Implications of the HPC-ABDS High Performance Computing Enhanced Apache Big Data Stack for workflows

Geoffrey Fox[a], Judy Qiu[a], Shantenu Jha[b], Supun Kamburugamuve[a] and Andre Luckow[b]
[a] School of Informatics and Computing, Indiana University, Bloomington, IN 47408, USA
[b] RADICAL, Rutgers University, Piscataway, NJ 08854, USA

1. Summary

We have introduced the software stack HPC-ABDS (High Performance Computing enhanced Apache Big Data Stack) to motivate an approach to high performance data analytics. The initial richness of the ABDS includes (cloud) workflow as well as Platform as a Service and system and heterogeneity management. We see these capabilities as relevant for the HPC workflow community but not yet fully exploited even though the distinction between HPPC workflows and data-intensive applications/analysis decrease. This paper argues the case for inclusion of HPC-ABDS approaches and components in any future system architecture, including integration of DevOps for systems management in the next-generation HPC workflow systems.

2. Introduction to HPC-ABDS

In previous work [1-4], we introduced the software stack HPC-ABDS shown online [5] and in the Figure with over 350 entries. These were combined with an application analysis [6-8] and used to motivate an approach to high performance data analytics including identification of a benchmarking set [9, 10]. We divided the stack into 21 architecture layers covering Message and Data Protocols, Distributed Coordination, Security & Privacy, Monitoring, Infrastructure Management, DevOps, Interoperability, File Systems, Cluster & Resource management, Data Transport, File management, NoSQL, SQL (NewSQL), Extraction Tools, Object-relational mapping, In-memory caching and databases, Inter-process Communication, Batch Programming model and Runtime, Stream Processing, High-level Programming, Application Hosting and PaaS, Libraries and Applications, Workflow and Orchestration. Further details of the stack can be found in an online course [11] that includes a section with approximately one slide (and associated lecture video) for each entry in Figure. The software in the figure comes from a variety of sources but we highlight the Apache contribution as it not only has contributed many key packages, it has established a very effective approach to software that has resonated with the cloud and big data commercial revolutions.

We suggest that HPC at all levels should carefully examine this software stack and adopt ABDS systems where relevant; this approach is likely to lead to a more sustainable software ecosystem where HPC can spend its scarce (relative to commercial world) resources on those areas where HPC has special expertise or requirements. We do not necessarily argue that ABDS is better than current HPC solutions but rather that adopting ABDS will in the long run lead to software environments that are cheaper and easier to maintain and have broader scope. We now briefly discuss some of the areas where HPC and ABDS have important overlaps and opportunities for integration.

Lower layers where HPC can make a major impact include scheduling where Apache technologies like Yarn and Mesos need to be compared with the sophisticated HPC approaches such as Slurm and Pilot jobs. Storage is another important area where HPC distributed and parallel storage environments need to be reconciled with the "data parallel" storage seen in HDFS in many ABDS systems. Further important issues are at the higher layers with data management, communication, (high layer or basic) programming, analytics and orchestration. These are areas where there is rapid commodity/commercial innovation and we briefly discuss them in order below. Much science data analysis is centered on files but we expect movement to the common commodity approaches of Object stores, SQL and NoSQL where latter has a proliferation of systems with different characteristics – especially in the data abstraction that varies over row/column, key-value, graph and documents. Note recent developments at the programming layer including Apache Hive and Drill, which offer high-layer access models such as SQL implemented on scalable NoSQL data systems. Maybe Drill can be generalized to offer traditional science interfaces such as FITS and HDF on ABDS data stores. The communication layer includes Publish-subscribe technology used in many approaches to streaming data as well the HPC communication technologies (MPI) which are much faster than most default Apache approaches but can be added [12] to some systems like Hadoop whose modern

version is modular and allows plug-ins for HPC stalwarts like MPI and sophisticated load balancers. The programming layer includes both the classic batch processing typified by Hadoop or Spark and streaming by Storm. The latter seems very relevant to the processing of the streaming observational data such as that from light sources and telescopes. The programming offerings differ in approaches to data model (key-value, array, pixel, bags, and graph), fault tolerance and communication. The trade-offs here have major performance implications.

