GridWay Scalability and Interoperation for DRMAA codes

José Luis Vázquez-Poletti
(on behalf of Eduardo Huedo)

dsa-research.org

Distributed Systems Architecture Research Group
Universidad Complutense de Madrid
Contents

1. The GridWay Metascheduler
2. The DRMAA standard and GridWay
3. GridWay Approach to Scalability and Interoperability
4. The CD-HIT Application

“The more man meditates upon good thoughts, the better will be his world and the world at large.”
1. The GridWay Metascheduler

What is GridWay?

GridWay is a Globus Toolkit component for meta-scheduling, creating a scheduler virtualization layer on top of Globus services (GRAM, MDS & GridFTP)

• For **project and infrastructure directors**
  • GridWay is an open-source community project, adhering to Globus philosophy and guidelines for collaborative development.

• For **system integrators**
  • GridWay is highly modular, allowing adaptation to different grid infrastructures, and supports several OGF standards.

• For **system managers**
  • GridWay gives a scheduling framework similar to that found on local LRM systems, supporting resource accounting and the definition of state-of-the-art scheduling policies.

• For **application developers**
  • GridWay implements the OGF standard DRMAA API (C, JAVA & more bindings), assuring compatibility of applications with LRM systems that implement the standard, such as SGE, Condor, Torque,...

• For **end users**
  • GridWay provides a LRM-like CLI for submitting, monitoring, synchronizing and controlling jobs, that could be described using the OGF standard JSDL.
1. The GridWay Metascheduler

Global Architecture of a Computational Grid

- **DRMAA**
  - .C, .java

- **Results**
  - CLI

- **Applications**
  - Standard API (OGF DRMAA)
  - Command Line Interface

- **Grid Meta-Scheduler**
  - open source
  - job execution management
  - resource brokering

- **Grid Middleware**
  - Globus services
  - Standard interfaces
  - end-to-end (e.g. TCP/IP)

- **Infrastructure**
  - highly dynamic & heterogeneous
  - high fault rate

- **PBS, SGE**
1. The GridWay Metascheduler

GridWay Internals

- DRMAA library
- CLI
- GridWay Core
  - Request Manager
  - Dispatch Manager
- GridWay Core
  - Job Pool
  - Host Pool
- Transfer Manager
- Execution Manager
- Information Manager
- GridFTP
- RFT
- pre-WS GRAM
- WS GRAM
- MDS2
- MDS2 GLUE
- MDS4

Job Submission
Job Monitoring
Job Control
Job Migration

Job Preparation
Job Termination
Job Migration

- Grid File Transfer Services
- Grid Execution Services
- Grid Information Services

Resource Discovery
Resource Monitoring
2. The DRMAA standard and GridWay

What is DRMAA?

- Distributed Resource Management Application API
  - http://www.drmaa.org/
- Open Grid Forum Standard

- Homogeneous interface to different Distributed Resource Managers (DRM):
  - SGE
  - Condor
  - PBS/Torque
  - GridWay
    - C
    - JAVA
    - Perl (GW 5.2+)
    - Ruby (GW 5.2+)
    - Python (GW 5.2+)
2. The DRMAA standard and GridWay

C Binding

• The native binding

• All the others are wrappers around this

• Features a dynamic library to link DRMAA applications with
  • They will automatically run on a Grid offered by GridWay

```c
drmaa_run_job
  (job_id,
   DRMAA_JOBNAME_BUFFER-1,
   jt,
   error,
   DRMAA_ERROR_STRING_BUFFER-1);
```
2. The DRMAA standard and GridWay

Java Binding

- Uses Java Native Interface (JNI)
  - performs calls to the C library to do the work
- Two versions of the DRMAA spec
  - 0.6
  - 1.0 - Not yet officially recommended by OGF

```java
session.runJob(jt);
```
2. The DRMAA standard and GridWay

Ruby Binding

- SWIG : C/C++ wrapper generator for scripting languages and Java
- SWIG binding for Ruby developed by dsa-research.org

```
(result, job_id, error)=drmaa_run_job(jt)
```
2. The DRMAA standard and GridWay

Python Binding

- SWIG binding developed by 3rd party
  - Author: Enrico Sirola
  - License: GPL  --> external download

\[(\text{result}, \text{job\_id}, \text{error}) = \text{drmaa\_run\_job}(\text{jt})\]

Perl Binding

- SWIG binding developed by 3rd party
  - Author: Tim Harsch
  - License: GPL  --> external download

\[($\text{result}, $\text{job\_id}, $\text{error}) = \text{drmaa\_run\_job}($\text{jt});\]
3. GridWay Approach to Scalability and Interoperability

