SWSA: Domain-Specific Software Architecture for Workflow-Based Science Data Systems

David Woollard§+
Neno Medvidovic§

University of Southern California§
Computer Science Department
Software Architecture Group

California Institute of Technology§+
NASA Jet Propulsion Laboratory
Data Management Group
Science Data System In Context

Science Data System

Space-based Instruments

Data Acquisition

Mission Operations

Ground Sensors

Scientific Community

Science Team

High Performance Computing

Workflow

Scientific Community

Data Analysis

Archiving

Mission Operations

Data Analysis

Archiving

Workflow
“in silico” Science In Context

- Prototypes of algorithms
- Developed by scientists
- Characterized by dynamism
- Lone Research Paradigm [Kepner03]
“in silico” Science In Context

- Activities include:
  - Scaling experiments
  - Replication
  - Calibration/Validation
    i.e., Science Operations

- Engineering concerns are introduced:
  - Throughput
  - Traceability
  - Reliability
    i.e., QoS
“in silico” Science In Context

- Publishing the results:
  - Papers
  - Peer Review
  - Federated Holdings (i.e., data grids)
Research Problem

• How can we, as software engineers and architects, support scientists in the production and distribution activities associated with Science Data Systems?

• Insights:
  – State of the practice scientific computing development methodologies lack support for engineering concerns
  – Scientists are not motivated to be good software engineers
  – Despite a wealth of computational resources (clusters, supercomputers, clouds) and software infrastructure (grids, workflow systems), scientists are not using them.
Our Thesis

• Workflow systems and grid provide processing model dynamism and hardware virtualization
• Much of the desired engineering functionality can be found in *Grid Services*:
  – Data Repositories
  – Provenance Repositories
  – Fault Registries

A Domain-Specific Software Architecture can provide a *separation of concerns* between the scientist with deep domain knowledge and the engineer’s interfaces to these Grid Services.
Workflow Systems

- Workflows are dynamic models, interpreted at \textit{run-time}, that capture both control flow and data flow
- Models orchestration of multiple executables, level-processors, and/or scientific tools
- Can be parameterized and replayed
OODT Process Control System Flow
Scientific Workflow Software Architecture

SWSA: A DSSA for Workflow-Based Science Data Systems - March 2nd, 2010
Co-Development Process

Existing Software Reuse
- Kernel Recovery

New Development
- Discovery Phase
- Workflow Dev.

Engineering Efforts
- Production Environment Setup
- Handler Specification

Integration

Cal/Val

Improvements
- Iterative Production Phases

SWSA: A DSSA for Workflow-Based Science Data Systems - March 2nd, 2010
Current Implementation

• Prism-MW
  – “Architectural” event-based middleware in Java
  – Components, connectors, and topology implemented as first class elements
    • Kernel wrappers implemented as event-based components with standard interfaces
  – Component and Connector behavior can be dynamically installed through factories at run-time
    • Custom event handlers for routing messages to engineering services are installed in invoking connectors
Prism-Middleware

- Originally developed for embedded systems
  - Light-weight
  - Performant
- Architectural constructs are first class elements
  - Couples architecture to source code
- Support for QoS monitoring, component migration
Performance Impacts

- **Computation Time in Fortran**
  - Fortran 77
  - Thin Wrapper
  - Full Wrapper

- **Spacetime in Fortran**
  - Fortran 77
  - Thin Wrapper
  - Full Wrapper

- **Computation Time in C**
  - Thin Wrapper
  - Full Wrapper

- **Spacetime in C**
  - Thin Wrapper
  - Full Wrapper
Decomposition Case Study

• Existing scientific software represents a significant investment
  – Complexity of the domain and code
  – Longevity

• In order to refactor existing scientific code in SWSA, decomposition into workflow components is necessary

• We recently performed a case study to explore methodologies for decomposing existing scientific systems
# Case Study

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
<th>SLOC</th>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler</td>
<td>Time-dependent Euler equation solver simulating flow in a channel with a &quot;bump&quot; on one of the walls. Jameson's finite volume scheme is used to discretize the domain and the discrete equations are then solved using a fourth order Runge-Kutta method with local timestepping.</td>
<td>832</td>
<td>24</td>
<td>2 Student Only Teams 3 Mixed Teams</td>
</tr>
<tr>
<td>MD</td>
<td>MD is an N-body molecular dynamics code modeling particles interacting under a Lennard-Jones potential in a cubic spatial volume with periodic boundary conditions.</td>
<td>399</td>
<td>18</td>
<td>2 Student Only Teams 2 Mixed Teams</td>
</tr>
<tr>
<td>Monte Carlo</td>
<td>A financial simulation that uses Monte Carlo techniques to price products derived from the price of an underlying asset.</td>
<td>1148</td>
<td>168</td>
<td>2 Student Only Teams 4 Mixed Teams</td>
</tr>
<tr>
<td>Ray Tracer</td>
<td>A 3-D ray tracer program that renders a scene containing 64 spheres at a resolution of NxN pixels.</td>
<td>548</td>
<td>48</td>
<td>2 Student Only Teams 2 Mixed Teams</td>
</tr>
<tr>
<td>Search</td>
<td>Search solves a game of connect-4 on a 6 x 7 board using an alpha-beta pruning technique. The search uses a 64Mb transposition table with the twobig replacement strategy.</td>
<td>410</td>
<td>27</td>
<td>1 Student Only Team 3 Mixed Teams</td>
</tr>
</tbody>
</table>
Initial Findings

Good News:
• General agreement about kernels
• Level of decomposition is still a trap
  – Bottom-up approach (e.g., clustering)

Bad News:
• More nuanced function classification required
  – Students replicated some functions across kernels
  – Some functions did not appear in student decompositions
• Working to find appropriate measures of success
Conclusions and Future Work

• Conclusions:
  – Production-style computational science, like that of science data systems, requires input from both scientists and engineers
  – The separation of concerns, access to desired grid serves and virtualization of the underlying computation platform can be achieved through a domain specific software architecture
  – We have implemented such as DSSA, called SWSA in Prism-MW and have explored both performance impact and also decomposition processes.

• Future Work
  – Computer-aided decomposition through static analysis and architectural recovery
  – Integration into grid systems (OODT)
Thank You