



Programming with the DRMAA OGF Standard

OGF25
Catania, Italy
March 2nd, 2009



José Luis Vázquez-Poletti
Distributed Systems Architecture Group
Universidad Complutense de Madrid





Contents

- 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

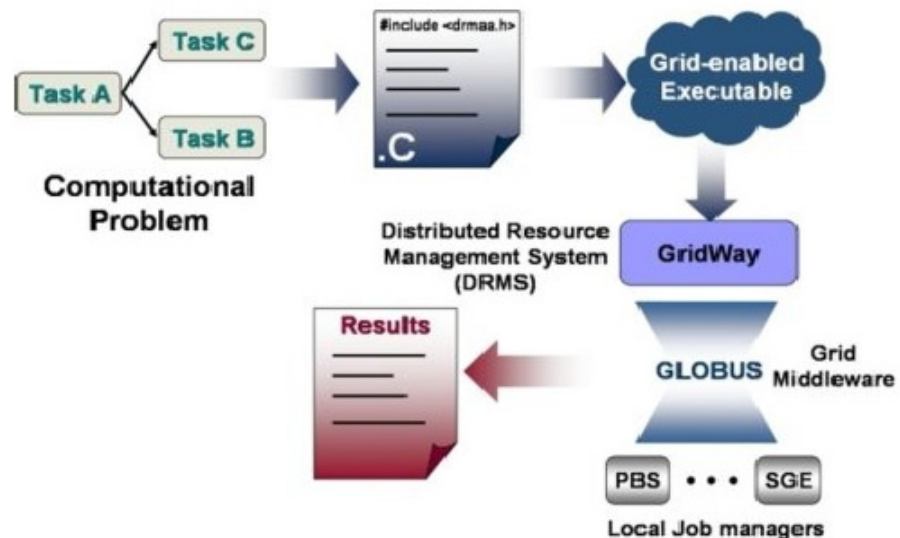
<http://www.drmaa.org/>

- Open Grid Forum Standard

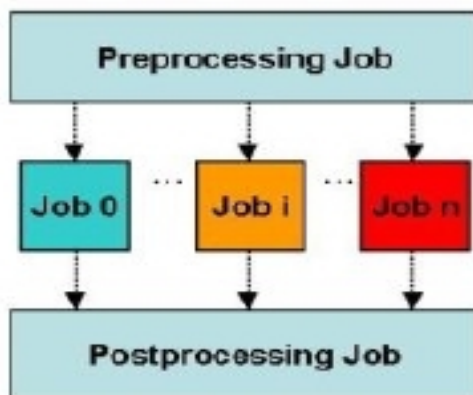


- Homogeneous interface to different Distributed Resource Managers (DRM):

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

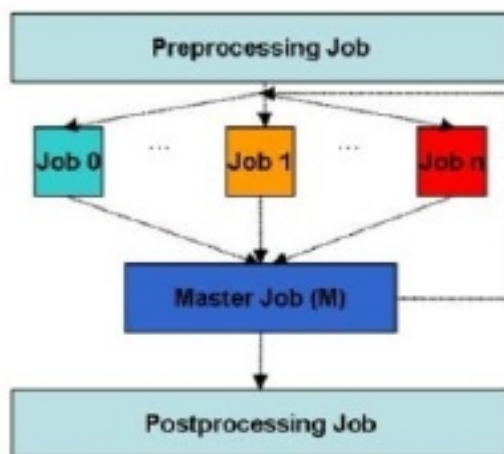


- Embarrassingly Distributed



```
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);
```



Contents

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

Program Structure and Compilation

- 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
```





The following code was extracted from **DRMAA C Howtos**

It's available at:

<http://www.gridway.org/doku.php?id=documentation:howto>

Check often this section for more
documentation and reference!





Contents

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



```
result = drmaa_init (NULL, error, DRMAA_ERROR_STRING_BUFFER-1);
```

```
if ( result != DRMAA_ERRNO_SUCCESS)
```

```
{
```

```
    fprintf(stderr,"drmaa_init() failed: %s\n", error);
```

```
    return -1;
```

```
}
```

```
else
```

```
{
```

```
printf("drmaa_init() success \n");
```

```
drmaa_get_contact(contact,
```

```
                DRMAA_ATTR_BUFFER-1,
```

```
                error,
```

```
                DRMAA_ERROR_STRING_BUFFER-1);
```

```
drmaa_version(&major,&minor,error,DRMAA_ERROR_STRING_BUFFER-1);
```

```
drmaa_get_DRM_system(drm,  
                    DRMAA_ATTR_BUFFER-1,  
                    error,  
                    DRMAA_ERROR_STRING_BUFFER-1);
```

```
drmaa_get_DRMAA_implementation(impl,  
                               DRMAA_ATTR_BUFFER-1,  
                               error,  
                               DRMAA_ERROR_STRING_BUFFER-1);
```

```
printf("Using %s, details:\n",impl);  
printf("\t DRMAA version %i.%i\n",major,minor);  
printf("\t DRMS %s (contact: %s)\n",drm,contact);
```



```
result = drmaa_exit (error, DRMAA_ERROR_STRING_BUFFER-1);
```

```
if ( result != DRMAA_ERRNO_SUCCESS)
```

```
{
```

```
    fprintf(stderr,"drmaa_exit() failed: %s\n", error);
```

```
    return -1;
```

```
}
```

```
else
```

```
    printf("drmaa_exit() success \n");
```



Contents

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

Template Creation

