Session 44:
Introduction to Scientific Workflows

Author/Presenter:
Gaurang Mehta & Ewa Deelman
Center For Grid Technologies, USC/ISI
Scientific Workflows

- Capture individual data transformation and analysis steps
- Allows to compose larger applications from individual application components
- The components can be independent or connected by some control flow/data flow dependencies.
- Provide automation
- Support easy analysis modifications
- Scaled up execution over several computational resources
- Foster collaborations
Types of Workflow Applications

• Providing a service to a community (Montage project)
  ♦ Data and derived data products available to a broad range of users
  ♦ A limited number of small computational requests can be handled locally
  ♦ For large numbers of requests or large requests need to rely on shared cyber infrastructure resources
  ♦ On-the-fly workflow generation, portable workflow definition

• Supporting community-based analysis (SCEC project)
  ♦ Codes are collaboratively developed
  ♦ Codes are “strung” together to model complex systems
  ♦ Ability to correctly connect components, scalability

• Processing large amounts of shared data on shared resources (LIGO project)
  ♦ Data captured by various instruments and cataloged in community data registries.
  ♦ Amounts of data necessitate reaching out beyond local clusters.
  ♦ Automation, scalability and reliability
Generating mosaics of the sky (Bruce Berriman, Caltech)

*The full moon is 0.5 deg. sq. when viewed from Earth, Full Sky is ~ 400,000 deg. sq.*

<table>
<thead>
<tr>
<th>Size of the mosaic is degrees square*</th>
<th>Number of jobs</th>
<th>Number of input data files</th>
<th>Number of Intermediate files</th>
<th>Total data footprint</th>
<th>Approx. execution time (20 procs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>232</td>
<td>53</td>
<td>588</td>
<td>1.2GB</td>
<td>40 mins</td>
</tr>
<tr>
<td>2</td>
<td>1,444</td>
<td>212</td>
<td>3,906</td>
<td>5.5GB</td>
<td>49 mins</td>
</tr>
<tr>
<td>4</td>
<td>4,856</td>
<td>747</td>
<td>13,061</td>
<td>20GB</td>
<td>1hr 46 mins</td>
</tr>
<tr>
<td>6</td>
<td>8,586</td>
<td>1,444</td>
<td>22,850</td>
<td>38GB</td>
<td>2 hrs. 14 mins</td>
</tr>
<tr>
<td>10</td>
<td>20,652</td>
<td>3,722</td>
<td>54,434</td>
<td>97GB</td>
<td>6 hours</td>
</tr>
</tbody>
</table>
Issues Critical to Scientists

- **Reproducibility** of scientific analyses and processes is at the core of the scientific method
  - Scientific versus Engineering reproducibility
  - Workflows give us the opportunity to provide reproducibility
- Scientists consider the “capture and generation of provenance information as a critical part of the workflow-generated data”
- “Sharing workflows is an essential element of education, and acceleration of knowledge dissemination.”

NSF Workshop on the Challenges of Scientific Workflows, 2006, [www.isi.edu/nsf-workflows06](http://www.isi.edu/nsf-workflows06)
Workflow Lifecycle

Reuse
- Data Products
  - Adapt, Modify
- Workflow and Component Libraries
  - Workflow Template
  - Workflow Instance
  - Map to available resources
  - Resource, Application Component Descriptions

Creation
- Data, Metadata Catalogs
  - Populate with data
  - Executable Workflow

Scheduling/Execution
- Compute, Storage and Network Resources
  - Distributed
  - Execute

Mapping
Workflow Creation

- Design a workflow
  - Find the right components
  - Set the right parameters
  - Find the right data
  - Connect appropriate pieces together
  - Find the right fillers
- Support both experts and novices
Wings: Workflow Instance Generation and Selection (Y. Gil, USC/ISI)

WINGS

Workflow Selection

Workflow Template

Data Selection

Workflow Instance

Workflow Libraries

Ontologies: Domain terms, Component types, Workflow Products

Data Repositories

Workflow Creation

Application Components

Component Specification

Workflow Management System

“Show me workflows that prune MT rules”

“Run this workflow with the WSJ-04 data set”

