Big Data and Extreme Scale Computing, 2nd Series, (BDEC2) Workshop 1: Bloomington, Indiana, November 28-30, 2018 Call for White Papers

The primary goal of the BDEC2 workshops is to help develop and build consensus around a new, shared, advanced cyberinfrastructure platform (ACP) that can address the disruptive challenges currently roiling the research ecosystems of nearly every field of science and engineering. Unsurprisingly, *the critical considerations in the design of such an ACP are the goals, strategies, and the requirements of the scientific application communities that will use it*. Unfortunately, the sea change we are witnessing will make those elements harder to specify. In the past, requests for application requirements have typically been met with 1) descriptions of current and near-future application bottlenecks and 2) a request for more or better versions of technologies that are either already are or will soon be available. As the BDEC Pathways report argues, however, the tsunami of change that many see coming will demand that research communities re-envision how their goals and requirements must evolve to best exploit the opportunities for discovery and innovation being created by the device and data-saturated world that is now emerging.

The BDEC Pathways report identifies at least three prominent trends that application communities should take into account as they think through their cyberinfrastructure needs for the future:

- *Novel models of integrated inquiry*: Unprecedented new methods in high-end data analysis (HDA) that are being pioneered in Big Data communities, such as Deep Learning, will increasingly be combined and integrated with the simulation-centric approaches of traditional high performance computing (HPC). In other words, the impact of the digital revolution on scientific methodologies has entered a dramatic new phase.
- *Support for advanced data logistics*: It seems clear that many, if not most fields will require new cyberservices that can help manage the logistics of data pouring out of devices at all levels of the "digital continuum." This continuum stretches from the proliferating host of new instruments coming online, to the sensor networks and cyber-physical systems now spreading out across the human and natural landscape, to the massive results of extreme-scale simulations, which threaten to swamp even the largest HPC systems.
- *Interfaces to commercial cyberinfrastructure*: As public funding for shared community resources continues to recede, massive and administratively centralized resources of different commercial cloud providers are delivering an ever increasing share of the computing, storage, and networking resources that different science and engineering

communities require. This situation is unlikely to change any time soon, especially for the majority of researchers who work in the "long tail" of a given field

Against the background of these broad concerns, *participants in the BDEC2 workshop are invited to submit a 1-2 page White Paper/Statement of Interest in areas of relevance to the goals of the BDEC2 workshops*. The particular (although not exclusive) focus of the Bloomington workshop is on *application community goals and requirements*, which are especially pertinent to the ACP design challenge. These goals and requirements are likely to be substantially affected, if not dramatically transformed, by the three factors mentioned, as well as others. Both application and technology focussed white papers and indeed in between are welcome.

White papers will be selected for presentation at the meeting based upon relevance of submissions, as well as the balance of strategic vision and expertise. However, all white papers submitted will be published online with the other products of the workshop, and short summaries of those papers which are not presented will be given at the meeting. White papers are due by Nov. 5 and should be sent to Terry Moore (tmoore@icl.utk.edu). If you have any questions, please contact BDEC2 workshop organizers: Jack Dongarra (dongarra@icl.utk.edu), Pete Beckman (beckman@mcs.anl.gov), Geoffrey Fox (gcf@indiana.edu), and Dan Reed (dan.reed@utah.edu).

A Collection of White Papers from the BDEC2 Workshop in Bloomington, IN

November 28–30, 2018

Acknowledgment

This workshop was supported in part by the National Science Foundation under Grant No. 1849625.

A vision for a validated distributed knowledge base of material behavior at extreme conditions using the Advanced Cyberinfrastructure Platform

James Ahrens and Christopher M. Biwer

White paper response to Big Data and Extreme Scale Computing, 2nd Series, (BDEC2)
Workshop 1: Bloomington, Indiana

As materials age or undergo extreme conditions (eg. shock), their physical properties can leave acceptable ranges which jeopardizes their safety and effectiveness in applications. Beamline facilities are a unique tool to probe how these materials properties (eg. strength and ductility) vary as a function of composition, time, stress, and other conditions [1,2]. This is critical knowledge to inform models predicting the performance of materials in applications. Due to the importance of these experiments in understanding materials, Europe, Japan, and the United States are commissioning X-Ray Free Electron Laster (XFEL) facilities with increased beam intensity, shot repetition rates, and detector sizes that enable new experiments as well as higher throughput rates for experiments [3,4,5]. Technological advances at synchrotron X-ray and the emerging X-ray Free Electron Laser (XFEL) facilities have realized data collection frequencies at 100 Hz, and MHz repetition rates are anticipated in the next few years [4].

Despite the significant advancements at facilities, open questions remain how data analysis methods will be applied given the data rates at these facilities, and in particular, whether the feedback from a real-time analysis can improve experimental outcomes at the facility, when an experiment is typically a few days. Requirements for local computer systems to triage the raw data and super computers to process the resulting information have been identified, similar to the envisioned ACP architecture.

The XFEL community has only begun to grapple with these data analysis challenges, and therefore, now is the opportune time to direct the cyberinfrastructure development. The BDEC call for papers identifies three approaches that the scientific community should use to address their cyberinfrastructure needs, and the following three sections describe unique requirements and ideas emerging from the XFEL community in the context of the three BDEC approaches.

Novel and/or converged models of inquiry: Predictive modeling of materials in applications relies on accurately parameterized models that describe the material's strength and plasticity (ie. tendency to deform). The parameterization and validation of the constitutive strength and plasticity models requires comparing simulations of the experiment and experimental results. Manual efforts to perform this parameterization of the strength and plasticity models using experimental data (which may be indirect measurements of the model parameters) often lead to non-unique parameterizations without uncertainty. Machine learning techniques such as Gaussian process modeling [6] provides a statistical framework to infer model parameter values and to quantify the uncertainty of each parameter. Gaussian process modeling uses an ensemble of hydrodynamic simulations to construct an emulator (ie. a surrogate model) that mimics the outputs of the computationally intensive hydrodynamics simulations. The emulator is calibrated with experimental data to infer the strength and plasticity model parameters which are inputs to the hydrodynamics simulation of the experiment.

The distribution in the parameter space and quantity of experimental data have a direct impact on the uncertainty of model parameters in Gaussian process modeling. Therefore, a strategy for experimental design can be to perform new experiments that minimize the uncertainty of the inferred parameters, which improves the parameterization of the strength and plasticity models. This approach to experimental design maximizes the science capability of beamline experiments which have a limited allotment of time to use the facility.

Support for advanced data logistics: Due to the time constraints an experiment has at these facilities, any analysis designed to provide real-time feedback, to direct experimental design, must be executed quickly.

However, Gaussian process modeling uses a Markov-chain Monte Carlo [7,8] which can require time equal to a significant portion of the experimental time at the beamline facility. One strategy for reducing the analysis time in Gaussian process modeling is pre-computing components of the analysis prior to arrival at the beamline facility. For example, the step which uses an ensemble of simulations to construct an emulator could potentially be computed beforehand. Aside from expediting the parameterization analyses, an emulator can predict the output of the hydrodynamics simulation for a given set of parameters in less than a second. Therefore, the emulator could be used at the beamline facility to stage and execute approximations of the simulations at a significantly reduced computational cost. Further reductions in the analysis time could be achieving using approximate algorithms that trade computational time at the cost of increasing the uncertainty of model parameters.

Support for interfaces to commercial cyberinfrastructure: The concept of optimizing of the experimental facilities resources can be extended to optimizing the computing resources. Since an experiment could benefit multiple research groups, or sites, the flexible cost model of commercial cyberinfrastructure could be shared amongst sites. Sites could choose to contribute to the cost of analysis in order to improve metrics such as time to completion, accuracy (ie. uncertainty reduction), data transfer, and resources costs that align with their particular research goals. The data and simulations are shared between all the sites, and therefore the sites collectively improve the quality of the analysis as well as reduce the overall number of computing resources required.

Vision: Competitive/cooperative science teams build their own repositories for raw data, simulation models, and models of previous experiments at experimental and supercomputing sites on the ACP. There is a multi-resolution transfer of raw simulation and experimental data between these sites, focused by the teams, on improving local repositories of selected materials in physical regions where modeled predictions differ from the real-world results. Each team collectively improves their data to increase their understanding. Exchanges both pull and push based on complex cost metrics that optimize what each site wants to achieve. Open science strategies could lower the cost of data based on its age, rewarding teams with privileged access and exchange capabilities for a fixed window of time.

[1] H. M. Rietveld. J. Appl. Cryst. **2**, 65-71 (1969).
[2] L. M. Barker and R. E. Hollenbach. J. Appl. Phys. **43**, 4669 (1972).
[3] U. Zastrau et al. REPORT-2017-004. XFEL.EU TR-2017-001 (2017).
[4] J. Thayer et al. Adv. Struct. Chem. Imaging **3**(1):3 (2017).
[5] S. Owada et al. J. Synchrotron Rad. **25**, 282-288 (2018).
[6] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, J. Am. Stat. Assoc. **103**, 570 (2008).
[7] N. Metropolis et al. J. Chem. Phys. **21**, 1087–1092 (1953).
[8] W. K. Hastings. Biometrika **57**, 97–109 (1970).

Objective Driven Computational Experiment Design: An ExaLearn Perspective

Francis J. Alexander[1] and Shantenu Jha[1,2]
[1] **Brookhaven National Laboratory**
[2]**Rutgers University**

Overview

A fundamental problem that currently pervades diverse areas of science and engineering is the need to design expensive computational campaigns (experiments) that are robust in the presence of substantial uncertainty. A particular interest lies in effectively achieving specific objectives for systems that cannot be completely identified. For example, there may be "big data" but the data size may still pale in comparison with the complexity of the system, or the available data may be scarce due to the prohibitive cost of the relevant experiments.

In current practice, the methodologies by which experiments inform theory, and theory guides experiments, remain ad hoc, particularly when the physical systems under study are multiscale, large-scale, and complex. Off-the-shelf machine learning methods are not the answer—these methods have been successful in problems for which massive amounts of data are available and for which a predictive capability does not rely upon the constraints of physical laws. The need to address this fundamental problem has become urgent, as computational campaigns at pre-exascale, and soon exascale, will entail models that span wider ranges of scales, represent richer interacting physics, and inform decisions of greater societal consequence.

To facilitate the design of computational campaigns across multiple scientific domains, diverse objectives and measures of robustness, we are developing a computational capability for **objective driven experimental design** (ODED) using RADICAL-Cybertools as part of the recently funded DOE ECP Co-Design Center "ExaLearn". This ODED framework will support the integration of scientific prior knowledge on the system with data generated via simulations, quantify the uncertainty relative to the objective, and design optimal experiments that can reduce the uncertainty and thereby directly contribute to the attainment of the objective.

Importance of ODED at Exascale

Although ODED has been always been important, its significance increases drastrically at the exascale. First, computational resources are too expensive for the design of computational campaigns to be conducted in an ad hoc fashion and for the resulting data not to be exploited to its very fullest. Second, is the need to enhance if not preserve computational efficiency at exascale. The ability to apply high-performance computing capabilities at scale, leads to the possibility of greater inefficiency in computational exploration. For example, greater computational capacity might generate relatively greater correlations, and thus less independent data, or lesser sampling.

The interaction between models and data occurs in two directions: (i) The problem of how to use multi-modal data to inform complex models in the presence of uncertainty, and (ii) How, where, when, and from which source to acquire simulation data to optimally inform models with respect to a particular goal or goals is fundamentally an optimal experimental design problem. Creating the conceptual and technological framework in which models optimally learn from data and data acquisition is optimally guided by models—presents significant challenges systems of interest are complex, multiscale, strongly interacting/correlated, *and* uncertain. These challenges must be overcome to realize the promise of efficiency and effective exascale computing. Our framework is designed to ideally support the mathematical developments while being agnostic of any specific formulation.

Example Science Driver: Objective Driven Computational Drug Design

The strength of drug binding is determined by a thermodynamic property known as the binding free energy. One promising technology for estimating binding free energies and the influence of protein and ligand composition upon them is molecular dynamics (MD). A diversity of methodologies have been developed to calculate binding affinities; MD sampling and blind tests show that many have considerable predictive potential. With the demands of clinical decision support and drug design applications in mind, several computational protocols to compute binding free energies have been designed. Different protocols (algorithmic methods) typically involve compounds with a wide range of chemical properties which can impact not only the time to convergence, but the type of sampling required to gain accurate results [1,2]. The advantages of determining optimal computational campaigns include: (i) Greater sampling and higher throughput of drug candidates; (ii) more accurate binding affinity calculations, and (iii) Efficient resource utilization.

Mathematical Formulation of Objective Driven Drug Design Campaign: We believe the problem of determining an optimal computational campaign for a given objective (`O`) under a given constraint (`C`) can be formulated as: Imagine there are M different stochastic algorithms which calculate the same quantity of interest but with different variances (errors) for the same amount of computational resource. Conversely, the distribution of run times (t) for a given algorithm at a prescribed variance level (σ) on a given computing resource R is given by P$_R$(σ,t). A priori we dont know this distribution P$_R$(σ,t), but we can learn it from ongoing experiments. One possible constraint (`C`) could be a fixed amount of overall computational resources available to run the algorithms (which can be run in parallel); another constraint could be to get the "optimal" answer in a given wall clock time. An objective could be to find the selection of algorithms so as to minimize the variance of the estimate of the mean, for the given constraints.

Software System for High-Performance Objective Driven Experimental Design

To promote interoperability and reuse across different scientific problems, objectives and optimization criterion, the software systems for high-performance optimal experimental design must be architected and implemented to support diverse usage modes, different combinations of capabilities with minimal customization, refactoring or new development.

At Brookhaven National Laboratory, we are developing the ODED framework to determine "optimal" surrogates for expensive cosmological simulations, to optimize computational campaign configurations for drug discovery and optimal model selection for materials and climate science probelms. The ODED framework is being developed in collaboration with RADICAL Laboratory at Rutgers and it leverages the RADICAL-Cybertools – a Building Blocks (BB) approach for HPC middleware [3]. It will utilize existing BB for sophisticated and scalable management and execution of ensemble and optimization style workflows.

References

[1] J Dakka, K Farkas-Pall, V Balasubramanian, M Turilli, S Wan, D Wright, S Zasada, P Coveney, S Jha: Enabling Trade-offs Between Accuracy and Computational Cost: Adaptive Algorithms to Reduce Time to Clinical Insight. CCGrid 2018: 572-577

[2] Concurrent and Adaptive Extreme Scale Binding Free Energy Calculations. J Dakka, K Farkas-Pall, M Turilli, S Wan, D Wright, P Coveney, S Jha: published in IEEE eScience 2018 (arXiv 1801.01174)

[3] M Turilli, A Merzky, V Balasubramanian, Shantenu Jha: Building Blocks for Workflow System Middleware. CCGrid 2018: 348-349

The Sigma Data Processing Architecture: Leveraging Future Data for Extreme-Scale Data Analytics to Enable High-Precision Decisions

Gabriel Antoniu[1], Alexandru Costan[1], Maria S. Pérez[2], Nenad Stojanovic[3]

[1]Univ Rennes, Inria, CNRS, IRISA

[2]Universidad Politécnica de Madrid

[3]Nissatech

6 November 2018

This white paper introduces several key principles based on which HPC-Big Data convergence can be achieved: 1) use future (simulated) data to substantially enrich knowledge obtained based on historical (past) data; 2) enable high-precision analytics thanks to hybrid modeling combining simulation and data-driven models; 3) enable unified data processing thanks to a data processing framework able to relevantly leverage and combine stream processing and batch processing in situ and in transit.

Due to an ever-growing digitalization of the everyday life, massive amounts of data start to be accumulated, providing larger and large volumes of historical data (**past data**) on more and more monitored systems. At the same time, an up-to-date vision of the actual status of these systems is offered by the increasing number of sources of real-time data (**present data**). Today's data analytics systems correlate these two types of data (past and present) to predict the future evolution of the systems to enable decision making. However, the relevance of such decisions is limited by the knowledge already accumulated in the past.

At the same time, companies and organizations do not only want to learn from the past, but also aim to get a precise understanding on what might happen next in any circumstances, in particular how to react to unknown events. A timely and relevant example is the connected vehicle (e.g., vessel or car) with autonomy facilities: on one side, computational simulation models [Chinesta2013] are used to simulate the vehicle's behavior in various hypothetical conditions in order to improve its design; on the other side, data-driven analytics models [Ibanez2017] are used to monitor and control the system in real time during its operation, to support vehicle motion through advanced sensing. Combining knowledge from both models appears as a high-potential opportunity to substantially improve the vehicle's performance and maintenance process.

More generally, we witness a huge expansion of complex systems and processes where the knowledge acquired based on **past data** can substantially be extended with what could be called **future data** generated by simulations of the system behavior under various hypothetical conditions that have not been met in the past. This can provide a richer tool for much deeper interpretation of measured real-time data, enabling much more relevant decision making, beyond what is currently enabled by history-based prediction methods.

Challenges. Enabling the use of the future data jointly with past and present data leads to several challenges:
The overall challenge is to define and validate the most appropriate methodology that supports the combination of the two modelling paradigms in a way that exploits the potential for synergies. This translates into:

- A **challenge regarding the data analytics model**: combine computational and data-driven analytics models through hybrid modeling.
- A **challenge regarding the data processing architecture**: efficiently integrate simulations and data analytics through a unified data processing architecture combining traditional Big Data processing (batch- and stream-based) with HPC techniques for data processing (in situ, in transit) to support hybrid analytics models.
- **A challenge regarding continuous model improvement**: simultaneously refine both the data-driven model and the computational model through continuous learning, with the goal of optimizing the real-world system behavior.

Future Data: from Big Data to Extreme Data. Combining computation-driven and data-driven analytics can reach full potential only if the two types of data analytics efficiently leverage each other to detect the best opportunities to improve the system operation, but also to react in an optimal way to critical situations. This leads to high challenges related to the extreme scale of data management in terms of both volume and velocity.

- First, the number of hypothetical scenarios, the possibility to simulate them with a virtually unlimited combination of parameters and the possibility to run joint data analysis correlating such hypothetical data with past measured data as well as with real-time data coming from the real system may produce immense amounts of data to process (**extreme volume**).
- At the same time, performing such a complex analysis on a virtually immense number of scenarios in order to find the most efficient way to react in real time to some critical situations that require very fast decision poses a challenge in terms of **extreme velocity** for data processing, which can require extreme-scale computations performed on extreme-scale HPC infrastructures.

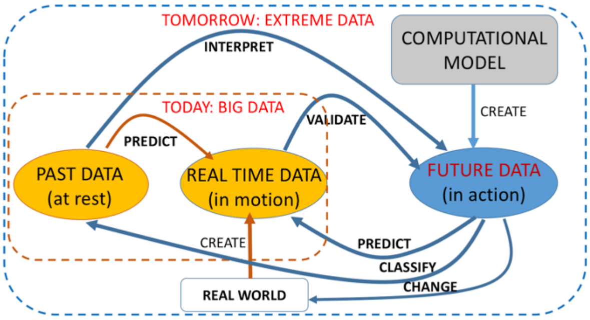


Figure 1. From Big Data to Extreme Data

The need for Hybrid Analytics Models. So far, data-driven analytics and simulation-based modelling have been developed and used separately. They involve quite different cultures and types of expertise: numerical models typically simulated in HPC environments on one side; Big Data analytics models and technologies on clouds on the other side. The two types of modelling provide complementary views:

- Data-driven analytics enables understanding/learning the real system's behavior and its rationale from past data, leveraging machine learning/deep learning techniques;
- Simulation based on complex computational models can provide data to forecast the system's performance in various possible situations (including some that have never occurred yet).

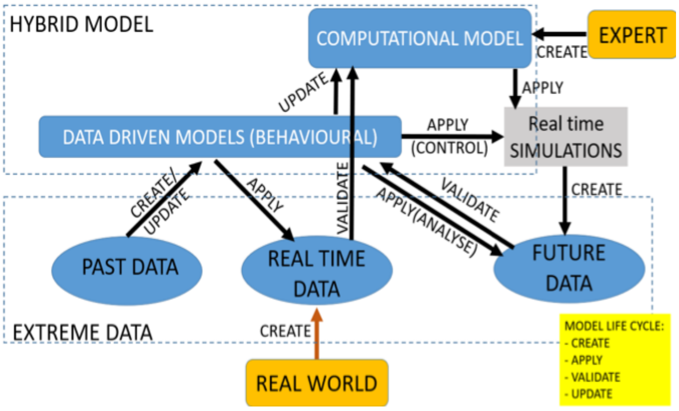


Figure 2. Hybrid Analytics Models

By combining information from both models, hybrid analytics models can be built: they can substantially augment knowledge for understanding and predicting the system's behavior: the knowledge based on potential future behavior will substantially enrich the knowledge acquired based on past behavior.

Continuous model improvement. The quality of the simulation model impacts the quality of the future data they generate and, thereby, the accuracy of the predictions of future events. To improve the simulation model, several techniques can be used. First, acquired data from the real system can be used for calibration. Second, the model can also be corrected based on experience with the system operation, by exploiting a behavioral model built based on deviations between predictions and measurements (hybrid model). Finally, in order to avoid jeopardizing real-time feedback, this process should consider only valuable, high-impact data that increases the knowledge encapsulated into the model correction. The challenge is to succeed in efficiently leveraging these relevant data as a key approach to enrich the physics-based simulation model, allowing it to produce better (faster and more accurate) future data. At the same time, the data-driven model is subject to continuous improvement, by assimilating simulation data.

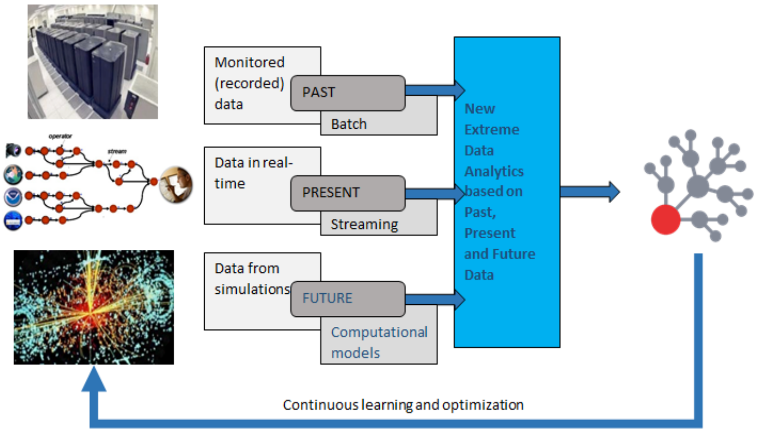

operator
stream
Monitored (recorded) data
PAST
Batch
Data in real-time
PRESENT
Streaming
Data from simulations
FUTURE
Computational models
New Extreme Data Analytics based on Past, Present and Future Data
Continuous learning and optimization

Figure 3. Using learning to continuously update and refine the models.

