Panel on Merge or Split: Mutual Influence between Big Data and HPC Techniques

IEEE International Workshop on High-Performance Big Data Computing In conjunction with The 30th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2016)

In Chicago Hyatt Regency, Chicago, Illinois USA, Friday, May 27th, 2016

http://web.cse.ohio-state.edu/~luxi/hpbdc2016

Geoffrey Fox

May 27, 2016

gcf@indiana.edu

http://www.dsc.soic.indiana.edu/, http://spidal.org/ http://hpc-abds.org/kaleidoscope

Panel Topics

- **What is the impact of Big Data techniques on HPC?**
  - Software sustainability model from Apache community
  - Functionality in data area from streaming to repository to NOSQL to Graph
  - Parallel computing paradigm useful in simulations as well as big data
  - DevOps gives sustainability and interoperability
- **What is the impact of HPC techniques on Big Data?**
  - Performance of mature hardware, algorithms and software
- **Future mutual influence between HPC and Big Data techniques?**
  - HPC-ABDS(Apache Big Data Stack) Software Stack; take best of each world
  - Integrated environments that approach **data and model** components of Big data and simulations; **use HPC-ABDS for Exascale and Big Data**
  - Work with Apache and Industry
  - Specifying stacks and benchmarks with DevOps

Kaleidoscope of (Apache) Big Data Stack (ABDS) and HPC Technologies

Green is HPC work of NSF14-43054

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

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

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

12) Object-relational mapping: Hibernate, OpenJPA, EclipseLink, DataNucleus, ODBC/JDBC

HPC-ABDS

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

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

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

Cross-Cutting Functions

1) **Message and Data Protocols:** Avro, Thrift, Protobuf

2) **Distributed Coordination: ** Google Chubby, Zookeeper, Giraffe, JGroups

3) **Security & Privacy:** InCommon, Eduroam OpenStack Keystone, LDAP, Sentry, Sqrrl, OpenID, SAML OAuth

4) **Monitoring:** Ambari, Ganglia, Nagios, Inca

21 layers Over 350 Software Packages

January 29 2016

	Big Data ABDS	HPC, Cluster
17. Orchestration	Beam, Crunch, Tez, Cloud Dataflow	Kepler, Pegasus, Taverna
16. Libraries	MLlib/Mahout, TensorFlow, CNTK, R, Python	ScaLAPACK, PETSc, Matlab
15A. High Level Programming	Pig, Hive, Drill	Domain-specific Languages
15B. Platform as a Service	App Engine, BlueMix, Elastic Beanstalk	XSEDE Software Stack
Languages	Java, Erlang, Scala, Clojure, SQL, SPARQL, Python	Fortran, C/C++, Python
14B. Streaming	Storm, Kafka, Kinesis	
13,14A. Parallel Runtime	Hadoop, MapReduce	MPI/OpenMP/OpenCL CUDA, Exascale Runtime
2. Coordination	Zookeeper	
12. Caching	Memcached	
11. Data Management	Hbase, Accumulo, Neo4J, MySQL	iRODS
10. Data Transfer	Sqoop	GridFTP
9. Scheduling	Yarn, Mesos	Slurm
8. File Systems	HDFS, Object Stores	Lustre
1, 11A Formats	Thrift, Protobuf	FITS, HDF
5. IaaS	OpenStack, Docker	Linux, Bare-metal, SR-IOV
Infrastructure	CLOUDS	SUPERCOMPUTERS

HPC-ABDS Integrated Software

Implementing HPC-ABDS

- Build **HPC data analytics library** – NSF14-43054 Dibbs SPIDAL building blocks
- Define Java Grande as approach and runtime
- Software Philosophy – **enhance existing ABDS** rather than building standalone software
  - Heron, Storm, Hadoop, Spark, Hbase, Yarn, Mesos
- Working with Apache; how should one do this?
  - Establish a standalone HPC project
  - Join existing Apache projects and contribute HPC enhancements
- Experimenting first with Twitter (Apache) Heron to build HPC Heron that supports science use cases (big images)

