Porting Fusion and Biomed Applications using DRMAA API and Globus GridWay

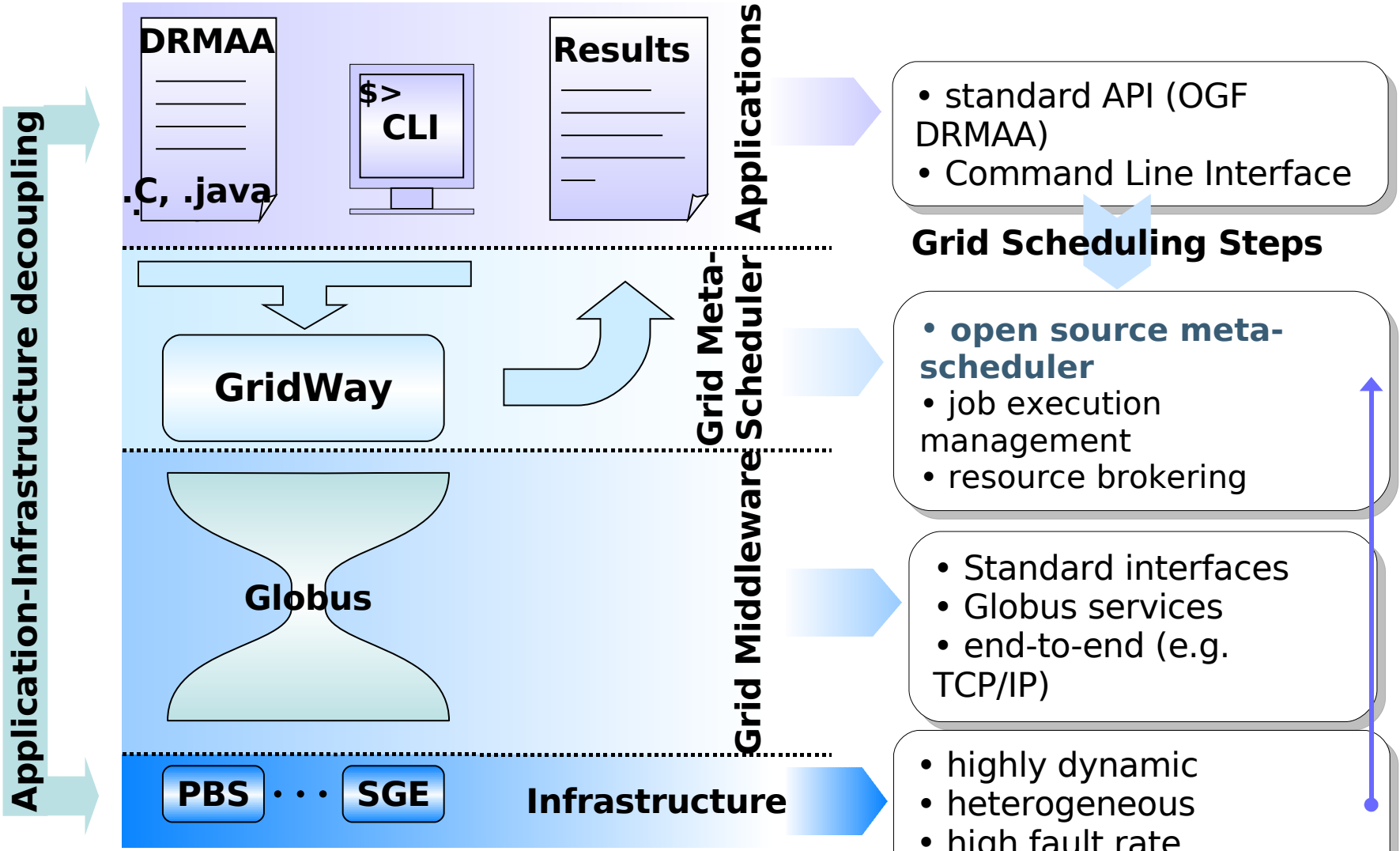
JOINT EGEE AND SEE-GRID SUMMER SCHOOL ON GRID APPLICATION SUPPORT
Budapest, Hungary
June 29, 2007

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

Contents

1. **A Global Vision**
2. GridWay Out There
3. Interacting With GridWay
   - 3.1. Command Line Interface
   - 3.2. DRMAA API (C Binding)
4. A Short Demo

