March 13, 2005
GGF13 DRMAA Tutorial

DRMAA for GridWay

Ignacio Martín Llorente
asds.dacya.ucm.es/nacho

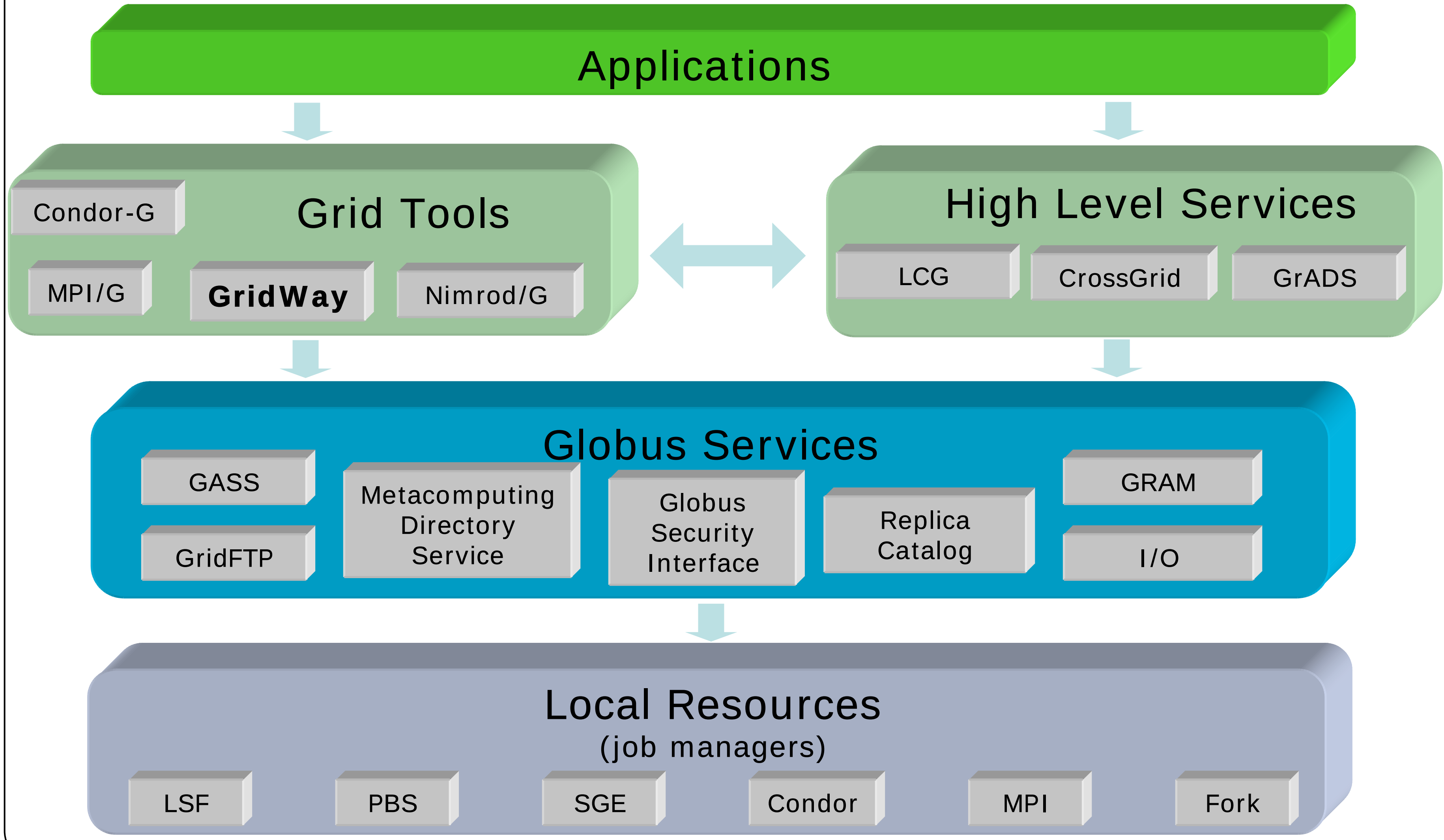
Grupo de Arquitectura de Sistemas Distribuidos y Seguridad
Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Laboratorio de Computación Avanzada, Simulación y Aplicaciones Telemáticas
Centro de Astrobiología CSIC/INTA
Asociado al NASA Astrobiology Institute

