User Tools and Languages for Graph-based Grid Workflows
User Tools and Languages for Graph-based Grid Workflows

Global Grid Forum 10 – Berlin, Germany
Grid Workflow Workshop

Andreas Hoheisel
(andreas.hoheisel@first.fraunhofer.de)

Fraunhofer Institute for Computer Architecture and Software Technology – Berlin, Germany
Outline

Fraunhofer Resource Grid

Grid Job Orchestration
  Petri nets
  Grid Job Builder

Workflow Enactment
  Dynamic Workflow Refinement
  Fault management
  Parameter Studies
  Grid Job Handler

Conclusions and future work
Fraunhofer Resource Grid (FhRG)

Objectives

Development and implementation of a stable and robust Grid infrastructure

Software layer on top of Globus to enable fast realizations of distributed applications (computational science & engineering)

Integration of available resources

Open Source Software:
http://www.eXeGrid.net/
Resource Repository

- automatic selection of suitable grid resources
- description of selected resources / corresponding informations with XML

Grid Job Builder

- new grid resources available?
- visual grid job authoring by end-user
- representation as resource set
- description of workflow as Petri net, GRDL enhancement
- parsing / extraction of available grid resources

Grid Job Handler

- mapping of grid job, generating new GLOBUS job
- parsing of grid job informations / Instructions etc.

Problem Definition of Grid User, Web-Interface as Entry Point (Portal)

GRDL

Soap

save

local home

load

GJDL

Execution

Soap
Grid Job Orchestration
What is a Grid Job?

<table>
<thead>
<tr>
<th>Grid job</th>
<th>Composition of Grid resources forming grid applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Software, hardware, data, (people)</td>
</tr>
<tr>
<td>Atomic job</td>
<td>Single task, indivisible component of a Grid job</td>
</tr>
<tr>
<td></td>
<td>Execution of a software component with input data</td>
</tr>
<tr>
<td></td>
<td>Future: Invocation of a WebService method call</td>
</tr>
<tr>
<td>Component Model</td>
<td>Loosely coupled software components</td>
</tr>
<tr>
<td></td>
<td>Executables that read input files and write output files</td>
</tr>
<tr>
<td></td>
<td>Communication via files and GridFTP</td>
</tr>
<tr>
<td>GJobDL</td>
<td>Description of Grid jobs on an abstract level</td>
</tr>
<tr>
<td></td>
<td>Independent from Grid infrastructure</td>
</tr>
<tr>
<td></td>
<td>Connecting software components and data</td>
</tr>
<tr>
<td></td>
<td>Based on Petri Nets</td>
</tr>
<tr>
<td></td>
<td>XML</td>
</tr>
</tbody>
</table>
### Different Grid Workflow Approaches

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherent model</td>
<td>Workflow is defined inside the software components (e.g. MPI, CORBA, Cactus)</td>
</tr>
<tr>
<td>External model</td>
<td>Workflow is defined on top of software components</td>
</tr>
<tr>
<td></td>
<td>Complete view of workflow</td>
</tr>
<tr>
<td></td>
<td>(e.g. scripts or graphs)</td>
</tr>
</tbody>
</table>

**Scripts**
- GridAnt, JPython (XCAT)

**Combined scripts + graphs**
- WSFL, XLANG, BPEL4WS, UNICORE, GSFL

**Graphs**
- DAG: Condor DAGman, Symphony
- Petri net: GJobDL (Fraunhofer Resource Grid)
ERAMAS – Pollutant Transport in the Atmosphere:
Accident → Source → Atmospheric Transport → Exposure
Why Petri Nets?

Problem
Description of complex workflows of grid jobs

DAG
Directed Acyclic Graph (see Condor, Cactus, UNICORE)
no loops

\[
\text{PARENT A CHILD B C} \\
\text{PARENT B C CHILD D}
\]

Petri Nets
Graphical flow control of discrete systems
Directed graph, can be made Turing complete
### Petri Nets

<table>
<thead>
<tr>
<th>Places</th>
<th>Files, buffers, control places</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitions</td>
<td>Software components, control transitions</td>
</tr>
<tr>
<td>Arcs from places to transitions (Place is input place of transition)</td>
<td></td>
</tr>
<tr>
<td>Arcs from transitions to places (Place is output place of transition)</td>
<td></td>
</tr>
<tr>
<td>Tokens</td>
<td>Data, State (done, failed)</td>
</tr>
</tbody>
</table>

**Rule**
A transition is activated if all input places are filled with tokens and all output places have not reached their maximum capacity of tokens

