Toward A Unified HPC and Big Data Runtime

Joshua Suetterlein
University of Delaware
Email: jodasue@udel.edu

Joshua Landwehr, Joseph Manzano, Andres Marquez
Pacific Northwest National Lab
Email: {joshua.landwehr, joseph.manzano, andres.marquez}@pnnl.gov

***Abstract*—The landscape of high performance computing (HPC) has radically changed over the past decade as the community has well surpassed Petascale performance and aims for Exascale. In this effort, chip fabrication and hardware architects have been directly challenged by the fundamentals of physics of chip manufacturing. The effects of these challenges have extended beyond the underlying hardware requiring the attention of the entire stack. As the fight for raw performance continues, a new field in computing has emerged. Big Data Analytics has been heralded as the fourth paradigm of science by turning enormous volumes of data into actionable knowledge. Big Data's influence spans various interests including, commercial, political, and scientific fields.**

While HPC and Big Data seem to approach knowledge discovery from two disparate angles, the technical challenges they face place them on a converging path. Both are required to address scalability, data movement, energy efficiency, and resiliency in large computing systems. Furthermore, in future systems, the copious parallelism will be capable of overwhelming the I/O. This makes the previous methodology of performing scientific simulations and then analyzing the results post mortem unreasonable, giving rise to in-situ analytic techniques.

From a HPC perspective, fine-grain event driven execution models have been proposed as a flexible and efficient model to utilize the underlying hardware. We propose extending a fine-grain execution model to support Big Data techniques as a method to efficiently utilize Exascale resources and as means to join scientific simulation and its analysis overcoming hardware limits (such as limited I/O bandwidth) and reducing the overall time to knowledge discovery.

I. Introduction

Computing plays a pivotal and multifaceted role in how we explore the world around us. Advanced computer systems are used to model, analyze, and interpret data collected from a range of diverse sensors, scientific instruments, and complex simulations. Scientific applications running on these systems include particle physics, DNA-sequencing, and climate modeling and analysis[1], [2], [3]. By providing systems that have more memory and computation power, scientists are enabled to push the boundaries of human knowledge. For this reason, computer scientists, architects, and manufacturers strive to produce larger, faster, and more efficient systems.

Big Data analytics and machine learning has been heralded as the emerging fourth paradigm of scientific discovery [4]. Differing from other paradigms, Big Data analytics are concerned with reliably processing large volumes of data to provide actionable intelligence. Traditionally Big Data is characterized by volume, variety, and velocity [5]. The amount of data generated daily is already exuberant and continues to grow exponentially. This data emanates from numerous sources at a very high rate in varied forms both structured and unstructured. Big Data has gained recognition since it enables query driven insights from trends obscured by the sheer size of the data required to process. While Big Data analytics solicits the interests of corporations and governments alike, it has also provided unique opportunities for science. Fields such as high particle physics, climate science, combustion, biology and genomics, and neutron science are poised to benefit from Big Data analytics and machine learning [3].

The High Performance Computing (HPC) community has pushed the frontiers of computing well beyond Petascale and is nearing Exascale as new systems are capable of achieving a peak performance of hundreds of Petaflops [6]. These systems continue to grow in size and complexity challenging not only manufacturers and system architects, but also application developers and scientists. Chip fabrication and manufacturing is starting to reach fundamental physical limitations such as power/heat dissipation properties of current materials [7], parasitic capacitance and process variation impact at smaller fabrication scales [8], [9], limits on lithography processes [10], among others. Many of these challenges have in some form been passed to the system software stack. Ultimately, these obstacles require solutions as the alternative further passes the burden onto application developers and scientists which in turn obstructs scientific discovery.

II. Problem Formulation

The joining HPC and Big Data has been realized in the new emerging field of High Performance Data Analysis (HPDA) coined by the International Data Corporation (IDC) [11]. This marriage comes at a time when the distinction of both HPC and Big Data workloads continues to blur. The increased resolution and complexity of current models and simulations push the boundaries of HPC workflows. All while Big Data queries and analysis are becoming increasingly more computationally complex [12]. HPDA serves as a newly minted rallying point for hardware manufactures as well as computer scientists seeking to exploit the strengths of HPC and Big Data analytics.

In addition to converging workloads, both HPC and Big Data face several key challenges including energy/power efficiency, exploiting large scale concurrency, effectively handling memory, tolerating network latencies both within and outside a node, and resiliency [2]. From the HPC perspective, one worrisome trend is the drastic decrease of I/O bandwidth to

compute resources [13]. While the core counts are increasing by orders of magnitude, I/O bandwidth is scaling linearly. This furthers the need for exploring in-situ analytics under the HPDA banner.