Contents

1. A Workload Management Tool for Computational Grids
2. The End-user Perspective
3. Sample Application: Computing π on the Grid
4. DRMAA Implementation
5. Future Work

GridWay in the Grid Layers



1. A Workload Management Tool for Globus

User on a Globus Grid

Where do I execute my job?	resource selection
What do I need (files, ...)?	job preparation
How do I execute my job?	job submission
How is my job doing?	job monitoring
Can I use a better host?	job migration
How do I retrieve job output?	job termination

Goal

To provide an **easier** and more **efficient** execution of jobs (***submit & forget***) on ***heterogeneous***, ***dynamic*** and ***loosely coupled*** Grids.

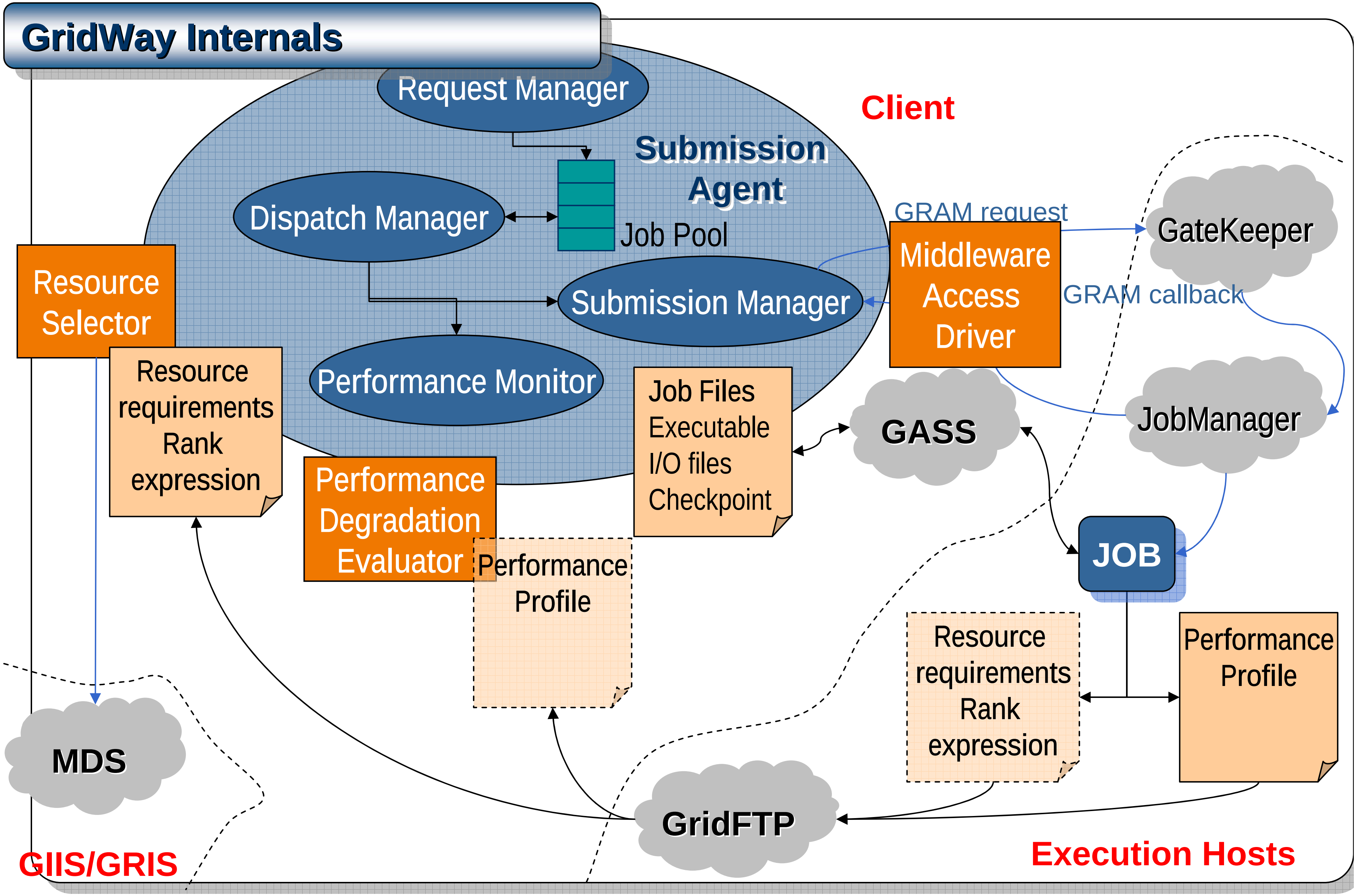
Design Guidelines

- **Easy to Adapt** (modular design)
- **Easy to Scale** (decentralized architecture)
- **Easy to Deploy** (user, standard services)

