Big Data Software and Applications

Lecture notes I524

Editor:
Gregor von Laszewski
laszewski@gmail.com

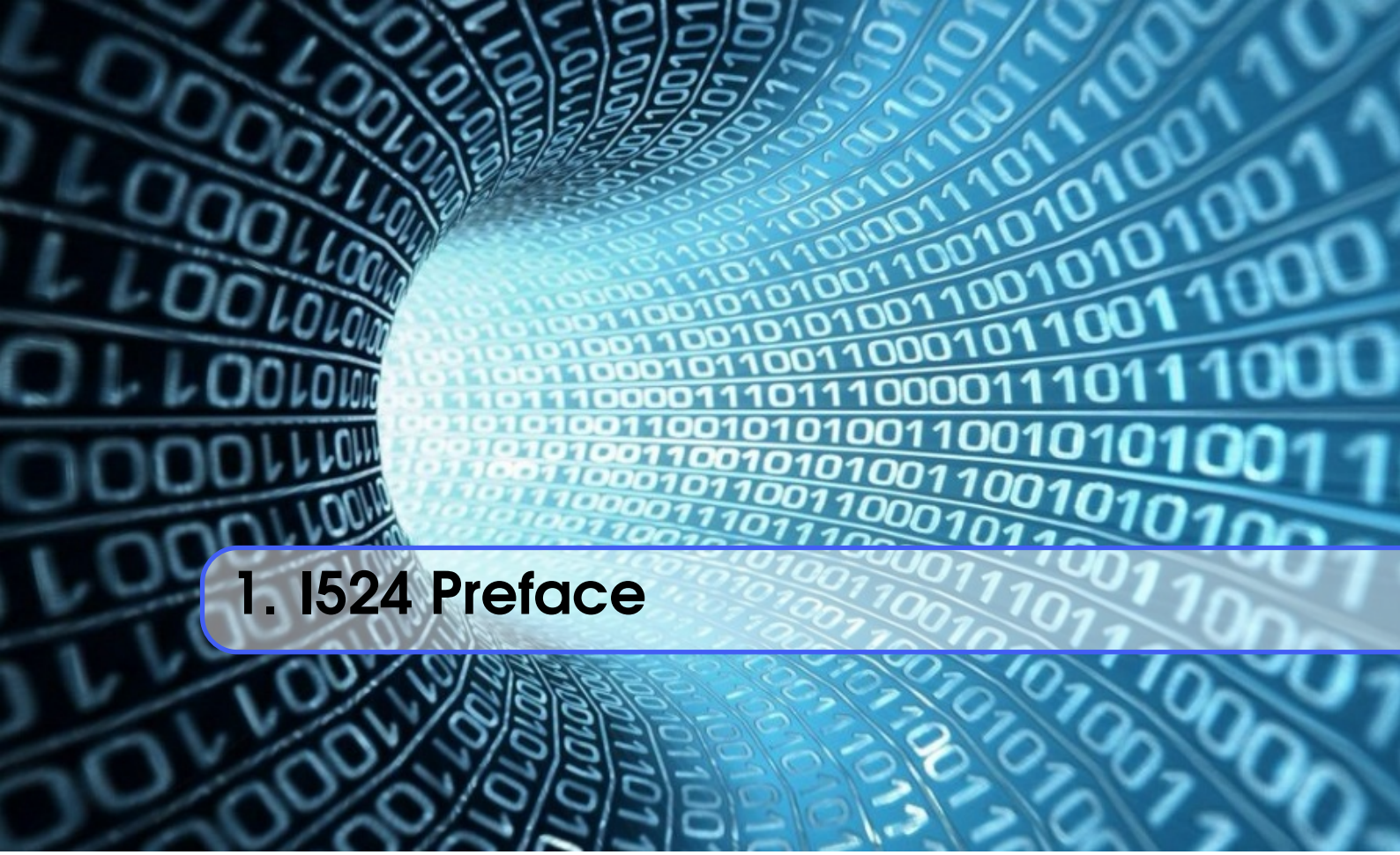
PUBLISHED BY INDIANA UNIVERSITY

HTTPS://GITHUB.COM/CLOUDMESH/CLASSES

First printing, January 2017

Contents

Contents



1. I524 Preface

1.1 About

This Web page is hosted at

- `https://cloudmesh.github.io/classes/`

The Piazza group is available at:

- `https://piazza.com/class/ix39m27czn5uw`

The PDF version of a **subset** if the information posted on the Web page is located at:

- `https://cloudmesh.github.io/classes/i524-notes.pdf`

This PDF document will be updated on a weekly basis and we will integrate what we have taught you in class and serve as lecture notes. However do not forget a lot of information is on teh Web Page and in piazza. So these PDF notes will not be sufficient for you for this class.

1.2 Disclaimer

It is normal that questions posed by the students in the online and residential classes lead to improvements and clarifications of the content published on the class Web pages. Hence you need to revisit the page updates. Just as in other years we provide a changelog. Changes will not lead to any requirements change to the class but only improving the content. However, we have learned from previous classes that additional homework may need to be added in case the class participants need to grasp a concept better or simplify their work.

We have two ways of providing content to the class:

- we release content only on weekly basis
- we release content even in draft form

We decided to do the later so you may have the ability to see what is comming up. Please do not mistake this as a changing requirement. This is an opportunity for you. You are not required to look at lectures or assignments that are not yet released for this class.

1.3 Conventions

$* When showing examples of commands, the *$ symbol precedes the actual command. So, the other lines are the output obtained after executing the command. An example invoking the ls command follows:

```
$ ls
```

PORTALNAME In some examples we refer to your portal name as the *PORTALNAME* you have on FutureSystems.

USERNAME In some examples we refer to your local computers name as *USERNAME*. Your portal name and your local name may be different.

Menu selections: *Start –> Programs*

Man page: *ls(1)*

1.4 Instructors

The course presents lectures in online form given by the instructors listed bellow. Many others have helped making this material available and may not be listed here. For this class support is provided by

- Gregor von Laszewski (PhD)
- Badi' Abdul-Wahid (PhD)
- Jerome Mitchell (Teaching Assistant)
- Miao Zhang (Teaching Assistant)
- Tony Liu (Teaching Assistant)
- Vibhatha Abeykoon (Teaching Assistant)
- Dimitar Nikolov (Teaching Assistant)

1.4.1 Dr. Gregor von Laszewski

Gregor von Laszewski is an Assistant Director of Cloud Computing in the DSC. His is also an Adj. Assoc. Professor of the Intelligent Systems Engineering Department at

Indiana University. He held a position at Argonne National Laboratory from Nov. 1996 – Aug. 2009 where he was last a scientist and a fellow of the Computation Institute at University of Chicago. During the last two years of that appointment he was on sabbatical and held a position as Associate Professor and the Director of a Lab at Rochester Institute of Technology focussing on Cyberinfrastructure. He received a Masters Degree in 1990 from the University of Bonn, Germany, and a Ph.D. in 1996 from Syracuse University in computer science. He was involved in Grid computing since the term was coined. He was the lead of the Java Commodity Grid Kit (`http://www.cogkit.org`) which provides till today a basis for many Grid related projects including the Globus toolkit. Current research interests are in the areas of Cloud computing. He is leading the effort to develop a simple IaaS client available at as OpenSource project at `http://cloudmesh.github.io/client/`

His Web page is located at `http://gregor.cyberaide.org`. To contact him please send mail to laszewski@gmail.com. For class related e-mail please use Piazza for this class.

In his free time he teaches Lego Robotics to high school students. In 2015 the team won the 2nd prize in programming design in Indiana. If you like to volunteer helping in this effort please contact him.

He offers also the opportunity to work with him on interesting independent studies. Current topics include but are not limited to

- cloudmesh
- big data for NIST
- big data benchmarking.
- scientific impact of supercomputer and data centers.
- STEM and other educational activities while using robotics or big data

Please contact me if you are interested in this.

1.4.2 Dr. Geoffrey C. Fox

Greoffrey C. Fox received a Ph.D. in Theoretical Physics from Cambridge University and is now distinguished professor of Informatics and Computing, and Physics at Indiana University where he is director of the Digital Science Center, Chair of Department of Intelligent Systems Engineering and Director of the Data Science program at the School of Informatics and Computing. He previously held positions at Caltech, Syracuse University and Florida State University after being a postdoc at the Institute of Advanced Study at Princeton, Lawrence Berkeley Laboratory and Peterhouse College Cambridge. He has supervised the PhD of 68 students and published around 1200 papers in physics and

computer science with an index of 70 and over 26000 citations. He currently works in applying computer science from infrastructure to analytics in Biology, Pathology, Sensor Clouds, Earthquake and Ice-sheet Science, Image processing, Deep Learning, Manufacturing, Network Science and Particle Physics. The infrastructure work is built around Software Defined Systems on Clouds and Clusters. The analytics focuses on scalable parallelism.

He is involved in several projects to enhance the capabilities of Minority Serving Institutions. He has experience in online education and its use in MOOCs for areas like Data and Computational Science. He is a Fellow of APS (Physics) and ACM (Computing).

1.4.3 Dr. Badi' Abdul-Wahid

Badi' received a Ph.D. in Computer Science at the University of Notre Dame under Professor Jesus Izaguirre. The primary focus of his graduate work was the development of scalable, fault-tolerant, elastic distributed applications for running Molecular Dynamics simulations.

At Indiana University, Badi' works with the FutureSystems project on a NIST-funded study whose goal is to understand patterns in the development and usage of Big Data Analysis pipelines.

1.4.4 Teaching Assistants

Jerome Mitchell

Jerome Mitchell is a Ph.D candidate in computer science at Indiana University and is interested in coupling the fields of computer and polar science. He has participated in the United State Antarctic Program, (USAP), where he collaborated with a multidisciplinary team of engineers and scientists to design a mobile robot for harsh polar environments to autonomously collect ice sheet data, decrease the human footprint of polar expeditions, and enhance measurement precision. His current work include: using machine learning techniques to help polar scientists identify bedrock and internal layers in radar imagery.

He has also been involved in facilitating workshops to educate faculty and students on the importance of parallel and distributed computing at minority-serving institutions.

Dimitar Nikolov

Dimitar Nikolov is a PhD student in the Computer Science program at Indiana University since August 2008. His research interests include online social networks, tagging systems and web mining. As part of his PhD he conducts research with the NaN group, led by Professor Fil Menczer. His thesis topic is TBD.

Vibhatha Abeykoon

Vibhatha Abeykoon is a PhD student in the Intelligent Systems Engineering program at Indiana University since August 2016. His research interests include machine learning, singal processing, source seperation, deep learning, big data and distributed systems. As part of his PhD he conducts research with Professor Geoffrey C. Fox and Assistant Professor Minje Kim. His thesis topic is TBD.

Miao Zhang

Miao Zhang PhD student working primarily on computer network related stuff. Currently focuses on network performance monitoring and forecasting.

Tony Liu

Tony Liu is a PhD student in the Intelligent Systems Engineering program at Indiana University since August 2016. He received his Master degree in Computer Science program in May 2016 at IU. His research interests are high performance computing, computer systems and computer architecture. He used to work under Professor Thomas Sterling and Professor Andrew Lumsdaine from Center for Research in Extreme Scale Technology (CREST) on HPX-5 runtime system project. His currently works on benchmarking Intel Xeon Phi Knights Landing processor with Caffe under Professor Judy Qiu and Professor Lei Jiang.

1.5 Contributing

Contributing content to this web page is easy. First you have to fork the repository while going to the link:

- `https://github.com/cloudmesh/classes`

And press the fork icon. Now you can clone or directly manipulate your fork from the web browser. If you clone, you need to make sure you clone from your fork.

Next, you cen cd to the *classes* directory and make the modifications. Check them locally with:

```
make html
make view
```

To change things you can simply edit files in the docs/source directory. Commit the changes and push them into your local fork. Now you can create on the web page for the classes a pull request.

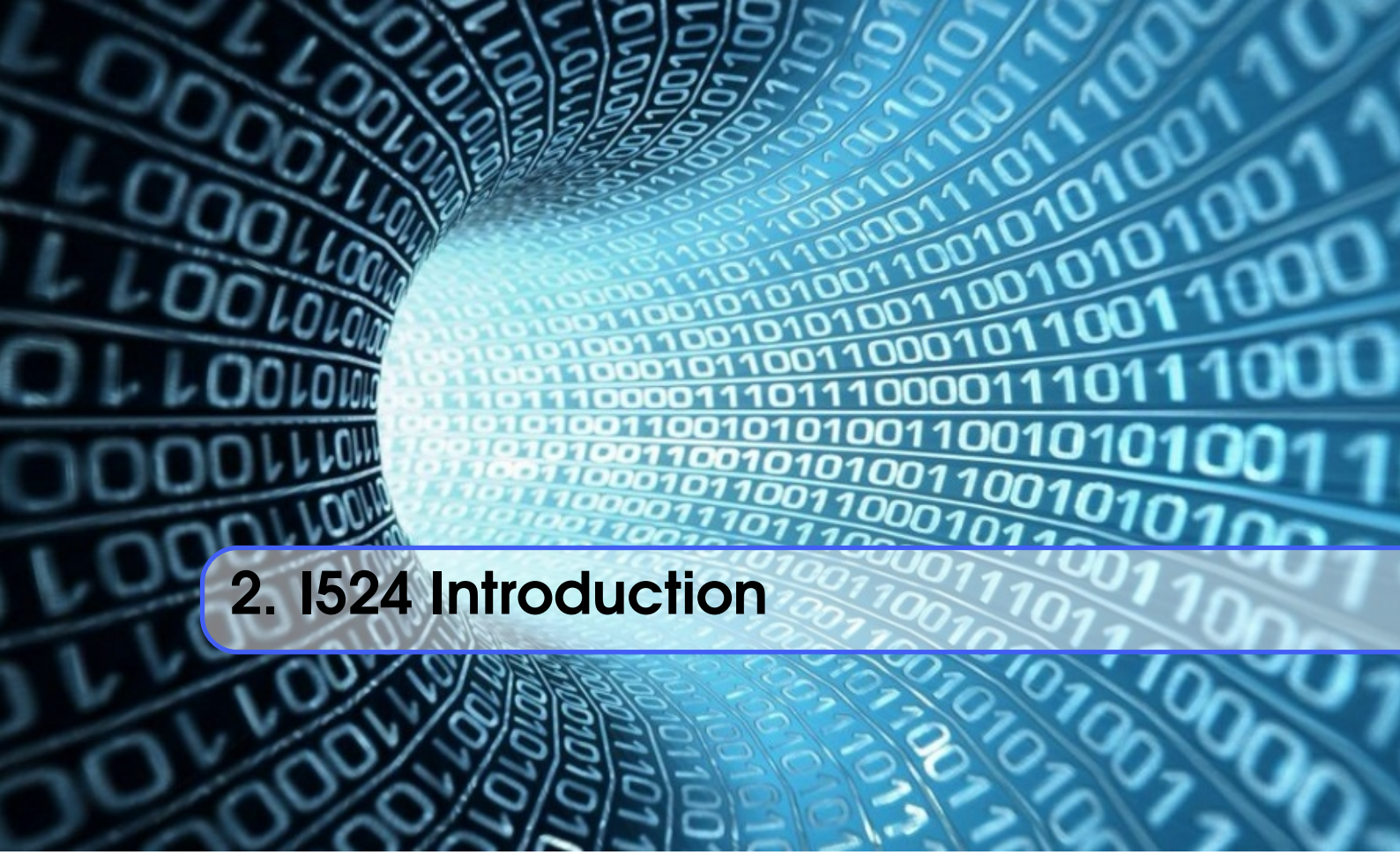
If accepted the request will be merged and you will be credited via your github account. Make sure you use git config to set your name and e-mail.

Please also make sure you do not confuse github and gitlab. While this web page is located in github, your activities for the class are done in gitlab.

1.5.1 Exercise

Contrib.1: Identify a spelling error in the web page, or another item to improve. Fork the Web page, fix the error and create a pull request.

Contrib.2: Identify a section that is not covered by this material, but could be useful. Add such a section and create a pull request so your contribution can be added. Work with others that review your section before submitting so we make sure no one else is working on this already. If they do we bring you in contact with them.



2. I524 Introduction

2.1 i524 Overview

This course studies software used in many commercial activities related to Big Data. The backdrop for course contains more than 370 software subsystems illustrated in Figure 1. We will describe the software architecture represented by this collection and work towards identifying best practices to deploy, access and interface with them. Topics of this class will include:

1. The cloud computing architecture underlying open source big data software and frameworks and contrast of them to high performance computing
2. The software architecture with its different layers covering broad functionality and rationale for each layer.
3. We will go through selected big data stack software from the list of more than 370
4. We will be identifying how we can create and replicate software environments based on software deployed and used on clouds while using Containers, OpenStack and Ansible playbooks.
5. Students will chose a number of open source members of the list each and create repeatable deployments as illustrated in class.
6. The main activity of the course will be building a significant project using multiple subsystems combined with user code and data. Projects will be suggested or students can chose their own. A project report will summarize the work conducted.
7. Topics taught in this class will be very relevant for industry as you are not only exposed to big data, but you will also be practically exposed to DevOps and collaborative code development tools as part of your homework and project assignment.

Students of this class will need to conduct their project deployments in python using

Kaleidoscope of (Apache) Big Data Stack (ABDS) and HPC Technologies	
Cross-Cutting Functions 1) **Message and Data Protocols:** Avro, Thrift, Protobuf	17) **Workflow-Orchestration:** ODE, ActiveBPEL, Airavata, Pegasus, Kepler, Swift, Taverna, Triana, Trident, BioKepler, Galaxy, IPython, Dryad, Naiad, Oozie, Tez, Google FlumeJava, Crunch, Cascading, Scalding, e-Science Central, Azure Data Factory, Google Cloud Dataflow, NiFi (NSA), Jitterbit, Talend, Pentaho, Apatar, Docker Compose, KeystoneML 16) **Application and Analytics:** Mahout , MLlib , MLbase, DataFu, R, pbdR, Bioconductor, ImageJ, OpenCV, Scalapack, PetSc, PLASMA MAGMA, Azure Machine Learning, Google Prediction API & Translation API, mlpy, scikit-learn, PyBrain, CompLearn, DAAL(Intel), Caffe, Torch, Theano, DL4j, H2O, IBM Watson, Oracle PGX, GraphLab, GraphX, IBM System G, GraphBuilder(Intel), TinkerPop, Parasol, Dream:Lab, Google Fusion Tables, CINET, NWB, Elasticsearch, Kibana, Logstash, Graylog, Splunk, Tableau, D3.js, three.js, Potree, DC.js, TensorFlow, CNTK
2) **Distributed Coordination:** Google Chubby, Zookeeper, Giraffe, JGroups	15B) **Application Hosting Frameworks:** Google App Engine, AppScale, Red Hat OpenShift, Heroku, Aerobatic, AWS Elastic Beanstalk, Azure, Cloud Foundry, Pivotal, IBM BlueMix, Ninefold, Jelastic, Stackato, appfog, CloudBees, Engine Yard, CloudControl, dotCloud, Dokku, OSGi, HUBzero, OODT, Agave, Atmosphere 15A) **High level Programming:** Kite, Hive, HCatalog, Tajo, Shark, Phoenix, Impala, MRQL, SAP HANA, HadoopDB, PolyBase, Pivotal HD/Hawq, Presto, Google Dremel, Google BigQuery, Amazon Redshift, Drill, Kyoto Cabinet, Pig, Sawzall, Google Cloud DataFlow, Summingbird, Lumberyard
3) **Security & Privacy:** InCommon, Eduroam OpenStack Keystone, LDAP, Sentry, Sqrrl, OpenID, SAML OAuth	14B) **Streams:** Storm, S4, Samza, Granules, Neptune, Google MillWheel, Amazon Kinesis, LinkedIn, Twitter Heron, Databus, Facebook Puma/Ptail/Scribe/ODS, Azure Stream Analytics, Floe, Spark Streaming, Flink Streaming, DataTurbine 14A) **Basic Programming model and runtime, SPMD, MapReduce:** Hadoop, Spark, Twister, MR-MPI, Stratosphere (Apache Flink), Reef, Disco, Hama, Giraph, Pregel, Pegasus, Ligra, GraphChi, Galois, Medusa-GPU, MapGraph, Totem 13) **Inter process communication Collectives, point-to-point, publish-subscribe:** MPI, HPX-5, Argo BEAST HPX-5 BEAST PULSAR, Harp, Netty, ZeroMQ, ActiveMQ, RabbitMQ, NaradaBrokering, QPid, Kafka, Kestrel, JMS, AMQP, Stomp, MQTT, Marionette Collective, **Public Cloud:** Amazon SNS, Lambda, Google Pub Sub, Azure Queues, Event Hubs 12) **In-memory databases/caches:** Gora (general object from NoSQL), Memcached, Redis, LMDB (key value), Hazelcast, Ehcache, Infinispan, VoltDB, H-Store
4) **Monitoring:** Ambari, Ganglia, Nagios, Inca	12) **Object-relational mapping:** Hibernate, OpenJPA, EclipseLink, DataNucleus, ODBC/JDBC
21 layers **Over 350 Software Packages**	12) **Extraction Tools:** UIMA, Tika 11C) **SQL(NewSQL):** Oracle, DB2, SQL Server, SQLite, MySQL, PostgreSQL, CUBRID, Galera Cluster, SciDB, Rasdaman, Apache Derby, Pivotal Greenplum, Google Cloud SQL, Azure SQL, Amazon RDS, Google F1, IBM dashDB, N1QL, BlinkDB, Spark SQL
	11B) **NoSQL:** Lucene, Solr, Solandra, Voldemort, Riak, ZHT, Berkeley DB, Kyoto/Tokyo Cabinet, Tycoon, Tyrant, MongoDB, Espresso, CouchDB, Couchbase, IBM Cloudant, Pivotal Gemfire, HBase, Google Bigtable, LevelDB, Megastore and Spanner, Accumulo, Cassandra, RYA, Sqrrl, Neo4J, graphdb, Yarcdata, AllegroGraph, Blazegraph, Facebook Tao, Titan:db, Jena, Sesame **Public Cloud:** Azure Table, Amazon Dynamo, Google DataStore
January 29 2016	11A) **File management:** iRODS, NetCDF, CDF, HDF, OPeNDAP, FITS, RCFile, ORC, Parquet
	10) **Data Transport:** BitTorrent, HTTP, FTP, SSH, Globus Online (GridFTP), Flume, Sqoop, Pivotal GPLOAD/GPFDIST
	9) **Cluster Resource Management**: Mesos, Yarn, Helix, Llama, Google Omega, Facebook Corona, Celery, HTCondor, SGE, OpenPBS, Moab, Slurm, Torque, Globus Tools, Pilot Jobs
	8) **File systems:** HDFS, Swift, Haystack, f4, Cinder, Ceph, FUSE, Gluster, Lustre, GPFS, GFFS **Public Cloud:** Amazon S3, Azure Blob, Google Cloud Storage
	7) **Interoperability:** Libvirt, Libcloud, JClouds, TOSCA, OCCI, CDMI, Whirr, Saga, Genesis
	6) **DevOps:** Docker (Machine, Swarm), Puppet, Chef, Ansible, SaltStack, Boto, Cobbler, Xcat, Razor, CloudMesh, Juju, Foreman, OpenStack Heat, Sahara, Rocks, Cisco Intelligent Automation for Cloud, Ubuntu MaaS, Facebook Tupperware, AWS OpsWorks, OpenStack Ironic, Google Kubernetes, Buildstep, Gitreceive, OpenTOSCA, Winery, CloudML, Blueprints, Terraform, DevOpSlang, Any2Api
Green is work of NSF14-43054	5) **IaaS Management from HPC to hypervisors:** Xen, KVM, QEMU, Hyper-V, VirtualBox, OpenVZ, LXC, Linux-Vserver, OpenStack, OpenNebula, Eucalyptus, Nimbus, CloudStack, CoreOS, rkt, VMware ESXi, vSphere and vCloud, Amazon, Azure, Google and other public Clouds **Networking:** Google Cloud DNS, Amazon Route 53

Figure 2.1: Big data relevant technology layers

ansible and enabling a software stack that is useful for a big data analysis. While it is not necessary to know either python or ansible to take the class it is important that you have knowledge of a programming language so you can enhance your knowledge on them throughout the class and succeed. You will be expected to have a computer on which you have python 2.7.x installed. You will be using chameleon and possibly our local cloud. Optionally some projects may use docker.

Figure 2 illustrates that you can follow the components of the class in a variety of ways and in parallel. For example, you do not have to wait to start the project or to find out more about any of the subsystems.

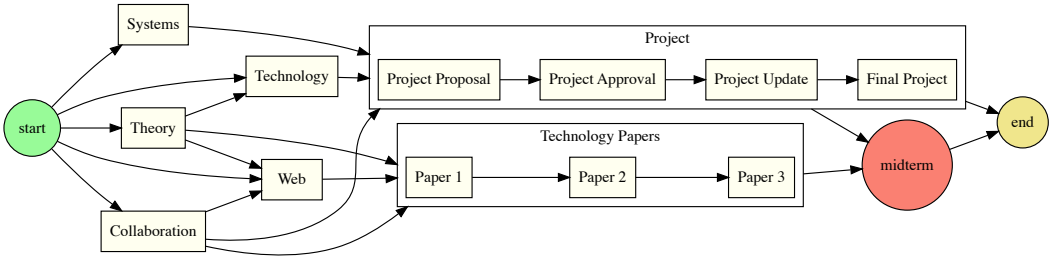


Figure 2: Components of the Class

Note: You do not have to take I523 in order to take I524.

For previous I523 class participants: While I523 is a beginners class I524 is a more advanced class and we expect that you know python which you hopefully have learned as part of I523 while doing a software project. If not, make sure you learn it before you take this class or consider **significant** additional time needed to learn it for the class.

Residential students need to enroll early so we avoid the situation like last year where we had many signing up, but did not even show up to the first lecture. I have asked that students from I523 have preference, but I am not sure if we can enforce this. So enroll ASAP. Those that are on the waiting list are recommended to show up in the first class. It is likely that you can join as others drop.

2.1.1 Meeting Times

The classes are published online. Residential students at Indiana University will participate in a discussion taking place at the following time:

- Monday 09:30am - 10:45am EST, I2 130

For the 100% online students see the office hours.

2.1.2 Online Meetings

For the zoom information please go to

`https://iu.instructure.com/courses/1603897/assignments/syllabus`

A doodle was used and all students that answered the doodle have times that they specified. We covered 100% the time for the students through the following schedule:

All times are in Eastern Standard Time.

Day of Week	Meetings
Monday	8-9am Office Hours 9:30-10:45am Residential Lecture 6-7pm Office Hours
Tuesday	1-2pm Office Hours 4-5pm Office Hours
Wednesday	6-7pm Office Hours
Thursday	6-7pm Office Hours (Gregor)
Friday	4-5pm Office Hours
Saturday	8-9pm Office Hours
Sunday	9-10am Office Hours 8-9pm Office Hours

2.1.3 Who can take the class?

- Although Undergrads can take this class it will be thaught as graduate class. Make sure you have enough time and fulfill the prerequisites such as knowing a programming language well. You need to have enough time to learn python if you do not know it.
- You can take I524 without taking I523, but you must be proficient in python. Overlap between I523 and I524 only relates to some introduction lectures and naturally lectures from the systems track such as github, report writing, introduction to python.
- Online students
- Residential students

2.1.4 Homework

Grading policies are listed in Table 1.

Table 2.1: Table 1: Grading

Percent	Description
10%	Class participation and contribution to Web pages.
30%	Three unique technology papers per student of the 370 systems. Each paper as at least 2 pages per technology without references.
60%	Project code and report with at least 6 pages without references. Much shorter reports will be returned without review. Do not artificially inflate contents.

- **Technology papers:** Technology papers must be non-overlapping in the entire class. As we have over 370 such technologies we should have enough for the entire class. If you see technologies missing, let us know and we see how to add them. Technology papers could be a survey of multiple technologies or an indepth analysis of a particular technology.
- **Technology paper groups:** Groups of up to three students can work also on the technology papers. However the workload is not reduced, you will produce 3 times the number of group members technology papers of unique technologies. However, you can have multiple coauthors for each paper (up to thre) that are part of your group. Please do not ask us how many technology papers you need to write if you are in a group. The rule is clearly specified. Example: Your group has 3 members, each of them has to procude 3 unique papers, thus you have to produce 9 unique technology papers for this group. If you have 2 members you have to produce 6, if you work alone you have to produce 3.
- **Technology deployment Homework:** Each student will develop as a preparation for the project a deployment of a technology. Points may depend on completeness, effort of the deployment. Technology deployments should as much as possible be non overlapping. In many cases you chose wisely such deployments may line up with your technology papers as you can add a section reporting on your achievement and experience with such deployments.
- **Project groups:** Groups of up to three students can work on a project but workload increases with each student and a work break down must be provided. More than three students are not allowed. If you work in a group you will be asked to deploy a larger system or demonstrate deployability on multiple clouds or container frameworks while benchmarking and comparing them. A group project containing 2 or 3 team members should not look like a project done by an individual. Please plan careful and make sure all team members contribute.
- **Frequent checkins**: It is **important** to make frequent and often commits to the github repository as the activities will be monitored and will be integrated into the project grade. Note that paper and project will take a considerable amount of time and doing proper time management is a must for this class. Avoid starting your project late. Procrastination does not pay off.
- **No bonus projects:** This class will not have any bonus projects or regrading requests. Instead you need to focus your time on the papers and the project assignments and homework.
- **Voluntary work:** You are welcome to conduct assignments and excerises you find

on the class Web page on your own. However they are not graded or considered for extra credit.

- **Late homework**: Any late homework will be receiving a 10% grade reduction. As this is a large class and the assignments are not standard multiple choice questions, grading will take a considerable time. Some homework can not be delivered late as they are related to establish communication with you. Such **deadline specific** homework will receive 0 points in case they are late. See course calendar. It is the student's responsibility to upload submissions well ahead of the deadline to avoid last minute problems with network connectivity, browser crashes, cloud issues, etc.
- **Chance for publishing a paper:** If however you find that the work you do could lead to a publishable paper, you could work together with the course instructor as coauthors to conduct such an activity. However, this is going to be a significant effort and you need to decide if you like to conduct this. In such cases if the work is sufficient for publication submission, an A+ for the class could be considered. It will be a lot of work. The length of such a paper is typically 10-12 high quality pages including figures and references. We may elect for the final submission to use a different LaTeX style

2.1.5 Prerequisites

We expect you are familiar with:

- Linux and the Operating system on which you will focus your deployment.
- Note that Windows as OS will not be sufficient as Ansible is not supported on it. However you can use virtualbox or log onto one of the clouds to get access to an OS that supports ansible. So you can use your Windows computer if it is powerful enough.
- Python 2.7.x (we will not use python 3 for this class as it is not yet portable with all systems) Although python is considered to be a straight forward language to learn, students that have not done any programming my find it challanging.
- Familaiarity with th Python eco system. The best way to install python on a computer is to use virtualenv, and pip (which we will teach you as part of the class).
- Familiarity with an editor such as emacs, vi, jedit, pyCharm, eclipse, or other that you can use to program in and write your reports.

If you are not familiar with these technologies, we expect that you get to know them before or during class. This may pose additional time commitment.

2.1.6 Open Source Publication of Homework

As this class is about open source technologies, we will be using such technologies to gather the homework submissions. We will not be using CANVAS so we teach you these technologies that are often mandated in industry. CANVAS is not.

As a consequence all technology papers from all students will be available as a single big technical report. To achieve this all reports must be written in the same format. This wil be LaTeX and all references have to be provided a bibtex file while you use jabref.

Alternatively lyx.org can be used, if you prefer to edit latex in *what you see is almost what you get* format. The use of sharelatex or overleave or lyx.org is allowed.

2.1.7 Piazza

All communication will be done via Piazza. We will not read e-mail send to our university or private e-mails. All instructors are following this rule. Any mail that is not send via Piazza will be **not read** and **deleted**. This is also true for any mail send to the inbox system in CANVAS. We found CANVAS a not scalable solution for our class and will not use CANVAS for reaching out to you. If you need a different mechanism to communicate with us, please ask on Piazza how to do that. Please note that private posts in piazza are shared among all instructors and TAs.

To sign up in piazza please follow this link:

- `https://piazza.com/iu/spring2017/i524`

We have created a number of piazza folder to organize the posts int topics. Thes folders are:

help: Our help folder is just like a ticket system that is monitored by the TA's. Once you file a question here, a TA will attend to it, and work with you. Once the issue you had is resolved, the TA is marking it as resolved. If you need to reopen a help message, please mark it again as unresolved or post a follow up.

project: Questions related to projects are posted here.

logistics: Question regarding the logistics of the class are posted here. This includes questions about communication, meeting times, and other administrative activities.

papers: Questions regarding the paper are posted here.

grading: Questions regarding grades are posted here.

clouds: Questions regarding cloud resources are posted here.

faq: Some questions will be transformed to FAQ's that we post here. Note also that we have an FAQ on the class Web page that you may want to visit. We try to move important FAQ's from Piazza into the Web page, so it is important that you check both.

meetings: Here we will post times for meetings with TA's and instructors that are not yet posted on the Web page as part of the regular meeting times. Class participants are allowed to attend any Zoom meeting that we annonce in this folder. For online students we will also determine a time for regulare meetings. The TAs are required to hold 10 hours of meeting times upon request with you. Please make use of this.

other: In case no folder matches for your question use other.

2.1.8 Tips on how to achieve your best

While teaching our classes we noticed the following tips to achieve your best:

- Listening to the lectures
- **Set aside enough undisturbed time for the class**. Switch off facebook, twitter, or other distracting social media systems when focussing on the class.

- **Ask for help**. The TAs can schedule custom help office hours on appointment during reasonable times.
- Do not **Procrastinate**.
- Do not **take your other classes more serious**.
- **Start the project in the first 4 weeks of the class**
- Be aware that this class is not based on a text book and what this implies.
- Do not overestimating the technical abilities.
- Do not underestimating the time it takes to do the project.
- Do not forget to include benchmarks in your project.
- Unnecessarily struggling with LaTeX as you do not use an example we provide.
- Trying to do things just on Windows which is typically more difficult than using Linux.
- Not having a computer that is up to date. Update your memory and have a SSD
- Ignoring obvious security rules and not integrating ssh form the start into your projects.
- Not posting passwords into git. For example git does **not** allow to **easily** completely delete files that contain secret information such as passwords. It takes significant effort to do that. Make sure you do add in git on individual files and never just a bulk add.
- Having your coleagues do the work for you
- Underestimating the **time** it takes to do deployments
- Not reading our piazza posts and repeating the same question over and over
- Use Piazza to communicate and not CANVAS or e-mail.
- When you receive an e-mail from piazza, reply to it while clicking on the link instead of replying via e-mail directly. This is more reliable.

2.1.9 Submissions

Your papers and projects will be developed on GitHub and submitted using Pull Requests. The process is as follows:

1. fork the sp17-i524 repository.
2. clone your fork and commit and push your changes.
3. submit a pull request to the master branch of the origin repository.

See the repository for details on the individual assignments. As it will periodically be updated, make sure you are familiar with the process of Syncing a fork.

Some things to keep in mind:

- space on github is limited, so do not add datasets to the repository. Any datasets you use should be publicly hosted and deployed as part of your project ansible deployment scripts.
- never add ssh private keys to the repository. This results in a security risk, possible point deductions, and lots of time and effort to fix.
- all work will be licensed under the Apache 2 open source license.
- all submissions and discussion will be visible to the world.

2.1.10 Selected Project Ideas

Students can select from a number of project ideas. We will improve and add additional ideas throughout the semester. Students can also conduct their own projects. We recommend that you identify a project idea by the end of the first month of the class. Example project descriptions that you may want to take a look at include:

- robotswarm
- dockerswarm
- kubernetes
- slurmcluster
- authordisambiguity_b
- NIST Big Data Working group examples: Selected and approved use case from `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-3.pdf`
- Selected examples from Fall I523: Some students may have created an example as part of I523. Not all examples created as part of this class qualify for a I524 project. Please contact Gregor von Laszewski via Piazza to discuss suitability of your previous I523 project. If such a project is selected, approved and used it is expected it is significantly enhanced.
- Cloudmesh Enhancements: A number of projects could center around the enhancements of cloudmesh for the improvement of big data projects using virtual machines and containers. This includes:
  - Development of REST services for cloudmesh while using cloudmesh client
  - Development of benchmarking examples while using cloudmesh client
  - Development of a better Azure interface to additinal services
  - Development of a better AWS interfac to additinal services
  - Development of a Web interface while using django
  - SLURM integration to create virtual clusters on comet
  - Port cloudmesh client to Windows 10
  - Integrate docker into cloudmesh and demonstrate its use
  - Integrate kubernetes into cloudmesh and demonstrate its use
  - Expand the HPC capabilities of cloudmesh

2.1.11 Software Project

For a software project you have the choice of working indifidualy or working in a team of up to three students. You can use the **search teammate** folder to find and form groups:

- `https://piazza.com/class/ix39m27czn5uw?cid=5`

The following artifacts are part of the deliverables for a project

Code: You must deliver the code in github. The code must be compilable and a TA may try to replicate to run your code. You MUST avoid lengthy install descriptions and everything must be installable from the command line. We will check submission. All team members must be responsible for one part of the project.

Project Report: A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 4 pages, one student in the project
- 6 pages, two students in the project
- 8 pages, three students in the project

Work Breakdown: The report contains in an appendix a section that is only needed for team projects. Include in the section a short but sufficiently detailed work breakdown documenting what the team has done. Back it up with commit information from github. Such as how many commits and lines of code a team member has contributed. The section does not count towards the overall length of the paper.

In addition the graders will check the history of checkins to verify each team member has used github to checkin their contributions frequently. E.g. if we find that one of the students has not checked in code or documentation in the same way at other teammates, it will be questioned. An oral exam may be scheduled to verify that the student has contributed to the project. In an oral exam the student must be familiar with **all** aspects of the project not just the part you contributed.

License: All projects are developed under an open source license such as Apache 2.0 License. You will be required to add a LICENCE.txt file and if you use other software identify how it can be reused in your project. If your project uses different licenses, please add in a README.md file which packages are used and which license these packages have while adding a licenses file.

Reproducability: The reproducability of your code will be tested twice. It is tetes by another student or team, it is also tested by a TA. A report of the testing team is provided. Your team will also be responsible for executing as many tests as you have team members on other projects. A reproducability statement should be written with details about functionality, readbility, and report quality. This statement does not have to be written in latex but uses RST.

Requirements:

- Use of cloud resources mandatory, can be substituted by kubeernetes or docker swarm
- Deployment must be done with ansible
- A Makefile or a cmd file as discussed in class is needed to deploy the software, start the program, conduct a parameter study/benchmark
- Report
- If project is conducted in a team at least two clouds are to be benchmarked and contrasted 2 team members = 2 clouds, 3 team members = 3 clouds. cloud could also be kubernetes or docker swarm
- Cloudmesh client is to be used to start the virtual cluster in order to avoid reinventing the wheel
- Cloudmesh contains deployments for hadoop and spark. If these technologies are used, it has to be shown that if the student(s) elect to write a new ansible script for it that it is better than the once provided by cloudmesh. Proof is to be provided by reproducible benchmarks. If this can not be achieved the student(s) have to write an additional ansible script for a technologie listed in class or approved by the professor.

Additional links form another class: This other class contains a section deployment

projects that may inspire you. You can look at suggestions and conduct them, the rules listed under requirements above applies: e.g. deployment must be done in ansible and it must be done on a cloud, kubernetes, or docker swarm. I524 will not focus on analytics. However you still are able to do that, but it still must contain a deployment portion.

- projects

2.1.12 Report Format

All reports will be using the format specified in Section *Report Format*.

There will be **NO EXCEPTION** to this format. Documents not following this format and are not professionally looking, will be returned without review.

2.1.13 Github repositories

Class content repository: `https://github.com/cloudmesh/classes`

Class homework repository: `https://github.com/cloudmesh/sp17-i524`

2.1.14 Code Repositories Deliverables

Code repositories are for code, if you have additional libraries or data that are needed you need to develop a script or use a DevOps framework to install such software. They **must** not be checked into github. Thus zip files and .class, .o, precompiled python, .exe, core dumps, and other such files files are not permissible in the project. If we find such files you will get a 20% deduction in your grade. Each project must be reproducible with a simple script. An example is:

```
git clone ....
make install
make run
make view
```

Which would use a simple make file to install, run, and view the results. Naturally you can use ansible or shell scripts. It is not permissible to use GUI based DevOps preinstalled frameworks (such as the one you may have installed in your company or as part of another project). Everything must be installable form the command line.

2.1.15 Learning Outcomes

Students will

1. gain broad understanding of Big Data applications and open source technologies supporting them.
2. have intense programming experience in Python and ansible and DevOps.
3. use open source technologies to manage code in large groups of individuals.

4. be able to communicate reserach in professional scientific reports.

Outcome 1 is supported by a series of lectures around open source technologies for big data.

Outcome 2 is supported by a significant software project that will take up a considerable amount of time to plan and execute.

Outcome 1 and 4 is supported by writing 3 technology papers and a project report that is shared with all students. Students can gain additional insight from reading and reviewing other students contributions.

Outcome 3 is supported by using piazza and github as well as contributiong to the class Web page with git pull requests.

2.1.16 Academic Integrity Policy

We take academic integrity very seriously. You are required to abide by the Indiana University policy on academic integrity, as described in the Code of Student Rights, Responsibilities, and Conduct, as well as the Computer Science Statement on Academic Integrity (`http://www.soic.indiana.edu/doc/graduate/graduate-forms/Academic-Integrity-Guideline-FINAL-2015.pdf`). It is your responsibility to understand these policies. Briefly summarized, the work you submit for course assignments, projects, quizzes, and exams must be your own or that of your group, if group work is permitted. You may use the ideas of others but you must give proper credit. You may discuss assignments with other students but you must acknowledge them in the reference section according to scholarly citation rules. Please also make sure that you know how to not plagiarize text from other sources while reviewing citation rules.

We will respond to acts of plagiarism and academic misconduct according to university policy. Sanctions typically involve a grade of 0 for the assignment in question and/or a grade of F in the course. In addition, University policy requires us to report the incident to the Dean of Students, who may apply additional sanctions, including expulsion from the university.

Students agree that by taking this course, papers and source code submitted to us may be subject to textual similarity review, for example by Turnitin.com. These submissions may be included as source documents in reference databases for the purpose of detecting plagiarism of such papers or codes.

It is not acceptable to use for pay services to conduct your project. Please be aware that we monitor such services and have TAs speaking various languages and know about these services even in different countries. Also do not just translate a report written by someone in a different language and claim it to be your project.

2.1.17 Links

This page is conveniently managed with git. The location for the changes can be found at

- `https://cloudmesh.github.io/classes/`

The repository is at

- `https://github.com/cloudmesh/classes`

Issues can be submitted at

- `https://github.com/cloudmesh/classes/issues`

Or better use piazza so you notify us in our discussion lists.

- `https://piazza.com/iu/i524`

If you detect errors, you could also create a merge request at

- `https://github.com/cloudmesh/classes`

2.1.18 Course Numbers

This course is offered for Graduate (and Undergraduate students with permission) at Indiana University and as an online course. To Register, for University credit please go to:

- `http://registrar.indiana.edu/browser/soc4172/INFO/INFO-I524.shtml`
- `http://registrar.indiana.edu/browser/soc4172/ENGR/ENGR-E599.shtml`

Please, select the course that is most suitable for your program:

```
INFO-I 524  BIG DATA SOFTWARE AND PROJECTS (3 CR)    Von Laszewski G
         Above class open to graduates only
         Above class taught online
         Discussion (DIS)
      30672 RSTR     09:30A-10:45A   M      I2 130    Von Laszewski G
         Above class meets with ENGR-E 599
INFO-I 524  BIG DATA SOFTWARE AND PROJECTS (3 CR)
      30673 RSTR     ARR             ARR    ARR       Von Laszewski G
         Above class open to graduates only
         This is a 100% online class taught by IU Bloomington. No
         on-campus class meetings are required. A distance education
         fee may apply; check your campus bursar website for more
         information

 ENGR-E 599  TOPICS IN INTELL SYS ENGINEER (3 CR)
       VT: BIG DATA SOFTWARE AND PROJECTS
          ***** RSTR     ARR             ARR    ARR       Von Laszewski G
             Discussion (DIS)
       VT: BIG DATA SOFTWARE AND PROJECTS
          33924 RSTR     09:30A-10:45A   M      I2 130    Von Laszewski G
             Above class meets with INFO-I 524
```

2.1.19 References

`http://hpc-abds.org/kaleidoscope/`

2.2 I524 Calendar

Warning: This calendar may be updated based on experience from the class. Please check back here.