**Refinement**
A single part of a Petri Net can be replaced by a sub Petri Net

**Description of state**
A Petri Net describes workflow and state of a system
Petri Nets
(from van der Aalst und Kumar, 2000)

Task

Sequence

Choice
Petri Nets

Condition

![Petri Net Diagram]

- **begin**
  - true: **is_true** → **re_1**
  - false: **is_false** → **re_2**

- **end**
Petri Nets

Parallel execution with synchronization

\[ \text{AND-split} \rightarrow \text{re}_1 \rightarrow \text{re}_2 \rightarrow \ldots \rightarrow \text{re}_n \rightarrow \text{AND-join} \]
Petri Nets

Parallel execution without synchronization
Petri Nets

Wait all with time out
Petri Nets

Wait any with time out
Petri Nets

While ... do
In the Lagrangian approach of the dispersion calculation, concentrations of gases are represented by particles as well as real particulate matter. The model uses winds and fluctuation computed by the Diagnostic Windfield Model (DWM) to predict the pathways of individual particles or air volumes. As long as an output time step has not been reached, the individual particles can be considered as fully independent. This makes the model well suited for numerical parallelization.

The software provides the following ports as connections:

<table>
<thead>
<tr>
<th>INCOMING</th>
<th>OUTGOING</th>
</tr>
</thead>
<tbody>
<tr>
<td>latin</td>
<td>dataset (latin.dat)</td>
</tr>
<tr>
<td>line</td>
<td>dataset (line.dat)</td>
</tr>
<tr>
<td>logos</td>
<td>dataset (logos.dat)</td>
</tr>
<tr>
<td>dim</td>
<td>dataset (dim.dat)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Workflow Enactment
Dynamic Workflows

Refinement

The Grid Job Handler supplements the workflow during runtime by introducing additional tasks.

File transfer

A GridFTP task may be introduced automatically to transfer files that are not available on the remote computer.

Software deployment

A software deployment task may be introduced to install software components on a remote computer.
Implicit Fault Management

Grid middleware
Fault management that is included in the Grid middleware

Petri Net refinement
Fault management tasks are introduced automatically if the submission or execution of an atomic task fails

failed

\[
\text{count} < 4 \quad \text{failed}
\]

re-schedule max. 3 times
Explicit Fault Management

Petri Net workflow model

The user defines the fault management explicitly by including user-defined fault management tasks in the Petri Net of his Grid job.
Parameter Study

n = total number of runs (e.g. 10,000)
p = number of parallel runs (e.g. 128)
Application flow of Grid Job Handler

- Read the GJobDL document
- Create Petri Net from this job description
- Verify the Petri Net (well-formedness, liveliness, deadlocks, pits, ...)
- Start the Grid Job (own thread)
  - Collect all activated transitions
  - Evaluate conditions
  - Invoke resource mapping → repository, (meta-)scheduler
  - Refine the Petri Net
    → insert GridFTPs, fault management, etc. if necessary
  - Create and submit atomic jobs using grid middleware (e.g. GRAM)
  - The transition fires, if atomic job is “done“ or has “failed“.
Conclusions and Future Work
# Conclusions

## Description of workflow
External, graph-based workflow definition
GJobDL uses Petri Nets instead of directed acyclic graphs to model workflow of Grid jobs

## Petri Nets
Easy orchestration of complex workflows, including conditions and loops

## Dynamic workflow model
Petri Nets can be refined and modified during runtime
Adding new tasks to the workflow, e.g.:
- transfer tasks
- software deployment tasks
- fault management tasks

## Fault Management
implicit $\rightarrow$ automatic, included in Grid middleware
explicit $\rightarrow$ user-defined, included in workflow model
Future Work

Tight coupling scheme? Now: one transition $\rightarrow$ one executable
Future: one transition $\rightarrow$ one method call (?) $\rightarrow$ Grid Service (OGSA), Web Service

Simulation of Petri Nets Prediction of Petri Nets for advanced reservation of resources ($\rightarrow$ scheduler) based on software and hardware benchmarks

Fault management Workflow check pointing, recovery of grid jobs
More Information:  
http://www.fhrg.fhg.de/  
http://www.andreas-hoheisel.de/  
andreas.hoheisel@first.fraunhofer.de

Wednesday, March 10th,  
5:30pm – 7:00pm  
Senate Hall and Senate Hall corridor.  
Please join us for some German beer and pretzels sponsored by the Fraunhofer Grid Alliance