Enabling Hybrid Analytics: the Sigma Architecture for Data Processing. Traditional *data-driven analytics* relies on *Big Data processing* techniques, consisting of *batch processing* and *real-time (stream) processing*, potentially combined in a so-called *Lambda architecture*. This architecture attempts to balance latency, throughput, and fault-tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data.

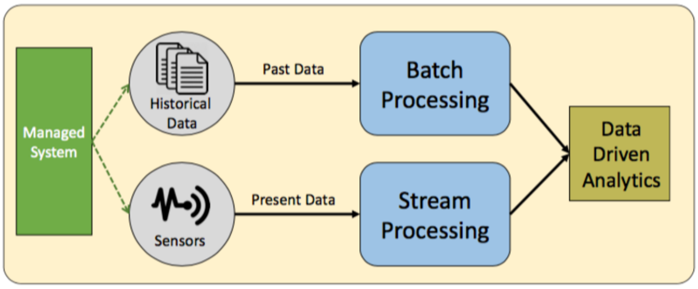


Figure 4. The Lambda data processing architecture.

On the other side, *simulation-driven analytics* is based on computational (usually physics-based) simulations of complex phenomena, which often leverage HPC infrastructures. The need to get fast and relevant insights from massive amounts of data generated by extreme-scale simulations led to the emergence of *in situ* and *in transit* processing approaches [Bennet2012]: they allow data to be visualized and processed interactively in real-time as data are produced, while the simulation is running.

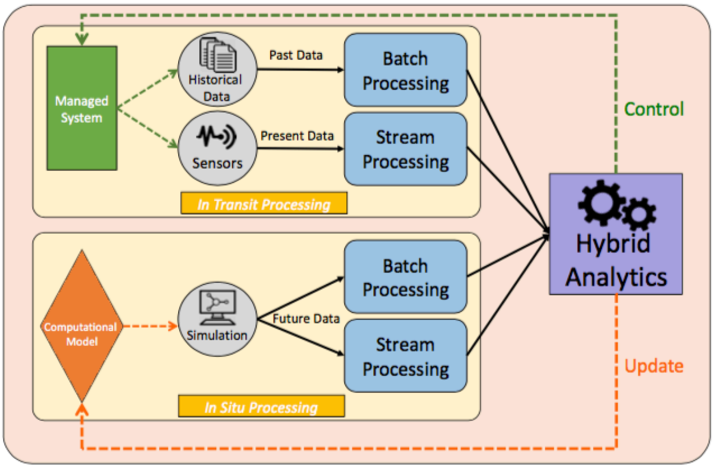


Figure 5. The Sigma data processing architecture.

To support hybrid analytics and continuous model improvement, we propose to combine the above data processing techniques in what we will call the **Sigma architecture**, a HPC-inspired extension of the Lambda architecture for Big Data processing. Its instantiation in specific application settings depends of course of the specific application requirements and of the constraints that may be induced by the underlying infrastructure. Its main conceptual strength consists in the ability to leverage in a unified, consistent framework, data processing techniques that became reference in HPC in the Big Data communities respectively, without however being combined so far for joint usage in converged environments.

Conclusion

We believe the principles sketched out above put in place a sound base for making a step further towards the convergence of the HPC and Big Data Analytics areas. The Sigma architecture can be established as a new paradigm enabling convergence at data processing level. Based on it, hybrid modeling can become a reference paradigm for data analytics. We see these two paradigms as key enablers of extreme data analytics for emerging application scenarios that will efficiently leverage converged HPC-Big Data infrastructures to enable unprecedentedly high decision making in complex environments.

References

[Chinesta2013] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar et A. Huerta. Parametric PGD based computational vademecum for efficient design, optimization and control. Archives of Computational Methods in Engineering, 20/1, 31-59, 2013.

[Ibanez2017] R. Ibanez, E. Abisset-Chavanne, J.V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta. A Manifold-Based Methodological Approach to Data-Driven Computational Elasticity and Inelasticity. Archives of Computational Methods in Engineering, 2017.

[Bennet2012] J.C. Bennet, H. Abbasi, P.-T. Bremer, R. Grout et al. Combining in-situ and in-transit processing to enable extreme-scale scientific analysis. In Proc. ACM SC'12, Salt Lake City, Nov. 2012.

[Carbone2015] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas, Apache Flink: Stream and batch processing in a single engine, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 36 (4).

[Marcu2017] O. Marcu, A. Costan, G. Antoniu, M.S. Pérez, R. Tudoran, S. Bortoli and B. Nicolae, Towards a Unified Storage and Ingestion Architecture for Stream Processing. IEEE International Conference on Big Data (Big Data), 2402–2407; 2017. doi: 10.1109/BigData.2017.8258196

Workflow environments for advanced cyberinfrastructure platforms

Rosa M Badia, Barcelona Supercomputing Center

Progress in science is deeply bound to the effective use of computing infrastructures and to the efficient extraction of knowledge from vast amounts of data. This involves large amounts of data coming from different sources, that follow a cycle composed of pre-processing steps for data curation and preparation for subsequent computing steps, and later analysis and analytics steps applied to results.

In the design of an advanced cyberinfrastructure platform (ACP), one of the key elements is how to describe the applications to be executed in such platform. Most of the times these applications are not standalone, but involve a set of sub-applications or steps composing a workflow. The scientists then rely on effective environments to describe their workflows and engines to manage them in complex infrastructures. From our point of view, current methodologies to describe workflows should be extended in several ways to fulfil the requirements of novel ACPs.

First, most of current approaches focus more on the computing part of the workflow, ignoring the data dimension. Current and, even more, future scenarios will involve large amounts of data coming from different sources (myriad of remote sensors, major scientific instruments, satellites, etc). As described in the BDEC white paper [bdec_paper], while the scientific process involves High-end Data Analysis **(**HDA) steps (abduction and induction), and High-Performance Computing (HPC) steps (deduction), current scientific workflows are performing the three different steps of the scientific process with separated methodologies and tools, with a lack of integration and lack of common view of the whole process. One of the BDEC recommendations is to address the basic problem of the split between the two paradigms: the HPC/HDA software ecosystem split.

We believe that the split between HPC and HDA is due to the fragmentation of the traditional scientific computational workflows into separated components, which use different programming models and different environments for HDA and HPC, with a lack of a global perspective. The huge amount of data and its format heterogeneity, both for the data generated from observations and from deduction, makes very difficult the generation of scientific conclusions. The developers of scientific applications are faced with all this amount of data, large number of data analytics methodologies and HPC tools. Therefore, there is a need for workflow environments and tools for the development of scientific workflows following a holistic approach where both data and computing are integrated in a single flow built on simple, high-level interfaces. Topics of research are novel ways to express the workflows that integrate the different data and compute processes, dynamic runtimes to support the execution of the workflows in complex and heterogeneous computing infrastructures in an efficient way, both in terms of performance and energy.

Besides, the focus of the different scientific and technological user profiles involved in the process may differ. While the emphasis from the computer science point of view has traditionally been on the programming models and applications used to make the predictions and simulations (deduction phase), the scientific application developers give much more emphasis on the data aspect of the problem: metadata that describes the data and traceability of the data that describes how it has been generated or transformed is even more important for them (abduction and induction phases). For this reason, it is common to see scientific workflows environments originated by scientific communities that run very inefficiently (i.e., low CPU utilization) in large HPC systems, while workflow environments that are able to get better performance are not adopted by scientific communities. This can be generalized by differentiating multiple abstraction levels seen by different user profiles that are involved in the use/development of a scientific workflow, from the final user that seeks a higher level of abstraction to define its application, to the computer engineer that faces herself with a lower level of abstraction with all the details of the infrastructure. In this sense, the design of new workflow environments should follow a multidisciplinary approach, with experts from different Computer Science (CS) fields (machine learning, parallelism, distributed computing) and from application fields involved, in order to define these new methodologies that will support advances in scientific research and knowledge progress. Application providers will contribute to the research together with the CS experts in order to design the workflow environments that better reflect the specific way of understanding their scientific workflows. In this sense, for the different areas of application, different solutions can be designed. What is more, different abstraction levels on the workflow methodologies can be considered to meet the expectations of different scientific and technological user profiles (final user, application developer, research support, computer engineer).

And finally, all these differences and complexities, we need to add the complexity of the current computational infrastructure. We are faced with new processor architectures and of different types (general purpose processors, graphic processors, programmable devices), new persistent storage technologies and new ways of interconnecting all the elements of these complex systems. HPC systems coupled with public and private Cloud infrastructures, and what is more, the systems where future scientific workflows are to be executed will also include edge devices, sensors and scientific instruments that will be able to do computation in the edge and to stream continuous flows of data. Similarly, the scientists expect results to be streamed out for monitoring, steering and visualization of the scientific results to enable interactivity. It is necessary to provide the workflows with powerful runtimes, able to make autonomous decisions in order to execute the scientific workflows in efficient ways in complex data and computing infrastructures, both in terms of performance and energy consumption. The runtime should be able to take decisions in a very dynamic fashion, to enable the exploration of the workflow design space in an intelligent manner, to boost the time to solution. Techniques such as automatic parallelization, machine learning, optimization of data and metadata management, should be present in the runtime. Also, the runtime will be able to deal with the vast and heterogeneous nature of the infrastructures, being able to get the best from them, but keeping the scientific workflow agnostic of them.

[bdec_paper] Asch, M., T. Moore, R. Badia, M. Beck, P. Beckman, T. Bidot, F. Bodin et al. "Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry." The International Journal of High Performance Computing Applications 32, no. 4 (2018): 435-479.

Convergence of data generation and analysis in the biomolecular simulation community

Oliver Beckstein[1], Geoffrey Fox[2], Shantenu Jha[3,4]

[1]Arizona State University, Tempe AZ <obeckste@asu.edu>; [2]Indiana University, Bloomington IN <gcf@indiana.edu>; [3]Rutgers University, Piscataway NJ; [4]Brookhaven National Laboratory <shantenu.jha@rutgers.edu>

The changing nature of biomolecular simulations

In the biomolecular simulation (BMS) community, classical molecular dynamics (MD) simulations enable the elucidation of the relationship between the structure of biomolecules such as proteins, nucleic acids, or lipids and their function via their dynamics. MD simulations account for approximately one quarter of the service units used on XSEDE resources. Although traditionally the generation of the data has been the computational bottleneck and has been highly optimized, more and more the analysis of the data is becoming a rate limiting step. Within the NSF DIBBs SPIDAL project we have been working on leveraging HPC resources for the analysis of BMS data [1], starting from two widely adopted software packages in the community, cpptraj [2] and MDAnalysis [3,4].

Current state of the art simulations are performed at the atomic level and include the niomolecules and their environment such as water, ions, lipids, and small molecules. Typical system sizes range from $O(10^3)$ to $O(10^6)$ atoms with some exceptionally large systems up to $\sim 10^8$ [5]. Simulations integrate the equations of motion of all atoms using femtosecond timesteps. The positions (and possibly velocities) of the atoms are saved to a trajectory file at regular intervals, typically every 1 to 100 ps. Current simulation lengths typically achieve ≤ 10 μs although on special hardware up to 1 ms has been achieved for small systems with $O(10^4)$ atoms [6], while massively distributed simulations can produce aggregate data of up to 6 ms [7]. Advances in hardware (GPUs, FPGA/custom hardware, exascale resources such as Summit) and software (e.g. GPU-optimized codes) lead to a steady increase in the trajectory sizes [8], currently in the hundreds of GB to a few TB range.

A common approach is to run a single or a few repeats of simulations for a fixed condition with the aim to capture equilibrium behavior. Long continuous trajectories have provided valuable insights in protein folding in a unbiased manner [6]. However, increasingly the emphasis is on sampling of rare events and quantitative predictions of free energies and rates, which necessitates enhanced sampling approaches [9, 10, 11] that run ensembles of tens to hundreds of coupled simulations. Exascale computing promises to make such calculations much more feasible and more widespread. The trajectories from enhanced sampling runs have to be analyzed as a single dataset; the size of the datasets (approaching hundreds of TB [12]) will make it infeasible to move the data away from the HPC system where they were produced and their analysis will take too long with current serial approaches. The challenge becomes to design computational environments that support both data generation and analysis efficiently and to develop analysis software that makes best use of resources that have been geared towards data production.

From offline to online analysis

Thus, concomitant with increased computing capabilities is the opportunity and the need for sophisticated and efficient analysis of unprecedented volumes of data generated from simulations. The temporal coupling of Molecular Dynamics simulations generating data (producer) to analysis of the data (consumer) can be classified into three broad categories,

I. *Data Reduction*: This is classic scenario, where in-situ (real-time) analysis of data is performed to reduce the volume of data that needs to eventually be stored or output to disc. Original drivers of data reduction were poor file system performance, but recent advances in the ability to "compute only what you need" [13] scenarios has resulted in several approaches to analysing data once and only once.

2. *Streaming Data into Analysis*: There have been advances in stream-based algorithms of traditional analysis algorithms which benefit from incremental data availability, thus necessitating the ability of large volumes of data to be streamed directly from simulations to analysis. The need to stream data directly into simulations is not confined to stream-based analysis algorithms; several online learning algorithms [14] benefit from increased and incremental data.

3. *Adaptive Simulations*: Arguably the coupled simulation-analysis scenario that has received the greatest attention thus far, is the general class of algorithms referred to as adaptive algorithms, and in particular adaptive ensembles simulations [15]. In adaptive algorithms the intermediate data generated by simulations is used to guide the evolution of the next stage of simulations. Traditional examples of these include Markov State Model (MSM) and variants thereof, but recently more sophisticated ML-driven approaches to steering simulations (ML-driven-MD) have been both proposed and implemented [11]. The motivation for adaptive simulations varies from "better, faster and greater" sampling of a very large phase space [15], to the efficient utilization of limited computing resources [16]. Ref [17] discusses a software system that supports multiple adaptive algorithms that significantly increase simulations efficiency.

AI-driven analysis and simulation

ML is being used to analyze the results of molecular dynamics simulations (e.g., binding affinities [12], folding [14], phase diagrams [20], or Tang's contribution to this meeting predicting stability). An exciting idea is to use AI-driven analysis to advance MD simulations or whole ensembles of simulations based on the phase space that has already been sampled. Such AI-driven autotuning has been shown to be able to increase time steps in QM MD [19] and suggest parameters (such as timestep size, spatial meshes, internal polarization densities) to be used in the simulation [21] but more widespread application of these ideas will likely require meeting the challenges of converging simulations and analysis as outlined above.

References

[1] I. Paraskevakos, A. Luckow, M. Khoshlessan, G. Chantzialexiou, T. E. Cheatham, O. Beckstein, G. Fox, and S. Jha. Task-parallel analysis of molecular dynamics trajectories. In ICPP 2018: 47th International Conference on Parallel Processing, August 13–16, 2018,

Eugene, OR, USA, New York, NY, USA, August 13–16 2018. Association for Computing Machinery, ACM.

[2] D. R. Roe and T. E. Cheatham. PTRAJ and CPPTRAJ: Software for processing and analysis of molecular dynamics trajectory data. Journal of Chemical Theory and Computation, 9(7):3084–3095, 2013.

[3] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. J Comp Chem, 32:2319–2327, 2011.

[4] R. J. Gowers, M. Linke, J. Barnoud, T. J. E. Reddy, M. N. Melo, S. L. Seyler, D. L. Dotson, J. Domanski, S. Buchoux, I. M. Kenney, and O. Beckstein. MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations. In S. Benthall and S. Rostrup, editors, Proceedings of the 15th Python in Science Conference, pages 98–105, Austin, TX, 2016. SciPy.

[5] G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken, and P. Zhang. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. Nature, 497(7451):643–6, May 2013.

[6] D. E. Shaw, R. O. Dror, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers, E. Chow, M. P. Eastwood, D. J. Ierardi, J. L. Klepeis, J. S. Kuskin, R. H. Larson, K. Lindorff-Larsen, P. Maragakis, M. A. Moraes, S. Piana, Y. Shan, and B. Towles. Millisecond-scale molecular dynamics simulations on anton. In SC '09: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, pages 1–11, New York, NY, USA, 2009. ACM.

[7] S. Chen, R. P. Wiewiora, F. Meng, N. Babault, A. Ma, W. Yu, K. Qian, H. Hu, H. Zou, J. Wang, S. Fan, G. Blum, F. Pittella-Silva, K. A. Beauchamp, W. Tempel, H. Jiang, K. Chen, R. Skene, Y. G. Zheng, P. J. Brown, J. Jin, C. Luo, J. D. Chodera, and M. Luo. The dynamic conformational landscapes of the protein methyltransferase SETD8. bioRxiv, 2018. DOI: 10.1101/438994

[8] T. Cheatham and D. Roe. The impact of heterogeneous computing on workflows for biomolecular simulation and analysis. Computing in Science Engineering, 17(2):30–39, 2015.

[9] T. Maximova, R. Moffatt, B. Ma, R. Nussinov, and A. Shehu. Principles and overview of sampling methods for modeling macromolecular structure and dynamics. PLoS Comput Biol, 12(4):1–70, 04 2016.

[10] M. C. Zwier and L. T. Chong. Reaching biological timescales with all-atom molecular dynamics simulations. Curr Opin Pharmacol, 10(6):745–52, Dec 2010.

[11] J. D. Chodera and F. Noé. Markov state models of biomolecular conformational dynamics. Current Opinion in Structural Biology, 25:135 – 144, 2014. Theory and simulation / Macromolecular machines.

[12] A. Pérez, G. Martínez-Rosell, and G. D. Fabritiis. Simulations meet machine learning in structural biology. Current Opinion in Structural Biology, 49:139 – 144, 2018.

[13] I. Foster, M. Ainsworth, B. Allen, J. Bessac, F. Cappello, J. Y. Choi, E. Constantinescu, P. E. Davis, S. Di, W. Di, H. Guo, S. Klasky, K. K. Van Dam, T. Kurc, Q. Liu, A. Malik, K. Mehta, K.

Mueller, T. Munson, G. Ostouchov, M. Parashar, T. Peterka, L. Pouchard, D. Tao, O. Tugluk, S. Wild, M. Wolf, J. M. Wozniak, W. Xu, and S. Yoo. Computing just what you need: Online data analysis and reduction at extreme scales. In F. F. Rivera, T. F. Pena, and J. C. Cabaleiro, editors, Euro-Par 2017: Parallel Processing, pages 3–19, Cham, 2017. Springer International Publishing.

[14] D. Bhowmik, M. T. Young, S. Gao, and A. Ramanathan. Deep clustering of protein folding simulations. bioRxiv, 2018. Doi: 10.1101/339879

[15] Peter M. Kasson and Shantenu Jha. Adaptive ensemble simulations of biomolecules. Current Opinion in Structural Biology 2018. 52:87-94. DOI:10.1016/j.sbi.2018.09.005

[16] Concurrent and Adaptive Extreme Scale Binding Free Energy Calculations. Jumana Dakka, Kristof Farkas-Pall, Matteo Turilli, David W Wright, Peter V Coveney, Shantenu Jha, published in IEEE eScience 2018 (arXiv 1801.01174)

[17] Adaptive Ensemble Biomolecular Simulations at Scale. Vivek Balasubramanian, Travis Jensen, Matteo Turilli, Peter Kasson, Michael Shirts, Shantenu Jha. 2018 (arXiv 1804.04736)

[19] V. Botu and R. Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. International Journal of Quantum Chemistry, 115(16):1074–1083, 2014.

[20] M. Spellings and S. C. Glotzer. Machine learning for crystal identification and discovery. AIChE Journal, 64(6):2198–2206, 2018.

[21] JCS Kadupitiya, Geoffrey C. Fox and Vikram Jadhao, “Machine Learning for Parameter Auto-tuning in Molecular Dynamics Simulations: Efficient Dynamics of Ions near Polarizable Nanoparticles”, paper in preparation

Glimpsing a Yottascale Data Ecosystem when the Fog Lifts

Micah Beck (mbeck@utk.edu), Terry Moore (tmoore@icl.utk.edu)

As the BDEC Pathways report[1] and other contemporary sources make clear, in the new era of data intensive science and engineering, nearly every field confronts formidable problems of “data logistics,” i.e., of the management of the time sensitive positioning and processing of data relative to both its intended users and the public or private resources available to them. Since there is widespread consensus that the volume of digital data worldwide is increasing exponentially, with plausible projections putting the total at around 20ZB by 2020, this situation is bound to get worse. Given the correlated, explosive growth in prolific data generators, ranging from the Internet of Things (IoT) and mobile devices, to scientific, engineering, medical, military and industrial equipment and infrastructure of all kinds, the reality of a Yottascale (10^{24}) digital universe is just over the horizon. Creating a widely shareable infrastructure that can provide the computing, storage/buffer, and communication services required to support data logistics for data ecosystems at that scale involves challenges that are obviously formidable, to say the least.

Beyond the notorious five V’s of data—volume, velocity, variety, variability, and value—what makes this problem especially “wicked” is the fact that, in a world of utterly pervasive IoT, mobile devices, and cyber-physical systems, *the data producers are everywhere* and this means, in turn, that the nodes of our general purpose Advanced Cyberinfrastructure Platform (ACP), our fabric for the “data periphery,” will have to be everywhere too. We know that ACP nodes (in a wide range of sizes and capabilities) will have to be deployed ubiquitously in the data periphery because at least some important applications will require low latency response, or local data reduction, or high confidence security/privacy, or highly survivable services, etc. So when we think about a future defining ACP, we are necessarily thinking about something with the same kind of boundary defying footprint as the Internet, with nodes deployed at and interoperable across every level of the ecosystem: wrist, pocket, purse, car, home, farm, office, building, campus, town, city, region, nation, globe. The question is “How can we create an ACP with that kind of deployment scalability?”