1. A Global Vision



1. A Global Vision

Workload Management

- Advanced scheduling functionalities and new scheduling policies support
- Failure detection and recovery
- Accounting service
- Simple jobs, job arrays and DAGs

User Interface

- Complete support for the OGF DRMAA Standard (C & JAVA)
- DRM-like command line interface

Integration

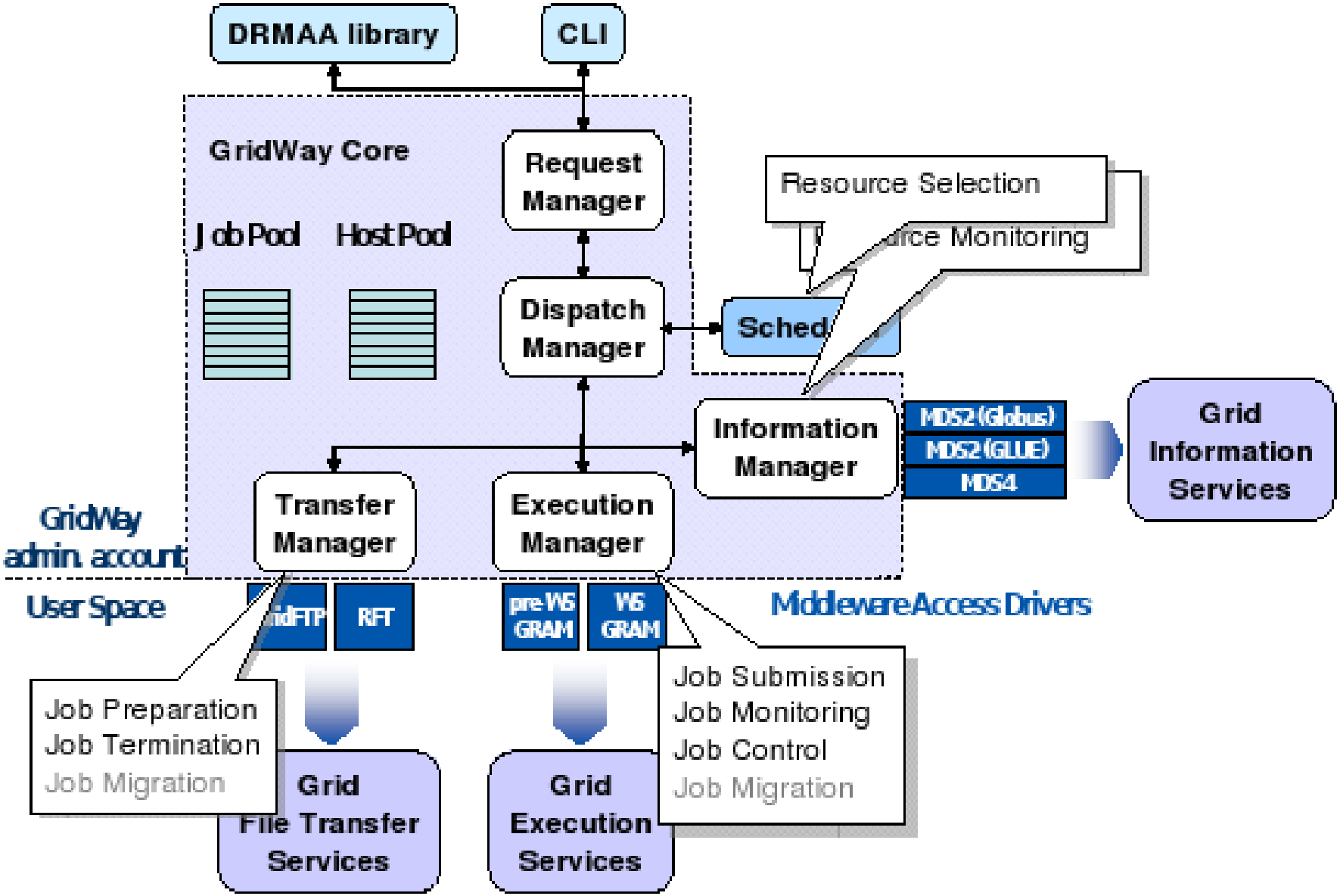
- Deployment of new services is not needed
- Interface with new grid services
- Interoperability with other infrastructures

