A Tale of Two Convergences: Applications and Computing Platforms

Geoffrey C. Fox*, Shantenu Jha†,
* School of Informatics and Computing, Indiana University, Bloomington, IN 47408, USA
Email: gcf@indiana.edu
† Electrical and Computer Engineering, RADICAL, Rutgers University, Piscataway, NJ 08854, USA
Email: shantenu.jha@rutgers.edu

***Abstract*—There are two important types of convergence that will shape the near term future of computing sciences. The first is the convergence between HPC and Cloud platforms for science. The second is the integration between Simulations and Big Data applications. We believe understanding these trends is not just a matter of ideal speculation but is important in particular to conceptualize and design future computing platforms for Science. This paper presents our analysis of the convergence between simulations and big-data applications as well as selected research about managing the convergence between HPC and Cloud platforms.**

I. Applications and Platforms: Status Quo

On the one hand, there is a march towards the exascale computing platforms. On the other hand, there is a proliferation of software systems to support data- intensive applications with an accompanying cloud infrastructure of well publicized dramatic and increasing size and sophistication.

Traditional simulations involve applications of differential equation based models that need very fine space and time steps and this leads to numerical formulations that need the memory and compute power of traditional HPC resources to solve individual problems (capability computing). A big data application does not typically need a full HPC system for However there are different types of parallelism that can/need to be exploited in order for data-intensive analytics to scale.

Ref. [1] was an initial attempt to try to understand the convergence in high-performance computing and data-intensive computing. It was by necessity both high-level and broad reaching. In this paper we build upon initial work in Ref. [1] and we revisit the convergence problem by decomposing along two different trends: platforms (high-performance and cloud platforms) and applications (simulations and big- data).

Understanding these trends is important: (i) to engineer the platforms of the future, that might support both HPC and data-intensive problems, (ii) allow efficient sharing of large scale resources running simulations and data analytics; (iii) the need for higher performance Big Data algorithms; (iv) a richer software environment for research community building on many "big data" tools, and (v) Facilitate a sustainability model for HPC, as it does not have resources to build and maintain a full software stack.

II. Understanding Applications

Needless to say there are many similarities between between data-intensive and simulation applications. Some high-level differences worth a brief mention are:

- Classic Non-iterative MapReduce is major paradigm in data-intensive sciences, but it is not a common simulation paradigm except where "reduce" summarizes pleasingly parallel execution as in some Monte Carlo simulations
- Data intensive applications often have large collective communication, whereas classic simulation has a lot of smallish point-to-point messages which motivates the MapCollective model
- Simulations tend to need high precision and very accurate results (partly because of differential operators), however, data- intensive problems often don't need high accuracy as seen in trend to low precision (16 or 32 bit) deep learning networks, as there are no derivatives and the data has inevitable errors.

In order to understand and analyze systematically these differences we examined extensively the landscape of applications across the HPC and data- intensive spectrum. For example in Ref. [**?**], [2] on examining applications with common characteristics, we introduced the concept of Ogres, and 64 Convergence Diamonds (features). Ogres provide a means of understanding and characterizing the most common application characteristics found across the two paradigms. Ogres provide a classification and structure including, (i) classic MPI-based simulations, (ii) pleasingly parallel and workflow systems, and (iii) data-intensive applications epitomized by deep learning. Full details of Ogres and their facets can be found in Ref. [2].

We introduce four Ogres views — classification dimensions or features. These views are:

1) Problem Architecture: Related to the machine architecture needed to support application and describes properties of problem such as Pleasing Parallel or Uses Collective Communication.
2) Execution View: Describes issues such as I/O versus compute rates, iterative nature and regularity of computation and the classic Vs of Big Data defining problem size, rate of change, etc. Execution facets allow the separation of "Data" and "Model" for both simulations

and data-intensive applications.

3) Data Source and Style views include specifying how the data is collected, stored and accessed. For example: Streaming, files versus objects, HDFS vs. Lustre
4) Processing view describe types of processing steps including nature of algorithms and kernels used by model e.g. Linear Programming, Learning, Maximum Likelihood, Spectral methods, Mesh type. It incoporates aspects of key simulation kernels and in particular includes facets seen in NAS Parallel Benchmarks and Berkeley Dwarfs

Of course there are other ways of looking at the Ogres and our work should be treated as an initial suggestion for further discussion.

