Federated Computing for the Masses – Aggregating Resources to Tackle Large-scale Engineering Problems

Javier Diaz-Montes
Rutgers University
Piscataway, NJ 08854, USA
javidiaz@rdi2.rutgers.edu

Yu Xie
Iowa State University
Ames, IA 50011, USA
yuxie@iastate.edu

Ivan Rodero
Rutgers University
Piscataway, NJ 08854, USA
irodero@rutgers.edu

Jaroslaw Zola
Rutgers University
Piscataway, NJ 08854, USA
jaroslaw.zola@rutgers.edu

Baskar Ganapathysubramanian
Iowa State University
Ames, IA 50011, USA
baskarg@iastate.edu

Manish Parashar
Rutgers University
Piscataway, NJ 08854, USA
parashar@rutgers.edu

ABSTRACT

The ever-growing complexity of scientific and engineering problems continues to pose new requirements and challenges for computing and data management. The analysis of high-dimensional parameter spaces, uncertainty quantification by stochastic sampling, or statistical significance assessment through resampling, are just few examples of a broad class of problems that are becoming increasingly important in a wide range of application domains. These "ensemble" (also termed as many task computing or MTC) applications consist of a set of heterogeneous computationally intensive, and independent or loosely coupled tasks, and can easily consume millions of core-hours on any state-of-the-art HPC resource. While many of these problems are conveniently parallel, their collective complexity exceeds computational time and throughput that average user can obtain from a single computational center. For instance, the fluid flow problem we consider comprises more than ten thousand MPI tasks, and would require approximately 1.5 million core-hours to solve on the *Stampede* cluster at TACC – one of the most powerful machines within XSEDE [3]. Although XSEDE allocations of that size are not uncommon, the heavy utilization of *Stampede*, and its typical queue waiting times make it virtually impossible to execute that number of tasks within an acceptable time limit. The problem becomes even more complex if we take into account that individual tasks are heterogeneous, and add in the possibility of failures that are not uncommon in large-scale multi-user systems.

The above constraints are not unique to one particular problem or a system. Rather, they represent common obstacles that can limit the scale of problems that can be considered by an ordinary researcher on a single, even very powerful, system. Importantly, this trend continues and one can only expect that more and more users will require computational throughput that cannot be delivered just by one resource. In order to overcome these limitations two important questions have to be addressed. First, how to empower a researcher with computational capability that is compatible to what currently is reserved for the "elite" problems. Second, how to deliver this capability in a user-centered way. In this work, we argue that both these questions can be answered by implementing a federation model in which a user, without any special privileges, can seamlessly aggregate multiple, globally distributed and heterogeneous HPC resources exploiting their intrinsic capabilities. Our focus is on empowering average user with aggregated computational capabilities typically reserved for selected high-profile problems. To achieve this, we propose to aggregate heterogeneous HPC resources in the spirit of how volunteer computing assembles desktop computers. Specifically, we describe a model of computational federation that i) is extremely easy to deploy and offers an intuitive API to meet expectations and needs of average user; ii) encapsulates cloud-like capabilities, e.g. on-demand resource provisioning, elasticity and resilience, to provide sustainable computational throughput; iii) provides strong fault-tolerance guarantees through constant monitoring of tasks and resources; iv) bridges multiple, highly heterogeneous resources, e.g. servers, clusters, supercomputers and clouds, to effectively exploit their intrinsic capabilities; v) transparently leverages the security mechanisms provided by each system.

In our model, the underlying infrastructure is presented as a single pool of resources regardless of their physical location. The design is based on four layers, where the bottom layer is responsible for the interaction with physical resources, and the top layer contains the actual user application. The appropriate provisioning of resources in accordance with user provided policies is realized by the cross-layer autonomic manager. In this model, the security is built on top of the underlying infrastructure, which can be X.509 certificates or public/private key authentication. The schematic representation of the design is presented in Figure 1.

