

Handbook of Big Data Applications and Analytics

Theory and Practice

Gregor von Laszewski

laszewski@gmail.com

Copyright © 2017 Gregor von Laszewski

laszewski@gmail.com

[HTTPS://GITHUB.COM/CLOUDMESH/CLASSES](https://github.com/couldmesh/classes)

First printing, October 2017



Contents

I

Preface

1	Introduction	21
1.1	About	22
1.2	Citation	22
1.3	Contributors	22
1.4	Conventions	23
1.4.1	Videos	23
1.4.2	Slides	23
1.4.3	Images	23

II

Documenting Scientific Research

2	Documenting Scientific Research	27
2.1	Plagiarism	28
2.2	Writing a Scientific Article or Conference Paper	29
2.2.1	Professional Paper Format	29
2.2.2	Submission Requirements	30
2.2.3	Microsoft Word vs. L ^A T _E X	30
2.2.4	Working in a Team	30
2.2.5	Timemanagement	31
2.2.6	Paper and Report Checklist	31
2.2.7	Example Paper	33
2.2.8	Creating the PDF from LaTe _X on your Computer	34

2.2.9	Class Specific README.md	34
2.2.10	Exercise	35
3	Introduction to \LaTeX	37
3.1	Installation	38
3.1.1	Local Install	38
3.1.2	Online Services	39
3.2	Basic \LaTeX Elements	40
3.2.1	Characters	40
3.2.2	Highlighting Text	40
3.2.3	Sections	41
3.2.4	Empty Lines	41
3.2.5	Itemize	41
3.2.6	Enumerate	41
3.2.7	Descriptions	42
3.2.8	Images	42
3.2.9	Tables	42
3.2.10	Labels	43
3.2.11	Mathematics	43
3.3	Advanced topics	44
3.3.1	ACM and IEEE Proceedings Format	44
3.3.2	Generating and Managing Images	44
3.3.3	Slides	45
3.3.4	Useful Online Information about \LaTeX	45
3.3.5	\LaTeX vs. X	46
3.4	Editing	47
3.4.1	Emacs	47
3.4.2	Vi/Vim	47
3.4.3	TeXshop	48
3.4.4	LyX	48
3.4.5	WYSIWYG locally	48
3.4.6	Markdown and \LaTeX	49
3.4.7	Including RST into \LaTeX	49
3.4.8	pyCharm	50
3.4.9	MSWord	50
3.5	The \LaTeX Cycle	50
3.6	Tips	51
4	Managing Bibliographies	53
4.0.1	Integrating Bibliographies	54
4.1	Entry types	55
4.1.1	Source code References	55
4.1.2	Researching proper bibtex entries	57
4.1.3	Article in a conference proceedings	59
4.1.4	What are the differnt entry types and fields	62
4.1.5	InProceedings	62
4.1.6	TechReport	63
4.1.7	Article	64
4.1.8	Proceedings	64

4.1.9	Wikipedia Entry	65
4.1.10	Blogs	66
4.1.11	Web Page	66
4.1.12	Book	67
4.2	Integrating Bibtex entries into Other Systems	69
4.2.1	Bibtex import to MSWord	69
4.3	Other Reference Managers	70
4.3.1	Endnote	70
4.3.2	Mendeley	70
4.3.3	Zotero	70
5	Editors	71
5.1	Basic Emacs	72
6	Other Formats	75
6.1	reStructuredText	76
6.1.1	Links	76
6.1.2	Source	76
6.1.3	Sections	76
6.1.4	Listtable	76
6.1.5	Exceltable	77
6.1.6	Boxes	77
6.1.7	Sidebar directive	78
6.1.8	Sphinx Prompt	78
6.1.9	Programm examples	78
6.1.10	Hyperlinks	78
6.1.11	Todo	79
6.2	Markdown	80
6.3	Communicating Research in Other Ways	81
6.3.1	Blogs	81
6.3.2	Sphinx	81
6.3.3	Notebooks	82

III

Big Data Applications

6.4	Introduction	85
6.4.1	Course Motivation	86
6.5	Overview of Data Science	92
6.5.1	Data Science generics and Commercial Data Deluge	92
6.5.2	Data Deluge and Scientific Applications and Methodology	94
6.5.3	Clouds and Big Data Processing; Data Science Process and Analytics	95
6.5.4	Clouds	96
6.6	Health Informatics Case Study	98
6.6.1	X-Informatics Case Study: Health Informatics	98
6.7	e-Commerce and LifeStyle Case Study	103
6.7.1	Recommender Systems: Introduction	103
6.7.2	Recommender Systems: Examples and Algorithms	105

6.7.3	Item-based Collaborative Filtering and its Technologies	106
6.8	Physics Case Study	108
6.8.1	Looking for Higgs Particles, Bumps in Histograms, Experiments and Accelerators (Part 1)	108
6.8.2	Looking for Higgs Particles: Random Variables, Physics and Normal Distributions	110
6.8.3	Looking for Higgs Particles: Random Numbers, Distributions and Central Limit Theorem (Part 3)	111
6.9	Radar Case Study	113
6.9.1	Introduction	113
6.9.2	Remote Sensing	113
6.9.3	Ice Sheet Science	113
6.9.4	Global Climate Change	113
6.9.5	Radio Overview	114
6.9.6	Radio Informatics	114
6.10	Sensors Case Study	115
6.10.1	Internet of Things	115
6.10.2	Robotics and IOT Expectations	115
6.10.3	Industrial Internet of Things	115
6.10.4	Sensor Clouds	115
6.10.5	Earth/Environment/Polar Science data gathered by Sensors	116
6.10.6	Ubiquitous/Smart Cities	116
6.10.7	U-Korea (U=Ubiquitous)	116
6.10.8	Smart Grid	116
6.10.9	Resources	116
6.11	Sports Case Study	118
6.11.1	Sports Informatics I : Sabermetrics (Basic)	118
6.11.2	Sports Informatics II : Sabermetrics (Advanced)	119
6.11.3	PITCHf/X	119
6.11.4	Sports Informatics III : Other Sports	120
6.12	Big Data Use Cases Survey	122
6.12.1	Overview of NIST Big Data Public Working Group (NBD-PWG) Process and Results	122
6.12.2	51 Big Data Use Cases	124
6.12.3	Features of 51 Big Data Use Cases	126
6.13	Web Search and Text Mining	131
6.13.1	Web Search and Text Mining I	131
6.13.2	Web and Document/Text Search: The Problem	131
6.13.3	Information Retrieval leading to Web Search	132
6.13.4	History behind Web Search	132
6.13.5	Key Fundamental Principles behind Web Search	132
6.13.6	Information Retrieval (Web Search) Components	132
6.13.7	Search Engines	132
6.13.8	Boolean and Vector Space Models	132
6.13.9	Web crawling and Document Preparation	133
6.13.10	Indices	133
6.13.11	TF-IDF and Probabilistic Models	133
6.13.12	Resources	133
6.13.13	Web Search and Text Mining II	133
6.13.14	Data Analytics for Web Search	134

6.13.15	Link Structure Analysis including PageRank	134
6.13.16	Web Advertising and Search	134
6.13.17	Clustering and Topic Models	134
6.13.18	Resources	134
6.14	Technology Training - kNN & Clustering	135
6.14.1	Recommender Systems - K-Nearest Neighbors	135
6.14.2	Clustering and heuristic methods	135

IV

Development

6.15	Cloud Computing Technology for Big Data Applications & Analytics (will be updated)	141
6.15.1	Parallel Computing: Overview of Basic Principles with familiar Examples	141
6.15.2	Cloud Computing Technology Part I: Introduction	142
6.15.3	Cloud Computing Technology Part II: Software and Systems	144
6.15.4	Cloud Computing Technology Part III: Architectures, Applications and Systems	145
6.15.5	Cloud Computing Technology Part IV: Data Systems	147

V

Python

7	Introduction	151
7.1	Introduction to Python	152
7.1.1	About the Tutorial	152
7.1.2	Links	153
8	Install	155
8.1	Python Installation	156
8.1.1	Managing custom Python installs	156
8.1.2	Instalation without pyenv	159
8.1.3	Anaconda and Miniconda	159
8.2	Interactive Python	162
8.3	REPL (Read Eval Print Loop)	162
8.4	Python 3 Features in Python 2	163
9	Language	165
9.1	Statements and Strings	166
9.2	Variables	166
9.3	Data Types	166
9.3.1	Booleans	166
9.3.2	Numbers	167
9.4	Module Management	168
9.4.1	Import Statement	168
9.4.2	The from ... import Statement	168
9.5	Date Time in Python	168

9.6	Control Statements	170
9.6.1	Comparision	170
9.6.2	Iteration	170
9.7	Datatypes	171
9.7.1	Lists	171
9.7.2	Sets	172
9.7.3	Removal and Testing for Membership in Sets	173
9.7.4	Dictionaries	173
9.7.5	Dictionary Keys and Values	174
9.7.6	Counting with Dictionaries	174
9.8	Functions	175
9.9	Classes	176
9.10	Database Access	177
9.11	Modules	177
9.12	Installing Libraries	178
9.13	Using pip to Install Packages	179
9.14	GUI	179
9.14.1	GUIZero	179
9.14.2	Kivy	179
9.15	Formatting and Checking Python Code	179
9.16	Using autopep8	180
9.17	Further Learning	180
9.18	Writing Python 3 Compatible Code	180
9.19	Using Python on FutureSystems	180
9.20	Ecosystem	181
9.20.1	pypi	181
9.20.2	Alternative Installations	181
9.20.3	Autoenv: Directory-based Environments	182
9.21	Resources	183
9.22	Jupyter Notebook Tutorials	184
9.23	Exercises	184
10	Cloudmesh Command Shell	185
10.1	CMD5	186
10.1.1	Resources	186
10.1.2	Creating a Python Development Environment	186
10.1.3	Installation from source	186
10.1.4	Execution	187
10.1.5	Create your own Extension	187
10.1.6	Excercise	188
10.2	Code Management	189

11 Technology Overview	195
11.1 Workflow-Orchestration	195
11.1.1 ODE	195
11.1.2 ActiveBPEL	195
11.1.3 Airavata	196
11.1.4 Pegasus	196
11.1.5 Kepler	196
11.1.6 Swift	196
11.1.7 Taverna	197
11.1.8 Triana	197
11.1.9 Trident	197
11.1.10 BioKepler	197
11.1.11 Galaxy	198
11.1.12 Jupyter and IPython	198
11.1.13 Dryad	198
11.1.14 Naiad	199
11.1.15 Oozie	199
11.1.16 Tez	199
11.1.17 Google FlumeJava	200
11.1.18 Crunch	200
11.1.19 Cascading	200
11.1.20 Scalding	201
11.1.21 e-Science Central	201
11.1.22 Azure Data Factory	201
11.1.23 Google Cloud Dataflow	201
11.1.24 NiFi (NSA)	202
11.1.25 Jitterbit	202
11.1.26 Talend	202
11.1.27 Pentaho	202
11.1.28 Apatar	203
11.1.29 Docker Compose	203
11.1.30 KeystoneML	203
11.2 Application and Analytics	203
11.2.1 Mahout (383)	203
11.2.2 MLlib	204
11.2.3 MLbase	204
11.2.4 DataFu	204
11.2.5 R	204
11.2.6 pbdR	205
11.2.7 Bioconductor	205
11.2.8 ImageJ	205
11.2.9 OpenCV	205
11.2.10 Scalapack	205
11.2.11 PetSc	206
11.2.12 PLASMA MAGMA	206
11.2.13 Azure Machine Learning	206
11.2.14 Google Prediction API & Translation API	207
11.2.15 scikit-learn	207

11.2.16 PyBrain (529)	208
11.2.17 CompLearn	208
11.2.18 DAAL(Intel)	208
11.2.19 Caffe	208
11.2.20 Torch	209
11.2.21 Theano	209
11.2.22 DL4j	209
11.2.23 H2O	209
11.2.24 IBM Watson	210
11.2.25 Oracle PGX	210
11.2.26 GraphLab	210
11.2.27 GraphX	210
11.2.28 IBM System G	211
11.2.29 GraphBuilder(Intel)	211
11.2.30 TinkerPop	211
11.2.31 Parasol	212
11.2.32 Dream:Lab	212
11.2.33 Google Fusion Tables	212
11.2.34 CINET	212
11.2.35 NWB	213
11.2.36 Elasticsearch	213
11.2.37 Kibana	213
11.2.38 Logstash	214
11.2.39 Graylog	214
11.2.40 Splunk	214
11.2.41 Tableau	215
11.2.42 D3.js	215
11.2.43 three.js	215
11.2.44 Potree	215
11.2.45 DC.js	215
11.2.46 TensorFlow	216
11.2.47 CNTK	216
11.3 Application Hosting Frameworks	216
11.3.1 Google App Engine	216
11.3.2 AppScale	217
11.3.3 Red Hat OpenShift	217
11.3.4 Heroku	217
11.3.5 Aerobatic	217
11.3.6 AWS Elastic Beanstalk	218
11.3.7 Azure	218
11.3.8 Cloud Foundry	218
11.3.9 Pivotal	219
11.3.10 IBM BlueMix	219
11.3.11 (Ninefold)	219
11.3.12 Jelastic	219
11.3.13 Stackato	219
11.3.14 appfog	219
11.3.15 CloudBees	220
11.3.16 Engine Yard (573)	220
11.3.17 (CloudControl)	220
11.3.18 dotCloud (428)	220

11.3.19 Dokku	221
11.3.20 OSGi	221
11.3.21 HUBzero	221
11.3.22 OODT	221
11.3.23 Agave	221
11.3.24 Atmosphere	221
11.4 High level Programming	222
11.4.1 Kite	222
11.4.2 Hive	222
11.4.3 HCatalog	222
11.4.4 Tajo	222
11.4.5 Shark	223
11.4.6 Phoenix	223
11.4.7 Impala	224
11.4.8 MRQL	224
11.4.9 SAP HANA	224
11.4.10 HadoopDB	225
11.4.11 PolyBase	225
11.4.12 Pivotal HD/Hawq	225
11.4.13 Presto	226
11.4.14 Google Dremel	226
11.4.15 Google BigQuery	226
11.4.16 Amazon Redshift	226
11.4.17 Drill	227
11.4.18 Kyoto Cabinet	227
11.4.19 Pig	227
11.4.20 Sawzall	227
11.4.21 Google Cloud DataFlow	228
11.4.22 Summingbird	228
11.4.23 Lumberyard	228
11.5 Streams	229
11.5.1 Storm	229
11.5.2 S4	229
11.5.3 Samza	229
11.5.4 Granules	230
11.5.5 Neptune	230
11.5.6 Google MillWheel	230
11.5.7 Amazon Kinesis	230
11.5.8 LinkedIn	231
11.5.9 Twitter Heron	231
11.5.10 Databus	231
11.5.11 Facebook Puma/Ptail/Scribe/ODS	231
11.5.12 Azure Stream Analytics	232
11.5.13 Floe	232
11.5.14 Spark Streaming (563)	232
11.5.15 Flink Streaming	232
11.5.16 DataTurbine	232
11.6 Basic Programming Model and Runtime, SPMD, MapReduce	233
11.6.1 Hadoop	233
11.6.2 Spark (541)	233

11.6.3	Twister	233
11.6.4	MR-MPI	234
11.6.5	Stratosphere (Apache Flink)	234
11.6.6	Reef	234
11.6.7	Disco	234
11.6.8	Hama	235
11.6.9	Giraph	235
11.6.10	Pregel	235
11.6.11	Pegasus	235
11.6.12	Ligra	235
11.6.13	GraphChi	235
11.6.14	Galois	236
11.6.15	Medusa-GPU	236
11.6.16	MapGraph	236
11.6.17	Totem	236
11.7	Inter process communication Collectives	236
11.7.1	point-to-point	236
11.7.2	(a) publish-subscribe: MPI	236
11.7.3	(b) publish-subscribe: Big Data	236
11.7.4	HPX-5	237
11.7.5	Argo BEAST HPX-5 BEAST PULSAR	237
11.7.6	Harp	237
11.7.7	Netty	237
11.7.8	ZeroMQ	238
11.7.9	ActiveMQ	238
11.7.10	RabbitMQ	238
11.7.11	NaradaBrokering	239
11.7.12	QPid	239
11.7.13	Kafka	239
11.7.14	Kestrel	239
11.7.15	JMS	240
11.7.16	AMQP	240
11.7.17	Stomp	240
11.7.18	MQTT	240
11.7.19	Marionette Collective	241
11.7.20	Public Cloud: Amazon SNS	241
11.7.21	Lambda	241
11.7.22	Google Pub Sub	242
11.7.23	Azure Queues	242
11.7.24	Event Hubs	242
11.8	In-memory databases/caches	242
11.8.1	Gora (general object from NoSQL)	242
11.8.2	Memcached	243
11.8.3	Redis	243
11.8.4	LMDB (key value)	244
11.8.5	Hazelcast	244
11.8.6	Ehcache	244
11.8.7	Infinispan	244
11.8.8	VoltDB	245
11.8.9	H-Store	245

11.9 Object-relational mapping	245
11.9.1 Hibernate	245
11.9.2 OpenJPA	246
11.9.3 EclipseLink	246
11.9.4 DataNucleus	246
11.9.5 ODBC/JDBC	246
11.10 Extraction Tools	247
11.10.1 UIMA	247
11.10.2 Tika	247
11.11 SQL(NewSQL)	247
11.11.1 Oracle	247
11.11.2 DB2	247
11.11.3 SQL Server	248
11.11.4 SQLite	248
11.11.5 MySQL	248
11.11.6 PostgreSQL	249
11.11.7 CUBRID	249
11.11.8 Galera Cluster	249
11.11.9 SciDB	250
11.11.10 Rasdaman	250
11.11.11 Apache Derby	250
11.11.12 Pivotal Greenplum	250
11.11.13 Google Cloud SQL	251
11.11.14 Azure SQL	251
11.11.15 Amazon RDS	251
11.11.16 Google F1	251
11.11.17 IBM dashDB	252
11.11.18 N1QL	252
11.11.19 BlinkDB	252
11.11.20 Spark SQL	252
11.12 NoSQL	252
11.12.1 Lucene	252
11.12.2 Solr	252
11.12.3 Solandra	252
11.12.4 Voldemort	253
11.12.5 Riak	253
11.12.6 ZHT	253
11.12.7 Berkeley DB	254
11.12.8 Kyoto/Tokyo Cabinet	254
11.12.9 Tycoon	254
11.12.10 Tyrant	254
11.12.11 MongoDB	255
11.12.12 Espresso	255
11.12.13 CouchDB	255
11.12.14 Couchbase Server	256
11.12.15 IBM Cloudant	256
11.12.16 Pivotal Gemfire (6)	256
11.12.17 HBase	257
11.12.18 Google Bigtable	257
11.12.19 LevelDB	257

11.12.20	Megastore and Spanner	257
11.12.21	Accumulo	258
11.12.22	Cassandra	258
11.12.23	RYA	258
11.12.24	sqrrl	259
11.12.25	Neo4J	259
11.12.26	graphdb	259
11.12.27	arcdata	259
11.12.28	AllegroGraph	260
11.12.29	Blazegraph	260
11.12.30	Facebook Tao	260
11.12.31	Titan:db	260
11.12.32	lena	261
11.12.33	Sesame	261
11.12.34	Public Cloud: Azure Table	261
11.12.35	Amazon Dynamo	262
11.12.36	Google DataStore	262
11.13	File management	262
11.13.1	iRODS	262
11.13.2	CDF	263
11.13.3	HDF	263
11.13.4	OPeNDAP	263
11.13.5	FITS	263
11.13.6	RCFfile	264
11.13.7	ORC	264
11.13.8	Parquet	264
11.14	Data Transport	264
11.14.1	BitTorrent	264
11.14.2	HTTP	265
11.14.3	FTP	265
11.14.4	SSH	265
11.14.5	Globus Online (GridFTP)	265
11.14.6	Flume	265
11.14.7	Sqoop	266
11.14.8	Pivotal GPLOAD/GPFDIST	266
11.15	Cluster Resource Management	266
11.15.1	Mesos	266
11.15.2	Yarn	267
11.15.3	Helix	267
11.15.4	Llama	267
11.15.5	Google Omega	267
11.15.6	Facebook Corona	267
11.15.7	Celery	268
11.15.8	HTCondor	268
11.15.9	SGE	268
11.15.10	OpenPBS	269
11.15.11	Moab	269
11.15.12	Slurm (551)	269
11.15.13	Torque	270
11.15.14	Globus Tools	270

11.15.15	Pilot Jobs	270
11.16	File systems	270
11.16.1	HDFS	270
11.16.2	Swift	270
11.16.3	Haystack	270
11.16.4	f4	271
11.16.5	Cinder	271
11.16.6	Ceph	271
11.16.7	FUSE	272
11.16.8	Gluster	272
11.16.9	Lustre	272
11.16.10	GPFS	272
11.16.11	IBM Spectrum Scale	272
11.16.12	GFFS	273
11.16.13	Public Cloud: Amazon S3	273
11.16.14	Azure Blob	273
11.16.15	Google Cloud Storage	274
11.17	Interoperability	274
11.17.1	cloudmesh	274
11.17.2	Libvirt	274
11.17.3	Libcloud	274
11.17.4	JClouds	275
11.17.5	TOSCA	275
11.17.6	OCCI	275
11.17.7	CDMI	276
11.17.8	Whirr	276
11.17.9	Saga	276
11.17.10	Genesis	277
11.18	DevOps	277
11.18.1	Docker (Machine, Swarm)	277
11.18.2	Puppet	277
11.18.3	Chef	278
11.18.4	Ansible	278
11.18.5	SaltStack	278
11.18.6	Boto	278
11.18.7	Cobbler	279
11.18.8	Xcat	279
11.18.9	Razor	279
11.18.10	Quju	279
11.18.11	Foreman	280
11.18.12	OpenStack Heat	280
11.18.13	Sahara	280
11.18.14	Rocks	280
11.18.15	Cisco Intelligent Automation for Cloud	280
11.18.16	Ubuntu MaaS	281
11.18.17	Facebook Tupperware	281
11.18.18	AWS OpsWorks	281
11.18.19	OpenStack Ironic	281
11.18.20	Google Kubernetes	281
11.18.21	Buildstep	282

11.18.22	Gitreceive	282
11.18.23	OpenTOSCA	282
11.18.24	Winery	282
11.18.25	CloudML	283
11.18.26	Blueprints	283
11.18.27	Terraform	283
11.18.28	DevOpSlang	284
11.18.29	Any2Api	284
11.19	IaaS Management from HPC to hypervisors	284
11.19.1	Xen	284
11.19.2	KVM	285
11.19.3	QEMU	285
11.19.4	Hyper-V	285
11.19.5	VirtualBox	286
11.19.6	OpenVZ	286
11.19.7	LXC	286
11.19.8	Linux-Vserver	286
11.19.9	OpenStack	287
11.19.10	OpenNebula	287
11.19.11	Eucalyptus	287
11.19.12	Nimbus	287
11.19.13	CloudStack	288
11.19.14	CoreOS	288
11.19.15	kt	288
11.19.16	Mware ESXi	289
11.19.17	Sphere and vCloud	289
11.19.18	Amazon	289
11.19.19	Azure	290
11.19.20	Google and other public Clouds	290
11.19.21	Networking: Google Cloud DNS	290
11.19.22	Amazon Route 53	290
11.20	Cross-Cutting Functions	290
11.20.1	Ambari	291
11.20.2	Ganglia	291
11.20.3	Nagios (413)	291
11.20.4	Inca	291
11.20.5	InCommon	292
11.20.6	Eduroam (168)	292
11.20.7	OpenStack Keystone	292
11.20.8	LDAP	293
11.20.9	Sentry	293
11.20.10	Sqrrl	293
11.20.11	OpenID	293
11.20.12	AML OAuth	294
11.20.13	Google Chubby	294
11.20.14	Zookeeper	294
11.20.15	Giraffe	294
11.20.16	Groups	295
11.21	Message and Data Protocols	295
11.21.1	Avro	295

11.21.2 Thrift	295
11.21.3 Protobuf	295
11.22 Technologies To Be Integrated	295
11.22.1 Snort	295
11.22.2 Fiddler	296
11.22.3 Zeppelin	296
11.22.4 Open MPI	296
11.22.5 Apache Tomcat	296
11.22.6 Apache Beam	297
11.22.7 Cloudability	297
11.22.8 CUDA	297
11.22.9 Blaze	297
11.22.10 CDAP	298
11.22.11 Apache Arrow	298
11.22.12 OpenRefine	298
11.22.13 Apache OODT	298
11.22.14 Omid	299
11.22.15 Askalon	299
11.22.16 Apache Ant	299
11.22.17 XD	300
11.22.18 Wink	300
11.22.19 Apache Apex	300
11.22.20 Apache Knox	300
11.22.21 Apache Apex	301
11.22.22 Robot Operating System (ROS)	301
11.22.23 Apache Flex	301
11.22.24 Apache Ranger	302
11.22.25 Google Cloud Machine Learning	302
11.22.26 Karajan	302
11.23 Excercise	303

Bibliography	307
Index	343



Preface

1	Introduction	21
1.1	About	
1.2	Citation	
1.3	Contributors	
1.4	Conventions	



1. Introduction

 chapter/preface/about.tex

1.1 About

The document is based on selected material published at the following Web page

- <https://cloudmesh.github.io/classes/>

It is part of a class taught at Indiana University. The class communication takes place at:

- <https://piazza.com/class/ix39m27czn5uw>

The PDF version will be made in future available at

- <https://github.com/laszewski/laszewski.github.io/raw/master/papers/vonLaszewski-bigdata.pdf>

This PDF document will be updated based on feedback from the students and once we have now material available. For a more complete set of information we recommend the students to visit the Web page.

1.2 Citation

The bibtex entry for this document is

```
@TechReport{las17handbook,
  author = {Gregor von Laszewski},
  title = {Handbook of Big Data Applications and Analytics},
  institution = {Indiana University},
  year = {2017},
  OPTtype = {Draft},
  address = {Smith Research Center, Bloomington, IN 47408},
  month = dec,
  url={https://github.com/laszewski/laszewski.github.io/raw/master/papers/vonLaszewski-bigdata.pdf}
```

1.3 Contributors

We like to acknowledge the following contributors that helped on this document. Please notify us with your name and a brief command on what you contributed:

Descriptions provided in Section ?? were contributed by the following people that are either listed by full name or their github.com id:

Abhijit Thakre, Abhishek Gupta, Abhishek Naik, Ajit Balaga, Anurag Kumar Jain, Avadhoot Agasti, Badi' Abdul-Wahid, Cmbays, DIKSHA, Dimitar Nikolov, Govind, Govind Mishra, Grace Li, Gregor von Laszewski, Harshit Krishnakumar, Hyungro Lee, Jerome Mitchell, Jimmy Ardiansyah, Jon, Jon Montgomery, Jordan Simmons, Juliette Zerick, Karthik, Kumar Satyam, Mark McCombe, Matthew Lawson, Methkupalli Vasanth, Miao Jiang, Miao Zhang, Milind Suryawanshi, MilindSuryawanshi, Nandita Sathe, Naveen, Niteesh01, Piyush Rai, Piyush Shinde, Prashanth, Pratik Jain, Rahul Raghata, Rahul Singh, Ribka Rafael, Ronak Parekh, Saber Sheybani,

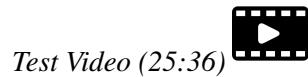
Sabyasachi Roy Choudhury, Sagar Vora, Sahiti Korrapati, Scott McClary, Sean Shiverick, SilviaKarim14, Sivaprasad Sushmita, Snehal Chemburkar, Sowmya Ravi, Srikanth Ramanam, Sunanda Unni, SushmitaSivaprasad, Tony Liu, Vasanth Methkupalli, Veera Marni, Vibhatha Abeykoon, Vibhatha Lakmal Abeykoon, Vishwanath Kodre, William H Knapp III, acastrob, ak.15, alyez, anveling, argetlam115, athakre, bhavesh37, coultes, cglmoocs, elenadesigner, eunosm3, harkrish1, jemitchell, justbbusy, jzerick, kartanba, karthick, karthick venkatesan, karthik-anba, kpvenkat, ksrivatsav, lmundia, miaozhan, michaelsmith1983, mmcccombe, nsathe, piyurai, pratik11jain, ronak1182, sabyasachi087, shah0112, sriramsitharaman, suunni, tifabi, tonythomascn, vasanth, vibhatha, vkodre, vlabeyko, xl41, yatinsharma7

John Doe He contributed to none of teh sections as this is just an example.

1.4 Conventions

1.4.1 Videos

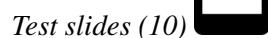
Videos to the class are refered to with embeded links into the PDF document as follows:



An index will also be available in the index page that lists on which page the video has been added.

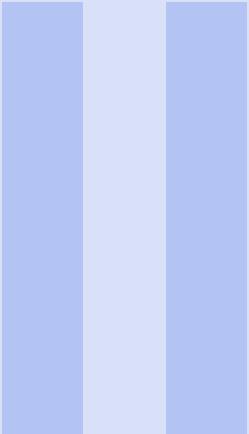
1.4.2 Slides

Sides



1.4.3 Images

The video icon was copied from <http://www.freeiconspng.com/img/8039>.



Documenting Scientific Research

2	Documenting Scientific Research	27
2.1	Plagiarism	
2.2	Writing a Scientific Article or Conference Paper	
3	Introduction to \LaTeX	37
3.1	Installation	
3.2	Basic \LaTeX Elements	
3.3	Advanced topics	
3.4	Editing	
3.5	The \LaTeX Cycle	
3.6	Tips	
4	Managing Bibliographies	53
4.1	Entry types	
4.2	Integrating Bibtex entries into Other Systems	
4.3	Other Reference Managers	
5	Editors	71
5.1	Basic Emacs	
6	Other Formats	75
6.1	reStructuredText	
6.2	Markdown	
6.3	Communicating Research in Other Ways	



2. Documenting Scientific Research

2.1 Plagiarism

In academic life it is important to understand and avoid plagiarism. Organizations and universities will have policies in place to address plagiarism. An example is provided for Indiana University [631]. We quote:

“Honesty requires that any ideas or materials taken from another source for either written or oral use must be fully acknowledged. Offering the work of someone else as one’s own is plagiarism. The language or ideas thus taken from another may range from isolated formulas, sentences, or paragraphs to entire articles copied from books, periodicals, speeches, or the writings of other students. The offering of materials assembled or collected by others in the form of projects or collections without acknowledgment also is considered plagiarism. Any student who fails to give credit for ideas or materials taken from another source is guilty of plagiarism.

(Faculty Council, May 2, 1961; University Faculty Council, March 11, 1975;
Board of Trustees, July 11, 1975”)

Faculty members at Universitys are also bound by policies that mandate reporting. At Indiana University the following policy applies (for a complete policy see the Web page):

“Should the faculty member detect signs of plagiarism or cheating, it is his or her most serious obligation to investigate these thoroughly, to take appropriate action with respect to the grades of students, and *in any event* to report the matter to the Dean for Student Services [or equivalent administrator]. The necessity to report every case of cheating, whether or not further action is desirable, arises particularly because of the possibility that this is not the student’s first offense, or that other offenses may follow it. Equity also demands that a uniform reporting practice be enforced; otherwise, some students will be penalized while others guilty of the same actions will go free.

(Faculty Council, May 2, 1961”)

Naturally if a student has any questions about understanding plagiarism the University can provide assistance. If a student is in doubt and asks for help this is not considered at that time plagiarism. As you can see from the previous policies, the faculty do not have any choice but reporting it to the university administration. Thus you must not hold them personally responsible as this is part of the tasks they are required to do if they like it or not. Instead, it is **the responsibility of the authors of the document** to assure no plagiarism occurs. If you are a student of a class that writes a paper or project report this naturally also all applies to you. In addition, if you work in a team you need to assure the entire team addresses plagiarism appropriately.

F chapter/lesson/doc/report-book.tex

2.2 Writing a Scientific Article or Conference Paper

An important part of any scientific research is to document it. This is often done through scientific conferences or journal articles. Hence it is important to learn how to prepare and submit such papers. Most conferences accept typically the papers in PDF format but require the papers to be prepared on MSWord or in LaTeX. While working with many students in the past we noticed however that those students using Word often spend unnecessarily countless hours on trying to make their papers beautiful while actually violating the template provided by the conference. Furthermore, we noticed that the same students had issues with bibliography management. Instead of Word helping the student it provided the illusion to be easier than LaTeX but when adding up the time spent on the paper we found that LaTeX actually saved time. This has been especially true with the advent of collaborative editing services such as sharelatex [537] and overleaf [458].

In this section we provide you with a professional template that is used for either system based on the ACM standard that you can use to write papers. Naturally this will be extremely useful if the quality of your research is strong enough to be submitted to a conference. We structure this section as follows. Although we do not recommend that you use MSWord for your editing of a scientific paper, we have included a short section about it and outline some of its pitfalls that initially you may not think is problematic, but has proven to be an issue with students. Next we will focus on introducing you to LaTeX and showcasing you the advantages and disadvantages. We will dedicate an entire section on bibliography management and teach you how to use jabref which clearly has advantages for us.

Having a uniform report format not only helps the students but also allows the comparison of paper length and effort as part of teaching a course. We have added an entire section to this chapter that discusses how we can manage a *Class Proceedings* form papers that are contributed by teams in the class.

2.2.1 Professional Paper Format

The report format we suggest here is based on the standard ACM proceedings format. It is of very high quality and can be adapted for your own activities. Moreover, it is possible to use most of the text to adapt to other formats in case the conference you intend to submit your paper to has a different format. The ACM format is always a good start.

Important is that you do not need to change the template but you can change some parameters in case you are not submitting the paper to a conference but use it for class papers. Certainly you should not change the spacing or the layout and instead focus on writing content. As for bibliography management we recommend you use jabref which we will introduce in Section ??.

We recommend that you carefully study the requirements for the report format. We would not want that your paper gets rejected by a journal, conference or the class just because you try to modify the format or do not follow the established publication guidelines.

The template we are providing is available from:

- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report>

Convenient compressed files are available at

- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report>.

`tar.gz`

- <https://github.com/cloudmesh/classes/tree/master/docs/source/format/report.zip>

You will find in it a modified ACM proceedings templates for Word and for LaTeX that has an identification box removed on the lower left hand side of the first page. This is done for classes so that you have more space to write. In case you must submit to a conference you can use the original ACM template. This template can be found at

2.2.2 Submission Requirements

Although the initial requirement for some conferences or journals is the document PDF, in many cases you must be prepared to provide the source when submitting to the conference. This includes the submission of the original images in an images folder. You may be asked to package the document into a folder with all of its sources and submit to the conference for professional publication.

2.2.3 Microsoft Word vs. \LaTeX

Microsoft word will provide you with the initial impression that you will save lots of time writing in it while you see the layout of the document. This will be initially true, but once you progress to the more challenging parts and later pages such as image management and bibliography management you will see some issues. These include that figure placement in Word needs to be done just right in order for images to be where they need. We have seen students spending hours with the placement of figures in a paper but when they did additional changes the images jumped around and were not at the place where the students expected them to be. So if you work with images, make sure you understand how to place them. Also always use relative caption counters so that if an image gets placed elsewhere the counter stays consistent. So never use just the number, but a reference to the figure when referring to it. Recently a new bibliography management system was added to Word. However, however it is not well documented and the references are placed in the system bibliography rather than a local managed bibliography. This may have severe consequences when working with many authors on a paper. The same is true when using Endnote. We have heard in many occasions that the combination of endnote and Word destroyed documents. You certainly do not want that to happen the day before your deadline. Also in classes we observed that those using LaTeX deliver better structured and written papers as the focus is on text and not beautiful layout.

For all these reasons we do not recommend that you use Word.

In LaTeX where we have an easier time with this as we can just ignore all of these issues due to relative good image placement and excellent support for academic reference management. Hence, it is in your best interest to use LaTeX. The information we provide here will make it easy for you to get started and write a paper in no time as it is just like filling out a form.

2.2.4 Working in a Team

Today research is done in potentially large research teams. This also includes the writing of a document. There are multiple ways this is done these days and depends on the system you chose.

In MSWord you can use skydrive, while for LaTeX you can use sharelatex and overleaf. However, in many cases the use of github is possible as the same groups that develop the code are also familiar with github. Thus we provide you here also with the introduction on how to write a document in

github while group members can contribute.

Here are the options:

LaTeX and git: This option will likely save you time as you can use jabref also for managing collaborative bibliographies and

sharelatex: an online tool to write latex documents

overleaf: an online tool to write latex documents

MS onedrive: It allows you to edit a word document in collaboration. We recommend that you use a local installed version of Word and do the editing with that, rather than using the online version. The online editor has some bugs. See also (untested): <http://www.paulkiddie.com/2009/07/jabref-exports-to-word-2007-xml/>, <http://usefulcodes.blogspot.com/2015/01/using-jabref-to-import-bib-to-microsoft.html>

Google Drive: google drive could be used to collaborate on text that is than pasted into document. However it is just a starting point as it does not support typically the format required by the publisher. Hence at one point you need to switch to one of the other systems.

2.2.5 Timemanagement

Obviously writing a paper takes time and you need to carefully make sure you devote enough time to it. The important part is that the paper should not be an after thought but should be the initial activity to conduct and execute your research. Remember that

1. It takes time to read the information
2. It takes time understand the information
3. It takes time to do the research

For deadlines the following will get you in trouble:

1. *There are still 10 weeks left till the deadline, so let me start in 4 week* Procrastination is your worst enemy.
2. If you work in a team that has time management issues address them immediately
3. Do not underestimate the time it takes to prepare the final submission into the submission system. Prepare automated scripts that can deliver the package for submission in minutes rather than hours by hand.

2.2.6 Paper and Report Checklist

In this section we summarize a number of checks that you may perform to make sure your paper is properly formatted and in excellent shape. Naturally this list is just a partial list and if you find things we should add here, let us know.

- Have you written the report in the specified format?
- Have you included an acknowledgement section?
- Have you included the paper in the submission system (In our case git). This includes all images, bibliography files and other material that is needed to build the paper from scratch?
- Have you added the bibliography file that you managed (In our case jabref to make it simple for you)?
- Have you specified proper identification in the submission system for your submission. This is typically a form or ASCII text that needs to be filled out and follows a very particular format. In our case it is a README.md file that includes a homework ID, names of the authors, and e-mails)?
- In case you used word have you also provided the jabref?

- In case of a class and if you do a multi-author paper, have you added a work-breakdown section in the appendix describing who did what in the paper?
- IN case you have an appenix it is included fater the bibliography
- Have you spellchecked the paper?
- Have you grammar chacked the paper?
- Are you using **a** and **the** properly?
- Have you made sure you do not plagiarize?
- Is the title properly capitalized?
- Have you not used phrases such as shown in the Figure below, but instead used as shown in Figure 3 when referring to the 3rd figure? Numbers in LaTeX are done with `\label{}` after captions and `\ref{}` in the text (See examples in the LaTeX section). In word you must use relative numbering.
- Have you capitalized “Figure 3”, “Table 1”?
- Have you removed any figure that is not referred explicitly in the text. E. ech figure needs a text such as *As shown in Figure ..* or similar.
- Are the figure captions bellow the figures and not on top. (Do not include the titles of the figures in the figure itself but instead use the caption or that information?)
- When using tables have you put the table caption above the table?
- When using image have you put the table caption bellow the image?
- Make the figures large enough so we can read the details. If needed make the figure over two columns?
- Do not worry about the figure placement if they are at a different location than you think. Figures are allowed to float. In many submissions you may have to place all figures at the end of the paper so you can focus on content rather than placing figures. In addition it will help you to refer to the figures by index.
- Are all figures and tables at the end?
- In case you copied a figure from another paper you need to ask for copyright permission. In case of a class paper you **must** include a reference to the original at the end of the figure caption.
- Do not use the word “I” instead use “we” even if you are the sole author.
- Do not use the phrase “In this paper/report we show” instead use “We show”. It is not important if this is a paper or a report and does not need to be mentioned.
- Do not artificially inflate your paper if you are bellow the page limit and have nothing to say anymore.
- If your paper limit is 12 pages but you want to hand in 120 pages, please check first ;-) If your page limoit is 2 pages but you hand in 4 thats is no issue.
- Do not use the characters & # % _ in the paper if you use LaTeX. If you use them you probably need a bakslash in front of them.
- Latex uses double single open quotes and double single closed quotes for quotes. Have you made sure you replaced them?
When using quotes in LaTeX, do not use the double quote but instead use two single quotes such as ‘‘This is a quote’’. THis will place the proper quotes in the text. To only use the quotes when you literraly quote from other papers. Never use a quote to emphasize a thext. For that you use `\emph{this is emphasize}` resulting in *This is emphasized*.
- If you want to say and do not use & but use the word *and*. If you need tou use it be reminded to write it as \&
- Pasting and copying from the Web often results in non ascii characters to be used in your text, please remove them and replace accordingly. This includes some form of – that you may see showing up as fi in pdf

- Is your Abstract not a proposal? Abstracts are no proposals, e.g. This paper intends to show If the paper intends to show you are still in the draft phase of the paper. However, if you say We show ... That would be good. Let us just assume you intended to show something but did not achieve then you can say We intended to show this but we showed it was not possible. As you can see not only the intention is communicated, but the result. If you just focus on the intent that's just a proposal and is not a proper abstract.
- Are you not using the word paper in your writeup? Abstracts and the entire paper should not have the word paper in it.
- If your paper is an introduction or overview paper, please do not assume the reader to be an expert. Provide enough material for the paper to be useful for an introduction into the topic.
- Are your references correct? References to a paper are no afterthought, they should be properly cited. Use Jabref and make sure the citation type of the reference is correct and fill out as many fields as you can. Some journals and conferences have for example special requirements that go beyond the requirements of for example Jabref. One example is that many conferences require you that when you cite papers from another conference to augment the conference not only with the location where the conference took place, but also with the dates the conference took place. Unfortunately, this information is often only available through additional Google queries and many reference entries you find in the internet do not have this information readily available.

In case of a class

- Check in your current work of the paper on a weekly basis to show consistent progress.
- Please use the dedicated report format for class. It may not be the ACM or IEEE format, but may have some additions that make management of bibliographies easier. Do follow our instructions for bibliographies.

In case you are allowed to use Word in class, such as the one we teach at IU, the following applies in addition:

- Are you managing your references in Jabref and Endnote (we need both)
- Are you using the right template we have a special 2 column template for the class that is a modified version from the 2 column ACM template
- Are you using built-in numbered section management? MSWord has Sections that must be used
- Are you using real bulleted lists in Word and not just a "*" or a "-"?
- Have you carelessly pasted and copied into the document without using proper formats. E.g. in MSWord this is a problem. You need to fix the format and use the built-in format. Note that if you paste wrong you affect the format styles.
- Have you created not only a .docx document but also the PDF.
- Make sure you use .docx and not .doc

If you observe something missing let us know.

2.2.7 Example Paper

An example report in PDF format is available:

- report.pdf

2.2.8 Creating the PDF from LaTeX on your Computer

Latex can be easily installed on any computer as long as you have enough space. Furthermore if your machine can execute the make command we have provided in the standard report format a simple Makefile that allows you to do editing with immediate preview as documented in the LaTeX lesson.

2.2.9 Class Specific README.md

For the class we will manage all papers via github.com. You will be added to our github at

- <https://github.com/bigdata-i523>

and assigned an hid (homework index directory) directory with a unique hid number for you. In addition, once you decide for a project, you will also get a project id (pid) and a directory in which you place the projects. Projects must not be placed in hid directories as they are treated differently and a class proceedings is automatically created based on your submission.

As part of the hid directory, you will need to create a README.md file in it, that **must** follow a specific format. The good news is that we have developed an easy template that with common sense you can modify easily. The template is located at

- <https://raw.githubusercontent.com/bigdata-i523/sample-hid000/master/README.md>

As the format may have been updated over time it does not hurt to revisit it and compare with your README.md and make corrections. It is important that you follow the format and not eliminate the lines with the three quotes. The text in the quotes is actually yaml. yaml is a data format the any data scientist must know. If you do not, you can look it up. However, if you follow our rules you should be good. If you find a rule missing for our purpose, let us know. We like to keep it simple and want you to fill out the *template* with your information.

Simple rules:

- replace the hid number with your hid number.
- naturally if you see sample- in the directory name you need to delete that as your directory name does not have sample- in it.
- do not ignore where the author is to be placed, it is in a list starting with a -
- there is always a space after a -
- do not introduce empty lines
- do not use TAB and make sure your editor does not bay accident automatically creates tabs. This is probably the most frequent error we see.
- do not use any : & _ in the attribute text including titles
- an object defined in the README.md must have on a single type field. for example in the project section. Make sure you select only one type and delete the other
- in case you have long paragraphs you can use the > after the abstract
- Once you understood how the README.md works, please delete the comment section.
- Add a chapter topic that your paper belongs to

2.2.10 Exercise

Report.1: Install latex and jabref on your system

Report.2: Check out the report example directory. Create a PDF and view it. Modify and recompile.

Report.4: Learn about the different bibliographic entry formats in bibtex

Report.5: What is an article in a magazine? Is it really an Article or a Misc?

Report.6: What is an InProceedings and how does it differ from Conference?

Report.7: What is a Misc?

Report.8: Why are spaces, underscores in directory names problematic and why should you avoid using them for your projects

Report.9: Write an objective report about the advantages and disadvantages of programs to write reports.

Report.10: Why is it advantageous that directories have no underscore or space in the name?



3. Introduction to \LaTeX

 chapter/lesson/doc/latex.tex

Mastering a text processing system is an essential part of a researcher's life. Not knowing how to use a text processing system can slow down the productivity of research drastically.

3.1 Installation

\LaTeX is available on all modern computer systems. A very good installation for OSX is available at:

- <https://tug.org/mactex/>

However, if you have older versions on your systems you may have to first completely uninstall them.

3.1.1 Local Install

Installing \LaTeX is trivial, and is documented on the internet very well. However, it requires sufficient space and time as it is a large environment. A system such as TeX Live takes in full install about 5.5 GB. In addition to \LaTeX we recommend that you install jabref and use it for bibliography management.

Thus you will have the most of them on your system.

- pdflatex: the latex program producing pdf
- bibtex: to create bibliographies
- jabref: GUI application to bibtex files (<http://www.jabref.org/>)

Make sure you check that these programs are there, for example with the Linux commands:

```
which pdflatex
which bibtex
which jabref (on OSX you may have an icon for it)
```

If these commands are missing, please install them. For the newest documentation on installation of \LaTeX we recommend you look up the installation for your specific OS.

Install on Ubuntu 16.04

The easiest way to install it on ubuntu is to use the terminal and type in (make sure you have enough space):

```
sudo apt-get install texlive-full
```

One of the best editors for \LaTeX is emacs as you can also do bibliography management with it and not just \LaTeX . However, other editors are available including:

- Kile, TeXworks, JLatexEditor, Gedit \LaTeX Plugin, TeXMaker

Please look up how to install them if you like to use them. TeXMaker is popular, However I find the combination of emacs and latexmk superior. TeXmaker is installed with:

```
sudo apt-get install texmaker
```

Other installations:

- kile is installed by default

- <https://www.tug.org/texworks/> (Works on ubuntu, Windows, OSX)

LaTeX for OSX

- <https://www.latex-project.org/get/>

LaTeX for Windows

- <https://www.latex-project.org/get/>

3.1.2 Online Services

Sharelatex

ShareLaTeX is an online, collaborative LaTeX editor that makes the creation, preview, and sharing of LaTeX documents easy through a web-based interface. Those that like to use latex, but do not have it installed on their computers may want to look at the following video:

Video: <https://youtu.be/PfhSOjuQk8Y>

Video with cc: <https://www.youtube.com/watch?v=8IDCGTFXoBs>

ShareLaTeX not only allows you to edit online, but allows you to share your documents in a group of up to three. Licenses are available if you need more than three people in a team.

IU Licensed ShareLaTeX

At IU we has a license for the ShareLaTeX service available to School of Informatics and Computing and Engeneering students, faculty, and staff only on the Bloomington campus.

You can create a free ShareLaTeX account but the free accounts have limitations. Adding your account to the IU license will give you access to advanced features, including unlimited sharing. It will also allow GitHub integration. THis however only works with the commercial github.com and not the IU Enterprise GitHub at github.iu.edu. As we require in our courses github.com you will be able to use it.

Please note that this license is only available to School of Informatics and Computing students, faculty, and staff on the Bloomington campus. Students must be enrolled in one of the SoIC degree programs on the Bloomington campus to be eligible. Students in other degree programs (even those taking SoIC classes) are not eligible.

If you want to use this service, please do and be aware of the following:

1. Go to the ShareLaTeX site and register. Please note that you **must** use either an @indiana.edu or @iu.edu email address when you register. If you use any other email address, we will not be able to add you to our site license. You are also required to use your IU passphrase as your ShareLaTeX password. Once you have registered, send an email to soichelp@indiana.edu asking to have your sharelatex account added to the IU license.
2. In your request, you must include the following: The IU email address you used when you registered (which must be in either the @indiana.edu or @iu.edu domain) A statement indicating that you understand that the ShareLaTeX service cannot be used for any sensitive data
3. Note that the ShareLaTeX service is **not** qualified for any sensitive data. This includes all data in the Critical, Restricted, and University-Internal categories as defined in the Data Classifications Page.

Overleaf

Overleaf.com is a collaborative latex editor. In its free version it has a very limited disk space. However it comes with a Rich text mode that allows you to edit the document in a preview mode. The free templates provided do not include ACM template, but you are allowed to use the OSA template.

Features of overleaf are documented at: <https://www.overleaf.com/benefits>

Paperia

We do not know where this service is located. However it offers similar services as ShareLatex and Overleaf.

- <https://papeeria.com/>

3.2 Basic LaTeX Elements

Often researchers may be initially overwhelmed with all the features that \LaTeX provides. However, it is much simpler than you initially believe. In Chapter ?? we introduced you towards using an article template. As a template is provided you can just look at the elements in that article and modify or copy them while adapting the content. Thus, it is more like filling out a form. You do not have to learn much and you can learn as you go. We are providing in this chapter some basic \LaTeX elements that will help you getting started quickly while serving you as a reminder what how to do certain things in \LaTeX .

3.2.1 Characters

\LaTeX is a command language and as such uses some special characters as part of the language. Thus if you want to use these characters either in your text or bibliography you need to be especially careful about. These characters include % \$ # _

Other than in hyperref links and urls you need to put a backslash in front of them. For example to print a % in the text you need to use:

\%

Furthermore the character " is not at all used as discussed in the next section.

3.2.2 Highlighting Text

Quotes are not written with the " character, but are embedded in two left single quotes and two right single quotes:

' 'This is a quote''

which will result in:

"This is a quote"

In many papers we see that the quote is misused while putting quotes around a word. However quotes are often just used to quote a text from another paper. Instead of using quotes authors may actually emphasize a word. \LaTeX has a special command for that using:

{\em this is emphasized}

resulting in

this is emphasized

To write a text as bold (which should also be avoided as bold is typically used in section headers), you can use:

```
{\em this is bold fett}
```

resulting in

this is bold fett

3.2.3 Sections

LaTeX provides a convenient mechanism to structure a paper with sections and subsections. This is achieved with the following commands:

```
\section{This is a Section}
\subsection{This is a Subsection}
\subsubsection{This is a Subsubsection}
```

Once you use one of these commands the next paragraph will start below the section command.

In addition you have the command:

```
\paragraph{This is a paragraph.}
```

The line is behind the paragraph heading

The command is special as it does not introduce a new line between the Heading and the next line even if you include empty lines

3.2.4 Empty Lines

Multiple empty lines will be reduced to a single empty line.

3.2.5 Itemize

Itemized lists can be written as:

```
\begin{itemize}
  \item First item
  \item Second item
\begin{itemize}
```

resulting in

- First item
- Second item

3.2.6 Enumerate

Enumerations can be written as:

```
\begin{enumerate}
  \item First item
```

```
\item Second item
\begin{enumerate}
resulting in
```

1. First item
2. Second item

3.2.7 Descriptions

Description lists can be written as:

```
\begin{itemize}
\item[Cloud] My definition of a Cloud.
\item[Big Data] My definition of Big Data
\begin{itemize}
```

Cloud: My definition of a Cloud

Big Data: My definition of Big Data

3.2.8 Images

Figures are extremely easy to handle by including them from source. We never worry about the placement as LaTeX does typically a very good job of doing this.:

In Figure \ref{F:graph} we show a black and white graph about

```
\begin{figure}
\includegraphics[width=\columnwidth]{images/graph.pdf}
\caption{A sample black and white graphic. \cite{las17graph}}
\label{F:graph}
\end{figure}
```

Note that las17graph must be a label of a valid bibtex entry. This is needed if you have copied the image from elsewhere to avoid plagiarism. However, if you came up with the graph yourself than you do not need a citation.

We recommend that you place in your paper drafts all images at the which can be done with the endfloat package

This can be enabled if you include the following lines before begin document command:

```
\usepackage{endfloat}
\renewcommand{\efloatseparator}{\mbox{}}
```

```
\begin{document}
```

3.2.9 Tables

tables from csv tables by hand

3.2.10 Labels

As we saw already for figures and tables it is recommended to use the label and ref commands to refer to figure or table numbers. This applies also to sections. Thus I can place a label after a section:

```
\section{Introduction}\label{S:introduction}
```

and write elsewhere in the paper:

```
As we showcased in Section \ref{S:introduction}
```

Furthermore to conveniently distinguish sections tables and figures, we use the prefix S T F followed by a colon for the label. This helps organizing your paper in case you have many labels.

3.2.11 Mathematics

One of the strength of LaTeX is the ability to write easily sophisticated mathematical expressions on paper with high quality. A good online resource is provided by the following online resource from which we have copied some examples:

- <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

To activate them use

```
\usepackage{amsmath}
```

at the beginning of the document after the document class

Exponents are using the \wedge character:

```
$ (a + b)^2 = a^2 + 2ab + b^{c+2} $
```

$$(a - b)^2 = a^2 - 2ab + b^2$$

Greek letters are referred to by their name proceeded by the slash:

```
$$ \alpha \beta \gamma \Gamma \pi \Pi \phi $$
```

$$\alpha\beta\gamma\Gamma\pi\Pi\phi$$

Limits can be written as follows:

```
$$ \lim_{x \rightarrow \infty} \exp(-x) = 0 $$
```

$$\lim_{x \rightarrow \infty} \exp(-x) = 0$$

Fractions are indicated by the frac command, and binomials by binom:

```
$ \frac{n!}{k!(n-k)!} = \binom{n}{k} $
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Matrices can be created as follows:

```

A_{m,n} =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}

```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

3.3 Advanced topics

3.3.1 ACM and IEEE Proceedings Format



Figure 3.1: The look of the ACM and IEEE format templates

- <http://www.acm.org/publications/proceedings-template>
- https://www.ieee.org/conferences_events/conferences/publishing/templates.html

3.3.2 Generating and Managing Images

To produce high quality images the programs PowerPoint and omnigraffle on OSX are recommended. When using powerpoint please keep the image ratio to 4x3 as they produce nice size graphics which you also can use in your presentations. When using other ratios they may not fit in presentations and thus you may increase unnecessarily your work. We do not recommend vizio as it is not universally available and produces images that in case you have to present them in a slide presentation does not easily reformat if you do not use 4x3 aspect ratio.

Naturally, graphics should be provided in SVG or PDF format so they can scale well when we look at the final PDF. Including PNG, gif, or jpeg files often do not result in the necessary resolution or the files become real big. For this reason we for example can also not recommend tools such as tablauer as they do not provide proper exports to high quality publication formats. For interactive display such tool may be good, but for publications it produces inferior formatted images.

We recommend that all images be stored into a folder called images in the same directory where your \LaTeX main document resides.

3.3.3 Slides

Slides are best produced with the seminar package:

```
\documentclass{seminar}
```

```
\begin{slide}
```

```
Hello World on slide 1
```

```
\end{slide}
```

The text between slides is ignored

```
\begin{slide}
```

```
Hello World on slide 2
```

```
\end{slide}
```

However, in case you need to have a slide presentation we recommend you use ppt. Just paste and copy content from your PDF or your \LaTeX source file into the ppt.

3.3.4 Useful Online Information about \LaTeX

Latex Sheet: <https://wch.github.io/latexsheet/latexsheet.pdf>

Latex Short: <http://tug.ctan.org/info/lshort/english/lshort.pdf>

Wikibook: <https://en.wikibooks.org/wiki/LaTeX>

Wikibook (PDF) : <https://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>

Links to books: <https://latexforhumans.wordpress.com/2008/10/11/the-best-guides-to-latex/>

Links to books: <https://www.latex-project.org/help/books/>

LaTeXe: The \LaTeX Reference Manual provides a good introduction to Latex.

- \LaTeX Users and Reference Guide, by Leslie Lamport https://www.amazon.com/LaTeX-Document-Preparation-System/dp/0201529831/ref=sr_1_2?s=books&ie=UTF8&qid=1507114870&sr=1-2&keywords=lamport
- \LaTeX an Introduction, by Helmut Kopka https://www.amazon.com/Guide-LaTeX-4th-Helmut-Kopka/dp/0321173856/ref=pd_lpo_sbs_14_t_0?_encoding=UTF8&psc=1&refRID=2BB4APDFEX34A4JM65ZB
- The \LaTeX Companion, by Frank Mittelbach <https://www.amazon.com/LaTeX-Companion-Techniques-Computer-Science/dp/0201362996>

3.3.5 \LaTeX vs. X

We will refrain from providing a detailed analysis on why we use \LaTeX in many cases versus other technologies. In general, we find that \LaTeX :

- is incredibly stable
- produces high-quality output
- is platform independent
- has lots of templates
- has been around for many years so it works well
- removes you from the pain of figure placements
- focusses you on content rather than the appearance of the paper
- integrates well with code repositories such as git to write collaborative papers.
- has superior bibliography integration
- has a rich set of tools that make using \LaTeX easier
- authors do not play with layouts much so papers in a format are uniform

In case you need a graphical view to edit \LaTeX or LateX exportable files you also find AucTeX and Lyx.

Word

Word is arguably available to many, but if you work on Linux you may be out of luck. Also Word often focusses not on structure of the text but on its appearance. Many students abuse Word and the documents in Word become a pain to edit with multiple users. Recently Microsoft has offered online services to collaborate on writing documents in groups which work well. Integration with bibliography managers such as endnote or Mendeley is possible.

However, we ran into issues whenever we use word:

- Word tends sometimes to crash for unknown reasons and we lost a lot of work
- Word has some issues with the bibliography managers and tends to crash sometimes for unknown reasons.
- Word is slow with integration to large bibliographies.
- Figure placement in Word in some formats is a disaster and you will spend many hours to correct things just to find out that if you make small changes you have to spend additional many hours to get used to the new placement. We have not yet experienced a word version where we have not lost images. Maybe that has changed, so let us know

However, we highly recommend the collaborative editing features of Word that work on a paragraph and not letter level. Thus saving is essential so you do not block other people from editing the paragraph.

Google Docs

Unfortunately, many useful features got lost in the new google docs. However, it is great to collaborate quickly online, share thoughts and even write your latex documents together if you like (just copy your work in a file offline and use latex to compile it ;-)

The biggest issue we have with Google Docs is that it does not allow the support of 2 column formats, that the bibliography integration is non-existent and that paste and copy from web pages and images encourages unintended plagiarism when collecting information without annotations (\LaTeX and Word are prone to this too, but we found from experience that it tends to happen more with Google docs users).

A Place for Each

When looking at the tools we find a place for each:

Google docs: Short meeting notes, small documents, quick online collaborations to develop documents collaboratively at the same time.

Word: Available to many, supports 2 column format, supports paragraph based collaborative editing, Integrates with bibliography managers.

LaTeX: Reduces failures, great offline editing, superior bibliography management, superior image placement, runs everywhere. Great collaborative editing with sharelatex, allows easy generation of proceedings written by hundreds of people with shared index.

The best choice for your class: LaTeX

3.4 Editing

3.4.1 Emacs

The text editor emacs provides a great basis for editing TeX and LaTeX documents. Both modes are supported. In addition there exists a color highlight module enabling the color display of LaTeX and TeX commands. On OSX aquemacs and carbon emacs have build in support for LaTeX. Spell checking is done with flyspell in emacs.

Aquamacs

Aquamacs is an editor based on GNU Emacs that runs on OSX and integrates with the OSX desktop. This is for many the preferred editor on OSX for \LaTeX .

<http://aquamacs.org>

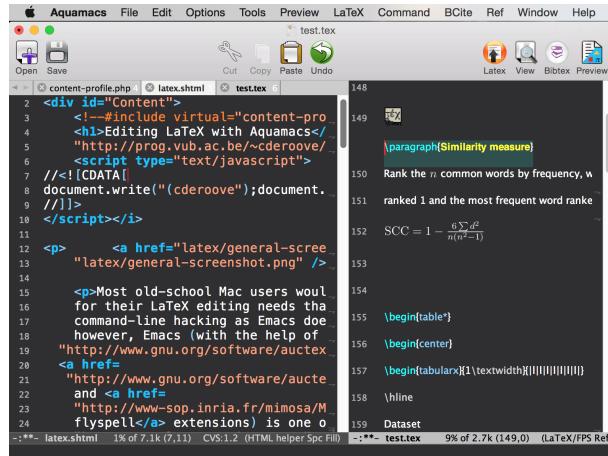


Figure 3.2: Aquamacs

3.4.2 Vi/Vim

Another popular editor is vi or vim. It is less feature rich but many programmers are using it. As it can edit ASCII text you can edit LaTeX. With the LaTeX add-ons to vim, vim becomes similar powerful while offering help and syntax highlighting for LaTeX as emacs does. (The authors still prefer emacs)

3.4.3 TeXshop

Other editors such as TeXshop are available which provide a more integrated experience. However, we find them at times to stringent and prefer editors such as emacs.

3.4.4 LyX

We have made very good experiences with Lyx. You must assure that the team you work with uses it consistently and that you all use the same version.

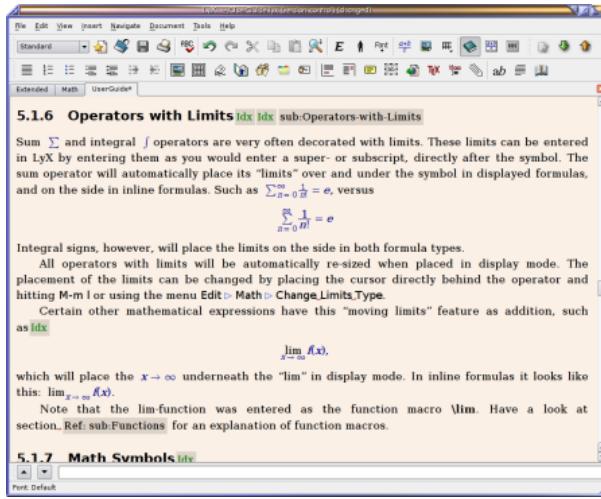


Figure 3.3: Lyx

Using the ACM templates is documented here:

- <https://wiki.lyx.org/Examples/AcmSiggraph>

On OSX it is important that you have a new version of LaTeX and Lyx installed. As it takes up quite some space, you may want to delete older versions. The new version of LyX comes with the acmsigplan template included. However on OSX and other platforms the .cls file is not included by default. However the above link clearly documents how to fix this.

3.4.5 WYSIWYG locally

We have found that editors such as Lyx and Auctex provide very good WYSIWYG alike features. However, we found an even easier way while using skim, a pdf previewer, in conjunction with emacs and latexmk. This can be achieved while using the following command assuming your latex file is called ‘report.tex’:

```
latexmk -pvc -view=pdf report
```

This command will update your pdf previewer (make sure to use skim) whenever you edit the file report.tex and save it. It will maintain via skim the current position, thus you have a real great way of editing in one window, while seeing the results in the other.

Skim can be found at: <http://skim-app.sourceforge.net/>

3.4.6 Markdown and \LaTeX

It may come as a surprise to many that one can actually write simple \LaTeX documents also in markdown Syntax or mix section written in markdown while others are written in \LaTeX . To do so all you ahve to do is place the markdown text in a separate file. Let us call the file `content.md` which has the following lines included in it:

```
# Section

* item a
* item b
```

Obviously, we would have to convert this to \LaTeX . Luckily there is a very useful program called `pandoc` that does this for you. YOu could make the translation in the shell, but you could also make the translation locally on your computer while allowing \LaTeX to start up external programs. This is achieved with the `write18` command and allowing \LaTeX explicitly to call external programs. Please inspect the following latex file that includes a template on how to do this. We assume the file is called `markdown.tex` for our example.

```
\documentclass{article}

\include{graphicx}
\newcommand{\tightlist}{{}

\begin{document}
\immediate\write18{pandoc content.md -o content.tex}

\input{content}

\end{document}
```

Now to generate the PDF we simply have to call the following command that include the `-shell-escape` flag to allow the execution of `write18` embedded commands:

```
pdflatex -shell-escape markdown-test
```

The output will be `markdown.pdf` with the content from the markdown file translated. Doing this naturally allows you to write large portions in markdown and automatically include them in your \LaTeX document. Hence, you can use editors such as Macdown to initially work in semi WYSIWYG mode and do fairly straight forward edition. Naturally the same can be done in RST. Naturally the most elementary features are supported. For more sophisticated features, please use \LaTeX directly.

3.4.7 Including RST into \LaTeX

`content.rst`:

Section

- * item a
- * item b

`sample.tex`:

```
\documentclass{article}

\include{graphicx}
\newcommand{\tightlist}{}{}

\begin{document}
\immediate\write18{pandoc content.rst -o content.tex}

\input{content}

\end{document}
```

3.4.8 pyCharm

TODO: comment on how we can use pycharm for editing and what the limitations are.

3.4.9 MSWord

it is possible to use Word.

be careful with

3.5 The \LaTeX Cycle

To create a PDF file from latex yo need to generate it following a simple development and improvement cycle.

First, Create/edit ASCII source file with `file.tex` file:

```
emacs file.tex
```

Create/edit bibliography file:

```
jabref refs.bib
```

Create the PDF:

```
pdflatex file
bibtex file
pdflatex file
pdflatex file
```

View the PDF:

```
open file
```

It not only showcases you an example file in ACM 2 column format, but also integrates with a bibliography. Furthermore, it provides a sample Makefile that you can use to generate view and recompile, or even autogenerated. A compilation would look like:

```
make
make view
```

If however you want to do things on change in the tex file you can do this automatically simply with:

```
make watch
```

for make watch its best to use skim as pdf previewer

3.6 Tips

Including figures over two columns:

- <http://tex.stackexchange.com/questions/30985/displaying-a-wide-figure-in-a-two-column>
- positioning figures with textwidth and columnwidth https://www.sharelatex.com/learn/Positioning_images_and_tables
- An organization as the author. Assume the author is National Institute of Health and want to have the author show up, please do:
`key= {National Institute of Health},
author= {{National Institute of Health}},
Please note the {{ }}`
- words containing ‘fi’ or ‘ffi’ showing blank places like below after recompiling it: find as nd efficiency as e ciency
You copied from word or PDF ff which is actually not an ff, but a condensed character, change it to ff and ffi, you may find other such examples such as any non ASCII character. A degree is for example another common issue in data science.
- do not use | & and other latex characters in bibtex references, instead use , and the word and
- If you need to use _ it is _ but if you use urls leave them as is
- We do recommend that you use sharelatex and jabref for writing papers. This is the easiest solution and beats in many cases MSWord as you can focus on writing and not on formatting.



4. Managing Bibliographies

 chapter/lesson/doc/bibtex.tex

4.0.1 Integrating Bibliographies

LaTeX integrates very well with bibtex. There are several pre-formatted styles available. It includes also styles for ACM and IEEE bibliographies. For the ACM style we recommend that you replace abbrv.bst with abbrvurl.bst, add hyperref to your usepackages so you can also display URLs in your citations:

```
\bibliographystyle{IEEETran}
\bibliography{references.bib}
```

Then you have to run latex and bibtex in the following order:

```
latex file
bibtex file
latex file
latex file
```

or simply call make from our makefile.

The reason for the multiple execution of the latex program is to update all cross-references correctly. In case you are not interested in updating the library every time in the writing progress just postpone it till the end. Missing citations are viewed as [?].

Two programs stand out when managing bibliographies: emacs and jabref:

- <http://www.jabref.org/>

Other programs such as Mendeley, Zotero, and even endnote integrate with bibtex. However their support is limited, so we recommend that you just use jabref. Furthermore its free and runs on all platforms.

jabref

Jabref is a very simple to use bibliography manager for LaTeX and other systems. It can create a multitude of bibliography file formats and allows upload in other online bibliography managers.

- Installation: Go to <http://www.jabref.org/> and click download
- Video: <https://youtu.be/cMtYOHCHZ3k>
- Video with cc: <https://www.youtube.com/watch?v=QVbifcLgMic>

jabref and MSWord

According to others it is possible to integrate jabref references directly into MSWord. This has been conducted so far however only on a Windows computer.

We have not tried this ourselves, but give it as a potential option.

Here are the steps the need to be done:

1. Create the Jabref bibliography just like in presented in the Jabref video
2. After finishing adding your sources in Jabref, click File -> export
3. Name your bibliography and choose MS Office 2007 (*.xml) as the file format. Remember the location of where you saved your file.
4. Open up your word document. If you are using the ACM template, go ahead and remove the template references listed under Section 7. References

5. In the MS Word ribbon choose ‘References’
6. Choose ‘Manage Sources’
7. Click ‘Browse’ and locate/select your Jabref xml file
8. You should now see your references appear in the left side window. Select the references you want to add to your document and click the ‘copy’ button to move them from the left side window to the right window.
9. Click the ‘Close’ button
10. In the MS Word Ribbon, select ‘Bibliography’ under the References tab
11. Click ‘Insert Bibliography’ and your references should appear in the document
12. Ensure references are of Style: IEEE. Styles are located in the References tab under ‘Manage Sources’

As you can see there is significant effort involved, so we do recommend you use LaTeX as you can focus there on content rather than dealing with complex layout decisions. This is especially true, if your papers have figures or tables, or you need to add references.

4.1 Entry types

In this section we will explain how to find and properly generate bibliographic entries. We are using bibtex for this as it is easy to use and generates reasonable entries that can be included in papers. What we like to achieve in this section is not to just show you a final entry, but to document the process on how that entry was derived. This will allow you to replicate or learn from the process to apply to your own entries.

We will address a number of important entry types which includes:

- wikipedia entries
- github entries
- books
- articles in a scientific journal
- articles in a conference
- articles in magazines (non scientific)
- blogs

4.1.1 Source code References

We will learn how to cite a source code from a publicly hosted repository. Such repositories are frequently used and include, for example github, bitbucket, sourcefore, or your Universities code repository as long as it is publicly reachable. As changes can occur on these repositories, it is important that the date of access is listed in the entry or even the release version of the source code.

Let us without bias chose a random source code entry that has been contributed by a student as follows:

```
@Misc{gonzalez_2015,
  Title = {Buildstep},
  Author = {Gonzalez, Jose and Lindsay, Jeff},
  HowPublished = {Web Page},
  Month = {Jul},
  Note = {Accessed: 2017-1-24},
  Year = 2015,
```

```

Key = {www-buildstep},
Url = {https://github.com/progium/buildstep}
}

```

Is this entry correct? Let us analyse.

Entry type Misc

First, it seems appropriate to use a `@misc` entry. We correctly identify this is a misc entry as it is online available. More recent version of bibtex include also the type `@online` for it. However, in order to maintain compatibility to older formats we chose simply Misc here and if we really would need to we could replace it easily

Label

Typically the Label should contain 3 letters from an author name, short year and the short name of the publication to provide maximum information regarding the publication. Underscores need to be replaced by dashes or removed. However as this is a github repository it is better to integrate this into the label. Hence, we simply use the github-projectname (in our case github-buildstep, out of convention we only use lower case letters).

Author

Unless the last name contains spaces, it should be first name followed by the last name with multiple authors separated with “and”.

Key

In this case the key field can be removed as the entry has an author field entry. If there was no author field, we could use key to specify the alphabetical ordering based on the specified key. Note that a key is not the label. In fact in our original entry the key field was wrongly used and the student did not understand that the key is used for sorting.

Howpublished

Since the source is a github project repository, the howpublished field shall hold the value {Code Repository} rather than a web page. If the url specified was a normal webpage, the {Web Page} entry would be valid.

Month

The lowercase month is, used for international notation since months are not capitalized in some other languages.

Owner

In class we introduced the convention to put the student HID in it. If multiple students contributed, add them with space separation.

Accessed

As we do not yet typically an accessed field, we simply include it in the note field. This is absolutely essential as code can change and when we read the code we looked at a particular snapshot in time.

In addition it is often necessary to record the actual version of the code. Typically this can also be done with the month and year field while relying on a release date

Final Entry

Filling out as many fields as possible with information for this entry we get:

```
@Misc{github-buildstep,
  Title = {Buildstep},
  Author = {Jose Gonzalez and Jeff Lindsay},
  HowPublished = {Code Repository},
  Year = {2015},
  Month = jul,
  Note = {Accessed: 2017-1-24},
  Url = {https://github.com/progium/buildstep},
  Owner = {S17-I0-3025},
}
```

We are using the release date in the year and month field as this project uses this for organizing releases. However, other project may have release versions so you would have in addition to using the data also to include the version in the note field such as:

```
Note = {Version: 1.2.3, Accessed: 2017-1-24},
```

All those that helped should add your HID to this entry with a space separated from each other

4.1.2 Researching proper bibtex entries

Article in a journal

Many online bibtex entries are wrong or incomplete. Often you may find via google a bibtex entry that may need some more research. Lets assume your first google query returns a publication and you cite it such as this:

```
@Unpublished{unpublished-google-sawzall,
  Title = {{Interpreting the Data: Parallel Analysis with Sawzall}},
  Author = {{Rob Pike, Sean Dorward, Robert Griesemer, Sean Quinlan}},
  Note = {accessed 2017-01-28},
  Month = {October},
  Year = {2005},
  Owner = {for the purpose of this discussion removed},
  Timestamp = {2017.01.31}
}
```

Could we improve this entry to achieve your best? We observe:

1. The author field has a wrong entry as the , is to be replaced by an and.
2. The author feild has authors and thus must not have a {{ }}
3. The url is missing, as the simple google search actually finds a PDF document.

Let us investigate a bit more while searching for the title. We find

- A) https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj_ytSA-PDRAhUH8IMKHaomC-oQFggaMAA&url=https%3A%2F%2Fresearch.google.com%2Farchive%2Fsawzall-sciprog.pdf&usg=AFQjCNHSSfKBwbxVAVPQ0td4rTjitKucpA&sig2=vbiVzi36B3gGFjIzlUKBDA&bvm=bv

- B) <https://research.google.com/pubs/pub61.html>
- C) <http://dl.acm.org/citation.cfm?id=1239658>

Let us look at A)

As you can see from the url this is actually some redirection to a google web page which probably is replaced by B as its from google research. So let us look at B)

Now when you look at the link we find the url <https://research.google.com/archive/sawzall-sciprog.pdf> which redirects you to the PDF paper.

When we go to B) we find surprisingly a bibtex entry as follows:

```
@article{61,
  title = {Interpreting the Data: Parallel Analysis with Sawzall},
  author = {Rob Pike and Sean Dorward and Robert Griesemer and Sean Quinlan},
  year = 2005,
  URL = {https://research.google.com/archive/sawzall.html},
  journal = {Scientific Programming Journal},
  pages = {277--298},
  volume = {13}
}
```

Now we could say lets be satisfied, but C) seems to be even more interesting as its from a major publisher. So lets just make sure we look at C)

If you go to C, you find under the colored box entitled Tools and Resources a link called **bibtex**. Thus it seems a good idea to click on it. This will give you:

```
@article{Pike:2005:IDP:1239655.1239658,
  author = {Pike, Rob and Dorward, Sean and Griesemer, Robert and Quinlan, Sean},
  title = {Interpreting the Data: Parallel Analysis with Sawzall},
  journal = {Sci. Program.},
  issue_date = {October 2005},
  volume = {13},
  number = {4},
  month = oct,
  year = {2005},
  issn = {1058-9244},
  pages = {277--298},
  numpages = {22},
  url = {http://dx.doi.org/10.1155/2005/962135},
  doi = {10.1155/2005/962135},
  acmid = {1239658},
  publisher = {IOS Press},
  address = {Amsterdam, The Netherlands, The Netherlands},
}
```

Now we seem to be at a position to combine our search result as neither entry is sufficient. As the doi number properly specifies a paper (look up what a doi is) we can replace the url with one that we find online, such as the one we found in A) Next we see that all fields in B are already covered in C, so we take C) and add the url. Now as the label is great and uniform for ACM, but for us a bit less convenient as its difficult to remember, we just change it while for example using authors, title, and year information. Lets also make sure to do mostly lowercase in the label just as a convention.

Thus our entry looks like:

```
@article{pike05swazall,
    author = {Pike, Rob and Dorward, Sean and Griesemer, Robert and Quinlan, Sean},
    title = {Interpreting the Data: Parallel Analysis with Sawzall},
    journal = {Sci. Program.},
    issue_date = {October 2005},
    volume = {13},
    number = {4},
    month = oct,
    year = {2005},
    issn = {1058-9244},
    pages = {277--298},
    numpages = {22},
    url = {https://research.google.com/archive/sawzall-sciprog.pdf},
    doi = {10.1155/2005/962135},
    acmid = {1239658},
    publisher = {IOS Press},
    address = {Amsterdam, The Netherlands, The Netherlands},
}
```

As you can see properly specifying a reference takes multiple google queries and merging of the results you find from various returns. As you still have time to correct things I advise that you check your references and correct them. If the original reference would have been graded it would have been graded with a “fail” instead of a “pass”.

4.1.3 Article in a conference proceedings

Lets look at a second obvious example that needs improvement:

```
@InProceedings{wettinger-any2api,
    Title          = {Any2API - Automated APIfication},
    Author         = {Wettinger, Johannes and
                      Uwe Breitenb\\"ucher
                      and Frank Leymann},
    Booktitle      = {Proceedings of the 5th International
                      Conference on Cloud Computing and
                      Services Science},
    Year           = {2015},
    Pages          = {475486},
    Publisher      = {SciTePress},
    ISSN           = {2326-7550},
    Owner          = {S17-I0-3005},
    Url            = {https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38}
```

As you can see this entry seems to define all required fields, so we could be tempted to stop here. But its good to double check. Lets do some queries against ACM, . and google scholar, so we just type in the title, and if this is in a proceedings they should return hopefully a predefined bibtex record for us.

Lets query:

```
google: googlescholar Any2API Automated APIfication
```

We get:

- https://scholar.google.de/citations?view_op=view_citation&hl=en&user=j6lIXt0AAAAJ&citation_for_view=j6lIXt0AAAAJ:8k81kl-MbHgC

On that page we see Cite

So we find a PDF at <https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf>

Lets click on this and the document includes a bibtex entry such as:

```
@inproceedings{Wettinger2015,
  author= {Johannes Wettinger and Uwe Breitenb\{\"u\}cher and Frank
           Leymann},
  title = {Any2API - Automated APIfication},
  booktitle = {Proceedings of the 5th International Conference on Cloud
               Computing and Service Science (CLOUDER)},
  year = {2015},
  pages = {475--486},
  publisher = {SciTePress}
}
```

Now lets add the URL and owner:

```
@inproceedings{Wettinger2015,
  author= {Johannes Wettinger and Uwe Breitenb\{\"u\}cher and Frank
           Leymann},
  title = {Any2API - Automated APIfication},
  booktitle = {Proceedings of the 5th International Conference on Cloud
               Computing and Service Science (CLOUDER)},
  year = {2015},
  pages = {475--486},
  publisher = {SciTePress},
  url ={https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf},
  owner = {S17-I0-3005},
}
```

Should we be satisfied? No, even our original information we gather provided more information.

So lets continue. Lets googlesearch different queries with ACM or IEEE and the title. When doing the IEEE in the example we find an entry called

dlp: Frank Leyman

Lets look at it and we find two entries:

```
@inproceedings{DBLP:conf/closer/WettingerBL15,
  author    = {Johannes Wettinger and
              Uwe Breitenb\{\"u\}cher and
              Frank Leymann},
  title     = {{\{ANY2API\}} - Automated APIfication - Generating APIs for Executables
              to Ease their Integration and Orchestration for Cloud Application
              Deployment Automation},
```

```

booktitle = {{CLOSER} 2015 - Proceedings of the 5th International Conference on
             Cloud Computing and Services Science, Lisbon, Portugal, 20-22 May,
             2015.},
pages      = {475--486},
year       = {2015},
crossref   = {DBLP:conf/closer/2015},
url        = {http://dx.doi.org/10.5220/0005472704750486},
doi        = {10.5220/0005472704750486},
timestamp  = {Tue, 04 Aug 2015 09:28:21 +0200},
biburl     = {http://dblp.uni-trier.de/rec/bib/conf/closer/WettingerBL15},
bibsource  = {dblp computer science bibliography, http://dblp.org}
}

@proceedings{DBLP:conf/closer/2015,
editor      = {Markus Helfert and
               Donald Ferguson and
               V{\'e}ronique M{\'e}ndez Mu{\~n}oz},
title       = {{CLOSER} 2015 - Proceedings of the 5th International Conference on
               Cloud Computing and Services Science, Lisbon, Portugal, 20-22 May,
               2015}},
publisher   = {SciTePress},
year        = {2015},
isbn        = {978-989-758-104-5},
timestamp  = {Tue, 04 Aug 2015 09:17:34 +0200},
biburl     = {http://dblp.uni-trier.de/rec/bib/conf/closer/2015},
bibsource  = {dblp computer science bibliography, http://dblp.org}
}

```

So lets look at the entry and see how to get a better one for our purpose to combine them. When using jabref, you see optional and required fields, we want to add as many as possible, regardless if optional or required, so Lets do that (I write here in ASCII as easier to document:

```

@InProceedings{,
author      = {},
title       = {},
OPTcrossref = {},
OPTkey      = {},
OPTbooktitle = {},
OPTyear     = {},
OPTeditor   = {},
OPTvolume   = {},
OPTnumber   = {},
OPTseries   = {},
OPTpages    = {},
OPTmonth    = {},
OPTaddress  = {},
OPTorganization = {},
OPTpublisher = {},
OPTnote     = {},
OPTannote   = {},
}
```

```
url = {}
}
```

So lets copy and fill out the **form** from our various searches:

```
@InProceedings{Wettinger2015any2api,
  author    = {Johannes Wettinger and
               Uwe Breitenb\"{u}cher and
               Frank Leymann},
  title     = {{ANY2API - Automated APIfication - Generating APIs for Executables
               to Ease their Integration and Orchestration for Cloud Application
               Deployment Automation}},
  booktitle = {{CLOSER 2015 - Proceedings of the 5th International Conference on
               Cloud Computing and Services Science}},
  year      = {2015},
  editor    = {Markus Helfert and
               Donald Ferguson and
               V{\'e}ctor M{\'e}ndez Mu{\~n}oz},
  publisher = {SciTePress},
  isbn      = {978-989-758-104-5},
  pages     = {475--486},
  month     = {20-22 May},
  address   = {Lisbon, Portugal},
  doi       = {10.5220/0005472704750486},
  url       = {https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf},
  owner     = {S17-I0-3005},
}
```

4.1.4 What are the differnt entry types and fields

We were asked what are the different entry types and fields, so we did a google query and found the following useful information. please remember that we also have fields such as doi, owner, we will add status ={pass/fail} at time of grading to indicate if the reference passes or fails. We may assign this to you so you get familiar with the identification if a refernce is ok or not.

Please see <https://en.wikipedia.org/wiki/BibTeX>

4.1.5 InProceedings

Please fill out

```
@InProceedings{,
  author = {},
  title = {},
  OPTcrossref = {},
  OPTkey = {},
  OPTbooktitle = {},
  OPTyear = {},
  OPTeditor = {},
  OPTvolume = {},
  OPTnumber = {},
```

```

OPTseries =    {},
OPTpages =    {},
OPTmonth =    {},
OPTaddress =  {},
OPTorganization = {},
OPTpublisher = {},
OPTnote =     {},
OPTannote =   {},
url = {}

}

@inproceedings{vonLaszewski15tas,
author =      {DeLeon, Robert L. and Furlani, Thomas R. and Gallo,
               Steven M. and White, Joseph P. and Jones, Matthew
               D. and Patra, Abani and Innus, Martins and Yearke,
               Thomas and Palmer, Jeffrey T. and Sperhac, Jeanette
               M. and Rathsam, Ryan and Simakov, Nikolay and von
               Laszewski, Gregor and Wang, Fugang},
title =       {{TAS View of XSEDE Users and Usage}},
booktitle =   {Proceedings of the 2015 XSEDE Conference: Scientific
               Advancements Enabled by Enhanced
               Cyberinfrastructure},
series =      {XSEDE '15},
year =        2015,
isbn =        {978-1-4503-3720-5},
location =   {St. Louis, Missouri},
pages =       {21:1--21:8},
articleno =   21,
numpages =   8,
url =         {http://doi.acm.org/10.1145/2792745.2792766},
doi =         {10.1145/2792745.2792766},
acmid =       2792766,
publisher =   {ACM},
address =     {New York, NY, USA},
keywords =    {HPC, SUPReMM, TAS, XDMoD, XSEDE usage, XSEDE users},
}

```

4.1.6 TechReport

Please fill out

```

@TechReport{,
author =      {},
title =       {},
institution = {},
year =        {},
OPTkey =     {},
OPTtype =    {},
OPTnumber =  {},
OPTaddress = {}
}
```

```

OPTmonth =      {},
OPTnote =      {},
OPTannote =    {},
url = {}

}

@TechReport{las05exp,
  title =  {{The Java CoG Kit Experiment Manager}},
  Author =  {von Laszewski, Gregor},
  Institution =  {Argonne National Laboratory},
  Year =  2005,
  Month =  jun,
  Number =  {P1259},
  url =  {https://laszewski.github.io/papers/vonLaszewski-exp.pdf}
}

```

4.1.7 Article

Please fill out

```

@Article{,
  author =      {},
  title =       {},
  journal =     {},
  year =        {},
  OPTkey =      {},
  OPTvolume =   {},
  OPTnumber =   {},
  OPTpages =    {},
  OPTmonth =    {},
  OPTnote =     {},
  OPTannote =   {},
  url =         {}
}

@Article{las05gridhistory,
  title =  {{The Grid-Idea and Its Evolution}},
  author =  {von Laszewski, Gregor},
  journal =  {Journal of Information Technology},
  year =  2005,
  month =  jun,
  number =  6,
  pages =  {319-329},
  volume =  47,
  doi =    {10.1524/jitit.2005.47.6.319},
  url =  {https://laszewski.github.io/papers/vonLaszewski-grid-idea.pdf}
}

```

4.1.8 Proceedings

Please fill out

```

@Proceedings{,
    title =      {},
    year =       {},
    OPTkey =     {},
    OPTbooktitle = {},
    OPTeditor =   {},
    OPTvolume =   {},
    OPTnumber =   {},
    OPTseries =   {},
    OPTaddress =  {},
    OPTmonth =    {},
    OPTorganization = {},
    OPTpublisher = {},
    OPTnote =     {},
    OPTannote =   {},
    url =        {}
}

@Proceedings{las12fedcloud-proc,
    title = {{FederatedClouds '12: Proceedings of the 2012
              Workshop on Cloud Services, Federation, and the 8th
              Open Cirrus Summit}},
    year = 2012,
    address = {New York, NY, USA},
    editor = {vonLaszewski, Gregor and Robert Grossman and Michael
              Kozuch and Rick McGee and Dejan Milojicic},
    publisher = {ACM},
    ISBN = {978-1-4503-1754-2},
    location = {San Jose, California, USA},
    url =
        {http://dl.acm.org/citation.cfm?id=2378975&picked=prox&cfid=389635474&cft}
}

```

4.1.9 Wikipedia Entry

Please fill out

```

@Misc{,
    OPTkey =     {},
    OPTauthor =  {},
    OPTtitle =   {},
    OPThowpublished = {},
    OPTmonth =   {},
    OPTyear =    {},
    OPTnote =    {},
    OPTannote =  {},
    url =        {}
}

@Misc{www-ode-wikipedia,
    Title = {Apache ODE},

```

```

HowPublished = {Web Page},
Note =      {Accessed: 2017-2-11},
Key =        {Apache ODE},
Url =        {https://en.wikipedia.org/wiki/Apache_ODE}
}

```

4.1.10 Blogs

Please fill out

```

@Misc{,
OPTkey =      {},
OPTauthor =   {},
OPTtitle =    {},
OPThowpublished = {},
OPTmonth =    {},
OPTyear =     {},
OPTnote =     {},
OPTannote =   {},
OPTurl =      {}
}

@Misc{www-clarridge-discoproject-blog,
title =  {Disco - A Powerful Erlang and Python Map/Reduce
           Framework},
author = {Clarridge, Tait},
howpublished = {Blog},
month =  may,
note =   {Accessed: 25-feb-2017},
year =   2014,
url =   {http://www.taitclarridge.com/techlog/2014/05/disco-a-powerful-erlang-and-python-m}
}

```

4.1.11 Web Page

Please fill out

```

@Misc{,
OPTkey =      {},
OPTauthor =   {},
OPTtitle =    {},
OPThowpublished = {},
OPTmonth =    {},
OPTyear =     {},
OPTnote =     {},
OPTannote =   {},
url =        {}
}

@Misc{www-cloudmesh-classes,
OPTkey =      {}
}

```

```

author = {von Laszewski, Gregor},
title = {Cloudmesh Classes},
howpublished = {Web Page},
OPTmonth = {},
OPTyear = {},
OPTnote = {},
OPTannote = {},
url = {https://cloudmesh.github.io/classes/}
}

@Misc{www-awslambda,
title = {AWS Lambda},
author = {{Amazon}},
key = {AWS Lambda},
howpublished = {Web Page},
url = {https://aws.amazon.com/lambda/faqs/}
}

```

4.1.12 Book

Given the following entry. What is the proper entry for this book. Provide rationale:

```

@Book{netty-book,
Title = {Netty in Action},
Author = {Maurer, Norman and Wolfthal, Marvin},
Publisher = {Manning Publications},
Year = {2016},
}

```

To obtain the record of a book you can look at many information sources. The can include:

- <https://www.manning.com/books/netty-in-action>
- <https://www.amazon.com/Netty-Action-Norman-Maurer/dp/1617291471>
- <http://www.barnesandnoble.com/w/netty-in-action-norman-maurer/1117342155?ean=9781617291470#productInfoTabs>
- <http://www.powells.com/book/netty-in-action-9781617291470/1-0>

Furthermore, we need to consider the entry of a book, we simply look it up in emacs where we find the following but add the owner and the url field:

```

@Book{,
ALTAuthor = {},
ALTeditor = {},
title = {},
publisher = {},
year = {},
OPTkey = {},
OPTvolume = {},
OPTnumber = {},
OPTseries = {},
OPTaddress = {},
OPTedition = {},
OPTmonth = {},

```

```

OPTnote =    {},
OPTannote =   {},
owner =      {},
url =  {}

}

```

In summary we find the following fields:

Required fields: author/editor, title, publisher, year

Optional fields: volume/number, series, address, edition, month, note, key

We apply the following to fill out the fields.

address: The address is the Publisher's address. Usually just the city, but can be the full address for lesser-known publishers.

author: The name(s) of the author(s) (in the case of more than one author, separated by and). Names can be written in one of two forms: Donald E. Knuth or Knuth, Donald E. or van Halen, Eddie. Please note that Eddie van Halen would result in a wrong name. For our purpose we keep nobelity titles part of the last name.

edition: The edition of a book, long form (such as "First" or "Second")

editor: The name(s) of the editor(s)

key: A hidden field used for specifying or overriding the alphabetical order of entries (when the "author" and "editor" fields are missing). Note that this is very different from the key that is used to cite or cross-reference the entry.

label: The label field should contain three letters from the auth field, a short year reference and a short name of the publication to provide the maximum information regarding the publication. Underscores should be replaced with dashes or removed completely.

month: The month of publication or, if unpublished, the month of creation. Use three-letter abbreviations for this field in order to account for languages that do not capitalize month names. Additional information for the day can be included as follows: aug #“~10,”

publisher: The publisher's name

series: The series of books the book was published in (e.g. "The Hardy Boys" or "Lecture Notes in Computer Science")

title: The title of the work. As the capitalization depends on the bibliography style and the language used we typically use camel case. To force capitalization of a word or its first letter you can use the curly braces, '{ }'. To keep the title in camel case simple use title = { {My Title} }

type: The field overriding the default type of publication (e.g. "Research Note" for techreport, "{PhD} dissertation" for phdthesis, "Section" for inbook/incollection) volume The volume of a journal or multi-volume book year The year of publication (or, if unpublished, the year of creation)

While applying the above rules and tips we summarize what we have done for this entry:

1. Search for the book by title/Author on ACM (<http://dl.acm.org/>) or Amazon or barnesandnoble or upcitemdb (<http://upcitemdb.com>). These services return bibtex entrie that you can improve.
2. Hence one option is to get the ISBN of the book. For "Mesos in action" from upcitemdb we got the ISBN as "9781617 292927". This is the 13 digit ISBN. The first 3 digits (GS1 code) can be skipped. Using the rest of 10 digits "1617 292927", Add in JabRef in Optional Fields->ISBN.

However it is fine to just specify the full number.

We can also return a bibtex entry generated while using Click on the "Get BibTex from ISBN".

Now we get more information on this book entry from ISBN. We can opt either the original or newly searched entry for the below bibtex fields or merge as appropriate. URL may not match from where we initially read the book, however there is option to put your original url or newly searched url. EAN, Edition, Pages,url,published date etc. Do a search on amazon for “ASIN”. Can skip if not available. Sometime we get ASIN for a different publication, maybe a paperback ASIN={B01MT311CU} We can add it as it becomes easier to search

doi: If you can find a doi numer you should also add it. IN this case we could not locate one.

As a result we obtain the entry:

```
@Book{netty-book,
  title = {Netty in Action},
  publisher = {Manning Publications Co.},
  year = {2015},
  author = {Maurer, Norman and Wolfthal, Marvin Allen},
  address = {Greenwich, CT, USA},
  edition = {1st},
  isbn = {1617291471},
  asin = {1617291471},
  date = {2015-12-23},
  ean = {9781617291470},
  owner = {S17-I0-3022 S17-I0-3010 S17-I0-3012},
  pages = {296},
  url = {http://www.ebook.de/de/product/21687528/norman_maurer_netty_in_action.html},
}
```

4.2 Integrating Bibtex entries into Other Systems

We have not tested any of this

4.2.1 Bibtex import to MSWord

XML import

Please respond back to us if you have used this and give feedback.

1. In JabRef, export the bibliography in MS Word 2008 xml format
2. Name the file Sources.xml (case sensitive)
3. In OSX with MS Word 2015: Go to /Library/Containers/com.microsoft.word/Data/Library/Application Support/Microsoft/Word/
4. Rename the original Sources.xml file to Sources.xml.bak
5. Copy the generated Sources.xml in this folder
6. Restart MS Word.

We do not know what needs to be done in case you need to make changes to the references. Please report back your experiences. To avoid issues we recommend that you use LaTeX. and not MSWord.

BibTex4Word

We have not tried this:

- <http://www.ee.ic.ac.uk/hp/staff/dmb/perl/index.html>

You are highly recommended to use Jabref for bibliography management in this class. Here is an

introductory video on Jabref: <https://youtu.be/roi7vezNmfo?t=8m6s>

4.3 Other Reference Managers

Please note that you should first decide which reference manager you like to use. In case you for example install zotero and mendeley, that may not work with word or other programs.

4.3.1 Endnote

Endnote os a reference manager that works with Windows. Many people use Endnote. However, in the past, Endnote has caused complications when dealing with collaborative management of references. Its price is considerable. We have lost many hours of work because of instability of Endnote in some cases. As a student, you may be able to use Endnote for free at Indiana University.

- <http://endnote.com/>

4.3.2 Mendeley

Mendeley is a free reference manager compatible with Windows Word 2013, Mac Word 2011, LibreOffice, BibTeX. Videos on how to use it are available at:

- <https://community.mendeley.com/guides/videos>

Installation instructions are available at

- <https://www.mendeley.com/features/reference-manager/>

When dealing with large databases, we found the integration of Mendeley into word slow.

4.3.3 Zotero

Zotero is a free tool to help you collect, organize, cite, and share your research sources. Documentation is available at

- <https://www.zotero.org/support/>

The download link is available from

- <https://www.zotero.org/>

We have limited experience with Zotero



5. Editors



chapter/lesson/doc/emacs.tex

5.1 Basic Emacs

One of the most useful short manuals for emacs is the following reference card. It takes some time to use this card efficiently, but the most important commands are written on it. Generations of students have literally been just presented with this card and they learned emacs from it.

- <https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>

There is naturally also additional material available and a great manual. You could also look at

- <https://www.gnu.org/software/emacs/tour/>

From the last page we have summarized the most useful and **simple** features. And present them here. One of the hidden gems of emacs is the ability to recreate replayable macros which we include here also. You ought to try it and you will find that for data science and the cleanup of data emacs (applied to smaller datasets) is a gem.

Notation

Key	Description
C	Control
M	Esc (meta character)

In the event of an emergency...

Here's what to do if you've accidentally pressed a wrong key:

If you executed a command and Emacs has modified your buffer, use C-/ to undo that change. If you pressed a prefix key (e.g. C-x) or you invoked a command which is now prompting you for input (e.g. Find file: ...), type C-g, repeatedly if necessary, to cancel. C-g also cancels a long-running operation if it appears that Emacs has frozen.

Moving around in buffers can be done with cursor keys, or with the following key combinations:

Key	Description
C-f Forw	ard one character
C-n Next	line
C-b Back	one character
C-p Prev	ious line

Here are some ways to move around in larger increments:

Key	Description
C-a Begi	nning of line
M-f Forw	ard one word
M-a Prev	ious sentence
M-v Prev	ious screen
M-< Begi	nning of buffer

Key	Description
C-e	End of line
M-b	Back one word
M-e	Next sentence
C-v	Next screen
M->	End of buffer

You can jump directly to a particular line number in a buffer:

Key	Description
M-g g	Jump to specified line

Searching is easy with the following commands

Key	Description
C-s	Incr emental search forward
C-r	Incr emental search backward

Replace

Key	Description
M-%	Query replace

Killing (“cutting”) text

Key	Description
C-k	Kill line

Yanking

Key	Description
C-y	Yank s last killed text

Macros

Keyboard Macros

Keyboard macros are a way to remember a fixed sequence of keys for later repetition. They’re handy for automating some boring editing tasks.

Key	Description
M-x (Start recording macro
M-x)	Stop recording macro

Key	Description
M-x e	Play back macro once
M-5 M-x-e	Play back macro 5 times

Modes

“Every buffer has an associated major mode, which alters certain behaviors, key bindings, and text display in that buffer. The idea is to customize the appearance and features available based on the contents of the buffer.” modes are typically activated by ending such as .py, .java, .rst, ...

Key	Description
M-x python-mode	Mode for editing Python files
M-x auto-fill-mode	Wraps your lines automatically when they get longer than 70 characters.
M-x flyspell-mode	Highlights misspelled words as you type.



6. Other Formats

 chapter/lesson/doc/rst.tex

6.1 reStructuredText

reStructuredText (RST) pur[pose] is to provide an easy-to-read, what-you-see-is-what-you-get plaintext markup syntax and parser system. With its help you can develop documentation not only for stand alone documentation, simple web pages, an in-line program documentation (such as Python). RST is extensible and new features can be added. It is used in sphinx as one of its supported formats.

6.1.1 Links

- RST Sphinx documentation: <http://www.sphinx-doc.org/en/stable/rest.html>
- RST Syntax: <http://docutils.sourceforge.net/rst.html>
- Important extensions: <http://sphinx-doc.org/ext/todo.html>

Cheatcheat:

- <http://github.com/ralsina/rst-cheatsheet/raw/master/rst-cheatsheet.pdf>
- <http://docutils.sourceforge.net/docs/ref/rst/directives.html>

6.1.2 Source

The source for this page is located at

- <https://raw.githubusercontent.com/cloudmesh/classes/master/docs/source/lesson/doc/rst.rst>

This way you can look at the source on how we create this page.

6.1.3 Sections

with overline, for parts * with overline, for chapters =, for sections -, for subsections ^, for subsubsections ", for paragraphs

RST allows to specify a number of sections. You can do this with the various underlines:

```
*****
Chapter
*****
Section
=====
Subsection
-----
Subsubsection
~~~~~
Paragraph
~~~~~
```

6.1.4 Listtable

```
.. csv-table:: Eye colors
```

```
:header: "Name", "Firstname", "eyes"
:widths: 20, 20, 10

"von Laszewski", "Gregor", "gray"
```

6.1.5 Exceltable

we have integrated Excel table from <http://pythonhosted.org//sphinxcontrib-exceltable/> into our sphinx allowing the definition of more elaborate tables specified in excel. However the most convenient way may be to use list-tables. The documentation to list tables can be found at <http://docutils.sourceforge.net/docs/ref/rst/directives.html#list-table>

6.1.6 Boxes

Seealso

```
.. seealso:: This is a simple **seealso** note.
```

Note

This is a **note** box.