As we all know, the conventional answer to this question today is to use Virtual Machines and Containers to move miniaturized versions of the Cloud data center model into nodes located throughout the network: the Cloud is supposed to roll out across the data periphery as Fog computing. This model is characterized by persistent processes and file or database systems that allocate resources indefinitely on specific nodes. It further uses embedded state to implement high level services, even though creating sufficiently flexible service/state management functions for them (e.g., migration and fault tolerance) is difficult. But the model has been extremely successful inside commercial cloud data centers and content delivery networks, where problems with automated management have been countered by concentrating on application domains that generate a lot of income. It has been so successful in fact that the prevailing opinion in the ICT community is that no other model is viable for the great data periphery: technological path dependence and overwhelming commercial power dictate an inescapable destiny, regardless of the experience and principles of “Computer Science.”

The authors of this position paper have significant doubts about this dominant consensus. In view of the history of distributed systems over the past three decades, the fundamental problem we see is that any ACP that converges over the top of the conventional “three-silo” model—process+file system+Internet—will prove too complicated to permit scientific applications at scale (especially those that incorporate the data periphery) to be highly automated, that is, automated in the way the Internet has been. This is illustrated, for instance, by the fact that CDN’s are relatively expensive to run, so that the USGS cannot afford to pay the cost of making all its satellite data publicly available without throttling bandwidth. If we believe plans for the future of scientific cyberinfrastructure are fated to collide with the silos in the Fog or at the Edge, then perhaps some, even many, of the applications research communities will want to deploy have requirements that are simply ruled out.

We believe that designing an affordable and sustainable ACP for the extreme scale data ecosystems of science's future, an ACP that rules in as many applications as possible, is *a grand challenge problem* that calls for a radically different model from the current conceptions of Fog and Edge computing. We can understand why the problem of designing such an APC is so formidable by considering four plausible design constraints. A scalable and sustainable ACP must

- *Combine global interoperability with diverse, community specific policies*: Service definition interoperability is widely agreed to be fundamental to the success of a sustainable ACP. But as BDEC Pathways report reminds us, "As a practical matter, real interoperation is achieved by the definition and use of effective spanning layers,"[2] i.e. a common service interface that aggregates access to heterogeneous resources in support of a generalized set of applications that need to use such resources. If the ACP's global services layer (e.g., layer three in the Internet model, see Figure 1A) is also the global spanning layer (i.e.," the narrow waist of the hourglass"), then any use of the system must go through that interface, and therefore must be open to all system users. However, the current Internet shows that this approach leaves the system wide open to all manner of malicious ingenuity by bad actors, making acceptable—let alone community specific—security, privacy, and federation policies extremely difficult to achieve. If we want a shared, pervasive ACP that provides global interoperability, it is doubtful that building on the current Internet silo (see Figure 1B) will get the community to that goal.
- *Manage tradeoffs between node autonomy and manageability*: Logical centralization characteristic of current Fog models enables focused "command and control" of resources, but can suffer from bottlenecks, latencies and other data logistics issues due to the separation between the policy and processing planes. On the other hand, physical distribution enables local control, but makes global optimization (e.g., for performance and reliability) and resource allocation difficult. The design of current operating systems and networks conflate the placement of resources with their control. Can we design distributed systems in a way that provides appropriate performance and reliability in widely different environments by adapting the separation of data and control in diverse and flexible ways?
- *Create a topologically enabled interface to system resources for good data logistics*: Effective and efficient management proximity/locality/logistics in a shared ACP demands 1) awareness of where the data and the resources to control and process it are, and 2) some control over when, where and in what form it goes next. The Internet's spanning layer hides topology from higher layers for good reasons, but in doing so it restricts their ability to perform data logistics optimizations. Can we achieve necessary tradeoffs between abstracting away from topology and heterogeneity serving multiple client communities by moving the spanning layer below layer 3?
- *Future-proof the design to maximize its sustainability*: If a shared ACP is going to be sustainable, it will have to be able to absorb successive waves of innovation in the hardware substrate it runs on and in the applications that people want to run on it. Two factors combine to make this an extremely difficult problem. On one hand, network effects inevitably make the spanning layer of any universal bearer platform a point of design ossification, inhibiting future innovation at that layer [3]. On the other hand, any standard tends to restrict heterogeneity at the layer at which it is imposed. Taken together, these two facts imply that the lower the spanning layer is placed, the more sustainable the infrastructure will be, since a lower layer standard can allow many choices at the layers above it, whereas a high level standard locks in choices that it inherits from the limitations of the silo layers that it relies on. This poses a fundamental problem for designing a sustainable ACP with the common service interface at a very low layer: how can the resources of its intermediate node be modeled in a manner that is general enough to support all necessary services but still be simple, generic and limited portable to almost any hardware substrate?

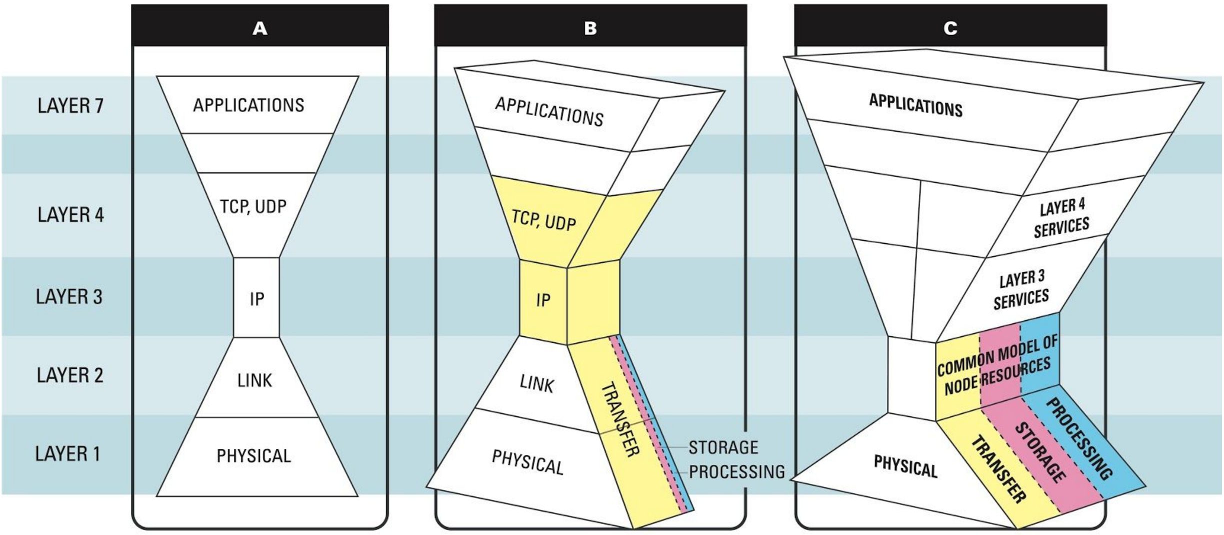


Figure 1: Moving the common interface from layer 3 of the Internet silo (B) to a lower layer (C) that exposes all its basic resources.

What we are proposing is to make the model of the intermediate node—the node operating system, if you will—the basis of interoperability (see Interoperable Convergence of Storage, Networking and Computation, FICC 2019, [4]). This means exposing a standards-compliant platform on which layer 3 services can be built, establishing interoperability below that layer, but not by adopting a uniform model of local networking. Instead the focus moves to the intermediate nodes themselves, virtualizing all of their local services including communication between "virtually adjacent" nodes. The fundamental abstraction on which these local services are built is the memory buffer or storage block, which is the common building block of operating system services.

If interoperability is moved to the resources of the intermediate node then the services built at "layer 3" need not be just those that are usually considered to be "communication" but can be much more general (see Figure 1C). For instance, a CDN can be a layer 3 service, but so can a distributed file system or a distributed facility that implements computing and caching of results in some particular domain of interest. The idea is to create a platform that can support distributed services in a fundamentally different way than the Internet approach (i.e., using stateless transmission to tie together servers located outside the network core) and not just a variant that loosens up the emphasis on "stateless" and "transmission".

The goal is to deploy nodes that support data transfer, persistence and transformation while being operated in a manner similar to the network, by engineers that monitor and adjust configuration but not by system administrators that get involved with the applications and their users imposing overhead and requiring fallback to complex manual procedures. This is the area where Data Logistics research has something unique to offer. Then there is a lot of engineering and further research to do in realizing the architectural vision.

[1] M. Asch *et al.*, “Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry,” *Int. J. High Perform. Comput. Appl.*, vol. 32, no. 4, pp. 435–479, 2018.

[2] D. D. Clark, “Interoperation, Open Interfaces, and Protocol Architecture,” in *The Unpredictable Certainty: Information Infrastructure through 2000*, Washington, DC, USA: National Academy Press, 1997, pp. 133–144.

[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[4] Micah Beck, Terry Moore, Piotr Luszczek, and Anthony Danalis, “Interoperable Convergence of Storage, Networking, and Computation,” presented at Future of Information and Communication Conference (FICC) 2019, to appear. (https://arxiv.org/abs/1706.07519)

Extreme Heterogeneity for S_n Transport Codes

Sunita Chandrasekaran, Assistant Professor, University of Delaware, schandra@udel.edu
(Collaborators: Oscar Hernandez, Wayne Joubert, Oak Ridge National Lab)

Architectures are rapidly evolving, and the future machines are expected to be equipped with large number of different types of devices offering billion-way concurrency. Such extreme heterogeneity demands that we rethink algorithms, languages, programming models among other components in order to increase parallelism from a programming standpoint in order to be able to migrate large scale applications to these massively powerful platforms. Although directive-based programming models is an effective solution as it allows programmers to worry less about programming and more about science, expressing complex parallel patterns in these models can be a daunting task especially when the goal is to match the performance that the hardware platforms are ready to offer. One of such complex parallel patterns, is the wavefront-based motif. This particular pattern has posed numerous challenges and been studied about at length since 1986 [1]. In scientific codes, this pattern has been observed in several structured and unstructured applications including S_n radiation transport codes, linear equation solvers, bioinformatics, nuclear reaction, etc.

Challenges

The primary challenge faced when attempting to develop a domain specific abstraction for wavefronts that can be implemented on the wide variety of hardware, as well as providing enough foresight into hardware trends to allow for adoption on future extreme heterogenous architectures with little to no modifications to the code. An important secondary challenge revolves around maintaining performance-portability while undergoing re-implementation of an abstraction, assuming the primary challenge is already met with a solution.

In the case of wavefront-based parallel patterns, this is extremely difficult. In addition to having a complex parallel pattern with regards to computation, different wavefront codes have non-uniform data dependencies. The commonality between these codes is that all data dependencies are met upstream. In a serial implementation, this implies that all data dependencies for a given iteration are satisfied by a previous iteration. There is no reliance on future computation. However, this becomes a huge problem when implementing a parallel wavefront code because we must structure our problem space in such a way that we expose parallelism, while ensuring that all data dependencies are met. This seems fairly trivial at first glance, but different wavefront codes have different amounts of data dependencies. They are all in the upstream direction, but we have no way of knowing what number of neighboring cells in the upstream direction a particular algorithm will examine. Some algorithms, such as ORNL’s Minisweep proxy code for Denovo radiation transport application, examine just a single neighboring cell in each upstream direction. Minisweep was recently parallelized and accelerated on state-of-the-art systems [5]. Others, like Smith-Waterman, trace back through an unknown number of neighboring cells until they find a particular value. This makes incorporating a data representation into a high-level abstraction very tricky.

Research Direction
Abstract machine model: Modern compute node hardware has an execution hierarchy. For example, a compute node may be composed of multiple GPUs, each with multiple cores possessing hardware threads and employing vector units composed of vector lanes. Some of these have co-located memories, for example node main memory, GPU high bandwidth memory or GPU shared memory associated with a streaming multiprocessor (SM) core. Execution threads are also associated with each level: for NVIDIA GPUs, in-warp threads execute in lock-step within a warp, in-threadblock threads are associated with an SM, and the thread grid is associated with the GPU. One can thus view a node as a hierarchy of execution units, memory places and compute threads, and in particular hardware threads can be thought of as indexed as a tuple depending on the location in the hierarchy. Threads also have characteristics based on location, e.g., thread synchronization across different cores of a node may be impossible or much slower compared to on-core synchronization. Likewise, memories at different levels have different speeds, and thread access to memories may have NUMA effects depending on the level. Note that these concepts readily apply to heterogeneous node as well as homogeneous systems.

State-of-the-art Research Wavefront parallelization approaches have been mapped to CPUs, GPUs, FPGAs Cell BEs. These approaches include blocking, loop permutation and skewing implemented within a polyhedral compiler transformation framework, adopting Lamport’s original parallel pipelined wavefront ‘hyperplane’ algorithm for certain applications [2-4]. Other approaches include HPCS languages, preliminary studies that use TBB, Cilk, CnC. In spite of these efforts, it is clear that most of these strategies either do not solve the wavefront parallel pattern itself but offer solutions to a specific similar problem type or require the user to use a low-level programming language to create loop transformations incurring a steep learning curve or provide a hardware-specific solution. It is hard and almost impossible for a scientific code developer to take these solutions for his/her own code and target modern platforms. Such gaps in the state-of- the-art work serve as a motivation for us to rethink what we need to support such computational motifs on extreme heterogeneous systems.

Maturity: The above state-of-the-art summary indicates that in spite of existing efforts, they cannot be leveraged by scientific developers for their applications. We propose to create an abstract parallelism model and create language extensions at the high-level to tackle this problem.

Timeliness: Architectures are becomingly increasingly parallel with fat and thin cores along with several tiers of memory hierarchy. Low-level programming model may achieve the best performance but expecting a domain scientist to be an expert low-level programmer to tap into the systems’ potential may not happen.

Uniqueness: There are several computational motifs that exposes wavefront-based pattern but currently there is no high-level solution to map such a pattern to extreme heterogeneity system. It is timely to create a high-level language extension with an abstract parallel model in mind to tackle this problem.

Novelty: The state-of-the-art paragraph captures existing methods that tackles this problem. However, those solutions cannot be adopted by scientific applications, as they are either too low-level or tied to a specific framework that is not being used by these applications. Moreover, it is not humanely possible to update the code every time the hardware changes. There is no work to the best of our knowledge that is exploring this approach. We aim to work with programming model standard communities and the vendors, explore a suitable solution that can be used to develop a solution applicable to several computational motifs exposing this parallel pattern.

REFERENCES

[1] Wolfe, M., 1986. Loops skewing: The wavefront method revisited. International Journal of Parallel Programming, 15(4), pp.279-293.
[2] DOE Accelerated Strategic Computing Initiative, The ASCI Sweep3D Benchmark Code, 1995
[3] Lamport, L., 1974. The parallel execution of DO loops. Communications of the ACM, 17(2), pp.83-93.
[4] Pennycook, S.J., Hammond, S.D., Mudalige, G.R., Wright, S.A. and Jarvis, S.A., 2011. On the acceleration of wavefront applications using distributed many-core architectures. The Computer Journal, 55(2), pp.138-153
[5] R. Searles, S. Chandrasekaran, W. Joubert, O. Hernandez, “MPI + OpenACC: Accelerating Radiation Transport Mini-Application, Minisweep, on Heterogeneous Systems,” in Computer Physics Communications, CPC 2018. DOI: 10.1016/j.cpc.2018.10.007

Development of a parallel algorithm for whole genome alignment for rapid delivery of personalized genomics

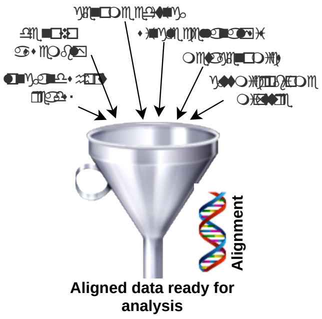
Sunita Chandrasekaran, Assistant Professor, University of Delaware, schandra@udel.edu

The Next Generation Sequencing (NGS) instruments are producing large volumes of data that is making the whole genome sequencing (WGS) a very important step for genomics research. Information gained from such volumes of data have been key to drug development and personalized medicine. Massive computing power offers tremendous capability to unwrap the complexity of biological systems and efficiently handle such massive genome (big) datasets. This challenge falls right into the intersection of computing systems and biology stimulating algorithmic innovation with sequence alignment on novel platforms.

With powerful sequencing data generated from instruments such as Illumina and Oxford Nanopore (long reads – a recent advancement) and a potentially transformative sequence alignment algorithm, we can make genomics a daily commodity. The state-of-the-art aligner is Burrows Wheeler Aligner (BWA) that is also tightly integrated into the gold-standard GATK best practices workflow, however BWA was not originally designed to take advantage of massively parallel processors. As a result, the algorithm is slow, memory inefficient, non-adaptable to hardware accelerators, non-portable across platforms and does now work well on long reads and only works well on short reads. An efficient aligner for long reads is very important as long reads are transforming our ability to assemble highly complex genomes, which the short reads cannot. BLASR is the state-of-the-art aligner for long reads but this algorithm has not been evaluated on massive computing systems, yet.

This demands the biology community along with the CS community to re-envision their goals and requirements in order to best exploit the tremendous opportunities the novel hardware platforms have to offer.

To address this challenge of bridging the gap between big data and HPC, our on-going research aims to create a novel parallel algorithm that can perform whole genome sequencing (WGS) faster while consuming less memory and not losing accuracy or sensitivity in comparison to classic sequence aligners. The most creative aspect of the proposed research is the seeding and the alignment algorithmic novelty that we carefully design after scanning a plethora of existing literature work and understanding the shortcomings of the same.



Our algorithm will be integrated into the widely popular and highly recommended industry standard Genome Analysis Toolkit (GATK) best practices workflow that is considered gold standard for variant calling (a process to identify variants from sequence data).

We plan to exploit the massive capability of hardware resources by creating a scalable algorithm that will not be suitable for just human genome but can also align sequences that are 3000 times larger than human genome, for example whiskfern. This way our algorithm will be propelling other areas of biology. Figure 1 indicates the benefits of such a faster sequence alignment.

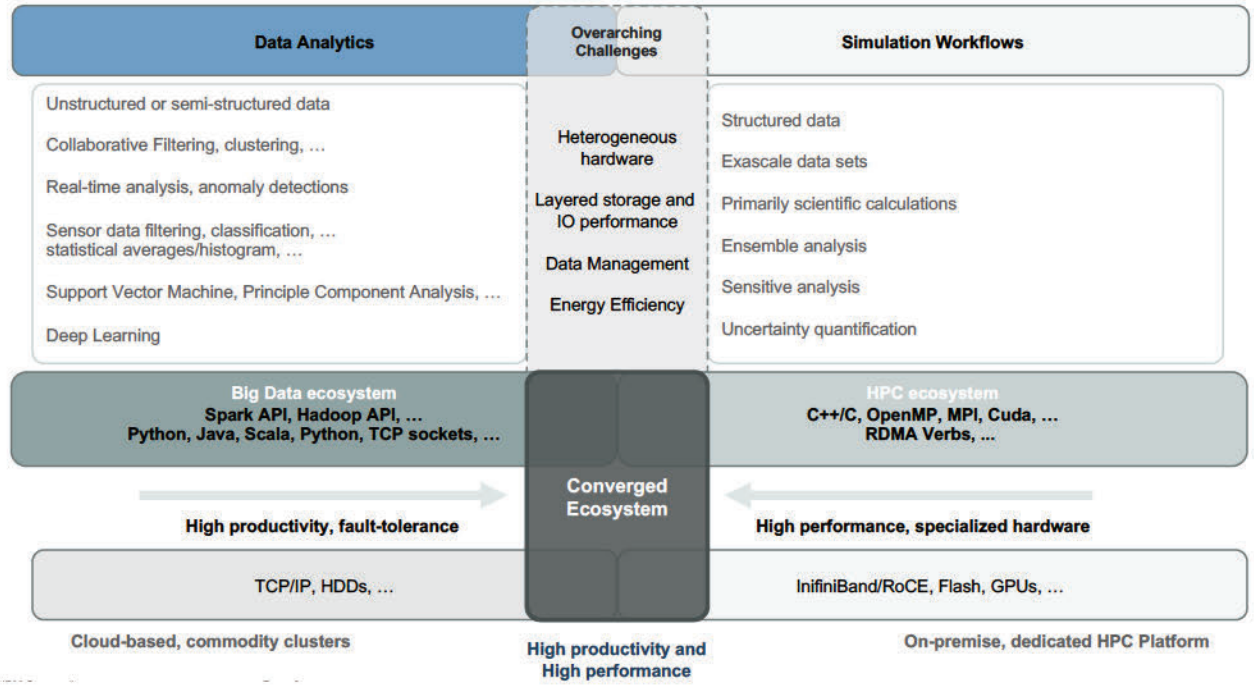
The results of this fast WGS alignment with hybrid DNA read length will enable faster translation of basic scientific findings into personalized therapeutic interventions for patients thus increasing survival rates. Our algorithm will be used for sequence alignment of pediatric cancer dataset from children of age 0-12 years from Nemours/Alfred I. duPont Hospital for Children. The algorithm will also be used on project that uses millions of veterans' health data in order to understand the complex genetic underpinnings that affect medical disorders, drug interactions, drug specificity, and individuals' responses to pharmaceuticals. Our novel algorithm will also be used for sequencing 2500 human genome taken from 26 distinct groups of people from the world. In addition to human genome research where rapid alignment can cause significant improvement in the quality of care provided to the patients, our scalable algorithm can also propel other areas of biology such as sequencing whiskfern that is 3000 times larger than human genome. We aim to look into the field of genome editing (CRISPR) as well where we will soon see a huge leap in sequencing demand. By making our algorithm and the software open-source we are enabling wider adoption and participation from the community.

Towards a Converged Software Ecosystem for Data Analytics and Extreme-Scale Computing

Carlos Costa, IBM T.J. Watson Research Center – chcost@us.ibm.com

The explosion of data generated by emerging extreme-scale simulation has been motivating a paradigm shift in scientific discovery: from simulation-centric to data-centric discovery. The unprecedented scale of data sets and the rise of powerful data analytics techniques made a data-driven approach crucial in advancing scientific inquiry in many disciplines in science and engineering.

In these emerging scientific workflows, which often embody the entire inference cycle of discovery, data analytics is a key element in enhancing traditional simulation. Rich and scalable data analytics support for in-situ analysis is expected to enable real-time extraction of actionable information and in-depth post-processing in large amounts of simulation data. A converged software ecosystem for data analytics and extreme-scale simulation would enable existing simulation workflows to be more intelligent, more productive, and more robust, and enable bigger or newer and more accurate science.