Residential classes meet Mondays 09:30A-10:45A, I2 130

Description	**Date/Due Dates**	No
Class Begins	Mon, Jan 9	
Assignment of HID	Fri, Jan 13	
Piazza access verified (HD)	Mon, Jan 16, 9am	C1
Surveys (HD)	Mon, Jan 16, 9am	C2
MLK Jr. Day. Good time for additional class work	Mon, Jan 16	
Github access (needed for TechList)	Mon, Jan 30, 9am	C3
Acces and use of cloud verified (HD)	Mon, Jan 30, 9am	C4
Acces to python 2.7.x verified (HD)	Mon, Jan 30, 9am	C5
Access to Chameleon Cloud (see piazza)	Mon, Jan 30, 9am	C5
TechList.1a - 1.c	Mon, Feb 15, 9am	T0a
Technology Paper 1	Mon, Feb 27, 9am	PA1
BibTeX open discussions on Piazza	Overdue	T0c
Review/fixes of BibTex open discussion on Piazza	Mon, Feb 27, 9am	T0c
TechList.1d and Techlist 2	Mon, Feb 27, 9am	T0b
Python cmd homework	See assignment link	PR1
TechList Peer Review	See assignment link	T0d
Ansible Deployment of Emacs	See assignment link	PR2
Project Information (nothing new)	Sun, Mar 13	
Auto Witdraw	Sun, Mar 12	
Project Execution plan and draft due (HD)	Mon, Mar 13, 9am	P1
Spring Break. Good time for additional class work	Mar 12 - Mar 19	
Technology Paper 2 due	Mon, Mar 27, 9am	PA2
Project Updates due (HD)	Mon, Mar 27, 9am	P2
Project due	Mon, Apr 24, 9am	P3
Last day to submit late Homework	May 1	
Ends	Fri, May 5	

- (HD) hard deadlines must be done in order to obtain full points. These deadlines are important to assure you have access to the resources for the class.
- Programming of A! can be substituted by a Paper 3

2.2.1 Comments

- Any late homework will have an automatic 10% grade deduction.
- Any late homework may result in substential delay in grading (one month or more).
- Hard deadlines can not receive any points for late submissions as they are essential to the communication and operation of the class. If you can naturally not communicate with we can not review your work or you can not even execute your work.
- Experience shows that those using additional time during the spring break do typically better. We recommend that you use this time wisely.
- You can start earlier if you like to prepare for this class, to for example learn Python and ansible. However, lectures may change.
- It would be a mistke not to start working on your project by February 1st. You will run out of time. In order to accomodate for this we have segnificantly reduced other homework requirements in contrast to previous classes we taught.

2.2.2 Official University calendar

- `http://registrar.indiana.edu/official-calendar/official-calendar-spring.shtml`

2.3 I524 Lectures

Warning: This page is under construction, but most lectures are already available. All tracks will change considerably. If you want to work ahead, start with the theory track.

Warning: At the end of the page you find a link to unreleased lectures.

Based on our experience with residential and online classes we will for the first time not require that you have to do the class videos at a prticular time once they are released. This however has the danger that you are not watching them at all and you cheat yourself as you do not allow yourself the educational lessons that this class offers to you. It also requires you to assemble your own schedule for watching the videos that will have to be managed through github as part of a README.md file in your git repository. You will need to do the technology track, the communications track, as well as the theory track.

Theory Track: Some lectures have been designed to introduce you to a number of technologies. These lectures are of more theoretical nature and do not require much hands on activities. Thus you can start them any time.

Collaboration Track: These lectures provide the tools for you to collaborate with your peers and with instructors.

Systems Track: These lectures cover topics that are fundamental to executing your project.

Technology Track: These are lectures with strong technology content and introduce you to using a selected number of technologies as part of the class. It is expected that you will use them as part of the project. Instead of slowing you down with graded homework we expect that you learn these technologies and reuse them as part of the project. It would be a big mistake to start the project 2 weeks befor the semester ends, you will not succeed. You must start your project in the first month of the course. Progress is reported on monthly basis while the report is updated and snapshot every month. We will monitor your progress and include them into the discussion grade. For residential students ther should be no reason why you can not provide a monthly update. For online students a valid update would be: "I changed my company and could not work on the project due to moving". This will give you some points if submitted in time. However, if you submit nothing, we will not issue any points.

2.3.1 Lectures

2.3.2 Lectures - Theory Track

Table 2.2: Theory Track

Topic	Description	Resources	Length	Available
Overview	Course Overview	Slides		Jan 1
	Class Overview - Part 1	Video	11:29	Jan 1
	Class Overview - Part 2	Video	04:10	Jan 1
	Class Overview - Part 3	Video	12:41	Jan 1
Web Page	Course Web Page	Web Page		Jan 1
	Class Web Page - Part 1	Video	11:25	Jan 1
	Class Web Page - Part 2	Video	17:31	Jan 1
Techlist.1	TechList.1 Web Page	Web Page		Jan 1
	TechList.1 Homework	Video	40:08	Jan 14
Introduction	Course Introduction	Slides		Jan 14
	Introduction	Video	0:13:59	Jan 14

Continued on next page

Table 2.2 – continued from previous page

Topic	Description	Resources	Length	Available
	Introduction - Real World Big Data	Video	0:15:28	Jan 14
	Introduction - Basic Trends and Jobs	Video	0:10:57	Jan 14
Acess Patterns	Data Access Patterns and Introduction to using HPC-ABDS	Slides		Jan 14
	1. Introduction to HPC-ABDS Software and Access Patterns	Video - Resource 1	0:27:45	Jan 14
	2. Science Exam-ples (Data Access Patterns)	Video - Resource 2	0:18:38	Jan 14
	3. Remaining General Access Patterns	Video	0:11:26	Jan 14

Continued on next page

Table 2.2 – continued from previous page

Topic	Description	Resources	Length	Available
	4. Summary of HPC-ABDS Layers 1 - 6	Video	0:14:32	Jan 14
	5. Summary of HPC-ABDS Layers 7 - 13	Video	0:30:52	Jan 14
	6. Summary of HPC-ABDS Layers 14 - 17	Video	0:28:02	Jan 14
	Final Part Summary of Stack	Video	0:20:20	Jan 14
Application Structure	Big Data Application Structure	Slides		Jan 14
	NIST Big Data Sub Groups	Video	0:23:25	Jan 14
	Big Data Patterns - Sources of Parallelism	Video	0:23:51	Jan 14
	First and Second Set of Features	Video	0:18:26	Jan 14

Continued on next page

Table 2.2 – continued from previous page

Topic	Description	Resources	Length	Available
	Machine Learning Aspect of Second Feature Set and the Third Set	Video	0:18:38	Jan 14
Application Aspects	Aspects of Big Data Applications	Slides		Jan 14
	Other sources of use cases and Classical Databases/SQL Solutions	Video	0:16:50	Jan 14
	SQL Solutions - Machine Learning Example - and MapReduce	Video	0:18:49	Jan 14
	Clouds vs HPC - Data Intensive vs. Simulation Problems	Video	0:20:26	Jan 14
Applications	Big Data Applications and Generalizing their Structure	Slides		Jan 14
	NIST UseCases and Image Based Applications Examples I	Video	0:25:20	Jan 14
	Image Based Applications II	Video	0:15:23	Jan 14
	Internet of Things Based Applications	Video	0:25:25	Jan 14
	Big Data Patterns - the Ogres and their Facets I	Video	0:22:44	Jan 14

Continued on next page

Table 2.2 – continued from previous page

Topic	Description	Resources	Length	Available
	Facets of the Big Data Ogres II	Video	0:15:09	Jan 14
Other	More of Software Stack	Video	0:24:00	Jan 14

2.3.3 Lectures - Collaboration Track

Table 2.3: Collaboration Track

Topic	Description	Resources	Length
Organization	Lessons vs Lectures	Web Page	
Piazza	Information about Piazza	PDF	
Web Page	Contributiing to the Web Page	Web Page	
Github	Overview and Introduction	Web page	
	Install Instructions	Web page	
	config	Video	2:47
	fork	Video	1:41
	checkout	Video	3:11
	pull	Video	4:26
	branch	Video	2:25
	merge	Video	4:50
	rebase	Video	4:20
	GUI	Video	3:47
	Windows - unsupported	Video	1:25
Paper	How to write a paper by Simon Peyton Jones	Video	34:24
	LaTeX - Overview of LaTeX Resources	Web Page	
	(optional) ShareLaTeX	Video	8:49
	jabref	Video	14:41
	bibtex	Web Page	
	Report Format	Web Page - Git - PDF	
RST	(Draft) Restructured Text	Web Page	

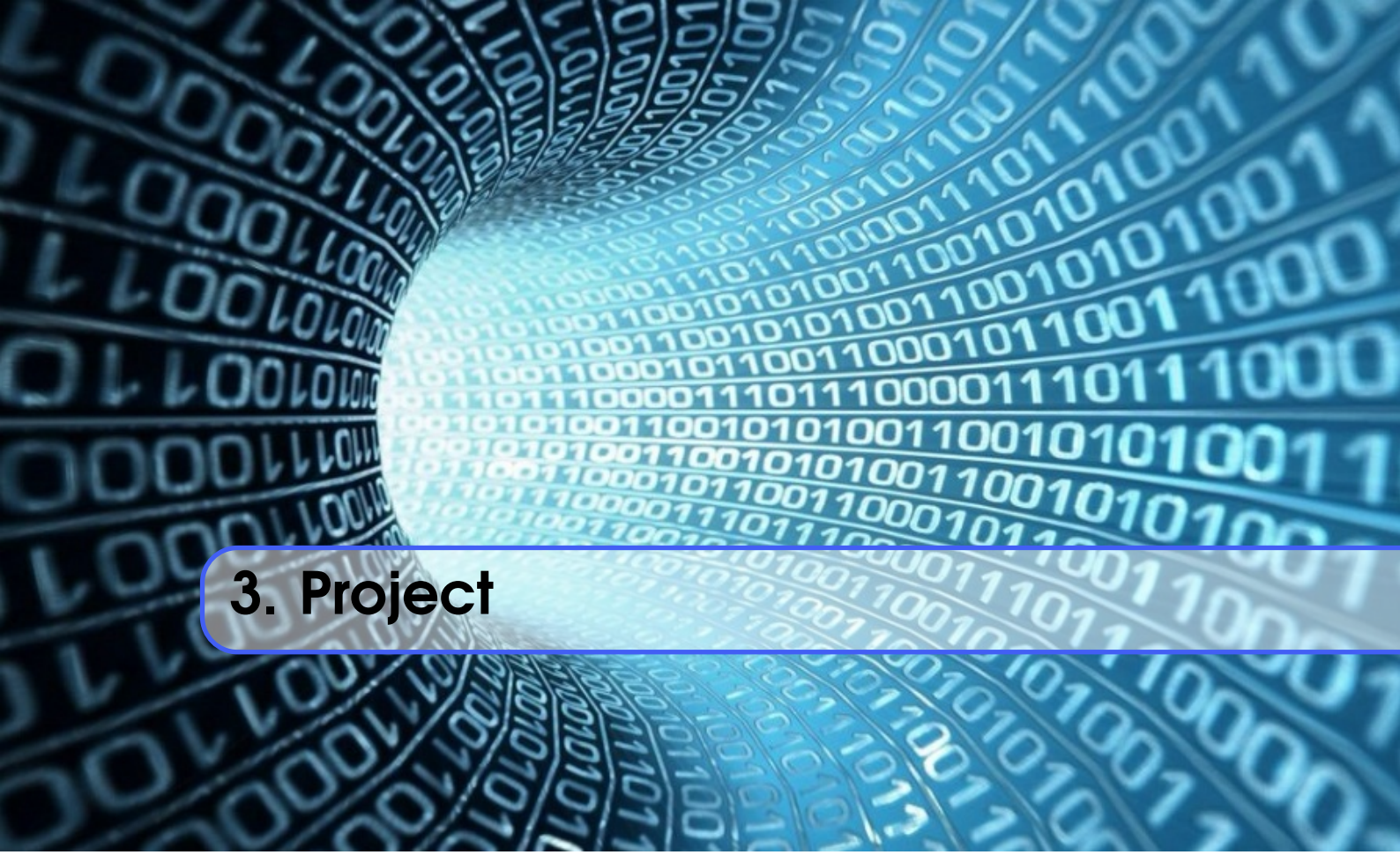
2.3.4 Lectures - Systems

Table 2.4: Systems Track

Treat	Quantity	Description	Length	Available
Ubuntu	Development OS for the class	*Web page*		7 Jan
Virtualbox	Virtualbox for class	*Web page*		7 Jan
	Instalation of ubuntu in virtualbox	Video		7 Jan
	Instalation of guest additions in virtualbox	Video		7 Jan
Shell	Linux Shell	Video | *Web Page*		7 Jan
Python	Introduction to Python	*Web page*		7 Jan
	Python for Big Data	*Web Page*		7 Jan
	Python CMD	*Web Page*		7 Jan
	(Optional) Python pyenv	*Web Page*		Feb 24
	(Draft - Advanced) Python Fingerprint example	*Web page*		7 Jan
	PyCharm	Video		7 Jan
Refcards	Reference cards	*Web Page*		7 Jan
Emacs	(Optional) Useful emacs commands	*Web Page*		7 Jan
Cloudmesh Client	Installation of Cloudmesh Client	Video | *Web Page*		14 Feb
Ansible	Starting Point	Video | *Web Page*		
	Roles and Others	Video | *Web Page*		
	Ansible Galaxy	*Web Page*		

2.3.5 Unreleased Lectures

A list of unrelease lectures that we are currently working on is available here: ref-unreleased



3. Project

3.1 Project

The main activity of the course will be building a significant project using multiple subsystems combined with user code and data. Projects will be suggested or students can chose their own. A project report will summarize the work conducted.

Topics taught in this class will be very relevant for industry as you are not only exposed to big data, but you will also be practically exposed to DevOps and collaborative code development tools as part of your homework and project assignment.

3.1.1 Project Selection and Approval

Each project must be approved by the TAs and the Professor. This is done in an itterative process in which the students needs to first provide a 2 page description of the project detailing the project and its execution plan. This is a *snapshot* of a draft of the actual report that will be handed in at the end. Thus, you do not have to write in your proposal the words, we propose, or in this proposal. You can directly write in this report ... You can actually write directly the project report and leave sections out or put in TBD as text. Important is that the abstract and the introduction is there, as well as an execution plan (e.g. reminding you what you can do realistically when).

3.1.2 Selected Project Ideas

Students can select from a number of project ideas. We will improve and add additional ideas throughout the semester. Students can also conduct their own projects. We

recommend that you identify a project idea by the end of the first month of the class. Example project descriptions that you may want to take a look at include:

- robotswarm
- dockerswarm
- kubernetes
- slurmcluster
- authordisambiguity_b
- NIST Big Data Working group examples: Selected and approved use case from `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-3.pdf`
- Selected examples from Fall I523: Some students may have created an example as part of I523. Not all examples created as part of this class qualify for a I524 project. Please contact Gregor von Laszewski via Piazza to discuss suitability of your previous I523 project. If such a project is selected, approved and used it is expected it is significantly enhanced.
- Cloudmesh Enhancements: A number of projects could center around the enhancements of cloudmesh for the improvement of big data projects using virtual machines and containers. This includes:
  - Development of REST services for cloudmesh while using cloudmesh client
  - Development of benchmarking examples while using cloudmesh client
  - Development of a better Azure interface to additinal services
  - Development of a better AWS interfac to additinal services
  - Development of a Web interface while using django
  - SLURM integration to create virtual clusters on comet
  - Port cloudmesh client to Windows 10
  - Integrate docker into cloudmesh and demonstrate its use
  - Integrate kubernetes into cloudmesh and demonstrate its use
  - Expand the HPC capabilities of cloudmesh

3.1.3 Other Examples

Example projects can also include the following, but must include a benchmark

- deploy Apache Spark on top of Hadoop
- deploy Apache Pig on top of Hadoop
- deploy Apache Storm
- deploy Apache Flink
- deploy a Tensorflow cluster
- deploy a PostgreSQL cluster
- deploy a MongoDB cluster
- deploy a CouchDB cluster
- deploy a Memcached cluster
- deploy a MySQL cluster
- deploy a Redis cluster
- deploy a Mesos cluster
- deploy a Hadoop cluster

- deploy a docker swarm cluster
- deploy NIST Fingerprint Matching
- deploy NIST Human Detection and Face Detection
- deploy NIST Live Twitter Analysis
- deploy NIST Big Data Analytics for Healthcare Data and Health Informatics
- deploy NIST Data Warehousing and Data mining

3.1.4 Datasets

Links Datasets that may inspire a project and benchmarking them are

- `https://www.data.gov/`
- `https://github.com/caesar0301/awesome-public-datasets`
- `https://aws.amazon.com/public-data-sets/`
- `https://www.kaggle.com/datasets`
- `https://cloud.google.com/bigquery/public-data/github`
- `https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public`

For NIST Projects:

- NIST Special Database 27A [4GB]
- INRIA Person Dataset
- Healthcare data from CMS
- Uber Ride Sharing GPS Data
- Census Data

3.1.5 For previous I523 class participants

If you have not yet done an ansible deployment as part of your I523 project you are allowed to continue it as part of this class. Please note that the focus of I523 allowed you to not conduct a deployment and a benchmark. I524 **requires** you to conduct a deployment with ansible and cloudmesh client, as well as benchmarking the application on a real cloud (e.g. chameleoncloud.org).

3.1.6 Is there a sample report?

Due to the variability of the project we also do not have a sample report for a sucessfully conducted project. However the papers written in class as well as the homework to develop an ansible deployment will provide you with sufficient clarity how to be successful.

3.1.7 Project Deployments

Students of this class will need to conduct their project deployments in Python using ansible and enabling a software stack that is useful for a big data analysis. You will be expected to have a computer on which you have python 2.7.x installed. You will be using chameleoncloud.org and possibly our local cloud. Optionally some projects may use

docker. If your project uses docker you can use docker files, but you still need to show its running on 3 different computers.

If your project uses neither, you have to make sure that you hand in a software stack deployment done very well on some software related to the 300+ software systems. You can pick what you want, but shuld not be as simple as installing emacs or R. For example a sharded mongodb or cansandra deployment, a distributed deployment of hadoop (some students asked for this one despite that we had already one like this. Based on student feedback we allow you to do that. and many others, ask for approval however).

3.1.8 Technology deployment Homework

Some students may elect to chose as the homework to deploy a technology with ansible, a technology that is actually used as part of the project. This is naturally a very good way of minimizing your work while building and expanding upon the technology homework you elect to conduct. Points may depend on completeness, effort of the deployment.Technology deployments should as much as possible be non overlapping. In many cases you chose wisely such deployments may line up with your technology papers as you can add a section reporting on your achievement and experience with such deployments.

3.1.9 Group Work

Groups of up to three students can work on a project but workload increases with each student and a work break down must be provided. More than three students are not allowed. If you work in a group you will be asked to deploy a larger system or demonstrate deployability on multiple clouds or container frameworks while benchmarking and comparing them. A group project containing 2 or 3 team members should not look like a project done by an individual. Please plan careful and make sure all team members contribute.

As we get this question often: No we will not allow more than three students to participate in a project. Please do not ask.

3.1.10 We monitor progress for grades

We monitor your progress in Github and you will get *Discussion* points for this. Thus it is imperative you do **Frequent checkins**: It is **important** to make frequent and often commits to the github repository as the activities will be monitored and will be integrated into the project grade. For example, if you elect to just check in your project at the end of the semester while not using github, you will miss points.

3.1.11 Time Management

Note that paper and project will take a considerable amount of time and doing proper time management is a must for this class. Avoid starting your project late. Procrastination does not pay off. Too often we see a student starting their project in the week before it is due.

We can guarantee you this will be problematic. To force you to think about your time management we require that your report contains a section **Project Execution Plan**, that documents when you approximately do what.

3.1.12 Focus on your project

We will not accept any bonus projects or secondary projects as we want that you focus on your class project. If you would have time to do a second project, we recommend you add or integrate it in your actual project spo you can achieve your best. One exccelent project is better than two good projects.

3.1.13 Chance for publishing a paper

If however you find that the work you do could lead to a publishable paper, you could work together with the course instructor as coauthors to conduct such an activity. However, this is going to be a significant effort and you need to decide if you like to conduct this. In such cases if the work is sufficient for publication submission, an A+ for the class could be considered. It will be a lot of work. The length of such a paper is typically 10-12 high quality pages including figures and references. We may elect for the final submission to use a different LaTeX style

3.1.14 Piazza

All project related discussions must be conducted in the **piazza** folder.

3.1.15 Grading

Some students form the class asked for a precise grading scheme. However, based on previous pobservation with other classes a truly outstanding project will not really need a grading scheme.

However as we got asked we propose the following:

```
ansible 30%
benchmarking 30%
paper 30%
wow factor 10%
```

The *wow factor** is given if one of the three other components of the project is impressively well done.

Be reminded that the benchmark must involve multiple vms In case you work as team, the benchmark must include multiple clouds‘

3.1.16 Tips

The following tips have been issued and especially apply to the project:

- **Start the project in the first 4 weeks of the class** starting means reading thinking and potentially discussing with other students or TAs
- Do not underestimating the time it takes to do the project.
- Do not forget to include benchmarks in your project.
- Unnecessarily struggling with LaTeX as you do not use an example we provide.
- Not having a computer that is up to date. Update your memory and have a SSD
- Ignoring obvious security rules and not integrating ssh form the start into your projects.
- Not posting passwords into git. For example git does **not** allow to **easily** completely delete files that contain secret information such as passwords. It takes significant effort to do that. Make sure you do add in git on individual files and never just a bulk add.
- Having your coleagues do the work for you
- Underestimating the **time** it takes to do deployments
- Not reading our piazza posts and repeating the same question over and over
- In case of questions ask.

3.1.17 Artifacts

The following artifacts are part of the deliverables for a project

Code: You must deliver the code in github. The code must be compilable and a TA may try to replicate to run your code. You MUST avoid lengthy install descriptions and everything must be installable from the command line. We will check submission. All team members must be responsible for one part of the project.

Project Report: A report must be produced while using the format discussed in the Report Format section. The following length is required:

- 4 pages, one student in the project
- 6 pages, two students in the project
- 8 pages, three students in the project

Work Breakdown: The report contains in an appendix a section that is only needed for team projects. Include in the section a short but sufficiently detailed work breakdown documenting what the team has done. Back it up with commit information from github. Such as how many commits and lines of code a team member has contributed. The section does not count towards the overall length of the paper.
In addition the graders will check the history of checkins to verify each team member has used github to checkin their contributions frequently. E.g. if we find that one of the students has not checked in code or documentation in the same way at other teammates, it will be questioned. An oral exam may be scheduled to verify that the student has contributed to the project. In an oral exam the student must be familiar with **all** aspects of the project not just the part you contributed.

License: All projects are developed under an open source license such as Apache 2.0 License. You will be required to add a LICENCE.txt file and if you use other software identify how it can be reused in your project. If your project uses different licenses, please add in a README.md file which packages are used and which license these packages have while adding a licenses file.

Reproducability: The reproducability of your code is anticipated to be tested twice. It

is tetes by another student or team, it is also tested by a TA. A report of the testing team is provided. Your team will also be responsible for executing as many tests as you have team members on other projects. A reproducability statement should be written with details about functionality, readbility, and report quality. This statement does not have to be written in latex but uses RST.

Requirements:

- Use of cloud resources is mandatory, can be substituted by kubeernetes or docker swarm
- Deployment must be done with ansible
- A Makefile or a cmd file as discussed in class is needed to deploy the software, start the program, conduct a parameter study/benchmark
- Report
- Cloudmesh client is to be used to start the virtual cluster/multiple vms in order to avoid reinventing the wheel
- Cloudmesh contains deployments for hadoop and spark. If these technologies are used, it has to be shown that if the student(s) elect to write a new ansible script for it that it is better than the once provided by cloudmesh. Proof is to be provided by reproducible benchmarks. If this can not be achieved the student(s) have to write an additional ansible script for a technologie listed in class or approved by the professor.

3.1.18 Report Format

All reports will be using the format specified in Section *Report Format*.

There will be **NO EXCEPTION** to this format. Documents not following this format and are not professionally looking, will be returned without review. The format is the same format that we use for the technology papers. Some additional information is provided in the technology paper template.

3.1.19 Github repositories

Class homework repository: `https://github.com/cloudmesh/sp17-i524`

3.1.20 Code Repositories Deliverables

Code repositories are for code, if you have additional libraries or data that are needed you need to develop a script or use a DevOps framework to install such software. They **must** not be checked into github. Thus zip files and .class, .o, precompiled python, .exe, core dumps, and other such files files are not permissible in the project. If we find such files you will get a 20% deduction in your grade. Each project must be reproducible with a simple script. An example is:

```
git clone ....
make install
```

```
make run
make view
```

Which would use a simple make file to install, run, and view the results. Naturally you can use ansible or shell scripts. It is not permissible to use GUI based DevOps preinstalled frameworks (such as the one you may have installed in your company or as part of another project). Everything must be installable form the command line. In many cases it is better not to use shell scripts but actually use the python CMD or even better the CMD5 tools as presented in class

3.1.21 Submission

The project is submitted into github into your project directory. We will refine this section, but the code must be submitted here. No compiled code or data is accepted in this directory. We expect you make weekly pull requests.

If you are working in a team, we will set up a "special project directory" directory for you, so you need to announce teams on Piazza. A post will be made to collect the team information.

Woring Alone

A README.rst file needs to be included that contains the following information (please be mindfull with the spaces, there is an empty line between each field. Additional fields may need to be added as the project proceeds:

```
group: no

project_url: url to the project directory

title: Your Project Title in CamelCase

author: Firstname Lastname

HID: your HID

piazza: your piazza id

github: your github id

repository: the link to the report folder

proposal: report-proposal.pdf

proposal_submission: mm/dd/2017 hh:mmam

report: report.pdf
```

```
report_submission: mm/dd/2017 hh:mmam

status: short one line non breaking sentance about where you are (updated␣
 ↪weekly)

dataset_url: url of the dataset, do not store in repo

deployment: short description of what you deploy

abstract: a copy of the abstract, make sure to use proper
  indentation in RST format

Bibtex Entry
------------

@TechReport{Project_ID_or_HID-project,
  author =      {},
  title =       {},
  institution = {Indiana University},
  year =        {2017},
  type =        {Class Project Report},
  number =      {your HID or project id},
  address =     {Course I524, Spring 2017},
  month =       apr,
  url =         (url of the report.pdf}
}
```

Working in a team

YOu will need to communicate via Piazza with the TAs that will set up a repository for you. All github names of all team members will need to be listed in that request.

Each author has to go to their HID repository and fill out the README.rst while making sure the values ar set as follows:

```
group: yes

project_url: url to the project directory, that will be assigned to you
```

AFter the project directory is created, fill out the README.rst, just as if you do it for a single user, but add in the Author field the list of authors. Use a comma to separate authors.

Please note that we create automatically a proceedings from the README.rst from all students. If you have not filled out the README.rst we will not be able to see your submission.

3.2 Datasets

Below are links to collections of datasets that may be of use for homework assignments or projects.

- `https://www.data.gov/`
- `https://github.com/caesar0301/awesome-public-datasets`
- `https://aws.amazon.com/public-data-sets/`
- `https://www.kaggle.com/datasets`
- `https://cloud.google.com/bigquery/public-data/github`
- `https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public`

For NIST Projects:

- NIST Special Database 27A [4GB]
- INRIA Person Dataset
- Healthcare data from CMS
- Uber Ride Sharing GPS Data
- Census Data

3.3 Report Format

Over the years we got tired of student that asked us how many pages a report needs to be and than turn around and play with spacing, fonts and other space manipulations to circumvent these recommendations. Thus we have adopted a much simpler approach. All reports **must** be written in the same format that we define on this page. Thus we require that all the reports and papers be written in LaTeX while using our **trivial** example template(s).

The template for the report is available from:

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

An example report in PDF format is available:

- report.pdf

It includes some very simple makefile and allows you to do editing with immediate preview as documented in the LaTeX lesson. Due to LaTeX being a trivial ASCII based format and its superior bibliography management you wil save yourself many hours of work.

In case you are in a team, you can use either github/gitlab while collaboratively developing the LaTeX document, use sharelatex, or overleaf.

Your final submission will include the bibliography file as a separate document. All images must be placed in an images folder and submitted in your repository with the originals. When using sharelatex or overleaf you must replicate the directory layout carefully. YOu

must also make sure that all files and directories in sharelatex you use be copied back to github.

Warning: There will be **NO EXCEPTION** to this format. Hence if you do not know latex we recommend you get familiar with it. Documents not written in LaTeX that do not follow the specified format and are not accompanied by references managed with jabref and are not spell checked will be returned without review.

Warning: We found that students using MsWord or Google docs produce generally inferior reports with the danger of having a lower grade. Hence, in order to help you achieve the best grade possible, we no longer accept reports using these tools and require that all documents be written in LaTeX.

3.3.1 Report Checklist

This incomplete list may serve as a way to check if you follow the rules

1. Have you written the report in LaTeX in the specified format?
2. Have you included an Acknowledgement section?
3. Have you included the report in gitlab?
4. Have you specified the HID, names, and e-mails of all team members in your report. E.g. the Real Names that are registered in Canvas?
5. Have you included the project number in the report?
6. Have you included all images in native and PDF format in gitlab in the images folder?
7. Have you added the bibliography file that you managed with jabref
8. Have you added an appendix describing who did what in the project or report?
9. Have you spellchecked the paper?
10. Have you made sure you do not plagiarize?
11. 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?
12. Have you capitalizezed “Figure 3”, “Table 1”, ... ?
13. Any figure that is not referred to explicitly in the text must be removed?
14. Are the figure captions bellow the figures and not on top. (Do not include the titles of the figures but instead use the caption or that information?
15. When using tables put the table caption on top?
16. Make the figures large enough so we can see it, regardless of page restrictions. If needed make the figure over two columns?
17. Do not worry about the figure placement if they are at a different location than you think. Figures are allowed to float. If you want you can place all figures at the end of the report?
18. Do not use the word “I”?
19. Do not artificially inflate your report if you are bellow the page limit and have

nothing to say anymore?

20. If your paper limit is 12 pages but you want to hand in 120 pages, please check first with an instructor ;-)
21. Check in your current work of the report on a weekly basis to show consistent progress?
22. Is in your report directory a README.rst file in it as shown in the example project that we introduced you to?

3.3.2 Exercise

Report.1: Install latex and jabref on your system

Report.2: Check out the project-000 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 realy 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 are lowercas have no underscore or space in the name?

4. I524 FAQ

4.1 FAQ

4.1.1 How do I ask a question?

We are using piazza for asking question. Our intend is to first gather the answer on piazza and than move the answer in some form to the web page if appropriate.

However asking a quaestion may require some effort on your part.

We suggest the following

- Before asking the quetion look at the Class Web site and in piazza. Both have search functions that you can use.
- Do not burry your question in an unrelated post. INstead use the **New Post** button
- Provide enough information in the questions. THis may include:
  - yur name
  - your HID
  - the exact URL where the issue aooruce (if related to the Web page or piaza)
  - detailed description of what the error is about
  - Make use of the online sessions so you can demonstrate the issue if it complicated to describe

A student als pointed to the following post:

`http://www.techsupportalert.com/content/how-ask-question-when-you-want-technical-help.htm`

4.1.2 What are the prerequisites for this class?

We have communicated the prerequisites to the university, but they may have forgotten to add them to the class. The perquisites can be found in the appropriate class overview. Please review them carefully.

See:

- I524 *Prerequisites*

4.1.3 Why is this class not hosted on EdX?

I523 and I524 were the first classes hosted on EdX. We wanted to continue offering them on EdX. However, the university system administrators mentioned to us that their systems are not ready to support this class in EdX. Unfortunately, this is out of our control and we hope that the university will soon update to an EdX system that can host our classes.

4.1.4 Why are you not using CANVAS for communicating with students?

We have found that Piazza supports our needs for communication better than Canvas.

4.1.5 Why are you using github for submitting projects and papers?

This class is about open source technology and we like that you benefit from material others in the class are developing or have developed. All assignments are openly submitted to the class github for everyone to see. As part of the goal of this class is to develop reusable deployments. Such reuse is only possible if the code is publicly available and others can benefit from it.

The technology papers are made accessible so you can read other technology papers and can get an introduction in technologies that you may not yet know about. It also allows others to contribute to your technology papers and improve them.

4.1.6 I am full time student at IUPUI. Can I take the online version?

Yes you can.

If you are an international student, I suggest you verify this with the office and the registrar. There may be some restrictions for international students. Also some degree programs may have a limit or do not allow to take online classes. It will be up to you to verify the requirements with the appropriate administrators.

4.1.7 I am a residential student at IU. Can I take the online version only?

We recommend you take the residential class.

If you are an international student or a student of a particular degree program restrictions may be placed in if and how many online courses you can take. It will be up to you to

contact the appropriate administrative departments including the international student office to verify what is allowed for you. In general international students have such restrictions. Please find out what they are and which section of the course is appropriate for you.

4.1.8 The class is full what do I do?

1. Make sure to put yourself on the waiting list.
2. If you are a residential student show up on the first class in the specified lecture room. More likely than not some students will enroll in more classes than they can do and places will free up. We will create a list and discuss with the registrar what to do.

4.1.9 Do I need to buy a textbook?

No, the resources will be provided for every unit. However, we recommend that you identify useful books for the class that can help you. Examples include

1. "Taming The Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", Bill Franks Wiley ISBN: 978-1-118-20878-6
2. "Doing Data Science: Straight Talk from the Frontline", Cathy O'Neil, Rachel Schutt, O'Reilly Media, ISBN 978-1449358655

If you find good books, we like to add them here.

4.1.10 Why is there no textbook?

We cover a wide range of topics and their subject-matter is constantly undergoing changes. A textbook would be out of date by the time of publishing.

4.1.11 Do I need a computer to participate in this class?

If you are an online student you need access to a computer. If you are a residential student the facilities provided by SOIC will be sufficient. However, as you study involves computers, its probably important to evaluate if a computer will make your work easier.

If it comes to what computer to buy we really do not have a good recommendation as this depends on your budget. A computer running Linux or OSX makes programming probably easier. A windows computer has the advantage of also being able to run Word and ppt (so does OSX). A cheap machine with multiple cores and sufficient memory (16GB+) is a good idea. A SSD will make access to data especially if large data snappy.

For this reason I myself use a Mac, but you probably can get much cheaper machines with similar specs elsewhere.

Other students bought themselves a cheap computer and installed Linux on it so they do not interfere with their work machines or with Windows. Given how inexpensive computers these days are this may be a reasonable idea. However, do not go too cheap have enough memory and use an SSD if you can.

4.1.12 Representative Bibliography

1. Big data: The next frontier for innovation, competition, and productivity
2. Big Data Spring 2015 Class

4.1.13 Where is the official IU calendar for the Fall?

Please follow this link

4.1.14 How to write a research article on computer science?

1. A good lecture about this is presented by Simon Peyton Jones, Microsoft Research
   `https://www.youtube.com/watch?v=g3dkRsTqdDA`

Other resources may inspire you also:

1. `https://globaljournals.org/guidelines-tips/research-paper-publishing`
2. `http://www.cs.columbia.edu/-hgs/etc/writing-style.html`
3. `https://www.quora.com/How-do-I-write-a-research-paper-for-a-computer-science-journal`

4.1.15 Which bibliography manager is required for the class?

We require you use jabref:

1. `http://www.jabref.org/`

4.1.16 Can I use endnote or other bibliography managers?

No. Jabref is best for us and we do require that you hand in all bibliographies while cleaning and transferring them to jabref. We will not accept any other bibliography tool such as:

1. `http://endnote.com/`
2. `http://libguides.utoledo.edu/c.php?g=284330&p=1895338`
3. `https://www.mendeley.com/`
4. `https://community.mendeley.com/guides/using-citation-editor/05-creating-bibliography`
5. `https://www.zotero.org`

4.1.17 Plagiarism test and resources related to that

1. `https://www.grammarly.com/plagiarism-checker`
2. `http://turnitin.com/`
3. `http://www.plagscan.com/plagiarism-check/`

4.1.18 How many hours will this course take to work on every week?

This question can not rely be answered precisely. Typically we have 2-3 hours video per week. However starting from that its difficult to put a real number down as things may also depend on your background.

- The programming load is modest, but requires knowledge in python and linux systems which you may have to learn outside of this class.
- Some students have more experience than others, thus it may be possible to put in 6 hours per week overall, but other may have to put in 12 hours, while yet others may enjoy this class so much that they spend a lot more hours.
- We will certainly not stopping you form spending time in the class. It will be up to you to figure out how much time you will spend.
- Please remember that procrastination will not pay of in this class.
- The project or term paper will take a significant amount of time.

4.1.19 Is all classes material final?

No. Class material can change. Please remember that in a normal class you will be given several hours of lectures a week. They will be released on a weekly basis. What we do here is to release the material as much as possible upfront and **correct** them when we find it necessary to provide improvements or additions. Additionally, we integrate your feedback into the classes. If you find errors on the class Web page or have additions that you want to add, we would like to hear from you. Pull requests can be issued by you so your contributions get acknowledged and rewarded as part of the grade.

4.1.20 What are the changes to the web page?

The changes we make are typically fixing errata or clarification of content. We do attempt to indicate when major change is made.

4.1.21 What lectures should I learn when?

The class is structured in lectures that you can listen to at any time. If you have difficulties with organizing your own calendar, we will develop a sample calendar for you. Please contact us. However we have undergraduates, graduates, residential and online students. We even have students that can only work part of the semester while they use their vacation. Hence, it is impossible for us to provide an exact calendar that satisfies all the different types of students. Hence we appeal to your organizational skills to create a “study” plan for you during the first week of the semester that works for you.

We recommend to do the theory lectures as quickly as possible, but also start learning ansible at the same time as this will be part of your project. You will fail if you assume you can do the project in 2 weeks. You will need to work on it all semester long on weekly basis, starting with learning how to use ansible and cloud resources.

4.1.22 I524: Why are you doing the papers?

Part of doing research is to communicate your thoughts on topics and to be able to analyze and evaluate technologies that may or may not be useful for you. Our goal within this class is for the first time to gather a significant portion of the technologies that you hear about in class and that you get exposed to as part of the technology list into a "proceedings" developed by all students in class. The papers serve also the dual purpose of you learning how to write a paper and use bibliographies.

4.1.23 I524: Why are there no homework to test me on skills such as ansible or python?

We used to do smaller homework in previous classes to evaluate you on your skills. However we found that they did not reflect real-world use cases. By focusing on the project instead, you will be forced to develop these skills.

However, we can provide you with additional ungraded homework that you can conduct to test your skills if you like. Please let us know if you like to do that and we can assign such homework to you.

4.1.24 I524: Why not use chef or another DevOps framework?

We used to use chef and other DevOps frameworks. However we found that for a class grading can not be uniformly conducted while using too many frameworks. We also found that the value of learning on how to collaboratively contribute as part of an opensource class was diminished while a small group were choosing other technologies. These groups complained later on that they had too much work and could not benefit from other students. Hence we make is simple. All DevOps must be provided in ansible. All programming must be provided in python if not an explicit reason exist to use another language or technology such as R or technologies such as neo4j. However all deployment must be done in python and ansible.

4.1.25 I am lost?

Please contact the instructors for your class.

4.1.26 I do not like Technology/Topic/Project/etc?

Please contact the instructors for your class.

4.1.27 I am not able to attend the online hours

Typically we provide many different times for meetings via Zoom. We even schedule within reason special sessions. All of them are however during reasonable hours in United States Easter Standard Time.

4.1.28 Do I need to attend the online sessions?

No. But you can ask any question you want. We found that in previous classes that some students clearly benefitted from online sessions. If you attend them make sure you have a working and tested microphone if possible.

4.1.29 What are the leaning outcomes?

If you feel that they are not clearly stated as part of the course please contact us so that we can clarify the material.

4.1.30 There are so many messages on Piazza I can not keep up.

Residential students typically participate in live lectures in which we discuss with each other important aspects of a topic. As an online class may not have such a lecture, the piazza posts are just a replacement of them. It is required that you read the posts and decide which of them are relevant for you. In a lecture room you will find also that one student asks a question, while the professor answers the question to the entire class.

4.1.31 I find the hosting Web confusing

Once in a while we find that a student finds the hosting of the class material on the class Web page confusing. This confusion can be overcome by doing the following:

1. You may have to take time to explore the Web page and identify what needs to be done for the class. However each class has a clear overview page.
2. You may have to learn to get used to a class that allows you to work ahead.
3. You may have to learn to appreciate the additional material that assist in learning about python, ansible, LaTex, or the many other topics
4. Please do not blame the instructors for things that are out of their control: You may not be aware that it is not the instructors fault that the university is not able to provide us with an EdX server that works for us. Our choice would be to use EdX.

4.1.32 I524: I do not know python. What do I do?

This class requires python. Please learn it. We will be using ansible for the project. This you can acquire as part of the class through self study. There is a section under lessosn that has some elementary python included.

4.1.33 How to solve merge conflict in Pull Request?

Make sure you have upstream repo defined:

```
$ git remote add upstream https://github.com/cloudmesh/classes
```

Backup all your changed files - just in case you need them while merging the changes back

Get latest from upstream:

```
$ git rebase upstream/master
```

In this step, the conflicting file shows up (in my case it was refs.bib):

```
$ git status
```

should show the name of the conflicting file:

```
$ git diff <file name>
```

should show the actual differences. In some cases, it is easy to simply take latest version from upstream and reapply your changes. So you can decide to checkout one version earlier of the specific file.

Note: You can find the version number with:

```
$ git log --oneline
```

You can checkout a specific version with:

```
$ git checkout <version number - e.g. ed13c06> <file name>
```

At this stage, the re-base should be complete. So, you need to commit and push the changes to your fork:

```
$ git commit
$ git rebase origin/master
$ git push
```

Then reapply your changes to refs.bib - simply use the backedup version and use the editor to redo the changes.

At this stage, only refs.bib is changed:

```
$ git status
```

should show the changes only in refs.bib.

Commit this change using:

```
$ git commit -a -m "new:usr: <message>"
```

And finally push the last commited change:

```
$ git push
```

The changes in the file to resolve merge conflict automatically goes to the original pull request and the pull request can be merged automatically

4.1.34 Building cloudmesh/classes in local machine

If you experience following errors, please follow the guideline explained below. Make sure to do the following steps first:

```
sudo apt-get install libssl-dev
```

Follow this link for more info

- `http://cloudmesh.github.io/client/system.html#ubuntu-14-04-15-04`

Pip will give the following error if you have not installed the library:

Pip installation error when installing requirements.:

```
error: command 'x86_64-linux-gnu-gcc' failed with exit status 1
```

4.1.35 Rolling back uninstall of cryptography

> Command
> "/usr/bin/python -u -c "import setuptools, tokenize;__file__='/tmp/pip-build-1vi4of/cryptography/setup.py';f=getattr(tokenize, 'open', open)(__file__);code=f.read().replace('rn', 'n');f.close();exec(compile(code, __file__, 'exec'))" install –record /tmp/pip-gNcw68-record/install-record.txt –single-version-externally-managed –compile" failed with error code 1 in /tmp/pip-build-1vi4of/cryptography/

Trying to build the source with this error:

```
$ make
cd docs; make html
make[1]: Entering directory '/home/albefrt/Documents/github/cloudmesh/
 ↪classes/docs'
sphinx-build -b html -d build/doctrees source build/html
Running Sphinx v1.5.2
Extension error:
Could not import extension sphinxcontrib.fulltoc (exception: No module␣
 ↪named fulltoc)
Makefile:54: recipe for target 'html' failed
make[1]: *** [html] Error 1
make[1]: Leaving directory '/home/sabyasachi/Documents/github/cloudmesh/
 ↪classes/docs'
Makefile:18: recipe for target 'doc' failed
make: *** [doc] Error 2
```

4.1.36 How to sole Merge Conflict in a Pull Request?

Warning: THis FAQ seems duplicated. Also you are allowed to point to content where thsi is already explained with a link. so you do not have to duplicate.

Steps followed to solve merge conflict in pull request.

Make sure you have upstream repo defined:

```
$ git remote add upstream https://github.com/cloudmesh/classes
```

Backup all your changed files - just in case you need them while merging the changes back

Get latest from upstream:

```
$ git rebase upstream/master
```

In this step, the conflicting file shows up (in my case it was refs.bib):

```
$ git status
```

should show the name of the conflicting file:

```
$ git diff <file name>
```

should show the actual differences. May be in some cases, It is easy to simply take latest version from upstream and reapply your changes.

So you can decide to checkout one version earlier of the specific file. At this stage, the re-base should be complete. So, you need to commit and push the changes to your fork:

```
$ git commit
$ git rebase origin/master
$ git push
```

Then reapply your changes to refs.bib - simply use the backedup version and use the editor to redo the changes.

At this stage, only refs.bib is changed:

```
$ git status
```

should show the changes only in refs.bib. Commit this change using:

```
$ git commit -a -m "new:usr: <message>"
```

And finally push the last commited change:

```
$ git push
```

The changes in the file to resolve merge conflict automatically goes to the original pull request and the pull request can be merged automatically

4.1.37 Cheat sheet for Linux commands

Usage of a particular command and all the attributes associated with it, use 'man' command. Avoid using 'rm -r' command to delete files recursively. A good way to avoid accidental deletion is to include the following in your .bash_profile file:

```
alias e=open_emacs
alias rm='rm -i'
alias mv='mv -i'
alias h='history'
```

More Information

`https://cloudmesh.github.io/classes/lesson/linux/refcards.html`

4.1.38 Tips: TechList.1 homework

Warning: why is this not placed in techlist-hw.rst?

4.1.39 Citations

Do not mention the authors of a citation that you use.

Example do not say:

As Gregor von Laszewski pointed out with flowery words in an article published recently .... [1]

Instead use: In [1] ...

Naturally you should use the cite command.

4.1.40 Spelling

- use a space after periods, and commas in a centence
- use a spellchecker
- do the indentation properly as demonstrated in the examples. (use fixed width font to edit RST to see it more easily)

4.1.41 Github

- when dounig your pull request, make sure you do not have any conflists, rebase if needed

4.1.42 Rubric

We already commented on what a good entry looks like so its rather simple, avoid plagiarism, subsections in the text, keep bullet lists minimal, be short but provide enough detail, dont just copy from the web page, relate technology to big data if you can

- a write a good introdcution to the technology that summarizes what it is (and if possible how it relates to big data)
- include the most important refernces and prepare them in correct bibtex format
- check in your contribution (obviously if you can not do that ask for help form the TAs so you get educated on git)
- you get 50% of your points from the writeup and 50% of the points from the bibliography

You are allowed to work in teams to improve your own submissions.

4.1.43 Timeliness

You will safe yourself a lot of hazle if you check in your assignment early. ON the last day typically a lot of checkins happen and may require you to do a rebase. The sooner you do it the easier for you.

4.1.44 Outdated Tech ology

One of the technology assigned to me is 'Ninefold'. It seems ninefold has shutdown their cloud service on January 30, 2016. Should I write a tech summary for ninefold or do we have remove this from the techlist as it is no longer in operation?

