The GridWay Metascheduler

Based on the Original Presentation from Tino Vázquez and Eduardo Huedo
http://www.gridway.org
at the
Open Grid Forum 21
Seattle, October 18, 2007
Contents

• Introduction
• What is GridWay?
• Architecture
• Components
• Scheduling Policies
• Examples of Grid Deployments
Introduction

- **Resource selection**: Where do I execute my job?
- **Resource preparation**: What do I need?
- **Job submission**: How do I submit my job?
- **Job monitoring**: How is my job doing?
- **Job migration**: Is there any better resource?
- **Job termination**: How do I get my output?
Introduction

- **Meta-scheduler**: Job to resource (*other schedulers*) matching (*execution management*).

- **Goal**: Optimize the performance according to a given metric (*performance model*):
  - Global Throughput
  - Resource usage
  - Application – Stand-alone, HPC, HTC and self-adaptive
  - User usage

- **Grid characteristics**
  - Heterogeneity (job requirements)
  - Dynamism (high fault rate, load, availability, price)
  - Site autonomy
What is GridWay?

The GridWay meta-scheduler is a scheduler virtualization layer on top of basic Globus services (GRAM, MDS & GridFTP)

For the user
  – A LRM-like environment for submitting, monitoring, and controlling jobs

For the developer
  – An standard-base development framework for Grid Applications

For the sysadmin
  – A policy-driven job scheduler
  – User-side Grid Accounting

For the Grid architect / solution provider
  – A modular component to use different infrastructures
  – A key component to deploy different Grids (enterprise, partner, utility…)

**Architecture**

- **Applications**
  - LRM-like Command Line Interface
  - OGF DRMAA C & JAVA Bindings
  - OGF JSDL (Posix & HPC profiles)
  - Array jobs, DAG workflows and MPI jobs

- **Grid Meta-Scheduler**
  - Advanced (Grid-aware) scheduling policies
  - Fault detection & recovery

- **Grid Middleware**
  - Straightforward deployment (basic services)
  - Components to deploy different Grids

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

- **Applications**
  - DRMAA .C, .java
  - CLI

- **Application-Infrastructure decoupling**
  - Globus Services
  - GridWay
  - PBS • • • SGE
Matching Resources for each job (user)

Resource Policies
- Rank Expressions
- Fixed Priority
- User Usage History
- Failure Rate

Job Policies
- Fixed Priority
- Urgent Jobs
- User Share
- Deadline
- Waiting Time

Grid Scheduling = Job + Resource Policies
1. **Installing GridWay standalone**
   - Uncompress tarball -> gw-<version>.tar.gz
   - ./configure
     - There are many options -- check them out in the manual
   - make
   - make install

2. **Enabling GridWay in Globus**
   - ./configure --enable-gridway

More information in the Installation & Configuration Guide
www.gridway.org/documentation/guides.php
Configuration

- $GW\_LOCATION/etc/gwd.conf
  - Configuration options for the GridWay daemon (GWD)

- $GW\_LOCATION/etc/sched.conf
  - Configuration options for GridWay built-in scheduling policies

- $GW\_LOCATION/etc/job_template.default
  - Default values for job template

- $GW\_LOCATION/etc/gwrc
  - Default environment variables for MADs

More information in the Installation & Configuration Guide
www.gridway.org/documentation/guides.php
Enterprise Grids

Characteristics

- “Small” scale infrastructures (campus/enterprise) with one meta-scheduler instance
- Resources within the same administration domain that may be running different LRMS and be geographically distributed

Goal & Benefits

- Integrate heterogeneous systems
- Improve return of IT investment
- Performance/Usage maximization
Enterprise Grids

Architecture

Applications
- Users
  - DRMAA interface
  - Portal
  - Command Line Interface

Middleware
- GridWay
  - One meta-scheduler
  - Grid-wide policies
- Globus
  - ....
- SGE Cluster
- PBS Cluster
- LSF Cluster

