Towards Distributed E* Computing

Shantenu Jha, Andre Merzky, Matteo Turilli

Rutgers University, Piscataway, NJ 08854, USA

Exascale computing 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 and synergistic pathway to deliver science at the exascale. We consider three reasons: (i) DEC is required by new science and 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.

EC presents unprecedented requirements both in terms of the flexibility and functionality of infrastructure, and application adaptivity and execution requirements. Functional requirements offered by the infrastructure include but are not limited to as addressing large degrees of distribution, and interoperability across heterogeneity. Consider for example the Square Kilometer Array (SKA) or the Large Synoptic Survey Telescope (LSST), with their large and continuous data output requiring to be stored and processed on global scale, or the next generation combustion with requirements of computational steering at scale to support enhanced scientific knowledge and discovery.

Traditional HPC/supercomputing class applications are becoming increasingly complex, multi-component and workflow based, as well as reliant upon multiple distributed data sources. In such cases, off-loading suitable components from the single leadership machine to a larger number but less powerful machines is an important requirement of efficient resource utilization. As demonstrated by the next-generation of high-energy physics (e.g. Big Panda project) and bioinformatics applications, there is a critical need to execute traditionally "small and distributed" workloads across distributed federated resources, which include leadership class machines, such as Titan. Thus the arrow of federation runs in both directions: federate small with large resources, and large with small resources.

If extreme scale computing is to be democratized to include the long-tail of science, the community has to break free of the model of "an anointed few (researchers) on the top-end machine" that has hitherto dominated. An immediate consequence of this requirement, is the need to support 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/thro!ugh multiple levels.

Towards Distributed Exascale Computing: Our Viewpoint

It is critical to establish that our viewpoint of DEC does not replace but complements traditional viewpoints of Exascale Computing, The fundamental question we seek to shed light on is, "To distribute or not to distribute?", which is associated with the question of "how to distribute?". And, although some of the research questions that emerge are seemingly familiar, the scale, the dynamism, increasingly sophisticated application requirements and the capabilities of the underlying infrastructure frame the specific context. We employ an abstractions-based and model-driven approach, whereby abstractions can be modeled (in more ways than one), and each of the models in turn can be implemented in multiple ways.

Before we layout our viewpoint, it is worth mentioning that there are three different types of models that our viewpoint will utilize: (i) Conceptual, (ii) Analytical and, (iii) Prototypical. All three model types enable reasoning; conceptual models guide implementation designs, but a conceptual model does not lend itself to any performance insight or analysis. An analytical model in contrast does permit a numerical performance analysis (often via simulations), while a prototypical model is an (mostly parametrizable) implementation facilitating experiments. If we assume the existence of an axis, one direction of which points towards specificity (and thus performance prediction and measurements), then a prototypical model is the most specific, whilst a conceptual model is the most non-specific.

In our viewpoint there are four vertices – Applications (A), Infrastructure (I), Federation (F) of infrastructure, and application Execution Plan (EP), that need to be considered to conceptualize DEC. We think the challenges inherent, both in implementing as well as modeling DEC, are at different levels of granularity than in traditional exascale computing. For example, we do not need specific fine-level architecture or energy considerations (we will eventually use them); we need coarser-grained and general models of applications and infrastructure. Given our belief of how DEC will be used, we posit a fundamental role for federation of infrastructure and thereby models of federation. Dynamic fluctuation in properties (mostly performance) will likely play a greater role at extreme-scales, thus we must be able to model time-dependent capabilities. Coupling the two, a requirement of modeling DEC is the need to include dynamic infrastructure composition and adaptive runtime execution of applications.

In this white paper, we provide initial conceptual models of A, I, and F. Where possible, we "sketch" analytical and prototypical models, that will ultimately enable us to compare, contrast and predict different applications and execution on flexibly federated infrastructure to support DES. This white paper outlines some initial ideas and a research agenda that paves a pathway to DEC. There remain a multitude of fundamental challenges that need addressing, understanding, as

well as a large set of assumptions need to be established on a firm footing.

A: A Conceptual Model for Applications*

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 approach, by expressing application requirements so that they can be mapped to infrastructure capabilities with support from execution plans. Admittedly, in order to do so, application models require models of similar granularity and specificity on Infrastructure level and Federation level; we discuss I* and F* as candidate models below.