Kindly refer:

`http://ninefold.com/`

`http://ninefold.com/news/`

Note: Outdated and unnecessary technologies will be removed by the TAs.

4.1.45 Techlist 1 and Paper 1 : Pagecount

TechLIst = a couple of paragraphs (so real short, see the NAGIOS example

Paper 1 = 2 pages in the format we specified, images and refs not included. See at the end of the paper format for a suitable layout

PS: If your paper is longer or if it a paragraph short that does not matter to us, important is the content

4.1.46 Tips to Install Virtualbox

A video on how to install virtual box on windows 10 can be seen as part of an unrelated course on youtube at

`https://www.youtube.com/watch?v=XvCUpZuHgvo`

It is a bit wordy as the presenter complains about the difficulties to record videos on windows 10, and talks about his course, so just ignore these portions. Naturally use whatever is the newest version. Here is one for Windows 8 which also contains ubuntu install (use the one above on how to install vb on windows 10 and ignore that part form the window bellow)

`https://www.youtube.com/watch?v=13GS1cLyk-E`

4.1.47 Do I generate the SSH key on Ubuntu VM ?

I have installed Ubuntu(on virtual box) on my windows 10 system. I wanted to confirm if the SSH key should be created on the Ubuntu VM? Yes we need to generate ssh on Ubuntu VM, because even it is a VM or a real machine we have to set up ssh in order to work with ssh based communication, in order to maintain security when you are using an application like Github.

You need to generate SSH, no matter what operating system you are using or on which operating system you are running VM.

First let us revisit what an ssh key is for. A key pair has a public and a private key pair. If a remote machine has the public key from another machine you will be able to login to that machine form the machine where you have created the public and private key pair from. Some services do require key authentication. Such services include:

1. login to any virtual machine
2. using github
3. login to the login nodes of futuresystems

Thus if you like ta access any of them any computer on which you want to access them from need a key pair. (or key as we sometimes abbreviate).

So if you like to access from your ubuntu vm future systems which you want you need one, if you want to access githu, you need one, if you want to login to vas on chameleon cloud you need one, if you want to login to vas on jetstreem you need one, if you want .... you need one.

So the answer is yes. Under no circumstances copy the private key to another computer as that is a security violation. You can only copy the public key. That is the reason its called public. On each machine where you like to access these services you need to create a different key and add the public key to the remote services/machines you want to access.

4.1.48 Ways to run Ubuntu on Windows 10

There are multiple ways to get ubuntu onto Windows.

a) The recommended way to do it is via virtual box which seems to work for most, but requires sometimes that the bios settings need to be adjusted. Naturally we do not know what your bios settings are so you need to figure this out from the internet. However in 99% of the cases virtual box works nicely.A student tip describes what needs to be done:

You need the virtual box software (`https://www.virtualbox.org/wiki/Downloads`) that corresponds to the operating system running on the physical machine in front of you. Then download the Ubuntu 16.04 .iso file (`https://www.ubuntu.com/download/desktop`) to your computer. Start virtual box. I think a wizard starts to guide you through setting up a new virtual machine when you choose "new". Then brows to where you downloaded the iso file and click on it. you will have to start this and ubuntu will start installing. (improve this description if something is not clear)

b) the other way of installing bash on windows is as subsystem as documented by your fellow students. This may not fulfill the requirements of running ansible, but it will help you to get started quickly while running bash on your host directly. It is often referred to as "ubuntu on windows".

`http://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10`

If you want to use one method, do a)

How can I download lecture sildes ?

Please refer to the following link. `https://cloudmesh.github.io/classes/i524/lectures.html`

4.1.49 Don't use Anaconda

We use python 2.7.13 for this class. It is better to use Virtualenv and pip. And for the IDE, you can use PyCharm. This is the open source way of doing python, while we use 2.7 because not everything is yet available in 3.5. We do not recommend Anaconda or Canopy. In fact we found issues with both. Especially with Canopy. It was incompatible with libraries the open source community uses and it negatively effected a students system wide python install. We had to reinstall python completely after we uninstalled canopy. Unfortunately it did cost us a lot of time to fix this. TAs will not provide any help in case you use anaconda or canopy.

4.1.50 Using SSH Key for Git Push

When you cloned your repository did you use SSH rather than HTTPS? Your clone command should look like this:

$ git clone git@github.com:YOUR_USERNAME/classes.git

You can use git remote set-url as described here to change from HTTPS to SSH: `https://help.github.com/articles/changing-a-remote-s-url/`

Changing the origin remote (as opposed to both origin and upstream) will be sufficient, since this is the only one you push into.

4.1.51 How to properly research a bibtex entry?

Often you may find via google a bibtex entry that may need some more reserach. Lets assume your first google quesry 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?

1. firts of all 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.

So let us investigate a bit more. Let us search for the title. So we find

1. `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.146073913,d.amc`
2. `https://research.google.com/pubs/pub61.html`
3. `http://dl.acm.org/citation.cfm?id=1239658`

Let us look at A)

As you can see from the url this is actualy 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 lats 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 entries and get a nice bibtex reference. 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 field sin B are already coverd in C, so we take C) and add the url. Now as the label is graet 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 finding a refernce takes multiple google quesries and merging of the results you find from various returms. As you still have time to correct things I advise that you check your refernces and correct them. If the original refernce would have been graded it would have been graded with a “fail” instead of a “pass”.

4.1.52 A second example

Lets look at a second obvious example that needs improvement:

```
@InProceedings{wettinger-any2api,
  Title                    = {Any2API - Automated APIfication},
  Author                   = {Wettinger, Johannes and
                              Uwe Breitenb{\"u}cher
                              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-IO-3005},
  Url                      = {https://pdfs.semanticscholar.org/1cd4/
↪4b87be8cf68ea5c4c642d38678a7b40a86de.pdf}
}
```

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 jst 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 incldes 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 (CLOSER)},
  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 (CLOSER)},
  year = {2015},
  pages = {475--486},
  publisher = {SciTePress},
  url ={https://pdfs.semanticscholar.org/1cd4/
↪4b87be8cf68ea5c4c642d38678a7b40a86de.pdf},
  owner = {S17-IO-3005},
}
```

Should we be satisfied? No, even our original information we gathere 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{\'{\i}}ctor 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 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 =         {}
}
```

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{\'{\i}}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-IO-3005},
}
```

4.1.53 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 refernce passes or fails. We may assign this to you so you get familiar with the identification if a referncei is ok or not.

Please see `https://en.wikipedia.org/wiki/BibTeX`

4.1.54 Can I write the papers on OSX?

Yes of course you can write papers on OSX. But we support for Ubuntu 16.04, because we consider it as the main OS that we use in this class. You can use, VM to install Ubuntu and use it for class work.

4.1.55 What is the nature of team collaboration on papers

You can build teams of three. You need to yourself build the team. The web page tells you that there will be no reduction in numbers of papers you write = number of team members * 3, papers can not be combined.

4.1.56 What is the nature of team collaboration on papers

You can build teams of three. You need to yourself build the team. The web page tells you that there will be no reduction in numbers of papers you write = number of team members * 3, papers can not be combined.

4.1.57 What are the due dates for assignments

Due dates are posed on the Web page calendar.

4.1.58 What are good places to find refernce entries?

- `https://scholar.google.com/`
- `http://dl.acm.org/`
- `http://ieeexplore.ieee.org/`
- `http://dblp.uni-trier.de/`
- `http://academic.research.microsoft.com/`

4.1.59 How to install Matplotlib?

Follow the installation in the class documentation properly.

Install the requirements:

```
$ pip install -r requirements.txt
```

Install matplotlib using pip:

```
$ pip install matplotlib
```

Install python Tkinter packages:

```
$ sudo apt-get install python-tk
```

4.1.60 How to test if your OS can install cloudmesh_client

> In installation of Cloudmesh Client, there may be extra packages that has to be installed.Missing a few dependencies for cryptography.

Since this SO question keeps coming up I'll drop a response here too (I am one of the pyca/cryptography developers). Here's what you need to reliably install pyca/cryptography on the 3 major platforms.

Please note in all these cases it is highly recommended that you install into a virtualenv and not into the global package space. This is not specific to cryptography but rather is generic advice to keep your Python installation reliable. The global package space in OS provided Pythons is owned by the system and installing things via pip into it is asking for trouble.

4.1.61 Windows

Upgrade to the latest pip (8.1.2 as of June 2016) and just pip install cryptography

cryptography and cffi are both shipped as statically linked wheels.

4.1.62 OS X

Ono OSX you need to install xcode.

Upgrade to the latest pip (8.1.2 as of June 2016) and just pip install cryptography

cryptography and cffi are both shipped as statically linked wheels. This will work for pyenv Python, system Python, homebrew Python, etc. As long as you're on the latest pip you won't even need a compiler.

4.1.63 Linux

On Linux you'll need a C compiler, libffi + its development headers, and openssl + its development headers.

Debian or Ubuntu derived distributions

apt-get install build-essential libssl-dev libffi-dev python-dev followed by:

```
pip install cryptography
```

Red Hat derived distributions:

```
yum install gcc openssl-devel libffi-devel python-devel followed by

pip install cryptography
```

4.1.64 Tips to write a Good Paper

This that must be avoided to write a good paper.

Why Technology xyz

(and makeing sure to include a ? ;)

Instead give the section a good name that is not a question, such as

Introduction

Design

Architecture

Performance

Comparison

Big Data Use cases

Conclusion

And there are many more different things.

Make sure to write a good paper avoiding these headings when you start a sub-section in your paper.

5. I524 Technology Collection

5.1 HID Assignment

As part of the class you will be assigne a Homework ID (HID). Some assignments in the class will use this HID to identify which homework you will be doing. Technologies listed with (1) behind it are for the homework TechList.1 and Technologies with a (2) are for TechList.2

Note: The folloing list is the original assignment of the technologies to HIDs. The mapping from the HID to names is stored at this time in Piazza at `https://piazza.com/class/ix39m27czn5uw?cid=33` please make sure we did not make a mistake and if so, please notify us.

Table 5.1: Mappings of HIDs to Techs

Name	HID	Technologies
Sheybani Moghadam Saber	S17-ER-1001	Azure Queues (1) – Sentry (1) – Tableau (1) – Berkeley DB (1) – ODE (1) – OpenStack Keystone (1) – Globus Tools (2)
	•	•

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
Agasti Avadhoot	S17-IO-3000	SQL Server (1) – Nimbus (1) – Taverna (1) – Chef (1) – Tyrant (1) – FITS (1) – DataTurbine (2)
Bays Christopher	S17-IO-3002	TensorFlow (1) – Azure Stream Analytics (1) – Ambari (1) – Galaxy (1) – Bioconductor (1) – OPeNDAP (2) – BlinkDB (2)
Carmickle Ricky	S17-IO-3003	QPid (1) – Stomp (1) – Apatar (1) – Google FlumeJava (1) – Sqrrl (1) – Scalding (2) – OSGi (2)
Coulter Cory	S17-IO-3004	appfog (1) – Dream:Lab (1) – MySQL (1) – ZHT (1) – RYA (1) – Summingbird (2) – SQLite (2)
Gupta Abhishek	S17-IO-3005	Amazon Kinesis (1) – Inca (1) – Gora (general object from NoSQL) (1) – RabbitMQ (1) – JClouds (1) – Megastore and Spanner (2) – Any2Api (2)
Kodre Vishwanath	S17-IO-3008	CINET (1) – Linux-Vserver (1) – Networking: Google Cloud DNS (1) – Talend (1) – Haystack (1) – PolyBase (2) – Docker (Machine, Swarm) (2)
Kshirsagar Hemant	S17-IO-3009	Flink Streaming (1) – Solr (1) – JGroups (1) – Azure SQL (1) – HDF (1) – Torque (2) – Databus (2)
Lawson Matthew	S17-IO-3010	Azure (1) – Couchbase (1) – Public Cloud: Azure Table (1) – Sawzall (1) – Phoenix (1) – CouchDB (2) – Disco (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
McClary Scott	S17-IO-3011	ZeroMQ (1) – Blueprints (1) – Trident (1) – e-Science Central (1) – Winery (1) – Crunch (2) – Airavata (2)
McCombe Mark	S17-IO-3012	MRQL (1) – AWS OpsWorks (1) – GPFS (1) – Hazelcast (1) – Google Bigtable (1) – Google Prediction API & Translation API (2)
Mwangi Leonard	S17-IO-3013	Google Prediction API and Translation API (1) – LMDB (key value) (1) – QEMU (1) – BioKepler (1) – Google Cloud Dataflow (1) – Pregel (2)
Rai Piyush	S17-IO-3014	Riak (1) – Ehcache (1) – Xen (1) – Zookeeper (1) – SSH (1) – SciDB (2)
Roy Choudhury Sabyasachi	S17-IO-3015	Lucene (1) – pbdR (1) – Protobuf (1) – Galera Cluster (1) – Cassandra (1) – Mbase (2)
Rufael Ribka	S17-IO-3016	DC.js (1) – Aerobatic (1) – CoreOS (1) – AMQP (1) – Argo BEAST HPX-5 BEAST PULSAR (1) – Apache Derby (2)
Sathe Nandita	S17-IO-3017	Facebook Tao (1) – MongoDB (1) – Amazon (1) – Kafka (1) – Amazon Dynamo (1) – Blazegraph (2)
Shane Kevin	S17-IO-3018	PostgreSQL (1) – Impala (1) – Hadoop (1) – Floe (1) – VirtualBox (1) – Ubuntu MaaS (2) – Pig (2)
Smith Michael	S17-IO-3019	Stackato (1) – Parasol (1) – vSphere and vCloud (1) – Totem (1) – Libvirt (1) – Xcat (2) – InCommon (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
Suryawanshi Milind	S17-IO-3020	AppScale (1) – CloudControl (1) – Google Fusion Tables (1) – Yarn (1) – TinkerPop (1) – LinkedIn (2) – CloudML (2)
Thakre Abhijit	S17-IO-3021	CUBRID (1) – MR-MPI (1) – NWB (1) – Cascading (1) – BitTorrent (1) – Tez (2) – Rocks (2)
Unni Sunanda	S17-IO-3022	Juju (1) – Netty (1) – FUSE (1) – Google Chubby (1) – Mesos (1) – Pivotal GPLOAD/GPFDIST (2) – Yarcdata (2)
Venkatesan Karthick	S17-IO-3023	H-Store (1) – Kyoto Cabinet (1) – Globus Online (GridFTP) (1) – Sahara (1) – DataFu (1) – Facebook Tupperware (2) – Lambda (2)
Vuppada Ashok	S17-IO-3024	NiFi (NSA) (1) – LXC (1) – Helix (1) – IBM dashDB (1) – Puppet (1) – Google Cloud SQL (2) – Giraph (2)
Yezerets Helen	S17-IO-3025	Voldemort (1) – Buildstep (1) – OCCI (1) – SAP HANA (1) – HPX-5 (1) – IPython (2) – CloudMesh (2)
	•	•
Akurati Niteesh Kumar	S17-IR-2001	Celery (1) – GraphBuilder(Intel) (1) – HTCondor (1) – HUBzero (1) – Gitreceive (1) – Pivotal Greenplum (2) – Infinispan (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
ARDIANSYAH JIMMY	S17-IR-2002	Stratosphere (Apache Flink) (1) – ActiveBPEL (1) – Google Dremel (1) – ImageJ (1) – IBM Cloudant (1) – Kepler (2) – Amazon Redshift (2)
Balaga Ajit	S17-IR-2004	PLASMA MAGMA (1) – Samza (1) – Azure Blob (1) – OpenVZ (1) – Jelastic (1) – Jupyter (2) – Kibana (2)
Chemburkar Snehal Shrish	S17-IR-2006	Cinder (1) – Spark (1) – R (1) – dotCloud (1) – Pivotal Gemfire (1) – PyBrain (2) – Engine Yard (2)
Anbazhagan Karthik	S17-IR-2008	Kestrel (1) – Scalapack (1) – HadoopDB (1) – OODT (1) – Thrift (1) – Mahout (2) – Moab (2)
Jain Anurag Kumar	S17-IR-2011	DL4j (1) – Solandra (1) – CloudStack (1) – Logstash (1) – Ansible (1) – Hyper-V (2) – Swift (2)
Jain Pratik	S17-IR-2012	GraphLab (1) – GFFS (1) – Lustre (1) – Reef (1) – Harp (1) – LevelDB (2) – Event Hubs (2)
Korrapati Sahiti	S17-IR-2013	Flume (1) – OpenCV (1) – ORC (1) – VMware ESXi (1) – Hama (1) – DevOpSlang (2) – Accumulo (2)
Krishnakumar Harshit	S17-IR-2014	Sqoop (1) – Pivotal (1) – Google MillWheel (1) – iRODS (1) – VoltDB (1) – OpenPBS (2) – Kite (2)
Lingampalli Anvesh Nayan	S17-IR-2016	ActiveMQ (1) – OpenTOSCA (1) – Avro (1) – SaltStack (1) – Whirr (1) – MLlib (2) – GraphChi (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
Marni Veera	S17-IR-2017	Eduroam (1) – Potree (1) – Pivotal HD/Hawq (1) – Docker Compose (1) – OpenNebula (1) – point-to-point (2) – Neptune (2)
Merugureddy Bhavesh Reddy	S17-IR-2018	CompLearn (1) – OpenID (1) – Cisco Intelligent Automation for Cloud (1) – Pentaho (1) – scikit-learn (1) – Google and other public Clouds (2) – Llama (2)
Methkupalli Vasanth	S17-IR-2019	Oracle (1) – CNTK (1) – Twister (1) – NetCDF (1) – Oozie (1) – KeystoneML (2) – Lumberyard (2)
Mishra Govind	S17-IR-2021	Docker Machine and Swarm (1) – Shark (1) – Ligra (1) – Redis (1) – Facebook Puma/Ptail/Scribe/ODS (1) – AWS Elastic Beanstalk (2) – Facebook Corona (2)
Naik Abhishek	S17-IR-2022	Sesame (1) – Pilot Jobs (1) – Red Hat OpenShift (1) – Google Pub Sub (1) – Boto (1) – Triana (2) – IBM System G (2)
Parekh Ronak	S17-IR-2024	Cobbler (1) – GraphX (1) – Memcached (1) – graphdb (1) – LDAP (1) – Spark SQL (2) – Splunk (2)
Raghatate Rahul	S17-IR-2026	Ceph (1) – CDF (1) – Jitterbit (1) – Naiad (1) – publish-subscribe: MPI (1) – Google F1 (2) – NaradaBrokering (2)
Ramachandran Shahidhya	S17-IR-2027	DataNucleus (1) – Razor (1) – Twitter Heron (1) – Amazon RDS (1) – SAML OAuth (1) – (Dryad) (2) – DB2 (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
Ramanam Srikanth	S17-IR-2028	Spark Streaming (1) – Libcloud (1) – Google Kubernetes (1) – mlpy (1) – Dokku (1) – N1QL (2) – PetSc (2)
Ramaraju Naveenkumar	S17-IR-2029	Galois (1) – Slurm [Slu] (1) – Giraffe (1) – Azure Machine Learning (1) – Ninefold (1) – CDMI (2) – OpenStack Ironic (2)
Ravi Sowmya	S17-IR-2030	UIMA (1) – Jena (1) – Tycoon (1) – Azure Data Factory (1) – Google Cloud DataFlow (1) – Medusa-GPU (2) – Neo4J (2)
Satyam Kumar	S17-IR-2031	Google Cloud Storage (1) – EclipseLink (1) – Torch (1) – Caffe (1) – Parquet (1) – Rasdaman (2) – DAAL(Intel) (2)
Sharma Yatin	S17-IR-2034	rkt (1) – Heroku (1) – Pegasus (1) – Drill (1) – Titan:db (1) – OpenStack (2) – Espresso (2)
Shinde Piyush	S17-IR-2035	ODBC/JDBC (1) – f4 (1) – Oracle PGX (1) – Eucalyptus (1) – D3.js (1) – Ganglia (2) – Amazon Route 53 (2)
Singh Rahul	S17-IR-2036	OpenStack Heat (1) – Saga (1) – Agave (1) – Storm (1) – JMS (1) – Graylog (2) – Google App Engine (2)
Sitharaman Sriram	S17-IR-2037	Public Cloud: Amazon SNS (1) – FTP (1) – HBase (1) – MQTT (1) – RCFile (1) – OpenJPA (2) – SGE (2)
Sivaprasad Sushmita	S17-IR-2038	Terraform (1) – H2O (1) – KVM (1) – Cloud Foundry (1) – CloudBees (1) – Marionette Collective (2) – three.js (2)

Continued on next page

Table 5.1 – continued from previous page

Name	HID	Technologies
Suri Naren	S17-IR-2039	TOSCA (1) – HTTP (1) – IBM BlueMix (1) – Google Omega (1) – Gluster (1) – Google DataStore (2) – MapGraph (2)
Vora Sagar	S17-IR-2041	IBM Watson (1) – Public Cloud: Amazon S3 (1) – Kyoto/Tokyo Cabinet (1) – Elasticsearch (1) – Tajo (1) – Google BigQuery (2) – S4 (2)
Yadav Diksha	S17-IR-2044	AllegroGraph (1) – Theano (1) – Atmosphere (1) – Granules (1) – HDFS (1) – Hibernate (2) – Hive (2)
	•	•
TA	S17-TS-0001	Mahout (1)
TA	S17-TS-0001	Tika (1)
TA	S17-TS-0003	HCatalog (1)
TA	S17-TS-0004	Foreman (1)
TA	S17-TS-0005	Genesis (1)
TA	S17-TS-0006	Presto (1)
TA	S17-TS-0007	Nagios (1)

5.2 Technologies

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`

5.2.1 Workflow-Orchestration

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 *[1]*. 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 *[2]*. 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 *[3]*.

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 use to administer, to control and to integrate web services into a larger application. *[4]*
3. Airavata
   Apache Airavata *[5]* 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 *[6]*, SEAGrid *[7]* and GenApp *[8]*).
4. Pegasus
   The Pegasus *[9]* is workflow management system that alows 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.
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 *[10]*.

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. *[11]*
   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.
7. Taverna
   Taverna is workflow management system. According to *[12]*, Taverna is transitioning to Apache Incubator as of Jan 2017. Taverna suite includes 2 products:
   (a) Taverna Workbench is desktop client where user can define the workflow.
   (b) 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 *[13]* paper, the formal syntax and operational semantics of Taverna is explained.
8. Triana
   [14] Triana is an open source problem solving software that comes with powerful data analysis tools. 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.
9. Trident
   In *[15]*, 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, *[15]* 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 *[16]*, 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 *[15]*, 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."

10. BioKepler
    BioKepler is a Kepler module of scientific workflow components to execute a set of bioinformatics tools using distributed execution patterns *[17]*. 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 *[18]*. BioKepler contains over 40 example workflows that demonstrate the actors and directors *[19]*.
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. *[20]* 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. *[21]* 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." *[22]* provides the project documents for Ansible Galaxy on github.
12. IPython
13. Jupyter
    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. *[23]* 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. *[24]* 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. *[25]* 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. *[26]* 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. *[27]* 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. *[28]*

14. (Dryad)
15. Naiad

    Naiad *[29]* 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 *[30]*.

    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.
16. 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 *[31] [32]*. Oozie *[33]* 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 effective ly. As demand for workflow and the sophistication of applications increase, it must continue to mature in these areas *[31]*.Is well integr ated with Hadoop security. Is the only workflow manager with built-in Hadoo p actions, making workflow development, maintenance and troubleshooting easi er. 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 *[31]*. Gets callbacks from MapReduce jobs so it knows when they finish and whether they hang without expensive polling. Oozie Coordinat or allows triggering actions when files arrive at HDFS. Also supported by Hadoop vendors *[31]*.
17. 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. *[34]* 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 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.

18. Google FlumeJava
19. Crunch
    Arvados Crunch *[35]* is a containerized workflow engine for running complex, multi-part pipelines or workflows in a way that is flexible, scalable, and supports versioning, reproducibilty, and provenance while running in virtualized computing environments. The Arvados Crunch *[36]* 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.
20. Cascading
    [37] 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.
21. Scalding
22. e-Science Central
    In *[38]*, 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 *[38]*, 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 *[39]*, 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 *[38]*, 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 *[38]*, "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."

23. 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 *[40]*. 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 *[41]* 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.
24. 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 *[42]*. 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 *[43]* the underlying data piping and process infrastructure.
25. NiFi (NSA)
    [44] Defines NiFi as "An Easy to use, powerful and realiable system to process and distribute data". This tool aims at automated data flow from sources with different sizes , formats and following diffent protocals to the centralized location or destination. *[45]*.
    This comes equipped with an easy use UI where the data flow can be conrolled with a drag and a drop. NiFi was initiatially developed by NSA (called Niagarafiles) using the concepts of flowbased programming and latter submitted to Apachi Software foundation. *[46]*
26. Jitterbit

Jitterbit *[47]* 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 *[48]*. 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 *[49]*. 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.

27. 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.
28. 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 *[50]*. 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 *[51]*. 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.
29. Apatar
30. 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

enviorment . 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. :cite:'docker-book' Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands. 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 ,:cite:'www-docker"'you can scale out your application as if it were running on a single, huge computer" as swarm can be scaled upto 1000 Nodes or upto 50,000 containers

31. 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 *[52]*. This approach increases ease of use and higher performance over existing systems for large scale learning *[52]*. 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 *[53]*. 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 *[54]*.

5.2.2 Application and Analytics

32. Mahout *[55]*
    "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 algorithms for Scala + Apache Spark, H2O, Apache Flink (3) Samsara, a vector math experimentation environment with R-like syntax which works at scale"
33. MLlib
34. MLbase
    MLBase *[56]* is a distributed machine learning system built with Apache Spark *[57]*. 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 *[58]*, *[59]* and *[60]* are available on distributed machine learning with MLBase.

35. 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 *[61]* 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.
36. 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 *[62]*. 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. It's strength lies in analyzing data using its rich library but falls short when working with very large datasets *[63]*.
37. pbdR
    Programming with Big Data in R (pbdR) *[64]* 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 accross 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. Currenly there are two implementation of pbdR, one Rmpi and another being pdbMpi. Rmpi uses SPMD parallelism while pbdRMpi uses manager/worker parallelism.
38. 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. *[65]* 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" *[66]* 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 "genemoic 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.

39. 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 scientifif 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. *[67]*
40. 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) *[68]*. OpenCV 3.2 is the latest version of the software that is currently available *[69]*.
41. 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. *[70]*
    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.
42. PetSc
43. 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. *[71] [72]* 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: *[73] [74]* 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. *[75][76][77]*

44. 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 *[78].* 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.
45. 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 *[79].*

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 *[80]*.

46. mlpy
    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. *[81]* explains the significanceq 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. *[82]* 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.
47. 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 *[83]*. 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 sciki-learn *[84]*. 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.
48. PyBrain *[85]*
    The goal of PyBrain is to provide flexible, easyto-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.

49. CompLearn
Complearn 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 complearn 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 *[86]*. 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 *[86]*. Complearn provides immediate to access every core functionality in all the major languages as it is designed to be extensible.
50. DAAL(Intel)
DAAL stands for Data Analytics Acceleration Library. DAAL is software library offered by Intel which is written in C++, python, and Java which implements algorithm for doing efficient and optimized data analysis tasks to solve big-data problems. *[87]*. 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 inlude 10-12 other important algorithms.
[88] 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.
51. Caffe
Caffe is a deep learning framework made with three terms namely expression, speed and modularity *[89]*. 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 comes 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 bu academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia.
52. Torch

Torch is a open source machine learning library, a scientific computing framework *[90]* .It implements LuaJIT programming language and implements C/CUDA. It implements N-dimensional array. It does routines of indexing, slicing, transposing etc. It has in 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 of 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.

53. 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 *[91]*.
54. DL4j
    DL4j stands for Deeplearning4j. *[92]* 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 a 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.
55. H2O
    It is an open source software for big data analysis. It was launched by the Start-up H2O in 2011. [www-H2O-website] 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. It is written in Java. [www-H20-book] 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. 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. *[93]* It can be used on cloud computing environments such as Amazon and Microsoft Azure.
56. IBM Watson

IBM Watson *[94]* 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. *[95]* 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. *[95]* mentions that today Watson is available as a set of open source APIs and Software As a Service product as well.

57. 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" *[96]*. Graphs can be loaded from various sources like SQL and NoSQL databases, Apache Spark and Hadoop *[97]*.
58. GraphLab
    GraphLab *[98]* 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.
59. GraphX
    GraphX is Apache Spark's API for graph and graph-parallel computation. *[99]*
    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. *[100]*
It gets developed as a part of Apache Spark project. It thus gets tested and updated with each Spark release.

60. IBM System G
[101] IBM System G provides a set of Cloud and Graph computing tools and solutions for Big Data. In fact, the G stands for Graph and typically spans a database, visualization, analytics library, middleware and Network Science Analytics tools. *[102]* It assists the easy creating of graph stores and queries and exploring them via interactive visualizations. 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. *[103]* Some of the typical applications wherein it can be used include Expertise Location, Commerce, Recommendation, Watson, Cybersecurity, etc.
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.
61. GraphBuilder(Intel)
Intel GraphBuilder for Apache Hadoop V2 is a software that is used to build graph data models easily enabiling 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.:cite:*graphbuilder*
62. TinkerPop
ThinkerPop is a graph computing framework from Apache software foundation. :cite :*www-ApacheTinkerPop* 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. *[104]* 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. *[105]* 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.

63. 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. *[106]* 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.
64. Dream:Lab
    DREAM:Lab stands for "Distributed Research on Emerging Applications and Machines Lab." *[107]* DREAM:Lab is centered around distributed systems research to enable expeditious utilization of distributed data and computing systems. *[107]* DREAM:Lab utilizes the "capabilities of hundereds of personal computers" to allow access to supercomputing resources to average individuals. *[108]* The DREAM:Lab pursues this goal by utilizing distributed computing. *[108]* Distributed computing consists of independent computing resources that communicate with each other over a network. *[109]* 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. *[109]*
65. 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. *[110]* It used for data management, visualizing data (e.g. pie-charts, bar-charts, lineplot, scatterplot, timelines) *[111]* , 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". *[112]* 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.
66. CINET
    A representation of connected entities such as "physical, biological and social phenomena"[www-bi.vt.edu] predictive model. Network science has grown its importance understanding these phenomena Cyberinfrastructure is middleware tool helps study Network science, [www-portal.futuresystems.org/projects/233] "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 cant be achieve by desktop. Cyberinfrastructure provides cloud based infrastructure (e.g. FutureGrid) as well as use of HPC (e.g. Shadowfax, Pecos). With use of advance intelligent Job mangers, it select 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 analysist.

67. NWB
[113] NWB stands for Network workbench is analysis, modelling and visualization toolkit for the network scientists. It provides an environment which help 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 helps 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.
68. Elasticsearch
Elasticsearch *[114]* 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. *[115]* 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 *[116]* 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". *[117]* 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. *[118]* A log parsing engine "Logstash" and analytics and visualization platform "Kibana" are also developed alongside Elasticsearch forming a single package.
69. Kibana
Kibana is an open source data visualization plugin for Elasticsearch. *[119]* 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. *[120]* 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. *[121]* Elasticsearch

is a search engine based on Lucene. *[122]* 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. *[123] [124]*

70. 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. *[125]* 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.
71. 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. *[126] [127]* Graylog allows us to centrally collect and manage log messages of an organization's complete infrastructure. A user can perform search on terrabytes 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. *[128]* 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.
72. 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 *[129]*. 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 *[129]*.
73. Tableau

[130] 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. *[131]* 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.

74. 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 *[132]*.
    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.
75. three.js
76. Potree
    Potree *[133]* 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. It efficiency can be improved by using techiques such as divide and conquer as disscused in a conference paper Taming the beast: Free and Open Source massive cloud point cloud web visualization *[134]*. It has also been widely used in works involving spatio-temporal data where the changes in geographical features are across time *[135]*.
77. DC.js
    According to *[136]*: "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 anlysis 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.
78. TensorFlow
    TensorFlow is a platform that provides a software library for expressing and executing machine learning algorithms. *[137]* states TensorFlow has a flexible architecture

allowing it to be executed with minimal change to many hetegeneous 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 netowrk models, and it has been used for conducting research and for deploying machine learning systems into production across more than a dozen areas". *[138]* 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.

79. 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 *[139]* . It can be included as a library in your Python or C++ pro grams, or used as a standalone machine learning tool through its own model description language (BrainScript). *[140]* Latest Version:2017-02-10. V 2.0 Beta 11 Release

5.2.3 Application Hosting Frameworks

80. 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. *[141]*
    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. *[142]*
81. 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. *[143]* 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. [www-apscale-deployment]

82. Red Hat OpenShift
    [www-paas] OpenShift was launched as a PaaS (Platform as a Service) by Red Hat in the Red Hat Summit, 2011. *[144]* 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. Thus, *[145]* OpenShift 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-blog] OpenShift is powered by Origin, which is in turn built using Docker container packaging and Kubernetes container cluster. Due to this, OpenShift offers a lot of options, including online, on-premise and open source project options.
83. Heroku
    Heroku *[146]* 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 *[147]*, 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.
84. Aerobatic
    According to *[148]*: 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.
85. AWS Elastic Beanstalk
    [www-amazon elastic beanstalk] AWS Elastic Beanstalk is an orchestration service

offered from Amazon Web Services which provides user with a platform for easy and quiclk deployment of their WebApps and services. [amazon elastic beanstalk-book] Amazon Elastic BeanStack automatically handles the deployement details of capacity provisioning by Amazon Cloud Watch, Elastic Load Balancing, Auto-scaling, and application health monitoring of the WebApps and service. 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 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 different container types for each language such as Apache Tomcat for Java applications, Apache HTTP Server for PHP applications Docker, GO and many more for specific languanges where the container defines the infrastructure and softwarestack to be used for a given enviorment.:cite:*www-amazon elastic beanstalk* "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."

86. 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." *[149]*

    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 *[150]*. For instance, Azure users can access predictive analytics, Bots and Blockchain-as-a-Service *[150]* as well as more-basic computing, networking, storage, database and management components *[151]*. 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*, *[152]* even though IBM (*SoftLayer [153]* and *Bluemix [154]*) and Google (*Google Cloud Platform*) *[155]* offer IaaS to the market. As of January 2017, Azure's datacenters span 32 Microsoft-defined *regions*, or 38 *declared regions*, throughout the world. *[150]*

87. Cloud Foundry

    It is an open source software with multi cloud application .It is a platform for running applications and services. *[156]* 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). 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 *[157]* , 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.

88. Pivotal
    Pivotal Software, Inc. (Pivotal) is a software and services company. It offeres 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 *[158]*.
89. IBM BlueMix
90. (Ninefold)
    The Australian based cloud computing platform has shut down their services since January 30, 2016. Refer *[159]*
91. 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. *[160]* 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. *[161]*
92. 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). *[162]* 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. *[163]* This capability user has to not have to make choices of flexibility over security of sensitive data when choosing a cloud computing platform.
93. appfog
    According to *[164]*, "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. *[165]* PaaS provides an environment for the creation of software. *[165]* The underlying support infrastructure that AppFog provides includes things such as runtime, middleware,

o/s, virtualization, servers, storage, and networking. *[166]* AppFog is based on VMWare's CloudFoundry project. *[164]* It gets things such as MySQL, Mongo, Reddis, memCache, etc. running and then manages them. *[167]*

94. CloudBees
    [168] Cloudbees provides Platform as a Service (PaaS) solution, which is a cloud service for Java applications. 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. *[169]* The platform ships with verified security and enhancements assuring less risk for sharing sensitive information. It simplies the task of getting the platform accessed by every user using the feature "Jenkins Sprawl".
95. Engine Yard *[170]*
    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.
96. (CloudControl)
    No Longer active as of Feb. 2016 *[171]*
97. dotCloud *[172]*
    dotCloud services were shutdown on February 29,2016.
98. Dokku
99. OSGi
100. 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.:cite:*hubzerowebsite* 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.:cite:*hubzeropaper2010*

101. 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. *[173]*
102. Agave
Agave is an open source, application hosting framework and provides a platform-as-a-service solution for hybrid computing. *[174]* 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. *[175]*
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. *[174]* Agave shall read the JSON file, formalize the parameters, execute the user program and dump the output to the requested destination.
103. 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 *[176]*. 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 *[177]*.

5.2.4 High level Programming

104. 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 *[178]*. 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 relavant examples/syntax and also tries to spot errors in the programs *[179]*.
105. 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 August2008. 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 [hive]
106. HCatalog
107. Tajo
Apache Tajo *[180]* 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. *[181]* 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 anaysis 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,RCFile, 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. *[181]* 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.

108. Shark

Data Scientists when working on huge data sets try to extract meaning and interpret the data to enhance insight about the various patterns, oppurtunities and possiblities that the dataset has to offer. *[182]* At a traditional EDW(Enterprrise Data Warehouse) a simple data manipulation can be perfpormed using SQL queries but we have to rely on other systems to apply the machine learning on thoese data.Apache Shark is a distributed query engine developed by the open source community whoese goal is to provide a a unified system for easy data manipulation using SQL and pushing sophisticated analysis towards the data.

[182] Shark is a data Warehouse system built on top of Apache Spark which does the parallel data execution and is capable of deep data analysis using the Resilient Distributed Datasets(RDD) memory abstraction which unifies the SQL query processing engine with analytical algorithms based on this common abstraction allowing the two to run 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 thoushands o nodes in a fault-toleranat manner

[182] "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". Shark can run both on the StandAlone Mode and Cluster Mode. *[182]* Shark can answer the queries 40X faster than Apache Hive and can machine learning programs 25X faster than MapReduce programmes. in Apache hadoop on large data sets.Thus, this new data analysis system performs query processing and complex analytics(iterative Machine learning) at scale and efficiently recovers form the failures midway

109. 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 accessiblity 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. *[183]* Although Salesforce.com initially *open-sourced* it via Github, by May of 2014 it had become a top-level Apache project. *[184]*

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." *[185]* 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. *[186]*

Despite adding an abstraction layer, Phoenix can actually speed up queries because it optimizes the query during the translation process. *[183]* For example, "Phoenix

beats Hive for a simple query spanning 10M-100M rows." *[187]*
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. *[188]*

110. Impala
111. MRQL
MapReduce Query Language (MRQL, pronounced miracle) "is a query processing and optimization system for large-scale, distributed data analysis" *[189]*. 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 *[189]*.
MRQL was created in 2011 by Leaonids Fegaras *[190]* 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 *[191]*.
112. SAP HANA
As noted in *[192]*, 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 *[193]*. It provides extensive administrative, security features and data access that ensures high data availability, data protection and data quality.
113. 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. *[194]*
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.
114. PolyBase
115. Pivotal HD/Hawq
Pivotal HDB is the Apache Hadoop native SQL database powered by Apache HAWQ *[195]* 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, roboust ANSI SQL compliance and integrated Apache MADlib for machine learning *[196]*.

116. Presto
Presto *[197]* is an open-source distributed SQL query engine that supports interactive analytics on large datasets. It allows interfacing with a variety of data sources such as Hive, Cassandra, RDBMSs and proprietary data source. Presto is used at a number of big-data companies such as Facebook, Airbnb and Dropbox. Presto's performance compares favorably to similar systems such as Hive and Stinger *[198]*.
117. 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. *[199]* 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 use for a variety of jobs including analyzing web-crawled documents, detecting e-mail spam, working through application crash reports.
118. Google BigQuery
Google BigQuery *[200]* is an enterprise data warehouse used for large scale data analytics. *[201]* 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 Googe BigQuery a user can control access to both the project and the data based on the his business needs which gives the ability to others to view and even query the data. *[200]* 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 pr python. BigQuery can also be accessed using a variety of third party tool. 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 *[201]* 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 *[202]* Capacitor columnar data format, and offers the standard database concepts of tables, partitions, columns, and rows.
119. Amazon Redshift
Amazon Redshift is a fully managed, petabyte-scale data werehouse service in the cloud. Redshift service manages all of the workof setting up, operating and scalling

a data werehouse. 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 *[203]*. Redshift is built on thet top of technology from the Massive Paraller Processing (MPP) data-werehouse company ParAccel which based on PostgresSQL 8.0.2 to PostgresSQL 9.x with capabilty to handle analytics workloads on large- scale dataset stored by a column-oriented DBMS principle *[204]*.

120. Drill
Apache Drill *[205]* 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
121. Kyoto Cabinet
Kyoto Cabinet as specified in *[206]* 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).
122. Pig
123. 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. *[207]*
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. *[208]*
Addtionally, 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". *[207]*

Important components of the Sawzall language include: *szl*, the binary containing the code compiler and byte-code interpreter that executes the program; the *libszl* 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. *[209]*

124. Google Cloud DataFlow

Google Cloud DataFlow *[210]* 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 *[210]*, 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 *[211]* the performance of MapReduce started deteriorating while facing multiple petabytes of data whereas Google Cloud Dataflow is apparently better at handling enormous datasets. *[210]* 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.

125. Summingbird

According to :cite:'summingbirdgit', "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 :cite:'boykin2014summingbird'. It combines online and batch MapReduce computations into one framework :cite:'boykin2014summingbird'. It utilizes the platforms Hadoop for batch and Storm for online process execution :cite:'boykin2014summingbird'. The open-source Hadoop implementation of MapReduce is a tool which those responsible for data management use to handle problems related to big data :cite:'boykin2014summingbird'. Summingbird uses an algebraic structure called a commutative semigroup to perform aggregations of both batch and online processes :cite:'boykin2014summingbird'. A commutative semigroup is a particular type of semigroup "where the associated binary operation is also commutative" :cite:'boykin2014summingbird'. The types of data that Summingbird

takes as inputs are streams and snapshots :cite:'boykin2014summingbird'. The types of data Summingbird jobs generate are called stores and sinks :cite:'boykin2014summingbird'. Stores are "an abstract model of a key-value store" while sinks are unaggregated tuples from a producer :cite:'boykin2014summingbird'. 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 :cite:'boykin2014summingbird'.

126. 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) *[212]*. Lumberyard's core engine technology is based on Crytek's CryEngine *[213]*. The goal is "creating experiences that embrace the notion of a player, broadcaster, and viewer all joining together"*[212]*. 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 *[214]*.

5.2.5 Streams

127. Storm
Apache Storm is an open source distributed computing framework for analyzing big data in real time. *[215]* 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. *[216]*

128. S4
S4 *[217]* 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. *[218]* 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. *[217]* 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. *[219]* In S4, both the platform and the

applications are built by dependency injection, and configured through independent modules.

129. 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. *[220]* 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). *[221]* 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 *[222] [220]*.

130. Granules
Granules in 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 *[223]*.

131. Neptune

132. 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 *[224]*.

133. Amazon Kinesis
Kinesis is Amazon's *[225]* 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 is 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 used to understand the amount of data being processed by Kinesis. User can use this 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 *[226]* in the area of real-time processing provides following key benefits: ease of use, parellel processing, scalable, cost effective, fault tolerant and highly available.
134. LinkedIn
135. 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 *[227]* 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 *[228]*)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.
136. Databus
137. Facebook Puma/Ptail/Scribe/ODS
The real time data Processing at Fcabook is carried out using the technologies like Scibe,PTail, Puma and ODS. While designing the system, facebook primarily focused on the five key decissions that the system should incorporate and that included Ease of Use, Performance , Fault-tolerance , Scalability and

Correctness.:cite:'www-facebook' "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". :cite:'www-facebook'Fcabook 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 made the system more fault toleranat and scalable. :cite:'facebook-paper-2017' The large infrastructure of facebook comprises of hundreds of systems distributed across multiple data centers that needs a continious monitoring to track their health and performance.Which is done by Operational Data Store(ODS).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 collect, aggregate 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 avalible for both batch and real-time processing. Ptail is an internal tool built to aggregate data from multiple Scribe stores and 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 i fairly real time

138. 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 usings real-time event processing engine. *[229]* Jobs are authored by "specifying the input source of the streaming data, the output sink for the results of your job, and a data tranformation 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. *[230]* Available documentation to get started with Azure Stream Analytics. *[231]* Azure Stream Analytics has a development project available on github.

139. Floe
140. Spark Streaming
141. Flink Streaming
142. DataTurbine

Data Turbine *[232]* 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.
[233] 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.

5.2.6 Basic Programming model and runtime, SPMD, MapReduce

143. Hadoop
144. Spark *[57]*
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.
145. 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 *[234]*. 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 archite cture 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. *[235] [236] [237]*.
146. MR-MPI
[238] 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.

147. 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 *[239]*.

148. Reef
REEF (Retainable Evaluator Execution Framework) *[240]* 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.

149. 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." *[241]* 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." *[242]* 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." *[243]* 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. *[244]*
b. DISCO from the research group Service Engineering (SE), *[245]* 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. *[246]*

150. 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 *[247]*. 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 *[248]*.

151. Giraph
    Apache Giraph is an iterative graph processing system built for big data *[249]*.It utilizes Hadoop Mapreduce technology for processing graphs *[250]* Giraph was initially developed by Yahoo based on the paper published by Google on Pregel. *[251]* 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). *[252]*
152. Pregel
153. Pegasus
    See #4 above.
154. 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 in that 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.
    :cite:'ligra-paper-2013 '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
    :cite:'ligra-paper-2' 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 is large enough for any of the graphs reported in the papers mentioned above.
155. GraphChi
156. Galois
    Galois system was built by intelligent software systems team at University of Texas, Austin. As explained in *[253]*, "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 *[254]*. Galois currently supports C++ and Java.
157. 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 *[255]* 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 *[255]* 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. *[256]* Medusa has proved to be a powerful framework for networked digital audio and video framework. *[256]* By exploiting the APIs it takes a modular approach to construct complex graph systems.

158. MapGraph
159. 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. *[257]*

5.2.7 Inter process communication Collectives

160. point-to-point
161. (a) publish-subscribe: MPI
see `http://www.slideshare.net/Foxsden/high-performance-processing-of-streaming-data`

161. (a) publish-subscribe: Big Data
Publish/Subscribe (Pub/Sub) *[258]* 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) *[259]*.
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 *[260]*. 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.

2. HPX-5
   Based on *[261]*, 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 *[262]*.
3. Argo BEAST HPX-5 BEAST PULSAR
   Search on the internet was not successsful.
4. Harp
   Harp *[263]* 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.
5. Netty
   Netty *[264]* "is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients". Netty *[265]* "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.
6. ZeroMQ
   In *[266]*, 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 *[267]*, 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 *[266]*, 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.
7. ActiveMQ

8. RabbitMQ
   RabbitMQ is a message broker *[268]* 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:
   (a) Simple messaging: producers write messages to the queue and consumers read messages from the the queue. This is synonymous to a simple message queue.
   (b) Producer-consumer: Producers produce messages and consumers receive messages from the queue. The messages are delivered to multiple consumers in round robin manner.
   (c) 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.
   (d) Routing: In this mode consumers can subscribe to a subset of messages instead of receiving all messages from the queue.
   (e) 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.
   (f) 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 *[269]* (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"
9. NaradaBrokering
   NaradaBrokering *[270]*, 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 *[271]*. 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 *[272]*.

10. QPid
11. 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. :cite:'www-kafka'
12. 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 sppark 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 *[273]*.
13. 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. *[274]*
    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. *[275]*

14. AMQP
    [276] 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 timly 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.
15. Stomp
16. MQTT
    According to *[277]*, 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 *[278]*, which monitors river and floodplains through a set of sensors.
17. Marionette Collective
18. 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 *[279]*, 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 *[280]* 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 corressponding 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 *[281]*, 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.

19. Lambda
AWS Lambda is a product from amazon which facilitates serverless computing *[282]*.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 *[283]*.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.
20. Google Pub Sub
[284] Google Pub/Sub provides an asynchronous messaging facility which assists the communication between independent applications. 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. *[285]* 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.
21. Azure Queues
Azure Queues storage is a Microsoft Azure service, providing inter -process communication by message passing *[286]*. 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 *[287]*. 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 *[288]*. 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.

22. 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 ingress massive amounts of telemetry into the cloud and read the data from multiple applications using publish-subscribe semantics. *[289]* 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.

5.2.8 In-memory databases/caches

183. Gora (general object from NoSQL)
Gora is a in-memory data model *[290]* which also provides persistence to the big data. Gora provides persistence to different types of data stores. Primary goals of Gora are:
    (a) data persistence
    (b) indexing
    (c) data access
    (d) analysis
    (e) 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:
    (a) define the bean used as model for persistence
    (b) use gora compiler to compile the bean
    (c) create a mapping file to map bean definition to datastore
    (d) update gora.properties to specify the datastore to use
    (e) 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.
184. Memcached
Memcached is a free and open-source, high performance, distributed memory object caching system. *[291]* 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.

185. Redis
Redis (Remote Dictionary Server) is an open source ,in-memory, key-value database which is commonly referred as a data structure server. :cite:'redis-book-2011' "It is called a data structure server and not simply a key-value store because Redis implements datastructure which allows keys to contain binary safe strings ,hashes,sets and sortedsets, as well as lists" .Redis's exceptional performance, simplicity to use and implement, and atomic manipulation of data structures lends itself to solving problems that are difficult or perform poorly when implemented with traditional relational databases. :cite:'redis-book-2016' "Salivator Sanfilippo(Creator of open-sorce 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 intyractable problems."
:cite:'redis-book-2016' A very popular use pattern for Redis is an in-memory cache for web-applications. The second popular use pattern for REDIS is for metric storage of such quantitative data such as web page usage and user behaviour on gamer leaderboards where using a bit operations on strings, Redis very efficently stores binary information on a particular characteristics.The third popular Redis use pattern is a communication layer between different systems through a publish/subscribe(pub/sub for short), where one can post 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 message. The Comapnies 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.

186. LMDB (key value)
LMDB (Lighting memory-mapped Database) is a high performance embedded transactional database in form of a key-value store *[292]*. 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 *[293]* 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 *[294]*.

187. Hazelcast
Hazelcast is a java based, in memory data grid *[295]*. It is open source software, released under the Apache 2.0 License *[296]*. 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 *[295]*.

188. 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 *[297]*. It offers storage tires such that less frequently data could be moved to slower tires *[298]*. 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.
189. Infinispan
190. 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) *[299]*. 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 [voltdb-wiki].
191. H-Store
H-Store is an in memory and parallel database management system for on-line transaction processing (OLTP). Specifically , *[300]* 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 *[301]* "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 *[302]* 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.

5.2.9 Object-relational mapping

192. 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 [hibernate]
193. OpenJPA
According to *[303]*, 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. *[304]* 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.
194. 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. *[305]*. 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.
195. 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 *[306]* 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 *[307]*.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 *[308]*.

196. ODBC/JDBC
Open Database Connectivity (ODBC) is an open standard application programming interface (API) for accessing database management systems (DBMS) *[309]*. 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 *[310]*. 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.

5.2.10 Extraction Tools

197. 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 , *[311]* 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 *[312]* 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.

381. 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. *[313]*"

5.2.11 SQL(NewSQL)

198. Oracle
Oracle database is an object-relational database management system by Oracle. Following are some of the key features of Oracle *[314]* 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 *[314]* Oracle Database 12c Release 1: 12.1 (Patch set as of June 2013)
199. DB2
200. SQL Server
SQL Server *[315]* is a relational database management system from Microsoft. As of Jan 2017, SQL Server is available in below editions
    (a) Standard - consists of core database engine
    (b) Web - low cost edition for web hosting
    (c) Business Intelligence - includes standard edition and business intelligence tools like PowerPivot, PowerBI, Master Data Services
    (d) Enterprise - consists of core database engine and enterprise services like cluster manager
    (e) SQL Server Azure - *[316]* core database engine integrated with Microsoft Azure cloud platform and available in platform-as-a-service mode.

    In the book *[317]*, the technical architecture of SQL Server in OLTP(online transaction processing), hybrid cloud and business intelligence modes is explained in detail.
201. SQLite
SQLite is a severless SQL database engine whose source code resides in the public domain :cite:'sqliteabout'. SQLite databases, including tables, indices, and views, reside on a single file on the disk :cite:'sqliteabout'. It has a compact library, often taking up less than KiB of space, depending on the particular configuration :cite:'sqliteabout'. Performance is the tradeoff with the smaller size, i.e. performance usually runs faster when given more memory :cite:'sqliteabout'. SQLite transactions comply with the ACID (Atomicity, Consistency, Isolation, Durability) :cite:'acid' properties :cite:'sqliteabout'. SQLite does not require administration or configuration :cite:'sqliteover'. 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) :cite:'sqliteover' SQLite does not compare directly with clien/server databases such as MySQL as they are both trying to solve different problems :cite:'sqlitewhentouse'. 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 :cite:'sqlitewhentouse' 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 :cite:'sqlitewhentouse'.

202. MySQL
MySQL is a relational database management system. *[318]* SQL is an acronym for Structured Query Language and is a standardized language used to interact with the databases. *[318]* Databases provide structure to a collection of data while. *[318]* A database management system allows for the addition, accessing, and processing of the data stored in a database. *[318]* 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. *[319]*
203. PostgreSQL
204. 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. *[320]* 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.
205. Galera Cluster
Galera cluster *[321]* 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).
206. SciDB
207. Rasdaman
Rasdaman is an specialized database management system which adds capabilities for storage and retrival of massive multi-dimensional array, such as sensors,image, and statistics data. *[322]* 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.
[323]: 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.
208. Apache Derby

[324]: 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.
[325]: 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 deployement 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). *[326]*:

209. Pivotal Greenplum
210. 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 *[327]*. Cloud SQL database aims at developers to focus on app development leaving database adminstitation to a minimum. This can be understood as 'My SQL 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. *[328]*
211. Azure SQL
212. 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 *[329]* 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 *[330]*) 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.
213. 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 *[331]*.
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 *[332]* . 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.

214. 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 *[333]* .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. *[334]*
215. N1QL
216. BlinkDB
217. 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 *[335]*. 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 *[336]*. 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.

5.2.12 NoSQL

218. Lucene
Apache Lucene *[337]* 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.

219. Solr
220. 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 disributed using Cassandra's data model. *[338]*
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. *[339]*
221. Voldemort
According to *[340]*, 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 *[341]*, the Voldemart'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 Voldermort 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.
222. Riak
Riak is a set of scalable distributed NoSQL databases developed by Basho Technologies. Riak KV is a key-value *[342]* 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 *[343]*. Riak S2 is designed to store large objects like media files and software binaries *[344]*. 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.
223. ZHT
According to *[345]*, "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. *[346]* To retrieve a value

in a distributed hash table, one needs to find the node that is responsible for the managing the key value pair of interest. *[346]* 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. *[346]* In a ZHT, however, every node contains information concerning the location of every other node. *[347]* Through this approach, ZHT aims to provide "high availability, good fault tolerance, high throughput, and low latencies, at extreme scales of millions of nodes." *[347]* 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. *[347]*

224. Berkeley DB
Berkeley DB is a family of open source, NoSQL key-value database libraries. *[348]* 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. *[349]* 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.
[350] 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.
225. Kyoto/Tokyo Cabinet
Tokyo Cabinet *[351]* and Kyoto Cabinet *[352]* 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). *[351]* 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 retrive a record using the key. Both have smaller size of database file, faster processing speed and provide effective backup procedures.
226. Tycoon
Tycoon/ Kyoto Tycoon *[353]* is a lightweight database server developed by FLL labs and is a distributed Key-value store *[354]*. 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 *[355]* 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.

227. 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 *[356]* explains the results of performance experiments between Tyrant and Memcached + MySQL.
Tyrant was written and maintained by FAL Labs *[357]*. However, according to FAL Labs, their latest product *[358]* Kyoto Tycoon is more powerful and convenient server than Tokyo Tyrant.

228. 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.

229. Espresso
Espresso *[359]* 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 heirarchical data model in which the databases and table schema are defined in JSON.Some of the key component of Espresso include : 1)Router: which is a stateless HTTP Proxy and also acts as a 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 partition. 3) Helix: is responsible for cluster management in Espresso. 4) Databus: are responsible for capturing change 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.

230. 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, *[360]* 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. *[361]* 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. *[362]* CouchDB serves as the basis of Couchbase, Inc's Couchbase Server.

231. Couchbase Server
Couchbase, Inc. offers Couchbase Server (CBS) to the marketplace as a NoSQL, document-oriented database alternative to traditional relationship- oriented database managgement 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. *[363]*
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". *[364]*
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++" *[365]*, It runs on an Erlang virtual machine known as BEAM. *[366]*
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. *[367]*
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. *[368]*
Finally, CB uses the Apache 2.0 License, one of several open-source license

alternatives. *[369]*

232. 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 *[370]*.

233. Pivotal Gemfire *[371]*
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 remaining untouched. Additionally, the support for multi-site configurations enable the user to scale horizontally between different distributed systems spread over a wide geographical network.

234. HBase
Apache Hbase is a distributed column-oriented database which is built on top of HDFS (Hadoop Distributed File System).According to *[372]*, It is a 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 identied 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.

235. Google Bigtable
Google Bigtable is a NoSQL database service, built upon several Google technologies, including Google File System, Chubby Lock Service, and SSTable *[373]*. Designed for Big Data, Bigtable provides high performance and low latency and scales to hundreds of petabytes *[373]*. 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 *[374] [375]*. Since May 6, 2015, Bigtable has been available to the public as Cloud Bigtable *[375]*.

236. LevelDB

LevelDB is a light-weight, single-purpose library for persistence with bindings to many platforms. *[376]* 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.

237. Megastore and Spanner
Spanner *[377]* 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 *[378]* 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.

238. 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 *[379]*.

239. Cassandra
Apache Cassandra *[380]* is an open-source distributed database managemment for handling large volume of data accross comodity 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.
240. RYA
Rya is a "scalable system for storing and retrieving RDF data in a cluster of nodes." *[381]* RDF stands for Resource Description Framework. *[381]* RDF is a model that facilitates the exchange of data on a network. *[382]* RDF utilizes a form commonly referred to as a triple, an object that consists of a subject, predicate, and object. *[381]* These triples are used to describe resources on the Internet. *[381]* Through new storage and querying techniques, Rya aims to make accessing RDF data fast and easy. *[383]*
241. Sqrrl
242. Neo4J
Neo4J *[384]* 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. *[385]* 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. *[386]* The Neo4J is often used for clustering. It offers two feature clustering solutions: Causal Clustering and Highly available clustering. *[387]* Casual clustering focuses on safety, scalability and causal consistency in the graph. *[388]* The highly available cluster places importance to fault tolerance as each instance in the cluster has full copies of data in their local database.
243. graphdb
A Graph Database is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data. *[389]* 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.
244. 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 *[390]* 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 *[391]* 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 *[392]* 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" *[393]* outlines the implementation of graph mining algorithms using SPARQL.

245. 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." *[394]* 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) as `http://example.name#Professor12 http://xmlns.com/foaf/0.1/teacheshttp`: //example.name#students. Using this representation, semantic data can be queried. *[395]*

246. 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.

247. 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. :cite:’www-tao’.
248. Titan:db
Titan:db *[396]* 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 *[397]*. 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.
249. Jena
Jena is an open source Java Framework provided by Apache for semantic web applications. (*[398]*) 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. *[399]* ‘ 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 *[399]* 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.
250. Sesame
Sesame is framework which can be used for the analysis of RDF (Resource Description Framework) data. Resource Description Framework (RDF) *[400]* 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. *[401]* Sesame has now officially been integrated into RDF4J Eclipse project. Sesame takes in the natively written code as the input and then performs a series of transformations, generating kernels for various platforms. *[402]* In order to achieve this, it makes use of the feature identifier, impact predictor, source-to-source translator and the auto-tuner. 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.

251. Public Cloud: Azure Table
Microsoft offers its NoSQL Azure Table product to the market as a low-cost, fast and scalable data storage option. *[403]* 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. *[404]*
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*. *[405]*
Azure Table is considered best-suited for infrequently accessed data storage.
252. Amazon Dynamo
Amazon explains DynamoDB as :cite:'www.dyndb' 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

'

253. Google DataStore

5.2.13 File management

254. 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 *[406]*. 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. *[407]*

255. NetCDF
NetCDF is a set of software libraries and self-describing, machine-indepen dent 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) Commun ity Programs (UCP). Unidata is funded primarily by the National Science F oundation *[408] [409]* . The purpose of the Netwo rk Common Data Form(netCDF) interface is to support the creation, efficient access, and sharing of data in a form that is self-describing, portable, co mpact, 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 *[408]*. NetCDF consists of: a) A conceptual data model b) A set of binary data formats c) A set of APIs for C/Fortran/Java
256. CDF
Common Data Format *[410]* 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 *[411]*.
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 *[412]*.
Compared to HDF format *[413]*, 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).

257. HDF
258. OPeNDAP
259. FITS

    FITS stand for 'Flexible Image Trasnport System'. It is a standard data format used in astronomy. FITS data format is endorsed by NASA and International Astronomical Union. According to *[414]*, 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 *[415]* newer version of FITS standard document was released in July 2016. FITS can be used for digitization of contents like books and magzines. Vatican Library *[416]* used FITS for long term preservation of their book, manuscripts and other collection. Matlab, a language used for technical computing supports fits *[417]*. The 2011 paper *[418]* explains how to perform processing of astronomical images on Hadoop using FITS.
260. RCFile

    RCFile (Record Columnar File) *[419]* 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 [www-rcfile]. According to *[420]*, 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!
261. 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 in to 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 *[421]*.
262. 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 *[422]*. 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.

5.2.14 Data Transport

263. 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 swarning and tracking. *[423]* It divides the files in large number of chunck and as soon as file is received it can be server 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 swarn. Bittorrent trackers keeps list of files available for transfer and helps the swarn 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 μTorrent, qBittorrent.
264. HTTP
265. FTP
According to *[424]* 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 *[425]* 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.
266. SSH
SSH is a cryptographic network protocol *[426]* 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 *[427]*.

267. 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 *[428]* 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 .
268. Flume
Flume is distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of log data [apche-flume]. 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 *[429]*.
269. Sqoop
Apache Sqoop is a tool to transfer large amounts of data between Apache Hadoop and sql databases *[430]*. 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 *[431]*.
270. Pivotal GPLOAD/GPFDIST
Greenplum Database *[432]* 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 *[433]* 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.
gpload *[432]* is data loading utility is used to load data into Greenplum's external

table in parallel.
Google has an invention *[434]* relating to integrating map-reduce processing techniques into a distributed relational database. An embodiment of the invention is implemented by Greenplum as gpfdist.

5.2.15 Cluster Resource Management

271. Mesos
Apache Mesos *[435]* 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 *[436]*, termed Dominant Resource Fairness (DRF) *[437]* 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" *[438]*
272. Yarn
Yarn (Yet Another Resource Negotiator) is Apache Hadoop's cluster management project *[439]* . 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. Resource manager 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) *[440]*. 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. *[441]*
273. Helix
Helix is a data management system getting developed by IBM which helps the users to do explitory 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 dispite 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. *[442]*

274. 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 *[443]*. 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 *[443]*.
275. Google Omega
276. Facebook Corona
Corona is a new scheduling framework developed by facebook which seperates 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 it's limitations with the scalibility as when the number of jobs at facebook grew in the next few years.:cite:*www-facebook corona* Another limitation of Hadoop was it was a pull-based scheduling model as the task trackerhave to provide a heartbeat to the job tracker to indicate that it is running which associates with a pre-defined delay,that was problematic for small jobs. 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 configuration – so slots are wasted anytime the cluster workload does not fit the static configuration.
:cite:'www-facebook corona'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. 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 it uses a push-based scheduling whoese 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 decsions.Corona is used as an integeral part of the Facebook's data infrastructure and is helping power big data analytics for teams across the company.
277. 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"*[444]*.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"*[445]*

278. 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"*[446]*
279. SGE
According to *[447]*, 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 delpoyment phasae for IBM's 64-bit operating system z/OS. Standard Grid cluster comprises of one master host and many execution hosts. There is a 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).
280. 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 *[448]*. 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 *[449]*.

281. 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).:cite:*www-moab*
282. Slurm *[450]*
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, *[451]* 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 mangers 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 *[452]*.
283. Torque
284. Globus Tools
[453] 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. *[454]* 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 *[455]* Since 2000, companies like Fujitsu, IBM, NEC and Oracle have pursued Grid strategies based on the Globus Toolkit.

285. 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 *[456]* Falkon lightweight framework and *[457]* 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.
[458] 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.

5.2.16 File systems

286. 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:cite:*www-hdfs*.
287. Swift
288. 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. [www-project-haystack.org]
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.
289. 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 *[459]*.

290. Cinder
"Cinder is a block storage service for Openstack" *[460]*. 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 *[461]*. 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 *[460]*.

291. 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 *[462]*. 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) *[463]*, 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) *[464]*.
The Ceph Filesystem *[465]* 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.

292. FUSE
FUSE (Filesystem in Userspace) *[466]* "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 *[467]* 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 *[468]*. 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.

293. Gluster
294. Lustre
The Lustre file system *[469]* 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.
295. GPFS
IBM General Parallel File System (GPFS) was rebranded to IBM Spectrum Scale on February 17, 2015 *[470]*. See 380.

380. IBM Spectrum Scale
General Parallel File System (GPFS) was rebranded as IBM Spectrum Scale on February 17, 2015 *[470]*.
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” *[470]* 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 *[470]*. 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 *[471]*.

296. GFFS
The Global Federated File System (GFFS) *[472]* 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.

297. Public Cloud: Amazon S3
Amazon Simple Storage Service (Amazon S3) *[473]* 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 auotmatic 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. *[474]* 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 and object which might be an file or folder, and (4) Perform operations on the object which is stored in the cloud.

298. 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 *[475]* 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 offe- red by Windows Azure namely block, append and page blobs. *[476]* 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. *[475]*

299. Google Cloud Storage
Google Cloud Storage is the cloud enabled storage offered by Google. *[477]* 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 contains 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.

5.2.17 Interoperability

300. 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. *[478]* 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. *[479]* It is accessible by many operating systems such as Linux, FreeBSD, Mac OS, and Windows OS.
301. 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.
302. JClouds
[480] 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 *[481]* 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:

(a) 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

(b) 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.

(c) Provider: a particular cloud vendor is a provider. Jcloud uses provider information to initialize its context.

(d) 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.

303. TOSCA

304. 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" *[482]*. 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 require 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. *[483]*

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" *[484]*.
OCCI RESTful HTTP Protocol describes communications between server and client on OCCI platform via HTTP protocol *[485]*. 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 *[486]* 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 depricated and will be removed from the next major version *[487]*.

305. CDMI
The Storage Networking Industry Association (SNIA) *[488]* 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 to for providers and users of cloud is Cloud Data Management Interface (CDMI). According latest issue of CDMI *[489]*, "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 *[490]*.
306. Whirr
307. 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. *[491]* 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." *[492]*

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. *[491]*

308. Genesis

5.2.18 DevOps

309. 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 includig the stack it runs on, dependencies it is associated with and everything the application requirs to run within an isolated enviorment . 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. :cite:'docker-book' Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands. 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 ,:cite:'www-docker"'you can scale out your application as if it were running on a single, huge computer" as swarm can be scaled upto 1000 Nodes or upto 50,000 containers
310. Puppet
Puppet is an open source software configuration management tool *[493]*.This aims at automatic configuration of the software applications and infrastructure. This configuration is done using the easy to use languge. Puppet works on major linux distributions and also on microsoft windows , it is also cross-platform application making it easy to manage and portable. *[494]*
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 rund on master server. The connection between 'Puppet Master' and 'Puppet agent' will be established using thesecured SSL connection. The configiration at client will be validated as per the set up in Puppet master at a predefined interval. If configration at client is not matching with the master puppet agent fetches the equired changes from master. *[495]*

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. *[493]*

311. 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 *[496]* 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 *[497]*
312. Ansible
Ansible is an IT automation tool that automates cloud provisioning, configuration management, and application deployment. *[498]* 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. *[499]*
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.
313. SaltStack
314. Boto
[500] The latest version of Boto is Boto3. *[501]* Boto3 is the Amazon Web Services (AWS) :cite:Software Development Kit (SDK) for Python. It enables the :cite:Python developers to make use of services like Amazon S3 :cite:and Amazon EC2. *www-boto3-documentation* It provides :cite:object oriented APIs along with low-level direct service :cite:access. It provides simple in-built functions and :cite:interfaces to work with Amazon S3 and EC2.
[502] Boto3 has two distinct levels of APIs - client and resource. 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.
315. 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 *[503]*.It is a nifty piece of code that assemble s all the usual setup bits required for a large network installation like TFTP, DNS, PXE installation trees. and automates the process[1].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.

316. Xcat
xCAT is defined as extreme cloud/cluster administration toolkit. Tnd his 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. *[504]* xCAT works with various cluster types such as high performance computing, horizontal scaling web farms, administrative, and operating systems. *[505]*
317. 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 *[506]* 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 *[507]*. Models can include steps for handoff to a DevOps system or to any other system capable of controlling the node.
318. CloudMesh
319. Juju
Juju (formerly Ensemble) *[508]* 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 *[509]* 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.
320. Foreman
321. OpenStack Heat
Openstack Heat, a template deployment service was the project launched by Openstack, a cloud operating system similar to AWS Cloud Formation. *[510]* 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" *[511]*

Once the execution enviroment 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.

322. Sahara
The Sahara product provides users with the capability to provision data processing frameworks (such as Hadoop, Spark and Storm) on OpenStack *[512]* by specifying several parameters such as the version,cluster topology and hardware node details.As specified in *[513]* 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.
323. Rocks
[514] Rocks provides open cluster distribution solution is buid targetting the scientist with less cluster experience to ease the process of deployment,managing,upgrading and scaling high performance parallel computing cluster. It was initially build 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 institution for their projects.
324. 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 *[515]*. 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 *[516]*. 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.
325. Ubuntu MaaS
326. Facebook Tupperware
Facebook Tupperware is a system which provisions services by taking requirements from engineers and mapping them to actual hardware allocations using containers *[517]*.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 Databse. 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.

327. AWS OpsWorks
AWS Opsworks is a configuration service provided by Amazon Web Services that uses Chef, a Ruby and Erlang based configuration management tool *[518]*, 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 *[519]*.
328. OpenStack Ironic
Ironic *[520]* 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.
329. Google Kubernetes
Google Kubernetes is a cluster management platform developed by Google. According to *[521]* 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 extensible primarily through Kubernetes API. Kubernetes follows a master-slave architecture, according to *[522]* 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. Kunernetes

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.

330. 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 *[523]*. 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. *[524]*.
331. 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 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”.:cite:*lindsay2016*
332. OpenTOSCA
333. Winery
Eclipse Winery *[525]* is a “web-based environment to graphically model [Topology and Orchestration Specification for Cloud Applications] TOSCA topologies and plans managing these topologies.” Winery *[526]* 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 *[526]* 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 *[526]* 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.”
334. CloudML
335. Blueprints
In *[527]*, it is explained that “IBM Blueprint has been replaced by IBM Blueworks Live.” In *[528]*, IBM Blueworks Live is described “as a cloud-based business process modeller, belonging under the set of IBM SmartCloud applications” that as *[529]* states “drive[s] out inefficiencies and improve[s] business operations.” Similarly to Google Docs, IBM Blueworks 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 Blueworks Live are both simple to use in a

collaborative manner, *[528]* explains that IBM Blueworks Live has the "capabilities to implement more complex models."

336. 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. It's latest release is Terraform 0.8 According to the website *[530]* 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 *[531]* 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.
337. 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 *[532]*.
338. Any2Api
This framework *[533]* 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:
    (a) Invokers, which are capable of running a given type of executable for example cookbook invoker can be used to run Chef cookbooks
    (b) Scanners, which are capable of scanning modules of certain type for example cookbook scanner scans Chef cookbooks.
    (c) API impl generators, which are doingthe actual work to generate the API implementation.

    The final API implementation *[534]* 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.

5.2.19 IaaS Management from HPC to hypervisors

339. Xen
Xen is the only open-source bare-metal hypervisor based on microkernel design *[535]*. 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 *[536]*. 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 "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) *[537]*.
340. KVM
[538] 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. 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 it's ease of use, it can be completely controlled by ourselves and there is an ease for migration from or to other platforms. *[539]* 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.
341. QEMU
QEMU (Quick Emulator) is a generic open source hosted hypervisor *[540]* 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 *[541]*.

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.

342. 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 include Hyper-V, in which it would launch both Xbox OS and Windows 10. *[542]*
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.
343. VirtualBox
344. 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. *[543]* 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 archite- cture 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. *[544]* 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, *[545]* 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 beancounters. 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.

345. 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 *[546]*. LXC are similar to the treditional virtual machines but instead of having seperate kernel process for the guest operating system being run, containers would share the kernal process with the host operating system. This is made possible with the implementation of namespaces and cgroups. *[547]*
Containers are light weighed (As guest operating system loading and booting is eleminated) and more customizable compared to VM technologies.The basis for docker developement is also LXC. *[548]*. Linux containers would work on the major distributions of linux this would not work on Microsoft Windows.

346. Linux-Vserver
Linux-VServers are used on web hosting services, pooling resources and containing any security breach. [www.linux-vserver.org/Paper] “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 provate 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.

347. OpenStack
OpenStack *[549]* 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 *[550]*.
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 [Apache-License]

348. OpenNebula
According to OpenNebula webpage *[551]* it provides simple but feature-rich and flexible solutions for the comprehensive management of virtualized data centers to enable private, public and hybrid laaS clouds. It is a cloud computing platform for managing heterogenous distributed data centers infrastructures. The OpenNebula toolkit includes features for management, scalability, security and accounting. It used in various sectors like hosting providers, telecom providers, telecom operators, IT service providers, supercomputing centers, research labs, and international research projects *[552]*. More about OpenNebula can be found in the following paper that is published at ieee computer society *[553]*
349. 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 physical resources *[554]*. 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 *[555]*.
350. Nimbus
Nimbus Infrastructure *[556]* 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 *[557]* 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 *[558]*, 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.
351. 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. *[559]*
ColudStack 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. *[560]*
352. CoreOS
[561] 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 environements 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.

353. rkt
rkt is an container manager developed by CoreOS *[562]* designed for Linux clusters. It is an alternative for Docker runtime and is designed for server environments with high security and composibity 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-priviliged 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 *[563]*.

354. VMware ESXi
VMware ESXi (formerly ESX) is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers *[564]*. 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 *[565]*.

355. vSphere and vCloud
vSphere was developed by VMware and is a cloud computing virtualization platform. *[566]* 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 areexist. Similarly, vCloud is also a suite of applications but for establishing an infrastructure for a private cloud. *[567]* 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.

356. 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. :cite:'www-aws'
    Improve: all of them need bibentries
357. Azure
358. 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 *[568]*. 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 *[569]*. 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 *[570]*. Windows Azure services platform, Amazon elastic compute cloud and Sun cloud are few examples of public clouds.
359. 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.

360. 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 *[571]*. 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 *[572]*.

5.2.20 Cross-Cutting Functions

Monitoring

361. Ambari
Apache Amabari 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. *[573]* 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. *[574]* provides documentation about Ambari, including a quick start guide for installing a cluster with Ambari. *[575]* provides the project documents for ambari on github.
362. 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 *[576]*.
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 *[577]*. 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.
363. Nagios *[578]*

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 *[579]*: 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. *[580]* Besides its open source components, Nagios also has commercial products to serve needing clients.

364. Inca
Inca is a grid monitoring *[581]* software suite. It provides grid monitoring features. These monitoring features provide operators failure trends, debugging support, email notifications, environmental issues etc. *[582]*. 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

365. 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’‘. *[583]* 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 globa scale.

Currently any two and four year higher education institution that is accredited is eligble for joining InCommon.

366. Eduroam *[584]*
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 reasearch 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 instituion connected through eduroam. According to the orgnizatioin it uses a combination of radius-based infrastructuor with 802.1X standard techonology to provide roaming acess across reasearch and educational networks. The role of the RADIUS hierarchy is to forward user crednetials to the users home instituion where they can be verified. This proved to be a successful solution when compared to other traditonal ways like using MAC-adress, SSID, WEP, 802.1x(EAP-TLS, EAP-TTLS), VPN Clients, Mobile-IP etc which have their own short comings when used for this purpose *[585]*. Today by enabling eduroam users get access to internet across 70 countries and tens of thousands of access points worldwide.

367. OpenStack Keystone
[586] 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 *[587]*. 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 *[588]*.
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 *[589]*.

368. 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. *[590]*
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.

369. Sentry
[591] "Apache Sentry 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."
370. Sqrrl
371. 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. *[592]* 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 *[593]*. 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.
372. SAML OAuth
As explained in *[594]*, 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

373. Google Chubby
Chubby Distributed lock service *[595]* 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 *[596]*, 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.
374. 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 *[597]*. 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 *[598]* by stamping each update with a number. Clients can also set a watch on a znode to be notified of any change *[599]*. The performance of the ZooKeeper is optimum for "read-dominant" workloads. It's maintained by Apache and is open-source.
375. 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 *[600]*. 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.
376. JGroups

Message and Data Protocols

377. Avro
378. 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. *[601]* 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. *[602]*

379. Protobuf
Protocol Buffer *[603]* 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 apllication 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.

5.2.21 New Technologies (To Be Integrated by the AIs)

381. Snort
[604] 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.
382. 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 *[605]*.
383. Zeppelin
Apache Zeppelin *[606]* 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.

384. Open MPI
The Open MPI Project *[607]* 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 *[608]* 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.

385. Apache Tomcat
Apache tomcat is an open source java servlet container. *[609]* It is used in IT industry as a HTTP web server which listens to the requests made by web client and send reponses. 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. *[610]* 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.

386. 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 *[611]* (perhaps in a completely different language). Google contributed its Dataflow SDK, the Dataflow model and three runners *[612]* 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. *[613]* It should be grouped with the technologies in the *Interoperability* section.

387. 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. *[614]*

388. 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. *[615].* CUDA is used for accelerated rendering of 3D graphics, accelerated interconversion of video file formats, encryption, decryption and compression of files. It is also usedd for distributed calculations, face recognition and distributed computing. *[615]*

389. 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. *[616]*

389. CDAP
CDAP *[617]* 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 *[618]* 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 *[619]* provide explain how to implement apps on CDAP platform.

389. Apache Arrow
Apache arrow allows execution engines to utilize what is known as Single Input multiple data (SIMD). *[620]* 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.

390. 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 *[621]*.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.

391. Apache OODT
Apache Object Oriented Data Technology (OODT) *[622]* 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. *[623]* 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.

389. Omid
Omid is a "flexible, reliable, high performant and scalable ACID transactional framework" *[624]* 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. [www-www-yahooomid].
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 *[191]*

390. *[625]* Askalon was developed at the University of Innsbruck. 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.

391. 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 *[626]*.

392. 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 libxlc to the remote network or local unix socket. *[627]*. It implements the under previlized conatiners by default adding more security. It works with Image based work flow supports online snapshopping and live container migration. *[628]*.It was build with aim of providing VM like virtulization with container like performance. *[629]*

393. Wink

Apache wink *[630]* 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.

394. Apache Apex

Apache Apex is “a YARN(Hadoop 2.0)-native platform that unifies cloud and batch processing” *[631]*.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 Marching, Query and Scripting, Stream manipulators, Parsers and UI & Charting operators *[632]*.

Apache Knox

According to :cite:’knox’, “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. :cite:’fielding’ API Gateways manage concerns related to “Authentication, Transport Security, Load-balancing, Request Dispatching (including fault tolerance and service discovery), Depenency Resolution, Transport Transformations.” :cite:’peyrott’ 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.” :cite:’knox’ 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.” :cite:’knox’ Some Apache Hadoop Services that integrate with Knox are, “Ambari, WebHDFS (HDFS), Templeton (Hcatalog), Stargate (Hbase), Oozie, Hive/JDBC, Yarn RM, [and] Storm.” :cite:’knox’ Apache Knox has a configuration driven method to aid in the addition of new routing services. :cite:’knox’ This allows support for new and custom Apache Hadoop REST APIs to be added to the Knox gateway quickly and easily. :cite:’knox’ This technology would be best placed under the interoperability category.

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 *[633]*. 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 *[634]*. 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 *[635]*.

5.2.22 Excersise

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 invesitgate 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 plagearism and use proper quotations or better rewrite the text.

You must look at technologies-hw to sucessfully complete the homework

A video about this hoemwork 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 realated 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. Fith you can push. Six 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 grammer 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 groundbraking work conducted in England, Allan Turing, introduced the turing machine in the years 1936-37 [2]. Its definition is base 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 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 *New Technologies* 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 *HID Assignment page*.

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.

Open Discussion: For useful information on how to correctly create BibTeX entries, see and contribute to these open discussion threads Piazza.

5.2.23 References

5.3 References

6. Document Preparation

6.1 Basic Emacs

One of the most useful refernces for emacs is the following refrence card. It takes some time to use this card efficiently, but the most important commands are written on it. Generations of students have litterally been just presented with thsi 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 graet 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	Forward one character
C-n	Next line
C-b	Back one character
C-p	Previous line

Here are some ways to move around in larger increments:

Key	Description
C-a	Beginning of line
M-f	Forward one word
M-a	Previous sentence
M-v	Previous screen
M-<	Beginning of buffer
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

Seaching is easy with the following commands

Key	Description
C-s	Incremental search forward
C-r	Incremental search backward

Replace

Key	Description
M-%	Query replace

Killing (“cutting”) text

Key	Description
C-k	Kill line

Yanking

Key	Description
C-y	Yanks 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
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.

6.2 LaTeX

6.2.1 Introduction

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

The information provided here is not intended to replace one of the many text books available about LaTeX. For the beginning you might be just fine with the documentation provided here. For serious users I recommend to purchase a book. Examples for books include

- LaTeX Users and Reference Guide, by Leslie Lamport
- LaTeX an Introduction, by Helmut Kopka
- The LaTeX Companion, by Frank Mittelbach

If you do not want to buy a book you can find a lot of useful information in the LaTeX reference manual.

6.2.2 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 incredible stable
- produces high quality output
- is platform independent
- has lots of templates
- has been around for many years so it works well
- removes you form the pain of figure placements
- focusses you on content rather tan 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 look. 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: reduce 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

6.2.3 Editing

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 aquaemacs and carbon emacs have build in support for LaTeX. Spell checking is done with flyspell in emacs.

Vi/Vim

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

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/

6.2.4 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.

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 ma 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.

6.2.5 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.

Note: Skim can be found at: `http://skim-app.sourceforge.net/`

6.2.6 Installation

Local Install

Installing LaTeX is trivial, but requires sufficient space and time as it is a large environment. 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: less fancy GUI to bibtex files

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 instal them.

Online Services

Sharelatex

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.

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, put you are allowed to use the OSA template.

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

6.2.7 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
```