Comparison between Data Intensive and Simulation Problems

It is useful to understand the aspects of data-intensive applications that are unique and those that are similar to traditional compute-intensive simulations. In general, data-intensive applications are generally more heterogeneous than compute-intensive simulation problems. Typically a data pipeline (or workflows) comprises of multiple steps: data ingest, transfer, pre-processing, several rounds of processing (e. g. for cleaning, fusing, computation of summary statistics) and advanced analytics. Each step of the pipeline can be characterized according to computational characteristics facet: (i) by the size of the input, intermediate and output data, (ii) data access pattern (sequential, random) and (iii) computational characteristics (e.g. the parallelisms deployed).

The analytics part of the pipeline is compute-intensive and thus, resemble many characteristics of traditional simulations problems. For example, many analytics and machine learning problems can be formulated with linear algebra or n-body (see seven giants [3]). Thus, analytical kernels (e. g. linear algebra libraries, such as BLAS, SCALAPACK) provide the basis for data analytics. For example, machine learning algorithm, such as SVM or principal component analysis (PCA) rely on dense and sparse linear algebra. Often these analytical kernels are implemented using low-level libraries using fine-grained, tightly coupled parallelism often implemented with MPI, which yield into better performance than shoehorning the problem into a rigid MapReduce programming model. However, there is also a lack of scalable analytics algorithms that are able to operate on high-dimensional, sparse datasets.

We use Ogres (facets) to facilitate this comparision. There are some clear similarities: Embarassingly parallel, BSP and SPMD are common in both arenas. However, the Classic MapReduce architecture is a major Big Data paradigm, but has much less common in simulations with one example between the execution of multiple simulations (as in Quantum Monte Carlo) followed by a reduce operation to collect the results of different simulations. The Iterative Map-Collective architecture is common in Big Data analytics, such as in clustering where there is no local graph structure and the parallel algorithms involve large-scale collectives but no point to point communication. The same structure is seen in N-body (long range force) or other "all-pairs" simulations without the locality typical from discretizing differential operators.

Many simulation problems have the Map-Communication architecture with numerous small point-to-point messages coming from local interactions between points defining system to be simulated. The importance of sparse data structures and algorithms is well understood in simulations and is seen in some Big Data problems such as PageRank, which calculates the leading eigenvector of the sparse matrix formed by internet site links. Other Big Data sparse data structures are seen in user-item ratings and bags of words problems. Most items are rated by few users and many documents contain a small fraction of the word vocabulary. However important data analytics involve full matrix algorithms; for example recent papers [4], [5] on a new Multi- Dimensional Scaling method use conjugate gradient solvers with full matrices.

Note that there are similarities between some Big Data graph problems and particle simulations with an unusual potential defined by the graph node connectivity. Both use the Map-Communication architecture and the links in a Big Data graph are equivalent to strength of force between the graph nodes considered as particles. In this analogy, many Big Data problems are "long range force" corresponding to a graph where all nodes are linked to each other. As in simulation cases, these O(N2) problems are typically very compute intense but straightforward to parallelize efficiently. It is interesting to consider the analogue of the "fast multipole" methods for the fully connected Big Data problems which can dramatically improve the performance to O(N) or O(N log N). Finally note the network connections used in deep learning are sparse but in recent image interpretation studies [**?**], the network weights are block sparse (corresponding to links to pixel blocks) and can be formulated as full matrix operations with GPUs and MPI running efficiently with these blocks.

The above discussion focuses on a qualitative comparison of Big Data applications with traditional simulation (HPC) applications visualization, comparing the structure. As shown here there are similarities as well as points of distinction. It is likely however, that there will be significant differences in the "computational feature" facet of the two application classes, viz., the distribution of the values of different ratios (e.g., ratio of computing to I/O, ratio of memory to I/O etc.) characterizing the computational feature will be different. We will investigate both quantitative and qualitative differences in future work.