```
.. note:: This is a **note** box.
```

Warning

note the space between the directive and the text

```
.. warning:: note the space between the directive and the text
```

Others

This is an **attention** box.

```
.. attention:: This is an **attention** box.
```

This is a **caution** box.

```
.. caution:: This is a **caution** box.
```

This is a **danger** box.

```
.. danger:: This is a **danger** box.
```

This is a **error** box.

```
.. error:: This is a **error** box.
```

This is a **hint** box.

```
.. hint:: This is a **hint** box.
```

This is an **important** box.

```
.. important:: This is an **important** box.
```

This is a **tip** box.

```
.. tip:: This is a **tip** box.
```

6.1.7 Sidebar directive

It is possible to create sidebar using the following code:

```
.. sidebar:: Sidebar Title
    :subtitle: Optional Sidebar Subtitle

    Subsequent indented lines comprise
    the body of the sidebar, and are
    interpreted as body elements.
```

Sidebar Title: Optional Sidebar Subtitle

Subsequent indented lines comprise the body of the sidebar, and are interpreted as body elements.

6.1.8 Sphinx Prompt

```
.. prompt:: bash, cloudmesh$

    wget -O cm-setup.sh http://bit.ly/cloudmesh-client-xenial
    sh cm-setup.sh
```

6.1.9 Programm examples

You can include code examples and bash commands with two colons.

This is an example for python:

```
print ("Hallo World")
```

This is an example for a shell command:

```
$ ls -lisa
```

6.1.10 Hyperlinks

Direct links to html pages can be done with:

`'This is a link to an html page <hadoop.html>'_`

Note that this page could be generated from an rst page

Links to the FG portal need to be formulated with the portal tag:

`:portal:'List to FG projects </projects/all>'`

In case a subsection has a link declared you can use :ref: (this is the preferred way as it can be used to point even to subsections):

`:ref:'Connecting private network VMs clusters <_s_vpn>'`

A html link can be created anywhere in the document but must be unique. for example if you place:

`.. _s_vpn:`

in the text it will create a target to which the above link points when you click on it

6.1.11 Todo

```
.. todo:: an example
```

Todo

an example

F chapter/lesson/doc/markdown.tex

6.2 Markdown

The content from this section originates from see: <https://en.wikipedia.org/wiki/Markdown>.

Markdown is a simple markup language, however there is no precise standard defined for it and implementations may have features not supported by other implementations. Nevertheless, it provides as simple and easy way to quickly develop clean looking documents.

There are several tools that make markdown attractive allowing to write the text in one window while at the same time seeing the rendered output in another.

This includes

Macdown An editor for markdown targeted on OSX

To convert the markdown to other formats with pandoc

Heading

Sub-heading

Another deeper heading

Paragraphs are separated
by a blank line.

Two spaces at the end of a line leave a
line break.

Text attributes `_italic_`, `*italic*`, `bold`, `**bold**`, ‘monospace’.

Horizontal rule:

Bullet list:

- * apples
- * oranges
- * pears

Numbered list:

1. apples
2. oranges
3. pears

A [link] (<http://example.com>).

F chapter/lesson/doc/type.tex

6.3 Communicating Research in Other Ways

Naturally, writing papers is not the only way to communicate your research with others. We find that today we see additional pathways for communication including blogs, twitter, facebook, e-mail, Web pages, and electronic notebooks. Let us revisit some of them and identify when they are helpful.

6.3.1 Blogs

blog: noun, a regularly updated website or web page, typically one run by an individual or small group, that is written in an informal or conversational style.

Advantages:

- encourages spontaneous posts
- encourages small short contributions
- chronologically ordered
- standard software exists to set up blogs
- online services exists to set up blogs

Disadvantages:

- structuring data is difficult (some blog software support it)
- not suitable for formal development of a paper
- often lack of sophisticated track change features
- no collaborative editing features

6.3.2 Sphinx

Sphinx (<http://www.sphinx-doc.org/>) is a tool that creates integrated documentation from a markup language while.

Advantages:

- output formats: html, LaTeX, PDF, ePub
- integrates well with directory structure
- powerful markup language (reStructuredText)
- can be hosted on GitHub via GitHub pages
- can integrate other renderers such as Markdown
- automatic table of contents, table of index
- code documentation integration
- search
- written in Python and using bash, so extensions and custom automation are possible

Disadvantage:

- requires compile step
- When using markdown GitHub can render individual page

Others:

- Read the Docs (<https://readthedocs.org/>)

- Doxygen (<http://www.stack.nl/~dimitri/doxygen/>)
- MkDocs (<http://www.mkdocs.org/>)

6.3.3 Notebooks

Jupyter

The Jupyter Notebook (<http://jupyter.org/>) is an open-source web application allowing users to create and share documents that contain live code, equations, visualizations and explanatory text. Use cases include data cleaning and transformation, numerical simulation, statistical modeling, machine learning.

Advantages:

- Integrates with python
- Recently other programming languages have been integrated
- Allows experimenting with settings
- Allows a form of literate programming while mixing documentation with code
- automatically renders on github
- comes with web service that allows hosting

Disadvantage:

- mostly encourages short documents
- mark up language is limited
- editing in ASCII is complex and Web editing is preferred

Apache Zeppelin

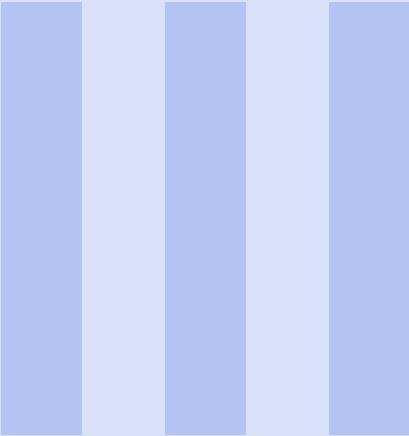
A Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and hadoop. It integrates a web-based notebook with data ingestion, data exploration, visualization, sharing and collaboration features to Hadoop and Spark.

Advantages:

- integration to various framework
- Web framework
- integration with spark, hadoop

Disadvantages:

- larger framework
- must leverages existing deployments of spark, hadoop



Big Data Applications

- 6.4 Introduction
- 6.5 Overview of Data Science
- 6.6 Health Informatics Case Study
- 6.7 e-Commerce and LifeStyle Case Study
- 6.8 Physics Case Study
- 6.9 Radar Case Study
- 6.10 Sensors Case Study
- 6.11 Sports Case Study
- 6.12 Big Data Use Cases Survey
- 6.13 Web Search and Text Mining
- 6.14 Technology Training - kNN & Clustering

 chapter/theory/introduction.tex

6.4 Introduction

You may find that some videos may have a different lesson, section or unit number. Please ignore this. In case the content does not correspond to the title, please let us know.

This section has a technical overview of course followed by a broad motivation for course hosted at www-cloudmesh-classes.

The course overview covers it's content and structure. It presents an introduction to general field of Big Data and Analytics. We are especially analysing the many different application areas in which Big Data can be applied. As Big Data is typically not just used in isolation but is part of a larger Informatics issue for a particular field we also use the term X-Informatics, where X defines a usecase or area of specialization in which Big Data is applied to. As such we organize the class around the the *Rallying Cry* of course: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics.

The courses is set up as a number of lessons that are typically between 20 minutes to an hour. The lessons are either provided as written documents or as video lectures. They are enhanced by an in person meeting that takes place either in a lecture room for residential students or as online meeting for online students.

The course covers a mix of applications (the X in X-Informatics) and technologies needed to support the field electronically i.e. to process the application data. The overview ends with a discussion of course content at highest level. The course starts with a motivation summarizing clouds and data science, then units describing applications in areas such as Physics, e-Commerce, Web Search and Text mining, Health, Sensors and Remote Sensing). These are interspersed with discussions of infrastructure (clouds) and data analytics (algorithms like clustering and collaborative filtering used in applications). The course uses Python as primary programming language. We will be introducing practical use of cloud resources so that you have the oportunity to explore example analytics applications on smaller data sets that you define.

The course motivation starts with striking examples of the data deluge with examples from research, business and the consumer. The growing number of jobs in data science is highlighted. He describes industry trend in both clouds and big data. Then the cloud computing model developed at amazing speed by industry is introduced. The 4 paradigms of scientific research are described with growing importance of data oriented version. He covers 3 major X-informatics areas: Physics, e-Commerce and Web Search followed by a broad discussion of cloud applications. Parallel computing in general and particular features of MapReduce are described.

We discuss in this course include the following topics. We may change the order of the topics to allow for maximal flexibility and parallel learning experiences.

Writing Track:

- Writing a short review article
- Writing a porject or term report

Theory Track:

- Motivation: Big Data and the Cloud; Centerpieces of the Future Economy
- Introduction: What is Big Data, Data Analytics
- Use Cases: Big Data Use Cases Survey

- Use Case, Physics Discovery of Higgs Particle
- Use Case: e-Commerce and Lifestyle with recommender systems
- Use Case: Web Search and Text Mining and their technologies
- Use Case: Sports
- Use Case: Health
- Use Case: Sensors
- Use Case: Radar for Remote Sensing.
- Parallel Computing Overview and familiar examples
- Cloud Technology for Big Data Applications & Analytics

Practice Track:

- Python for Big Data Applications and Analytics: NumPy, SciPy, Matplotlib
- Using FutureGrid for Big Data Applications and Analytics Course
- Using Chameleon Cloud for Big Data Applications and Analytics Course
- [optional] Using Plotviz Software for Displaying Point Distributions in 3D
- Recommender Systems - K-Nearest Neighbors, Clustering and heuristic methods
- PageRank
- Kmeans
- MapReduce
- Kmeans and MapReduce Parallelism

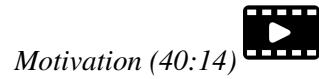
6.4.1 Course Motivation

We motivate the study of X-informatics by describing data science and clouds. He starts with striking examples of the data deluge with examples from research, business and the consumer. The growing number of jobs in data science is highlighted. He describes industry trend in both clouds and big data.

He introduces the cloud computing model developed at amazing speed by industry. The 4 paradigms of scientific research are described with growing importance of data oriented version. He covers 3 major X-informatics areas: Physics, e-Commerce and Web Search followed by a broad discussion of cloud applications. Parallel computing in general and particular features of MapReduce are described. He comments on a data science education and the benefits of using MOOC's.

Emerging Technologies

This presents the overview of talk, some trends in computing and data and jobs. Gartner's emerging technology hype cycle shows many areas of Clouds and Big Data. We highlight 6 issues of importance: economic imperative, computing model, research model, Opportunities in advancing computing, Opportunities in X-Informatics, Data Science Education



Motivation (40:14)

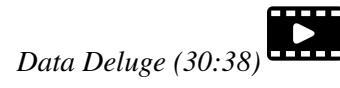


Motivation (30)

Data Deluge

We give some amazing statistics for total storage; uploaded video and uploaded photos; the social media interactions every minute; aspects of the business big data tidal wave; monitors of aircraft

engines; the science research data sizes from particle physics to astronomy and earth science; genes sequenced; and finally the long tail of science. The next slide emphasizes applications using algorithms on clouds. This leads to the rallying cry “Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics educated in data science” with a catalog of the many values of X”Astronomy, Biology, Biomedicine, Business, Chemistry, Climate, Crisis, Earth Science, Energy, Environment, Finance, Health, Intelligence, Lifestyle, Marketing, Medicine, Pathology, Policy, Radar, Security, Sensor, Social, Sustainability, Wealth and Wellness”



Jobs

Jobs abound in clouds and data science. There are documented shortages in data science, computer science and the major tech companies advertise for new talent.



Industrial Trends

Trends include the growing importance of mobile devices and comparative decrease in desktop access, the export of internet content, the change in dominant client operating systems, use of social media, thriving Chinese internet companies.



Digital Disruption of Old Favorites

Not everything goes up. The rise of the Internet has led to declines in some traditional areas including Shopping malls and Postal Services.



Digital Distruption and transformation (32:54)



Digital Distruption and transformation (28)

Computing Model

Industry adopted clouds which are attractive for data analytics

Clouds and Big Data are transformational on a 2-5 year time scale. Already Amazon AWS is a lucrative business with almost a \$4B revenue. We describe the nature of cloud centers with economies of scale and gives examples of importance of virtualization in server consolidation. Then key characteristics of clouds are reviewed with expected high growth in Infrastructure, Platform and Software as a Service.



Computing Model I (24:03)



Computing Model I (14)



Computing Model II (28:18)



Computing Model II (27)

Research Model

4th Paradigm; From Theory to Data driven science?

We introduce the 4 paradigms of scientific research with the focus on the new fourth data driven methodology.



Research Model (7:33)



Research Model (4)

Data Science Process

We introduce the DIKW data to information to knowledge to wisdom paradigm. Data flows through cloud services transforming itself and emerging as new information to input into other transformations.



Data Science Process (15:42)

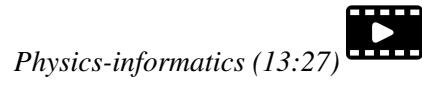


Data Science Process (10)

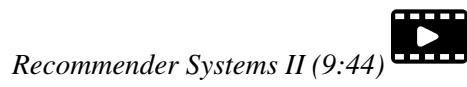
Physics-Informatics

Looking for Higgs Particle with Large Hadron Collider LHC

We look at important particle physics example where the Large hadron Collider has observed the Higgs Boson. He shows this discovery as a bump in a histogram; something that so amazed him 50 years ago that he got a PhD in this field. He left field partly due to the incredible size of author lists on papers.

**Recommender Systems**

Many important applications involve matching users, web pages, jobs, movies, books, events etc. These are all optimization problems with recommender systems one important way of performing this optimization. We go through the example of Netflix ~ everything is a recommendation and muses about the power of viewing all sorts of things as items in a bag or more abstractly some space with funny properties.

**Web Search and Information Retrieval**

This course also looks at Web Search and here we give an overview of the data analytics for web search, Pagerank as a method of ranking web pages returned and uses material from Yahoo on the subtle algorithms for dynamic personalized choice of material for web pages.

**Cloud Application in Research**

We describe scientific applications and how they map onto clouds, supercomputers, grids and high throughput systems. He likes the cloud use of the Internet of Things and gives examples.




Cloud Applications in Research (20)

Parallel Computing and MapReduce

We define MapReduce and gives a homely example from fruit blending.


Computing and MapReduce (14:02)
Computing and MapReduce (9)

Data Science Education

We discuss one reason you are taking this course ~~ Data Science as an educational initiative and aspects of its Indiana University implementation. Then general; features of online education are discussed with clear growth spearheaded by MOOC's where we use this course and others as an example. He stresses the choice between one class to 100,000 students or 2,000 classes to 50 students and an online library of MOOC lessons. In olden days he suggested "hermit's cage virtual university" ~~ gurus in isolated caves putting together exciting curricula outside the traditional university model. Grading and mentoring models and important online tools are discussed. Clouds have MOOC's describing them and MOOC's are stored in clouds; a pleasing symmetry.


Data Science Education (28:08)
Data Science Education (19)

Conclusions

The conclusions highlight clouds, data-intensive methodology, employment, data science, MOOC's and never forget the Big Data ecosystem in one sentence "Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics educated in data science"


Conclusions (4:59)
Conclusions (4)

Resources

- <http://www.gartner.com/technology/home.jsp> and many web links
- Meeker/Wu May 29 2013 Internet Trends D11 Conference <http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013>
- <http://cs.metrostate.edu/~sbd/slides/Sun.pdf>
- Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks Wiley ISBN: 978-1-118-20878-6
- Bill Ruh http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- <http://www.genome.gov/sequencingcosts/>

- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon
- <http://www.microsoft.com/en-us/news/features/2012/mar12/03-05CloudComputingJobs.aspx>
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
- http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
- <http://www.google.com/green/pdfs/google-green-computing.pdf>
- <http://www.wired.com/wired/issue/16-07>
- <http://research.microsoft.com/en-us/collaboration/fourthparadigm/>
- Jeff Hammerbacher <http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf>
- <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>
- <http://www.interactions.org/cms/?pid=1032811>
- <http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg/>
- <http://www.sciencedirect.com/science/article/pii/S037026931200857X>
- <http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-rec>
- http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
- <http://en.wikipedia.org/wiki/PageRank>
- <http://pages.cs.wisc.edu/~beechung/icml11-tutorial/>
- <https://sites.google.com/site/opensourceiotcloud/>
- <http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/>
- <http://blog.coursera.org/post/49750392396/on-the-topic-of-boredom>
- <http://x-informatics.appspot.com/course>
- <http://iucloudsummerschool.appspot.com/preview>
- https://www.youtube.com/watch?v=M3jcSCA9_hM

 chapter/theory/overview.tex

6.5 Overview of Data Science

What is Big Data, Data Analytics and X-Informatics?

The course introduction starts with X-Informatics and its rallying cry. The growing number of jobs in data science is highlighted. The first unit offers a look at the phenomenon described as the Data Deluge starting with its broad features. Data science and the famous DIKW (Data to Information to Knowledge to Wisdom) pipeline are covered. Then more detail is given on the flood of data from Internet and Industry applications with eBay and General Electric discussed in most detail.

In the next unit, we continue the discussion of the data deluge with a focus on scientific research. He takes a first peek at data from the Large Hadron Collider considered later as physics Informatics and gives some biology examples. He discusses the implication of data for the scientific method which is changing with the data-intensive methodology joining observation, theory and simulation as basic methods. Two broad classes of data are the long tail of sciences: many users with individually modest data adding up to a lot; and a myriad of Internet connected devices – the Internet of Things.

We give an initial technical overview of cloud computing as pioneered by companies like Amazon, Google and Microsoft with new centers holding up to a million servers. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing with a comparison to supercomputing. Features of the data deluge are discussed with a salutary example where more data did better than more thought. Then comes Data science and one part of it ~~ data analytics ~~ the large algorithms that crunch the big data to give big wisdom. There are many ways to describe data science and several are discussed to give a good composite picture of this emerging field.

6.5.1 Data Science generics and Commercial Data Deluge

We start with X-Informatics and its rallying cry. The growing number of jobs in data science is highlighted. This unit offers a look at the phenomenon described as the Data Deluge starting with its broad features. Then he discusses data science and the famous DIKW (Data to Information to Knowledge to Wisdom) pipeline. Then more detail is given on the flood of data from Internet and Industry applications with eBay and General Electric discussed in most detail.

TBD (45)



What is X-Informatics and its Motto

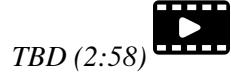
This discusses trends that are driven by and accompany Big data. We give some key terms including data, information, knowledge, wisdom, data analytics and data science. WE introduce the motto of the course: Use Clouds running Data Analytics Collaboratively processing Big Data to solve problems in X-Informatics. We list many values of X you can defined in various activities across the world.

TBD (9:49)

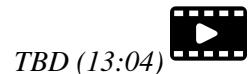


Jobs

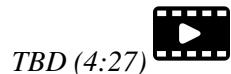
Big data is especially important as there are some many related jobs. We illustrate this for both cloud computing and data science from reports by Microsoft and the McKinsey institute respectively. We show a plot from LinkedIn showing rapid increase in the number of data science and analytics jobs as a function of time.

**Data Deluge: General Structure**

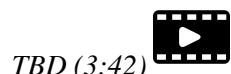
We look at some broad features of the data deluge starting with the size of data in various areas especially in science research. We give examples from real world of the importance of big data and illustrate how it is integrated into an enterprise IT architecture. We give some views as to what characterizes Big data and why data science is a science that is needed to interpret all the data.

**Data Science: Process**

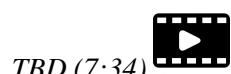
We stress the DIKW pipeline: Data becomes information that becomes knowledge and then wisdom, policy and decisions. This pipeline is illustrated with Google maps and we show how complex the ecosystem of data, transformations (filters) and its derived forms is.

**Data Deluge: Internet**

We give examples of Big data from the Internet with Tweets, uploaded photos and an illustration of the vitality and size of many commodity applications.

**Data Deluge: Business**

We give examples including the Big data that enables wind farms, city transportation, telephone operations, machines with health monitors, the banking, manufacturing and retail industries both online and offline in shopping malls. We give examples from ebay showing how analytics allowing them to refine and improve the customer experiences.



Resources

- <http://www.microsoft.com/en-us/news/features/2012/mar12/03-05CloudComputingJobs.aspx>
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Anjul Bhambhani http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Jeff Hammerbacher <http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf>
- <http://www.economist.com/node/15579717>
- <http://cs.metrostate.edu/~sbd/slides/Sun.pdf>
- <http://jess3.com/geosocial-universe-2/>
- Bill Ruh http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- <http://www.hsph.harvard.edu/ncb2011/files/ncb2011-z03-rodriguez.pptx>
- Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html

6.5.2 Data Deluge and Scientific Applications and Methodology

Overview

We continue the discussion of the data deluge with a focus on scientific research. He takes a first peek at data from the Large Hadron Collider considered later as physics Informatics and gives some biology examples. He discusses the implication of data for the scientific method which is changing with the data-intensive methodology joining observation, theory and simulation as basic methods. We discuss the long tail of sciences; many users with individually modest data adding up to a lot. The last lesson emphasizes how everyday devices ~~ the Internet of Things ~~ are being used to create a wealth of data.



TBD (22) PDF

Science & Research

We look into more big data examples with a focus on science and research. We give astronomy, genomics, radiology, particle physics and discovery of Higgs particle (Covered in more detail in later lessons), European Bioinformatics Institute and contrast to Facebook and Walmart.



TBD (11:27)



TBD (11:49)

Implications for Scientific Method

We discuss the emergence of a new fourth methodology for scientific research based on data driven inquiry. We contrast this with third ~~ computation or simulation based discovery - methodology which emerged itself some 25 years ago.



TBD (5:07)

Long Tail of Science

There is big science such as particle physics where a single experiment has 3000 people collaborate!.Then there are individual investigators who don't generate a lot of data each but together they add up to Big data.



TBD (2:10)

Internet of Things

A final category of Big data comes from the Internet of Things where lots of small devices ~~ smart phones, web cams, video games collect and disseminate data and are controlled and coordinated in the cloud.



TBD (5:45)

Resources

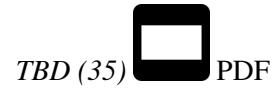
- <http://www.economist.com/node/15579717>
- Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceedings of HPC 2012 Conference at Cetraro, Italy June 28 2012
- http://grids.ucs.indiana.edu/ptliupages/publications/Clouds_Technical_Computing_FoxGannonv2.pdf
- <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>
- <http://www.genome.gov/sequencingcosts/>
- <http://www.quantumdiaries.org/2012/09/07/why-particle-detectors-need-a-trigger/atlasmgg>
- <http://salsahpc.indiana.edu/dlib/articles/00001935/>
- http://en.wikipedia.org/wiki/Simple_linear_regression
- <http://www.ebi.ac.uk/Information/Brochures/>
- <http://www.wired.com/wired/issue/16-07>
- <http://research.microsoft.com/en-us/collaboration/fourthparadigm/>
- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon <https://sites.google.com/site/opensourceiotcloud/>

6.5.3 Clouds and Big Data Processing; Data Science Process and Analytics

Overview

We give an initial technical overview of cloud computing as pioneered by companies like Amazon, Google and Microsoft with new centers holding up to a million servers. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing with a comparison to supercomputing.

He discusses features of the data deluge with a salutary example where more data did better than more thought. He introduces data science and one part of it ~~ data analytics ~~ the large algorithms that crunch the big data to give big wisdom. There are many ways to describe data science and several are discussed to give a good composite picture of this emerging field.



TBD (35) PDF

6.5.4 Clouds

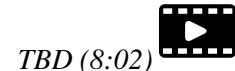
We describe cloud data centers with their staggering size with up to a million servers in a single data center and centers built modularly from shipping containers full of racks. The benefits of Clouds in terms of power consumption and the environment are also touched upon, followed by a list of the most critical features of Cloud computing and a comparison to supercomputing.



TBD (16:04) MP4

Features of Data Deluge I

Data, Information, intelligence algorithms, infrastructure, data structure, semantics and knowledge are related. The semantic web and Big data are compared. We give an example where “More data usually beats better algorithms”. We discuss examples of intelligent big data and list 8 different types of data deluge



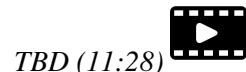
TBD (8:02)



TBD (6:24)

Data Science Process

We describe and critique one view of the work of a data scientist. Then we discuss and contrast 7 views of the process needed to speed data through the DIKW pipeline.



TBD (11:28)

Data Analytics



TBD (30)

We stress the importance of data analytics giving examples from several fields. We note that better analytics is as important as better computing and storage capability. In the second video we look at High Performance Computing in Science and Engineering: the Tree and the Fruit.



TBD (7:28)



TBD (6:51)

Resources

- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of

Science October 2012 Dennis Gannon

- Dan Reed Roger Barga Dennis Gannon Rich Wolski http://research.microsoft.com/en-us/people/barga/sc09_cloud.pdf
- <http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-power-use-of-data-centers/>
- <http://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data-centers/>
- <http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf>
- Bina Ramamurthy <http://www.cse.buffalo.edu/~bina/cse487/fall2011/>
- Jeff Hammerbacher <http://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf>
- Jeff Hammerbacher <http://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf>
- Anjul Bhambhani http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- <http://cs.metrostate.edu/~sbd/slides/Sun.pdf>
- Hugh Williams http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Tom Davenport http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- http://www.mckinsey.com/mgi/publications/big_data/index.asp
- <http://cra.org/ccc/docs/nitrdsymposium/pdfs/keyes.pdf>

 chapter/theory/health.tex

6.6 Health Informatics Case Study

This section starts by discussing general aspects of Big Data and Health including data sizes, different areas including genomics, EBI, radiology and the Quantified Self movement. We review current state of health care and trends associated with it including increased use of Telemedicine. We summarize an industry survey by GE and Accenture and an impressive exemplar Cloud-based medicine system from Potsdam. We give some details of big data in medicine. Some remarks on Cloud computing and Health focus on security and privacy issues.

We survey an April 2013 McKinsey report on the Big Data revolution in US health care; a Microsoft report in this area and a European Union report on how Big Data will allow patient centered care in the future. Examples are given of the Internet of Things, which will have great impact on health including wearables. A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative. The final topic is Genomics, Proteomics and Information Visualization.

6.6.1 X-Informatics Case Study: Health Informatics

Overview

131 (Health)



This section starts by discussing general aspects of Big Data and Health including data sizes, different areas including genomics, EBI, radiology and the Quantified Self movement. We review current state of health care and trends associated with it including increased use of Telemedicine. We summarize an industry survey by GE and Accenture and an impressive exemplar Cloud-based medicine system from Potsdam. We give some details of big data in medicine. Some remarks on Cloud computing and Health focus on security and privacy issues.

We survey an April 2013 McKinsey report on the Big Data revolution in US health care; a Microsoft report in this area and a European Union report on how Big Data will allow patient centered care in the future. Examples are given of the Internet of Things, which will have great impact on health including wearables. A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative. The final topic is Genomics, Proteomics and Information Visualization.

Big Data and Health

This lesson starts with general aspects of Big Data and Health including listing subareas where Big data important. Data sizes are given in radiology, genomics, personalized medicine, and the Quantified Self movement, with sizes and access to European Bioinformatics Institute.

Big Data and Health (10:02)

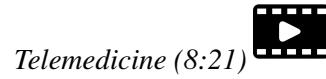


Status of Healthcare Today

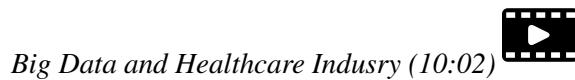
This covers trends of costs and type of healthcare with low cost genomes and an aging population. Social media and government Brain initiative.

*Status of Healthcare Today (16:09)***Telemedicine (Virtual Health)**

This describes increasing use of telemedicine and how we tried and failed to do this in 1994.

*Telemedicine (8:21)***Big Data and Healthcare Industry**

Summary of an industry survey by GE and Accenture.

*Big Data and Healthcare Indusry (10:02)***Medical Big Data in the Clouds**

An impressive exemplar Cloud-based medicine system from Potsdam.

*Medical Big Data in the Clouds (15:02)***Medical image Big Data***Midical Image Big Data (6:33)***Clouds and Health***Clouds and Health (4:35)***McKinsey Report on the big-data revolution in US health care**

This lesson covers 9 aspects of the McKinsey report. These are the convergence of multiple positive changes has created a tipping point for innovation; Primary data pools are at the heart of the big data revolution in healthcare; Big data is changing the paradigm: these are the value pathways; Applying early successes at scale could reduce US healthcare costs by \$300 billion to \$450 billion; Most new big-data applications target consumers and providers across pathways; Innovations are weighted towards influencing individual decision-making levers; Big data innovations use a range of public, acquired, and proprietary data types; Organizations implementing a big data transformation should provide the leadership required for the associated cultural transformation; Companies must develop a range of big data capabilities.

*McKinsey Report (14:53)***Microsoft Report on Big Data in Health**

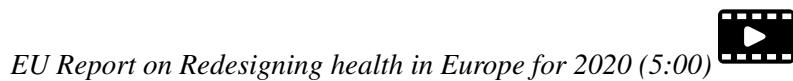
This lesson identifies data sources as Clinical Data, Pharma & Life Science Data, Patient & Consumer Data, Claims & Cost Data and Correlational Data. Three approaches are Live data feed,

Advanced analytics and Social analytics.



EU Report on Redesigning health in Europe for 2020

This lesson summarizes an EU Report on Redesigning health in Europe for 2020. The power of data is seen as a lever for change in My Data, My decisions; Liberate the data; Connect up everything; Revolutionize health; and Include Everyone removing the current correlation between health and wealth.



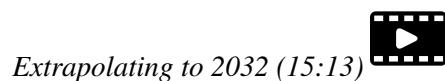
Medicine and the Internet of Things

The Internet of Things will have great impact on health including telemedicine and wearables. Examples are given.



Extrapolating to 2032

A study looks at 4 scenarios for healthcare in 2032. Two are positive, one middle of the road and one negative.



Genomics, Proteomics and Information Visualization

A study of an Azure application with an Excel frontend and a cloud BLAST backend starts this lesson. This is followed by a big data analysis of personal genomics and an analysis of a typical DNA sequencing analytics pipeline. The Protein Sequence Universe is defined and used to motivate Multi dimensional Scaling MDS. Sammon's method is defined and its use illustrated by a metagenomics example. Subtleties in use of MDS include a monotonic mapping of the dissimilarity function. The application to the COG Proteomics dataset is discussed. We note that the MDS approach is related to the well known chisq method and some aspects of nonlinear minimization of chisq (Least Squares) are discussed.



Next we continue the discussion of the COG Protein Universe introduced in the last lesson. It is shown how Proteomics clusters are clearly seen in the Universe browser. This motivates a side remark on different clustering methods applied to metagenomics. Then we discuss the Generative Topographic Map GTM method that can be used in dimension reduction when original data is in a metric space and is in this case faster than MDS as GTM computational complexity scales like N

not N squared as seen in MDS.

Examples are given of GTM including an application to topic models in Information Retrieval. Indiana University has developed a deterministic annealing improvement of GTM. 3 separate clusterings are projected for visualization and show very different structure emphasizing the importance of visualizing results of data analytics. The final slide shows an application of MDS to generate and visualize phylogenetic trees.

 *Genomics, Proteomics and Information Visualization I (10:33)*

 *Genomics, Proteomics and Information Visualization: II (7:41)*

 *131 (Proteomics and Information Visualization)*

Resources

- <https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives>
- <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>
- <http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%202020100923.pdf>
- <http://quantifiedself.com/larry-smarr/>
- <http://www.ebi.ac.uk/Information/Brochures/>
- <http://www.kpcb.com/internet-trends>
- <http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quan>
- <http://www.siam.org/meetings/sdm13/sun.pdf>
- http://en.wikipedia.org/wiki/Calico_%28company%29
- http://www.slideshare.net/GSW_Worldwide/2015-health-trends
- <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Cha>
- <http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-p>
- <http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-health>
- http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
- <http://www.mckinsey.com/~media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolution%20in%20US%20health%20care/The%20big-data%20revolution%20in%20US%20health%20care%20Accelerating%20value%20and%20innovation.ashx>
- <https://partner.microsoft.com/download/global/40193764>
- http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_health-eu-for2020-ehtf-report2012.pdf
- <http://www.kpcb.com/internet-trends>
- <http://www.liveathos.com/apparel/app>
- <http://debategraph.org/Poster.aspx?aID=77>
- <http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon>
- <http://www.delsall.org>
- http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
- <http://www.geatbx.com/docu/fcnindex-01.html>
- <https://wiki.nci.nih.gov/display/CIP/CIP+Survey+of+Biomedical+Imaging+Archives>
- <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>

- 20the%20data%20come%20from%20v7.pdf
- <http://www.ieee-icsc.org/ICSC2010/Tony%20Hey%20-%2020100923.pdf>
 - <http://quantifiedself.com/larry-smarr/>
 - <http://www.ebi.ac.uk/Information/Brochures/>
 - <http://www.kpcb.com/internet-trends>
 - <http://www.slideshare.net/drsteventucker/wearable-health-fitness-trackers-and-the-quan>
 - <http://www.siam.org/meetings/sdm13/sun.pdf>
 - http://en.wikipedia.org/wiki/Calico_%28company%29
 - http://www.slideshare.net/GSW_Worldwide/2015-health-trends
 - <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Cha>
 - <http://www.slideshare.net/schappy/how-realtime-analysis-turns-big-medical-data-into-p>
 - <http://medcitynews.com/2013/03/the-body-in-bytes-medical-images-as-a-source-of-health>
 - http://healthinformatics.wikispaces.com/file/view/cloud_computing.ppt
 - <http://www.mckinsey.com/~media/McKinsey/dotcom/Insights/Health%20care/The%20big-data%20revolution%20in%20US%20health%20care/The%20big-data%20revolution%20in%20US%20health%20care%20Accelerating%20value%20and%20innovation.ashx>
 - <https://partner.microsoft.com/download/global/40193764>
 - http://ec.europa.eu/information_society/activities/health/docs/policy/taskforce/redesigning_health-eu-for2020-ehtf-report2012.pdf
 - <http://www.kpcb.com/internet-trends>
 - <http://www.liveathos.com/apparel/app>
 - <http://debategraph.org/Poster.aspx?aID=77>
 - <http://www.oerc.ox.ac.uk/downloads/presentations-from-events/microsoftworkshop/gannon>
 - <http://www.delsall.org>
 - http://salsahpc.indiana.edu/millionseq/mina/16SrRNA_index.html
 - <http://www.geatbx.com/docu/fcnindex-01.html>

F chapter/theory/lifestyle.tex

6.7 e-Commerce and LifeStyle Case Study

Recommender systems operate under the hood of such widely recognized sites as Amazon, eBay, Monster and Netflix where everything is a recommendation. This involves a symbiotic relationship between vendor and buyer whereby the buyer provides the vendor with information about their preferences, while the vendor then offers recommendations tailored to match their needs. Kaggle competitions help improve the success of the Netflix and other recommender systems. Attention is paid to models that are used to compare how changes to the systems affect their overall performance. It is interesting that the humble ranking has become such a dominant driver of the world's economy. More examples of recommender systems are given from Google News, Retail stores and in depth Yahoo! covering the multi-faceted criteria used in deciding recommendations on web sites.

The formulation of recommendations in terms of points in a space or bag is given where bags of item properties, user properties, rankings and users are useful. Detail is given on basic principles behind recommender systems: user-based collaborative filtering, which uses similarities in user rankings to predict their interests, and the Pearson correlation, used to statistically quantify correlations between users viewed as points in a space of items. Items are viewed as points in a space of users in item-based collaborative filtering. The Cosine Similarity is introduced, the difference between implicit and explicit ratings and the k Nearest Neighbors algorithm. General features like the curse of dimensionality in high dimensions are discussed. A simple Python k Nearest Neighbor code and its application to an artificial data set in 3 dimensions is given. Results are visualized in Matplotlib in 2D and with Plotviz in 3D. The concept of a training and a testing set are introduced with training set pre labeled. Recommender system are used to discuss clustering with k-means based clustering methods used and their results examined in Plotviz. The original labelling is compared to clustering results and extension to 28 clusters given. General issues in clustering are discussed including local optima, the use of annealing to avoid this and value of heuristic algorithms.

6.7.1 Recommender Systems: Introduction

We introduce Recommender systems as an optimization technology used in a variety of applications and contexts online. They operate in the background of such widely recognized sites as Amazon, eBay, Monster and Netflix where everything is a recommendation. This involves a symbiotic relationship between vendor and buyer whereby the buyer provides the vendor with information about their preferences, while the vendor then offers recommendations tailored to match their needs, to the benefit of both.

There follows an exploration of the Kaggle competition site, other recommender systems and Netflix, as well as competitions held to improve the success of the Netflix recommender system. Finally attention is paid to models that are used to compare how changes to the systems affect their overall performance. It is interesting how the humble ranking has become such a dominant driver of the world's economy.

Recommender Systems as an Optimization Problem

We define a set of general recommender systems as matching of items to people or perhaps collections of items to collections of people where items can be other people, products in a store, movies, jobs, events, web pages etc. We present this as “yet another optimization problem”.



Recommender Systems I (8:06)

Recommender Systems Introduction

We give a general discussion of recommender systems and point out that they are particularly valuable in long tail of items (to be recommended) that aren't commonly known. We pose them as a rating system and relate them to information retrieval rating systems. We can contrast recommender systems based on user profile and context; the most familiar collaborative filtering of others ranking; item properties; knowledge and hybrid cases mixing some or all of these.



Recommender Systems Introduction (12:56)

Kaggle Competitions

We look at Kaggle competitions with examples from web site. In particular we discuss an Irvine class project involving ranking jokes.



Kaggle Competitions: (3:36)

Examples of Recommender Systems

We go through a list of 9 recommender systems from the same Irvine class.



Examples of Recommender Systems (1:00)

Netflix on Recommender Systems

This is Part 1.

We summarize some interesting points from a tutorial from Netflix for whom “everything is a recommendation”. Rankings are given in multiple categories and categories that reflect user interests are especially important. Criteria used include explicit user preferences, implicit based on ratings and hybrid methods as well as freshness and diversity. Netflix tries to explain the rationale of its recommendations. We give some data on Netflix operations and some methods used in its recommender systems. We describe the famous Netflix Kaggle competition to improve its rating system. The analogy to maximizing click through rate is given and the objectives of optimization are given.



Netflix on Recommender Systems (14:20)

Consumer Data Science

Here we go through Netflix's methodology in letting data speak for itself in optimizing the recommender engine. An example is given on choosing self produced movies. A/B testing is discussed with examples showing how testing does allow optimizing of sophisticated criteria. This lesson is concluded by comments on Netflix technology and the full spectrum of issues that are involved including user interface, data, AB testing, systems and architectures. We comment on optimizing for a household rather than optimizing for individuals in household.



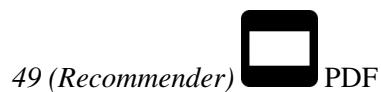
Resources

- <http://www.slideshare.net/xamat/building-largescale-realworld-recommender-systems-rec>
- http://www.ifi.uzh.ch/ce/teaching/spring2012/16-Recommender-Systems_Slides.pdf
- <https://www.kaggle.com/>
- http://www.ics.uci.edu/~welling/teaching/CS77Bwinter12/CS77B_w12.html
- Jeff Hammerbacher <https://berkeleydatascience.files.wordpress.com/2012/01/20120117berkeley1.pdf>
- <http://www.techworld.com/news/apps/netflix-foretells-house-of-cards-success-with-cass>
- https://en.wikipedia.org/wiki/A/B_testing
- <http://www.infoq.com/presentations/Netflix-Architecture>

6.7.2 Recommender Systems: Examples and Algorithms

We continue the discussion of recommender systems and their use in e-commerce. More examples are given from Google News, Retail stores and in depth Yahoo! covering the multi-faceted criteria used in deciding recommendations on web sites. Then the formulation of recommendations in terms of points in a space or bag is given.

Here bags of item properties, user properties, rankings and users are useful. Then we go into detail on basic principles behind recommender systems: user-based collaborative filtering, which uses similarities in user rankings to predict their interests, and the Pearson correlation, used to statistically quantify correlations between users viewed as points in a space of items.



Recap and Examples of Recommender Systems

We start with a quick recap of recommender systems from previous unit; what they are with brief examples.



Examples of Recommender Systems

We give 2 examples in more detail: namely Google News and Markdown in Retail.



Recommender Systems in Yahoo Use Case Example

We describe in greatest detail the methods used to optimize Yahoo web sites. There are two lessons discussing general approach and a third lesson examines a particular personalized Yahoo page with its different components. We point out the different criteria that must be blended in making decisions; these criteria include analysis of what user does after a particular page is clicked; is the user satisfied and cannot that we quantified by purchase decisions etc. We need to choose Articles, ads, modules, movies, users, updates, etc to optimize metrics such as relevance score, CTR, revenue, engagement. These lesson stress that if though we have big data, the recommender data is sparse. We discuss the approach that involves both batch (offline) and on-line (real time) components.



User-based nearest-neighbor collaborative filtering

Collaborative filtering is a core approach to recommender systems. There is user-based and item-based collaborative filtering and here we discuss the user-based case. Here similarities in user rankings allow one to predict their interests, and typically this quantified by the Pearson correlation, used to statistically quantify correlations between users.



Vector Space Formulation of Recommender Systems

We go through recommender systems thinking of them as formulated in a funny vector space. This suggests using clustering to make recommendations.



Resources

- <http://pages.cs.wisc.edu/~beechung/icml11-tutorial/>

6.7.3 Item-based Collaborative Filtering and its Technologies

We move on to item-based collaborative filtering where items are viewed as points in a space of users. The Cosine Similarity is introduced, the difference between implicit and explicit ratings

and the k Nearest Neighbors algorithm. General features like the curse of dimensionality in high dimensions are discussed.

18 (*Filtering*)  PDF

Item-based Collaborative Filtering

We covered user-based collaborative filtering in the previous unit. Here we start by discussing memory-based real time and model based offline (batch) approaches. Now we look at item-based collaborative filtering where items are viewed in the space of users and the cosine measure is used to quantify distances. WE discuss optimizations and how batch processing can help. We discuss different Likert ranking scales and issues with new items that do not have a significant number of rankings.

 *Item Based Filtering (11:18)*

 *k Nearest Neighbors and High Dimensional Spaces (7:16)*

k Nearest Neighbors and High Dimensional Spaces

We define the k Nearest Neighbor algorithms and present the Python software but do not use it. We give examples from Wikipedia and describe performance issues. This algorithm illustrates the curse of dimensionality. If items were a real vectors in a low dimension space, there would be faster solution methods.

 *k Nearest Neighbors and High Dimensional Spaces (10:03)*

 chapter/theory/physics.tex

6.8 Physics Case Study

This section starts by describing the LHC accelerator at CERN and evidence found by the experiments suggesting existence of a Higgs Boson. The huge number of authors on a paper, remarks on histograms and Feynman diagrams is followed by an accelerator picture gallery. The next unit is devoted to Python experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals. Then random variables and some simple principles of statistics are introduced with explanation as to why they are relevant to Physics counting experiments. The unit introduces Gaussian (normal) distributions and explains why they seen so often in natural phenomena. Several Python illustrations are given. Random Numbers with their Generators and Seeds lead to a discussion of Binomial and Poisson Distribution. Monte-Carlo and accept-reject methods. The Central Limit Theorem concludes discussion.

6.8.1 Looking for Higgs Particles, Bumps in Histograms, Experiments and Accelerators (Part 1)

This unit is devoted to Python and Java experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals. The lectures use Python but use of Java is described.

 20 (Higgs)

Files: HiggsClassI-Sloping.py </files/python/physics/mr_higgs/higgs_classI_sloping.py>_

Looking for Higgs Particle and Counting Introduction

We return to particle case with slides used in introduction and stress that particles often manifested as bumps in histograms and those bumps need to be large enough to stand out from background in a statistically significant fashion.

 Discovery of Higgs Particle (13:49)

We give a few details on one LHC experiment ATLAS. Experimental physics papers have a staggering number of authors and quite big budgets. Feynman diagrams describe processes in a fundamental fashion.

 Looking for Higgs Particle and Counting Introduction II (7:38)

Physics-Informatics Looking for Higgs Particle Experiments

We give a few details on one LHC experiment ATLAS. Experimental physics papers have a staggering number of authors and quite big budgets. Feynman diagrams describe processes in a fundamental fashion.

 Looking for Higgs Particle Experiments (9:29)

Accelerator Picture Gallery of Big Science

This lesson gives a small picture gallery of accelerators. Accelerators, detection chambers and magnets in tunnels and a large underground laboratory used fpr experiments where you need to be shielded from background like cosmic rays.



Accelerator Picture Gallery of Big Science (11:21)

Resources

- <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>
- <http://www.sciencedirect.com/science/article/pii/S037026931200857X>
- <http://www.nature.com/news/specials/lhc/interactive.html>

Looking for Higgs Particles: Python Event Counting for Signal and Background (Part 2)

This unit is devoted to Python experiments looking at histograms of Higgs Boson production with various forms of shape of signal and various background and with various event totals.



29 (Higgs II) PDF

Files:

- HiggsClassI-Sloping.py </files/python/physics/mr_higgs/higgs_classI_sloping.py>
- HiggsClassIII.py </files/python/physics/number_theory/higgs_classIII.py>
- HiggsClassIIUniform.py </files/python/physics/mr_higgs/higgs_classII_uniform.py>

Physics Use Case II 1: Class Software

We discuss how this unit uses Java and Python on both a backend server (FutureGrid) or a local client. WE point out useful book on Python for data analysis. This builds on technology training in Section 3.



Higgs Particle Events and Counting (9:30)

This video contains Java information, but we are no longer using Java in this class.

Physics Use Case II 2: Event Counting

We define “event counting” data collection environments. We discuss the python and Java code to generate events according to a particular scenario (the important idea of Monte Carlo data). Here a sloping background plus either a Higgs particle generated similarly to LHC observation or one observed with better resolution (smaller measurement error).



Event Counting (7:02)

Physics Use Case II 3: With Python examples of Signal plus Background

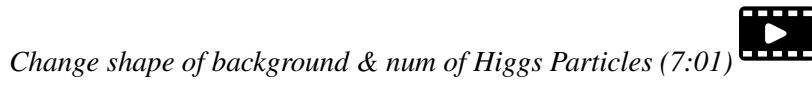
This uses Monte Carlo data both to generate data like the experimental observations and explore effect of changing amount of data and changing measurement resolution for Higgs.



With Python examples of Signal plus Background (7:33)

Physics Use Case II 4: Change shape of background & num of Higgs Particles

This lesson continues the examination of Monte Carlo data looking at effect of change in number of Higgs particles produced and in change in shape of background.



Change shape of background & num of Higgs Particles (7:01)

Resources

- Python for Data Analysis: Agile Tools for Real World Data By Wes McKinney, Publisher: O'Reilly Media, Released: October 2012, Pages: 472.
- <http://jwork.org/scavis/api/>
- <https://en.wikipedia.org/wiki/DataMelt>

6.8.2 Looking for Higgs Particles: Random Variables, Physics and Normal Distributions

We introduce random variables and some simple principles of statistics and explains why they are relevant to Physics counting experiments. The unit introduces Gaussian (normal) distributions and explains why they seen so often in natural phenomena. Several Python illustrations are given. Java is currently not available in this unit.

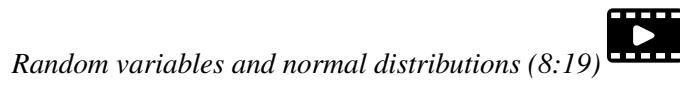


39 (Higgs)

HiggsClassIII.py </files/python/physics/number_theory/higgs_classIII.py>

Statistics Overview and Fundamental Idea: Random Variables

We go through the many different areas of statistics covered in the Physics unit. We define the statistics concept of a random variable.



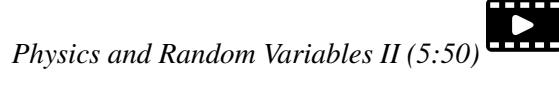
Random variables and normal distributions (8:19)

Physics and Random Variables

We describe the DIKW pipeline for the analysis of this type of physics experiment and go through details of analysis pipeline for the LHC ATLAS experiment. We give examples of event displays showing the final state particles seen in a few events. We illustrate how physicists decide what's going on with a plot of expected Higgs production experimental cross sections (probabilities) for signal and background.



Physics and Random Variables I (8:34)



Physics and Random Variables II (5:50)

Statistics of Events with Normal Distributions

We introduce Poisson and Binomial distributions and define independent identically distributed (IID) random variables. We give the law of large numbers defining the errors in counting and leading to Gaussian distributions for many things. We demonstrate this in Python experiments.

*Statistics of Events with Normal Distributions (11:25)***Gaussian Distributions**

We introduce the Gaussian distribution and give Python examples of the fluctuations in counting Gaussian distributions.

*Gaussian Distributions (9:08)***Using Statistics**

We discuss the significance of a standard deviation and role of biases and insufficient statistics with a Python example in getting incorrect answers.

*Using Statistics (14:02)***Resources**

- <http://indico.cern.ch/event/20453/session/6/contribution/15?materialId=slides>
- <http://www.atlas.ch/photos/events.html>
- <https://cms.cern/>

6.8.3 Looking for Higgs Particles: Random Numbers, Distributions and Central Limit Theorem (Part 3)

We discuss Random Numbers with their Generators and Seeds. It introduces Binomial and Poisson Distribution. Monte-Carlo and accept-reject methods are discussed. The Central Limit Theorem and Bayes law concludes discussion. Python and Java (for student - not reviewed in class) examples and Physics applications are given.

*44 (Higgs III) PDF*

Files:

- HiggsClassIII.py </files/python/physics/calculated_dice_roll/higgs_classIV_seeds.py>

Generators and Seeds

We define random numbers and describe how to generate them on the computer giving Python examples. We define the seed used to define to specify how to start generation.

*Higgs Particle Counting Errors (6:28)*



Generators and Seeds II (7:10)

Binomial Distribution

We define binomial distribution and give LHC data as an example of where this distribution valid.



Binomial Distribution: (12:38)

Accept-Reject

We introduce an advanced method **accept/reject** for generating random variables with arbitrary distributions.



Accept-Reject (5:54)

Monte Carlo Method

We define Monte Carlo method which usually uses accept/reject method in typical case for distribution.



Monte Carlo Method (2:23)

Poisson Distribution

We extend the Binomial to the Poisson distribution and give a set of amusing examples from Wikipedia.



Poisson Distribution (4:37)

Central Limit Theorem

We introduce Central Limit Theorem and give examples from Wikipedia.



Central Limit Theorem (4:47)

Interpretation of Probability: Bayes v. Frequency

This lesson describes difference between Bayes and frequency views of probability. Bayes's law of conditional probability is derived and applied to Higgs example to enable information about Higgs from multiple channels and multiple experiments to be accumulated.



Interpretation of Probability (12:39)

Resources

..bibliography:: physics-references.bib

 chapter/theory/radar.tex

6.9 Radar Case Study

The changing global climate is suspected to have long-term effects on much of the world's inhabitants. Among the various effects, the rising sea level will directly affect many people living in low-lying coastal regions. While the ocean's thermal expansion has been the dominant contributor to rises in sea level, the potential contribution of discharges from the polar ice sheets in Greenland and Antarctica may provide a more significant threat due to the unpredictable response to the changing climate. The Radar-Informatics unit provides a glimpse in the processes fueling global climate change and explains what methods are used for ice data acquisitions and analysis.

58 (Radar)



6.9.1 Introduction

This lesson motivates radar-informatics by building on previous discussions on why X-applications are growing in data size and why analytics are necessary for acquiring knowledge from large data. The lesson details three mosaics of a changing Greenland ice sheet and provides a concise overview to subsequent lessons by detailing explaining how other remote sensing technologies, such as the radar, can be used to sound the polar ice sheets and what we are doing with radar images to extract knowledge to be incorporated into numerical models.

Radar Informatics (3:31)



6.9.2 Remote Sensing

This lesson explains the basics of remote sensing, the characteristics of remote sensors and remote sensing applications. Emphasis is on image acquisition and data collection in the electromagnetic spectrum.

Remote Sensing (6:43)



6.9.3 Ice Sheet Science

This lesson provides a brief understanding on why melt water at the base of the ice sheet can be detrimental and why it's important for sensors to sound the bedrock.

Ice Sheet Science (1:00)



6.9.4 Global Climate Change

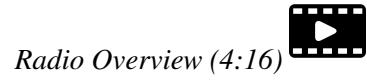
This lesson provides an understanding and the processes for the greenhouse effect, how warming effects the Polar Regions, and the implications of a rise in sea level.



Global Climate Change (2:51)

6.9.5 Radio Overview

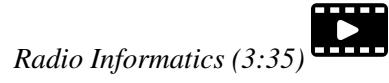
This lesson provides an elementary introduction to radar and its importance to remote sensing, especially to acquiring information about Greenland and Antarctica.



Radio Overview (4:16)

6.9.6 Radio Informatics

This lesson focuses on the use of sophisticated computer vision algorithms, such as active contours and a hidden markov model to support data analysis for extracting layers, so ice sheet models can accurately forecast future changes in climate.

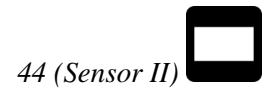
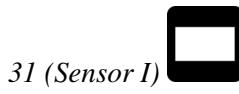


Radio Informatics (3:35)

 chapter/theory/sensor.tex

6.10 Sensors Case Study

We start with the Internet of Things IoT giving examples like monitors of machine operation, QR codes, surveillance cameras, scientific sensors, drones and self driving cars and more generally transportation systems. We give examples of robots and drones. We introduce the Industrial Internet of Things IIoT and summarize surveys and expectations Industry wide. We give examples from General Electric. Sensor clouds control the many small distributed devices of IoT and IIoT. More detail is given for radar data gathered by sensors; ubiquitous or smart cities and homes including U-Korea; and finally the smart electric grid.



6.10.1 Internet of Things

There are predicted to be 24-50 Billion devices on the Internet by 2020; these are typically some sort of sensor defined as any source or sink of time series data. Sensors include smartphones, webcams, monitors of machine operation, barcodes, surveillance cameras, scientific sensors (especially in earth and environmental science), drones and self driving cars and more generally transportation systems. The lesson gives many examples of distributed sensors, which form a Grid that is controlled by a cloud.



6.10.2 Robotics and IOT Expectations

Examples of Robots and Drones.



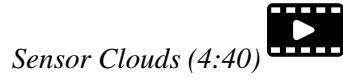
6.10.3 Industrial Internet of Things

We summarize surveys and expectations Industry wide.



6.10.4 Sensor Clouds

We describe the architecture of a Sensor Cloud control environment and gives example of interface to an older version of it. The performance of system is measured in terms of processing latency as a function of number of involved sensors with each delivering data at 1.8 Mbps rate.



6.10.5 Earth/Environment/Polar Science data gathered by Sensors

This lesson gives examples of some sensors in the Earth/Environment/Polar Science field. It starts with material from the CReSIS polar remote sensing project and then looks at the NSF Ocean Observing Initiative and NASA's MODIS or Moderate Resolution Imaging Spectroradiometer instrument on a satellite.



Earth/Environment/Polar Science data gathered by Sensors (4:58)

6.10.6 Ubiquitous/Smart Cities

For Ubiquitous/Smart cities we give two examples: Iniquitous Korea and smart electrical grids.



Ubiquitous/Smart Cities (1:44)

6.10.7 U-Korea (U=Ubiquitous)

Korea has an interesting positioning where it is first worldwide in broadband access per capita, e-government, scientific literacy and total working hours. However it is far down in measures like quality of life and GDP. U-Korea aims to improve the latter by Pervasive computing, everywhere, anytime i.e. by spreading sensors everywhere. The example of a 'High-Tech Utopia' New Songdo is given.



U-Korea (U=Ubiquitous) (2:49)

6.10.8 Smart Grid

The electrical Smart Grid aims to enhance USA's aging electrical infrastructure by pervasive deployment of sensors and the integration of their measurement in a cloud or equivalent server infrastructure. A variety of new instruments include smart meters, power monitors, and measures of solar irradiance, wind speed, and temperature. One goal is autonomous local power units where good use is made of waste heat.



Smart Grid (6:04)

6.10.9 Resources

- <https://www.gesoftware.com/minds-and-machines>
- <https://www.gesoftware.com/predix>
- <https://www.gesoftware.com/sites/default/files/the-industrial-internet/index.html>
- <https://developer.cisco.com/site/eiot/discover/overview/>
- <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Industrial-Internet-Char.pdf>
- <http://www.gesoftware.com/ge-predictivity-infographic>
- <http://www.gettransportation.com/railconnect360/rail-landscape>

- <http://www.gesoftware.com/sites/default/files/GE-Software-Modernizing-Machine-to-Mach.pdf>

 chapter/theory/sport.tex

6.11 Sports Case Study

Sports sees significant growth in analytics with pervasive statistics shifting to more sophisticated measures. We start with baseball as game is built around segments dominated by individuals where detailed (video/image) achievement measures including PITCHf/x and FIELDf/x are moving field into big data arena. There are interesting relationships between the economics of sports and big data analytics. We look at Wearables and consumer sports/recreation. The importance of spatial visualization is discussed. We look at other Sports: Soccer, Olympics, NFL Football, Basketball, Tennis and Horse Racing.

6.11.1 Sports Informatics I : Sabermetrics (Basic)

Unit Overview

This unit discusses baseball starting with the movie Moneyball and the 2002-2003 Oakland Athletics. Unlike sports like basketball and soccer, most baseball action is built around individuals often interacting in pairs. This is much easier to quantify than many player phenomena in other sports. We discuss Performance-Dollar relationship including new stadiums and media/advertising. We look at classic baseball averages and sophisticated measures like Wins Above Replacement.


40 (Overview)

Introduction and Sabermetrics (Baseball Informatics) Lesson

Introduction to all Sports Informatics, Moneyball The 2002-2003 Oakland Athletics, Diamond Dollars economic model of baseball, Performance - Dollar relationship, Value of a Win.


Introduction and Sabermetrics (Baseball Informatics) Lesson (31:4)

Basic Sabermetrics

Different Types of Baseball Data, Sabermetrics, Overview of all data, Details of some statistics based on basic data, OPS, wOBA, ERA, ERC, FIP, UZR.


Basic Sabermetrics (26:53)

Wins Above Replacement

Wins above Replacement WAR, Discussion of Calculation, Examples, Comparisons of different methods, Coefficient of Determination, Another, Sabermetrics Example, Summary of Sabermetrics.


Wins Above Replacement (30:43)

Resources

- <http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-power>

- <http://www.sloansportsconference.com/>
- <http://sabr.org/>
- <http://en.wikipedia.org/wiki/Sabermetrics>
- http://en.wikipedia.org/wiki/Baseball_statistics
- <http://www.sportvision.com/baseball>
- <http://m.mlb.com/news/article/68514514/mlbam-introduces-new-way-to-analyze-every-play>
- <http://www.fangraphs.com/library/offense/offensive-statistics-list/>
- http://en.wikipedia.org/wiki/Component_ERA
- <http://www.fangraphs.com/library/pitching/fip/>
- <http://nomaas.org/2012/05/a-look-at-the-defense-the-yankees-d-stinks-edition/>
- http://en.wikipedia.org/wiki/Wins_Above_Replacement
- <http://www.fangraphs.com/library/misc/war/>
- http://www.baseball-reference.com/about/war_explained.shtml
- http://www.baseball-reference.com/about/war_explained_comparison.shtml
- http://www.baseball-reference.com/about/war_explained_position.shtml
- http://www.baseball-reference.com/about/war_explained_pitch.shtml
- <http://www.fangraphs.com/leaders.aspx?pos=all&stats=bat&lg=all&qual=y&type=8&season=2014&month=0&season1=1871&ind=0>
- <http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-ii/>
- <http://battingleadoff.com/2014/01/08/comparing-the-three-war-measures-part-iii/>
- http://en.wikipedia.org/wiki/Coefficient_of_determination
- http://www.sloansportsconference.com/wp-content/uploads/2014/02/2014_SSAC_Data-driven-Method-for-In-game-Decision-Making.pdf
- <https://courses.edx.org/courses/BUx/SABR101x/2T2014/courseware/10e616fc7649469ab4457ae>

6.11.2 Sports Informatics II : Sabermetrics (Advanced)

This unit discusses ‘advanced sabermetrics’ covering advances possible from using video from PITCHf/X, FIELDf/X, HITf/X, COMMANDf/X and MLBAM.

41 (Sporta II)

Pitching Clustering

A Big Data Pitcher Clustering method introduced by Vince Gennaro, Data from Blog and video at 2013 SABR conference.

Pitching Clustering (20:59)

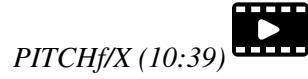
Pitcher Quality

Results of optimizing match ups, Data from video at 2013 SABR conference.

Pitcher Quality (10:02)

6.11.3 PITCHf/X

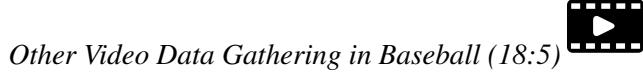
Examples of use of PITCHf/X.



PITCHf/X (10:39)

Other Video Data Gathering in Baseball

FIELDf/X, MLBAM, HITf/X, COMMANDf/X.



Other Video Data Gathering in Baseball (18:5)

Resources

- <http://vincegennaro.mlblogs.com/>
- https://www.youtube.com/watch?v=H-kx-x_d0Mk
- <http://www.sportvision.com/media/pitchfx-how-it-works>
- <http://www.baseballprospectus.com/article.php?articleid=13109>
- <http://baseball.physics.illinois.edu/FastPFXGuide.pdf>
- <http://baseball.physics.illinois.edu/FieldFX-TDR-GregR.pdf>
- <http://www.sportvision.com/baseball/fieldfx>
- <http://regressing.deadspin.com/mlb-announces-revolutionary-new-fielding-tracking-syst>
- <http://grantland.com/the-triangle/mlb-advanced-media-play-tracking-bob-bowman-intervi>
- <http://www.sportvision.com/baseball/hitfx>
- <https://www.youtube.com/watch?v=YkjtnuNmK74>

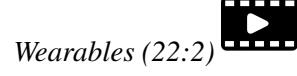
6.11.4 Sports Informatics III : Other Sports

We look at Wearables and consumer sports/recreation. The importance of spatial visualization is discussed. We look at other Sports: Soccer, Olympics, NFL Football, Basketball, Tennis and Horse Racing.

44 (Sports III)

Wearables

Consumer Sports, Stake Holders, and Multiple Factors.



Wearables (22:2)

Soccer and the Olympics

Soccer, Tracking Players and Balls, Olympics.



Soccer and the Olympics (8:28)

Spatial Visualization in NFL and NBA

NFL, NBA, and Spatial Visualization.



Spatial Visualization in NFL and NBA (15:19)

Tennis and Horse Racing

Tennis, Horse Racing, and Continued Emphasis on Spatial Visualization.

Video8:52Tennis and Horse Racinghttps://www.youtube.com/watch?v=2P-pismFSrI

Resources

- http://www.sloansportsconference.com/?page_id=481&sort_cate=Research%20Paper
- http://www.slideshare.net/Tricon_Infotech/big-data-for-big-sports
- <http://www.slideshare.net/BrandEmotivity/sports-analytics-innovation-summit-data-power>
- <http://www.liveathos.com/apparel/app>
- <http://www.slideshare.net/elew/sport-analytics-innovation>
- <http://www.wired.com/2013/02/catapult-smartball/>
- http://www.sloansportsconference.com/wp-content/uploads/2014/06/Automated_Playbook_Generation.pdf
- <http://autoscout.adsc.illinois.edu/publications/football-trajectory-dataset/>
- http://www.sloansportsconference.com/wp-content/uploads/2012/02/Goldsberry_Sloan_Submission.pdf
- <http://gamesetmap.com/>
- <http://www.trakus.com/technology.asp#tNetText>

 chapter/theory/usecases.tex

6.12 Big Data Use Cases Survey

This section covers 51 values of X and an overall study of Big data that emerged from a NIST (National Institute for Standards and Technology) study of Big data. The section covers the NIST Big Data Public Working Group (NBD-PWG) Process and summarizes the work of five subgroups: Definitions and Taxonomies Subgroup, Reference Architecture Subgroup, Security and Privacy Subgroup, Technology Roadmap Subgroup and the Requirements and Use Case Subgroup. 51 use cases collected in this process are briefly discussed with a classification of the source of parallelism and the high and low level computational structure. We describe the key features of this classification.

6.12.1 Overview of NIST Big Data Public Working Group (NBD-PWG) Process and Results

This unit covers the NIST Big Data Public Working Group (NBD-PWG) Process and summarizes the work of five subgroups: Definitions and Taxonomies Subgroup, Reference Architecture Subgroup, Security and Privacy Subgroup, Technology Roadmap Subgroup and the Requirements and Use Case Subgroup. The work of latter is continued in next two units.

 45 (Overview)

Introduction to NIST Big Data Public Working Group (NBD-PWG) Process

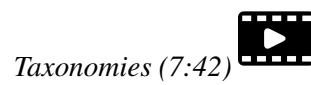
The focus of the (NBD-PWG) is to form a community of interest from industry, academia, and government, with the goal of developing a consensus definitions, taxonomies, secure reference architectures, and technology roadmap. The aim is to create vendor-neutral, technology and infrastructure agnostic deliverables to enable big data stakeholders to pick-and-choose best analytics tools for their processing and visualization requirements on the most suitable computing platforms and clusters while allowing value-added from big data service providers and flow of data between the stakeholders in a cohesive and secure manner.

 Introduction (13:02)

Definitions and Taxonomies Subgroup

The focus is to gain a better understanding of the principles of Big Data. It is important to develop a consensus-based common language and vocabulary terms used in Big Data across stakeholders from industry, academia, and government. In addition, it is also critical to identify essential actors with roles and responsibility, and subdivide them into components and sub-components on how they interact/ relate with each other according to their similarities and differences.

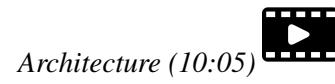
For Definitions: Compile terms used from all stakeholders regarding the meaning of Big Data from various standard bodies, domain applications, and diversified operational environments. For Taxonomies: Identify key actors with their roles and responsibilities from all stakeholders, categorize them into components and subcomponents based on their similarities and differences. In particular data Science and Big Data terms are discussed.



Taxonomies (7:42)

Reference Architecture Subgroup

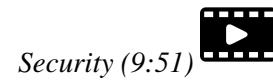
The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus-based approach to orchestrate vendor-neutral, technology and infrastructure agnostic for analytics tools and computing environments. The goal is to enable Big Data stakeholders to pick-and-choose technology-agnostic analytics tools for processing and visualization in any computing platform and cluster while allowing value-added from Big Data service providers and the flow of the data between the stakeholders in a cohesive and secure manner. Results include a reference architecture with well defined components and linkage as well as several exemplars.



Architecture (10:05)

Security and Privacy Subgroup

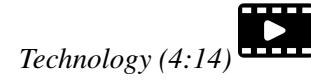
The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus secure reference architecture to handle security and privacy issues across all stakeholders. This includes gaining an understanding of what standards are available or under development, as well as identifies which key organizations are working on these standards. The Top Ten Big Data Security and Privacy Challenges from the CSA (Cloud Security Alliance) BDWG are studied. Specialized use cases include Retail/Marketing, Modern Day Consumerism, Nielsen Homescan, Web Traffic Analysis, Healthcare, Health Information Exchange, Genetic Privacy, Pharma Clinical Trial Data Sharing, Cyber-security, Government, Military and Education.



Security (9:51)

Technology Roadmap Subgroup

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus vision with recommendations on how Big Data should move forward by performing a good gap analysis through the materials gathered from all other NBD subgroups. This includes setting standardization and adoption priorities through an understanding of what standards are available or under development as part of the recommendations. Tasks are gather input from NBD subgroups and study the taxonomies for the actors' roles and responsibility, use cases and requirements, and secure reference architecture; gain understanding of what standards are available or under development for Big Data; perform a thorough gap analysis and document the findings; identify what possible barriers may delay or prevent adoption of Big Data; and document vision and recommendations.



Technology (4:14)

Interfaces subgroup

This subgroup is working on the following document: *NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interface*.

This document summarizes interfaces that are instrumental for the interaction with Clouds, Containers, and HPC systems to manage virtual clusters to support the NIST Big Data Reference Architecture (NBDRA). The Representational State Transfer (REST) paradigm is used to define these interfaces allowing easy integration and adoption by a wide variety of frameworks. . This volume, Volume 8, uses the work performed by the NBD-PWG to identify objects instrumental for the NIST Big Data Reference Architecture (NBDRA) which is introduced in the NBDIF: Volume 6, Reference Architecture.

This presentation was given at the *2nd NIST Big Data Public Working Group (NBD-PWG) Workshop* in Washington DC in June 2017. It explains our thoughts on deriving automatically a reference architecture from the Reference Architecture Interface specifications directly from the document.

The workshop Web page is located at

- <https://bigdatawg.nist.gov/workshop2.php>

The agenda of the workshop is as follows:

- https://bigdatawg.nist.gov/2017_NIST_Big_Data_PWG_WorkshopAgenda_with_Speakers_Bio.pdf

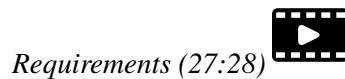
The Webcast of the presentation is given below, while you need to fast forward to a particular time

- Webcast: Interface subgroup: <https://www.nist.gov/news-events/events/2017/06/2nd-nist-big-data-public-working-group-nbd-pwg-workshop>
 - see: Big Data Working Group Day 1, part 2 Time start: 21:00 min, Time end: 44:00
- Slides: <https://github.com/cloudmesh/cloudmesh.rest/blob/master/docs/NBDPWG-vol8.pptx?raw=true>
- Document: <https://github.com/cloudmesh/cloudmesh.rest/raw/master/docs/NIST.SP.1500-8-draft.pdf>

You are welcome to view other presentations if you are interested.

Requirements and Use Case Subgroup Introduction

The focus is to form a community of interest from industry, academia, and government, with the goal of developing a consensus list of Big Data requirements across all stakeholders. This includes gathering and understanding various use cases from diversified application domains. Tasks are to gather use case input from all stakeholders; derive Big Data requirements from each use case; analyze/prioritize a list of challenging general requirements that may delay or prevent adoption of Big Data deployment; develop a set of general patterns capturing the “essence” of use cases (not done yet) and work with Reference Architecture to validate requirements and reference architecture by explicitly implementing some patterns based on use cases. The progress of gathering use cases (discussed in next two units) and requirements systemization are discussed.



6.12.2 51 Big Data Use Cases

This unit consists of one or more slides for each of the 51 use cases - typically additional (more than one) slides are associated with pictures. Each of the use cases is identified with source of parallelism and the high and low level computational structure. As each new classification topic is introduced we briefly discuss it but full discussion of topics is given in following unit.


100 (51)

Government Use Cases

This covers Census 2010 and 2000 - Title 13 Big Data; National Archives and Records Administration Accession NARA, Search, Retrieve, Preservation; Statistical Survey Response Improvement (Adaptive Design) and Non-Traditional Data in Statistical Survey Response Improvement (Adaptive Design).



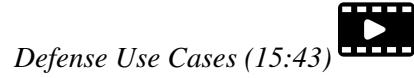
Commercial Use Cases

This covers Cloud Eco-System, for Financial Industries (Banking, Securities & Investments, Insurance) transacting business within the United States; Mendeley - An International Network of Research; Netflix Movie Service; Web Search; IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System; Cargo Shipping; Materials Data for Manufacturing and Simulation driven Materials Genomics.



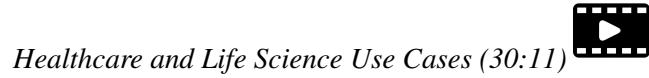
Defense Use Cases

This covers Large Scale Geospatial Analysis and Visualization; Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance and Intelligence Data Processing and Analysis.



Healthcare and Life Science Use Cases

This covers Electronic Medical Record (EMR) Data; Pathology Imaging/digital pathology; Computational Bioimaging; Genomic Measurements; Comparative analysis for metagenomes and genomes; Individualized Diabetes Management; Statistical Relational Artificial Intelligence for Health Care; World Population Scale Epidemiological Study; Social Contagion Modeling for Planning, Public Health and Disaster Management and Biodiversity and LifeWatch.



Deep Learning and Social Networks Use Cases

This covers Large-scale Deep Learning; Organizing large-scale, unstructured collections of consumer photos;Truthy: Information diffusion research from Twitter Data; Crowd Sourcing in the Humanities as Source for Bigand Dynamic Data; CINET: Cyberinfrastructure for Network (Graph) Science and Analytics and NIST Information Access Division analytic technology performance measurement, evaluations, and standards.


*Deep Learning and Social Networks Use Cases (14:19)***Research Ecosystem Use Cases**

DataNet Federation Consortium DFC; The ‘Discinnet process’, metadata -big data global experiment; Semantic Graph-search on Scientific Chemical and Text-based Data and Light source beamlines.


*Research Ecosystem Use Cases (9:09)***Astronomy and Physics Use Cases**

This covers Catalina Real-Time Transient Survey (CRTS): a digital, panoramic, synoptic sky survey; DOE Extreme Data from Cosmological Sky Survey and Simulations; Large Survey Data for Cosmology; Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle and Belle II High Energy Physics Experiment.


*Astronomy and Physics Use Cases (17:33)***Environment, Earth and Polar Science Use Cases**

EISCAT 3D incoherent scatter radar system; ENVRI, Common Operations of Environmental Research Infrastructure; Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets; UAVSAR Data Processing, DataProduct Delivery, and Data Services; NASA LARC/GSFC iRODS Federation Testbed; MERRA Analytic Services MERRA/AS; Atmospheric Turbulence - Event Discovery and Predictive Analytics; Climate Studies using the Community Earth System Model at DOE’s NERSC center; DOE-BER Subsurface Biogeochemistry Scientific Focus Area and DOE-BER AmeriFlux and FLUXNET Networks.


*Environment, Earth and Polar Science Use Cases (25:29)***Energy Use Case**

This covers Consumption forecasting in Smart Grids.


Energy Use Case (4:01)

6.12.3 Features of 51 Big Data Use Cases

This unit discusses the categories used to classify the 51 use-cases. These categories include concepts used for parallelism and low and high level computational structure. The first lesson is an introduction to all categories and the further lessons give details of particular categories.

Summary of Use Case Classification I

This discusses concepts used for parallelism and low and high level computational structure. Parallelism can be over People (users or subjects), Decision makers; Items such as Images, EMR, Sequences; observations, contents of online store; Sensors – Internet of Things; Events; (Complex) Nodes in a Graph; Simple nodes as in a learning network; Tweets, Blogs, Documents, Web Pages etc.; Files or data to be backed up, moved or assigned metadata; Particles/cells/mesh points. Low level computational types include PP (Pleasingly Parallel); MR (MapReduce); MRStat; MRIter (Iterative MapReduce); Graph; Fusion; MC (Monte Carlo) and Streaming. High level computational types include Classification; S/Q (Search and Query); Index; CF (Collaborative Filtering); ML (Machine Learning); EGO (Large Scale Optimizations); EM (Expectation maximization); GIS; HPC; Agents. Patterns include Classic Database; NoSQL; Basic processing of data as in backup or metadata; GIS; Host of Sensors processed on demand; Pleasingly parallel processing; HPC assimilated with observational data; Agent-based models; Multi-modal data fusion or Knowledge Management; Crowd Sourcing.



Summary of Use Case Classification (23:39)

Database(SQL) Use Case Classification

This discusses classic (SQL) database approach to data handling with Search&Query and Index features. Comparisons are made to NoSQL approaches.



Database (SQL) Use Case Classification (11:13)

NoSQL Use Case Classification

This discusses NoSQL (compared in previous lesson) with HDFS, Hadoop and Hbase. The Apache Big data stack is introduced and further details of comparison with SQL.



NoSQL Use Case Classification (11:20)

Use Case Classifications I

This discusses a subset of use case features: GIS, Sensors. the support of data analysis and fusion by streaming data between filters.



Use Case Classifications I (12:42)

Use Case Classifications II

This discusses a subset of use case features: Pleasingly parallel, MRStat, Data Assimilation, Crowd sourcing, Agents, data fusion and agents, EGO and security.



Use Case Classifications II (20:18)

Use Case Classifications III

This discusses a subset of use case features: Classification, Monte Carlo, Streaming, PP, MR, MRStat, MRIter and HPC(MPI), global and local analytics (machine learning), parallel computing, Expectation Maximization, graphs and Collaborative Filtering.



Use Case Classifications III (17:25)

Resources

- NIST Big Data Public Working Group (NBD-PWG) Process <https://www.nist.gov/el/cyber-physical-systems/big-data-pwg>
- Big Data Definitions: <http://dx.doi.org/10.6028/NIST.SP.1500-1> (link is external)
- Big Data Taxonomies: <http://dx.doi.org/10.6028/NIST.SP.1500-2> (link is external)
- Big Data Use Cases and Requirements: <http://dx.doi.org/10.6028/NIST.SP.1500-3> (link is external)
- Big Data Security and Privacy: <http://dx.doi.org/10.6028/NIST.SP.1500-4> (link is external)
- Big Data Architecture White Paper Survey: <http://dx.doi.org/10.6028/NIST.SP.1500-5> (link is external)
- Big Data Reference Architecture: <http://dx.doi.org/10.6028/NIST.SP.1500-6> (link is external)
- Big Data Standards Roadmap: <http://dx.doi.org/10.6028/NIST.SP.1500-7> (link is external)

Some of the links bellow may be outdated. Please let us know the new links and notify us of the outdated links.

- DCGSA Standard Cloud: <https://www.youtube.com/watch?v=14Qii7T8zeg>
- On line 51 Use Cases <http://bigdatawg.nist.gov/usecases.php>
- Summary of Requirements Subgroup http://bigdatawg.nist.gov/_uploadfiles/M0245_v5_6066621242.docx
- Use Case 6 Mendeley <http://mendeley.com%20http://dev.mendeley.com>
- Use Case 7 Netflix <http://www.slideshare.net/xamat/building-largescale-realworld-recommendations>
- Use Case 8 Search http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013, http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html, http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws, http://www.slideshare.net/beechung/recommender-systems-tutorialpart1intro, http://www.worldwidewebsize.com/
- Use Case 9 IaaS (Infrastructure as a Service) Big Data Business Continuity & Disaster Recovery (BC/DR) Within A Cloud Eco-System provided by Cloud Service Providers (CSPs) and Cloud Brokerage Service Providers (CBSPs) <http://www.disasterrecovery.org/>
- Use Case 11 and Use Case 12 Simulation driven Materials Genomics <https://www.materialsproject.org/>
- Use Case 13 Large Scale Geospatial Analysis and Visualization <http://www.opengeospatial.org/standards, http://geojson.org/, http://earth-info.nga.mil/publications/specs/printed/CADRG/cadrg.html>
- Use Case 14 Object identification and tracking from Wide Area Large Format Imagery (WALF) Imagery or Full Motion Video (FMV) - Persistent Surveillance <http://www.militaryaerospace.com/topics/m/video/79088650/persistent-surveillance-relies-on-extraction.htm, http://www.defencetalk.com/wide-area-persistent-surveillance-revolutionizes-tactical-situational-awareness>

- Use Case 15 Intelligence Data Processing and Analysis http://www.afcea-aberdeen.org/files/presentations/AFCEAAberdeen_DCGSA_COLWells_PS.pdf, http://stids.c4i.gmu.edu/papers/STIDSPapers/STIDS2012/_T14/_SmithEtAl/_HorizontalIntegrationOfWarfi.pdf, http://stids.c4i.gmu.edu/STIDS2011/papers/STIDS2011_CR_T1_SalmenEtAl.pdf, <https://www.youtube.com/watch?v=14Qii7T8zeg>, <http://dcgsa.apg.army.mil/>
- Use Case 16 Electronic Medical Record (EMR) Data: Regenstrief Institute, Logical observation identifiers names and codes, Indiana Health Information Exchange, Institute of Medicine Learning Healthcare System
- Use Case 17 Pathology Imaging/digital pathology; <https://web.cci.emory.edu/confluence/display/PAIS> , <https://web.cci.emory.edu/confluence/display/HadoopGIS>
- Use Case 19 Genome in a Bottle Consortium: www.genomeinabottle.org
- Use Case 20 Comparative analysis for metagenomes and genomes <http://img.jgi.doe.gov/>
- Use Case 25 Biodiversity and LifeWatch
- Use Case 26 Deep Learning: Recent popular press coverage of deep learning technology: <http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a.html> , <http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence.html> , http://www.wired.com/2013/06/andrew_ng/, A recent research paper on HPC for Deep Learning: http://www.stanford.edu/~acoates/papers/CoatesHuvalWangWuNgCatanzaro_icml2013.pdf, Widely-used tutorials and references for Deep Learning: http://ufldl.stanford.edu/wiki/index.php/Main_Page, <http://deeplearning.net/>
- Use Case 27 Organizing large-scale, unstructured collections of consumer photos <http://vision.soic.indiana.edu/projects/disco/>
- Use Case 28 Truthy: Information diffusion research from Twitter Data <http://truthy.indiana.edu/> , <http://cnets.indiana.edu/groups/nan/truthy/> , <http://cnets.indiana.edu/groups/nan/despic/>
- Use Case 30 CINET: Cyberinfrastructure for Network (Graph) Science and Analytics http://cinet.vbi.vt.edu/cinet_new/
- Use Case 31 NIST Information Access Division analytic technology performance measurement, evaluations, and standards <http://www.nist.gov/itl/iad/>
- Use Case 32 DataNet Federation Consortium DFC: The DataNet Federation Consortium, iRODS
- Use Case 33 The ‘Discinnet process’, metadata < - > big data global experiment <http://www.discinnet.org/>
- Use Case 34 Semantic Graph-search on Scientific Chemical and Text-based Data http://www.eurekalert.org/pub_releases/2013-07/aiop-ffm071813.php , <http://xpdb.nist.gov/chemblast/pdb.pl>
- Use Case 35 Light source beamlines <http://www-als.lbl.gov/> , <https://www1.aps.anl.gov/>
- Use Case 36 CRTS survey, CSS survey ; For an overview of the classification challenges, see, e.g., <http://arxiv.org/abs/1209.1681>
- Use Case 37 DOE Extreme Data from Cosmological Sky Survey and Simulations <http://www.lsst.org/lssst/> , <http://www.nersc.gov/> , <http://www.nersc.gov/assets/Uploads/HabibcosmosimV2.pdf>
- Use Case 38 Large Survey Data for Cosmology <http://desi.lbl.gov/> , <http://www.darkenergysurvey.org/>
- Use Case 39 Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf> , http://www.es.net/assets/pubs_presos/High-throughput-lessons-from-the-LHC-experience.Johnston.TNC2013.pdf

- Use Case 40 Belle II High Energy Physics Experiment <http://belle2.kek.jp/>
- Use Case 41 EISCAT 3D incoherent scatter radar system <https://www.eiscat3d.se/>
- Use Case 42 ENVRI, Common Operations of Environmental Research Infrastructure, ENVRI Project website, ENVRI Reference Model, ENVRI deliverable D3.2 : Analysis of common requirements of Environmental Research Infrastructures, ICOS, Euro-Argo, EISCAT 3D, LifeWatch, EPOS, EMSO
- Use Case 43 Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets <https://www.cresis.ku.edu/>
- Use Case 44 UAVSAR Data Processing, Data Product Delivery, and Data Services <http://uavasar.jpl.nasa.gov/>, <http://www.asf.alaska.edu/program/sdc>, <http://geo-gateway.org/main.html>
- Use Case 47 Atmospheric Turbulence - Event Discovery and Predictive Analytics <http://oceanworld.tamu.edu/resources/oceanography-book/teleconnections.htm>, <http://www.forbes.com/sites/toddwoody/2012/03/21/meet-the-scientists-mining-big-data-to-pr>
- Use Case 48 Climate Studies using the Community Earth System Model at DOE's NERSC center <http://www-pcmdi.llnl.gov/>, <http://www.nersc.gov/>, <http://science.energy.gov/ber/research/cesd/>, <http://www2.cisl.ucar.edu/>
- Use Case 50 DOE-BER AmeriFlux and FLUXNET Networks <http://ameriflux.lbl.gov/>, <http://www.fluxdata.org/default.aspx>
- Use Case 51 Consumption forecasting in Smart Grids <http://smartgrid.usc.edu/>, http://ganges.usc.edu/wiki/Smart_Grid,
https://www.ladwp.com/ladwp/faces/ladwp/aboutus/a-power/a-p-smartgridla?_afrLoop=157401916661989&_afrWindowMode=0&_afrWindowId=null#%40%3F_afrWindowId%3Dnull%26_afrLoop%3D157401916661989%26_afrWindowMode%3D0%26_adf.ctrl-state%3Db7yulr4rl_17, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6475927>

 chapter/theory/web.tex

6.13 Web Search and Text Mining

This section starts with an overview of data mining and puts our study of classification, clustering and exploration methods in context. We examine the problem to be solved in web and text search and note the relevance of history with libraries, catalogs and concordances. An overview of web search is given describing the continued evolution of search engines and the relation to the field of Information.

The importance of recall, precision and diversity is discussed. The important Bag of Words model is introduced and both Boolean queries and the more general fuzzy indices. The important vector space model and revisiting the Cosine Similarity as a distance in this bag follows. The basic TF-IDF approach is discussed. Relevance is discussed with a probabilistic model while the distinction between Bayesian and frequency views of probability distribution completes this unit.

We start with an overview of the different steps (data analytics) in web search and then goes key steps in detail starting with document preparation. An inverted index is described and then how it is prepared for web search. The Boolean and Vector Space approach to query processing follow. This is followed by Link Structure Analysis including Hubs, Authorities and PageRank. The application of PageRank ideas as reputation outside web search is covered. The web graph structure, crawling it and issues in web advertising and search follow. The use of clustering and topic models completes the section.

6.13.1 Web Search and Text Mining I

The unit starts with the web with its size, shape (coming from the mutual linkage of pages by URL's) and universal power laws for number of pages with particular number of URL's linking out or in to page. Information retrieval is introduced and compared to web search. A comparison is given between semantic searches as in databases and the full text search that is base of Web search. The origin of web search in libraries, catalogs and concordances is summarized. DIKW – Data Information Knowledge Wisdom – model for web search is discussed. Then features of documents, collections and the important Bag of Words representation. Queries are presented in context of an Information Retrieval architecture. The method of judging quality of results including recall, precision and diversity is described. A time line for evolution of search engines is given.

Boolean and Vector Space models for query including the cosine similarity are introduced. Web Crawlers are discussed and then the steps needed to analyze data from Web and produce a set of terms. Building and accessing an inverted index is followed by the importance of term specificity and how it is captured in TF-IDF. We note how frequencies are converted into belief and relevance.

56 (Web Search and Text Mining) 

6.13.2 Web and Document/Text Search: The Problem

 Text Mining (9:56)

This lesson starts with the web with its size, shape (coming from the mutual linkage of pages by

URL's) and universal power laws for number of pages with particular number of URL's linking out or in to page.

6.13.3 Information Retrieval leading to Web Search



Information Retrieval (6:06)

Information retrieval is introduced A comparison is given between semantic searches as in databases and the full text search that is base of Web search. The ACM classification illustrates potential complexity of ontologies. Some differences between web search and information retrieval are given.

6.13.4 History behind Web Search



Web Search History (5:48)

The origin of web search in libraries, catalogs and concordances is summarized.

6.13.5 Key Fundamental Principles behind Web Search



Principles (9:30)

This lesson describes the DIKW – Data Information Knowledge Wisdom – model for web search. Then it discusses documents, collections and the important Bag of Words representation.

6.13.6 Information Retrieval (Web Search) Components



Fundametal Principals of Web Search (5:06)

This describes queries in context of an Information Retrieval architecture. The method of judging quality of results including recall, precision and diversity is described.

6.13.7 Search Engines



Search Engines (3:08)

This short lesson describes a time line for evolution of search engines. The first web search approaches were directly built on Information retrieval but in 1998 the field was changed when Google was founded and showed the importance of URL structure as exemplified by PageRank.

6.13.8 Boolean and Vector Space Models

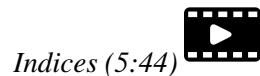


Boolean and Vector Space Model (6:17)

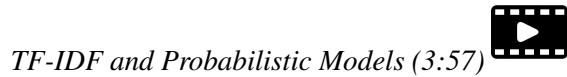
This lesson describes the Boolean and Vector Space models for query including the cosine similarity.

6.13.9 Web crawling and Document Preparation*Web crawling and Document Preparation (4:55)*

This describes a Web Crawler and then the steps needed to analyze data from Web and produce a set of terms.

6.13.10 Indices*Indices (5:44)*

This lesson describes both building and accessing an inverted index. It describes how phrases are treated and gives details of query structure from some early logs.

6.13.11 TF-IDF and Probabilistic Models*TF-IDF and Probabilistic Models (3:57)*

It describes the importance of term specificity and how it is captured in TF-IDF. It notes how frequencies are converted into belief and relevance.

6.13.12 Resources

- http://saedsayad.com/data_mining_map.htm
- http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
- The Web Graph: an Overview: www.youtube.com/watch?v=yPFi6xFnDHE&feature=youtu.be
Jean-Loup Guillaume and Matthieu Latapy <https://hal.archives-ouvertes.fr/file/index/docid/54458/filename/webgraph.pdf>
- Constructing a reliable Web graph with information on browsing behavior, Yiqun Liu, Yufei Xue, Danqing Xu, Rongwei Cen, Min Zhang, Shaoping Ma, Liyun Ru www.sciencedirect.com/science/article/pii/S0167923612001844
- <http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws>

6.13.13 Web Search and Text Mining II*33 (Text Mining)*

PDF

We start with an overview of the different steps (data analytics) in web search. This is followed by Link Structure Analysis including Hubs, Authorities and PageRank. The application of PageRank ideas as reputation outside web search is covered. Issues in web advertising and search follow. This leads to emerging field of computational advertising. The use of clustering and topic models completes unit with Google News as an example.

6.13.14 Data Analytics for Web Search*Web Search and Text Mining II (6:11)*

This short lesson describes the different steps needed in web search including: Get the digital data (from web or from scanning); Crawl web; Preprocess data to get searchable things (words, positions); Form Inverted Index mapping words to documents; Rank relevance of documents with potentially sophisticated techniques; and integrate technology to support advertising and ways to allow or stop pages artificially enhancing relevance.

6.13.15 Link Structure Analysis including PageRank*Realated Applications (17:24)*

The value of links and the concepts of Hubs and Authorities are discussed. This leads to definition of PageRank with examples. Extensions of PageRank viewed as a reputation are discussed with journal rankings and university department rankings as examples. There are many extension of these ideas which are not discussed here although topic models are covered briefly in a later lesson.

6.13.16 Web Advertising and Search*Web Advertising and Search (9:02)*

Internet and mobile advertising is growing fast and can be personalized more than for traditional media. There are several advertising types Sponsored search, Contextual ads, Display ads and different models: Cost per viewing, cost per clicking and cost per action. This leads to emerging field of computational advertising.

6.13.17 Clustering and Topic Models*Clustering and Topic Models (6:21)*

We discuss briefly approaches to defining groups of documents. We illustrate this for Google News and give an example that this can give different answers from word-based analyses. We mention some work at Indiana University on a Latent Semantic Indexing model.

6.13.18 Resources

- <http://www.ifis.cs.tu-bs.de/teaching/ss-11/irws>
- <https://en.wikipedia.org/wiki/PageRank>
- http://webcourse.cs.technion.ac.il/236621/Winter2011-2012/en/ho_Lectures.html
- Meeker/Wu May 29 2013 Internet Trends D11 Conference <http://www.slideshare.net/kleinerperkins/kpcb-internet-trends-2013>

F chapter/theory/clustering.tex

6.14 Technology Training - kNN & Clustering

This section is meant to provide a discussion on the k th Nearest Neighbor (kNN) algorithm and clustering using K-means. Python version for kNN is discussed in the video and instructions for both Java and Python are mentioned in the slides. Plotviz is used for generating 3D visualizations.

6.14.1 Recommender Systems - K-Nearest Neighbors

We discuss simple Python k Nearest Neighbor code and its application to an artificial data set in 3 dimensions. Results are visualized in Matplotlib in 2D and with Plotviz in 3D. The concept of training and testing sets are introduced with training set pre-labelled.

Files:

- kNN.py </files/python/knn/kNN.py>
- kNN_Driver.py </files/python/knn/kNN_Driver.py>
- DatingTesting2.txt </files/python/knn/dating_test_set2.txt>
- clusterFinal-M3-C3Dating-ReClustered.pviz </files/python/knn/clusterFinal-M3-C3Dating-ReClustered.pviz>
- DatingRating-OriginalLabels.pviz </files/python/knn/dating_rating_original_labels.pviz>
- clusterFinal-M30-C28.pviz </files/python/knn/clusterFinal-M30-C28.pviz>

Python k 'th Nearest Neighbor Algorithms

This lesson considers the Python k Nearest Neighbor code found on the web associated with a book by Harrington on Machine Learning. There are two data sets. First we consider a set of 4 2D vectors divided into two categories (clusters) and use $k=3$ Nearest Neighbor algorithm to classify 3 test points. Second we consider a 3D dataset that has already been classified and show how to normalize. In this lesson we just use Matplotlib to give 2D plots.

3D Visualization

The lesson modifies the online code to allow it to produce files readable by PlotViz. We visualize already classified 3D set and rotate in 3D.

Testing k 'th Nearest Neighbor Algorithms

The lesson goes through an example of using k NN classification algorithm by dividing dataset into 2 subsets. One is training set with initial classification; the other is test point to be classified by $k=3$ NN using training set. The code records fraction of points with a different classification from that input. One can experiment with different sizes of the two subsets. The Python implementation of algorithm is analyzed in detail.

6.14.2 Clustering and heuristic methods

We use example of recommender system to discuss clustering. The details of methods are not discussed but k -means based clustering methods are used and their results examined in Plotviz. The original labelling is compared to clustering results and extension to 28 clusters given. General

issues in clustering are discussed including local optima, the use of annealing to avoid this and value of heuristic algorithms.

Files:

- Fungi_LSU_3_15_to_3_26_zeroidx.pviz </files/python/plotviz/fungi_lsu_3_15_to_3_26_zeroidx.pviz>
- DatingRating-OriginalLabels.pviz </files/python/plotviz/datingrating_originallabels.pviz>
- clusterFinal-M30-C28.pviz </files/python/plotviz/clusterFinal-M30-C28.pviz>
- clusterFinal-M3-C3Dating-ReClustered.pviz </files/python/plotviz/clusterfinal_m3_c3dating_reclustered.pviz>

Kmeans Clustering

We introduce the k means algorithm in a gentle fashion and describes its key features including dangers of local minima. A simple example from Wikipedia is examined.

Clustering of Recommender System Example

Plotviz is used to examine and compare the original classification with an “optimal” clustering into 3 clusters using a fancy deterministic annealing method that is similar to k means. The new clustering has centers marked.

Clustering of Recommender Example into more than 3 Clusters

The previous division into 3 clusters is compared into a clustering into 28 separate clusters that are naturally smaller in size and divide 3D space covered by 1000 points into compact geometrically local regions.

Local Optima in Clustering

This lesson introduces some general principles. First many important processes are “just” optimization problems. Most such problems are rife with local optima. The key idea behind annealing to avoid local optima is described. The pervasive greedy optimization method is described.

Clustering in General

The two different applications of clustering are described. First find geometrically distinct regions and secondly divide spaces into geometrically compact regions that may have no “thin air” between them. Generalizations such as mixture models and latent factor methods are just mentioned. The important distinction between applications in vector spaces and those where only inter-point distances are defined is described. Examples are then given using PlotViz from 2D clustering of a mass spectrometry example and the results of clustering genomic data mapped into 3D with Multi Dimensional Scaling MDS.

Heuristics

Some remarks are given on heuristics; why are they so important why getting exact answers is often not so important?

Resources

- <https://en.wikipedia.org/wiki/Kmeans>
- http://grids.ucs.indiana.edu/ptliupages/publications/DACIDR_camera_ready_v0.3.pdf
- <http://salsahpc.indiana.edu/millionseq/>

- <http://salsafungiphy.blogspot.com/>
- <https://en.wikipedia.org/wiki/Heuristic>

Development

IV

6.15 Cloud Computing Technology for Big Data Applications & Analytics (will be updated)

We describe the central role of Parallel computing in Clouds and Big Data which is decomposed into lots of “Little data” running in individual cores. Many examples are given and it is stressed that issues in parallel computing are seen in day to day life for communication, synchronization, load balancing and decomposition. Cyberinfrastructure for e-moreorlessanything or moreorlessanything-Informatics and the basics of cloud computing are introduced. This includes virtualization and the important ‘as a Service’ components and we go through several different definitions of cloud computing.

Gartner’s Technology Landscape includes hype cycle and priority matrix and covers clouds and Big Data. Two simple examples of the value of clouds for enterprise applications are given with a review of different views as to nature of Cloud Computing. This IaaS (Infrastructure as a Service) discussion is followed by PaaS and SaaS (Platform and Software as a Service). Features in Grid and cloud computing and data are treated. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

Cloud (Data Center) Architectures with physical setup, Green Computing issues and software models are discussed followed by the Cloud Industry stakeholders with a 2014 Gartner analysis of Cloud computing providers. This is followed by applications on the cloud including data intensive problems, comparison with high performance computing, science clouds and the Internet of Things. Remarks on Security, Fault Tolerance and Synchronicity issues in cloud follow. We describe the way users and data interact with a cloud system. The Big Data Processing from an application perspective with commercial examples including eBay concludes section after a discussion of data system architectures.

6.15.1 Parallel Computing: Overview of Basic Principles with familiar Examples

We describe the central role of Parallel computing in Clouds and Big Data which is decomposed into lots of “Little data” running in individual cores. Many examples are given and it is stressed that issues in parallel computing are seen in day to day life for communication, synchronization, load balancing and decomposition.

- Slides: (33 Pages) https://drive.google.com/file/d/0B8936_ytjfjmZDRqMldzSVhVem8/view?usp=sharing

Decomposition

We describe why parallel computing is essential with Big Data and distinguishes parallelism over users to that over the data in problem. The general ideas behind data decomposition are given followed by a few often whimsical examples dreamed up 30 years ago in the early heady days of parallel computing. These include scientific simulations, defense outside missile attack and computer chess. The basic problem of parallel computing – efficient coordination of separate tasks processing different data parts – is described with MPI and MapReduce as two approaches. The challenges of data decomposition in irregular problems is noted.

- Video 1: 8:51: Decomposition: <https://drive.google.com/file/d/0B10f61fJF7WsWXgtNndQNOJydkk/view?usp=sharing>

- Video 2: 13:22: Examples of Application: <https://drive.google.com/file/d/0B10f61fJF7WsQ1pMLWhXGview?usp=sharing>
- Video 3: 9:22: Decomposition Strategies: <https://drive.google.com/file/d/0B10f61fJF7WsaVZLOEUzcview?usp=sharing>

Parallel Computing in Society

This lesson from the past notes that one can view society as an approach to parallel linkage of people. The largest example given is that of the construction of a long wall such as that (Hadrian's wall) between England and Scotland. Different approaches to parallelism are given with formulae for the speed up and efficiency. The concepts of grain size (size of problem tackled by an individual processor) and coordination overhead are exemplified. This example also illustrates Amdahl's law and the relation between data and processor topology. The lesson concludes with other examples from nature including collections of neurons (the brain) and ants.

- Video 1: 8:24: Parallel Computing in Society I: <https://drive.google.com/file/d/0B10f61fJF7WsY3hEeTJvTFYtN2s/view?usp=sharing>
- Video 2: 8:01: Parallel Computing in Society II: <https://drive.google.com/file/d/0B10f61fJF7WsU1ROMmpNNt1UTUU/view?usp=sharing>

Parallel Processing for Hadrian's Wall

This lesson returns to Hadrian's wall and uses it to illustrate advanced issues in parallel computing. First We describe the basic SPMD – Single Program Multiple Data – model. Then irregular but homogeneous and heterogeneous problems are discussed. Static and dynamic load balancing is needed. Inner parallelism (as in vector instruction or the multiple fingers of masons) and outer parallelism (typical data parallelism) are demonstrated. Parallel I/O for Hadrian's wall is followed by a slide summarizing this quaint comparison between Big data parallelism and the construction of a large wall.

- Video: 9:24: Processing for Hadrian's Wall: <https://drive.google.com/file/d/0B10f61fJF7WsNETLOTNview?usp=sharing>

Resources

- Solving Problems in Concurrent Processors-Volume 1, with M. Johnson, G. Lyzenga, S. Otto, J. Salmon, D. Walker, Prentice Hall, March 1988.
- Parallel Computing Works!, with P. Messina, R. Williams, Morgan Kaufman (1994). <http://www.netlib.org/utk/lsi/pcwLSI/text/>
- The Sourcebook of Parallel Computing book edited by Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, Morgan Kaufmann, November 2002.
- Geoffrey Fox Computational Sciences and Parallelism to appear in Encyclopedia on Parallel Computing edited by David Padua and published by Springer. http://grids.ucs.indiana.edu/ptliupages/publications/SpringerEncyclopedia_Fox.pdf

6.15.2 Cloud Computing Technology Part I: Introduction

We discuss Cyberinfrastructure for e-moreorlessanything or moreorlessanything-Informatics and the basics of cloud computing. This includes virtualization and the important ‘as a Service’ components and we go through several different definitions of cloud computing. Gartner’s Technology Landscape includes hype cycle and priority matrix and covers clouds and Big Data. The unit concludes with

two simple examples of the value of clouds for enterprise applications. Gartner also has specific predictions for cloud computing growth areas.

- Slides: (45 pages) https://drive.google.com/file/d/0B8936_ytjfjmdmF2Uy1vWS0xTFU/view?usp=sharing

Cyberinfrastructure for E-MoreOrLessAnything

This introduction describes Cyberinfrastructure or e-infrastructure and its role in solving the electronic implementation of any problem where e-moreorlessanything is another term for moreorlessanything-Informatics and generalizes early discussion of e-Science and e-Business.

- Video: 13:34: Cloud Computing Introduction Part1: <https://drive.google.com/file/d/0B10f61fJF7WsbXpEdF8zWFh4aXc/view?usp=sharing>

What is Cloud Computing: Introduction

Cloud Computing is introduced with an operational definition involving virtualization and efficient large data centers that can rent computers in an elastic fashion. The role of services is essential – it underlies capabilities being offered in the cloud. The four basicaaS's – Software (SaaS), Platform (PaaS), Infrastructure (IaaS) and Network (NaaS) – are introduced with ResearchaaS and other capabilities (for example SensorsaaS are discussed later) being built on top of these.

- Video: 12:01: What is Cloud Computing Intro: <https://drive.google.com/file/d/0B10f61fJF7WsdDdsYkw0dXdHS1U/view?usp=sharing>

What and Why is Cloud Computing: Several Other Views I

This lesson contains 5 slides with diverse comments on “what is cloud computing” from the web.

- Video 1: 5:25: Other Views I: <https://drive.google.com/file/d/0B10f61fJF7WsNm1jVVJMUVpCU1U/view?usp=sharing>
- Video 2: 6:41: Other Views II: <https://drive.google.com/file/d/0B10f61fJF7WsV1RJcldzRlctRlk/view?usp=sharing>
- Video 3: 7:27: Other Views III: <https://drive.google.com/file/d/0B10f61fJF7WsOUlxVHZ4M1NORXc/view?usp=sharing>

Gartner's Emerging Technology Landscape for Clouds and Big Data

This lesson gives Gartner's projections around futures of cloud and Big data. We start with a review of hype charts and then go into detailed Gartner analyses of the Cloud and Big data areas. Big data itself is at the top of the hype and by definition predictions of doom are emerging. Before too much excitement sets in, note that spinach is above clouds and Big data in Google trends.

- Video: 11:26: Gartners Emerging Technology Landscape: <https://drive.google.com/file/d/0B10f61fJF7WsaTg5aEZ0cHJuM0k/view?usp=sharing>

Simple Examples of use of Cloud Computing

This short lesson gives two examples of rather straightforward commercial applications of cloud computing. One is server consolidation for multiple Microsoft database applications and the second is the benefits of scale comparing gmail to multiple smaller installations. It ends with some fiscal comments.

- Video: 3:26: Sample Examples: <https://drive.google.com/file/d/0B10f61fJF7WsLTBoMONpYzVx0Hc/view?usp=sharing>

[view?usp=sharing](#)

Value of Cloud Computing

Some comments on fiscal value of cloud computing.

- Video: 4:20: Value of Cloud Computing: <https://drive.google.com/file/d/0B10f61fJF7WsSFdfZ0hodl>
[view?usp=sharing](#)

Resources

- <http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing>
- <http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis>
- <https://setandbma.wordpress.com/2012/08/10/hype-cycle-2012-emerging-technologies/>
- <http://insights.dice.com/2013/01/23/big-data-hype-is-imploding-gartner-analyst-2/>
- http://research.microsoft.com/pubs/78813/AJ18_EN.pdf
- <http://static.googleusercontent.com/media/www.google.com/en//green/pdfs/google-green-computing.pdf>

6.15.3 Cloud Computing Technology Part II: Software and Systems

We cover different views as to nature of architecture and application for Cloud Computing. Then we discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

- Slides (32 pages): https://drive.google.com/file/d/0B8936_ytjfjmUH1EVG1wSUhDNnM/
[view?usp=sharing](#)

What is Cloud Computing

This lesson gives some general remark of cloud systems from an architecture and application perspective.

- Video: 6:20: What is Cloud Computing: <https://drive.google.com/file/d/0B10f61fJF7WsY1RhOHU5ci>
[view?usp=sharing](#)

Introduction to Cloud Software Architecture: IaaS and PaaS I

We discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We cover different views as to nature of architecture and application for Cloud Computing. Then we discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

- Video: 7:42: Intro to IaaS and PaaS I: <https://drive.google.com/file/d/0B10f61fJF7WsUm1XanBaaWtp>
[view?usp=sharing](#)
- Video: 6:42: Intro to IaaS and PaaS II: <https://drive.google.com/file/d/0B10f61fJF7WsMXpfTTlvNDBk>
[view?usp=sharing](#)

We discuss cloud software for the cloud starting at virtual machine management (IaaS) and the

broad Platform (middleware) capabilities with examples from Amazon and academic studies. We cover different views as to nature of architecture and application for Cloud Computing. Then we discuss cloud software for the cloud starting at virtual machine management (IaaS) and the broad Platform (middleware) capabilities with examples from Amazon and academic studies. We summarize the 21 layers and almost 300 software packages in the HPC-ABDS Software Stack explaining how they are used.

- Video: 7:42: Software Architecture: <https://youtu.be/1AnyJYyh490>
- Video: 6:43: IaaS and PaaS II: <https://youtu.be/hVpFAUHcAd4>

Using the HPC-ABDS Software Stack

Using the HPC-ABDS Software Stack.

- Video: 27:50: ABDS: <https://drive.google.com/file/d/0B10f61fJF7WsUTd1Nm1YWDUyT1E/view?usp=sharing>

Resources

- <http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis>
- http://research.microsoft.com/en-us/people/barga/sc09_cloudcomp_tutorial.pdf
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Yousef_Khalidi.pdf
- <http://cloudonomic.blogspot.com/2009/02/cloud-taxonomy-and-ontology.html>

6.15.4 Cloud Computing Technology Part III: Architectures, Applications and Systems

We start with a discussion of Cloud (Data Center) Architectures with physical setup, Green Computing issues and software models. We summarize a 2014 Gartner analysis of Cloud computing providers. This is followed by applications on the cloud including data intensive problems, comparison with high performance computing, science clouds and the Internet of Things. Remarks on Security, Fault Tolerance and Synchronicity issues in cloud follow.

- Slides (64 pages): https://drive.google.com/file/d/0B8936_ytjfjmTHlzcGN3SzFNTTA/view?usp=sharing

Cloud (Data Center) Architectures

Some remarks on what it takes to build (in software) a cloud ecosystem, and why clouds are the data center of the future are followed by pictures and discussions of several data centers from Microsoft (mainly) and Google. The role of containers is stressed as part of modular data centers that trade scalability for fault tolerance. Sizes of cloud centers and supercomputers are discussed as is “green” computing.

- Video 1: 8:38: Coud Architecture: <https://drive.google.com/file/d/0B10f61fJF7WsYkxKe1V2bT1MZ1I/view?usp=sharing>
- Video 2: 9:59: Cloud Data Center Architecture: <https://drive.google.com/file/d/0B10f61fJF7WsRHJhN3VMaDJLTG8/view?usp=sharing>

Analysis of Major Cloud Providers

Gartner 2014 Analysis of leading cloud providers.

- video: 21:40: Analysis of Major Cloud Providers: <https://drive.google.com/file/d/0B10f61fJF7WsUXBjRUJpX1BaSjA/view?usp=sharing>

Commercial Cloud Storage Trends

Use of Dropbox, iCloud, Box etc.

- video: 3:07: Commercial Storage Trends: <https://drive.google.com/file/d/0B10f61fJF7WsZjR5VHQ2M/view?usp=sharing>

Cloud Applications I

This short lesson discusses the need for security and issues in its implementation. Clouds trade scalability for greater possibility of faults but here clouds offer good support for recovery from faults. We discuss both storage and program fault tolerance noting that parallel computing is especially sensitive to faults as a fault in one task will impact all other tasks in the parallel job.

- Video 1: 7:57: Cloud Applications I: <https://drive.google.com/file/d/0B10f61fJF7WsYX1KVXk0aG8tZ/view?usp=sharing>
- Video 2: 7:44: Cloud Applications II: <https://drive.google.com/file/d/0B10f61fJF7WseGVUNHhGTHpZ/view?usp=sharing>

Science Clouds

Science Applications and Internet of Things.

- video: 19:26: Science Clouds: <https://drive.google.com/file/d/0B10f61fJF7Wsd0lZejhPTkItZEE/view?usp=sharing>

Security

This short lesson discusses the need for security and issues in its implementation.

- video: 2:34: Security: <https://drive.google.com/file/d/0B10f61fJF7WsajE4QkljRUExLWM/view?usp=sharing>

Comments on Fault Tolerance and Synchronicity Constraints

Clouds trade scalability for greater possibility of faults but here clouds offer good support for recovery from faults. We discuss both storage and program fault tolerance noting that parallel computing is especially sensitive to faults as a fault in one task will impact all other tasks in the parallel job.

- video: 8:55: Comments on Fault Tolerance and Synchronicity Constraints: <https://drive.google.com/file/d/0B10f61fJF7WsdHRZV1VrTk1WYVE/view?usp=sharing>

Resources

- <http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing>
- <http://www.eweek.com/c/a/Cloud-Computing/AWS-Innovation-Means-Cloud-Domination-307831>
- CSTI General Assembly 2012, Washington, D.C., USA Technical Activities Coordinating Committee (TACC) Meeting, Data Management, Cloud Computing and the Long Tail of Science October 2012 Dennis Gannon.
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Yousef_Khalidi.pdf

- <http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pu>
- <https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data->
- <http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterlectuse2011finalversion.pdf>
- <http://www.slideshare.net/JensNimis/cloud-computing-tutorial-jens-nimis>
- <http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus>
- <http://www.venus-c.eu/Pages/Home.aspx>
- Geoffrey Fox and Dennis Gannon Using Clouds for Technical Computing To be published in Proceedings of HPC 2012 Conference at Cetraro, Italy June 28 2012 http://grids.ucs.indiana.edu/ptliupages/publications/Clouds_Technical_Computing_FoxGannonv2.pdf
- <https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf>
- Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks Wiley ISBN: 978-1-118-20878-6
- Anjul Bhambhri, VP of Big Data, IBM http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Conquering Big Data with the Oracle Information Model, Helen Sun, Oracle
- Hugh Williams VP Experience, Search & Platforms, eBay <http://businessinnovation.berkeley.edu/fisher-cio-leadership-program/>
- Dennis Gannon, Scientific Computing Environments, http://www.nitrd.gov/nitrdgroups/images/7/73/D_Gannon_2025_scientific_computing_environments.pdf
- http://research.microsoft.com/en-us/um/redmond/events/cloudfutures2012/tuesday/Keynote_OpportunitiesAndChallenges_Yousef_Khalidi.pdf
- <http://www.datacenterknowledge.com/archives/2011/05/10/uptime-institute-the-average-pu>
- <https://loosebolts.wordpress.com/2008/12/02/our-vision-for-generation-4-modular-data->
- <http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterlectuse2011finalversion.pdf>
- <http://searchcloudcomputing.techtarget.com/feature/Cloud-computing-experts-forecast-th>
- <http://www.slideshare.net/botchagalupe/introduction-to-clouds-cloud-camp-columbus>
- <http://www.slideshare.net/woorung/trend-and-future-of-cloud-computing>
- <http://www.venus-c.eu/Pages/Home.aspx>
- <http://www.kpcb.com/internet-trends>

6.15.5 Cloud Computing Technology Part IV: Data Systems

We describe the way users and data interact with a cloud system. The unit concludes with the treatment of data in the cloud from an architecture perspective and Big Data Processing from an application perspective with commercial examples including eBay.

- Slides: (49 pages): <https://drive.google.com/file/d/0B10f61fJF7WsN1RPVFLUGJLZGs/view?usp=sharing>

The 10 Interaction scenarios (access patterns) I

The next 3 lessons describe the way users and data interact with the system.

- Video: 10:26: The 10 Interaction scenarios I: <https://drive.google.com/file/d/0B10f61fJF7WsWldDNm1oNXdPQmc/view?usp=sharing>

The 10 Interaction scenarios. Science Examples

This lesson describes the way users and data interact with the system for some science examples.

- Video: 16:34: The 10 Interaction scenarios. Science Examples: <https://drive.google.com/file/d/0B10f61fJF7WsQT1vLWs4cm5NRE0/view?usp=sharing>

Remaining general access patterns

This lesson describe the way users and data interact with the system for the final set of examples.

- Video: 11:36: Access Patterns: <https://drive.google.com/file/d/0B10f61fJF7WsYVVRWmdpanV4Vlk/view?usp=sharing>

Data in the Cloud

Databases, File systems, Object Stores and NOSQL are discussed and compared. The way to build a modern data repository in the cloud is introduced.

- Video: 10:24: Data in the Cloud: <https://drive.google.com/file/d/0B10f61fJF7WsRzR6eHZwelVu0G8/view?usp=sharing>

Applications Processing Big Data

This lesson collects remarks on Big data processing from several sources: Berkeley, Teradata, IBM, Oracle and eBay with architectures and application opportunities.

- Video: 8:45: Processing Big Data: <https://drive.google.com/file/d/0B10f61fJF7WsUG9UVGF0QXNXbn/view?usp=sharing>

Resources

- http://bigdatawg.nist.gov/_uploadfiles/M0311_v2_2965963213.pdf
- <https://dzone.com/articles/hadoop-t-etl>
- <http://venublog.com/2013/07/16/hadoop-summit-2013-hive-authorization/>
- <https://indico.cern.ch/event/214784/session/5/contribution/410>
- http://asd.gsfc.nasa.gov/archive/hubble/a_pdf/news/facts/FS14.pdf
- <http://blogs.teradata.com/data-points/announcing-teradata-aster-big-analytics-appliance/>
- <http://wikibon.org/w/images/2/20/Cloud-BigData.png>
- <http://hortonworks.com/hadoop/yarn/>
- <https://berkeleydatascience.files.wordpress.com/2012/01/20120119berkeley.pdf>
- http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html



Python

7	Introduction	151
7.1	Introduction to Python	
8	Install	155
8.1	Python Installation	
8.2	Interactive Python	
8.3	REPL (Read Eval Print Loop)	
8.4	Python 3 Features in Python 2	
9	Language	165
9.1	Statements and Strings	
9.2	Variables	
9.3	Data Types	
9.4	Module Management	
9.5	Date Time in Python	
9.6	Control Statements	
9.7	Datatypes	
9.8	Functions	
9.9	Classes	
9.10	Database Access	
9.11	Modules	
9.12	Installing Libraries	
9.13	Using pip to Install Packages	
9.14	GUI	
9.15	Formatting and Checking Python Code	
9.16	Using autopep8	
9.17	Further Learning	
9.18	Writing Python 3 Compatible Code	
9.19	Using Python on FutureSystems	
9.20	Ecosystem	
9.21	Resources	
9.22	Jupyter Notebook Tutorials	
9.23	Exercises	
10	Cloudmesh Command Shell	185
10.1	CMD5	
10.2	Code Management	



7. Introduction

 chapter/prg/python-intro.tex

7.1 Introduction to Python

Portions of this lesson have been adapted from the official Python Tutorial copyright Python Software Foundation.

Python is an easy to learn programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's simple syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter can be extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

Python is an interpreted, dynamic, high-level programming language suitable for a wide range of applications.

The philosophy of python is summarized in The Zen of Python as follows:

- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

The main features of Python are:

- Use of indentation whitespace to indicate blocks
- Object orient paradigm
- Dynamic typing
- Interpreted runtime
- Garbage collected memory management
- a large standard library
- a large repository of third-party libraries

Python is used by many companies (such as Google, Yahoo!, CERN, NASA) and is applied for web development, scientific computing, embedded applications, artificial intelligence, software development, and information security, to name a few.

7.1.1 About the Tutorial

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well. At the end of this lesson you will be able to:

- use Python
- use the interactive Python interface
- understand the basic syntax of Python
- write and run Python programs stored in a file

- have an overview of the standard library
- install Python libraries using pyenv or if it is not available virtualenv

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules.

In order to conduct this lesson you need

- A computer with Python 2.7.13 or 3.6.2
- Familiarity with command line usage
- A text editor such as PyCharm, emacs, vi or others. You should identify which works best for you and set it up.

7.1.2 Links

- Python
- Pip
- Virtualenv
- NumPy
- SciPy
- Matplotlib
- Pandas
- pyenv
- PyCharm

Python module of the week is a Web site that provides a number of short examples on how to use some elementary python modules. Not all modules are equally useful and you should decide if there are better alternatives. However for beginners this site provides a number of good examples

- Python 2: <https://pymotw.com/2/>
- Python 3: <https://pymotw.com/3/>



8. Install

 chapter/prg/python-install.tex

8.1 Python Installation

Python is easy to install and very good instructions for most platforms can be found on the python.org Web page. We will be using Python 2.7.13 and/or Python 3 in our activities.

To manage python modules, it is useful to have pip package installation tool on your system.

In the tutorial, we assume that you have a computer with python installed. However, we also recommend that for the class you use Python's virtualenv (see below) to isolate your development Python from the system installed Python.

8.1.1 Managing custom Python installs

Often you have your own computer and you do not like to change its environment to keep it in pristine condition. Python comes with many libraries that could for example conflict with libraries that you have installed. To avoid this it is best to work in an isolated python we can use tools such as virtualenv, pyenv or pyvenv for 3.6.2. Which you use depends on you, but we highly recommend pyenv if you can.

Managing Multiple Python Versions with Pyenv

Python has several versions that are used by the community. This includes Python 2 and Python 3, but alls different management of the python libraries. As each OS may have their own version of python installed. It is not recommended that you modify that version. Instead you may want to create a localized python installation that you as a user can modify. To do that we recommend *pyenv*. Pyenv allows users to switch between multiple versions of Python (<https://github.com/yyuu/pyenv>). To summarize:

- users to change the global Python version on a per-user basis;
- users to enable support for per-project Python versions;
- easy version changes without complex environment variable management;
- to search installed commands across different python versions;
- integrate with tox (<https://tox.readthedocs.io/>).

Instalation without pyenv

If you need to have more than one python version installed and do not want or can use pyenv, we recommend you download and install python 2.7.13 and 3.6.2 from [python.org](http://www.python.org/downloads/) (<https://www.python.org/downloads/>)

Disabeling wrong python installs on OSX

While working with students we have seen at times that they take other classes either at universities or online that teach them how to program in python. Unfortuanatley, although they seem to do that they often ignore to teach you how to properly install python. I just reacheits had a students that had installed python 7 times on his OSX machine, while another student had 3 different instalations, all of which confliced with each other as they were not set up properly.

We recommend that you inspect if you have a files such as `~/.bashrc` or `~/.bashrc_profile` in your ehome directory and identify if it activates various versions of python on your computer. If so you

could try to deactivate them while outcommenting the various versions with the # character at the beginning of the line, start a new terminal and see if the terminal shell still works. Than you can follow our instructions here while using an install on pyenv.

Install pyenv on OSX from git

This is our recommended way to install pyenv on OSX:

```
$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv  
$ git clone https://github.com/pyenv/pyenv-virtualenv.git ~/.pyenv/plugins/pyenv-virtualenv  
$ git clone https://github.com/yyuu/pyenv-virtualenvwrapper.git ~/.pyenv/plugins/pyenv-virt  
$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile  
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile
```

Instalation of Homebrew

Before installing anything on your computer make sure you have enough space. Use in the terminal the command:

```
$ df -h
```

which gives your an overview of your file system. If you do not have enough space, please make sure you free up unused files from your drive.

In many occasions it is beneficial to use readline as it provides nice editing features for the terminal and xz for completion. First, make sure you have xcode installed:

```
$ xcode-select --install
```

Next install homebrew, pyenv, pyenv-virtualenv and pyenv-virtualwrapper. Additionally install readline and some compression tools:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/in  
brew update  
brew install readline xz
```

Install pyenv on OSX with Homebrew

We describe here a mechanism of installing pyenv with homebrew. Other mechanisms can be found on the pyenv documentation page (<https://github.com/yyuu/pyenv-installer>). You must have homebrew installed as discussed in the previous section.

To install pyenv with homebrew execute in the terminal:

```
brew install pyenv pyenv-virtualenv pyenv-virtualenvwrapper
```

Install pyenv on Ubuntu

The following steps will install pyenv in a new ubuntu 16.04 distribution.

Start up a terminal and execute in the terminal the following commands. We recommend that you do it one command at a time so you can observe if the command succeeds:

```
$ sudo apt-get update  
$ sudo apt-get install git python-pip make build-essential libssl-dev  
$ sudo apt-get install zlib1g-dev libbz2-dev libreadline-dev libsdlite3-dev  
$ sudo pip install virtualenvwrapper
```

```
$ git clone https://github.com/yyuu/pyenv.git ~/.pyenv
$ git clone https://github.com/pyenv/pyenv-virtualenv.git ~/.pyenv/plugins/pyenv-virtualenv
$ git clone https://github.com/yyuu/pyenv-virtualenvwrapper.git ~/.pyenv/plugins/pyenv-virtualenvwrapper

$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
$ echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
```

Now that you have installed pyenv it is not yet activated in your current terminal. The easiest thing to do is to start a new terminal and type in:

```
which pyenv
```

If you see a response pyenv is installed and you can proceed with the next steps.

Please remember whenever you modify .bashrc or .bash_profile you need to start a new terminal.

Install Different Python Versions

Pyenv provides a large list of different python versions. To see the entire list please use the command:

```
$ pyenv install -l
```

However, for us we only need to worry about python 2.7.13 and python 3.6.2 (once 3.6.2 becomes available we will use that). You can now install different versions of python into your local environment with the following commands:

```
$ pyenv install 2.7.13
$ pyenv install 3.6.2
```

You can set the global python default version with:

```
$ pyenv global 2.7.13
```

Type the following to determine which version you activated:

```
$ pyenv version
```

Type the following to determine which versions you have available:

```
$ pyenv versions
```

Associate a specific environment name with a certain python version, use the following commands:

```
$ pyenv virtualenv 2.7.13 ENV2
$ pyenv virtualenv 3.6.2 ENV3
```

In the example above, ENV2 would represent python 2.7.13 while ENV3 would represent python 3.6.2. Often it is easier to type the alias rather than the explicit version.

Set up the Shell

To make all work smoothly from your terminal, you can include the following in your .bashrc files:

```
export PYENV_VIRTUALENV_DISABLE_PROMPT=1
eval "$(pyenv init -)"
eval "$(pyenv virtualenv-init -)"
```

```
__pyenv_version_ps1() {  
    local ret=$?;  
    output=$(pyenv version-name)  
    if [[ ! -z $output ]]; then  
        echo -n "($output)"  
    fi  
    return $ret;  
}  
  
PS1="\$(__pyenv_version_ps1) ${PS1}"
```

We recommend that you do this towards the end of your file.

Switching Environments

After setting up the different environments, switching between them is now easy. Simply use the following commands:

```
(2.7.13) $ pyenv activate ENV2  
(ENV2) $ pyenv activate ENV3  
(ENV3) $ pyenv activate ENV2  
(ENV2) $ pyenv deactivate ENV2  
(2.7.13) $
```

To make it even easier, you can add the following lines to your .bash_profile file:

```
alias ENV2="pyenv activate ENV2"  
alias ENV3="pyenv activate ENV3"
```

If you start a new terminal, you can switch between the different versions of python simply by typing:

```
$ ENV2  
$ ENV3
```

8.1.2 Instalation without pyenv

If you need to have more than one python version installed and do not want or can use pyenv, we recommend you download and install python 2.7.13 and 3.6.2 from python.org (<https://www.python.org/downloads/>)

Make sure pip is up to date

As you will want to install other packages, make sure pip is up to date:

```
pip install pip -U  
pyenv virtualenv anaconda3-4.3.1 ANA3  
pyenv activate ANA3
```

8.1.3 Anaconda and Miniconda

We do not recommend that you use anaconda or miniconda as it may interfere with your default python interpreters and setup.

Please note that beginners to python should always use anaconda or miniconda only after they have

installed pyenv and use it. For this class neither anaconda nor miniconda is required. In fact we do not recommend it. We keep this section as we know that other classes at IU may use anaconda. We are not aware if these classes teach you the right way to install it, with *pyenv*.

Miniconda

This section about miniconda is experimental and has not been tested. We are looking for contributors that help completing it. If you use anaconda or miniconda we recommend to manage it via pyenv.

To install mini conda you can use the following commands:

```
$ mkdir ana
$ cd ana
$ pyenv install miniconda3-latest
$ pyenv local miniconda3-latest
$ pyenv activate miniconda3-latest
$ conda create -n ana anaconda
```

To activate use:

```
$ source activate ana
```

To deactivate use:

```
$ source deactivate
```

To install cloudmesh cmd5 please use:

```
$ pip install cloudmesh.cmd5
$ pip install cloudmesh.sys
```

Anaconda

This section about anaconda is experimental and has not been tested. We are looking for contributors that help completing it.

You can add anaconda to your pyenv with the following commands:

```
pyenv install anaconda3-4.3.1
```

To switch more easily we recommend that you use the following in your .bash_profile file:

```
alias ANA="pyenv activate anaconda3-4.3.1"
```

Once you have done this you can easily switch to anaconda with the command:

```
$ ANA
```

Terminology in anaconda could lead to confusion. Thus we like to point out that the version number of anaconda is unrelated to the python version. Furthermore, anaconda uses the term root not for the root user, but for the originating directory in which the anaconda program is installed.

In case you like to build your own conda packages at a later time we recommend that you install the conda-build package:

```
$ conda install conda-build
```

When executing:

```
pyenv versions
```

you will see after the install completed the anaconda versions installed:

```
pyenv versions
system
2.7.13
2.7.13/envs/ENV2
3.6.2
3.6.2/envs/ENV3
ENV2
ENV3
* anaconda3-4.3.1 (set by PYENV_VERSION environment variable)
```

Let us now create virtualenv for anaconda:

```
$ pyenv virtualenv anaconda3-4.3.1 ANA
```

To activate it you can now use:

```
$ pyenv ANA
```

However, anaconda may modify your .bashrc or .bash_profile files and , may result in incompatibilities with other python versions. For this reason we recommend not to use it. If you find ways to get it to work reliably with other versions, please let us know and we update this tutorial.

To install cloudmesh cmd5 please use:

```
$ pip install cloudmesh.cmd5
$ pip install cloudmesh.sys
```

Exercise

Epyenv.1: Write installation instructions for an operating system of your choice and add to this documentation.

Epyenv.2: Replicate the steps above, so you can type in ENV2 and ENV3 in your terminals to switch between python 2 and 3.

virtualenv

environment while using virtualenv,. Documentation about it can be found at:

```
* https://virtualenv.pypa.io
```

The installation is simple once you have pip installed. If it is not installed you can say:

```
$ easy_install pip
```

After that you can install the virtual env with:

```
$ pip install virtualenv
```

To setup an isolated environment for example in the directory ~/ENV please use:

```
$ virtualenv ~/ENV
```

To activate it you can use the command:

```
$ source ~/ENV/bin/activate
```

you can put this command in your .bashrc or .bash_profile files so you do not forget to activate it. Instructions for this can be found in our lesson on Linux <bashrc>.

 chapter/prg/python-interactive.tex

8.2 Interactive Python

Python can be used interactively. Start by entering the interactive loop by executing the command:

```
$ python
```

You should see something like the following:

```
Python 2.7.13 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The `>>>` is the prompt for the interpreter. This is similar to the shell interpreter you have been using.

Often we show the prompt when illustrating an example. This is to provide some context for what we are doing. If you are following along you will not need to type in the prompt.

This interactive prompt does the following:

- *read* your input commands
- *evaluate* your command
- *print* the result of evaluation
- *loop* back to the beginning.

This is why you may see the interactive loop referred to as a **REPL**: Read-Evaluate-Print-Loop.

8.3 REPL (Read Eval Print Loop)

We have so far seen a few examples of types: **strings**, **bools**, **ints**, and **floats**. A **type** indicates that values of that type support a certain set of operations. For instance, how would you exponentiate a string? If you ask the interpreter, this results in an error:

```
>>> "hello"**3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
```

There are many different types beyond what we have seen so far, such as **dictionaries**, **lists**, **sets**. One handy way of using the interactive python is to get the type of a value using `'type()'`:

```
::
>>> type(42) <type 'int'> >>> type(hello) <type 'str'> >>> type(3.14) <type 'float'>
```

You can also ask for help about something using `help()`:

```
::
>>> help(int) >>> help(list) >>> help(str)
.. tip::
```

Using `help()` opens up a pager. To navigate you can use the spacebar to go down a page w to go up

a page, the arrow keys to go up/down line-by-line, or q to exit.

8.4 Python 3 Features in Python 2

As mentioned earlier, we assume you will use Python 2.7.X because there are still some libraries that haven't been ported to Python 3. However, there are some features of Python 3 we can and want to use in Python 2.7. Before we do anything else, we need to make these features available to any subsequent code we write:

```
>>> from __future__ import print_function, division
```

The first of these imports allows us to use the `print` function to output text to the screen, instead of the `print` statement, which Python 2 uses. This is simply a design decision that better reflects Python's underlying philosophy.

The second of these imports makes sure that the `division` operator behaves in a way a newcomer to the language might find more intuitive. In Python 2, `division /` is *floor division* when the arguments are integers, meaning that `5 / 2 == 2`, for example. In Python 3, `division /` is *true division*, thus `5 / 2 == 2.5`.



9. Language

 chapter/prg/python.tex

9.1 Statements and Strings

Let us explore the syntax of Python. Type into the interactive loop and press Enter:

```
>>> print("Hello world from Python!")
Hello world from Python!
```

What happened: the `print` function was given a **string** to process. A string is a sequence of characters. A **character** can be alphabetic (A through Z, lower and upper case), numeric (any of the digits), white space (spaces, tabs, newlines, etc), syntactic directives (comma, colon, quotation, exclamation, etc), and so forth. A string is just a sequence of the character and typically indicated by surrounding the characters in double quotes.

Standard output is discussed in the `../lesson/linux/shell` lesson.

So, what happened when you pressed Enter? The interactive Python program read the line `print "Hello world from Python!"`, split it into the `print` statement and the `"Hello world from Python!"` string, and then executed the line, showing you the output.

9.2 Variables

You can store data into a **variable** to access it later. For instance, instead of:

```
>>> print('Hello world from Python!')
```

which is a lot to type if you need to do it multiple times, you can store the string in a variable for convenient access:

```
>>> hello = 'Hello world from Python!'
>>> print(hello)
Hello world from Python!
```

9.3 Data Types

9.3.1 Booleans

A **boolean** is a value that indicates *truthiness* of something. You can think of it as a toggle: either “on” or “off”, “one” or “zero”, “true” or “false”. In fact, the only possible values of the **boolean** (or `bool`) type in Python are:

- True
- False

You can combine booleans with **boolean operators**:

- and
- or

```
>>> print(True and True)
True
>>> print(True and False)
```

```

False
>>> print(False and False)
False
>>> print(True or True)
True
>>> print(True or False)
True
>>> print(False or False)
False

```

9.3.2 Numbers

The interactive interpreter can also be used as a calculator. For instance, say we wanted to compute a multiple of 21:

```

>>> print(21 * 2)
42

```

We saw here the print statement again. We passed in the result of the operation $21 * 2$. An **integer** (or **int**) in Python is a numeric value without a fractional component (those are called **floating point** numbers, or **float** for short).

The mathematical operators compute the related mathematical operation to the provided numbers. Some operators are:

- * — multiplication
- / — division
- + — addition
- - — subtraction
- ** — exponent

Exponentiation is read as $x^{**}y$ is x to the y th power:

$$x^y$$

You can combine **floats** and **ints**:

```

>>> print(3.14 * 42 / 11 + 4 - 2)
13.9890909091
>>> print(2**3)
8

```

Note that **operator precedence** is important. Using parenthesis to indicate affect the order of operations gives a difference results, as expected:

```

>>> print(3.14 * (42 / 11) + 4 - 2)
11.42
>>> print(1 + 2 * 3 - 4 / 5.0)
6.2
>>> print( (1 + 2) * (3 - 4) / 5.0 )
-0.6

```

9.4 Module Management

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference. A module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.

9.4.1 Import Statement

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module. The from...import Statement Python's from statement lets you import specific attributes from a module into the current namespace. The from...import has the following syntax - from modname:

```
import name1[, name2[, ... nameN]]
```

When the interpreter encounters an import statement, it imports the module if the module is present in the search path. A search path is a list of directories that the interpreter searches before importing a module.

9.4.2 The from ... import Statement

Python's from statement lets you import specific attributes from a module into the current namespace. The from ... import has the following syntax:

```
::
```

```
from module1 import name1[, name2[, ... nameN]]
```

9.5 Date Time in Python

The datetime module supplies classes for manipulating dates and times in both simple and complex ways. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation. For related functionality, see also the time and calendar modules.

The import Statement You can use any Python source file as a module by executing an import statement in some other Python source file.

```
>>>from datetime import datetime
```

This module offers a generic date/time string parser which is able to parse most known formats to represent a date and/or time.

```
>>>from dateutil.parser import parse
```

pandas is an open source Python library for data analysis that needs to be imported.

```
>>>import pandas as pd
```

Create a string variable with the class start time

```
>>>fall_start = '08-21-2017'
```

Convert the string to datetime format

```
>>>datetime.strptime(fall_start, '%m-%d-%Y')
datetime.datetime(2017, 8, 21, 0, 0)
```

Creating a list of strings as dates

```
>>>class_dates = ['8/25/2017', '9/1/2017', '9/8/2017', '9/15/2017', '9/22/2017', '9/29/2017']
```

Convert Class_dates strings into datetime format and save the list into variable a

```
>>>a = [datetime.strptime(x, '%m/%d/%Y') for x in class_dates]
```

Use parse() to attempt to auto-convert common string formats. Parser must be a string or character stream, not list.

```
>>>parse(fall_start)
datetime.datetime(2017, 8, 21, 0, 0)
```

Use parse() on every element of the Class_dates string.

```
>>>[parse(x) for x in class_dates]
[datetime.datetime(2017, 8, 25, 0, 0),
 datetime.datetime(2017, 9, 1, 0, 0),
 datetime.datetime(2017, 9, 8, 0, 0),
 datetime.datetime(2017, 9, 15, 0, 0),
 datetime.datetime(2017, 9, 22, 0, 0),
 datetime.datetime(2017, 9, 29, 0, 0)]
```

Use parse, but designate that the day is first.

```
>>>parse (fall_start, dayfirst=True)
datetime.datetime(2017, 8, 21, 0, 0)
```

Create a dataframe. A DataFrame is a tabular data structure comprised of rows and columns, akin to a spreadsheet, database table. DataFrame as a group of Series objects that share an index (the column names).

```
>>>import pandas as pd
>>>data = {'class_dates': ['8/25/2017 18:47:05.069722', '9/1/2017 18:47:05.119994',
                           '9/8/2017 18:47:05.178768', '9/15/2017 18:47:05.230071',
                           '9/22/2017 18:47:05.230071', '9/29/2017 18:47:05.280592'],
           'complete': [1, 0, 1, 1, 0, 1]}
>>>df = pd.DataFrame(data, columns = ['class_dates', 'complete'])
>>>print(df)
      class_dates  complete
0  8/25/2017 18:47:05.069722        1
1   9/1/2017 18:47:05.119994        0
2   9/8/2017 18:47:05.178768        1
3   9/15/2017 18:47:05.230071        1
4   9/22/2017 18:47:05.230071        0
5   9/29/2017 18:47:05.280592        1
```

Convert df['date'] from string to datetime

```
>>>import pandas as pd
>>>pd.to_datetime(df['class_dates'])
0    2017-08-25 18:47:05.069722
1    2017-09-01 18:47:05.119994
```

```

2 2017-09-08 18:47:05.178768
3 2017-09-15 18:47:05.230071
4 2017-09-22 18:47:05.230071
5 2017-09-29 18:47:05.280592
Name: class_dates, dtype: datetime64[ns]

```

9.6 Control Statements

9.6.1 Comparision

Computer programs do not only execute instructions. Occasionally, a choice needs to be made. Such as a choice is based on a condition. Python has several conditional operators:

```

> greater than
< smaller than
== equals
!= is not

```

Conditions are always combined with variables. A program can make a choice using the if keyword. For example:

```

>>> x = int(input("Guess x:"))
>>> if x == 4:
...     print('You guessed correctly!')
... <ENTER>

```

In this example, *You guessed correctly!* will only be printed if the variable x equals to four (see table above). Python can also execute multiple conditions using the elif and else keywords.

```

>>> x = int(input("Guess x:"))
>>> if x == 4:
...     print('You guessed correctly!')
... elif abs(4 - x) == 1:
...     print('Wrong guess, but you are close!')
... else:
...     print('Wrong guess')
... <ENTER>

```

9.6.2 Iteration

To repeat code, the for keyword can be used. For example, to display the numbers from 1 to 10, we could write something like this:

```

>>> for i in range(1, 11):
...     print('Hello!')

```

The second argument to range, *11*, is not inclusive, meaning that the loop will only get to *10* before it finishes. Python itself starts counting from 0, so this code will also work:

```

>>> for i in range(0, 10):
...     print(i + 1)

```

In fact, the range function defaults to starting value of *0*, so the above is equivalent to:

```

>>> for i in range(10):

```

```
...     print(i + 1)
```

We can also nest loops inside each other:

```
>>> for i in range(0,10):
...     for j in range(0,10):
...         print(i, ', ',j)
... <ENTER>
```

In this case we have two nested loops. The code will iterate over the entire coordinate range (0,0) to (9,9)

9.7 Datatypes

9.7.1 Lists

see: https://www.tutorialspoint.com/python/python_lists.htm

Lists in Python are ordered sequences of elements, where each element can be accessed using a 0-based index.

To define a list, you simply list its elements between square brackets ‘[]’:

```
>>> names = ['Albert', 'Jane', 'Liz', 'John', 'Abby']
>>> names[0] # access the first element of the list
'Albert'
>>> names[2] # access the third element of the list
'Liz'
```

You can also use a negative index if you want to start counting elements from the end of the list. Thus, the last element has index -1, the second before last element has index -2 and so on:

```
>>> names[-1] # access the last element of the list
'Abby'
>>> names[-2] # access the second last element of the list
'John'
```

Python also allows you to take whole slices of the list by specifying a beginning and end of the slice separated by a colon ‘::’:

```
::
>>> names[1:-1] # the middle elements, excluding first and last ['Jane', 'Liz', 'John']
```

As you can see from the example above, the starting index in the slice is inclusive and the ending one, exclusive.

Python provides a variety of methods for manipulating the members of a list.

You can add elements with append‘:

```
>>> names.append('Liz')
>>> names
['Albert', 'Jane', 'Liz', 'John', 'Abby', 'Liz']
```

As you can see, the elements in a list need not be unique.

Merge two lists with ‘extend‘:

```
>>> names.extend(['Lindsay', 'Connor'])
>>> names
['Albert', 'Jane', 'Liz', 'John', 'Abby', 'Liz', 'Lindsay', 'Connor']
```

Find the index of the first occurrence of an element with ‘index’:

```
>>> names.index('Liz')
2
```

Remove elements by value with ‘remove’:

```
>>> names.remove('Abby')
>>> names
['Albert', 'Jane', 'Liz', 'John', 'Liz', 'Lindsay', 'Connor']
```

Remove elements by index with ‘pop’:

```
>>> names.pop(1)
'Jane'
>>> names
['Albert', 'Liz', 'John', 'Liz', 'Lindsay', 'Connor']
```

Notice that pop returns the element being removed, while remove does not.

If you are familiar with stacks from other programming languages, you can use insert and ‘pop’:

```
>>> names.insert(0, 'Lincoln')
>>> names
['Lincoln', 'Albert', 'Liz', 'John', 'Liz', 'Lindsay', 'Connor']
>>> names.pop()
'Connor'
>>> names
['Lincoln', 'Albert', 'Liz', 'John', 'Liz', 'Lindsay']
```

The Python documentation contains a full list of list operations.

To go back to the range function you used earlier, it simply creates a list of numbers:

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(2, 10, 2)
[2, 4, 6, 8]
```

9.7.2 Sets

Python lists can contain duplicates as you saw above:

```
>>> names = ['Albert', 'Jane', 'Liz', 'John', 'Abby', 'Liz']
```

When we don’t want this to be the case, we can use a set:

```
>>> unique_names = set(names)
>>> unique_names
set(['Lincoln', 'John', 'Albert', 'Liz', 'Lindsay'])
```

Keep in mind that the *set* is an unordered collection of objects, thus we can not access them by index:

```
>>> unique_names[0]
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'set' object does not support indexing
```

However, we can convert a set to a list easily:

```
>>> unique_names = list(unique_names) >>> unique_names ['Lincoln', 'John', 'Albert', 'Liz',
'Lindsay'] >>> unique_names[0] 'Lincoln'
```

Notice that in this case, the order of elements in the new list matches the order in which the elements were displayed when we create the set (we had set(['Lincoln', 'John', 'Albert', 'Liz', 'Lindsay'])) and now we have ['Lincoln', 'John', 'Albert', 'Liz', 'Lindsay']). You should not assume this is the case in general. That is, don't make any assumptions about the order of elements in a set when it is converted to any type of sequential data structure.

You can change a set's contents using the add, remove and update methods which correspond to the append, remove and extend methods in a list. In addition to these, *set* objects support the operations you may be familiar with from mathematical sets: *union*, *intersection*, *difference*, as well as operations to check containment. You can read about this in the Python documentation for sets.

9.7.3 Removal and Testing for Membership in Sets

One important advantage of a *set* over a *list* is that **access to elements is fast**. If you are familiar with different data structures from a Computer Science class, the Python list is implemented by an array, while the set is implemented by a hash table.

We will demonstrate this with an example. Let's say we have a list and a set of the same number of elements (approximately 100 thousand):

```
>>> import sys, random, timeit
>>> nums_set = set([random.randint(0, sys.maxint) for _ in range(10**5)])
>>> nums_list = list(nums_set)
>>> len(nums_set)
100000
```

We will use the *timeit* Python module to time 100 operations that test for the existence of a member in either the list or set:

```
>>> timeit.timeit('random.randint(0, sys.maxint) in nums', setup='import random; nums=%s' %
0.0004038810729980469
>>> timeit.timeit('random.randint(0, sys.maxint) in nums', setup='import random; nums=%s' %
0.3980541229248047
```

The exact duration of the operations on your system will be different, but the take away will be the same: searching for an element in a set is orders of magnitude faster than in a list. This is important to keep in mind when you work with large amounts of data.

9.7.4 Dictionaries

One of the very important data structures in python is a dictionary also referred to as *dict*.

A dictionary represents a key value store:

```
>>> person = {'Name': 'Albert', 'Age': 100, 'Class': 'Scientist'}
>>> print("person['Name']: ", person['Name'])
```

```
person['Name']: Albert
>>> print("person['Age']: ", person['Age'])
person['Age']: 100
```

You can delete elements with the following commands:

```
>>> del person['Name'] # remove entry with key 'Name'
>>> person
{'Age': 100, 'Class': 'Scientist'}
>>> person.clear()      # remove all entries in dict
>>> person
{}
>>> del person          # delete entire dictionary
>>> person
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    NameError: name 'person' is not defined
```

You can iterate over a dict:

```
>>> person = {'Name': 'Albert', 'Age': 100, 'Class': 'Scientist'}
>>> for item in person:
...     print(item, person[item])
...     <ENTER>
Age 100
Name Albert
Class Scientist
```

9.7.5 Dictionary Keys and Values

You can retrieve both the keys and values of a dictionary using the `keys()` and `values()` methods of the dictionary, respectively:

```
>>> person.keys()
['Age', 'Name', 'Class']
>>> person.values()
[100, 'Albert', 'Scientist']
```

Both methods return lists. Notice, however, that the order in which the elements appear in the returned lists (Age, Name, Class) is different from the order in which we listed the elements when we declared the dictionary initially (Name, Age, Class). It is important to keep this in mind: **you can't make any assumptions about the order in which the elements of a dictionary will be returned by the `keys()` and `values()` methods.**

However, you can assume that if you call `keys()` and `values()` in sequence, the order of elements will at least correspond in both methods. In the above example Age corresponds to 100, Name to 'Albert', and Class to Scientist, and you will observe the same correspondence in general as long as **keys() and values() are called one right after the other.**

9.7.6 Counting with Dictionaries

One application of dictionaries that frequently comes up is counting the elements in a sequence. For example, say we have a sequence of coin flips:

```
>>> import random
>>> die_rolls = [random.choice(['heads', 'tails']) for _ in range(10)]
>>> die_rolls
['heads', 'tails', 'heads', 'tails', 'heads', 'heads', 'tails', 'heads', 'heads', 'heads']
```

The actual list die_rolls will likely be different when you execute this on your computer since the outcomes of the die rolls are random.

To compute the probabilities of heads and tails, we could count how many heads and tails we have in the list:

```
>>> counts = {'heads': 0, 'tails': 0}
>>> for outcome in coin_flips:
...     assert outcome in counts
...     counts[outcome] += 1
...     <ENTER>
>>> print('Probability of heads: %.2f' % (counts['heads'] / len(coin_flips)))
Probability of heads: 0.70
>>> print('Probability of tails: %.2f' % (counts['tails'] / sum(counts.values())))
Probability of tails: 0.30
```

In addition to how we use the dictionary counts to count the elements of coin_flips, notice a couple things about this example:

1. We used the assert outcome in counts statement. The assert statement in Python allows you to easily insert debugging statements in your code to help you discover errors more quickly. assert statements are executed whenever the internal Python `__debug__` variable is set to True, which is always the case unless you start Python with the -O option which allows you to run *optimized* Python.
2. When we computed the probability of tails, we used the built-in sum function, which allowed us to quickly find the total number of coin flips. sum is one of many built-in function you can read about here.

9.8 Functions

You can reuse code by putting it inside a function that you can call in other parts of your programs. Functions are also a good way of grouping code that logically belongs together in one coherent whole. A function has a unique name in the program. Once you call a function, it will execute its body which consists of one or more lines of code:

```
def check_triangle(a, b, c):
    return \
        a < b + c and a > abs(b - c) and \
        b < a + c and b > abs(a - c) and \
        c < a + b and c > abs(a - b)

print(check_triangle(4, 5, 6))
```

The def keyword tells Python we are defining a function. As part of the definition, we have the function name, `check_triangle`, and the parameters of the function – variables that will be populated when the function is called.

We call the function with arguments 4, 5 and 6, which are passed in order into the parameters a,

b and c. A function can be called several times with varying parameters. There is no limit to the number of function calls.

It is also possible to store the output of a function in a variable, so it can be reused.

```
def check_triangle(a, b, c):
    return \
        a < b + c and a > abs(b - c) and \
        b < a + c and b > abs(a - c) and \
        c < a + b and c > abs(a - b)

result = check_triangle(4, 5, 6)
print(result)
```

9.9 Classes

A class is an encapsulation of data and the processes that work on them. The data is represented in member variables, and the processes are defined in the methods of the class (methods are functions inside the class). For example, let's see how to define a Triangle class:

```
class Triangle(object):

    def __init__(self, length, width, height, angle1, angle2, angle3):
        if not self._sides_ok(length, width, height):
            print('The sides of the triangle are invalid.')
        elif not self._angles_ok(angle1, angle2, angle3):
            print('The angles of the triangle are invalid.')

        self._length = length
        self._width = width
        self._height = height

        self._angle1 = angle1
        self._angle2 = angle2
        self._angle3 = angle3

    def _sides_ok(self, a, b, c):
        return \
            a < b + c and a > abs(b - c) and \
            b < a + c and b > abs(a - c) and \
            c < a + b and c > abs(a - b)

    def _angles_ok(self, a, b, c):
        return a + b + c == 180

triangle = Triangle(4, 5, 6, 35, 65, 80)
```

Python has full object-oriented programming (OOP) capabilities, however we can not cover all of them in a quick tutorial, so please refer to the Python docs on classes and OOP.

9.10 Database Access

see: https://www.tutorialspoint.com/python/python_database_access.htm

9.11 Modules

Make sure you are no longer in the interactive interpreter. If you are you can type quit() and press Enter to exit.

You can save your programs to files which the interpreter can then execute. This has the benefit of allowing you to track changes made to your programs and sharing them with other people.

Start by opening a new file hello.py in the Python editor of your choice. If you don't have a preferred editor, we recommend PyCharm.

Now write this simple program and save it:

```
from __future__ import print_statement, division
print("Hello world!")
```

As a check, make sure the file contains the expected contents on the command line:

```
$ cat hello.py
from __future__ import print_statement, division
print("Hello world!")
```

To execute your program pass the file as a parameter to the python command:

```
$ python hello.py
Hello world!
```

Files in which Python code is stored are called **modules**. You can execute a Python module from the command line like you just did, or you can import it in other Python code using the import statement.

Let's write a more involved Python program that will receive as input the lengths of the three sides of a triangle, and will output whether they define a valid triangle. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides and greater than the difference of the lengths of the other two sides.:

```
"""Usage: check_triangle.py [-h] LENGTH WIDTH HEIGHT
```

Check if a triangle is valid.

Arguments:

LENGTH	The length of the triangle.
WIDTH	The width of the triangle.
HEIGHT	The height of the triangle.

Options:

-h	--help
"""	

```
from __future__ import print_function, division
from docopt import docopt
```

```

if __name__ == '__main__':
    args = docopt(__doc__)
    a, b, c = int(args['LENGTH']), int(args['WIDTH']), int(args['HEIGHT'])
    valid_triangle = \
        a < b + c and a > abs(b - c) and \
        b < a + c and b > abs(a - c) and \
        c < a + b and c > abs(a - b)
    print('Triangle with sides %d, %d and %d is valid: %r' %
          a, b, c, valid_triangle
    )

```

Assuming we save the program in a file called `check_triangle.py`, we can run it like so:

```
$ python check_triangle.py 4 5 6
Triangle with sides 4, 5 and 6 is valid: True
```

Let break this down a bit.

1. We are importing the `print_function` and `division` modules from Python 3 like we did earlier in this tutorial. It's a good idea to always include these in your programs.
2. We've defined a boolean expression that tells us if the sides that were input define a valid triangle. The result of the expression is stored in the `valid_triangle` variable. Inside are `true`, and `False` otherwise.
3. We've used the backslash symbol `\` to format our code nicely. The backslash simply indicates that the current line is being continued on the next line.
4. When we run the program, we do the check if `__name__ == '__main__'`. `__name__` is an internal Python variable that allows us to tell whether the current file is being run from the command line (value `__name__`), or is being imported by a module (the value will be the name of the module). Thus, with this statement we're just making sure the program is being run by the command line.
5. We are using the `docopt` module to handle command line arguments. The advantage of using this module is that it generates a usage help statement for the program and enforces command line arguments automatically. All of this is done by parsing the docstring at the top of the file.
6. In the `print` function, we are using Python's string formatting capabilities to insert values into the string we are displaying.

9.12 Installing Libraries

Often you may need functionality that is not present in Python's standard library. In this case you have two options:

- implement the features yourself
- use a third-party library that has the desired features.

Often you can find a previous implementation of what you need. Since this is a common situation, there is a service supporting it: the Python Package Index (or PyPi for short).

Our task here is to install the `autopep8` tool from PyPi. This will allow us to illustrate the use of virtual environments using the `pyenv` or `virtualenv` command, and installing and uninstalling PyPi packages using `pip`.

9.13 Using pip to Install Packages

Let's now look at another important tool for Python development: the Python Package Index, or PyPI for short. PyPI provides a large set of third-party python packages. If you want to do something in python, first check pypi, as odd are someone already ran into the problem and created a package solving it.

In order to install package from PyPI, use the pip command. We can search for PyPI for packages:

```
$ pip search --trusted-host pypi.python.org autopep8 pylint
```

It appears that the top two results are what we want so install them:

```
$ pip install --trusted-host pypi.python.org autopep8 pylint
```

This will cause pip to download the packages from PyPI, extract them, check their dependencies and install those as needed, then install the requested packages.

You can skip ‘–trusted-host pypi.python.org’ option if you have patched urllib3 on Python 2.7.9.

9.14 GUI

9.14.1 GUIZero

Install guizero with the following command:

```
sudo pip3 install guizero
```

For a comprehensive tutorial on guizero, click [here](#).

9.14.2 Kivy

You can install Kivy on OSX as follows:

```
brew install pkg-config sdl2 sdl2_image sdl2_ttf sdl2_mixer gstreamer
pip install -U Cython
pip install kivy
pip install pygame
```

A hello world program for kivy is included in the cloudmesh.robot repository. Which you can fine here

- <https://github.com/cloudmesh/cloudmesh.robot/tree/master/projects/kivy>

To run the program, please download it or execute it in cloudmesh.robot as follows:

```
cd cloudmesh.robot/projects/kivy
python swim.py
```

To create stand alone packages with kivy, please see:

- <https://kivy.org/docs/guide/packaging-osx.html>

9.15 Formatting and Checking Python Code

First, get the bad code:

```
$ wget --no-check-certificate http://git.io/pXqb -O bad_code_example.py
```

Examine the code:

```
$ emacs bad_code_example.py
```

As you can see, this is very dense and hard to read. Cleaning it up by hand would be a time-consuming and error-prone process. Luckily, this is a common problem so there exist a couple packages to help in this situation.

9.16 Using autopep8

We can now run the bad code through autopep8 to fix formatting problems:

```
$ autopep8 bad_code_example.py >code_example_autopep8.py
```

Let us look at the result. This is considerably better than before. It is easy to tell what the example1 and example2 functions are doing.

It is a good idea to develop a habit of using autopep8 in your python-development workflow. For instance: use autopep8 to check a file, and if it passes, make any changes in place using the -i flag:

```
$ autopep8 file.py      # check output to see of passes  
$ autopep8 -i file.py # update in place
```

If you use pyCharm you have the ability to use a similar function while pressing on Inspect Code.

9.17 Further Learning

There is much more to python than what we have covered here:

- conditional expression (if, if...then, ‘if..elif..then’)
- function definition(def)
- class definition (class)
- function positional arguments and keyword arguments
- lambda expression
- iterators
- generators
- loops
- docopts
- humanize

9.18 Writing Python 3 Compatible Code

To write python 2 and 3 compatible code we recommend that you take a look at: http://python-future.org/compatible_idioms.html

9.19 Using Python on FutureSystems

This is only important if you use FutureSystems resources.

In order to use Python you must log into your FutureSystems account. Then at the shell prompt execute the following command:

```
$ module load python
```

This will make the python and virtualenv commands available to you.

The details of what the module load command does are described in the future lesson modules.

9.20 Ecosystem

9.20.1 pypi

Link: [pypi](#)

The Python Package Index is a large repository of software for the Python programming language containing a large number of packages [link]. The nice think about pipy is that many packages can be installed with the program ‘pip’.

To do so you have to locate the <package_name> for example with the search function in pypi and say on the commandline:

```
pip install <package_name>
```

where pagage_name is the string name of the package. an example would be the package called cloudmesh_client which you can install with:

```
pip install cloudmesh_client
```

If all goes well the package will be installed.

9.20.2 Alternative Installations

The basic installation of python is provided by python.org. However others claim to have alternative environments that allow you to install python. This includes

- Canopy
- Anaconda
- IronPython

Typically they include not only the python compiler but also several useful packages. It is fine to use such environments for the class, but it should be noted that in both cases not every python library may be available for install in the given environment. For example if you need to use cloudmesh client, it may not be available as conda or Canopy package. This is also the case for many other cloud related and useful python libraries. Hence, we do recommend that if you are new to python to use the distribution from python.org, and use pip and virtualenv.

Additionally some python version have platform specific libraries or dependencies. For example coca libraries, .NET or other frameworks are examples. For the assignments and the projects such platform dependent libraries are not to be used.

If however you can write a platform independent code that works on Linux, OSX and Windows while using the python.org version but develop it with any of the other tools that is just fine. However it is up to you to guarantee that this independence is maintained and implemented. You do have to write requirements.txt files that will install the necessary python libraries in a platform independent fashion. The homework assignment PRG1 has even a requirement to do so.

In order to provide platform independence we have given in the class a “minimal” python version that we have tested with hundreds of students: python.org. If you use any other version, that is your

decision. Additionally some students not only use python.org but have used iPython which is fine too. However this class is not only about python, but also about how to have your code run on any platform. The homework is designed so that you can identify a setup that works for you.

However we have concerns if you for example wanted to use chameleon cloud which we require you to access with cloudmesh. cloudmesh is not available as conda, canopy, or other framework package. Cloudmesh client is available from pypi which is standard and should be supported by the frameworks. We have not tested cloudmesh on any other python version than python.org which is the open source community standard. None of the other versions are standard.

In fact we had students over the summer using canopy on their machines and they got confused as they now had multiple python versions and did not know how to switch between them and activate the correct version. Certainly if you know how to do that, than feel free to use canopy, and if you want to use canopy all this is up to you. However the homework and project requires you to make your program portable to python.org. If you know how to do that even if you use canopy, anaconda, or any other python version that is fine. Graders will test your programs on a python.org installation and not canopy, anaconda, ironpython while using virtualenv. It is obvious why. If you do not know that answer you may want to think about that every time they test a program they need to do a new virtualenv and run vanilla python in it. If we were to run two installs in the same system, this will not work as we do not know if one student will cause a side effect for another. Thus we as instructors do not just have to look at your code but code of hundreds of students with different setups. This is a non scalable solution as every time we test out code from a student we would have to wipe out the OS, install it new, install a new version of whatever python you have elected, become familiar with that version and so on and on. This is the reason why the open source community is using python.org. We follow best practices. Using other versions is not a community best practice, but may work for an individual.

We have however in regards to using other python version additional bonus projects such as

- deploy run and document cloudmesh on ironpython
- deploy run and document cloudmesh on anaconda, develop script to generate a conda package from github
- deploy run and document cloudmesh on canopy, develop script to generate a conda package from github
- deploy run and document cloudmesh on ironpython
- other documentation that would be useful

9.20.3 Autoenv: Directory-based Environments

We do not recommend that you use autoenv. Instead we recommend that you use pyenv.

Link: Autoenv <<https://pypi.python.org/pypi/autoenv/0.2.0>>

If a directory contains a .env file, it will automatically be executed when you cd into it. It's easy to use and install.

This is useful for

- auto-activating virtualenvs
- project-specific environment variables

To use it add the ENV you created with virtualenv into .env file within your project directory:

```
$ echo "source ~/ENV/bin/activate" > yourproject/.env
$ echo "echo 'whoa'" > yourproject/.env
```

```
$ cd project  
whoa
```

To install it on Mac OS X use Homebrew:

```
$ brew install autoenv  
$ echo "source $(brew --prefix autoenv)/activate.sh" >> ~/.bash_profile
```

To install it using pip use:

```
$ pip install autoenv  
$ echo "source `which activate.sh`" >> ~/.bashrc
```

To install it using git use:

```
$ git clone git://github.com/kennethreitz/autoenv.git ~/.autoenv  
$ echo 'source ~/.autoenv/activate.sh' >> ~/.bashrc
```

Before sourcing activate.sh, you can set the following variables:

- ‘AUTOENV_AUTH_FILE’: Authorized env files, defaults to ~/.autoenvAuthorized
- ‘AUTOENV_ENV_FILENAME’: Name of the .env file, defaults to .env
- ‘AUTOENV_LOWER_FIRST’: Set this variable to flip the order of .env files executed

Autoenv overrides cd. If you already do this, invoke autoenv_init within your custom cd after sourcing activate.sh.

Autoenv can be disabled via unset cd if you experience I/O issues with certain file systems, particularly those that are FUSE-based (such as smbnetfs).

9.21 Resources

If you are unfamiliar with programming in Python, we also refer you to some of the numerous online resources. You may wish to start with Learn Python or the book Learn Python the Hard Way. Other options include Tutorials Point or Code Academy, and the Python wiki page contains a long list of references for learning as well. Additional resources include:

- <https://virtualenvwrapper.readthedocs.io>
- <https://github.com/yyuu/pyenv>
- <https://amaral.northwestern.edu/resources/guides/pyenv-tutorial>
- <https://godjango.com/96-django-and-python-3-how-to-setup-pyenv-for-multiple-pythons/>
- <https://www.accelebrate.com/blog/the-many-faces-of-python-and-how-to-manage-them/>
- <http://ivory.idyll.org/articles/advanced-swc/>
- <http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html>
- <http://www.youtube.com/watch?v=0vJJ1VBVTFg>
- <http://www.korokithakis.net/tutorials/python/>
- <http://www.afterhoursprogramming.com/tutorial/Python/Introduction/>
- <http://www.greenteapress.com/thinkpython/thinkCSpy.pdf>
- <https://docs.python.org/3.3/tutorial/modules.html>
- https://www.learnpython.org/en/Modules/_and_/_Packages
- <https://docs.python.org/2/library/datetime.html>
- https://chrisalbon.com/python/strings/_to/_datetime.html

A very long list of useful information are also available from

- <https://github.com/vinta/awesome-python>

- https://github.com/rasbt/python_reference

This list may be useful as it also contains links to data visualization and manipulation libraries, and AI tools and libraries. Please note that for this class you can reuse such libraries if not otherwise stated.

9.22 Jupyter Notebook Tutorials

A Short Introduction to Jupyter Notebooks and NumPy To view the notebook, open this link in a background tab <<https://nbviewer.jupyter.org/>> and copy and paste the following link in the URL input area <<https://cloudmesh.github.io/classes/lesson/prg/Jupyter-NumPy-tutorial-I523.ipynb>> Then hit Go!

9.23 Exercises

E_{Python.1}: Write a python program called iterate.py that accepts an integer n from the command line. Pass this integer to a function called iterate.

The iterate function should then iterate from 1 to n. If the ith number is a multiple of three, print “multiple of 3”, if a multiple of 5 print “multiple of 5”, if a multiple of both print “multiple of 3 and 5”, else print the value.

- E_{Python.2}:**
1. Create a pyenv or virtualenv ~/ENV
 2. Modify your ~/.bashrc shell file to activate your environment upon login.
 3. Install the docopt python package using pip
 4. Write a program that uses docopt to define a commandline program. Hint: modify the iterate program.
 5. Demonstrate the program works and submit the code and output.