A great example is provided at:

- https://gitlab.com/cloudmesh/project-000/tree/master/report

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 autogenerate. 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
```

Note: for make watch its best to use skim as pdf previewer

6.2.8 Generating 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 rations 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 tablaeu 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.

6.2.9 Bibliographies

LaTeX integrates very well with bibtex. There are several preformatted 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}
```

Than 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 cand create a multitude of bibliography file formats and allows upload in other online bibliography managers.

Video: `https://youtu.be/cMtYOHCHZ3k`

Video with cc: `https://www.youtube.com/watch?v=QVbifcLgMic`

jabref and MSWord

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

Note: 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 involve, 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 has figures or tables, or you need to add references.

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.

Endnote

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

- `http://endnote.com/`

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 Mendeleys integration into word slow.

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

6.2.10 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.

6.2.11 Links

- The LaTeX Reference Manual provides a good introduction to Latex.

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.

6.2.12 Tips

Including figures over two columns:

- `http://tex.stackexchange.com/questions/30985/displaying-a-wide-figure-in-a-two-column-document`
- positioning figures with textwidth and columnwidth `https://www.sharelatex.com/learn/Positioning_images_and_tables`
- An organization as 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.

6.3 Bibliography Management

Warning: THis page is under active development and students are encouraged to contribute

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

6.3.1 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 ov access is listed in the entry or even the release version of the source code.

Let us without bias chose a random source dode 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/progrium/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/progrium/buildstep},
  Owner =       {S17-IO-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},
```

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

---

6.3.2 Wikipedia Entry

Please fill out

6.3.3 Web Page

Please fill out

6.3.4 InProceedings

Please fill out

6.3.5 TechReport

Please fill out

6.3.6 Article

Please fill out

6.3.7 Proceedings

Please fill out

6.4 What is the entry for a Blog

In this example, we show how to arrive at the BibTeX entry for `http://dataconomy.com/2014/08/google-cloud-dataflow/`

A Misc entry has the following fields -

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

}

To start with we go to the blog URL. From the blog we get the details for author, title, month, year,publisher and URL. The howpublished field we add blog, since it is a blog post.

@Misc{www-google-cloud-dataflow, author = {Eileen McNulty}, title = {Understanding Big Data: What is Google Cloud Dataflow?}, publisher = {Dataconomy.com}, month = aug, year = {2014}, howpublished = {Blog}, url = {`http://dataconomy.com/2014/08/google-cloud-dataflow`}, key = {Dataconomy}, owner = {S17-IR-2006 S17-IR-2026 S17-IR-2035}, note = {Accessed: 13-Feb-2017} }

Misc: Misc type is used when nothing else fits the type of entry. For a Misc entry we have do not have any required fields

Optional fields: author, title, howpublished, month, year, note. Since this is blog we use the misc entry. Online entry type is also available but in order to maintain compatibility with older formats we choose the Misc.

Key: When the author and editor information is missing we use the key field for alphabetizing, cross-referencing, and creating a label. One should not confuse the key with the label that appears in the cite command and at the beginning of the database entry. key = {organisation}

Label: The label field is should contain 3 letters from[auth] 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. Here since this is an online reference we use www- since it allows us to quickly search through a paper that we write and identifying online references, to judge if we should replace them with articles in journals or proceedings.

Author: This field should include all the authors for your entry. Author names should be separated using "and". We can write the author names in two forms:

```
Example 1: Type 1 -- Donald E. Knuth
           Type 2 -- Knuth, Donald E.
Example 2: Type 1 -- Eddie van Halen
           Type 2 -- van Halen, Eddie
```

The second type should be used for authors with more than two names to differentiate between middle name and last names.

Title: This is the title of the work. The capitalization depends on the bibliography style and the language used. For words that have to be capitalized (such as a proper noun), enclose the word (or its first letter) in braces.

Howpublished: Since the source is a blog, the howpublished field shall hold the value {Blog} rather than a web page. If the url specified was a normal webpage, the {Web Page} entry would be valid.

Month: It's best to use the three-letter abbreviations for the month, rather than spelling out the month yourself. This lets the bibliography style be consistent, since month in not capitalized in some languages. And if you want to include information for the day of the month, the month field is usually the best place. Example: month = aug # "~10,"

Owner: the HID mapping each student and their technologies

6.5 What is the entry for a 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

Furtheromore wee 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 =          {},
  ownwer =       {},
  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 caitalization 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 t 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 youst 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-IO-3022 S17-IO-3010 S17-IO-3012},
  pages = {296},
  url = {http://www.ebook.de/de/product/21687528/norman_maurer_netty_in_
↪action.html},
```

```
}
```

6.6 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 aone 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.

6.6.1 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`

6.6.2 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.