1. A Global Vision

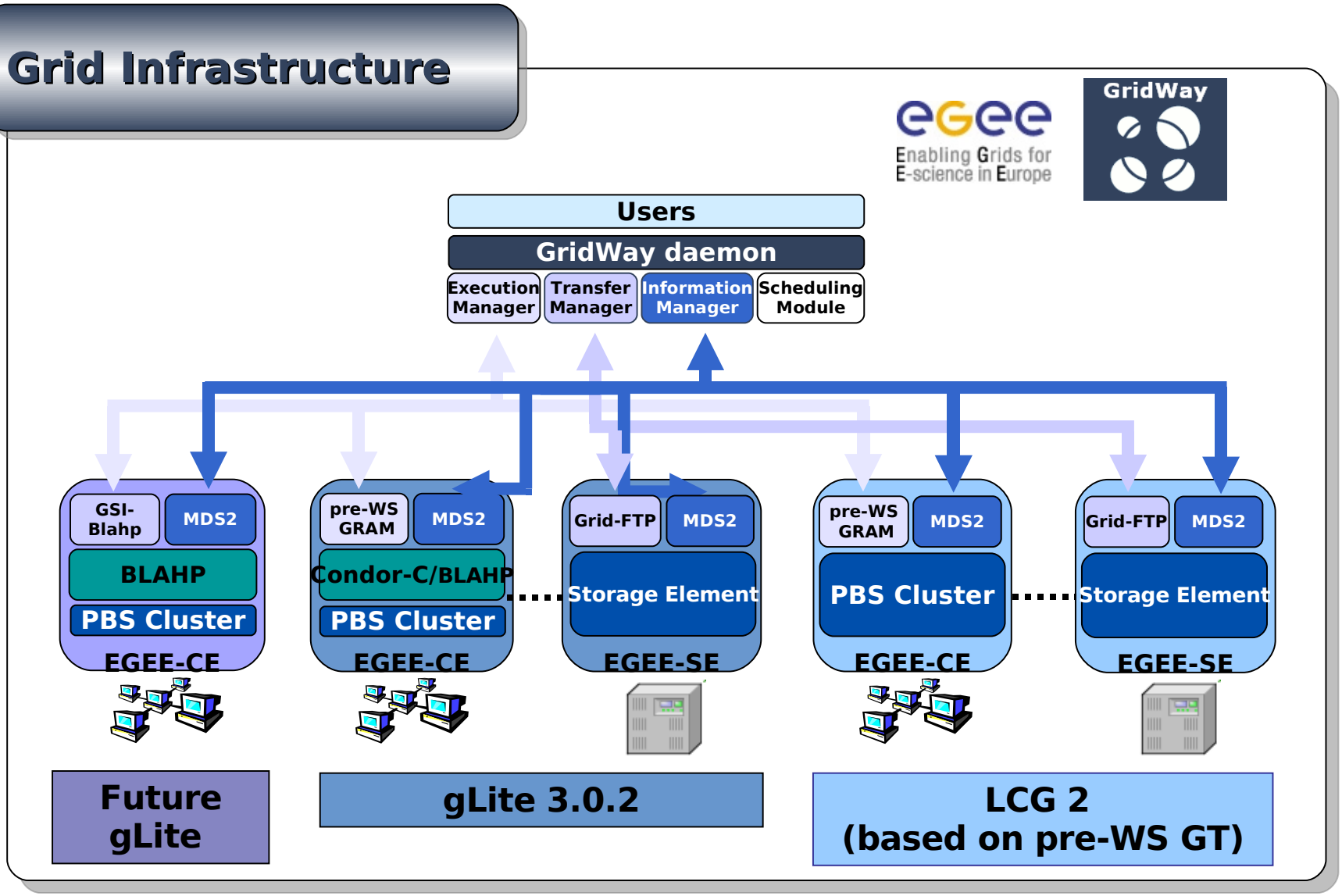
GridWay complements gLite

FEATURES	BENEFITS
DRMAA OGF Standard support (C & JAVA)	Application compatibility with DRM systems which implement the Standard: SGE, Torque, ...
DRM-like Command Line Interface (allows users to submit, kill, migrate, query state and synchronize jobs)	CLI similar to that found in local resource managers
Lightweight Middleware	Higher Efficiency for certain application profiles
Site-level Accounting and scheduling policies	Use analysis, profiling utilization and user monitoring
Minimum installation requisites	Easy deployment and mantainment
Interoperability	Simultaneous access to different infrastructures