Our model of an application, labelled A*, will model workload decompositions. Decomposed workloads imply a need to coordinate the dependencies and concurrencies of sub-workloads (tasks), which in turn implies the need for communication and data exchange. A* will support the ability to distinguish execution planning requiremetns for different infrastructure federations. For example, a master-worker based execution plan which distributes a loosely-coupled fine-grained workload decomposition (application) will require different communication capabilities on a small federation of a large, tightly coupled compute resources (such as federated DOE leadership class machines), compared to those 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 specific (qualitative and quantitative) properties to the model components, to support the analysis and simulation of an application's runtime behavior and performance. 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 sub-problems 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: A Conceptual Model for Infrastructure*

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, support the execution of 100 tasks each of 10 hours duration and 10^5 flops within a duration of 100 hours.

Particularly relevant for our infrastructure modeling are capabilities related to workload management and introspection – they need to satisfy the capability requirements posed by multiple types of applications but need to also expose enough information for the quantitative assessment of workloads execution – which are part of the A* and F* models.

F: A Conceptual Model of Federation*

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 on 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 aim at minimizing the requirements on the federated infrastructures. Infrastructures are assumed to be free to choose how to expose their capabilities. Therefore, we focus not on modeling yet another middleware that would need to be deployed on each infrastructures but on abstracting and composing the capabilities exposed by the infrastructures into a coherent ensemble.

Furthermore, our goal is also to model a federation capable of supporting multiple execution strategies for different types of scientific workloads. Accordingly, we conceptualize and will eventually prototype a federation with an inherently flexible architecture, something that goes well beyond the limits of tailored user interfaces.

Our conceptual model of federation leverages the three core notions of 'service', 'composition' and 'overlay'. We model the capabilities exposed by the federation as services. Services can be composed as needed by the application layer as no specific composition pattern is imposed by the architectural design of the federation. Specific services are used to create ad hoc and ephemeral workload management overlays that allows for diversified execution strategies on the federated infrastructures.

Moving towards an eventual analytical model, our approach to a federation of infrastructures will require a design grounded on well-defined and standard- based interfaces, uniform communication protocols, a connector-based architecture and an application-based workload manager. Services will be designed as self-contained, independent units and interfaces and uniform communication protocols will be used to expose their capabilities. The heterogeneity of the underlying infrastructures will be addressed by means of dedicated connectors while the 'resource containers', inspired to the vastly successful pilot abstractions, will the base for the creation of workload management overlays.

The modeling of a federation of distributed resources based on services and standard-based interfaces shares multiple elements with the original vision for Grid Computing. Nonetheless, they 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.

Execution Planning

We define execution-planning as the time-dependent composition (or decompositions) and placement of applications as a function of the (dynamic) federated infrastructure available to it. Execution planning provides the conceptual basis for distinguishing the execution of different workloads on the same infrastructure, or the same workload on different infrastructure. Akin to dynamical aspects of infrastructure federation, execution plans do not need to be static through the lifetime of an application.

Given the range of application types and characteristics that need to be modeled, the space of Execution Plans is vast and oftentimes more nuanced than just HPC or HTC — two commonly assumed execution plans. The distinction between HTC and HPC is at best artificial and contextual. As a simple, but effective illustration, we note that depending upon "urgency" and other considerations, an infrastructure that is otherwise used to support HPC workloads can be made to support HTC execution plans.

Advantages and novelty of our Simulation and Modeling Approach

Our work represents initial and arguably novel step on the path towards an integrated model of distributed exascale computing. Once completed, our conceptual, analytical and prototypical models will offer qualitative and quantitative elements for simulating and predicting the execution of scientific workloads in a distributed exascale environment.

Our modeling activity is shaped 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*. We have completed a conceptual model of pilot abstractions called P*, and are making advances in developing and using a model for affinity to determine the optimal workload decomposition, distribution and placement. Finally, we have formalized many elements of A* in order to understand synthetic and data-intensive application.

We argue that a novel model of federation is an essential component to achieve DES. Departing from an established tradition, the model of federation we propose does does not require specific middleware and interfaces at the infrastructure level. The conceptual model of this type of federation is well underway alongside the creation of a prototype for the distributed execution of synthetic workloads across heterogeneous, production infrastructural.

Overall, the entirety of our modeling effort is centered around the application domain. Overlays and resource containers make workload management accessible to the application domain by simplifying the process and abstracting away the differences among infrastructures. This shift has far-reaching consequences when considering the very same possibility to qualify and quantify the execution of distributed scientific workloads at exascale.

The pathway to DEC entails both conceptual complexity 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.

Acknowledgements

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