6.6.3 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
~~~~~~~~~~~~~~~~~~~~~
```

6.6.4 Listtable

```
.. csv-table:: Eye colors
   :header: "Name", "Firstname", "eyes"
   :widths: 20, 20, 10

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

Table 6.1: a title

Name	Firstname	eyes
von Laszewski	Gregor	gray

6.6.5 Exceltable

we have integrated Excel table from `http://pythonhosted.org//sphinxcontrib-exceltable/` intou our sphinx allowing the definition of more elaborate tables specified in excel. Howere 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`

6.6.6 Boxes

Seealso

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

See also:

This is a simple **seealso** note.

Note

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

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

Warning

Warning: note the space between the directive and the text

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

Others

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.6.7 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.

6.6.8 Programm examples

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

This isa na example for python:

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

This is an example for a shell command:

```
$ ls -lisa
```

6.6.9 Hyperlinks

Direct links to html pages can ve 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 prefered 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

6.6.10 Todo

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

Todo

an example

7. Linux

7.1 Linux Shell

There are many good tutorials out there that explain why one needs a linux shell and not just a GUI. Randomly we picked the firts one that came up with a google query (This is not an endorsement for the material we point to, but could be a worth while read for someone that has no experience in Shell programming:

- `http://linuxcommand.org/lc3_learning_the_shell.php`

Certainly you are welcome to use other resources that may suite you best. We will however summarize in table form a number of useful commands that you may als find in a link to a RefCard.

- `http://www.cheat-sheets.org/#Linux`

7.1.1 File commands

Find included a number of commands related to file manipulation.

Command	Description
ls	Directory listing
ls -lisa	list details
cd *dirname*	Change directory to *dirname*
mkdir *dirname*	create the directory
pwd	print working directory
rm *file*	remove the file
cp *a b*	copy file *a* to *b*
mv *a b*	move/rename file *a* to *b*
cat *a*	print content of file*a*
less *a*	print paged content of file *a*
head -5 *a*	Display first 5 lines of file *a*
tail -5 *a*	Display last 5 lines of file *a*

7.1.2 Search commands

Find included a number of commands related to seraching.

Command	Description
fgrep	TBD
grep -R “xyz” .	TBD
find . -name “*.py” | TBD |	

7.1.3 Help

Find included a number of commands related to manual pages.

Command	Description
man *command*	manual page for the *command*

7.1.4 Keyboard Shortcuts

These shortcuts will come in handy. Note that many overlap with emacs short cuts.

Keys	Description
Up Arrow	Show the previous command
Ctrl + z	Stops the current command
	resume with fg in the foreground
	resume with bg in the background
Ctrl + c	Halts the current command
Ctrl + l	Clear the screen
Ctrl + a	Return to the start of the command you’re typing
Ctrl + e	Go to the end of the command you’re typing
Ctrl + k	Cut everything after the cursor to a special clipboard
Ctrl + y	Paste from the special clipboard
Ctrl + d	Log out of current session, similar to exit

7.1.5 .bashrc and .bash_profile

Warning: Not yet implemented.

7.1.6 Exercise

Linux.1: Familiarize yourself with the commands
Linux.2: Find more commands that you find useful and add them to this page.
Linux.3: Use the *sort* command to sort all lines of a file while removing duplicates.

7.2 Refcards

We present you with a list of useful short refrence cards. This cards can be extremly useful to remind yourself about some important commands and features. Having them could simplify your interaction with the systems, We not only collected here some refcards about Linux, but also about other useful tools and services.

If you like to add new topics, let us know via your contribution (see the contribution section).

Emacs	`https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf`
Vi	`http://www.ks.uiuc.edu/Training/Tutorials/Reference/virefcard.pdf`
Linux	`http://www.cs.jhu.edu/~joanne/unixRC.pdf`
Makefile	`http://www.tofgarion.net/lectures/IN323/refcards/refcardMakeIN323.pdf`
R	`https://cran.r-project.org/doc/contrib/Short-refcard.pdf`
Python	`https://dzone.com/refcardz/core-python`
Python Data	`https://dzone.com/refcardz/data-mining-discovering-and`
SQL	`http://www.digilife.be/quickreferences/QRC/MySQL-4.02a.pdf`
Vim	`http://michaelgoerz.net/refcards/vimqrc.pdf`
LaTeX	`https://wch.github.io/latexsheet/latexsheet.pdf`
Git	`https://education.github.com/git-cheat-sheet-education.pdf`
Open-stack	`http://docs.openstack.org/user-guide/cli_cheat_sheet.html`
Open-stack	`http://cmias.free.fr/IMG/pdf/rc208_010d-openstack_2.pdf`
RST	`https://github.com/ralsina/rst-cheatsheet/blob/master/rst-cheatsheet.pdf`

Others:

- Numpi/Pandas: http://www.cheat-sheets.org/saved-copy/NumPy_SciPy_Pandas_Quandl_Cheat_Sheet.pdf
- Cheat Sheets: http://www.cheat-sheets.org/
- Python Tutorial: http://fivedots.coe.psu.ac.th/Software.coe/learnPython/Cheat%20Sheets/python2.pdf
- Python: http://www.cheat-sheets.org/saved-copy/PQRC-2.4-A4-latest.pdf
- Python: https://www.cheatography.com/davechild/cheat-sheets/python/pdf/
- Python API Index: http://overapi.com/python
- Python 3: https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf

7.3 Using SSH Keys

If you do not know what ssh is we recommend that you read up on it . However, the simple material presented here will help you etting started quickly. It can however not replace the more comprehensive documentation.

To access remote resources this is often achieved via SSH. You need to provide a public ssh key to FutureSystem. We explain how to generate a ssh key, upload it to the FutureSystem portal and log onto the resources. This manual covers UNIX, Mac OS X.

7.3.1 Using SSH from Windows

Hint: For Linux users, please skip to the section *Generate a SSH key*

Hint: For Mac users, please skip to the section *Using SSH on Mac OS X*

Warning: For this class we recommend that you use a virtual machine via virtual box and use the Linux ssh instructions. The information here is just provided for completness and no support will be offered for native windows support.

Windows users need to have some special software to be able to use the SSH commands. If you have one that you are comfortable with and know how to setup key pairs and access the contents of your public key, please feel free to use it.

The most popular software making ssh clients available to Windows users include

- cygwin
- putty

- or installing a virtualiztion software and running Linux virtual machine on your Windows OS.
- using chocolatey
- using bash ubuntu under WIndows 10 (we need a contribution on this)

We will be discussing here how to use it in Powershell with the help of chopolatey. Other options may be better suited for you and we leave it up to you to make this decission. In general we recommend that you use an ubuntu OS either on bare hardware or a virtual machine. Naturally your computer must support this. It will be up to you to find such a computer.

However if you want a unix like environments with ssh you can use Chocolatey.

Chocolatey is a software management tool that mimics the install experience that you have on Linux and OSX. It has a repository with many packages. Before using and installing a package be aware of the consequences when installing software on your computer. Please be aware that there could be malicious code offered in the chocolatey repository although the distributors try to remove them.

The installation is sufficently explained at

- `https://chocolatey.org/install`

Once installed you have a command choco and you should make sure you have the newest version with

```
choco upgrade chocolatey
```

Now you can browse packages at

- `https://chocolatey.org/packages`

Search for openssh and see the results. You may find different versions. Select the one that most suits you and satisfies your security requirements as well as your architecture. Lets assume you chose the Microsoft port, than you can install it with:

```
choco install win32-openssh
```

Warning: If you have a different version such as a 64 bit version please find teh appropriate commands

Other packages of interest include

- LaTeX:: *choco install miktex*
- jabref: *choco install jabref*
- pycharm: *choco install pycharm-community*
- python 2.7.11: *choco install python2*
- pip: *choco install pip*
- virtual box: *choco install virtualbox*

- emacs: *choco install emacs*
- lyx: *choco install lyx*
- vagrant: *choco install vagrant*

Before installing any of them evaluate if you need them.

7.3.2 Using SSH on Mac OS X

Mac OS X comes with an ssh client. In order to use it you need to open the `Terminal.app` application. Go to `Finder`, then click `Go` in the menu bar at the top of the screen. Now click `Utilities` and then open the `Terminal` application.

7.3.3 Generate a SSH key

First we must generate a ssh key with the tool ssh-keygen. This program is commonly available on most UNIX systems (this includes Cygwin if you installed the ssh module or use our pre-generated cygwin executable). It will ask you for the location and name of the new key. It will also ask you for a passphrase, which you **MUST** provide. Some teachers and teaching assistants advice you to not use passphrases. This is **WRONG** as it allows someone that gains access to your computer to also gain access to all resources that have the public key. Also, please use a strong passphrase to protect it appropriately.

In case you already have a ssh key in your machine, you can reuse it and skip this whole section.

To generate the key, please type:

Example:

```
ssh-keygen -t rsa -C localname@indiana.edu
```

This command requires the interaction of the user. The first question is:

```
Enter file in which to save the key (/home/localname/.ssh/id_rsa):
```

We recommend using the default location ~/.ssh/ and the default name id_rsa. To do so, just press the enter key.

Note: Your *localname* is the username on your computer.

The second and third question is to protect your ssh key with a passphrase. This passphrase will protect your key because you need to type it when you want to use it. Thus, you can either type a passphrase or press enter to leave it without passphrase. To avoid security problems, you **MUST** chose a passphrase. Make sure to not just type return for an empty passphrase:

```
Enter passphrase (empty for no passphrase):
```

and:

```
Enter same passphrase again:
```

If executed correctly, you will see some output similar to:

```
  Generating public/private rsa key pair.
  Enter file in which to save the key (/home/localname/.ssh/id_rsa):
  Enter passphrase (empty for no passphrase):
  Enter same passphrase again:
  Your identification has been saved in /home/localname/.ssh/id_rsa.
  Your public key has been saved in /home/localname/.ssh/id_rsa.pub.
  The key fingerprint is:
  34:87:67:ea:c2:49:ee:c2:81:d2:10:84:b1:3e:05:59 localname@indiana.edu
  The key's random art image  File "/Users/grey/.pyenv/versions/2.7.13/
 ↪envs/ENV2/lib/python2.7/site-packages/traitlets/config/application.py",␣
 ↪line 445, in initialize_subcommand
  subapp = import_item(subapp)
File "/Users/grey/.pyenv/versions/2.7.13/envs/ENV2/lib/python2.7/site-
 ↪packages/ipython_genutils/importstring.py", line 31, in import_item
  module = __import__(package, fromlist=[obj])
```

ImportError: No module named nbconvert.nbconvertapp is:

```
+--[ RSA 2048]----+
|.+...Eo= .       |
| ..=.o + o +o    |
|O.  o o +.o      |
| = .   . .       |
+-----------------+
```

Once, you have generated your key, you should have them in the .ssh directory. You can check it by

```
$ cat ~/.ssh/id_rsa.pub
```

If everything is normal, you will see something like:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCXJH2iG2FMHqC6T/U7uB8kt
6KlRh4kUOjgw9sc4Uu+Uwe/EwDOwk6CBQMB+HKb9upvCRW/851UyRUagtlhgy
thkoamyiOVvhTVZhj61pTdhyl1t8hlkoL19JVnVBPP5kIN3wVyNAJjYBrAUNW
4dXKXtmfkXp98T3OW4mxAtTH434MaT+QcPTcxims/hwsUeDAVKZY7UgZhEbiE
xxkejtnRBHTipiOWO3WO5TOUGRW7EuKf/4ftNVPilCO4DpfY44NFG1xPwHeim
Uk+t9h48pBQj16FrUCpOrSO2Pj+4/9dNeS1kmNJu5ZYS8HVRhvuoTXuAY/UVc
ynEPUegkp+qYnR user@myemail.edu
```

7.3.4 Add or Replace Passphrase for an Already Generated Key

In case you need to change your change passphrase, you can simply run "ssh-keygen -p" command. Then specify the location of your current key, and input (old and) new passphrases. There is no need to re-generate keys:

```
ssh-keygen -p
```

You will see the following output once you have completed that step:

```
Enter file in which the key is (/home/localname/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/localname/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

7.3.5 Upload the key to gitlab

Follow the instructions provided here:

- `http://docs.gitlab.com/ce/ssh/README.html`

7.3.6 Exercise

SSH.1: create an SSH key pair

SSH.2: upload the key to github and/or gitlab. Create a fork in git and use your ssh key to clone and commit to it

SSH.3: Get an account on futuresystems.org (if you are authorized to do so). Upload your key to futuresystems.org. Login to india.futuresystems.org Note. that this could take some time as administrators need to approve you. Be patient.

7.4 Ubuntu Development Configurations

7.4.1 Development Configuration

The documentation on how to configure the virtual machine and install many useful programs is posted at:

- `https://github.com/cloudmesh/ansible-cloudmesh-ubuntu-xenial`

You simply have to execute the following commands in the terminal of the virtual machine. In order to eliminate confusion with other terminals, we use the prefix *vm*> *$* to indicate any command that is to be started on the virtual machine. Otherwise it is clear from the context:

```
vm>$ wget https://raw.githubusercontent.com/cloudmesh/ansible-cloudmesh-
 ↪ubuntu-xenial/master/bootstrap.sh
vm>$ bash bootstrap.sh
```

A video showcasing this install is available:

- Video: `https://youtu.be/YqXIj_Wzfsc`

A video showcasing the upload to gitlab from within the vm using commandline tools

- Video: `https://youtu.be/EnpneUY82I8`

7.5 Virtual Box Installation and Instructions

For development purposes we recommend tha you use for this class an ubuntu virtual machine that you set up with the help of virtualbox.

Only after you have successfully used ubuntu in a virtual machine you will be allowed to use virtual machine son clouds.

A "cloud drivers license test" will be conducted to let you gain access to the cloud infrastructure. We will announce this test. Before you have not passed the test, you will not be able to use the clouds. Furthermore, you do not have to ask us for join requests before you have not passed the test. Please be patient. Only students enrolled in the class can get access to the cloud.

7.5.1 Creation

First you will need to install virtualbox. It is easy to install and details can be found at

- `https://www.virtualbox.org/wiki/Downloads`

After you have installed virtualbox you also need to use an image. For this class we will be using ubuntu Desktop 16.04 which you can find at:

- `http://www.ubuntu.com/download/desktop`

Please note the recommended requirements that also apply to a virtual machine:

- 2 GHz dual core processor or better
- 2 GB system memory
- 25 GB of free hard drive space

A video to showcase such an install is available at:

- Video: `https://youtu.be/NWibDntN2M4`

Warning: If you specify your machien too small you will not be able to install the development environment. Gregor used on his machine 8gb of RAM and 20GB diskspace.

Please let us know the smalest configuration that works for you and share this in Piaza. Only update if yours is smaller and works than a previous post. If not do not post.

7.5.2 Guest additions

The virtual guest additions allow you to easily do the following tasks:

- Resize the windows of the vm
- Copy and paste content between the Guest operating system and the host operating system windows.

This way you can use many native programs on you host and copy contents easily into for example a terminal or an editor that you run in the Vm.

A video is located at

- Video: `https://youtu.be/wdCoiNdn2jA`

Note: Please reboot the machine after installation and configuration.

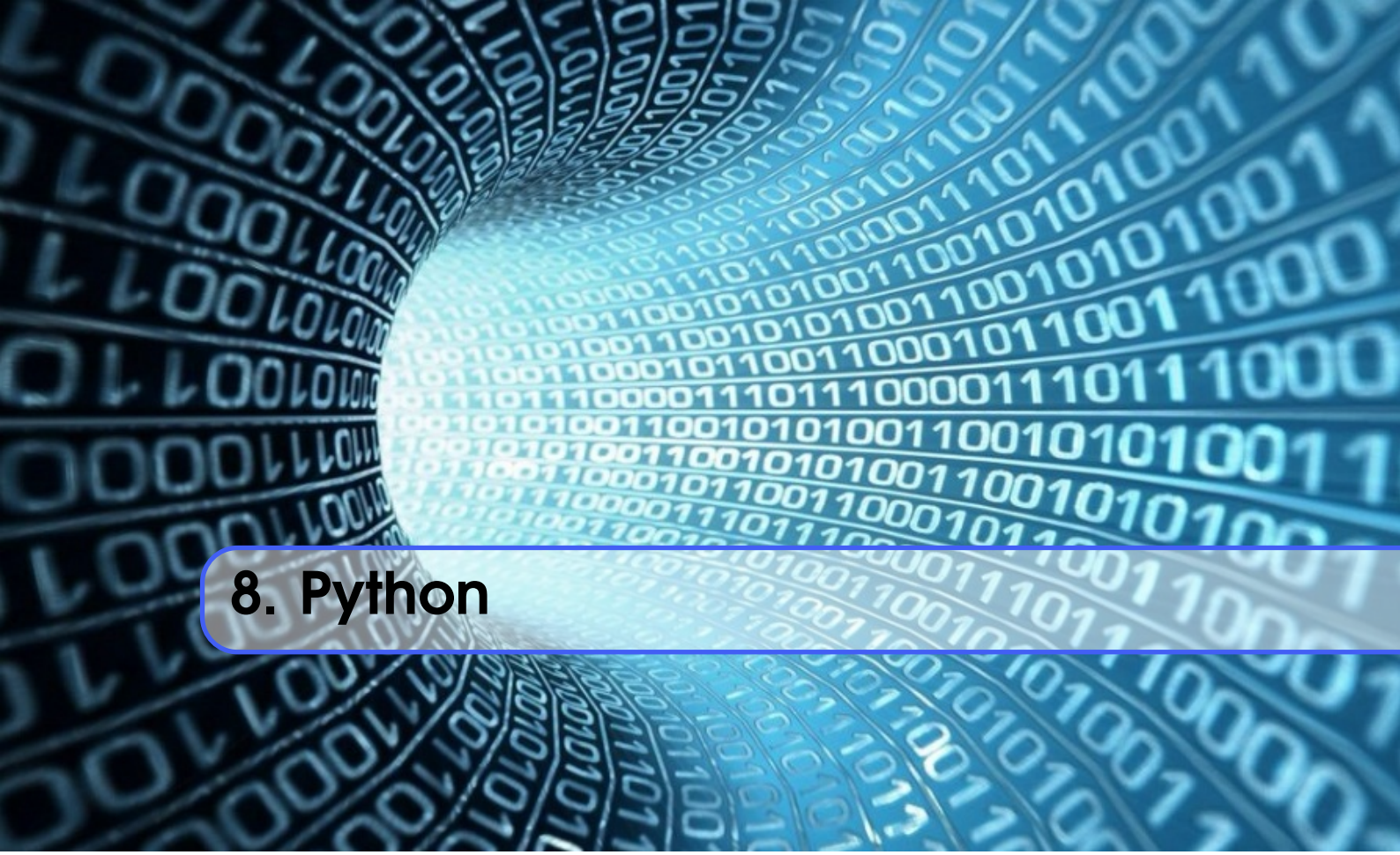
On OSX you can once you have enabled bidirectional copying in the Device tab with

OSX -> Vbox: *command c -> shift CONTRL v*
Vbox to OSX: *shift CONTRL v -> shift CONTRL v*

On Windows the key combination is naturally different. Please consult your windows manual.

7.5.3 Exercise

Virtualbox.1: Install ubuntu desktop on your computer with guest additions.
Virtualbox.2: Make sure you know how to paste and copy between your host and guest operating system
Virtualbox.3: Install the programs defined by the development configuration



8. Python

8.1 Introduction to Python

8.1.1 Acknowledgments

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

8.1.2 Description

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 is easily 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.

8.1.3 Philosophy

Python is an interpreted, dynamic, high-level programming language suitable for a wide range of applications. The The Zen of Python summarizes some of its philosophy including:

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

8.1.4 Features

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.

8.1.5 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.

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.

8.1.6 Prerequisite

In order to conduct this lesson you should

- A computer with Python 2.7.x (and preferably, virtualenv)
- Familiarity with command line usage
- A text editor such as PyCharm, emacs, vi or others. You should identity which works best for you and set it up.

8.1.7 Dependencies

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

8.1.8 Learning Goals

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 `virtualenv`

8.1.9 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 not Python 3.

In addition to Python, 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.

8.1.10 virtualenv

Often you have your own computer and you do not like to change its environment to keep it in pristine condition. Python comes with mnay libraries that could for example conflict with libraries that you have installed. To avoid this it is bets to work in an isolated python 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 command so you do not forget to activate it. :ref:‘‘Instructions for this can be found in our lesson on Linux <bashrc>‘.

8.1.11 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.

Tip: 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**: **R**ead-**E**valuate-**P**rint-**L**oop.

8.1.12 Python 3 Features

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
```

Note: 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.

Note: The second of these imports makes sure that the division operator behaves in a way a newcomer to the language might find more intruitive. 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`.

8.1.13 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 a 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.

Tip: 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.

8.1.14 Variables and Simple Data Types

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!
```

8.1.15 Booleans

A **boolean** is a value that indicates the "truthness" 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
```

8.1.16 Numbers and Math

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 yth power:

$$x^y$$

You can combine **float**s and **int**s:

```
>>> 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
```

8.1.17 Types and Using the REPL

We have so far seen a few examples of types: **string**s, **bool**s, **int**s, and **float**s. 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**s, **list**s, **set**s. 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.

8.1.18 Control Statements

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>
```

8.1.19 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)

8.1.20 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 brackest `[]`:

```
>>> >>> 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 specifing 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]
```

8.1.21 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.

8.1.22 Removal and Testing for Membership

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' % str(nums_set), number=100)
0.0004038810729980469
>>> timeit.timeit('random.randint(0, sys.maxint) in nums', setup='import␣
↪random; nums=%s' % str(nums_list), number=100)
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.

8.1.23 Dictionaries

One of the very important datastructures in python is a dictionary also refered 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
```

8.1.24 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**.

8.1.25 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.

8.1.26 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 **module**s. You can execute a Python module form 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 traingle.
  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 are 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.

8.1.27 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)
```

8.1.28 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 Aobject-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.

8.1.29 Database Access

see: `https://www.tutorialspoint.com/python/python_database_access.htm`

8.1.30 Installing Libraries

Often you may need functionality that is not present in Python's standard library. In this case you have two option:

- 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 if virtual environments using the `virtualenv` command, and installing and uninstalling PyPi packages using `pip`.

8.1.31 Virtual Environments

Often when you use shared computing resources, such as `india.futuresystems.org` you will not have permission to install applications in the default global location.

Let's see where `grep` is located:

```
$ which grep
/bin/grep
```

It seems that there are many programs installed in `/bin` such as `mkdir` and `pwd`:

```
$ ls /bin
alsacard    dbus-cleanup-sockets  env             hostname         mailx  ␣
 ↪       pwd
alsaunmute  dbus-daemon           ex              igawk            mkdir  ␣
 ↪       raw
...
```

If we wished to add a new program it seems like putting it in `/bin` is the place to start. Let's create an empty file `/bin/hello-$PORTALNAME`:

```
$ touch /bin/hello-$(whoami)
touch: cannot touch `/bin/hello-albert': Permission denied
```

Tip: Recall that $PORTALNAME is your username on FutureSystems, which can also be obtained using the `whoami` shell command. t seems that this is not possible. Since `india` is a shared resources not all users should be allowed to make changes that could affect everyone else. Only a small number of users, the administrators, have the ability to globally modify the system.

We can still create our program in our home directory:

```
$ touch ~/hello-$(whoami)
```

but this becomes cumbersome very quickly if we have a large number of programs to install. Additionally, it is not a good idea to modify the global environment of one's computing system as this can lead to instability and bizarre errors.

A virtual environment is a way of encapsulating and automating the creation and use of a computing environment that is consistent and self-contained.

The tool we use with Python to accomplish this is called `virtualenv`.

Let's try it out. Start by cleaning up our test earlier and going into the home directory:

```
$ rm ~/hello-$(whoami)
$ cd ~
```

Now lets create a virtual env:

```
$ virtualenv ENV
PYTHONHOME is set.  You *must* activate the virtualenv before using it
New python executable in ENV/bin/python
Installing setuptools.............done.
Installing pip...............done.
```

When using `virtualenv` you pass the directory where you which to create the virtual environment, in this case `ENV` in the current (home) directory. We are then told that we must activate the virtual environment before using it and that the python program, setuptools, and pip are installed.

Let's see what we have:

```
$ ls ENV/bin
activate  activate.csh  activate.fish  activate_this.py  easy_install
easy_install-2.7  pip  pip-2.7  python  python2  python2.7
```

It seems that there are several programs installed. Let's see where our current `python` is and what happens after activating this environment:

```
$ which python
/N/soft/python/2.7/bin/python
$ source ENV/bin/activate
(ENV) $ which python
~/ENV/bin/python
```

Important: As virtualenv stated, you **must** activate the virtual environment before it can be used.

Tip: Notice how the shell prompt changed upon activation.

8.1.32 Fixing Bad Code

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.

I'm going to demonstrate creating a user python environment, installing a couple packages from pypi, and use them to examine some code.

First, get the bad code like so:

```
$ wget --no-check-certificate http://git.io/pXqb -O bad_code_example.py
```

Let's examine the code:

```
$ nano 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.

8.1.33 Using pip to Install Packages

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.

Note: You can skip '–trusted-host pypi.python.org' option if you have a patch on urllib3 on Python 2.7.9.

8.1.34 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's 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
```

8.1.35 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

Note: you can receive extra credit if you contribute such a section of your choice addressing the above topics

---

8.1.36 Writing Python 3 Compatible Code

see: `http://python-future.org/compatible_idioms.html`

8.1.37 Using Python on FutureSystems

> **Warning:** 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.

Tip: The details of what the `module load` command does are described in the future lesson modules.

8.1.38 Exercises

Lab - Python - FizzBuzz

Write a python program called fizzbuzz.py that accepts an integer n from the command line. Pass this integer to a function called fizzbuzz.

The fizzbuzz function should then iterate from 1 to n. If the ith number is a multiple of three, print "fizz", if a multiple of 5 print "buzz", if a multiple of both print "fizzbuzz", else print the value.

Lab - Python - Setup for FutureSystems

1. Create a 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 FizzBuzz program.
5. Demonstrate the program works and submit the code and output.

8.1.39 Ecosystem

Autoenv: Directory-based Environments

Link: *Autoenv <https://pypi.python.org/pypi/autoenv/0.2.0>*

> **Warning:** We do not recommend that you use autoenv. INstead we recommend that you use pyenv. For this class neither is important.

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 great for...

- auto-activating virtualenvs
- project-specific environment variables

Here is how 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
```

Here is how to install. Mac OS X Using Homebrew:

```
$ brew install autoenv
$ echo "source $(brew --prefix autoenv)/activate.sh" >> ~/.bash_profile
```

Using pip:

```
$ pip install autoenv
$ echo "source `which activate.sh`" >> ~/.bashrc
```

Using git:

```
$ 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 `~/.autoenv_authorized`
- `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`).

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.

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 form 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 form pypi which is standard

and should be supported by the frameworks. We have not tested cloudmesh on any other python version then 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 canpoy, 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 instals 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 an 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 anaconde, develop script to generate a conda packge form github
- deploy run and document cloudmesh on canopy, develop script to generate a conda packge form github
- deploy run and document cloudmesh on ironpython
- other documentation that would be useful

8.1.40 Useful Ecosystem Links

- `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/`

8.1.41 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:

- `http://ivory.idyll.org/articles/advanced-swc/`
- `http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html`
- `http://www.youtube.com/watch?v=0vJJlVBVTFg`
- `http://www.korokithakis.net/tutorials/python/`
- `http://www.afterhoursprogramming.com/tutorial/Python/Introduction/`
- `http://www.greenteapress.com/thinkpython/thinkCSpy.pdf`

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.

8.2 pyenv

pyenv allows users to switch between multiple versions of Python [`https://github.com/yyuu/pyenv`].

pyenv allows:

- 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 accross different python versions;
- integrate with tox [`https://tox.readthedocs.io/en/latest/`].

8.2.1 Install pyenv on OSX

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`]. 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/install)"
brew update
brew install pyenv pyenv-virtualenv pyenv-virtualenvwrapper
brew install readline xz
```

8.2.2 Python Versions

In order to use pyenv, you need to have more than one python version installed. We recommend you download and install python 2.7.13 and 3.6 from python.org [https://www.python.org/downloads/]

8.2.3 Install Different Python Versions

You can now install different versions of python into your local environment with the following commands:

```
$ pyenv install 2.7.13
$ pyenv install 3.6.0
```

You can set the global python defualt version with:

```
$ pyenv global 2.7.13
```

Type the following to determine which versions you have available:

```
$ pyenv version
```

Associate a specifc environment name with a certain python version, use the following commands:

```
$ pyenv virtualenv 2.7.13 ENV2
$ pyenv virtualenv 3.6.0 ENV3
```

In the example above, *ENV2* would represent python 2.7.13 while *ENV3* would represent python 3.6.0. Often it is easier to type the alias rather than the explicit version.

8.2.4 Set up the Shell

To make all work smoothly from your terminal, you can include the follwowing 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}"
```

8.2.5 Switching Environments

After setting up the different environments, switching between them is now easy. Simply use the following commands:

```
(2.7.13) laptop~ gregor$ pyenv activate ENV2
(ENV2) laptop~ gregor$ pyenv activate ENV3
(ENV3) laptop~ gregor$ pyenv activate ENV2
(ENV2) laptop~ gregor$ pyenv deactivate ENV2
(2.7.13) laptop~ gregor$
```

To make it even easier, you can add the following lines to your *.bash_provild* 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
```

Try it out.

8.2.6 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
```

8.2.7 Excersise

pyenv.1: Write installation instructions for an operating system of your choice and add to this documentation.

pyenv.2: Replicate the steps above, so you can type in ENV2 and ENV3 in your terminals to switch between python 2 and 3.

8.3 Python for Big Data

8.3.1 An Example with Pandas, NumPy and Matplotlib

In this example, we will download some traffic citation data for the city of Bloomington, IN, load it into Python and generate a histogram. In doing so, you will be exposed to important Python libraries for working with big data such as numpy, pandas and matplotlib.

Set Up Directories and Get Test Data

Data.gov is a government portal for open data and the city of Bloomington, Indiana makes available a number of datasets there.

We will use traffic citations data for 2016.

To start, let's create a separate directory for this project and download the CSV data:

```
$ cd ~/projects/i524
$ mkdir btown-citations
$ cd btown-citations
$ wget https://data.bloomington.in.gov/dataset/c543f0c1-1e37-46ce-a0ba-
 ↪e0a949bd248a/resource/24841976-fd35-4483-a2b4-573bd1e77cfb/download/2016-
 ↪first-quarter-citations.csv
```

Depending on your directory organization, the above might be slightly different for you.

If you go to the link to data.gov for Bloomington above, you will see that the citations data is organized per quarter, so there are a total of four files. Above, we downloaded the data for the first quarter. Go ahead and download the remaining three files with `wget`.

In this example, we will use three modules, `numpy`, `pandas` and `matplotlib`. If you set up `virtualenv` as described in the *Python tutorial*, the first two of these are already installed for you. To install `matplotlib`, make sure you've activated your `virtualenv` and use `pip`:

```
$ source ~/ENV/bin/activate
$ pip install matplotlib
```

If you are using a different distribution of Python, you will need to make sure that all three of these modules are installed.

Load Data in Pandas

From the same directory where you saved the citations data, let's start the Python interpreter and load the citations data for Q1 2016

```
$ python
>>> from __future__ import division, print_function
>>> import numpy as np
>>> import pandas as pd
>>> import matplotlib.pyplot as plt
>>> data = pd.read_csv('2016-first-quarter-citations.csv')
```

If the first `import` statement seems confusing, take a look at the *Python tutorial*. The next three `import` statements load each of the modules we will use in this example. The final line uses Pandas' `read_csv` function to load the data into a Pandas `DataFrame` data structure.

Working with DataFrames

You can verify that you are working with a `DataFrame` and use some of its methods to take a look at the structure of the data as follows:

```
>>> type(data)
<class 'pandas.core.frame.DataFrame'>
>>> data.index
Int64Index([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,
...
197, 198, 199, 200, 201, 202, 203, 204, 205, 206],
dtype='int64', length=200)
>>> data.columns
Index([u'Citation Number', u'Date Issued', u'Time Issued', u'Location ',
u'District', u'Cited Person Age', u'Cited Person Sex',
u'Cited Person Race', u'Offense Code', u'Offense Description',
u'Officer Age', u'Officer Sex', u'Officer Race'],
dtype='object')
>>> data.dtypes
Citation Number               object
Date Issued                   object
Time Issued                   object
Location                      object
District                      object
Cited Person Age             float64
Cited Person Sex              object
Cited Person Race             object
Offense Code                  object
Offense Description           object
Officer Age                  float64
Officer Sex                   object
Officer Race                  object
dtype: object
>>> data.shape
(200, 15)
```

As you can see from the `columns` field, when the CSV file was read, the header line was used to populate the name of the columns in the `DataFrame`. In addition, you will notice that `read_csv` correctly inferred the data type of some columns like *Age*, but not of others like *Date Issued* and *Time Issued*. `read_csv` is a very customizable function and in general, you can correct issues like this using the `dtype` and `converters` parameters. In this specific case, it makes more sense to combine the *Date Issued* and *Time Issued* columns into a new column containing a time stamp. We will see how to do this shortly.

You can also look at the data itself with the `DataFrame`'s `head()` and `tail()` methods:

```
>>> data.head()
<Output omitted for brevity>
>>> data.tail()
<Output omitted for brevity>
```

In addition to letting you examine your data easily, ‘‘DataFrame‘‘s have methods that help you deal with missing values:

```
>>> data = data.dropna(how='any')
>>> data.shape
```

Adding columns to the data is also easy. Here, we add two columns. First, a datetime column that is a combination of the `Date Issued` and `Time Issued` columns originally in the data. Second, a column identifying what day of the week each citation was given. To understand this example better, take a look at the Python docs for the `strptime` and `strftime` functions in the `datetime` module linked above.

```
>>> from datetime import datetime
>>> data['DateTime Issued'] = data.apply(
...  lambda row: datetime.strptime(row['Date Issued'] + ':' + row['Time␣
↪Issued'], '%m/%d/%y:%I:%M %p'), axis=1
... )
>>> data.columns
>>> data['Day of Week Issued'] = data.apply(
...  lambda row: datetime.strftime(row['DateTime Issued'], '%A'), axis=1
... )
```

Plotting with Matplotlib and NumPy

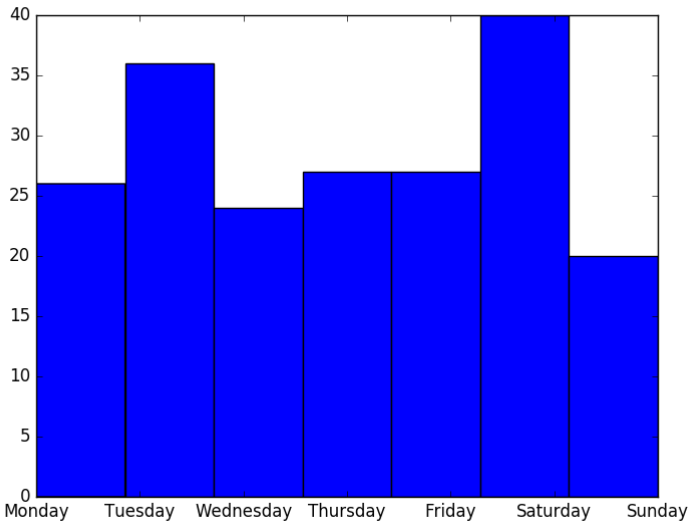
Let's say we want to see how many citations were given each day of the week. We gather the data first:

```
>>> days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
↪'Saturday', 'Sunday']
>>> dow_data = [days.index(dow) for dow in data['Day of Week Issued']]
>>> dow_data
<Output omitted for brevity>
```

Then we use `matplotlib` to plot it:

```
>>> fig = plt.figure()
>>> ax = fig.add_subplot(1, 1, 1)
>>> plt.hist(dow_data, bins=len(days))
>>> plt.xticks(range(len(days)), days)
>>> plt.show()
```

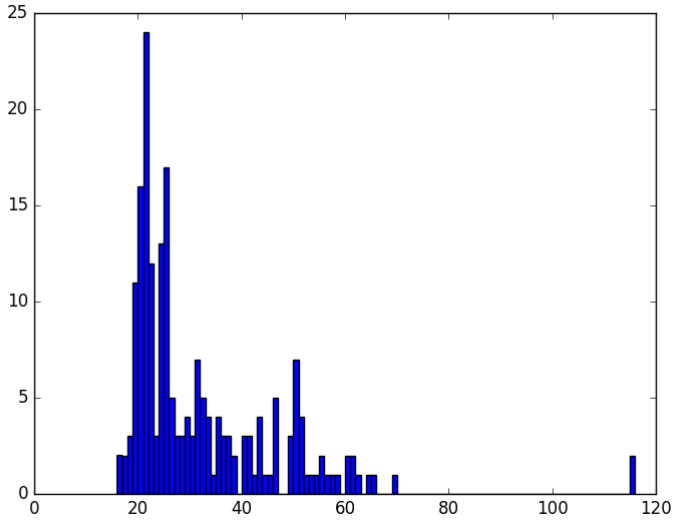
You should see something like this on your screen:



More *DataFrame* Manipulation and Plotting

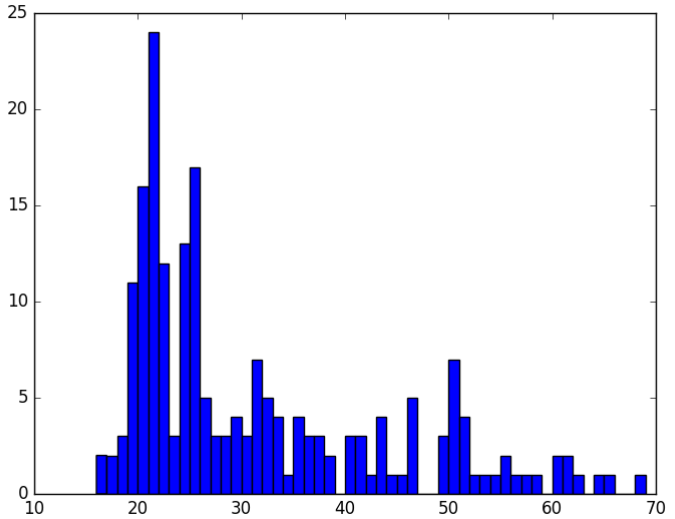
`DataFrame``s and ``numpy` give us other ways to manipulate data. For example, we can plot a histogram of the ages of violators like this:

```
>>> ages = data['Cited Person Age'].astype(int)
>>> fig = plt.figure()
>>> ax = fig.add_subplot(1, 1, 1)
>>> plt.hist(ages, bins=np.max(ages) - np.min(ages))
>>> plt.show()
```



Surprisingly, we see some 116 year-old violators! This is probably an error in the data, so we can remove these data points easily and plot the histogram again:

```
>>> ages = ages[ages < 100]
>>> fig = plt.figure()
>>> ax = fig.add_subplot(1, 1, 1)
>>> plt.hist(ages, bins=np.max(ages) - np.min(ages))
>>> plt.show()
```



Saving Plots to PDF

Oftentimes, you will want to save your `matplotlib` graph as a PDF or an SVG file instead of just viewing it on your screen. For both, we need to create a `figure` and plot the histogram as before:

```
>>> fig = plt.figure()
>>> ax = fig.add_subplot(1, 1, 1)
>>> plt.hist(ages, bins=np.max(ages) - np.min(ages))
```

Then, instead of calling `plt.show()` we can invoke `plt.savefig()` to save as SVG:

```
>>> plt.savefig('hist.svg')
```

If we want to save the figure as PDF instead, we need to use the `PdfPages` module together with `savefig()`:

```
>>> import matplotlib.patches as mpatches
>>> from matplotlib.backends.backend_pdf import PdfPages
>>> pp = PdfPages('hist.pdf')
>>> fig.savefig(pp, format='pdf')
>>> pp.close()
```

Next Steps and Exercises

There is a lot more to working with `pandas`, `numpy` and `matplotlib` than we can show you here, but hopefully this example has piqued your curiosity.

Don't worry if you don't understand everything in this example. For a more detailed explanation on these modules and the examples we did, please take a look at the tutorials below. The `numpy` and `pandas` tutorials are mandatory if you want to be able to use these modules, and the `matplotlib` gallery has many useful code examples.

8.3.2 Summary of Useful Libraries

Numpy

- `http://www.numpy.org/`

According to the Numpy Web page "NumPy is a package for scientific computing with Python. It contains a powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities

Tutorial: `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`

MatplotLib

- `http://matplotlib.org/`

According the the Matplotlib Web page, "matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB®* or Mathematica®†), web application servers, and six graphical user interface toolkits."

Matplotlib Gallery: `http://matplotlib.org/gallery.html`

Pandas

- `http://pandas.pydata.org/`

According to the Pandas Web page, "Pandas is a library library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language."

In addition to access to charts via matplotlib it has elementary functionality for conduction data analysis. Pandas may be very suitable for your projects.

Tutorial: `http://pandas.pydata.org/pandas-docs/stable/10min.html`

Pandas Cheat Sheet: `https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas_Cheat_Sheet.pdf`

8.3.3 Other Useful Libraries

Scipy

- `https://www.scipy.org/`

According to the SciPy Web page, "SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

- NumPy
- IPython
- Pandas
- Matplotlib
- Sympy
- SciPy library

It is thus an agglomeration of useful pacakes and will prbably sufice for your projects in case you use Python.

ggplot

- `http://ggplot.yhathq.com/`

According to the ggplot python Web page ggplot is a plotting system for Python based on R's ggplot2. It allows to quickly generate some plots quickly with little effort. Often it may be easier to use than matplotlib directly.

seaborn

`http://www.data-analysis-in-python.org/t_seaborn.html`

The good library for plotting is called seaborn which is build on top of matplotlib. It provides high level templates for common statistical plots.

- Gallery: http://stanford.edu/~mwaskom/software/seaborn/examples/index.html
- Original Tutorial: http://stanford.edu/~mwaskom/software/seaborn/tutorial.html
- Additional Tutorial: https://stanford.edu/~mwaskom/software/seaborn/tutorial/distributions.html

Bokeh

Bokeh is an interactive visualization library with focus on web browsers for display. Its goal is to provide a similar experience as D3.js

- URL: http://bokeh.pydata.org/
- Gallery: http://bokeh.pydata.org/en/latest/docs/gallery.html

pygal

Pygal is a simple API to produce graphs that can be easily embedded into your Web pages. It contains annotations when you hover over data points. It also allows to present the data in a table.

- URL: http://pygal.org/

Network and Graphs

- igraph: http://www.pythonforsocialscientists.org/t_igraph.html
- networkx: https://networkx.github.io/

REST

- django REST FRamework http://www.django-rest-framework.org/
- flask https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask
- requests https://realpython.com/blog/python/api-integration-in-python/
- urllib2 http://rest.elkstein.org/2008/02/using-rest-in-python.html (not recommended)
- web http://www.dreamsyssoft.com/python-scripting-tutorial/create-simple-rest-web-service-with-python.php (not recommended)
- bottle http://bottlepy.org/docs/dev/index.html
- falcon https://falconframework.org/
- eve http://python-eve.org/
- https://code.tutsplus.com/tutorials/building-rest-apis-using-eve--cms-22961

8.3.4 Other Examples

- *Fingerprint Analysis*

8.4 *cmd* Module

8.4.1 Introduction

The Python *cmd* module is useful for any more involved command-line application. It is used in the Cloudmesh Project, for example, and students in *I524* have found it helpful in their projects. The Python *cmd* module contains a public class, *Cmd*, designed to be used as a base class for command processors such as interactive shells and other command interpreters.

8.4.2 *Hello, World* with *cmd*

This example shows a very simple command interpreter that simply responds to the *greet* command.

In order to demonstrate commands provided by *cmd*, let's save the following program in a file called *helloworld.py*.

```
from __future__ import print_function, division
import cmd


class HelloWorld(cmd.Cmd):
    '''Simple command processor example.'''

    def do_greet(self, line):
        if line is not None and len(line.strip()) > 0:
            print('Hello, %s!' % line.strip().title())
        else:
            print('Hello!')

    def do_EOF(self, line):
        print('bye, bye')
        return True


if __name__ == '__main__':
    HelloWorld().cmdloop()
```

A session with this program might look like this:

```
$ python helloworld.py

(Cmd) help

Documented commands (type help <topic>):
========================================
help
```

```
Undocumented commands:
======================
EOF  greet

(Cmd) greet
Hello!
(Cmd) greet albert
Hello, Albert!
<CTRL-D pressed>
(Cmd) bye, bye
```

The *Cmd* class can be used to customize a subclass that becomes a user-defined command prompt. After you have executed your program, commands defined in your class can be used. Take note of the following in this example:

- The methods of the class of the form *do_xxx* implement the shell commands, with *xxx* being the name of the command. For example, in the *HelloWorld* class, the function *do_greet* maps to the *greet* on the command line.
- The *EOF* command is a special command that is executed when you press CTRL-D on your keyboard.
- As soon as any command method returns *True* the shell application exits. Thus, in this example the shell is exited by pressing CTRL-D, since the *do_EOF* method is the only one that returns *True*.
- The shell application is started by calling the *cmdloop* method of the class.

8.4.3 A More Involved Example

Let's look at a little more involved example. Save the following code in a file called *calculator.py*.

```
from __future__ import print_function, division
import cmd


class Calculator(cmd.Cmd):
    prompt = 'calc >>> '
    intro = 'Simple calculator that can do addition, subtraction,␣
↪multiplication and division.'

    def do_add(self, line):
            args = line.split()
            total = 0
            for arg in args:
                    total += float(arg.strip())
            print(total)

    def do_subtract(self, line):
            args = line.split()
```

```
            total = 0
            if len(args) > 0:
                    total = float(args[0])
            for arg in args[1:]:
                    total -= float(arg.strip())
            print(total)

    def do_EOF(self, line):
            print('bye, bye')
            return True


if __name__ == '__main__':
    Calculator().cmdloop()
```

A session with this program might look like this:

```
$ python calculator.py
Simple calculator that can do addition, subtraction, multiplication and␣
 ↪division.
calc >>> help

Documented commands (type help <topic>):
========================================
help

Undocumented commands:
======================
EOF  add  subtract

calc >>> add
0
calc >>> add 4 5 6
15.0
calc >>> subtract
0
calc >>> subtract 10 2
8.0
calc >>> subtract 10 2 20
-12.0
calc >>> bye, bye
```

- In this case we are using the *prompt* and *intro* class variables to define what the default prompt looks like and a welcome message when the command interpreter is invoked.
- In the *add* and *subtract* commands we are using the *strip* and *split* methods to parse all arguments. If you want to get fancy, you can use Python modules like *getopts* or *argparse* for this, but this is not necessary in this simple example.

8.4.4 Help Messages

Notice that all commands presently show up as undocumented. To remedy this, we can define *help_* methods for each command:

```
from __future__ import print_function, division
import cmd


class Calculator(cmd.Cmd):
      prompt = 'calc >>> '
      intro = 'Simple calculator that can do addition, subtraction,␣
↪multiplication and division.'

      def do_add(self, line):
              args = line.split()
              total = 0
              for arg in args:
                      total += float(arg.strip())
              print(total)

      def help_add(self):
              print('\n'.join([
                      'add [number,]',
                      'Add the arguments together and display the total.'
              ]))

      def do_subtract(self, line):
              args = line.split()
              total = 0
              if len(args) > 0:
                      total = float(args[0])
              for arg in args[1:]:
                      total -= float(arg.strip())
              print(total)

      def help_subtract(self):
              print('\n'.join([
                      'subtract [number,]',
                      'Subtract all following arguments from the first␣
↪argument.'
              ]))

      def do_EOF(self, line):
              print('bye, bye')
              return True


if __name__ == '__main__':
```

```
    Calculator().cmdloop()
```

Now, we can obtain help for the *add* and *subtract* commands:

```
$ python calculator.py
Simple calculator that can do addition, subtraction, multiplication and␣
 ↪division.
calc >>> help

Documented commands (type help <topic>):
========================================
add  help  subtract

Undocumented commands:
======================
EOF

calc >>> help add
add [number,]
Add the arguments together and display the total.
calc >>> help subtract
subtract [number,]
Subtract all following arguments from the first argument.
calc >>> bye, bye
```

8.4.5 Useful Links

- Python Docs
- Python Module of the Week: cmd – Create line-oriented command processors

8.5 CMD5

CMD is a very useful package in python to create command line shells. However it does not allow the dynamic integration of newly defined commands. Furthermore, addition to cmd need to be done within the same source tree. To simplify developping commands by a number of people and to have a dynamic plugin mechnism, we developed cmd5. It is a rewrite on our ealier effords in cloudmesh and cmd3.

8.5.1 Resources

The source code for cmd5 is located in github:

- `https://github.com/cloudmesh/cmd5`

8.5.2 Installation from source

We recommend that you use a virtualenv either with virtualenv or pyenv. This can be either achieved vor virtualenv with:

```
virtualenv ~/ENV2
```

or for pyenv, with:

```
pyenev virtualenv 2.7.13 ENV2
```

Next, you must install cloudmesh_client with:

```
pip install cloudmesh_client
```

Now you need to get two source directories. We assume yo place them in ~/github:

```
mkdir ~/github
cd ~/github
git clone https://github.com/cloudmesh/cmd5.git
git clone https://github.com/cloudmesh/extbar.git
```

The cmd5 repository contains the shell, while the extbar directory contains the sample to add the dynamic commands foo and bar.

To install them simply to the following:

```
cd ~/github/cmd5
python setup.py install
cd ~/github/extbar
python setup.py install
```

8.5.3 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
```

8.5.4 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. This is defined in the setup.py file of your enhancement. The best way to create an enhancement is to take a look at the code in

- `https://github.com/cloudmesh/extbar.git`

Simply copy the code and modify the bar and foo commands to fit yor needs. 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:

```
cloudmsesh/ext/command/COMMANDNAME.py
```

An example for the bar command is presented at:

- `https://github.com/cloudmesh/extbar/blob/master/cloudmesh/ext/command/bar.py`

It shows how simple the command definition is (bar.py):

```
from __future__ import print_function
from cloudmesh_client.shell.command import command
from cloudmesh_client.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 used not to use docopts and argparse was used. However we noticed that for some contributions the lead to commands that were either not properly documented or the developers delivered ambiguous commands that resulted in confusion and wrong ussage by the users. Hence, we do recommend that you use docopts.

The transformation is enabled by the @command decorator that takes also the 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.

8.5.5 Excersise

CMD5.1: Install cmd5 on your computer.
CMD5.2: Write a new command with your firstname as teh command name.
CMD5.3: Write a new command and experiment with docopt syntax and argument interpretation of the dict with if conditions.
CMD5.4: If you have useful extensions that you like us to add by default, please work with us.

8.6 REST with Eve

8.6.1 Overview of REST

REST stands for REpresentational State Transfer. REST is an architecture style for designing networked applications. It is based on stateless, client-server, cacheable communications protocol. Although not based on http, in most cases, the HTTP protocol is used. In contrast to what some others write or say, REST is not a *standard*.

RESTful applications use HTTP requests to:

- post data: while creating and/or updating it,
- read data: while making queries, and
- delete data.

Hence REST uses HTTP for the four CRUD operations:

- Create

- Read
- Update
- Delete

As part of the HTTP protocol we have methods such as GET, PUT, POST, and DELETE. These methods can than be used to implement a REST service. As REST introduces collections and items we need to implement the CRUD functions for them. The semantics is explained in the Table illustrationg how to implement them with HTTP methods.

Table 8.1: IMplementing REST with HTTP methods

URL	GET	PUT	POST	DELETE
`http://../resources/`	List the URIs and perhaps other details of the collection's members.	Replace the entire collection with another collection.	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Delete the entire collection.
`http://../resources/item17`	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	Replace the addressed member of the collection, or if it does not exist, create it.	Not generally used. Treat the addressed member as a collection in its own right and create a new entry within it.	Delete the addressed member of the collection.

Source: `https://en.wikipedia.org/wiki/Representational_state_transfer`

8.6.2 REST and eve

Now that we have outlined the basic functionality that we need, we lke to introduce you to Eve that makes this process rather trivial. IN fact we will provide you with an implementation example that showcases that we can create REST services without writing a single line of code. The code for this is located at `https://github.com/cloudmesh/eve`

Installation

First we havt to install mongodb. The instalation will depend on your operating system. Note that for this example we do not need to integrate mongodb into the system upon reboot. IN fact for us it is better if we can start and stop the services by hand.

On ubuntu, you need to do the following steps:

```
TO BE CONTRIBUTED BY THE STUDENTS OF THE CLASS as homework
```

On windows 10, you need to do the following steps:

```
TO BE CONTRIBUTED BY THE STUDENTS OF THE CLASS as homework, if you
elect Windows 10
```

On OSX you can use homebrew and install it with

```
brew update
brew install mongodb
# brew install mongodb --with-openssl
```

Starting the service

We have provided a convenient Makefile that currently only works for OSX. But you can replicate the steps while looking at the targets we defined in the makefile in a shell program.