Data Analytics
Overarching Challenges
Simulation Workflows
Unstructured or semi-structured data
Collaborative Filtering, clustering, ...
Real-time analysis, anomaly detections
Sensor data filtering, classification, ... statistical averages/histogram, ...
Support Vector Machine, Principle Component Analysis, ...
Deep Learning
Heterogeneous hardware
Layered storage and IO performance
Data Management
Energy Efficiency
Structured data
Exascale data sets
Primarily scientific calculations
Ensemble analysis
Sensitive analysis
Uncertainty quantification
Big Data ecosystem
Spark API, Hadoop API, ...
Python, Java, Scala, Python, TCP sockets, ...
HPC ecosystem
C++/C, OpenMP, MPI, Cuda, ...
RDMA Verbs, ...
Converged Ecosystem
High productivity, fault-tolerance
High performance, specialized hardware
TCP/IP, HDDs, ...
InifiniBand/RoCE, Flash, GPUs, ...
Cloud-based, commodity clusters
High productivity and High performance
On-premise, dedicated HPC Platform

A converged ecosystem requires the effective co-deployment and integration of two traditionally disjoint software ecosystems: the Big Data and HPC ecosystems. While these ecosystems share some of the same overarching challenges imposed by trends in system design, distinct design goals and software development cultures make their integration challenging. The HPC community has been traditionally keen to specialization at both the software and hardware levels, often trading productivity for performance at the expense of increased overall system complexity. Conversely, the Big Data community evolved around the goal of enabling cost-effective parallel computing on cloud-based commodity clusters, for which high productivity and fault-tolerance are high priorities.

The challenges in reconciling these two approaches arise at multiple levels. At application and runtime levels, one challenge is how to leverage advantages in both ecosystems without disrupting APIs and breaking existing libraries and frameworks. For Big Data frameworks, that translates into how to leverage specialized hardware and enable effective scaling without disrupting APIs that many libraries depend on (e.g. Apache Spark's MLLib/GrapX, etc…). For traditional simulation code (e.g., MPI-based), the challenge translates into enabling more elastic deployment strategies with cloud technologies (e.g. containers and virtual machines) without sacrificing functionality and performance. At the data management level, one particularly difficult challenge is how to enable a data flow model that allows efficient data exchange across heterogenous processing frameworks (e.g., between an MPI application and a data analytics framework, such as Spark). Complex data-driven workflows typically expose both tightly and loosely coupled data sharing for which traditional low-level approaches, such as traditional message passing, are not efficient. At the resource management and scheduler level, a converged ecosystem would be required to support both batch and stream processing and allow the integration of on-premises and cloud platforms.

In response to these challenges, we envision a converged ecosystem that builds on the strengths of both ecosystems, adding extra services that facilitate interoperability. We have been working towards this vision in multiple fronts. In one front, we have been optimizing Apache Spark, adding support for specialized hardware, such as RDMA-enabled communication and GPU acceleration, and enhancing communication primitives for more effective scaling in large-systems [1][2][3]. These optimizations transparently enhance the performance of Spark libraries, making them more suitable for processing large data sets on extreme-scale platforms. In the data management front, we have been developing the Data Broker, a shared storage framework for data and message exchange. It provides a simple and intuitive API to access persistent or volatile storage through one or more distributed tuple-based namespaces, regardless of programming language and heterogeneity of the data [3][4]. It facilitates the integration of heterogenous workflows composed of simulation and data analytics. In the scheduler front, we have been identifying gaps and adding missing functionally to enable the deployment of simulation code with cloud schedulers, like Kubernetes [3].

While we believe these are still the first steps in the direction of a truly converged ecosystem, we are encouraged by the benefits we have been observing with the application of our efforts to a set of hybrid workflows. In collaboration with Lawrence Livermore National Laboratory (LLNL), we demonstrated the benefits of Spark optimizations for extreme-scale knowledge discovery on the Sierra System [3]. Also with LLNL, we demonstrated the value of the Data Broker for data exchange in a hybrid workflow involving molecular dynamics simulation and machine learning for lipid cell membrane simulation [3]. In collaboration with MIT-Harvard Broad Institute, we demonstrated significantly improved performance and scalability of a reference DNA variant discovery pipeline involving heterogenous steps [5]. These among other efforts have helped us to identify gaps and missing functionally and validate and refine the vision for a converged ecosystem.

References

[1] Leveraging Adaptive I/O to Optimize Collective Data Shuffling Patterns for Big Data Analytics. B Nicolae, CHA Costa, C Misale, K Katrinis, Y Park - IEEE Transactions on Parallel and Distributed Systems (TPDS), 2017. link

[2] Towards Memory-Optimized Data Shuffling Patterns for Big Data Analytics. B Nicolae, CHA Costa, C Misale, K Katrinis, Y Park - Cluster, Cloud and Grid Computing (CCGrid), 2016. link

[3] Converged Ecosystem for Data Analytics and Extreme-Scale Computing, Carlos Costa, Workshop I: Big Data Meets Large-Scale Computing, Part of NSF program: Science at Extreme Scales: Where Big Data Meets Large-Scale Computing, 2018. link

[4] Data Broker - https://github.com/IBM/data-broker

[5] Optimization of Genomics Analysis Pipeline for Scalable Performance in a Cloud Environment, CHA Costa, C Misale, F Liu, M Silva, H Franke, P Crumley, B D'Amora, IEEE International Conference on Bioinformatics and Biomedicine (industry track), 2018 (to appear).

[6] SparkGA: A Spark Framework for Cost Effective, Fast and Accurate DNA Analysis at Scale. H Mushtaq, F Liu, C Costa, G Liu, P Hofstee, Z Al-Ars - ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM-BCB), 2017. link

Digital Continuum in Materials Research

Digital transition applied to Material Nano-Characterization.

Keywords: Materials – Characterization – Data Management and workflow – DFT

Authors: Thierry Deutsch, Luigi Genovese (CEA – Grenoble, France)

1- Challenges in the field of materials simulations

Materials simulations, quantum-mechanical in particular, have become dominant and widely used tools for scientific discovery and technological advancement: since they are performed without any experimental input or parameter they can streamline, accelerate, support or replace actual physical experiments. The field of computer-assisted materials modelling, discovery, and engineering is extremely vital.

Enhancing the impact of quantum mechanical modelling on materials research will stretch our simulation capacity towards systems of increasing size and complexity and our predicting abilities towards chemical accuracy, thus requiring the deployment of ever more sophisticated (and expensive) modelling and theoretical methods. More and more frequently, the whole process relies on extensive, database-driven searches, where millions of calculations are deployed to probe unexplored vistas in materials' space, in turn accumulating **an ever-increasing treasure trove of curated, high-quality computational data**.

Community codes are often a jumble of mathematical libraries, equation solvers, and property calculators, resulting from unconcerted efforts of generations of students and postdocs, usually without specific training in IT, and aiming at quick results rather than at structured, easy-to-maintain and to-port software. As a consequence, porting those codes among hardware architectures has always required extensive recoding in the past. This cannot be sustained any longer in view of the considerable level of complexity reached by both community codes (several hundred thousands code lines each) and the forthcoming diverse, heterogeneous, and rapidly evolving hardware architectures.

The solution we have identified is to refactor the code base into multiple software layers, resulting from the assembly of weakly coupled components (modules and libraries), to be maintained and enhanced independently from each other, shared among different codes, and ported across different architectures. In the European project MAX, we will now leverage our past efforts **to design solutions that will work across entire classes of codes**, and deploy them on a larger and even more representative number of community codes that will be ready for production on pre-exascale machines and beyond by the completion of the present programme. This will allow us to achieve a substantial economy of scale and the entire scientific community to exploit solutions that can be adopted by community codes not yet represented in our consortium.

2- Challenges in the field of characterization techniques

In other way, progresses in characterization techniques have been constant over the last decades due to the continuous development of radiation sources (e.g. synchrotron), optical elements, nanometer positioning systems, 3D techniques etc... Development of techniques that are capable of **material or device characterization at the nanoscale** are often a basic requirement of many process or device developers. The techniques covered include electron microscopies (TEM and SEM) with associated spectroscopies (EELS, EDX) and imaging techniques (holography, tomography), ion spectroscopies (SIMS), XPS, NMR, Scanning Probe Microscopies, etc.

However this ever increasing performance in the sensitivity and accuracy of characterization tools has been often accompanied by an increasing amount of generated data. Now it is rather common to have 1 Terabytes per experiment which have to be processed and analyzed by means of models and simulations. The technicality of this processing is often far below that of the characterization one (home-made or generic software on personal computers).

It would then be very valuable to tailor, in the Materials characterization area, some new digital tools as Data Management, Data workflow management systems but also some Artificial Intelligence tools like machine learning ones to classify and organize data or to build models.

Data management and analysis are keys to fostering materials characterization, accelerate the use and the exploitation of these results, improving materials but also industrial production by a better quality control.

2- Towards a digital continuum between characterization and simulation

A paradigm shift for computational design and discovery is also ensuing, in which massive high-performance computing (HPC) and high-throughput computing (HTC) efforts are combined with high-performance data analytics (HPDA) to identify the most promising novel materials or those with improved or designed properties and performance. Such effort requires simulation codes able to deliver the predictive accuracy needed to sustain or streamline experiments; able to address the complexity of real-life conditions; able to exploit or drive the evolution of the current and forthcoming hardware platforms; and able to leverage the wealth of data that is generated or harvested in computations and characterization experiments alike. **The capabilities of codes need to be standardized in workflows that provide highly curated data on demand**, and that can be exploited easily using cloud technologies by the scientific and industrial community at large.

Moreover, these data workflows should be directly coupled with 3D characterization improving considerably the material analysis. Statistics of the structure of materials could be greatly improved by a systematic and automated analysis using a digital continuum between characterisation and simulation based on these complex data workflows.

CEA contact: Dr. Thierry DEUTSCH

thierry.deutsch@cea.fr, phone: +33 4 38 78 34 06

Convergence of AI, Big Data, Computing and IOT (ABCI)- Smart City as an Application Driver and Virtual Intelligence Management (VIM)

A White/Position Paper

Tarek El-Ghazawi

tarek@gwu.edu

The George Washington University

November 11, 2018

Background- This white paper advocates the need to establish rich case studies to evaluate and clearly formulate the research questions for the sought convergence. Further it will particularly cite Smart City as one promising such case study. It will also promote a couple of supporting concepts that can prove useful in the context of this convergence problem: namely the *productive system view (PSV)* and the *virtual intelligence management (VIM).*

The Need for Case Studies- The convergence of AI, HPC, Big Data, Cloud and IoT will require a close examination to define opportunities for *interoperability*, *co-design* and *productivity* to avoid mismatches, and exploit efficiencies while delivering the services needed seamlessly, promptly, cost-efficiently and without the users' worries or heroic efforts even at scale. In order to do so well from the beginning, a few critical case studies need to be identified, thoroughly researched, prototyped and experimented with in order to distill overarching guiding principles for the design of relevant infrastructures of hardware, software and governing standards as well as best practices for the the deployment of applications.

Smart City as a Case Study- Smart city is a rich environment for examining the convergence. An abundance of smart sensors/IoT devices could be found nearly anywhere. Services would be for the most part interactive requiring fast decision making thereby high-performance computing is a must. At one end, more heterogeneous HPC capabilities can be augmented at an Apache spark server side for more engaging processing, model based analyses, ML training/weight calculations and global decisions, while at the IoT device client side processing capabilities at the sensor side can be provisioned for real-time local decisions and apply ML style processing using the back-end calculated and updated weights. Other layers in the hierarchy may be present and should be efficiently leveraged. An examination of any impediments of achieving a rich hierarchy of seamless processing support with a variety of capabilities should be studied here.

The Need for a Productive System View: Programming and Execution Models- Developing applications for such a converged environment productively requires the creation of an integrated system view from the edge to the data center with an overarching programming model that allows domain scientists to harness the power of this environment easily. It also requires an execution model that leverages compiled applications and a run-time system to orchestrate activities belonging to one application across the layers and even an operating system-like service for inter-application coordination.

Virtual Intelligence Management (VIM): Smart city is one of the applications that require speed and interactive processing. The promise of virtual memory was to give the programmer/domain scientist the illusion that they have a very large and very fast memory at their disposal (on the application side) in a cost efficient manner through the wise management of memory resources based on the needs. Likewise, I envision here that processing (and even ML) and data resources should be managed in such converged environment wisely to provide the applications with powerful processing and data anywhere in a cost-efficient manner, thus supporting just-in-time, right-on-the-spot (or just-in-place) intelligence.

Testbeds, Benchmarking and Metrics: A number of testbeds encompassing the capabilities needed for a number of key application drivers must be selected. Application level benchmarks, as well as targeted and mirco-bencharmks and metrics need to be identified to facilitate the research studies in order to provide sufficient data for abstraction and co-design issues, for example. Testbed may be simulations, lab-based or instrumentation of existing real platforms.

Learning Systems for Deep Science

A white paper for the November 2018 Big Data and Extreme-scale Computing workshop

Ian Foster, Argonne National Laboratory and University of Chicago; foster@anl.gov

We are entering an era of what we may call *deep science*, in which machine learning (ML) and in particular deep learning (DL) methods are used increasingly to automate many elements of research workflows.

Motivations from molecular sciences: Materials science and chemistry illustrate many relevant issues. ML/DL methods are being used extensively [6], for example to extract knowledge from the scientific literature [15], process data from experiments [13], synthesize software to study specific problems [5], estimate properties of unfamiliar compounds [16], select the next compounds and materials to study [11], and design experiments and computations [14]. New approaches are being used to capture and organize large quantities of heterogeneous data [4] and associated models [8].

New challenges for computational infrastructure: New methods such as those just reviewed present major challenges for the computational technologies, methods, and infrastructure on which science has long relied. The following are just a few of the issues. High-end computing is no longer restricted to a few niche researchers and their esoteric applications, and big data processing is no longer the exclusive domain of big data specialists. Scientists, engineers, and technicians need massive computing to train ML/DL models and (in the aggregate) to serve such models. They need access to large quantities of training data, which must be collected at many locations and integrated and organized for effective use. They require access to specialized software for defining, training, applying, and interpreting models. The software lifecycle changes also, as for example when the "applications" used by scientists are models created by automated processes, deployed in various forms on different platforms (e.g., at edge devices in field experiments and laboratories [2]), and updated dynamically in response to new data. These new scenarios pose challenges for provenance and reproducibility.

The need for learning systems: Addressing these new demands will require significant evolution of scientific infrastructure at every level, from processors (e.g., new DL-optimized chips), computers, data systems, systems software, libraries [10], and networks to the design of scientific facilities (e.g., to deliver data and to support automated operations). In some cases (e.g., processor design), innovations will come primarily from industry; in others, science will need to innovate to address unique requirements. The end result will likely be new *learning systems* that integrate large-scale computing, storage, and networks into research environments in ways designed to satisfy voracious new demands for both large-scale and timely data and computation; deliver new methods to new communities via new services and APIs, for example for on-demand inference; support the resulting new workloads, for example via the use of serverless computing [12]; distribute and connect data and computation in new ways; and track and explicate increasingly complex computational results.

The vital role of cloud services. The value of cloud services as a convenient source of elastic computing and storage is well known. Less appreciated, but equally important, is their ability to host powerful automation services that can manage the complex workflows that underpin modern data-driven science [9]. The Globus service [7] illustrates the latter use: its cloud-hosted services are used by tens of thousands to manage a wide range of processes relating to authentication and authorization, data transfer and synchronization, and data publication, and are now being extended to manage data lifecycle issues such as data publication [1] and processing of data from experimental facilities [3].

References

[1] R. Ananthakrishnan, B. Blaiszik, K. Chard, R. Chard, B. McCollam, J. Pruyne, S. Rosen, S. Tuecke, and I. Foster. Globus platform services for data publication. In *Practice and Experience on Advanced Research Computing*, page 14. ACM, 2018.

[2] P. Beckman, R. Sankaran, C. Catlett, N. Ferrier, R. Jacob, and M. Papka. Waggle: An open sensor platform for edge computing. In *IEEE SENSORS*, pages 1–3. IEEE, 2016.

[3] B. Blaiszik, K. Chard, R. Chard, I. Foster, and L. Ward. Data automation at light sources. In *13th International Conference on Synchrotron Radiation Instrumentation*. 2018.

[4] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, and I. Foster. The Materials Data Facility: Data services to advance materials science research. *Journal of Materials*, 68(8):2045–2052, 2016.

[5] V. Botu, R. Batra, J. Chapman, and R. Ramprasad. Machine learning force fields: Construction, validation, and outlook. *arXiv.org*, Oct. 2016.

[6] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, July 2018.

[7] K. Chard, S. Tuecke, and I. Foster. Globus: Recent enhancements and future plans. In *XSEDE16 Conference on Diversity, Big Data, and Science at Scale*, 2016.

[8] R. Chard, Z. Li, K. Chard, L. Ward, Y. Babuj, A. Woodard, S. Tuecke, B. Blaiszik, M. J. Franklin, and I. Foster. DLHub: Model and data serving for science. 2018.

[9] I. Foster and D. B. Gannon. *Cloud Computing for Science and Engineering*. MIT Press, 2017.

[10] A. Haidar, A. Abdelfattah, M. Zounon, P. Wu, S. Pranesh, S. Tomov, and J. Dongarra. The design of fast and energy-efficient linear solvers: On the potential of half-precision arithmetic and iterative refinement techniques. In *International Conference on Computational Science*, pages 586–600, 2018.

[11] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv.org*, Feb. 2018.

[12] H. Lee, K. Satyam, and G. Fox. Evaluation of production serverless computing environments. In *11th International Conference on Cloud Computing*, pages 442–450. IEEE, 2018.

[13] D. Pelt, K. Batenburg, and J. Sethian. Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks. *Journal of Imaging*, 4(11):128–20, Nov. 2018.

[14] F. Ren, L. Ward, T. Williams, K. J. Laws, C. Wolverton, J. Hattrick-Simpers, and A. Mehta. Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Science Advances*, 4(4):eaaq1566, 2018.

[15] M. C. Swain and J. M. Cole. ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016.

[16] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2:16028, 2016.

Computational and Data Intelligence linked in the Intelligent Aether for Applications

Geoffrey Fox November 27, 2018

We are in the midst of yet another amazing computing technology-driven revolution that is seen in the advances in machine and deep learning, cloud computing, internet of things (edge computing), data and computational science. These advances are happening in spite of the slowdown in Moore's law which can be combatted by custom designs (neuromorphic or FPGA's for example), major hardware advances (e.g. quantum computing) or in the near term, by the systematic use of high-performance computing HPC technology. More importantly, we expect that the pervasive use of training and learning, the intelligent aether, can potentially lead to huge performance advances and counter the slowdown of Moore's law. All these approaches need to preserve the ease of use of current big data systems.

As well as the advances identified above, another important trend is the broadening of the importance of computing across many different application fields which is occurring within research, commercial, and government sectors. This seen academically by the broad interest in interdisciplinary programs such as "Computer Science + X", "AI Driven X", computational thinking, and of course computational and data science. Increasingly programming is moving to a higher level as users have attractive front ends (Python Notebooks or Gateways) to build mashups (or workflows) of existing libraries. There are emerging backend technologies such as Function as a Service that support this trend. The Software 2.0 concept that one programs datasets, not machine instructions, is also relevant.

We can term the systems challenge as architecting the Global AI and Modeling Supercomputer GAMSC where Global captures the need to mashup services from many different sources; AI captures the incredible progress in machine learning (ML); Modeling captures both traditional large-scale simulations and the models and digital twins needed for data interpretation; Supercomputer captures that everything is huge and needs to be done quickly and often in real time for streaming applications. The GAMSC includes an intelligent HPC cloud linked via an intelligent HPC Fog to an intelligent HPC edge. We consider this distributed environment as a set of computational and data-intensive nuggets swimming in an intelligent aether. GAMSC requires parallel computing to achieve high performance on large ML and simulation nuggets and distributed system technology to build the aether and support the distributed but connected nuggets. In the latter respect, the intelligent aether mimics a grid but it is a data grid where there are computations but typically those associated with data (often from edge devices). So unlike the distributed simulation supercomputer that was often studied in previous grids, GAMSC is a supercomputer aimed at very different data intensive AI-enriched problems.

I believe that we need to re-use as much as possible of the powerful commercial big data technology which is captured by the HPC-ABDS -- High-Performance Computing Enhanced

Apache Big Data Stack -- concept. This has been developed in a large collaborative NSF SPIDAL (Scalable Parallel Interoperable Data Analytics Library) team science collaboration project with 7 institutions led by Indiana University. This project used applications (Network Science, Polar Science, Molecular Dynamics, Pathology and Health-related Imaging) to drive HPC enhanced core technologies. Applications to scientific, medical and engineering research are attractive and these plus applications aimed at the good of society are places where academia can make especially important contributions.

There is a rapid increase in the integration of ML and simulations. ML can analyze results, guide the execution and set up initial configurations (auto-tuning). This is equally true for AI itself -- the GAMSC will use itself to optimize its execution for both analytics and simulations. This is an important idea that will grow. In principle every transfer of control (job or function invocation, a link from device to the fog/cloud) should pass through an AI wrapper that learns from each call and can decide both if call needs to be executed (maybe we have learned the answer already and need not compute it) and how to optimize the call if it really needs to be executed. The digital continuum proposed by BDEC2 is an intelligent aether learning from and informing the interconnected computational actions that are embedded in the aether. Implementing the intelligent aether embracing and extending the edge, fog, and cloud is a major research challenge where bold new ideas are needed!

The new MIDAS middleware designed in SPIDAL has been engineered to support high-performance technologies and yet preserve the key features of the Apache Big Data Software. MIDAS seems well suited to build the prototype intelligent high-performance aether. Note this will mix many relatively small nuggets with AI wrappers generating parallelism from the number of nuggets and not internally to the nugget and its wrapper. However, there will be also large global jobs requiring internal parallelism for individual large-scale machine learning or simulation tasks. Thus parallel computing and distributed systems (grids) must be linked in a deep fashion although the key parallel computing ideas needed for ML are closely related to those already developed for simulations and our expertise is immediately applicable. This technology development needs to continue being driven by applications with health, social science, and scientific research being very attractive.

