On Development of Grid-enabled Applications and Service-Oriented Scientific Environments

O.V. Sukhoroslov

Centre for Grid Technologies and Distributed Computing ISA RAS
Moscow Institute for Physics and Technology
Grid(-enabled) Application

- Application that runs on (uses) grid resources
- Often not written from scratch but ported to grid

Application types
- Single node / Multiple nodes
- Tightly-coupled (MPI) / Loosely-coupled (Bag-of-Tasks)
- CPU-intensive / Data-intensive
- Batch job / 24x7 Service
Development Challenges

- Low-level grid access mechanisms
- Interoperability between grids
- Application porting
- Implementation of distributed coordination, load balancing, fault recovery...
- Providing application to users as a service
Development Tools

- Service-Oriented Toolkits
- Application Frameworks
- Infrastructure Interfaces
Development Tools

Service-Oriented Toolkits

Application Frameworks

Infrastructure Interfaces
Infrastructure Interfaces

- High-level APIs and programming libraries for accessing distributed computing resources
  - clusters
  - grid infrastructures
  - desktop grids
  - clouds
  - ...

- Should hide the complexity and heterogeneity of the underlying middleware
Accessing gLite-based Grid (EGEE)

- Command-line Interface (User Interface)
  - Not cross platform (Scientific Linux)
- Java APIs
  - Expose low-level operations
  - Scattered among several packages with complex external dependencies
  - Non-trivial configuration in the absence of User Interface
jLite

- Simple high-level Java API
  - Similar to gLite UI commands (20/80% principle)
  - Complete gLite job management lifecycle
  - Normal, collection and parametric job types
- Easy to install
  - All external dependencies included
  - No need to install gLite User Interface
- Platform-independent
  - Runs on any Java-capable platform

http://jlite.googlecode.com/
Hiding Heterogeneity

- Can we provide unified access to different grid infrastructures?
  - Enable aggregation / simultaneous use of resources from different grids
  - Enable easy application porting to / migration between grid infrastructures

- Grid standardization (JSDL, BES)
- Unified API (SAGA) + adaptors
- Bridging technologies (EDGeS)
Development Tools

- Service-Oriented Toolkits
- Application Frameworks
- Infrastructure Interfaces
Application Frameworks

- Ready-to-use implementations of common patterns of distributed computing
  - coordination, load balancing, fault recovery...

- Provide high-level programming models which simplify application development

- Allow developer to concentrate on implementation of problem-specific parts of application
Bag-of-Tasks Applications

- Applications composed of many independent tasks
  - Parameter study
  - Monte-Carlo simulations
  - ...
- Can have thousands of tasks
- Non-trivial for running in grids (scheduling, fault recovery, data management)
Master-Worker Pattern

Master

Independent Tasks

Worker

Worker

Worker

Worker

A

B

C

D

E

A

B

C

D

E

Worker

Worker

Worker

Worker
MaWo

- Master-worker framework
- Implements generic parts of master-worker pattern
  - worker allocation, communication with master, task scheduling, data transfer, failure recovery, etc.
- Simultaneous use of different types of computing resources
  - local workstations, clusters, grids (EGI/EGEE)
- Declarative description of BoT applications
  - arbitrary executable files

http://mawo.googlecode.com/
MaWo: Architecture

1. Starting workers

2. Sending tasks and collecting results

- MASTER
- Worker Allocator
- Application

- WORKER
- App

- WORKER
- App

- WORKER
- App
MaWo: Application Description

# Ray tracing with POV-Ray

JOB_NAME = povray-benchmark
JOB_DATA = input // job input directory
JOB_INIT = /bin/sh init.sh // initialize application on worker node (unpack POV-Ray)

TASK_NUM = 48 // number of tasks
TASK_CMD = $JOB_DIR/povray-3.6/povray
  $JOB_DIR/benchmark.ini
  +I$JOB_DIR/benchmark.pov
  +L$JOB_DIR/povray-3.6/include
  +FP +O$TASK_DIR/$TASK.ppm
  +SR{$TASK*(3840/$TASK_NUM)+1}
  +ER{($TASK+1)*(3840/$TASK_NUM)}
  // parameterized task command

TASK_OUTPUT = $TASK.ppm // task output files
JOB_OUTPUT = output // job output directory
# MaWo: Web Interface