Kaleidoscope of (Apache) Big Data Stack (ABDS) and HPC Technologies	
Cross-Cutting Functions	**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
1) **Message and Data Protocols:** Avro, Thrift, Protobuf	**16) Application and Analytics:** Mahout , MLlib , MLbase, DataFu, R, pbdR, Bioconductor, ImageJ, OpenCV, Scalapack, PetSc, 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, Google Fusion Tables, CINET, NWB, Elasticsearch, Kibana, Logstash, Graylog, Splunk, Tableau, D3.js, three.js, Potree, DC.js
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
3) **Security & Privacy:** InCommon, Eduroam OpenStack Keystone, LDAP, Sentry, Sqrrl, OpenID, SAML OAuth	**14B) Streams:** Storm, S4, Samza, Granules, Google MillWheel, Amazon Kinesis, LinkedIn Databus, Facebook Puma/Ptail/Scribe/ODS, Azure Stream Analytics, Floe **14A) Basic Programming model and runtime**, **SPMD, MapReduce:** Hadoop, Spark, Twister, MR-MPI, Stratosphere (Apache Flink), Reef, Hama, Giraph, Pregel, Pegasus, Ligra, GraphChi, Galois, Medusa-GPU, MapGraph, Totem
4) **Monitoring:** Ambari, Ganglia, Nagios, Inca	**13) Inter process communication Collectives, point-to-point, publish-subscribe:** MPI, 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
21 layers Over 350 Software Packages **May 15 2015**	**12) In-memory databases/caches:** Gora (general object from NoSQL), Memcached, Redis, LMDB (key value), Hazelcast, Ehcache, Infinispan
	12) Object-relational mapping: Hibernate, OpenJPA, EclipseLink, DataNucleus, ODBC/JDBC
	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
	11B) NoSQL: Lucene, Solr, Solandra, Voldemort, Riak, 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, 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, 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

One also sees systems like Apache Pig offering data parallel interfaces. At the high layer we see both programming models and Platform as a Service toolkits where the Google App Engine is well known but there are many entries including the recent BlueMix from IBM. The implementation of the analytics layer depends on details of orchestration and especially programming layers but probably most important are quality parallel algorithms. As many machine learning algorithms involve linear algebra, HPC expertise is directly applicable as is fundamental mathematics needed to develop O(NlogN) algorithms for analytics that are naively $O(N^2)$ [13].

3. Implications for Workflow

The orchestration or workflow layer has seen an explosion of recent activity in the commodity space although with systems like Pegasus, Taverna, Kepler, and Swift, HPC has substantial experience stemming from the Grid and service oriented architecture communities. There are ABDS orchestration dataflow systems like Tez but more interestingly projects like Apache Crunch with a data parallel emphasis based on ideas from Google FlumeJava. A modern version of the latter presumably underlies Google's recently announced Cloud Dataflow that unifies support of multiple batch and streaming components; a capability that we expect to become common. Cascading which is open source but not Apache is a popular ABDS workflow toolkit. NiFi is a recent Apache workflow environment from NSA, the US National Security Agency. NSA has a strong ABDS commitment having previously donated the popular NoSQL store Accumulo and spun off Sqrrl at the interface of big data and security. It seems important to evaluate the ABDS workflow systems compared to HPC approaches like Pegasus, Kepler, Taverna, and Swift; this evaluation should include ease of integration of other levels of HPC-ABDS including batch and streaming processing as well as SQL and NoSQL data systems. One area of HPC and ABDS commonality is the use of Python (IPython) to script orchestration environments.

We suggest that there we can find another important opportunity by looking at levels 5 (Infrastructure), 6 (DevOps), 15B (PaaS hosting environments) and 17 (Workflow and orchestration) which are naturally linked. At the IaaS level 5, there has been lots of recent progress in both hypervisor and Linux container-based virtual environments with OpenStack and Docker being key technologies. This progress has been accompanied by the rapid emergence of DevOps (level 6) to build software defined or automation systems. At level 6 one finds in particular, tools like Chef, Ansible, OpenStack Heat and Kubernetes. These DevOps tools define computer systems in a variety of languages and use this definition to automate system deployment. This approach was aimed at easing the systems administration task but has other interesting features. Firstly it makes interoperability at the infrastructure level easier as the DevOps tools can take the scripted system and deploy it on different hypervisors, containers or just bare metal without any user or administrative actions needed. Here standards at level 7 – especially libcloud or TOSCA and OCCI are important. One can use the same script on Docker, OpenStack or Amazon for example. Now let's understand importance of these ideas for workflow. DevOps defines a system for deployment whilst workflow defines systems for execution; these are different but closely related goals. This close relationship can be seen by comparing the two OASIS standards: BPEL for workflow and TOSCA for DevOps. They are very similar and have key people in common including the group at Stuttgart led by Frank Leymann. This group has developed the OpenTOSCA software automation environment and tools such as Winery that span orchestration and deployment. We don't recommend this particular approach – partly because BPEL was not so successful but do consider the linking of workflow with DevOps as an important area that should be included in any workflow initiative. We can quote the mission of OpenStack Heat "*The mission of the OpenStack Orchestration program is to create a human- and machine-accessible service for managing the entire lifecycle of infrastructure and applications within OpenStack clouds*". There are several other DevOps tools in the lifecycle and management area which could be very valuable in developing a robust HPC workflow stack that can run on a broad IaaS and not just OpenStack.

We can illustrate another idea with OpenStack Sahara that is a project to automatically build Hadoop systems on OpenStack. This combines DevOps with an idea often called PaaS or Platform as a Service, listed as level 15B in the Figure. PaaS identifies a set of popular tools and offers a complete managed environment supporting them. It can be considered as integrating DevOps and workflow for a particular application stack as Sahara does for Hadoop. Although some PaaS are focused, others like Heroku and Stackato support general application stacks. The integration with DevOps is illustrated by Dokku that

implements Heroku PaaS functionality on Docker where Heroku has buildpacks that are its DevOps style system definitions. The field is pretty incoherent but the trend is clear. Application stacks are defined as scripts which are used through the full life cycle from deployment through execution. These form well supported Platforms as a Service that can be deployed on a variety of IaaS. We suggest that HPC should investigate this model and build HPC systems in the DevOps supported PaaS style.

