Exploring the Use of Elastic Resource Federations for Enabling Large-scale Scientific Workflows

Javier Diaz-Montes
Rutgers Discovery Informatics Institute (RDI²)
javier.diazmontes@gmail.com

Yu Xie, Ivan Rodero, Jaroslaw Zola, Baskar Ganapathysubramanian, and Manish Parashar
Motivation

• A large class of problems in science and engineering:
  – Fit the MTC paradigm
  – Exceed computational time and throughput that an average user can get from a single data center
  – Have dynamic resource requirements

• Need to explore new federation models that can dynamically shape the infrastructure to meet requirements

• Software defined / User programmable federated infrastructure for the masses
Use Case: Fluid Flow in Microchannel

• Controlling fluid streams at microscale is of great importance for biological processing, creating structured materials, etc.
• Placing pillars of different dimensions, and at different offsets, allows “sculpting” the fluid flow in microchannels
• Four parameters affect the flow:
  – Microchannel height
  – Pillar location
  – Pillar diameter
  – Reynolds number

• Each point in the parameter space represents simulation using the Navier-Stokes equation (MPI-based software)
• Highly heterogeneous and computational cost is hard to predict a priori
Federation Overview

- User defined, dynamically created at runtime
- Sites can join and leave at any point
- Sites talk with each others to:
  - Identify themselves
  - Verify identity (x.509, public/private key,...)
  - Advertise their own resources and capabilities
  - Discover available resources
- Users can access the federation from any site
Federation Architecture

• Dynamic Federation built on top of CometCloud framework
• Federation is coordinated using Comet spaces at two levels
• Management space
  – Orchestrates resources in the federation
  – Interchange operational messages
• Shared execution spaces
  – Created on demand by agents
  – Provision local resources and cloudburst to public clouds or external HPC systems
Federation Site

- Gateway to the federation
- Autonomic Manager and Resource Manager
- Transparent coordination between site and execution spaces based on programming model
Experimental Setup

- 10 different HPC resources from 3 countries dynamically (and opportunistically) federated using a CometCloud-based infrastructure
- Experiment completely performed within user space (SSH)
- Fault-tolerance mechanisms to handle failed tasks
- Global view of the parameter space requires 12,400 simulations (three categories)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Cores</th>
<th>Network</th>
<th>Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excalibur</td>
<td>IBM BG/P</td>
<td>8,192</td>
<td>BG/P</td>
<td>LoadLeveler</td>
</tr>
<tr>
<td>Snake</td>
<td>Linux SMP</td>
<td>64</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Stampede</td>
<td>iDataPlex</td>
<td>1,024</td>
<td>IB</td>
<td>SLURM</td>
</tr>
<tr>
<td>Lonestar</td>
<td>iDataPlex</td>
<td>480</td>
<td>IB</td>
<td>SGE</td>
</tr>
<tr>
<td>Hotel</td>
<td>iDataPlex</td>
<td>256</td>
<td>IB</td>
<td>Torque</td>
</tr>
<tr>
<td>India</td>
<td>iDataPlex</td>
<td>256</td>
<td>IB</td>
<td>Torque</td>
</tr>
<tr>
<td>Sierra</td>
<td>iDataPlex</td>
<td>256</td>
<td>IB</td>
<td>Torque</td>
</tr>
<tr>
<td>Carver</td>
<td>iDataPlex</td>
<td>512</td>
<td>IB</td>
<td>Torque</td>
</tr>
<tr>
<td>Hermes</td>
<td>Beowulf</td>
<td>256</td>
<td>10 GbE</td>
<td>SGE</td>
</tr>
<tr>
<td>Libra</td>
<td>Beowulf</td>
<td>128</td>
<td>1 GbE</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Note: † – peak number of cores available to the experiment.
Summary of the Experiment

- 16 days, 12 hours, 59 minutes and 28 seconds of continuous execution
- 12,845 tasks processed (445 extra), 2,897,390 CPU-hours consumed, 400 GB of data generated
Throughput and Queue Waiting Time
Science Outcomes

• The most comprehensive data on the effect of pillars on microfluid channel flow
• Library of flow transformations
• Arranging pillars is possible to perform basic flow transformation
• What is the optimal pillar arrangement to achieve a desired flow output?
• Useful for medical diagnostics, smart materials engineering, and guiding chemical reactions
Lessons Learned: Federation Properties

• **Deployability**: Must be easy to deploy by a regular user without special privileges

• **Scalability and extended capacity**: Scale across geographically distributed resources

• **Elasticity**: Ability to scale up, down or out on-demand

• **Interoperability**: Interact with heterogeneous resources

• **Self-discovery**: Discovery mechanisms to provide a realistic view of the federation
Conclusions

• We focused on a class of MTC problems with dynamic and non-trivial computational requirements

• Demonstrated feasibility and capability of an elastic, dynamically federated infrastructure

• User-oriented / software defined approach - empower user with a simple mechanism to quickly federate resources

• Offer programming abstractions that allow users to build their federation
Acknowledgments
Thank You!

Javier Diaz-Montes, Ph.D.
Research Associate, Dept. of Electrical & Computer Engr.
Rutgers Discovery Informatics Institute (RDI²)
Rutgers, The State University of New Jersey

Email: javidiaz@rdi2.rutgers.edu
WWW: https://sites.google.com/site/javierdiazmontes/
CometCloud: http://cometcloud.org