1. A Global Vision

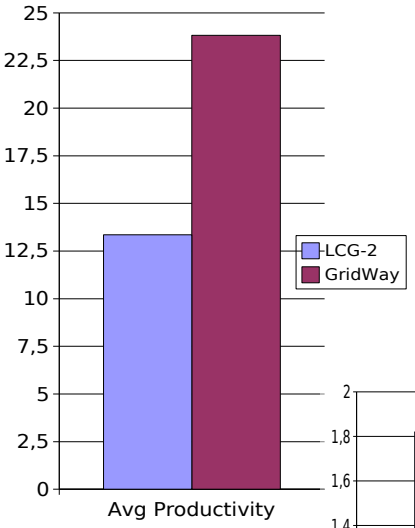


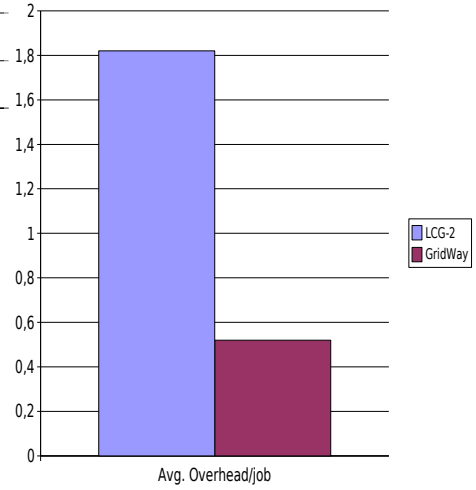
1. A Global Vision

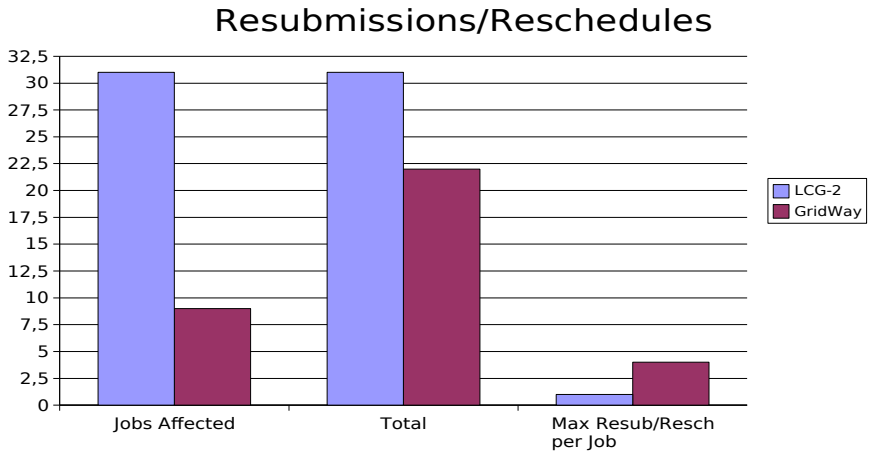


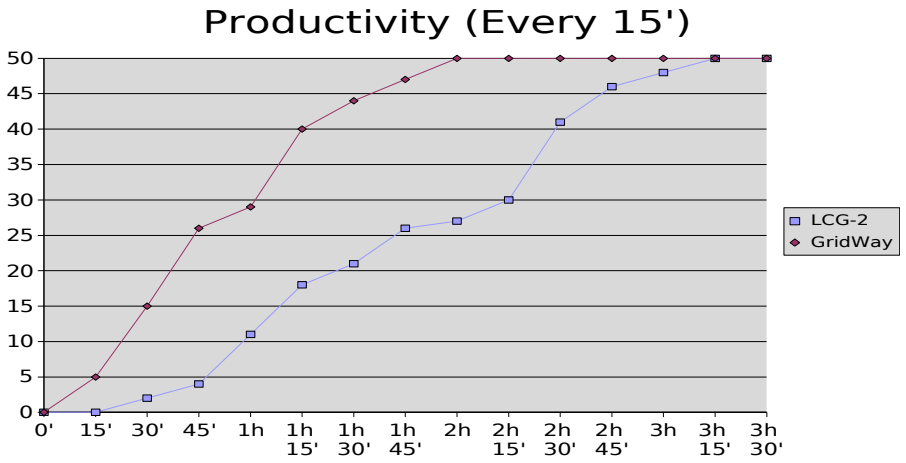
1. A Global Vision

GridWay / LCG-2 RB Comparative
EGEE 1st User Forum (CERN, 1st-3rd March 2006)