To demonstrate potential of the resulting federated infrastructure to address the computational requirements of real-world large-scale computational engineering problems, we

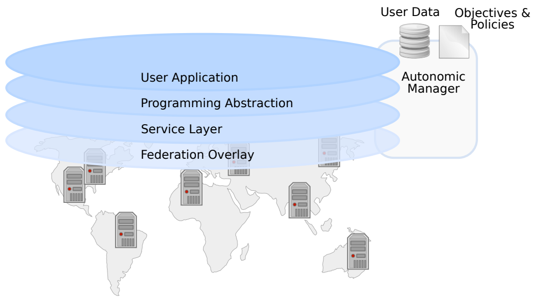

Figure 1: Multi-layer design of the proposed federation model. Here, the autonomic manager is a cross-layer component that based on user data and policies provisions appropriate resources.

performed the analysis of a high-dimensional parameter space in the fluid flow problem. Our focus on the fluid flow problem has been motivated by its great practical importance. The ability to control fluid streams at microscale has significant applications in many domains, including biological processing, guiding chemical reactions, and creating structured materials. Two of the authors, henceforth referred to as the end-user, are part of a team that recently discovered that placing pillars of different dimensions, and at different offsets, allows "sculpting" the fluid flow in microchannels [1]. The design and placement of sequences of pillars allows a phenomenal degree of flexibility to program the flow for various bio-medical and manufacturing applications. However, to achieve such a control it is necessary to understand how flow is affected by different input parameters. For this purpose the end-user has developed a parallel, finite element and MPI-based Navier-Stokes equation solver, which can be used to simulate flows in a microchannel with an embedded pillar obstacle. Here, the microchannel with the pillar is a building block that implements a fluid transformation. For a given combination of microchannel height, pillar location and diameter, and Reynolds number (4 variables), the solver captures both qualitative and quantitative characteristics of flow (see Figure 2).

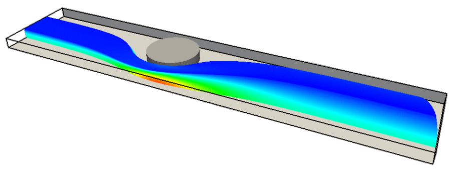

Figure 2: Example flow in a microchannel with a pillar. Four variables characterize the simulation: channel height, pillar location, pillar diameter, and Reynolds number. Please view in color.

The problem is challenging for several reasons. The search space consists of tens of thousands of points, and an individual simulation may take hundreds of core-hours, even when executed on a state-of-the-art HPC cluster. For example, the specific instance we consider requires 12,400 simulations. The individual tasks, although independent, are highly heterogeneous and their cost of execution is very difficult to estimate *a priori*, owing to varying resolution and mesh density required for different configurations. In our case, the cost may range from 100 core-hours to 100,000 core-hours per task executed on the IBM Blue Gene/P. Consequently, scheduling and coordination of the execution cannot be performed manually, and a single system cannot support it. Finally, because the non-linear solver is iterative, it may fail to converge for some combinations of input parameters, in which case fault-tolerance mechanisms should be engaged. The above properties make the problem impossible for the end-user to solve using the standard computational resources (e.g. computational allocation from XSEDE).

To perform this experiment, we implemented our federation model using the CometCloud framework [2]. Here, CometCloud provides a basic functionality such as autonomic capabilities, fault tolerance mechanisms, and the transparent access to heterogeneous resources. We integrated our scientific application with the framework using the master/worker programming model provided by CometCloud. The master component takes care of generating tasks, collecting results, verifying that all tasks executed properly, keeping log of the execution and resubmitting failed tasks. Meanwhile, workers sole responsibility is to execute tasks pulled from the task space.

To execute our case-study we federated 10 different HPC resources provided by six institutions from three countries. The experiment consumed over 2.5 million core-hours, and provided the most comprehensive data to-date on the effect of pillars on microfluid channel flow. Thanks to the library of flow configurations that we generated, we can now investigate the inverse problem and, for example, ask questions about the optimal pillar arrangement to achieve a desired flow output. The implications of such capability are far-reaching, with potential applications in medical diagnostics and smart materials engineering.

The success of this experiment clearly demonstrates the capability, feasibility, and advantages of a user-centered computational federation based on CometCloud. In the experiment, a regular user was able to solve a large scale computational engineering problem, within just two weeks. More importantly, this result was achieved in a few simple steps executed completely in a user-space. The user was required to provide kernels executed by the master and workers, and gained access to a unified and fault-tolerant computational platform with cloud-like capabilities that was able to sustain the computational throughput required to solve the problem. This result is of great relevance if we consider the growing complexity of computational engineering problems, that very often outpace the increase in performance of individual HPC resources.

References

[1] H. Amini, E. Sollier, M. Masaeli, et al. Engineering fluid flow using sequenced microstructures. *Nature Communications*, 2013.

[2] CometCloud Project. http://www.cometcloud.org/.

[3] XSEDE Project. https://www.xsede.org/.