III. Understanding Computing Platforms Trends

There are at least three trends that we see represented in any Future Platform for science research:

- The increasing power and complexity of modern HPC systems as exemplified by those involved in drive to build exascale class machines.
- The increasing use and sophistication of commercial and open cloud infrastructure (that can be used as IaaS, PaaS, SaaS, FaaS etc.)

- The increasing functionality and use of Big Data software systems in conjunction with HPC.

In addition, there is growing interest in streaming data and event based computing models such Amazon Lambda, IBM OpenWhisk and Function-as-a-Service (serverless computing). These general trends are likely to continue independent of specific technology trends which we classify as micro and macro architectural trends: *Microscopic Architecture:* The three primary microscopic architecture are: (i) Continuation of X86 systems, (ii) Many core systems (e.g., KNL) and (iii) non-traditional architectures (e.g., GPU, FPGA) etc. *Macroscopic Architecture:* The three primary macroscopic architectures are: (i) Data Center Model, (ii) Traditional supercomputers and, (iii) Clusters (with virtualization) such as those represented by NSF Comet.

Furthermore, any Future Platform must satisfy the following constraints:

- It must allow easy integration of public and private clouds and allow HPC and cloud approaches to run well and run together,
- It must allow the powerful features of modern clouds such as ABDS, XaaS to be usable on HPC hardware,
- It must support distributed data sources and repositories,
- It should support modern workflows and Pythonesque front ends; an area where simulations and Big Data have similar requirements.

We call such platforms HPC Cloud platforms. Independent of whether we consider "cloudification" of HPC, or the "HPCfication of Clouds". Independent of the directionality, the future platform will be a software- defined system that works across different types of macroscopic and microscopic architectures as well as for different applications systems. This requires the selective integration of the Apache Big-Data Stack (ABDS) capabilities appropriately implemented for supercomputing platforms.

Future platforms must support the analytics requirements from both ends of the spectrum: traditional simulation applications that need Big Data ("All exascale applications are data-intensive problems"), as well as data-intensive applications that will increasingly need high-performance capabilities (e.g., Deep Learning with HPC capabilities).

Given the current separation of characteristics of simulations and data-intensive applications this requires a convergence of capabilities: (i) there must be a richer set of analysis-as-a-service than currently available, (ii) future analysis and associated middleware must provide traditional performance capabilities, yet expose fundamentally new capabilities.

IV. HPC-ABDS: Supporting the Convergence of Applications and Platforms

Armed with an understanding of the spectrum of applications and platform trends, we will now discuss HPC-ABDS software stack that will support the high- performance analysis requirements on platforms resulting from the convergence of high-performance and cloud platforms [6], [7], [8].

In spite of many arguments, technology for data-processing like Spark, Flink, Hadoop, Storm, Heron are not designed to support parallel computing well and tend to get poor performance on those workloads that need tight task synchronization and/or use high performance hardware. A corollary of the huge success of unmodified Apache software results in a statement about the lack of classic parallel computing in commercial workloads. We know however, that such is not the case for scientific computing workloads, and thus without refactoring data-processing systems for parallelism they will not be very effective for scientific computing workloads.

In addition to using the rich functionality and usability of ABDS (Apache Big Data Stack) for HPC applications, the adoption of community open source sustainability models are also worthy. Further most ABDS components are optimized for fault-tolerance and usability and not performance ABDS run naturally on clouds and not HPC platforms where the cloud is logically centralized (even if physically distributed) but science data typically distributed.

Thus we have propose **HPC-ABDS** which uses HPC runtime and tools to enhance commercial data systems (ABDS). HPC-ABDS developed by the SPIDAL Project implements the High Performance Computing (HPC) enhanced Apache Big Data Stack (ABDS) uses the major open source Big Data software environment but develops the principles allowing use of HPC software and hardware to achieve good performance.