10. Cloudmesh Command Shell

F chapter/python-cmd5.tex

10.1 CMD5

Python's CMD (<https://docs.python.org/2/library/cmd.html>) is a very useful package to create command line shells. However it does not allow the dynamic integration of newly defined commands. Furthermore, additions to CMD need to be done within the same source tree. To simplify developing commands by a number of people and to have a dynamic plugin mechanism, we developed cmd5. It is a rewrite on our earlier efforts in cloudmesh client and cmd3.

10.1.1 Resources

The source code for cmd5 is located in github:

- <https://github.com/cloudmesh/cmd5>

10.1.2 Creating a Python Development Environment

We recommend that you use a virtualenv either with virtualenv or pyenv. This is in detail documented in the Section [section_pyenv](#).

10.1.3 Installation from source

Cmd5 can easily deployed with pip:

```
pip install cloudmesh.cmd5
```

In case you like to generate easily new cmd 5 commands we also recommend you install the cloudmesh sys command with:

```
pip install cloudmesh.sys
```

In case you like to work with the source please clone the following directories from github:

```
mkdir -p ~/github  
cd ~/github
```

```
git clone https://github.com/cloudmesh/cloudmesh.common.git  
git clone https://github.com/cloudmesh/cloudmesh.cmd5.git  
git clone https://github.com/cloudmesh/cloudmesh.sys.git
```

```
cd ~/github/cloudmesh.common  
python setup.py install  
pip install .
```

```
cd ~/github/cloudmesh.cmd5  
python setup.py install  
pip install .
```

```
cd ~/github/cloudmesh.sys  
python setup.py install  
pip install .
```

The common directory contains some useful libraries, the cmd5 repository contains the shell, while the sys directory contains a command to generate extensions to cloudmesh.

10.1.4 Execution

To run the shell you can activate it with the cms command. cms stands for cloudmesh shell:

```
(ENV2) $ cms
```

It will print the banner and enter the shell:

```
+-----+  
| / _ _ | | _ _ - - - - | | _ _ - - - - | | _ _ | | _ _ | | _ _ | | _ _ | | | | | | | | | |
| | | | | | / _ \ | | | | / _ ' | | ' _ \ | | / _ \ | | / _ \ | | ' _ \ | |  
| | | _ _ | | ( ) | | _ | | ( ) | | | | | | | | _ / \ _ \ | | | | | |  
| \ _ _ | | \ _ / \ _ , _ \ _ , | | | | | | | | | | | | | | | | | | | |  
+-----+  
| Cloudmesh CMD5 Shell |  
+-----+
```

```
cms>
```

To see the list of commands you can say:

```
cms> help
```

To see the manula page for a specific command, please use:

```
help COMMANDNAME
```

10.1.5 Create your own Extension

One of the most important features of CMD5 is its ability to extend it with new commands. This is done via packaged name spaces. We recommend you name is cloudmesh.mycommand, where mycommand is the name of the command that you like to create. This can easily be done while using the sys command:

```
cms sys command generate mycommand
```

It will download a template form cloudmesh called cloudmesh.bar and generate a new directory cloudmesh.mycommand with all the needed files to create your own command and register it dynamically with cloudmesh. All you have to do is to cd into the directory and install the code:

```
cd cloudmesh.mycommand  
python setup.py install  
pip install .
```

Adding your own command is easy. It is important that all objects are defined in the command itself and that no global variables be use in order to allow each shell command to stand alone. Naturally you should develop API libraries outside of the cloudmesh shell command and reuse them in order to keep the command code as small as possible. We place the command in:

```
cloudmesh/mycommand/command/mycommand.py
```

An example for the bar command is presented at:

- <https://github.com/cloudmesh/cloudmesh/blob/master/cloudmesh/bar/command/bar.py>

It shows how simple the command definition is (bar.py):

```
from __future__ import print_function
from cloudmesh.shell.command import command
from cloudmesh.shell.command import PluginCommand

class BarCommand(PluginCommand):

    @command
    def do_bar(self, args, arguments):
        """
        ::

            Usage:
                command -f FILE
                command FILE
                command list

            This command does some useful things.

        Arguments:
            FILE      a file name
        Options:
            -f        specify the file
        ...
        print(arguments)
```

An important difference to other CMD solutions is that our commands can leverage (besides the standrad definition), docopts as a way to define the manual page. This allows us to use arguments as dict and use simple if conditions to interpret the command. Using docopts has the advantage that contributors are forced to think about the command and its options and document them from the start. Previously we did not use but argparse and click. However we noticed that for our contributors both systems lead to commands that were either not properly documented or the developers delivered ambiguous commands that resulted in confusion and wrong ussage by subsequent users. Hence, we do recommend that you use docopts for documenting cmd5 commands. The transformation is enabled by the @command decorator that generates a manual page and creates a proper help message for the shell automatically. Thus there is no need to introduce a sepaarte help method as would normally be needed in CMD while reducing the effort it takes to contribute new commands in a dynamic fashion.

10.1.6 Excercise

ECMD5.1: Install cmd5 on your computer.

ECMD5.2: Write a new command with your firstname as the command name.

ECMD5.3: Write a new command and experiment with docopt syntax and argument interpretation of the dict with if conditions.

ECMD5.4: If you have useful extensions that you like us to add by default, please work with us.

 chapter/lesson/code.tex

10.2 Code Management

TBD

VI

Technologies

11 Technology Overview 195

- 11.1 Workflow-Orchestration
- 11.2 Application and Analytics
- 11.3 Application Hosting Frameworks
- 11.4 High level Programming
- 11.5 Streams
- 11.6 Basic Programming Model and Runtime, SPMD, MapReduce
- 11.7 Inter process communication Collectives
- 11.8 In-memory databases/caches
- 11.9 Object-relational mapping
- 11.10 Extraction Tools
- 11.11 SQL(NewSQL)
- 11.12 NoSQL
- 11.13 File management
- 11.14 Data Transport
- 11.15 Cluster Resource Management
- 11.16 File systems
- 11.17 Interoperability
- 11.18 DevOps
- 11.19 IaaS Management from HPC to hypervisors
- 11.20 Cross-Cutting Functions
- 11.21 Message and Data Protocols
- 11.22 Technologies To Be Integrated
- 11.23 Excercise

 chapter/i524/technologies.tex



11. Technology Overview

In this section we find a number of technologies that are related to big data. Certainly a number of these projects are hosted as an Apache project. One important resource for a general list of all apache projects is at

Apache projects: <https://projects.apache.org/projects.html?category>

11.1 Workflow-Orchestration

11.1.1 ODE

Apache ODE (Orchestration Director Engine) is an open source implementation of the WS-BPEL 2.0 standard. WS- BPEL which stands for Web Services Business Process Execution Language, is an executable language for writing business processes with web services [102]. It includes control structures like conditions or loops as well as elements to invoke web services and receive messages from services. ODE uses WSDL (Web Services Description Language) for interfacing with web services [434]. Naming a few of its features, It supports two communication layers for interacting with the outside world, one based on Axis2 (Web Services http transport) and another one based on the JBI standard. It also supports both long and short living process executions for orchestrating services for applications [433].

11.1.2 ActiveBPEL

Business Process Execution Language for Web Services (BPEL4WS or just BPEL) is an XML-based grammar for describing the logic to coordinate and control web services that seamlessly integrate people, processes and systems, increasing the efficiency and visibility of the business. ActiveBPEL is a robust Java/J2EE runtime environment that is capable of executing process definitions created to the Business Process Execution Language for Web Services. The ActiveBPEL also provides an administration interface that is accessible via web service invocations;and it can

also be used to administer, to control and to integrate web services into a larger application [10].

11.1.3 Airavata

Apache Airavata [202] is a software framework that enables you to compose, manage, execute, and monitor large scale applications and workflows on distributed computing resources such as local clusters, supercomputers, computational grids, and computing clouds. Scientific gateway developers use Airavata as their middleware layer between job submissions and grid systems. Airavata supports long running applications and workflows on distributed computational resources. Many scientific gateways are already using Airavata to perform computations (e.g. Ultrascan [488], SEAGrid [305] and GenApp [201]).

11.1.4 Pegasus

The Pegasus [465] is workflow management system that allows to compose and execute a workflow in an application in different environment without the need for any modifications. It allows users to make high level workflow without thinking about the low level details. It locates the required input data and computational resources automatically. Pegasus also maintains information about tasks done and data produced. In case of errors Pegasus tries to recover by retrying the whole workflow and providing check pointing at workflow-level. It cleans up the storage as the workflow gets executed so that data-intensive workflows can have enough required space to execute on storage-constrained resources. Some of the other advantages of Pegasus are: scalability, reliability and high performance. Pegasus has been used in many scientific domains like astronomy, bioinformatics, earthquake science , ocean science, gravitational wave physics and others.

11.1.5 Kepler

Kepler, scientific workflow application, is designed to help scientist, analyst, and computer programmer create, execute and share models and analyses across a broad range of scientific and engineering disciplines. Kepler can operate on data stored in a variety of formats, locally and over the internet, and is an effective environment for integrating disparate software components such as merging *R* scripts with compiled *C* code, or facilitating remote, distributed execution of models. Using Kepler's GUI, users can simply select and then connect pertinent analytical components and data sources to create a *scientific workflow*. Overall, the Kepler helps users share and reuse data, workflow, and components developed by the scientific community to address common needs [346].

11.1.6 Swift

Swift is a general-purpose, multi-paradigm, compiled programming language. It has been developed by Apple Inc. for iOS, macOS, watchOS, tvOS, and Linux. This programming language is intended to be more robust and resilient to erroneous code than Objective-C, and more concise. It has been built with the LLVM compiler framework included in Xcode 6 and later and, on platforms other than Linux. C, Objective-C, C++ and Swift code can be run within one program as Swift uses the Objective-C runtime library [697].

Swift supports the core concepts that made Objective-C flexible, notably dynamic dispatch, widespread late binding, extensible programming and similar features. Swift features have well-known safety and performance trade-offs. A system that helps address common programming errors like null pointers was introduced to enhance safety. Apple has invested considerable effort in

aggressive optimization that can flatten out method calls and accessors to eliminate this overhead to handle performance issues.

11.1.7 Taverna

Taverna is workflow management system. According to [580], Taverna is transitioning to Apache Incubator as of Jan 2017. Taverna suite includes 2 products:

1. Taverna Workbench is desktop client where user can define the workflow.
2. Taverna Server is responsible for executing the remote workflows.

Taverna workflows can also be executed on command-line. Taverna supports wide range of services including WSDL-style and RESTful Web Services, BioMart, SoapLab, R, and Excel. Taverna also support mechanism to monitor the running workflows using its web browser interface. In the [617] paper, the formal syntax and operational semantics of Taverna is explained.

11.1.8 Triana

Triana is an open source problem solving software that comes with powerful data analysis tools [615]. Having been developed at Cardiff University, it has a good and easy-to-understand User Interface and is typically used for signal, text and image processing. Although it has its own set of analysis tools, it can also easily be integrated with custom tools. Some of the already available toolkits include signal-analysis toolkit, an image-manipulation toolkit, etc. Besides, it also checks the data types and reports the usage of any incompatible tools. It also reports errors, if any, as well as useful debug messages in order to resolve them. It also helps track serious bugs, so that the program does not crash. It has two modes of representing the data - a text-editor window or a graph-display window. The graph-display window has the added advantage of being able to zoom in on particular features. Triana is specially useful for automating the repetitive tasks, like finding-and-replacing a character or a string.

11.1.9 Trident

In [204], it is explained that Apache Trident is a “high-level abstraction for doing realtime computing on top of [Apache] Storm.” Similarly to Apache Storm, Apache Trident was developed by Twitter. Furthermore, [204] introduces Trident as a tool that “allows you to seamlessly intermix high throughput (millions of messages per second), stateful stream processing with low latency distributed querying.” In [203], the five kinds of operations in Trident are described as “Operations that apply locally to each partition and cause no network transfer”, “repartitioning operations that repartition a stream but otherwise don’t change the contents (involves network transfer)”, “aggregation operations that do network transfer as part of the operation”, “operations on grouped streams” and “merges and joins.” In [204], these five kinds of operations (i.e. joins, aggregations, grouping, functions, and filters) and the general concepts of Apache Trident are described as similar to “high level batch processing tools like Pig or Cascading.”

11.1.10 BioKepler

BioKepler is a Kepler module of scientific workflow components to execute a set of bioinformatics tools using distributed execution patterns [97]. It contains a specialized set of actors called “bioActors” for running bioinformatic tools, directors providing distributed data-parallel(DPP) execution on Big Data platforms such as Hadoop and Spark they are also configurable and reusable

[96]. BioKepler contains over 40 example workflows that demonstrate the actors and directors [371].

11.1.11 Galaxy

Ansible Galaxy is a website platform and command line tool that enables users to discover, create, and share community developed roles. Users' GitHub accounts are used for authentication, allowing users to import roles to share with the ansible community. [211] describes how Ansible roles are encapsulated and reusable tools for organizing automation content. Thus a role contains all tasks, variables, and handlers that are necessary to complete that role. [275] depicts roles as the most powerful part of Ansible as they keep playbooks simple and readable. "They provide reusable definitions that you can include whenever you need and customize with any variables that the role exposes." [212] provides the project documents for Ansible Galaxy on github.

11.1.12 Jupyter and IPython

The Jupyter Notebook is a language-agnostic HTML notebook web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text [487]. The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results [226]. The Jupyter notebook combines two components:

1. A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.
2. Notebook documents: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

Notebooks may be exported to a range of static formats, including HTML (for example, for blog posts), reStructuredText, LaTeX, PDF, and slide shows, via the nbconvert command [227]. Notebook documents contains the inputs and outputs of a interactive session as well as additional text that accompanies the code but is not meant for execution [598]. In this way, notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics, and rich representations of resulting objects [312]. These documents are internally JSON files and are saved with the .ipynb extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues [652].

11.1.13 Dryad

Dryad is a general-purpose distributed execution engine for coarse-grain data-parallel applications. According to [164] it was created with the objective of automatically managing scheduling, distribution, fault tolerance etc. Dryad concentrates on the throughput instead of latency and it assumes that a private data centre is used. It creates a dataflow graph by using computational 'vertices' and communication 'channels'. The computational vertices are written using C++ base classes and objects. During runtime, the dataflow graph is parallelized by distributing the vertices across multiple processor cores on the same computer or different physical computers connected by a network. The Dryad runtime handles this scheduling without any explicit intervention. The data flow from one vertex to another is realized by TCP/IP streams, shared memory, or temporary files. In the directed acyclic graph created by Dryad, each vertex is a program and the edges represent

data channels. Each graph is represented as $G = (VG, EG, IG, OG)$ in [165] where VG is a sequence of vertices with EG directed edges and two sets IG is a subset of VG and OG is a subset of VG that indicate the input and output vertices respectively. Other technologies used for the same purpose as Dryad include Map Reduce, MPI etc.

11.1.14 Naiad

Naiad [410] is a distributed system based on computational model called *Timely Dataflow* developed for execution of data-parallel, cyclic dataflow programs. It provides an in-memory distributed dataflow framework which exposes control over data partitioning and enables features like the high throughput of batch processors, the low latency of stream processors, and the ability to perform iterative and incremental computations. The Naiad architecture consists of two main components: (1) incremental processing of incoming updates and (2) low-latency real-time querying of the application state.

Compared to other systems supporting loops or streaming computation, Naiad provides support for the combination of the two, nesting loops inside streaming contexts and indeed other loops, while maintaining a clean separation between the many reasons new records may flow through the computation [605].

This model enriches dataflow computation with timestamps that represent logical points in the computation and provide the basis for an efficient, lightweight coordination mechanism. All the above capabilities in one package allows development of High-level programming models on Naiad which can perform tasks as streaming data analysis, iterative machine learning, and interactive graph mining. On the contrary, it's public reusable low-level programming abstractions leads Naiad to outperforms many other data parallel systems that enforce a single high-level programming model.

11.1.15 Oozie

Oozie is a workflow manager and scheduler. Oozie is designed to scale in a Hadoop cluster. Each job will be launched from a different datanode [316] [655]. Oozie [441] is architected from the ground up for large-scale Hadoop workflow. Scales to meet the demand, provides a multi-tenant service, is secure to protect data and processing, and can be operated cost effectively. As demand for workflow and the sophistication of applications increase, it must continue to mature in these areas [316]. Is well integrated with Hadoop security. Is the only workflow manager with built-in Hadoop actions, making workflow development, maintenance and troubleshooting easier. It's UI makes it easier to drill down to specific errors in the data nodes. Proven to scale in some of the world's largest clusters [316]. Gets callbacks from MapReduce jobs so it knows when they finish and whether they hang without expensive polling. Oozie Coordinator allows triggering actions when files arrive at HDFS. Also supported by Hadoop vendors [316].

11.1.16 Tez

Apache Tez is open source distributed execution framework build for writing native YARN application. It provides architecture which allows user to convert complex computation as dataflow graphs and the distributed engine to handle the directed acyclic graph for processing large amount of data. It is highly customizable and pluggable so that it can be used as a platform for various application. It is used by the Apache Hive, Pig as execution engine to increase the performance of map reduce functionality [586]. Tez focuses on running application efficiently on Hadoop

cluster leaving the end user to concentrate only on its business logic. Tez provides features like distributed parallel execution on hadoop cluster, horizontal scalability, resource elasticity, shared library reusable components and security features. Tez provides capability to naturally map the algorithm into the hadoop cluster execution engine and it also provides the interface for interaction with different data sources and configurations.

Tez is client side application and just needs Tez client to be pointed to Tez jar libraries path makes it easy and quick to deploy. User can have multiple tez version running concurrently. Tez provides DAG API's which lets user define structure for the computation and Runtime API's which contain the logic or code that needs to be executed in each transformation or task.

11.1.17 Google FlumeJava

FlumeJava [195] is a java library that allows users to develop and run data parallel pipelines. Its goal is to allow a programmer to express his data-parallel computations in a clear way while simultaneously executing it in the best possible optimized manner. The MapReduce function eases the task of data parallelism. However, a pipeline of MapReduce functions is desired by many real time computation systems. FlumeJava provides these abstractions of data parallel computations by providing support for pipelined execution. To provide optimized parallel execution, FlumeJava defers the execution of these pipelines and instead constructs an execution plan dataflow graph depending on the results needed by each stage of the pipeline. “When the final results of the parallel operations are eventually needed, FlumeJava first optimizes the execution plan, and then executes the optimized operations on appropriate underlying primitives” [113]. FlumeJava library is written on top of the collection framework in Java.

When developing a large pipeline, it is timeconsuming to find a bug in the later stages and then re-compile and re-evaluate all the operations. FlumeJava library supports a cached execution mode to aid in this scenario. In this mode, it automatically creates temporary files to hold the outputs of each operation it executes [113]. Thus, rather than recomputing all the operations once the pipeline has been rectified to fix all the bugs, it simply reads the output from these temporary files and later deletes them once they are no longer in use.

11.1.18 Crunch

Arvados Crunch [147] is a containerized workflow engine for running complex, multi-part pipelines or workflows in a way that is flexible, scalable, and supports versioning, reproducibility, and provenance while running in virtualized computing environments. The Arvados Crunch [146] framework is designed to support processing very large data batches (gigabytes to terabytes) efficiently. Arvados Crunch increases concurrency by running tasks asynchronously, using many CPUs and network interfaces at once (especially beneficial for CPU-bound and I/O-bound tasks respectively). Crunch also tracks inputs, outputs, and settings so you can verify that the inputs, settings, and sequence of programs you used to arrive at an output is really what you think it was. Crunch ensures that your programs and workflows are repeatable with different versions of your code, OS updates, etc. and allows you to interrupt and resume long-running jobs consisting of many short tasks and maintains timing statistics automatically.

11.1.19 Cascading

[109] Cascading software authored by Chris Wensel is development platform for building the application in Hadoop. It basically act as an abstraction for Apache Hadoop used for creating

complex data processing workflow using the scalability of hadoop however hiding the complexity of mapReduce jobs. User can write their program in java without having knowledge of mapReduce. Applications written on cascading are portable.

Cascading Benefits 1. With Cascading application can be scaled as per the data sets. 2. Easily Portable 3. Single jar file for application deployment.

11.1.20 Scalding

11.1.21 e-Science Central

In [278], it is explained that e-Science Central is designed to address some of the pitfalls within current Infrastructure as a Service (e.g. Amazon EC2) and Platform as a Service (e.g. force.com) services. For instance, in [278], the “majority of potential scientific users, access to raw hardware is of little use as they lack the skills and resources needed to design, develop and maintain the robust, scalable applications they require” and furthermore “current platforms focus on services required for business applications, rather than those needed for scientific data storage and analysis.” In [75], it is explained that e-Science Central is a “cloud based platform for data analysis” which is “portable and can be run on Amazon AWS, Windows Azure or your own hardware.” In [278], e-Science Central is further described as a platform, which “provides both Software and Platform as a Service for scientific data management, analysis and collaboration.” This collaborative platform is designed to be scalable while also maintaining ease of use for scientists. In [278], “a project consisting of chemical modeling by cancer researchers” demonstrates how e-Science Central “allows scientists to upload data, edit and run workflows, and share results in the cloud.”

11.1.22 Azure Data Factory

Azure data factory is a cloud based data integration service that can ingest data from various sources, transform/ process data and publish the result data to the data stores. A data management gateway enables access to data on SQL Databases [535]. The data processing is done by It works by creating pipelines to transform the raw data into a format that can be readily used by BI Tools or applications. The services comes with rich visualization aids that aid data analysis. Data Factory supports two types of activities: data movement activities and data transformation activities. Data Movement [538] is a Copy Activity in Data Factory that copies data from a data source to a Data sink. Data Factory supports the following data stores. Data from any source can be written to any sink. Data Transformation: Azure Data Factory supports the following transformation activities such as Map reduce, Hive transformations and Machine learning activities. Data factory is a great tool to analyze web data, sensor data and geo-spatial data.

11.1.23 Google Cloud Dataflow

Google Cloud Dataflow is a unified programming model and a managed service for developing and executing a wide variety of data processing patterns (pipelines). Dataflow includes SDKs for defining data processing workflows and a Cloud platform managed services to run those workflows on a Google cloud platform resources such as Compute Engine, BigQuery amongst others [236]. Dataflow pipelines can operate in both batch and streaming mode. The platform resources are provided on demand, allowing users to scale to meet their requirements, it's also optimized to help balance lagging work dynamically.

Being a cloud offering, Dataflow is designed to allow users to focus on devising proper analysis without worrying about the installation and maintaining [320] the underlying data piping and

process infrastructure.

11.1.24 NiFi (NSA)

[64] Defines NiFi as “An Easy to use, powerful and reliable system to process and distribute data”. This tool aims at automated data flow from sources with different sizes , formats and following diffent protocols to the centralized location or destination [47].

This comes equipped with an easy use UI where the data flow can be controlled with a drag and a drop. NiFi was initiatially developed by NSA (called Niagarafiles) using the concepts of flowbased programming and latter submitted to Apache Software foundation [429].

11.1.25 Jitterbit

Jitterbit [330] is an integration tool that delivers a quick, flexible and simpler approach to design, configure, test, and deploy integration solutions. It delivers powerful, flexible, and easy to use integration solutions that connect modern on premise, cloud, social, and mobile infrastructures. Jitterbit employs high performance parallel processing algorithms to handle large data sets commonly found in ETL initiatives [329]. This allows easy synchronization of disparate computing platforms quickly. The Data Cleansing and Smart Reconstruction tools provides complete reliability in data extraction, transformation and loading.

Jitterbit employs a no-code GUI (graphical user interface) and work with diverse applications such as : ETL (extract-transform-load), SaaS (Software as a Service),SOA (service-oriented architecture).

Thus it provides centralized platform with power to control all data. It supports many document types and protocols: XML, web services, database, LDAP, text, FTP, HTTP(S), Flat and Hierarchic file structures and file shares [331]. It is available for Linux and Windows, and is also offered through Amazon EC2 (Amazon Elastic Compute Cloud). Jitterbit Data Loader for Salesforce is a free data migration tool that enables Salesforce administrators automated import and export of data between flat files, databases and Salesforce.

11.1.26 Talend

Talend is Apache Software Foundation sponsor Big data integration tool design to ease the development and integration and management of big data, Talend provides well optimised auto generated code to load transform, enrich and cleanse data inside Hadoop, where one don't need to learn write and maintain Hadoop and spark code. The product has 900+ inbuild components feature data integration

Talend features multiple products that simplify the digital transformation tools such as Big data integration, Data integration, Data Quality, Data Preparation, Cloud Integration, Application Integration, Master Data management, Metadata Manager. Talend Integration cloud is secure and managed integration Platform-as-a-service (iPaaS), for connecting, cleansing and sharing cloud on premise data.

11.1.27 Pentaho

Pentaho is a business intelligence corporation that provides data mining, reporting, dashboarding and data integration capabilities. Generally, organizations tend to obtain meaningful relationships and useful information from the data present with them. Pentaho addresses the obstacles that

obstruct them from doing so [466]. The platform includes a wide range of tools that analyze, explore, visualize and predict data easily which simplifies blending any data. The sole objective of pentaho is to translate data into value. Being an open and extensible source, pentaho provides big data tools to extract, prepare and blend any data [34]. Along with this, the visualizations and analytics will help in changing the path that the organizations follow to run their business. From spark and hadoop to noSQL, pentaho transforms big data into big insights.

11.1.28 Apatar

11.1.29 Docker Compose

Docker is an open-source container based technology. A container allows a developer to package up an application and all its part includig the stack it runs on, dependencies it is associated with and everything the application requirs to run within an isolated enviornment . Docker seperates Application from the underlying Operating System in a similar way as Virtual Machines seperates the Operating System from the underlying Hardware. Dockerizing an application is very lightweight in comparison with running the application on the Virtual Machine as all the containers share the same underlying kernel, the Host OS should be same as the container OS (eliminating guest OS) and an average machine cannot have more than few VMs running o them.

Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands [387]. You can use Machine to create Docker hosts on your local Mac or Windows box, on your company network, in your data center, or on cloud providers like AWS or Digital Ocean. For Docker 1.12 or higher swarm mode is integerated with the Docker Engine, but on the older versions with Machine's swarm option, we can configure a swarm cluster. Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine. With these pooled resources “you can scale out your application as if it were running on a single, huge computer” [159] as swarm can be scaled upto 1000 Nodes or upto 50,000 containers

11.1.30 KeystoneML

A framework for building and deploying large-scale machine-learning pipelines within Apache Spark. It captures and optimizes the end-to-end large-scale machine learning applications for high-throughput training in a distributed environment with a high-level API [560]. This approach increases ease of use and higher performance over existing systems for large scale learning [560]. It is designed to be a faster and more sophisticated alternative to SparkML, the machine learning framework that's a full member of the Apache Spark club. Whereas SparkML comes with a basic set of operators for processing text and numbers, KeystoneML includes a richer set of operators and algorithms designed specifically for natural language processing, computer vision, and speech processing [561]. It has enriched set of operations for complex domains:vision,NLP,Speech, plus,advanced math And is Integrated with new BDAS technologies: Velox, ml-matrix, soon Planck, TuPAQ and Sample Clean [702].

11.2 Application and Analytics

11.2.1 Mahout (383)

“Apache Mahout software provides three major features: (1) A simple and extensible programming environment and framework for building scalable algorithms (2) A wide variety of premade algo-

rithms for Scala + Apache Spark, H2O, Apache Flink (3) Samsara, a vector math experimentation environment with R-like syntax which works at scale”

11.2.2 MLLib

MLlib is Apache Spark’s scalable machine learning library [401]. Its goal is to make machine learning scalable and easy. MLlib provides various tools such as, algorithms, feature extraction, utilities for data handling and tools for constructing, evaluating, and tuning machine learning pipelines. MLlib uses the linear algebra package Breeze, which depends on netlib-java for optimized numerical processing. MLlib is shipped with Spark and supports several languages which provides functionality for wide range of learning settings. MLlib library includes Java, Scala and Python APIs and is released as a part of Spark project under the Apache 2.0 license [393].

11.2.3 MLbase

MLBase [158] is a distributed machine learning system built with Apache Spark [541]. Machine Learning (ML) and Statistical analysis are tools for extracting insights from big data. MLbase is a tool for execute machine learning algorithms on a scalable platform. It consist of three components MLLib, MLI and ML Optimizer. MLLib was initially developed as a part of MLBase project but is now a part of Apache Spark. MLI is an experimental API for developing ML algorithm and to extract information. It provides high-level abstraction to the core ML algorithms. A prototype is currently implemented against Spark. ML optimizer on the other hand is use to automate the MLI pipeline construction. It solves for the search problem over feature extractors and ML algorithms included in MLI and ML lib. This library is its in early stage and under active development. Publications like [562], [357] and [579] are available on distributed machine learning with MLBase.

11.2.4 DataFu

The Apache DataFu project was created out of the need for stable, well-tested libraries for large scale data processing in Hadoop. As detailed in [149] Apache DatFu consists of two libraries Apache DataFu Pig and Apache DataFu Hourglass. Apache DataFu Pig is a collection of useful user-defined functions for data analysis in Apache Pig. The functions are in areas of Statistics, Bag Operations, Set Operations, Sessions, Sampling, Estimation, Hashing and Link Analysis. Apache DataFu Hourglass is a library for incrementally processing data using Hadoop MapReduce. It is designed to make computations over sliding windows more efficient. For these types of computations, the input data is partitioned in some way, usually according to time, and the range of input data to process is adjusted as new data arrives. Hourglass works with input data that is partitioned by day, as this is a common scheme for partitioning temporal data.

11.2.5 R

R, a GNU project, is a successor to S - a statistical programming language. It offers a range of capabilities – “programming language, high level graphics, interfaces to other languages and debugging”. “R is an integrated suite of software facilities for data manipulation, calculation and graphical display”. The statistical and graphical techniques provided by R make it popular in the statistical community. The statistical techniques provided include linear and nonlinear modelling, classical statistical tests, time-series analysis, classification and clustering to name a few [497]. The number of packages available in R has made it popular for use in machine learning, visualization, and data operations tasks like data extraction, cleaning, loading, transformation, analysis, modeling

and visualization. Its strength lies in analyzing data using its rich library but falls short when working with very large datasets [485].

11.2.6 pbdR

Programming with Big Data in R (pbdR) [457] is an environment having series of R packages for statistical computing with Big Data using high-performance statistical computation. It uses R, a popular language between statisticians and data miners. *pbdR* focuses on distributed memory system, where data is distributed across several machines and processed in batch mode. It uses MPI for inter process communications. R focuses on single machines for data analysis using a interactive GUI. Currently there are two implementation of pbdR, one Rmpi and another being pdbMpI. Rmpi uses SPMD parallelism while pbdRMpi uses manager/worker parallelism.

11.2.7 Bioconductor

Bioconductor is an open source and open development platform used for analysis and understanding of high throughput genomic data. Bioconductor is used to analyze DNA microarray, flow, sequencing, SNP, and other biological data. All contributions to Bioconductor are under an open source license. [222] describes the goals of Bioconductor “include fostering collaborative development and widespread use of innovative software, reducing barriers to entry into interdisciplinary scientific research, and promoting the achievement of remote reproducibility of research results” [95] described that Bioconductor is primarily based on R, as most components of Bioconductor are released in R packages. Extensive documentation is provided for each Bioconductor package as vignettes, which include task-oriented descriptions for the functionalities of each package. Bioconductor has annotation functionality to associate “genomic data in real time with biological metadata from web databases such as GenBank, Entrez genes and PubMed.” Bioconductor also has tools to process genomic annotation data.

11.2.8 ImageJ

ImageJ is a Java-based image processing program developed at the National Institutes of Health (NIH). ImageJ was designed with an open architecture that provides extensibility via Java plugins and recordable macros. Using ImageJ’s built-in editor and a Java compiler, it has enabled to solve many image processing and analysis problems in scientific research from three-dimensional live-cell imaging to radiological image processing. ImageJ’s plugin architecture and built-in development environment has made it a popular platform for teaching image processing [302].

11.2.9 OpenCV

OpenCV stands for Open source Computer Vision. It was designed for computational efficiency and with a strong focus on real-time applications. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. It can take advantage of the hardware acceleration of the underlying heterogeneous compute platform as it is enabled with OpenCL(Open Computing Language) [1]. OpenCV 3.2 is the latest version of the software that is currently available [2].

11.2.10 Scalapack

ScaLAPACK is a library of high-performance linear algebra routines for parallel distributed memory machines. It solves dense and banded linear systems, least squares problems, eigenvalue problems,

and singular value problems. It is designed for heterogeneous computing and is portable on any computer that supports Message Passing Interface or Parallel Virtual Machine [527].

ScaLAPACK is a open source software package and is available from netlib via anonymous ftp and the World Wide Web. It contains driver routines for solving standard types of problems, computational routines to perform a distinct computational task, and auxiliary routines to perform a certain subtask or common low-level computation. ScaLAPACK routines are based on block-partitioned algorithms in order to minimize the frequency of data movement between different levels of the memory hierarchy.

11.2.11 PetSc

11.2.12 PLASMA MAGMA

PLASMA is built to address the performance shortcomings of the LAPACK and ScaLAPACK libraries on multicore processors and multi-socket systems of multicore processors and their inability to efficiently utilize accelerators such as Graphics Processing Units (GPUs). Real arithmetic and complex arithmetic are supported in both single precision and double precision. PLASMA has been designed by restructuring the software to achieve much greater efficiency, where possible, on modern computers based on multicore processors. PLASMA does not support band matrices and does not solve eigenvalue and singular value problems. Also, PLASMA does not replace ScaLAPACK as software for distributed memory computers, since it only supports shared-memory machines [106] [476]. Recent activities of major chip manufacturers, such as Intel, AMD, IBM and NVIDIA, make it more evident than ever that future designs of microprocessors and large HPC systems will be hybrid/heterogeneous in nature, relying on the integration (in varying proportions) of two major types of components: [161] [267] 1. Many-cores CPU technology, where the number of cores will continue to escalate because of the desire to pack more and more components on a chip while avoiding the power wall, instruction level parallelism wall, and the memory wall; 2. Special purpose hardware and accelerators, especially Graphics Processing Units (GPUs), which are in commodity production, have outpaced standard CPUs in floating point performance in recent years, and have become as easy, if not easier to program than multicore CPUs. While the relative balance between these component types in future designs is not clear, and will likely to vary over time, there seems to be no doubt that future generations of computer systems, ranging from laptops to supercomputers, will consist of a composition of heterogeneous components [359][613][614].

11.2.13 Azure Machine Learning

Azure Machine Learning is a cloud based service that can be used to do predictive analytics, machine learning or data mining. It has features like in-built algorithm library, machine learning studio and a webservice [260]. In built algorithm library has implementation of various popular machine learning algorithms like decision tree, SVM, linear regression, neural networks etc. Machine learning studio facilitates creation of predictive models using graphical user interface by dragging, dropping and connecting of different modules that can be used by people with minimal knowledge in the machine learning field. Machine learning studio is a free service for basic version and comes with a monthly charge for advanced versions. Apart from building models, studio also has options to do preprocessing like clean, transform and normalize the data. Webservice provides option to deploy the machine learning algorithm as ready to consume APIs that can be reused in future with minimal effort and can also be published.

11.2.14 Google Prediction API & Translation API

Google Prediction API & Translation API are part of Cloud ML API family with specific roles. Below is a description of each and their use.

Google Prediction API provides pattern-matching and machine learning capabilities. Built on HTTP and JSON, the prediction API uses training data to learn and consecutively use what has been learned to predict a numeric value or choose a category that describes new pieces of data. This makes it easier for any standard HTTP client to send requests to it and parse the responses. The API can be used to predict what users might like, categorize emails as spam or non-spam, assess whether posted comments sentiments are positive or negative or how much a user may spend in a day. Prediction API has a 6 month limited free trial or a paid use for \$10 per project which offers up to 10,000 predictions a day [238].

Google Translation API is a simple programmatic interface for translating an arbitrary string into any supported language. Google Translation API is highly responsive allowing websites and applications to integrate for fast dynamic translation of source text from source language to a target language. Translation API also automatically identifies and translate languages with a high accuracy from over a hundred different languages. Google Translation API is charged at \$20 per million characters making it an affordable localization solution. Translation API is also distributed in two editions, premium edition which is tailored for users with precise long-form translation services like livestream, high volumes of emails or detailed articles and documents. There's also standard edition which is tailored for short, real-time conversations [239].

mlpy is an open source python library made for providing machine learning functionality. It is built on top of popular existing python libraries of NumPy, SciPy and GNU scientific libraries (GSL). It also makes extensive use of Cython language. These form the prerequisites for mlpy. [18] explains the significance of its components: NumPy, SciPy provide sophisticated N-dimensional arrays, linear algebra functionality and a variety of learning methods, GSL, which is written in C, provides complex numerical calculation functionality.

mlpy provides a wide range of machine learning methods for both supervised and unsupervised learning problems. mlpy is multiplatform and works both on Python 2 and 3 and is distributed under GPL3. Mlpy provides both classic and new learning algorithms for classification, regression and dimensionality reduction. [402] provides a detailed list of functionality offered by mlpy. Though developed for general machine learning applications, mlpy has special applications in computational biology, particularly in functional genomics modeling.

11.2.15 scikit-learn

Scikit-learn is an open source library that provides simple and efficient tools for data analysis and data mining. It is accessible to everybody and reusable in various contexts. It is built on numpy, Scipy and matplotlib and is commercially usable as it is distributed under many linux distributions [105]. Through a consistent interface, scikit-learn provides a wide range of learning algorithms. Scikits are the names given to the modules for SciPy, a fundamental library for scientific computing and as these modules provide different learning algorithms, the library is named as scikit-learn [532]. It provides an in-depth focus on code quality, performance, collaboration and documentation. Most popular models provided by scikit-learn include clustering, cross-validation, dimensionality reduction, parameter tuning, feature selection and extraction.

11.2.16 PyBrain (529)

The goal of PyBrain is to provide flexible, easy-to-use algorithms that are not just simple but are also powerful for machine learning tasks. The algorithms implemented are Long Short-Term Memory (LSTM), policy gradient methods, (multidimensional) recurrent neural networks and deep belief networks. These algorithms include a variety of predefined environments and benchmarks to test and compare algorithms.

PyBrain provides a toolbox for supervised, unsupervised and reinforcement learning as well as black-box and multi-objective optimization as it is much larger than Python libraries.

PyBrain implements many recent learning algorithms and architectures while emphasizing on sequential and nonsequential data and tasks. These algorithms range from areas such as supervised learning and reinforcement learning to direct search / optimization and evolutionary methods. For application-oriented users, PyBrain contains reference implementations of a number of algorithms at the bleeding edge of research and this is in addition to standard algorithms which are not available in Python library. Besides this PyBrain sets itself apart by its versatility for composing custom neural networks architectures that range from (multi-dimensional) recurrent networks to restricted Boltzmann machines or convolutional networks.

11.2.17 ComLearn

ComLearn is a system that makes use of data compression methodologies for mining patterns in a large amount of data. So, it is basically a compression-based machine learning system. For identifying and learning different patterns, it provides a set of utilities which can be used in applying standard compression mechanisms. The most important characteristic of ComLearn is its power in mining patterns even in domains that are unrelated. It has the ability to identify and classify the language of different bodies of text [117]. This helps in reducing the work of providing background knowledge regarding a particular classification. It provides such generalization through a library that is written in ANSI C which is portable and works in many environments [117]. ComLearn provides immediate access to every core functionality in all the major languages as it is designed to be extensible.

11.2.18 DAAL(Intel)

DAAL stands for Data Analytics Acceleration Library. DAAL is a software library offered by Intel which is written in C++, Python, and Java which implements algorithms for doing efficient and optimized data analysis tasks to solve big-data problems [684]. The library is designed to use data platforms like Hadoop, Spark, R, and Matlab. The important algorithms which DAAL implements are 'Lower Order Moments' which is used to find out max, min, standard deviation of a dataset, 'Clustering' which is used to do unsupervised learning by grouping data into unlabelled group. It also includes 10-12 other important algorithms.

[307] It supports three processing modes namely batch processing, online processing and distributed processing. Intel DAAL addresses all stages of data analytics pipeline namely pre-processing, transformation, analysis, modelling, validation, and decision making.

11.2.19 Caffe

Caffe is a deep learning framework made with three terms namely expression, speed and modularity [726]. Using Expressive architecture, switching between CPU and GPU by setting a single flag to

train on a GPU machine then deploy to commodity cluster or mobile devices. Here the concept of configuration file will come without hard coding the values . Switching between CPU and GPU can be done by setting a flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

It can process over 60 million images per day with a single NVIIA k40 GPU It is being used by academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia.

11.2.20 Torch

Torch is an open source machine learning library, a scientific computing framework [669] .It implements LuaJIT programming language and implements C/CUDA. It implements N-dimensional array. It does routines of indexing, slicing, transposing etc. It has an interface to C language via scripting language LuaJIT. It supports different artificial intelligence models like neural network and energy based models. It is compatible with GPU. The core package is *torch*. It provides a flexible N dimensional array which supports basic routings. It has been used to build hardware implementation for data flows like those found in neural networks.

11.2.21 Theano

Theano is a Python library. It was written at the LISA lab. Initially it was created with the purpose to support efficient development of machine learning(ML) algorithms. Theano uses recent GPUs for higher speed. It is used to evaluate mathematical expressions and especially those mathematical expressions that include multi-dimensional arrays. Theano's working is dependent on combining aspects of a computer algebra system and an optimizing compiler. This combination of computer algebra system with optimized compilation is highly beneficial for the tasks which involves complicated mathematical expressions and that need to be evaluated repeatedly as evaluation speed is highly critical in such cases. It can also be used to generate customized C code for number of mathematical operations. For cases where many different expressions are there and each of them is evaluated just once, Theano can minimize the amount of compilation and analyses overhead [604].

11.2.22 DL4j

DL4j stands for Deeplearning4j [685]. It is a deep learning programming library written for Java and the Java virtual machine (JVM) and a computing framework with wide support for deep learning algorithms. Deeplearning4j includes implementations of the restricted Boltzmann machine, deep belief net, deep autoencoder, stacked denoising autoencoder and recursive neural tensor network, word2vec, doc2vec, and GloVe. These algorithms all include distributed parallel versions that integrate with Apache Hadoop and Spark. It is an open-source software released under Apache License 2.0.

Training with Deeplearning4j occurs in a cluster. Neural nets are trained in parallel via iterative reduce, which works on Hadoop-YARN and on Spark. Deeplearning4j also integrates with CUDA kernels to conduct pure GPU operations, and works with distributed GPUs.

11.2.23 H2O

It is an open source software for big data analysis. It was launched by the Start-up H2O in 2011. It provides an in-memory, distributed, fast and a scalable machine learning and predictive analytics

platform that allows the users to build machine learning models on big data [263]. It is written in Java. It is currently implemented in 5000 companies. It provides APIs for R(3.0.0 or later), Python(2.7.x, 3.5.x), Scala(1.4-1.6) and JSON [134]. The software also allows online scoring and modeling on a single platform. It is scalable and has a wide range of OS and language support. It works perfectly on the conventional operating systems, and big data systems such as Hadoop, Cloudera, MapReduce, HortonWorks. It can be used on cloud computing environments such as Amazon and Microsoft Azure [264].

11.2.24 IBM Watson

IBM Watson [299] is a super computer built on cognitive technology that processes information like the way human brain does by understanding the data in a natural language as well as analyzing structured and unstructured data. It was initially developed as a question and answer tool more specifically to answer questions on the quiz show *Jeopardy* but now it has been seen as helping doctors and nurses in the treatment of cancer. It was developed by IBM's DeepQA research team led by David Ferrucci. [300] illustrates that with Watson you can create bots that can engage in conversation with you. You can even provide personalized recommendations to Watson by understanding a user's personality, tone and emotion. Watson uses the Apache Hadoop framework in order to process the large volume of data needed to generate an answer by creating in-memory datasets used at run-time. Watson's DeepQA UIMA (Unstructured Information Management Architecture) annotators were deployed as mappers in the Hadoop Map-Reduce framework. Watson is written in multiple programming languages like Java, C++, Prolog and it runs on the SUSE Linux Enterprise Server. [300] mentions that today Watson is available as a set of open source APIs and Software As a Service product as well.

11.2.25 Oracle PGX

Numerous information is revealed from graphs. Information like direct and indirect relations or patterns in the elements of the data, can be easily seen through graphs. The analysis of graphs can unveil significant insights. Oracle PGX (Parallel Graph AnalytiX) is a toolkit for graph analysis. “It is a fast, parallel, in-memory graph analytic framework that allows users to load up their graph data, run analytic algorithms on them, and to browse or store the result” [454]. Graphs can be loaded from various sources like SQL and NoSQL databases, Apache Spark and Hadoop [460].

11.2.26 GraphLab

GraphLab [256] is a graph-based, distributed computation, high performance framework for machine learning written in C++. It is an open source project started by Prof. Carlos Guestrin of Carnegie Mellon University in 2009, designed considering the scale, variety and complexity of real world data. It integrates various high level algorithms such as Stochastic Gradient Descent, Gradient Descent & Locking and provides high performance experience. It includes scalable machine learning toolkits which has implementation for deep learning, factor machines, topic modeling, clustering, nearest neighbors and almost everything required to enhance machine learning models. This framework is targeted for sparse iterative graph algorithms. It helps data scientists and developers easily create and install applications at large scale.

11.2.27 GraphX

GraphX is Apache Spark's API for graph and graph-parallel computation [56].

GraphX provides:

Flexibility: It seamlessly works with both graphs and collections. GraphX unifies ETL, exploratory analysis, and iterative graph computation within a single system. You can view the same data as both graphs and collections, transform and join graphs with RDDs efficiently, and write custom iterative graph algorithms using the Pregel API.

Speed: Its performance is comparable to the fastest specialized graph processing systems while retaining Apache Spark's flexibility, fault tolerance, and ease of use.

Algorithms: GraphX comes with a variety of algorithms such as PageRank, Connected Components, Label propagations, SVD++, Strongly connected components and Triangle Count.

It combines the advantages of both data-parallel and graph-parallel systems by efficiently expressing graph computataion within the Spark data-parallel framework [724].

It gets developed as a part of Apache Spark project. It thus gets tested and updated with each Spark release.

11.2.28 IBM System G

IBM System G provides a set of Cloud and Graph computing tools and solutions for Big Data [293]. In fact, the G stands for Graph and typically spans a database, visualization, analytics library, middleware and Network Science Analytics tools. It assists the easy creating of graph stores and queries and exploring them via interactive visualizations [294]. Internally, it uses the property graph model for its working. It consists of five individual components - gShell, REST API, Python interface to gShell, Gremlin and a Visualizer. Some of the typical applications wherein it can be used include Expertise Location, Commerce, Recommendation, Watson, Cybersecurity, etc [373].

However, it is to be noted that the current version does not work in a distributed environment and it is planned that future versions would support it.

11.2.29 GraphBuilder(Intel)

Intel GraphBuilder for Apache Hadoop V2 is a software that is used to build graph data models easily enabling data scientists to concentrate more on the business solution rather than preparing/formatting the data. The software automates a)Data cleaning, b)transforming data and c)creating graph models with high throughput parallel processing using hadoop, with the help of prebuilt libraries. Intel Graph Builder helps to speed up the time to insight for data scientists by automating heavy custom workflows and also by removing the complexities of cluster computing for constructing graphs from Big Data. Intel Graph Building uses Apache Pig scripting language to simplify data preparation pipeline. “Intel Graph Builder also includes a connector that parallelizes the loading of the graph output into the Aurelius Titan open source graph database—which further speeds the graph processing pipeline through the final stage”. Finally being an open source there is a possibility of adding a load of functionalities by various contributors [308].

11.2.30 TinkerPop

TinkerPop is a graph computing framework from Apache software foundation [62]. Before coming under the Apache project, ThinkerPop was a stack of technologies like Blueprint, Pipes, Frames, Rexters, Furnace and Gremlin where each part was supporting graph-based application development. Now all parts are come under single TinkerPop project repo [167]. It uses Gremlin, a graph traversal machine and language. It allows user to write complex queries (traversal), that

can use for real-time transactional (OLTP) queries, graph analytic system (OLAP) or combination of both as in hybrid. Gremlin is written in java [59]. TinkerPop has an ability to create a graph in any size or complexity. Gremlin engine allows user to write graph traversal in Gremlin language, Python, JavaScript, Scala, Go, SQL and SPARQL. It is capable to adhere with small graph which requires a single machine or massive graphs that can only be possible with large cluster of machines, without changing the code.

11.2.31 Parasol

The parasol laboratory is a multidisciplinary research program founded at Texas A&M University with a focus on next generation computing languages. The core focus is centered around algorithm and application development to find solutions to data concentrated problems [461]. The developed applications are being applied in the following areas: computational biology, geophysics, neuroscience, physics, robotics, virtual reality and computer aided drug design(CAD). The program has organized a number of workshops and conferences in the areas such as software, intelligent systems, and parallel architecture.

11.2.32 Dream:Lab

DREAM:Lab stands for “Distributed Research on Emerging Applications and Machines Lab.” [459] DREAM:Lab is centered around distributed systems research to enable expeditious utilization of distributed data and computing systems [459]. DREAM:Lab utilizes the “capabilities of hundrededs of personal computers” to allow access to supercomputing resources to average individuals [502]. The DREAM:Lab pursues this goal by utilizing distributed computing [502]. Distributed computing consists of independent computing resources that communicate with each other over a network [154]. A large, complex computing problem is broken down into smaller, more manageable tasks and then these tasks are distributed to the various components of the distributed computing system [154].

11.2.33 Google Fusion Tables

Fusion Tables is a cloud based services, provided by Google for data management and integration. Fusion Tables allow users to upload the data in tabular format using data files like spreadsheet, CSV, KML, .tsv up to 250MB [241]. It used for data management, visualizing data (e.g. pie-charts, bar-charts, lineplot, scatterplot, timelines) [698] , sharing of tables, filter and aggregation the data. It allows user to take the data privately, within controlled collaborative group or in public. It allows to integrate the data from different tables from different users or tables. Fusion Table uses two-layer storage, Bigtable and Magastore. The information rows are stored in bigdata table called *Rows*, user can merge the multiple table in to one, from multiple users. “Megastore is a library on top of bigtable” [240]. Data visualization is one the feature, where user can see the visual representation of their data as soon as they upload it. User can store the data along with geospatial information as well.

11.2.34 CINET

A representation of connected entities such as “physical, biological and social phenomena” [119] predictive model. Network science has grown its importance understanding these phenomena Cyberinfrastructure is middleware tool helps study Network science, [343] “by providing unparalleled computational and analytic environment for researcher”.

Network science involves study of graph a large volume which requires high power computing which usually can't be achieved by desktop. Cyberinfrastructure provides cloud based infrastructure (e.g. FutureGrid) as well as use of HPC (e.g. Shadowfax, Pecos). With use of advanced intelligent Job managers, it selects the infrastructure smartly suitable for submitted job.

It provides structural and dynamic network analysis, has number of algorithms for "network analysis" such as shortest path, sub path, motif counting, centrality and graph traversal". CiNet has number of range of network visualization modules. CiNet is actively being used by several universities, researchers and analysts.

11.2.35 NWB

[432] NWB stands for Network workbench is analysis, modelling and visualization toolkit for the network scientists. It provides an environment which helps scientist researchers and practitioner to get online access to the shared resource environment and network datasets for analysis, modelling and visualization of large scale networking application. User can access this network datasets and algorithms previously obtained by doing lot of research and can also add their own datasets helping in speeding up the process and saving the time for redoing the same analysis.

NWB provides advanced tools for users to understand and interact with different types of networks. NWB members are largely the computer scientist, biologist, engineers, social and behavioural scientist. The platform helps the specialist researchers to transfer the knowledge within the broader scientific and research communities.

11.2.36 Elasticsearch

Elasticsearch [172] is a real time distributed, RESTful search and analytics engine which is capable of performing full text search operations for you. It is not just limited to full text search operations but it also allows you to analyze your data, perform CRUD operations on data, do basic text analysis including tokenization and filtering [174]. For example while developing an E-commerce website, Elasticsearch can be used to store the entire product catalog and inventory and can be used to provide search and autocomplete suggestions for the products. Elasticsearch is developed in Java and is an open source search engine which uses standard RESTful APIs and JSON on top of Apache's Lucene - which is a full text search engine library. Clinton Gormley & Zachary Tong [254] describes elastic search as "A distributed real time document store where every field is indexed and searchable". They also mention that "Elastic search is capable of scaling to hundreds of servers and petabytes of structured and unstructured data" [175]. mentions that Elastic search can be used on big data by using the Elasticsearch-Hadoop (ES-Hadoop) connector. ES-Hadoop connector lets you index the Hadoop data into the Elastic Stack to take full advantage of the Elasticsearch engine and returns output through Kibana visualizations [173]. A log parsing engine "Logstash" and analytics and visualization platform *Kibana* are also developed alongside Elasticsearch forming a single package.

11.2.37 Kibana

Kibana is an open source data visualization plugin for Elasticsearch [351]. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data [225]. The combination of Elasticsearch, Logstash, and Kibana (also known as ELK stack or Elastic stack) is available as products or service. Logstash provides an input stream to Elastic for storage and search, and Kibana

accesses the data for visualizations such as dashboards [282]. Elasticsearch is a search engine based on Lucene [176]. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Kibana makes it easy to understand large volumes of data. Its simple, browser-based interface enables you to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time [350] [310].

11.2.38 Logstash

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice [177]. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

11.2.39 Graylog

Graylog is an open source log management tool that allows an organization to assemble, organize and analyze large amounts of data from its network activity. It collects and aggregates events from a group of sources and presents data in a streamlines, simplified interface where one can drill down to significant metrics, identify key relationships, generate powerful data visualizations and derive actionable insights [380]. Graylog allows us to centrally collect and manage log messages of an organization's complete infrastructure [213]. A user can perform search on terabytes of log data to discover number of failed logins, find application errors across all servers or monitor the acivity of a suspicious user id. Graylog works on top of ElasticSearch and MongoDB to facilitate this high availability searching. Graylog provides visualization through creation of dashboards that allows a user to build pre-defined views on his data to assemble all of his important data only a single click away [257]. Any search result or metric shall be added as a widget on the dashboard to observe trends in one single location. These dashboards can also be shared with other users in the organization. Based on a user's recent search queries, graylog also allows you to distinguish data that are not searched upon very often and thus can be archived on cost effective storage drives. Users can also add certain trigger conditions that shall alert the system about performance issues, failed logins or exceptions in the flow of the application.

11.2.40 Splunk

Splunk is a platform for big data analytics. It is a software product that enables you to search, analyze, and visualize the machine-generated data gathered from the websites, applications, sensors, devices, and so on, that comprise your IT infrastructure or business [564]. After defining the data source, Splunk indexes the data stream and parses it into a series of individual events that you can view and search. It provides distributed search and MapReduce linearly scales search and reporting. It uses a standard API to connect directly to applications and devices. It was developed in response to the demand for comprehensible and actionable data reporting for executives outside a company's IT department [564].

11.2.41 Tableau

[577] Tableau is a family of interactive data visualization products focused on business intelligence. The different products which tableau has built are: Tableau Desktop, for individual use; Tableau Server for collaboration in an organization; Tableau Online, for Business Intelligence in the Cloud; Tableau Reader, for reading files saved in Tableau Desktop; Tableau Public, for journalists or anyone to publish interactive data online. [578]. Tableau uses VizQL as a visual query language for translating drag-and-drop actions into data queries and later expressing the data visually. Tableau also benefits from an Advanced In-Memory Technology for handling large amounts of data. The strengths of Tableau are mainly the ease of use and speed. However, it has a number of limitations, which the most prominent are unfitness for broad business and technical user, being closed-source, no predictive analytical capabilities and no support for expanded analytics.

11.2.42 D3.js

D3.js is a JavaScript library responsible for manipulating documents based on data. D3 helps in making data more interactive using HTML, SVG, and CSS. D3's emphasis on web standards makes it framework independent utilizing the full capabilities of modern browsers, combining powerful visualization components and a data-driven approach to DOM manipulation [148].

It assists in binding random data to a Document Object Model (DOM), followed by applying data-driven transformations to the document. It is very fast, supports large datasets and dynamic behaviours involving interaction and animation.

11.2.43 three.js

Three.js is an API library with about 650 contributions till date , where users can create and display an animated 3D computer graphics in a web browser. It is written in javascript and uses WebGL, HTML5 or SVG. Users can animate HTML elements using CSS3 or even import models from 3D modelling apps [606]. In order to display anything using three.js we need three basic features, which are scene, camera and renderer. This will result in rendering the scene with a camera. In addition to these three features , we can add animation, lights (ambience,spot lights, shadows), objects (lines , ribbons , particles) , geometry etc [607].

11.2.44 Potree

Potree [483] is a opensource tool powered by WebGL based viewer to visualize data from large point clouds. It started at the TU Wien, institute of Computer Graphics and Algorithms and currently begin continued under the Harvest4D project. Potree relies on reorganizing the point cloud data into an multi-resolution octree data structure which is time consuming. Its efficiency can be improved by using techniques such as divide and conquer as discussed in a conference paper Taming the beast: Free and Open Source massive cloud point cloud web visualization [386]. It has also been widely used in works involving spatio-temporal data where the changes in geographical features are across time [268].

11.2.45 DC.js

According to [153]: “DC.js is a javascript charting library with native crossfilter support, allowing exploration on large multi-dimensional datasets. It uses d3 to render charts in CSS-friendly SVG format. Charts rendered using dc.js are data driven and reactive and therefore provide instant

feedback to user interaction.” DC.js library can be used to perform data analysis on both mobile devices and different browsers. Under the dc namespace the following chart classes are included: barChart, boxplot, bubbleChart, bubbleOverlay, compositeChart, dataCount, dataGrid, dataTable, geoChoroplethChart, heatMap, legend, lineChart, numberDisplay, pieChart, rowChart, scatterPlot, selectMenu and seriesChart.

11.2.46 TensorFlow

TensorFlow is a platform that provides a software library for expressing and executing machine learning algorithms. [4] states TensorFlow has a flexible architecture allowing it to be executed with minimal change to many heterogeneous systems such as CPUs and GPUs of mobile devices, desktop machines, and servers. TensorFlow can “express a wide variety of algorithms, including training and inference algorithms for deep neural network models, and it has been used for conducting research and for deploying machine learning systems into production across more than a dozen areas”. [584] describes that TensorFlow utilizes data flow graphs in which the “nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.” TensorFlow was developed by the Google Brain Team and has a reference implementation that was released on 2015-11-09 under the Apache 2.0 open source license.

11.2.47 CNTK

The Microsoft Cognitive Toolkit - CNTK - is a unified deep-learning toolkit by Microsoft Research. It is in essence an implementation of Computational Network(CN) which supports both CPU and GPU. CNTK supports arbitrary valid computational networks and makes building DNNs, CNNs, RNNs, LSTMs, and other complicated networks as simple as describing the operations of the networks. The toolkit is implemented with efficiency in mind. It removes duplicate computations in both forward and backward passes, uses minimal memory needed and reduces memory reallocation by reusing them. It also speeds up the model training and evaluation by doing batch computation whenever possible [129]. It can be included as a library in your Python or C++ programs, or used as a standalone machine learning tool through its own model description language (BrainScript) [279]. Latest Version:2017-02-10. V 2.0 Beta 11 Release

11.3 Application Hosting Frameworks

11.3.1 Google App Engine

Google App Engine is a cloud computing platform to host your mobile or web applications on Google managed servers. Google App Engine provides automatic scaling for web applications, i.e it automatically allocates more resources to the application upon increase in the number of requests. It gives developers the freedom to focus on developing their code and not worry about the infrastructure. Google App Engine provides built-in services and APIs such as load balancing, automated security scanning, application logging, NoSQL datastores, memcache, and a user authentication API, that are a core part to most applications [65].

An App Engine platform can be run in either the Standard or the Flexible environment. Standard environment lays restrictions on the maximum number of resources an application can use and charges a user based on the instance hours used. The flexible environment as the name suggests provides higher flexibility in terms of resources and is charged based on the CPU and disk utilization. The App Engine requires developers to use only its supported languages and frameworks.

Supported languages are Java, Python, Ruby, Scala, PHP, GO, Node.js and other JVM oriented languages. The App Engine datastore uses a SQL like syntax called the GQL (Google Query Language) which works with non-relational databases when compared to SQL [66].

11.3.2 AppScale

AppScale is an application hosting platform. This platform helps to deploy and scale the unmodified Google App Engine application, which run the application on any cloud infrastructure in public, private and on premise cluster [68]. AppScale provide rapid, API development platform that can run on any cloud infrastructure. The platform separates the app logic and its service part to have control over application deployment, data storage, resource use, backup and migration. AppScale is based on Google's App Engine APIs and has support for Python, Go, PHP and Java applications. It supports single and multimode deployment, which will help with large, dataset or CPU. AppScale allows to deploy app in thee main mode i.e. dev/test, production and customize deployment [69].

11.3.3 Red Hat OpenShift

OpenShift was launched as a PaaS (Platform as a Service) by Red Hat in the Red Hat Summit, 2011 [512]. It is a cloud application development and hosting platform that envisages shifting of the developer's focus to development by automating the management and scaling of applications [509]. Thus, OpenShift [510] enables us to write our applications in any one web development language (using any framework) and it itself takes up the task of running the application on the web. This has its advantages and disadvantages - advantage being the developer doesn't have to worry about how the stuff works internally (as it is abstracted away) and the disadvantage being that he cannot control how it works, again because it is abstracted.

OpenShift is powered by Origin, which is in turn built using Docker container packaging and Kubernetes container cluster [511]. Due to this, OpenShift offers a lot of options, including online, on-premise and open source project options.

11.3.4 Heroku

Heroku [277] is a platform as a service that is used for building, delivering monitoring and scaling applications. It lets you develop and deploy application quickly without thinking about irrelevant problems such as infrastructure. Heroku also provides a secure and scalable database as a service with number of developers' tools like database followers, forking, data clips and automated health checks. It works by deploying to cedar stack [110], an online runtime environment that supports apps buit in Java, Node.js, Scala, Clojure, Python and PHP. It uses Git for version controlling. It is also tightly intergrated with Salesforce, providing seamless and smooth Heroku and Salesforce data synchronization enabling companies to develop and design creative apps that uses both platforms.

11.3.5 Aerobatic

According to [14]: Aerobatic is a platform that allows hosting static websites. It used to be an ad-on for Bitbucket but now Aerobatic is transitioning to standalone CLI(command Line Tool) and web dashboard . Aerobatic allows automatic builds to different branches. New changes to websites can be deployed using aero deploy command which can be executed from local desktop or any of CD tools and services like Jenkins, Codeship,Travis and so on. It also allows users to configure custom error pages and offers authentication which can also be customized. Aerobatic is backed by AWS cloud. Aerobatic has free plan and pro plan options for customers.

11.3.6 AWS Elastic Beanstalk

AWS Elastic Beanstalk is an orchestration service offered from Amazon Web Services which provides user with a platform for easy and quick deployment of their WebApps and services [27]. Amazon Elastic BeanStack automatically handles the deployment details of capacity provisioning by Amazon Cloud Watch, Elastic Load Balancing, Auto-scaling, and application health monitoring of the WebApps and service [339]. AWS Management Console allows the users to configure an automatic scaling mechanism of AWS Elastic Beanstalk. Elastic Load Balancing enables a load balancer, which automatically spreads the load across all running instances in an auto-scaling group based on metrics like request count and latency tracked by Amazon CloudWatch. Amazon CloudWatch tracks and stores per-instance metrics, including request count and latency, CPU, and RAM utilization. Elastic Beanstalk supports applications developed in Java, PHP, .NET, Node.js, Python, and Ruby as well as supports different container types for each language such as Apache Tomcat for Java applications, Apache HTTP Server for PHP applications Docker, GO and much more for specific languages where the container defines the infrastructure and software stack to be used for a given environment. “AWS Elastic Beanstalk runs on the Amazon Linux AMI and the Windows Server 2012 R2 AMI. Both AMIs are supported and maintained by Amazon Web Services and are designed to provide a stable, secure, and high-performance execution environment for Amazon EC2 Cloud computing”[27].

11.3.7 Azure

Microsoft Corporation (MSFT) markets its cloud products under the *Azure* brand name. At its most basic, Azure acts as an *infrastructure- as-a-service* (IaaS) provider. IaaS virtualizes hardware components, a key differentiation from other *-as-a-service* products. IaaS “abstract[s] the user from the details of infrasctructure like physical computing resources, location, data partitioning, scaling, security, backup, etc.” [682]

However, Azure offers a host of closely-related tool and products to enhance and improve the core product, such as raw block storage, load balancers, and IP addresses [397]. For instance, Azure users can access predictive analytics, Bots and Blockchain-as-a-Service [397] as well as more-basic computing, networking, storage, database and management components [396]. The Azure website shows twelve major categories under *Products* and twenty *Solution* categories, e.g., e-commerce or Business SaaS apps.

Azure competes against Amazon’s *Amazon Web Service*, [28] even though IBM (*SoftLayer* [296] and *Bluemix* [297]) and Google (*Google Cloud Platform*) [22] offer IaaS to the market. As of January 2017, Azure’s datacenters span 32 Microsoft-defined *regions*, or 38 *declared regions*, throughout the world. [397]

11.3.8 Cloud Foundry

It is an open source software with multi cloud application .It is a platform for running applications and services. It was originally developed by VMware and currently owned by Pivotal . It is written in Ruby and Go .It has a commercial version called Pivotal Cloud Foundry (PFC)[701]. Cloud Foundry is available as a stand alone software package, we can also deploy it to Amazon AWS as well as host it on OpenStack server , HP’s Helion or VMware’s vSphere as given in the blog [122] , it delivers quick application from development to deployment and is highly scalable. It has a DevOps friendly workflow. Cloud Foundry changes the way application and services are deployed and reduces the develop to deployment cycle time.

11.3.9 Pivotal

Pivotal Software, Inc. (Pivotal) is a software and services company. It offers multiple consulting and technology services, which includes Pivotal Web Services, which is an agile application hosting service. It has a single step upload feature *cf push*, another feature called Buildpacks lets us push applications written for any language like Java, Grails, Play, Spring, Node.js, Ruby on Rails, Sinatra or Go. Pivotal Web Services also allows developers to connect to 3rd party databases, email services, monitoring and more from the Marketplace. It also offers performance monitoring, active health monitoring, unified log streaming, web console built for team-based agile development [471].

11.3.10 IBM BlueMix

11.3.11 (Ninefold)

The Australian based cloud computing platform has shut down their services since January 30, 2016. Refer [426]

11.3.12 Jelastic

Jelastic (acronym for Java Elastic) is an unlimited PaaS and Container based IaaS within a single platform that provides high availability of applications, automatic vertical and horizontal scaling via containerization to software development clients, enterprise businesses, DevOps, System Admins, Developers, OEMs and web hosting providers [325]. Jelastic is a Platform-as-Infrastructure provider of Java and PHP hosting. It has international hosting partners and data centers. The company can add memory, CPU and disk space to meet customer needs. The main competitors of Jelastic are Google App Engine, Amazon Elastic Beanstalk, Heroku, and Cloud Foundry. Jelastic is unique in that it does not have limitations or code change requirements, and it offers automated vertical scaling, application lifecycle management, and availability from multiple hosting providers around the world [262].

11.3.13 Stackato

Hewlett Packard Enterprise or HPE Helion Stackato is a platform as a service(PaaS) cloud computing solution. The platform facilitates deployment of the user's application in the cloud and will function on top of an Infrastructure as a service(IaaS) [285]. Multiple cloud development is supported across AWS, vSphere, and Helion Openstack. The platform supports the following programming languages: native .NET support, java, Node.js, python, and ruby. This flexibility is advantageous compared to early PaaS solutions which would force the customer into utilizing a single stack. Additionally, this solution has the capacity to support private, public and hybrid clouds. [342] This capability user has to not have to make choices of flexibility over security of sensitive data when choosing a cloud computing platform.

11.3.14 appfog

According to [643], “AppFog is a platform as a service (PaaS) provider.” Platform as a service provides a platform for the development of web applications without the necessity of purchasing the software and infrastructure that supports it [345]. PaaS provides an environment for the creation of software [345]. The underlying support infrastructure that AppFog provides includes things such as runtime, middleware, o/s, virtualization, servers, storage, and networking [67]. AppFog is

based on VMWare's CloudFoundry project [643]. It gets things such as MySQL, Mongo, Reddis, memCache, etc. running and then manages them [622].

11.3.15 CloudBees

Cloudbees provides Platform as a Service (PaaS) solution, which is a cloud service for Java applications [124]. It is used to build, run and manage the web applications. It was created in 2010 by Jenkins. It has a continuous delivery platform for DevOps, and adds a enterprise-grade functionality with an expert level support. Cloudbees is better than the traditional Java platform as it requires no provision of the nodes, clusters, load balancers and databases. In cloudbees the environment is constantly managed and monitored where a metering and scale updating is done on a real time basis. The platform ships with verified security and enhancements assuring less risk for sharing sensitive information. It simplifies the task of getting the platform accessed by every user using the feature *Jenkins Sprawl* [125].

11.3.16 Engine Yard (573)

A deployment platform with fully managed services that combines high-end clustering resources to run Ruby and Rails applications in the cloud is offered by Engine Yard. It is designed as a platform-as-a-Service for Web application developers using Ruby on Rails, PHP and Node.js who requires the advantages of cloud computing. Amazon cloud is the platform where the Engine Yard perform its operations and accomplishes application stack for its users. Amazon allows as many as eight regions to Engine Yard to deploy its CPU instances in varying capacities such as normal, high memory and high CPU. According to customer requirements multiple software components are configured and processed when an instance is started in Engine Yard.

Engine Yard builds its version on Gentoo Linux and has non-proprietary approach to its stack. The stack includes HAProxy load balancer, Ngnix and Rack Web servers, Passenger and Unicorn app servers, as well as MySQL and PostgreSQL relational databases in addition to Ruby, PHP, and Node.js. The credibility of Engine Yard rests with orchestration and management as developers have option of performing functions in Amazon cloud. Standard operations management procedures are performed once the systems are configured and deployed. Key operations tasks such as performing backups, managing snapshots, managing clusters, administering databases and load balancing are taken care by Engine Yard.

Engine Yard users are empowered as they have more control over virtual machine instances. These instances are dedicated instances and are not shared with other users. As the instances are independent every user can exercise greater control over instances without interferences with other users.

11.3.17 (CloudControl)

No Longer active as of Feb. 2016 [670]

11.3.18 dotCloud (428)

dotCloud services were shutdown on February 29,2016.

11.3.19 Dokku**11.3.20 OSGi****11.3.21 HUBzero**

HUBzero is a collaborative framework which allows creation of dynamic websites for scientific research as well as educational activities. HUBzero lets scientific researchers work together online to develop simulation and modeling tools. These tools can help you connect with powerful Grid computing resources as well as rendering farms [651]. Thus allowing other researchers to access the resulting tools online using a normal web browser and launch simulation runs on the Grid infrastructure without having to download or compile any code. It is a unique framework with simulation and social networking capabilities [390].

11.3.22 OODT

The Apache Object Oriented Data Technology (OODT) is an open source data management system framework. OODT was originally developed at NASA Jet Propulsion Laboratory to support capturing, processing and sharing of data for NASA's scientific archives. OODT focuses on two canonical use cases: Big Data processing and on Information integration. It facilitates the integration of highly distributed and heterogeneous data intensive systems enabling the integration of different, distributed software systems, metadata and data. OODT is written in the Java, and through its REST API used in other languages including Python [440].

11.3.23 Agave

Agave is an open source, application hosting framework and provides a platform-as-a-service solution for hybrid computing [642]. It provides everything ranging from authentication and authorization to computational, data and collaborative services. Agave manages end to end lifecycle of an application's execution. Agave provides an execution platform, data management platform, or an application platform through which users can execute applications, perform operations on their data or simple build their web and mobile applications [15].

Agave's API's provide a catalog with existing technologies and hence no additional appliances, servers or other software needs to be installed. To deploy an application from the catalog, the user needs to host it on a storage system registered with Agave, and submit to `agave`, a JSON file that shall contain the path to the executable file, the input parameters, and specify the desired output location. Agave shall read the JSON file, formalize the parameters, execute the user program and dump the output to the requested destination [642].

11.3.24 Atmosphere

Atmosphere is developed by CyVerse (previously named as iPlant Collaborative). It is a cloud-computing platform. It allows one to launch his own “isolated virtual machine (VM) image [74]. It does not require any machine specification. It can be run on any device (tablet/desktop/laptop) and any machine(Linux/Windows/Max/Unix). User should have a CyVerse account and be granted permission to access to Atmosphere before he can begin using Atmosphere. No subscription is needed. Atmosphere is designed to execute data-intense bioinformatics tasks that may include a)Infrastructure as a Service (IaaS) with advanced APIs; b)Platform as a Service (PaaS), and c)Software as a Service (SaaS). On Atmosphere one has several images of virtual machine and user can launch any image or instance according to his requirements. The images launched by users can

be shared among different members as and when required [377].

11.4 High level Programming

11.4.1 Kite

Kite is a programming language designed to minimize the required experience level of the programmer. It aims to allow quick development and running time and low CPU and memory usage. Kite was designed with lightweight systems in mind. On OS X Leopard, the main Kite library is only 88KB, with each package in the standard library weighing in at 13-30KB. The main design philosophy is minimalism — only include the minimum necessary, while giving developers the power to write anything that they can write in other languages. Kite combines both object oriented and functional paradigms in the language syntax. One special feature is its use of the pipe character (!) to indicate function calls, as opposed to the period (.) or arrow (->) in other languages. Properties are still de-referenced using the period [155]. Kite also offers a digital assistant for programmers. Kite offers a product which sits as a sidebar in code editor and enables programmers to search for opensource codes to implement in their codes. It even provides relevant examples/syntax and also tries to spot errors in the programs [639].

11.4.2 Hive

The reason behind development of Hive is making it easier for end users to use Hadoop. Map reduce programs were required to be developed by users for simple to complex tasks. It lacked expressiveness like query language. So, it was a time consuming and difficult task for end users to use Hadoop. For solving this problem Hive was built in January 2007 and open sourced in August 2008. Hive is an open source data warehousing solution which is built on top of Hadoop. It structures data into understandable and conventional database terms like tables, columns, rows and partitions. It supports HiveQL queries which have structure like SQL queries. HiveQL queries are compiled to map reduce jobs which are then executed by Hadoop. Hive also contains Metastore which includes schemas and statistics which is useful in query compilation, optimization and data exploration [70]

11.4.3 HCatalog

11.4.4 Tajo

Apache Tajo [57] is a big data relational and distributed data warehouse system for Apache's Hadoop framework. It uses the Hadoop Distributed File System (HDFS) as a storage layer and has its own query execution engine instead of the MapReduce framework. Tajo is designed to provide low-latency and scalable ad-hoc queries, online aggregation, and ETL (extraction-transformation-loading process) on large-data sets which are stored on HDFS (Hadoop Distributed File System) and on other data sources [58]. Apart from HDFS, it also supports other storage formats as Amazon S3, Apache HBase, Elasticsearch etc. It provides distributed SQL query processing engine and even has query optimization techniques and provides interactive analysis on large-data sets. Tajo is compatible with ANSI/ISO SQL standard, JDBC standard. Tajo can also store data from various file formats such as CSV, JSON, RCF, SequenceFile, ORC and Parquet. It provides a SQL shell which allows users to submit the SQL queries. It also offers user defined functions to work with it which can be created in python. A Tajo cluster has one master node and a number of worker nodes [58]. The master node is responsible for performing the query planning and maintaining

a coordination among the worker nodes. It does this by dividing a query in small task which are assigned to the workers who have a local query engine for executing the queries assigned to them.

11.4.5 **Shark**

Data Scientists when working on huge data sets try to extract meaning and interpret the data to enhance insight about the various patterns, opportunities, and possibilities that the dataset has to offer [178]. At a traditional EDW (Enterprise Data Warehouse), a simple data manipulation can be performed using SQL queries but we have to rely on other systems to apply the machine learning algorithms on these data sets. Apache Shark is a distributed query engine developed by the open source community whose goal is to provide a unified system for easy data manipulation using SQL and pushing sophisticated analysis towards the data.

Shark is a data Warehouse system built on top of Apache Spark which does the parallel data execution and is also capable of deep data analysis using the Resilient Distributed Datasets(RDD) memory abstraction which unifies the SQL query processing engine with analytical algorithms [178]. Based on this common abstraction, it allows running two query in the same set of workers and share intermediate data. Since RDDs are designed to scale horizontally, it is easy to add or remove nodes to accommodate more data or faster query processing. Thus, it can be scaled to the large number of nodes in a fault-tolerant manner

”Shark is built on Hive Codebase and it has the ability to execute HIVE QL queries up to 100 times faster than Hive without making any change in the existing queries [178]. “Shark can run both on the Standalone Mode and Cluster-Mode. Shark can answer the queries 40X faster than Apache Hive and can run machine learning algorithms 25X faster than MapReduce programs in Apache Hadoop on large data sets” [178]. Thus, this new data analysis system performs query processing and complex analytics (iterative Machine learning) at scale and efficiently recovers from the failures.

11.4.6 **Phoenix**

In the first quarter of 2013, Salesforce.com released its proprietary SQL-like interface and query engine for HBase, *Phoenix*, to the open source community. The company appears to have been motivated to develop Phoenix as a way to 1) increase accessibility to HBase by using the industry-standard query language (SQL); 2) save users time by abstracting away the complexities of coding native HBase queries; and, 3) implementing query best practices by implementing them automatically via Phoenix [347]. Although Salesforce.com initially *open-sourced* it via Github, by May of 2014 it had become a top-level Apache project [679].

Phoenix, written in Java, ”compiles [SQL queries] into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets [55]. “ In addition, the program directs compute intense portions of the calls to the server. For instance, if a user queried for the top ten records across numerous regions from an HBase database consisting of a billion records, the program would first select the top ten records for each region using server-side compute resources. After that, the client would be tasked with selecting the overall top ten” [534].

Despite adding an abstraction layer, Phoenix can actually speed up queries because it optimizes the query during the translation process [347]. For example, “Phoenix beats Hive for a simple query spanning 10M-100M rows” [76].

Finally, another program can enhance HBase’s accessibility for those inclined towards graphical interfaces. SQuirell only requires the user to set up the JDBC driver and specify the appropriate

connection string. [576].

11.4.7 Impala

Cloudera Impala is Cloudera's open source massively parallel processing (MPP) SQL query engine for data stored in a computer cluster running Apache Hadoop [128]. It allows users to execute low latency SQL queries for data stored in HDFS and HBase, without any movement or transformation of data. The Apache Hive provides a powerful query mechanism for hadoop users, but the query response time are not acceptable due to Hive's reliance on MapReduce. Impala technology by Cloudera has its MPP query engine written in C++ replacing the Java engine proves to improve the interactive Hadoop queries and interactive query response time for hadoop users [196]. Impala is faster than Hive also because it executes the SQL queries natively without translating them into Hadoop MapReduce jobs, thus taking less time. Impala uses HiveQL as programming interface and also the Impala's Query Exec Engines are co-located with the HDFS data nodes, so that the data nodes and processing tasks are co-located, following the haddops paradigm [196]. Impala can also use Hbase as a data source. Thus, Impala can be considered as an extension to the Apache Hadoop, providing a better performance alternative to Hive-on-top-of-MapReduce model.

Hive and other frameworks built on MapReduce are best suited for long running batch jobs, such as those involving batch processing of Extract, Transform, and Load (ETL) type jobs [128]. The important applications of Impala are when the data is to be partially analyzed or when the same kind of query is to be processed several times from the dataset. When the data is to be partially analyzed, Impala uses parquet as the file format, which is developed by Twitter and Cloudera and it stores data in vertical manner [8]. When Parquet queries the dataset it only reads the column split part files rather than reading the entire dataset as compared to Hive.

11.4.8 MRQL

MapReduce Query Language (MRQL, pronounced miracle) "is a query processing and optimization system for large-scale, distributed data analysis" [52]. MRQL provides a SQL like language for use on Apache Hadoop, Hama, Spark, and Flink. MRQL allows users to perform complex data analysis using only SQL like queries, which are translated by MRQL to efficient Java code. MRQL can evaluate queries in Map-Reduce (using Hadoop), Bulk Synchronous Parallel (using Hama), Spark, and Flink modes [52].

MRQL was created in 2011 by Leonids Fegaras [727] and is currently in the Apache Incubator. All projects accepted by the Apache Software Foundation (ASF) undergo an incubation period until a review indicates that the project meets the standards of other ASF projects [51].

11.4.9 SAP HANA

As noted in [719], SAP HANA is in-memory massively distributed platform that consists of three components: analytics, relational ACID compliant database and application. Predictive analytics and machine learning capabilities are dynamically allocated for searching and processing of spatial, graphical, and text data. SAP HANA accommodates flexible development and deployment of data on premises, cloud and hybrid configurations. In a nutshell, SAP HANA acts as a warehouse that integrates live transactional data from various data sources on a single platform [436]. It provides extensive administrative, security features and data access that ensures high data availability, data protection and data quality.

11.4.10 HadoopDB

HadoopDB is a hybrid of parallel database and MapReduce technologies. It approaches parallel databases in performance and efficiency, yet still yields the scalability, fault tolerance, and flexibility of MapReduce systems. It is a free and open source parallel DBMS. The basic idea behind it is to give Hadoop access to multiple single-node DBMS servers (eg. PostgreSQL or MySQL) deployed across the cluster. It pushes as much as possible data processing into the database engine by issuing SQL queries which results in resembling a shared-nothing cluster of machines [266].

HadoopDB is more scalable than currently available parallel database systems and DBMS/MapReduce hybrid systems. It has been demonstrated on clusters with 100 nodes and should scale as long as Hadoop scales, while achieving superior performance on structured data analysis workloads.

11.4.11 PolyBase

“PolyBase is a technology that accesses and combines both non-relational and relational data, all from within SQL Server. It allows you to run queries on external data in Hadoop or Azure Blob storage acts mediator between SQL and non SQL data store it makes the analysis of the relation data and other data that is non structure to tables (Hadoop).” [480] Unless there is a way to transfer data between the data stores it is always difficult to do so. PolyBase bridges this gap by operating on data that is external to SQL server. It don’t require additional software, querying to external can be done with same syntax as querying a database table. This happens transparently behind the scene, no knowledge of Hadoop or Azure is required.

It can query data store in Hadoop using T-SQL, polybase also makes it easy to access the Azure blob data using T-SQL. There is no need for a separate ETL or import tool while importing data from Hadoop, “Azure blob storage or Azure Data Lake into relational tables. It leverages Microsoft’s Columnstore technology and analysis capabilities while importing” [480]. It also archives data into Hadoop Azure blob and data lake store in cost effective way.

Push computation to Hadoop. The query optimizer makes a cost-based decision to push computation to Hadoop and while doing so will improve query performance. It uses statistics on external tables to make the cost-based decision. Pushing computation creates MapReduce jobs and leverages Hadoop’s distributed computational resources. Scale compute resources. SQL Server PolyBase scale-out groups can be used to improve query performance. This enables parallel data transfer between SQL Server instances and Hadoop nodes, and it adds compute resources for operating on the external data.

11.4.12 Pivotal HD/Hawq

Pivotal HDB is the Apache Hadoop native SQL database powered by Apache HAWQ [474] for data science and machine learning workloads. It can be used to gain deeper and actionable insights into data with out the need from moving data to another platform to perfrom advanced analytics. Few important problems that Pivot HDB address are as follows Quickly unlock business insights with exceptional performance, Integrate SQL BI tools with confidence and Iterate advanced analytics and machine learning in database support. Pivotal HDB comes with an elastic SQL query engine which combines MPP-based analytical performance, robust ANSI SQL compliance and integrated Apache MADlib for machine learning [475].

11.4.13 Presto

```
.. include:: techs/presto.rst
```

11.4.14 Google Dremel

Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, Google Dremel is capable of running aggregation queries over trillion-row tables in seconds [392]. With Dremel, you can write a declarative SQL-like query against data stored in a read-only columnar format efficiently for analysis or data exploration. It's also possible to write queries that analyze billions of rows, terabytes of data, and trillions of records in seconds. Dremel can be used for a variety of jobs including analyzing web-crawled documents, detecting e-mail spam, working through application crash reports.

11.4.15 Google BigQuery

Google BigQuery [92] is an enterprise data warehouse used for large scale data analytics [93]. A user can store and query massive datasets by storing the data in BigQuery and querying the database using fast SQL queries using the processing power of Google's infrastructure. In Google BigQuery a user can control access to both the project and the data based on his business needs which gives the ability to others to view and even query the data [92]. BigQuery can scale the database from GigaBytes to PetaBytes. BigQuery can be accessed using a Web UI or a command-line tool or even by making calls to the BigQuery REST API using a variety of client libraries such as Java, .NET or Python. BigQuery can also be accessed using a variety of third party tools. BigQuery is fully managed to get started on its own, so there is no need to deploy any resources such as disks and virtual machines.

Projects in BigQuery [93] are top-level containers in Google Cloud Platform. They contain the BigQuery Data. Each project is referenced by a name and unique ID. Tables contain the data in BigQuery. Each table has a schema that describes field names, types, and other information. Datasets enable to organise and control access to the tables. Every table must belong to a dataset. A BigQuery data can be shared with others by defining roles and setting permissions for organizations, projects, and datasets, but not on the tables within them. BigQuery stores data in the [463] Capacitor columnar data format, and offers the standard database concepts of tables, partitions, columns, and rows.

11.4.16 Amazon Redshift

Amazon Redshift is a fully managed, petabyte-scale data warehouse service in the cloud. Redshift service manages all of the work of setting up, operating and scaling a data warehouse. AWS Redshift can perform these tasks including provisioning capacity, monitoring and backing up the cluster, and applying patches as well as upgrades to the Redshift's engine [26]. Redshift is built on the top of technology from the Massive Parallel Processing (MPP) data-warehouse company ParAccel which is based on PostgreSQL 8.0.2 to PostgreSQL 9.x with capability to handle analytics workloads on large-scale datasets stored by a column-oriented DBMS principle [674].

11.4.17 Drill

Apache Drill [163] is an open source framework that provides schema free SQL query engine for distributed large-scale datasets. Drill has an extensible architecture at its different layers. It does not require any centralized metadata and does not have any requirement for schema specification. Drill is highly useful for short and interactive ad-hoc queries on very large scale data sets. It is scalable to several thousands of nodes. Drill is also capable to query nested data in various formats like JSON and Parquet. It can query large amount of data at very high speed. It is also capable of performing discovery of dynamic schema. A service called ‘Drillbit’ is at the core of Apache Drill responsible for accepting requests from the client, processing the required queries, and returning all the results to the client. Drill is primarily focused on non-relational datastores, including Hadoop and NoSQL

11.4.18 Kyoto Cabinet

Kyoto Cabinet as specified in [362] is a library of routines for managing a database which is a simple data file containing records. Each record in the database is a pair of a key and a value. Every key and value is serial bytes with variable length. Both binary data and character string can be used as a key and a value. Each key must be unique within a database. There is neither concept of data tables nor data types. Records are organized in hash table or B+ tree. Kyoto Cabinet runs very fast. The elapsed time to store one million records is 0.9 seconds for hash database, and 1.1 seconds for B+ tree database. Moreover, the size of database is very small. The overhead for a record is 16 bytes for hash database, and 4 bytes for B+ tree database. Furthermore, scalability of Kyoto Cabinet is great. The database size can be up to 8EB (9.22e18 bytes).

11.4.19 Pig**11.4.20 Sawzall**

Google engineers created the domain-specific programming language (DSL) *Sawzall* as a productivity enhancement tool for Google employees. They targeted the analysis of large data sets with flat, but regular, structures spread across numerous servers. The authors designed it to handle “simple, easily distributed computations: filtering, aggregation, extraction of statistics,” etc. from the aforementioned data sets. [469]

In general terms, a Sawzall job works as follows: multiple computers each create a Sawzall instance, perform some operation on a single record out of (potentially) petabytes of data, return the result to an aggregator function on a different computer and then shut down the Sawzall instance.