```
TODO Provide a shell progra, that runs on all three operating
systems. To be completed by students of the class
```

When using the makefile you can start the services with:

```
make deploy
```

To test the services you can say:

```
make test
```

The program relies on evegenie that we will be added to the repository by executing

```
SOME MAKEFILE TARGET. TO BE COMPLETED BY STUDENT. ITS ALREADY IN MAKEFILE
```

TODO by student add logging

8.7 Python Homework

In this homework you need to complete a few simple Python exercises. Before you begin, please refer to the Python lessons on the course website:

1. *Python Introduction*
2. *Python for Big Data*
3. *Python cmd module*

8.7.1 Exercise 1

Finish implementing the *Calculator* class by adding *multiply* and *divide* methods.

8.7.2 Exercise 2

Create a command-line interpreter, *mycmd*, that has the following commands:

1. deploy
2. kill
3. benchmark
4. report

Each command should simply print out its name to the screen.

8.7.3 Exercise 3

In the *tutorial on Python for big data*, you saw how to plot a histogram of the ages of people who received traffic citations in Q1 2016. Generate the same histogram with the pygal Python module.

8.7.4 Submission

- For **residential students**, please have this ready on your GitHub accounts by class time on Monday, February 20th. We will pick students at random to demonstrate their solutions.
- For **online students**, you don't need to submit anything to us. However, we still strongly encourage you to do these exercises. They will help you get up to speed in Python if you're not familiar with it; or if you are, they might introduce you to some modules that you haven't used before. Of course, if you have any questions about the exercises, please discuss them with the AIs during office hours or on Piazza.

8.8 Python Fingerprint Example

This has moved. Please see the fingerprint matching example in ../../notebooks/index.

9. Ansible

9.1 Ansible I: Simplest Example

9.1.1 Prerequisite

In order to conduct this lesson:

- You can install Ubuntu 16.04 virtual machine on VirtualBox
- You can install software packages via 'apt-get' tool in Ubuntu virtual host
- You already reserved a virtual cluster (with at least 1 virtual machine in it) on some cloud. OR you can use VMs installed in VirtualBox instead.
- You set up SSH credentials and can login to your virtual machines.

9.1.2 What can you do with Ansible?

Humans do maintenance and configurations on computers. Essentially, we specify a list of operations and a list of target machines where the operations apply to. After defining these two critical lists, the creative works are done, and the rest labour can be automated. And this is what Ansible is used for.

Let us develop a sample from scratch, based on this paradigm.

9.1.3 A sample use case

In this example, we are going to use Ansible to install Apache server on our VMs.

- install Ansible tool on your machine first

```
$ sudo apt-get update
$ sudo apt-get install ansible
```

- prepare a working environment for your Ansible

```
$ mkdir ansible-apache
$ cd ansible-apache
```

- next, we are going to create a local configuration file for Ansible.

When you execute Ansible within this folder, this local configuration file is always going to overwrite a system level Ansible configuration. It is in general beneficial to keep custom configurations locally unless you absolutely believe it should be applied system wide. Create a file 'ansible.cfg' in this folder, and fill:

```
[defaults]
hostfile = hosts
```

This local configuration file simple tells that the target machines' names are given in a file named 'hosts'

Note: In your assignments, we choose to use `inventory` instead of `ansible.cfg`. More details will be given when we introduce `ansible-galaxy` in following chapters. At this moment, getting yourself familiar with some concepts of configuring the Ansible environment is good enough.

- specify hosts in the file

You should have accesses to all VMs listed in this file as part of our prerequisites. Create and edit file 'hosts':

```
[apache]
<server_ip> ansible_ssh_user=<server_username>
```

The name 'apache' in the brackets defines the server group name. We will use this name to refer to all server items in this group next. Fill in IP addresses of the virtual machines you launched in your VirtualBox and fire up these VMs in you VirtualBox.

- compose a playbook

A playbook tells Ansible what to do. it uses YAML Markup syntax. Create and edit a file with a proper name e.g. apache.yml as follow:

```
---
- hosts: apache #comment: apache is the group name we just defined
  become: yes #comment: this operation needs privilege access
  tasks:
```

```
  - name: install apache2 # text description
    apt: name=apache2 update_cache=yes state=latest
```

This block defines the target VMs and operations(tasks) need to apply. you may wonder what 'apt' means. here comes the concept of Ansible modules in next paragraph.

9.1.4 The concept of modules

'apt' is the module used in our sample playbook. it installs packages on Ubuntu for us.

Ansible relies on various kinds of modules to fulfil tasks on the remote servers. These modules are developed for particular tasks and take in related arguments. For instance, when we use 'apt' module, we certainly need to tell which package we intend to install. That is why we provide a value for the 'name' argument.

9.1.5 Run you playbook

In the same folder, execute

```
ansible-playbook apache.yml --ask-sudo-pass
```

After a successful run, open a browser and fill in your server IP. you should se a 'It works!' Apache2 Ubuntu default page. Make sure the security policy on your cloud opens port 80 to let the HTTP traffic go through.

Ansible playbook can have more complex and fancy structure and syntaxes. Go explore! this sample is based on: `https://www.digitalocean.com/community/tutorials/how-to-configure-apache-using-ansible-on-ubuntu-14-04`

We are going to offer an advanced Ansible in next chapter.

9.2 Ansible II: Roles

In this example, we are going to install R package onto your cloud VMs. R is a useful statistic programing language commonly used in many scientific and statistics computing projects, maybe also the one you chose for this class.

In addition to last basic example, we are going to illustrate the concept of Ansible Roles, install source code through Github, and make use of variables. These are key features you will find useful in your project deployments.

We are going to use a top-down fashion in this example. We first start from a playbook that is already good to go. You can execute this playbook (don't do it yet) to get R installed in your remote hosts. We then further complicate this concise playbook by introducing functionalities to do the same tasks but in different ways. Although these detours are not necessary in this simply case, it helps you grasp the power of Ansible and ease your life when they are needed in your real projects.

9.2.1 Prerequisite

- Finished Ansible Lesson I
- Have VMs reserved on cloud and SSH Key setup

9.2.2 A completed playbook

From the earlier example, we already can compose a playbook 'example.yml' that install a software package, for example:

```
---
- hosts: R_hosts
  become: yes
  tasks:
    - name: install the R package
      apt: name=r-base update_cache=yes state=latest
```

the hosts are defined in a file 'hosts', which we configured in 'ansible.cfg':

```
[R_hosts]
<cloud_server_ip> ansible_ssh_user=<cloud_server_username>
```

Note: In your assignments, we choose to use `inventory` instead of `ansible.cfg`. More details will be given when we introduce `ansible-galaxy` in following chapters. At this moment, getting yourself familiar with some concepts of configuring the Ansible environment is good enough.

This should get the installation job done. But we are going to extend it via new features next.

9.2.3 Introducing Roles

Role is an important concept used very often in large Ansible projects. You divide a series of tasks into different groups. Each group corresponds to certain role of the whole project.

For example, if your project is to deploy a web site, you may need to install the back end database, the web server that responses HTTP requests and the web application itself. They are three different roles and should carry out their own installation and configuration tasks.

Even though we only need to install the R package in this example, we can still do it by defining a role 'r'. Modify your 'example.yml' to be:

```
---
- hosts: R_hosts

  roles:
    - r
```

and create directory structure in your top project directory:

```
$ mkdir -p roles/r/tasks
$ touch roles/r/tasks/main.yml
```

edit this 'main.yml' to be:

```
---
- name: install the R package
  apt: name=r-base update_cache=yes state=latest
  become: yes
```

You probably already get the point. We take the 'tasks' section out of the one-for-all example.yml and re-organize them into roles. Each role specified in example.yml should have its own directory under roles/ and the tasks need be done by this role is listed in a file 'tasks/main.yml' as above.

9.2.4 Install source code from Github

Although R can be installed through the OS package manager (apt-get etc.), the software used in your projects may not. Many research projects are available by Git instead. Here we are going to show you how to install packages from their Git repositories. Instead of directly executing the module 'apt', we pretend Ubuntu does not provide this package and you have to find it on Git. The source code of R can be found at `https://github.com/wch/r-source.git`. We are going to clone it to a remote VM's hard drive, build the package and install the binary there.

To do so, we need a few new Ansible modules. You may remember from last example that Ansible modules assist us to do various kinds of jobs when fed correct arguments. No surprise, Ansible has a module 'git' to take care of git-related works, and a 'command' module to run shell commands. Let's modify 'roles/r/tasks/main.yml' to be:

```
---
- name: get R package source
  git:
    repo: https://github.com/wch/r-source.git
    dest: /tmp/R

- name: build and install R
  become: yes
  command: chdir=/tmp/R "{{ item }}"
  with_items:
    - ./configure
    - make
    - make install
```

The role 'r' carries out its two tasks now. One to clone the R source code into /tmp/R, the other uses a series of shell commands to build and install the packages.

Warning: Note that the commands executed by the second task may not be available on a fresh VM image. But the point of this example is to show an alternative way to install packages, so we conveniently assume conditions are all met.

9.2.5 Using variables in a separate file

We typed several string constants in our Ansible scripts so far. In general, it is a good practice to give these values names and use them by referring to their names. This way, you complex Ansible project can be less error prone. Create a file in the same directory, and name it 'vars.yml':

```
---
repository: https://github.com/wch/r-source.git
tmp: /tmp/R
```

Accordingly, we will update our 'example.yml':

```
---
- hosts: R_hosts
  vars_files:
    - vars.yml
  roles:
    - r
```

As shown, we specify a 'vars_files' telling the script that the file 'vars.yml' is going to supply variable values, whose keys are denoted by Double curly brackets like in 'roles/r/tasks/main.yml':

```
---
- name: get R package source
  git:
    repo: "{{ repository }}"
    dest: "{{ tmp }}"

- name: build and install R
  become: yes
  command: chdir="{{ tmp }}" "{{ item }}"
  with_items:
    - ./configure
    - make
    - make install
```

9.2.6 Summarize

Now, just edit the 'hosts' file with your target VMs' IP addresses and execute the playbook.

You should be able to extend the Ansible playbook for your project. Configuration tools like Ansible are important components to master the cloud environment. There is much to explore and it's worth it.

9.3 Ansible III: Ansible Galaxy

By finishing the first two chapters, you should be able to compose Ansible projects to install, configure or do other maintenances on your software packages. We introduced the powerful component `Roles` in the previous chapters, and emphasized the concepts of modularize and re-usability. With these preparations, we are ready to start working on Ansible Galaxy.

Think Ansible Galaxy as of an marketplace, where developers can share Ansible Roles to complete their system administration tasks. Roles exchanged in Ansible Galaxy community need to follow common conventions so that all participants know what to expect. We will illustrate details in this chapter.

It is good to follow the Ansible Galaxy standard during your development assignment as much as possible, however, you will submit your assignments to this class's repository not the global Galaxy community.

9.3.1 Ansible Galaxy helloworld

Let us start with a simplest case: We will build an Ansible Galaxy project. This project will install the Emacs software package on your localhost as the target host. It is a "helloworld" project only meant to get us familiar with Ansible Galaxy project structures.

create the directory

Setup your submission directory after you clone and rebased with `https://github.com/cloudmesh/sp17-i524`:

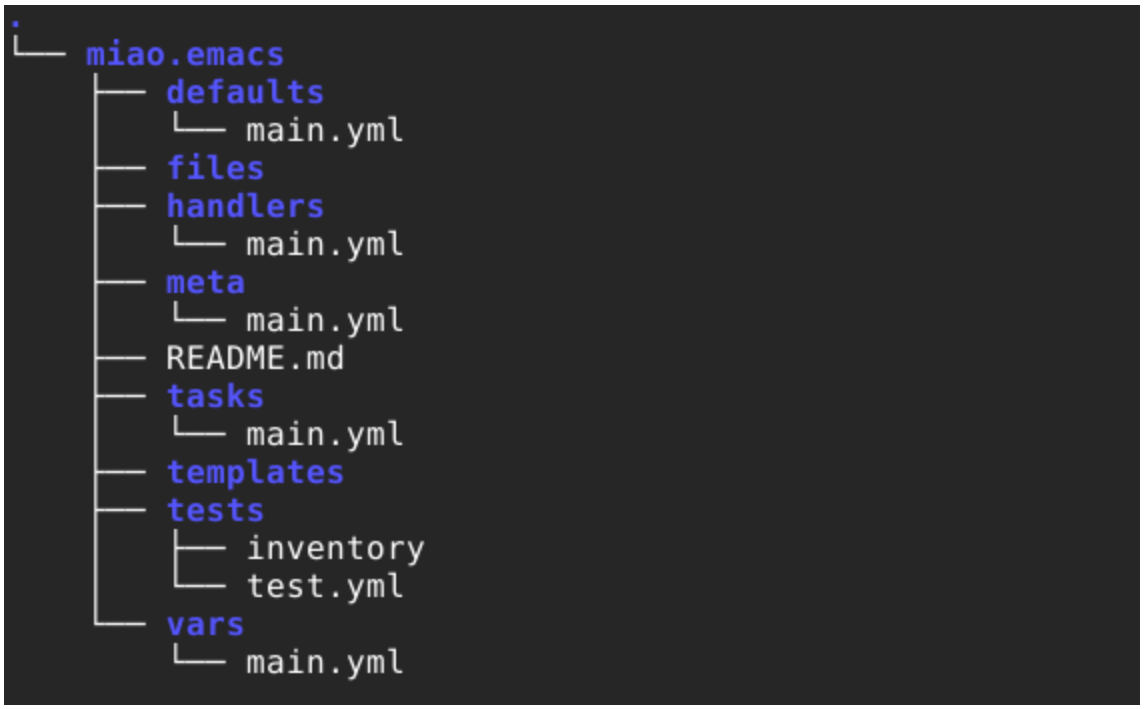
```
$ git rebase upstream/master
$ ./setup galaxy <your HID>
```

It will create a folder named after your HID inside directory galaxy/. Your Galaxy related assignments will be completed and submitted there. Go ahead and create files `README.md`, `playbook.yml`, `inventory` and a subdirectory `roles/` then. playbook.yml is your project playbook. It should perform the Emacs installation task by executing the corresponding role you will develop in the folder 'roles/'. The only difference is that we will construct the role with the help of ansible-galaxy this time.

Now, let ansible-galaxy initialize the directory structure for you:

```
$ cd roles
$ ansible-galaxy init <to-be-created-role-name>
```

The naming convention is to concatenate your name and the role name by a dot. Here is how it looks like:



fill in information

Let us fill in information to our project. There are several `main.yml` files in different folders, and we will illustrate their usages.

- defaults and vars

These folders should hold variables key-value pairs for your playbook scripts. We will leave them empty in this example.

- files

This folder is for files need to be copied to the target hosts. Data files or configuration files can be specified if needed. We will leave it empty too.

- templates

Similar missions to files/, templates is allocated for template files. Keep empty for a simple Emacs installation.

- handlers

This is reserved for services running on target hosts. For example, to restart a service under certain circumstance.

- tasks

This file is the actual script for all tasks. You can use the role you built previously for Emacs installation here:

```
---
- name: install Emacs on Ubuntu 16.04
  become: yes
  package: name=emacs state=present
```

- meta

Lastly, we provide necessary metatdata for our Ansible Galaxy project for shipping:

```
---
galaxy_info:
  author: <you name>
  description: emacs installation on Ubuntu 16.04
  license:
    - MIT
  min_ansible_version: 2.0
  platforms:
    - name: Ubuntu
      versions:
        - xenial
  galaxy_tags:
    - development

dependencies: []
```

test it out

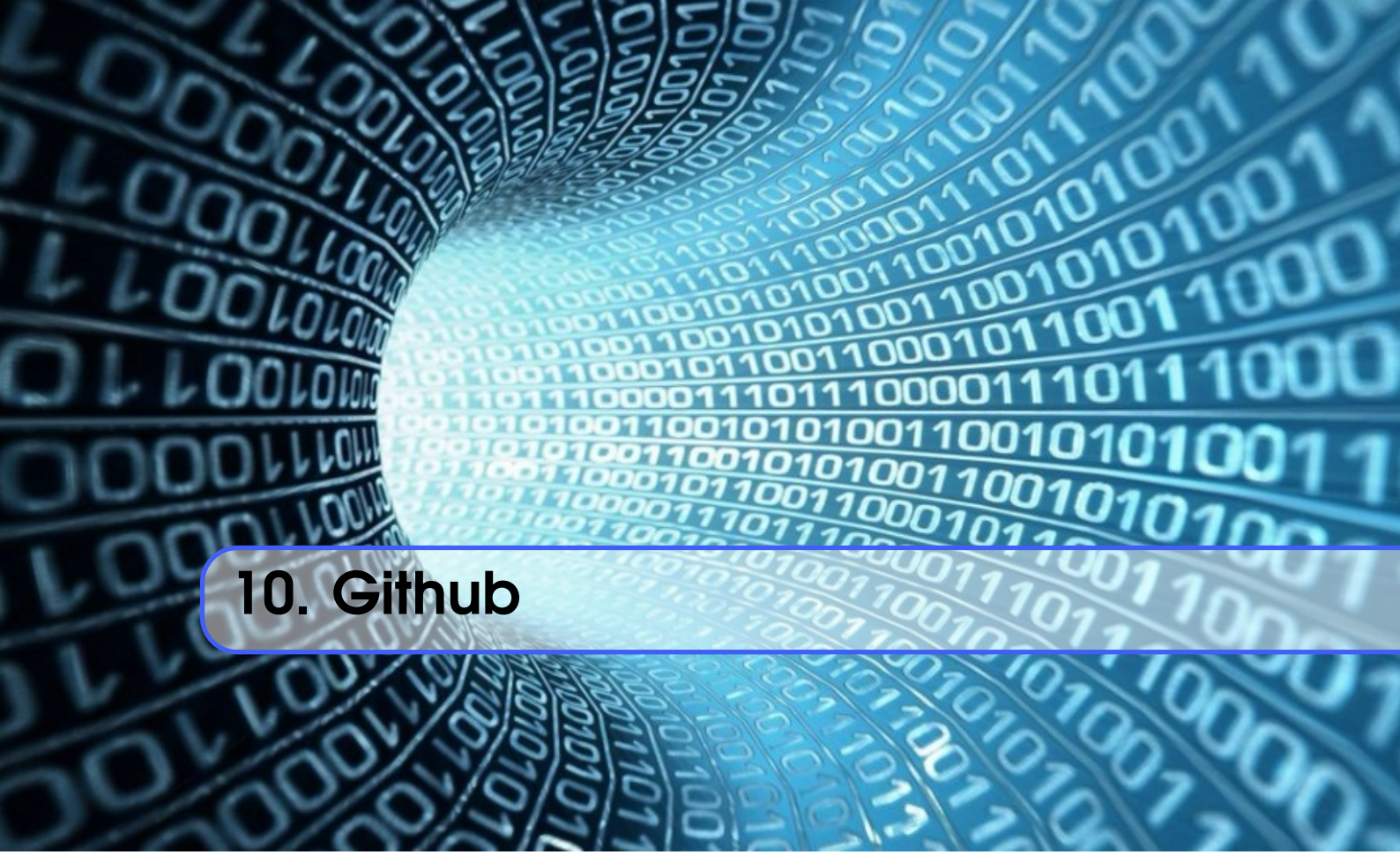
You have your Ansible Galaxy role ready now. To test it as a user, go to your HID directory and edit the other two files `inventory` and `playbook.yml`, which are already generated for you in directory `tests` by the script:

After running this playbook, you should have Emacs installed on localhost.

9.3.2 A Complete Ansible Galaxy Project

We are going to use ansible-galaxy to setup a sample project. This sample project will:

- use a cloud cluster with multiple VMs
- deploy Apache Spark on this cluster
- install a particular HPC application
- prepare raw data for this cluster to process
- run the experiment and collect results



10. Github

10.1 Github

Github is a code repository that allows the development of code in a distributed fashion. There are many good tutorials about github.

Some of them can be found on the github Web page. An interactive tutorial is for example available at

- `https://try.github.io/`

A more extensive list of tutorials can be found at

- `https://help.github.com/articles/what-are-other-good-resources-for-learning-git-and-github`

10.1.1 Video Lectures on Github

The github foundation has a number of excellent videos about git. If you are unfamiliar with git and you like to watch videos in addition to reading the docuentation we recommend these videos

- `https://www.youtube.com/user/GitHubGuides/videos`

Below we introduce some heavily used concepts:

Upload Key

Before you fork a repository, you need to upload a public key in order to access your repository. See lesson *Generate a SSH key* before you upload. Copy the contents of your

`.ssh/id_rsa.pub` file and add them to you github keys.

- `https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/`

Fork

Forking is the first step to contributing to projects on GitHub. Forking allows you to copy a repository and work on it under your own account. Next, creating a branch, making some changes, and offering a pull request to the original repository, rounds out your contribution to the open source project.

- `https://www.youtube.com/watch?v=5oJHRbqEofs`

Rebase

With Git Rebase allows you to integrate your code easily into a branch. Rebase allows you to take all of the work on your branch and relocate it to another point in your repository's history, encouraging you to branch early and often without the fear of messy merges or missing out on new development.

When you start editing your project, you diverge from the original version. During your developing, the original version may be updated, or other developers may have some of their branches implementing good features that you would like to include in your current work. That is when 'Rebase' becomes useful. When you 'Rebase' to certain points, could be a newer Master or other custom branch, consider you graft all your on-going work right to that point.

Rebase may fail, because some times it is impossible to achieve what we just described (conflicts!). For example, you and the to-be-rebased copy both edited some common text section. Once this happens, human intervention would be the only solution.

- `https://www.youtube.com/watch?v=SxzjZtJwOgo`

Remote

Collaborating with others involves managing these remote repositories and pushing and pulling data to and from them when you need to share work. Managing remote repositories includes knowing how to add remote repositories, remove remotes that are no longer valid, manage various remote branches and define them as being tracked or not, and more.

Though out this semester, you will typically work on two 'remote' repos. One is the office class repo, and another is the repo you forked from the class repo. The class repo is used as the centralized, authority and final version of all student submissions. The repo under your own Github account is for your personal storage. Only when you feel good about your personal repo, you should let us know that you want us to merge your repo into the class repo (by submitting a pull request as illustrated right next.)

- `https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes`

Pull Request

Pull requests are a means of starting a conversation about a proposed change back into a project. We'll be taking a look at the strength of conversation, integration options for fuller information about a change, and cleanup strategy for when a pull request is finished.

- `https://www.youtube.com/watch?v=d5wpJ5VimSU`

Checkout

Change where and what you're working on with the checkout command. Whether we're switching branches, wanting to look at the working tree at a specific commit in history, or discarding edits we want to throw away, all of these can be done with the checkout command.

- `https://www.youtube.com/watch?v=HwrPhOp6-aM`

Branch

Branches are an excellent way to not only work safely on features or experiments, but they are also the key element in creating Pull Requests on GitHub. Let's take a look at why we want branches, how to create and delete branches, and how to switch branches in this episode.

- `https://www.youtube.com/watch?v=H5GJfcp3p4Q`

Merge

Once you know branches, merging that work into master is the natural next step. Find out how to merge branches, identify and clean up merge conflicts or avoid conflicts until a later date. Lastly, we'll look at combining the merged feature branch into a single commit and cleaning up your feature branch after merges.

- `https://www.youtube.com/watch?v=yyLiplDQtf0`

GUI

Using Graphical User Interfaces can supplement your use of the command line to get the best of both worlds. GitHub for Windows and GitHub for Mac allow for switching to command line, ease of grabbing repositories from GitHub, and participating in a particular pull request. We'll also see the auto-updating functionality helps us stay up to date with stable versions of Git on the command line.

- `https://www.youtube.com/watch?v=BMYOs5jflGE`

Windows

This is a quick tour of GitHub for Windows. It offers GitHub newcomers a brief overview of what this feature-loaded version control tool and an equally powerful web application can do for developers, designers, and managers using Windows in both the open source and commercial software worlds. More: `http://windows.github.com`

- https://www.youtube.com/watch?v=YBbkvCrfDSo

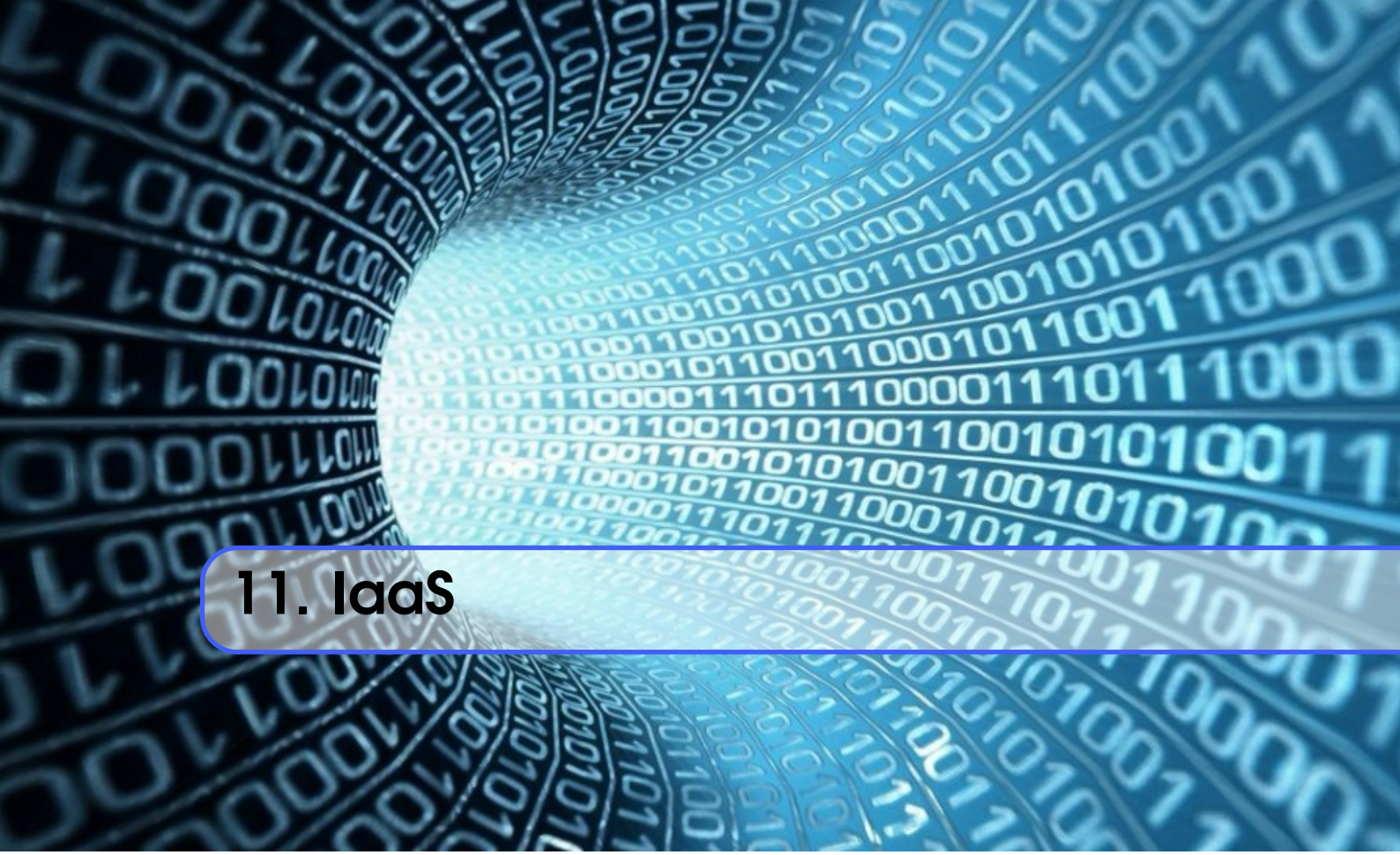
10.1.2 Exercise

Github.1: How do you set your favorite editor as a default with github config

Github.2: What is the difference between merge and rebase?

Github.3: Assume you have made a change in your local fork, however other users have since committed to the master branch, how can you make sure your commit works off from the latest information in the master branch?

Github.4: Find a spelling error in theo Web page or a contribution and create a pull request for it.



11. IaaS

11.1 Cloudmesh Client on Ubuntu 16.04

In this class we will be accessing for several projects multiple clouds. This could pose a significant development issue for some in the class. to simplify access we introduce you to one method that we recommend for this class. We will be using a project called cloudmesh_client that allows users to easily access multiple clouds from the commandline. Switching between clouds can be achieved with one variable. The advantages include:

- access to another cloud in case of failure
- relocation of experiments to see if they are robust
- performance benchmarking to compare clouds.

11.1.1 Installation

First, let us install cloudmesh client using pip and make sure you install the latest updates. Install cloudmesh client using pip

```
$ pip install -U cloudmesh_client
```

In order to make sure cloudmesh is running properly, enter cm in your command line. It will show a terminal in the following way.

```
$ cm
```

11.1.2 Setting Up

Now cloudmesh is installed locally. In order to run a virtual machine in chameleon cloud, there are few configurations that have to be done. You can find the configuration information in the following location.

`http://cloudmesh.github.io/client/configuration.html`

After setting up cloudmesh client locally, the yaml file can be opened by running the following command. You can use vim or vi instead of emacs to run this.

```
$ emacs ~/.cloudmesh/cloudmesh.yaml
```

examples:

```
$ vim ~/.cloudmesh/cloudmesh.yaml
$ vi ~/.cloudmesh/cloudmesh.yaml
$ gedit ~/.cloudmesh/cloudmesh.yaml
```

First the profile section must be updated as follows:

```
profile:
firstname: TBD
lastname: TBD
email: TBD
user: TBD
```

example configuration:

```
profile:
     firstname: Vibhatha
     lastname: Abeykoon
     email: vibhatha@gmail.com
     user: vibhatha
```

This must be filled when working on Cloudmesh set up. And this can be found in the configuration file in cm- yaml file.

11.1.3 Setting Up Chameleon Cloud

In the cloudmesh.yaml file, set chameleon cloud as the active cloud as shown below. Locate the attribute value in the:

```
active:
 - chameleon
```

Go to the following link and you can find the information regarding, the chameleon cloud setup.

`http://cloudmesh.github.io/client/configuration.html#chameleon-cloud`

The following parameters has to be replaced with corresponding values:

```
OS_PASSWORD: TBD
OS_TENANT_NAME: TBD
OS_TENANT_ID: TBD
OS_PROJECT_NAME: TBD
OS_USERNAME: TBD
```

Make sure you are following the above url. And after replacing all the TBD values, the configuration should look like as follows.

example configuration:

```
OS_PASSWORD: NOTMYPASSWORD
OS_TENANT_NAME: CH-818664
OS_TENANT_ID: CH-818664
OS_PROJECT_NAME: CH-818664
OS_USERNAME: vibhatha
```

Tip: If you don't want to put your cloud password in the yaml file, you can put `read` instead of the password in `OS_PASSWORD` field. In this way, every time you need to access the cloud, you will type in password.

Make sure the TENANT_NAME: CH-818664. You must be a member of the project in the Chameleon cloud, in order to gain access to the virtual machines.

Note: Replace all TBD values with correct values (only in profile section and chameleon cloud section).

`http://cloudmesh.github.io/client/configuration.html#chameleon-cloud`

11.1.4 Preaparing for Chameleon Access

To create a virtual machine we need first to make sure we start it on chamelon cloud. This we need to set the default cloud to be chameleon with the following command:

```
$ cm default cloud=chameleon
$ cm register profile
$ cm default user=YOURUSERNAME
```

Information about the configuration of cloudmesh can be retrieved by the following command:

```
$ cm info
```

Next we need to add the ssh key to the cloudmesh database by running the following command. Make sure you have already generated a ssh key with ssh-keygen. The command will add the default id_rsa.pub key to a local database:

```
$ cm key add --ssh
```

Not that the key is in our local cloudmesh database, we need to upload it to all active clouds. As we have just one active cloud it will uploade the key to the chamelon cloud once you execute the command:

```
$ cm key upload
```

Furthermore, we must be able to communicate with the virtualmachines. To communicate which ports we use we execute the secgroup command. To just use the defaults we execute the command:

```
$ cm secgroup upload
```

To see the details of the secgropus please use the command:

```
$ cm secgroup list
```

11.1.5 Boot Virtual Machine

Run the following command to boot the virtual machine:

```
$ cm vm boot
```

To see all vms just use the command:

```
$ cm vm list
```

11.1.6 Login to the vm

To login to the vm you need to have a publicly available (floating) ip. This can be achieved with the command:

```
$ cm vm ip assign
```

You can after this command has succeed login to the vm with the command:

```
$ cm vm ssh
```

After a successful launch it will show a similar console as shown below:

```
cc@hostname$-
```

Warning: Many errors could occur that are unrelated to cloudmesh client. Such errors could include network interruptions, resource starvation of cloudmesh, while either no vms can be started, they are out of ip addresses, ir they have a maintenance day. Please do not blame cloudmesh for such issues and explore first if they originate through such issues.

Step 7 : Remove Virtual Machine

To delete a virtual machine, run the following command (exit the vm first if necessary):

```
$ cm vm delete <name_of_vm>
```

Example:

```
$ cm vm delete vibhatha-001
```

To delete multiple virtual machines, run the following command:

```
$ cm vm delete <name_of_vm>*
```

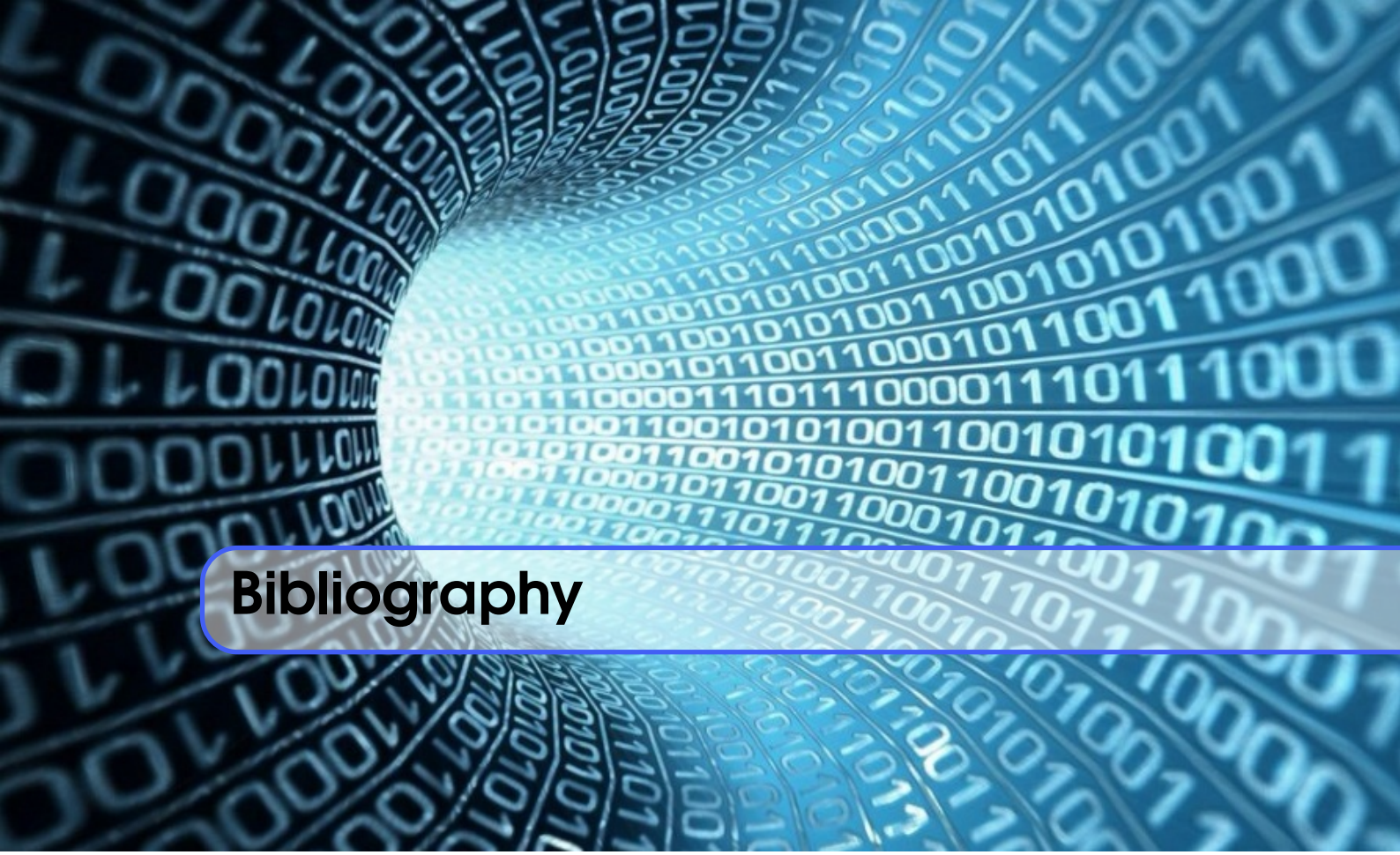
or with `--all` option:

```
$ cm vm delete --all
```

It is important that you delete or terminate the vm after you are done as chameleon cloud has a limited set of resources. we recommend that you do not keep a vm up for more than 6 hours. Please be aware when you delete a vm everything on that vm is deleted. hence we recommend you to make appropriate backups of the content in the vm and have scripts via ansible to recreate your softwarstack.

11.1.7 Exercise

cloudmesh.1: install cloudmesh, create a vm and delete it



Bibliography

[1] Business process execution language. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Business_Process_Execution_Language.

[2] Apache ode. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Apache_ODE.

[3] Ode website. Web Page. Accessed: 2017-2-11. URL: http://ode.apache.org/.

[4] ActiveBPEL. Communicating with the activebpel server administration interface via web services. Web Page, February 2017. Accessed: 2017-02-12. URL: http://www.activevos.com/content/developers/education/sample_active_bpel_admin_api/doc/index.html.

[5] Apache Software Foundation. Apache Airavata. Web page, 2016. Accessed: 2017-2-22. URL: http://airavata.apache.org.

[6] UltraScan Project. UltraScan Analysis Software. Web page, April 2015. Accessed: 2017-2-22. URL: http://ultrascan.uthscsa.edu.

[7] Indiana University, Apache Airavata, XSEDE, and NSF. SEAGrid Portal. Web page. Accessed: 2017-2-22. URL: https://seagrid.org.

[8] Apache Software Foundation. GenApp - Apache Airavata - Apache Software Foundation. Web page, August 2014. Accessed: 2017-2-22. URL: https://cwiki.apache.org/confluence/display/AIRAVATA/GenApp.

[9] Pegasus. Web Page. Accessed:2/3/2017. URL: https://pegasus.isi.edu/.

[10] Kepler Project. The kepler project. Web Page, February 2017. Accessed: 2017-02-12. URL: https://kepler-project.org.

[11] Wikipedia. Swift (programming language). Web Page, February 2017. Accessed: 2017-02-23. URL: https://en.wikipedia.org/wiki/Swift_(programming_language).

[12] Taverna. Web Page. URL: https://taverna.incubator.apache.org/introduction/.

[13] *Taverna Workflows: Syntax and Semantics*, IEEE, December 2007. URL: http://ieeexplore.ieee.org/document/4426917/, doi:10.1109/E-SCIENCE.2007.71.

[14] Triana documentation. Web Page, 2012. Accessed: 2017-2-20. URL: http://www.trianacode.org/.

[15] Trident tutorial. Web Page. Accessed: 2017-1-24. URL: http://storm.apache.org/releases/0.10.1/Trident-tutorial.html.

[16] Trident api overview. Web Page. Accessed: 2017-1-24. URL: http://storm.apache.org/releases/1.0.0/Trident-API-Overview.html.

[17] bioKepler. What is biokepler. WebPage. URL: http://www.biokepler.org/faq#what-is-biokepler.

[18] bioKepler. Demo workflow. WebPage. URL: http://www.biokepler.org/userguide#demos.

[19] Weizhong Li. Introduction to bioactors. In bioKepler, editor, *bioKepler Tools and Its Applications*, number 1 in 1st Workshop on bioKepler Tools and Its Applications, 31. San Diego Supercomputer Center 10100 Hopkins Dr, La Jolla, CA 92093, September 2012. UCSD;SDSC, bioKepler. URL: http://www.biokepler.org/sites/swat.sdsc.edu.biokepler/files/workshops/2012-09-05/slides/2012-09-05-02-Li.pdf.

[20] About ansible galaxy. Web Page. Accessed: 2017-2-06. URL: https://galaxy.ansible.com/intro/.

[21] Michael Heap. *Ansible From Beginner to Pro*. apress, 2016.

[22] GitHub ansible/ galaxy. Web Page. Accessed: 2017-2-06. URL: https://github.com/ansible/galaxy/.

[23] Project Jupyter. Web page. URL: http://www.jupyter.org.

[24] GitHub - jupyter/jupyter: Jupyter metapackage for installation, docs and chat. Web page. URL: https://github.com/jupyter/jupyter.

[25] GitHub - jupyter/notebook: Jupyter Interactive Notebook. Web page. URL: https://github.com/jupyter/notebook.

[26] The
Jupyter notebook — Jupyter Notebook 5.0.0.dev documentation. Web page. URL: https://jupyter-notebook.readthedocs.io/en/latest/index.html.

[27] IPython. Web page, February 2017. Page Version ID: 767266837. URL: https://en.wikipedia.org/w/index.php?title=IPython&oldid=767266837.

[28] What is the Jupyter Notebook? — Jupyter Notebook 5.0.0.dev documentation. Web page.

[29] Derek G. Murray, Frank McSherry, Rebecca Isaacs, Michael Isard, Paul Barham, and Martin Abadi. Naiad: a timely dataflow system. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, SOSP '13, 439–455. New York, NY, USA, 2013. ACM. Accessed: 2017-1-27. URL: http://doi.acm.org/10.1145/2517349.2522738, doi:10.1145/2517349.2522738.

[30] Chandu Thekkath. Naiad - microsoft research. webpage, October 2011. Accessed: 2017-1-27. URL: https://www.microsoft.com/en-us/research/project/naiad/.

[31] Mohammad Islam, Angelo K Huang, Mohamed Battisha, Michelle Chiang, Santhosh Srinivasan, Craig Peters, Andreas Neumann, and Alejandro Abdelnur. Oozie: towards a scalable workflow management system for hadoop. In *Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*, 4. ACM, 2012.

[32] Why Oozie? \textbar Just a simple Hadoop DBA. Web Page. URL: https://prodlife.wordpress.com/2013/12/09/why-oozie/.

[33] Oozie - Apache Oozie Workflow Scheduler for Hadoop. Web Page. URL: http://oozie.apache.org/.

[34] Tez. Web Page, 2017. URL: https://hortonworks.com/apache/tez.

[35] Curoverse, Inc. Arvados | Open Source Big Data Processing and Bioinformatics. Web Page, 2016. Accessed: 2017-2-22. URL: https://arvados.org.

[36] Curoverse, Inc. Arvados | Introduction to Crunch. Web Page, 2016. Accessed: 2017-2-22. URL: http://doc.arvados.org/user/tutorials/intro-crunch.html.

[37] Cascading. Web Page, 2017. URL: http://www.cascading.org/projects/cascading/.

[38] Hugo Hiden, Paul Watson, Simon Woodman, and David Leahy. E-science central: cloud-based e-science and its application to chemical property modelling. In *University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1227*. University of Newcastle upon Tyne, November 2010.

[39] Escience central overview. Web Page. Accessed: 2017-1-24. URL: https://bitbucket.org/digitalinstitute/esciencecentral/.

[40] 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/.

[41] 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.

[42] Google. Cloud dataflow. WebPage. URL: https://cloud.google.com/dataflow/.

[43] 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.

[44] Apachi nifi. Web Page. URL: https://nifi.apache.org.

[45] Apache nifi (aka hdf) data flow across data center. Web Page. URL: https://community.hortonworks.com/articles/9933/apache-nifi-aka-hdf-data-flow-across-data-center.html.

[46] 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.

[47] *Integration Made Easy - Jitterbit*. Jitterbit,Inc., 1 Kaiser Plaza Suite 701 Oakland, CA 94612. Accessed: 2017-1-26. URL: http://www.jitterbit.com/Files/Product/Jitterbit-General-Datasheet.pdf.

[48] Jitterbit. Data integration and etl | jitterbit | integrate data from any source. Web Page. accessed: 2017-1-26. URL: https://www.jitterbit.com/etl-data-integration/.

[49] *Technical Overview - Jitterbit*. Jitterbit,Inc, 2011. Accessed: 2017-1-26. URL: http://www.jitterbit.com/Files/Product/JitterbitTechnicalOverview.pdf.

[50] Pentaho. webpage. URL: https://en.wikipedia.org/wiki/Pentaho.

[51] Any analytics, any data, simplified. webpage. URL: http://www.pentaho.com/product/product-overview.

[52] Evan R Sparks, Shivaram Venkataraman, Tomer Kaftan, Michael J Franklin, and Benjamin Recht. Keystoneml: optimizing pipelines for large-scale advanced analytics. *arXiv preprint arXiv:1610.09451*, 2016.

[53] Evan Sparks and Shivaram Venkataraman. Building Large Scale Machine Learning Applications with Pipelines-(Ev.... URL: https://www.slideshare.net/SparkSummit/evan-sparks.

[54] Alex Woodie. Spark Gets New Machine Learning Framework: KeystoneML. URL: https://www.datanami.com/2015/05/26/spark-gets-new-machine-learning-framework-keystoneml/.

[55] Apache mahout. Web Page. URL: http://mahout.apache.org/.

[56] Distributed machine learning. Web page, feb 2017. Accessed 25-feb-2017. URL: http://www.mlbase.org/.

[57] Abdul Ghaffar Shoro and Tariq Rahim Soomro. Big data analysis: apache spark perspective. *Global Journal of Computer Science and Technology*, 2015. Accessed: 2017-1-21. URL: http://www.computerresearch.org/index.php/computer/article/view/1137/1124.

[58] E. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J. Gonzalez, J. Gonzalez, M. Franklin, M. I. Jordan, and T. Kraska. Mli: an api for distributed machine learning. In *MLI: An API for Distributed Machine Learning*. International Conference on Data Mining, 2013.

[59] T. Kraska, A. Talwalkar, J.Duchi, R. Griffith, M. Franklin, and M.I. Jordan. Mlbase: a distributed machine learning system. In *MLbase: A Distributed Machine Learning System*. Conference on Innovative Data Systems Research, 2013.

[60] A. Talwalkar, T. Kraska, R. Griffith, J. Duchi, J. Gonzalez, D. Britz, X. Pan, V. Smith, E. Sparks, A. Wibisono, M. J. Franklin, and M. I. Jordan. Mlbase: a distributed machine learning wrapper. In *MLbase: A Distributed Machine Learning Wrapper*. Big Learning Workshop at NIPS, 2012.

[61] Datafu. Web Page. Accessed:1/16/2017. URL: https://datafu.incubator.apache.org/.

[62] R: what is r? Web Page. Accessed: 2017-1-26. URL: https://www.r-project.org/about.html.

[63] Vignesh Prajapati. *Big data analytics with R and Hadoop*. Packt Publishing, 2013. ISBN 9781782163282.

[64] G. Ostrouchov, W.-C. Chen, D. Schmidt, and P. Patel. Web Page, 2012. URL: http://r-pbd.org/.

[65] Robert C. Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean YH Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004. URL: http://dx.doi.org/10.1186/gb-2004-5-10-r80, doi:10.1186/gb-2004-5-10-r80.

[66] About bioconductor. Web Page. Accessed: 2017-02-10. URL: https://www.bioconductor.org/about/.

[67] ImageJ. Imagej introduction. Web Page, February 2017. Accessed: 2017-02-12. URL: https://imagej.nih.gov/ij/docs/intro.html.

[68] web page. accessed 2017-02-13. URL: http://opencv.org/.

[69] web page. accessed 2017-02-13. URL: http://opencv.org/opencv-3-2.html.

[70] netlib. Scalapack users guide. Web Page. Accessed: 2017-02-12. URL: http://www.netlib.org/scalapack/slug/.

[71] Alfredo Buttari, Julien Langou, Jakub Kurzak, and Jack Dongarra. A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Computing*, 35(1):38–53, 2009. URL: http://www.sciencedirect.com/science/article/pii/S0167819108001117.

[72] PLASMA README. Web Page. URL: http://icl.cs.utk.edu/projectsfiles/plasma/html/README.html.

[73] Jack Dongarra, Mark Gates, Azzam Haidar, Jakub Kurzak, Piotr Luszczek, Stanimire Tomov, and Ichitaro Yamazaki. Accelerating numerical dense linear algebra calculations with GPUs. In *Numerical Computations with GPUs*, pages 3–28. Springer, 2014. URL: http://link.springer.com/chapter/10.1007/978-3-319-06548-9_1.

[74] 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*, 90. ACM, 2013. URL: http://dl.acm.org/citation.cfm?id=2503292.

[75] Jakub Kurzak, Hatem Ltaief, Jack Dongarra, and Rosa M. Badia. Scheduling dense linear algebra operations on multicore processors. *Concurrency and Computation: Practice and Experience*, 22(1):15–44, 2010. URL: http://onlinelibrary.wiley.com/doi/10.1002/cpe.1467/full.

[76] Stanimire Tomov, Jack Dongarra, and Marc Baboulin. Towards dense linear algebra for hybrid GPU accelerated manycore systems. *Parallel Computing*, 36(5):232–240, 2010. URL: http://www.sciencedirect.com/science/article/pii/S0167819109001276.

[77] Stanimire Tomov, Rajib Nath, Hatem Ltaief, and Jack Dongarra. Dense linear algebra solvers for multicore with GPU accelerators. In *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, 1–8. IEEE, 2010. URL: http://ieeexplore.ieee.org/abstract/document/5470941/.

[78] C.J. Gronlund, Gary Ericson, Larry Francs, and Paulette McKay. Azure machine learning. Web Page, January 2017. Accessed: 2017-2-27. 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.

[79] Google. Google cloud prediction api documentation. Web Page. Accessed 2017-1-26. URL: https://cloud.google.com/prediction/docs/.

[80] Google. Google cloud translation api documentation. Web Page. Accessed 2017-2-20. URL: https://cloud.google.com/translate/docs/.

[81] Davide Albanese, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman, and Cesare Furlanello. Mlpy: machine learning python. *CoRR*, 2012. URL: http://arxiv.org/abs/1202.6548.

[82] Mlpy documentation. Web Page, 2012. URL: http://mlpy.sourceforge.net/docs/3.5/.

[83] Jason Brownlee. A gentle introduction to scikit-learn: a python machine learning library. webpage, 2014. URL: http://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/.

[84] Scikit-learn. webpage. URL: https://en.wikipedia.org/wiki/Scikit-learn.

[85] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, and Thomas Rückstießand Jürgen Schmidhuber. Pybrain. *Journal of Machine Learning Research*, 11:743–746, March 2010. URL: http://dl.acm.org/citation.cfm?id=1756006.1756030.

[86] Rudi Cilibrasi, Anna Lissa Cruz, Steven de Rooij, and Maarten Keijzer. What is complearn. webpage. URL: http://complearn.org/.

[87] Daal intel wikipedia. Accessed: 02-23-2017. URL: https://en.wikipedia.org/wiki/Data_Analytics_Acceleration_Library.

[88] Daal intel official website. Accessed: 02-23-2017. URL: https://software.intel.com/en-us/intel-daal.

[89] Caffe-deep learning. Accessed: 02-06-2017. URL: http://caffe.berkeleyvision.org/.

[90] Torch-machine learning. Accessed: 02-06-2017. URL: https://en.wikipedia.org/wiki/Torch_(machine_learning).

[91] Theano. Web Page. Accessed: 2017-1-21. URL: http://deeplearning.net/software/theano/introduction.html.

[92] Wikipedia. Deeplearning4j. Web Page, February 2017. Accessed: 2017-02-11. URL: https://en.wikipedia.org/wiki/Deeplearning4j.

[93] H2O Wikipedia Documentation components. Web Page. URL: https://en.wikipedia.org/wiki/H2O_(software)l.

[94] Ibm watson. Web Page. Accessed: 2017-1-25. URL: https://en.wikipedia.org/wiki/Watson_(computer).

[95] Ibm watson product page. Web Page. Accessed: 2017-1-25. URL: https://www.ibm.com/watson.

[96] Oracle labs pgx. Web Page. Accessed: 2017-02-07. URL: https://docs.oracle.com/cd/E56133_01/2.3.1/index.html.

[97] Parallel graph analytics (pgx). Web Page. Accessed: 2017-02-11. URL: http://www.oracle.com/technetwork/oracle-labs/parallel-graph-analytics/overview/index.html.

[98] GraphLab. Graphlab. Web Page, December 2012. accessed 2017-01-28. URL: http://www.select.cs.cmu.edu/code/graphlab/.

[99] Apache Software Foundation. Apache spark graphx. Web Page, February 2017. accessed 2017-01-30. URL: http://spark.apache.org/graphx/.

[100] Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, and Ion Stoica. Graphx: a resilient distributed graph system on spark. In *First International Workshop on Graph Data Management Experiences and Systems*, GRADES '13, 2:1–2:6. New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2484425.2484427, doi:10.1145/2484425.2484427.

[101] Ibm system g documentation. Web Page, 2014. Accessed: 2017-2-19. URL: http://systemg.research.ibm.com/.

[102] Ibm system g documentation-2. Web Page, 2017. Accessed: 2017-2-17. URL: http://www.predictiveanalyticstoday.com/ibm-system-g-native-store/.

[103] Ching Yung Lin. Graph computing and linked big data. In *8th IEEE/ICSC International Conference on Semantic Computing*, 1–63. IBM Corporation, 2014. URL: http://ieee-icsc.org/icsc2014/GraphComputing_IBM_Lin.pdf.

[104] Dylan Raithel. Apache tinkerpop graduates to top-level project. Web Page, 2016. URL: https://www.infoq.com/news/2016/06/tinkerpop-top-level-apache/.

[105] Apache Tinker Pop Home. Apache tinkerpop home. Web Page, 2016. URL: https://tinkerpop.apache.org/.

[106] Parasol. Webpage. URL: https://parasol.tamu.edu/research.php.

[107] Overview. Online. The contact information listed on the webpage was for Yogesh Simmhan. URL: http://dream-lab.cds.iisc.ac.in/about/overview/.

[108] Siddharth Rao. Dream:lab – democratising computing through the cloud. Online, March 2016. URL: http://iisc.researchmedia.center/article/dreamlab-.

[109] John Denero. *CS61A: Online Textbook*. Berkeley, http://www-inst.eecs.berkeley.edu/~cs61a/sp12/book/, 2011. "This book is derived from the classic textbook Structure and Interpretation of Computer Programs by Abelson, Sussman, and Sussman. John Denero originally modified if for Python for the Fall 2011 semester." http://www-inst.eecs.berkeley.edu/~cs61a/sp12/book/. URL: http://www-inst.eecs.berkeley.edu/~cs61a/sp12/book/communication.html.

[110] Google. Fusiontablesupporthelp. Web Page, 2017. URL: https://support.google.com/fusiontables/answer/171181?hl=en.

[111] Wikipedia. Fusion table support. Web Page, Last updated in 1 November 2016. URL: https://support.google.com/fusiontables/answer/171181?hl=en.

[112] Google. Google fusion tables: data management, integration and collaboration in the cloud. 2012. URL: http://homes.cs.washington.edu/~alon/files/socc10.pdf.

[113] Nwb. Web Page, 2017. URL: http://nwb.cns.iu.edu/about.html.

[114] Elastic search. Web Page. Accessed: 2017-1-25. URL: https://www.elastic.co/products/elasticsearch.

[115] Elastic search getting started. Web Page. Accessed: 2017-1-25. URL: https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html.

[116] Clinton Gormley and Zachary Tong. *Elasticsearch - The Definitive Guide : A Distributed REAL-TIME SEARCH AND ANALYTICS ENGINE*. O'Reilly Media,Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472, 1st edition, 2015. ISBN 9781449358549.

[117] Elastic search on hadoop. Web Page. Accessed: 2017-1-25. URL: https://www.elastic.co/products/hadoop.

[118] Elastic Search. Web Page. Accessed: 2017-1-25. URL: https://en.wikipedia.org/wiki/Elasticsearch.

[119] Kibana: Explore, Visualize, Discover Data \textbar Elastic. Web Page. URL: https://www.elastic.co/products/kibana.

[120] GitHub - elastic/kibana: Kibana analytics and search dashboard for Elasticsearch. Web Page. URL: https://github.com/elastic/kibana.

[121] 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-

[122] Elasticsearch. Web Page, February 2017. Page Version ID: 767434249. URL: https://en.wikipedia.org/w/index.php?title=Elasticsearch&oldid=767434249.

[123] Kibana. Web Page, February 2017. Page Version ID: 767434139. URL: https://en.wikipedia.org/w/index.php?title=Kibana&oldid=767434139.

[124] Introduction \textbar Kibana User Guide [5.2] \textbar Elastic. Web Page. URL: https://www.elastic.co/guide/en/kibana/current/introduction.html.

[125] Elasticsearch. Logstash introduction. Web Page, February 2017. Accessed: 2017-02-11. URL: https://www.elastic.co/guide/en/logstash/current/introduction.html.

[126] Andy Lurie. Top 47 log management tools. webpage, May 2014. Accessed : 02-21-2017. URL: http://blog.profitbricks.com/top-47-log-management-tools.

[127] Luke Galea. Graylog2 optimization for high-log environments. webpage, jul 2012. Accessed : 02-21-2017. URL: https://dzone.com/articles/graylog2-optimization-high-log.

[128] Dashboards - 2.2.1 documentation. webpage, 2012. Accessed : 02-21-2017. URL: http://docs.graylog.org/en/2.2/pages/dashboards.html.

[129] Splunk, Inc. Splunk enterprise overview. Web Page, February 2015. accessed: 2017-02-18. URL: http://docs.splunk.com/Documentation/Splunk/6.5.2/Overview/AboutSplunkEnterprise.

[130] Tableau tutorial. Web Page. Accessed: 2017-2-10. URL: https://casci.umd.edu/wp-content/uploads/2013/12/Tableau-Tutorial.pdf.

[131] Tableau technology. Web Page. Accessed: 2017-2-10. URL: https://www.tableau.com/products/technology.

[132] D3 data-driven documents. Web Page. Accessed: 2017-02-11. URL: https://d3js.org/.

[133] Potree. URL: http://potree.org/wp/about/.

[134] Martinez-Rubi.O, Verhoeven.S, Van Meersbergen.M, Schûtz.M Van, Oosterom.P Gonçalves.R, and Tijssen.T. 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.

[135] H. F. Halaoui. A spatio temporal indexing structure for efficient retrieval and manipulation of discretely changing spatial data. *Journal of Spatial Science*, 53(2):1–12, 2008. URL: http://dx.doi.org/10.1080/14498596.2008.9635146, doi:10.1080/14498596.2008.9635146.

[136] DC.js. Dc.js - dimensional charting javascript library. Web Page, January 2017. accessed: 2017-01-21. URL: https://dc-js.github.io/dc.js/.

[137] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citroand GS. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: large-scale machine learning on heterogeneous distributed systems. Technical Report, Cornell University, March 2016. Accessed: 2017-1-24.

[138] TensorFlow. Web Page. Accessed: 2017-1-24. URL: https://www.tensorflow.org.

[139] 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.

[140] Home · Microsoft/CNTK Wiki · GitHub. Web Page. URL: https://github.com/Microsoft/CNTK/wiki.

[141] Appengine - platform as a service. webpage. Accessed : 02-22-2017. URL: https://cloud.google.com/appengine.

[142] Wikipedia. Google app engine. webpage, jan 2017. Accessed : 02-22-2017. URL: https://en.wikipedia.org/wiki/Google_App_Engine.

[143] AppScale. What-is-appscale. Web Page, 2016. URL: https://www.appscale.com/community/what-is-appscale/.

[144] developers-openshift components. Web Page. Accessed: 2017-1-26. URL: https://developers.openshift.com/.

[145] openshift components. Web Page. Accessed: 2017-1-26. URL: https://www.openshift.org/.

[146] Heroku. Web Page. Accessed:1/27/2017. URL: https://devcenter.heroku.com/.

[147] Cedar stack. Web Page. Accessed:1/27/2017. URL: https://devcenter.heroku.com/articles/stack#cedar.

[148] Aerobatic. Aerobatic - overview. Web Page, January 2017. accessed: 2017-01-25. URL: https://www.aerobatic.com/docs/overview/.

[149] Wikipedia. Cloud computing — wikipedia, the free encyclopedia. web page, jan 2017. Online; accessed 21-jan-2017. URL: https://en.wikipedia.org/wiki/Cloud_computing.

[150] Microsoft Corp. Web Page, jan 2017. Online; accessed 21-jan-2017. URL: https://azure.microsoft.com/en-us/.

[151] Microsoft Corp. Form 10-k. Web page, july 2016. Online; accessed 21-jan-2017. URL: https://www.sec.gov/Archives/edgar/data/789019/000119312516662209/d187868d10k.htm.

[152] Amazon.com, Inc. Web Page, jan 2017. Online; accessed 25-jan-2017. URL: https://aws.amazon.com.

[153] IBM Corp. Web Page, jan 2017. Online; accessed 25-jan-2017. URL: www.softlayer.com.

[154] IBM Corp. Web Page, jan 2017. Online; accessed 25-jan-2017. URL: https://www.ibm.com/cloud-computing/bluemix/.

[155] Alphabet, Inc. Web Page, jan 2017. Online; accessed 25-jan-2017. URL: https://cloud.google.com.

[156] Duncan C.E Winn. *Cloud Foundry-The Cloud Native Platform*. O'Reilley Media, 2016. URL: https://books.google.com/books?id=XP2wDAAAQBAJ&printsec=frontcover&dq=Cloud+foundry.

[157] Cloud foundry blog. Web Page. URL: http://cloudacademy.com/blog/cloud-foundry-benefits/.

[158] Pivotal. Web Page. URL: https://run.pivotal.io/features/.

[159] Ninefold. Ninefold news. Web Page, November 2015. Accessed: 2017-2-27. URL: http://ninefold.com/news/.

[160] Jelastic. Web Page, December 2016. Page Version ID: 754931676. URL: https://en.wikipedia.org/w/index.php?title=Jelastic&oldid=754931676.

[161] Grow Hosting Business with Software Platform. Web Page. URL: https://jelastic.com/cloud-business-for-hosting-providers/.

[162] HPE. Hpe helion stackato. Webpage. URL: https://www.hpe.com/us/en/software/multi-cloud-platform.html.

[163] Mike Kavis. The pros and cons of private and public paas. Webpage, 06 2013. URL: https://www.virtualizationpractice.com/the-pros-and-cons-of-private-and-public-paas-21961/.

[164] Pau Kiat Wee. *Instant AppFog*, chapter 1. Packt Publishing, July 2013. URL: https://www.packtpub.com/mapt/book/Web-Development/9781782167624/1/ch01lvl1sec03/So,+what+is+AppFog.

[165] Ben Kepes. Understanding the cloud computing stack: saas, paas, iaas. white paper, Racspace, 2015. URL: https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/.

[166] appfog. Overview. Online. URL: https://www.ctl.io/appfog/#Overview.

[167] 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/.

[168] Cloudbees wikipedia documentation. Web Page. URL: https://en.wikipedia.org/wiki/CloudBees.

[169] Cloudbees webpage documentation. Web Page. URL: https://www.cloudbees.com/products.

[170] Dan Sullivan. Paas provider comparison guide: engine yard - orchestration and management. Article, July 2013. Accessed: 02-26-2017. URL: http://www.tomsitpro.com/articles/paas-engine-yard-amazon-elastic-cloud-computing,2-578.html.

[171] Wikipedia. Cloudcontrol. Wiki Page, February 2016. URL: https://en.wikipedia.org/wiki/CloudControl.

[172] Jordan Novet. Dotcloud. Web Page, January 2016. Accessed: 2017-1-31. URL: http://venturebeat.com/2016/01/22/dotcloud-the-cloud-service-that-gave-birth-to-docker-is-shutting-down-on-febr

[173] Apache. About oodt. Web Page. Accessed: 2017-02-12. URL: http://oodt.apache.org/.

[174] 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)*. December 2015.

[175] Agave api home – features tab. webpage. Accessed : 02-04-2017. URL: https://agaveapi.co/platform/features/.

[176] At1. Web Page. Accessed: 2017-1-22. URL: http://www.cyverse.org/atmosphere.

[177] Martin Lisa, Cook Charis, Matasci Naim, Williams Jason, and Bastow Ruth. 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.

[178] DevTopics. Web Page. URL: http://www.devtopics.com/kite-obscure-programming-language-of-the-month/.

[179] VoltDB. Web Page. URL: https://www.wired.com/2016/04/kites-coding-asssitant-spots-errors-finds-better-open-source/.

[180] Apache tajo. Web Page. Accessed: 2017-1-27. URL: https://tajo.apache.org.

[181] Apache tajo quick guide. Web Page. Accessed: 2017-1-25. URL: https://www.tutorialspoint.com/apache_tajo/apache_tajo_quick_guide.htm.

[182] Engle, Cliff, Lupher, Antonio, Xin, Reynold, Zaharia, Matei, Franklin, Michael J., Shenker, Scott, Stoica, and Ion. Shark: fast data analysis using coarse-grained distributed memory. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, 689–692. New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2213836.2213934, doi:10.1145/2213836.2213934.

[183] Justin Kestelyn. Phoenix in 15 minutes or less. Web page, march 2013. Online; accessed 25-jan-2017. URL: http://blog.cloudera.com/blog/2013/03/phoenix-in-15-minutes-or-less/.

[184] Wikipedia. Apache phoenix —wikipedia, the free encyclopedia. Web page, jan 2017. Online; accessed 25-jan-2017. URL: https://en.m.wikipedia.org/wiki/Apache_Phoenix.

[185] Apache Software Foundation. Apache phoenix: oltp and operational analytics for apache hadoop. Web page, jan 2017. Online; accessed 25-jan-2017. URL: http://phoenix.apache.org/.

[186] Adam Seligman. Apache phoenix - a small step for big data. Web page, may 2014. Online; accessed 25-jan-2017. URL: https://developer.salesforce.com/blogs/developer-relations/2014/05/apache-phoenix-small-step-big-data.html.

[187] Abel Avram. Phoenix: running sql queries on apache hbase [updated]. Web page, jan 2013. Online; accessed 25-jan-2017. URL: https://www.infoq.com/news/2013/01/Phoenix-HBase-SQL.

[188] Istvan Szegedi. Apache phoenix - an sql driver for hbase. Web page, may 2014. Online; accessed 23-jan-2017. URL: https://bighadoop.wordpress.com/2014/05/17/apache-phoenix-an-sql-driver-for-hbase/.

[189] Apache Software Foundation. Apache mrql. Web Page, April 2016. accessed 2017-01-29. URL: https://mrql.incubator.apache.org/.

[190] Edward J. Yoon. Mrql - a sql on hadoop miracle. Web Page, January 2017. accessed 2017-01-29. URL: http://www.hadoopsphere.com/2013/04/mrql-sql-on-hadoop-miracle.html.

[191] Apache Software Foundation. Apache incubator. Web Page. accessed 2017-01-29. URL: http://incubator.apache.org/.

[192] SAP. What is sap hana. Web Page. Accessed: 2017-1-17. URL: http://www.sap.com/product/technology-platform/hana.html.

[193] Carl W Olofson. The analytic-transactional data platform: enabling the real-time enterprise (idc). Technical Report, International Data Corporation (IDC), Dec 2014. Accessed: 2017-1-17. URL: http://www.sap.com/documents/2016/08/3c4e546e-817c-0010-82c7-eda71af511fa.html.

[194] Bajda Pawlikowski, Azza Abouzeid, Daniel Abadi, and Avi Silberschatz. 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.

[195] Pivotal hawq. URL: http://hawq.incubator.apache.org/.

[196] Pivotal hdb. URL: https://pivotal.io/pivotal-hdb.

[197] Presto. Web Page. Accessed: 2017-1-24. URL: https://prestodb.io/.

[198] Yueguo Chen, Xiongpai Qin, Haoqiong Bian, Jun Chen, Zhaoan Dong, Xiaoyong Du, Yanjie Gao, Dehai Liu, Jiaheng Lu, and Huijie Zhang. *A Study of SQL-on-Hadoop Systems*, pages 154–166. Springer International Publishing, 2014.

[199] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: interactive analysis of web-scale datasets. *Communications of the ACM*, 54:114–123, June 2011. URL: http://cacm.acm.org/magazines/2011/6/108648-dremel-interactive-analysis-of-web-scale-datasets/fulltext, doi:10.1145/1953122.1953148.

[200] What is bigquery? google cloud platform. Web Page. Accessed: 2017-2-23. URL: https://cloud.google.com/bigquery.

[201] What is bigquery? bigquery documentation google cloud platform. Web Page. Accessed: 2017-2-23. URL: https://cloud.google.com/bigquery/what-is-bigquery.

[202] Mosha Pasumansky. Inside capacitor, bigquery's next-generation columnar storage format. Blog, April 2016. Accessed: 2017-2-23. URL: https://cloud.google.com/blog/big-data/2016/04/inside-capacitor-bigquerys-next-generation-columnar-storage-format.

[203] Amazon Web services. Amazon redshift management services. Web Page, February 2017. Accessed: 2017-02-12. URL: http://docs.aws.amazon.com/redshift/latest/mgmt/overview.html.

[204] Wikipedia. Amazon redshift. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Amazon_Redshift.

[205] Apache drill. Web Page. Accessed:2/4/2017. URL: https://drill.apache.org/.

[206] Kyoto cabinet. Web Page. Accessed:1/16/2017. URL: http://fallabs.com/kyotocabinet/.

[207] Rob Pike, Sean Dorward, Robert Griesemer, and Sean Quinlan. Interpreting the data: parallel analysis with sawzall. *Scientific Programming Journal*, 13(4):277–298, oct 2005. URL: https://research.google.com/archive/sawzall-sciprog.pdf, doi:10.1155/2005/962135.

[208] Ryan Rosario. Exciting tools for big data: s4, sawzall and mrjob! Web page, november 2010. Online; accessed 30-jan-2017. URL: http://www.bytemining.com/2010/11/exciting-tools-for-big-data-s4-sawzall-and-mrjob/.

[209] Alphabet, Inc. Szl - overview.wiki. Code Repository. Online; accessed 30-jan-2017. URL: https://code.google.com/archive/p/szl/wikis/Overview.wiki.

[210] Google. Cloud dataflow. Web Page. Accessed: 02/03/2016. URL: https://cloud.google.com/dataflow/.

[211] Eileen McNulty. What is google cloud dataflow? Web Page. Accessed: 02/03/2016. URL: http://dataconomy.com/2014/08/google-cloud-dataflow/.

[212] 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.

[213] Gamefromscratch. Hands On With Amazon's Lumberyard Game Engine - YouTube. URL: https://www.youtube.com/watch?v=FUDlTTbt4qE.

[214] Magister Vis. What Is Lumberyard? (Lumberyard Tutorials Series #1) - YouTube. URL: https://www.youtube.com/watch?v=Fxwo3KqSsUI.

[215] Muhammad Hussain Iqbal and Tariq Rahim Soomro. Big data analysis: apache storm perspective. In *International Journal of Computer Trends and Technology (IJCTT)-2015*. January 2015.

[216] Apache storm homepage. webpage. Accessed : 01-22-2017. URL: http://storm.apache.org/releases/1.0.2/Concepts.html.

[217] S4: distributed stream computing platform. Web Page. Accessed: 2017-02-24. URL: http://incubator.apache.org/s4/.

[218] S4: s4 0.6.0 overview. Web Page. Accessed: 2017-02-24. URL: http://incubator.apache.org/s4/doc/0.6.0/overview.

[219] Morel Matthieu. S4 0.5.0 overview. Web Page, February 2013. Accessed: 2017-02-24. URL: https://cwiki.apache.org/confluence/display/S4/S4+0.5.0+Overview.

[220] Apache Samza. Web Page, February 2017. Page Version ID: 764035647. URL: https://en.wikipedia.org/w/index.php?title=Apache_Samza&oldid=764035647.

[221] Samza. Web Page. URL: http://samza.apache.org/.

[222] Apache Samza, LinkedIn's Framework for Stream Processing. Web Page, January 2015. URL: https://thenewstack.io/apache-samza-linkedins-framework-for-stream-processing/.

[223] Pallickara Shrideep, Buddhika Thilina, Malensek Matthew, and Stern Ryan. Granules. Project, July 2016. URL: http://granules.cs.colostate.edu/.

[224] Tyler Akidau, Alex Balikov, Kaya Bekiroglu, Slava Chernyak, Josh Haberman, Reuven Lax, Sam McVeety, Daniel Mills, Paul Nordstrom, and Sam Whittle. Millwheel: fault-tolerant stream processing at internet scale. In *Very Large Data Bases*, 734–746. 2013.

[225] Kinesis - real-time streaming data in the aws cloud. Web Page. Accessed: 2017-01-17. URL: https://aws.amazon.com/kinesis/.

[226] Sumit Gupta Shilpi Saxena. *Real-Time Big Data Analytics*. Packt Publishing, 35 Livery Street, Birmingham B3 2PB, UK, 1st edition, 2016. ISBN 9781784391409.

[227] Twittter. Open sourcing twitter heron. Web Page. Accessed: 2017-02-04. URL: https://blog.twitter.com/2016/open-sourcing-twitter-heron.

[228] Twittter. Flying faster with twitter heron. Web Page. Accessed: 2017-02-04. URL: https://blog.twitter.com/2015/flying-faster-with-twitter-heron.

[229] Microsoft Azure real-time data analytics. Web Page. Accessed: 2017-2-03. URL: https://azure.microsoft.com/en-us/services/stream-analytics/.

[230] Microsoft Docs azure stream analytics documentation. Web Page. Accessed: 2017-2-03. URL: https://docs.microsoft.com/en-us/azure/stream-analytics/.

[231] Github azure/ azure-stream-analytics. Web Page. Accessed: 2017-2-03. URL: https://github.com/Azure/azure-stream-analytics/.

[232] Open source data turbine initiative. Web Page. Accessed: 2017-2-26. URL: http://dataturbine.org/.

[233] Tony Fountain, Sameer Tilak, Peter Shin, and Michael Nekrasov. The open source dataturbine initiative: empowering the scientific community with streaming data middleware. *Bulletin - Ecological Society of America*, 93(3):242–252, July 2012. URL: http://onlinelibrary.wiley.com/doi/10.1890/0012-9623-93.3.242/abstract.

[234] 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.

[235] Twister: Iterative MapReduce. Web Page. URL: http://www.iterativemapreduce.org/.

[236] 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.

[237] Bingjing Zhang, Yang Ruan, Tak-Lon Wu, Judy Qiu, Adam Hughes, and Geoffrey Fox. Applying twister to scientific applications. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, 25–32. IEEE, 2010.

[238] Mr-mpi. Web Page, 2017. URL: http://mapreduce.sandia.gov/doc.

[239] Wikipedia. Apache flink. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Apache_Flink.

[240] Byung-Gon Chun. Reefproposal - incubator. Web Page, August 2014. accessed 2017-01-28. URL: https://wiki.apache.org/incubator/ReefProposal.

[241] The Disco Project. What is disco. Web page, feb 2017. accessed 25-feb-2017. URL: http://disco.readthedocs.io/en/develop/intro.html.

[242] The Disco Project. Why not hadoop? Web page, feb 2017. accessed 25-feb-2017. URL: http://disco.readthedocs.io/en/develop/faq.html#why-not-hadoop.

[243] Tait Clarridge. Disco - a powerful erlang and python map/reduce framework. Blog, may 2014. accessed 25-feb-2017. URL: http://www.taitclarridge.com/techlog/2014/05/disco-a-powerful-erlang-and-python-mapreduce-framework.html.

[244] Nokia Corp. Web page, feb 2017. accessed 25-feb-2017. URL: http://www.nokia.com/en_int.

[245] ZHAW Zurich University of Applied Sciences. About us. Blog, feb 2017. accessed 25-feb-2017. URL: https://blog.zhaw.ch/icclab/ueber-uns/.

[246] mesz. First public release of disco - a new distributed computing orchestration framework. Blog, may 2016. accessed 25-feb-2016. URL: https://blog.zhaw.ch/icclab/first-beta-release-of-disco/.

[247] Apache. web-page. accessed 2017-02-13. URL: https://hama.apache.org/.

[248] James J. (Jong Hyuk) Park, Hai Jin, Young-Sik Jeong, and Muhammad Khurram Khan. *Advanced Multimedia and Ubiquitous Engineering*. Springer, 2016. accessed 2017-02-13.

[249] Apache giraph. web page. URL: https://giraph.apache.org.

[250] Apache giraph wiki. web page. URL: https://en.wikipedia.org/wiki/Apache_Giraph.

[251] Facebook's graph search puts apache giraph on the map. web page. URL: https://www.pcworld.com/article/2046680/facebooks-graph-search-puts-apache-giraph-on-the-map.html.

[252] 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.

[253] ISS team - University of Texas. Galois website. Web Page. Accessed: 2017-2-27. URL: www-galoisSite: http://iss.ices.utexas.edu/?p=projects/galois.

[254] Keshav Pingali, Donald Nguyen, Milind Kulkarni, Martin Burtscher, M. Amber Hassaan, Rashid Kaleem, Tsung-Hsien Lee, Andrew Lenharth, Roman Manevich, Mario Mendez-Lojo, Dimitrios Prountzos, and Xin Sui. The tao of parallelism in algorithms. In *The Tao of Parallelism in Algorithms*, 1–14. June 2011. Accessed: 2017-2-27. URL: http://iss.ices.utexas.edu/Publications/Papers/pingali11.pdf.

[255] Bingsheng He Jianlong Zhong. Medusa: simplified graph processing on gpus. *IEEE Transactions on Parallel and Distributed Systems*, 25(6):1–11, April 2013. URL: http://pdcc.ntu.edu.sg/xtra/paper/2013ago/Medusa_TPDS13.pdf.

[256] 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.

[257] Netsyslab. Totem. Webpage. URL: http://netsyslab.ece.ubc.ca/wiki/index.php/Totem.

[258] Vinay Jayarama Setty. *Publish/subscribe for large-scale social interaction: Design, analysis and resource provisioning*. PhD thesis, Department of Informatics,UNIVERSITY OF OSLO, Faculty of Mathematics and Natural Sciences, University of Oslo, January 2015. Accessed:2017-1-29. URL: https://www.duo.uio.no/bitstream/handle/10852/43117/1595-Setty-DUO-Thesis.pdf?sequence=1&isAllowed=y.

[259] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June

2003. Accessed: 2017-1-27. URL: http://doi.acm.org/10.1145/857076.857078, doi:10.1145/857076.857078.

[260] Christian Esposito, Massimo Ficco, Francesco Palmieri, and Aniello Castiglione. A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing. *Know.-Based Syst.*, 79(C):3–17, May 2015. Accessed: 2017-1-27. URL: http://dx.doi.org/10.1016/j.knosys.2014.05.003, doi:10.1016/j.knosys.2014.05.003.

[261] High performance parallex (hpx-5). Web Page. Accessed: 2017-1-17. URL: https://hpx.crest.iu.edu/.

[262] *HPX-5: user guide*. HPX-5. Accessed: 2017-1-17. URL: https://hpx.crest.iu.edu/users_guide.

[263] Harp. Harp. Web Page, January 2012. accessed 2017-01-28. URL: http://harpjs.com.

[264] Netty site. Web Page. URL: http://netty.io/.

[265] Norman Maurer and Marvin Wolfthal. *Netty in Action*. Manning Publications, Greenwich, CT, USA, 1st edition, 2015. ISBN 1617291471. URL: http://www.ebook.de/de/product/21687528/norman_maurer_netty_in_action.html.

[266] Distributed messaging. Web Page. Accessed: 2017-1-24. URL: http://zeromq.org.

[267] 0mq - the guide. Web Page. Accessed: 2017-1-24. URL: http://zguide.zeromq.org/page:all.

[268] RabbitMQ, components. Web Page. Accessed: 2017-01-19. URL: https://www.rabbitmq.com/.

[269] John O'Hara. Toward a commodity enterprise middleware. *Queue*, 5(4):48–55, 2007.

[270] Community Grids Lab IU. The naradabrokering project @ iu community grids laboratory. Web Page, November 2009. Accessed: 2017-2-15. URL: http://www.naradabrokering.org/.

[271] G. Fox and S. Pallickara. Deploying the naradabrokering substrate in aiding efficient web and grid service interactions. *Proceedings of the IEEE*, 93(3):564–577, March 2005. Accessed : 2017-02-15. doi:10.1109/JPROC.2004.842759.

[272] Community Grids Lab IU. Some of the salient features in naradabrokering. Web Page, November 2009. Accessed: 2017-2-15. URL: http://www.naradabrokering.org/.

[273] Agargenta. About kestrel. Web Page. Accessed: 2017-02-12. URL: https://github.com/twitter-archive/kestrel.

[274] Wikipedia link for jms. webpage. Accessed : 01-18-2017. URL: https://en.wikipedia.org/wiki/Java_Message_Service.

[275] Oracle documentation on jms. webpage. Accessed : 01-18-2017. URL: http://docs.oracle.com/javaee/6/tutorial/doc/bnceh.html.

[276] AMQP. Amqp is the internet protocol for business messaging. Web Page, January 2017. accessed: 2017-01-31. URL: http://www.amqp.org/about/what.

[277] Mqtt. Web Page. Accessed: 2017-1-27. URL: http://mqtt.org/.

[278] Mqtt floodnet. Web Page. Accessed: 2017-1-27. URL: http://mqtt.org/projects/floodnet.

[279] Amazon sns. Web Page. Accessed: 2017-02-02. URL: https://aws.amazon.com/sns/.

[280] Amazon
sns blog. Web Page. Accessed: 2017-02-02. URL: https://aws.amazon.com/blogs/aws/introducing-the-amazon-simple-notification-service/.

[281] Amazon sns faqs. Web Page. Accessed: 2017-02-02. URL: https://aws.amazon.com/sns/faqs/.

[282] Amazon. Awslambda. Web Page. Accessed:2/18/2017. URL: https://aws.amazon.com/lambda/faqs/.

[283] 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.

[284] What-is-google-pub-sub. Web Page. Accessed: 2017-2-09. URL: https://cloud.google.com/pubsub/docs/overview.

[285] Google-pub-sub-scalable-messaging-middleware. Web Page. Accessed: 2017-2-10. URL: https://cloud.google.com/pubsub/.

[286] Abraham Silberschatz, Peter B Galvin, Greg Gagne, and A Silberschatz. *Operating system concepts*. Volume 4. Addison-wesley Reading, 1998.

[287] 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.

[288] Microsoft azure - queues. Web Page. Accessed: 2017-2-10. URL: https://www.tutorialspoint.com/microsoft_azure/microsoft_azure_queues.htm.

[289] Microsoft. Event hubs. Web Page, December 2016. accessed 2017-02-25. URL: https://azure.microsoft.com/en-us/services/event-hubs/.

[290] Gora - in-memory data model and persistence for big data. Web Page. Accessed: 2017-01-18. URL: http://gora.apache.org/.

[291] Dormando. Memcached. Web Page, February 2015. accessed 2017-01-30. URL: http://www.memcached.org/.

[292] Wikipedia. Key-value database. WebPage, November 2016. URL: https://en.wikipedia.org/wiki/Key-value_database.

[293] Wikipedia. Relational database. WebPage, January 2017. URL: https://en.wikipedia.org/wiki/Relational_database.

[294] 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/.

[295] Wikipedia. Hazelcast. Web Page, January 2017. accessed 2017-01-29. URL: https://en.wikipedia.org/wiki/Hazelcast.

[296] Hazelcast. Open source in-memory data grid. Code Repository, January 2017. accessed 2017-01-29. URL: https://github.com/hazelcast/hazelcast.

[297] Ehcache - features. Web Page. Accessed: 2017-01-21. URL: http://www.ehcache.org/about/features.html.

[298] Ehcache - documentation. Web Page. Accessed: 2017-01-21. URL: http://www.ehcache.org/documentation/3.2/getting-started.html.

[299] VoltDB. Web Page. URL: https://www.voltdb.com/.

[300] H-store. Web Page. Accessed:1/16/2017. URL: http://hstore.cs.brown.edu/.

[301] Robert Kallman, Hideaki Kimura, Jonathan Natkins, Andrew Pavlo, Alexander Rasin, Stanley Zdonik, Evan P. C. Jones, Samuel Madden, Michael Stonebraker, Yang Zhang, John Hugg, and Daniel J. Abadi. H-Store: a high-performance, distributed main memory transaction processing system. *Proc. VLDB Endow.*, 1(2):1496–1499, 2008. URL: http://hstore.cs.brown.edu/papers/hstore-demo.pdf, doi:http://doi.acm.org/10.1145/1454159.1454211.

[302] H-storewiki. Web Page. Accessed:1/16/2017. URL: https://en.wikipedia.org/wiki/H-Store.

[303] Wikipedia. Apache openjpa. webpage, dec 2016. Accessed : 02-22-2017. URL: http://openjpa.apache.org/.

[304] The Apache Software Foundation. Apache openjpa - wikipedia. webpage, jan 2017. Accessed : 02-22-2017. URL: https://en.wikipedia.org/wiki/Apache_OpenJPA.

[305] Eclipselink. Accessed: 02-06-2017. URL: https://en.wikipedia.org/wiki/EclipseLink.

[306] Wikipedia. Datanucleus wiki. Web Page. Accessed: 2017-02-04. URL: https://en.wikipedia.org/wiki/DataNucleus.

[307] DataNucleus. Datanucleus. Web Page. Accessed: 2017-02-04. URL: http://www.datanucleus.com/.

[308] JPAB. Jpa performance benchmark. Web Page. Accessed: 2017-02-04. URL: http://www.jpab.org/DataNucleus.html.

[309] Open database connectivity. Web Page. Accessed: 2017-1-30. URL: https://en.wikipedia.org/wiki/Open_Database_Connectivity.

[310] Java database connectivity. Web Page. Accessed: 2017-1-30. URL: https://en.wikipedia.org/wiki/Java_Database_Connectivity.

[311] Wikipedia. Uima. Web Page. Accessed: 2017.02.03. URL: https://en.wikipedia.org/wiki/UIMA.