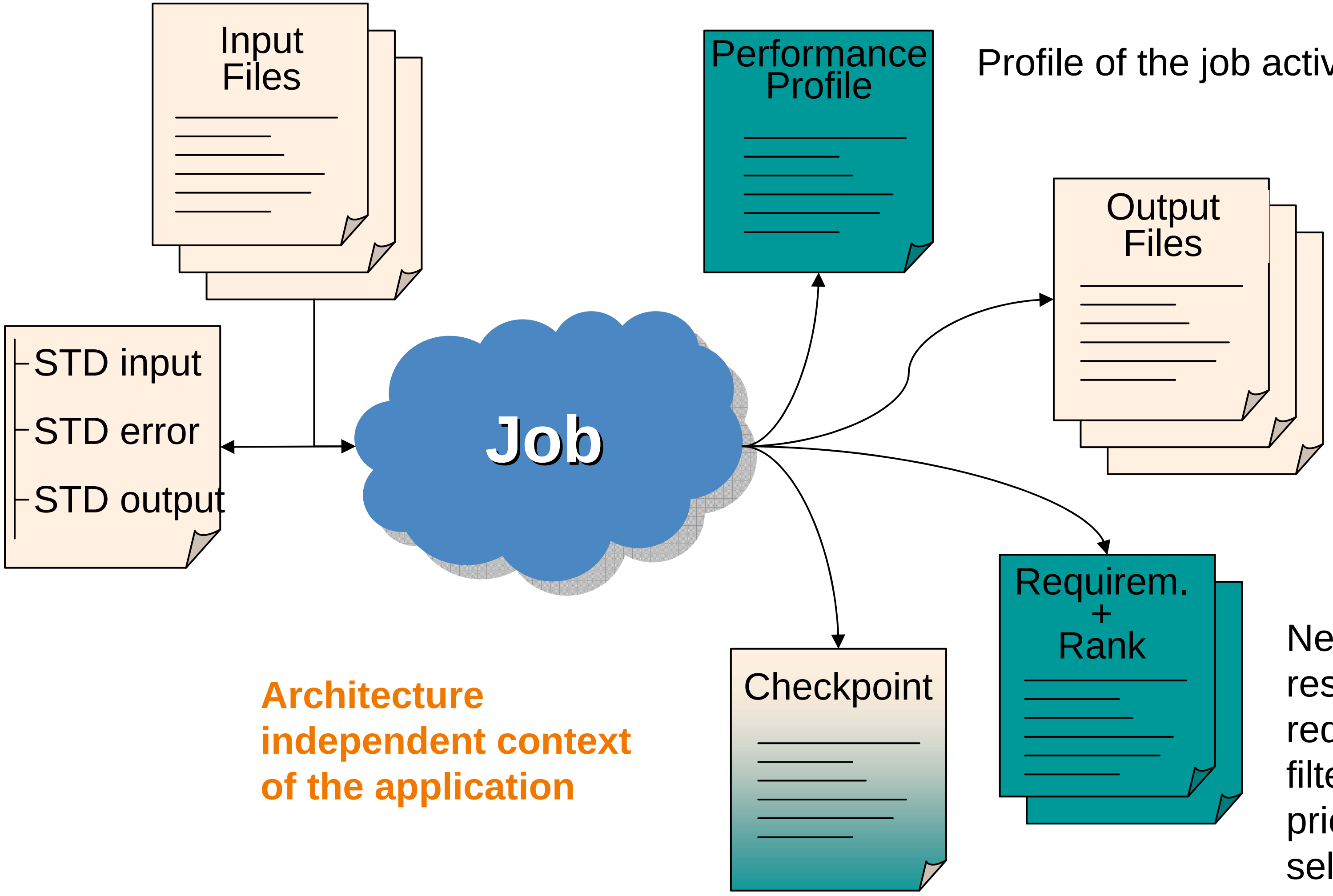
Features

- Adaptive scheduling
- Adaptive execution
- Self-adaptive applications
- Fault-tolerance

GridWay Internals



The Job Model



Profile of the job activity

Architecture independent context of the application

Needed for resource requirement filtering and priorizing the selected hosts

Job Template

```
# Scheduling Variables
DISCOVERY_TIME    = 60
DISCOVERY_TIMEOUT = 1200

# Performance Variables
POLL_TIMEOUT       = 60
SUSPENSION_TIMEOUT = 3000

# Behaviour in case of failure
ON_FAILURE        = reschedule
NUMBER_OF_RETRIES = 3

# Executable Variables
EXECUTABLE_FILE      =   /bin/ls
EXECUTABLE_ARGUMENTS =  -la
INPUT_FILES       =
OUTPUT_FILES    =
RESTART_FILES =

# Standard Streams
STDIN_FILE  =   /dev/null
STDOUT_FILE =   stdout_file.${GW_JOB_ID}
STDERR_FILE =   stderr_file.${GW_JOB_ID}

# Execution Modules
RESOURCE_SELECTOR  = scripts/rs_round_robin.sh

# Driver specific information and user-defined and module specific variables
...
```

2. The End-user Perspective

User Interface (Unix-like)

- **gwps:** display job information and status

```
JID AID TID DM         SM         GSM        STIME ETIME EXETIME EXIT HOST    TEMPLATE
0   --  --  submitted  prologue   --         --:-- --:-- --:--   --   columba SP.A
1   --  --  zombie     done       --         27:37 28:07 00:30   0    ursa    BT.A
7   --  --  pending    done       --         --:-- --:-- --:--   --   draco   SP.A
```