The engineer’s focus on simplicity and parallelization led to unconventional design choices. For instance, in contrast to most programming languages Sawzall operates on one data record at a time; it does not even preserve state between records. [519] Additionally, the language provides just a single primitive result function, the *emit* statement. The emitter returns a value from the Sawzall program to a designated virtual receptacle, generally some type of aggregator. In another example of pursuing language simplicity and parallelization, the aggregators remain separate from the formal Sawzall language (they are written in C++) because “some of the aggregation algorithms are sophisticated and best implemented in a native language [and] [m]ore important[ly] drawing an explicit line between filtering and aggregation enables a high degree of parallelism, even though it hides the parallelism from the language itself” [469].

Important components of the Sawzall language include: *szi*, the binary containing the code compiler and byte-code interpreter that executes the program; the *libszi* library, which compiles

and executes Sawzall programs “[w]hen szl is used as part of another program, e.g. in a [map-reduce] program”; the Sawzall language plugin, designated *protoc_gen_szl*, which generates Sawzall code when run in conjunction with Google’s own *protoc* protocol compiler; and libraries for intrinsic functions as well as Sawzall’s associated aggregation functionality. [21]

11.4.21 Google Cloud DataFlow

Google Cloud DataFlow [237] is a unified programming model that manages the deployment, maintenance and optimization of data processes such as batch processing, ETL etc. It creates a pipeline of tasks and dynamically allocates resources thereby maintaining high efficiency and low latency. According to [237], these capabilities make it suitable for solving challenging big data problems. Also, google DataFlow overcomes the performance issues faced by Hadoops Mapreduce while building pipelines. As stated in [391] the performance of MapReduce started deteriorating while facing multiple petabytes of data whereas Google Cloud Dataflow is apparently better at handling enormous datasets. [237]. Additionally Google Dataflow can be integrated with Cloud Storage, Cloud Pub/Sub, Cloud Datastore, Cloud Bigtable, and BigQuery. The unified programming ability is another noteworthy feature which uses Apache Beam SDKs to support powerful operations like windowing and allows correctness control to be applied to batch and stream data processes.

11.4.22 Summingbird

According to [100], “Summingbird is a library that lets you write MapReduce programs that look like native Scala or Java collection transformations and execute them on a number of well-known distributed MapReduce platforms, including Storm and Scalding.” Summingbird is open-source and is a domain-specific Scala implemented language [101]. It combines online and batch MapReduce computations into one framework [101]. It utilizes the platforms Hadoop for batch and Storm for online process execution [101]. The open-source Hadoop implementation of MapReduce is a tool which those responsible for data management use to handle problems related to big data [101]. Summingbird uses an algebraic structure called a commutative semigroup to perform aggregations of both batch and online processes [101]. A commutative semigroup is a particular type of semigroup “where the associated binary operation is also commutative” [101]. The types of data that Summingbird takes as inputs are streams and snapshots [101]. The types of data Summingbird jobs generate are called stores and sinks [101]. Stores are “an abstract model of a key-value store” while sinks are unaggregated tuples from a producer [101]. Summingbird aims to simplify the process of both batch and online analytics by exploiting “the formal properties of algebraic structures” to integrate the various modes of distributed processing [101].

11.4.23 Lumberyard

It is powerful and full-featured enough to develop triple-A, current-gen console games and is deeply integrated with AWS and Twitch(an online steaming service) [431]. Lumberyard’s core engine technology is based on Crytek’s CryEngine [215]. The goal is ”creating experiences that embrace the notion of a player, broadcaster, and viewer all joining together“[431]. Monetization for Lumberyard will come strictly through the use of Amazon Web Services’ cloud computing. If you use the engine for your game, you’re permitted to roll your own server tech, but if you’re using a third-party provider, it has to be Amazon [635].

11.5 Streams

11.5.1 Storm

Apache Storm is an open source distributed computing framework for analyzing big data in real time [313]. refers storm as the Hadoop of real time data. Storm operates by reading real time input data from one end and passes it through a sequence of processing units delivering output at the other end. The basic element of Storm is called topology. A topology consists of many other elements interconnected in a sequential fashion. Storm allows us to define and submit topologies written in any programming language.

Once under execution, a storm topology runs indefinitely unless killed explicitly. The key elements in a topology are the spout and the bolt. A spout is a source of input which can read data from various datasources and passes it to a bolt. A bolt is the actual processing unit that processes data and produces a new output stream. An output stream from a bolt can be given as an input to another bolt [572].

11.5.2 S4

S4 [48] is a distributed, scalable, fault-tolerant, pluggable platform that allows programmers to easily develop applications for processing continuous unbounded streams of data. It is built on similar concept of key-value pairs like the MapReduce. The core platform is written in Java [521]. S4 provides a runtime distributed platform that handles communication, scheduling and distribution across containers. The containers are called S4 nodes. The data is executed and processed on these S4 nodes. These S4 nodes are then deployed on S4 clusters. The user develops applications and deploys them on S4 clusters for its processing. The applications are built as a graph of Processing Elements (PEs) and Stream that interconnects the PEs. All PEs communicate asynchronously by sending events on streams. Events are dispatched to nodes according to their key in the program [48]. All nodes are symmetric with no centralized service and no single point of failure. Additionally there is no limit on the number of nodes that can be supported. [388]. In S4, both the platform and the applications are built by dependency injection, and configured through independent modules.

11.5.3 Samza

Apache Samza is an open-source near-realtime, asynchronous computational framework for stream processing developed by the Apache Software Foundation in Scala and Java [49]. Apache Samza is a distributed stream processing framework. It uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management. Samza processes streams. A stream is composed of immutable messages of a similar type or category. Messages can be appended to a stream or read from a stream. Samza supports pluggable systems that implement the stream abstraction: in Kafka a stream is a topic, in a database we might read a stream by consuming updates from a table, in Hadoop we might tail a directory of files in HDFS. Samza is a stream processing framework. Samza provides a very simple callback-based *process message* API comparable to MapReduce. Samza manages snapshotting and restoration of a stream processor's state. Samza is built to handle large amounts of state (many gigabytes per partition) [524]. Whenever a machine in the cluster fails, Samza works with YARN to transparently migrate your tasks to another machine. Samza uses Kafka to guarantee that messages are processed in the order they were written to a partition, and that no messages are ever lost. Samza is partitioned and distributed at every level. Kafka provides ordered, partitioned, replayable, fault-tolerant streams. YARN provides a distributed environment for Samza containers to run in. Samza works

with Apache YARN, which supports Hadoop's security model, and resource isolation through Linux CGroups [50] [49].

11.5.4 Granules

Granules are used for execution or processing of data streams in distributed environment. When applications are running concurrently on multiple computational resources, granules manage their parallel execution. The MapReduce implementation in Granules is responsible for providing better performance. It has the capability of expressing computations like graphs. Computations can be scheduled based on periodicity or other activity. Computations can be developed in C, C++, Java, Python, C#, R. It also provides support for extending basic Map reduce framework. Its application domains include hand writing recognition, bio informatics and computer brain interface [542].

11.5.5 Neptune

11.5.6 Google MillWheel

MillWheel is a framework for building low-latency data-processing applications. Users specify a directed computation graph and application code for individual nodes, and the system manages persistent state and the continuous flow of records, all within the envelope of the framework's fault-tolerance guarantees. Other streaming systems do not provide this combination of fault tolerance, versatility, and scalability. MillWheel allows for complex streaming systems to be created without distributed systems expertise. MillWheel's programming model provides a notion of logical time, making it simple to write time-based aggregations. MillWheel was designed from the outset with fault tolerance and scalability in mind. In practice, we find that MillWheel's unique combination of scalability, fault tolerance, and a versatile programming model [17].

11.5.7 Amazon Kinesis

Kinesis is Amazon's [353] real time data processing engine. It is designed to provide scalable, durable and reliable data processing platform with low latency. The data to Kinesis can be ingested from multiple sources in different format. This data is further made available by Kinesis to multiple applications or consumers interested in the data. Kinesis provides robust and fault tolerant system to handle this high volume of data. Data sharding mechanism in Kinesis makes it horizontally scalable. Each of these shards in Kinesis process a group of records which are partitioned by the shard key. Each record processed by Kinesis is identified by sequence number, partition key and data blob. Sequence number to records is assigned by the stream. Partition keys are used by partitioner(a hash function) to map the records to the shards i.e. which records should go to which shard. Producers like web servers, client applications, logs push the data to Kinesis whereas Kinesis applications act as consumers of the data from Kinesis engine. It also provides data retention for certain time for example 24 hours default. This data retention window is a sliding window. Kinesis collects lot of metrics which can be used to understand the amount of data being processed by Kinesis. User can use these metrics to do some analytics and visualize the metrics data. Kinesis is one of the tools part of AWS infrastructure and provides its users a complete software-as-a-service. Kinesis [540] in the area of real-time processing provides following key benefits: ease of use, parallel processing, scalable, cost effective, fault tolerant and highly available.

11.5.8 LinkedIn

LinkedIn is a social networking website for Business and employment [699]. LinkedIn has more than 400 million user profiles (as per 10 March 2016 news), and increasing at a rate of 2 new member every second [156]. LinkedIn provides different products like:

- People You May Know - Skill Endorsements - Jobs You May Be Interested In - News Feed Updates

Such products are based on big data. To achieve such big data tasks, LinkedIn has its ecosystem consist of Oracle, Hadoop, Pig, Hive, Azkaban (Workflow), Avro Data, Zookeeper, Aster Data, Data In- Apache Kafka, Data Out- Apache Kafka and Voldemort [156]. LinkedIn uses Hadoop and Aster Data as an analytics layer [496]. LinkedIn partitioned the user's data into separate DB's stored it in XML format. Voldemort is a key lookup system used to store the analytically-derived data for the products like "People You May Know". Voldemort stores the data in key-value form [496]. LinkedIn has exposed REST API to get the user data [374].

11.5.9 Twitter Heron

Heron is a real-time analytics platform that was developed at Twitter for distributed streaming processing. Heron was introduced at SIGMOD 2015 to overcome the shortcomings of Twitter Storm as the scale and diversity of Twitter data increased. As mentioned in [625] The primary advantages of Heron were: API compatible with Storm: Back compatibility with Twitter Storm reduced migration time. Task-Isolation: Every task runs in process-level isolation, making it easy to debug/ profile. Use of main stream languages: C++, Java, Python for efficiency, maintainability, and easier community adoption. Support for backpressure: dynamically adjusts the rate of data flow in a topology during run-time, to ensure data accuracy. Batching of tuples: Amortizing the cost of transferring tuples. Efficiency: Reduce resource consumption by 2-5x and Heron latency is 5-15x lower than Storm's latency. The architecture of Heron (as shown in [624]) uses the Storm API to submit topologies to a scheduler. The scheduler runs each topology as a job consisting of several containers. The containers run the topology master, stream manager, metrics manager and Heron instances. These containers are managed by the scheduler depending on resource availability.

11.5.10 Databus

11.5.11 Facebook Puma/Ptail/Scribe/ODS

The real time data Processing at Facebook is carried out using the technologies like Scribe, Ptail, Puma, and ODS. While designing the system, facebook primarily focused on the five key decisions that the system should incorporate which were Ease of Use, Performance, Fault-tolerance, Scalability, and Correctness. "The real time data analytics ecosystem at facebook is designed to handle hundreds of Gigabytes of data per second via hundreds of data pipelines and this system handles over 200,000 events per second with a maximum latency of 30 seconds" [185]. Facebook focused on the Seconds of latency while designing the system and not milliseconds as seconds are fast enough to for all the use case that needs to be supported, and it allowed facebook to use persistent message bus for data transport and this also made the system more fault tolerant and scalable [185]. The large infrastructure of facebook comprises of hundreds of systems distributed across multiple data centers that needs a continuous monitoring to track their health and performance which is done by Operational Data Store(ODS) [116]. ODS comprises of a time series database (TSDB), which is a query service, and a detection and alerting system. ODS's TSDB is built atop the HBase storage system. Time series data from services running on Facebook hosts is collected by the ODS

write service and written to HBase.

When the data is generated by the user from their devices, an AJAX request is fired to facebook, and these requests are then written to a log file using Scribe (distributed data transport system), this messaging system collects, aggregates, and delivers high volume of log data with few seconds of latency and high throughput. Scribe stores the data in the HDFS (Hadoop Distributed File System) in a tailing fashion, where the new events are stored in log files and the files are tailed below the current events. The events are then written into the storage HBase on distributed machines. This makes the data available for both batch and real-time processing. Ptail is an internal tool built to aggregate data from multiple Scribe stores. It then tails the log files and pulls data out for processing. Puma is a stream processing system which is the real-time aggregation/storage of data. Puma provides filtering and processing of Scribe streams (with a few seconds delay), usually Puma batches the storage per 1.5 seconds on average and when the last flush completes, then only a new batch starts to avoid the contention issues, which makes it fairly real time.

11.5.12 Azure Stream Analytics

Azure Stream Analytics is a platform that manages data streaming from devices, web sites, infrastructure systems, social media, internet of things analytics, and other sources using real-time event processing engine [82]. Jobs are authored by “specifying the input source of the streaming data, the output sink for the results of your job, and a data transformation expressed in a SQL-like language.” Some key capabilities and benefits include ease of use, scalability, reliability, repeatability, quick recovery, low cost, reference data use, user defined functions capability, and connectivity. [160] Available documentation to get started with Azure Stream Analytics. [80] Azure Stream Analytics has a development project available on github.

11.5.13 Floe

11.5.14 Spark Streaming (563)

Spark Streaming is a library built on top of Spark Core which enables Spark to process real-time streaming data. The streaming jobs can be written similar to batch jobs in Spark, using either Java, Scala or Python. The input to Spark Streaming applications can be fed from multiple data sources such HDFS, Kafka, Flume, Twitter, ZeroMQ, or custom-defined sources. It also provides a basic abstraction called Discretized Streams or DStreams to represent the continuous data streams. Spark’s API for manipulating these data streams is very similar to the Spark Core’s Resilient Distributed Dataset(RDD) API [559] which makes it easier for users to move between projects with stored and real-time data as the learning curve is short. Spark Streaming is designed to provide fault-tolerance, throughput, and scalability. Examples of streaming data are messages being published to a queue for real-time flight status update or the log files for a production server.

11.5.15 Flink Streaming

11.5.16 DataTurbine

Data Turbine [707] is open source engine that allows to stream data from various sources, process it and sink it to different destinations. The streaming sources can be labs, web cams and Java enabled cell phones. The sinks can be visualizations, interfaces and databases. Data Turbine can be used to stream data formats like numbers, text, sound and video.

[206] explains that the Data Turbine middleware provides the cyber-infrastructure that integrates

disparate elements of complex distributed real time application. Data Turbine acts as a middleware black box using which applications and devices can send and receive data. Data Turbine manages the management operations like memory and file management as well as book-keeping and reconnection logic. Data Turbine also provides Android based controller which allows algorithms to run close to sensors.

11.6 Basic Programming Model and Runtime, SPMD, MapReduce

11.6.1 Hadoop

Apache Hadoop is an open source framework written in Java that utilizes distributed storage and the MapReduce programming model for processing of big data. Hadoop utilizes commodity hardware to build fault tolerant clusters. Hadoop was developed based on papers published by Google on the Google File System (2003) and MapReduce (2004) [678].

Hadoop consists of several modules: the Cluster, Storage, Hadoop Distributed File System (HDFS) Federation, Yarn Infrastructure, MapReduce Framework, and the Hadoop Common Package. The Cluster is comprised of multiple machines, otherwise referred to as nodes. Storage is typically in the HDFS. HDFS federation is the framework responsible for this storage layer. YARN Infrastructure provides computational resources such as CPU and memory. The MapReduce layer is responsible for implementing MapReduce [135]. The Hadoop Common Package which includes operating and file system abstractions and JAR files needed to start Hadoop [678].

11.6.2 Spark (541)

Apache Spark which is an open source cluster computing framework has emerged as the next generation big data processing engine surpassing Hadoop MapReduce. “Spark engine is developed for in-memory processing as well a disk based processing. This system also provides large number of impressive high level tools such as machine learning tool M Lib, structured data processing, Spark SQL, graph processing took Graph X, stream processing engine called Spark Streaming, and Shark for fast interactive question device.” The ability of spark to join datasets across various heterogeneous data sources is one of its prized attributes. Apache Spark is not the most suitable data analysis engine when it comes to processing (1) data streams where latency is the most crucial aspect and (2) when the available memory for processing is restricted. “When available memory is very limited, Apache Hadoop Map Reduce may help better, considering huge performance gap.” In cases where latency is the most crucial aspect we can get better results using Apache Storm.

11.6.3 Twister

Twister is a new software tool released by Indiana University, which is an extension to MapReduce architectures currently used in the academia and industry [319]. It supports faster execution of many data mining applications implemented as MapReduce programs. Applications that currently use Twister include: K-means clustering, Google’s page rank, Breadth first graph search , Matrix multiplication, and Multidimensional scaling. Twister also builds on the SALSA team’s work related to commercial MapReduce runtimes, including Microsoft Dryad software and open source Hadoop software. SALSA project work is funded in part by an award from Microsoft, Inc. The architecture is based on pub/sub messaging that enables it to perform faster data transfers, minimizing the overhead of the runtime. Also, the support for long running processes improves the efficiency of the runtime for many iterative MapReduce computations. [623] [318] [729].

11.6.4 MR-MPI

[408] MR-MPI stands for Map Reduce-Message Passing Interface is open source library build on top of standard MPI. It basically implements mapReduce operation providing a interface for user to simplify writing mapReduce program. It is written in C++ and needs to be linked to MPI library in order to make the basic map reduce functionality to be executed in parallel on distributed memory architecture. It provides interface for c, c++ and python. Using C interface the library can also be called from Fortrain.

11.6.5 Stratosphere (Apache Flink)

Apache Flink is an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications. Apache Flink is used in big data application primarily involving analysis of data stored in Hadoop clusters. It also supports a combination of in-memory and disk-based processing as well as handles both batch and stream processing jobs, with data streaming the default implementation and batch jobs running as special-case versions of streaming application [677].

11.6.6 Reef

REEF (Retainable Evaluator Execution Framework) [107] is a scale-out computing fabric that eases the development of Big Data applications on top of resource managers such as Apache YARN and Mesos. It is a Big Data system that makes it easy to implement scalable, fault-tolerant runtime environments for a range of data processing models on top of resource managers. REEF provides capabilities to run multiple heterogeneous frameworks and workflows of those efficiently. REEF contains two libraries, Wake and Tang where Wake is an event-based-programming framework inspired by Rx and SEDA and Tang is a dependency injection framework inspired by Google Guice, but designed specifically for configuring distributed systems.

11.6.7 Disco

a. Disco from discoproject.org represents an implementation of mapreduce for distributed computing that benefits end users by relieving them of the need to handle “difficult technicalities related to distribution such as communication protocols, load balancing, locking, job scheduling, and fault tolerance.” [595] Its designers wrote the software in Erlang, an inherently fault tolerant language. In addition, Disco’s creators chose Erlang because they believe it best meets the software’s need to handle “tens of thousands of tasks in parallel.” [596] Python was used for Disco’s libraries. Finally, Disco supports pipelines, “a linear sequence of stages, where the outputs of each stage are grouped into the input of the subsequent stage.” [120] Its designers implemented Disco’s libraries in Python. Disco originated within Nokia Corp. to handle large data sets. Since then it has proven itself reliable in production environments outside of Nokia. [427]

b. DISCO from the research group Service Engineering (SE), [[www-discoabout-discoabstractionlayer](#)] serves as “an abstraction layer for OpenStack’s orchestration component [Heat]” SE based DISCO on its prior orchestration framework, Hurtle. The software sets up a computer cluster and deploys the user’s choice of distributed computing architecture onto the cluster based on setup inputs provided by the user. DISCO offers a command line interface via HTTP to directly access OpenStack. [[www-discodescribed-discoabstractionlayer](#)]

11.6.8 Hama

Apache Hama is a framework for Big Data analytics which uses the Bulk Synchronous Parallel (BSP) computing model, which was established in 2012 as a Top-Level Project of The Apache Software Foundation. It provides not only pure BSP programming model but also vertex and neuron centric programming models, inspired by Google's Pregel and DistBelief [35]. It avoids the processing overhead of MapReduce approach such as sorting, shuffling, reducing the vertices etc. Hama provides a message passing interface and each superstep in BSP is faster than a full job execution in MApReduce framework, such as Hadoop [462].

11.6.9 Giraph

Apache Giraph is an iterative graph processing system built for big data [42]. It utilizes Hadoop Mapreduce technology for processing graphs [43] Giraph was initially developed by Yahoo based on the paper published by Google on Pregel. [184] Facebook with some improvements on Giraph could analyze real world graphs up to a scale of a trillion. Giraph can directly interface with HDFS and Hive (As it's developed in Java). [528]

11.6.10 Pregel

11.6.11 Pegasus

See #4 above.

11.6.12 Ligra

Ligra is a Light Weight Graph Processing Framework for the graph manipulation and analysis in shared memory system. It is particularly suited for implementing on parallel graph traversal algorithms where only a subset of the vertices are processed in an iteration. The interface is lightweight as it supplies only a few functions. The Ligra framework has two very simple routines, one for mapping over edges and one for mapping over vertices.

The implementations of several graph algorithms like BFS, breadth-first search, betweenness centrality, graph radii estimation, graph-connectivity, PageRank and Bellman-Ford single-source shortest paths efficient and scalable, and often achieve better running times than ones reported by other graph libraries/systems [543]. Although the shared memory machines cannot be scaled to the same size as distributed memory clusters, but the current commodity single unit servers can easily fit graphs with well over a hundred billion edges in the shared memory systems that are large enough for any of the graphs reported in the paper [338].

11.6.13 GraphChi

GraphChi is a disk-based system for computing efficiently on graphs with large number of edges. It uses a well-known method to break large graphs into small parts, and executes data mining, graph mining, machine learning algorithms. GraphChi can process over one hundred thousand graph updates per second, while simultaneously performing computation [364]. GraphChi is a spin-off of the GraphLab. GraphChi brings web-scale graph computation, such as analysis of social networks, available to anyone with a modern laptop

11.6.14 Galois

Galois system was built by intelligent software systems team at University of Texas, Austin. As explained in [630], “Galois is a system that automatically executes ‘Galoized’ serial C++ or Java code in parallel on shared-memory machines. It works by exploiting amorphous data-parallelism, which is present even in irregular codes that are organized around pointer-based data structures such as graphs and trees”. By using Galois provided data structures programmers can write serial programs that gives the performance of parallel execution. Galois employs annotations at loop levels to understand correct context during concurrent execution and executes the code that could be run in parallel. The key idea behind Galois is Tao-analysis, in which parallelism is exploited at compile time rather than at run time by creating operators equivalent of the code by employing data driven local computation algorithm [470]. Galois currently supports C++ and Java.

11.6.15 Medusa-GPU

Graphs are commonly used data structures . However, developers may find it challenging to write correct and efficient programs. Furthermore, graph processing is further complicated by irregularities of graph structures. Medusa enables the developers to write sequential C/C++ code. According to [328] it provides a set of APIs which embraces a runtime system to automatically execute those APIs in parallel. A number of optimization techniques are implemented to improvise the efficiency of graph processing. The experimental results provided in the paper [328] demonstrate that (1) Medusa greatly simplifies implementation of GPGPU programs for graph processing, with many fewer lines of source code written by developers; (2) The optimization techniques significantly improve the performance of the runtime system, making its performance comparable with or better than manually tuned GPU graph operations. [108] Medusa has proved to be a powerful framework for networked digital audio and video framework. [108] By exploiting the APIs it takes a modular approach to construct complex graph systems.

11.6.16 MapGraph

11.6.17 Totem

Totem is a project to overcome the current challenges in graph algorithms. The project is research the Networked Systems Laboratory (NetSysLab) The issue resides in the scale of real world graphs and the inability to process them on platforms other than a supercomputer. Totem is based on a bulk synchronous parallel(BSP) model that can enable hybrid CPU/GPU systems to process graph based applications in a cost effective manner. [422]

11.7 Inter process communication Collectives

11.7.1 point-to-point

11.7.2 (a) publish-subscribe: MPI

see <http://www.slideshare.net/Foxsden/high-performance-processing-of-streaming-data>

11.7.3 (b) publish-subscribe: Big Data

Publish/Subscribe (Pub/Sub) [536] is a communication paradigm in which subscribers register their interest as a pattern of events or topics and then asynchronously receive events matching

their interest. On the other hand, publishers generate events that are delivered to subscribers with matching interests. In Pub/sub systems, publishers and subscribers need not know each other. Pub/sub technology is widely used for a loosely coupled interaction between disparate publishing data-sources and numerous subscribing data-sinks. The two most widely used pub/sub schemes are - Topic-Based Publish/Subscribe (TBPS) and Content-Based Publish/Subscribe (CBPS) [181].

Big Data analytics architecture are being built on top of a publish/subscribe service stratum, serving as the communication facility used to exchange data among the involved components [180]. Such a publish/subscribe service stratum brilliantly solves several interoperability issues due to the heterogeneity of the data to be handled in typical Big Data scenarios.

Pub/Sub systems are being widely deployed in Centralized datacenters, P2P environments, RSS feed notifications, financial data dissemination, business process management, Social interaction message notifications- Facebook, Twitter, Spotify, etc.

11.7.4 HPX-5

Based on [709], High Performance ParallelX (HPX-5) is an open source, distributed model that provides opportunity for operations to run unmodified on one-to-many nodes. The dynamic nature of the model accommodates effective “computing resource management and task scheduling”. It is portable and performance-oriented. HPX-5 was developed by IU Center for Research in Extreme Scale Technologies (CREST). Concurrency is provided by lightweight control object (LCO) synchronization and asynchronous remote procedure calls. ParallelX component allows for termination detection and supplies per-process collectives. It “addresses the challenges of starvation, latency, overhead, waiting, energy and reliability”. Finally, it supports OpenCL to use distributed GPU and coprocessors. HPX-5 could be compiled on various OS platforms , however it was only tested on several Linux and Darwin (10.11) platforms. Required configurations and environments could be accessed via [710].

11.7.5 Argo BEAST HPX-5 BEAST PULSAR

Search on the internet was not successful.

11.7.6 Harp

Harp [270] is a simple, easy to maintain, low risk and easy to scale static web server that also serves Jade, Markdown, EJS, Less, Stylus, Sass, and CoffeeScript as HTML, CSS, and JavaScript without any configuration and requires low cognitive overhead. It supports the beloved layout/partial paradigm and it has flexible metadata and global objects for traversing the file system and injecting custom data into templates. It acts like a lightweight web server that was powerful enough for me to abandon web frameworks for dead simple front-end publishing. Harp can also compile your project down to static assets for hosting behind any valid HTTP server.

11.7.7 Netty

Netty [423] “is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients”. Netty [389] “is more than a collection of interfaces and classes; it also defines an architectural model and a rich set of design patterns”. It is protocol agnostic, supports both connection oriented protocols using TCP and connection less protocols built using UDP. Netty offers performance superior to standard Java NIO API thanks to

optimized resource management, pooling and reuse and low memory copying.

11.7.8 ZeroMQ

In [140], ZeroMQ is introduced as a software product that can “connect your code in any language, on any platform” by leveraging “smart patterns like pub-sub, push-pull, and router-dealer” to carry “messages across inproc, IPC, TCP, TIPC, [and] multicast.” In [139], it is explained that ZeroMQ’s “asynchronous I/O model” causes this “tiny library” to be “fast enough to be the fabric for clustered products.” In [140], it is made clear that ZeroMQ is “backed by a large and open source community” with “full commercial support.” In contrast to Message Passing Interface (i.e. MPI), which is popular among parallel scientific applications, ZeroMQ is designed as a fault tolerant method to communicate across highly distributed systems.

11.7.9 ActiveMQ

Apache ActiveMQ is a powerful open source messaging and Integration Patterns server [11]. It is a message oriented middleware(MOM) for the Apache Software Foundation that provides high availability, reliability, performance, scalability and security for enterprise messaging [575]. The goal of ActiveMQ is to provide standard-based, message-oriented application integration across as many languages and platforms as possible. ActiveMQ implements the JMS spec and offers dozens of additional features and value on top of this specifications. ActiveMQ is used in many scenarios such as heterogeneous application integration, as a replacement for RPC and to loosen the coupling between applications.

11.7.10 RabbitMQ

RabbitMQ is a message broker [498] which allows services to exchange messages in a fault tolerant manner. It provides variety of features which “enables software applications to connect and scale”. Features are: reliability, flexible routing, clustering, federation, highly available queues, multi-protocol, many clients, management UI, tracing, plugin system, commercial support, large community and user base. RabbitMQ can work in multiple scenarios:

1. Simple messaging: producers write messages to the queue and consumers read messages from the the queue. This is synonymous to a simple message queue.
2. Producer-consumer: Producers produce messages and consumers receive messages from the queue. The messages are delivered to multiple consumers in round robin manner.
3. Publish-subscribe: Producers publish messages to exchanges and consumers subscribe to these exchanges. Consumers receive those messages when the messages are available in those exchanges.
4. Routing: In this mode consumers can subscribe to a subset of messages instead of receiving all messages from the queue.
5. Topics: Producers can produce messages to a topic multiple consumers registered to receive messages from those topics get those messages. These topics can be handled by a single exchange or multiple exchanges.
6. RPC:In this mode the client sends messages as well as registers a callback message queue. The consumers consume the message and post the response message to the callback queue.

RabbitMQ is based on AMPQ [435] (Advanced Message Queuing Protocol) messaging model. AMPQ is described as follows “messages are published to exchanges, which are often compared to

post offices or mailboxes. Exchanges then distribute message copies to queues using rules called bindings. Then AMQP brokers either deliver messages to consumers subscribed to queues, or consumers fetch/pull messages from queues on demand”

11.7.11 NaradaBrokering

NaradaBrokering [132], is a content distribution infrastructure for voluminous data streams. The substrate places no limits on the size, rate and scope of the information encapsulated within these streams or on the number of entities within the system. The smallest unit of this substrate called as broker, intelligently process and route messages, while working with multiple underlying communication protocols. The major capabilities of NaradaBrokering consists of providing a message oriented middleware (MoM) which facilitates communications between entities (which includes clients, resources, services and proxies thereto) through the exchange of messages and providing a notification framework by efficiently routing messages from the originators to only the registered consumers of the message in question [207]. Also, it provides salient stream oriented features such as their Secure end-to-end delivery, Robust disseminations, jitter reductions.

NaradaBrokering incorporates support for several communication protocol such as TCP, UDP, Multicast, HTTP, SSL, IPSec and Parallel TCP as well as supports enterprise messaging standards such as the Java Message Service, and a slew of Web Service specifications such as SOAP, WS-Eventing, WS-Reliable Messaging and WS-Reliability [131].

11.7.12 QPid

11.7.13 Kafka

Apache Kafka is a streaming platform, which works based on publish-subscribe messaging system and supports distributed environment.

Kafka lets you publish and subscribe to the messages. Kafka maintains message feeds based on ‘topic’. A topic is a category or feed name to which records are published. Kafka’s Connector APIs are used to publish the messages to one or more topics, whereas, Consumer APIs are used to subscribe to the topics.

Kafka lets you process the stream of data at real time. Kafka’s stream processor takes continual stream of data from input topics, processes the data in real time and produces streams of data to output topics. Kafka’s Streams API are used for data transformation.

Kafka lets you store the stream of data in distributed clusters. Kafka acts as a storage system for incoming data stream. As Kafka is a distributed system, data streams are partitioned and replicated across nodes.

Thus, a combination of messaging, storage and processing data stream makes Kafka a ‘streaming platform’. It can be used for building data pipelines where data is transferred between systems or applications. Kafka can also be used by applications that transform real time incoming data. [200]

11.7.14 Kestrel

Kestrel is a distributed message queue, with added features and bulletproofing, as well as the scalability offered by actors and the Java virtual machine. It supports multiple protocols: memcache: the memcache protocol; thrift: Apache Thrift-based RPC; text: a simple text-based protocol. Each queue is strictly ordered following the FIFO (first in, first out) principle. To keep up with

performance items are cached in system memory. Kestrel is more durable as queues are stored in memory for speed, but logged into a journal on disk so that servers can be shutdown or moved without losing any data. When kestrel starts up, it scans the journal folder and creates queues based on any journal files it finds there, to restore state to the way it was when it last shutdown (or was killed or died).

Kestrel uses a pull-based data aggregator system that convey data without prior definition on its destination. So the destination can be defined later on either storage system, like HDFS or NoSQL, or processing system, like storm and spark streaming. Each server handles a set of reliable, ordered message queues. When you put a cluster of these servers together, with no cross communication, and pick a server at random whenever you do a set or get, you end up with a reliable, loosely ordered message queue [348].

11.7.15 JMS

JMS (Java Messaging Service) is a java oriented messaging standard that defines a set of interfaces and semantics which allows applications to send, receive, create, and read messages. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous [663]. JMS overcomes the drawbacks of RMI (Remote Method Invocation) where the sender needs to know the method signature of the remote object to invoke it and RPC(Remote Procedure Call), which is tightly coupled i.e it cannot function unless the sender has important information about the receiver.

JMS establishes a standard that provides loosely coupled communication i.e the sender and receiver need not be present at the same time or know anything about each other before initiating the communication. JMS provides two communication domains. A point-to-point messaging domain where there is one producer and one consumer. On generating message, a producer simple pushes the message to a message queue which is known to the consumer. The other communication domain is publish/subscribe model, where one message can have multiple receivers [332].

11.7.16 AMQP

[32] AMQP stands for Advanced Message Queueing Protocol. AMQP is open interenet protocol that allows secure and reliable communication between applications in different orginization and different applications which are on diffferent platforms. AMQP allows businesses to implement middleware applications interoperability by allowing secure message transfer bewteen the applications on timely manner. AMQP is mainly used by financial and banking business. Other sectors that aslo use AMQP are Defence, Telecommunication, cloud Computing and so on. Apache Qpid, StormMQ, RabbitMQ, MQlight, Microsoft's Windows Azure Service Bus, IIT Software's SwiftMQ and JORAM are some of the products that implement AMQP protocol.

11.7.17 Stomp

11.7.18 MQTT

According to [406], Message Queueing Telemetry Transport (MQTT) protocol is an Interprocess communication protocol that could serve as better alternative to HTTP in certain cases. It is based on a publish-subscribe messaging pattern. Any sensor or remote machine can publish it's data and any registered client can subscribe the data. A broker takes care of the message being published by the remote machine and updates the subscriber in case of new message from the remote machine. The data is sent in binary format which makes it use less bandwidth. It is designed mainly to cater

to the needs to devices that has access to minimal network bandwidth and device resources without affecting reliability and quality assurance of delivery. MQTT protocol has been in use since 1999. One of the notable work is project Floodnet [407], which monitors river and floodplains through a set of sensors.

11.7.19 Marionette Collective

It is basically a framework for management of a system where the systems undergo an organized coordination resulting in an automated deployment of systems which creates an orderly workflow or a parallel wise job execution. It doesn't rely on central inventories such as SSH and uses tools such as Middleware :cite: 'www-marionette-webpage'. This gives an advantage of delivering a very scalable and quick execution environment. Mcollective gives us a huge advantage of working with a large number of servers , it uses publish/subscribe middleware for communicating with many hosts at once in a parallel manner. Mcollective allows us to interact with a cluster of servers at the same time, it allows us to use a simple command line to call remote agents and there isn't a centralized inventory. Mcollective uses a broadcast paradigm to distribute the requests , where all the servers receives the request at the same time which are also attached with a filter. The servers which match the filter will act on these requests.

11.7.20 Public Cloud: Amazon SNS

Amazon SNS is an Inter process communication service which gives the user simple, end-to-end push messaging service allowing them to send messages, alerts, or notifications. According to [554], it can be used to send a directed message intended for an entity or to broadcast messages to list of selected entities. It is an easy to use and cost effective mechanism to send push messages. Amazon SNS is compatible to send push notifications to iOS, Windows, Fire OS and Android OS devices.

According to [555] SNS system architecture consists of four elements: (1) Topics, (2) Owners, (3) Publishers, and (4) Subscribers. Topics are events or access points that identifies the subject of the event and can be accessed by an unique identifier(URI). Owners create topics and control all access to the topic and define the corresponding permission for each topic. Subscribers are clients (applications, end-users, servers, or other devices) that want to receive messages or notifications on specific topics of interest to them.Publishers send messages to topics. SNS matches the topic with the list of subscribers interested in the topic, and delivers the message to them.

According to [556], Amazon SNS follows pay as per usage. In general it is \$0.50 per 1 million Amazon SNS Requests.Amazon SNS supports notifications over multiple transport protocols such as HTTP/HTTPS, Email/Email-JSON, SQS(Message queue) and SMS.Amazon SNS can be used with other AWS services such as Amazon SQS, Amazon EC2 and Amazon S3.

11.7.21 Lambda

AWS Lambda is a product from amazon which facilitates serverless computing [77].AWS Lambda allows for running the code without the need for provisioning or managing servers, all server management is taken care by AWS.The code to be run on AWS Lambda is called a server function which can be written in Node.js,Python,Java,C#.Each Lambda function is to be stateless and any persistent data needs are to be handled through storage devices.AWS Lambda function can be setup using the AWS Lambda console where one can setup the function code and specify the event that triggers the functional call.AWS Lamda service supports multiple event sources as identified in [78].

AWS Lambda is designed to use replication and redundancy to provide for high availability both for the service itself and the function it runs. AWS Lambda automatically scales your application by running the code in response to each trigger. The code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload. Billing for AWS Lambda is based on the number of times the code executes and in 100 ms increments of the duration of the processing.

11.7.22 Google Pub Sub

Google Pub/Sub provides an asynchronous messaging facility which assists the communication between independent applications [252]. It works in real time and helps keep the two interacting systems independent. It is the same technology used by many of the Google apps like GMail, Ads, etc. and so integration with them becomes very easy. Some of the typical features it provides are: (1) Push and Pull - Google Pub/Sub integrates quickly and easily with the systems hosted on the Google Cloud Platform thereby supporting one-to-many, one-to-one and many-to-many communication, using the push and pull requests. (2) Scalability - It provides high scalability and availability even under heavy load without any degradation of latency. This is done by using a global and highly scalable design. (3) Encryption - It provides security by encryption of the stored data as well as that in transit. Other than these important features, it provides some others as well, like the usage of RESTful APIs, end-to-end acknowledgement, replicated storage, etc [251].

11.7.23 Azure Queues

Azure Queues storage is a Microsoft Azure service, providing inter-process communication by message passing [546]. A sender sends the message and a client receives and processes them. The messages are stored in a queue which can contain millions of messages, up to the total capacity limit of a storage account [79]. Each message can be up to 64 KB in size. These messages can then be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. Similar to the other message queue services, Azure Queues enables decoupling of the components [621]. It runs in an asynchronous environment where messages can be sent among the different components of an application. Thus, it provides an efficient solution for managing workflows and tasks. The messages can remain in the queue up to 7 days, and afterwards, they will be deleted automatically.

11.7.24 Event Hubs

Azure Event Hubs is a hyper-scale telemetry ingestion service. It collects, transforms, and stores millions of events. As a distributed streaming platform, it offers low latency and configurable time retention enabling one to ingest massive amounts of telemetry into the cloud and read the data from multiple applications using publish-subscribe semantics. [395] It is a highly scalable data streaming platform. Data sent to an Event Hub can be transformed and stored using any real-time analytics provider or batching/storage adapters. With the ability to provide publish-subscribe capabilities, Event Hubs serves as the “on ramp” for Big Data.

11.8 In-memory databases/caches

11.8.1 Gora (general object from NoSQL)

Gora is an in-memory data model [253] which also provides persistence to the big data. Gora provides persistence to different types of data stores. Primary goals of Gora are:

1. data persistence
2. indexing
3. data access
4. analysis
5. map reduce support

Unlike ORM models which mostly work with relational databases for example hibernate gora works for most type of data stores like documents, columnar, key value as well as relational. Gora uses beans to maintain the data in-memory and persist it on disk. Beans are defined using apache avro schema. Gora provides modules for each type of data store it supports. The mapping between bean definition and datastore is done in a mapping file which is specific to a data store. Type Gora workflow will be:

1. define the bean used as model for persistence
2. use gora compiler to compile the bean
3. create a mapping file to map bean definition to datastore
4. update gora.properties to specify the datastore to use
5. get an instance of corresponding data store using datastore factory.

Gora has a query interface to query the underlying data store. Its configuration is stored in gora.properties which should be present in classpath. In the file you can specify default data store used by Gora engine. Gora also has a CI/CD library call GoraCI which is used to write integration tests.

11.8.2 Memcached

Memcached is a free and open-source, high performance, distributed memory object caching system. [162] Although, generic in nature,it is intended for se in speeding up dynamic web applications by reducing the database load.

It can be thought of as a short term memory for your applications. Memcached is an in-memory key-value store for small chunks of arbitrary data from the results of database calls, API calls and page rendering. Its API is available in most of the popular languages. In simple terms, it allows you to take memory from parts of your system where you have more memory than you need and allocate it to parts of your system where you have less memory than you need.

11.8.3 Redis

Redis (Remote Dictionary Server) is an open source ,in-memory, key-value database which is commonly referred as a data structure server. “It is called a data structure server and not simply a key-value store because Redis implements data structure which allows keys to contain binary safe strings, hashes, sets, and sortedsets as well as lists” [382]. Redis’s better performance, easy to use and implement, and atomic manipulation of data structures lends itself to solving problems that are difficult to solve or perform poorly when implemented with traditional relational databases. “Salivator Sanfilippo (Creator of open-source database Redis) makes a strong case that Redis does not need to replace the existing database but is an excellent addition to an enterprise for new functionalities or to solve sometimes intractable problems.” [417]

A widely used use pattern for Redis is an in-memory cache for web-applications and the other being the use of pattern for REDIS for metric storage of such quantitative data such as the web page usage and user behavior on gamer leaderboards where using a bit operations on strings, Redis very efficiently stores binary information on a particular characteristics [417].The other popular Redis use pattern is a communication layer between different systems through a publish/subscribe (pub/sub for short), where one can post the message to one or more channels that can be acted upon by other systems that are subscribed to or listening to that channel for incoming messages. The Companies using REDIS includes Twitter to store the timelines of all the user , Pinterest stores the user follower graph, Github, popular web frameworks like Node.js , Django, Ruby-on-Rails etc.

11.8.4 LMDB (key value)

LMDB (Lighting memory-mapped Database) is a high performance embedded transactional database in form of a key-value store [671]. LMDB is designed around virtual memory facilities found in modern operating systems, multi-version concurrency control (MVCC) and single-level store (SLS) concepts. LMDB stores arbitrary key/data pairs as byte arrays, provides a range-based search capability, supports multiple data items for a single key and has a special mode for appending records at the end of the database (MDB_APPEND) which significantly increases its write performance compared to other similar databases.

LMDB is not a relational database [695] and strictly uses key-value store. Key-value databases allows one write at a time, the difference that LMDB highlights is that write transactions do not block readers nor do readers block writes. Also, it does allow multiple applications on the same system to open and use the store simultaneously which helps in scaling up performance [269].

11.8.5 Hazelcast

Hazelcast is a java based, in memory data grid [689]. It is open source software, released under the Apache 2.0 License [272]. Hazelcast enables predictable scaling for applications by providing in memory access to data. Hazelcast uses a grid to distribute data evenly across a cluster. Clusters allow processing and storage to scale horizontally. Hazelcast can run locally, in the cloud, in virtual machines, or in Docker containers. Hazelcast can be utilized for a wide variety of applications. It has APIs for many programing languages including Python, Java, Scala, C++, .NET and Node.js and supports any binary languages through an Open Binary Client Protocol [689].

11.8.6 Ehcache

EHCACHE is an open-source Java-based cache. It supports distributed caching and could scale to hundred of caches. It comes with REST APIs and could be integrated with popular frameworks like Hibernate [170]. It offers storage tires such that less frequently data could be moved to slower tires [171]. It's XA compliant and supports two- phase commit and recovery for transactions. It's developed and maintained by Terracotta and is available under Apache 2.0 license. It conforms to Java caching standard JSR 107.

11.8.7 Infinispan

Infinispan is a highly available, extremely scalable key/value data store and data grid platform. The design perspective of infinispan is exposing a distributed,highly concurrent data structure to make the most use of modern multi-core as well as multi-processor architectures. It is mostly used as a distributed cache, but also can be used as a object database or NoSQL key/value store [311].

Infinispan is mostly used as a cache store. It is predominantly used for applications that are clustered, and requires a cache coherency for data consistency. Infinispan is written in java and is open source. It is fully transactional. Infinispan is used to add clusterability as well as high availability to frameworks. Infinispan has many use-cases,they are: 1) it can be used as a distributed cache 2)Storage for temporal data, like web sessions, 3)Cross-JVM communication, 4)Shared storage, 5)In-memory data processing and analytics and 6)MapReduce Implementation in the In-Memory Data Grid. It is also used in research and academia as a framework for distribution execution and storage [306].

11.8.8 VoltDB

VoltDB is an in-memory database. It is an ACID-compliant RDBMS which uses a shared nothing architecture to achieve database parallelism. It includes both enterprise and community editions. VoltDB is a scale-out NewSQL relational database that supports SQL access from within pre-compiled Java stored procedures. VoltDB relies on horizontal partitioning down to the individual hardware thread to scale, k-safety (synchronous replication) to provide high availability, and a combination of continuous snapshots and command logging for durability (crash recovery) [640]. The in-memory, scale-out architecture couples the speed of traditional streaming solutions with the consistency of an operational database. This gives a simplified technology stack that delivers low-latency response times (1ms) and hundreds of thousands of transactions per second. VoltDB allows users to ingest data, analyze data, and act on data in milliseconds, allowing users to create per-person, real-time experiences [640].

11.8.9 H-Store

H-Store is an in memory and parallel database management system for on-line transaction processing (OLTP). Specifically , [286] illustrates that H-Store is a highly distributed, row-store-based relational database that runs on a cluster on shared-nothing, main memory executor nodes. As Noted in [284] “the architectural and application shifts have resulted in modern OLTP databases increasingly falling short of optimal performance.In particular, the availability of multiple-cores, the abundance of main memory, the lack of user stalls, and the dominant use of stored procedures are factors that portend a clean-slate redesign of RDBMSs”.The H-store which is a complete redesign has the potential to outperform legacy OLTP databases by a significant factor. As detailed in [287] H-Store is the first implementation of a new class of parallel DBMS, called NewSQL, that provides the high-throughput and high-availability of NoSQL systems, but without giving up the transactional guarantees of a traditional DBMS. The H-Store system is able to scale out horizontally across multiple machines to improve throughput, as opposed to moving to a more powerful , more expensive machine for a single-node system.

11.9 Object-relational mapping

11.9.1 Hibernate

Hibernate is an open source project which provides object relational persistence framework for applications in Java. It is an Object relational mapping library (ORM) which provides the framework for mapping object oriented model to relational database. It provides a query language, a caching layer and Java Management Extensions (JMX) support. Databases supported by Hibernate includes DB2, Oracle, MySQL, PostgreSQL. To provide persistence services, Hibernate uses database and configuration data. For using hibernate, firstly a java class is created which represents table in the database. Then columns in database are mapped to the instance variables of created Java class. Hibernate can perform database operations like select, insert, delete and update records in table by automatically creating query. Connection management and transaction management are provided by hibernate. Hibernate saves development and debugging time in comparison to JDBC. But it is slower at runtime as it generates many SQL statements at runtime. It is database independent. For batch processing it is advisable to use JDBC over Hibernate [83]

11.9.2 OpenJPA

According to [354], Apache OpenJPA is a Java persistence project developed by The Apache Software Foundation that can either be used as Plain old Java Object (POJO) or could be used in any Java EE compliant containers. It provides object relational mapping which effectively simplifies the storing of relational dependencies among objects in databases. [335] mentions that Kodo, an implementation of Java Data Objects acted as a precursor to the development of OpenJPA. In 2006, BEA Systems donated the majority of the source code of Kodo to The Apache Software Foundation under the name OpenJPA. Being a POJO, OPenJPA can be used without needing to extend prespecified classes, implementing predefined interfaces and inclusion of annotations. OPenJPA can be used in cases where the focus of the project is majorly on business logic and has no dependencies on enterprise frameworks. OPenJPA can be implemented across multiple operating systems, on account of its function of cross platform support. It is written in Java and a most recent stable release came out in April 20, 2016 under the version 2.4.1 with Apache License 2.0.

11.9.3 EclipseLink

EclipseLink is an open source persistence Services project from Eclipse foundation. It is a framework which provide developers to interact with data services including database and web services, Object XML mapping etc. [668]. This is the project which was developed out of Oracle's Toplink product. The main difference is EclipseLink does not have some key enterprise feature. Eclipselink support a number of persistence standard model like JPA, JAXB, JCA and Service Data Object. Like Toplink, the ORM (Object relational model) is the technique to convert incompatible type system in Object Oriented programming language. It is a framework for storing java object into relational database.

11.9.4 DataNucleus

DataNucleus (available under Apache 2 open source license) is a data management framework in Java. Formerly known as 'Java Persistent Objects' (JPOX) this was relaunched in 2008 as 'DataNucleus'. According to [662] DataNucleus Access Platform is a fully compliant implementation of the Java Persistent API (JPA) and Java Data Objects (JDO) specifications. It provides persistence and retrieval of data to a number of datastores using a number of APIs, with a number of query languages. In addition to object-relational mapping (ORM) it can also map and manage data from sources other than RDBMS (PostgreSQL, MySQL, Oracle, SQLServer, DB2, H2 etc.) such as Map-based (Cassandra, HBase), Graph-based (Neo4j), Documents (XLS, OOXML, XML, ODF), Web-based (Amazon S3, Google Storage, JSON), Doc-based (MongoDB) and Others (NeoDatis, LDAP). It supports the JPA (Uses JPQL Query language), JDO (Uses JDOQL Query language) and REST APIs [150]. DataNucleus products are built from a sequence of plugins where each of it is an OSGi bundle and can be used in an OSGi environment. Google App Engine uses DataNucleus as the Java persistence layer [336].

11.9.5 ODBC/JDBC

Open Database Connectivity (ODBC) is an open standard application programming interface (API) for accessing database management systems (DBMS) [442]. ODBC was developed by the SQL Access Group and released in September, 1992. Microsoft Windows was the first to provide an ODBC product. Later the versions for UNIX, OS/2, and Macintosh platforms were developed. ODBC is independent of the programming language, database system and platform.

Java Database Connectivity (JDBC) is a API developed specific to the Java programming language. JDBC was released as part of Java Development Kit (JDK) 1.1 on February 19, 1997 by Sun Microsystems [323]. The ‘java.sql’ and ‘javax.sql’ packages contain the JDBC classes. JDBC is more suitable for object oriented databases. JDBC can be used for ODBC compliant databases by using a JDBC-to-ODBC bridge.

11.10 Extraction Tools

11.10.1 UIMA

Unstructured Information Management applications (UIMA) provides a framework for content analytics. It searches unstructured data to retrieve specific targets for the user. For example, when a text document is given as input to the system, it identifies targets such as persons, places, objects and even associations. According to , [666] theUIMA architecture can be thought of as four dimensions: 1. Specifies component interfaces in analytics pipeline. 2. Describes a set of Design patterns. 3. Suggests two data representations: an in-memory representation of annotations for high-performance analytics and an XML representation of annotations for integration with remote web services. 4. Suggests development roles allowing tools to be used by users with diverse skills.

UIMA uses different, possibly mixed, approaches which include Natural Language Processing, Machine Learning, IR. UIMA supports multimodal analytics [550] which enables the system to process the resource fro various points of view. UIMA is used in several software projects such as the IBM Research’s Watson uses UIMA for analyzing unstructured data and Clinical Text Analysis and Knowledge Extraction System (Apache cTAKES) which is a UIMA-based system for information extraction from medical records.

11.10.2 Tika

“The Apache Tika toolkit detects and extracts metadata and text from over a thousand different file types (such as PPT, XLS, and PDF). All of these file types can be parsed through a single interface, making Tika useful for search engine indexing, content analysis, translation, and much more. [609]”

11.11 SQL(NewSQL)

11.11.1 Oracle

Oracle database is an object-relational database management system by Oracle. Following are some of the key features of Oracle [453] 1. ANSI SQL Compliance 2. Multi-version read consistency 3. Procedural extensions: PL/SQL and Java. Apart from above they are performance related features, including but not limited to: indexes, in-memory, partitioning, optimization. As of today the latest release of Oracle is [453] Oracle Database 12c Release 1: 12.1 (Patch set as of June 2013)

11.11.2 DB2

DB2 is a Relational DataBase Management System (RDBMS). Though initially introduced in 1983 by IBM to run exclusively on its MVS (Multiple Virtual Storage) mainframe platform, it was later extended to other operating systems like UNIX, Windows and Linux. It is used to store, analyze and retrieve the data and is extended with the support of Object-Oriented features and

non-relational structures with XML [152]. DB2 server editions include: Advanced Enterprise Server Edition and Enterprise Server Edition (AESE / ESE) designed for mid-size to large-size business organizations, Workgroup Server Edition (WSE) designed for Workgroup or mid-size business organizations, Express -C provides the capabilities of DB2 at no charge and can run on any physical or virtual systems, Express Edition designed for entry level and mid-size business organizations, Enterprise Developer Edition offers single application developer useful to design, build and prototype the applications for deployment on the IBM server. DB2 has APIs for REXX, PL/I, COBOL, RPG, FORTRAN, C++, C, Delphi, .NET CLI, Java, Python, Perl, PHP, Ruby, and many other programming languages. DB2 also supports integration into the Eclipse and Visual Studio integrated development environments [151].

11.11.3 SQL Server

SQL Server [567] is a relational database management system from Microsoft. As of Jan 2017, SQL Server is available in below editions

1. Standard - consists of core database engine
2. Web - low cost edition for web hosting
3. Business Intelligence - includes standard edition and business intelligence tools like PowerPivot, PowerBI, Master Data Services
4. Enterprise - consists of core database engine and enterprise services like cluster manager
5. SQL Server Azure - [81] core database engine integrated with Microsoft Azure cloud platform and available in platform-as-a-service mode.

In the book [400], the technical architecture of SQL Server in OLTP(online transaction processing), hybrid cloud and business intelligence modes is explained in detail.

11.11.4 SQLite

SQLite is a severless SQL database engine whose source code resides in the public domain [565]. SQLite databases, including tables, indices, and views, reside on a single file on the disk [565]. It has a compact library, often taking up less than KiB of space, depending on the particular configuration [565]. Performance is the tradeoff with the smaller size, i.e. performance usually runs faster when given more memory [565]. SQLite transactions comply with the ACID (Atomicity, Consistency, Isolation, Durability) [384] properties [565]. SQLite does not require administration or configuration [620]. There are some limitations associated with SQLite, such as the inability to perform Right Outer Joins, read-only views, and access permissions (other than those that are associated with regular file acces permissions) [620] SQLite does not compare directly with client/server databases such as MySQL as they are both trying to solve different problems [566]. While database engines such as MySQL aim to provide a shared database, with different access permissions to different individuals/applications, SQLite has the goal of being a local repository of data for applications [566] While SQLite is not appropriate for every situation, there certainly exists situations where it can prove to be a prudent choice for data management needs [566].

11.11.5 MySQL

MySQL is a relational database management system. [412] SQL is an acronym for Structured Query Language and is a standardized language used to interact with the databases. [412] Databases provide structure to a collection of data while. [412] A database management system allows for the addition, accessing, and processing of the data stored in a database. [412] Relational databases utilize tables that are broken down into columns, representing the various fields of the table, and rows, which correspond to individual entries in the table. [283]

11.11.6 PostgreSQL

PostgreSQL is an open-source relational database management system (DBMS). It runs on all the major operating systems like Linux, Mac OSX, Windows and UNIX. It supports the ACID (Atomicity, Consistency, Isolation and Durability) properties of a conventional DBMS. It supports the standard SQL:2008 data types like INTEGER, NUMERIC, etc. besides providing native interfaces for languages such as C++, C, Java and .Net [601].

With the release of its latest version 9.5, it has included new features like the UPSERT capability, Row Level security and multiple features to support Big Data. These new features rolled out in the latest version make PostgreSQL a very strong contender for modern use. UPSERT feature has predominantly been released for the application developers in order to help them simplify their web application and software development. UPSERT is basically a shorthand of “Insert, on conflict update”. Row Level Security (RLS), as the name suggests, enables the database administrators to control which particular rows could be updated by the users. This helps in ensuring that the users do not inadvertently update rows which they are not meant to. Features such as BRIN indexing, Faster sorts, CUBE, ROLLUP and GROUPING SETS, Foreign Data Wrappers and TABLESAMPLE were added as a part of the new Big Data features. Under BRIN indexing (Block Range Indexing), PostgreSQL supports creating small but powerful indexes for large tables. Using a new algorithm called as “abbreviated keys”, PostgreSQL can sort NUMERIC data very quickly. The CUBE, ROLLUP and GROUPING clauses enable the users to use just a single query to create myriad reports at different levels of summarization. Using the concept of Foreign Data Wrappers (FDWs), PostgreSQL can be used for querying Big Data systems like Cassandra and Hadoop. The TABLESAMPLE clause allows quick statistical sample generation of huge tables without any need to sort them [602].

11.11.7 CUBRID

CUBRID name is deduced from the combination of word CUBE(security within box) and BRIDGE(data bridge). It is an open source Relational DataBase Management System designed in C programming language with high performance, scalability and availability features. During its development by NCL, korean IT service provider the goal was to optimize database performance for web-applications. [144] Importantly most of the SQL syntax from MYSQL and ORACLE can work on cubrid.CUBRID also provides manager tool for database administration and migration tool for migrating the data from DBMS to CUBRID bridging the dbs. CUBRID enterprise version and all the tools are free and suitable database candidate for web-application development.

11.11.8 Galera Cluster

Galera cluster [214] is a type of database clustering which has all multiple masters and works on synchronous replication. At a deeper level, it was created by extending MySql replication API to provide all support for true multi master synchronous replication. This extended api is called as Write-Set Replication API and is the core of the clustering logic. Each transaction of wsrep API not only contains the record but also other meta-info to requires to commit each node separately or asynchronously. So though it seems synchronous logically but works independently on each node. The approach is also called virtually synchronous replication. This helps in directly read-write on a specific node and can lose a node without handling any complex failover scenarios (zero downtime).

11.11.9 SciDB

SciDB is an open source DBMS based on multi-dimensional array data model and runs on Linux platform. [571] The data store is optimized for mathematical operations such as linear algebra and statistical analysis. The data can be distributed across multiple nodes in a cluster.

The dimensions of the data can be either standard integers or user-defined types. Ragged arrays are also supported. The data is accessed through AQL, a SQL like language designed specifically for array operations. It supports operations such as to filter and join arrays and aggregation over the cell values. It has few similarities to Postgres in terms of user-defined scalar functions and storage manager. Old values of data are updated instead of being deleted to retain different versions of a cell. The arrays are divided into chunks and partitioned across the nodes in the cluster, with provision of caching some of them in the main memory.

11.11.10 Rasdaman

Rasdaman is an specialized database management system which adds capabilities for storage and retrieval of massive multi-dimensional array, such as sensors,image, and statistics data. [672] It is written in C++ language. For example, it can serve 1-D measurement data, 2-D satellite data, 3-D x/y/t image series and x/y/z exploration data, 4-D ocean and climate data, and much more.

[503]: Rasdaman servers provides functionality from geo service up to complex analytics which are related to spatio-temporal raster data. It also integrates smoothly with R, OpenLayers, NASA WorldWind etc. via APIs calls. It is massively used in the domains like earth, space, and social science related fields.

11.11.11 Apache Derby

[591]: Apache Derby is java based relational database system. Apache Derby has JDBC driver which can be used by Java based applications. Apache derby is part of the Apache DB subproject and licensed under Apache version 2.0.

[592]: Derby Embedded Database Engine is the database engine with JDBC and SQL as programming APIs. Client/Server functionality is achieved by Derby network server, it allows connection through TCP/IP using DRDA protocol. ij, database utility makes it possible for SQL scripts to be run on JDBC database. The dblook utility is the schema extraction tool. The sysinfo utility is used for displaying version of Java environment and Derby.

There are two deployment options for Apache Derby , embedded and Derby network server option. In embedded framework, Derby is started and stopped by the single user java application without any adiministration required. In the case of Derby network server configuration, Derby is started by multi user java application over TCP/IP. Since Apache Derby is written in Java, it runs on any certified JVM(Java Virtual Machine). [590]:

11.11.12 Pivotal Greenplum

Pivotal Greenplum is a commercial fully featured data warehouse. It is powered by Greenplum Database an open source initiative. “It is powered by advanced cost-based query optimizer thereby delivering high analytical query performance on large data volumes”. Pivotal Greenplum is uniquely focused on big data analytics [472].

The system consists of a master node, standy master node and segment nodes. The master node

consists of the catalog information whereas the data resides on the segment nodes. The segment nodes runs on one or more segments which are modified PostgreSQL databases and are assigned a content identifier. The data is distributed among these segment nodes. The segment node also supports bulk loading and unloading. The master node parses, optimizes an SQL query and dispatch it to all segment nodes. Therefore, it provides powerful and rapid analytics on petabyte scale data volumes [473].

11.11.13 Google Cloud SQL

Google Cloud SQL is a fully managed data base as service developed by Google where Google manages the backup, patching and replication of the databases etc [245]. Cloud SQL database aims at developers to focus on app development leaving database administration to a minimum. This can be understood as 'MySQL on Cloud' as most of the features from MySQL 5.7 are directly supported in Cloud SQL. The service is offered with 'Pay per use' providing the flexibility and 'better performance per dollar'. Cloud SQL is scalable up to 16 processor cores and more than 100GB of RAM. [246]

11.11.14 Azure SQL

11.11.15 Amazon RDS

According to Amazon Web Services, Amazon Relation Database Service (Amazon RDS) is a web service which makes it easy to setup, operate and scale relational databases in the cloud. As mentioned in [508] It allows to create and use MySQL, Oracle, SQL Server, and PostgreSQL databases in the cloud. Thus, codes, applications and tools used with existing databases can be used with Amazon RDS. The basic components of Amazon (As listed in [507]) RDS include: DB Instances: DB instance is an isolated database environment in the cloud. Regions and availability zones: Region is a data center location which contains Availability Zones. Availability Zone is isolated from failures in other Availability Zones. Security groups: controls access to DB instance by allowing access to IP address ranges or Amazon EC2 instances that is specified. DB parameter groups: manage configuration of DB engine by specifying engine configuration values that are applied to one or more DB instances of the same instance type. DB option groups: Simplifies data management through Oracle Application Express (APEX), SQL Server Transparent Data Encryption, and MySQL memcached support.

11.11.16 Google F1

F1 is a distributed relational database system built at Google to support the AdWords business. It is a hybrid database that combines high availability, the scalability of NoSQL systems like Bigtable, and the consistency and usability of traditional SQL databases. F1 is built on Spanner, which provides synchronous cross-datacenter replication and strong consistency [545].

F1 features include a strictly enforced schema, a powerful parallel SQL query engine, general transactions, change tracking and notification, and indexing, and is built on top of a highly-distributed storage system that scales on standard hardware in Google data centers. The store is dynamically sharded and is able to handle data center outages without data loss [544]. The synchronous cross-datacenter replication and strong consistency results in higher commit latency which can be overcome using hierarchical schema model with structured data types and through smart application design.

11.11.17 IBM dashDB

IBM dashDB is a data warehousing service hosted in cloud , This aims at integrating the data from various sources into a cloud data base. Since the data base is hosted in cloud it would have the benifits of a cloud like scalability and less maintainance. This data base can be configured as 'transaction based' or 'Analytics based' depending on the work load [3] .This is available through ibm blue mix cloud platform.

dash DB has build in analytics based on IBM Netezza Analytics in the PureData System for Analytics. Because of the build in analytics and support of in memory optimization promises better performance efficieny. This can be run alone as a standalone or can be connected to variousBI or analytic tools. [298]

11.11.18 N1QL**11.11.19 BlinkDB****11.11.20 Spark SQL**

Spark SQL is Apache Spark's module for working with structured data. Spark SQL is a new module that integrates relational processing with Spark's functional programming API [593]. It is used to seamlessly mix SQL queries with Spark programs. Spark SQL lets you query structured data inside Spark programs, using either SQL or a familiar DataFrame API. it offers much tighter integration between relational and procedural processing, through a declarative DataFrame API that integrates with procedural Spark code. Spark SQL reuses the Hive frontend and metastore, giving you full compatibility with existing Hive data, queries, and UDFs by installing it alongside Hive. Spark SQL includes a cost-based optimizer, columnar storage and code generation to make queries fast [9]. At the same time, it scales to thousands of nodes and multi hour queries using the Spark engine, which provides full mid-query fault tolerance.

11.12 NoSQL**11.12.1 Lucene**

Apache Lucene [46] is a high-performance, full-featured text search engine library. It is originally written in pure Java but also has been ported to few other languages chiefly python. It is suitable for applications that requires full-text search. One of the key implementation of Lucene is Internet search engines and local, single-site searching. Another important implementation usage is its recomendation system. The core idea of Lucene is to extract text from any document that contains text (not image) field, making it format idependent.

11.12.2 Solr**11.12.3 Solandra**

Solandra is a highly scalable real-time search engine built on Apache Solr and Apache Cassandra. Solandra simplifies maintaining a large scale search engine, something that more and more applications need. At its core, Solandra is a tight integration of Solr and Cassandra, meaning within a single JVM both Solr and Cassandra are running, and documents are stored and distributed using Cassandra's data model. [321]

Solandra supports most out-of-the-box Solr functionality (search, faceting, highlights), multi-master (read/write to any node). It features replication, sharing, caching, and compaction managed by Cassandra. [322]

11.12.4 Voldemort

According to [638], project Voldemort, developed by LinkedIn, is a non-relational database of key-value type that supports eventual consistency. The distributed nature of the system allows pluggable data placement and provides horizontal scalability and high consistency. Replication and partitioning of data is automatic and performed on multiple servers. Independent nodes that comprise the server support transparent handling of server failure and ensure absence of a central point of failure. Essentially, Voldemort is a hashtable. It uses APIs for data replication. In memory caching allows for faster operations. It allows cluster expansion with no data rebalancing. When Voldemort performance was benchmarked with the other key-value databases such as Cassandra, Redis and HBase as well as MySQL relational database [499], the Voldemort's throughput was twice lower than MySQL and Cassandra and six times higher than HBase. Voldemort was slightly underperforming in comparison with Redis. At the same time, it demonstrated consistent linear performance in maximum throughput that supports high scalability. The read latency for Voldemort was fairly consistent and only slightly underperformed Redis. Similar tendency was observed with the read latency that puts Voldemort in the cluster of databases that require good read-write speed for workload operations. However, the same authors noted that Voldemort required creation of the node specific configuration and optimization in order to successfully run a high throughput tests. The default options were not sufficient and were quickly saturated that stall the database.

11.12.5 Riak

Riak is a set of scalable distributed NoSQL databases developed by Basho Technologies. Riak KV is a key-value [85] database with time-to-live feature so that older data is deleted automatically. It can be queried through secondary indexes, search via Apache Solr, and MapReduce. Riak TS is designed for time-series data. It co-locates related data on the same physical cluster for faster access [86]. Riak S2 is designed to store large objects like media files and software binaries [87]. The databases are available in both open source and commercial versions with multicluster replication provided only in later. REST APIs are available for these databases.

11.12.6 ZHT

According to [581], “ZHT is a zero-hop distributed hash table.” Distributed hash tables effectively break a hash table up and assign different nodes responsibility for managing different pieces of the larger hash table. [700] To retrieve a value in a distributed hash table, one needs to find the node that is responsible for managing the key value pair of interest. [700] In general, every node that is a part of the distributed hash table has a reference to the closest two nodes in the node list. [700] In a ZHT, however, every node contains information concerning the location of every other node. [370] Through this approach, ZHT aims to provide “high availability, good fault tolerance, high throughput, and low latencies, at extreme scales of millions of nodes.” [370] Some of the defining characteristics of ZHT are that it is light-weight, allows nodes to join and leave dynamically, and utilizes replication to obtain fault tolerance among others. [370]

11.12.7 Berkeley DB

Berkeley DB is a family of open source, NoSQL key-value database libraries. [90] It provides a simple function-call API for data access and management over a number of programming languages, including C, C++, Java, Perl, Tcl, Python, and PHP. Berkeley DB is embedded because it links directly into the application and runs in the same address space as the application. [89] As a result, no inter-process communication, either over the network or between processes on the same machine, is required for database operations. It is also extremely portable and scalable, it can manage databases up to 256 terabytes in size.

[455] For data management, Berkeley DB offers advanced services, such as concurrency for many users, ACID transactions, and recovery.

Berkeley DB is used in a wide variety of products and a large number of projects, including gateways from Cisco, Web applications at Amazon.com and open-source projects such as Apache and Linux.

11.12.8 Kyoto/Tokyo Cabinet

Tokyo Cabinet [612] and Kyoto Cabinet [363] are libraries of routines for managing a database. The database normally is a simple data file containing records having a key value pair structure. Every key and value is serial bytes with variable length. Both binary data and character string can be used as a key and a value. There is no concept of data tables nor data types like RDBMS or DBMS. Records are organized in hash table, B+ tree, or fixed-length array. Tokyo and Kyoto cabinets both are developed as a successor of GDBM and QDBM which are library routines for managing database as well. Tokyo Cabinet is written in the C language, and is provided as API of C, Perl, Ruby, Java, and Lua. Tokyo Cabinet is available on platforms which have API conforming to C99 and POSIX. Whereas Kyoto Cabinet is written in the C++ language, and is provided as API of C++, C, Java, Python, Ruby, Perl, and Lua. Kyoto Cabinet is available on platforms which have API conforming to C++03 with the TR1 library extensions. Both are free software licenced under GNU (General Public Licence). [612] actually mentions that Kyoto Cabinet is more powerful and has convenient library structure than Tokyo and recommends people to use Kyoto. Since they use key-value pair concept, you can store a record with a key and a value, delete a record using the key and even retrieve a record using the key. Both have smaller size of database file, faster processing speed and provide effective backup procedures.

11.12.9 Tycoon

Tycoon/ Kyoto Tycoon [366] is a lightweight database server developed by FLL labs and is a distributed Key-value store [574]. It is very useful in handling cache data persistent data of various applications. Kyoto Tycoon is also a package of network interface to the DBM called Kyoto Cabinet [365] which contains a library of routines for managing a database. Tycoon is composed of a sever process that manger multiple databases. This renders high concurrency enabling it to handle more than 10 thousand connections at the same time.

11.12.10 Tyrant

Tyrant provides network interfaces to the database management system called Tokyo Cabinet. Tyrant is also called as Tokyo Tyrant. Tyrant is implemented in C and it provides APIs for Perl, Ruby and C. Tyrant provides high performance and concurrent access to Tokyo Cabinet. The blog [628] explains the results of performance experiments between Tyrant and Memcached + MySQL.

Tyrant was written and maintained by FAL Labs [627]. However, according to FAL Labs, their latest product [626] Kyoto Tycoon is more powerful and convenient server than Tokyo Tyrant.

11.12.11 MongoDB

MongoDB is a NoSQL database which uses collections and documents to store data as opposed to the relational database where data is stored in tables and rows. In MongoDB a collection is a container for documents, whereas a document contains key-value pairs for storing data. As MongoDB is a NoSQL database, it supports dynamic schema design allowing documents to have different fields. The database uses a document storage and data interchange format called BSON, which provides a binary representation of JSON-like documents.

MongoDB provides high data availability by way of replication and sharding. High cost involved in data replication can be reduced by horizontal data scaling by way of shards where data is scattered across multiple servers. It reduces query cost as the query load is distributed across servers. This means that both read and write performance can be increased by adding more shards to a cluster. Which document resides on which shard is determined by the shard key of each collection.

As far as data backup and restore is concerned the default MongoDB storage engines natively support backup of complete data. For incremental backups one can use MongoRocks that is a third party tool developed by Facebook.

11.12.12 Espresso

Espresso [309] is a document-oriented distributed data serving platform that plays an important role in LinkedIn's central data pipeline. It currently powers approximately 30 LinkedIn applications including Member Profile, InMail, etc and also hosts some of its most important member data. Espresso provides a hierarchical data model in which the databases and table schema are defined in JSON. Some of the key components of Espresso include : 1) Router: which is a stateless HTTP Proxy and also acts as an entry point for all client requests in Espresso. The Router uses local cached routing table to manage the partition among all the storage nodes within the cluster. 2) Storage Node: are the building blocks of the storage and each one of them hosts a set of partitions. 3) Helix: is responsible for cluster management in Espresso. 4) Databus: are responsible for capturing changes to transport source transactions in commit order.

All the above mentioned components together enable Espresso to achieve real-time secondary indexing, on-the-fly schema evolution and also a timeline consistent change capture stream.

11.12.13 CouchDB

The Apache Software Foundation makes CouchDB available as an option for those seeking an open-source, NoSQL, document-oriented database. CouchDB, or cluster of unreliable commodity hardware database, [368] stores data as a JSON-formatted document. Documents can consist of a variety of field types, e.g., text, booleans or lists, as well as metadata used by the software. [54] CouchDB does not limit the number of fields per document, and it does not require any two documents to consist of matching or even similar fields. That is, the document has structure, but the structure can vary by document. CouchDB coordinates cluster activities using the master-master mode by default, which means it does not have any one in charge of the cluster. However, a cluster can be set up to write all data to single node, which is then replicated across the cluster. Either way, the system can only offer eventual consistency. [141] CouchDB serves as the basis of Couchbase, Inc's Couchbase Server.

11.12.14 Couchbase Server

Couchbase, Inc. offers Couchbase Server (CBS) to the marketplace as a NoSQL, document-oriented database alternative to traditional relationship-oriented database management systems as well as other NoSQL competitors. The basic storage unit, a *document*, is a “data structure defined as a collection of named fields”. The document utilizes JSON, thereby allowing each document to have its own individual schema. [258]

CBS combines the in-memory capabilities of Membase with CouchDB’s inherent data store reliability and data persistency. Membase functions in RAM only, providing the highest-possible speed capabilities to end users. However, Membase’s in-ram existence limits the amount of data it can use. More importantly, it provides no mechanism for data recovery if the server crashes. Combining Membase with CouchDB provides a persistent data source, mitigating the disadvantages of either product. In addition, CouchDB + membase allows the data size “to grow beyond the size of RAM”. [104]

CBS is written in Erlang/OTP, but generally shortened to just Erlang. In actuality, it is written in “Erlang using components of OTP alongside some C/C++” [179]. It runs on an Erlang virtual machine known as BEAM. [686]

Out-of-the-box benefits of Erlang/OTP include dynamic type setting, pattern matching and, most importantly, actor-model concurrency. As a result, Erlang code virtually eliminates the possibility of inadvertent deadlock scenarios. In addition, Erlang/OTP processes are lightweight, spawning new processes does not consume many resources and message passing between processes is fast since they run in the same memory space. Finally, OTP’s process supervision tree makes Erlang/OTP extremely fault-tolerant. Error handling is indistinguishable from a process startup, easing testing and bug detection. [533]

CouchDB’s design adds another layer of reliability to CBS. CouchDB operates in *append-only* mode, so it adds user changes to the tail of database. This setup resists data corruption while taking a snapshot, even if the server continues to run during the procedure. [341]

Finally, CB uses the Apache 2.0 License, one of several open-source license alternatives. [553]

11.12.15 IBM Cloudant

Cloudant is based on both Apache-backed CouchDB project and the open source BigCouch project. IBM Cloudant is an open source non-relational, distributed database service as service (DBaaS) that provides integrated data management, search and analytics engine designed for web applications. Cloudant’s distributed service is used the same way as standalone CouchDB, with the added advantage of data being redundantly distributed over multiple machines [693].

11.12.16 Pivotal Gemfire (6)

A real-time, consistent access to data-intensive applications is provided by a open source, data management platform named Pivotal Gemfire. “GemFire pools memory, CPU, network resources, and optionally local disk across multiple processes to manage application objects and behavior”. The main features of Gemfire are high scalability, continuous availability, shared nothing disk persistence, heterogeneous data sharing and parallelized application behavior on data stores to name a few. In Gemfire, clients can subscribe to receive notifications to execute their task based on a specific change in data. This is achieved through the continuous querying feature which enables event-driven architecture. The shared nothing architecture of Gemfire suggests that each

node is self-sufficient and independent, which means that if the disk or caches in one node fail the remaining nodes remain untouched. Additionally, the support for multi-site configurations enable the user to scale horizontally between different distributed systems spread over a wide geographical network.

11.12.17 HBase

Apache Hbase is a distributed column-oriented database which is built on top of HDFS (Hadoop Distributed File System). According to [61], It is an open source, versioned, distributed, non-relational database modelled after Google's Bigtable. Similar to Bigtable providing harnessing distributed file storage system offered by Google file system, Apache Hbase provides similar capabilities on top of Hadoop and HDFS. Moreover, Hbase supports random, real-time CRUD (Create/Read/Update/Delete) operations.

Hbase is a type of NoSQL database and is classified as a key value store. In HBase, value is identified with a key where both of them are stored as byte arrays. Values are stored in the order of keys. HBase is a database system where the tables have no schema. Some of the companies that use HBase as their core program are Facebook, Twitter, Adobe, Netflix etc.

11.12.18 Google Bigtable

Google Bigtable is a NoSQL database service, built upon several Google technologies, including Google File System, Chubby Lock Service, and SSTable [235]. Designed for Big Data, Bigtable provides high performance and low latency and scales to hundreds of petabytes [235]. Bigtable powers many core Google products, such as Search, Analytics, Maps, Earth, Gmail, and YouTube. Bigtable also drives Google Cloud Datastore and influenced Spanner, a distributed NewSQL database also developed by Google [673] [680]. Since May 6, 2015, Bigtable has been available to the public as Cloud Bigtable [680].

11.12.19 LevelDB

LevelDB is a light-weight, single-purpose library for persistence with bindings to many platforms. [369] It is a simple open source on-disk key/value data store built by Google, inspired by BigTable and is used in Google Chrome and many other products. It supports arbitrary byte arrays as both keys and values, singular get, put and delete operations, batched put and delete, bi-directional iterators and simple compression using the very fast Snappy algorithm. It is hosted on GitHub under the New BSD License and has been ported to a variety of Unix-based systems, Mac OS X, Windows, and Android. It is not an SQL database and does not support SQL queries. Also, it has no support for indexes. Applications use LevelDB as a library, as it does not provide a server or command-line interface.

11.12.20 Megastore and Spanner

Spanner [136] is Google's distributed database which is used for managing all google services like play, gmail, photos, picasa, app engine etc Spanner is distributed database which spans across multiple clusters, datacenters and geo locations. Spanner is structured in such a way so as to provide non blocking reads, lock free transactions and atomic schema modification. This is unlike other noSql databases which follow the CAP theory i.e. you can choose any two of the three: Consistency, Availability and Partition-tolerance. However, spanner gives an edge by satisfying all three of these. It gives you atomicity and consistency along with availability, partition tolerance

and synchronized replication. Megastore bridges the gaps found in google's bigtable. As google realized that it is difficult to use bigtable where the application requires constantly changing schema. Megastore offers a solution in terms of semi-relational data model. Megastore [103] also provides a transactional database which can scale unlike relational data stores and synchronous replication. Replication in megastore is supported using Paxos. Megastore also provides versioning. However, megastore has a poor write performance and lack of a SQL like query language. Spanners basically adds what was missing in Bigtable and megastore. As a global distributed database spanner provides replication and globally consistent reads and writes. Spanner deployment is called universe which is a collections of zones. These zones are managed by singleton universe master and placement driver. Replication in spanner is supported by Paxos state machine. Spanner was put into evaluation in early 2011 as F1 backend(F1 is Google's advertisement system) which was replacement to mysql. Overall spanner fulfils the needs of relational database along with scaling of noSQL database. All these features make google run all their apps seamlessly on spanner infrastructure.

11.12.21 Accumulo

Apache Accumulo, a highly scalable structured store based on Google's BigTable, is a sorted, distributed key/value store that provides robust, scalable data storage and retrieval. Accumulo is written in Java and operates over the Hadoop Distributed File System (HDFS), which is part of the popular Apache Hadoop project. Accumulo supports efficient storage and retrieval of structured data, including queries for ranges, and provides support for using Accumulo tables as input and output for MapReduce jobs. Accumulo features automatic load-balancing and partitioning, data compression and fine-grained security labels. Much of the work Accumulo does involves maintaining certain properties of the data, such as organization, availability, and integrity, across many commodity-class machines [37].

11.12.22 Cassandra

Apache Cassandra [41] is an open-source distributed database management for handling large volume of data accross commodity servers. It works on asynchronous masterless replication technique leading to low latency and high availability. It is a hybrid between a key-value and column oriented database. A table in cassandra can be viewed as a multi dimensional map indexed by a key. It has its own "Cassandra Query language (CQL)" query language for data extraction and mining. One of the demerits of such structure is it does not support joins or subqueries. It is a java based system which can be administered by any JMX compliant tools.

11.12.23 RYA

Rya is a "scalable system for storing and retrieving RDF data in a cluster of nodes." [490] RDF stands for Resource Description Framework. [490] RDF is a model that facilitates the exchange of data on a network. [261] RDF utilizes a form commonly referred to as a triple, an object that consists of a subject, predicate, and object. [490] These triples are used to describe resources on the Internet. [490] Through new storage and querying techniques, Rya aims to make accessing RDF data fast and easy. [520]

11.12.24 Sqrrl**11.12.25 Neo4J**

Neo4J [665] is a popular ACID compliant graph database management system developed by Neo technology. In this database everything is stored as nodes or edges, both of which can be labeled. Labels help in narrowing and simplifying the search process through the database. [504] It is a highly scalable software and can be distributed across multiple machines. The graph query language that accompanies the software has traversal framework which makes it fast and powerful. [419] The Neo4J is often used for clustering. It offers two feature clustering solutions: Causal Clustering and Highly available clustering. [418] Casual clustering focuses on safety, scalability and causal consistency in the graph. [420] The highly available cluster places importance to fault tolerance as each instance in the cluster has full copies of data in their local database.

11.12.26 graphdb

A Graph Database is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data. [688] The Graph is a concept which directly relates the data items in the store. The data which is present in the store is linked together directly with the help of relationships. It can be retrieved with a single operation. Graph database allow simple and rapid retrieval of complex hierarchical structures that are difficult to model in relational systems.

There are different underlying storage mechanisms used by graph databases. Some graphdb depend on a relational engine and store the graph data in a table, while others use a key-value store or document-oriented database for storage. Thus, they are inherently caled as NoSQL structures. Data retrieval in a graph database requires a different query language other than SQL. Some of the query languages used to retrieve data from a graph database are Gremlin, SPARQL, and Cypher. Graph databases are based on graph theory. They employ the concepts of nodes, edges and properties.

11.12.27 Yarcdata

Yarcdata is Cray subsidiary providing Analytics products, namely the Urika Agile Analytics Platform and Graph Engine. Cray's Urika (Universal RDF Integration Knowledge Appliance) system [142] is a hardware platform designed specifically to provide high-speed graph-retrieval for relationship analytics. Urika is a massively parallel, multi-threaded, shared-memory computing device designed to store and retrieve massive graph datasets. The system can import and host massive heterogeneous graphs represented in the resource description framework (RDF) format and can retrieve descriptive graph patterns specified in a SPARQL query.

Urika-GD [143] is a big data appliance for graph analytics helps enterprises gain key insights by discovering relationships in big data. Its highly scalable, real-time graph analytics warehouse supports ad hoc queries, pattern-based searches, inferencing and deduction. The Urika-GD appliance complements an existing data warehouse or Hadoop® cluster by offloading graph workloads and interoperating within the existing analytics workflow

Cray Graph Engine [517] is a semantic database using Resource Description Framework (RDF) triples to represent the data, SPARQL as the query language and extensions to support mathematical algorithms.

The paper “Graph mining meets the semantic web” [526] outlines the implementation of graph mining algorithms using SPARQL.

11.12.28 AllegroGraph

“AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from their highly complex, distributed data that can’t be answered with conventional databases, i.e., it turns complex data into actionable business insights.” [19] It can be viewed as a closed source database that is used for storage and retrieval of data in the form of triples (triple is a data entity composed of subject-predicate-object like “Professor teaches students”). Information in a triplestore is retrieved using a query language. Query languages can be classified into database query languages or information retrieval query languages. The difference is that a database query language gives exact answers to exact questions, while an information retrieval query language finds documents containing requested information. Triple format represents information in a machine-readable format. Every part of the triple is individually addressable via unique URLs — for example, the statement “Professor teaches students” might be represented in RDF(Resource Description Framework). Using this representation, semantic data can be queried. [20]

11.12.29 Blazegraph

Blazegraph is a graph database also supporting property graph, capable of clustered deployment. A graph database is a NoSQL database. It is based on a graph theory of nodes and edges where each node represents an element such as user or business and each edge represents relationship between two nodes. It is mainly used for storing and analyzing data where maintaining interconnections is essential. Data pertaining to social media is best example where graph database can be used.

Blazegraph’s main focus is large scale complex graph analytics and query. The Blazegraph database runs on graphics processing units (GPU) to speed graph traversals. :cite ‘paper-blzgraph’

Lets now see how Blazegraph handles data. :cite ‘www-blzgraph’ **Blazegraph data can be accessed** using REST APIs.

Blazegraph supports Apache TinkerPop, which is a graph computing framework.

For graph data mining, Blazegraph implements GAS (Gather, Apply, Scatter) model as a service.

11.12.30 Facebook Tao

In the paper published in USENIX annual technical conference, Facebook Inc describes TAO (The Association and Objects) as :cite ‘book-tao’ a geographically distributed data store that provides timely access to the social graph for Facebook’s demanding workload using a fixed set of queries. It is deployed at Facebook for many data types that fit its model. The system runs on thousands of machines, is widely distributed, and provides access to many petabytes of data. TAO represents social data items as Objects (user) and relationship between them as Associations (liked by, friend of). TAO cleanly separates the caching tiers from the persistent data store allowing each of them to be scaled independently. To any user of the system it presents a single unified API that makes the entire system appear like 1 giant graph database. [634].

11.12.31 Titan:db

Titan:db [611] is a distributed graph database that can support of thousands of concurrent users interacting with a single massive graph database that is distributed over the clusters. It is open source with liberal Apache 2 license. Its main components are storage backend, search backend, and TinkerPop graph stack. Titan provides support for various storage backends and also linear

scalability for a growing data and user base. It inherits features such as ‘Gremlin’ query language and ‘Rexter’ graph server from TinkerPop [610]. For huge graphs, Titan uses a component called Titan-hadoop which compiles Gremlin queries to Hadoop MapReduce jobs and runs them on the clusters. Titan is basically optimal for smaller graphs.

11.12.32 Jena

Jena is an open source Java Framework provided by Apache for semantic web applications. ([641]) It provides a programmatic environment for RDF, RDFS and OWL, SPARQL, GRDDL, and includes a rule-based inference engine. Semantic web data differs from conventional web applications in that it supports a web of data instead of the classic web of documents format. The presence of a rule based inference engine enable Jena to perform a reasoning based on OWL and RDFS ontologies. [616] ‘The architecture of Jena contains three layers : Graph layer, model layer and Ontology layer. The graph layer forms the base for the architecture. It does not have an extensive RDF implementation and serves more as a Service provider Interface. According to [616] It provides classes/methods that could be further extended. The model layer extends the graph layer and provides objects of type ‘resource’ instead of ‘node’ to work with. The ontology layer enables one to work with triples.

11.12.33 Sesame

Sesame is framework which can be used for the analysis of RDF (Resource Description Framework) data. Resource Description Framework (RDF) [12] is a model that facilitates the interchange of data on the Web. Using RFD enables us to merge data even if the underlying schemas differ. Sesame has now officially been integrated into RDF4J Eclipse project [13]. Sesame takes in the natively written code as the input and then performs a series of transformations, generating kernels for various platforms. In order to achieve this, it makes use of the feature identifier, impact predictor, source-to-source translator and the auto-tuner [327]. The feature identifier is concerned with the extraction and detection of the architectural features that are important for application performance. The impact predictor determines the performance impact of the core features extracted above. A source-to-source translator transforms the input code into a parametrized one; while the auto-tuner helps find the optimal solution for the processor.