“Validate this workflow based on the component specs”

“Here is a new Rule pruning code, takes in a set of MT rules, is compiled for MPI”

Balatonfüred, Hungary - 6th-18th July 2008
Workflow Creation with Wings

Workflow Template
(Edited in Wings)

Workflow Instance
(Automatically generated by Wings)

Input Data Sets

Slides courtesy Yolanda Gil, USC/ISI

Balatonfüred, Hungary - 6th-18th July 2008
Bioinformatics applications

- Manual creation
- Semi-automation using custom software
- Issues:
  - Volatility of data in life sciences
  - Data and metadata storage
  - Integration of heterogeneous biological data
  - Visualisation of models
  - Brittleness
Create a gene list in Excel
Go to NCBI
Retrieve FASTA for each gene
DragoDB Blast each sequence
Copy/paste IDs into a spreadsheet
Run Repeat Masker on each sequence
copy/paste masked sequence into Excel
Run MacVector cut each seq with EcoR1
Taverna Workbench

Scufl Simple Conceptual Unified Flow Language
Taverna Writing, running workflows & examining results
SOAPLAB Makes applications available as services

Web Service
e.g. DDBJ BLAST

SOAPLAB Web Service
Any Application

Slides courtesy Katy Wolstencroft

Balatonfüred, Hungary - 6th-18th July 2008
Kepler (UCSD)

- Kepler is a software application for the analysis and modeling of scientific data
  - Builds on Ptolemy II framework and provides a GUI to construct workflows

- Actor Oriented Modeling
  - Each actor has input/output ports (data flows through the actors)
  - Parameters are static ports

- Data Connections
  - Unidirectional communication channels connect output to input ports

- Composite Actors
  - Wrap sub workflows
  - Arbitrary Nesting

- Directors
  - Define the execution semantics of workflow graph
  - executes workflow graph (some schedule)
  - sub-workflows may have different directors promotes reusability
... to an executable workflow (here: in KEPLER)

Slide Courtesy of Bertram Ludaescher
P-GRADE (SZTAKI)

- GRAPNEL - The Language
- GRM & Mercury - The Monitor
- P-GRADE Portal for web submissions
- ... You learnt more already today.
LIGO Inspiral Search Application

Use of high-level languages to describe workflows
Textual workflow representations

Inspiral workflow application is the work of Duncan Brown, Caltech, Scott Koranda, UW Milwaukee, and the LSC Inspiral group

Balatonfüred, Hungary · 6th-18th July 2008
Workflow Lifecycle

1. **Data Products**
   - Adapt, Modify
   - Workflow and Component Libraries
   - Workflow Template
   - Populate with data
   - Workflow Instance
   - Executable Workflow
     - Data, Metadata, Provenance Information
     - Compute, Storage and Network Resources

2. **Mapping**
   - Resource, Application Component Descriptions
   - Map to available resources

Balatonfüred, Hungary · 6th–18th July 2008
Why mapping?

- Many workflow systems support only executable workflow composition
- Abstraction provides
  - ease of use (do not need to worry about low-level execution details)
  - portability (can use the same workflow description to run on a number of TG resources and/or across them)
  - gives opportunities for optimization and fault tolerance
    - automatically restructure the workflow
    - automatically provide fault recovery (retry, choose different resource)
Specification: Place $Y = F(x)$ at $L$

Execution Environment: Distributed

- Find where $x$ is--- {S1, S2, ...}
- Find where $F$ can be computed--- {C1, C2, ...}
- Choose $c$ and $s$ subject to constraints (performance, space availability, ....)
- Move $x$ from $s$ to $c$
  
  - Move $F$ to $c$
- Compute $F(x)$ at $c$
- Move $Y$ from $c$ to $L$
- Register $Y$ in data registry
- Record provenance of $Y$, performance of $F(x)$ at $c$

Error! $x$ was not at $s$!

Error! $F(x)$ failed!

Error! $c$ crashed!