Pathways to Convergence – An Additional Scenario

Dennis Gannon

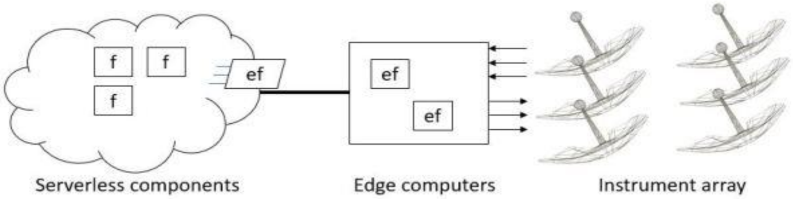
The frontier of computing is now, literally, at the edge of the network. The edge has long been the home of content distribution systems where content can be cached and access quickly, but now that the Internet-of-Things (IoT) is upon us, it is also increasingly important to do some of the computing at the edge. For example, if you have a thousand tiny sensors in a sensitive environment or farm and you need to control water from sprinklers or detect severe weather conditions, it is necessary to gather the data, do some analysis and signal an action. If the sensors are all sending WIFI messages they may be routable to the cloud for analysis, but a more common solution is to provide local computing that can do some event preprocessing and response while forwarding only summary data to the cloud or HPC system. That local computing is called the Edge, or if distributed as a system of edge servers, it is often called the Fog. These points are well documented in the report "Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry" by Asch et. al. (Hereafter referred to as the Pathways Report.) In this note we discuss an additional paradigm of computing not covered in that report: hardware and software microservices.

A microservice based application is based on decomposing the problem into several layers of independent software components that can be deployed and scaled independently. Suppose you are designing an application that must respond in near real-time to thousands of independent concurrent inputs. One layer of microservices can catch and clean the incoming data, which can then be forwarded to a second layer of analysis services that classify the input. A third layer can handle data logging, and another can deal with responses. The number of instances in each layer can be adjusted to match the input bandwidth requirements and scaled back when not needed. All the large commercial cloud providers base many, if not all, of their on-line services on this design. Examples include Netscape, Amazon.com, Twitter, Ebay, Uber, Google search, Bing and Azure's planet-scale CosmosDB. Applications running with thousands of instances are common. In addition to dynamic scaling, another advantage of this design paradigm is that services can be upgraded and replaced without taking the system down. (This is critical to DevOps design principles.)

Microservices are often packaged and deployed as containers running in massive-scale service fabrics like Kubernetes (open sourced by Google and now a standard). A related technology is Serverless computing in which the container instance is managed by a system that invokes the service as a function in response to an input signal. (Hardware microservices are small specialized devices that are deployed with the servers to execute microservices. A great example of hardware microservices is the FPGA nodes in the Azure Project Brainwave.)

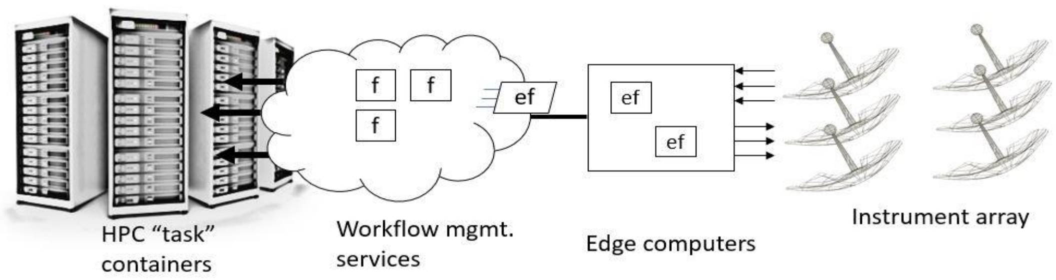
Migrating Microservices to the Edge.

If serverless functions are designed to respond to signals, that suggests that it should be possible to extend them to run in the edge servers rather than the cloud. AWS was the first to do this with a tool called GreenGrass that provides a special runtime system that allows us to push/migrate lambda functions or microservices from the data center to the edge. More recently Microsoft has introduced (and now open-sourced) Azure IoT Edge which is built on open container technologies. Using an instance of the open source Virtual Kubelet deployed on the edge devices, we can push our Kubernetes containers to run on the edge. You can think of a Kubelet as the part of Kubernetes that runs on a single node. This enables Kubernetes clusters to span across the cloud and edge as illustrated below.



Dynamically deploy edge functions from the cloud to edge or fog nodes as needed to optimize performance.

A major difference between this cloud-to-edge Kubernetes model and traditional HPC is that the former is optimized for long running systems while the latter is based on batch execution of tasks. IoT related activities require uninterrupted service. An ideal hybrid of cloud-edge and HPC would be allowing part of the HPC system to be partitioned off so that a system like Kubernetes can manage long-running containerized services (and interactive environments). Long-running workflow management services can adapt in real-time to events from the edge and submit big analysis singularity containers to the HPC side for execution.



Advanced Cyberinfrastructure Platform Design

Toshihiro Hanawa

Information Technology Center, The University of Tokyo, Japan

On an advanced cyberinfrastructure platform (ACP), data logistics should logically be partitioned into each project for security reasons. For example, medical image recognition for cancer detection and genomic analysis have to manage persistent personal characteristics, and such kind of data must be protected perfectly. In contrast in traditional HPC systems, many shared resources are used. The access control including file permission mechanism is used but there are strong demands for higher security.

To fulfill such demands, the platform should employ double protection like file access control & logical separation, and the policy of double protection is helpful for accountability to stakeholders.

To realize logical separation from the other project on ACP, network level and storage level separation are considerable. In terms of network level separation, Virtual LAN (VLAN) can manage multiple independent segments. pKey in InfiniBand is very similar technology to VLAN. To expand VLAN technology to resource management, the technologies for Software Defined Network (SDN) can be used. In a certain project, the sensors, computing resources, and storages must be enclosed into a single VLAN. In terms of storage level separation, NVMe over Fabrics (NVMeoF) is promising approach to assign the block device mapping to the host dynamically. NVMeoF can achieve high performance IO using enhancement of NVMe protocol for SSD drive.

Data acquisition through Internet and heavy computations like simulation, machine learning, and so on can be executed on different subpart of ACP. Data acquisition processes should be performed on the front-end nodes of ACP (or Virtual Machines). On the front-end node, roughly filtered data for integrity should be stored into the storage directly. After that, data will be preprocessed by various kinds of accelerators, for example, inference of machine learning, and dataset is generated. Finally, heavy computation process using huge generated dataset is executed by large-scale computing resource like traditional HPC systems.

The Challenges and opportunities of BDEC systems for Smart Cities

Masaaki Kondo
Graduate School of Information Science and Technology, The University of Tokyo
RIKEN Center for Computational Science

Introduction: It is expected that future BDEC systems will become a key component of a cyber-infrastructure for smart city applications such as intelligent transportation, disaster prevention and mitigation, optimization of an energy supply chain, and smart industry. In these applications, various information obtained in the physical world is processed / optimized in a cyber-world, and then feedback is provided to the physical world in realtime. For example, AI-based anomaly detection systems installed in production lines of smart factories gather the various types of sensor data equipped in target machines, execute inference to predict anomality in time series of data, and control the machines or provide reports to operators accordingly. While these systems need to perform inference at edge devices for lower latency and realtimeness, analyzing whole data corrected from sensors in centralized cloud servers is important to create an efficient inference / optimization engine. Therefore, tight collaboration and autonomous cooperation between edge devices and BDEC system are indispensable.

Challenges for BDEC systems: In order to realize effective and advanced smart city applications with BDEC systems, we need to address several challenges including the following aspects:

- Interoperability: Edge devices are diverse and usually have low performance processing units and small amount of memory capacity, they can perform only a part of the AI and optimization tasks. Interoperability between edge devices and BDEC systems needs to be provided for smart city applications so that some portions of a task can be flexibly assigned to an edge device or a BDEC system depending upon the characteristics of the devices and applications. We need to consider at least tools and an application programming interface for the interoperability.
- Efficient communication protocols: Communications between an edge devices and serves in a datacenter is a key for smart sensor or IoT applications. In these systems, simple communication protocols such as HTTP or MQTT are frequently used but they are not common in HPC systems. It is not clear BDEC systems can offer an efficient data communication performance with these protocols.
- Efficient data handling: While HPC systems are optimized for large data handling or burst data transfer, smart sensor applications usually generate and handle a small chunk of data at a time. When such a small data packet comes to an HPC system asynchronously and periodically, the overall system performance may degrade seriously. Future BDEC systems should have a special mechanism to handle a large amount of small data items.

Though we envision an improve human experience by smart city applications with BDEC systems, community wide discussions to explore the possible solutions for the above mentioned challenges are indispensable.

OneDataShare:
A Universal Data Sharing Building Block for Data-Intensive Applications

PI: Tevfik Kosar, University at Buffalo, SUNY

As data has become more abundant and data resources become more heterogeneous, accessing, sharing and disseminating these data sets become a bigger challenge. Using simple tools to remotely logon to computers and manually transfer data sets between sites is no longer feasible. Managed file transfer (MFT) services have allowed users to do more, but these services still rely on the users providing specific details to control this process, and they suffer from shortcomings including low transfer throughput, inflexibility, restricted protocol support, and poor scalability. OneDataShare is a universal data sharing building block for data-intensive applications, with three major goals: (1) optimization of end-to-end data transfers and reduction of the time to delivery of the data; (2) interoperation across heterogeneous data resources and on-the-fly inter-protocol translation; and (3) prediction of the data delivery time to decrease the uncertainty in real-time decision-making processes. These capabilities are being developed as a cloud-hosted service.

Goal 1: Reduce the time to delivery of the data. Large scale data easily generated in a few days may take weeks to transfer to the next stage of processing or to the long-term storage sites, even assuming high speed interconnect and the availability of resources to store the data. Through OneDataShare's application-level tuning and optimization of TCP-based data transfer protocols (e.g., GridFTP, SFTP, SCP, HTTP etc), the users will be able to obtain throughput close to the theoretical speeds promised by the high-bandwidth networks, and the performance of data movement will not be a major bottleneck for data-intensive applications any more. The time to the delivery of data will be greatly reduced, and the end-to-end performance of data-intensive applications relying on remote data will increase drastically.

Goal 2: Provide interoperation across heterogeneous data resources. In order to meet the specific needs of the users (i.e., scientists, engineers, educators etc), numerous data storage systems with specialized transfer protocols have been designed, with new ones emerging all the time. Despite the familiar file system-like architecture that underlies most of these systems, the protocols used to exchange data with them are mutually incompatible and require specialized software to use. The difficulties in accessing heterogeneous data storage servers and incompatible data transfer protocols discourage researchers from drawing from more than a handful of resources in their research, and also prevent them from easily disseminating the data sets they produce. OneDataShare will provide interoperation across heterogeneous data resources (both streaming and at-rest) and on-the-fly translation between different data transfer protocols. Sharing data between traditionally non-compatible data sources will become very easy and convenient for the scientists and other end users.

Goal 3: Decrease the uncertainty in real-time decision-making processes. The timely completion of some compute and analysis tasks may be crucial for especially mission-critical and real-time decision-making processes. If these compute and analysis tasks depend on the delivery of certain data before they can be processed and completed, then not only the timely delivery of the data but also the predictive ability for estimating the time of delivery becomes very important. This would allow the researchers/users to do better planning, and deal with the uncertainties associated with the delivery of data in real-time decision-making process. OneDataShare's data throughput and delivery time prediction service will eliminate possible long delays in completion of a transfer operation and increase utilization of end-system and network resources by giving an opportunity to provision these resources in advance with great accuracy. Also, this will enable the data schedulers to make better and more precise scheduling decisions by focusing on a specific time frame with a number of requests to be organized and scheduled for the best end-to-end performance.

In order to realize the above-mentioned goals, OneDataShare project produces the following tangible outputs: (1) implementation of novel and proven techniques (online optimization based on real-time probing, offline optimization based on historical data analysis, and combined optimization based on historical analysis and real-time tuning) for application-level tuning and optimization of the data transfer protocol parameters to achieve best possible end-to-end data transfer throughput; (2) development of a universal interface specification for heterogeneous data storage endpoints and a framework for on-the-fly data transfer protocol translation to provide interoperability between otherwise incompatible storage resources; (3) instrumentation of end-to-end data transfer time prediction capability, and feeding of it into real-time scheduling and decision making process for advanced provisioning, high-level planning, and co-scheduling of resources; (4) deployment of these capabilities as part of stand-alone OneDataShare cloud-hosted service to the end users with multiple flexible interfaces; and (5) integration of these capabilities with widely used data transfer (e.g., Globus) and workflow management (e.g., Swift, Pegasus) tools, and validation of them in real-life data-intensive applications.

Challenges Faced in the Development of OneDataShare

Challenge 1: Transferring large datasets especially with heterogeneous file sizes and dynamically changing background traffic causes inefficient utilization of the available network bandwidth. Small file transfers may cause the underlying transfer protocol not reaching the full network utilization due to short-duration transfers and connection start up/tear down overhead; and large file transfers may suffer from protocol inefficiency and end-system limitations. Application-level TCP tuning parameters such as pipelining, parallelism and concurrency are very effective in removing these bottlenecks, especially when used together and in correct combinations. However, predicting the best combination of these parameters requires highly complicated modeling since incorrect combinations can either lead to overloading of the network, inefficient utilization of the resources, or unacceptable prediction overheads.

Solution: In order to address this issue, we combined offline historical log analysis with online dynamic tuning. Our combined optimization technique initially uses historical data to derive network specific models of transfer throughput based on protocol parameters. Then by running sample transfers, it captures current load on the network which is fed into these models to increase the accuracy of our predictive modeling. Combining historical data analysis with real time sampling enables our algorithms to tune the application level data transfer parameters accurately and efficiently to achieve close-to-optimal end-to-end data transfer throughput with very low sampling overhead.

Challenge 2: There is a highly fragmented ecosystem of data transfer tools, with many different tools (all with widely differing interfaces) having been crafted for speaking the many protocols in widespread use today. This fragmentation is further amplified by research databases and storage services requiring special software to access, despite employing a common transfer protocol. Providing flexible and maintainable interoperability between these plethora of systems, tools, and protocols has been a challenging task.

Solution: To address this issue, we have devised an interface specification for heterogeneous data storage endpoints and a framework for on-the-fly data transfer protocol translation for interoperability across data resources. We call this framework Feather (Framework for Enacting Asynchronous Transactions on Heterogeneous Endpoint Resources). Modules built with Feather present a unified client interface for interacting with specialized data storage systems. A collection of such modules constitutes a protocol abstraction layer allowing Feather to act as a translation mediator between any combination of supported systems. Resource objects in Feather are stateless, and the instantiation of a resource object in Feather does not correspond to the creation of any resource on a storage endpoint.

Ongoing Challenge: Dependency on third party software and libraries is a big issue for the sustainability of OneDataShare software development. Upgrades and changes in third party software and libraries are sometimes not backward compatible (e.g., Dropbox API upgrade from v1 to v2) or they make the existing libraries completely obsolete (e.g., Globus ending the support for open-source Globus toolkit) which increases the amount of time and effort to sustain support for certain transfer protocols/tools. We have adapted a completely modular approach to decrease the impact of this issue on our integrated OneDataShare software development, but continue to look for other solutions.

Geospatial and Global Earth Mapping and Modelling Applications Requirements for the BDEC2 Workshop

William Kramer[1], University of Illinois at Urbana Champaign
with contributions from Paul Morin, Jonathan Pundsack, Claire Porter and others at the Polar Geospatial Center[2] at the University of Minnesota, and Brett Bode and Greg Bauer, NCSA[3] at University of Illinois at Urbana Champaign
November 28-30, 2018

This white paper is an initial and limited discussion of requirements and goals for geospatial information systems and applications. GIS is broad umbrella for a range of research uses from mapping and imaging the earth – not just the surface – to providing assessments of current and past conditions augmented with strong modeling capabilities, including the integration of physical models with social and environmental models. Geospatial information already integrates expensive sensors (e.g. high resolution satellites) with many, lower cost sensors (field level, UAVs, radar, lidar, sonar, etc.) at all scales, which all produce huge amounts of raw data and then uses various computational and analysis methods (image analysis, computer vision, model creation, ML/AI, modeling and simulation, …) to derive insights.

The scope of geospatial activities is too large and varied to incorporate into a single, short white paper, so the remaining paper will focus on the needs, requirements and impacts of just one fundamental enabling aspect for GIS - the new method of producing high resolution Digital Elevation Models (DEMs) from satellite images. In the last few years, new best of breed approaches created paradigm shifting map creation capabilities that greatly reduce the cost and improves the timeliness and resolution of traditional map making. The improvements include increasing the resolution of current elevation maps by more than 3 orders of magnitude (12.5^2 improvements in resolution), improving the time to production 58,500x in time to solution compared to a single workstation, and a 220 times reduction in cost, resulting in 9 orders of magnitude of overall productivity improvement. The proof of the paradigm shift has been shown in the ArcticDEM[4,5] and Reference Elevation Model of Antarctica[6] (REMA) projects that use the NCSA's Blue Waters NSF leadership computing system to generate essentially complete elevation maps using Illinois Innovation and NSF PRAC allocations. The improvements mean it is for the first time feasible to envision creating very accurate Digital Elevation Models on a global and frequent basis.

To give some idea of fundamental, albeit limited, requirements for computing and data required for creating two meter accurate DEMs of the entire landmass of the earth. The new five year EarthDEM project extends the impact and scope of these methods to the entire land mass of the earth.

Global DEM processing requirements

The goal of EarthDEM is use a global coverage set of in-track and cross-track satellite images to create two meter digital elevation models of the entire landmass of the earth and provide those to the public in mosaics and well as localized data sets.

[1] All error and mistakes in this document are the sole responsibility of the author and not the contributors

[2] https://www.pgc.umn.edu/data/arcticdem/

[3] http://www.ncsa.illinois.edu

[4]http://www.ncsa.illinois.edu/news/story/blue_waters_processes_final_installment_of_arcticdem_mapping_initiative

[5] https://bluewaters.ncsa.illinois.edu/liferay-content/document-library/18symposium-slides/porter.pdf

[6] http://www.ncsa.illinois.edu/news/story/ncsas_blue_waters_supercomputer_helps_map_the_poles

	Strips	Area – km^2	Estimated BW node-hours required to process one time	Data Required (PB)
In-track	336,492	538 million	54 million	5.38
Cross-Track	2,956,131	2,365 million	473 million	47.30
Total	3,292,623	2,903 million	527 million	52.68

Based on Polar Geospatial Center (PCG) staff estimates, the computational needs to create a single, global set of DEMs once is 527 million Blue Waters X86 node hours[7] that are needed to process all the EarthDEM Digital Globe/NGA imagery that could be provided. The Arctic and Antarctic areas represent about 1/6 of the earth's land mass surface area each. Given the need to do reprocessing for areas (e.g. if clouds are present in the first set of images) and other factors, the total processing of all the data is approximately 3.2-3.5 dedicated, sustained "Blue Waters Years" of computing.

For data requirements, each stripe averages 4 GB in size, and two strips are needed for every DEM. Two meter DEMs average 8 GBs. So, the processes consumes 8 GB and produces 8 GB per sample. This indicates in addition to over 527 billion node hours of computational time, the processing requires 26-30 PB is consumed and 26-30PB is produced. Since the original strip data flows from repositories specific for the satellites, and is stored in open access repositories, these 50+ PBs of data has to move within the period of the campaign. If you assume this is a yearly campaign, the average sustained data rates are ~10 kbps, but will have peaks where multiple streams of 8 GBs need to move before or after a job initiated.

Benefits and Impacts of DEM creation

Having accurate DEMs is a necessary condition for many other research and practical areas, all of which rely on additional processing and analysis to gain their final insights. For example, for hydrology, DEMs are used to develop very large graphs that can be used to analyze water flows and storage, flooding and drought potential, water use and watershed contamination. For weather and climate, accurate DEMs can be used for analyzing micro climate and micro weather predictions and well as enhancing the predictability of severe weather based on the land surface atmosphere interaction. High resolution DEMs are used for understanding earth surface changes, such as impacts of mud slides and earthquakes, analysis and potentially predication of volcano eruption, etc. DEMs can be used for natural resource management such as forest and agricultural management and wild life migration patterns. DEMs also can be used for analyzing ice melts, glacier retreats and advances, sea level rise, etc. all on a larger, much more timely manner and lower cost scale than previously possible. Urban analysis and planning also now can use high resolution DEMs since at, 2 meters, (and even 30 cm when high accurate reference elevations are present) clearly show buildings, roads, infrastructure, etc.

Computational and data analysis resources need to support all these derived uses and more areas that are enabled by the use of high resolution DEMs. Just as DEMs are now much more accurate, the derived investigations will want to do more accurate and detailed analysis, which will require significant increases in computational and data resources. Furthermore, to facilitate use of DEMs and other data products by others, effective infrastructure is needed to enable easier

[7] A serious attempt was made to port SETSM to the GPU based XK nodes in the BW PAID program, but the resulting code ran slower than the XE version. Further work would be needed to see if SETSM could benefit from GPU acceleration.

storage, annotation and retrieval of the original images, the DEMs, metadata and derived data products.

Frequency of DEM releases

ArcticDEM has made two major, complete releases of their mosaic DEM, the first after about 18 months of processing on Blue Waters and a second after about three years of processing. Antarctic REMA has made one data release at 8m after about 18-24 months of processing. The geospatial community who use DEMs desire complete global DEMs every couple of years, and more frequent DEMs in areas of high interest such as watersheds, areas of earth crust movement such as the Pacific "ring of fire, etc. maybe on monthly basis or even on demand basis. In order to do this on a regular and reliable basis will require a substantial fraction of exascale computing resources.

Other Geospatial BDEC activities

Of course, DEMs and their use are only one part of the geospatial research and production, many of which have similar levels of requirements. Many other methods are in use and will need to be supported on BDEC resources. Some of these areas are image classification, identification and isolation, crop and vegetation analysis, computer vision, weather and climate, oceanography, human population responses to events be they droughts and floods, to infectious disease outbreaks and other stimuli. All these methods require large amounts data from many sources and very significant amounts of computation and data analysis.