We have examined High-Performance Computing Enhanced Big Data Stack (HPC-ABDS) where we examined the addition of high performance runtime and components to Apache systems. We have highlighted the importance of the Big Data systems associated with Apache Foundation, such as Hbase, Hadoop, Spark, Storm etc., which we term the Apache Big Data Stack (ABDS), even though important components such as MongoDB and Tensorflow are not Apache projects. We note that most of these technologies are in principle usable on both HPC and Cloud IaaS systems, though in practice many challenges remain. Independent of the hardware infrastructure, there are even stronger forces driving the adoption of ABDS technologies. They offer usability, functionality and sustainability that is not available in the HPC ecosystem. A realization of the HPC-ABDS concept is provided by the SPIDAL project [9], [10] and discussed in publications [11], [2].

Some machine learning like topic modeling (LDA), clustering, deep learning, dimension reduction, graph algorithms involve Map-Collective or Map-Point to Point iterative structure and already benefit from HPC. However, in general, deep learning doesn't exhibit massive parallelism due to stochastic gradient descent using small mini-batches of training data, but deep learning does use small accelerator enhanced HPC clusters. If this were to change, this would have important implications for Deep learning and other data-intensive applications uptake on HPC platforms.

REFERENCES

[1] S. Jha, J. Qiu, A. Luckow, P. Mantha, and G. C.Fox, "A Tale of Two Data-Intensive Paradigms: Applications, Abstractions, and Architectures," in *Big Data (BigData Congress), 2014 IEEE International Congress on*, 2014, pp. 645 – 652.

[2] G. C. Fox, S. Jha, J. Qiu, and A. Luckow, "Towards an understanding of facets and exemplars of big data applications," in *Proceedings of the 20 Years of Beowulf Workshop on Honor of Thomas Sterling's 65th Birthday*, ser. doi: 10.1145/2737909.2737912. New York, NY, USA: ACM, 2015, pp. 7–16. [Online]. Available: http://doi.acm.org/10.1145/2737909.2737912

[3] Committee on the Analysis of Massive Data and Committee on Applied and Theoretical Statistics and Board on Mathematical Sciences and Their Applications and Division on Engineering and Physical Sciences, *Frontiers in Massive Data Analysis*. The National Academies Press, 2013.

[4] Y. Ruan and G. Fox, "A robust and scalable solution for interpolative multidimensional scaling with weighting," October 22-25 2013.

[5] Y. Ruan, G. L. House, S. Ekanayake, U. Schütte, J. D. Bever, H. Tang, and G. Fox, "Integration of clustering and multidimensional scaling to determine phylogenetic trees as spherical phylograms visualized in 3 dimensions," http://grids.ucs.indiana.edu/ptliupages/publications/PhylogeneticTreeDisplayWithClustering.pdf, pp. 26–29, May 26-29 2014.

[6] G. Fox, J. Qiu, and S. Jha, "High performance high functionality big data software stack," http://www.exascale.org/bdec/sites/www.exascale.org/bdec/files/whitepapers/fox.pdf, 2014.

[7] G. C. Fox, S. Jha, J. Qiu, and A. Luckow, "Ogres: A Systematic Approach to Big Data Benchmarks," http://www.exascale.org/bdec/sites/www.exascale.orgbdec/files/whitepapers/OgreFacets.pdf, Barcelona, January 29-30 2015.

[8] Geoffrey Fox, Judy Qiu, Shantenu Jha, Supun KamburugamuveSaliya Ekanayake, "Big Data, Simulations and HPC Convergence" Big Data and Extreme-Scale Computing (BDEC) Frankfurt June 15 to June 17, 2016 http://dx.doi.org/10.13140/RG.2.1.3112.2800.

[9] J. Qiu, S. Jha, A. Luckow, and G. C.Fox, "Towards HPC-ABDS: An Initial High-Performance Big Data Stack," March 18-21 2014. [Online]. Available: http://grids.ucs.indiana.edu/ptliupages/publications/nist-hpc-abds.pdf

[10] G. C. Fox, S. Jha, J. Qiu, and A. Luckow, "Towards an Understanding of Facets and Exemplars of Big Data Applications," October 14 2014. [Online]. Available: http://grids.ucs.indiana.edu/ptliupages/publications/OgrePaperv9.pdf

[11] G. C. Fox, J. Qiu, S. Kamburugamuve, S. Jha, and A. Luckow, "Hpc-abds high performance computing enhanced apache big data stack," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 2015, pp. 1057–1066.