Big data and HPC face many similar challenges, and share similar approaches in many aspects. As workloads continue to cross single domains and converge to a HPC/Big Data applications, it is important to explore the software stack to reduce inefficiencies. By joining Big Data and HPC the generation of data and its analysis can be overlapped reducing the overall time to discovery.

III. Background

An increasing sector inside the HPC community have proposed shifting execution models from current von-Neumann derived industrial standards like OpenMP and MPI to fine-grain, dataflow-inspired approaches [14]. The key insight being that a fine-granularity of work would be better capable of utilizing the massively concurrent underlying hardware. Such work has been explored in "Codelet" based frameworks such as ParalleX [15], SWARM [16], DARTS [17], Freshbreeze [18], and OCR [19]. We believe these frameworks to be potential vehicles to explore HPDA. One of the codelet based runtime, SWARM, has already served as inspiration and the backbone to a preliminary version of HAMR, a new commercially focused Big Data solution [20].

A. Codelet Execution Model

The codelet execution model explores asynchronous, fine-grain, event driven parallelism. Applications are represented by a graph of (ideally) functional fine-grain tasks called codelets. Codelets are connected by dependencies known as events. Events may go beyond simple data dependencies however, to incorporate control flow, data locality, and more. A codelet fires when all its required events have been satisfied. A codelet execution model is typically realized in the form of a runtime system executing a codelet application on a HPC cluster [21].

The codelet execution model is a promising solution for HPC because of its use of event driven fine-grain parallelism. BSP and fork-join models (e.g. University of Oxford's BSPLib and OpenMP v3.0) struggle to tolerate the varying latencies caused by NUMA domains [22]. Instead, computation units can be underutilized waiting for global barriers. Fine-grained event driven tasks can overcome these latencies by executing tasks not based on control-flow but rather their inherent data dependencies. Shifting from BSP style execution to a more stream based approach is already gaining popularity in Big Data. The exploration of Spark [23] (microbatching) and Storm [24] (streaming) compared to Hadoop's MapReduce [25], [26], parallels codelet based efforts in the HPC community. Moreover, these "streaming" Big Data technologies are still capable of supporting and accelerating the MapReduce paradigm easing the shift of application workloads.

IV. Approach

We believe codelet frameworks present a unique opportunity to join both HPC and Big Data. From the view of a programming model, codelet and Big Data frameworks are similar to Spark and Storm in that the application is divided into asynchronous tasks connected via data dependencies. These models' similarities are furthered when we observe their operational semantics. In all models data is streamed into and out of tasks, which are only executed when data arrives. Given these similarities, our intention is to augment a distributed high performance codelet model with Big Data semantics moving toward a unified framework for HPC and Big Data. These extensions will support MapReduce like operations such as map and reduce on partitions of data akin to Spark and Storm's semantics. Our intention is to treat the extensions as first class citizens in the programming model providing runtime support as shown in figure 1. This approach is preferable, rather than building Big Data layers on top of the Codelet framework, as fine-grain runtimes must have low overhead to be efficient. Moreover, intermixing fine-grain codelets with partition operators provides both HPC and Big Data users new facilities to explore parallelism and performance.

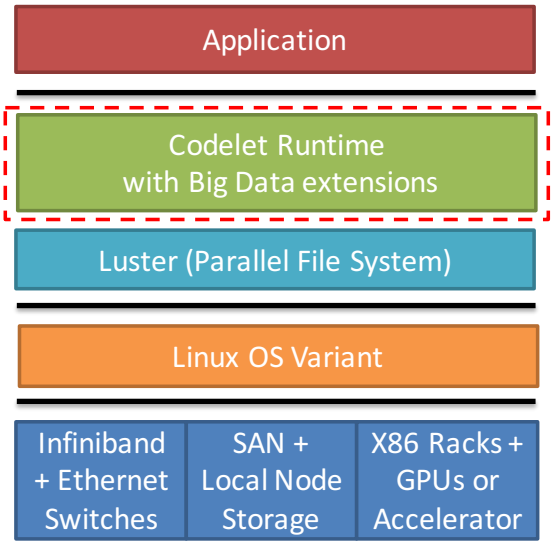

Fig. 1: Proposed HPC/Big Data stack.

V. Conclusion

The distinguishing features of HPC and Big Data workloads have begun to fade. This coupled with the technical challenges both fields face individually has given rise the the joining of the HPC and Big Data software stacks. We believed that the asynchronous, fine-grain, event driven codelet model can server as an effective vehicle in joining the two fields. Leveraging the similarities between current streaming Big Data and fine-grain models, we intend to provide Big Data extensions for the codelet model to explore in-situ analysis and furthering effective research in the HPDA field.

References

[1] R. Rosner *et al.*, "The opportunities and challenges of exascale computing," *US Dept. of Energy Office of Science, Summary Report of the*

Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee, 2010.

