

Programming with the DRMAA OGF Standard

José Luis Vázquez-Poletti (UCM)

**Introduction to gLite & RESPECT Tools at
EGEE'09 Conference (Barcelona)**

18-19 September 2009

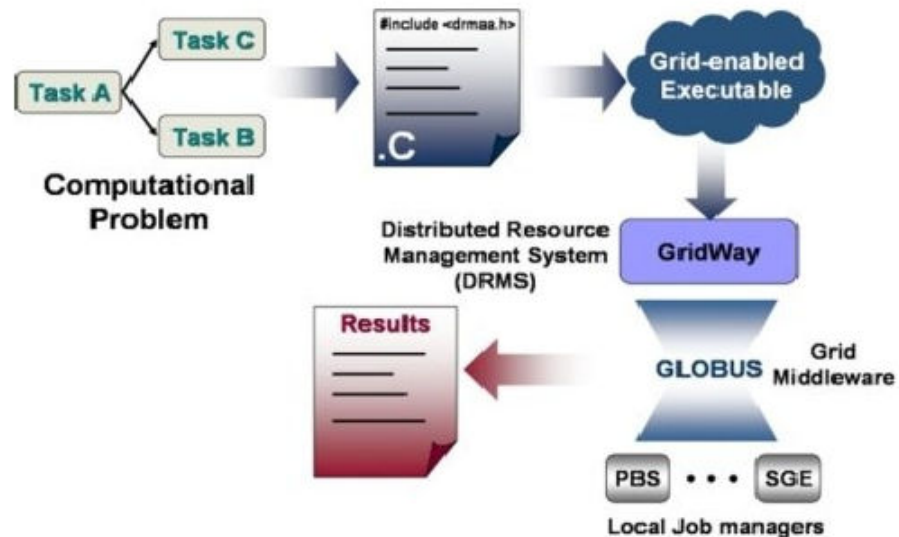
- 1. Introduction**
2. Program Structure and Compilation
3. DRMAA Sessions
4. Template Creation
5. Job Submission
6. Job Status and Control
7. Job Arrays

What is DRMAA?

- Distributed Resource Management Application API
 - <http://www.drmaa.org/>
- Open Grid Forum Standard
- Homogeneous interface to different Distributed Resource Managers (DRM):

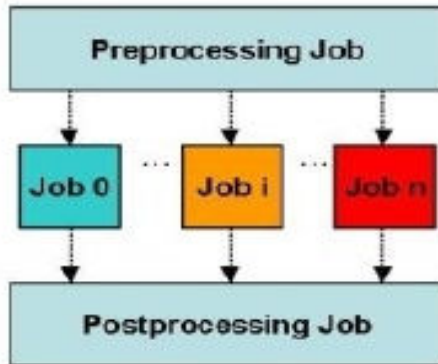


- **SGE**
- **Condor**
- **PBS/Torque**
- **GridWay**
 - C
 - JAVA
 - Perl
 - Ruby
 - Python



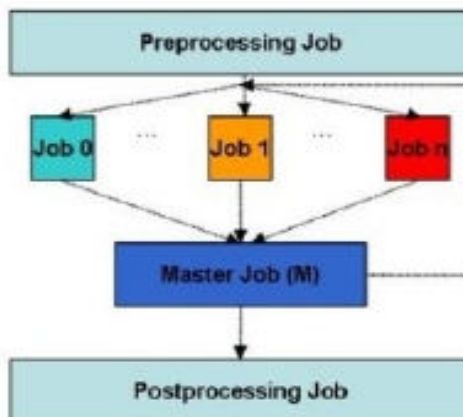
Application Profiles

- 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);
```

1. Introduction
- 2. Program Structure and Compilation**
3. DRMAA Sessions
4. Template Creation
5. Job Submission
6. Job Status and Control
7. Job Arrays

- Include the DRMAA library:

```
#include "drmaa.h"
```

- Verify the following environment variable (.bashrc):

```
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
```

1. Introduction
2. Program Structure and Compilation
- 3. DRMAA Sessions**
4. Template Creation
5. Job Submission
6. Job Status and Control
7. Job Arrays

Session Initialize