Pervasive, Personalized and Precision (P^3) analytics for massive bio-social systems

A white paper for the Big Data and Extreme-scale Computing workshop, November 2018

Madhav V. Marathe, Christopher L. Barrett
Biocomplexity Institute and Initiative & Department of Computer Science.
Email: {marathe,clb5xe}@virginia.edu

Motivation: We are concerned with the range of conceptual, theoretical, technical and use issues related to providing actionable information to individuals related to at-scale, massively interacting, bio-social/technical systems. These co-evolving environments are often very large but not large enough so that they can be approximated using mean field theories. They are comprised of intricately networked biological, social, informational, technological and infrastructural components. Examples abound: from natural and artificial ecosystems, including mega-cities and natural ecologies to human and animal immune systems. Such systems are characterized by decentralized coordination, adaptation and memory, heterogeneity and non-stationarity with various time-varying interactions. Perhaps most importantly, these networked systems co-evolve as a result of the dynamical interactions within and amongst them. At any time system knowledge is provisional and subject to revision, whether this knowledge involves historical, current or projected system behavior. As a result "best" estimates or "best" actions, in general, will change over time, experience and are sensitive to predecessor state histories. For example, the structure of a mega-city is not just dictated by legal framework and centralized governing bodies, but also and largely dictated by the various simultaneously co-defined functional networks of its inhabitants. This sort of decentralized, yet agentic, functionality and structure is constantly evolving at multiple time, space and social scales. Technologies ranging from pharmaceuticals to transportation and communication create ever new layers for functional embodiments, interaction and phenomenological innovations. Advances in computing, information sciences and sensor systems have led both to the emergence of entirely new social phenomena arising from massive interaction with the cognitive, biological and physical environments in the ICT layers and to the possibility of pervasive, personalized and precision analytics and decision making within, for, and among these systems. Such services need to be understood and supported at various levels of organizational hierarchy – from individual constituent elements to systemic level wherein certain centralized or understandable coordination of decentralized decision making might be possible.

Computational challenges: Interestingly, while advances in computing, communication and information technology has fueled the desire and vision of P^3 analytics --- this vision brings forth new challenges far greater than simple implementation and fielding of the individual computational technologies. Many of the challenges are beginning to be perceived and articulated in the so-called Big Data and extreme-scale computing worlds and certainly are applicable here. Some salient ones that are worth noting here include: (*i*) distributed, pervasive decision making in real-time and multiple stake holders – leading to systems that constantly sense, act and adapt in response to the bio-social system; (*ii*) inability to conduct typical experiments at scale that can often be done for physical systems (e.g. large scale disaster scenarios) – leading to the need for developing at-scale computational models and experiments to study such systems (*iii*) lack of fundamental and well accepted theories that describe how such systems work (one could argue that there is really no universal law describing human mobility – indeed movements will coevolve with the social structures we create) – leading to development of models that are informed by theories but also by observed data; and (*iv*) notion of validity and reproducibility is likely to be substantially different than physical sciences – leading to design and development of systems that are done so for a reason. Additionally we believe it is centrally important that we must actively develop in adversarial information environments, adopt a provisional knowledge perspective, and take very seriously the natural properties of spontaneously appearing and engineered decentralized agency.

An abductive multi-scale systems to support P^3 analytics and decision making: One critical aspect of provisional knowledge systems is the problem of imputing unclear and multi-scale complex system state for purposes of system management and the problem of *adapting action related knowledge*. We envision the need for *abductive systems* that support state and action reasoning for decision making in massive biosocial systems environments. Such systems, will need to constantly sense the environment , and modify their internal representation of the system based on the observations. They will also need to revisit their actions against current knowledge. In other words, decision making, sensing and analytics is done based on a provisional model and knowledge of the underlying system that evolves in time. We do not use the term incomplete knowledge (or information) since in many cases, the system itself is adapting and hence there is no clear notion of complete knowledge. Indeed there is no asymptotically true state over time in general for these rather radically nonstationary environments,. This view is inspired by a cognitive view of how natural decision making systems work. Such a view immediately reconciles with the fact that information is inherently noisy, incomplete and often manipulated. The notion of validity and best description of the system is based on the notion of abduction. It acknowledges that larger outcomes are really the result of individual decisions and outcomes and in this sense predictive or retrospective validity cannot be the only measure of success. One will have to extend the notion of abduction to distributed, local and hierarchical abduction: abduction by distributed agents for estimate their environment; abduction by agents based largely on local information and finally abduction at various organizational levels.

Prediction Science:
the 5th Paradigm Fusing the Computational Science and Data Science

by
Takemasa Miyoshi
RIKEN Center for Computational Science, Kobe, Japan
takemasa.miyoshi@riken.jp

High-performance computation (HPC) has consistently been extending the capability of more realistic simulations in various application fields. As a simulation becomes more precise and accurate, real-world data play an increasingly important role in improving the simulation. Moreover, real-time use of data will synchronize the simulation with the real world and enable predicting the future.

Numerical Weather Prediction (NWP) is a successful example of fusing the HPC-based "Big Simulation" with real-world "Big Data" through a method known as data assimilation. As the NWP model became more and more accurate, the relative importance of data assimilation has increased. In the past two decades, data assimilation was considered as important as the NWP model itself. As a result, data assimilation grew rapidly to be a major field in meteorology.

In the past BDEC meetings, I have presented our work on Big Data Assimilation in NWP. We assimilated roughly two orders of magnitude more data from a new radar system with a 100-m-mesh NWP model, 100 times more grid points per area than a typical high-resolution NWP system at 1-km resolution. We examined the feasibility of local severe storm prediction refreshed every 30 seconds, 120 times more rapidly than a typical hourly-update system. This will bring a revolution to weather forecasting, while such a revolutionary NWP system requires intense FLOPS and I/O speed to meet the strict real-time requirement for the 30-second update frequency.

The general concept of data assimilation is to fuse an HPC-based simulation in the cyberworld with the real-world data. This is a realization of a broader concept of cyber-physical system with HPC and Big Data. Here we propose to generalize the success of NWP and data assimilation to broader simulation fields. We have a high-precision simulation synchronized with real world and select the best future scenario based on the simulation with quantified uncertainties. This is what we call the Prediction Science, the 5th paradigm fusing the Computational Science (3rd science) and Data Science (4th science), extending the capability of prediction and control to large and complex systems with which traditional approaches have difficulties.

To create the new Prediction Science, it is essential to bring together the process-based model simulation and data science approaches such as AI and machine learning techniques through modern advanced mathematics including the uncertainty quantification (UQ). This requires new-generation computing resources with high computing capacity with fast networking and large storage access. Simulations will use more FLOPS with more I/O with increasing variety and volume of real-world data. AI and machine learning techniques will be integrated with data assimilation, but these have very different computational requirements. Namely, AI and machine learning techniques are efficient with

GPUs, while simulation and data assimilation codes are efficient with CPUs most of the time. It would not be easy to use GPUs for high-precision simulation and data assimilation. For integrated Prediction Science applications, we will need to have access to both architectures in a seamless manner.

I would like to bring to light the potential future directions of “Prediction Science” fusing Computational Science and Data Science and to discuss what computational needs are expected to enable prediction in broader areas. To make this happen, it will be essential to have synergistic interactions among computer scientists, experts in broad application areas, and theoretical and mathematical scientists.

High End Data Science and HPC for the Electrical Power Grid

Alex Pothen and Ariful Azad
Purdue University and Indiana University

The U.S. electrical power grid is the largest machine ever built, and the National Science Foundation, the Department of Energy and other federal agencies are calling for research to improve the power system reliability through wide area measurement and control by employing syncrophasor technology (Phasor Measurement Unit sensors, PMUs). Over 2500 PMU sensor units are deployed in the North American power grid. PMUs obtain 30-60 samples per second, generating Petabytes of data per day, and providing real-time situational awareness leading to early warning of grid events and dynamic behavior of the grid. It can help with strategies to recover from natural disasters, large variations in the power generation from renewable sources, cyberattacks, etc. These include planning ahead for events and shocks, decision and control as the event unfolds and for post-shock recovery to a steady state. The North American Power Grid Initiative (NASPI) promotes the use of syncrophasor data to estimate and control the state of the grid.

An important use of PMUs is in oscillation detection in the power grid, and the predominant computation here is computing the dominant singular values and singular vectors of a covariance matrix computed from the time varying data matrices from a formulation for controlling the grid. We have recently deployed randomized SVD algorithms and Lanczos algorithms on parallel computers to enable this computation without forming the covariance matrix itself, employing only matrix-vector products for this purpose [1]. We are able to compute SVDs for data from hundreds of PMU sensors in a matter of seconds.

A second application of high-end data analytics in the power grid that we have considered is in contingency analysis, where power grid operators have to determine how to stabilize the grid when a few transmission lines or generators should go down. We have modeled this scenario in the case of DC power flow and have shown that an augmented matrix formulation can be used to update solutions to Kirchoff's equations three orders of magnitude faster than a method that would solve the modified system of equations directly [2].

A third application of high-end data analytics in the power grid is in vulnerability analysis of power networks. Vulnerability analysis is aimed at detecting vulnerable sections of a power grid subject to cyberattack, natural disasters or mechanical failures. Traditionally, power grids and flows are analyzed using non-linear, numerical methods. However, under certain assumptions, graph-theoretical methods can be useful to study optimal power flow and vulnerability of power networks. It is widely believed that power grids exhibit the properties of scale-free networks, especially when a power network is described as a function of its “electrical topology” [3]. Consequently, power networks are robust to random attacks and failures, but they can be vulnerable to targeted attacks. To model power flow and vulnerability, prior work considered the maximum flow problem from graph theory [4]. However, as electricity flows differently than fluids, specialized flow models are required for power networks. To this end, an exciting avenue of research is in designing realistic flow models for power networks and using them to analyze dynamic power network with random failure of equipment. The problem is even more exciting when power flow can be controlled such as with Flexible Alternating Current Transmission System (FACTS) [5].

In developing algorithms and libraries for power grids, computer scientists are often challenged by lack of reliable data due to privacy and security concerns. One approach to tackle this challenge is to generate synthetic data based on public information about power stations and demographic data. For example, ARPA-E Grid Data program supports several projects to generate synthetic data with realistic generation and load profiles (https://electricgrids.engr.tamu.edu/). These datasets can be augmented further to make them reliable replicas of real grids without exposing secure information.

Today's dynamic and massive power grids require easy-to-use algorithms and libraries deployed on HPC infrastructures. A joint effort from power grid analysts, data scientists and HPC experts can play pivotal roles in rapid analysis of power networks for potential cyber threats, vulnerability and swift recovery schemes after an attack or mechanical failure. The HPC community can provide high-performance software that can be easily deployed to the cloud for out-of-the-box analysis.

References

[1] Tianying Wu, Vaithianathan ``Mani" Venkatasubramanian, and Alex Pothen, Fast parallel stochastic subspace algorithms for large-scale ambient oscillation monitoring, IEEE Transactions on the Smart Grid, 8(3), pp. 1494-1503, 2017.

[2] Yu-Hong Yeung, Alex Pothen, Mahantesh Halappanavar and Zhenyu Huang, AMPS: An augmented matrix formulation for principal submatrix updates with application to power grids, SIAM J. Scientific Computing, 39 (3), S809-S827, 2017.

[3] Paul Hines, and Seth Blumsack, A centrality measure for electrical networks, Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008.

[4] Ajendra Dwivedi, Xinghuo Yu, and Peter Sokolowski, Analyzing power network vulnerability with maximum flow based centrality approach, 8th IEEE International Conference on Industrial Informatics, 2010.

[5] Franziska Wegner, Network Flow Models for Power Grids. Diss. Karlsruhe Institute of Technology, 2014.

Real-Time Anomaly Detection from Edge to HPC-Cloud

Judy Qiu[1], Bo Peng[1], Ravi Teja[2], Sahil Tyagi[1], Chathura Widanage[1], Jon Koskey[3]

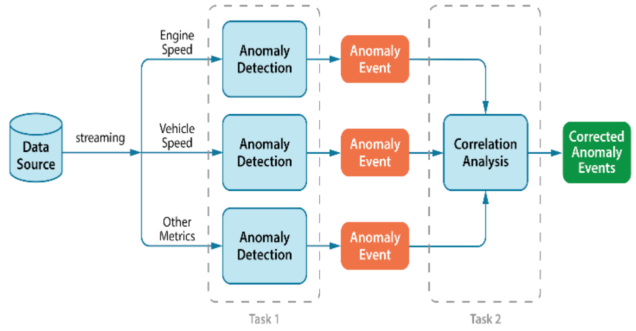
[1]Indiana University
[2]India Institute of Technology
[3]IndyCar

Abstract

Telemetry data plays an important role in many areas such as motor racing, meteorology, agriculture, transportation, manufacturing processes and energy monitoring to name a few. There lies a direct intersection between embedded computing and big data analysis. In the domain of motor racing, telemetry data is very widely used for analyzing or improving the performance of the racing cars and monitoring the effects of racing towards the physical status of the race car drivers. A large number of sensors, in the number of 100s, are fixed on the racing car. There are on-board and off-board electronic systems that transmit the sensor readings to teams on the pit and electromechanical systems that control the fuel utilization, throttling, etc. The generated sensor readings are then analyzed using several big data analysis methods, with detecting anomaly events at edge and conducting correlation analysis for high dimensional data on HPC and Cloud.

Keywords Time Series, Anomaly Detection, Online AI/ML

Challenges and Impact of Anomaly Detection on Time Series Datasets



Anomaly detection is a heavily studied area of data science and machine learning. It refers to the problem of finding patterns in data that do not conform to expected behavior [1]. Detection of anomalies, especially temporally in real-time streaming data, has significant importance to a wide variety of application domains, as it can give actionable information in critical scenarios. In this streaming application, data are observed sequentially, and the processing must be done in an online fashion, i.e., the algorithm cannot rely on any look-ahead procedures.

Traditional time-series modeling and forecasting models can be utilized to detect temporal anomalies. Approaches based on ARIMA are capable and effective for data seasonal patterns [2]. Techniques based on relative entropy, graph [3, 4] are also utilized to detect temporal anomalies. Another mainstream approach is to build simulation model with explicit domain knowledge for domain-specific applications. However, model-based approaches are limited for lack of generalizability. A novel approach was proposed in [5] to use Hierarchical Temporal Memory (HTM) [6, 7] networks to robustly detect anomalies on real-time data streams. HTM is a state-of-the-art online machine learning technology that aims to capture the structural and algorithmic properties of the neocortex. Figure 1 outlines the steps to create a complete anomaly detection system. The input time series x_t are fed to the HTM component. It models temporal patterns in a(x_t) and output a prediction in $\pi(x_t)$. Then by building a statistical model on the prediction error, $\pi(x_t)$ - a(x_{t-1}), anomaly likelihood score can be calculated on x_t.

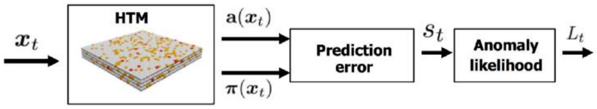
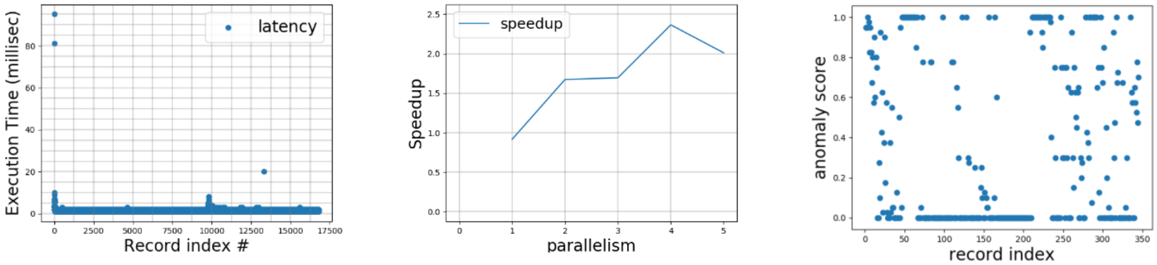


Figure 1 Anomaly Detection System Based on HTM

Convergence of HPC, Big Data and Machine Learning

We apply anomaly detection algorithms on the dataset of Indycar race held on May 28, 2017. The raw log file contains around 670 megabytes of data, amounting to roughly 368,000 individual records logged throughout the race from 33 racing cars with an average speed between 200 and 250 miles per hour. We run HTM detection at various parallelism on Apache Storm and measure the speedup with respect to the batch HTM job. We also match batch HTM with Storm on a parallelism of 1 to determine the overhead incurred in a distributed framework. The batch and stream experiments run on a single Linux node with 48 CPU cores of Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz. The total available memory is 125 gigabytes. The minimum Java heap size (-Xmn) set for the batch mode is 16GB.



(a) Execution latency in batch mode

(b) Speedup in HTM detection using Apache Storm

(c) Anomaly Scores of 345 injected anomalous datapoints

Figure 2 Performance and Validation of HTM over Streaming Dataset

The experiments run on the eRP (enahnced results protocol) data for car #9. This is due to the car (by driver Scott Dixon) suffered a car crash during the race and provides ample potential anomalies for us to detect. For the sake of simplicity, we predict anomalies based on the two parameters: ***time_of_day*** and ***vehicle_speed*** from Dixon's car, which has a dataset containing around 17,263 records. Fig 2(a) shows the execution latency (time taken to predict anomaly on a single input record) to process each record of car #9 on Apache Storm framework with parallelism of 1. An average execution time to predict an anomaly is 1.43 milliseconds. Fig 2(b) shows the speedup, the ratio of time to predict anomalies on a serial process to the time taken by running Storm at various parallelism levels. To establish a sense of ground truth on labels and validate our approach, we deliberately inject anomalies at 2% data fraction (~ 345 data points) at known indexes. We inject speeds of 0.00 mph (absolute vehicle halt) at various indexes. Anomaly score of 0.0 implies a normal event. The anomaly scores of each of the 345 data points is shown in Fig 2(c).

Conclusion

It is an important research topic on the convergence of HPC and Big Data to help people select appropriate computing hardware and software architectures based on the characteristics of different AI algorithms and applications. HTM is a special type of neural network on sparse data representation and operations for anomaly detection. HPC is needed to harness high-speed data with increasing complexity of predictive analytics, e.g. from all 33 cars, all 150 car sensors, all 10 timing sensors laid on the track with 1/10 millisecond accuracy, and 36 cameras streaming to 6 video feed servers, plus the possibility of sophisticated anomalies in issues like the way drivers take curves and strategies to deliver an exciting data-driven experience.

References

[1] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. ACM Comput. Surv., 41(3):15:1–15:58, July 2009.

[2] A. M. Bianco, M. G. Ben, E. J. Martnez, and V. J. Yohai. Outlier Detection in Regression Models with ARIMA Errors using Robust Estimates. Journal of Forecasting, 20(8):565–579.

[3] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery, 29(3):626–688, 2015.

[4] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest-based anomaly detection on streams. In International Conference on Machine Learning, pages 2712–2721, 2016.

[5] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262:134–147, Nov. 2017.

[6] J. Hawkins and S. Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. Frontiers in neural circuits, 10:23, 2016.

[7] Y. Cui, S. Ahmad, and J. Hawkins. Continuous online sequence learning with an unsupervised neural network model. Neural computation, 28(11):2474–2504, 2016.

Semantic Segmentation of Underwater Sonar Imagery based on Deep Learning

Maryam Rahnemoonfar
Texas A&M University-Corpus Christi

Extensive degradation of seagrass beds is taking place in coastal areas around the globe because of natural and human induced disturbances. These negative impacts affect approximately 65% of the original seagrass communities, mainly in Europe, North America, and Australia. Mapping of seagrass degradation due to natural and human disturbances such as potholes and propeller scars is essential to estimating overall abundance, disturbance regimes, and the overall health of related marine systems.

It is difficult to detect seagrass disturbances under water with optical sensors because light is attenuated as it passes through the water column and reflects back from the benthos causing errors in calculations. Underwater acoustic techniques have allowed many advances in the field of remote sensing and these techniques can be used to produce a high-definition, 2-D sonar image of seagrass ecosystems. In this study we use sonar sensors for pattern identification in seagrass.

Recent years have witnessed enormous advancement in the pattern recognition research based on deep learning. Majority of deep learning methods are developed for RGB imagery. However, for many applications such as detecting objects underwater other types of sensors such as sonar or radar are required. Here we developed a new deep learning framework based on Dilated Convolution and Inception Densenet to perform semantic segmentation for automatic extraction of potholes in underwater sonar imagery. Side scan sonar images usually contain speckle noise and uneven illumination across the image. Moreover, disturbance presents complex patterns where most segmentation techniques will fail. We tested our proposed approach on a collection of underwater sonar images taken from Laguna Madre in Texas. Experimental results in comparison with the ground-truth and state-of-the-art semantic segmentation methods show the efficiency and improved accuracy of our proposed method.

Title:

Elevating the Edge to be a Peer of the Cloud

Abstract

Technological forces and novel applications are the drivers that move the needle in systems and networking research, both of which have reached an inflection point. On the technology side, there is a proliferation of sensors in the spaces in which humans live that become more intelligent with each new generation. This opens immense possibilities to harness the potential of inherently distributed multimodal networked sensor platforms (aka Internet of Things - IoT platforms) for societal benefits. On the application side, large-scale situation awareness applications (spanning healthcare, transportation, disaster recovery, and the like) are envisioned to utilize these platforms to convert sensed information into actionable knowledge. The sensors produce data 24/7. Sending such streams to the cloud for processing is sub-optimal for several reasons. First, often there may not be any actionable knowledge in the data streams (e.g., no action in front of a camera), wasting limited backhaul bandwidth to the core network. Second, there is usually a tight bound on latency between sensing and actuation to ensure timely response for situation awareness. Lastly, there may be other non-technical reasons, including sensitivity for the collected data leaving the locale. Sensor sources themselves are increasingly becoming mobile (e.g., self-driving cars). This suggests that provisioning application components that process sensor streams cannot be statically determined but may have to occur dynamically.

All the above reasons suggest that processing should take place in a geo-distributed manner near the sensors. Fog/Edge computing envisions extending the utility computing model of the cloud to the edge of the network. We go further and assert that the edge should become a peer of the cloud. This white paper is aimed at identifying the challenges in accomplishing the seamless integration of the edge with the cloud as peers. Specifically, we want to raise questions pertaining to (a) frameworks (NOSQL databases, pub/sub systems, distributed programming idioms) for facilitating the composition of complex latency sensitive applications at the edge; (b) geo-distributed data replication and consistency models commensurate with network heterogeneity while being resilient to coordinated power failures; and (c) support for rapid dynamic deployment of application components, multi-tenancy, and elasticity while recognizing that both computational, networking, and storage resources are limited at the edge.