Error! there is not enough space at $L$!
Some challenges in workflow mapping

- Automated management of data
- Efficient mapping of workflow instances to resources
  - Runtime Performance
  - Data space optimizations
  - Fault tolerance (involves interfacing with the workflow execution system)
    - Recovery by replanning
    - plan “B”
  - Scalability
- Providing feedback to the user
  - Feasibility, time estimates
Mapping Correctly (Pegasus)

- Select where to run the computations
  - Apply a scheduling algorithm
    - HEFT, min-min, round-robin, random
    - Schedule in a data-aware fashion (data transfers, amount of storage)
    - The quality of the scheduling depends on the quality of information
  - Transform task nodes into nodes with executable descriptions
    - Execution location
    - Environment variables initializes
    - Appropriate command-line parameters set

- Select which data to access
  - Add stage-in nodes to move data to computations
  - Add stage-out nodes to transfer data out of remote sites to storage
  - Add data transfer nodes between computation nodes that execute on different resources

- Add nodes to create an execution directory on a remote site
Additional Mapping Elements (Pegasus)

• Cluster compute nodes in small granularity applications
• Add data cleanup nodes to remove data from remote sites when no longer needed
  ◆ reduces workflow data footprint
• Add nodes that register the newly-created data products
• Provide provenance capture steps
  ◆ Information about source of data, executables invoked, environment variables, parameters, machines used, performance
• Scale matters—today we can handle:
  ◆ 1 million tasks in the workflow instance (SCEC)
  ◆ 10TB input data (LIGO)
SCEC and Pegasus-WMS on the TeraGrid

Job Count / Time Duration vs Date and Hour for site USC

Total number of jobs 884,787
Total time : 230 CPU days

Balatonfüred, Hungary - 6th-18th July 2008
Workflow Lifecycle

Data Products

Adapt, Modify

Workflow and Component Libraries

Workflow Template

Populate with data

Workflow Instance

Map to available resources

Data, Metadata Catalogs

Resource, Application Component Descriptions

Data, Metadata, Provenance Information

Scheduling/Execution

Execute

Executable Workflow

Compute, Storage and Network Resources
Execution Environment

Globus and Condor Services for job scheduling
Globus Services for data transfer and Cataloging

Information Services:
--- information about data location
--- information about the execution sites
Challenges in Workflow Execution

- Resource provisioning
  - Which resources to provision if many possibilities?
  - How many resources to provision?
  - For how long?

- Fault Tolerance
  - How to recognize different types of failures
  - How to recover from failures?

- Efficient collaboration between the data and computation management systems

- Debugging
  - How to relate the workflow result (outcome) to workflow specification

- Executing in a number of environments
  - Grid, P2P, Web services
Triana, the GAT and the GAP

Grid Computing
- Job Submission
- File services
- A Graphical Grid Computing Environment or Portal

Service Based Computing:
- Deployment, discovery and communication with distributed services e.g. P2P and (GSI) Web services

GAT Interface
- Condor
- Unicore
- GridFTP
- GRMS
- WSRF
- Globus RLS
- PBS
- GridLab
- .NET
- SSH
- SGE
- LDR
- Other...

GAP Interface
- P2PS
- JXTA
- Web Services
- UDDI
- SOAP
- Grid services

Slide Courtesy of Ian Taylor
ASKALON Workflow Composition and Runtime Environment

UML-based Workflow Composition

AGWL

<agwl>
  <parallel>
    activity
  </parallel>
</agwl>

Runtime Middleware Services

Execution Engine

Scheduler

Resource Manager

Data Repository

Job

The Grid

Globus toolkit

Performance Analysis

Slide Courtesy of Thomas Fahringer
Dynamic Bindings of Workflow Abstract - Concrete

Abstract Workflow

Concrete Workflow

Resource Manager

Web Services

Executables

Activity Type (abstract)

Activity Deployment

Node 1

Node 2

Node 3

Node 4
Execution Engine

- Workflow controller
  - Converts XML-based specification (AGWL) to internal representation
  - Executes the workflow according to control and data flow dependencies
    - One separate Controller for every workflow instance

- Event system
  - Other components can subscribe to the internal events
  - e.g. logging, controller, tool (WS-Notification), ...