Contents

1. A Global Vision
2. **GridWay Out There**
3. Interacting With GridWay
   3.1. Command Line Interface
   3.2. DRMAA API (C Binding)
4. A Short Demo

2. GridWay Out There

Some Projects and Infrastructures

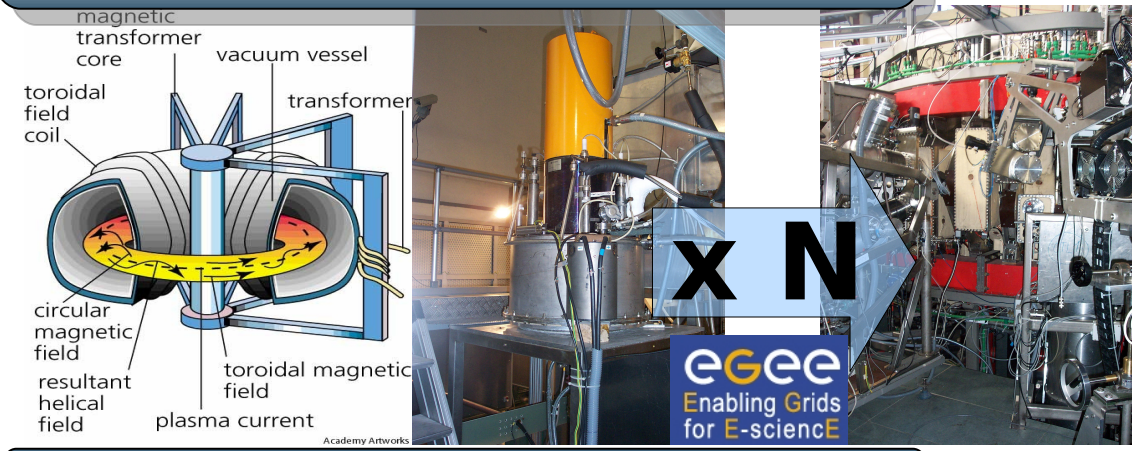
- IRISGrid
- Politecnico di Torino
- CABGrid (Centro de Astrobiología)
- C2VO (Universidad de Castilla La Mancha)
- Grid en ESAC (Agencia Espacial Europea)
- CRO-GRID (Croacia)
- Sun Microsystems Solution Center World Grid
- Infraestructura EGEE
- Proyecto BeinGRID
- GridX1 (Canadian Grid for HEP applications)
- Universidade do Porto
- Madras Institute of Technology
- National Center for High-Performance Computing