```
int drmaa_init (const char *contact, char *error_diagnosis, size_t error_diag_len)
```

- **Parameters:**
 - contact: Must be NULL for GridWay
 - error_diagnosis: Buffer where error will be stored
 - error_diag_len: Error buffer size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
 - DRMAA_ERRNO_INVALID_CONTACT_STRING
 - DRMAA_ERRNO_ALREADY_ACTIVE_SESSION

Session Finalize

```
int drmaa_exit (char *error_diagnosis, size_t error_diag_len)
```

- **Parameters:**

- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_NO_ACTIVE_SESSION

- **Other functions:**

- drmaa_get_contact - gives more information and uses the following:
 - drmaa_version
 - drmaa_get_DRM_system
 - drmaa_get_DRMAA_implementation

1. Introduction
2. Program Structure and Compilation
3. DRMAA Sessions
- 4. Template Creation**
5. Job Submission
6. Job Status and Control
7. Job Arrays

Job Template assignment

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

- **Parameters:**

- jt: Pointer to Job Template
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
- DRMAA_ERRNO_INVALID_CONTACT_STRING
- DRMAA_ERRNO_ALREADY_ACTIVE_SESSION

Set scalar attribute

```
int drmaa_set_attribute (drmaa_job_template_t *jt, const char *name,  
                        const char *value, char *error_diagnosis, size_t error_diag_len)
```

- **Parameters:**
 - jt: Pointer to Job Template
 - name: Attribute name
 - value: Attribute value
 - error_diagnosis: Buffer where error will be stored
 - error_diag_len: Error buffer size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_INVALID_ARGUMENT
 - DRMAA_ERRNO_NO_MEMORY
 - DRMAA_ERRNO_NO_ACTIVE_SESSION

Set vector attribute (i.e. string with executable arguments)

```
int drmaa_set_vector_attribute (drmaa_job_template_t *jt, const char *name,
                              const char *value[], char *error_diagnosis, size_t error_diag_len)
```

- **Parameters:**
 - jt: Pointer to Job Template
 - name: Attribute name
 - value: Attribute value
 - error_diagnosis: Buffer where error will be stored
 - **error_diag_len: Error buffer size**
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_INVALID_ARGUMENT
 - DRMAA_ERRNO_NO_MEMORY
 - DRMAA_ERRNO_NO_ACTIVE_SESSION

1. Introduction
2. Program Structure and Compilation
3. DRMAA Sessions
4. Template Creation
- 5. Job Submission**
6. Job Status and Control
7. Job Arrays

Simple job submission

```
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)
```

- **Parameters:**

- jobid: Job ID assigned by GridWay
- job_id_len: Job ID buffer size
- jt: Pointer to Job Template
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_INTERNAL_ERROR
- DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
- DRMAA_ERRNO_TRY_LATER
- DRMAA_ERRNO_NO_ACTIVE_SESSION

Wait for job execution

```
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)
```

- **Parameters:**

- jobid: Job ID assigned by GridWay
- job_id_out: Done Job ID
- job_id_out_len: job_id_out buffer size
- stat: job_id_out exit code
- timeout: DRMAA_TIMEOUT_WAIT_FOREVER, DRMAA_TIMEOUT_NO_WAIT or n_seconds
- rusage: Where remote resource usage values will be stored
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_INVALID_ARGUMENT
- DRMAA_ERRNO_INVALID_JOB
- DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
- DRMAA_ERRNO_NO_RUSAGE

Get exit code

```
int drmaa_wexitstatus (int *exit_status, int stat, char *error_diagnosis, size_t error_diag_len)
```

- **Parameters:**
 - `exit_status`: Where the exit code will be stored
 - `stat`: Status code of a done job
 - `error_diagnosis`: Buffer where error will be stored
 - `error_diag_len`: Error buffer size
- **Returns:**
 - `DRMAA_ERRNO_SUCCESS`
 - `DRMAA_ERRNO_NO_ACTIVE_SESSION`

Get remote use stats