- **gwhistory:** display job execution history

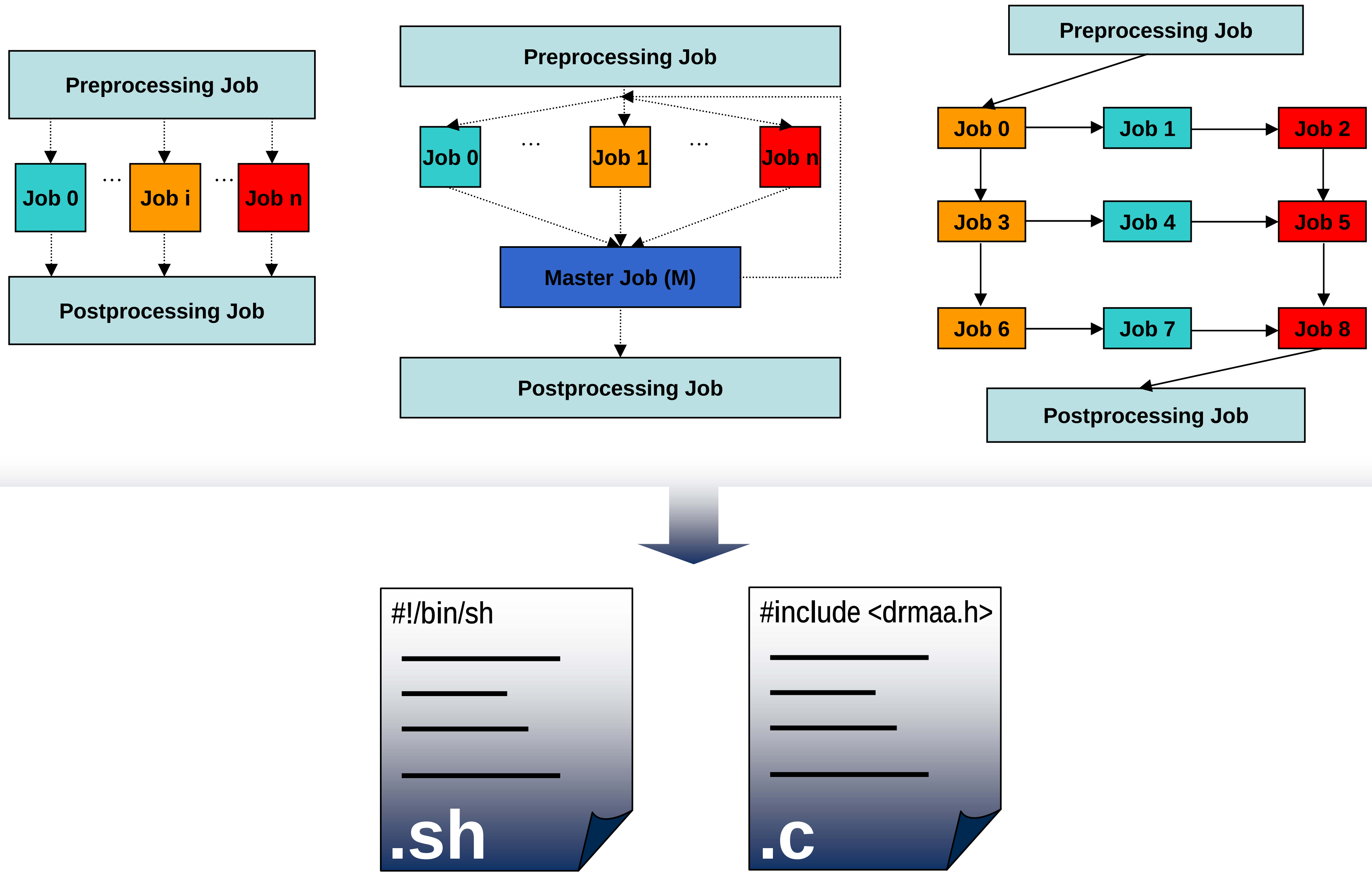
```
HOST                 RANK STIME      ETIME      EXETIME    MIGRATION_REASON
columba.dacya.ucm.es 100  --:--      --:--      --:--      --
ursa.dacya.ucm.es/jobmanager-grd 50   27:41      27:52      00:11      discovery timeout
```

- **gwkill:** signals a job (kill, stop, resume, reschedule)
- **gwsubmit:** submits a job, or an array job
- **gwwait:** waits for zombie state of a job (any, all, set)