Some Application Porting Areas

- Life-Sciences
- Aerospace
- Fusion Physics
- Computational Chemistry

2. GridWay Out There

MAssive RAy TRAcing in Fusion Plasmas



MINISTERIO DE EDUCACIÓN Y CIENCIA

Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

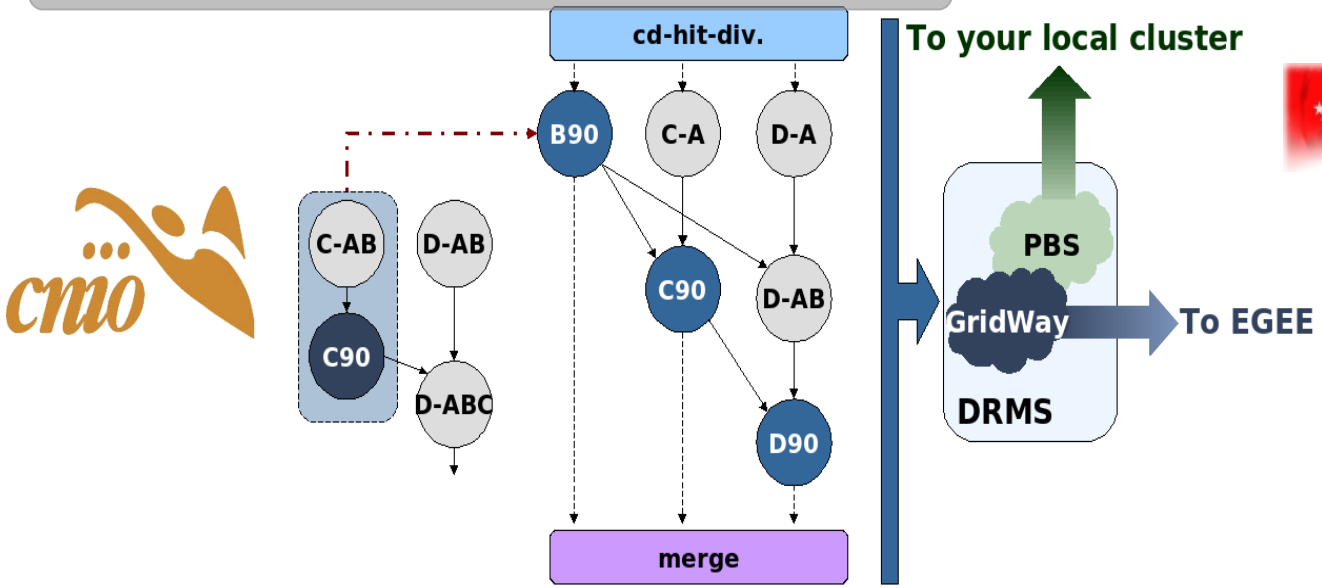
Executable: *Truba*

- 1,8 MB – 9' – 50 Executions

Input files =~ 70 KB

Output files =~ 549 KB

CD-HIT



cnio

GRIDIMadrid

eGee
Enabling Grids
for E-sciencE

Contents

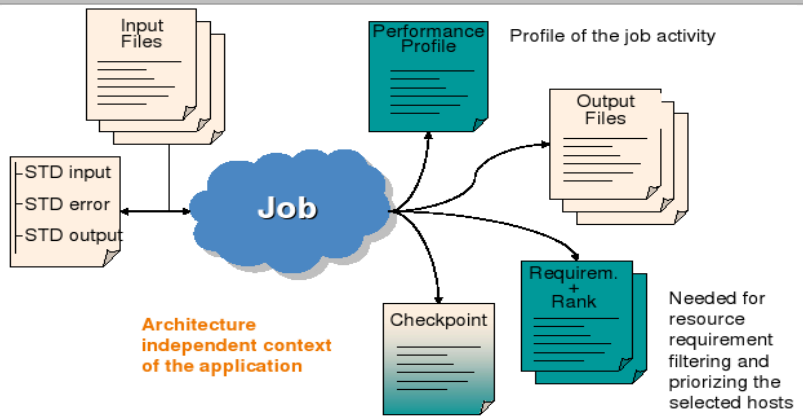
1. A Global Vision
2. GridWay Out There
3. **Interacting With GridWay**
   - **3.1. Command Line Interface**
   - **3.2. DRMAA API (C Binding)**
4. A Short Demo

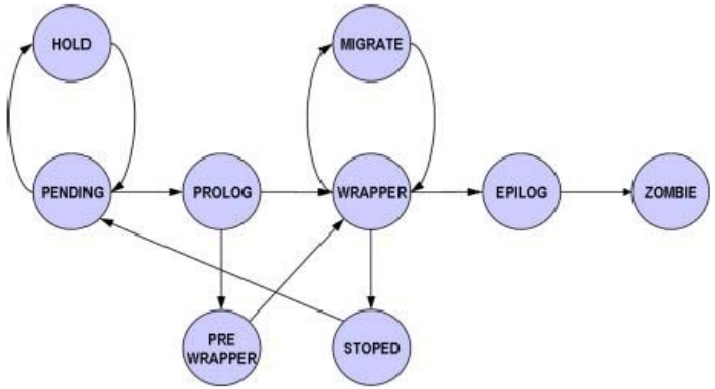
3.1 Command Line Interface

Main Commands

- **gwps:** Shows job information and state
- **gwhistory:** Shows execution history
- **gwkill:** Sends signals to a job (kill, stop, resume, reschedule)
- **gwsubmit:** Submits a job or array
- **gwwait:** Waits for job's end (any, all, set)
- **gwuser:** User Monitoring
- **gwhost:** Host Monitoring
- **gwacct:** Accounting