[312] Slideshare. Apache uima introduction. Web Page. Accessed: 02/03/2016. URL: http://www.slideshare.net/teofili/apache-uima-introduction.

[313] Apache tika. Web Page. URL: https://tika.apache.org/.

[314] Oracle Database - Wikipedia. Web Page. URL: https://en.wikipedia.org/wiki/Oracle_Database.

[315] Sql server wiki. Web Page. URL: https://en.wikipedia.org/wiki/Microsoft_SQL_Server.

[316] Sql server azure. Web Page. URL: https://azure.microsoft.com/en-us/services/sql-database/?b=16.50.

[317] Ross Mistry and Stacia Misner. *Introducing Microsoft SQL Server 2014 Technical Overview*. Microsoft Press, 2014. ISBN 978-0-7356-8475-1.

[318] mysql. Mysql 5.7 reference manual, what is mysql. Online. URL: https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html.

[319] HowStuffWorks.com. What are relational databases? Online, March 2001. URL: http://computer.howstuffworks.com/question599.htm.

[320] Cubrid. Web Page, 2017. URL: http://www.cubrid.org/.

[321] Galera cluster. Web Page, 2017. URL: http://galeracluster.com/.

[322] Rasdaman array dbms wikipedia. Accessed: 02-23-2017. URL: https://en.wikipedia.org/wiki/Rasdaman).

[323] Rasdaman array dbms official website. Accessed: 02-23-2017. URL: http://www.rasdaman.org/.

[324] The Apache Software Foundation. Apache derby. Web Page, October 2016. accessed: 2017-02-18. URL: https://db.apache.org/derby/.

[325] The Apache Software Foundation. Apache derby project charter. Web page, September 2016. accessed: 2017-02-18. URL: https://db.apache.org/derby/derby_charter.html.

[326] The Apache Software Foundation. Getting started with derby. Web Page, September 2015. accessed: 2017-02-18. URL: https://db.apache.org/derby/docs/10.12/getstart/index.html.

[327] Google cloud sql. web page. URL: https://cloud.google.com/sql/.

[328] Google cloud sql faq. Webpage. URL: https://cloud.google.com/sql/faq#version.

[329] Amazon RDS. Amazonrds. Web Page. Accessed: 2017-02-04. URL: http://searchaws.techtarget.com/definition/Amazon-Relational-Database-Service-RDS.

[330] 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.

[331] Jeff Shute, Radek Vingralek, Bart Samwel, Ben Handy, Chad Whipkey, Eric Rollins, Mircea Oancea, Kyle Littlefield, David Menestrina, Stephan Ellner, John Cieslewicz, Ian Rae, Traian Stancescu, and Himani Apte. F1: a distributed sql database that scales. *Proc. VLDB Endow.*, 6(11):1068–1079, August 2013. Accessed:2017-02-15. URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41344.pdf, doi:10.14778/2536222.2536232.

[332] Jeff Shute, Mircea Oancea, Stephan Ellner, Ben Handy, Eric Rollins, Bart Samwel, Radek Vingralek, Chad Whipkey, Xin Chen, Beat Jegerlehner, Kyle Littlefield, and Phoenix Tong. 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, 777–778. New York, NY, USA, 2012. ACM. Accessed : 2017-02-15. URL: http://doi.acm.org/10.1145/2213836.2213954, doi:10.1145/2213836.2213954.

[333] 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.

[334] Ibm dashdb. Web Page. URL: https://www.ibm.com/analytics/us/en/technology/cloud-data-services/dashdb/.

[335] The Apache Software Foundation. Spark sql. Web Page, February 2016. accessed 2017-02-19. URL: http://spark.apache.org/sql/.

[336] ACM, Inc. Spark sql: relational data processing in spark. Web Page, February 2016. accessed 2017-02-19. URL: http://dl.acm.org/citation.cfm?id=2742797.

[337] Apache lucene. Web Page, January 2017. URL: http://lucene.apache.org/.

[338] Jake Luciani. Solandra wiki. Code Repository, February 2017. Accessed: 2017-02-12. URL: https://github.com/tjake/Solandra/wiki/Solandra-Wiki.

[339] Jake Luciani. Solandra wiki. Code Repository, February 2017. Accessed: 2017-02-12. URL: https://github.com/tjake/Solandra.

[340] LinkedIn. Project voldemort. Web Page. Accessed: 2017-1-17. URL: http://www.project-voldemort.com/voldemort/.

[341] Tilmann Rabl, Sergio Gómez-Villamor, Mohammad Sadoghi, Victor Muntés-Mulero, Hans-Arno Jacobsen, and Serge Mankovskii. Solving big data challenges for enterprise application performance management. *Proc. VLDB Endow.*, 5(12):1724–1735, August 2012. URL: https://arxiv.org/pdf/1208.4167.pdf, doi:10.14778/2367502.2367512.

[342] Riak-kv - nosql key value database. Web Page. Accessed: 2017-01-20. URL: http://basho.com/products/riak-kv/.

[343] Riak-ts - nosql time series database. Web Page. Accessed: 2017-01-20. URL: http://basho.com/products/riak-ts/.

[344] Riak-s2 - cloud object storage software. Web Page. Accessed: 2017-01-20. URL: http://basho.com/products/riak-s2/.

[345] Illinois Institute of Technology Department of Computer Science. Zht: a zero-hop distributed hashtable. Online. URL: http://datasys.cs.iit.edu/projects/ZHT/.

[346] Brandon Wiley. Distributed hash ttable, part 1. *Linux Journal*, October 2003. From issue number 114. URL: http://www.linuxjournal.com/article/6797?page=0,0.

[347] T. Li, X. Zhou, K. Brandstatter, D. Zhao, K. Wang, A. Rajendran, Z. Zhang, and I. Raicu. Zht: a light-weight reliable persistent dynamic scalable zero-hop distributed hash table. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, 775–787. May 2013. URL: http://datasys.cs.iit.edu/publications/2013_IPDPS13_ZHT.pdf, doi:10.1109/IPDPS.2013.110.

[348] Berkeley db wiki. Web Page. Accessed: 2017-2-11. URL: https://en.wikipedia.org/wiki/Berkeley_DB.

[349] Berkeley db tutorial and reference guide. Web Page. Accessed: 2017-2-11. URL: https://web.stanford.edu/class/cs276a/projects/docs/berkeleydb/reftoc.html.

[350] Oracle website. Web Page. Accessed: 2017-2-11. URL: http://www.oracle.com/technetwork/database/database-technologies/berkeleydb.

[351] Tokyo cabinet. Web Page. Accessed: 2017-1-27. URL: http://fallabs.com/tokyocabinet/.

[352] Kyoto cabinet. Web Page. Accessed: 2017-1-27. URL: http://fallabs.com/kyotocabinet/.

[353] Fal labs. Shijo cabinet: a handy cache/storage server. Web Page. Accessed: 02/03/2016. URL: http://fallabs.com/kyototycoon/.

[354] Nick Sullivan. Kyoto tycoon secure replication. Web Page. Accessed: 02/03/2016. URL: https://blog.cloudflare.com/kyoto-tycoon-secure-replication/.

[355] Fal labs. Nijo cabinet: a straightforward implementation of dbm. Web Page. Accessed: 02/03/2016. URL: http://fallabs.com/kyotocabinet/.

[356] Tyrant blog. Web Page. URL: https://www.percona.com/blog/2009/10/19/mysql_memcached_tyrant_part3/.

[357] Tyrant fallabs. Web Page. URL: http://fallabs.com/tokyotyrant/.

[358] Kyoto tycoon. Web Page. URL: http://fallabs.com/kyototycoon/.

[359] 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-linkedins-hot-new-distributed-document-store.

[360] Joe Lennon. Exploring couchdb: a document-oriented database for web applications. Web page, mar 2009. accessed feb-26-2017. URL: http://www.ibm.com/developerworks/opensource/library/os-couchdb/index.html.

[361] Apache Software Foundation. 1.1. document storage. Web page, feb 2017. accessed 26-feb-2017. URL: http://docs.couchdb.org/en/stable/intro/overview.html.

[362] Couchbase,
Inc. Couchbase and apache couchdb compared. Web page, feb 2017. accessed 26-feb-2017. URL: https://www.couchbase.com/couchbase-vs-couchdb.

[363] Rick Grehan. Nosql showdown: mongodb vs. couchbase. Web page, march 2013. Online; accessed 29-jan-2017. URL: http://www.infoworld.com/article/2613970/nosql/nosql-showdown--mongodb-vs--couchbase.html.

[364] Martin Brown. The technology behind couchbase. Web page, march 2012. Online; accessed 29-jan-2017. URL: https://www.safaribooksonline.com/blog/2012/03/01/the-technology-behind-couchbase/.

[365] Erlang Central. Couchbase performance and scalability: iterating with dtrace observability. Web page, march 2012. Online; accessed 29-jan-2017. URL: http://erlangcentral.org/videos/couchbase-performance-and-scalability-iterating-with-dtrace-observability/#.WI5uYephnRY.

[366] Wikipedia. Erlang (programming language) — wikpedia, the free encyclopedia. Web page, jan 2017. Online; accessed: 29-jan-2017. URL: https://en.wikipedia.org/wiki/Erlang_(programming_language).

[367] Sean Lynch. Why Membase Uses Erlang. web page, October 2010. accessed 2017-01-29. URL: https://blog.couchbase.com/why-membase-uses-erlang.

[368] Riyad Kalla. Well put!
when should you use mongodb vs couchbase versus redis... Web page, october 2011. Online; accessed 29-jan-2017. URL: http://rick-hightower.blogspot.com/2014/04/well-put-when-should-you-use-mongodb-vs.html.

[369] Russell Smith. What are the advantages and disadvantages of using mongodb vs couchdb vs cassandra vs redis? Web page, november 2015. Online; accessed 29-jan-2017. URL: https://www.quora.com/What-are-the-advantages-and-disadvantages-of-using-MongoDB-vs-CouchDB-vs-Cass

[370] Wikipedia. Ibm cloudant. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Cloudant.

[371] About pivotal gemfire. Web Page. Accessed: 2017-01-28. URL: http://gemfire.docs.pivotal.io/gemfire/getting_started/gemfire_overview.html.

[372] Apache hbase. Web Page. Accessed: 2017-1-26. URL: https://hbase.apache.org/.

[373] Google. Cloud bigtable. Web Page. accessed 2017-01-29. URL: https://cloud.google.com/bigtable/.

[374] Wikipedia. Spanner (database). Web Page, October 2016. accessed 2017-01-29. URL: https://en.wikipedia.org/wiki/Spanner_(database).

[375] Wikipedia. Bigtable. Web Page, January 2017. accessed 2017-01-29. URL: https://en.wikipedia.org/wiki/Bigtable.

[376] LevelDB. Leveldb. Web Page, February 2017. accessed 2017-02-25. URL: leveldb.org.

[377] James C Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Jeffrey John Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, and others. Spanner: google's globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):8, 2013. URL: http://dl.acm.org/ft_gateway.cfm?id=2491245&type=pdf.

[378] Mikio L. Braun. Magastore, Spanner - distributed databases. Web Page, 11 mar 2013. Accessed: 2017-01-28. URL: http://blog.mikiobraun.de/2013/03/more-google-papers-megastore-spanner-voted-commits.html.

[379] *Apache Accumulo User Manual Version 1.8*. accessed 2017-02-26. URL: https://accumulo.apache.org/1.8/accumulo_user_manual.html.

[380] Apache cassandra. Web Page, 2016. URL: http://cassandra.apache.org/.

[381] 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, 4:1–4:8. New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2347673.2347677, doi:10.1145/2347673.2347677.

[382] RDF Working Group. Resource description framework (rdf). Online, February 2014. URL: https://www.w3.org/RDF/.

[383] Apache Rya. Apache rya. Online. URL: https://rya.apache.org/.

[384] Wikipedia. Neo4j. Web page. Last Accessed: 2017.02.25. URL: https://en.wikipedia.org/wiki/Neo4j.

[385] Syed Ali Raza. Neo4j. Presentation/ Slides. Last Accessed: 2017.02.25. URL: https://www.slideshare.net/aliraza995/neo4j-graph-storage-27104408.

[386] Neo4j. Chapter 4. clustering. Web page. Last Accessed: 2017.02.25. URL: https://neo4j.com/docs/operations-manual/current/clustering/.

[387] Neo4j. Causal cluster. Web page. Last Accessed: 2017.02.25. URL: https://neo4j.com/docs/operations-manual/current/clustering/.

[388] Neo4j. Highly avaailable cluster. Web page. Last Accessed: 2017.02.25. URL: https://neo4j.com/docs/operations-manual/current/clustering/high-availability/architecture/.

[389] Wikipedia. Graph database. Web Page, February 2017. accessed 2017-01-30. URL: https://en.wikipedia.org/wiki/Graph_database.

[390] Cray Inc. Cray yarcdata urika appliance. Web Page, 2017. URL: http://www.cray.com/products/Urika.aspx.

[391] Cray Inc. Cray urika-gd technical specification. techreport, Cray Inc, 2014. URL: http://www.cray.com/sites/default/files/resources/Urika-GD-TechSpecs.pdf.

[392] Bloor Robin and Jozwiak Rebecca. The nature of graph data. Technical Report, The Bloor Group, Austin, TX 78720|512-524–3689, 2017. URL: http://web.cray.com/bloor-graph-data.

[393] Lee Sangkeun, Rangan Sukumar Sreenivas, and Lim Seung-Hwan. Graph mining meets the semantic web. In Johannes Gehrke, Wolfgang Lehner, and Kyuseok Shim, editors, *2015 31st IEEE International Conference on Data Engineering Workshops*, 53–58. IEEE, April 2015. URL: https://www.researchgate.net/profile/Sreenivas_Rangan_Sukumar/publication/273755615_Graph_Mining_Meets_the_Semantic_Web/links/550a27cb0cf20ed529e26260.pdf, doi:10.1109/ICDEW.2015.7129544.

[394] Allegro. Web Page. Accessed: 2017-1-25. URL: http://allegrograph.com/.

[395] Allegrow. Web Page. Accessed: 2017-1-20. URL: https://en.wikipedia.org/wiki/AllegroGraph.

[396] Titan db. Web Page. Accessed:2/4/2017. URL: http://titan.thinkaurelius.

[397] Tinkerpop. Web Page. Accessed:2/6/2017. URL: http://tinkerpop.apache.org/.

[398] w3. Apache jena. Web Page. Accessed: 2016.02.03. URL: https://www.w3.org/2001/sw/wiki/Apache_Jena.

[399] Craig Trim. Jen: semantic web framework. Web Page. Accessed: 02/03/2016. URL: http://trimc-nlp.blogspot.com/2013/06/introduction-to-jena.html.

[400] RDF components. Web Page. Accessed: 2017-1-25. URL: https://www.w3.org/RDF/.

[401] sesame components. Web Page. Accessed: 2017-1-26. URL: https://projects.eclipse.org/projects/technology.rdf4j.

[402] 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*, 70–73. 2013.

[403] Ken Smith. Azure: what to use, what to avoid. Web page, october 2014. Online; accessed 28-jan-2017. URL: http://blog.wouldbetheologian.com/2014/10/azure-what-to-use-what-to-avoid.html.

[404] The Windows Club. Understanding blob, queue and table storage for windows azure. Web page, jan 2017. Online; accessed 28-jan-2017. URL: http://www.thewindowsclub.com/understanding-blobqueuetable-storage-windows-azure.

[405] Microsoft Corp. Designing a scalable partitioning strategy for azure table storage. Web page, jan 2017. Online; accessed 28-jan-2017. URL: https://docs.microsoft.com/en-us/rest/api/storageservices/fileservices/designing-a-scalable-partitioning-strategy-for-azure-table-storage.

[406] iRod. Web Page. URL: https://irods.org/.

[407] iRod. Web Page. URL: https://github.com/irods/irods.

[408] 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.

[409] NetCDF: Introduction and Overview. Web Page. URL: https://www.unidata.ucar.edu/software/netcdf/docs_rc/.

[410] NASA/GSFC. Cdf home page. webpage, 2017. accessed: 2017-1-28. URL: http://cdf.gsfc.nasa.gov/.

[411] NASA/GSFC. *CDF User's Guide (V3.3.6)*. NASA/GSFC Space Physics Data Facility, NASA / Goddard Space Flight Center, Greenbelt, Maryland 20771 (U.S.A.), version 3.3.6 edition, October 2016. Accessed: 2017-1-28. URL: http://spdf.gsfc.nasa.gov/pub/software/cdf/doc/cdf363/cdf363ug.pdf.

[412] Cdf,common data format (multidimensional datasets). Web page, March 2014. Accessed: 2017-1-28. URL: http://www.digitalpreservation.gov/formats/fdd/fdd000226.shtml#useful.

[413] Wikipedia. Hierarchical data format — wikipedia, the free encyclopedia". Web Page, February 2017. Online,accessed: 2017-1-28. URL: https://en.wikipedia.org/wiki/Hierarchical_Data_Format.

[414] Fits nasa. web. URL: https://fits.gsfc.nasa.gov/.

[415] Fits news. Web Page. URL: https://fits.gsfc.nasa.gov/fits_standard.html.

[416] Fits vatican library. Web Page. URL: https://www.vatlib.it/home.php?pag=digitalizzazione&ling=eng.

[417] 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.

[418] *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.

[419] Rcfilecat - apache hive. Web Page. Accessed: 2017-1-27. URL: https://cwiki.apache.org/confluence/display/Hive/RCFileCat.

[420] Yongqiang He, Rubao Lee, Yin Huai, Zheng Shao, Namit Jain, Xiaodong Zhang, and Zhiwei Xu. 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*, 1199–1208. IEEE, 2011.

[421] Background. web-page. accessed 2017-02-13. URL: https://orc.apache.org/docs/.

[422] Parquet. Accessed: 02-06-2017. URL: https://parquet.apache.org/documentation/latest.

[423] Bittorrent. Web Page, 2017. URL: https://www.lifewire.com/how-torrent-downloading-works-2483513.

[424] Ftp wikipedia. Web Page. Accessed: 2017-02-02. URL: https://cwiki.apache.org/confluence/display/Hive/RCFileCat.

[425] Rfc114 specification. Web Page. Accessed: 2017-02-02. URL: https://cwiki.apache.org/confluence/display/Hive/RCFileCat.

[426] Ssh - wikipedia. Web Page. Accessed: 2017-01-26. URL: https://en.wikipedia.org/wiki/Secure_Shell.

[427] Openssh - wikipedia. Web Page. Accessed: 2017-01-26. URL: https://en.wikipedia.org/wiki/OpenSSH.

[428] Globus online (gridftp). Web Page. Accessed:1/16/2017. URL: https://en.wikipedia.org/wiki/GridFTP.

[429] IBM. What is flume? web page. accessed 201-02-06. URL: https://www-01.ibm.com/software/data/infosphere/hadoop/flume/.

[430] The Apache Software Foundation. Web Page. URL: http://sqoop.apache.org/.

[431] Wikipedia. Web Page. URL: https://en.wikipedia.org/wiki/Sqoop.

[432] Sunila Gollapudi. *Getting Started with Greenplum for Big Data Analytics*. Packt Publishing, October 2013. ISBN 1782177043. URL: http://www.ebook.de/de/product/21653990/sunila_gollapudi_getting_started_with_greenplum_for_big_data_analytics.html.

[433] Pivotal Software Inc. Gpfdist. Web Page, 2017. URL: http://gpdb.docs.pivotal.io/4330/utility_guide/admin_utilities/gpfdist.html.

[434] Jeffrey Ira Cohen, Luke Lonergan, and Caleb E. WeltonCohen. Integrating map-reduce into a distributed relational database. December 2016. URL: https://www.google.com/patents/US9514188.

[435] Mesos site. Web Page. URL: http://mesos.apache.org/.

[436] Abed Abu-Dbai, David Breitgand, Gidon Gershinsky, Alex Glikson, and Khalid Ahmed. Enterprise resource management in mesos clusters. In *Proceedings of the 9th ACM International on Systems and Storage Conference*, SYSTOR '16, 17:1–17:1. New York, NY, USA, 2016. ACM. URL: http://doi.acm.org/10.1145/2928275.2933272, doi:10.1145/2928275.2933272.

[437] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: fair allocation of multiple resource types. In *NSDI*, volume 11, 24–24. 2011.

[438] Roger Ignazio. *Mesos in Action*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2016. ISBN 1617292923. URL: http://dl.acm.org/citation.cfm?id=3006364.

[439] 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/.

[440] 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.

[441] Hadoop Apache. Apache software foundation. Web Page, 2016. URL: https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html.

[442] 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.

[443] Lars Kotthoff. Llama: leveraging learning to automatically manage algorithms. URL: https://arxiv.org/abs/1306.1031.

[444] Celery. Webpage. URL: http://www.celeryproject.org/.

[445] Celery - distributed task queue. Web Page. URL: http://docs.celeryproject.org/en/latest/index.html.

[446] What is htcondor? Web Page. URL: https://research.cs.wisc.edu/htcondor/description.html.

[447] Wikipedia. Oracle grid engine - wikipedia. webpage, feb 2017. Accessed : 02-22-2017. URL: https://en.wikipedia.org/wiki/Oracle_Grid_Engine.

[448] Wikipedia. Web Page. URL: https://en.wikipedia.org/wiki/Portable_Batch_System.

[449] PBS. Web Page. URL: http://www.pbspro.org/.

[450] Slurm. Web Page. URL: https://slurm.schedmd.com/.

[451] SchedMd. Slurm website. Web Page, March 2013. Accessed: 2017-2-27. URL: https://slurm.schedmd.com/overview.html.

[452] SchedMd. Slurm supported platforms. Web Page, April 2015. Accessed: 2017-2-27. URL: https://slurm.schedmd.com/platforms.html.

[453] Borja Sotomayor and Lisa Childers. *Globus\textregistered Toolkit 4: Programming Java Services*. Morgan Kaufmann, 2006.

[454] Ian Foster. Globus toolkit version 4: software for service-oriented systems. *Journal of computer science and technology*, 21(4):513, 2006.

[455] About the globus toolkit. Web Page. Accessed: 2017-2-26. URL: http://toolkit.globus.org/toolkit/about.html.

[456] Ioan Raicu, Yong Zhao, Catalin Dumitrescu, Ian Foster, and Mike Wilde. Falkon: a fast and light-weight task execution framework. In *ACM SC07*. 2007.

[457] Jik-Soo Kim, Seungwoo Rho, Seoyoung Kim, Sangwan Kim, Seokkyoo Kim, and Soonwook Hwang. Htcaas: leveraging distributed supercomputing infrastructures for large-scale scientific computing. In *ACM MTAGS 13*. 2013.

[458] Matteo Turilli, Mark Santcroos, and Shantenu Jha. A comprehensive perspective on pilot-job systems. In *ACM arXiv*, 1–26. March 2016.

[459] Subramanian Muralidhar, Wyatt Lloyd, Sabyasachi Roy, Cory Hill, Ernest Lin, Weiwen Liu, Satadru Pan, Shiva Shankar, Viswanath Sivakumar, Linpeng Tang, and others. F4: facebook's warm blob storage system. In *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*, 383–398. USENIX Association, 2014.

[460] Cinder - openstack. Web Page. Accessed: 2017-1-21. URL: https://wiki.openstack.org/wiki/Cinder.

[461] Dan Radez. *OpenStack Essentials*. Packt Publishing Ltd., 2015. ISBN 978-1-78398-708-5. URL: http://ebook.konfigurasi.net/Openstack/OpenStack%20Essentials.pdf.

[462] Red Hat, Inc. Ceph homepage-ceph. Web Page, 2017. Accessed: 2017-1-26. URL: https://ceph.com/.

[463] Red Hat,Inc. Ceph architecture-ceph documentation. Web Page, 2017. Accessed: 2017-1-24. URL: http://docs.ceph.com/docs/master/architecture/.

[464] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. Ceph: a scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*, OSDI '06, 307–320. Berkeley, CA, USA, November 2006. USENIX Association. Accessed: 2017-1-26. URL: https://www.usenix.org/legacy/event/osdi06/tech/full_papers/weil/weil.pdf.

[465] Red Hat,Inc. Ceph filesystem - ceph documentation. Web Page. Accessed: 2017-1-24. URL: http://docs.ceph.com/docs/master/cephfs/.

[466] Fuse site. Web Page. URL: http://fuse.sourceforge.net.

[467] Xianbo Zhang, David Du, Jim Hughes, Ravi Kavuri, and Sun StorageTek. 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. URL: https://www.dtc.umn.edu/publications/reports/2006_11.pdf, doi:10.1.1.184.1133.

[468] 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)*, 509–518. Institute of Electrical and Electronics Engineers (IEEE), Dec 2016. doi:10.1109/ICPADS.2016.0074.

[469] OpenSFS, EOFS. Welcome to the official home of the lustre filesystem. Web Page, December 2016. accessed 2017-01-28. URL: http://lustre.org/.

[470] Wikipedia. Ibm general parallel file system. Web Page, January 2017. accessed 2017-01-29. URL: https://en.wikipedia.org/wiki/IBM_General_Parallel_File_System.

[471] IBM. Ibm spectrum scale. Web Page. accessed 2017-01-29. URL: http://www-03.ibm.com/systems/storage/spectrum/scale/.

[472] PmWiki. Gffs – global federated file system. Web Page, January 2012. accessed 2017-01-28. URL: http://genesis2.virginia.edu/wiki/Main/GFFS.

[473] Amazon s3. Web Page. Accessed: 2017-1-27. URL: https://aws.amazon.com/s3/.

[474] Using Amazon S3. Web Page. Accessed: 2017-1-27. URL: http://docs.aws.amazon.com/AmazonS3/latest/gsg/CopyingAnObject.html.

[475] 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/.

[476] Get started with Azure Blob storage (object storage) using .NET \textbar Microsoft Docs. Web Page. URL: https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-how-to-use-blobs.

[477] Google cloud storage. Accessed: 02-06-2017. URL: https://cloud.google.com/storage/docs.

[478] Libvirt virtualization api. Web Page. URL: https://libvirt.org/.

[479] Tim Jones. Anatomy of the libvirt virtualization. Webpage, 01 2010. URL: https://www.ibm.com/developerworks/library/l-libvirt/.

[480] Antonio Esposito Beniamino Di Martino, Giuseppina Cretella. *Cloud Portability and Interoperability*. Springer International Publishing, New York City, USA, illustrated edition, 2015. ISBN 331913700X, 9783319137001.

[481] JClouds - the java multi-cloud toolkit. Web Page. Accessed: 2017-01-20. URL: https://jclouds.apache.org/.

[482] Open Grid Forum. Open cloud computing interface. Web Page. Accessed: 2017-1-17. URL: http://occi-wg.org/.

[483] Ralf Nyren, Andy Edmonds, Alesander Papaspyrou, Thijs Metsch, and Boris Parak. Open cloud computing interface core. OGF Published Document GWD-R-P.221, Global Grid Forum, Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA, September 2016. Accessed: 2017-1-17. URL: https://www.ogf.org/documents/GFD.221.pdf.

[484] Michel Drescher, Boris Parak, and David Wallom. Occi compute resource templates profile. OGF Published Document GWD-R-P.222, Global Grid Forum, Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA, April 2015. Accessed: 2017-1-17. URL: https://www.ogf.org/documents/GFD.222.pdf.

[485] Ralf Nyren, Andy Edmonds, and Thijs Metsch atnd Boris Parak. Open cloud computing interface - http protocol. OGF Published Document GWD-R-P.223, Global Grid Forum, Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA, September 2016. Accessed: 2017-1-17. URL: https://www.ogf.org/documents/GFD.223.pdf.

[486] Ralf Nyren, Florian Feldhaus, Boris Parak, and Zdenek Sustr. Open cloud computing interface -json rendering. OGF Published Document GWD-R-P.226, Global Grid Forum, Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA, September 2016. Accessed: 2017-1-17. URL: https://www.ogf.org/documents/GFD.226.pdf.

[487] Andy Edmonds and Thijs Metsch. Open cloud computing interface - text rendering. OGF Published Document GWD-R-P.229, Global Grid Forum, Open Grid Forum, P.O. Box 1738, Muncie IN 47308, USA, September 2016. Accessed: 2017-1-17. URL: https://www.ogf.org/documents/GFD.229.pdf.

[488] Storage Networking Industry Association. About the SNIA. Web Page. Accessed: 2017-2-27. URL: https://www.snia.org/about.

[489] Storage Networking Industry Association. *Cloud Data Management Interface*. Storage Networking Industry Association, Colorado Springs, CO, 1.1.1 edition, March 2015. Accessed: 2017-2-27. URL: https://www.snia.org/sites/default/files/CDMI_Spec_v1.1.1.pdf.

[490] Storage Networking Industry Association. Cloud data management interface. Web Page, March 2015. Accessed: 2017-2-27. URL: https://www.snia.org/cdmi.

[491] Shantenu Jha, Hartmut Kaiser, Andre Merzky, and Ole Weidner. Grid interoperability at the application level using saga. In *07 Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*. December 2007.

[492] Open grid forum - document. webpage. Accessed : 02-01-2017. URL: https://www.ogf.org/documents/GFD.90.pdf.

[493] Puppet software. Web Page. URL: https://en.wikipedia.org/wiki/Puppet_(software).

[494] Puppet faq. Web Page. URL: https://puppet.com/product/faq.

[495] How puppet works. Web Page. URL: http://www.slashroot.in/puppet-tutorial-how-does-puppet-work.

[496] Matthias Marschall. *Chef Infrastructure Automation Cookbook*. Packt Publishing, 2013. ISBN 9351105164 and 9789351105169.

[497] Chef commercial support. Web Page. URL: https://www.chef.io/support/.

[498] Red Hat, Inc. Ansible documentation. Web Page, February 2015. Accessed: 2017-02-12. URL: https://docs.ansible.com/ansible/index.html.

[499] Wikipedia. Ansible (software). Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Ansible_(software).

[500] boto components. Web Page. Accessed: 2017-1-25. URL: http://boto.cloudhackers.com/en/latest/.

[501] boto-github components. Web Page. Accessed: 2017-1-25. URL: https://github.com/boto/boto3.

[502] boto-amazon-python-sdk components. Web Page. Accessed: 2017-1-26. URL: https://aws.amazon.com/sdk-for-python/.

[503] Modine Austin. Cobbler. Web Page, February 2008. accessed 2017-01-30. URL: http://www.theregister.co.uk/2008/06/19/red_hat_summit_2008_cobbler/.

[504] xcat. Extreme cloud/cluster administration toolkit. Webpage, 2015. URL: http://xcat-docs.readthedocs.io/en/stable/.

[505] ibm. Extreme cloud administration toolkit. Webpage. URL: http://www-03.ibm.com/systems/technicalcomputing/xcat/.

[506] PuppetLabsRazor. Puppetlabsrazor. Web Page. Accessed: 2017-02-04. URL: https://github.com/puppetlabs/Razor/wiki.

[507] PuppetLabsRazor.
Puppetlabsrazor. Web Page. Accessed: 2017-02-04. URL: https://puppet.com/blog/introducing-razor-a-next-generation-provisioning-solution.

[508] 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.

[509] Canonical Ltd. Juju. Web Page. URL: https://www.ubuntu.com/cloud/juju.

[510] Blog on introduction to heat. webpage. Accessed : 01-15-2017. URL: http://blog.scottlowe.org/2014/05/01/an-introduction-to-openstack-heat/.

[511] Wikipedia link for heat. webpage. Accessed : 01-15-2017. URL: https://wiki.openstack.org/wiki/Heat.

[512] Openstack. Web Page. Accessed:1/16/2017. URL: https://www.openstack.org/.

[513] Sahara. Web Page. Accessed:1/16/2017. URL: http://docs.openstack.org/developer/sahara/.

[514] Rocks. Web Page, 2017. URL: https://www.rockscluster.org.

[515] Cloud and systems management. webpage. URL: http://www.cisco.com/c/en/us/products/cloud-systems-management.

[516] 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.

[517] Prashanth. Facebook tupperware. Blog, December 2015. Accessed:2/18/2017. URL: http://blog.cspp.in/index.php/2015/12/17/facebooks-tupperware/.

[518] Wikipedia. Chef (software). Web Page, January 2017. accessed 2017-01-26. URL: https://en.wikipedia.org/wiki/Chef_(software).

[519] Amazon. Aws opsworks. Web Page. accessed 2017-01-25. URL: https://aws.amazon.com/opsworks/.

[520] Devananda Van Der Veen and Llama Wrangler. Introduction to Ironic. Web Page, March 2015. Accessed: 2017-2-27. URL: https://docs.openstack.org/developer/ironic/deploy/user-guide.html.

[521] Kubernetes. Kubernetes site. Web Page. URL: https://kubernetes.io/docs/whatisk8s/.

[522] Wikipedia. Kubernetes wiki. Web Page. URL: https://en.wikipedia.org/wiki/Kubernetes.

[523] Moritz Plassnig. Heroku-style application deployments with docker - dzone cloud. Web Page, November 2015. Accessed: 2017-1-17. URL: https://dzone.com/articles/heroku-style-application-deployments-with-docker.

[524] Jose Gonzalez and Jeff Lindsay. Buildstep. Code repository, July 2015. Accessed: 2017-1-24. URL: https://github.com/progrium/buildstep.

[525] Eclipse winery. Web Page. Accessed: 2017-1-24. URL: https://projects.eclipse.org/projects/soa.winery.

[526] Oliver Kopp, Tobias Binz, Uwe Breitenbücher, and Frank Leymann. Winery – a modeling tool for tosca-based cloud applications. In *11\textsuperscript th International Conference on Service-Oriented Computing*, 700–704. Springer, December 2013.

[527] Exercise: analyze business processes with ibm bpm blueprint. Web Page. Accessed: 2017-1-24. URL: http://www.ibm.com/developerworks/downloads/soasandbox/blueprint.html.

[528] Blueworkslive. Web Page. Accessed: 2017-1-24. URL: https://www.blueworkslive.com/home.

[529] Ibm blueworks live. Web Page. Accessed: 2017-1-24. URL: https://en.wikipedia.org/wiki/IBM_Blueworks_Live.

[530] Terraform components. Web Page. URL: https://www.terraform.io/intro/index.html.

[531] James Turnbull. *The Terraform Book*. Number 9780988820258. Turnbull Press; 110 edition (November 26, 2016), 2016.

[532] 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), 108–122. SpringerVerlag, 2014. accessed 2017-02-26.

[533] Johannes Wettinger, Uwe Breitenbücher, and Frank Leymann. Any2api automated apification. In Markus Helfert, Donald Ferguson, and V\'ıctor Méndez Muñoz, editors, *CLOSER 2015 - Proceedings of the 5th International Conference on Cloud Computing and Services Science*, 475–486. Lisbon, Portugal, 20-22 May 2015. SciTePress. URL: https://pdfs.semanticscholar.org/1cd4/4b87be8cf68ea5c4c642d38678a7b40a86de.pdf, doi:10.5220/0005472704750486.

[534] Johannes Wettinger. any2api - the better way to create awesome apis. Web Page. Accessed: 2017-02-02. URL: http://www.any2api.org/.

[535] Xen - wikipedia. Web Page. Accessed: 2017-02-04. URL: https://en.wikipedia.org/wiki/Xen.

[536] Xen project overview. Web Page. Accessed: 2017-02-04. URL: https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview.

[537] Xen feature list. Web Page. Accessed: 2017-02-04. URL: https://wiki.xenproject.org/wiki/Xen_Project_4.7_Feature_List.

[538] Kvm wikipedia documentation. Web Page. URL: https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine.

[539] Kvm webpage documentation. Web Page. URL: http://www.linux-kvm.org/page/Main_Page.

[540] Wikipedia. Hypervisor. WebPage, February 2017. URL: https://en.wikipedia.org/wiki/Hypervisor.

[541] Qemu. Qemu. WebPage, February 2017. URL: http://wiki.qemu-project.org/index.php/Main_Page.

[542] Wikipedia. Hyper-v. Web Page, February 2017. Accessed: 2017-02-23. URL: https://en.wikipedia.org/wiki/Hyper-V.

[543] OpenVZ. Web Page, February 2017. Page Version ID: 764498783. URL: https://en.wikipedia.org/w/index.php?title=OpenVZ&oldid=764498783.

[544] How To Create OpenVZ Container In OpenVZ \textbar Unixmen. Web Page. URL: https://www.unixmen.com/how-to-create-openvz-container-in-openvz/.

[545] Features. Web Page. URL: https://openvz.org/Features.

[546] Linux containers. web page. URL: https://en.wikipedia.org/wiki/LXC.

[547] Why use lxc (linux containers) ? web page. URL: http://www.jpablo128.com/why-use-lxc-linux-containers/.

[548] Linux containers in use. web page. URL: http://www.infoworld.com/article/3072929/linux/containers-101-linux-containers-and-docker-explained.html.

[549] Software >> openstack open source cloud computing software. Web Page. Online; accessed 23-Feb-2017. URL: https://www.openstack.org/software/.

[550] Foundation >> openstack open source cloud computing software. Web Page. Online; accessed 23-Feb-2017. URL: https://www.openstack.org/foundation/.

[551] About opennebula. URL: https://opennebula.org/about/technology/.

[552] Opennebula wiki. URL: https://opennebula.org/about/technology/.

[553] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. Iaas cloud architecture from virtualized datacenters to federated cloud infrastructures. In IEEE, editor, *Computer*, volume 45, 65–72. IEEE Computer Society, IEEE, 2012. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6165242&isnumber=6383143, doi:10.1109/MC.2012.76.

[554] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 124–131. IEEE Computer Society, 2009.

[555] 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.

[556] Nimbus wiki. Web Page. URL: https://en.wikipedia.org/wiki/Nimbus_(cloud_computing).

[557] Nimbus. Web Page. URL: http://www.nimbusproject.org/doc/nimbus/platform/.

[558] *Rebalancing in a multi-cloud environment in Science Cloud '13*, ACM, 2013. URL: http://dl.acm.org/citation.cfm?id=2465854, doi:10.1145/2465848.2465854.

[559] The Apache Software Foundation. Apache cloudstack. Web Page, February 2017. Accessed: 2017-02-11. URL: https://cloudstack.apache.org/.

[560] Wikipedia. Apache cloudstack. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Apache_CloudStack.

[561] CoreOS. Why coreos. Web Page, January 2017. accessed: 2017-01-23. URL: https://coreos.com/why/.

[562] Coreos. Web Page. Accessed:1/27/2017. URL: https://www.coreos.com/.

[563] Coreos/rkt. Web Page. Accessed:1/27/2017. URL: https://github.com/coreos/rkt/.

[564] Wikipedia. web-page. accessed 2017-02-13. URL: https://en.wikipedia.org/wiki/VMware_ESXi.

[565] VMware. web-page. accessed 2017-02-13. URL: http://www.vmware.com/products/esxi-and-esx.html.

[566] vmware. Vcloud. Webpage. URL: http://www.vmware.com/products/vcloud-suite.html.

[567] Bipin. Difference between vsphere, esxi and vcenter. Webpage, 08 2012. URL: http://www.mustbegeek.com/difference-between-vsphere-esxi-and-vcenter/.

[568] Google cloud. Webpage. URL: https://en.wikipedia.org/wiki/Google_Cloud.

[569] Google cloud platform. Webpage. URL: https://cloud.google.com.

[570] What is a public cloud? Webpage. URL: http://www.interoute.com/cloud-article/what-public-cloud.

[571] Amazon route 53. Web Page. Accessed: 2017-02-24. URL: https://en.wikipedia.org/wiki/Amazon_Route_53.

[572] What is amazon route 53? Web Page. Accessed: 2017-02-24. URL: http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html.

[573] Hortonworks apache ambari. Web Page. Accessed: 2017-2-04. URL: http://hortonworks.com/apache/ambari/.

[574] Ambari. Web Page. Accessed: 2017-2-04. URL: https://ambari.apache.org/.

[575] Github apache/ ambari. Web Page. Accessed: 2017-2-04. URL: https://github.com/apache/ambari/.

[576] Ganglia monitoring system. Web Page. Accessed: 2017-02-24. URL: http://ganglia.info/.

[577] Ganglia (software). Web Page. Accessed: 2017-02-24. URL: https://en.wikipedia.org/wiki/Ganglia_(software).

[578] Nagios components. Web Page. Accessed: 2017-1-11. URL: https://www.nagios.org/projects/.

[579] David Josephsen. *Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks*. Prentice Hall Press, Upper Saddle River, NJ, USA, 2nd edition, 2013. ISBN 013313573X, 9780133135732.

[580] C. Issariyapat, P. Pongpaibool, S. Mongkolluksame, and K. Meesublak. Using nagios as a groundwork for developing a better network monitoring system. In *2012 Proceedings of PICMET '12: Technology Management for Emerging Technologies*, 2771–2777. July 2012.

[581] Jinjun Chen Lizhe Wang, Wei Jie. *Grid Computing: Infrastructure, Service, and Applications*. Taylor & Francis, 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487, 2009. ISBN 1420067664.

[582] Inca - periodic, automated, user-level cyberinfrastructure testing. Web Page. Accessed: 2017-01-16. URL: `http://inca.sdsc.edu/`.

[583] InCommon. What is the incommon federation? Webpage. URL: `https://spaces.internet2.edu/download/attachments/2764/final_InCommon.pdf`.

[584] Eduroam. Web Page. URL: `https://www.eduroam.org/about/`.

[585] Licia Florio and Klaas Wierenga. Eduroam, providing mobility for roaming users. *TERENA*, 2005. URL: `https://www.terena.org/activities/tf-mobility/docs/ppt/eunis-eduroamfinal-LF.pdf`.

[586] Keystone wiki. Web Page. Accessed: 2017-2-12. URL: `https://wiki.openstack.org/wiki/Keystone`.

[587] 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*, 283–288. IEEE, 2015.

[588] Openstack keystone verification. Web Page. Accessed: 2017-2-12. URL: `https://www.cloudberrylab.com/blog/openstack-keystone-authentication-explained/`.

[589] Keystone architecture. Web Page. Accessed: 2017-2-12. URL: `http://docs.openstack.org/developer/keystone/architecture.html`.

[590] TechTarget. Lightweight directory access protocol. Web Page, February 2017. accessed 2017-01-30. URL: `http://searchmobilecomputing.techtarget.com/definition/LDAP`.

[591] Apache sentry tutorial. Web Page. Accessed: 2017-2-11. URL: `https://cwiki.apache.org/confluence/display/SENTRY/Sentry+Tutorial`.

[592] Openid. website. URL: `http://openid.net/`.

[593] Openid. webpage. URL: `https://en.wikipedia.org/wiki/OpenID`.

[594] 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`.

[595] Mike Burrows. Chubby site. Web Page. URL: https://research.google.com/archive/chubby.html.

[596] 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)*, 1–10. August 2016. URL: http://ieeexplore.ieee.org/abstract/document/7568499/, doi:10.1109/ICCCN.2016.7568499.

[597] Zookeeper - overview. Web Page. Accessed: 2017-01-23. URL: https://zookeeper.apache.org/doc/trunk/zookeeperOver.html.

[598] Zookeeper - wikipedia. Web Page. Accessed: 2017-01-23. URL: https://en.wikipedia.org/wiki/Apache_ZooKeeper.

[599] Ibm - what is zookeeper. Web Page. Accessed: 2017-01-23. URL: http://www-01.ibm.com/software/data/infosphere/hadoop/zookeeper/.

[600] Xuanhua Shi, Haohong Lin, Hai Jin, Bing Bing Zhou, Zuoning Yin, Sheng Di, and Song Wu. Giraffe: a scalable distributed coordination service for large-scale systems. In *GIRAFFE: A Scalable Distributed Coordination Service for Large-scale Systems*, 1–10. September 2014. Accessed: 2017-2-27. URL: http://www.mcs.anl.gov/papers/P5157-0714.pdf.

[601] Mark Slee, Aditya Agarwal, and Marc Kwiatkowski. Thrift: scalable cross-language services implementation. *Thrift*, 2007. URL: https://thrift.apache.org/static/files/thrift-20070401.pdf.

[602] Apache. About thrift. Web Page. Accessed: 2017-02-12. URL: https://thrift.apache.org/.

[603] Protocol buffer. Web Page, September 2016. URL: https://developers.google.com/protocol-buffers/.

[604] Wikipedia. Snort software. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Snort_(software).

[605] Wikipedia. Fiddler software. Web Page, February 2017. Accessed: 2017-02-12. URL: https://en.wikipedia.org/wiki/Fiddler_(software).

[606] Apache Zeppelin. Apache zeppelin 0.7.0 documentation. Web Page, February 2017. Accessed: 2017-2-27. URL: https://zeppelin.apache.org/docs/0.7.0/.

[607] The Open MPI Project. Open MPI: Open Source High Performance Computing. Web Page, February 2017. Accessed: 2017-2-22. URL: https://www.open-mpi.org.

[608] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, 97–104. Budapest, Hungary, September 2004.

[609] Apache org. Apache tomcat - official website. Accessed: 02-24-2017. URL: http://tomcat.apache.org/.

[610] Wikipedia. Apache tomcat- wikipedia website. Accessed: 02-24-2017. URL: https://en.wikipedia.org/wiki/Apache_Tomcat.

[611] Ian Pointer. Apache beam want to be uber-api for big data. Web page, apr 2016. accessed 25-feb-2017. URL: http://www.infoworld.com/article/3056172/application-development/apache-beam-wants-to-be-uber-api-for-big-data.html.

[612] Alex Woodie. Apache beam's ambitious goal: unify big data development. Web page, apr 2016. accessed 25-feb-2017. URL: https://www.datanami.com/2016/04/22/apache-beam-emerges-ambitious-goal-unify-big-data-development/.

[613] Jean-Baptiste Onofre. The future of apache beam, now a top-level apache software foundation project. Blog, jan 2017. accessed 25-feb-2017. URL: https://www.talend.com/blog/2017/01/13/future-apache-beam-now-top-level-apache-software-foundation-project/.

[614] Cloudability Inc. Cloudability cost management tools | cloudability. Web Page, February 2017. Accessed: 2017-02-23. URL: https://www.cloudability.com/product/.

[615] Wikipedia. Cuda. Web Page, February 2017. Online; accessed 25-Feb-2017. URL: https://en.wikipedia.org/wiki/CUDA.

[616] Continuum Analytics. Blaze. Web page, 2015. accessed 25-feb-2017. URL: http://blaze.readthedocs.io/en/latest/index.html#.

[617] CASK. Cask - the first unified integration platform for big data. Web Page. Online; accessed 18-Feb-2017. URL: http://cask.co/products/cdap/.

[618] 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.

[619] Cdap applications. Code Repository, May 2015. Accessed: 2017-2-18. URL: https://github.com/caskdata/cdap-apps.

[620] Apache. Apache arrow. Webpage. URL: http://arrow.apache.org/.

[621] OpenRefine. Openrefine. Web Page, February 2016. accessed 2017-01-13. URL: http://openrefine.org/.

[622] Apache OODT. Web Page. Accessed: 2017-2-25. URL: http://oodt.apache.org/.

[623] Getting started. Web Page. Accessed: 2017-2-25. URL: http://oodt.apache.org/documentation.htm.

[624] Apache Software Foundation. Omid project incubation status. Web Page, April 2016. accessed 2017-02-25. URL: https://mrql.incubator.apache.org/.

[625] Florin Pop, Joanna Kolodziej, and Beniamino DiMartino. *Resource Management for Big Data Platforms*, pages 49–50. Springer Nature, 2016.

[626] The Apache Software Foundation. Web Page. URL: http://ant.apache.org/.

[627] Understanding lxc and lxd. web page. URL: http://thevarguy.com/open-source-application-software-companies/understanding-lxc-and-lxd-canonicals-open-source-containe.

[628] Lxd an hypervisor for containers. web page. URL: https://lists.linuxcontainers.org/pipermail/lxc-devel/2014-November/010817.html.

[629] The lxd container hypervisor. web page. URL: https://www.ubuntu.com/containers/lxd.

[630] Apache wink. Web Page. Accessed: 2017-02-24. URL: http://wink.apache.org/index.html.

[631] Wikipedia. Apache apex wiki. Web Page. Accessed: 2017-02-26. URL: https://en.wikipedia.org/wiki/Apache_Apex.

[632] Amol Kekre. Apache apex blog. Web Page, September 2015. Accessed: 2017-02-26. URL: https://www.datatorrent.com/blog/introducing-apache-apex-incubating/.

[633] Apache Apex. Application developer guide. Web-page. URL: https://apex.apache.org/docs/apex/application_development/#application-developer-guide.

[634] Apache Apex. Operator development guide. Web-page. accessed 2017-02-27. URL: https://apex.apache.org/docs/apex/operator_development/.

[635] Desmond Chan. Big data with apache apex. Web-page, January 2016. accessed 2017-02-27. URL: https://jaxenter.com/big-data-apache-apex-122839.html.

[1] Gregor von Laszewski, Fugang Wang, Hyungro Lee, Heng Chen, and Geoffrey C. Fox. Accessing Multiple Clouds with Cloudmesh. In *Proceedings of the 2014 ACM International Workshop on Software-defined Ecosystems*, BigSystem '14, 21–28. New York, NY, USA, 2014. ACM. URL: http://doi.acm.org/10.1145/2609441.2609638, doi:10.1145/2609441.2609638.