```
int drmaa_get_next_attr_name (drmaa_attr_names_t *values, char *value, size_t value_len
```

- **Parameters:**
 - values: Value list
 - value: Actual value name
 - value_len: Actual value size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_INVALID_ARGUMENT
 - DRMAA_ERRNO_NO_ACTIVE_SESSION
 - DRMAA_ERRNO_NO_MORE_ELEMENTS
- **NOTE:** Values must be released with `drmaa_release_attr_values`

Delete Job Template

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

- **Parameters:**
 - jt: Pointer to Job Template
 - error_diagnosis: Buffer where error will be stored
 - error_diag_len: Error buffer size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_NO_ACTIVE_SESSION

1. Introduction
2. Program Structure and Compilation
3. DRMAA Sessions
4. Template Creation
5. Job Submission
- 6. Job Status and Control**
7. Job Arrays

Get Job Status

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

- **Parameters:**

- job_id: Job ID
- remote_ps: Actual job status
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

remote_ps values:

- DRMAA_PS_QUEUED_ACTIVE
- DRMAA_PS_RUNNING
- DRMAA_PS_USER_ON_HOLD
- DRMAA_PS_DONE
- DRMAA_PS_FAILED
- DRMAA_PS_UNDETERMINED

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_INTERNAL_ERROR
- DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
- DRMAA_ERRNO_INVALID_JOB
- DRMAA_ERRNO_NO_ACTIVE_SESSION

- Translation of remote_ps:

- **const char * drmaa_gw_status (int drmaa_state)**

Wait Job Finalization

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

- **Parameters:**

- job_ids[]: Job ID list (ends with NULL)
- timeout: Max waiting time
- dispose: Kill (1) or not (0) the job
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- DRMAA_ERRNO_SUCCESS
- DRMAA_ERRNO_INVALID_ARGUMENT
- DRMAA_ERRNO_INVALID_JOB
- DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
- DRMAA_ERRNO_NO_ACTIVE_SESSION

Set Job Control Signals

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

- **Parameters:**

- job_id: Job ID
- action: Control signal
- error_diagnosis: Buffer where error will be stored
- error_diag_len: Error buffer size

- **Returns:**

- *DRMAA_ERRNO_SUCCESS*
- *DRMAA_ERRNO_INTERNAL_ERROR*
- *DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE*
- *DRMAA_ERRNO_NO_ACTIVE_SESSION*
- *DRMAA_ERRNO_INVALID_ARGUMENT*
- *DRMAA_ERRNO_INVALID_JOB*
- *DRMAA_ERRNO_HOLD_INCONSISTENT_STATE*
- *DRMAA_ERRNO_RELEASE_INCONSISTENT_STATE*
- *DRMAA_ERRNO_RESUME_INCONSISTENT_STATE*
- *DRMAA_ERRNO_SUSPEND_INCONSISTENT_STATE*

Actions:

- DRMAA_CONTROL_SUSPEND
- DRMAA_CONTROL_RESUME
- DRMAA_CONTROL_TERMINATE
- DRMAA_CONTROL_HOLD
- DRMAA_CONTROL_RELEASE

1. Introduction
2. Program Structure and Compilation
3. DRMAA Sessions
4. Template Creation
5. Job Submission
6. Job Status and Control
7. **Job Arrays**

Submit Job Array

```
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)
```

- **Parameters:**
 - jobids: Vector where Job IDs will be stored
 - jt: Pointer to Job Template
 - start: First job index
 - end: Last job index
 - incr: Increment used for obtaining job total number (GridWay uses 1)
 - error_diagnosis: Buffer where error will be stored
 - error_diag_len: Error buffer size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_INTERNAL_ERROR
 - DRMAA_ERRNO_DRM_COMMUNICATION_FAILURE
 - DRMAA_ERRNO_TRY_LATER
 - DRMAA_ERRNO_NO_ACTIVE_SESSION
 - DRMAA_ERRNO_NO_MEMORY

Obtain Next Job ID

```
int drmaa_get_next_job_id (drmaa_job_ids_t *values, char *value, size_t value_len)
```

- **Parameters:**
 - values: Job ID list
 - value: Actual Job ID
 - value_len: Actual Job ID size
- **Returns:**
 - DRMAA_ERRNO_SUCCESS
 - DRMAA_ERRNO_INVALID_ARGUMENT
 - DRMAA_ERRNO_NO_ACTIVE_SESSION
 - DRMAA_ERRNO_NO_MORE_ELEMENTS

**Thank you
for your attention!**