Application Model and Lifecycle





3.1 Command Line Interface

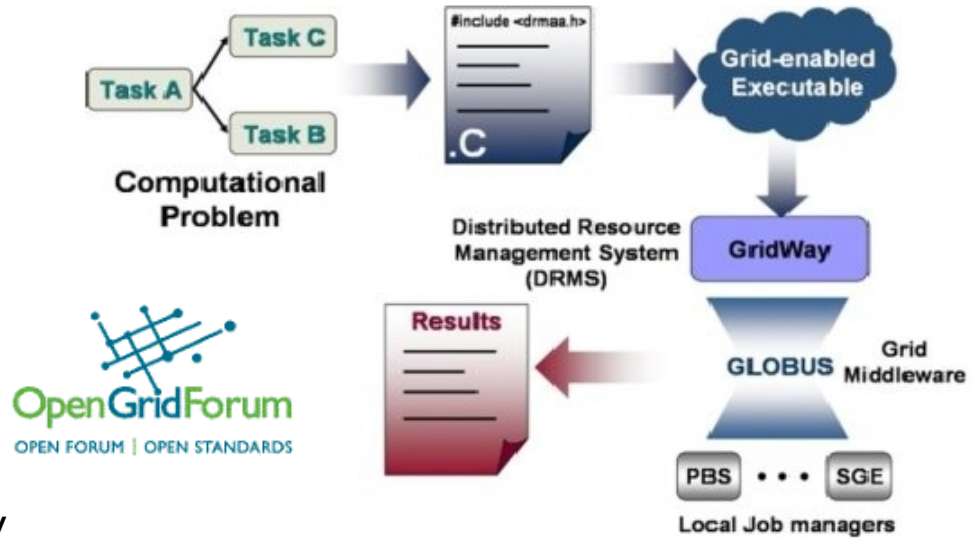
Job Template (Ex.)

```
# Execution variables
EXECUTABLE  = job
ARGUMENTS   = ${TASK_ID} ${TOTAL_TASKS} 100000
ENVIRONMENT = LD_LIBRARY_PATH=/usr/local/lib

# Resource selection parameters
REQUIREMENTS = HOSTNAME= "*.dacya.ucm.es"
RANK         = CPU_MHZ

# I/O files
INPUT_FILES  = my_inputfile
OUTPUT_FILES = my_outputfile

# Standard streams
STDOUT_FILE = stdout_file.${TASK_ID}
STDERR_FILE = stderr_file.${TASK_ID}
… other variables related to checkpointing, fault
tolerance, performance,…
```

3.2 DRMAA API (C Binding)

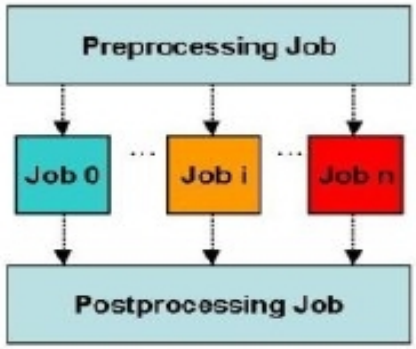
DRMAA API

- **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
    - Soon!
      - Perl & Ruby



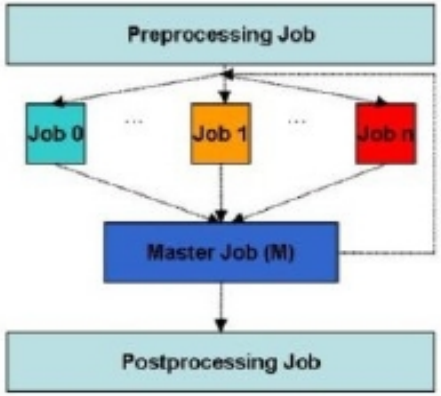
Application Profiles with DRMAA

- Embarrasingly 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 env variable (.bashrc)

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GW_LOCATION/lib/
```

Include the compiling options for DRMAA

```
-L $GW_LOCATION/lib
-I $GW_LOCATION/include
-ldrmaa
```

```
gcc ejemplo.c -L $GW_LOCATION/lib
      -I $GW_LOCATION/include -ldrmaa -o ejemplo
```

3.2 DRMAA API (C Binding)

DRMAA Sessions

Initialize DRMAA Session

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

Finalize DRMAA Session

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

Job Template Creation

Job Template Allocation

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

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)
```

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)
```

3.2 DRMAA API (C Binding)

Job Submission

Submit a Job

```
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 the Job

```
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)
```