4. Conclusions

We have found it very fruitful to consider HPC-ABDS, which merges High Performance Computing with the Commodity Big Data Stack ABDS. In this white paper we have focused on the special implications of this approach for workflow and orchestration. Our analysis suggests an important broadening of HPC workflow to consider both ABDS workflow systems and the integration of workflow with DevOps and PaaS ideas. We stress that our ideas are synergistic with the concept of Software as a Service argued for at the workshop.

References

1. Judy Qiu, Shantenu Jha, Andre Luckow, and Geoffrey C.Fox, *Towards HPC-ABDS: An Initial High-Performance Big Data Stack*, in *Building Robust Big Data Ecosystem ISO/IEC JTC 1 Study Group on Big Data*. March 18-21, 2014. San Diego Supercomputer Center, San Diego. http://grids.ucs.indiana.edu/ptliupages/publications/nist-hpc-abds.pdf.
2. Geoffrey Fox, Judy Qiu, and Shantenu Jha, *High Performance High Functionality Big Data Software Stack*, in *Big Data and Extreme-scale Computing (BDEC)*. 2014. Fukuoka, Japan. http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/fox.pdf.
3. Shantenu Jha, Judy Qiu, Andre Luckow, Pradeep Mantha, and Geoffrey C. Fox, *A Tale of Two Data-Intensive Approaches: Applications, Architectures and Infrastructure*, in *3rd International IEEE Congress on Big Data Application and Experience Track*. June 27- July 2, 2014. Anchorage, Alaska. http://arxiv.org/abs/1403.1528.
4. Geoffrey Fox, Judy Qiu, Shantenu Jha, Supun Kamburugamuve, and Andre Luckow, *HPC-ABDS High Performance Computing Enhanced Apache Big Data Stack*, in *Invited talk at 2nd International Workshop on Scalable Computing For Real-Time Big Data Applications (SCRAMBL'15) at.CCGrid2015, the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2015, IEEE. Shenzhen, Guangdong, China. http://dsc.soic.indiana.edu/publications/HPC-#ABDSDescribedv2.pdf.
5. *HPC-ABDS Kaleidoscope of over 300 Apache Big Data Stack and HPC Tecnologies*. [accessed 2014 April 8]; Available from: http://hpc-abds.org/kaleidoscope/.
6. Geoffrey C. Fox, Shantenu Jha, Judy Qiu, and Andre Luckow, *Towards an Understanding of Facets and Exemplars of Big Data Applications*, in *20 Years of Beowulf: Workshop to Honor Thomas Sterling's 65th Birthday* October 14, 2014. Annapolis http://grids.ucs.indiana.edu/ptliupages/publications/OgrePaperv9.pdf.
7. Geoffrey Fox and Wo Chang, *Big Data Use Cases and Requirements*, in *1st Big Data Interoperability Framework Workshop: Building Robust Big Data Ecosystem ISO/IEC JTC 1 Study Group on Big Data* March 18 - 21, 2014. San Diego Supercomputer Center, San Diego. http://grids.ucs.indiana.edu/ptliupages/publications/NISTUseCase.pdf.
8. *NIST Big Data Use Case & Requirements*. 2013 [accessed 2015 March 1]; Available from: http://bigdatawg.nist.gov/V1_output_docs.php.
9. Geoffrey C.Fox, Shantenu Jha, Judy Qiu, and Andre Luckow, *Ogres: A Systematic Approach to Big Data Benchmarks*, in *Big Data and Extreme-scale Computing (BDEC)* January 29-30, 2015. Barcelona. http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/OgreFacets.pdf.
10. Geoffrey C. FOX , Shantenu JHA, Judy QIU, Saliya EKANAYAKE, and Andre LUCKOW, *Towards a Comprehensive Set of Big Data Benchmarks*. February 15, 2015. http://grids.ucs.indiana.edu/ptliupages/publications/OgreFacetsv9.pdf.
11. Geoffrey Fox. *Data Science Curriculum: Indiana University Online Class: Big Data Open Source Software and Projects*. 2015 [accessed 2015 March 31]; Available from: http://bigdataopensourceprojects.soic.indiana.edu/.
12. Bingjing Zhang, Yang Ruan, and Judy Qiu, in *IEEE International Conference on Cloud Engineering (IC2E)*. March 9-12, 2015. Tempe AZ. http://grids.ucs.indiana.edu/ptliupages/publications/HarpQiuZhang.pdf.
13. Committee on the Analysis of Massive Data; Committee on Applied and Theoretical Statistics; Board on Mathematical Sciences and Their Applications; Division on Engineering and Physical Sciences; National Research Council, *Frontiers in Massive Data Analysis*. 2013: National Academies Press. http://www.nap.edu/catalog.php?record_id=18374