



## Programming with the DRMAA OGF Standard

PPAM 2007  
Gdansk, Poland  
September 9-12, 2007



**Tino Vázquez**  
**Distributed Systems Architecture Group**  
**Universidad Complutense de Madrid**

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

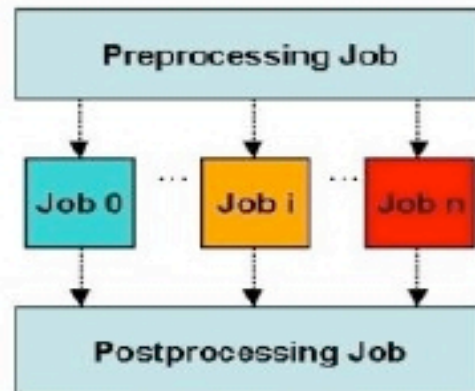
## What is DRMAA?

- 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 (coming soon!)
    - Ruby (coming soon!)
    - Python (coming soon!)



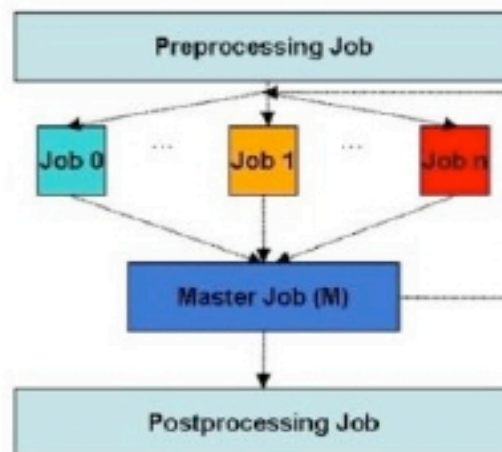
## Application Profiles

- 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

## Get Job Status

```
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!**