Get the exit code

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

Get remote resource usage stats

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

Remove Job Template

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

3.2 DRMAA API (C Binding)

Job Status and Control

Get Job status

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

Wait Job ending

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

Send control signals to Job

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

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)
```

Get next Job id

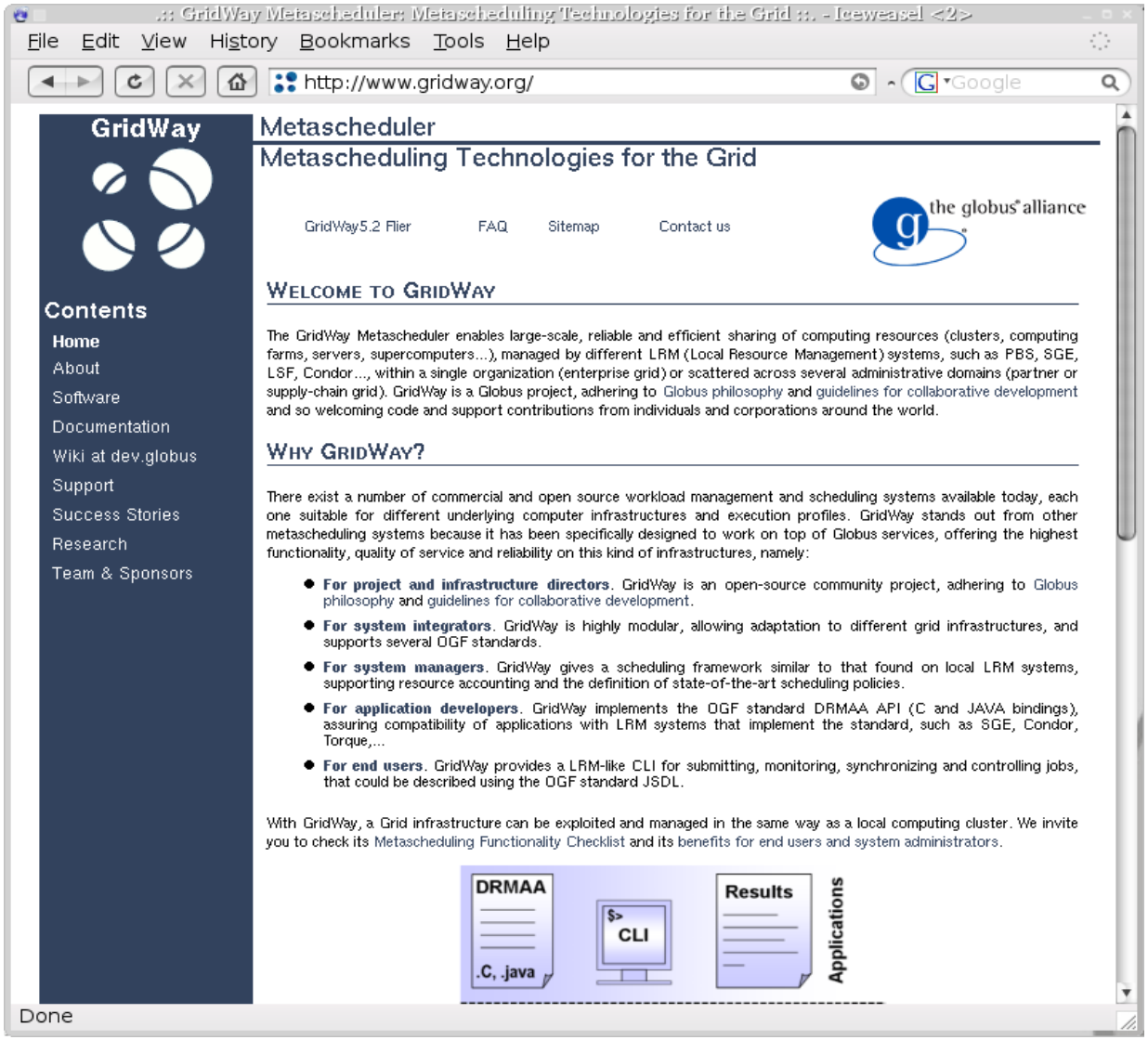
```
int drmaa_get_next_job_id (drmaa_job_ids_t *values, char *value, size_t value_len)
```

Contents

1. A Global Vision
2. GridWay Out There
3. Interacting With GridWay
   3.1. Command Line Interface
   3.2. DRMAA API (C Binding)
4. **A Short Demo**

For More Information

http://www.GridWay.org/



Thank you for your attention!