Distributed Exascale Computing: An AIMES Perspective

Shantenu Jha[1], Andre Merzky[1], Matteo Turilli[1], Daniel S. Katz[2] and Jon Weissman[3]

[1]*Rutgers University*, [2]*University of Chicago* [3]*University of Minnesota*

The Research Problem

Exascale Computing (EC) usually refers to the technology of using a single tightly-coupled computing resource to deliver performance in the exaflops region. Nonetheless, there are many reasons to believe that Distributed Exascale Computing (DEC) is an important complementary and synergistic pathway to deliver science at the exascale. We consider three reasons: (i) DEC is required by new science and to support novel usage modes, (ii) DEC supports the need for existing applications to scale in multiple dimensions, and (iii) DEC will further the democratization of extreme-scale science.

HPC/Supercomputing class applications as those under development for the Square Kilometer Array (SKA), the Large Synoptic Survey Telescope (LSST), or for the next generation of combustion, high-energy physics (e.g. Big Panda) and bioinformatics are becoming increasingly complex, multi-component, workflow based, and reliant upon multiple distributed data sources. In this scenario, the integration between single leadership machines and larger number of less powerful machines is necessary to promote efficient resource utilization. Such an integration runs in both directions: small resources need to be integrated with large resources and vice versa.

Furthermore, if EC is to be democratized to include the long-tail of science, it has to break free of “an anointed few (researchers) on the top-end machine” scenario that has hitherto dominated. An immediate consequence of this requirement is the need to support at exascale a broad range of applications over a broad set of resources with varying capabilities. This entails the federation, aggregation and integration of distributed resources at, across, and through multiple levels.

Our Viewpoint

The fundamental question we seek to address in the context of DEC is, “To distribute or not to distribute?”, which is associated with “how to distribute?” (so as to achieve EC). Although some of the research questions that emerge are seemingly familiar, the scale, the dynamism, the requirements of increasingly sophisticated applications and the capabilities of the underlying infrastructure frame the specific context.

We employ an abstractions-based and model-driven approach, whereby an abstraction of an element of the DEC landscape can be modeled in more ways than one, and each of the models in turn can have multiple implementations. Specifically, we utilize three types of model: (i) Conceptual, (ii) Analytical and, (iii) Prototypical. Conceptual models guide implementation designs; analytical models permit a numerical performance analysis mostly via simulations; and prototypical models are an often parametrizable implementation that facilitate experiments. All three model types enable reasoning, and will ultimately enable us to compare, contrast and predict different applications and execution plans to support DES.

Our research approach to DEC does not replace but complements traditional viewpoints of EC. We believe that implementing and modeling DEC poses challenges that are at different levels of granularity than in traditional EC. Fine-level architecture or energy considerations pertain to single, tightly-coupled computing resources. Conversely, coarse-grained properties need to be modeled when considering dynamic infrastructure composition and adaptive runtime execution of applications.

Research Approach

Our approach to conceptualize DEC has three vertices: Applications (A), Infrastructure (I), Federation (F) of infrastructures. Here we provide initial conceptual models of A, I, and F, and we assume Application Execution Plans (EP) as the time-dependent composition (or decompositions) and placement of applications as a function of the (dynamic) federated infrastructure available to it.

A: Applications Models*

At a conceptual level, applications can be modeled in different but inter-related ways: (i) as a definition of a workload; (ii) as a set of semantic components representing a workload; (iii) as a sequence or composition of infrastructure capabilities executing a workload. As our goal is to develop an integrated model of DEC, we focus on the third way.

Our model of an application, labeled A*, will enable reasoning about decomposition of a workload into tasks. Decomposed workloads imply a need to coordinate the dependencies and concurrencies of their tasks, which in turn implies the need for communication and data exchange.

A* will support the ability to distinguish execution planning requirements for different federations of infrastructures. Consider, for example, a master-worker based execution plan which distributes a loosely-coupled, fine-grained workload decomposition. The communication capabilities for such a plan will differ when distributed on a small federation of a large, tightly-coupled compute resources (such as federated DOE leadership class machines); or on a large federation of small, loosely coupled compute resources (as exemplified by BOINC-based resources).

Further refinement of the conceptual application model will add qualitative and quantitative properties to support the analysis and simulation of an application’s runtime behavior and performance. In turn, such analytical modeling of applications will also provide stringent requirements on the properties of the federation and infrastructure capabilities. For example, runtime requirements for the execution of a well defined application workload of, say, 10^{19} flops of coupled tasks with infrequent exchange of large data sets will inform the required infrastructure capabilities, and also the performance properties of federations and infrastructures required to support that application workload.