Bio

Professor Umakishore Ramachandran received his Ph. D. in Computer Science from the University of Wisconsin, Madison in 1986, and has been on the faculty of Georgia Tech since then. For two years (July 2003 to August 2005) he served as the Chair of the Core Computing Division within the College of Computing. His fields of interest include parallel and distributed systems, computer architecture, and operating systems. He has authored over 100 technical papers and is best known for his work in Distributed Shared Memory (DSM) in the context of the Clouds operating system; and more recently for his work in stream-based distributed programming in the context of the Stampede system. Currently, he is leading a project that deals with large-scale situation awareness using distributed camera networks and multi-modal sensing with applications to surveillance, connected vehicles, and transportation. He led the definition of the curriculum and the implementation for an online MS program in Computer Science (OMSCS) using MOOC technology for the College of Computing, which is currently providing an opportunity for students to pursue a low-cost graduate education in computer science internationally. He has so far graduated 30 Ph.D. students who are well placed in academia and industries. He is currently advising 5 Ph.D. students. He is the recipient of an

NSF PYI Award in 1990, the Georgia Tech doctoral thesis advisor award in 1993, the College of Computing Outstanding Senior Research Faculty award in 1996, the College of Computing Dean's Award in 2003 and 2014, the College of Computing William ``Gus" Baird Teaching Award in 2004, the ``Peter A. Freeman Faculty Award" from the College of Computing in 2009 and in 2013, the Outstanding Faculty Mentor Award from the College of Computing in 2014, and became an IEEE Fellow in 2014.

Smart Community CyberInfrastructure at the Speed of Life

A White Paper for BDEC2 by Glenn Ricart, US Ignite and U. Utah

At the Speed of Life

Smart Community cyberinfrastructure collects data at the speed of life and similarly often needs to respond in sync with the real world. If vehicles will communicate and coordinate their way through intersections without stopping, there must be reliable and timely information about the vehicles and their location, motion, and acceleration/deceleration fed into complex cyberphysical system decision-making which in turn is sent back to and verified by the vehicles in a trust-worthy way. All of that must occur in at most a few thousandths of a second. Smart community cyberinfrastructure must respond at the speed of life. (This is not unlike real-time data collection in HPC science experiments where the real-time results are coupled to and alter the experimental controls in real-time.)

Over time, smart and connected communities are finding a growing number of cyberphysical systems that improve their communities and its healthcare, public safety, education, economic development, and recreation, but which need to reliably operate at the speed of life in that community.

Characterizing the Smart Community Applications of the Future

As part of the NSF-sponsored *Looking Beyond the Internet* series of workshops held in 2015-2016, Glenn Ricart and Prasad Calyam held a workshop on *Applications and Services in the Year 2021*. That report noted that several of the sciences increasingly would rely on data-driven and AI-driven systems with strict **time constraints**. Data will most often be presented in streams. It often will be voluminous. Reliability and calibration may not be as high as in HPC scenarios due to commercial cost constraints. Context and metadata will be as important as the data itself. There are serious requirements for being immune to attack that don't arise in scientific instrumentation scenarios. While many lessons of HPC and distributed systems are highly useful, they are not sufficient in this environment.

In addition, sensor-driven data is often of use primarily in localized environments. In computer science terms, it has high **locality**. In addition, the data itself often is **perishable**. It may well be archived for future study, but its actionable information value fades as newer values from the same stream arrive.

Enter Einstein

In these circumstances of perishable data with high locality and time constraints for actions taken, the ability to use the currently dominant model of sending all data to huge and highly efficient datacenters or supercomputers is limited due to the speed of light in fiber. To achieve adequate response time and sufficiently reliable service, edge computing has become an important tool.

The Edges and their Clouds

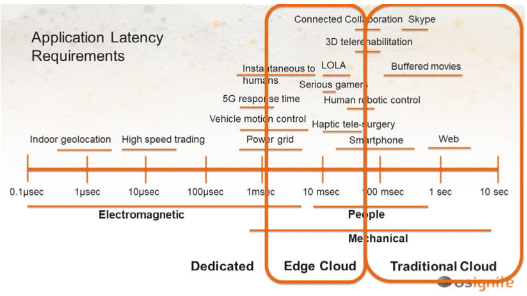
The leading edge of edge computing thinking is multiple edges. Different portions of an application may have differing requirements for response time, reliability, security, etc. Each microservice invoked has its own requirements which may be most efficiently handled through a network of edge clouds. It may often be better to move computation to the data than data to the computation. Computation and data need each other; they are cyberinfrastructure duals. It's a challenging and unsolved multi-dimensional optimization in time, space, and capability at the very least. Coordinated multi-edge and multi-service orchestration and hypervision will need to be part of the answer.

At the *Second National Research Platform Workshop: Toward A National Big Data Superhighway* (August 6-7, 2018), a major advance was recognition of the role of microservices and containers in high performance platforms. This means HPC and big data are moving to embrace distributed systems techniques while distributed systems are moving to embrace HPC and big data techniques.

Due to the current lack of suitable multiple-edge cyberinfrastructure, Rick McGeer, the author, and US Ignite are using an NSF grant to develop Edge-Net, a Kubernetes-based distributed container and VM driven viral edge cloud designed to service challenging smart community applications and other science.

Application Requirements

The author works primarily in the world of smart and connected community applications and services and their infrastructure. If selected to present this white paper at the BDEC2 conference, most of the time would be spent on smart community requirements for cyberinfrastructure, including volume, time constraints, reliability, resistance to attack, economics, veracity, privacy, and value to society. US Ignite has more than a hundred applications and services developed or under development in its 26 Smart Gigabit Communities, and the most demanding of these in terms of data and computation will be discussed along with their requirements. Examples will come from transportation, energy management, healthcare, and education, depending on the time available. To prototype these applications, we've had to handcraft some edge clouds and national network paths. For a few, we have videos in which the PIs describe them.



Related Sources of Application Requirements

Additional sources of smart and connected community application requirements come from the Global City Teams Challenge and their Action Cluster blueprints, and from the EU effort for FIWARE Smart Cities. In Asia, smart city requirements are present in the NSF-funded CENTRA effort. The EU, Asia, and Brazil have committed to joint smart communities applications and infrastructure research via GEFI. US Ignite, together with Northeastern University, is responsible for the PAWR advanced wireless research which addresses the high bandwidth and low latency wireless infrastructure needed by these applications and services.

US Ignite is working with the Alliance for Telecommunications Industry Solutions (ATIS) to develop a data exchange for smart community information. This effort is intended to help make smart community data more comparable and actionable, improving the deployability of smart community applications.

Glenn Ricart is co-founder of the San Diego Supercomputer Center and co-author of the Federal HPC Bluebooks in the 1990s. He was mentor to the West Big Data Hub during its formation. His current effort is US Ignite, a nonprofit where he's helping solve the chicken-and-egg problems in Smart and Connected community applications and their edge clouds. He holds an adjunct appointment at the University of Utah School of Computing.

Big Data and Extreme Scale Computing, 2nd Series (BDEC2) White Paper - Statement of Interest from the Square Kilometre Array Organisation (SKAO)

The Square Kilometre Array (SKA) and the High Luminosity - Large Hadron Collider (HL-LHC) are Landmark projects in the area of Big Data and Extreme Scale Computing. In 2017 we launched a formal collaboration on extreme scale computing to work together on areas of mutual interest.

The SKA project will build a new astronomical observatory spanning three continents - Europe, Africa and Australasia - and provide the scientific community with a step-change in radio astronomy capability with two new telescopes: one operating in the frequency range 50-350 MHz - SKA1-LOW in Western Australia - the other operating in the frequency range 350 MHz to 15 GHz - SKA1-MID in South Africa. Raw data rates in the petabit/s range need to be reduced through a number of processing stages to output science data products in the rate of GB/s.

The existing Large Hadron Collider at CERN will undergo a major upgrade in the early 2020s. In the HL-LHC phase the data rates will grow by a factor of 10 with respect of the current running conditions and the complexity of the physics events will also increase considerably. We foresee the need to store and process exascale-level of data while we do not foresee an increase of the budget for computing hardware.

Both projects are large and complex big data challenges. In addition to enormous on-line real-time data and processing challenges, several hundred PB/year will be transported around the globe to a network of centres for further analysis. The SKA Regional Centres (SRCs) will be similar to the Worldwide LHC Computing Grid operated by CERN and its partners. The data centres will likely be realised by a network of cyberinfrastructure shared with other science communities.

The specific application requirements are unique, but there is also commonality with other disciplines. Below we discuss the three areas highlighted in the call for contributions, give our perspective on these matters, highlighting where the our computing requirements perhaps differ, but also where we share the concerns with other data-intensive sciences communities.

1. Novel models of integrated inquiry.

   Neither the LHC’s existing applications or those envisaged for the SKA once in operation can be classed as typical HPC workloads. We do recognise the and concur with the recommendations laid out on p4 of the BDEC “Pathways to Convergence” report regarding decentralized edge systems and centralised facilities and the need for these infrastructures to support a variety of complex workflows generating considerable volumes of data, some with demanding real-time characteristics, over wide geographical areas. Both projects are keen to further understand how developments in machine learning and deep learning might be applied to problems ranging from new end-user analysis methods and tools to aid visualization and comprehension of huge data sets, to more efficient algorithms for data reduction and ways of improving the reliability, availability and maintainability of very complex scientific instruments and their supporting computing infrastructure across the globe. A recently held workshop at the Alan Turing Institute in London brought together members of the high-energy physics and astronomy communities with a view to identifying areas of commonality and avenues for future collaboration. Extending these conversations further to include representatives from across the communities represented in BDEC2 is of real interest to us.

2. Support for advanced data logistics.

   The ‘extreme scale’ data production of the likes of SKA and HL-LHC will require ways of developing new cost-effective and user-friendly cyber-infrastructures that are able to capture, analyse and store vast quantities of data for decades. The challenge will be to ensure that these infrastructures will span continents and be able to utilise a heterogeneous pool of resources: HTC, HPC, clouds, CDNs etc. We refer to this model as a ‘data lake’ (but this is perhaps a misnomer in that it differs to how the term is interpreted in a commercial ICT context). The LHC experience shows that delivering data to processing cores in typical HPC systems in an efficient way is a concern. The problems are twofold: efficient ingest of large volumes of data into the receiving nodes within the HPC platform; and the subsequent efficient and timely placement of data on to processing nodes when using shared parallel file systems and typical HPC batch scheduling methods. Traditional HPC techniques do not work well at petascale and certainly won’t at exascale. The way the convergence of techniques from the HPC and HPDA worlds evolves will be key to data-intensive science communities such as ours being able to fully exploit the full range of cyber-infrastructures expected to emerge in the coming decade.

   The infrastructure we will use within the SRCs and WLCG will most likely be shared with peers in other communities: we do not envisage member states funding new infrastructure exclusively for use by the SKA or LHC science community. Therefore it is vital to work with other projects to look at how common tools and techniques around virtualization, containerization in particular (e.g. the “hour glass” model), along with common analytical front-end platforms for driving varied workflows can be developed to meet our needs.

3. Interfaces to commercial cyberinfrastructure.

   SKA and HL-LHC view commercial cyberinfrastructure as part of the overall landscape but we are not concerned with users working at the ‘long tail’ since the user communities will be highly organised around the SRC and WLCG modes of operation. We see the use of commercial clouds as potentially complementary to dedicated resources within the regions around the globe. Current analysis of the overall cost of using commercial cloud suggest that it is not an affordable way of providing the majority of cycles our communities require. (But of course the long-term costs of commercial clouds vs. on-premise provision are hard to predict.) The LHC community has carried out large scale demonstrator projects including US labs (BNL and FermiLab) in order to reach these conclusions. Regardless of the uncertain long-term financial picture, both the SKA and LHC communities would like to fully explore how infrastructure owned by the projects might be seamlessly integrated with publicly funded shard cyberinfrastructures and privately operated public clouds. It should be noted that the appetite for using the leading commercial cyber-infrastructures will likely vary from region to region. For example, the European Commission is clearly concerned about developing deeper dependencies on US-based tech monopolies, and is therefore funding programmes that might lead to greater “technological sovereignty” such as the European Processor Initiative and the European Open Science Cloud. Similarly, in China there are number of exascale initiatives that will be based on homegrown technology, and the country’s science community will have its own commercial cloud infrastructure to use such as Alibaba. As a global project, the SKA needs to develop strategies that bear in mind these regional differences and what drives them, and develop its SRC network accordingly.

The BDEC2 community might also wish to consider whether a collective approach could be adopted to more effectively lobby governments and argue for higher funding levels to meet the challenges we face.

Big Data & Extreme Scale Computing, 2nd Series, (BDEC2)
Workshop 1: Bloomington, Indiana, November 28-30, 2018

BDEC-2 WHITE PAPER:
Category/Focus Area:

Novel models of integrated inquiry: Unprecedented new methods in high-end data analysis (HDA) that are being pioneered in Big Data communities, such as Deep Learning, will increasingly be combined and integrated with the simulation-centric approaches of traditional high performance computing (HPC). In other words, the impact of the digital revolution on scientific methodologies has entered a dramatic new phase.

Author: William Tang (Princeton University/PPPL)

Background & Challenge: Accelerating delivery of accurate predictions in key scientific domains of current interest can best be accomplished by engaging modern big-data-driven statistical methods featuring machine learning/deep learning/artificial intelligence (ML/DL/AI). Associated techniques have been formulated and adapted to enable new avenues of data-driven discovery in prominent scientific applications areas such as the quest to deliver Fusion Energy -- identified by the 2015 CNN "Moonshots for the 21st Century" series as one of 5 exciting grand challenges. An especially time-urgent and very challenging problem facing the development of a fusion energy reactor is the need to reliably predict and avoid large-scale major disruptions in magnetically-confined tokamak systems such as the EUROfusion Joint European Torus (JET) today and the burning plasma ITER device in the near future. Significantly improved methods of prediction with better than 95% predictive capability are required to provide sufficient advanced warning for disruption avoidance or mitigation strategies to be effectively applied before critical damage can be done to ITER -- a ground-breaking $25B international burning plasma experiment with the potential capability to exceed "breakeven" fusion power by a factor of 10 or more. This truly formidable task demands accuracy beyond the near-term reach of hypothesis-driven /"first-principles" extreme-scale computing (HPC) simulations that dominate current research and development in the field.

Approach & Advances: Recent HPC- relevant advances in the deployment of deep learning convolutional and recurrent neural nets have been demonstrated in exciting scaling studies of Princeton's Deep Learning Code -- "FRNN (Fusion Recurrent Neural Net) Code -- on modern GPU systems. This is clearly a "big-data" project in that it has direct access to the huge EUROFUSION/JET disruption data base of over a half-petabyte to drive these supervised machine-learning studies. FRNN implements a distributed data parallel synchronous stochastic gradient approach with "Tensorflow" libraries at the backend and MPI for communication. This deep learning software has demonstrated excellent scaling up to 6000 GPU's on Titan that has enabled clear progress toward the goal of establishing the practical feasibility of using leadership class supercomputers to greatly enhance training of neural nets that can enable transformational impact on key discovery science application domains such as Fusion Energy Science. In addition to (1) Titan at the OLCF, powerful systems currently deployed by the Princeton U/PPPL deep learning software include: (2) Japan's "Tsubame 3" system with 3000 P-100 GPU's; and (3) OLCF'S "Summit" system during it's Early Access Phase. Achieving accelerated progress in statistical Deep Learning/AI software trained on very large data sets hold exciting promise for delivering much-needed predictive tools capable of accelerating scientific knowledge discovery in HPC. The associated creative methods being developed also has significant potential for cross-cutting benefit to a number of important application areas in science and industry. This work was recently awarded the 2018

NVIDIA Global Achievement Award – (https://insidehpc.com/2018/03/princeton-team-using-deep-learning-develop-fusion-energy/; and https://www.princeton.edu/news/2018/04/02/william-tang-wins-2018-global-impact-award-advance- development-ai-software-help.) Moreover, the project on "Accelerated Deep Learning Discovery in Fusion Energy Science" has been selected as one of the DOE-ALCF-21 Early Science Projects that will feature advanced INTEL architectures: https://www.alcf.anl.gov/articles/alcf-selects-data-and-learning-projects-aurora-early-science-program; https://www.hpcwire.com/off-the-wire/deep-learning-to-predict-fusion-disruptions-picked-for-first-us-exascale-system/

Relevance AI/Deep Learning Co-Design: The applied math and computer science algorithms being developed in our exemplar Tokamak fusion DL/AI project have significant potential for cross-cutting benefit to a number of important application areas in science and industry. For example, this work is featured as the Plasma Fusion "exemplar" in the Big Data & Extreme Computing (BDEC) Report (J. Dongarra, et al. 2018, https://www.exascale.org/bdec/). In addition, as evident from examining some of the hyper-parameter tuning workflows being developed in the fusion energy application studies, there are some similarities with those displayed in deep-learning projects such as "Candle" – the DOE/NIH exascale deep learning and simulation studies of precision medicine for Cancer. Accordingly, algorithmic formulations with results (including ROC curves) from current work provide opportunities for productive cross-disciplinary comparisons of methodologies with associated "lessons learned" for co-design. This would also provide a desirable step forward to the delivery of modern DL/AI software capabilities that can be largely "machine and hardware agnostic" with inherent adaptability to expected improvements in architectural designs. As highlighted in IBM's Power-9 rollout, it was noted that "whether its GPU accelerators or FPGA's, our aim is to provide the links and hooks to give all an equal footing in the new server."

Accelerated progress in the further development and deployment of advanced DL/AI R&D for the prediction and mitigation of disruptions in burning plasma fusion Tokamak systems will be significantly enhanced by cross-cutting engagement with leading DOE laboratory scientists. As noted earlier, exciting progress can be realistically anticipated from the collaboration between the clean fusion energy "FRNN" deep learning /AI project at Princeton U/PPPL and the cancer precision medicine "Candle" project at ANL. For example, mutually addressing generically similar but highly challenging hyper-parameter tuning workflow complexities holds great promise for benefiting both application domains. In addition, the new DL/AI capability to include for the first time multi-dimensional signals into the pre-disruption classifiers opens the door for delivery of such classifiers via path-to-exascale HPC simulations – a major step forward in illustrating possible practical "convergence" between big-data-driven AI/DL with exascale HPC simulations.

It is important to highlight at this point another hot topical area of strong interest -- the *demonstration of the actual connection of DL/AI prediction to control in real-world situations.* The fusion energy DL/AI exemplar provides a natural pathway to do so – i.e. moving from accurate prediction to actual control in an active tokamak laboratory environment. This is a key goal which will require collaborative development -- with diagnostics experts -- of actuators capable of showing how reinforcement learning, inference, etc. can positively impact real-time control in a realistic laboratory environment. For example, the leadership of the DIII-D tokamak experiment at General Atomics in San Diego, CA. has already expressed enthusiasm to deploy the Princeton FRNN "predictors" on their plasma control system (PCS). While we continue work on further improving our DL/AI disruption predictor, it's first deployment on an actual PCS promises to be quite exciting. Realistic control theory capabilities connected to DL/AI predictors will in general be a huge area of growth.in many practical application domains.

A position paper

Memory-Storage Hierarchy

Osamu Tatebe
University of Tsukuba
tatebe@cs.tsukuba.ac.jp

The performance gap between CPU and storage is growing wider and wider. Big data applications and deep learning requires further storage performance than checkpointing in HPC applications. To fill the gap, a burst buffer has been proposed [1], and deployed at several supercomputing centers. The burst buffer exhibits the buffering to users explicitly and allows to know users the inconsistent state between the burst buffer and the parallel file system, which provides high storage performance. Currently, it is used to stage-in/out input/output files and to store temporal files during job executions. We, JCAHPC, deployed and operated the burst buffer for a year, and found several issues related to the burst buffer, including maturity of the software, fault handling, capacity management, performance improvement, and metadata performance. Most of these issues will be fixed in a couple of years.

Next step will be how to exploit node local non-volatile memory. Byte addressable non-volatile memory will help to improve transaction performance and write-ahead logging or journaling performance, which will be some evolution of the storage system. Key issue is how to fill the gap between high-bandwidth memory (HBM) and the parallel file system. Between them, there are DRAM, NV-RAM, NVMe SSD, and large capacity SSD. Current burst buffer solution is one approach, but there is a plenty of opportunity for system design research to efficiently utilize this memory-storage hierarchy.

Reference

[1] John Bent, et al., "PLFS: A Checkpoint Filesystem for Parallel Applications", Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009

Cyberinfrastructures for In Situ Data Analytics for Next Generation Molecular Dynamics Workflows

Michela Taufer[1], Michel Cuendet[2], Ewa Deelman[3], Trilce Estrada[4], Rafael Ferreira Da Silva[2], Harrel Weinstein[2]

[1] University of Tennessee Knoxville
[2] Weill Cornell Medical College of CORNELL University
[3] University of South California
[4] University of New Mexico

Molecular dynamics simulations studying the classical time evolution of a molecular system at atomic resolution are widely recognized in the fields of chemistry, material sciences, molecular biology and drug design; these simulations are one of the most common simulations on supercomputers. Next-generation supercomputers will have dramatically higher performance than do current systems, generating more data that needs to be analyzed (i.e., in terms of number and length of molecular dynamics trajectories). The coordination of data generation and analysis cannot rely on manual, centralized approaches as it does now. This interdisciplinary project integrates research from various areas across programs such as computer science, structural molecular biosciences, and high performance computing to transform the centralized nature of the molecular dynamics analysis into a distributed approach that is predominantly performed in situ. Specifically, this effort combines machine learning and data analytics approaches, workflow management methods, and high performance computing techniques to analyze molecular dynamics data as it is generated, save to disk only what is really needed for future analysis, and annotate molecular dynamics trajectories to drive the next steps in increasingly complex simulations' workflows.

This project tackle the data challenge of data analysis of molecular dynamics simulations on the next-generation supercomputers by (1) creating new in situ methods to trace molecular events such as conformational changes, phase transitions, or binding events in molecular dynamics simulations at runtime by locally reducing knowledge on high-dimensional molecular organization into a set of relevant structural molecular properties; (2) designing new data representations and extend unsupervised machine learning techniques to accurately and efficiently build an explicit global organization of structural and temporal molecular properties; (3) integrating simulation and analytics into complex workflows for runtime detection of changes in structural and temporal molecular properties; and (4) developing new curriculum material, online courses, and online training material targeting data analytics. The project's harnessed knowledge of molecular structures' transformations at runtime can be used to steer simulations to more promising areas of the simulation space, identify the data that should be written to congested parallel file systems, and index generated data for retrieval and post-simulation analysis. Supported by this knowledge, molecular dynamics workflows such as replica exchange simulations, Markov state models, and the string method with swarms of trajectories can be executed ?from the outside? (i.e., without reengineering the molecular dynamics code).