Examples

European Space Astronomy Center
- Data Analysis from space missions
- DRMAA

UABGrid, University of Alabama
- Bioinformatics applications
Partner Grids

Characteristics
- “Large” scale infrastructures with one or several meta-schedulers
- Resources belong to different administrative domains

Goal & Benefits
- Large-scale, secure and reliable sharing of resources
- Support collaborative projects
- Access to higher computing power to satisfy peak demands
Partner Grids

Architecture

Applications

Users

(Virtual) Organization

Users

• DRMAA interface
• Science Gateways

GridWay

• Multiple metaschedulers
• (V)Organization-wide policies

GridWay

GridWay

Middleware

Globus

Globus

Globus

Infrastructure

SGE Cluster

PBS Cluster

LSF Cluster

• Multiple Admin. Domains
• Multiple Organizations

Examples

EGEE-II
• gLite-LHC interoperability
• Virtual Organizations
  Fusion: Massive Ray Tracing
  Biomed: CD-HIT (Workflow)

AstroGrid-D, German Astronomy Community Grid
• Supercomputing resources
• Astronomy-specific resources
• GRAM interface
A Tool for Interoperability

- Different Middlewares (e.g. WS and pre-WS)
- Different Data/Execution architectures
- Different Information models
- Integration through adapters
- Global DN’s
- Demo in June 2007 at TeraGrid07
Questions?
Using GridWay: CLI
• User set up
• Job Definition
• Job Life-cycle
• Job Submission
• Job & Host Monitoring
• Job Control
• Sample Session
User Set Up

- Usually in a multi-user setting (single-user also possible)
- User environment (sh)
  - export GW_LOCATION=/usr/local/gw
  - export PATH=$PATH:$GW_LOCATION/bin
  - CLASSPATH=$GW_LOCATION/lib/drmaa.jar:$CLASSPATH
  - LD_LIBRARY_PATH=$GW_LOCATION/lib:$LD_LIBRARY_PATH
- Check $GW_LOCATION
  - etc configuration files
  - share/doc documents (www.gridway.org)
  - share/examples templates and howtos
  - var log information (debugging)
Job Definition

- Each job is defined within an experiment directory (default paths)
- Execution variables
  - EXECUTABLE = bin.$\{ARCH\}$
  - ARGUMENTS = $\{TASK_ID\}$
  - ENVIRONMENT = SCRATCH_DIR=/tmp (Also GW_* vars. are set)
- I/O files relative to exp dir (also abs path, file://, gsiftp://, http://)
  - INPUT_FILES = param.$\{TASK_ID\}$ param, inputfile
  - OUTPUT_FILES = outputfile, bin bin. $\{ARCH\}$
- Standard streams
  - STDIN_FILE = /dev/null
  - STDOUT_FILE = stdout_file.$\{JOB_ID\}$
  - STDERR_FILE = stderr_file.$\{JOB_ID\}$
Job Definition

- **Resource selection parameters**
  - REQUIREMENTS = ARCH = "i686" & CPU_MHZ > 1000
  - RANK = (CPU_MHZ * 2) + FREE_MEM_MB

- **Job Type**
  - TYPE = mpi
  - NP = 16

- **Advanced definition parameters**
  - Checkpointing parameters
  - Failure handling
  - Performance
  - Re-scheduling
  - Execution Configuration
Job Definition

Job Submission Description Language

- describing the job requirements for submission to resources (equivalent to job templates)