```
rc = drmaa_allocate_job_template(jt, error,  
                                DRMAA_ERROR_STRING_BUFFER);  
  
if ( rc != DRMAA_ERRNO_SUCCESS)  
{  
    fprintf(stderr,  
            "drmaa_allocate_job_template() failed: %s\n", error);  
    exit(-1);  
}
```

```
rc = drmaa_set_attribute(*jt,  
                        DRMAA_WD,  
                        cwd,  
                        error,  
                        DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_set_attribute(*jt,  
                          DRMAA_JOB_NAME,  
                          "jobtemplate",  
                          error,  
                          DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_set_attribute(*jt,  
                          DRMAA_REMOTE_COMMAND,  
                          "/bin/ls",  
                          error,  
                          DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_set_vector_attribute(*jt,  
                                DRMAA_V_ARGV,  
                                args,  
                                error,  
                                DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_set_attribute(*jt,  
                          DRMAA_OUTPUT_PATH,  
                          ":stdout."DRMAA_GW_JOB_ID,  
                          error,  
                          DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_set_attribute(*jt,  
                          DRMAA_ERROR_PATH,  
                          ":stderr."DRMAA_GW_JOB_ID,  
                          error,  
                          DRMAA_ERROR_STRING_BUFFER);
```

```
rc = drmaa_delete_job_template(jt,  
                                error,  
                                DRMAA_ERROR_STRING_BUFFER-  
1);
```



Contents

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

```
rc = drmaa_run_job(job_id,  
                   DRMAA_JOBNAME_BUFFER-1,  
                   jt,  
                   error,  
                   DRMAA_ERROR_STRING_BUFFER-1);
```

```
if ( rc != DRMAA_ERRNO_SUCCESS)  
{  
    fprintf(stderr, "drmaa_run_job() failed: %s\n", error);  
    return -1;  
}  
  
fprintf(stdout, "Your job has been submitted with id: %s\n",  
        job_id);
```



Contents

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

```
rc = drmaa_job_ps(id, &status, error,  
                  DRMAA_ERROR_STRING_BUFFER-1);
```

```
if ( rc != DRMAA_ERRNO_SUCCESS)  
{  
    fprintf(stderr, "drmaa_job_ps() failed: %s\n", error);  
    exit(-1);  
}
```

```
fprintf(stdout, "Job state is: %s\n", drmaa_gw_strstatus(status));
```

```
rc = drmaa_set_attribute(jt,  
                        DRMAA_JS_STATE,  
                        DRMAA_SUBMISSION_STATE_HOLD,  
                        error,  
                        DRMAA_ERROR_STRING_BUFFER-1);
```

```
rc = drmaa_run_job(job_id,  
                  DRMAA_JOBNAME_BUFFER-1,  
                  jt,  
                  error,  
                  DRMAA_ERROR_STRING_BUFFER-1);
```

```
rc = drmaa_control(job_id,  
                  DRMAA_CONTROL_RELEASE,  
                  error,  
                  DRMAA_ERROR_STRING_BUFFER-1);
```

```
rc = drmaa_synchronize(job_ids,  
                        DRMAA_TIMEOUT_WAIT_FOREVER,  
                        0,  
                        error,  
                        DRMAA_ERROR_STRING_BUFFER-1);
```

```
rc = drmaa_control(job_id,  
                   DRMAA_CONTROL_TERMINATE,  
                   error,  
                   DRMAA_ERROR_STRING_BUFFER-1);
```



Contents

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

```
rc = drmaa_run_bulk_jobs(&jobids,  
                          jt,  
                          0,  
                          1,  
                          1,  
                          error,  
                          DRMAA_ERROR_STRING_BUFFER-1);
```

```
fprintf(stderr,"Bulk job successfully submitted IDs are:\n");  
do  
{
```

```
    rc = drmaa_get_next_job_id(jobids, value,  
                               DRMAA_ATTR_BUFFER-1);
```

```
    if ( rc == DRMAA_ERRNO_SUCCESS )  
        fprintf(stderr,"\t%s\n", value);  
}while (rc != DRMAA_ERRNO_NO_MORE_ELEMENTS );
```

```
rc = drmaa_synchronize(job_ids,  
                        DRMAA_TIMEOUT_WAIT_FOREVER,  
                        0,  
                        error,  
                        DRMAA_ERROR_STRING_BUFFER-1);
```

```
do  
{
```

```
rcj = drmaa_get_next_job_id(jobids, value,  
                             DRMAA_ATTR_BUFFER-1);
```

```
drmaa_wait(value,  
            job_id_out,  
            DRMAA_JOBNAME_BUFFER-1,  
            &stat,  
            DRMAA_TIMEOUT_WAIT_FOREVER,  
            &rusage,  
            error,  
            DRMAA_ERROR_STRING_BUFFER-1);
```

```
drmaa_wexitstatus(&stat,stat,error,  
                  DRMAA_ERROR_STRING_BUFFER-1);
```

```
fprintf(stderr, "Rusage for task %s (exit code %i)\n", value,
          stat);

do
{
    rc = drmaa_get_next_attr_value(rusage, value,
                                   DRMAA_ATTR_BUFFER-1);
    if ( rc == DRMAA_ERRNO_SUCCESS )
        fprintf(stderr, "\t%s\n", value);
}while (rc != DRMAA_ERRNO_NO_MORE_ELEMENTS );
drmaa_release_attr_values(rusage);
}
}while (rcj != DRMAA_ERRNO_NO_MORE_ELEMENTS );
drmaa_release_job_ids(jobids);
```



**Thank you
for your attention!**