11.12.34 Public Cloud: Azure Table

Microsoft offers its NoSQL Azure Table product to the market as a low-cost, fast and scalable data storage option. [552] Table stores data as collections of key-value combinations, which it terms *properties*. Table refers to a collection of properties as an *entity*. Each entity can contain a mix of properties. The mix of properties can vary between each entity, although each entity may consist of no more than 255 properties. [603]

Although data in Azure Table will be structured via key-value pairs, Table provides just one mechanism for the user to define relationships between entities: the entity’s *primary key*. The primary key, which Microsoft sometimes calls a *clustered index*, consists of a PartitionKey and a RowKey. The PartitionKey indicates the group, a.k.a partition, to which the user assigned the entity. The RowKey indicates the entity’s relative position in the group. Table sorts in ascending order by the PartitionKey first, then by the RowKey using lexical comparisons. As a result, numeric sorting requires fixed-length, zero-padded strings. For instance, Table sorts *111* before *2*, but will sort *111* after *002*. [398]

Azure Table is considered best-suited for infrequently accessed data storage.

11.12.35 Amazon Dynamo

Amazon explains DynamoDB as [[www.dynamodb](http://www.dynamodb.amazonaws.com)] a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications. DynamoDB can be easily integrated with big-data processing tools like Hadoop. It can also be integrated with AWS Lambda, an event driven platform, which enables creating applications that can automatically react to data changes. At present there are certain limits to DynamoDB. Amazon has listed all the limits in a web page titled *Limits in DynamoDB* <http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html>

11.12.36 Google DataStore

Google Cloud Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development [231]. Though Cloud Datastore interface has many of the features similar to traditional databases, but as a NoSQL database, it differs from the SQL in the way as it describes relationships between various data objects. It also provides a number of features that relational databases are not optimally suited to provide, including high-performance at a very large scale and high-reliability. The Google Cloud DataStore can have different kinds of properties for the same kind of entities, unlike the Relational Database where they are represented in rows. For example, the difference between entities can have the properties with the same name but having different values. The flexible schema maps naturally to object-oriented and scripting languages.

Non-relational databases have become popular recently, especially for web applications that require high-scalability and performance with high-availability. Non-relational databases such as Cloud DataStore let developers to choose an optimal balance between strong consistency and eventual consistency for each application. This allows developers to combine the benefits of both the database structures [250]. Datastore is designed to automatically scale to very large data sets, allowing applications to maintain high performance as they receive more traffic. Datastore also provides a number of features that relational databases are not optimally suited to provide, including high-performance at a very large scale and high-reliability [231].

11.13 File management

11.13.1 iRODS

The Integrated Rule-Oriented Data System (iRODS) is open source data management software. iRODS is released as a production-level distribution aimed at deployment in mission critical environments. It virtualizes data storage resources, so users can take control of their data, regardless of where and on what device the data is stored. The development infrastructure supports exhaustive testing on supported platforms. The plugin architecture supports microservices, storage systems, authentication, networking, databases, rule engines, and an extensible API [314]. iRODS implements data virtualization, allowing access to distributed storage assets under a unified namespace, and freeing organizations from getting locked in to single-vendor storage solutions. iRODS enables data discovery using a metadata catalog that describes every file, every directory, and every storage resource in the iRODS Zone. iRODS automates data workflows, with a rule engine that permits any action to be initiated by any trigger on any server or client in the Zone. iRODS enables secure

collaboration, so users only need to log in to their home Zone to access data hosted on a remote Zone. [315]

255. NetCDF

NetCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array oriented scientific data. NetCDF was developed and is maintained at Unidata , part of the University Corporation for Atmospheric Research (UCAR) Community Programs (UCP). Unidata is funded primarily by the National Science Foundation [169] [421] . The purpose of the Network Common Data Form(netCDF) interface is to support the creation, efficient access, and sharing of data in a form that is self-describing, portable, compact, extendible, and archivable Version 3 of netCDF is widely used in atmospheric and ocean sciences due to its simplicity. NetCDF version 4 has been designed to address limitations of netCDF version 3 while preserving useful forms of compatibility with existing application software and data archives [169]. NetCDF consists of: a) A conceptual data model b) A set of binary data formats c) A set of APIs for C/Fortran/Java

11.13.2 CDF

Common Data Format [415] is a conceptual data abstraction for storing, manipulating, and accessing multidimensional data sets. CDF differs from traditional physical file formats by defining form and function as opposed to a specification of the bits and bytes in an actual physical format.

CDF's integrated dataset is composed by following two categories :(a)Data Objects - scalars, vectors, and n-dimensional arrays.(b)Metadata - set of attributes describing the CDF in global terms or specifically for a single variable [414].

The self-describing property (metadata) allows CDF to be a generic, data-independent format that can store data from a wide variety of disciplines. Hence, the application developer remains insulated from the actual physical file format for reasons of conceptual simplicity, device independence, and future expandability.CDF data sets are portable on any of the CDF-supported platforms and accessible with CDF applications or layered tools. To ensure the data integrity in a CDF file, checksum method using MD5 algorithm is employed [157].

Compared to HDF format [690], CDF permitted cross-linking data from different instruments and spacecraft in ISTP with one development effort. CDF is widely supported by commercial and open source data analysis/visualization software such as IDL, MATLAB, and IBM's Data Explorer (XP).

11.13.3 HDF

11.13.4 OPeNDAP

11.13.5 FITS

FITS stand for 'Flexible Image Transport System'. It is a standard data format used in astronomy. FITS data format is endorsed by NASA and International Astronomical Union. According to [190], FITS can be used for transport, analysis and archival storage of scientific datasets and support multi-dimensional arrays, tables and headers sections. FITS is actively used and developed - according to [191] newer version of FITS standard document was released in July 2016. FITS can be used for digitization of contents like books and magazines. Vatican Library [188] used FITS for long term preservation of their book, manuscripts and other collection. Matlab, a language used for technical computing supports fits [189]. The 2011 paper [340] explains how to perform processing of astronomical images on Hadoop using FITS.

11.13.6 RCFfile

RCFile (Record Columnar File) [506] is a big data placement data structure that supports fast data loading and query processing coupled with efficient storage space utilization and adaptive to dynamic workload environments. It is designed for data warehousing systems that uses map-reduce. The data is stored as a flat file comprising of binary key/value pairs. The rows are partitioned first and then the columns are partitioned in each row and the respective meta-data for each row is stored in the key part for that row and the values comprises of the data part of the row. Storing the data in this format enables RCFile to accomplish fast loading and query processing. A shell utility is available for reading RCFile data and metadata [505]. According to [274], RCFile has been chosen in Facebook data warehouse system as the default option. It has also been adopted by Hive and Pig, the two most widely used data analysis systems developed in Facebook and Yahoo!

11.13.7 ORC

ORC files were created as part of the initiative to massively speed up Apache Hive and improve the storage efficiency of data stored in Apache Hadoop. ORC is a self-describing type-aware columnar file format designed for Hadoop workloads. It is optimized for large streaming reads, but with integrated support for finding required rows quickly. Storing data in a columnar format lets the reader read, decompress, and process only the values that are required for the current query. Because ORC files are type-aware, the writer chooses the most appropriate encoding for the type and builds an internal index as the file is written. ORC files are divided into stripes that are roughly 64MB by default. The stripes in a file are independent of each other and form the natural unit of distributed work. Within each stripe, the columns are separated from each other so the reader can read just the columns that are required [84].

11.13.8 Parquet

Apache parquet is the column Oriented data store for Apache Hadoop ecosystem and available in any data processing framework, data model or programming language [199]. It stores data such that the values in each column are physically stored in contiguous memory locations. As it has the columnar storage, it provides efficient data compression and encoding schemes which saves storage space as the queries that fetch specific column values need not read the entire row data and thus improving performance. It can be implemented using the Apache Thrift framework which increases its flexibility to work with a number of programming languages like C++, Java, Python, PHP, etc.

11.14 Data Transport

11.14.1 BitTorrent

BitTorrent is P2P communication protocol commonly used for sending and receiving the large digital files like movies and audioclips. In order to upload and download file, user have to download bittorrent client which implement the bittorrent protocol. BitTorrent uses the principle of swarming and tracking. [99] It divides the files in large number of chunk and as soon as file is received it can be served to the other users for downloading. So rather than downloading one entire large file from one source, user can download small chunk from the different sources of linked users in swarm. BitTorrent trackers keeps list of files available for transfer and helps the swarm user find each other.

Using the protocol, machine with less configuration can serve as server for distributing the files. It result in increase in the downloading speed and reduction in origin server configuration.

Few popular bittorrent client in μ Torrent, qBittorrent.

11.14.2 HTTP

11.14.3 FTP

According to [208] FTP is an acronym for File Transfer Protocol. It is network protocol standard used for transferring files between two computer systems or between a client and a server. It is part of the Application layer of the Internet Protocol Suite and works along with HTTP/SSH. It follows a client-server model architecture. Secure systems asks the client to authenticate themselves using a Username and Password registered with the server to access the files via FTP. The specification for FTP was first written by Abhay Bhushan [[www-rfc114](#)] in 1971 and is termed as RFC114. The current specification, RFC959 in use was written in 1985. Several other versions of the specification are available which provides firewall friendly FTP access, additional security extensions, support for IPV6 and passive mode file access respectively. FTP can be used in command line in most of the operating systems to transfer files. There are FTP clients such as WinSCP, FileZilla etc. which provides a graphical user interface to the clients to authenticate themselves (sign on) and access the files from the server.

11.14.4 SSH

SSH is a cryptographic network protocol [568] to provide a secure channel between two clients over an unsecured network. It uses public-key cryptography for authenticating the remote machine and the user. The public-private key pairs could be generated automatically to encrypt the network connection. ssh-keygen utility could be used to generate the keys manually. The public key then could be placed on the all the computers to which the access is required by the owner of the private key. SSH runs on the client-server model where a server listens for incoming ssh connection requests. It's generally used for remote login and command execution. It's other important uses include tunneling(required in cloud computing) and file transfer(SFTP). OpenSSH is an open source implementation of network utilities based on SSH [569].

11.14.5 Globus Online (GridFTP)

GridFTP is a enhancement on the File Tranfer Protocol (FTP) which provides high-performance , secure and reliable data transfer for high-bandwidth wide-area networks. As noted in [259] the most widely used implementation of GridFTP is Globus Online. GridFTP achieves efficient use of bandwidth by using multiple simultaneous TCP streams. Files can be downloaded in pieces simultaneously from multiple sources; or even in separate parallel streams from the same source. GridFTP allows transfers to be restarted automatically and handles network unavailability with a fault tolerant implementation of FTP. The underlying TCP connection in FTP has numerous settings such as window size and buffer size. GridFTP allows automatic (or manual) negotiation of these settings to provide optimal transfer speeds and reliability .

11.14.6 Flume

Flume is distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of log data [589]. Flume was created to allow you to flow data from a source into your Hadoop® environment. In Flume, the entities you work with are called sources, decorators, and sinks. A source can be any data source, and Flume has many predefined source adapters. A sink is the target of a specific operation. A decorator is an operation on the stream that can transform the

stream in some manner, which could be to compress or uncompress data, modify data by adding or removing pieces of information, and more [292].

11.14.7 Sqoop

Apache Sqoop is a tool to transfer large amounts of data between Apache Hadoop and sql databases [587]. The name is a Portmanteau of SQL + Hadoop. It is a command line interface application which supports incremental loads of complete tables, free form (custom) SQL Queries and allows the use of saved and scheduled jobs to import latest updates made since the last import. The imports can also be used to populate tables in Hive or Hbase. Sqoop has the option of export, which allows data to be transferred from Hadoop into a relational database. Sqoop is supported in many different business integration suits like Informatica Big Data Management, Pentaho Data Integration, Microsoft BI Suite and Couchbase [658].

11.14.8 Pivotal GPOLOAD/GPFDIST

Greenplum Database [232] is a shared nothing, massively parallel processing solution built to support next generation data warehousing and Big Data analytics processing. In its new distribution under Pivotal, Greenplum Database is called Pivotal(Greenplum) Database.

gpfdist [255] is Greenplum's parallel file distribution program. It is used by readable external tables and gpload to serve external table files to all Greenplum Database segments in parallel. It is used by writable external tables to accept output streams from Greenplum Database segments in parallel and write them out to a file.

gupload [232] is data loading utility is used to load data into Greenplum's external table in parallel.

Google has an invention [130] relating to integrating map-reduce processing techniques into a distributed relational database. An embodiment of the invention is implemented by Greenplum as gpfdist.

11.15 Cluster Resource Management

11.15.1 Mesos

Apache Mesos [394] abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems to easily be built and run effectively. The Mesos kernel runs on every machine and provides applications (e.g., Hadoop, Spark, Kafka, Elasticsearch) with API's for resource management and scheduling across entire datacenter and cloud environments.

The resource scheduler of Mesos supports a generalization of max-min fairness [7], termed Dominant Resource Fairness (DRF) [224] scheduling discipline, which allows to harmonize execution of heterogeneous workloads (in terms of resource demand) by maximizing the share of any resource allocated to a specific framework.

Mesos uses containers for resource isolation between processes. In the context of Mesos, the two most important resource-isolation methods to know about are the control groups (cgroups) built into the Linux kernel, and Docker. The difference between using hyper-V, Docker containers, cgroup is described in detail in the book "Mesos in action" [301]

11.15.2 Yarn

Yarn (Yet Another Resource Negotiator) is Apache Hadoop's cluster management project [127]. It's a resource management technology which make a pace between, the way applications use Hadoop system resources & node manager agents. Yarn, "split up the functionalities of resource management and job scheduling/monitoring". The NodeManager watch the resource (cpu, memory, disk, network) usage the container and report the same to ResourceManager. ResourceManager will take a decision on allocation of resources to the applications. ApplicationMaster is a library specific to application, which requests/negotiate resources from ResourceManager and launch and monitoring the task with NodeManager(s) [570]. ResourceManager have two majors: Scheduler and ApplicationManager. Scheduler have a task to schedule the resources required by the application. ApplicationManger holds the record of application who require resource. It validates (whether to allocate the resource or not) the application's resource requirement and ensure that no other application already have register for the same resource requirement. Also it keeps the track of release of resource. [265]

11.15.3 Helix

Helix is a data management system getting developed by IBM which helps the users to do exploratory analysis of the data received from various sources following different formats. This system would help orgnaize the data by providing links between data collected across various sources despite of the knowledge of the data sources schemas. It also aims at providing the data really required for the user by extracting the important information from the data. This would plan to target the issue by mainataining the "knowledge base of schemas" and "context-dependent dynamic linkage", The system can get the schema details either from the knowledge base being maintained or can even get the schema from the data being received. As the number of users for helix increases the linkages gets stronger and would provide better data quality. [182]

11.15.4 Llama

Llama stands for leveraging learning to automatically manage algorithms. There has been a phenomenal improvement in algorithm portfolio and selection approaches. The main drawback of them is that their implementation is specific to a problem domain and customized which leads to the difficulty of exploring new techniques for certain problem domains. Llama has been developed to provide an extensible toolkit which can initiate exploration of a variety of portfolio techniques over a wide range of problem domains. It is modular and implemented as an R package. It leverages the extensive library of machine learning algorithms and techniques in R [356]. Llama can be regarded as a framework which provides the prerequisites for initiating automatic portfolio selectors. It provides a set of methods for combining several trivial approaches of portfolio selection into sophisticated techniques. The primary reason behind the introduction of Llama was to help the researchers working in algorithm selection, algorithm portfolios, etc. and can be just used as a tool for designing the systems [356].

11.15.5 Google Omega

11.15.6 Facebook Corona

Corona is a new scheduling framework developed by facebook which separates the cluster resource management from job coordination. Facebook, employed the MapReduce implementation from Apache Hadoop since 2011 for job scheduling. The scheduling MapReduce framework has its

limitations with the scalability as when the number of jobs at facebook grew in the next few years. Another limitation of Hadoop was it was a pull-based scheduling model as the task tracker have to provide a heartbeat to the job tracker to indicate that it is running which associated with a pre-defined delay, that was problematic for small jobs [183]. Hadoop MapReduce is also constrained by its static slot-based resource management model where a MapReduce cluster is divided into a fixed number of map and reduce slots based on a static configurations so the slots are not utilized completely anytime the cluster workload does not fit the static configuration.

Corona improves over the Hadoop MapReduce by introducing a cluster manager whose only purpose is to track the nodes in the cluster and the amount free resources [183]. A dedicated job tracker is created for each job and can run either in the same process as the client (for small jobs) or as a separate process in the cluster (for large jobs). The other difference is that it uses a push-based scheduling whose implementation does not involve a periodic heartbeat and thus scheduling latency is minimized. The cluster manager also implements a fair-share scheduling as it has access to the full snapshot of the cluster for making the scheduling decisions. Corona is used as an integral part of the Facebook's data infrastructure and is helping power big data analytics for teams across the company.

11.15.7 Celery

“Celery is an asynchronous task queue/job queue based on distributed message passing. The focus of celery is mostly on real-time operation, but it equally scheduling. In celery there are execution units, called tasks, are executed concurrently on a single or more worker servers using multiprocessing, Eventlet,or gevent. Tasks can execute asynchronously (in the background) or synchronously (wait until ready). Celery is easy to integrate with web framework. Celery is written in python whereas the protocol can be implemented in any language” [111]. “Celery is a simple, flexible, and reliable distributed system to process vast amounts of messages,while providing operations with the tools required to maintain such a system”[112]

11.15.8 HTCondor

HTCondor is a specialized workload management system for compute-intensive jobs. HTCondor provides various features like a)job queuing mechanism, b)scheduling policy, c)resource monitoring, d)priority scheme and e)resource management just as other full-featured batch systems. “Users submit their serial or parallel jobs to HTCondor,HTCondor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion”. HTCondor can be used to manage a cluster of dedicated compute nodes. HTCondor uses unique mechanisms to harness wasted CPU power from idle deskto workstations. “The ClassAd mechanism in HTCondor provides an extremely flexible and expressive framework for matching resource requests (jobs) with resource offers (machines). Jobs can easily state both job requirements and job preferences”. “HTCondor incorporates many of the emerging Grid and Cloud-based computing methodologies and protocols” [650]

11.15.9 SGE

According to [452], Sun Grid Engine (SGE) renamed to Oracle Grid Engine (OGE) is a grid computing cluster software system. Grid Engine is a high performance computing cluster used for managing job queueing in distributed and parallel environment. It can accept, schedule, dispatch and manage the execution of single, parallel user jobs in a remote or distributed manner. It also manages the resource allocation to those jobs. The resources can be anything like processors,

storage, RAM and licenses for softwares. The latest stable release of OGE is termed as 6.2u8 which came out in October 1,2012.

OGE supports a vast array of features like: Topology-aware scheduling and thread binding, advanced fault tolerance mechanisms for job scheduling, web interface based status reporting and ability to use different scheduling algorithms,etc. OGE runs on several platforms including AIX, BSD, Linux, Solaris, OS X, Tru64, Windows, etc. It is under deployment phases for IBM's 64-bit operating system z/OS. Standard Grid cluster comprises of one master host and many execution hosts. There is an option of creating shadow master hosts which would take the master's place incase of a system crash. Notable deployments of OGE include: TSUBAME supercomputer at the Tokyo Institute of Technology,Ranger at the Texas Advanced Computing Center (TACC) and San Diego Supercomputer Center (SDSC).

11.15.10 OpenPBS

Portable Batch System (or simply PBS) is the name of computer software that performs job scheduling. Its primary task is to allocate computational tasks, i.e., batch jobs, among the available computing resources. It is often used in conjunction with UNIX cluster environments [659]. OpenPBS is the original open source version of PBS. There are more commercialized versions of the same software. One of the key feature of OpenPBS is that it supports millions of cores with fast job dispatch and minimal latency. It meets unique site goals and SLAs by balancing job turnaround time and utilization with optimal job placement. OpenPBS also includes automatic fail-over architecture with no single point of failure – jobs are never lost, and jobs continue to run despite failures. It is built upon a Flexible Plugin Framework which simplifies administration with enhanced visibility and extensibility [464].

11.15.11 Moab

Moab HPC Suite is a workload management and resource orchestration platform that automates the scheduling, managing, monitoring, and reporting of HPC workloads on massive scale. It uses multi-dimensional policies and advanced future modeling to optimize workload start and run times on diverse resources. It integrates and accelerates the workloads management across independent clusters by adding grid-optimized job submission. Moab's unique intelligent and predictive capabilities evaluate the impact of future orchestration decisions across diverse workload domains (HPC, HTC, Big Data, and Cloud VMs)[403].

11.15.12 Slurm (551)

Simple Linux Utility for Resource Management (SLURM) workload manager is an open source, scalable cluster resource management tool used for job scheduling in small to large Linux cluster using multi-core architecture. As per, [530] SLURM has three key functions. First, it allocates resources to users for some duration with exclusive and/or non-exclusive access. Second, it enables users to start, execute and monitor jobs on the resources allocated to them. Finally, it intermediates to resolve conflicts on resources for pending work by maintaining them in a queue. The slurm architecture has following components: a centralized manager to monitor resources and work, may have a backup manager, daemon on each server to provide fault-tolerant communications, an optional daemon for clusters with multiple managers and tools to initiate, terminate and report about jobs in a graphical view with network topology. It also provides around twenty additional plugins that could be used for functionalities like accounting, advanced reservation, gang scheduling, back fill scheduling and multifactor job prioritization. Though originally developed for Linux, SLURM

also provides full support on platforms like AIX, FreeBSD, NetBSD and Solaris [531].

11.15.13 Torque

11.15.14 Globus Tools

[558] The Globus Toolkit is an open source toolkit organized as a collection of loosely coupled components. These components consist of services, programming libraries and development tools designed for building Grid-based applications. GT components fall into five broad domain areas: Security, Data Management, Execution Management, Information Services, and Common Runtime. [197] These components enable a broader "Globus ecosystem" of tools and components that build on or interoperate with GT functionality to provide a wide range of useful application-level functions. www-about-globus [230] Since 2000, companies like Fujitsu, IBM, NEC and Oracle have pursued Grid strategies based on the Globus Toolkit.

11.15.15 Pilot Jobs

In pilot job, an application acquires a resource so that it can be delegated some work directly by the application; instead of requiring some job scheduler. The issue of using a job scheduler is that a waiting queue is required. Few examples of Pilot Jobs are the [501] Falkon lightweight framework and [352] HTCaaS. Pilot jobs are typically associated with both Parallel computing as well as Distributed computing. Their main aim is to reduce the dependency on queues and the associated multiple wait times.

Using pilot jobs enables us to have a multilevel technique for the execution of various workloads. This is so because the jobs are typically acquired by a placeholder job and they relayed to the workloads [618].

11.16 File systems

11.16.1 HDFS

Hadoop provides distributed file system framework that uses Map reduce (Distributed computation framework) for transformation and analyses of large dataset. Its main work is to partition the data and other computational tasks to be performed on that data across several clusters. HDFS is the component for distributed file system in Hadoop. An HDFS cluster primarily consists of a Name Node and Data Nodes. Name Node manages the file system metadata such as access permission, modification time, location of data and Data Nodes store the actual data. When user applications or Hadoop frameworks request access to a file in HDFS, Name Node service responds with the Data Node locations for the respective individual data blocks that constitute the whole of the requested file[273].

11.16.2 Swift

11.16.3 Haystack

Haystack is an open source project working with data from internet of Things, aim to standardise the semantic data model generated from smart devices, homes, factories etc. It include automation, control, energy, HVAC, lighting and other environmental systems. [271]

Building block of Project haystack is on TagModel tagging of metadata stored in key/value pair

applied to entity such id, dis, sites, geoAddr, tz. Structure the primary structure of haystack is based on three entities, Site location of single unit, equip physical or logical piece of equipment within site, point sensor, actuator or setpoint value for equip, it also includes weather outside weather condition. TimeZone time series data is most important factor it is foundation for sensor and operational data. Captured data not always associated with measurable unit, however it provides facility to associate the data points. Commonly Supported units like Misc, Area, Currency, Energy, Power, Temperature, Temperature differential, Time, Volumetric Flow. The data often represented in 2D tabular form for tagged entities. It supports the query language for filtering over the data, data exposed through REST API in JSON format.

11.16.4 f4

As the amount of data Facebook stores continues to increase, the need for quick access and efficient storage of data continues to rise. Facebook stores a class of data in Binary Large OBjects (BLOBs), which can be created once, read many times, never modified, and sometimes deleted. Haystack, Facebook's traditional BLOB storage system is becoming increasingly inefficient. The storage efficiency is measured in the effective-replication-factor of BLOBs.

f4 BLOB storage system provides an effective-replication-factor lower than that of Haystack. f4 is simple, modular, scalable, and fault tolerant. f4 currently stores over 65PBs of logical BLOBs, with a reduced effective-replication-factor from 3.6 to either 2.8 or 2.1 [409].

11.16.5 Cinder

"Cinder is a block storage service for Openstack" [118]. Openstack Compute uses ephemeral disks meaning that they exist only for the life of the Openstack instance i.e. when the instance is terminated the disks disappear. Block storage system is a type of persistent storage that can be used to persist data beyond the life of the instance. Cinder provides users with access to persistent block-level storage devices. It is designed such that users can create block storage devices on demand and attach them to any running instances of OpenStack Compute [500]. This is achieved through the use of either a reference implementation(LVM) or plugin drivers for other storage. Cinder virtualizes the management of block storage devices and provides end users with a self-service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device [118].

11.16.6 Ceph

Ceph is open-source storage platform providing highly scalable object, block as well as file-based storage. Ceph is a unified, distributed storage system designed for excellent performance, reliability and scalability [514]. Ceph Storage clusters are designed to run using an algorithm called CRUSH (Controlled Replication Under Scalable Hashing) which replicates and re-balance data within the cluster dynamically to ensure even data distribution across cluster and quick data retrieval without any centralized bottlenecks.

Ceph's foundation is the Reliable Autonomic Distributed Object Store (RADOS) [516], which provides applications with object, block, and file system storage in a single unified storage cluster—making Ceph flexible, highly reliable and easy to manage. Ceph decouples data and metadata operations by eliminating file allocation tables and replacing them with generating functions which allows RADOS to leverage intelligent OSDs to manage data replication, failure detection and recovery, low-level disk allocation, scheduling, and data migration without encumbering any central

server(s) [644].

The Ceph Filesystem [515] is a POSIX-compliant filesystem that uses a Ceph Storage Cluster to store its data. Ceph's dynamic subtree partitioning is a uniquely scalable approach, offering both efficiency and the ability to adapt to varying workloads. Ceph Object Storage supports two compatible interfaces: Amazon S3 and Openstack Swift.

11.16.7 FUSE

FUSE (Filesystem in Userspace) [209] "is an interface for userspace programs to export a filesystem to the Linux kernel". The FUSE project consists of two components: the fuse kernel module and the libfuse userspace library. libfuse provides the reference implementation for communicating with the FUSE kernel module. The code for FUSE itself is in the kernel, but the filesystem is in userspace. As per the 2006 paper [730] on HPTFS which has been built on top of FUSE. It mounts a tape as normal file system based data storage and provides file system interfaces directly to the application. Another implementation of FUSE FS is CloudBB [725]. Unlike conventional filesystems CloudBB creates an on-demand two-level hierarchical storage system and caches popular files to accelerate I/O performance. On evaluating performance of real data-intensive HPC applications in Amazon EC2/S3, results show CloudBB improves performance by up to 28.7 times while reducing cost by up to 94.7% compared to the ones without CloudBB.

Some more implementation examples of FUSE are - mp3fs (A VFS to convert FLAC files to MP3 files instantly), Copy-FUSE(To access cloud storage on Copy.com), mtpfs(To mount MTP devices) etc.

11.16.8 Gluster

11.16.9 Lustre

The Lustre file system [448] is an open-source, parallel file system that supports many requirements of leadership class HPC simulation environments and Enterprise environments worldwide. Because Lustre file systems have high performance capabilities and open licensing, it is often used in supercomputers. Lustre file systems are scalable and can be part of multiple computer clusters with tens of thousands of client nodes, tens of petabytes of storage on hundreds of servers, and more than a terabyte per second of aggregate I/O throughput. Lustre file systems a popular choice for businesses with large data centers, including those in industries such as meteorology, simulation, oil and gas, life science, rich media, and finance. Lustre provides a POSIX compliant interface and many of the largest and most powerful supercomputers on Earth today are powered by the Lustre file system.

11.16.10 GPFS

IBM General Parallel File System (GPFS) was rebranded to IBM Spectrum Scale on February 17, 2015 [694]. See 380.

11.16.11 IBM Spectrum Scale

General Parallel File System (GPFS) was rebranded as IBM Spectrum Scale on February 17, 2015 [694].

Spectrum Scale is a clustered file system, developed by IBM, designed for high performance. It

"provides concurrent high-speed file access to applications executing on multiple nodes of clusters" [694] and can be deployed in either shared-nothing or shared disk modes. Spectrum Scale is available on AIX, Linux, Windows Server, and IBM System Cluster 1350 [694]. Due to its focus on performance and scalability, Spectrum Scale has been utilized in compute clusters, big data and analytics - including support for Hadoop Distributed File System (HDFS), backups and restores, and private clouds [291].

11.16.12 GFFS

The Global Federated File System (GFFS) [478] is a computing technology that allows linking of data from Windows, Mac OS X, Linux, AFS, and Lustre file systems into a global namespace, making them available to multiple systems. It is a federated, secure, standardized, scalable, and transparent mechanism to access and share resources across organizational boundaries. It is useful when, for data resources, boundaries do not require application modification and do not disrupt existing data access patterns. It uses FUSE to handle access control and allows research collaborators on remote systems to access a shared file system. Existing applications can access resources anywhere in the GFFS without modification. It helps in rapid development of code, which can then be exported via GFFS and implemented in-place on a given computational resource or Science Gateway.

11.16.13 Public Cloud: Amazon S3

Amazon Simple Storage Service (Amazon S3) [25] is storage object which provides a simple web service interface to store and retrieve any amount of data from anywhere on the web. With Amazon S3, users can store as much data as they want and can scale it up and down based on the requirements. For developers Amazon S3 provides full REST API's and SDK's which can be integrated with third-party technologies. Amazon S3 is also deeply integrated with other AWS services to make it easier to build solutions that use a range of AWS services which include Amazon CloudFront, Amazon CloudWatch, Amazon Kinesis, Amazon RDS, Amazon Glacier etc. Amazon S3 provides automatic encryption of data once the data is uploaded in the cloud. Amazon S3 uses the concept of Buckets and Objects for storing data wherein Buckets are used to store objects. Amazon S3 services can be used using the Amazon Console Management. [632] The steps for using the Amazon S3 are as follows: (1) Sign up for Amazon S3 (2) After sign up, create a Bucket in your account, (3) Create an object which might be a file or folder, and (4) Perform operations on the object which is stored in the cloud.

11.16.14 Azure Blob

Azure Blob storage is a service that stores unstructured data in the cloud as objects/blobs. Blob storage can store any type of text or binary data, such as a document, media file, or application installer [33]. Blob storage is also referred to as object storage. The word 'Blob' expands to Binary Large OBject. There are three types of blobs in the service offered by Windows Azure namely block, append and page blobs. [223] 1. Block blobs are collection of individual blocks with unique block ID. The block blobs allow the users to upload large amount of data. 2. Append blobs are optimized blocks that helps in making the operations efficient. 3. Page blobs are compilation of pages. They allow random read and write operations. While creating a blob, if the type is not specified they are set to block type by default. All the blobs must be inside a container in your storage. Azure Blob storage is a service for storing large amounts of unstructured object data, such as text or binary data, that can be accessed from anywhere in the world via HTTP or HTTPS. You

can use Blob storage to expose data publicly to the world, or to store application data privately. Common uses of Blob storage include serving images or documents directly to a browser, storing files for distributed access, streaming video and audio, storing data for backup and restore, disaster recovery, and archiving and storing data for analysis by an on-premises or Azure-hosted service. Azure Storage is massively scalable and elastic with an auto-partitioning system that automatically load-balances your data. Blob storage is a specialized storage account for storing your unstructured data as blobs (objects) in Azure Storage. Blob storage is similar to existing general-purpose storage accounts and shares all the great durability, availability, scalability, and performance features. Blob storage has two types of access tiers that can be specified, hot access tier, which will be accessed more frequently, and a cool access tier, which will be less frequently accessed. There are many reasons why you should consider using BLOB storage. Perhaps you want to share files with clients, or off-load some of the static content from your web servers to reduce the load on them. [33]

11.16.15 Google Cloud Storage

Google Cloud Storage is the cloud enabled storage offered by Google. [247] It is unified object storage. To have high availability and performance among different regions in the geo-redundant storage offering. If you want high availability and redundancy with a single region one can go for "Regional" storage. Nearline and Coldline' are the different archival storage techniques. "Nearline" storage offering is for the archived data which the user access less than once a month . "Coldline" storage is the storage which is used for the data which is touched less than once a year.

All the data in Google Cloud storage belongs inside a project. A project will contain different buckets. Each bucket has different objects. We need to make sure that the name of the bucket is unique across all Google cloud name space . And the name of the objects should unique in a bucket.

11.17 Interoperability

11.17.1 cloudmesh

11.17.2 Libvirt

Libvirt is an open source API to manage hardware virtualization developed by Red Hat. It is a standard C library but has accessibility from other languages such as Python, Perl, Java and others. [372] Multiple virtual machine monitors(VMM) or hypervisors are supported such as KVM,QEMU, Xen, Virtuozzo, VMWare ESX, LXC, and BHyve. It can be divided into five categories such as hypervisor connection, domain, network, storage volume and pool. [333] It is accessible by many operating systems such as Linux, FreeBSD, Mac OS, and Windows OS.

11.17.3 Libcloud

:cite::‘www-libcloudwiki‘ Libcloud is a python library that allows to interact with several popular cloud service providers. It is primarily designed to ease development of software products that work with one or more cloud services supported by Libcloud. It provides a unified API to interact with these different cloud services. Current API includes methods for list, reboot, create, destroy, list images and list sizes. :cite::‘www-libclouddoc‘ lists Libcloud key component APIs Compute, Storage, Load Balancers, DNS, Container and Backup. Compute API allows users to manage cloud servers. Storage API allows users to manage cloud object storage and also provides CDN management functionality. Load balancer, DNS and Backup API’s allows users to manage their respective functionalities, as services, and related products of different cloud service providers.

Container API allows users to deploy containers on to container virtualization platforms. Libcloud supports Python 2, Python 3 and PyPy.

11.17.4 JClouds

[91] Primary goals of cross-platform cloud APIs is that application built using these APIs can be seamlessly ported to different cloud providers. The APIs also bring interoperability such that cloud platforms can communicate and exchange information using these common or shared interfaces. Jclouds or apache jclouds [324] is a java based library to provide seamless access to cloud platforms. Jclouds library provides interfaces for most of cloud providers like docker, openstack, amazon web services, microsoft azure, google cloud engine etc. It will allow users build applications which can be portable across different cloud environments. Key components of jcloud are:

1. Views: abstracts functionality from a specific vendor and allow user to write more generic code. For example odbc abstracts the underlying relational data source. However, odbc driver converts to native format. In this case user can switch databases without rewriting the application. Jcloud provide following views: blob store, compute service, loadBalancer service
2. API: APIs are requests to execute a particular functionality. Jcloud provide a single set of APIs for all cloud vendors which is also location aware. If a cloud vendor doesn't support customers from a particular region the API will not work from that region.
3. Provider: a particular cloud vendor is a provider. Jcloud uses provider information to initialize its context.
4. Context: it can be termed as a handle to a particular provider. Its like a ODBC connection object. Once connection is initialized for a particular database, it can used to make any api call.

Jclouds provides test library to mock context, APIs etc to different providers so that user can write unit test for his implementation rather than waiting to test with the cloud provider. Jcloud library certifies support after testing the interfaces with live cloud provider. These features make jclouds robust and adoptable, hiding most of the complexity of cloud providers.

11.17.5 TOSCA

11.17.6 OCCI

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API that provides specifications and remote management for the development of "interoperable tools" [711]. It supports IaaS, PaaS and SaaS and focuses on integration, portability, interoperability, innovation and extensibility. It provides a set of documents that describe an OCCI Core model, contain best practices of interaction with the model, combined into OCCI Protocols, explain methods of communication between components via HTTP protocol introduced in the OCCI Renderings, and define infrastructure for IaaS presented in the OCCI Extensions.

The current version 1.2 OCCI consists of seven documents that identify required and optional components. Of the Core Model. In particular, the following components are required to implement: a)Core Model, b)HTTP protocol, c)Text rendering and d)JSON rendering. Meanwhile, Infrastructure, Platform and SLA models are optional. The OCCI Core model defines instance types and

provides a layer of abstraction that allows the OCCI client to interact with the model without knowing of its potential structural changes. The model supports extensibility via inheritance and using mixin types that represent ability to add new components and capabilities at run-time. [712]

The OCCI Protocol defines the common set of names provided for the IaaS cloud services user that specify requested system requirements. It is often denoted as "resource templates" or "flavours" [715].

OCCI RESTful HTTP Protocol describes communications between server and client on OCCI platform via HTTP protocol [713]. It defines a minimum set of HTTP headers and status codes to ensure compliance with the OCCI Protocol. Separate requirements for Server and Client for versioning need to be implemented using HTTP 'Server' header and 'User-Agent' header respectively.

JSON rendering [714] protocol provides JSON specifications to allow "render OCCI instances independently of the protocol being used." In addition, it provides details of the JSON object declaration, OCCI Action Invocation, object members required for OCCI Link Instance Rendering, "location maps to OCCI Core's source and target model attributes and kind maps to OCCI Core's target" to satisfy OCCI Link Instance Source/Target Rendering requirements. Finally, it specifies various attributes and collection rendering requirements. The text rendering process is deprecated and will be removed from the next major version [716].

11.17.7 CDMI

The Storage Networking Industry Association (SNIA) [71] is a non-profit organization formed by various companies, suppliers and consumers of data storage and network products. SNIA defines various standards to ensure the quality and interoperability of various storage systems. One of the standards defined by SNIA for providers and users of cloud is Cloud Data Management Interface (CDMI). According latest issue of CDMI [72], "CDMI International Standard is intended for application developers who are implementing or using cloud storage. It documents how to access cloud storage and to manage the data stored there." It defines functional interface for applications that will use cloud for various functionalities like create, retrieve, update and delete data elements from the cloud. These interface could be used to manage containers along with the data. The interface could be used by administrative and management applications as well. Also, the CDMI specification uses RESTful principles in the interface design. All the standards issued on CDMI can be found on SNIA web page [73].

11.17.8 Whirr

Apache Whirr is a set of libraries for running cloud services, which provides a cloud-neutral way to run services [653]. This is achieved by using cloud-neutral provisioning and storage libraries such as jclouds and libcloud. Whirr's API should be built on top these libraries and is not exposed to the users. It is also a common service API, in which the details of its working are, particular to the service. Whirr provides smart defaults for services by which any properly configured system can run quickly, while still being able to override settings as needed. Whirr can also be used as a command line tool for deploying clusters. It uses low level API libraries to work with providers which was mentioned in the [654].

11.17.9 Saga

SAGA(Simple API for Grid Applications) provides an abstraction layer to make it easier for applications to utilize and exploit infra effectively. With infrastructure being changed continuously its becoming difficult for most applications to utilize the advances in hardware. SAGA API provides a high level abstraction of the most common Grid functions so as to be independent of

the diverse and dynamic Grid environments [326]. This shall address the problem of applications developers developing an application tailored to a specific set of infrastructure. SAGA allows computer scientists to write their applications at high level just once and not to worry about low level hardware changes. SAGA provides this high level interface which has the underlying mechanisms and adapters to make the appropriate calls in an intelligent fashion so that it can work on any underlying grid system. "SAGA was built to provide a standardized, common interface across various grid middleware systems and their versions" [234].

As SAGA is to be implemented on different types of middleware it does not specify a single security model but provides hooks to interfaces of various security models. The SAGA API provides a set of packages to implement its objectivity : SAGA supports data management, resource discovery, asynchronous notification, event generation, event delivery etc. It does so by providing set of functional packages namely SAGA file package, replica package, stream package, RPC package, etc. SAGA provides interoperability by allowing the same application code to run on multiple grids and also communicate with applications running on others [326].

11.17.10 Genesis

11.18 DevOps

11.18.1 Docker (Machine, Swarm)

Docker is an open-source container-based technology. A container allows a developer to package up an application and all its part including the stack it runs on, dependencies it is associated with and everything the application requires to run within an isolated environment. Docker separates Application from the underlying Operating System in a similar way as Virtual Machines separates the Operating System from the underlying hardware. Dockerizing an application is lightweight in comparison with running the application on the Virtual Machine as all the containers share the same underlying kernel, the Host OS should be same as the container OS (eliminating guest OS) and an average machine cannot have more than few VMs running on them.

Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands [387]. You can use Machine to create Docker hosts on your local Mac or Windows machine, on your company network, in your data center, or on cloud providers like AWS or Digital Ocean. For Docker 1.12 or higher swarm mode is integrated with the Docker Engine, but on the older versions with Machine's swarm option, user can configure a swarm cluster. Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine. "With these pooled resources user can scale out your application as if it were running on a single, huge computer" [159]. Docker Swarm can be scaled up to 1000 Nodes or up to 50,000 containers

11.18.2 Puppet

Puppet is an open source software configuration management tool [492].This aims at automatic configuration of the software applications and infrastructure. This configuration is done using the easy to use language. Puppet works on major linux distributions and also on microsoft windows , it is also cross-platform application making it easy to manage and portable. [491]

Puppet works with a client server model. All the clients (nodes) which needs to be managed will have 'Puppet Agent' installed and 'Puppet Master' contains the configuration for different hosts this demon process runs on master server. The connection between 'Puppet Master' and 'Puppet

agent' will be established using the secured SSL connection. The configuration at client will be validated as per the set up in Puppet master at a predefined interval. If configuration at client is not matching with the master puppet agent fetches the required changes from master. [280]

Puppet is developed by Puppet Labs using ruby language and released as GNU General Public License (GPL) until version 2.7.0 and the Apache License 2.0 after that. [492]

11.18.3 Chef

Chef is a configuration management tool. It is implemented in Ruby and Erlang. Chef can be used to configure and maintain servers on-premise as well as cloud platforms like Amazon EC2, Google Cloud Platform and Open Stack. The book [385] explains the use of concept called 'recipes' in Chef to manage server applications and utilities such as database servers like MySQL, or HTTP servers like Apache HTPP and systems like Apache Hadoop.

Chef is available in open source version and it also has commercial products for the companies which need it [115]

11.18.4 Ansible

Ansible is an IT automation tool that automates cloud provisioning, configuration management, and application deployment. [513] Once Ansible gets installed on a control node, which is an agentless architecture, it connects to a managed node through the default OpenSSH connection type. [675]

As with most configuration management softwares, Ansible distinguishes two types of servers: controlling machines and nodes. First, there is a single controlling machine which is where orchestration begins. Nodes are managed by a controlling machine over SSH. The controlling machine describes the location of nodes through its inventory.

Ansible manages machines in an agent-less manner. Ansible is decentralized, if needed, Ansible can easily connect with Kerberos, LDAP, and other centralized authentication management systems.

11.18.5 SaltStack

SaltStack (also Salt) platform is a Python-based open-source configuration management software and remote execution engine, which makes systems and configuration management software for the orchestration and automation of CloudOps, ITOps and DevOps at scale [523]. SaltStack is used to manage all the data center things including any cloud, infrastructure, virtualization, application stack, software or code. Salt is built on two major concepts, which are clearly mentioned in [411] as remote execution and configuration management. In the remote execution system, Salt leverages Python to accomplish complex tasks with single-function calls. The configuration management system in Salt, called States, builds upon the remote execution foundation to create repeatable, enforceable configuration for the minions (connects to the master and treats the master as the source)

11.18.6 Boto

The latest version of Boto is Boto3 [218]. Boto3 is the Amazon Web Services (AWS) Development Kit (SDK) for Python [221]. It enables the Python developers to make use of services like Amazon S3 and Amazon EC2 [220]. It provides object oriented APIs along with low-level direct service [219]. It provides simple in-built functions and interfaces to work with Amazon S3 and EC2.

Boto3 has two distinct levels of APIs - client and resource [220]. One-to-one mappings to underlying HTTP API is provided by the client APIs. Resource APIs provide resource objects and collections to perform various actions by accessing the attributes. Boto3 also comes with 'waiters'. Waiters are used for polling status changes in AWS, automatically. Boto3 has these waiters for both the APIs - client as well as resource.

11.18.7 Cobbler

Cobbler is a Linux provisioning system that facilitates and automates the network based system installation of multiple computer operating systems from a central point using services such as DHCP, TFTP and DNS [404]. It is a nifty piece of code that assembles all the usual setup bits required for a large network installation like TFTP, DNS, PXE installation trees and automates the process. It can be configured for PXE, reinstallations and virtualized guests using Xen, KVM or VMware. Cobbler interacts with the koan program for re-installation and virtualization support. Cobbler builds the Kickstart mechanism and offers installation profiles that can be applied to one or many machines. Cobbler has features to dynamically change the information contained in a kickstart template (definition), either by passing variables called ksmeta or by using so-called snippets.

11.18.8 Xcat

xCAT is defined as extreme cloud/cluster administration toolkit. This open source software was developed by IBM and utilized on clusters based on either linux or a version of UNIX called AIX. With this service administrator is enabled with a number of capabilities including parallel system management, provision OS usage on virtual machines, and manage all systems remotely. [720] xCAT works with various cluster types such as high performance computing, horizontal scaling web farms, administrative, and operating systems. [295]

11.18.9 Razor

Razor is a hardware provisioning application, developed by Puppet Labs and EMC. Razor was introduced as open, pluggable, and programmable since most of the provisioning tools that existed were vendor-specific, monolithic, and closed. According to [493] it can deploy both bare-metal and virtual systems. During boot the Razor client automatically discovers the inventory of the server hardware – CPUs, disk, memory, etc., feeds this to the Razor server in real-time and the latest state of every server is updated. It maintains a set of rules to dynamically match the appropriate operating system images with server capabilities as expressed in metadata. User-created policy rules are referred to choose the preconfigured model to be applied to a new node. The node follows the model's directions, giving feedback to Razor as it completes various steps as specified in [494]. Models can include steps for handoff to a DevOps system or to any other system capable of controlling the node.

11.18.10 Juju

Juju (formerly Ensemble) [88] is software from Canonical that provides open source service orchestration. It is used to easily and quickly deploy and manage services on cloud and physical servers. Juju charms can be deployed on cloud services such as Amazon Web Services (AWS), Microsoft Azure and OpenStack. It can also be used on bare metal using MAAS. Specifically [337] lists around 300 charms available for services available in the Juju store. Charms can be written in

any language. It also supports Bundles which are pre-configured collection of Charms that helps in quick deployment of whole infrastructure.

11.18.11 Foreman

11.18.12 OpenStack Heat

Openstack Heat, a template deployment service was the project launched by Openstack, a cloud operating system similar to AWS Cloud Formation. [379] states - Heat is an orchestration service which allows us to define resources over the cloud and connections amongst them using a simple text file called referred as a ‘template’ . "A Heat template describes the infrastructure for a cloud application in a text file that is readable and writable by humans, and can be checked into version control" [276].

Once the execution environment has been setup and a user wants to modify the architecture of resources in the future, a user needs to simply change the template and check it in. Heat shall make the necessary changes. Heat provides 2 types of template - HOT(Heat Orchestration Template) and CFN (AWS Cloud Formation Template). The HOT can be defined as YAML and is not compatible with AWS. The CFN is expressed as JSON and follows the syntax of AWS Cloud Formation and thus is AWS compatible. Further, heat provides an additional @parameters section in its template which can be used to parameterize resources to make the template generic.

11.18.13 Sahara

The Sahara product provides users with the capability to provision data processing frameworks (such as Hadoop, Spark and Storm) on OpenStack [449] by specifying several parameters such as the version,cluster topology and hardware node details. As specified in [522] the solution allows for fast provisioning of data processing clusters on OpenStack for development and quality assurance and utilisation of unused computer power from a general purpose OpenStack IaaS Cloud. Sahara is managed via a REST API with a User Interface available as part of OpenStack Dashboard.

11.18.14 Rocks

[518] Rocks provides open cluster distribution solution is build targeting the scientist with less cluster experience to ease the process of deployment, managing, upgrading and scaling high performance parallel computing cluster. It was initially built on Linux however the latest version Rocks 6.2 Sidewinder is also available on CentOS. Rocks can help create a cluster in few days with default configuration and software packages. Rocks distribution package comes with high-performance distributed and parallel computing tools. It is used by NASA, the NSA , IBM Austin Research LAB, US Navy and many other institutions for their projects.

11.18.15 Cisco Intelligent Automation for Cloud

Cisco Intelligent automation for cloud desires to help different service providers and software professionals in delivering highly secure infrastructure as a service on demand. It provides a foundation for organizational transformation by expanding the uses of cloud technology beyond its infrastructure [121]. From a single self-service portal, it automates standard business processes and sophisticated data center which is beyond the provision of virtual machines. Cisco Intelligent automation for cloud is a unified cloud platform that can deliver any type of service across mixed environments [399]. This leads to an increase in cloud penetration across different business and IT

holdings. Its services range from underlying infrastructure to anything-as-a-service by allowing its users to evaluate, transform and deploy the IT and business services in a way they desire.

11.18.16 Ubuntu MaaS

11.18.17 Facebook Tupperware

Facebook Tupperware is a system which provisions services by taking requirements from engineers and mapping them to actual hardware allocations using containers [486]. Facebook Tupperware simplifies the task of configuring and running services in production and allows engineers to focus on actual application logic. The tupperware system consists of a Scheduler , Agent process and a Server Database. The Scheduler consists of set of machines with one of them as master and the others in standby. The machines share state among them. The Agent process runs on each and every machine and manages all the tasks and co-ordinates with the Scheduler. The Server database stores the details of resources available across machines which is used by the scheduler for scheduling jobs and tasks. Tupperware allows for sandboxing of the tasks which allows for isolation of the tasks. Initially isolation was implemented using chroots but now it is switched to Linux Containers(LXC) .The configuration for the container is done by a specific config file written in a dialect of python by the owner of the process.

11.18.18 AWS OpsWorks

AWS Opsworks is a configuration service provided by Amazon Web Services that uses Chef, a Ruby and Erlang based configuration management tool [681], to automate the configuration, deployment, and management of servers and applications. There are two versions of AWS Opsworks. The first, a fee based offering called AWS OpsWorks for Chef Automate, provides a Chef Server and suite of tools to enable full stack automation. The second, AWS OpsWorks Stacks, is a free offering in which applications are modeled as stacks containing various layers. Amazon Elastic Cloud Compute (EC2) instances or other resources can be deployed and configured in each layer of AWS OpsWorks Stacks [23].

11.18.19 OpenStack Ironic

Ironic [633] project is developed and supported by OpenStack. Ironic provisions bare metal machines instead of virtual machines and functions as hypervisor API that is developed using open source technologies like Preboot Execution Environment (PXE), Dynamic Host Configuration Protocol (DHCP), Network Bootstrap Program (NBP), Trivial File Transfer Protocol (TFTP) and Intelligent Platform Management Interface (IPMI). A properly configured Bare Metal service with the Compute and Network services, could provision both virtual and physical machines through the Compute service's API. But, the number of instance actions are limited, due to physical servers and switch hardware. For example, live migration is not possible on a bare metal instance. The Ironic service has five key components. A RESTful API service, through which other components would interact with the bare metal servers, a Conductor service, various drivers, messaging queue and a database. Ironic could be integrated with other OpenStack projects like Identity (keystone), Compute (nova), Network (neutron), Image (glance) and Object (swift) services.

11.18.20 Google Kubernetes

Google Kubernetes is a cluster management platform developed by Google. According to [358] is an open source system for "automating deployment, scaling and management of containerized

applications". It primarily manages clusters through containers as they decouple applications from the host operating system dependencies and allowing their quick and seamless deployment, maintenance and scaling.

Kubernetes components are designed to be extensible primarily through Kubernetes API. Kubernetes follows a master-slave architecture, according to [664] Kubernetes Master controls and manages the clusters workload and communications of the system. Its main components are etcd, API server, scheduler and controller manager. The individual Kubernetes nodes are the workers where containers are deployed. The components of a node are Kubelet, Kube-proxy and cAdvisor. Kubernetes makes it easier to run application on public and private clouds. It is also said to be self-healing due to features like auto-restart and auto-scaling.

11.18.21 Buildstep

Buildsteps is an open software developed under MIT license. It is a base for Dockerfile and it activates Heroku-style application. Heroku is a platform-as-service (PaaS) that automates deployment of applications on the cloud. The program is pushed to the PaaS using git push, and then PaaS detects the programming language, builds, and runs application on a cloud platform [477]. Buildstep takes two parameters: a tar file that contains the application and a new application container name to create a new container for this application. Build script is dependent on buildpacks that are pre-requisites for buildstep to run. The builder script runs inside the new container. The resulting build app can be run with Docker using docker build -t your_app_name command. [233].

11.18.22 Gitreceive

Gitreceive is used to create an ssh+git user which can accept repository pushes right away and also triggers a hook script. Gitreceive is used to push code anywhere as well as extend your Git workflow. "Gitreceive dynamically creates bare repositories with a special pre-receive hook that triggers your own general gitreceive hook giving you easy access to the code that was pushed while still being able to send output back to the git user" Gitreceive can also be used to provide feedback to the user not only just to trigger code on git push. Gitreceive can be used for the following: "a)for putting a git push deploy interface in front of App Engine b)Run your company build/test system as a separate remote c)Integrate custom systems into your workflow d)Build your own Heroku e)Push code anywhere" [708].

11.18.23 OpenTOSCA

The Topology and Orchestration Specification for Cloud Applications, TOSCA is a new standard facilitating platform independent description of Cloud applications. OpenTOSCA is a runtime for TOSCA-based Cloud applications. The runtime enables fully automated plan-based deployment and management of applications defined in the OASIS TOSCA packaging format CSAR, Cloud Service ARchive. The key tasks of OpenTOSCA, are to operate management operations, run plans, and manage state of the TOSCA [94].

11.18.24 Winery

Eclipse Winery [205] is a "web-based environment to graphically model [Topology and Orchestration Specification for Cloud Applications] TOSCA topologies and plans managing these topologies." Winery [355] is a "tool offering an HTML5-based environment for graph-based modeling of application topologies and defining reusable component and relationship types." This web-based

[355] interface enables users to drag and drop icons to create automated "provisioning, management, and termination of applications in a portable and interoperable way." Essentially, this web-based interface [355] allows users to create an application topology, which "describes software and hardware components involved and relationships between them" as well a management plan, which "captures knowledge [regarding how] to deploy and manage an application."

11.18.25 CloudML

CloudML a research project initiated by SINTEF in 2011 [547]. Cloud computing facilitates to shared and virtualized computer capabilities like storage, memory, CPU, GPU and networks, to user. There is multiple cloud provider, also the IaaS(Infrastructure-as-a-service) and PaaS(Platform-as-a-service). To operate multiple cloud for applications, which requires multiple private, public, or hybrid clouds, limit the capability of each cloud solution. Solution provided by such cloud will get incompatible with others. So, to providing the solution which can compatible with multi-cloud platform is a tedious job. To achieve this CloudML provides a "domain-specific modelling language along with run time environment" [547]. It provides the interoperability and provide vendor lock-in, also it provides the solution on specification of provisioning, deployment, and adaptation concerns of multi-cloud systems. At design time as well as runtime [547]. CloudML provides two level of abstraction while developing model for multi-cloud application:

- Cloud Provider-Independent Model (CPIM), this specifies the provisioning and deployment.
- Cloud Provider-Specific Model (CPSM), which filters the provisioning and deployment of multiple cloud application, according to its cloud.

This two abstract approach help CloudML to achieve the multi-cloud application support [548].

11.18.26 Blueprints

In [289], it is explained that "IBM Blueprint has been replaced by IBM Bluworks Live." In [288], IBM Bluworks Live is described "as a cloud-based business process modeller, belonging under the set of IBM SmartCloud applications" that as [290] states "drive[s] out inefficiencies and improve[s] business operations." Similarly to Google Docs, IBM Bluworks Live is "designed to help organizations discover and document their business processes, business decisions and policies in a collaborative manner." While Google Docs and IBM Bluworks Live are both simple to use in a collaborative manner, [288] explains that IBM Bluworks Live has the "capabilities to implement more complex models."

11.18.27 Terraform

Terraform, developed by HashiCorp, is an infrastructure management tool, it has an open source platform as well as an enterprise version and uses infrastructure as a code to increase operator productivity. Its latest release is Terraform 0.8 According to the website [585] it enables users to safely and predictably create, change and improve the production infrastructure and codifies APIs into declarative configuration files that can be shared amongst other users and can be treated as a code, edited, reviewed and versioned at the same time. The book [619] explains that it can manage the existing and popular service it provides as well as create customized in-house solutions. It builds an execution plan that describes what it can do next after it reaches a desired state to accomplish the goal state. It provides a declarative executive plan which is used for creating applications and implementing the infrastructures. Terraform is mainly used to manage cloud based and SaaS infrastructure, it also supports Docker and VMWare vSphere.

11.18.28 DevOpSlang

DevOpSlang serves as means of collaboration and provides the foundation to automate deployment and operations of an application. Technically, it is a domain specific language based on JavaScript Object Notation (JSON). JSON Schema is used to define a formal schema for DevOpSlang and complete JSON Schema definition of DevOpSlang is publicly available on GitHub project DevOpSlang: <http://github.com/jojow/devopslang> Devopsfiles are the technical artifacts (Unix shell commands, Chef Scripts, etc.) rendered using DevOpSlang to implement operations. Beside some meta data such as 'version' and 'author' Devopsfile defines operations like 'start' consisting of a single or multiple actions which specifies the command to run the application. Similarly, a 'build' operation can be defined to install the dependencies required to run the application. Different abstraction levels may be combined consistently such as a 'deploy' operation consisting of actions on the level of Unix shell commands and actions using portable Chef cookbooks [646].

11.18.29 Any2Api

This framework [647] allows user to wrap an executable program or scripts, for example scripts, chef cookbooks, ansible playbooks, juju charms, other compiled programs etc. to generate APIs from your existing code. These APIs are also containerized so that they can be hosted on a docker container, vagrant box etc Any2Api helps to deal with problems like scale of application, technical expertise, large codebase and different API formats. The generated API hide the tool specific details simplifying the integration and orchestration different kinds of artifacts. The APIfication framework contains various modules:

1. Invokers, which are capable of running a given type of executable for example cookbook invoker can be used to run Chef cookbooks
2. Scanners, which are capable of scanning modules of certain type for example cookbook scanner scans Chef cookbooks.
3. API impl generators, which are doingthe actual work to generate the API implementation.

The final API implementation [645] is is packages with executable in container. The module is packaged as npm module. Currently any2api-cli provides a command line interface and web based interface is planned for future development. Any2Api is very useful for by devops to orchestrate open source ecosystem without dealing with low level details of chef cookbook or ansible playbook or puppet. It can also be very useful in writing microservices where services talk to each other using well defined APIs.

11.19 IaaS Management from HPC to hypervisors

11.19.1 Xen

Xen is the only open-source bare-metal hypervisor based on microkernel design [721]. The hypervisor runs at the highest privilege among all the processes on the host. It's responsibility is to manage CPU and memory and handle interrupts [722]. Virtual machines are deployed in the guest domain called DomU which has no access privilege to hardware. A special virtual machine is deployed in the control domain called Domain 0. It contains hardware drivers and the toolstack to control the VMs and is the first VM to be deployed. Xen supports both Paravirtualization and hardware assisted virtualization. The hypervisor itself has a very small footprint. It's being actively maintained by Linux Foundation under the trademark *XEN Project*. Some of the features included in the latest releases include em Reboot-free Live Patching (to enable application of security patches

without rebooting the system) and KCONFIG support (compilation support to create a lighter version for requirements such as embedded systems) [723].

11.19.2 KVM

It is an acronym for Kernel-based Virtual Machine for the Linux Kernel that turns it into a hypervisor upon installation. It was originally developed by Qumranet in 2007 [360]. It has a kernel model and uses kernel as VMM. It only supports fully virtualized VMs. It is very active for Linux users due to its ease of use, it can be completely controlled by ourselves and there is an ease for migration from or to other platforms. It is built to run on a x86 machine on an Intel processor with virtualization technology extensions (VT-x) or an AMD-V. It supports 32 and 64 bit guests on a 64 bit host and hardware visualization features. The supported guest systems are Solaris, Linux, Windows and BSD Unix [361].

11.19.3 QEMU

QEMU (Quick Emulator) is a generic open source hosted hypervisor [692] that performs hardware virtualization (virtualization of computers as complete hardware platform, certain logical abstraction of their componentry or only the certain functionality required to run various operating systems) :cite-‘www-qemu’ and also emulates CPUs through dynamic binary translations and provides a set of device models, enabling it to run a variety of unmodified guest operating systems.

When used as an emulator, QEMU can run Operating Systems and programs made for one machine (ARM board) on a different machine (e.g. a personal computer) and achieve good performance by using dynamic translations. When used as a virtualizer, QEMU achieves near native performance by executing the guest code directly on the host CPU. QEMU supports virtualization when executing under the Xen hypervisor or using KVM kernel module in Linux [495].

Compared to other virtualization programs like VMWare and VirtualBox, QEMU does not provide a GUI interface to manage virtual machines nor does it provide a way to create persistent virtual machine with saved settings. All parameters to run virtual machine have to be specified on a command line at every launch. It’s worth noting that there are several GUI front-ends for QEMU like virt-manager and gnome-box.

11.19.4 Hyper-V

Hyper-V is a native hypervisor which was first released alongside Windows Server 2008. It is available free of charge for all the Windows Server and some client operating systems since the release. Microsoft Hyper-V, is also codenamed as Viridian and formerly known as Windows Server Virtualization, is a native hypervisor. Xbox One also includes Hyper-V, in which it would launch both Xbox OS and Windows 10. [691]

Hyper-V is used to create virtual machines on x86-64 systems which are running Windows. Windows 8 onwards, Hyper-V supersedes Windows Virtual PC as the hardware virtualization component of the client editions of Windows NT. A server computer running Hyper-V can be configured to expose individual virtual machines to one or more networks.

11.19.5 VirtualBox

11.19.6 OpenVZ

OpenVZ (Open Virtuozzo) is an operating system-level virtualization technology for Linux. It allows a physical server to run multiple isolated operating system instances, called containers, virtual private servers, or virtual environments (VEs). OpenVZ is similar to Solaris Containers and LXC. [451] While virtualization technologies like VMware and Xen provide full virtualization and can run multiple operating systems and different kernel versions, OpenVZ uses a single patched Linux kernel and therefore can run only Linux. All OpenVZ containers share the same architecture and kernel version. This can be a disadvantage in situations where guests require different kernel versions than that of the host. However, as it does not have the overhead of a true hypervisor, it is very fast and efficient. Memory allocation with OpenVZ is soft in that memory not used in one virtual environment can be used by others or for disk caching. [281] While old versions of OpenVZ used a common file system (where each virtual environment is just a directory of files that is isolated using chroot), current versions of OpenVZ allow each container to have its own file system. OpenVZ has four main features, [186] 1. OS virtualization: A container (CT) looks and behaves like a regular Linux system. It has standard startup scripts; software from vendors can run inside a container without OpenVZ-specific modifications or adjustment; A user can change any configuration file and install additional software; Containers are completely isolated from each other and are not bound to only one CPU and can use all available CPU power. 2. Network virtualization: Each CT has its own IP address and CTs are isolated from the other CTs meaning containers are protected from each other in the way that makes traffic snooping impossible; Firewalling may be used inside a CT 3. Resource management: All the CTs are use the same kernel. OpenVZ resource management consists of four main components: two-level disk quota, fair CPU scheduler, disk I/O scheduler, and user bean counters. 4. Checkpointing and live migration: Checkpointing allows to migrate a container from one physical server to another without a need to shutdown/restart a container. This feature makes possible scenarios such as upgrading your server without any need to reboot it: if your database needs more memory or CPU resources, you just buy a newer better server and live migrate your container to it, then increase its limits.

11.19.7 LXC

LXC (Linux Containers) is an operating-system-level virtualization method for running multiple isolated Linux systems (containers) on a control host using a single Linux kernel [375]. LXC are similar to the traditional virtual machines but instead of having separate kernel process for the guest operating system being run, containers would share the kernel process with the host operating system. This is made possible with the implementation of namespaces and cgroups. [656]

Containers are light weighed (As guest operating system loading and booting is eliminated) and more customizable compared to VM technologies. The basis for docker development is also LXC. [376]. Linux containers would work on the major distributions of linux this would not work on Microsoft Windows.

11.19.8 Linux-Vserver

Linux-VServers are used on web hosting services, pooling resources and containing any security breach. [484] “Linux servers consist of three building blocks Hardware, Kernel and Applications” the purpose of kernel is to provide abstraction layer between hardware and application. Linux-Vserver provides VPS securely partitioning the resources on computer system in such a way that process cannot mount denial of service out of the partition.

It utilises the power of Linux kernel and top of it with additional modification provides secure layer to each process (VPS) feel like it is running separate system. By providing context separation, context capabilities, each partition called as security context, chroot barrier created on private directory of each VPS to prevent unauthorized modification. Booting VPS in new secure context is just matter of booting server, context is so robust to boot many server simultaneously.

The virtual servers shares same system calls, shares common file system, process within VS are queued to same scheduler that of host allowing guest process to run concurrently on SMP systems. No additional overhead of network virtualization. These few advantages of Linux-VServer.

11.19.9 OpenStack

OpenStack [557] is a free and open source cloud operating system mostly deployed as infrastructure as a service(IaaS) that allows us to control large pool of computers, storage, and networking resources. OpenStack is managed by OpenStack Foundation [198].

Just like cloud, OpenStack provides infrastructure which runs as platform upon which end users can create applications. Key components of OpenStack include: Nova: which is the primary computing engine, Swift: which is a storage system for object and files, Neutron: which ensures effective communication between each of the components of the OpenStack. Other components include: Cinder, Horizon, Keystone, Glance, Ceilometer and Heat. The main goal of Openstack is to allow business to build Amazon-like cloud services in their own data centers. OpenStack is licensed under the Apache 2.0 license [45]

11.19.10 OpenNebula

According to OpenNebula webpage [5] it provides simple but feature-rich and flexible solutions for the comprehensive management of virtualized data centers to enable private, public and hybrid IaaS clouds. It is a cloud computing platform for managing heterogeneous distributed data centers infrastructures. The OpenNebula toolkit includes features for management, scalability, security and accounting. It is used in various sectors like hosting providers, telecom providers, telecom operators, IT service providers, supercomputing centers, research labs, and international research projects [446]. More about OpenNebula can be found in the following paper that is published at IEEE computer society [405]

11.19.11 Eucalyptus

Eucalyptus is a Linux-based open source software framework for cloud computing that implements Infrastructure as a Service (IaaS). IaaS are systems that give users the ability to run and control entire virtual machine instances deployed across a variety of physical resources [430]. Eucalyptus is an acronym for “Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems.”

A Eucalyptus private cloud is deployed on an enterprise’s data center infrastructure and is accessed by users over the enterprise’s intranet. Sensitive data remains entirely secure from external interference behind the enterprise firewall [597].

11.19.12 Nimbus

Nimbus Infrastructure [425] is an open source IaaS implementation. It allows deployment of self-configured virtual clusters and it supports configuration of scheduling, networking leases, and

usage metering.

Nimbus Platform [424] provides an integrated set of tools which enable users to launch large virtual clusters as well as launch and monitor the cloud apps. It also includes service that provides auto-scaling and high availability of resources deployed over multiple IaaS cloud. The Nimubs Platform tools are cloudinit.d, Phantom and Context Broker. In this paper [166], the use of Nimbus Phantom to deploy auto-scaling solution across multiple NSF FutureGrid clouds is explained. In this implementation Phantom was responsible for deploying instances across multiple clouds and monitoring those instance. Nimbus platform supports Nimbus, Open Stack, Amazon and several other clouds.

11.19.13 CloudStack

Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform. It uses existing hypervisors such as KVM, VMware vSphere, and XenServer/XCP for virtualization. In addition to its own API, CloudStack also supports the Amazon Web Services (AWS) API and the Open Cloud Computing Interface from the Open Grid Forum. [594]

CloudStack features like built-in high-availability for hosts and VMs, AJAX web GUI for management, AWS API compatibility, Hypervisor agnostic, snapshot management, usage metering, network management (VLAN's, security groups), virtual routers, firewalls, load balancers and multi-role support. [676]

11.19.14 CoreOS

[138] states that “CoreOS is a linux operating system used for clustered deployments.” CoreOS allows applications to run on containers. CoreOS can be run on clouds, virtual or physical servers. CoreOS allows the ability for automatic software updates inorder to make sure containers in cluster are secure and reliable. It also makes managing large cluster environments easier. CoreOS provides open source tools like CoreOS Linux, etcd,rkt and flannel. CoreOS also has commercial products Kubernetes and CoreOS stack. In CoreOS linux service discovery is achieved by etcd, applications are run on Docker and process management is achieved by fleet.

11.19.15 rkt

rkt is an container manager developed by CoreOS [137] designed for Linux clusters. It is an alternative for Docker runtime and is designed for server environments with high security and compositiby requirement. It is the first implementation of the open container standard called *App Container* or *appc* specification but not the only one. It is a standalone tool that lives outside of the core operating system and can be used on variety of platforms such as Ubuntu, RHEL, CentOS, etc. rkt implements the facilities specified by the App Container as a command line tool. It allows execution of App Containers with pluggable isolation and also varying degrees of protection. Unlike Docker, rkt runs containers as un-privileged users making it impossible for attackers to break out of the containers and take control of the entire physical server. rkt’s primary interface comprises a single executable allowing it easily integrate with existing init systems and also advanced cluster environments. rkt is open source and is written in the Go programming language [229].

11.19.16 VMware ESXi

VMware ESXi (formerly ESX) is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers [660]. The name ESX originated as an abbreviation of Elastic Sky X. ESXi installs directly onto your physical server enabling it to be partitioned into multiple logical servers referred to as virtual machines. Management of VMware ESXi is done via APIs. This allows for an *agent-less* approach to hardware monitoring and system management. VMware also provides remote command lines, such as the vSphere Command Line Interface (vCLI) and PowerCLI, to provide command and scripting capabilities in a more controlled manner. These remote command line sets include a variety of commands for configuration, diagnostics and troubleshooting. For low-level diagnostics and the initial configuration, menu-driven and command line interfaces are available on the local console of the server [636].

11.19.17 vSphere and vCloud

vSphere was developed by VMware and is a cloud computing virtualization platform. [637] vSphere is not one piece of software but a suite of tools that contains software such as vCenter, ESXi, vSphere client and a number of other technologies. ESXi server is a type 1 hypervisor on a physical machine of which all virtual machines are installed. The vSphere client then allows administrators to connect to the ESXi and manage the virtual machines. The vCenter server is a virtual machine that is also installed on the ESXi server which is used in environments when multiple ESXi servers are exist. Similarly, vCloud is also a suite of applications but for establishing an infrastructure for a private cloud. [98] The suite includes the vsphere suite, but also contains site recovery management for disaster recovery, site networking and security. Additionally, a management suite that can give a visual of the infrastructure to determine where potential issues might arise.

11.19.18 Amazon

Amazon's AWS (Amazon Web Services) is a provider of Infrastructure as a Service (IaaS) on cloud. It provides a broad set of infrastructure services, such as computing, data storage, networking and databases. One can leverage AWS services by creating an account with AWS and then creating a virtual server, called as an instance, on the AWS cloud. In this instance you can select the hard disk volume, number of CPUs and other hardware configuration based on your application needs. You can also select operating system and other software required to run your application. AWS lets you select from the countless services. Some of them are mentioned below:

- Amazon Elastic Computer Cloud (EC2) - Amazon Simple Storage Service (Amazon S3) - Amazon CloudFront - Amazon Relational Database Service (Amazon RDS) - Amazon SimpleDB - Amazon Simple Notification Service (Amazon SNS) - Amazon Simple Queue Service (Amazon SQS) - Amazon Virtual Private Cloud (Amazon VPC)

Amazon EC2 and Amazon S3 are the two core IaaS services, which are used by cloud application solution developers worldwide. [582]

Improve: all of them need bibentries

11.19.19 Azure**11.19.20 Google and other public Clouds**

A public cloud is a scenario where a provider provides services such as infrastructure or applications to the public over the internet. Google cloud generally refers to services such as cloud print, connect, messaging, storage and platform [243]. Google cloud print allows a print-aware application on a device, installed on a network, to provide prints to any printer on that network. Cloud connect allows an automatic storage and synchronization of Microsoft word documents, power-points and excel sheets to Google docs while preserving the Microsoft office formats. In certain cases, developers require important notifications to be sent to applications targeting android operating system. Google cloud messaging provides such services. Google cloud platform allows the developers to deploy their mobile, web and backend solutions on a highly scalable and reliable infrastructure [244]. It gives developers a privilege of using any programming language. Google cloud platform provides a wide range of products and services including networking, storage, machine learning, big data, authentication and security, resource management, etc. In general, public clouds provide services to different end users with the usage of the same shared infrastructure [648]. Windows Azure services platform, Amazon elastic compute cloud and Sun cloud are few examples of public clouds.

11.19.21 Networking: Google Cloud DNS

Under the umbrella of google cloud platform, helps user to publish their domain using Google's infrastructure. It is highly scalable, low latency, high availability DNS service residing on infrastructure same as google.

It is build around projects a resource container, domain for access control, and billing configuration. Managed zones holds records for same DNS name. The resource record sets collection holds current state of the DNS that make up managed zones it is unmodifiable or cannot be modified easily and changes to record sets. It supports A address records, AAAA IPv6, CAA Certificate authority, CNAME canonical name, MX mail exchange, NAPTR naming authority pointer, NS Name server record, SOA start of authority, SPF Sender policy framework, "SRV" service locator, TXT text record.

11.19.22 Amazon Route 53

Amazon Route 53 is a DNS (Domain Name System) service that gives developers and businesses a reliable way to route end users to Internet applications. The number 53 refers to TCP or UDP port 53, where DNS server requests are addressed [24].

When using Route 53 as your DNS provider, in case of a recursion, the query of fetching an IP address (of a website or application) always goes to the closest server location to reduce query latency. The Route 53 server returns the IP address enabling the browser to load the website or application. Route 53 can also be used for registering domain names and arranging DNS health checks to monitor the server [649].

11.20 Cross-Cutting Functions

Monitoring

11.20.1 Ambari

Apache Ambari is an open source platform that enables easy management and maintenance of Hadoop clusters, regardless of cluster size. Ambari has a simplified Web UI and robust REST API for automating and controlling cluster operations. [31] illustrates Ambari to provide key benefits including easy installation, configuration, and management with features such as Smart Configs and cluster recommendations and Ambari Blueprints, to provide repeatable and automated cluster creation. Ambari provides a centralized security setup that automates security capabilities of clusters. Ambari provides a holistic view for cluster monitoring and provides visualizations for operation metrics. [29] provides documentation about Ambari, including a quick start guide for installing a cluster with Ambari. [30] provides the project documents for ambari on github.

11.20.2 Ganglia

Ganglia is a scalable distributed monitoring system for high-performance computing systems (clusters and grids). It is a BSD-licensed open-source project that grew out of the University of California, Berkeley Millennium Project which was initially funded in large part by the National Partnership for Advanced Computational Infrastructure (NPACI) and National Science Foundation RI Award EIA-9802069 [216].

It relies on a multicast-based listen/announce protocol to monitor state within clusters. It uses a tree of point-to-point connections amongst representative cluster nodes to unite clusters and aggregate their state [217]. It leverages technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on thousands of clusters around the world, handling clusters with 2000 nodes.

11.20.3 Nagios (413)

Nagios is a platform, which provides a set of software for network infrastructure monitoring. It also offers administrative tools to diagnose when failure events happen, and to notify operators when hardware issues are detected. Specifically, illustrates that Nagios is consist of modules including [334]: a core and its dedicated tool for core configuration, extensible plugins and its frontend. Nagios core is designed with scalability in mind. Nagios contains a specification language allowing for building an extensible monitoring systems. Through the Nagios API components can integrate with the Nagios core services. Plugins can be developed via static languages like C or script languages. This mechanism empowers Nagios to monitor a large set of various scenarios yet being very flexible. [317] Besides its open source components, Nagios also has commercial products to serve needing clients.

11.20.4 Inca

Inca is a grid monitoring [378] software suite. It provides grid monitoring features. These monitoring features provide operators failure trends, debugging support, email notifications, environmental issues etc. [303]. It enables users to automate the tests which can be executed on a periodic basis. Tests can be added and configured as and when needed. It helps users with different portfolios like system administrators, grid operators, end users etc Inca provides user-level grid monitoring. For each user it stores results as well as allows users to deploy new tests as well as share the results with other users. The incat web ui allows users to view the status of test, manage test and results. The architectural blocks of inca include report repository, agent, data consumers and depot. Reporter is

an executable program which is used to collect the data from grid source. Reporters can be written in perl and python. Inca repository is a collection of pre build reporters. These can be accessed using a web url. Inca repository has 150+ reporters available. Reporters are versioned and allow automatic updates. Inca agent does the configuration management. Agent can be managed using the incat web ui. Inca depot provides storage and archival of reports. Depot uses relational database for this purpose. The database is accessed using hibernate backend. Inca web UI or incat provides real time as well as historical view of inca data. All communication between inca components is secured using SSL certificates. It requires user credentials for any access to the system. Credentials are created at the time of the setup and installation. Inca's performance has been phenomenal in production deployments. Some of the deployments are running for more than a decade and has been very stable. Overall Inca provides a solid monitoring system which not only monitors but also detects problems very early on.

Security & Privacy

11.20.5 InCommon

The mission of InCommon is to “create and support a common trust framework for U.S. education and research. This includes trustworthy shared management of access to on-line resources in support of education and research in the United States.” [304] This mission ultimately is a simplification and an elimination of the need for multiple accounts across various websites that are at risk of data spills or misuse. In the academic setting, this helps assist researchers to focus on their area of study, and enabling the cross collaboration which is happening on a global scale. Currently any two and four year higher education institution that is accredited is eligible for joining InCommon.

11.20.6 Eduroam (168)

Eduroam is an initiative started in the year 2003 when the number of personal computers with in the academia are growing rapidly. The goal is to solve the problem of secure access to WI-FI due to increasing number of students and research teams becoming mobile which was increasing the administrative problems for provide access to WI-FI. Eduroam provides any user from an eduroam participating site to get network access at any institution connected through eduroam. According to the organization it uses a combination of radius-based infrastructure with 802.1X standard technology to provide roaming access across research and educational networks. The role of the RADIUS hierarchy is to forward user credentials to the users home institution where they can be verified. This proved to be a successful solution when compared to other traditional ways like using MAC-address, SSID, WEP, 802.1x(EAP-TLS, EAP-TTLS), VPN Clients, Mobile-IP etc which have their own short comings when used for this purpose [194]. Today by enabling eduroam users get access to internet across 70 countries and tens of thousands of access points worldwide.

11.20.7 OpenStack Keystone

[349] Keystone is the identity service used by OpenStack for authentication (authN) and high-level authorization (authZ). There are two authentication mechanisms in Keystone, UUID token, and PKI. Universally unique identifier (UUID) is a 128-bit number used to identify information (user). Each application after each request of the client checks token validity online. PKI was introduced later and improved the security of Keystone [145]. In PKI, each token has its own digital signature that can be checked by any service and OpenStack application with no necessity to ask for Keystone database [126].

Thus, Keystone enables ensuring user's identity with no need to transmit its password to applications. It has recently been rearchitected to allow for expansion to support proxying external services and AuthN/AuthZ mechanisms such as oAuth, SAML and openID in future versions [450].

11.20.8 LDAP

LDAP stands for Lightweight Directory Access Protocol. It is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the Internet or on corporate internet. [583]

LDAP is a lightweight (smaller amount of code) version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network. In a network, a directory tells you where in the network something is located. On TCP/IP networks (including the Internet), the domain name system (DNS) is the directory system used to relate the domain name to a specific network address (a unique location on the network). However, you may not know the domain name. LDAP allows you to search for an individual without knowing where they're located (although additional information will help with the search). An LDAP directory can be distributed among many servers. Each server can have a replicated version of the total directory that is synchronized periodically. An LDAP server is called a Directory System Agent (DSA). An LDAP server that receives a request from a user takes responsibility for the request, passing it to other DSAs as necessary, but ensuring a single coordinated response for the user.

11.20.9 Sentry

"Apache Sentry [63] is a granular, role-based authorization module for Hadoop. Sentry provides the ability to control and enforce precise levels of privileges on data for authenticated users and applications on a Hadoop cluster. Sentry currently works out of the box with Apache Hive, Hive Metastore/HCatalog, Apache Solr, Impala and HDFS (limited to Hive table data). Sentry is designed to be a pluggable authorization engine for Hadoop components. It allows the client to define authorization rules to validate a user or application's access requests for Hadoop resources. Sentry is highly modular and can support authorization for a wide variety of data models in Hadoop."

11.20.10 Sqrrl

11.20.11 OpenID

OpenID is an authentication protocol that allows users to log in to different websites, which are not related, using the same login credentials for each, i.e. without having to create separate id and password for all the websites. The login credentials used are of the existing account. The password is known only to the identity provider and nobody else which relieves the users' concern about identity being known to an insecure website. [444] It provides a mechanism that makes the users control the information that can be shared among multiple websites. OpenID is being adopted all over the web. Most of the leading organizations including Microsoft, Facebook, Google, etc. are accepting the OpenIDs [445]. It is an open source and not owned by anyone. Anyone can use OpenID or be an OpenID provider and there is no need for an individual to be approved.

11.20.12 SAML OAuth

As explained in [525], Security Assertion Markup Language (SAML) is a secured XML based communication mechanism for communicating identities between organizations. The primary use case of SAML is Internet SSO. It eliminates the need to maintain multiple authentication credentials in multiple locations. This enhances security by elimination opportunities for identity theft/Phishing. It increases application access by eliminating barriers to usage. It reduces administration time and cost by excluding the effort to maintain duplicate credentials and helpdesk calls to reset forgotten passwords. Three entities of SAML are the users, Identity Provider (IdP-Organization that maintains a directory of users and an authentication mechanism) and Service Provider(SP-Hosts the application /service). User tries to access the application by clicking on a link or through an URL on the internet. The Federated identity software running in the IdP validates the user's identity and the user is then authenticated. A specifically formatted message is then communicated to the federated identity software running at SP. SP creates a session for the user in the target application and allows the user to get direct access once it receives the authorization message from a known identity provider.

Distributed Coordination

11.20.13 Google Chubby

Chubby Distributed lock service [242] is intended for use within a loosely-coupled distributed system consisting of moderately large numbers of small machines connected by a high-speed network. Asynchronous consensus is solved by the Paxos protocol. The implementation in Chubby is based on coarse grained lock server and a library that the client applications link against. As per the 2016 paper [16], an open-source implementation of the Google Chubby lock service was provided by the Apache ZooKeeper project. ZooKeeper used a Paxos-variant protocol Zab for solving the distributed consensus problem. Google stack and Facebook stack both use versions of zookeeper.

11.20.14 Zookeeper

Zookeeper provides coordination services to distributed applications. It includes synchronization, configuration management and naming services among others. The interfaces are available in Java and C [731]. The services themselves can be distributed across multiple Zookeeper servers to avoid single point of failure. If the leader fails to answer, the clients can fall-back to other nodes. The state of the cluster is maintained in an in-memory image along with a persistent storage file called znode by each server. The cluster namespace is maintained in a hierarchical order. The changes to the data are totally ordered [732] by stamping each update with a number. Clients can also set a watch on a znode to be notified of any change [733]. The performance of the ZooKeeper is optimum for read-dominant workloads. It's maintained by Apache and is open-source.

11.20.15 Giraffe

Giraffe is a scalable distributed coordination service. Distributed coordination is a media access technique used in distributed systems to perform functions like providing group membership, gaining lock over resources, publishing, subscribing, granting ownership and synchronization together among multiple servers without issues. Giraffe was proposed as alternative to coordinating services like Zookeeper and Chubby which were efficient only in read-intensive scenario and small ensembles. To overcome this three important aspects were included in the design of Giraffe [539].

First feature is Giraffe uses interior-node joint trees to organize coordination servers for better scalability. Second, Giraffe uses Paxos protocol for better consistency and to provide more fault-tolerance. Finally, Giraffe also facilitates hierarchical data organization and in-memory storage for high throughput and low latency.

11.20.16 JGroups

11.21 Message and Data Protocols

11.21.1 Avro

Apache Avro is a data serialization system, which provides rich data structures, remote procedure call(RPC), a container file to store persistent data and simple integration with dynamic languages [40]. Avro depends on schemas, which are defined with JSON. This facilitates implementation in other languages that have the JSON libraries. The key advantages of Avro are schema evolution - Avro will handle the missing/extra/modified fields, dynamic typing - serialization and deserialization without code generation, untagged data - data encoding and faster data processing by allowing data to be written without overhead.

11.21.2 Thrift

The Apache Thrift software framework, for scalable cross-language services development, combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml and Delphi and other languages. [549] It includes a complete stack for creating clients and servers. It includes a server infrastructure to tie the protocols and transports together. There are blocking, non-blocking, single and multithreaded servers available. Thrift was originally developed at Facebook, it was open sourced in April 2007 and entered the Apache Incubator in May, 2008. It became an Apache TLP in October, 2010. [608]

11.21.3 Protobuf

Protocol Buffer [489] is a way to serialize structured data into binary form (stream of bytes) in order to transfer it over wires or for storage. It is used for inter application communication or for remote procedure call (RPC). It involves a interface description that describes the structure of some data and a program that can generate source code or parse it back to the binary form. It emphasizes on simplicity and performance over xml. Though xml is more readable but requires more resources in parsing and storing. This is developed by Google and available under open source licensing. The parser program is available in many languages including java and python.

11.22 Technologies To Be Integrated

11.22.1 Snort

[696] Snort is a Network Intrusion Prevention System (NIPS) and Network Intrusion Detection System (NIDS). Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching. These basic services have many purposes including application-aware triggered quality of service, to de-prioritize bulk traffic when

latency-sensitive applications are in use. The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans. Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyze it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.

11.22.2 Fiddler

Fiddler is an HTTP debugging proxy server application. Fiddler captures HTTP and HTTPS traffic and logs it for the user to review by implementing man-in-the-middle interception using self-signed certificates. Fiddler can also be used to modify (fiddle with) HTTP traffic for troubleshooting purposes as it is being sent or received.[5] By default, traffic from Microsoft's WinINET HTTP(S) stack is automatically directed to the proxy at runtime, but any browser or Web application (and most mobile devices) can be configured to route its traffic through Fiddler [687].

11.22.3 Zeppelin

Apache Zeppelin [728] provides an interactive environment for big data data analytics on applications using distributed data processing systems like Hadoop and Spark. It supports various tasks like data ingestion, data discovery, data visualization, data analytics and collaboration. Apache Zeppelin provides built-in Apache Spark integration and is compatible with many languages/data-processing backends like Python, R, SQL, Cassandra and JDBC. It also supports adding new language backend. Zeppelin also lets users to collaborate by sharing their Notebooks, Paragraph and has option to broadcast any changes in realtime.

11.22.4 Open MPI

The Open MPI Project [600] is an open source Message Passing Interface implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI is therefore able to combine the expertise, technologies, and resources from all across the High Performance Computing community in order to build the best MPI library available. Open MPI offers advantages for system and software vendors, application developers and computer science researchers. Open MPI [210] provides functionality that has not previously been available in any single, production-quality MPI implementation, including support for all of MPI-2, multiple concurrent user threads, and multiple options for handling process and network failures.

11.22.5 Apache Tomcat

Apache tomcat is an open source java servlet container. [456] It is used in IT industry as a HTTP web server which listens to the requests made by web client and send responses. The main components of tomcat are cataline, coyote and jasper. The most stable version of Apache Tomcat server is version 8.5.11. Apache tomcat is released under Apache License version 2. [667] As it is cross platform, it can run in any platform or OS like Windows, UNIX, AIX or SOLARIS etc. It is basically an integral part of many java based web application.