I: Infrastructure Models*

Our abstraction of infrastructure is based upon a set of capabilities exposed by that infrastructure, *without* regards to its internal properties, or specific mechanisms that are used to provide these capabilities. For example, rather than suggesting that a specific machine has 10^6 cores, we model a machine to say that it supports the execution of 100 tasks each of 10 hours duration and 10^5 flops within a duration of 100 hours.

Workload management capabilities need to be flexible enough to satisfy the requirements posed by multiple types of applications. Analogously, introspective capabilities intended as those functionalities that allows for an application to gather data about the states of the infrastructure where it is running, need to also expose enough information for the qualitative and quantitative assessment of workloads execution. The infrastructure model will enable workload management to be based upon the concept of capabilities.

F: Federation Models*

Existing and proposed infrastructures show a vast heterogeneity of capabilities, and an even larger heterogeneity in implementations. At increasing scale, this makes standardization and interoperation at infrastructure level difficult. This presents the need for a type of federation which combines and exposes diverse infrastructure capabilities to applications in a consistent and scalable way.

We define a ‘federation of infrastructures’ as a set of services that allow for the dynamic creation of workload management overlays on independent infrastructures that expose heterogeneous capabilities.

We minimize the requirements on the federated infrastructures by focusing not on modeling yet another middleware but on abstracting and composing the capabilities they expose into a coherent ensemble. Furthermore, we conceptualize and will eventually prototype a federation capable of supporting multiple execution strategies for different types of scientific workloads thanks to an inherently flexible architecture.

Our conceptual and eventual analytical model of federation leverages the three core notions of ‘service’, ‘composition’ and ‘overlay’. We will model the capabilities exposed by the federation as self-contained, independent services with well-defined interfaces and uniform communication protocols. Services will be composed as needed by the application layer as no composition pattern will be imposed by the architectural design of the federation. The heterogeneity of the underlying infrastructures will be addressed by means of dedicated connectors while ‘resource containers’, inspired to the vastly successful pilot abstractions, will be the base for the creation of workload management overlays.

Our modeling of a federation of distributed resources shares multiple elements with the original vision for Grid Computing. Nonetheless, the two approaches significantly diverge when considering that no specific middleware is required, rigid service interdependencies are avoided and the workload management is shifted from the domain of the middleware to that of the application.

Assessment

Challenges Addressed: This position paper outlines some initial ideas towards integrated modeling of DEC, and a research agenda that will be required en route. Once completed, our conceptual, analytical and prototypical models will offer qualitative and quantitative elements for simulating and predicting the execution of a range of scientific workloads in a distributed exascale environment.

The pathway to DEC entails both conceptual and implementation complexity. Whereas the focus is on addressing the former, we believe modeling may help manage implementation complexity by providing abstractions and conceptual models to develop effective and scalable tools and techniques. Also, as we refine and formulate specific models of A, I and F, we will address the many ways to compose (or decompose) applications and federate infrastructures, and in turn guiding their interfaces and software design.

Uniquness: Ultimately, exascale capabilities will be achieved by scaling-up leadership class machines, scaling-out DoE infrastructure such as the Open Science Grid. Our modeling activity is motivated by the critical need to seamless integrate these infrastructure from the application and resource management perspective. Although, our models are built around the properties of current production infrastructures, however, monolithic and idealized models are avoided in favor of a careful separation of concerns among A*, I* and F*.

Novelty: We argue that a novel model of federation is an essential component for future DEC. Departing from an established tradition, the model of federation we propose does not require specific middleware and interfaces at the infrastructure level. Overlays and resource containers make workload management accessible to the application domain by simplifying the process and abstracting away the differences among infrastructures; this has far-reaching consequences when qualifying and quantifying the execution of distributed scientific workloads.

Maturity: We have completed a conceptual model (called P*) of pilot abstractions for resource management. We are making advances in developing a model for affinity between distributed compute, data and computational resources; our model will guide reasoning to determine effective workload decomposition, distribution and placement. We have formalized many elements of A* in order to understand synthetic and data-intensive application, as well as the creation of a prototype for the distributed execution of synthetic workloads across heterogeneous, production infrastructures. Furthermore, a conceptual model of this type of federation is well underway.

Effort: There remain fundamental challenges that need addressing. For example, a large set of assumptions need to be explicitly worked through for each of the models before they can be integrated into an end-to-end capability. We have sketched what is arguably an ambitious three-to-five year research agenda.

Acknowledgements

This work is funded by Department of Energy Award (ASCR) DE-FG02-12ER26115. We acknowledge important and useful discussions with Mark Santcroos (Rutgers)