Cyberinfrastructure Tools for Precision Agriculture in the 21st Century

Michela Taufer[1] and Rodrigo Vargas[2]

[1] University of Tennessee Knoxville
[2] University of Delaware

This interdisciplinary project applies computer science approaches and computational resources to large multidimensional environmental datasets, and synthesizes this information into finer resolution, spatially explicit products that can be systematically analyzed with other variables. The main emphasis is ecoinformatics, a branch of informatics that analyzes ecological and environmental science variables such as information on landscapes, soils, climate, organisms, and ecosystems. The project focuses on synthesis/computational approaches for producing high-resolution soil moisture datasets, and the pilot application is precision agriculture. The effort combines analytical geospatial approaches, machine learning methods, and high performance computing (HPC) techniques to build cyberinfrastructure tools that can transform how ecoinformatics data is analyzed.

The investigators build upon publicly available data collections (soil moisture datasets, soil properties datasets, and topography datasets) to develop: (1) tools based on machine-learning techniques to downscale coarse-grained data to fine-grained datasets of soil moisture information; (2) tools based on HPC techniques to estimate the degree of confidence and the probabilities associated with the temporal intervals within which soil-moisture-base changes, trends, and patterns occur; and (3) data- and user- interfaces integrating data preprocessing to deal with data heterogeneity and inaccuracy, containerized environments to assure portability, and modeling techniques to represent temporal and spatial patterns of soil moisture dynamics. The tools will inform precision agriculture through the generation and use of unique information on soil moisture for the coterminous United States. Accessibility for field practitioners (e.g., local soil moisture information) is made possible through lightweight virtualization, mobile devices, and web applications.

This award by the Office of Advanced Cyberinfrastructure is jointly supported by the Division of Earth Sciences within the NSF Directorate for Geosciences.

Toward integration of multi-SPMD programming model and advanced cyberinfrastructure platform

Miwako Tsuji
RIKEN Center for Computational Sceince

Agenda : In this paper, we introduce a multi SPMD (mSPMD) programming model, which combines a workflow paradigm and a distributed parallel programming model. Then, we discuss about current issues in the mSPMD regarding data transfer. At the end, we describe future plan to integrate the mSPMD and advanced cyberinfrastructure platform (ACP).

A multi-SPMD programming model

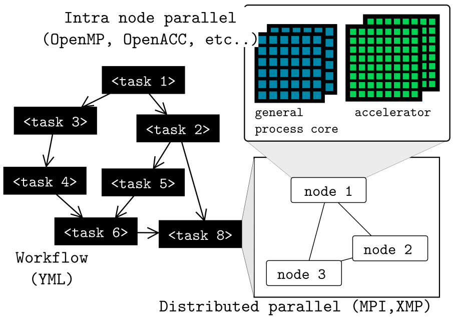


Figure 1: Overview of the multi SPMD programming model

In order to address reliability, fault tolerance, and scaling problems in future large scale systems, we have proposed a multi SPMD (mSPMD) programming model and have been developing a development and execution environment for the mSPMD programming model[4, 3]. The mSPMD programming model combines a workflow paradigm and a distributed parallel programming model for future large scale systems, which would be highly hierarchical architecture with nodes of many-core processors, accelerators and other different architectures.

Figure 1 shows the overview of the mSPMD programming model. Each task in a workflow can be a distributed parallel program. A PGAS language called XcalableMP [5] has been supported to describe tasks in a workflow. To describe and manage dependency between tasks, we adopt YML[1] workflow development and execution environment.

Since tasks in a workflow can be executed in parallel in the mSPMD programming model, some "heavy" tasks can be executed in parallel to speed up the workflow. Another advantage of the mSPMD is that a huge distributed parallel program can be decomposed into several moderate sized sub-programs based on the workflow paradigm to avoid the communication overhead.

Current "BigData" issues in mSPMD programming model

One of important pieces that the mSPMD programming model is missing is an intelligent implementation of the data transfer method between tasks. As shown in Figure 2, our current implementation of the data transfer between tasks strongly relies on a network file system (NFS) and MPI-IO functions. After a task writes a data to a NFS, the other tasks which use the data are started and read the data from the NFS. The advantage of using NFS and MPI-IO are (1) portability (2) auto check-pointing and (3) ease of use for application developers since the MPI-IO function calls can be generated automatically based on XcalableMP declarations.

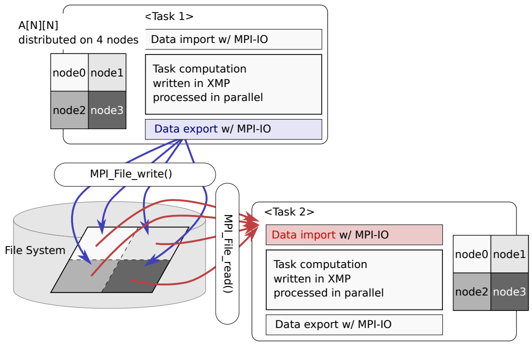


Figure 2: Data transfer between tasks

The disadvantages of using NFS are speed and performance instability. To overcome this, we are investigating the combination of file-IO and data servers[2]. Additionally, as future works, we will investigate advanced software and hardware infrastructures such as ADIOS library, data-compression hardware, burst buffer.

Toward integration of multi-SPMD programming model and advanced cyberinfrastructure platform

In addition to the advantages described above, the mSPMD can combine several parallel libraries and existing parallel programs easily to compose a complex application for a heterogeneous architecture.

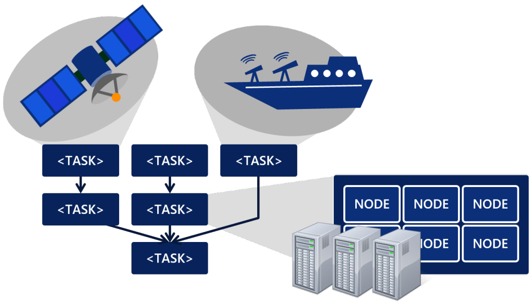


Figure 3: Integration of mSPMD and ACP

The mSPMD may provide a new method for data processing and data logistic among different systems. So far, we have focused on the data processing in an HPC cluster and the data dependencies between them. However, we consider the workflow paradigm is also useful to manage data dependencies in/from cyber-physical systems. An unified workflow approach to describe dependencies among data generations, data processings, HPC simulations to update data, etc... should be important to develop complicated applications. As future works, we will integrate the mSPMD programming model and ACP. As shown in Figure 3, our workflow paradigm will orchestrate data dependencies not only between traditional distributed parallel programs, but also from/to various sensing and processing devices.

References

[1] Olivier Delannoy, Nahid Emad, and Serge Petiton. Workflow global computing with yml. In *The 7th IEEE/ACM International Conference on Grid Computing*, pages 25–32, 2006.

[2] Thomas Dufaud, Miwako Tsuji, and Mitsuhisa Sato. Design of data management for multi-spmd workflow programming model. In *Proceedings of the 4th International Workshop on Extreme Scale Programming Models and Middleware, SC18*. ACM, 2018.

[3] Miwako Tsuji, Serge Petiton, and Mitsuhisa Sato. Fault tolerance features of a new multi-spmd programming/execution environment. In *Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware, SC15*, pages pp.20–27 doi:10.1145/2832241.2832243. ACM, 2015.

[4] Miwako Tsuji, Mitsuhisa Sato, Maxime Hugues, and Serge Petiton. Multiple-SPMD programming environment based on PGAS and workflow toward post-petascale computing. In *Proceedings of the 2013 International Conference on Parallel Processing (ICPP-2013)*, pages 480–485. IEEE, 2013.

[5] XcalableMP. http://www.xcalablemp.org/.

Autonomous Experimentation as a Paradigm for Materials Discovery

Kevin G. Yager, *Center for Functional Nanomaterials, Brookhaven National Laboratory*

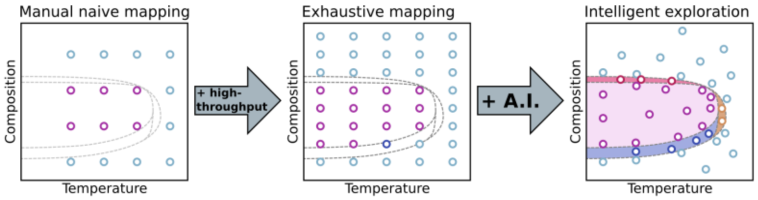
Concept

One of the grand challenges of twenty-first-century materials science is the rational design of new materials, where given a desired material functionality, the material structure is predicted; and for that particular structure, we can design appropriate constituents and assembly processes. To address this challenge, it is critical to understand the relationships between constituents, processing, and resultant materials structure and function. With the needs for material functionality becoming more diverse, stringent and sophisticated, the complexity of materials continues to increase. The relevant parameter space expands correspondingly, arising from both the multi-com ponent nature of functional materials and a multitude of processing conditions. All this implies that optimizing functionality requires strategic exploration of the vast parameter space that is associated with complex materials. To meet this challenge, the way we investigate materials needs to evolve, to become more efficient and intelligent.

An emerging paradigm to address this complexity is *autonomous experimentation*, wherein experimental synthesis and data collection are automated, and machine-learning algorithms are used to select experiments to conduct based on the evolving dataset. Implemented properly, these methods enable intelligent exploration of the enormous parameter spaces of materials science—that is, every single sample synthesis and measurement step is selected so as to yield maximal value (scientific insight) such that one can iterate towards a desired material property, or a answer a desired scientific question, as rapidly as possible.

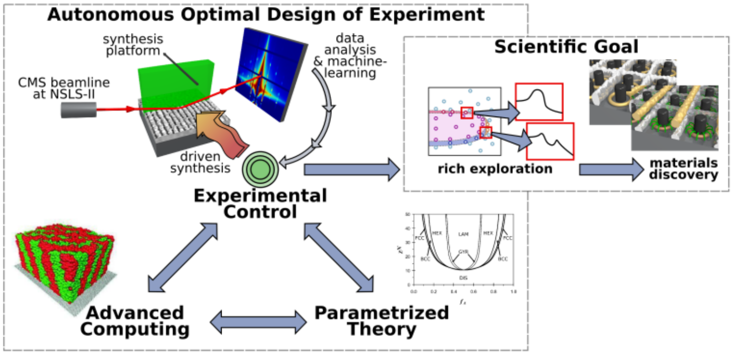
Recent work

Brookhaven National Laboratory has focused on developing these concepts and deploying them in the context of x-ray scattering, which is a powerful and rapid probe of material structure that can even measure materials in-situ (as they are being synthesized or processed). In this work, we have developed analytic[1, 2] and deep learning[3-7] methods for classifying or healing[8, 9] x-ray scattering datasets, and algorithms for optimal decision-making in an experimental context. We have demonstrated how physics-informed deep learning can deliver substantial performance improvements. For instance, we created a multi-channel convolution neural network in which initial data transformations are carefully selected by domain experts to highlight features of interest. For x-ray scattering data, decomposing the raw detector image into a matrix of Fourier-Bessel coefficients efficiently highlights symmetry information in the raw signal.



When these methods are combined and deployed at a synchrotron x-ray scattering beamline, they enable the instrument to autonomously explore material science problems. For instance, the beamline was able to efficiently image the structure of a nanoparticle thin film, where it first measured a low-resolution (coarse) image of the sample makeup, and then selected follow-up experiments so as to achieve higher imaging resolution in areas of interest (especially edges in the image). In another set of experiments, the beamline autonomously measured a large set of sample (using robotics to select samples from a queue) and was able not only to measure these samples in an efficient ordering, but also to suggest what follow-up samples should be synthesized next. Finally, this research program has demonstrated how these methods can be combined with combinatorial sample preparation. For instance, sample libraries can be synthesized by creating continuous gradients (of, e.g., composition); subsequent autonomous study allows these spaces to be mapped efficiently.

Future Needs



A key route towards improvement of the autonomous experimentation paradigm is to enable input of known material physics. This existing understanding both constrains the exploration problem, providing initial estimates of material behavior for guiding exploration, and also provides a framework into which newly-acquired data can be fit. Materials physics can be captured by appropriate simulation tools, such as molecular dynamics or field theories. However, materials models are typically computationally expensive, especially when they are attempting to capture the non-equilibrium aspects of realistic materials, and must thus explicitly simulation material evolution. Thus, a key challenge in the integration of materials models into autonomous workflows is to merge high-performance computing into a real-time experimental context.

Progress will require developments along three key vectors:

1. Software platforms that allow one to seamlessly integrate different computation inputs. In particular, the ability for experiments to select from a menu of decision-making algorithms, and to easily accept input from arbitrary materials modeling code.
2. New models of materials physics must be developed that provide reasonable predictive power while being computationally tractable. Machine-learning approximants can be trained based on the outputs of rigorous models. Ideally infrastructure would be developed to connect models of different fidelity, allowing both rapid input from approximants, as well as intermittent input from expensive models.
3. Infrastructure to enable timely (i.e. *during* experimentation) access to significant computing power, through connecting to existing HPC clusters, or by accessing novel distributed resources. Access must be rapid and elastic, able to handle the inconsistent and ‘bursty’ nature of experimental data collection, while also scaling favorable to handle the changing complexity of the underlying physics models.

Outlook

Autonomous experimentation has the potential to radical transform scientific study, by liberating human scientists to focus on high-level conceptual understanding, while having scientific instruments automatically handle sample management, processing, and high-speed decision-making. In the future, this paradigm must leverage large computational resources in order to provide real-time inputs from computationally-expensive materials modeling. New cyber infrastructure is critically required to enable timely and cost-effective access to elastic computing resources.

References

1. K. G. Yager and P. W. Majewski, *J. Appl. Crystallogr.*, 2014, **47**, 1855-1865.
2. J. R. Lhermitte, A. Stein, C. Tian, Y. Zhang, L. Wiegart, A. Fluerasu, O. Gang and K. G. Yager, *IUCrJ*, 2017, **4**, 604-613.
3. H. Huang, S. Yoo, K. Kaznatcheev, K. G. Yager, F. Lu, D. Yu, O. Gang, A. Fluerasu and H. Qin, presented in part at the Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Republic of Korea, 2014.
4. M. H. Kiapour, K. Yager, A. C. Berg and T. L. Berg, Applications of Computer Vision (WACV), 2014.
5. B. Wang, Z. Guan, S. Yao, H. Qin, M. H. Nguyen, K. G. Yager and D. Yu, Scientific Data Summit (NYSDS), New York, 2016.
6. B. Wang, K. G. Yager, D. Yu and M. H. Nguyen, Applications of Computer Vision (WACV), 2017.
7. Z. Guan, H. Qin, K. G. Yager, Y. Choo and D. Yu, *British Machine Vision Conference*, 2018, **0828**, 1-10.
8. J. Liu, J. Lhermitte, Y. Tian, Z. Zhang, D. Yu and K. G. Yager, *IUCrJ*, 2017, **4**, 455-465.
9. J. Liu and K. G. Yager, *IUCrJ*, 2018, **5**, 737-752.

BigDataBench: A Scalable and Unified Big Data and AI Benchmark Suite

Wanling Gao*†, Jianfeng Zhan*†, Lei Wang*, Chunjie Luo*†, Daoyi Zheng‡, Xu Wen*†, Rui Ren*†, Chen Zheng*,
Xiwen He§, Hainan Ye§, Haoning Tang¶, Zheng Cao‖, Shujie Zhang** and Jiahui Dai§
*Institute of Computing Technology, Chinese Academy of Sciences
†University of Chinese Academy of Sciences, China
‡Baidu
§Beijing Academy of Frontier Sciences and Technology
¶Tencent
‖Alibaba
**Huawei

***Abstract*—Several fundamental changes in technology indicate domain-specific hardware and software co-design is the only path left. In this context, architecture, system, data management, and machine learning communities pay greater attention to innovative big data and AI algorithms, architecture, and systems. Unfortunately, complexity, diversity, frequently-changed workloads, and rapid evolution of big data and AI systems raise great challenges. First, the traditional benchmarking methodology that creates a new benchmark or proxy for every possible workload is not scalable, or even impossible for Big Data and AI benchmarking. Second, it is prohibitively expensive to tailor the architecture to characteristics of one or more application or even a domain of applications.**

We consider each big data and AI workload as a pipeline of one or more classes of units of computation performed on different initial or intermediate data inputs, each class of which we call a data motif. We propose a scalable benchmarking methodology that uses the combination of one or more data motifs—to represent diversity of big data and AI workloads. Following this methodology, we present a unified big data and AI benchmark suite—BigDataBench 4.0, publicly available from http://prof.ict.ac.cn/BigDataBench. This unified benchmark suite sheds new light on domain-specific hardware and software co-design: tailoring the system and architecture to characteristics of the unified eight data motifs other than one or more application case by case.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The traditional benchmark methodology that creates a new benchmark or proxy for every possible workload is prohibitively costly and hence not scalable, or even impossible for Big Data and AI benchmarking. First, there are many classes of big data and AI applications. Even for Internet services, there are several important application domains, e.g., search engines, social networks, and e-commerce. The value of big data and AI also drives the emergence of innovative application domains. Meanwhile, data (sizes, types, sources, and patterns) have a great impact on workload behaviors and performance significantly [1], [2], so comprehensive and representative real-world data sets should be included.Second, at an earlier stage, it is usually difficult to justify porting a full-scale end-to-end Big data or AI application to a new computer system or architecture simply to obtain a benchmark number [3]; while at a later stage, kernels alone are insufficient to completely assess the performance potential of a new system or architecture on real-world data sets and applications [3]. Meanwhile, the benchmarks should be consistent across different communities for the co-design of software and hardware. Third, the correctness of results and performance figures must be easily verifiable [3]. To some extent, too complex workloads, i.e., full-scale end-to-end Big Data or AI applications raise difficulties in reproducibility and interpretability of performance data [1].

As modern big data and AI workloads are not only diverse, but also fast changing and expanding, it also raises great challenges in domain-specific hardware and software co-design. Even the agile hardware development methodology and tools are adopted [4], it is prohibitively expensive to tailor the architecture to characteristics of one or more application or even a domain of applications, and hence building domain-specific hardware and software systems case by case should be avoided.

This paper presents our joint research efforts on a scalable and unified Big Data and AI benchmarking suite with several industrial partners. On the basis of our previous work [1] that identifies eight data motifs—taking up most of the run time among a wide variety of big data and AI workloads, we propose a scalable benchmarking methodology that uses the combination of one or more data motifs—including *Matrix, Sampling, Transform, Graph, Logic, Set, Sort and Statistic computation* to represent diversity of big data and AI workloads. Our benchmark suite includes micro benchmarks, each of which is a single data motif, the component benchmarks, each of which consists of the combination of one or more data motifs with different weights in terms of runtime, and end-to-end application benchmarks, which are combinations of component benchmarks.

Following this methodology, we present a unified big data and AI benchmark suite—BigDataBench 4.0, publicly available from http://prof.ict.ac.cn/BigDataBench. BigDataBench 4.0 provides 13 representative real-world data sets and 47 big data and AI benchmarks of seven workload types: online service, offline analytics, graph analytics, AI, data warehouse, NoSQL, and streaming. Also, for each workload type, we provide diverse implementations using state-of-the-art and state-of-the-practise software stacks. Data varieties are considered with the whole spectrum of data types including structured, semi-structured, and unstructured data. Using real data sets as the seed, the data generators [5] are provided to generate the data with a specific scale.

II. BigDataBench 4.0: Big Data and AI Benchmark Suite

Circling around the data motifs identified from these application domains, we define the specifications of micro benchmarks—each of which is a single data motif, component benchmarks—each of which is a combination of data motifs with different weights, and application benchmarks—each of which represents an end-to-end applications. On the basis of the data motif-based benchmarking methodology, we make benchmark decisions and build BigDataBench 4.0. Please note that due to the space limitation, the detailed methodology and decisions about BigDataBench 4.0 are illustrated in our technical report [6].

1) Workloads Diversity: After investigating fundamental components in application domains, we provide a suite of micro benchmarks and component benchmarks. Totally, BigDataBench 4.0 provides 13 representative real-world data sets and 47 big data and AI benchmarks of seven workload types: online service, offline analytics, graph analytics, AI, data warehouse, NoSQL, and streaming.

For big data, we provide diverse workloads covering data mining, machine learning, natural language processing and computer vision techniques. For AI, we identify representative and widely used data motifs in a wide variety of deep learning networks (i.e. convolution, relu, sigmoid, tanh, fully connected, max/avg pooling, cosine/batch normalization and dropout) and then implement each single motif and motif combinations as micro benchmarks and component benchmarks. The AI component benchmarks include Alexnet [7], Googlenet [8], Resnet [9], Inception_Resnet V2 [10], VGG16 [11], DCGAN [12], WGAN [13], Seq2Seq [14] and Word2vec [15], which are important state-of-the-art networks in AI.

2) Representative Real-world Data Set: To cover a full spectrum of data characteristics, we collect 13 representative data sets, including different data sources (text, table, graph, and image), and data types of structured, un-structured, semi-structured. Further, big data generation tools are provided to suit for different cluster scales, including text, table, matrix and graph generators.

3) State-of-the-art Techniques: To perform apple-to-apple comparisons, we provide diverse implementations using the state-of-the-art techniques. For offline analytics, we provide Hadoop, Spark, Flink and MPI implementations. For graph analytics, we provide Hadoop, Spark GraphX, Flink Gelly and GraphLab implementations. For AI, we provide TensorFlow, Caffe and PyTorch implementations. For data warehouse, we provide Hive, Spark-SQL and Impala implementations. For NoSQL, we provide MongoDB and HBase implementations. For streaming, we provide Spark streaming and JStorm implementations.

References

[1] W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, F. Tang, B. Xie, C. Zheng, X. Wen, X. He, H. Ye, and R. Ren, "Data motifs: A lens towards fully understanding big data and ai workloads," *Parallel Architectures and Compilation Techniques (PACT), 2018 27th International Conference on*, 2018.

[2] B. Xie, J. Zhan, X. Liu, W. Gao, Z. Jia, X. He, and L. Zhang, "Cvr: Efficient vectorization of spmv on x86 processors," in *2018 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, 2018.

[3] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber *et al.*, "The nas parallel benchmarks," *The International Journal of Supercomputing Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[4] J. Hennessy and D. Patterson, "A new golden age for computer architecture: Domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development." 2018.

[5] Z. Ming, C. Luo, W. Gao, R. Han, Q. Yang, L. Wang, and J. Zhan, "Bdgs: A scalable big data generator suite in big data benchmarking," *arXiv preprint arXiv:1401.5465*, 2014.

[6] W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, R. Ren, C. Zheng, G. Lu, J. Li, Z. Cao *et al.*, "Bigdatabench: A dwarf-based big data and ai benchmark suite," *arXiv preprint arXiv:1802.08254*, 2018.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.