11.22.6 Apache Beam

Apache Beam attempts to abstract away the need to write code for multiple data-oriented workflows, e.g., batch, interactive and streaming, as well as multiple big data tools, e.g., Storm, Spark and Flink. Instead, Beam attempts to automagically map a dataflow process written in Java or Python to the target runtime environment via *runners*. As a result, switching a data processing routine from Spark to Flink only requires changing the target runtime environment as opposed to re-writing the entire process [479] (perhaps in a completely different language). Google contributed its Dataflow SDK, the Dataflow model and three runners [703] to the Apache Software Foundation in the first half of 2016. The ASF elevated Beam to a Top-Level project in January 2017. Jean-Baptiste Onofre of French tech company Talend, and a frequent Apache project contributor, champions the project. [437] It should be grouped with the technologies in the *Interoperability* section.

11.22.7 Cloudability

Cloudability is a financial management tool for analyzing and monitoring all cloud expenses across an organization. It can be used for cost monitoring, usage rightsizing, reserved instance planning, cost allocation, role-based visibility. It aggregates expenditures into reports, helps identify opportunities for reducing costs, offers budget alerts and recommendations via SMS and email, and provides APIs for connecting cloud billing and usage data to any business or financial system. [123]

11.22.8 CUDA

It is a parallel computing platform and application programming interface(API) model created by Nvidia. It allows software developers to use a CUDA-enabled graphics processing unit for general purpose processing. The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels. CUDA platform has advantages such as scattered reads i.e the code can read from arbitrary addresses in memory, unified virtual memory, unified memory, faster downloads and readbacks to and from the GPU and full support for integer and bitwise operations. [683]. CUDA is used for accelerated rendering of 3D graphics, accelerated interconversion of video file formats, encryption, decryption and compression of files. It is also used for distributed calculations, face recognition and distributed computing. [683]

11.22.9 Blaze

Blaze library translates NumPy/Pandas-like syntax to data computing systems (e.g. database , in-memory, distributed-computing). This provides Python users with a familiar interface to query data in a variety of other data storage systems. One Blaze query can work across data ranging from a CSV file to a distributed database.

Blaze presents a pleasant and familiar interface regardless of what computational solution or database we use (e.g. Spark, Impala, SQL databases, No-SQL data-stores, raw-files). It mediates the users interaction with files, data structures, and databases, optimizing and translating the query as appropriate to provide a smooth and interactive session. It allows the data scientists and analyst to write their queries in a unified way that does not have to change because the data is stored in another format or a different data-store. [133]

11.22.10 CDAP

CDAP [705] stands for Cask Data Application Platform. CDAP is an application development platform using which developers can build, deploy and monitor applications on Apache Hadoop. In a typical CDAP application, a developer can ingest data, store and manage datasets on Hadoop, perform batch mode data analysis, and develop web services to expose the data. They can also schedule and monitor the execution of the application. This way, CDAP enables the developers to use single platform to develop the end to end application on Apache Hadoop.

CDAP documentation [706] explains the important CDAP concepts of CDAP Dataset, CDAP Application and CDAP Services. CDAP Datasets provide logical abstraction over the data stored in Hadoop. CDAP Applications provide containers to implement application business logic in open source processing frameworks like map reduce, Spark and real time flow. CDAP applications also provide standardize way to deploy and manage the apps. CDAP Services provide services for application management, metadata management, and streams management. CDAP can be deployed on various Hadoop Platforms such as Apache Hadoop, Cloudera Hadoop, Hortonworks Hadoop and Amazon EMR. CDAP sample apps [228] provide explain how to implement apps on CDAP platform.

11.22.11 Apache Arrow

Apache arrow allows execution engines to utilize what is known as Single Input multiple data (SIMD). [36] This SIMD is an operation that allows modern processors to take advantage of this engine. Peformance is enhanced by grouping relevant data as close as possible in a column format. Many programming languages are supported such a Java, C, C++, Python and it is anticipated that languages will be added as it grows. It is still in early developemnt but has released a 0.1.0 build.

11.22.12 OpenRefine

OpenRefine (formerly GoogleRefine) is an open source tool that is dedicated to cleaning messy data. With the help of this user-friendly tool you can explore huge data sets easily and quickly even if the data is a little unstructured. It allows you to load data, understand it, clean it up, reconcile it, and augment it with data coming from the web [447].It operates on rows of data which have cells under columns, which is very similar to relational database tables. One OpenRefine project is one table. The user can filter the rows to display using facets that define filtering criteria. most operations in OpenRefine are done on all visible rows: transformation of all cells in all rows under one column, creation of a new column based on existing column data, etc. All actions that were done on a dataset are stored in a project and can be replayed on another dataset. It has a huge community with lots of contributors meaning that the software is constantly getting better and better.

11.22.13 Apache OODT

Apache Object Oriented Data Technology (OODT) [438] is a distributed data management technology that helps to integrate and archive your processes, your data, and its metadata. OODT allows to generate, process, manage and analyze distributed and heterogeneous data enabling integration of different, distributed software systems. Apache OODT uses structured XML-based capturing of the processing pipeline which is used to create, edit, manage and provision workflow and task execution. OODT is written in Java programming language and provides its own set of APIs for storing and processing data. [439] It provides three core services. A File Manager is responsible

for tracking file locations, their metadata, and for transferring files from a staging area to controlled access storage. A Workflow Manager captures control flow and data flow for complex processes, and allows for reproducibility and the construction of scientific pipelines. A Resource Manager handles allocation of workflow tasks and other jobs to underlying resources, e.g., Python jobs go to nodes with Python installed on them similarly jobs that require a large disk or CPU are properly sent to those nodes that fulfill those requirements. OODT is now supported with Apache Mesos and Grid Computing which can allow for creating of highly distributed, scalable data platforms that can process large amounts of data. OODT technology is used in NASA's Jet Propulsion Labatory.

11.22.14 Omid

Omid is a “flexible, reliable, high performant and scalable ACID transactional framework” [53] for NoSQL databases, developed by Yahoo for HBase and contributed to the Apache community. Most NoSQL databases, do not natively support ACID transactions. Omid employs a lock free approach from concurrency and can scale beyond 100,000 transactions per second. At Yahoo, millions of transactions per day are processed by Omid. [467].

Omid is currently in the Apache Incubator. All projects accepted by the Apache Software Foundation (ASF) undergo an incubation period until a review indicates that the project meets the standards of other ASF projects [51]

11.22.15 Askalon

Askalan was developed at the University of Innsbruck [481]. It is application development as well as a runtime environment. It allows easy execution of distributed work flow applications in service oriented grids. It uses a Service Oriented Architecture. Also, for its Grid middleware it uses the Globus Toolkit. The work flow applications are developed using Abstract Grid Work flow Language (AGWL). The architecture has various components like the resource broker responsible for brokerage functions like management and reservation, information service for the discovery and organization of resources and data, metascheduler for mapping in the Grid, performance analysis for unification of performance monitoring and integration of the results and the Askalon scheduler.

The Metascheduler is of special significance since it consists of two major components - the workflow converter and the scheduling engine. The former is responsible for conversion of traditional workflows into directed acyclic graphs (DAGs) while the later one is responsible for the scheduling of workflows for various specific tasks. It has a conventional pluggable architecture which allows easy integration of various services. By default, the Heterogeneous Earliest Finish Time (HEFT) is used as the primary scheduling algorithm.

11.22.16 Apache Ant

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks. Ant is written in Java. Users of Ant can develop their own “antlibs” containing Ant tasks and types, and are offered a large number of ready-made commercial or open-source “antlibs”. Ant is extremely flexible and does not impose coding conventions or directory layouts to the Java projects which adopt it as a build

tool. Software development projects looking for a solution combining build tool and dependency management can use Ant in combination with Apache Ivy. The Apache Ant project is part of the Apache Software Foundation [588].

11.22.17 LXD

LXD is a demon processes established to manage the containers. It can be understood as hypervisor for linux containers. It is implemented by exporting RESTful API for libxl to the remote network or local unix socket. [629]. It implements the under privileged conatiners by default adding more security. It works with Image based work flow supports online snapshopping and live container migration. [381].It was build with aim of providing VM like virtulization with container like performance. [599]

11.22.18 Wink

Apache wink [60] provides a framework to develop and use RESTful web services. It implements using JAX-RS v1.1 specification. The project provides server module which integrates with all popular web servers and a client module which can used to write RESTful web services. This project will be integrated with Geronimo and other opensource REST projects to build a vendor neutral community. Currently IBM and HP have taken lead. IBM is writing a full JAX-RS implementation while HP is working on RESTful SDK for client and server components. Portion of initial project was also taken from Apache CXF which uses other Apache components like commons-codec, commons-logging, Apache-Abdera. Apache wink will simply web services development using one single standard.

11.22.19 Apache Apex

Apache Apex is “a YARN(Hadoop 2.0)-native platform that unifies cloud and batch processing” [661].This project was developed under Apache License 2.0 and was driven by Data Torrent. It can be used for processing both streams of data and static files making it more relevant in the context of present day internet and social media. It is aimed at leveraging the present Hadoop platform and reducing the learning curve for development of applications over it. It is aimed at It can used through a simple API. It enables reuse of code by not having to make drastic changes to the applications by providing interoperability with existing technology stack. It leverages the existing Hadoop platform investments.

Apart from the Apex core component, it also has Apex Malhar which provides a library of connectors and logic functions. It provides connectors to existing file systems, message systems and relational, NoSQL and Hadoop databases, social media. It also provides a library of compute operators like Machine Learning, Stats and Math, Pattern Matching, Query and Scripting, Stream manipulators, Parsers and UI & Charting operators [344].

11.22.20 Apache Knox

According to [44], “the Apache Knox Gateway is a REST API Gateway for interacting with Apache Hadoop clusters.” REST stands for Representational State Transfer and is web architectural style designed for distributed hypermedia systems and defines a set of constraints. [187] API Gateways manage concerns related to “Authentication, Transport Security, Load-balancing, Request Dispatching (including fault tolerance and service discovery), Dependency Resolution, Transport Transformations.” [468] Although every Apache Hadoop cluster has its own set of REST APIs,

Knox will represent all of them as “a single cluster specific application context path.” [44] Knox protects Apache Hadoop clusters, by way of its gateway function, by aiding “the control, integration, monitoring and automation of critical administrative and analytical needs.” [44] Some Apache Hadoop Services that integrate with Knox are, “Ambari, WebHDFS (HDFS), Templeton (Hcatalog), Stargate (Hbase), Oozie, Hive/JDBC, Yarn RM, [and] Storm.” [44] Apache Knox has a configuration driven method to aid in the addition of new routing services. [44] This allows support for new and custom Apache Hadoop REST APIs to be added to the Knox gateway quickly and easily. [44] This technology would be best placed under the interoperability category.

11.22.21 Apache Apex

The Apex platform is designed to process real-time events with streaming data natively in Hadoop. The platform handles application execution, dynamic scaling, state checkpointing and recovery, etc. This allows the users to focus on writing their application logic without mixing operational and functional concerns [38]. In the platform, building a streaming application is easy and intuitive.

An application may consist of one or more operators each of which define some logical operation to be done on the tuples arriving at the operator. These operators are connected together to form streams. A streaming application is represented by a DAG that consists of operators and streams [39]. The Apex platform comes with support for web services and metrics. This enables ease of use and easy integration with current data pipeline components. DevOps teams can monitor data in action using existing systems and dashboards with minimal changes, thereby easily integrating with the current setup. With different connectors and the ease of adding more connectors, Apex easily integrates with an existing dataflow [114].

11.22.22 Robot Operating System (ROS)

The aptly-named *Robot Operating System*, or ROS, provides a framework for writing operating systems for robots. ROS offers “a collection of tools, libraries, and conventions [meant to] simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms” [443]. ROS’ designers, the Open Source Robotics Foundation, hereinafter OSRF or the Foundation, attempt to meet the aforementioned objective by implementing ROS as a modular system. That is, ROS offers a core set of features, such as inter-process communication, that work with or without pre-existing, self-contained components for other tasks.

The OSRF designed ROS as a distributed, modular system. The OSRF maintains a subset of essential features for ROS, i.e., *ROS core*, to provide an extensible platform for other roboticists. The Foundation also coordinates the maintenance and distribution of a vast array of ROS add-ons, referred to as modules. ROS’ core consists of the following components: a) communications infrastructure; b) robot-specific features; and, c) tools. The modules, analogous to packages in Linux repositories or libraries in other software packages such as *R*, provide solutions for numerous robot-related problems. General categories include a) drivers, such as sensor and actuator interfaces; b) platforms, for steering and image processing, etc.; c) algorithms, for task planning and obstacle avoidance; and, d) user interfaces, such as tele-operation and sensor data display [416].

11.22.23 Apache Flex

Apache Flex [192] is an open source application framework for building and maintaining mobile and web applications that deploy consistently on multiple browsers, desktops and mobile devices. It was initially developed by Macromedia and then acquired by Adobe Systems. It was later donated to the

Apache Software Foundation in 2011 [193]. It can pull data from multiple back-end sources such as Java, Spring, PHP, Ruby, .NET, Adobe ColdFusion, and SAP and display it visually allowing users to drill down into the data for deeper insight and even change the data and have it automatically updated on the back end [657].

11.22.24 Apache Ranger

Apache Ranger [704] is open source software project designed to provide centralized security services to various components of Apache Hadoop. Apache Hadoop provides various mechanism to store, process and access the data. Each Apache tool has its own security mechanism. This increases administrative overhead and is also error prone. Apache Ranger fills this gap to provide a central security and auditing mechanism for various Hadoop components [717]. Using Ranger, Hadoop administrators can perform security administration tasks using a central UI or Restful web services. He can define policies which enable users/user-groups to perform specific action using Hadoop components and tools. Ranger provides role based access control for datasets on Hadoop at column and row level. The blog article [718] explains that the row level filtering and dynamic data masking are most important features of Apache Ranger. Ranger also provides centralized auditing of user acces and security related administrative actions.

11.22.25 Google Cloud Machine Learning

Google Could Machine Learning is a Googles cloud based managed system for building machine learning model, capable to work on any type and volume of data. User can create their own machine learning model using GoogleTensorFlow framework, which helps to use the range of Google products from Google Photos to Google Cloud Speech. We can build our machine learning model regardless the size, google will managed it infrastructure according to requirement. User can immediately host the created model and start predicting on new data [248].Cloud Machine Learning provides two important things:

- Help user to train the machine learning model at large scale with the help of TensorFlow training application.
- User can host the trained model on cloud, this will help to use the large and new data available on cloud, which help in creating good model.

Google CloudML will help user to focus on model instead of hardware configuration and resource management [249].

11.22.26 Karajan

Karajan is used to allow users to describe various workflows using XML [482]. It also uses a custom yet user friendly language called K. The advantages of using XML and K is that we can use Directed Acyclic Graphs (DAGs) to describe hierarchical workflows. Besides, it is also very easy to handle concurrency using trivial programming constructs like if/while orders. It can also use tools such as Globus GRAM for parallel or distributed execution of various workflows. From an architectural perspective, Karajan mainly consists of three components: Workflow engine, that monitors the execution and is responsible for the higher level interaction with higher level components like the Graphical User Interface Module (GUI) for the description of various workflows; Workflow service, that is used to allow the execution of various workflows using specific functionalities that can be accessed by the workflow engine using specific libraries; and the Checkpointing subsystem that monitors and checks the current state of the workflow. Karajan is typically used as a scientific

workflow scheduling technique for various Big Data platforms.

The Karajan code, that can be obtained from Java CoG Kit CVS archive has two interfaces: the command line interface (CLI) and the GUI. The CLI can be accessed via bin/karajan and provides a minimalist interface that is non-interactive and doesn't provide much feedback on the execution status. As against this, the GUI can be accessed via bin/karajan-gui and provides an enriched interface that provides visual features to determine the execution status besides being interactive in real time [367].

11.23 Excercise

TechList.1: In class you will be given an HID and you will be assigned a number of technologies that you need to research and create a summary as well as one or more relevant references to be added to the Web page. All technologies for TechList.1 are marked with a (1) behind the technology. An example text is given for Nagios in this page. Please create a pull request with your responses. You are responsible for making sure the request shows up and each commit is using gitchangelog in the commit message::

new:usr: added paragraph about <PUTTECHHERE>

You can create one or more pull requests for the technology and the references. We have created in the referens file a placeholder using your HID to simplify the management of the references while avoiding conflicts. For the technologies you are responsible to investigate them and write an academic summary of the technology. Make sure to add your reference to refs.bib. Many technologies may have additional references than the Web page. Please add the most important once while limiting it to three if you can. Avoid plagiarism and use proper quotations or better rewrite the text.

You must look at :doc:‘technologies-hw’ to successfully complete the homework

A video about this homework is posted at <https://www.youtube.com/watch?v=roi7vezNmfo> showing how to do references in emacs and jabref, it shows you how to configure git, it shows you how to do the fork request while asking you to add “new:usr” to the commit messages). As this is a homework related video we put a lot of information in it that is not only useful for beginners. We recommend you watch it.

This homework can be done in steps. First you can collect all the content in an editor. Second you can create a fork. Third you can add the new content to the fork. Fourth you can commit. Fifth you can push. Sixth if the TAs have commend improve. The commit message must have new:usr: at the beginning.

While the Nagios entry is a good example (make sure grammar is ok the Google app engine is an example for a bad entry).

Do Techlist 1.a 1.b 1.c first. We will assign Techlist 1.d and TechList 2 in February.

TechList.1.a: Complete the pull request with the technologies assigned to you. Details for the assignment are posted in Piazza. Search for TechList.

TechList.1.b: Identify how to cite. We are using *scientific* citation formats such as IEEEtran, and ACM. We are **not** using citation formats such as Chicago, MLA, or ALP. The later are all for non scientific publications and thus of no use to us. Also when writing about a technology do not use the names of the person, simply say something like. In [1] the definition of a turing machine is given as follows, ... and do not use elaborate sentences such as: In his groundbreaking work conducted in England, Allan Turing, introduced the turing machine in the years 1936-37 [2]. Its definition is based on ... The difference is clear, while the first focusses on results and technological concepts, the second introduces a colorful description that is more suitable for a magazine or a computer history paper.

TechList 1.c: Learn about plagiarism and how to avoid it. Many Web pages will conduct self

advertisement while adding suspicious and subjective adjectives or phrases such as cheaper, superior, best, most important, with no equal, and others that you may not want to copy into your descriptions. Please focus on facts, not on what the author of the Web page claims.

TechList 1.d: Identify technologies from the Apache Project <https://projects.apache.org/> or other Big Data related Web pages and projects that are not yet listed here. Add them at the end of the Technologies page under the :ref:`New Technologies <new-techs>` section, together with a description and appropriate references just like you did for your list of technologies in TechList 1a-1c. As part of your paragraph, please suggest a section where you think is best to add the technologies. Once the new technologies have been submitted, the AIs will integrate them in the appropriate sections. Please, only add new techs to the last section, otherwise it will be easy to introduce conflicts in the file.

TechList.2: In this hopweork we provide you with additional technologies that you need to complete. They are marked with (2) in the :doc:`HID Assignment page <hids-techs>`.

TechList.3: Identify technologies that are not listed here and add them. Provide a description and a reference just as you did before. Before you add a technology, verify that it is not on the **new technologies** list already. Duplicated entries will be merged.

TechList.4: For useful information on how to correctly create BibTeX entries, see and contribute to :ref:`these open discussion threads Piazza <bibtex-discussions>`.

Appendix



Bibliography	307
Index	343



Bibliography

References

- [1] web page. accessed 2017-02-13. URL: <http://opencv.org/> (cited on page 205).
- [2] web page. accessed 2017-02-13. URL: <http://opencv.org/opencv-3-2.html> (cited on page 205).
- [3] *5 Things to Know about dashDB*. web page. URL: https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_dashdb_placeholder?lang=en (cited on page 252).
- [4] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Technical report. Accessed: 2017-1-24. Cornell University, Mar. 2016 (cited on page 216).
- [5] *About OpenNebula*. URL: <https://opennebula.org/about/technology/> (cited on page 287).
- [6] *About Pivotal Gemfire*. Web Page. Version 9.0.1. Accessed: 2017-01-28. URL: http://gemfire.docs.pivotal.io/gemfire/getting_started/gemfire_overview.html (cited on page 256).
- [7] Abed Abu-Dbai et al. “Enterprise Resource Management in Mesos Clusters”. In: *Proceedings of the 9th ACM International on Systems and Storage Conference*. SYSTOR ’16. Haifa, Israel: ACM, 2016, 17:1–17:1. ISBN: 978-1-4503-4381-7. DOI: 10.1145/2928275.2933272. URL: <http://doi.acm.org/10.1145/2928275.2933272> (cited on page 266).
- [8] ACADGILD. *Beginner’s Guide for Impala*. Web page. Online; accessed 7-Apr-2017. Mar. 2016. URL: <https://acadgild.com/blog/beginners-guide-impala/> (cited on page 224).
- [9] ACM, Inc. *Spark SQL: Relational Data Processing in Spark*. Web Page. accessed 2017-02-19. Feb. 2016. URL: <http://dl.acm.org/citation.cfm?id=2742797> (cited on page 252).
- [10] ActiveBPEL. *Communicating with the ActiveBPEL Server Administration Interface via Web Services*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: <http://www.activevos.com/>

- com/content/developers/education/sample_active_bpel_admin_api/doc/index.html (cited on page 196).
- [11] *ActiveMQ technology*. webpage. accessed: 2017-2-4. URL: <http://activemq.apache.org/> (cited on page 238).
 - [12] Aduna. *RDF components*. Web Page. Accessed: 2017-1-25. URL: <https://www.w3.org/RDF/> (cited on page 261).
 - [13] Aduna. *sesame components*. Web Page. Accessed: 2017-1-26. URL: <https://projects.eclipse.org/projects/technology.rdf4j> (cited on page 261).
 - [14] Aerobatic. *Aerobatic - Overview*. Web Page. accessed: 2017-01-25. Jan. 2017. URL: <https://www.aerobatic.com/docs/overview/> (cited on page 217).
 - [15] *Agave API Home – Features Tab*. webpage. Accessed : 02-04-2017. URL: <https://agaveapi.co/platform/features/> (cited on page 221).
 - [16] A. Ailijiang, A. Charapko, and M. Demirbas. “Consensus in the Cloud: Paxos Systems Demystified”. In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. Aug. 2016, pages 1–10. DOI: 10.1109/ICCCN.2016.7568499. URL: <http://ieeexplore.ieee.org/abstract/document/7568499/> (cited on page 294).
 - [17] Tyler Akidau et al. “MillWheel: Fault-Tolerant Stream Processing at Internet Scale”. In: *Very Large Data Bases*. 2013, pages 734–746 (cited on page 230).
 - [18] Davide Albanese et al. “mlpy: Machine Learning Python”. In: *CoRR* abs/1202.6548 (2012). URL: <http://arxiv.org/abs/1202.6548> (cited on page 207).
 - [19] *Allegro*. Web Page. Accessed: 2017-1-25. URL: <http://allegrograph.com/> (cited on page 260).
 - [20] *Allegrow*. Web Page. Accessed: 2017-1-20. URL: <https://en.wikipedia.org/wiki/AllegroGraph> (cited on page 260).
 - [21] Alphabet, Inc. *szl - Overview.wiki*. Code Repository. Online; accessed 30-jan-2017. URL: <https://code.google.com/archive/p/szl/wikis/Overview.wiki> (cited on page 228).
 - [22] Alphabet, Inc. Web Page. Online; accessed 25-jan-2017. Jan. 2017. URL: %7Bhttps://cloud.google.com%7D (cited on page 218).
 - [23] Amazon. *AWS OpsWorks*. Web Page. accessed 2017-01-25. URL: <https://aws.amazon.com/opsworks/> (cited on page 281).
 - [24] *Amazon Route 53*. Web Page. Accessed: 2017-02-24. URL: https://en.wikipedia.org/wiki/Amazon_Route_53 (cited on page 290).
 - [25] *Amazon S3*. Web Page. Accessed: 2017-1-27. URL: <https://aws.amazon.com/s3/> (cited on page 273).
 - [26] Amazon Web services. *Amazon Redshift Management Services*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: <http://docs.aws.amazon.com/redshift/latest/mgmt/overview.html> (cited on page 226).
 - [27] Amazon Web Services,Inc. *AWS Elastic Beanstalk*. Web Page. accessed 2017-2-14. URL: <https://aws.amazon.com/elasticbeanstalk/> (cited on page 218).
 - [28] Amazon.com, Inc. Web Page. Online; accessed 25-jan-2017. Jan. 2017. URL: %7Bhttps://aws.amazon.com%7D (cited on page 218).
 - [29] *Ambari*. Web Page. Accessed: 2017-2-04. URL: <https://ambari.apache.org/> (cited on page 291).
 - [30] *Github apache/ambari*. Web Page. Accessed: 2017-2-04. URL: <https://github.com/apache/ambari/> (cited on page 291).
 - [31] *Hortonworks Apache Ambari*. Web Page. Accessed: 2017-2-04. URL: <http://hortonworks.com/apache/ambari/> (cited on page 291).

-
- [32] AMQP. *AMQP is the Internet Protocol for Business Messaging*. Web Page. accessed: 2017-01-31. Jan. 2017. URL: <http://www.amqp.org/about/what> (cited on page 240).
 - [33] *An Introduction to Windows Azure BLOB Storage*. Web Page. July 2013. URL: <https://www.simple-talk.com/cloud/cloud-data/an-introduction-to-windows-azure-blob-storage/> (visited on 02/13/2017) (cited on pages 273, 274).
 - [34] *Any Analytics, Any Data, Simplified*. webpage. URL: <http://www.pentaho.com/product/product-overview> (cited on page 203).
 - [35] Apache. web-page. accessed 2017-02-13. URL: <https://hama.apache.org/> (cited on page 235).
 - [36] Apache. *Apache arrow*. Webpage. URL: <http://arrow.apache.org/> (cited on page 298).
 - [37] *Apache Accumulo User Manual Version 1.8*. accessed 2017-02-26. URL: https://accumulo.apache.org/1.8/accumulo_user_manual.html (cited on page 258).
 - [38] Apache Apex. *Application Developer Guide*. Web-page. URL: https://apex.apache.org/docs/apex/application_development/#application-developer-guide (cited on page 301).
 - [39] Apache Apex. *Operator Development Guide*. Web-page. accessed 2017-02-27. URL: https://apex.apache.org/docs/apex/operator_development/ (cited on page 301).
 - [40] Apache Avro. webpage. accessed: 2017-2-6. URL: <https://avro.apache.org/docs/current/> (cited on page 295).
 - [41] Apache Cassandra. Web Page. 2016. URL: <http://cassandra.apache.org/> (cited on page 258).
 - [42] Apache Giraph. web page. URL: <https://giraph.apache.org> (cited on page 235).
 - [43] Apache Giraph Wiki. web page. URL: https://en.wikipedia.org/wiki/Apache_Giraph (cited on page 235).
 - [44] Apache Knox. *Apache Knox Home*. Website. Feb. 2017. URL: <https://knox.apache.org/> (cited on pages 300, 301).
 - [45] *Apache License, Version 2.0*. Web Page. Online; accessed 23-Feb-2017. URL: <https://www.apache.org/licenses/LICENSE-2.0> (cited on page 287).
 - [46] Apache Lucene. Web Page. Jan. 2017. URL: <http://lucene.apache.org/> (cited on page 252).
 - [47] *Apache Nifi (aka HDF) data flow across data center*. Web Page. URL: <https://hortonworks.com/articles/9933/apache-nifi-aka-hdf-data-flow-across-data-center.html> (cited on page 202).
 - [48] *S4: Distributed Stream Computing Platform*. Web Page. Accessed: 2017-02-24. URL: <http://incubator.apache.org/s4/> (cited on page 229).
 - [49] Apache Samza. en. Web Page. Page Version ID: 764035647. Feb. 2017. URL: https://en.wikipedia.org/w/index.php?title=Apache_Samza&oldid=764035647 (visited on 02/13/2017) (cited on pages 229, 230).
 - [50] Apache Samza, LinkedIn's Framework for Stream Processing. Web Page. Jan. 2015. URL: <https://thenewstack.io/apache-samza-linkedin-s-framework-for-stream-processing/> (visited on 02/13/2017) (cited on page 230).
 - [51] Apache Software Foundation. *Apache Incubator*. Web Page. accessed 2017-01-29. URL: <http://incubator.apache.org/> (cited on pages 224, 299).
 - [52] Apache Software Foundation. *Apache MRQL*. Web Page. accessed 2017-01-29. Apr. 2016. URL: <https://mrql.incubator.apache.org/> (cited on page 224).
 - [53] Apache Software Foundation. *Omid Project Incubation Status*. Web Page. accessed 2017-02-25. Apr. 2016. URL: <https://mrql.incubator.apache.org/> (cited on page 299).

- [54] Apache Software Foundation. *1.1. Document Storage*. Web page. accessed 26-feb-2017. Feb. 2017. URL: <http://docs.couchdb.org/en/stable/intro/overview.html> (cited on page 255).
- [55] Apache Software Foundation. *Apache Phoenix: OLTP and operational analytics for Apache Hadoop*. Web page. Online; accessed 25-jan-2017. Jan. 2017. URL: <http://phoenix.apache.org/> (cited on page 223).
- [56] Apache Software Foundation. *Apache Spark GraphX*. Web Page. accessed 2017-01-30. Feb. 2017. URL: <http://spark.apache.org/graphx/> (cited on page 210).
- [57] *Apache Tajo*. Web Page. Accessed: 2017-1-27. URL: <https://tajo.apache.org> (cited on page 222).
- [58] *Apache Tajo Quick Guide*. Web Page. Accessed: 2017-1-25. URL: https://www.tutorialspoint.com/apache_tajo/apache_tajo_quick_guide.htm (cited on page 222).
- [59] Apache Tinker Pop Home. *Apache TinkerPop Home*. Web Page. 2016. URL: <https://tinkerpop.apache.org/> (cited on page 212).
- [60] *Apache Wink*. Web Page. Accessed: 2017-02-24. URL: <http://wink.apache.org/index.html> (cited on page 300).
- [61] *Apache Hbase*. Web Page. Accessed: 2017-1-26. URL: <https://hbase.apache.org/> (cited on page 257).
- [62] ApacheTinkerPopDoc. *Apache TinkerPop*. Web Page. 2016. URL: <http://tinkerpop.apache.org/docs/3.1.1-incubating/tutorials/getting-started/> (cited on page 211).
- [63] *Apache Sentry Tutorial*. Web Page. Accessed: 2017-2-11. URL: <https://cwiki.apache.org/confluence/display/SENTRY/Sentry+Tutorial> (cited on page 293).
- [64] *Apache NiFi*. Web Page. URL: <https://nifi.apache.org> (cited on page 202).
- [65] *AppEngine - platform as a service*. webpage. Accessed : 02-22-2017. URL: <https://cloud.google.com/appengine> (cited on page 216).
- [66] Wikipedia. *Google App Engine*. webpage. Accessed : 02-22-2017. Jan. 2017. URL: https://en.wikipedia.org/wiki/Google_App_Engine (cited on page 217).
- [67] appfog. *Overview*. Online. URL: <https://www.ctl.io/appfog/#Overview> (cited on page 219).
- [68] AppScale. *what-is-appscale*. Web Page. 2016. URL: <https://www.appspot.com/community/what-is-appscale/> (cited on page 217).
- [69] Appscale. *why-use-appscale*. Web Page. 2016. URL: <https://www.appspot.com/get-started/deployment-types/> (cited on page 217).
- [70] Thusoo Ashish et al. *Hive – A Petabyte Scale Data Warehouse Using Hadoop*. Paper. Apr. 2010. URL: <http://datamining.uos.ac.kr/wp-content/uploads/2015/07/Hive-A-Petabyte-Scale-Data-Warehouse-Using-Hadoop.pdf> / (cited on page 222).
- [71] Storage Networking Industry Association. *About the SNIA*. Web Page. Accessed: 2017-2-27. URL: <https://www.snia.org/about> (cited on page 276).
- [72] Storage Networking Industry Association. *Cloud Data Management Interface*. 1.1.1. Accessed: 2017-2-27. Storage Networking Industry Association. Colorado Springs, CO, Mar. 2015. URL: https://www.snia.org/sites/default/files/CDMI_Spec_v1.1.1.pdf (cited on page 276).
- [73] Storage Networking Industry Association. *Cloud Data Management Interface*. Web Page. Accessed: 2017-2-27. Mar. 2015. URL: <https://www.snia.org/cdmi> (cited on page 276).
- [74] *at1*. Web Page. Accessed: 2017-1-22. URL: <http://www.cyverse.org/atmosphere> (cited on page 221).

- [75] Atlassian. *eScience Central Overview*. Web Page. Accessed: 2017-1-24. URL: <https://bitbucket.org/digitalinstitute/esciencecentral/> (cited on page 201).
- [76] Abel Avram. *Phoenix: Running SQL Queries on Apache HBase [Updated]*. Web page. Online; accessed 25-jan-2017. Jan. 2013. URL: <https://www.infoq.com/news/2013/01/Phoenix-HBase-SQL> (cited on page 223).
- [77] Amazon. *AWSLambda*. Web Page. Accessed: 2/18/2017. URL: <https://aws.amazon.com/lambda/faqs/> (cited on page 241).
- [78] Amazon. *AWSLambdaEvent*. Web Page. Accessed: 2/18/2017. URL: <http://docs.aws.amazon.com/lambda/latest/dg/invoking-lambda-function.html#intro-core-components-event-sources> (cited on page 241).
- [79] *Get started with Azure Queue storage using .NET*. Web Page. Accessed: 2017-2-10. URL: <https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-how-to-use-queues> (cited on page 242).
- [80] *Github Azure/ azure-stream-analytics*. Web Page. Accessed: 2017-2-03. URL: <https://github.com/Azure/azure-stream-analytics/> (cited on page 232).
- [81] *SQL Server Azure*. Web Page. URL: <https://azure.microsoft.com/en-us/services/sql-database/?b=16.50> (cited on page 248).
- [82] *Microsoft Azure Real-time data analytics*. Web Page. Accessed: 2017-2-03. URL: <https://azure.microsoft.com/en-us/services/stream-analytics/> (cited on page 232).
- [83] Vasavi*1 B et al. *HIBERNATE TECHNOLOGY FOR AN EFFICIENT BUSINESS APPLICATION EXTENSION*: Paper. June 2011. URL: <https://www.rroij.com/open-access/hibernate-technology-for-an-efficient-business-application-extension-118-125.pdf> (cited on page 245).
- [84] *Background*. web-page. accessed 2017-02-13. URL: <https://orc.apache.org/docs/> (cited on page 264).
- [85] *Riak-KV - NOSQL Key Value Database*. Web Page. Accessed: 2017-01-20. URL: <http://basho.com/products/riak-kv/> (cited on page 253).
- [86] *Riak-TS - NOSQL Time Series Database*. Web Page. Accessed: 2017-01-20. URL: <http://basho.com/products/riak-ts/> (cited on page 253).
- [87] *Riak-S2 - Cloud Object Storage Software*. Web Page. Accessed: 2017-01-20. URL: <http://basho.com/products/riak-s2/> (cited on page 253).
- [88] Kent Baxley, JD la Rosa, and Mark Wenning. “Deploying workloads with Juju and MAAS in Ubuntu 14.04 LTS”. In: *Deploying workloads with Juju and MAAS in Ubuntu 14.04 LTS*. Dell Inc, Technical White Paper, May 2014 (cited on page 279).
- [89] *Berkeley DB Tutorial and Reference Guide*. Web Page. Accessed: 2017-2-11. URL: <https://web.stanford.edu/class/cs276a/projects/docs/berkeleydb/reftoc.html> (cited on page 254).
- [90] *Berkeley DB Wiki*. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Berkeley_DB (cited on page 254).
- [91] Antonio Esposito Beniamino Di Martino Giuseppina Cretella. *Cloud Portability and Interoperability*. illustrated. New York City, USA: Springer International Publishing, 2015. ISBN: 331913700X, 9783319137001 (cited on page 275).
- [92] *What is BigQuery? Google Cloud Platform*. Web Page. Accessed: 2017-2-23. URL: <https://cloud.google.com/bigquery> (cited on page 226).
- [93] *What is BigQuery? BigQuery Documentation Google Cloud Platform*. Web Page. Accessed: 2017-2-23. URL: <https://cloud.google.com/bigquery/what-is-bigquery> (cited on page 226).
- [94] Tobias Binz et al. *OpenTOSCA – A Runtime for TOSCA-Based Cloud Applications*. Edited by Samik Basu et al. Springer Berlin Heidelberg, 2013, pages 692–695. ISBN: 978-3-642-

- 45005-1. URL: http://dx.doi.org/10.1007/978-3-642-45005-1_62 (cited on page 282).
- [95] *About Bioconductor*. Web Page. Accessed: 2017-02-10. URL: <https://www.bioconductor.org/about/> (cited on page 205).
- [96] bioKepler. *Demo Workflow*. WebPage. URL: <http://www.biokepler.org/userguide#demos> (cited on page 198).
- [97] bioKepler. *What is bioKepler*. WebPage. URL: <http://www.biokepler.org/faq#what-is-biokepler> (cited on page 197).
- [98] Bipin. *Difference between vSphere, ESXi and vCenter*. Webpage. Aug. 2012. URL: <http://www.mustbegeek.com/difference-between-vsphere-esxi-and-vcenter/> (cited on page 289).
- [99] BITTORRENT. Web Page. 2017. URL: <https://www.lifewire.com/how-torrent-downloading-works-2483513> (cited on page 264).
- [100] Oscar Boykin et al. *Summingbird*. Website. README.md, github. URL: <https://github.com/twitter/summingbird> (cited on page 228).
- [101] Oscar Boykin et al. “Summingbird: A framework for integrating batch and online map-reduce computations”. In: *Proceedings of the VLDB Endowment* 7.13 (2014), pages 1441–1451 (cited on page 228).
- [102] *Business Process Execution Language*. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Business_Process_Execution_Language (cited on page 195).
- [103] Mikio L. Braun. *Magastore, Spanner - distributed databases*. Web Page. Accessed: 2017-01-28. Nov. 2013. URL: <http://blog.mikiobraun.de/2013/03/more-google-papers-megastore-spanner-voted-commits.html> (cited on page 258).
- [104] Martin Brown. *The Technology Behind Couchbase*. Web page. Online; accessed 29-jan-2017. Mar. 2012. URL: <https://www.safaribooksonline.com/blog/2012/03/01/the-technology-behind-couchbase/> (cited on page 256).
- [105] Jason Brownlee. *A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library*. webpage. Apr. 16, 2014. URL: <http://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/> (cited on page 207).
- [106] Alfredo Buttari et al. “A class of parallel tiled linear algebra algorithms for multicore architectures”. In: *Parallel Computing* 35.1 (2009), pages 38–53. URL: <http://www.sciencedirect.com/science/article/pii/S0167819108001117> (visited on 02/13/2017) (cited on page 206).
- [107] Byung-Gon Chun. *REEFProposal - Incubator*. Web Page. accessed 2017-01-28. Aug. 2014. URL: <https://wiki.apache.org/incubator/ReefProposal> (cited on page 234).
- [108] AT&T Laboratories Cambridge. *The Medusa Applications Environment*. Web page. Last accessed: 2017.02.25. URL: <http://www.cl.cam.ac.uk/research/dtg/attarchive/medusa.html> (cited on page 236).
- [109] CASCADING. Web Page. 2017. URL: <http://www.cascading.org/projects/cascading/> (cited on page 200).
- [110] Cedar stack. Web Page. Accessed: 1/27/2017. URL: <https://devcenter.heroku.com/articles/stack#cedar> (cited on page 217).
- [111] Celery. Webpage. URL: <http://www.celeryproject.org/> (cited on page 268).
- [112] Celery - Distributed Task Queue. Web Page. URL: <http://docs.celeryproject.org/en/latest/index.html> (cited on page 268).
- [113] Craig Chambers et al. “FlumeJava: Easy, Efficient Data-Parallel Pipelines”. In: *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. Ac-

- cessed : 04-09-2017. 2 Penn Plaza, Suite 701 New York, NY 10121-0701, 2010, pages 363–375. URL: <http://dl.acm.org/citation.cfm?id=1806638> (cited on page 200).
- [114] Desmond Chan. *Big Data with Apache Apex*. Web-page. accessed 2017-02-27. Jan. 2016. URL: <https://jaxenter.com/big-data-apache-apex-122839.html> (cited on page 301).
- [115] *Chef Commercial Support*. Web Page. URL: <https://www.chef.io/support/> (cited on page 278).
- [116] Guoqiang Jerry Chen et al. “Realtime Data Processing at Facebook”. In: *Proceedings of the 2016 International Conference on Management of Data*. SIGMOD ’16. San Francisco, California, USA: ACM, 2016, pages 1087–1098. ISBN: 978-1-4503-3531-7. DOI: 10.1145/2882903.2904441. URL: <http://doi.acm.org/10.1145/2882903.2904441> (cited on page 231).
- [117] Rudi Cilibrasi et al. *What is CompLearn*. webpage. URL: <http://complearn.org/> (cited on page 208).
- [118] *Cinder - Openstack*. Web Page. Accessed: 2017-1-21. URL: <https://wiki.openstack.org/wiki/Cinder> (cited on page 271).
- [119] *CINET - CyberInfrastructure for Network Science*. Web Page. URL: www.bi.vt.edu (cited on page 212).
- [120] Tait Clarridge. *Disco - A Powerful Erlang and Python Map/Reduce Framework*. Blog. accessed 25-feb-2017. May 2014. URL: <http://www.taitclarridge.com/techlog/2014/05/disco-a-powerful-erlang-and-python-mapreduce-framework.html> (cited on page 234).
- [121] *Cloud and systems management*. webpage. URL: <http://www.cisco.com/c/en/us/products/cloud-systems-management> (cited on page 280).
- [122] Vineet Badola. *Cloud Foundry Blog*. Blog. Sept. 2015. URL: <http://cloudacademy.com/blog/cloud-foundry-benefits/> (cited on page 218).
- [123] Cloudability Inc. *Cloudability Cost Management Tools | Cloudability*. Web Page. Accessed: 2017-02-23. Feb. 2017. URL: <https://www.cloudability.com/product/> (cited on page 297).
- [124] *Cloudbees Wikipedia Documentation*. Web Page. Accessed: 2017-02-13. URL: <https://en.wikipedia.org/wiki/CloudBees> (cited on page 220).
- [125] *Cloudbees Webpage Documentation*. Web Page. Accessed: 2017-02-13. URL: <https://www.cloudbees.com/products> (cited on page 220).
- [126] *OpenStack Keystone Verification*. Web Page. Accessed: 2017-2-12. URL: <https://www.cloudberrylab.com/blog/openstack-keystone-authentication-explained/> (cited on page 292).
- [127] cloudera. *Untangling Apache Hadoop YARN Part 1 Cluster and YARN Basics*. Web Page. 2015. URL: <https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/> (cited on page 267).
- [128] Cloudera Inc. *Cloudera Impala Overview*. Web page. Online; accessed 7-Apr-2017. URL: https://www.cloudera.com/documentation/enterprise/5-3-x/topics/impala_intro.html (cited on page 224).
- [129] *CNTK/CNTKBook-20160217.pdf at master · Microsoft/CNTK · GitHub*. Web Page. URL: <https://github.com/Microsoft/CNTK/blob/master/Documentation/CNTK-TechReport/lyx/CNTKBook-20160217..pdf> (visited on 02/13/2017) (cited on page 216).
- [130] Jeffrey Ira Cohen, Luke Lonergan, and Caleb E. WeltonCohen. *Integrating map-reduce into a distributed relational database*. grant. Dec. 2016. URL: <https://www.google.com/patents/US9514188> (cited on page 266).

- [131] Community Grids Lab IU. *Some of the salient features in naradabrokering*. Web Page. Accessed: 2017-2-15. 501 N. MORTON ST, SUITE 224 BLOOMINGTON IN 47404: Pervasive Technology Labs at Indiana University, Nov. 2009. URL: <http://www.naradabrokering.org/> (cited on page 239).
- [132] Community Grids Lab IU. *The NaradaBrokering Project @ IU Community Grids Laboratory*. Web Page. Accessed: 2017-2-15. 501 N. MORTON ST, SUITE 224 BLOOMINGTON IN 47404: Pervasive Technology Labs at Indiana University, Nov. 2009. URL: <http://www.naradabrokering.org/> (cited on page 239).
- [133] Continuum Analytics. *Blaze*. Web page. accessed 25-feb-2017. 2015. URL: <http://blaze.readthedocs.io/en/latest/index.html#> (cited on page 297).
- [134] Darren Cook. *Practical Machine Learning with H2O*. O'Reilly Media, Dec. 2017, page 300. ISBN: 978-1-4919-6454-5. URL: <https://books.google.com/books?id=nJWmDQAAQBAJ&pg=PP2&dq=h2o+software> (cited on page 210).
- [135] Emilio Coppa. *Hadoop Architecture Overview*. Code Repository. accessed 2017-03-23. URL: <http://ercoppa.github.io/HadoopInternals/HadoopArchitectureOverview.html> (cited on page 233).
- [136] James C Corbett et al. "Spanner: Google's globally distributed database". In: *ACM Transactions on Computer Systems (TOCS)* 31.3 (2013), page 8. URL: http://dl.acm.org/ft_gateway.cfm?id=2491245&type=pdf (cited on page 257).
- [137] CoreOS. Web Page. Accessed: 1/27/2017. URL: <https://www.coreos.com/> (cited on page 288).
- [138] CoreOS. *Why CoreOS*. Web Page. accessed: 2017-01-23. Jan. 2017. URL: <https://coreos.com/why/> (cited on page 288).
- [139] iMatix Corporation. *0MQ - The Guide*. Web Page. Accessed: 2017-1-24. URL: <http://zguide.zeromq.org/page:all> (cited on page 238).
- [140] iMatix Corporation. *Distributed Messaging*. Web Page. Accessed: 2017-1-24. URL: <http://zeromq.org> (cited on page 238).
- [141] Couchbase, Inc. *Couchbase and Apache CouchDB compared*. Web page. accessed 26-feb-2017. Feb. 2017. URL: <https://www.couchbase.com/couchbase-vs-couchdb> (cited on page 255).
- [142] Cray Inc. *Cray Yarcdata Urika appliance*. Web Page. 2017. URL: <http://www.cray.com/products/Urika.aspx> (cited on page 259).
- [143] Cray Inc. *Cray Urika-GD Technical Specification*. Technical report. Cray Inc, 2014. URL: <http://www.cray.com/sites/default/files/resources/Urika-GD-TechSpecs.pdf> (cited on page 259).
- [144] *CUBRID*. Web Page. 2017. URL: <http://www.cubrid.org/> (cited on page 249).
- [145] Baojiang Cui and Tao Xi. "Security analysis of openstack keystone". In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2015 9th International Conference on*. IEEE. 2015, pages 283–288 (cited on page 292).
- [146] Curoverse, Inc. *Arvados | Introduction to Crunch*. Web Page. Accessed: 2017-2-22. URL: <http://doc.arvados.org/user/tutorials/intro-crunch.html> (cited on page 200).
- [147] Curoverse, Inc. *Arvados | Open Source Big Data Processing and Bioinformatics*. Web Page. Accessed: 2017-2-22. 2016. URL: <https://arvados.org> (cited on page 200).
- [148] *D3 Data-Driven Documents*. Web Page. Accessed: 2017-02-11. URL: <https://d3js.org/> (cited on page 215).
- [149] *DataFu*. Web Page. Accessed: 1/16/2017. URL: <https://datafu.incubator.apache.org/> (cited on page 204).

- [150] DataNucleus. *DataNucleus Support*. Web Page. Accessed: 2017-02-04. URL: <http://www.datanucleus.com/> (cited on page 246).
- [151] Wikipedia. *IBM DB2-Wikipedia*. Web Page. Online, Accessed: 2017-02-24. Feb. 2017. URL: https://en.wikipedia.org/wiki/IBM_DB2 (cited on page 248).
- [152] *DB2 Introduction*. Web Page. Online, Accessed: 2017-02-24. June 2016. URL: https://www.tutorialspoint.com/db2/db2_introduction.htm (cited on page 248).
- [153] DC.js. *dc.js - Dimensional Charting Javascript Library*. Web Page. accessed: 2017-01-21. Jan. 2017. URL: <https://dc-js.github.io/dc.js/> (cited on page 215).
- [154] John Denero. *CS61A: Online Textbook*. This book is derived from the classic textbook Structure and Interpretation of Computer Programs by Abelson, Sussman, and Sussman. John Denero originally modified it for Python for the Fall 2011 semester. <http://www-inst.eecs.berkeley.edu/cs61a/sp12/book/>. Berkeley, 2011. URL: <http://www-inst.eecs.berkeley.edu/~cs61a/sp12/book/communication.html> (cited on page 212).
- [155] DevTopics. Web Page. URL: <http://www.devtopics.com/kite-obscure-programming-language-of-the-month/> (cited on page 222).
- [156] DeZyre. *How LinkedIn Uses Hadoop To Leverage Big Data Analytics*. Web page. accessed 10-March-2016. This page was last modified on 18 March 2017, at 22:27. URL: <https://www.dezyre.com/article/how-linkedin-uses-hadoop-to-leverage-big-data-analytics/229> (cited on page 231).
- [157] *CDF,Common Data Format (multidimensional datasets)*. Web page. Accessed: 2017-1-28. Mar. 2014. URL: <http://www.digitalpreservation.gov/formats/fdd/fdd000226.shtml#useful> (cited on page 263).
- [158] *Distributed Machine Learning*. Web page. Accessed 25-feb-2017. Feb. 2017. URL: <http://www.mlbase.org/> (cited on page 204).
- [159] *docker,swarm*. WebPage. Accessed: 2017-8-02. URL: <https://www.docker.com/products/docker-swarm> (cited on pages 203, 277).
- [160] *Microsoft Docs Azure Stream Analytics Documentation*. Web Page. Accessed: 2017-2-03. URL: <https://docs.microsoft.com/en-us/azure/stream-analytics/> (cited on page 232).
- [161] Jack Dongarra et al. “Accelerating numerical dense linear algebra calculations with GPUs”. In: *Numerical Computations with GPUs*. Springer, 2014, pages 3–28. URL: http://link.springer.com/chapter/10.1007/978-3-319-06548-9_1 (visited on 02/13/2017) (cited on page 206).
- [162] Dормандо. *Memcached*. Web Page. accessed 2017-01-30. Feb. 2015. URL: <http://www.memcached.org/> (cited on page 243).
- [163] *Apache Drill*. Web Page. Accessed: 2/4/2017. URL: <https://drill.apache.org/> (cited on page 227).
- [164] Wikipedia. *Dryad(Programming)-Wikipedia*. Web Page. Online, Accessed: 2017-02-24. Nov. 2016. URL: [https://en.wikipedia.org/wiki/Dryad_\(programming\)](https://en.wikipedia.org/wiki/Dryad_(programming)) (cited on page 198).
- [165] Michael Isard et al. “Dryad: distributed data-parallel programs from sequential building blocks”. In: *ACM SIGOPS operating systems review*. Volume 41. 3. Online, Accessed: 2017-02-24. ACM. 2007, pages 59–72. URL: <https://www.microsoft.com/en-us/research/wp-content/uploads/2007/03/eurosys07.pdf> (cited on page 199).
- [166] *Rebalancing in a multi-cloud environment in Science Cloud '13*. ACM, 2013, pages 21–28. ISBN: 978-1-4503-1979-9. DOI: 10.1145/2465848.2465854. URL: <http://dl.acm.org/citation.cfm?id=2465854> (cited on page 288).

-
- [167] Dylan Raithel. *Apache TinkerPop Graduates to Top-Level Project*. Web Page. 2016. URL: <https://www.infoq.com/news/2016/06/tinkerpop-top-level-apache/> (cited on page 211).
 - [168] EDUROAM. Web Page. URL: <https://www.eduroam.org/about/> (cited on page 292).
 - [169] UCAR Edward Hartnett and Rew RK. “Experience with an enhanced netCDF data model and interface for scientific data access”. In: *24th Conference on IIPS*. 2008 (cited on page 263).
 - [170] *Ehcache - Features*. Web Page. Accessed: 2017-01-21. URL: <http://www.ehcache.org/about/features.html> (cited on page 244).
 - [171] *Ehcache - Documentation*. Web Page. Accessed: 2017-01-21. URL: <http://www.ehcache.org/documentation/3.2/getting-started.html> (cited on page 244).
 - [172] *Elastic Search*. Web Page. Accessed: 2017-1-25. URL: <https://www.elastic.co/products/elasticsearch> (cited on page 213).
 - [173] *Elastic Search*. Web Page. Accessed: 2017-1-25. URL: <https://en.wikipedia.org/wiki/Elasticsearch> (cited on page 213).
 - [174] *Elastic Search Getting Started*. Web Page. Accessed: 2017-1-25. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html> (cited on page 213).
 - [175] *Elastic Search on Hadoop*. Web Page. Accessed: 2017-1-25. URL: <https://www.elastic.co/products/hadoop> (cited on page 213).
 - [176] *Elasticsearch*. en. Web Page. Page Version ID: 767434249. Feb. 2017. URL: <https://en.wikipedia.org/w/index.php?title=Elasticsearch&oldid=767434249> (visited on 02/27/2017) (cited on page 214).
 - [177] Elasticsearch. *Logstash Introduction*. Web Page. Accessed: 2017-02-11. Feb. 2017. URL: <https://www.elastic.co/guide/en/logstash/current/introduction.html> (cited on page 214).
 - [178] Engle et al. “Shark: Fast Data Analysis Using Coarse-grained Distributed Memory”. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’12. Scottsdale, Arizona, USA: ACM, 2012, pages 689–692. ISBN: 978-1-4503-1247-9. DOI: 10.1145/2213836.2213934. URL: <http://doi.acm.org/10.1145/2213836.2213934> (cited on page 223).
 - [179] Erlang Central. *Couchbase Performance and Scalability: Iterating with DTrace Observability*. Web page. Online; accessed 29-jan-2017. Mar. 2012. URL: <http://erlangcentral.org/videos/couchbase-performance-and-scalability-iterating-with-dtrace-observability/#.WI5uYephnRY> (cited on page 256).
 - [180] Christian Esposito et al. “A Knowledge-based Platform for Big Data Analytics Based on Publish/Subscribe Services and Stream Processing”. In: *Know.-Based Syst.* 79.C (May 2015). Accessed: 2017-1-27, pages 3–17. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2014.05.003. URL: <http://dx.doi.org/10.1016/j.knosys.2014.05.003> (cited on page 237).
 - [181] Patrick Th. Eugster et al. “The Many Faces of Publish/Subscribe”. In: *ACM Comput. Surv.* 35.2 (June 2003). Accessed: 2017-1-27, pages 114–131. ISSN: 0360-0300. DOI: 10.1145/857076.857078. URL: <http://doi.acm.org/10.1145/857076.857078> (cited on page 237).
 - [182] *Exploring Big Data with Helix: Finding Needles in a Big Haystack*. Web Page. URL: https://sigmodrecord.org/publications/sigmodRecord/1412/pdfs/09_industry_Ellis.pdf (cited on page 267).
 - [183] Facebook Inc. *Under the Hood: Scheduling MapReduce jobs more efficiently with Corona*. Web Page. accessed 2017-2-13. Nov. 2012. URL: <https://www.facebook.com/notes/>

- facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920/ (cited on page 268).
- [184] *Facebook's Graph Search puts Apache Giraph on the map.* web page. URL: <https://www.pcworld.com/article/2046680/facebook-graph-search-puts-apache-giraph-on-the-map.html> (cited on page 235).
- [185] *Facebook's New Realtime Analytics System: HBase To Process 20 Billion Events Per Day.* Web Page. Accessed:2017-2-8. Mar. 2011. URL: <http://highscalability.com/blog/2011/3/22/facebook-new-realtime-analytics-system-hbase-to-process-20.html> (cited on page 231).
- [186] *Features.* Web Page. URL: <https://openvz.org/Features> (visited on 02/13/2017) (cited on page 286).
- [187] Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures". Doctoral Dissertation. University of California, Irvine, 2000. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (cited on page 300).
- [188] *FITS Vatican Library.* Web Page. URL: <https://www.vatlib.it/home.php?pag=digitalizzazione&ling=eng> (cited on page 263).
- [189] *Fits Matlab.* Web Page. URL: <https://www.mathworks.com/help/matlab/import-export/importing-flexible-image-transport-system-fits-files.html?requestedDomain=www.mathworks.com> (cited on page 263).
- [190] *FITS Nasa.* web. URL: <https://fits.gsfc.nasa.gov/> (cited on page 263).
- [191] *FITS News.* Web Page. URL: https://fits.gsfc.nasa.gov/fits_standard.html (cited on page 263).
- [192] Apache Software Foundation. *About Apache Flex.* Web Page. Accessed: 2017-03-01. Feb. 2017. URL: <http://flex.apache.org/about-whatis.html> (cited on page 301).
- [193] Joab Jackson. "Adobe donates Flex to Apache". In: *The IDG News Service* (Nov. 2011). Accessed: 2011-11-17. URL: https://www.techworld.com.au/article/407714/adobe_donates_flex_apache (cited on page 302).
- [194] Licia Florio and Klaas Wierenga. "Eduroam, providing mobility for roaming users". In: TERENA (2005). URL: <https://www.terena.org/activities/tf-mobility/docs/ppt/eunis-eduroamfinal-LF.pdf> (cited on page 292).
- [195] Google. *Google App Engine.* webpage. Accessed : 04-09-2017. URL: <https://research.google.com/pubs/pub35650.html> (cited on page 200).
- [196] For Dummies, Inc. *Cloudera Impala and Hadoop.* Web page. Online; accessed 7-Apr-2017. URL: <http://www.dummies.com/programming/big-data/hadoop/cloudera-impala-and-hadoop/> (cited on page 224).
- [197] Ian Foster. "Globus toolkit version 4: Software for service-oriented systems". In: *Journal of computer science and technology* 21.4 (2006), page 513 (cited on page 270).
- [198] *Foundation » OpenStack Open Source Cloud Computing Software.* Web Page. Online; accessed 23-Feb-2017. URL: <https://www.openstack.org/foundation/> (cited on page 287).
- [199] Apache Software Foundation. *Apache Parquet.* Web Page. Accessed: 02-06-2017. URL: <https://parquet.apache.org/documentation/latest> (cited on page 264).
- [200] Apache Software Foundation. *Kafka-A distributed streaming platform.* Web Page. URL: <https://kafka.apache.org/> (cited on page 239).
- [201] Apache Software Foundation. *GenApp - Apache Airavata - Apache Software Foundation.* Web page. Accessed: 2017-2-22. Aug. 2014. URL: <https://cwiki.apache.org/confluence/display/AIRAVATA/GenApp> (cited on page 196).

-
- [202] Apache Software Foundation. *Apache Airavata*. Web page. Accessed: 2017-2-22. 2016. URL: <http://airavata.apache.org> (cited on page 196).
 - [203] The Apache Software Foundation. *Trident API Overview*. Web Page. Accessed: 2017-1-24. URL: <http://storm.apache.org/releases/1.0.0/Trident-API-Overview.html> (cited on page 197).
 - [204] The Apache Software Foundation. *Trident Tutorial*. Web Page. Accessed: 2017-1-24. URL: <http://storm.apache.org/releases/0.10.1/Trident-tutorial.html> (cited on page 197).
 - [205] The Eclipse Foundation. *Eclipse Winery*. Web Page. Accessed: 2017-1-24. URL: <https://projects.eclipse.org/projects/soa.winery> (cited on page 282).
 - [206] Tony Fountain et al. “The Open Source DataTurbine Initiative: Empowering the Scientific Community with Streaming Data Middleware”. In: *Bulletin - Ecological Society of America* 93.3 (July 2012), pages 242–252. URL: <http://onlinelibrary.wiley.com/doi/10.1890/0012-9623-93.3.242/abstract> (cited on page 232).
 - [207] G. Fox and S. Pallickara. “Deploying the NaradaBrokering Substrate in Aiding Efficient Web and Grid Service Interactions”. In: *Proceedings of the IEEE* 93.3 (Mar. 2005). Accessed : 2017-02-15, pages 564–577. ISSN: 0018-9219. DOI: 10.1109/JPROC.2004.842759 (cited on page 239).
 - [208] *FTP Wikipedia*. Web Page. Accessed: 2017-02-02. URL: <https://cwiki.apache.org/confluence/display/Hive/RCFileCat> (cited on page 265).
 - [209] *FUSE Site*. Web Page. URL: <http://fuse.sourceforge.net> (cited on page 272).
 - [210] Edgar Gabriel et al. “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”. In: *Proceedings, 11th European PVM/MPI Users’ Group Meeting*. Budapest, Hungary, Sept. 2004, pages 97–104. URL: <https://www.open-mpi.org/papers/euro-pvmm MPI-2004-overview/euro-pvmm MPI-2004-overview.pdf> (cited on page 296).
 - [211] *About Ansible Galaxy*. Web Page. Accessed: 2017-2-06. URL: <https://galaxy.ansible.com/intro/> (cited on page 198).
 - [212] *GitHub ansible/galaxy*. Web Page. Accessed: 2017-2-06. URL: <https://github.com/ansible/galaxy/> (cited on page 198).
 - [213] Luke Galea. *Graylog2 optimization for High-log Environments*. webpage. Accessed : 02-21-2017. July 2012. URL: <https://dzone.com/articles/graylog2-optimization-high-log> (cited on page 214).
 - [214] *Galera Cluster*. Web Page. 2017. URL: <http://galeracluster.com> (cited on page 249).
 - [215] Gamefromscratch. *Hands On With Amazon’s Lumberyard Game Engine - YouTube*. URL: <https://www.youtube.com/watch?v=FUD1TTbt4qE> (visited on 02/27/2017) (cited on page 228).
 - [216] *Ganglia Monitoring System*. Web Page. Accessed: 2017-02-24. URL: <http://ganglia.info/> (cited on page 291).
 - [217] *Ganglia (software)*. Web Page. Accessed: 2017-02-24. URL: [https://en.wikipedia.org/wiki/Ganglia_\(software\)](https://en.wikipedia.org/wiki/Ganglia_(software)) (cited on page 291).
 - [218] Mitch Garnaat. *boto components*. Web Page. Accessed: 2017-1-25. URL: <http://boto.cloudhackers.com/en/latest/> (cited on page 278).
 - [219] Mitch Garnaat. *boto3-documentation components*. Web Page. Accessed: 2017-1-26. URL: <https://boto3.readthedocs.io/en/latest/> (cited on page 278).
 - [220] Mitch Garnaat. *boto-amazon-python-sdk components*. Web Page. Accessed: 2017-1-26. URL: <https://aws.amazon.com/sdk-for-python/> (cited on pages 278, 279).
 - [221] Mitch Garnaat. *boto-github components*. Web Page. Accessed: 2017-1-25. URL: <https://github.com/boto/boto3> (cited on page 278).

- [222] Robert C. Gentleman et al. “Bioconductor: open software development for computational biology and bioinformatics”. In: *Genome Biology* 5.10 (2004), R80. ISSN: 1474-760X. DOI: 10.1186/gb-2004-5-10-r80. URL: <http://dx.doi.org/10.1186/gb-2004-5-10-r80> (cited on page 205).
- [223] *Get started with Azure Blob storage (object storage) using .NET | Microsoft Docs*. Web Page. URL: <https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-how-to-use-blobs> (visited on 02/13/2017) (cited on page 273).
- [224] Ali Ghodsi et al. “Dominant Resource Fairness: Fair Allocation of Multiple Resource Types.” In: *NSDI*. Volume 11. 11. 2011, pages 24–24 (cited on page 266).
- [225] *GitHub - elastic/kibana: Kibana analytics and search dashboard for Elasticsearch*. Web Page. URL: <https://github.com/elastic/kibana> (visited on 02/27/2017) (cited on page 213).
- [226] *GitHub - jupyter/jupyter: Jupyter metapackage for installation, docs and chat*. Web page. URL: <https://github.com/jupyter/jupyter> (visited on 02/27/2017) (cited on page 198).
- [227] *GitHub - jupyter/notebook: Jupyter Interactive Notebook*. Web page. URL: <https://github.com/jupyter/notebook> (visited on 02/27/2017) (cited on page 198).
- [228] *CDAP Applications*. Code Repository. Accessed: 2017-2-18. May 2015. URL: <https://github.com/caskdata/cdap-apps> (cited on page 298).
- [229] *coreos/rkt*. Web Page. Accessed: 1/27/2017. URL: <https://github.com/coreos/rkt/> (cited on page 288).
- [230] *About the Globus Toolkit*. Web Page. Accessed: 2017-2-26. URL: <http://toolkit.globus.org/toolkit/about.html> (cited on page 270).
- [231] Google Inc. *Google Cloud Datastore Overview*. Web page. Online; accessed 9-Apr-2017. URL: <https://cloud.google.com/datastore/docs/concepts/overview> (cited on page 262).
- [232] Sunila Gollapudi. *Getting Started with Greenplum for Big Data Analytics*. Packt Publishing, Oct. 25, 2013. 172 pages. ISBN: 1782177043. URL: http://www.ebook.de/de/product/21653990/sunila_gollapudi_getting_started_with_greenplum_for_big_data_analytics.html (cited on page 266).
- [233] Jose Gonzalez and Jeff Lindsay. *Buildstep*. Code repository. Accessed: 2017-1-24. July 2015. URL: <https://github.com/progrium/buildstep> (cited on page 282).
- [234] Tom Goodale et al. *A Simple API for Grid Applications*. webpage. Accessed: 02-01-2017. Sept. 2013. URL: <https://www.ogf.org/documents/GFD.90.pdf> (cited on page 277).
- [235] Google. *Cloud Bigtable*. Web Page. accessed 2017-01-29. URL: <https://cloud.google.com/bigtable/> (cited on page 257).
- [236] Google. *CLOUD DATAFLOW*. WebPage. URL: <https://cloud.google.com/dataflow/> (cited on page 201).
- [237] Google. *Cloud Dataflow*. Web Page. Accessed: 02/03/2016. URL: <https://cloud.google.com/dataflow/> (cited on page 228).
- [238] Google. *Google Cloud Prediction API Documentation*. Web Page. Accessed 2017-1-26. URL: <https://cloud.google.com/prediction/docs/> (cited on page 207).
- [239] Google. *Google Cloud Translation API Documentation*. Web Page. Accessed 2017-2-20. URL: <https://cloud.google.com/translate/docs/> (cited on page 207).
- [240] Google. *Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud*. 2012. URL: <http://homes.cs.washington.edu/~alon/files/socc10.pdf> (cited on page 212).
- [241] Google. *FusionTableSupportHelp*. Web Page. 2017. URL: <https://support.google.com/fusiontables/answer/171181?hl=en> (cited on page 212).

- [242] Mike Burrows. *Chubby Site*. Web Page. URL: <https://research.google.com/archive/chubby.html> (cited on page 294).
- [243] *Google Cloud*. Webpage. URL: https://en.wikipedia.org/wiki/Google_Cloud (cited on page 290).
- [244] *Google cloud platform*. Webpage. URL: <https://cloud.google.com> (cited on page 290).
- [245] *Google Cloud SQL*. web page. URL: <https://cloud.google.com/sql/> (cited on page 251).
- [246] *Google Cloud SQL FAQ*. Webpage. URL: <https://cloud.google.com/sql/faq#version> (cited on page 251).
- [247] *Google Cloud Storage Documentation*. Web Page. Accessed: 02-06-2017. URL: <https://cloud.google.com/storage/docs> (cited on page 274).
- [248] Google Developers. *Cloud Machine Learning*. Web page. accessed 16-March-2017. URL: <https://cloud.google.com/ml-engine/> (cited on page 302).
- [249] Google Developers. *Cloud ML Engine Overview*. Web page. accessed 16-March-2017. URL: <https://cloud.google.com/ml-engine/docs/concepts/technical-overview> (cited on page 302).
- [250] Google Inc. *Balancing Strong and Eventual Consistency with Google Cloud Datastore*. Web page. Online; accessed 9-Apr-2017. URL: <https://cloud.google.com/datastore/docs/articles/balancing-strong-and-eventual-consistency-with-google-cloud-datastore/> (cited on page 262).
- [251] Google Inc. *Google-pub-sub-scalable-messaging-middleware*. Web Page. Accessed: 2017-2-10. URL: <https://cloud.google.com/pubs/> (cited on page 242).
- [252] Google Inc. *What-is-Google-Pub-Sub*. Web Page. Accessed: 2017-2-09. URL: <https://cloud.google.com/pubs/docs/overview> (cited on page 242).
- [253] *Gora - in-memory data model and persistence for big data*. Web Page. Accessed: 2017-01-18. URL: <http://gora.apache.org/> (cited on page 242).
- [254] Clinton Gormley and Zachary Tong. *Elasticsearch - The Definitive Guide : A Distributed REAL-TIME SEARCH AND ANALYTICS ENGINE*. 1st. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc, 2015. ISBN: 9781449358549 (cited on page 213).
- [255] Pivotal Software Inc. *gpfdist*. Web Page. 2017. URL: http://gpdb.docs.pivotal.io/4330/utility_guide/admin_utilities/gpfdist.html (cited on page 266).
- [256] GraphLab. *GraphLab*. Web Page. accessed 2017-01-28. Dec. 2012. URL: <http://www.select.cs.cmu.edu/code/graphlab/> (cited on page 210).
- [257] *Dashboards - 2.2.1 documentation*. webpage. Accessed : 02-21-2017. 2012. URL: <http://docs.graylog.org/en/2.2/pages/dashboards.html> (cited on page 214).
- [258] Rick Grehan. *NoSQL Showdown: MongoDB vs. Couchbase*. Web page. Online; accessed 29-jan-2017. Mar. 2013. URL: <http://www.infoworld.com/article/2613970/nosql/nosql-showdown--mongodb-vs--couchbase.html> (cited on page 256).
- [259] *Globus Online (GridFTP)*. Web Page. Accessed:1/16/2017. URL: <https://en.wikipedia.org/wiki/GridFTP> (cited on page 265).
- [260] C.J. Gronlund et al. *Azure Machine Learning*. Web Page. Accessed: 2017-2-27. Jan. 2017. URL: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-machine-learning#what-is-machine-learning-in-the-microsoft-azure-cloud> (cited on page 206).
- [261] RDF Working Group. *Resource Description Framework (RDF)*. Online. Feb. 2014. URL: <https://www.w3.org/RDF/> (cited on page 258).
- [262] *Grow Hosting Business with Software Platform*. Web Page. URL: <https://jelastic.com/cloud-business-for-hosting-providers/> (visited on 02/13/2017) (cited on page 219).

-
- [263] *H2O Website Documentation components*. Web Page. URL: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/welcome.html> (cited on page 210).
 - [264] *H2O Wikipedia Documentation components*. Web Page. Accessed : 2017-02-12. URL: [https://en.wikipedia.org/wiki/H2O_\(software\)1](https://en.wikipedia.org/wiki/H2O_(software)1) (cited on page 210).
 - [265] Hadoop Apache. *Apache Software Foundation*. Web Page. 2016. URL: <https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html> (cited on page 267).
 - [266] Bajda Pawlikowski et al. *An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads*. Web Page. Accessed: 2017-02-12. URL: <db.cs.yale.edu/hadoopdb/hadoopdb.html> (cited on page 225).
 - [267] Azzam Haidar, Jakub Kurzak, and Piotr Luszczek. “An improved parallel singular value algorithm and its implementation for multicore hardware”. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, page 90. URL: <http://dl.acm.org/citation.cfm?id=2503292> (visited on 02/13/2017) (cited on page 206).
 - [268] H. F. Halaoui. “A spatio temporal indexing structure for efficient retrieval and manipulation of discretely changing spatial data”. In: *Journal of Spatial Science* 53.2 (2008), pages 1–12. DOI: 10.1080/14498596.2008.9635146. URL: <http://dx.doi.org/10.1080/14498596.2008.9635146> (cited on page 215).
 - [269] Matthew Hardin. *Understanding LMDB Database File Sizes and Memory Utilization*. WebPage. May 2016. URL: <https://symas.com/understanding-lmdb-database-file-sizes-and-memory-utilization/> (cited on page 244).
 - [270] Harp. *Harp*. Web Page. accessed 2017-01-28. Jan. 2012. URL: <http://harpjs.com> (cited on page 237).
 - [271] *Haystack*. Web Page. URL: www.project-haystack.org (cited on page 270).
 - [272] Hazelcast. *Open Source In-Memory Data Grid*. Code Repository. accessed 2017-01-29. Jan. 2017. URL: <https://github.com/hazelcast/hazelcast> (cited on page 244).
 - [273] *hdfs*. Web Page. Accessed: 2017-1-21. URL: <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html> (cited on page 270).
 - [274] Yongqiang He et al. “RCFile: A fast and space-efficient data placement structure in MapReduce-based warehouse systems”. In: *Data Engineering (ICDE), 2011 IEEE 27th International Conference on Engineering*. IEEE. 2011, pages 1199–1208 (cited on page 264).
 - [275] Michael Heap. *Ansible From Beginner to Pro*. Edited by L. Corrigan et al. apress, 2016 (cited on page 198).
 - [276] *Heat - Openstack*. webpage. Accessed : 01-15-2017. URL: <https://wiki.openstack.org/w/index.php?title=Heat&oldid=100000> (cited on page 280).
 - [277] *Heroku*. Web Page. Accessed:1/27/2017. URL: <https://devcenter.heroku.com/> (cited on page 217).
 - [278] Hugo Hiden et al. *e-Science Central: Cloud-based e-Science and its application to chemical property modelling*. Technical report. Newcastle, England: University of Newcastle upon Tyne, Nov. 2010. URL: http://eprint.ncl.ac.uk/file_store/production/179301/8EA32A5C-DE97-44A8-869B-81731044F2A8.pdf (cited on page 201).
 - [279] *Home · Microsoft/CNTK Wiki · GitHub*. Web Page. URL: <https://github.com/Microsoft/CNTK/wiki> (visited on 02/13/2017) (cited on page 216).
 - [280] *How Puppet Works*. Web Page. URL: <http://www.slashroot.in/puppet-tutorial-how-does-puppet-work> (cited on page 278).

- [281] *How To Create OpenVZ Container In OpenVZ | Unixmen*. Web Page. URL: <https://www.unixmen.com/how-to-create-openvz-container-in-openvz/> (visited on 02/13/2017) (cited on page 286).
- [282] *How To Use Logstash and Kibana To Centralize Logs On Ubuntu 14.04*. Web Page. URL: <https://www.digitalocean.com/community/tutorials/how-to-use-logstash-and-kibana-to-centralize-and-visualize-logs-on-ubuntu-14-04> (visited on 02/27/2017) (cited on page 214).
- [283] HowStuffWorks.com. *What are relational databases?* Online. Mar. 2001. URL: <http://computer.howstuffworks.com/question599.htm> (cited on page 248).
- [284] Robert Kallman et al. "H-Store: a High-Performance, Distributed Main Memory Transaction Processing System". In: *Proc. VLDB Endow.* 1.2 (2008), pages 1496–1499. ISSN: 2150-8097. DOI: <http://doi.acm.org/10.1145/1454159.1454211>. URL: <http://hstore.cs.brown.edu/papers/hstore-demo.pdf> (cited on page 245).
- [285] HPE. *HPE Helion Stackato*. Webpage. URL: <https://www.hpe.com/us/en/software/multi-cloud-platform.html> (cited on page 219).
- [286] *H-Store*. Web Page. Accessed:1/16/2017. URL: <http://hstore.cs.brown.edu/> (cited on page 245).
- [287] *H-StoreWiki*. Web Page. Accessed:1/16/2017. URL: <https://en.wikipedia.org/wiki/H-Store> (cited on page 245).
- [288] IBM. *BlueworksLive*. Web Page. Accessed: 2017-1-24. URL: <https://www.blueworkslive.com/home> (cited on page 283).
- [289] IBM. *Exercise: Analyze business processes with IBM BPM Blueprint*. Web Page. Accessed: 2017-1-24. URL: <http://www.ibm.com/developerworks/downloads/soasandbox/blueprint.html> (cited on page 283).
- [290] IBM. *IBM Blueworks Live*. Web Page. Accessed: 2017-1-24. URL: https://en.wikipedia.org/wiki/IBM_Blueworks_Live (cited on page 283).
- [291] IBM. *IBM Spectrum Scale*. Web Page. accessed 2017-01-29. URL: <http://www-03.ibm.com/systems/storage/spectrum/scale/> (cited on page 273).
- [292] IBM. *What is Flume?* web page. accessed 201-02-06. URL: <https://www-01.ibm.com/software/datainfosphere/hadoop/flume/> (cited on page 266).
- [293] IBM Corporation. *IBM System G documentation*. Web Page. Accessed: 2017-2-19. IBM, 2014. URL: <http://systemg.research.ibm.com/> (cited on page 211).
- [294] IBM Corporation. *IBM System G documentation-2*. Web Page. Accessed: 2017-2-17. Predictive Analytics Today, 2017. URL: <http://www.predictiveanalyticstoday.com/ibm-system-g-native-store/> (cited on page 211).
- [295] ibm. *Extreme cloud administration toolkit*. Webpage. URL: <http://www-03.ibm.com/systems/technicalcomputing/xcat/> (cited on page 279).
- [296] IBM Corp. Web Page. Online; accessed 25-jan-2017. Jan. 2017. URL: www.softlayer.com (cited on page 218).
- [297] IBM Corp. Web Page. Online; accessed 25-jan-2017. Jan. 2017. URL: <https://www.ibm.com/cloud-computing/bluemix/> (cited on page 218).
- [298] *IBM dashDB*. Web Page. URL: <https://www.ibm.com/analytics/us/en/technology/cloud-data-services/dashdb/> (cited on page 252).
- [299] *IBM Watson*. Web Page. Accessed: 2017-1-25. URL: [https://en.wikipedia.org/wiki/Watson_\(computer\)](https://en.wikipedia.org/wiki/Watson_(computer)) (cited on page 210).
- [300] *IBM Watson Product Page*. Web Page. Accessed: 2017-1-25. URL: <https://www.ibm.com/watson> (cited on page 210).

- [301] Roger Ignazio. *Mesos in Action*. 1st. Greenwich, CT, USA: Manning Publications Co., May 17, 2016. 272 pages. ISBN: 1617292923. URL: <http://dl.acm.org/citation.cfm?id=3006364> (cited on page 266).
- [302] ImageJ. *ImageJ Introduction*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: <https://imagej.nih.gov/ij/docs/intro.html> (cited on page 205).
- [303] Inca - Periodic, automated, user-level cyberinfrastructure testing. Web Page. Accessed: 2017-01-16. URL: <http://inca.sdsc.edu/> (cited on page 291).
- [304] InCommon. *What is the InCommon Federation?* Webpage. URL: https://spaces.internet2.edu/download/attachments/2764/final_InCommon.pdf (cited on page 292).
- [305] Indiana University et al. *SEAGrid Portal*. Web page. Accessed: 2017-2-22. July 2016. URL: <https://seagrid.org> (cited on page 196).
- [306] Infinispan. Webpage. URL: <https://en.wikipedia.org/wiki/Infinispan> (cited on page 244).
- [307] Intel® Data Analytics Acceleration Library (Intel® DAAL). Web Page. Accessed: 02-23-2017. URL: <https://software.intel.com/en-us/intel-daal> (cited on page 208).
- [308] Intel Graph Analytics Solutions. Intel® Graph Builder for Apache Hadoop® Software v2. Web Page. URL: <https://www-ssl.intel.com/content/www/us/en/software/intel-graph-builder-for-apache-hadoop-software-v2-product-detail.html> (cited on page 211).
- [309] Introducing Espresso - LinkedIn's hot new distributed document store | LinkedIn Engineerin. Web Page. Online; accessed 23-Feb-2017. URL: <https://engineering.linkedin.com/espresso/introducing-espresso-linkedin-s-hot-new-distributed-document-store> (cited on page 255).
- [310] Introduction | Kibana User Guide [5.2] | Elastic. Web Page. Docs/Kibana/Reference/5.2. URL: <https://www.elastic.co/guide/en/kibana/current/introduction.html> (visited on 02/27/2017) (cited on page 214).
- [311] Introduction to Infinispan. Distributed in-memory key/value data grid and cache. Webpage. URL: <http://infinispan.org/about/> (cited on page 244).
- [312] IPython. en. Web page. Page Version ID: 767266837. Feb. 2017. URL: <https://en.wikipedia.org/w/index.php?title=IPython&oldid=767266837> (visited on 02/27/2017) (cited on page 198).
- [313] Muhammad Hussain Iqbal and Tariq Rahim Soomro. “Big Data Analysis: Apache Storm Perspective”. In: *International Journal of Computer Trends and Technology (IJCTT)-2015*. Jan. 2015 (cited on page 229).
- [314] iRod. Web Page. URL: <https://irods.org/> (cited on page 262).
- [315] iRod. Web Page. URL: <https://github.com/irods/irods> (cited on page 263).
- [316] Mohammad Islam et al. “Oozie: towards a scalable workflow management system for hadoop”. In: *Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*. ACM. 2012, page 4 (cited on page 199).
- [317] C. Issariyapat et al. “Using Nagios as a groundwork for developing a better network monitoring system”. In: *2012 Proceedings of PICMET '12: Technology Management for Emerging Technologies*. July 2012, pages 2771–2777 (cited on page 291).
- [318] IU 'Twister' software improves Google's MapReduce for large-scale scientific data analysis. Web Page. URL: <https://phys.org/news/2010-03-iu-twister-software-google-mapreduce.html> (visited on 02/13/2017) (cited on page 233).
- [319] IU Twister software improves Google's MapReduce for large-scale scientific data analysis: IU News Room: Indiana University. Web Page. URL: <http://newsinfo.iu.edu/news-archive/13726.html> (visited on 02/13/2017) (cited on page 233).

- [320] Jacob Jackson. *Google service analyzes live streaming data*. WebPage. June 2014. URL: <http://www.infoworld.com/article/2607938/data-mining/google-service-analyzes-live-streaming-data.html> (cited on page 201).
- [321] Jake Luciani. *Solandra Wiki*. Code Repository. Accessed: 2017-02-12. Feb. 2017. URL: <https://github.com/tjake/Solandra/wiki/Solandra-Wiki> (cited on page 252).
- [322] Jake Luciani. *Solandra Wiki*. Code Repository. Accessed: 2017-02-12. Feb. 2017. URL: <https://github.com/tjake/Solandra> (cited on page 253).
- [323] *Java Database Connectivity*. Web Page. Accessed: 2017-1-30. URL: https://en.wikipedia.org/wiki/Java_Database_Connectivity (cited on page 247).
- [324] *JClouds - The Java Multi-Cloud Toolkit*. Web Page. Accessed: 2017-01-20. URL: <https://jclouds.apache.org/> (cited on page 275).
- [325] *Jelastic*. en. Web Page. Page Version ID: 754931676. Dec. 2016. URL: <https://en.wikipedia.org/w/index.php?title=Jelastic&oldid=754931676> (visited on 02/13/2017) (cited on page 219).
- [326] Shantenu Jha et al. “Grid Interoperability at the Application Level Using SAGA”. In: *07 Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*. Dec. 2007. ISBN: 0769530648 (cited on page 277).
- [327] Ana Lucia Varbanescu Jianbin Fang and Henk Sips. “Sesame: A User-Transparent Optimizing Framework for Many-Core Processors”. In: *IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. 2013, pages 70–73 (cited on page 261).
- [328] Bingsheng He Jianlong Zhong. “Medusa: Simplified Graph Processing on GPUs”. In: *IEEE Transactions on Parallel and Distributed Systems* 25.6 (Apr. 2013), pages 1–11. URL: http://pdcc.ntu.edu.sg/xtra/paper/2013ago/Medusa_TPDS13.pdf (cited on page 236).
- [329] Jitterbit. *Data Integration and ETL | Jitterbit | Integrate data from any source*. Web Page. accessed: 2017-1-26. Jitterbit. URL: <https://www.jitterbit.com/etl-data-integration/> (cited on page 202).
- [330] *Integration Made Easy - Jitterbit*. Accessed: 2017-1-26. Jitterbit, Inc. 1 Kaiser Plaza Suite 701 Oakland, CA 94612. URL: <http://www.jitterbit.com/Files/Product/Jitterbit-General-Datasheet.pdf> (cited on page 202).
- [331] *Technical Overview - Jitterbit*. Accessed: 2017-1-26. Jitterbit, Inc. 2011. URL: <http://www.jitterbit.com/Files/Product/JitterbitTechnicalOverview.pdf> (cited on page 202).
- [332] *The Java EE 6 Tutorial*. webpage. Accessed : 01-18-2017. URL: <http://docs.oracle.com/javaee/6/tutorial/doc/bnceh.html> (cited on page 240).
- [333] Tim Jones. *Anatomy of the libvirt virtualization*. Webpage. Jan. 2010. URL: <https://www.ibm.com/developerworks/library/l-libvirt/> (cited on page 274).
- [334] David Josephsen. *Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks*. 2nd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2013. ISBN: 013313573X, 9780133135732 (cited on page 291).
- [335] The Apache Software Foundation. *Apache OpenJPA - Wikipedia*. webpage. Accessed : 02-22-2017. Jan. 2017. URL: https://en.wikipedia.org/wiki/Apache_OpenJPA (cited on page 246).
- [336] JPAB. *JPA Performance Benchmark*. Web Page. Accessed: 2017-02-04. URL: <http://www.jpab.org/DataNucleus.html> (cited on page 246).
- [337] Canonical Ltd. *Juju*. Web Page. URL: <https://www.ubuntu.com/cloud/juju> (cited on page 279).
- [338] Guy E. Blelloch Julian Shun and Laxman Dhulipala. “Smaller and Faster: Parallel Processing of Compressed Graphs with Ligra+”. In: *ACM SIGPLAN Notices - PPoPP '13DCC '15*

- Proceedings of the 2015 Data Compression Conference.* Pittsburgh, Pennsylvania USA: IEEE Computer Society Washington, DC, USA, 2015, pages 403–412. ISBN: 978-1-4799-8430-55. DOI: 10.1109/DCC.2015.8. URL: <http://dl.acm.org/citation.cfm?id=2860198> (cited on page 235).
- [339] Steven van Wel Jurg Vliet Flavia Paganelli and Dara Dowd. *Elastic Beanstalk*. 1st edition. O'Reilly Media, Inc., 2011. ISBN: 1449306640 (cited on page 218).
- [340] *Astronomical Image Processing with Hadoop*. Volume 442. Astronomical Data Analysis Software and Systems XX. ASP Conference Proceedings, July 2011. URL: <http://adsabs.harvard.edu/abs/2011ASPC..442...93W> (cited on page 263).
- [341] Riyad Kalla. *Well put! When should you use MongoDB vs Couchbase versus Redis...* Web page. Online; accessed 29-jan-2017. Oct. 2011. URL: <http://rick-hightower.blogspot.com/2014/04/well-put-when-should-you-use-mongodb-vs.html> (cited on page 256).
- [342] Mike Kavis. *THE PROS AND CONS OF PRIVATE AND PUBLIC PAAS*. Webpage. June 2013. URL: <https://www.virtualizationpractice.com/the-pros-and-cons-of-private-and-public-paas-21961/> (cited on page 219).
- [343] Keith-Bisset. *CINET - A Cyber-Infrastructure for Network Science*. Web Page. URL: www.portal.futuresystems.org/project/233 (cited on page 212).
- [344] Amol Kekre. *Apache Apex blog*. Web Page. Accessed: 2017-02-26. Sept. 2015. URL: <https://www.datatorrent.com/blog/introducing-apache-apex-incubating/> (cited on page 300).
- [345] Ben Kepes. *Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS*. white paper. Rackspace, 2015. URL: <https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/> (cited on page 219).
- [346] Kepler Project. *The Kepler Project*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: <https://kepler-project.org> (cited on page 196).
- [347] Justin Kestelyn. *Phoenix in 15 Minutes or Less*. Web page. Online; accessed 25-jan-2017. Mar. 2013. URL: <http://blog.cloudera.com/blog/2013/03/phoenix-in-15-minutes-or-less/> (cited on page 223).
- [348] Agargent. *About kestrel*. Web Page. Accessed: 2017-02-12. URL: <https://github.com/twitter-archive/kestrel> (cited on page 240).
- [349] Keystone Wiki. Web Page. Accessed: 2017-2-12. URL: <https://wiki.openstack.org/wiki/Keystone> (cited on page 292).
- [350] Kibana. en. Web Page. Page Version ID: 767434139. Feb. 2017. URL: <https://en.wikipedia.org/w/index.php?title=Kibana&oldid=767434139> (visited on 02/27/2017) (cited on page 214).
- [351] Kibana: Explore, Visualize, Discover Data |Elastic. Web Page. URL: <https://www.elastic.co/products/kibana> (visited on 02/27/2017) (cited on page 213).
- [352] Jik-Soo Kim et al. “HTCaaS: Leveraging Distributed Supercomputing Infrastructures for Large-Scale Scientific Computing”. In: *ACM MTAGS 13*. 2013 (cited on page 270).
- [353] *Kinesis - real-time streaming data in the AWS cloud*. Web Page. Accessed: 2017-01-17. URL: <https://aws.amazon.com/kinesis/> (cited on page 230).
- [354] Wikipedia. *Apache OpenJPA*. webpage. Accessed : 02-22-2017. Dec. 2016. URL: <http://openjpa.apache.org/> (cited on page 246).
- [355] Oliver Kopp et al. “Winery – A Modeling Tool for TOSCA-based Cloud Applications”. In: *11th International Conference on Service-Oriented Computing*. Springer, Dec. 2013, pages 700–704. URL: <http://www.iaas.uni-stuttgart.de/RUS-data/INPROC->

- 2013-46%5C%20-%5C%20Winery_A_Modeling_Tool_for_TOSCA-based_Cloud_Applications.pdf (cited on pages 282, 283).
- [356] Lars Kotthoff. *LLAMA: Leveraging Learning to Automatically Manage Algorithms*. Research paper. Apr. 30, 2014. URL: <https://arxiv.org/abs/1306.1031> (cited on page 267).
- [357] T. Kraska et al. “MLbase: A Distributed Machine Learning System”. In: *MLbase: A Distributed Machine Learning System*. Conference on Innovative Data Systems Research. 2013 (cited on page 204).
- [358] Kubernetes. *Kubernetes Site*. Web Page. URL: <https://kubernetes.io/docs/whatisk8s/> (cited on page 281).
- [359] Jakub Kurzak et al. “Scheduling dense linear algebra operations on multicore processors”. In: *Concurrency and Computation: Practice and Experience* 22.1 (2010), pages 15–44. URL: <http://onlinelibrary.wiley.com/doi/10.1002/cpe.1467/full> (visited on 02/13/2017) (cited on page 206).
- [360] *KVM Wikipedia Documentation*. Web Page. Accessed : 2017-02-12. URL: https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine (cited on page 285).
- [361] *KVM webpage documentation*. Web Page. Accessed : 2017-02-13. URL: http://www.linux-kvm.org/page/Main_Page (cited on page 285).
- [362] *Kyoto Cabinet*. Web Page. Accessed:1/16/2017. URL: <http://fallabs.com/kyotocabinet/> (cited on page 227).
- [363] *Kyoto Cabinet*. Web Page. Accessed: 2017-1-27. URL: <http://fallabs.com/kyotocabinet/> (cited on page 254).
- [364] Aapo Kyrola, Guy Blelloch, and Carlos Guestrin. “GraphChi: Large-scale Graph Computation on Just a PC”. In: *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*. USENIX Association, 2012, pages 31–46. ISBN: 978-1-931971-96-6. URL: <http://dl.acm.org/citation.cfm?id=2387880.2387884> (cited on page 235).
- [365] Fal labs. *Nijo Cabinet: a straightforward implementation of DBM*. Web Page. Accessed: 02/03/2016. URL: <http://fallabs.com/kyotocabinet/> (cited on page 254).
- [366] Fal labs. *Shijo Cabinet: a handy cache/storage server*. Web Page. Accessed: 02/03/2016. URL: <http://fallabs.com/kyototycoon/> (cited on page 254).
- [367] Gregor von Laszewski and Mike Hategan. “Using Karajan”. In: *Java CoG Kit Karajan/Gridant Workflow Guide*. 2005. URL: <https://pdfs.semanticscholar.org/48e1/31ca4e7a76ca98c43d46975cd79c3dbfd4cb.pdf> (cited on page 303).
- [368] Joe Lennon. *Exploring CouchDB: A document-oriented database for Web applications*. Web page. accessed feb-26-2017. Mar. 2009. URL: <http://www.ibm.com/developerworksopensource/library/os-couchdb/index.html> (cited on page 255).
- [369] LevelDB. *LevelDB*. Web Page. accessed 2017-02-25. Feb. 2017. URL: leveldb.org (cited on page 257).
- [370] T. Li et al. “ZHT: A Light-Weight Reliable Persistent Dynamic Scalable Zero-Hop Distributed Hash Table”. In: *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*. May 2013, pages 775–787. DOI: 10.1109/IPDPS.2013.110. URL: http://datasys.cs.iit.edu/publications/2013_IPDPS13_ZHT.pdf (cited on page 253).
- [371] Weizhong Li. “Introduction to bioActors”. In: *bioKepler Tools and Its Applications*. Edited by bioKepler. 1st Workshop on bioKepler Tools and Its Applications 1. UCSD;SDSC. San Diego Supercomputer Center 10100 Hopkins Dr, La Jolla, CA 92093: bioKepler, Sept. 2012, page 31. URL: <http://www.biokepler.org/sites/swat.sdsc.edu>.