- jsdl2gw command to do the translation

```xml
...<jsdl:Application>
  <jsdl:ApplicationName>ls</jsdl:ApplicationName>
  <jsdl-posix:POSIXApplication>
    <jsdl-posix:Executable>/bin/ls</jsdl-posix:Executable>
    <jsdl-posix:Argument>-la file.txt</jsdl-posix:Argument>
    <jsdl-posix:Environment name="PATH">/sbin</jsdl-posix:Environment>
  </jsdl-posix:POSIXApplication>
...```
Job Life-cycle
Job Submission

- Simple Jobs

  $ gwsubmit example/jt

- Array Jobs ($TASK_ID and custom parametric var.)

  $ gwsubmit -v -n 4 pi.jt

  ARRAY ID: 0
  TASK JOB
  0    3
  1    4
  2    5
  3    6
DAG Workflows

- $ gwsubmit -v -t A.jt
  JOB ID: 5

- $ gwsubmit -v -t B.jt -d "5"
  JOB ID: 6

- $ gwsubmit -v -t C.jt -d "5"
  JOB ID: 7

- $ gwsubmit -t C.jt -d "6 7"
For More Information

- User guide
- Command reference guide

http://www.gridway.org/documentation/documentation/guides.php
Questions?
Developing Applications with GridWay: DRMAA
Contents

- Introduction
- Program Structure and Compilation
- DRMAA Directives and Functions
- More Information
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 implementation
  - C & JAVA
  - Perl, Ruby & Python
    > coming soon, check the next development release GW 5.3, due end of october 2007
Programming Model

Computational Problem

Task A

Task B

Task C

#include <drmaa.h>

Grid-enabled Executable

Distributed Resource Management System (DRMS)

GridWay

GLOBUS

PBS

SGE

Local Job managers

Results
Application Profiles

- **Embarrassingly Distributed**

  ```
  rc = drmaa_init(contact, err);
  // Execute initial job and wait for it
  rc = drmaa_run_job(job_id, jt, err);
  rc = drmaa_wait(job_id, &stat, timeout, rusage, err);
  // Execute n jobs simultaneously and wait
  rc = drmaa_run_bulk_jobs(job_ids, jt, 1, JOB_NUM, 1, err);
  rc = drmaa_synchronize(job_ids, timeout, 1, err);
  // Execute final job and wait for it
  rc = drmaa_run_job(job_id, jt, err);
  rc = drmaa_wait(job_id, &stat, timeout, rusage, err);
  rc = drmaa_exit(err_diag);
  ```

- **Master-Worker**

  ```
  rc = drmaa_init(contact, err_diag);
  // Execute initial job and wait for it
  rc = drmaa_run_job(job_id, jt, err_diag);
  rc = drmaa_wait(job_id, &stat, timeout, rusage, err_diag);
  while (exitstatus != 0)
  {
    // Execute n Workers concurrently and wait
    rc = drmaa_run_bulk_jobs(job_ids, jt, 1, JOB_NUM, 1, err_diag);
    rc = drmaa_synchronize(job_ids, timeout, 1, err_diag);
    // Execute the Master, wait and get exit code
    rc = drmaa_run_job(job_id, jt, err_diag);
    rc = drmaa_wait(job_id, &stat, timeout, rusage, err_diag);
    rc = drmaa_wexitstatus(&exitstatus, stat, err_diag);
  }
  rc = drmaa_exit(err_diag);
  ```
Program Structure and Compilation

- Include the DRMAA library:
  ```
  #include "drmaa.h"
  ```

- Verify the following environment variable:
  ```
  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GW_LOCATION/lib/
  ```

- Include the compiling and linking options for DRMAA:
  ```
  -L $GW_LOCATION/lib
  -I $GW_LOCATION/include
  -ldrmaa
  ```

- Example:
  ```
  $ gcc example.c -L $GW_LOCATION/lib \
  -I $GW_LOCATION/include -ldrmaa -o example
  ```
Session Management

- Initialization and finalization

```c
int drmaa_init (const char *contact, char *error_diagnosis, size_t error_diag_len);
int drmaa_exit (char *error_diagnosis, size_t error_diag_len);
```
## Auxiliary Functions

### Getting Information

- `const char * drmaa_strerror (int drmaa_errno);`
- `int drmaa_get_contact (char *contact, size_t contact_len, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_version (unsigned int *major, unsigned int *minor, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_get_DRM_system (char *drm_system, size_t drm_system_len, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_get_DRMAA_implementation (char *drmaa_impl, size_t drmaa_impl_len, char *error_diagnosis, size_t error_diag_len);`
Job Template

- Allocation and Deletion