**Definition (by OGF GIN-CG)**

- **Interoperability**: The native ability of Grids and Grid technologies to interact directly via common open standards in the near future.
  - A rather long-term solution within production e-Science infrastructures.
  - GridWay provides support for established standards: DRMAA, JSDL, WSRF…

- **Interoperation**: What needs to be done to get production Grid and e-Science infrastructures to work together as a short-term solution. Two alternatives:
  - **Adapters**: "A device that allows one system to connect to and work with another".
    - Change the middleware/tools to insert the adapter
  - **Gateways**: adapters implemented as a service.
    - No need to change the middleware/tools

GridWay provides both adapters (Middleware Access Drivers, MADs) and a gateway (GridGateWay, WSRF GRAM service encapsulating GridWay).

GridWay’s light concept helps to maintain **Scalability**.
3. GridWay Approach to Scalability and Interoperability

How do we achieve interoperability

- By using adapters:

  “A device that allows one system to connect to and work with another”
3. GridWay Approach to Scalability and Interoperability

**EGEE**

- The Enabling Grids for E-sciencE European Commission funded project brings together scientists and engineers from more than 240 institutions in 45 countries world-wide to provide a seamless Grid infrastructure for e-Science that is available to scientists 24 hours-a-day.

**Interoperability Issues**

- Execution Manager Driver for preWS
- Different data staging philosophy
  - Cannot stage to front node
  - Don’t know Execution Node beforehand
  - SOLUTION : Wrapper
- Virtual Organization support
3. GridWay Approach to Scalability and Interoperability

Open Science Grid

- The Open Science Grid brings together a distributed, peta-scale computing and storage resources into a uniform shared cyberinfrastructure for large-scale scientific research. It is built and operated by a consortium of universities, national laboratories, scientific collaborations and software developers.

Interoperability Issues

- MDS2 info doesn’t provide queue information
  - static monitoring

- Globus container running in a non standard port
  - MAD modification
3. GridWay Approach to Scalability and Interoperability

TeraGrid

- TeraGrid is an open scientific discovery infrastructure combining leadership class resources at eleven partner sites to create an integrated, persistent computational resource

- Interoperability Issues
  - Separated Staging Element and Working Node
    - Shared homes
    - Use of SE_HOSTNAME
    - Mix of static and dynamic data
  - Support for raw rsl extensions
    - To bypass GRAM and get info to DRMS
4. The CD-HIT Application

Application Description

• “Cluster Database at High Identity with Tolerance”
• Protein (and also DNA) clustering
  • Compares protein DB entries
  • Eliminates redundancies
• Example: Used in UniProt for generating UniRef data sets
• Our case: Widely used in the Spanish National Oncology Research Center (CNIO)
  • Input DB: 504,876 proteins / 435MB
• Infeasible to be executed on single machine
  • Memory requirements
    • Total execution time
• UniProt is the world's most comprehensive catalog of information on proteins. CD-HIT program is used to generate the UniRef reference data sets, UniRef90 and UniRef50.
• CD-HIT is also used at the PDB to treat redundant sequences
4. The CD-HIT Application

CD-HIT Parallel

- Execute cd-hit in **parallel mode**
- **Idea:** divide the input database to compare each division in parallel
  - Divide the input db
  - Repeat
    - Cluster the first division (cd-hit)
    - Compare others against this one (cd-hit-2d)
  - Merge results
- Speed-up the process and deal with **larger databases**
- **Computational characteristics**
  - Variable degree of parallelism
  - Grain must be adjusted
4. The CD-HIT Application

Database division/merging is performed in the front-end

- Several structures to invoke the underlying DRMS
- PBS, SGE and ssh
4. The CD-HIT Application

Merge sequential tasks to reduce overhead.

Provide a uniform interface (DRMAA) to interact with different DRMS. Some file manipulation still needed.
4. The CD-HIT Application

Running with 10 divisions

• Using previous set-up on TG, EGEE, OSG and UCM local cluster
4. The CD-HIT Application

Job States - Running with 14 divisions
Who’s behind the GridWay Metascheduler?

- Ignacio M. Ilorente (Leader)
- Rubén S. Montero
- Eduardo Huedo
- José Herrera
- José Luis Vázquez-Poletti
- Javier Fontán
- Tino Vázquez

Want to participate?

Visit http://www.gridway.org/ now!
Questions?

Thank you for your attention!

謝謝