[2] R. Lucas *et al.*, "Ten technical approaches to address the challenges of exascale computing," *US Dept. of Energy Office of Science, Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee*, 2014.

[3] J. Chen, A. Choudhary, S. Feldman, B. Hendrickson, C. Johnson, R. Mount, V. Sarkar, V. White, and D. Williams, "Synergistic challenges in data-intensive science and exascale computing," *DOE ASCAC Data Subcommittee Report, Department of Energy Office of Science*, 2013.

[4] R. Kitchin, "Big data, new epistemologies and paradigm shifts," *Big Data & Society*, vol. 1, no. 1, p. 2053951714528481, 2014.

[5] D. Laney, "3D data management: Controlling data volume, velocity, and variety," tech. rep., META Group, February 2001.

[6] J. Hack and M. Papka, "Big data: Next-generation machines for big science," *Computing in Science Engineering*, vol. 17, pp. 63–65, July 2015.

[7] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, (New York, NY, USA), pp. 365–376, ACM, 2011.

[8] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Varius: A model of process variation and resulting timing errors for microarchitects," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 21, pp. 3–13, Feb 2008.

[9] T. Cui, Q. Xie, Y. Wang, S. Nazarian, and M. Pedram, "7nm finfet standard cell layout characterization and power density prediction in near- and super-threshold voltage regimes," in *Green Computing Conference (IGCC), 2014 International*, pp. 1–7, Nov 2014.

[10] "Euv deal raises questions," 2014.

[11] S. Conway, C. Dekate, and E. Joseph, "Worldwide high-performance data analysis 20142018 forecast," tech. rep., International Data Corporation, May 2014.

[12] T. B, R. S. Wagh, and B. S, "Article: High performance computing and big data analytics - paradigms and challenges," *International Journal of Computer Applications*, vol. 116, pp. 28–33, April 2015. Full text available.

[13] B. Hendrikson *et al.*, "Data crosscutting requirements review," *US Dept. of Energy Office of Science, Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee*, 2014.

[14] S. Zuckerman, J. Suetterlein, R. Knauerhase, and G. R. Gao, "Using a "codelet" program execution model for exascale machines: Position paper," in *Proceedings of the 1st International Workshop on Adaptive Self-Tuning Computing Systems for the Exaflop Era*, EXADAPT '11, (New York, NY, USA), pp. 64–69, ACM, 2011.

[15] A. Tabbal, M. Anderson, M. Brodowicz, H. Kaiser, and T. Sterling, "Preliminary design examination of the parallex system from a software and hardware perspective," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 4, pp. 81–87, 2011.

[16] C. Lauderdale and R. Khan, "Towards a codelet-based runtime for exascale computing: Position paper," in *Proceedings of the 2Nd International Workshop on Adaptive Self-Tuning Computing Systems for the Exaflop Era*, EXADAPT '12, (New York, NY, USA), pp. 21–26, ACM, 2012.

[17] J. Suetterlein, S. Zuckerman, and G. Gao, "An implementation of the codelet model," in *Euro-Par 2013 Parallel Processing* (F. Wolf, B. Mohr, and D. an Mey, eds.), vol. 8097 of *Lecture Notes in Computer Science*, pp. 633–644, Springer Berlin Heidelberg, 2013.

[18] J. B. Dennis, G. R. Gao, and X. X. Meng, "Experiments with the fresh breeze tree-based memory model," *Comput. Sci.*, vol. 26, pp. 325–337, June 2011.

[19] T. Mattson *et al.*, "Ocr the open community runtime interface v0.9," tech. rep., OCR Working Group, 2014. https://xstackwiki.modelado.org/images/1/13/Ocr-v0.9-spec.pdf.

[20] B. Hellig, S. Turner, R. Collier, and L. Zheng, "Beyond mapreduce: The next generation of big data analytics," 2014.

[21] G. R. Gao, J. Suetterlein, and S. Zuckerman, "Toward an execution model for extreme-scale systems runnemede and beyond.," Tech. Rep. CAPSL Technical Memo 104, Department of Electrical and Computer Engineering, University of Delaware, Newark, Delaware, May 2011.

[22] E. Garcia, D. Orozco, R. Khan, I. E. Venetis, K. Livingston, and G. R. Gao, "Dynamic percolation: A case of study on the shortcomings of traditional optimization in many-core architectures," in *Proceedings of the 9th Conference on Computing Frontiers*, CF '12, (New York, NY, USA), pp. 245–248, ACM, 2012.

[23] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2010.

[24] M. T. Jones, "Process real-time big data with twitter storm: An introduction to streaming big data," tech. rep., IBM, Nov. 2012.

[25] D. Gillick, A. Faria, and J. DeNero, "Mapreduce: Distributed computing for machine learning," 2006.

[26] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, pp. 107–113, Jan. 2008.