```c
int drmaa_allocate_job_template (drmaa_job_template_t **jt, char *error_diagnosis, size_t error_diag_len);
int drmaa_delete_job_template (drmaa_job_template_t *jt, char *error_diagnosis, size_t error_diag_len);
```

- Parameter Setting/Getting

```c
int drmaa_set_attribute (drmaa_job_template_t *jt, const char *name, const char *value, char *error_diagnosis, size_t error_diag_len);
int drmaa_get_attribute (drmaa_job_template_t *jt, const char *name, char *value, size_t value_len, char *error_diagnosis, size_t error_diag_len);
int drmaa_set_vector_attribute (drmaa_job_template_t *jt, const char *name, const char *value[], char *error_diagnosis, size_t error_diag_len);
int drmaa_get_vector_attribute (drmaa_job_template_t *jt, const char *name, drmaa_attr_values_t **values, char *error_diagnosis, size_t error_diag_len);
int drmaa_get_attribute_names (drmaa_attr_names_t **values, char *error_diagnosis, size_t error_diag_len);
int drmaa_get_vector_attribute_names (drmaa_attr_names_t **values, char *error_diagnosis, size_t error_diag_len);
```
DRMAA_REMOTE_COMMAND
DRMAA_V_ARGV
DRMAA_V_ENV
DRMAA_INPUT_PATH
DRMAA_OUTPUT_PATH
DRMAA_ERROR_PATH
DRMAA_WD
DRMAA_JOB_NAME
DRMAA_JS_STATE
DRMAA_SUBMISSION_STATE_ACTIVE
DRMAA_SUBMISSION_STATE_HOLD
DRMAA_PLACEHOLDER_HD
DRMAA_PLACEHOLDER_INCR
DRMAA_PLACEHOLDER_WD
DRMAA_DEADLINE_TIME
DRMAA_DEADLINE_TIME
DRMAA_DEADLINE_TIME
GridWay Specific Job Template Compilation

DRMAA_GW_TOTAL_TASKS
DRMAA_GW_JOB_ID
DRMAA_GW_TASK_ID
DRMAA_GW_PARAM
DRMAA_GW_MAX_PARAM
DRMAA_GW_ARCH
DRMAA_V_GW_INPUT_FILES
DRMAA_V_GW_OUTPUT_FILES
DRMAA_V_GW_RESTART_FILES
DRMAA_GW_RESCHEDULE_ON_FAILURE
DRMAA_GW_NUMBER_OF_RETRIES
DRMAA_GW_RANK
DRMAA_GW_REQUIREMENTS
DRMAA_GW_TYPE
DRMAA_GW_TYPE_SINGLE
DRMAA_GW_TYPE_MPI
DRMAA_GW_NP
Simple Job Submission

```c
int drmaa_run_job (char *job_id, size_t job_id_len, drmaa_job_template_t *jt, char *
error_diagnosis, size_t error_diag_len);
```
Wait for Job Completion

```c
int drmaa_wait (const char *job_id, char *job_id_out, size_t job_id_out_len, int *stat,
                signed long timeout, drmaa_attr_values_t **rusage, char *error_diagnosis,
                size_t error_diag_len);
```

> **job_id** value could be

  DRMAA_JOB_IDS_SESSION_ANY or
  DRMAA_JOB_IDS_SESSION_ALL
Interpreting Job Status Code

- `int drmaa_wexitstatus (int *exit_status, int stat, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_wifexited (int *exited, int stat, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_wifsignaled (int *signaled, int stat, char *error_diagnosis, size_t error_diag_len);`
- `int drmaa_wtermsig (char *signal, size_t signal_len, int stat, char *error_diagnosis, size_t error_diag_len);`
String Lists

```c
int drmaa_get_next_attr_name (drmaa_attr_names_t *values, char *value, size_t value_len);
int drmaa_get_next_attr_value (drmaa_attr_values_t *values, char *value, size_t value_len);
int drmaa_get_num_attr_names (drmaa_attr_names_t *values, size_t *size);
int drmaa_get_num_attr_values (drmaa_attr_values_t *values, size_t *size);
void drmaa_release_attr_names (drmaa_attr_names_t *values);
void drmaa_release_attr_values (drmaa_attr_values_t *values);
```
Job Status and Control

- **Get Job Status**