- biokepler/files/workshops/2012-09-05/slides/2012-09-05-02-Li.pdf (cited on page 198).
- [372] *Libvirt virtualization API*. Web Page. URL: <https://libvirt.org/> (cited on page 274).
- [373] Ching Yung Lin. "Graph Computing and Linked Big Data". In: *8th IEEE/ICSC International Conference on Semantic Computing*. IBM Corporation, 2014, pages 1–63. URL: http://ieee-icsc.org/icsc2014/GraphComputing_IBM_Lin.pdf (cited on page 211).
- [374] LinkedIn Developers. *Getting started with the REST API*. Web page. accessed 17-March-2017. URL: <https://developer.linkedin.com/docs/rest-api> (cited on page 231).
- [375] *Linux containers*. web page. URL: <https://en.wikipedia.org/wiki/LXC> (cited on page 286).
- [376] *Linux containers in use*. web page. URL: <http://www.infoworld.com/article/3072929/linux/containers-101-linux-containers-and-docker-explained.html> (cited on page 286).
- [377] Martin Lisa et al. *Data mining with iPlant*: Paper. Oct. 2014. URL: <https://academic.oup.com/jxb/article/66/1/1/2893405/Data-mining-with-iPlantA-meeting-report-from-the> (cited on page 222).
- [378] Jinjun Chen Lizhe Wang Wei Jie. *Grid Computing: Infrastructure, Service, and Applications*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487: Taylor & Francis, 2009. ISBN: 1420067664 (cited on page 291).
- [379] Scott Lowe. *An introduction to Openstack Heat*. webpage. Accessed : 01-15-2017. May 2014. URL: <http://blog.scottlowe.org/2014/05/01/an-introduction-to-openstack-heat/> (cited on page 280).
- [380] Andy Lurie. *Top 47 Log Management Tools*. webpage. Accessed : 02-21-2017. May 2014. URL: <http://blog.profitbricks.com/top-47-log-management-tools> (cited on page 214).
- [381] *LXD an hypervisor for containers*. web page. URL: <https://lists.linuxcontainers.org/pipermail/lxc-devel/2014-November/010817.html> (cited on page 300).
- [382] Tiago Macedo and Fred Oliveira. *Redis Cookbook: Practical Technique for fast Data Manipulation*. Sonoma County, California, United States: O'Reilly Media, Aug. 2011. ISBN: 978-1-4493-0503-1 (cited on page 243).
- [383] *Apache Mahout*. Web Page. URL: <http://mahout.apache.org/> (cited on page 203).
- [384] Jose C Elias Margaret Rouse. *ACID (atomicity, consistency, isolation, and durability)*. Website. July 2006. URL: <http://searchsqlserver.techtarget.com/definition/ACID> (cited on page 248).
- [385] Matthias Marschall. *Chef Infrastructure Automation Cookbook*. Packt Publishing, 2013. ISBN: 9351105164 and 9789351105169 (cited on page 278).
- [386] Martinez-Rubi.O et al. *Taming the beast: Free and open-source massive point cloud web visualization*. 2015. URL: <http://repository.tudelft.nl/islandora/object/uuid:0472e0d1-ec75-465a-840e-fd53d427c177?collection=research> (cited on page 215).
- [387] Karl Matthias and Sean P. Kane. *Docker Up and Running, Shipping Reliable Containers In Production*. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O'REILLY, 2015. ISBN: 978-1-491-91757-2 (cited on pages 203, 277).
- [388] Morel Matthieu. *S4 0.5.0 Overview*. Web Page. Accessed: 2017-02-24. Feb. 2013. URL: <https://cwiki.apache.org/confluence/display/S4/S4+0.5.0+Overview> (cited on page 229).
- [389] Norman Maurer and Marvin Wolfthal. *Netty in Action*. 1st. Greenwich, CT, USA: Manning Publications, Dec. 23, 2015. 296 pages. ISBN: 1617291471. URL: <http://www.ebook.com>.

- de / de / product / 21687528 / norman_maurer.netty_in_action.html (cited on page 237).
- [390] M. McLennan and R. Kennell. “HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering”. In: *Computing in Science Engineering* 12.2 (Mar. 2010), pages 48–53. ISSN: 1521-9615. DOI: 10.1109/mcse.2010.41 (cited on page 221).
- [391] Eileen McNulty. *What Is Google Cloud Dataflow?* Web Page. Accessed: 02/03/2016. URL: <http://dataconomy.com/2014/08/google-cloud-dataflow/> (cited on page 228).
- [392] Sergey Melnik et al. “Dremel: Interactive Analysis of Web-Scale Datasets”. In: *Communications of the ACM* 54 (June 2011), pages 114–123. ISSN: 0001-0782. DOI: 10.1145/1953122.1953148. URL: <http://cacm.acm.org/magazines/2011/6/108648-dremel-interactive-analysis-of-web-scale-datasets/fulltext> (cited on page 226).
- [393] Xiangrui Meng et al. “MLlib: Machine Learning in Apache Spark”. In: *CoRR* abs/1505.06807 (2015). URL: <http://arxiv.org/abs/1505.06807> (cited on page 204).
- [394] *Mesos Site*. Web Page. URL: <http://mesos.apache.org/> (cited on page 266).
- [395] Microsoft. *Event Hubs*. Web Page. accessed 2017-02-25. Dec. 2016. URL: <https://azure.microsoft.com/en-us/services/event-hubs/> (cited on page 242).
- [396] Microsoft Corp. *Form 10-K*. Web page. Online; accessed 21-jan-2017. July 2016. URL: <https://www.sec.gov/Archives/edgar/data/789019/000119312516662209/d187868d10k.htm> (cited on page 218).
- [397] Microsoft Corp. Web Page. Online; accessed 21-jan-2017. Jan. 2017. URL: <https://azure.microsoft.com/en-us/> (cited on page 218).
- [398] Microsoft Corp. *Designing a Scalable Partitioning Strategy for Azure Table Storage*. Web page. Online; accessed 28-jan-2017. Jan. 2017. URL: <https://docs.microsoft.com/en-us/rest/api/storageservices/fileservices/designing-a-scalable-partitioning-strategy-for-azure-table-storage> (cited on page 261).
- [399] Stuart Miniman. *Cisco Moves Up the Cloud Stack with Intelligent Automation*. webpage. 2011. URL: http://wikibon.org/wiki/v/Cisco_Moves_Up_the_Cloud_Stack_with_Intelligent_Automation (cited on page 280).
- [400] Ross Mistry and Stacia Misner. *Introducing Microsoft SQL Server 2014 Technical Overview*. Microsoft Press, 2014. ISBN: 978-0-7356-8475-1 (cited on page 248).
- [401] *Apache Spark’s MLlib*. Web Page. URL: <http://spark.apache.org/mllib/> (cited on page 204).
- [402] *MLPY documentation*. Web Page. 2012. URL: <http://mlpy.sourceforge.net/docs/3.5/> (cited on page 207).
- [403] AdaptiveComputing. *Moab Cluster Suite*. Web Page. Accessed: 2017-02-24. URL: [https://www.adaptivecomputing.com/products/](http://www.adaptivecomputing.com/products/) (cited on page 269).
- [404] Modine Austin. *Cobbler*. Web Page. accessed 2017-01-30. Feb. 2008. URL: http://www.theregister.co.uk/2008/06/19/red_hat_summit_2008_cobbler/ (cited on page 279).
- [405] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. “IaaS Cloud Architecture From Virtualized Datacenters to Federated Cloud Infrastructures”. In: *Computer*. Edited by IEEE. Volume 45. 12. IEEE Computer Society. IEEE, 2012, pages 65–72. DOI: 10.1109/MC.2012.76. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6165242&isnumber=6383143> (cited on page 287).
- [406] *MQTT*. Web Page. Accessed: 2017-1-27. URL: <http://mqtt.org/> (cited on page 240).
- [407] *MQTT Floodnet*. Web Page. Accessed: 2017-1-27. URL: <http://mqtt.org/projects/floodnet> (cited on page 241).

- [408] *MR-MPI*. Web Page. 2017. URL: <http://mapreduce.sandia.gov/doc> (cited on page 234).
- [409] Subramanian Muralidhar et al. “f4: Facebook’s warm blob storage system”. In: *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*. USENIX Association. 2014, pages 383–398 (cited on page 271).
- [410] Derek G. Murray et al. “Naiad: A Timely Dataflow System”. In: *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. SOSP ’13. Accessed: 2017-1-27. Farmington, Pennsylvania: ACM, 2013, pages 439–455. ISBN: 978-1-4503-2388-8. DOI: 10.1145/2517349.2522738. URL: <http://doi.acm.org/10.1145/2517349.2522738> (cited on page 199).
- [411] Colton Myers. *Learning SaltStack*. 2nd. Packt Publishing, 2015. ISBN: 978-1-78439-460-8, 1784394602. URL: <https://books.google.com/books?id=pVnsrQEACAAJ> (cited on page 278).
- [412] mysql. *MySQL 5.7 Reference Manual, What is MySQL*. Online. URL: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html> (cited on page 248).
- [413] *Nagios components*. Web Page. Accessed: 2017-1-11. URL: <https://www.nagios.org/projects/> (cited on page 291).
- [414] NASA/GSFC. *CDF User’s Guide (V3.3.6)*. Version 3.3.6. Accessed: 2017-1-28. NASA/GSFC Space Physics Data Facility. NASA / Goddard Space Flight Center, Greenbelt, Maryland 20771 (U.S.A.), Oct. 2016. URL: <http://spdf.gsfc.nasa.gov/pub/software/cdf/doc/cdf363/cdf363ug.pdf> (cited on page 263).
- [415] NASA/GSFC. *CDF Home Page*. webpage. accessed: 2017-1-28. NASA, Goddard Space Flight Center, 2017. URL: <http://cdf.gsfc.nasa.gov/> (cited on page 263).
- [416] National Instruments. *A Layered Approach to Designing Robot Software*. Web page. accessed 18-mar-2017. Mar. 2017. URL: <http://www.ni.com/white-paper/13929/en/> (cited on page 301).
- [417] Jeremy Nelson. *Mastering Redis*. Packt Publishing, May 2016. ISBN: 978-1783988181 (cited on page 243).
- [418] Neo4j. *Causal Cluster*. Web page. Last Accessed: 2017.02.25. URL: <https://neo4j.com/docs/operations-manual/current/clustering/> (cited on page 259).
- [419] Neo4j. *Chapter 4. Clustering*. Web page. Last Accessed: 2017.02.25. URL: <https://neo4j.com/docs/operations-manual/current/clustering/> (cited on page 259).
- [420] Neo4j. *Highly Available cluster*. Web page. Last Accessed: 2017.02.25. URL: <https://neo4j.com/docs/operations-manual/current/clustering/high-availability/architecture/> (cited on page 259).
- [421] *NetCDF: Introduction and Overview*. Web Page. URL: https://www.unidata.ucar.edu/software/netcdf/docs_rc/ (visited on 02/13/2017) (cited on page 263).
- [422] Netsyslab. *TOTEM*. Webpage. URL: <http://netsyslab.ece.ubc.ca/wiki/index.php/Totem> (cited on page 236).
- [423] *Netty Site*. Web Page. URL: <http://netty.io/> (cited on page 237).
- [424] *Nimbus*. Web Page. URL: <http://www.nimbusproject.org/doc/nimbus/platform/> (cited on page 288).
- [425] *Nimbus Wiki*. Web Page. URL: [https://en.wikipedia.org/wiki/Nimbus_\(cloud_computing\)](https://en.wikipedia.org/wiki/Nimbus_(cloud_computing)) (cited on page 287).
- [426] Ninefold. *Ninefold news*. Web Page. Accessed: 2017-2-27. Nov. 2015. URL: <http://ninefold.com/news/> (cited on page 219).
- [427] Nokia Corp. Web page. accessed 25-feb-2017. Feb. 2017. URL: http://www.nokia.com/en_int (cited on page 234).

-
- [428] Jordan Novet. *dotCloud*. Web Page. Accessed: 2017-1-31. Jan. 2016. URL: <http://venturebeat.com/2016/01/22/dotcloud-the-cloud-service-that-gave-birth-to-docker-is-shutting-down-on-february-29/> (cited on page 220).
 - [429] NSA 'NiFi' Big Data Automation Project Out In The Open. Web Page. URL: <http://www.forbes.com/sites/adrianbridgwater/2015/07/21/nsa-nifi-big-data-automation-project-out-in-the-open/#6b37cac55d9a> (cited on page 202).
 - [430] Daniel Nurmi et al. "The eucalyptus open-source cloud-computing system". In: *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE Computer Society. 2009, pages 124–131 (cited on page 287).
 - [431] Christian Nutt. *Gamasutra - Amazon launches new, free, high-quality game engine: Lumberyard*. URL: http://www.gamasutra.com/view/news/265425/Amazon_launches_new_free_highquality_game_engine_Lumberyard.php (visited on 02/27/2017) (cited on page 228).
 - [432] NWB. Web Page. 2017. URL: <http://nwb.cns.iu.edu/about.html> (cited on page 213).
 - [433] *ODE Website*. Web Page. Accessed: 2017-2-11. URL: <http://ode.apache.org/> (cited on page 195).
 - [434] *Apache ODE*. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Apache_ODE (cited on page 195).
 - [435] John O'Hara. "Toward a commodity enterprise middleware". In: *Queue* 5.4 (2007), pages 48–55 (cited on page 238).
 - [436] Carl W Olofson. *The Analytic-Transactional Data Platform: Enabling the Real-Time Enterprise (IDC)*. Technical report. Accessed: 2017-1-17. International Data Corporation (IDC), Dec. 2014. URL: <http://www.sap.com/documents/2016/08/3c4e546e-817c-0010-82c7-eda71af511fa.html> (cited on page 224).
 - [437] Jean-Baptiste Onofre. *The Future of Apache Beam, Now a Top-Level Apache Software Foundation Project*. Blog. accessed 25-feb-2017. Jan. 2017. URL: <https://www.talend.com/blog/2017/01/13/future-apache-beam-now-top-level-apache-software-foundation-project/> (cited on page 297).
 - [438] *Apache OODT*. Web Page. Accessed: 2017-2-25. URL: <http://oodt.apache.org/> (cited on page 298).
 - [439] *Getting Started*. Web Page. Accessed: 2017-2-25. URL: <http://oodt.apache.org/documentation.htm> (cited on page 298).
 - [440] Apache. *About OODT*. Web Page. Accessed: 2017-02-12. URL: <http://oodt.apache.org/> (cited on page 221).
 - [441] *Oozie - Apache Oozie Workflow Scheduler for Hadoop*. Web Page. URL: <http://oozie.apache.org/> (visited on 02/13/2017) (cited on page 199).
 - [442] *Open Database Connectivity*. Web Page. Accessed: 2017-1-30. URL: https://en.wikipedia.org/wiki/Open_Database_Connectivity (cited on page 246).
 - [443] Open Source Robotics Foundation. *About ROS*. Web page. accessed 16-mar-2017. Mar. 2017. URL: <http://www.ros.org/about-ros/> (cited on page 301).
 - [444] *OpenID*. website. URL: <http://openid.net/> (cited on page 293).
 - [445] *OpenID*. webpage. URL: <https://en.wikipedia.org/wiki/OpenID> (cited on page 293).
 - [446] *OpenNebula Wiki*. URL: <https://opennebula.org/about/technology/> (cited on page 287).
 - [447] OpenRefine. *OpenRefine*. Web Page. accessed 2017-01-13. Feb. 2016. URL: <http://openrefine.org/> (cited on page 298).
 - [448] OpenSFS, EOFS. *Welcome to the official home of the Lustre filesystem*. Web Page. accessed 2017-01-28. Dec. 2016. URL: <http://lustre.org/> (cited on page 272).

- [449] *OpenStack*. Web Page. Accessed:1/16/2017. URL: <https://www.openstack.org/> (cited on page 280).
- [450] *Keystone Architecture*. Web Page. Accessed: 2017-2-12. URL: <http://docs.openstack.org/developer/keystone/architecture.html> (cited on page 293).
- [451] *OpenVZ*. en. Web Page. Page Version ID: 764498783. Feb. 2017. URL: <https://en.wikipedia.org/w/index.php?title=OpenVZ&oldid=764498783> (visited on 02/13/2017) (cited on page 286).
- [452] Wikipedia. *Oracle Grid Engine - Wikipedia*. webpage. Accessed : 02-22-2017. Feb. 2017. URL: https://en.wikipedia.org/wiki/Oracle_Grid_Engine (cited on page 268).
- [453] *Oracle Database - Wikipedia*. Web Page. URL: https://en.wikipedia.org/wiki/Oracle_Database (visited on 02/13/2017) (cited on page 247).
- [454] *Oracle Labs PGX*. Web Page. Accessed: 2017-02-07. URL: https://docs.oracle.com/cd/E56133_01/2.3.1/index.html (cited on page 210).
- [455] *Oracle Website*. Web Page. Accessed: 2017-2-11. URL: <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb> (cited on page 254).
- [456] Apache org. *Apache Tomcat*. Web Page. Accessed: 02-24-2017. URL: <http://tomcat.apache.org/> (cited on page 296).
- [457] G. Ostrouchov et al. Web Page. 2012. URL: <http://r-pbd.org/> (cited on page 205).
- [458] Overleaf. URL: <https://www.overleaf.com> (cited on page 29).
- [459] *Overview*. Online. The contact information listed on the webpage was for Yogesh Simmhan. URL: <http://dream-lab.cds.iisc.ac.in/about/overview/> (cited on page 212).
- [460] *Parallel Graph Analytics (PGX)*. Web Page. Accessed: 2017-02-11. URL: <http://www.oracle.com/technetwork/oracle-labs/parallel-graph-analytics/overview/index.html> (cited on page 210).
- [461] Parasol. Webpage. URL: <https://parasol.tamu.edu/research.php> (cited on page 212).
- [462] James J. (Jong Hyuk) Park et al. *Advanced Multimedia and Ubiquitous Engineering*. Edited by James J. (Jong Hyuk) Park et al. accessed 2017-02-13. Springer, 2016 (cited on page 235).
- [463] Mosha Pasumansky. *Inside Capacitor, BigQuery's next-generation columnar storage format*. Blog. Accessed: 2017-2-23. Apr. 2016. URL: <https://cloud.google.com/blog/big-data/2016/04/inside-capacitor-bigquery-s-next-generation-columnar-storage-format> (cited on page 226).
- [464] PBS. Web Page. URL: <http://www.pbspro.org/> (cited on page 269).
- [465] Pegasus. Web Page. Accessed:2/3/2017. URL: <https://pegasus.isi.edu/> (cited on page 196).
- [466] *Pentaho*. webpage. URL: <https://en.wikipedia.org/wiki/Pentaho> (cited on page 203).
- [467] Francisco Perez-Sorrosal et al. *Omid's First Step in the Apache Community*. Web Page. accessed 2017-02-25. Sept. 2016. URL: <https://yahoobeng.tumblr.com/post/151015726181/omids-first-step-in-the-apache-community> (cited on page 299).
- [468] Sebastian Peyrott. *API Gateway. An Introduction to Microservices, Part 2*. Website. Blog Post. Sept. 2015. URL: <https://auth0.com/blog/an-introduction-to-microservices-part-2-API-gateway/> (cited on page 300).
- [469] Rob Pike et al. “Interpreting the Data: Parallel Analysis with Sawzall”. In: *Scientific Programming Journal* 13.4 (Oct. 2005), pages 277–298. ISSN: 1058-9244. DOI: 10.1155/2005/962135. URL: <https://research.google.com/archive/sawzall-sciprog.pdf> (cited on page 227).

- [470] Keshav Pingali et al. “The Tao of Parallelism in Algorithms”. In: *The Tao of Parallelism in Algorithms*. Accessed: 2017-2-27. June 2011, pages 1–14. URL: <http://iss.ices.utexas.edu/Publications/Papers/pingali11.pdf> (cited on page 236).
- [471] Pivotal. Web Page. URL: <https://run.pivotal.io/features/> (cited on page 219).
- [472] *Pivotal Greenplum. The Open Source Massively Parallel Data Warehouse*. Webpage. URL: <https://pivotal.io/pivotal-greenplum> (cited on page 250).
- [473] *Pivotal Greenplum Database*. Webpage. URL: https://en.wikipedia.org/wiki/Pivotal_Greenplum_Database (cited on page 251).
- [474] *Pivotal HAWQ*. URL: <http://hawq.incubator.apache.org/> (cited on page 225).
- [475] *Pivotal HDB*. URL: <https://pivotal.io/pivotal-hdb> (cited on page 225).
- [476] *PLASMA README*. Web Page. URL: <http://icl.cs.utk.edu/projectsfiles/plasma/html/README.html> (visited on 02/13/2017) (cited on page 206).
- [477] Moritz Plassnig. *Heroku-style Application Deployments with Docker - DZone Cloud*. Web Page. Accessed: 2017-1-17. Nov. 2015. URL: <https://dzone.com/articles/heroku-style-application-deployments-with-docker> (cited on page 282).
- [478] PmWiki. *GFFS – Global Federated File System*. Web Page. accessed 2017-01-28. Jan. 2012. URL: <http://genesis2.virginia.edu/wiki/Main/GFFS> (cited on page 273).
- [479] Ian Pointer. *Apache Beam want to be uber-API for big data*. Web page. accessed 25-feb-2017. Apr. 2016. URL: <http://www.infoworld.com/article/3056172/application-development/apache-beam-wants-to-be-uber-api-for-big-data.html> (cited on page 297).
- [480] *PolyBase*. Web Page. URL: www.microsoft.com/en-us/library/mt143171.aspx (cited on page 225).
- [481] Florin Pop, Joanna Kolodziej, and Beniamino DiMartino. “Resource Management for Big Data Platforms”. In: Springer Nature, 2016, pages 49–50 (cited on page 299).
- [482] Florin Pop, Joanna Kolodziej, and Beniamino DiMartino. “Resource Management for Big Data Platforms”. In: Springer Nature, 2016, pages 50–51 (cited on page 302).
- [483] *Potree*. URL: <http://potree.org/wp/about/> (cited on page 215).
- [484] Herbert Pötzl. *Linux-Vserver*. Web Page. URL: www.linux-vserver.org (cited on page 286).
- [485] Vignesh Prajapati. *Big data analytics with R and Hadoop*. Packt Publishing, 2013. ISBN: 9781782163282 (cited on page 205).
- [486] Prashanth. *Facebook Tupperware*. Blog. Accessed: 2/18/2017. Dec. 2015. URL: <http://blog.cspp.in/index.php/2015/12/17/facebook-tupperware/> (cited on page 281).
- [487] *Project Jupyter*. Web page. URL: <http://www.jupyter.org> (visited on 02/27/2017) (cited on page 198).
- [488] UltraScan Project. *UltraScan Analysis Software*. Web page. Accessed: 2017-2-22. Apr. 2015. URL: <http://ultrascan.uthscsa.edu> (cited on page 196).
- [489] *Protocol Buffer*. Web Page. Sept. 2016. URL: <https://developers.google.com/protocol-buffers/> (cited on page 295).
- [490] Roshan Punnoose, Adina Crainiceanu, and David Rapp. “Rya: A Scalable RDF Triple Store for the Clouds”. In: *Proceedings of the 1st International Workshop on Cloud Intelligence. Cloud-I '12*. Istanbul, Turkey: ACM, 2012, 4:1–4:8. ISBN: 978-1-4503-1596-8. DOI: 10.1145/2347673.2347677. URL: <http://doi.acm.org/10.1145/2347673.2347677> (cited on page 258).
- [491] *Puppet FAQ*. Web Page. URL: <https://puppet.com/product/faq> (cited on page 277).
- [492] *Puppet software*. Web Page. URL: [https://en.wikipedia.org/wiki/Puppet_\(software\)](https://en.wikipedia.org/wiki/Puppet_(software)) (cited on pages 277, 278).

-
- [493] PuppetLabsRazor. *Home.PuppetLabs/Razor Wiki*. Web Page. Accessed: 2017-02-04. URL: <https://github.com/puppetlabs/Razor/wiki> (cited on page 279).
 - [494] PuppetLabsRazor. *Introducing Razor; a next Generation provisioning Solution-Puppet*. Web Page. Accessed: 2017-02-04. URL: <https://puppet.com/blog/introducing-razor-a-next-generation-provisioning-solution> (cited on page 279).
 - [495] Qemu. *QEMU*. WebPage. Feb. 2017. URL: http://wiki.qemu-project.org/index.php/Main_Page (cited on page 285).
 - [496] Quora. *LinkedIn*. Web page. accessed 18-March-2017. URL: <https://www.quora.com/What-is-LinkedIn-s-database-architecture-like> (cited on page 231).
 - [497] R: What is R? Web Page. Accessed: 2017-1-26. URL: <https://www.r-project.org/about.html> (cited on page 204).
 - [498] RabbitMQ, components. Web Page. Accessed: 2017-01-19. URL: <https://www.rabbitmq.com/> (cited on page 238).
 - [499] Tilmann Rabl et al. “Solving Big Data Challenges for Enterprise Application Performance Management”. In: *Proc. VLDB Endow.* 5.12 (Aug. 2012), pages 1724–1735. ISSN: 2150-8097. DOI: 10.14778/2367502.2367512. URL: <https://arxiv.org/pdf/1208.4167.pdf> (cited on page 253).
 - [500] Dan Radez. *OpenStack Essentials*. Packt Publishing Ltd., May 26, 2015. 182 pages. ISBN: 978-1-78398-708-5. URL: <http://ebook.konfigurasi.net/Openstack/OpenStack%5C%20Essentials.pdf> (cited on page 271).
 - [501] Ioan Raicu et al. “Falkon: a Fast and Light-weight tasK executiON framework”. In: *ACM SC07*. 2007 (cited on page 270).
 - [502] Siddharth Rao. *DREAM:Lab – Democratising Computing through the Cloud*. Online. Mar. 2016. URL: <http://iisc.researchmedia.center/article/dreamlab-%E2%80%93-democratising-computing-through-cloud> (cited on page 212).
 - [503] Rasdaman raster array database. Web Page. Accessed: 02-23-2017. URL: <http://www.rasdaman.org/> (cited on page 250).
 - [504] Syed Ali Raza. *Neo4j*. Presentation/ Slides. Last Accessed: 2017.02.25. URL: <https://www.slideshare.net/aliraza995/neo4j-graph-storage-27104408> (cited on page 259).
 - [505] RCFFile Cat. Web Page. Accessed: 2017-02-02. URL: <https://cwiki.apache.org/confluence/display/Hive/RCFFileCat> (cited on page 264).
 - [506] RCFFileCat - Apache Hive. Web Page. Accessed: 2017-1-27. URL: <https://cwiki.apache.org/confluence/display/Hive/RCFFileCat> (cited on page 264).
 - [507] Amazon RDS. *What is Amazon RDS*. Web Page. Accessed: 2017-02-04. URL: <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide>Welcome.html#Welcome.Concepts> (cited on page 251).
 - [508] Amazon RDS. *What is Amazon RDS?Definition*. Web Page. Accessed: 2017-02-04. URL: <http://searchaws.techtarget.com/definition/Amazon-Relational-Database-Service-RDS> (cited on page 251).
 - [509] Red Hat, Inc. *developers-openshift components*. Web Page. Accessed: 2017-1-26. URL: <https://developers.openshift.com/> (cited on page 217).
 - [510] Red Hat, Inc. *openshift components*. Web Page. Accessed: 2017-1-26. URL: <https://www.openshift.org/> (cited on page 217).
 - [511] Red Hat, Inc. *openshift-blog components*. Web Page. Accessed: 2017-1-27. URL: <https://blog.openshift.com/getting-started-with-openshift-origin-the-open-source-platform-as-a-service-paaS/> (cited on page 217).

- [512] Red Hat, Inc. *paas-openshift components*. Web Page. Accessed: 2017-1-25. URL: <https://blog.openshift.com/announcing-openshift-origin-the-open-source-platform-as-a-service-paaS/> (cited on page 217).
- [513] Red Hat, Inc. *Ansible Documentation*. Web Page. Accessed: 2017-02-12. Feb. 2015. URL: <https://docs.ansible.com/ansible/index.html> (cited on page 278).
- [514] Red Hat, Inc. *Ceph Homepage-Ceph*. Web Page. Accessed: 2017-1-26. Red Hat, Inc., 2017. URL: <https://ceph.com/> (cited on page 271).
- [515] Red Hat, Inc. *Ceph Filesystem - Ceph Documentation*. Web Page. Accessed: 2017-1-24. Red Hat, Inc. URL: <http://docs.ceph.com/docs/master/cephfs/> (cited on page 272).
- [516] Red Hat, Inc. *Ceph Architecture-Ceph Documentation*. Web Page. Accessed: 2017-1-24. 2017. URL: <http://docs.ceph.com/docs/master/architecture/> (cited on page 271).
- [517] Bloor Robin and Jozwiak Rebecca. *THE NATURE OF GRAPH DATA*. Technical report. Austin, TX 78720|512-524-3689: The Bloor Group, 2017. URL: <http://web.cray.com/bloor-graph-data> (cited on page 259).
- [518] Rocks. Web Page. 2017. URL: <https://www.rockscluster.org> (cited on page 280).
- [519] Ryan Rosario. *Exciting Tools for Big Data: S4, Sawzall and mrjob!* Web page. Online; accessed 30-jan-2017. Nov. 2010. URL: <http://www.bytetmining.com/2010/11/exciting-tools-for-big-data-s4-sawzall-and-mrjob/> (cited on page 227).
- [520] Apache Rya. *Apache Rya*. Online. URL: <https://rya.apache.org/> (cited on page 258).
- [521] *S4: S4 0.6.0 overview*. Web Page. Accessed: 2017-02-24. URL: <http://incubator.apache.org/s4/doc/0.6.0/overview> (cited on page 229).
- [522] *Sahara*. Web Page. Accessed: 1/16/2017. URL: <http://docs.openstack.org/developer/sahara/> (cited on page 280).
- [523] *saltstack components*. webpage. accessed: 2017-2-4. URL: <https://saltstack.com/> (cited on page 278).
- [524] *Samza*. Web Page. URL: <http://samza.apache.org/> (visited on 02/13/2017) (cited on page 229).
- [525] Abhijeet Sandil. *SSO Strategy: Authentication (SAML) –vs– Authorization (OAuth)*. Web Page. Accessed: 2017-02-04. URL: <https://www.linkedin.com/pulse/sso-strategy-authentication-vs-authorization-saml-oauth-sandil> (cited on page 294).
- [526] Lee Sangkeun, Rangan Sukumar Sreenivas, and Lim Seung-Hwan. “Graph mining meets the semantic web.” In: *2015 31st IEEE International Conference on Data Engineering Workshops*. (Seoul, South Korea). Edited by Johannes Gehrke, Wolfgang Lehner, and Kyuseok Shim. IEEE. Apr. 2015, pages 53–58. DOI: 10.1109/ICDEW.2015.7129544. URL: https://www.researchgate.net/profile/Sreenivas_Rangan_Sukumar/publication/273755615_Graph_Mining_Meets_the_Semantic_Web/links/550a27cb0cf20ed529e26260.pdf (cited on page 259).
- [527] netlib. *ScalAPACK Users Guide*. Web Page. Accessed: 2017-02-12. URL: <http://www.netlib.org/scalapack/slug/> (cited on page 206).
- [528] *Scaling Apache Giraph to a trillion edges*. web page. URL: <https://www.facebook.com/notes/facebook-engineering/scaling-apache-giraph-to-a-trillion-edges/10151617006153920> (cited on page 235).
- [529] Tom Schaul et al. “PyBrain”. In: *Journal of Machine Learning Research* 11 (Mar. 2010). Edited by Soeren Sonnenburg, pages 743–746. ISSN: 1532-4435. URL: <http://dl.acm.org/citation.cfm?id=1756006.1756030> (cited on page 208).
- [530] SchedMd. *Slurm website*. Web Page. Accessed: 2017-2-27. Mar. 2013. URL: <https://slurm.schedmd.com/overview.html> (cited on page 269).

- [531] SchedMd. *Slurm Supported Platforms*. Web Page. Accessed: 2017-2-27. Apr. 2015. URL: <https://slurm.schedmd.com/platforms.html> (cited on page 270).
- [532] scikit-Learn. webpage. URL: <https://en.wikipedia.org/wiki/Scikit-learn> (cited on page 207).
- [533] Sean Lynch. *Why Membase Uses Erlang*. web page. accessed 2017-01-29. Oct. 2010. URL: %7B<https://blog.couchbase.com/why-membase-uses-erlang%7D> (cited on page 256).
- [534] Adam Seligman. *Apache Phoenix - A Small Step for Big Data*. Web page. Online; accessed 25-jan-2017. May 2014. URL: <https://developer.salesforce.com/blogs/developer-relations/2014/05/apache-phoenix-small-step-big-data.html> (cited on page 223).
- [535] James Serra. *What is Azure Data Factory?* Web Page. Accessed: 02/03/2016. URL: <http://www.jamesserra.com/archive/2014/11/what-is-azure-data-factory/> (cited on page 201).
- [536] Vinay Jayarama Setty. “Publish/subscribe for large-scale social interaction: Design, analysis and resource provisioning”. Accessed:2017-1-29. PhD thesis. Faculty of Mathematics and Natural Sciences, University of Oslo: Department of Informatics,UNIVERSITY OF OSLO, Jan. 2015. URL: <https://www.duo.uio.no/bitstream/handle/10852/43117/1595-Setty-DUO-Thesis.pdf?sequence=1&isAllowed=y> (cited on page 236).
- [537] ShareLaTex. URL: <https://www.sharelatex.com> (cited on page 29).
- [538] Caro Caserio Sharon Lo Sreedhar Pelluru. *Introduction to Azure Data Factory Service, a data integration service in the cloud*. Web Page. Accessed: 02/03/2016. URL: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-introduction> (cited on page 201).
- [539] Xuanhua Shi et al. “GIRAFFE: A Scalable Distributed Coordination Service for Large-scale Systems”. In: *GIRAFFE: A Scalable Distributed Coordination Service for Large-scale Systems*. Accessed: 2017-2-27. Sept. 2014, pages 1–10. URL: <http://www.mcs.anl.gov/papers/P5157-0714.pdf> (cited on page 294).
- [540] Sumit Gupta Shilpi Saxena. *Real-Time Big Data Analytics*. 1st. 35 Livery Street, Birmingham B3 2PB, UK: Packt Publishing, 2016. ISBN: 9781784391409 (cited on page 230).
- [541] Abdul Ghaffar Shoro and Tariq Rahim Soomro. “Big data analysis: Apache spark perspective”. In: *Global Journal of Computer Science and Technology* 15.1 (2015). Accessed: 2017-1-21. ISSN: 0975-4172. URL: <http://www.computerresearch.org/index.php/computer/article/view/1137/1124> (cited on pages 204, 233).
- [542] Pallickara Shrideep et al. *granules*. Project. July 2016. URL: <http://granules.cs.colostate.edu/> (cited on page 230).
- [543] Julian Shun and Guy E. Blelloch. “Ligra: A Lightweight Graph Processing Framework for Shared Memory”. In: *ACM SIGPLAN Notices - PPoPP '13*. Pittsburgh, Pennsylvania, USA: ACM New York, NY, USA, 2013. ISBN: 978-1-4503-1922-5. DOI: 10.1145/2517327.2442530. URL: <http://dl.acm.org/citation.cfm?id=2442530> (cited on page 235).
- [544] Jeff Shute et al. “F1: The Fault-tolerant Distributed RDBMS Supporting Google’s Ad Business”. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’12. Accessed : 2017-02-15. Scottsdale, Arizona, USA: ACM, 2012, pages 777–778. ISBN: 978-1-4503-1247-9. DOI: 10.1145/2213836.2213954. URL: <http://doi.acm.org/10.1145/2213836.2213954> (cited on page 251).
- [545] Jeff Shute et al. “F1: A Distributed SQL Database That Scales”. In: *Proc. VLDB Endow.* 6.11 (Aug. 2013). Accessed:2017-02-15, pages 1068–1079. ISSN: 2150-8097. DOI: 10.14778/2536222.2536232. URL: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41344.pdf> (cited on page 251).

- [546] Abraham Silberschatz et al. *Operating system concepts*. Volume 4. Addison-wesley Reading, 1998 (cited on page 242).
- [547] SINTEF. *CloudML*. Web page. accessed 15-March-2017. 2013. URL: <http://cloudml.org/> (cited on page 283).
- [548] SINTEF Git. *CloudML Wiki*. Web page. accessed 16-March-2017. Last updated in 2 Oct 2015. URL: <https://github.com/SINTEF-9012/cloudml/wiki> (cited on page 283).
- [549] Mark Slee, Aditya Agarwal, and Marc Kwiatkowski. “Thrift: Scalable Cross-Language Services Implementation”. In: *Thrift* (Apr. 1, 2007). URL: <https://thrift.apache.org/static/files/thrift-20070401.pdf> (cited on page 295).
- [550] Slideshare. *Apache UIMA Introduction*. Web Page. Accessed: 02/03/2016. URL: <http://www.slideshare.net/teofili/apache-uima-introduction> (cited on page 247).
- [551] *SLURM*. Web Page. URL: <https://slurm.schedmd.com/> (cited on page 269).
- [552] Ken Smith. *Azure: What to Use, What to Avoid*. Web page. Online; accessed 28-jan-2017. Oct. 2014. URL: <http://blog.wouldbethelogian.com/2014/10/azure-what-to-use-what-to-avoid.html> (cited on page 261).
- [553] Russell Smith. *What are the advantages and disadvantages of using MongoDB vs CouchDB vs Cassandra vs Redis?* Web page. Online; accessed 29-jan-2017. Nov. 2015. URL: <https://www.quora.com/What-are-the-advantages-and-disadvantages-of-using-MongoDB-vs-CouchDB-vs-Cassandra-vs-Redis> (cited on page 256).
- [554] Amazon SNS. Web Page. Accessed: 2017-02-02. URL: <https://aws.amazon.com/sns/> (cited on page 241).
- [555] Amazon SNS Blog. Web Page. Accessed: 2017-02-02. URL: <https://aws.amazon.com/blogs/aws/introducing-the-amazon-simple-notification-service/> (cited on page 241).
- [556] Amazon SNS FAQs. Web Page. Accessed: 2017-02-02. URL: <https://aws.amazon.com/sns/faqs/> (cited on page 241).
- [557] Software » OpenStack Open Source Cloud Computing Software. Web Page. Online; accessed 23-Feb-2017. URL: <https://www.openstack.org/software/> (cited on page 287).
- [558] Borja Sotomayor and Lisa Childers. *Globus® Toolkit 4: Programming Java Services*. Morgan Kaufmann, 2006 (cited on page 270).
- [559] Apache Software Foundation. *Spark Programming Guide - Spark 2.1.0 Documentation*. Web Page. Accessed: 04-09-2017. URL: <http://spark.apache.org/docs/latest/programming-guide.html#resilient-distributed-datasets-rdds> (cited on page 232).
- [560] Evan R Sparks et al. “KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics”. In: *arXiv preprint arXiv:1610.09451* (2016) (cited on page 203).
- [561] Evan Sparks and Shivaram Venkataraman. *Building Large Scale Machine Learning Applications with Pipelines-(Ev...)* URL: <https://www.slideshare.net/SparkSummit/evan-sparks> (visited on 02/27/2017) (cited on page 203).
- [562] E. Sparks et al. “MLI: An API for Distributed Machine Learning”. In: *MLI: An API for Distributed Machine Learning*. IEEE 13th International Conference on Data Mining. 2013 (cited on page 204).
- [563] Apache Software Foundation. *Spark Streaming - Spark 2.1.0 Documentation*. Web Page. Accessed: 04-09-2017. URL: <http://spark.apache.org/docs/latest/streaming-programming-guide.html> (cited on page 232).
- [564] Splunk, Inc. *Splunk Enterprise Overview*. Web Page. accessed: 2017-02-18. Feb. 2015. URL: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Overview/AboutSplunkEnterprise> (cited on page 214).

-
- [565] SQLite. *About SQLite*. Website. URL: <https://www.sqlite.org/about.html> (cited on page 248).
 - [566] SQLite. *Appropriate Uses For SQLite*. Website. URL: <https://www.sqlite.org/whentouse.html> (cited on page 248).
 - [567] *SQL Server Wiki*. Web Page. URL: https://en.wikipedia.org/wiki/Microsoft_SQL_Server (cited on page 248).
 - [568] *SSH - Wikipedia*. Web Page. Accessed: 2017-01-26. URL: https://en.wikipedia.org/wiki/Secure_Shell (cited on page 265).
 - [569] *OpenSSH - Wikipedia*. Web Page. Accessed: 2017-01-26. URL: <https://en.wikipedia.org/wiki/OpenSSH> (cited on page 265).
 - [570] *Difference between Application Manager and Application Master in YARN?* StackOverflow. 2015. URL: <http://stackoverflow.com/questions/30967247/difference-between-application-manager-and-application-master-in-yarn> (cited on page 267).
 - [571] Michael Stonebraker. “SciDB: An Open-Source DBMS for Scientific Data”. In: *ERCIM News* 89 (Apr. 2012), page 56. URL: <https://ercim-news.ercim.eu/en89/special/scidb-an-open-source-dbms-for-scientific-data> (cited on page 250).
 - [572] *Apache Storm - Concepts*. webpage. Accessed: 01-22-2017. URL: <http://storm.apache.org/releases/1.0.2/Concepts.html> (cited on page 229).
 - [573] Dan Sullivan. *Paas Provider Comparison Guide: Engine Yard - Orchestration and Management*. Article. Accessed: 02-26-2017. July 2013. URL: http://www.tomsitpro.com/articles/paas-engine-yard-amazon-elastic-cloud-computing_2-578.html (cited on page 220).
 - [574] Nick Sullivan. *Kyoto Tycoon Secure Replication*. Web Page. Accessed: 02/03/2016. URL: <https://blog.cloudflare.com/kyoto-tycoon-secure-replication/> (cited on page 254).
 - [575] Bruce Synder and Rob Davies. *ActiveMQ in Action*. 1st. Manning publications, 2013. ISBN: 1933988940, 9781933988948 (cited on page 238).
 - [576] Istvan Szegedi. *Apache Phoenix - An SQL Driver for HBase*. Web page. Online; accessed 23-jan-2017. May 2014. URL: <https://bighadoop.wordpress.com/2014/05/17/apache-phoenix-an-sql-driver-for-hbase/> (cited on page 224).
 - [577] *Tableau Tutorial*. Web Page. Accessed: 2017-2-10. URL: <https://casci.umd.edu/wp-content/uploads/2013/12/Tableau-Tutorial.pdf> (cited on page 215).
 - [578] *Tableau Technology*. Web Page. Accessed: 2017-2-10. URL: <https://www.tableau.com/products/technology> (cited on page 215).
 - [579] A. Talwalkar et al. “MLbase: A Distributed Machine Learning Wrapper”. In: *MLbase: A Distributed Machine Learning Wrapper*. Big Learning Workshop at NIPS. 2012 (cited on page 204).
 - [580] *Taverna*. Web Page. URL: <https://taverna.incubator.apache.org/introduction/> (cited on page 197).
 - [581] Illinois Institute of Technology Department of Computer Science. *ZHT: a zero-hop distributed hashtable*. Online. URL: <http://datasys.cs.iit.edu/projects/ZHT/> (cited on page 253).
 - [582] Techopedia. *Amazon Web Services (AWS)*. Web Page. URL: <https://www.techopedia.com/definition/26426/amazon-web-services-aws> (cited on page 289).
 - [583] TechTarget. *Lightweight Directory Access Protocol*. Web Page. accessed 2017-01-30. Feb. 2017. URL: <http://searchmobilecomputing.techtarget.com/definition/LDAP> (cited on page 293).

- [584] *TensorFlow*. Web Page. Accessed: 2017-1-24. URL: <https://www.tensorflow.org> (cited on page 216).
- [585] *Terraform components*. Web Page. Accessed: 2017-02-12. URL: <https://www.terraform.io/intro/index.html> (cited on page 283).
- [586] *Tez*. Web Page. 2017. URL: <https://hortonworks.com/apache/tez> (cited on page 199).
- [587] The Apache Software Foundation. Web Page. URL: <http://sqoop.apache.org/> (cited on page 266).
- [588] The Apache Software Foundation. Web Page. URL: <http://ant.apache.org/> (cited on page 300).
- [589] The Apache Software Foundation. *Apache Flume*. web page. accessed 2017-02-06. URL: <https://flume.apache.org/index.html> (cited on page 265).
- [590] The Apache Software Foundation. *Getting Started with Derby*. Web Page. accessed: 2017-02-18. Sept. 2015. URL: <https://db.apache.org/derby/docs/10.12/getstart/index.html> (cited on page 250).
- [591] The Apache Software Foundation. *Apache Derby*. Web Page. accessed: 2017-02-18. Oct. 2016. URL: <https://db.apache.org/derby/> (cited on page 250).
- [592] The Apache Software Foundation. *Apache Derby Project Charter*. Web page. accessed: 2017-02-18. Sept. 2016. URL: https://db.apache.org/derby/derby_charter.html (cited on page 250).
- [593] The Apache Software Foundation. *Spark SQL*. Web Page. accessed 2017-02-19. Feb. 2016. URL: <http://spark.apache.org/sql/> (cited on page 252).
- [594] The Apache Software Foundation. *Apache CloudStack*. Web Page. Accessed: 2017-02-11. Feb. 2017. URL: <https://cloudstack.apache.org/> (cited on page 288).
- [595] The Disco Project. *What is Disco*. Web page. accessed 25-feb-2017. Feb. 2017. URL: <http://disco.readthedocs.io/en/develop/intro.html> (cited on page 234).
- [596] The Disco Project. *Why Not Hadoop?* Web page. accessed 25-feb-2017. Feb. 2017. URL: <http://disco.readthedocs.io/en/develop/faq.html#why-not-hadoop> (cited on page 234).
- [597] *The Eucalyptus Open-Source Private Cloud*. Web Page. Accessed: 2017-02-11. URL: <http://www.cloudbook.net/resources/stories/the-eucalyptus-open-source-private-cloud> (cited on page 287).
- [598] *The Jupyter notebook — Jupyter Notebook 5.0.0.dev documentation*. Web page. URL: <https://jupyter-notebook.readthedocs.io/en/latest/index.html> (visited on 02/27/2017) (cited on page 198).
- [599] *The LXD container hypervisor*. web page. URL: <https://www.ubuntu.com/containers/lxd> (cited on page 300).
- [600] The Open MPI Project. *Open MPI: Open Source High Performance Computing*. Web Page. Accessed: 2017-2-22. Feb. 2017. URL: <https://www.open-mpi.org> (cited on page 296).
- [601] The PostgreSQL Global Development Group. *PostgreSQL 9.5: UPSERT, Row Level Security, and Big Data*. Web Page. Accessed: 2017-4-10. URL: <https://www.postgresql.org/about/> (cited on page 249).
- [602] The PostgreSQL Global Development Group. *PostgreSQL 9.5: UPSERT, Row Level Security, and Big Data*. Web Page. Accessed: 2017-4-10. URL: <https://www.postgresql.org/about/news/1636/> (cited on page 249).
- [603] The Windows Club. *Understanding Blob, Queue and Table Storage for Windows Azure*. Web page. Online; accessed 28-jan-2017. Jan. 2017. URL: <http://www.thewindowsclub.com/understanding-blobqueueutable-storage-windows-azure> (cited on page 261).

- [604] *theano*. Web Page. Accessed: 2017-1-21. URL: <http://deeplearning.net/software/theano/introduction.html> (cited on page 209).
- [605] Chandu Thekkath. *Naiad - Microsoft Research*. webpage. Accessed: 2017-1-27. Microsoft, Oct. 2011. URL: <https://www.microsoft.com/en-us/research/project/naiad/> (cited on page 199).
- [606] *Three.js*. Web Page. Accessed: 2017-02-27. Mar. 2017. URL: <https://en.wikipedia.org/wiki/Three.js> (cited on page 215).
- [607] *Creating a scene*. Web Page. Accessed: 2017-02-27. Dec. 2016. URL: https://threejs.org/docs/index.html#Manual/Getting_Started/Creating_a_scene (cited on page 215).
- [608] Apache. *About Thrift*. Web Page. Accessed: 2017-02-12. URL: <https://thrift.apache.org/> (cited on page 295).
- [609] *Apache Tika*. Web Page. URL: <https://tika.apache.org/> (cited on page 247).
- [610] *TinkerPop*. Web Page. Accessed: 2/6/2017. URL: <http://tinkerpop.apache.org/> (cited on page 261).
- [611] *Titan DB*. Web Page. Accessed: 2/4/2017. URL: <http://titan.thinkaurelius> (cited on page 260).
- [612] *Tokyo Cabinet*. Web Page. Accessed: 2017-1-27. URL: <http://fallabs.com/tokyocabinet/> (cited on page 254).
- [613] Stanimire Tomov, Jack Dongarra, and Marc Baboulin. “Towards dense linear algebra for hybrid GPU accelerated manycore systems”. In: *Parallel Computing* 36.5 (2010), pages 232–240. URL: <http://www.sciencedirect.com/science/article/pii/S0167819109001276> (visited on 02/13/2017) (cited on page 206).
- [614] Stanimire Tomov et al. “Dense linear algebra solvers for multicore with GPU accelerators”. In: *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*. IEEE, 2010, pages 1–8. URL: http://link.springer.com/chapter/10.1007/978-3-319-06548-9_1 (visited on 02/13/2017) (cited on page 206).
- [615] Cardiff University. *Triana Documentation*. Web Page. Accessed: 2017-2-20. Cardiff University, 2012. URL: <http://www.trianacode.org/> (cited on page 197).
- [616] Craig Trim. *Jen: Semantic Web Framework*. Web Page. Accessed: 02/03/2016. URL: <http://trimc-nlp.blogspot.com/2013/06/introduction-to-jena.html> (cited on page 261).
- [617] *Taverna Workflows: Syntax and Semantics*. IEEE, Dec. 2007. ISBN: 0-7695-3064-8. DOI: 10.1109/E-SCIENCE.2007.71. URL: <http://ieeexplore.ieee.org/document/4426917/> (cited on page 197).
- [618] Matteo Turilli, Mark Santcroos, and Shantenu Jha. “A Comprehensive Perspective on Pilot-Job Systems”. In: *ACM arXiv*. Mar. 2016, pages 1–26 (cited on page 270).
- [619] James Turnbull. *The Terraform Book*. 110th edition. 9780988820258. Turnbull Press, Nov. 2016 (cited on page 283).
- [620] tutorialspoint. *SQLite Overview*. Website. URL: https://www.tutorialspoint.com/sqlite/sqlite_overview.htm (cited on page 248).
- [621] *Microsoft Azure - Queues*. Web Page. Accessed: 2017-2-10. URL: https://www.tutorialspoint.com/microsoft_azure/microsoft_azure_queues.htm (cited on page 242).
- [622] Dylan Tweney. *AppFog gives developers an easier way to deploy cloud apps (interview)*. Online. May 2012. URL: <http://venturebeat.com/2012/05/15/appfog-gives-developers-an-easier-way-to-deploy-cloud-apps-interview/> (cited on page 220).

- [623] *Twister: Iterative MapReduce*. Web Page. URL: <http://www.iterativemapreduce.org/> (visited on 02/13/2017) (cited on page 233).
- [624] Twitter. *Flying faster with Twitter Heron*. Web Page. Accessed: 2017-02-04. URL: <https://blog.twitter.com/2015/flying-faster-with-twitter-heron> (cited on page 231).
- [625] Twitter. *Open Sourcing Twitter Heron*. Web Page. Accessed: 2017-02-04. URL: <https://blog.twitter.com/2016/open-sourcing-twitter-heron> (cited on page 231).
- [626] *Kyoto Tycoon*. Web Page. URL: <http://fallabs.com/kyototycoon/> (cited on page 255).
- [627] *Tyrant FALLabs*. Web Page. URL: <http://fallabs.com/tokyotyrant/> (cited on page 255).
- [628] *Tyrant Blog*. Web Page. URL: https://www.percona.com/blog/2009/10/19/mysql-memcached_tyrant_part3/ (cited on page 254).
- [629] *Understanding LXC and LXD*. web page. URL: <http://thevar guy.com/open-source-application-software-companies/understanding-lxc-and-lxd-canonicals-open-source-containe> (cited on page 300).
- [630] ISS team - University of Texas. *Galois website*. Web Page. Accessed: 2017-2-27. URL: <http://iss.ices.utexas.edu/?p=projects/galois> (cited on page 236).
- [631] *University Policies: Cheating and Plagiarism, ACA-72*. Web Page. updates 1975. 1961. URL: <https://policies.iu.edu/policies/aca-72-cheating-plagiarism/index.html> (cited on page 28).
- [632] *Using Amazon S3*. Web Page. Accessed: 2017-1-27. URL: <http://docs.aws.amazon.com/AmazonS3/latest/gsg/CopyingAnObject.html> (cited on page 273).
- [633] Devananda Van Der Veen and Llama Wrangler. *Introduction to Ironic*. Web Page. Accessed: 2017-2-27. Mar. 2015. URL: <https://docs.openstack.org/developer/ironic/deploy/user-guide.html> (cited on page 281).
- [634] Venkat Venkataramani. *What is the TAO cache used for at Facebook*. Web Page. June 2013. URL: <https://www.quora.com/What-is-the-TAO-cache-used-for-at-Facebook> (cited on page 260).
- [635] Magister Vis. *What Is Lumberyard? (Lumberyard Tutorials Series #1) - YouTube*. URL: <https://www.youtube.com/watch?v=Fwo3KqSsUI> (visited on 02/27/2017) (cited on page 228).
- [636] VMware. web-page. accessed 2017-02-13. URL: <http://www.vmware.com/products/esxi-and-esx.html> (cited on page 289).
- [637] vmware. *vCloud*. Webpage. URL: <http://www.vmware.com/products/vcloud-suite.html> (cited on page 289).
- [638] LinkedIn. *Project Voldemort*. Web Page. Accessed: 2017-1-17. URL: <http://www.project-voldemort.com/voldemort/> (cited on page 253).
- [639] VoltDB. Web Page. URL: <https://www.wired.com/2016/04/kites-coding-assistant-spots-errors-finds-better-open-source/> (cited on page 222).
- [640] VoltDB. Web Page. URL: <https://www.voltdb.com/> (cited on page 245).
- [641] w3. *Apache Jena*. Web Page. Accessed: 2016.02.03. URL: https://www.w3.org/2001/sw/wiki/Apache_Jena (cited on page 261).
- [642] Liya Wang, Peter Van Buren, and Doreen Ware. “Architecting a distributed bioinformatics platform with iRODS and iPlant Agave API”. In: *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*. Dec. 2015. ISBN: 9781467397957 (cited on page 221).

- [643] Pau Kiat Wee. "Instant AppFog". In: Packt Publishing, July 2013. Chapter 1. URL: <https://www.packtpub.com/mapt/book/Web-Development/9781782167624/1/ch01lvl1sec03/So,+what+is+AppFog%3F> (cited on pages 219, 220).
- [644] Sage A. Weil et al. "Ceph: A scalable, high-performance distributed file system". In: *Proceedings of the 7th symposium on Operating systems design and implementation*. OSDI '06. Accessed: 2017-1-26. Seattle, Washington: USENIX Association, Nov. 2006, pages 307–320. ISBN: 1-931971-47-1. URL: https://www.usenix.org/legacy/event/osdi06/tech/full_papers/weil/weil.pdf (cited on page 272).
- [645] Johannes Wettinger. *any2api - The better way to create awesome APIs*. Web Page. Accessed: 2017-02-02. URL: <http://www.any2api.org/> (cited on page 284).
- [646] Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann. "DevOpSlang – Bridging the Gap Between Development and Operations". In: *Proceedings of the 3rd European Conference on ServiceOriented and Cloud Computing (ESOCC 2014)*. Lecture Notes in Computer Science (LNCS). accessed 2017-02-26. SpringerVerlag, 2014, pages 108–122 (cited on page 284).
- [647] Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann. "Any2API Automated APIfication". In: *CLOSER 2015 - Proceedings of the 5th International Conference on Cloud Computing and Services Science*. Edited by Markus Helfert, Donald Ferguson, and Víctor Méndez Muñoz. Lisbon, Portugal: SciTePress, 20-22 May 2015, pages 475–486. DOI: 10.5220/0005472704750486. URL: <https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf> (cited on page 284).
- [648] *What is a public cloud?* Webpage. URL: <http://www.interoute.com/cloud-article/what-public-cloud> (cited on page 290).
- [649] *What Is Amazon Route 53?* Web Page. Accessed: 2017-02-24. URL: <http://docs.aws.amazon.com/Route53/latest/DeveloperGuide>Welcome.html> (cited on page 290).
- [650] *What is HTCondor?* Web Page. URL: <https://research.cs.wisc.edu/htcondor/description.html> (cited on page 268).
- [651] *What is hubzero?* Web Page. URL: <https://hubzero.org/documentation/1.0.0/installation> (cited on page 221).
- [652] *What is the Jupyter Notebook? — Jupyter Notebook 5.0.0.dev documentation*. Web page. (Visited on 02/27/2017) (cited on page 198).
- [653] *Whirr technology*. webpage. accessed: 2017 – 2 – 4. URL: <https://whirr.apache.org/> (cited on page 276).
- [654] *Whirr technology*. webpage. accessed: 2017 – 2 – 4. URL: <http://www.slideshare.net/huguk/apache-whirr> (cited on page 276).
- [655] *Why Oozie? | Just a simple Hadoop DBA*. Web Page. URL: <https://prodlife.wordpress.com/2013/12/09/why-oozie/> (visited on 02/13/2017) (cited on page 199).
- [656] *Why use LXC (LinuX Containers) ?* web page. URL: <http://www.jpablo128.com/why-use-lxc-linux-containers/> (cited on page 286).
- [657] Wikipedia. *Apache Flex*. Web Page. Accessed: 2017-03-06. Mar. 2017. URL: https://en.wikipedia.org/wiki/Apache_Flex (cited on page 302).
- [658] Wikipedia. Web Page. URL: <https://en.wikipedia.org/wiki/Sqoop> (cited on page 266).
- [659] Wikipedia. Web Page. URL: https://en.wikipedia.org/wiki/Portable_Batch_System (cited on page 269).
- [660] Wikipedia. web-page. accessed 2017-02-13. URL: https://en.wikipedia.org/wiki/VMware_ESXi (cited on page 289).
- [661] Wikipedia. *Apache Apex wiki*. Web Page. Accessed: 2017-02-26. URL: https://en.wikipedia.org/wiki/Apache_Apex (cited on page 300).

- [662] Wikipedia. *DataNucleus Wiki*. Web Page. Accessed: 2017-02-04. URL: <https://en.wikipedia.org/wiki/DataNucleus> (cited on page 246).
- [663] Wikipedia. *Java Message Service - Wikipedia*. webpage. Accessed : 01-18-2017. URL: https://en.wikipedia.org/wiki/Java_Message_Service (cited on page 240).
- [664] Wikipedia. *Kubernetes Wiki*. Web Page. URL: <https://en.wikipedia.org/wiki/Kubernetes> (cited on page 282).
- [665] Wikipedia. *Neo4j*. Web page. Last Accessed: 2017.02.25. URL: <https://en.wikipedia.org/wiki/Neo4j> (cited on page 259).
- [666] Wikipedia. *UIMA*. Web Page. Accessed: 2017.02.03. URL: <https://en.wikipedia.org/wiki/UIMA> (cited on page 247).
- [667] Wikipedia. *Apache tomcat — Wikipedia, The Free Encyclopedia*. [Accessed: 02-24-2017]. 2010. URL: https://en.wikipedia.org/wiki/Apache_Tomcat (cited on page 296).
- [668] Wikipedia. *EclipseLink — Wikipedia, The Free Encyclopedia*. [Accessed: 02-06-2017]. 2010. URL: <https://en.wikipedia.org/wiki/EclipseLink> (cited on page 246).
- [669] Wikipedia. *Torch (Machine Learning) — Wikipedia, The Free Encyclopedia*. [Accessed: 02-06-2017]. 2014. URL: [https://en.wikipedia.org/wiki/Torch_\(machine_learning\)](https://en.wikipedia.org/wiki/Torch_(machine_learning)) (cited on page 209).
- [670] Wikipedia. *CloudControl*. Wiki Page. Feb. 2016. URL: <https://en.wikipedia.org/wiki/CloudControl> (cited on page 220).
- [671] Wikipedia. *Key-value database*. WebPage. Nov. 2016. URL: https://en.wikipedia.org/wiki/Key-value_database (cited on page 244).
- [672] Wikipedia. *Rasdaman — Wikipedia, The Free Encyclopedia*. [Accessed: 02-23-2017]. 2016. URL: <https://en.wikipedia.org/wiki/Rasdaman> (cited on page 250).
- [673] Wikipedia. *Spanner (Database)*. Web Page. accessed 2017-01-29. Oct. 2016. URL: [https://en.wikipedia.org/wiki/Spanner_\(database\)](https://en.wikipedia.org/wiki/Spanner_(database)) (cited on page 257).
- [674] Wikipedia. *Amazon Redshift*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: https://en.wikipedia.org/wiki/Amazon_Redshift (cited on page 226).
- [675] Wikipedia. *Ansible (software)*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: [https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software)) (cited on page 278).
- [676] Wikipedia. *Apache CloudStack*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: https://en.wikipedia.org/wiki/Apache_CloudStack (cited on page 288).
- [677] Wikipedia. *Apache Flink*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: https://en.wikipedia.org/wiki/Apache_Flink (cited on page 234).
- [678] Wikipedia. *Apache Hadoop*. Web Page. accessed 2017-03-18. Mar. 2017. URL: https://en.wikipedia.org/wiki/Apache%5C_Hadoop (cited on page 233).
- [679] Wikipedia. *Apache Phoenix — Wikipedia, The Free Encyclopedia*. Web page. Online; accessed 25-jan-2017. Jan. 2017. URL: https://en.m.wikipedia.org/wiki/Apache%5C_Phoenix (cited on page 223).
- [680] Wikipedia. *Bigtable*. Web Page. accessed 2017-01-29. Jan. 2017. URL: <https://en.wikipedia.org/wiki/Bigtable> (cited on page 257).
- [681] Wikipedia. *Chef (Software)*. Web Page. accessed 2017-01-26. Jan. 2017. URL: [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software)) (cited on page 281).
- [682] Wikipedia. *Cloud computing — Wikipedia, The Free Encyclopedia*. web page. Online; accessed 21-jan-2017. Jan. 2017. URL: https://en.wikipedia.org/wiki/Cloud_computing (cited on page 218).
- [683] Wikipedia. *CUDA*. Web Page. Online; accessed 25-Feb-2017. Feb. 2017. URL: <https://en.wikipedia.org/wiki/CUDA> (cited on page 297).

-
- [684] Wikipedia. *Data Analytics Acceleration Library* — Wikipedia, The Free Encyclopedia. [Accessed: 02-23-2017]. 2017. URL: https://en.wikipedia.org/wiki/Data_Analytics_Acceleration_Library (cited on page 208).
 - [685] Wikipedia. *Deeplearning4j*. Web Page. Accessed: 2017-02-11. Feb. 2017. URL: <https://en.wikipedia.org/wiki/Deeplearning4j> (cited on page 209).
 - [686] Wikipedia. *Erlang (programming language)* — Wikipedia, The Free Encyclopedia. Web page. Online; accessed: 29-jan-2017. Jan. 2017. URL: [https://en.wikipedia.org/wiki/Erlang_\(programming_language\)](https://en.wikipedia.org/wiki/Erlang_(programming_language)) (cited on page 256).
 - [687] Wikipedia. *Fiddler Software*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: [https://en.wikipedia.org/wiki/Fiddler_\(software\)](https://en.wikipedia.org/wiki/Fiddler_(software)) (cited on page 296).
 - [688] Wikipedia. *Graph Database*. Web Page. accessed 2017-01-30. Feb. 2017. URL: https://en.wikipedia.org/wiki/Graph_database (cited on page 259).
 - [689] Wikipedia. *Hazelcast*. Web Page. accessed 2017-01-29. Jan. 2017. URL: <https://en.wikipedia.org/wiki/Hazelcast> (cited on page 244).
 - [690] Wikipedia. *Hierarchical Data Format* — Wikipedia, The Free Encyclopedia. Web Page. Online,accessed: 2017-1-28. Feb. 2017. URL: https://en.wikipedia.org/wiki/Hierarchical_Data_Format (cited on page 263).
 - [691] Wikipedia. *Hyper-V*. Web Page. Accessed: 2017-02-23. Feb. 2017. URL: <https://en.wikipedia.org/wiki/Hyper-V> (cited on page 285).
 - [692] Wikipedia. *Hypervisor*. WebPage. Feb. 2017. URL: <https://en.wikipedia.org/wiki/Hypervisor> (cited on page 285).
 - [693] Wikipedia. *IBM Cloudant*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: <https://en.wikipedia.org/wiki/Cloudant> (cited on page 256).
 - [694] Wikipedia. *IBM General Parallel File System*. Web Page. accessed 2017-01-29. Jan. 2017. URL: https://en.wikipedia.org/wiki/IBM_General_Parallel_File_System (cited on pages 272, 273).
 - [695] Wikipedia. *Relational database*. WebPage. Jan. 2017. URL: https://en.wikipedia.org/wiki/Relational_database (cited on page 244).
 - [696] Wikipedia. *Snort Software*. Web Page. Accessed: 2017-02-12. Feb. 2017. URL: [https://en.wikipedia.org/wiki/Snort_\(software\)](https://en.wikipedia.org/wiki/Snort_(software)) (cited on page 295).
 - [697] Wikipedia. *Swift (programming language)*. Web Page. Accessed: 2017-02-23. Feb. 2017. URL: [https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)) (cited on page 196).
 - [698] Wikipedia. *Fusion Table Support*. Web Page. Last updated in 1 November 2016. URL: <https://support.google.com/fusiontables/answer/171181?hl=en> (cited on page 212).
 - [699] Wikipedia. *LinkedIn*. Web page. accessed 17-March-2017. This page was last modified on 18 March 2017, at 22:27. URL: <https://cloud.google.com/ml-engine/docs/concepts/technical-overview> (cited on page 231).
 - [700] Brandon Wiley. “Distributed Hash TTable, Part 1”. In: *Linux Journal* 114 (Oct. 2003). From issue number 114. URL: <http://www.linuxjournal.com/article/6797?page=0,0> (cited on page 253).
 - [701] Duncan C.E Winn. *Cloud Foundry-The Cloud Native Platform*. O'Reilley Media, 2016, page 70. ISBN: 978-4919-6573-3. URL: <https://books.google.com/books?id=XP2wDAAAQBAJ&printsec=frontcover&dq=Cloud+foundry> (cited on page 218).
 - [702] Alex Woodie. *Spark Gets New Machine Learning Framework: KeystoneML*. URL: <https://www.datanami.com/2015/05/26/spark-gets-new-machine-learning-framework-keystoneml/> (visited on 02/27/2017) (cited on page 203).

-
- [703] Alex Woodie. *Apache Beam's Ambitious Goal: Unify Big Data Development*. Web page. accessed 25-feb-2017. Apr. 2016. URL: <https://www.datanami.com/2016/04/22/apache-beam-emerges-ambitious-goal-unify-big-data-development/> (cited on page 297).
 - [704] Apache Software Foundation. *Apache Ranger*. Web Page. Online; accessed 9-Mar-2017. URL: <http://ranger.apache.org/> (cited on page 302).
 - [705] CASK. *CASK - The First Unified Integration Platform for Big Data*. Web Page. Online; accessed 18-Feb-2017. URL: <http://cask.co/products/cdap/> (cited on page 298).
 - [706] CASK. *Getting Started Developing with CDAP*. Web Page. Online; accessed 18-Feb-2017. URL: <http://docs.cask.co/cdap/current/en/developers-manual/getting-started/index.html> (cited on page 298).
 - [707] Open Source Data Turbine Initiative. Web Page. Accessed: 2017-2-26. URL: <http://dataturbine.org/> (cited on page 232).
 - [708] Jeff Lindsay. *Gitreceive*. Web Page. Accessed: 2017-2-13. Feb. 2016. URL: <https://github.com/progrium/gitreceive> (cited on page 282).
 - [709] High Performance ParalleX (HPX-5). Web Page. Accessed: 2017-1-17. URL: <https://hpx.crest.iu.edu/> (cited on page 237).
 - [710] HPX-5: user guide. Accessed: 2017-1-17. HPX-5. URL: https://hpx.crest.iu.edu/users_guide (cited on page 237).
 - [711] Open Grid Forum. *Open Cloud Computing Interface*. Web Page. Accessed: 2017-1-17. URL: <http://occi-wg.org/> (cited on page 275).
 - [712] Ralf Nyren et al. *Open Cloud Computing Interface Core*. OGF Published Document GWD-R-P.221. Accessed: 2017-1-17. Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA: Global Grid Forum, Sept. 2016. URL: <https://www.ogf.org/documents/GFD.221.pdf> (cited on page 275).
 - [713] Ralf Nyren, Andy Edmonds, and Thijs Metsch atnd Boris Parak. *Open Cloud Computing Interface - HTTP Protocol*. OGF Published Document GWD-R-P.223. Accessed: 2017-1-17. Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA: Global Grid Forum, Sept. 2016. URL: <https://www.ogf.org/documents/GFD.223.pdf> (cited on page 276).
 - [714] Ralf Nyren et al. *Open Cloud Computing Interface -JSON Rendering*. OGF Published Document GWD-R-P.226. Accessed: 2017-1-17. Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA: Global Grid Forum, Sept. 2016. URL: <https://www.ogf.org/documents/GFD.226.pdf> (cited on page 276).
 - [715] Michel Drescher, Boris Parak, and David Wallom. *OCCI Compute Resource Templates Profile*. OGF Published Document GWD-R-P.222. Accessed: 2017-1-17. Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA: Global Grid Forum, Apr. 2015. URL: <https://www.ogf.org/documents/GFD.222.pdf> (cited on page 276).
 - [716] Andy Edmonds and Thijs Metsch. *Open Cloud Computing Interface - Text Rendering*. OGF Published Document GWD-R-P.229. Accessed: 2017-1-17. Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA: Global Grid Forum, Sept. 2016. URL: <https://www.ogf.org/documents/GFD.229.pdf> (cited on page 276).
 - [717] Hortonworks. *Apache Ranger - Overview*. Web Page. Online; accessed 9-Mar-2017. URL: https://hortonworks.com/apache/ranger/#section_2 (cited on page 302).
 - [718] Syed Mahmood and Srikanth Venkat. *FOR YOUR EYES ONLY: DYNAMIC COLUMN MASKING & ROW-LEVEL FILTERING IN HDP2.5*. Web Page. Online; accessed 9-Mar-2017. Sept. 2016. URL: <https://hortonworks.com/blog/eyes-dynamic-column-masking-row-level-filtering-hdp2-5/> (cited on page 302).
 - [719] SAP. *What is SAP HANA*. Web Page. Accessed: 2017-1-17. URL: <http://www.sap.com/product/technology-platform/hana.html> (cited on page 224).

-
- [720] xcat. *Extreme cloud/cluster administration toolkit*. Webpage. 2015. URL: <http://xcat-docs.readthedocs.io/en/stable/> (cited on page 279).
 - [721] *Xen - Wikipedia*. Web Page. Accessed: 2017-02-04. URL: <https://en.wikipedia.org/wiki/Xen> (cited on page 284).
 - [722] *Xen Project Overview*. Web Page. Accessed: 2017-02-04. URL: https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview (cited on page 284).
 - [723] *Xen Feature List*. Web Page. Accessed: 2017-02-04. URL: https://wiki.xenproject.org/wiki/Xen_Project_4.7_Feature_List (cited on page 285).
 - [724] Reynold S. Xin et al. “GraphX: A Resilient Distributed Graph System on Spark”. In: *First International Workshop on Graph Data Management Experiences and Systems*. GRADES ’13. New York, New York: ACM, 2013, 2:1–2:6. ISBN: 978-1-4503-2188-4. DOI: 10.1145/2484425.2484427. URL: <http://doi.acm.org/10.1145/2484425.2484427> (cited on page 211).
 - [725] T. Xu, K. Sato, and S. Matsuoka. “CloudBB: Scalable I/O Accelerator for Shared Cloud Storage”. In: *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*. Institute of Electrical and Electronics Engineers (IEEE), Dec. 2016, pages 509–518. DOI: 10.1109/ICPADS.2016.0074 (cited on page 272).
 - [726] Shelhamer Yangqing Jia Evan. *Caffe | Deep Learning Framework*. Web Page. Accessed: 02-06-2017. URL: <http://caffe.berkeleyvision.org/> (cited on page 208).
 - [727] Edward J. Yoon. *MRQL - a SQL on Hadoop Miracle*. Web Page. accessed 2017-01-29. Jan. 2017. URL: <http://www.hadoopsphere.com/2013/04/mrql-sql-on-hadoop-miracle.html> (cited on page 224).
 - [728] Apache Zeppelin. *Apache Zeppelin 0.7.0 Documentation*. Web Page. Accessed: 2017-2-27. Feb. 2017. URL: <https://zeppelin.apache.org/docs/0.7.0/> (cited on page 296).
 - [729] Bingjing Zhang et al. “Applying twister to scientific applications”. In: *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE. 2010, pages 25–32 (cited on page 233).
 - [730] Xianbo Zhang et al. “Hptfs: A high performance tape file system”. In: *Proceedings of 14th NASA Goddard/23rd IEEE conference on Mass Storage System and Technologies*. Ieee Computer Society (September 1995), 2006. DOI: 10.1.1.184.1133. URL: https://www.dtc.umn.edu/publications/reports/2006_11.pdf (cited on page 272).
 - [731] *Zookeeper - Overview*. Web Page. Accessed: 2017-01-23. URL: <https://zookeeper.apache.org/doc/trunk/zookeeperOver.html> (cited on page 294).
 - [732] *Zookeeper - Wikipedia*. Web Page. Accessed: 2017-01-23. URL: https://en.wikipedia.org/wiki/Apache_ZooKeeper (cited on page 294).
 - [733] *IBM - What is Zookeeper*. Web Page. Accessed: 2017-01-23. URL: <http://www-01.ibm.com/software/data/infosphere/hadoop/zookeeper/> (cited on page 294).



Index

L

Latex

cycle	36
Elements.....	26
description, 28	
enumerate, 27	
highlight text, 26	
images, 28	
itemiz, 27	
labels, 29	
mathematics, 29	
sections, 27	
tables, 28	
installation	24
OSX, 25	
ubuntu, 24	
Windows, 25	
markdown	35
other documentation.....	31
overleaf.....	26
proceedings	
acm, 30	
ieee, 30	
Sharelatex.....	25
slides	31

P

Python

2 and 3	149
data types	152
interactive	148
numbers	153
REPL	148
statements	152
strings	152
variables	152

S

Slides

About

10 (Test slides), 9

Health

Health (131), 84

Proteomics and Information Visualization (131), 87

Introduction

10 (Data Science Process), 74

14 (Computing Model I), 74

16 (Industrial Trends II), 73

16 (Industrial Trends), 73

19 (Data Science Education), 76

20 (Cloud Applications in Research), 76

20 (Data Deluge), 73

21 (Industrial Trends III), 73

27 (Computing Model II), 74

28 (Digital Disruption and transformation), 74

- 30 (Motivation), 72
- 4 (Conclusions), 76
- 4 (Research Model), 74
- 6 (Physics-inforamtics), 75
- 6 (Recommender Systems II), 75
- 6 (Web Search and Information Retrieval), 75
- 8 (Jobs), 73
- 9 (Computing and MapReduce), 76
- 9 (Recommender Systems I), 75
- Lifestyle**
 - Filtering (18), 93
 - Recommender (45), 89
 - Recommender (49), 91
- Overview**
 - 22 (TBD), 80
 - 30 (TBD), 82
 - 35 (TBD), 82
 - 45 (TBD), 78
- Physics**
 - Higgs (20), 94
 - Higgs (39), 96
 - Higgs II (29), 95
 - Higgs III (44), 97
- Radar**
 - Radar (58), 99
- Sensor**
 - Sensor I (31), 101
 - Sensor II (44), 101
- Sport**
 - Overview (40), 104
 - Sporta II (41), 105
 - Sports III (44), 106
- Usecases**
 - 51 (100), 111
 - Features (43), 112
 - Overview (45), 108
- Web**
 - Text Mining (33), 119
 - Web Search and Text Mining (56), 117
- Video**
 - About
 - Test Video (25:36), 9
 - Health
 - CC) Genomics, Proteomics and Information Visualization (6:56), 86
 - Big Data and Health (10:02), 84
- Big Data and Healthcare Indusry (10:02), 85**
- Clouds and Health (4:35), 85**
- EU Report on Redesigning health in Europe for 2020 (5:00), 86**
- Extrapolating to 2032 (15:13), 86**
- Genomics, Proteomics and Information Visualization (6:56), 86**
- Genomics, Proteomics and Information Visualization I (10:33), 87**
- Genomics, Proteomics and Information Visualization: II (7:41), 87**
- McKinsey Report (14:53), 85**
- Medical Big Data in the Clouds (15:02), 85**
- Medicine and the Internet of Things (8:17), 86**
- Microsoft Report on Big Data in Health (2:26), 86**
- Midical Image Big Data (6:33), 85**
- Status of Healthcare Today (16:09), 85**
- Telemedicine (8:21), 85**
- Introduction**
 - Cloud Applications in Research (33:51), 75
 - Computing and MapReduce (14:02), 76
 - Computing Model I (24:03), 74
 - Computing Model II (28:18), 74
 - Conclusions (4:59), 76
 - Data Deluge (30:38), 73
 - Data Science Education (28:08), 76
 - Data Science Process (15:42), 74
 - Digital Disruption and transformation (32:54), 74
 - Industrial Trends III (30:13), 73
 - Industrial Trends (19:25), 73
 - Industrial Trends II (16:54), 73
 - Jobs (9:39), 73
 - Motivation (40:14), 72
 - Physics-informatics (13:27), 75
 - Recommender Systems I (12:21), 75
 - Recommender Systems II (9:44), 75
 - Research Model (7:33), 74
 - Web Search and Information Retrieval (12:05), 75
- Lifestyle**
 - Case Study of Recommender systems (3:21), 92
 - Consumer Data Science (13:04), 91

- Examples of Recommender Systems (1:00), 90
Examples of Recommender Systems (8:34), 92
Item Based Filtering (11:18), 93
k Nearest Neighbors and High Dimensional Spaces (10:03), 93
k Nearest Neighbors and High Dimensional Spaces (7:16), 93
Kaggle Competitions: (3:36), 90
Netflix on Recommender Systems (14:20), 90
Recap and Examples of Recommender Systems (5:48), 91
Recap of Recommender Systems II (8:46), 92
Recap of Recommender Systems III (10:48), 92
Recommender Systems I (8:06), 90
Recommender Systems Introduction (12:56), 90
User-based nearest-neighbor collaborative filtering I (7:20), 92
User-based nearest-neighbor collaborative filtering II (7:29), 92
Vector Space Formulation of Recommender Systems new (9:06), 92
Overview
TBD (11:27), 80
TBD (11:28), 82
TBD (11:49), 80
TBD (13:04), 79
TBD (16:04), 82
TBD (2:10), 81
TBD (2:58), 79
TBD (3:42), 79
TBD (4:27), 79
TBD (5:07), 80
TBD (5:45), 81
TBD (6:00), 79
TBD (6:24), 82
TBD (6:51), 82
TBD (7:28), 82
TBD (7:34), 79
TBD (8:02), 82
TBD (9:37), 79
TBD (9:49), 78
Physics
Accelerator Picture Gallery of Big Sci-
ence (11:21), 95
Accept-Reject (5:54), 98
Binomial Distribution: (12:38), 98
Central Limit Theorem (4:47), 98
Change shape of background & num of Higgs Particles (7:01), 96
Discovery of Higgs Particle (13:49), 94
Event Counting (7:02), 95
Gaussian Distributions (9:08), 97
Generators and Seeds II (7:10), 98
Higgs Particle Counting Errors (6:28), 97
Higgs Particle Events and Counting (9:30), 95
Interpretation of Probability (12:39), 98
Looking for Higgs Particle and Counting Introduction II (7:38), 94
Looking for Higgs Particle Experiments (9:29), 94
Monte Carlo Method (2:23), 98
Physics and Random Variables I (8:34), 96
Physics and Random Variables II (5:50), 96
Poisson Distribution (4:37), 98
Random variables and normal distributions (8:19), 96
Statistics of Events with Normal Distributions (11:25), 97
Using Statistics (14:02), 97
With Python examples of Signal plus Background (7:33), 96
Radar
Global Climate Change (2:51), 100
Ice Sheet Science (1:00), 99
Radar Informatics (3:31), 99
Radio Informatics (3:35), 100
Radio Overview (4:16), 100
Remote Sensing (6:43), 99
Sensor
Earth/Environment/Polar Science data gathered by Sensors (4:58), 102
Industrial Internet of Things (1:24:02), 101
Internet of Things (12:36), 101
Robotics and IoT Expectations (8:05), 101
Sensor Clouds (4:40), 101
Smart Grid (6:04), 102

- U-Korea (U=Ubiquitous) (2:49), 102
 Ubiquitous/Smart Cities (1:44), 102
Sport
 Basic Sabermetrics (26:53), 104
 Introduction and Sabermetrics (Baseball Informatics) Lesson (31:4), 104
 Other Video Data Gathering in Baseball (18:5), 106
 Pitcher Quality (10:02), 105
 PITCHf/X (10:39), 106
 Pitching Clustering (20:59), 105
 Soccer and the Olympics (8:28), 106
 Spatial Visualization in NFL and NBA (15:19), 106
 Wearables (22:2), 106
 Wins Above Replacement (30:43), 104
Use cases
 Architecture (10:05), 109
 Astronomy and Physics Use Cases (17:33), 112
 Commercial Use Cases (17:43), 111
 Database (SQL) Use Case Classification (11:13), 113
 Deep Learning and Social Networks Use Cases (14:19), 112
 Defense Use Cases (15:43), 111
 Energy Use Case (4:01), 112
 Environment, Earth and Polar Science Use Cases (25:29), 112
 Government Use Cases (17:43), 111
 Healthcare and Life Science Use Cases (30:11), 111
 Introduction (13:02), 108
 NoSQL Use Case Classification (11:20), 113
 Requirements (27:28), 110
 Research Ecosystem Use Cases (9:09), 112
 Security (9:51), 109
 Summary of Use Case Classification (23:39), 113
 Taxonomies (7:42), 109
 Technology (4:14), 109
 Use Case Classifications I (12:42), 113
 Use Case Classifications II (20:18), 113
 Use Case Classifications III (17:25), 114
Web
 Boolean and Vector Space Model (6:17), 118
 Clustering and Topic Models (6:21), 120
 Fundametal Principals of Web Search (5:06), 118
 Indices (5:44), 119
 Information Retrieval (6:06), 118
 Principles (9:30), 118
 Realated Applications (17:24), 120
 Search Engines (3:08), 118
 Text Mining (9:56), 117
 TF-IDF and Probabilistic Models (3:57), 119
 Web Advertising and Search (9:02), 120
 Web crawling and Document Preparation (4:55), 119
 Web Search and Text Mining II (6:11), 120
 Web Search History (5:48), 118

W

- | | |
|-----------------|----|
| Writing | 15 |
| Checklist | 17 |