**Job Progress**

- Tasks: **1000**  Done: 403  Assigned: 11  Unassigned: 586

## Workers

<table>
<thead>
<tr>
<th>ID</th>
<th>Host</th>
<th>Status</th>
<th>Registered</th>
<th>Last Heartbeat</th>
<th>Uptime</th>
<th>Done Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>dcs.isa.ru</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:03:30 +0400</td>
<td>Sun, 29 Mar 2009 15:05:57 +0400</td>
<td>3747</td>
<td>48</td>
</tr>
<tr>
<td>1</td>
<td>gn8</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:05:42 +0400</td>
<td>Sun, 29 Mar 2009 15:05:55 +0400</td>
<td>3612</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>mc18.nesc.ed.ac.uk</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:07:28 +0400</td>
<td>Sun, 29 Mar 2009 15:05:56 +0400</td>
<td>3507</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>mc17.nesc.ed.ac.uk</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:07:30 +0400</td>
<td>Sun, 29 Mar 2009 15:05:57 +0400</td>
<td>3506</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>dgt07.ui.savba.sk</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:09:20 +0400</td>
<td>Sun, 29 Mar 2009 15:05:56 +0400</td>
<td>3396</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>dgt06.ui.savba.sk</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:09:22 +0400</td>
<td>Sun, 29 Mar 2009 15:05:55 +0400</td>
<td>3392</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>dgt05.ui.savba.sk</td>
<td>WORKING</td>
<td>Sun, 29 Mar 2009 14:09:22 +0400</td>
<td>Sun, 29 Mar 2009 15:05:55 +0400</td>
<td>3392</td>
<td>19</td>
</tr>
</tbody>
</table>
Development Tools

Service-Oriented Toolkits

Application Frameworks

Infrastructure Interfaces
Grid Application as a Service

- **Motivation**
  - Ease of use and support
  - Scientific cooperation
  - Publication of scientific results
  - Commercialization

- **Actions needed**
  - Describe service functionality
  - Define and implement protocol for remote access to service functionality
  - Deploy and maintain server listening to client requests
MathCloud

- Service-oriented environment for mathematical research
- Provides access to services for solving various types of mathematical problems
- Key goals: simplicity, ease of use, openness
- Based on modern Web and Grid technologies
  - RESTful Web services (instead of “Big” Web Services, SOAP, WSRF, etc.)
  - Services can use grid resources (grid-enabled services)

http://www.mathcloud.org/
Service-Oriented Science

• ...So-called service-oriented architectures define standard interfaces and protocols that allow developers to encapsulate information tools as services that clients can access without knowledge of, or control over, their internal workings. Thus, tools formerly accessible only to the specialist can be made available to all...

• service-oriented science refers to scientific research enabled by distributed networks of interoperating services

MathCloud Service

- RESTful Web service
  - Conforms to unified REST API
- Stateless “Web function” abstraction
  - ServiceX (in1, in2) → {out1, out2, out3}
  - Input/output parameters are encoded with JSON and described with JSON Schema
- Supports long-running jobs
  - Check job status, get job result, cancel job...
- Supports large data parameters
  - Passed as URIs (HTTP, FTP, GridFTP...)

MathCloud Service
MathCloud: Technology

• Service container
  • Implements unified REST API
  • Supports quick porting of command-line applications
  • Can run jobs in EGI/EGEE grid

• Workflow Management System
  • Web-based graphical workflow editor
  • Workflow storage/runtime with REST API
  • Publication of workflows as new services
MathCloud: Workflow Editor
MathCloud: Application Example

- V.V. Voloshinov, S.A. Smirnov
  Error-free Inversion of Ill-conditioned Matrices in Distributed Computing System of RESTful services of Computer Algebra

- S.A. Smirnov
  On Development of RESTful Web Service for a Computer Algebra System in MathCloud Environment
Thank you!
Talk Outline

- Grid Application Development Challenges
- Accessing Grid Infrastructure
- Application Programming Frameworks
- Grid Applications as Services
Related Projects

- myGrid (UK)
  - Набор средств для e-Science и e-Laboratories
  - Big Web Services (SOAP, WSDL)
  - Taverna - workflow editor
  - myExperiment – workflow repository
  - BioCatalogue – service catalogues
  - Ontology/Feta – semantic annotation and discovery
  - SoapLab/Gowlab – “application to service” wrapper
Future Work

- Security
- Service catalogue
- Data management
- Platform for hosting services
- Universal toolkit for building service-oriented scientific environments