  ```c
  int drmaa_job_ps (const char *job_id, int *remote_ps, char *error_diagnosis, size_t error_diag_len);
  ```

  - `remote_ps` returns `DRMAA_PS_QUEUED_ACTIVE`, `DRMAA_PS_RUNNING`, `DRMAA_PS_USER_ON_HOLD`, `DRMAA_PS_DONE`, `DRMAA_PS_FAILED` or `DRMAA_PS_UNDETERMINED`

- **Job Control**

  ```c
  int drmaa_control (const char *jobid, int action, char *error_diagnosis, size_t error_diag_len);
  ```

  - `action` value can be `DRMAA_CONTROL_SUSPEND`, `DRMAA_CONTROL_RESUME`, `DRMAA_CONTROL_TERMINATE`, `DRMAA_CONTROL_HOLD` or `DRMAA_CONTROL_RELEASE`
### Synchronize Jobs

```c
int drmaa_synchronize (const char *job_ids[], signed long timeout, int dispose, char *error_diagnosis, size_t error_diag_len);
```

- **timeout** value could be
  - `DRMAA_TIMEOUT_WAIT_FOREVER` or
  - `DRMAA_TIMEOUT_NO_WAIT`
Job Submission

Bulk Job Submission

```c
int drmaa_run_bulk_jobs (drmaa_job_ids_t **jobids, drmaa_job_template_t *jt, int start, int end, int incr, char *error_diagnosis, size_t error_diag_len);
```
## Helper Functions

### String Lists

```
int drmaa_get_next_job_id (drmaa_job_ids_t *values, char *value, size_t value_len);
int drmaa_get_num_job_ids (drmaa_job_ids_t *values, size_t *size);
void drmaa_release_job_ids (drmaa_job_ids_t *values);
```
**DRMAA Error Codes**

<table>
<thead>
<tr>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRMAA_ERRNO_SUCCESS</td>
</tr>
<tr>
<td>DRMAA_ERRNO_INTERNAL_ERROR</td>
</tr>
<tr>
<td>DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE</td>
</tr>
<tr>
<td>DRMAA_ERRNO_AUTH_FAILURE</td>
</tr>
<tr>
<td>DRMAA_ERRNO_INVALID_ARGUMENT</td>
</tr>
<tr>
<td>DRMAA_ERRNO_NO_ACTIVE_SESSION</td>
</tr>
<tr>
<td>DRMAA_ERRNO_NO_MEMORY</td>
</tr>
<tr>
<td>DRMAA_ERRNO_INVALID_CONTACT_STRING</td>
</tr>
<tr>
<td>DRMAA_ERRNO_DEFAULT_CONTACT_STRING_ERROR</td>
</tr>
<tr>
<td>DRMAA_ERRNO_DRMS_INIT_FAILED</td>
</tr>
<tr>
<td>DRMAA_ERRNO_ALREADY_ACTIVE_SESSION</td>
</tr>
<tr>
<td>DRMAA_ERRNO_DRMS_EXIT_ERROR</td>
</tr>
<tr>
<td>DRMAA_ERRNO_INVALID_ATTRIBUTE_FORMAT</td>
</tr>
<tr>
<td>DRMAA_ERRNO_INVALID_ATTRIBUTE_VALUE</td>
</tr>
<tr>
<td>DRMAA_ERRNO_CONFLICTING_ATTRIBUTE_VALUES</td>
</tr>
<tr>
<td>DRMAA_ERRNO_TRY_LATER</td>
</tr>
<tr>
<td>DRMAA_ERRNO_DENIED_BY_DRM</td>
</tr>
</tbody>
</table>

...
For More Information

- Application Developer Guide (DRMAA C /JAVA bindings)
- DRMAA C Howtos
- DRMAA JAVA Howtos
- DRMAA C Reference
- DRMAA JAVA Reference
- DRMAA JAVA TestSuite

http://www.gridway.org/documentation/documentation.php
Questions?