Accelerating Parameter Sweep Workflows by Utilizing Ad-hoc Network Computing Resources: an Ecological Example

Jianwu Wang¹, Ilkay Altintas¹, Parviez R. Hosseini², Derik Barseghian², Daniel Crawl¹, Chad Berkley³, Matthew B. Jones³

¹ San Diego Supercomputer Center, UCSD, U.S.A.
² The Consortium for Conservation Medicine, Wildlife Trust, U.S.A.
³ National Center for Ecological Analysis and Synthesis, UCSB, U.S.A.
Outline

• Introduction
• Theoretical Ecology Use Case
• Background
  – Kepler
  – Master-Slave Architecture
• Our Approach
  – Distributed Composite Actor
  – Provenance Collection
• Results
• Conclusion and Future Work
Introduction

• Many scientific computing problems have linear or greater time complexity based on parameter configuration ranges.

• Domain scientists should be able to easily leverage distributed computing resources with little knowledge of the underlying techniques.

• We will discuss a distributed execution framework, called Master-Slave Distribution, to distribute sub-workflows to ad-hoc network computing resources.
Characteristics of the use case

- **Parameter Sweep**: independent multiple execution, i.e., “embarrassingly parallel problems”
- Smooth Transition of Computation Environments
- Partial Workflow Distribution
- Provenance Collection
Background - Kepler

• **Actor-oriented Modeling**
  - All these actors inherit the same interfaces, such as `prefire()`, `fire()` and `postfire()`

• **Model of Computation**
  - Synchronous Data Flow (SDF) director: actors execute sequentially
  - Process Network (PN) director: each actor has its own execution thread and execute in parallel
Background - Conceptual Architecture for Workflow Distributed Execution

A High-Level Distributed Execution Framework for Scientific Workflows
Our Approach - Distributed Composite Actor

- As the role of Master, each token received by this Actor is distributed to a Slave node, executed, and the results returned.
- Different behavior with different computation models
Our Approach - Provenance Collection

- By collecting workflow structure and executions, our provenance framework make it easier for users to track data files for large parameter sweeps.
- It can be configured to support centralized or decentralized provenance information recording.
Results - Workflow
Results - Usability

- Users use the DistributedCompositeActor just like the common composite actor

- Interaction for execution environment transition
## Results - Experiment

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Execution Time (minutes)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>b=&lt;0.1, 0.2, 0.3, 0.4&gt;, d=&lt;2.5, 3.5&gt;, X=3, Y=3, E=4</td>
<td>SDF locally</td>
<td>PN locally</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.35</td>
</tr>
<tr>
<td>b=&lt;0.1, 0.2, 0.3, 0.4&gt;, d=&lt;2.5, 3.5&gt;, X=8, Y=8, E=30</td>
<td>32.21</td>
<td>32.24</td>
</tr>
<tr>
<td>b=&lt;0.1, 0.2, 0.3, 0.4&gt;, d=&lt;2.5, 3.5&gt;, X=16, Y=16, E=30</td>
<td>502.2</td>
<td>510</td>
</tr>
</tbody>
</table>

### Testbed Constitution

<table>
<thead>
<tr>
<th></th>
<th>OS</th>
<th>Memory</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notebook</td>
<td>Window XP</td>
<td>2 GB</td>
<td>2.00 GHz Duo Core</td>
</tr>
<tr>
<td>Desktop</td>
<td>Mac OS X</td>
<td>2 GB</td>
<td>2.80 GHz Duo Core</td>
</tr>
</tbody>
</table>

![Configuration Graph](image)
Conclusion and Future Work

• A distributed execution framework in the Kepler
  - Distribute sub-workflows to ad-hoc network computing resources
  - Applicable to parameter sweep applications to realize parallel independent execution

• Future Work
  - Generalize for Cluster, Grid, and Cloud platforms.
  - Categorize different distributed approaches in Kepler to match different requirements
• **Thanks!**

• **For More Information:**
  - Distributed Execution Interest Group of Kepler: [https://dev.kepler-project.org/developers/interest-groups/distributed](https://dev.kepler-project.org/developers/interest-groups/distributed)
  - Contact: jianwu@sdsu.edu