- Logging and database
  - For post-mortem performance analysis

- GT4 WSRF wrapper
  - Send WS-Notifications to the portal

- **Scheduler**
  - Receives jobs ready to execute from the task loop
  - Retrieves the resources with available from GridARM
  - Assigns the task to the best machine according to the selection criteria
    - Clock speed * no free processors
    - Prediction information, memory available, ...

Slide Courtesy of Thomas Fahringer
Efficient data handling (Pegasus)

• Input data is staged dynamically
• New data products are generated during execution
• For large workflows 10,000+ files
  ♦ Similar order of intermediate and output files
  ♦ Total space occupied is far greater than available space—failures occur

• Solution:
  ♦ Determine which data are no longer needed and when
  ♦ Add nodes to the workflow do cleanup data along the way

• Issues:
  ♦ minimize the number of nodes and dependencies added so as not to slow down workflow execution
  ♦ deal with portions of workflows scheduled to multiple sites

Joint work with Rizos Sakellariou, Manchester University, CCGrid 2007, Scientific Programming Journal, 2007
Dynamic Cleanup with Montage Workflow

Adding cleanup nodes to the workflow

1.25GB versus 4.5 GB
26% improvement

56% improvement

Balatonfüred, Hungary - 6th-18th July 2008

Full workflow:
185,000 nodes
466,000 edges
10 TB of input data
1 TB of output data.
Workflow Lifecycle

Reuse

Data Products

Adapt, Modify

Workflow and Component Libraries

Workflow Template

Populate with data

Data, Metadata Catalogs

Workflow Instance

Map to available resources

Executable Workflow

Resource, Application Component Descriptions

Data, Metadata, Provenance information

Execute

Compute, Storage and Network Resources

Balatonfüred, Hungary - 6th-18th July 2008
Challenges in reuse and sharing

- How to find what is already there
- How to determine the quality of what’s there
- How to invoke an existing workflow
- How to share a workflow with a colleague
- How to share a workflow with a competitor
Workflow Reuse within a system

- Template and component libraries
- Keyword and semantic search capabilities
- Composite Workflows
Broad sharing: the new frontier

- MyExperiment in the UK (University of Manchester), a repository of workflows
  http://www.myexperiment.org/

- How do you share workflows across different workflow systems?
  - How to write a workflow in Wings and execute in Taverna?
  - NSF/Mellon Workshop on Scientific and Scholarly Workflow, 2007
    https://spaces.internet2.edu/display/SciSchWorkflow/Home

- How do you interpret results from one workflow when you are using a different workflows system (provenance-level interoperability)
  - Provenance challenge http://twiki.ipaw.info/
  - Open provenance model
    http://eprints.ecs.soton.ac.uk/14979/1/opm.pdf
Workflows do not need to be exposed to the end user Portals, Providing high-level Interfaces

EarthWorks Project (SCEC), lead by with J. Muench P. Maehling, H. Francoeur, and others

Scientific Workflow Systems Summary

- Scientific workflows are used in a number of disciplines: astronomy, bioinformatics, earth sciences, ecology, physics.

- Many workflow systems, each with their own strengths and weaknesses
  - Taverna—good UI, built for bioinformatics applications (web service-based)
  - Kepler—good UI, support for different semantics between workflow components (streaming data)
  - Triana—good UI, support for execution in grid-based and service-based environments
  - P-Grade – good UI, you learnt more today then I can mention.
  - Pegasus—deals with mapping abstract workflows to resources, issues of performance, scalability (portable across environments). Eclipse GUI framework in development.
  - DAGMan—scalable and reliable workflow execution engine
  - Pegasus-WMS = Pegasus + DAGMan

- BPEL—developed within the business community, often too complex for scientific workflows
  - Scientific tools often either abstract the BPEL layer or provide a hand-tailored solution
Relevant Links

- DAGMan: [www.cs.wisc.edu/condor/dagman](http://www.cs.wisc.edu/condor/dagman)
- Kepler: [http://www.kepler-project.org/](http://www.kepler-project.org/)
- Pegasus: [http://pegasus.isi.edu](http://pegasus.isi.edu)
- P-Grade: [http://www.lpds.sztaki.hu/pgrade/](http://www.lpds.sztaki.hu/pgrade/)