HPC-ABDS Mapping of Dibbs NSF14-43054 project

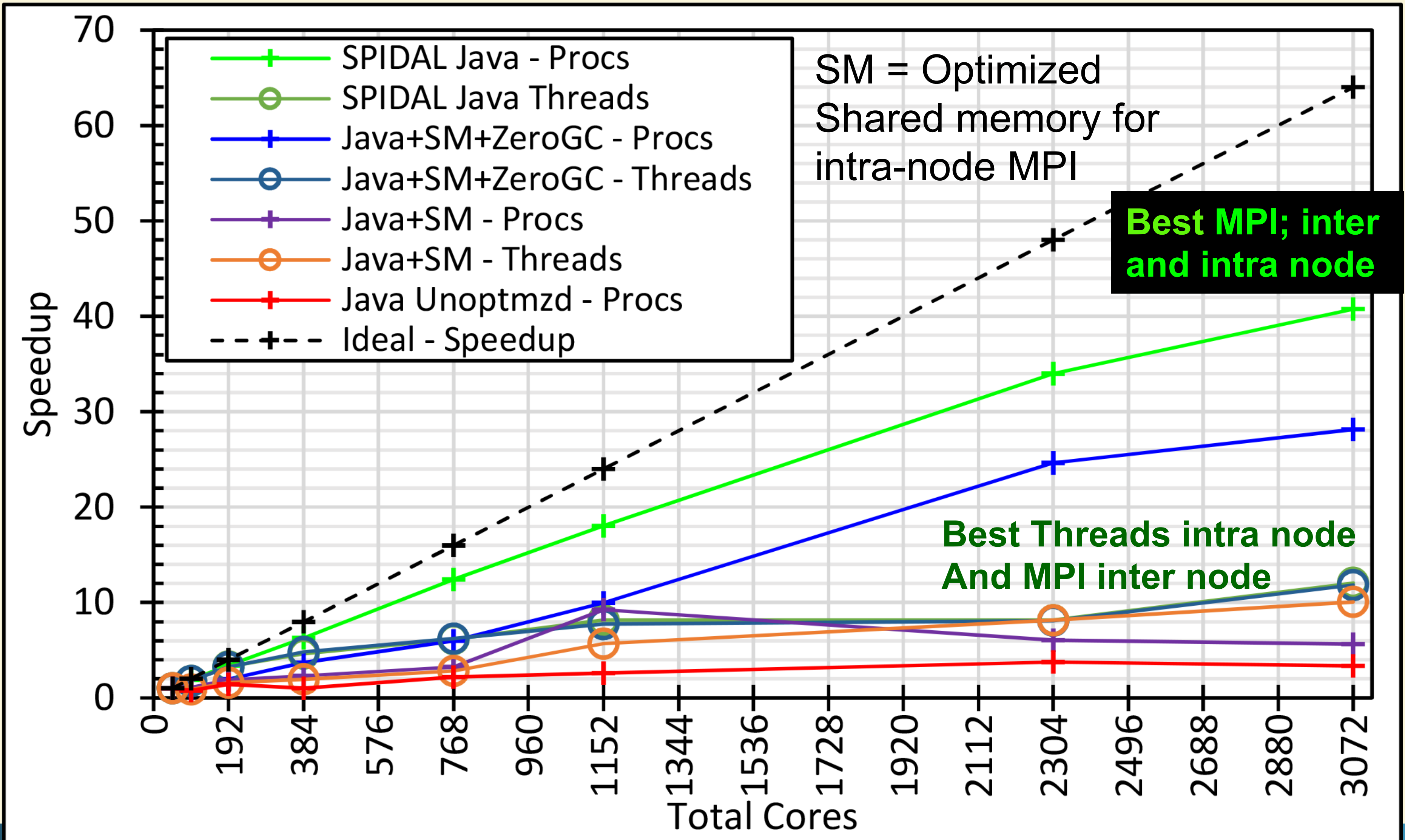
- **Level 17: Orchestration:** Apache Beam (Google Cloud Dataflow) integrated with Cloudmesh on HPC cluster
- **Level 16: Applications:** Datamining for molecular dynamics, Image processing for remote sensing and pathology, graphs, streaming, bioinformatics, social media, financial informatics, text mining
- **Level 16: Algorithms:** Generic and custom for applications **SPIDAL**
- **Level 14: Programming:** Storm, Heron (Twitter replaces Storm), Hadoop, Spark, Flink. Improve Inter- and Intra-node performance
- **Level 13: Communication:** Enhanced Storm and Hadoop using HPC runtime technologies, Harp
- **Level 11: Data management:** Hbase and MongoDB integrated via use of Beam and other Apache tools; enhance Hbase
- **Level 9: Cluster Management:** Integrate Pilot Jobs with Yarn, Mesos, Spark, Hadoop; integrate Storm and Heron with Slurm
- **Level 6: DevOps:** Python Cloudmesh virtual Cluster Interoperability

Constructing HPC-ABDS Exemplars

- This is one of next steps in NIST Big Data Working Group
- Philosophy: jobs will run on virtual clusters defined on variety of infrastructures: HPC, SDSC Comet, OpenStack, Docker, AWS, Virtualbox
- Jobs are defined hierarchically as a combination of Ansible (preferred over Chef as Python) scripts
- Scripts are invoked on Infrastructure (**Cloudmesh** Tool)
- INFO 524 **"Big Data Open Source Software Projects"** IU Data Science class required final project to be defined in Ansible and decent grade required that script worked (On NSF Chameleon and FutureSystems)
  - 80 students gave 37 projects with ~20 pretty good such as
  - "Machine Learning benchmarks on Hadoop with HiBench" Hadoop/YARN, Spark, Mahout, Hbase
  - "Human and Face Detection from Video" Hadoop, Spark, OpenCV, Mahout, MLLib
- Build up **curated collection of Ansible scripts** defining use cases for benchmarking, standards, education

Java MPI performs better than Threads

128 24 core Haswell nodes on SPIDAL DA-MDS Code



Big Data - Big Simulation (Exascale) Convergence

- Discuss **Data** and **Model** together as built around problems which combine them, but we can get insight by separating which allows better understanding of **Big Data - Big Simulation "convergence"**
- Big Data implies Data is large but Model varies
  - e.g. **LDA** with many topics or **deep learning** has large model
  - Clustering or Dimension reduction can be quite small
- **Simulations** can also be considered as **Data** and **Model**
  - **Model** is solving particle dynamics or partial differential equations
  - **Data** could be small when just boundary conditions
  - **Data** large with data assimilation (weather forecasting) or when data visualizations are produced by simulation
- **Data** often static between iterations (unless streaming); **Model** varies between iterations
- Take 51 NIST and other use cases → derive multiple specific features
- Generalize and systematize with features termed "facets"
- **50 Facets (Big Data) or 64 Facets (Big Simulation and Data)** divided into 4 sets or views where each view has "similar" facets
  - Allows one to study coverage of benchmark sets and architectures

64 Features in 4 views for Unified Classification of Big Data and Simulation Applications