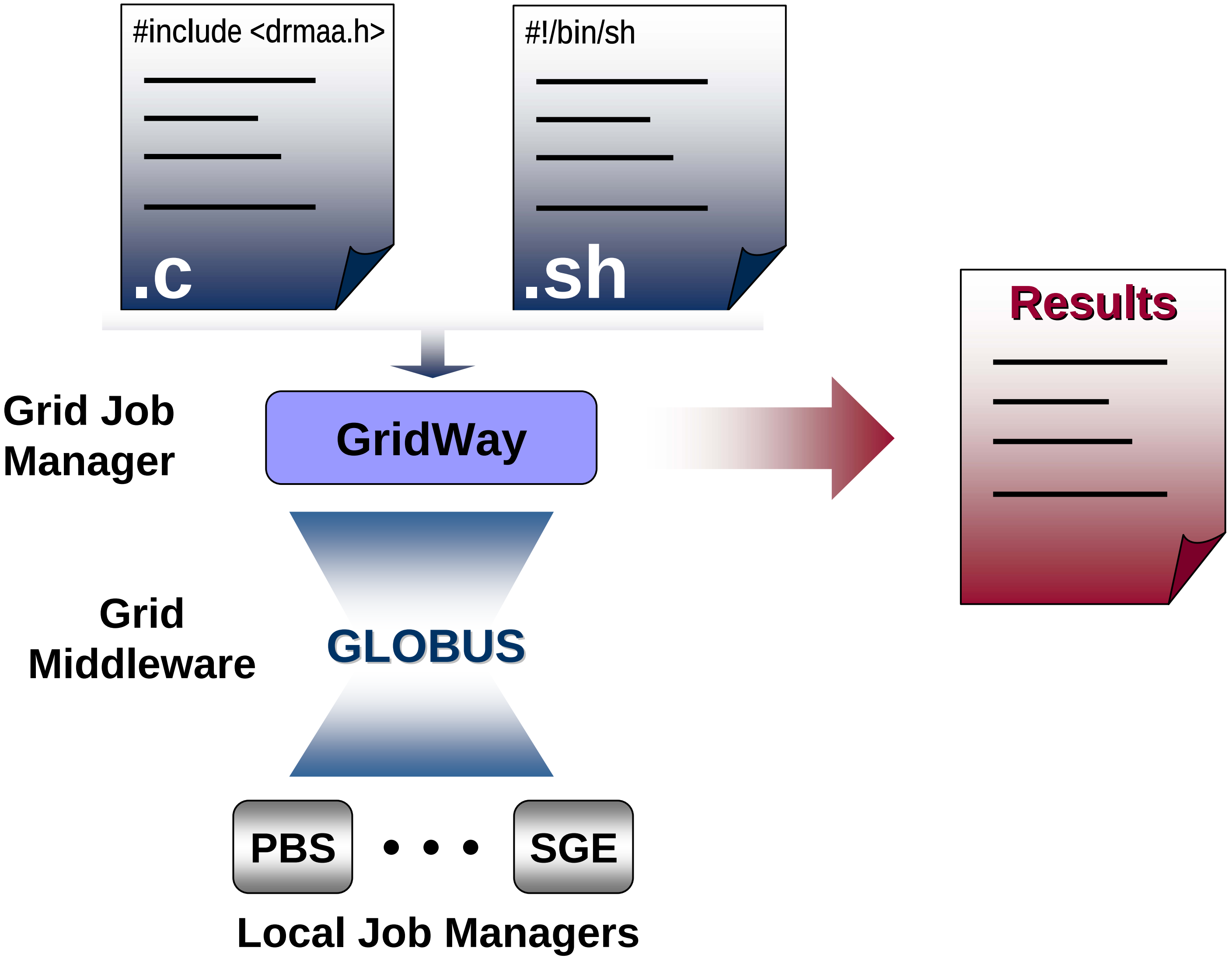
Client API

Handles **job submission**, **monitoring** and **control**, and retrieval of **finished job status**. ***(DRMAA subset)***

Complex Application Definition



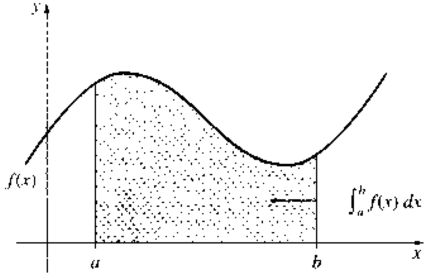
Complex Application Execution



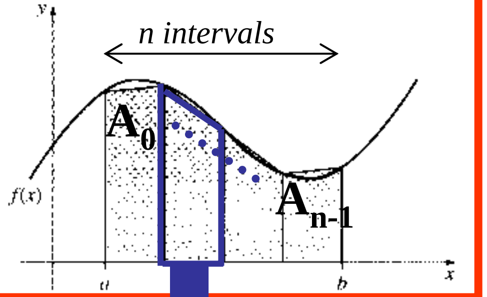
3. Sample Application: Computing π on the Grid

Numerical Approximation

$$\pi = \int_0^1 \frac{4}{1+x^2}dx$$

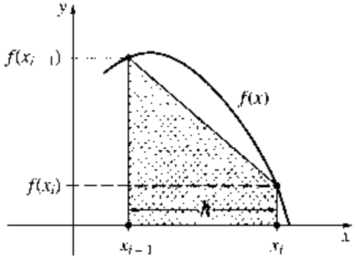


$$\pi = \sum_{i=0}^{n-1} A_i$$



Numerical Integration

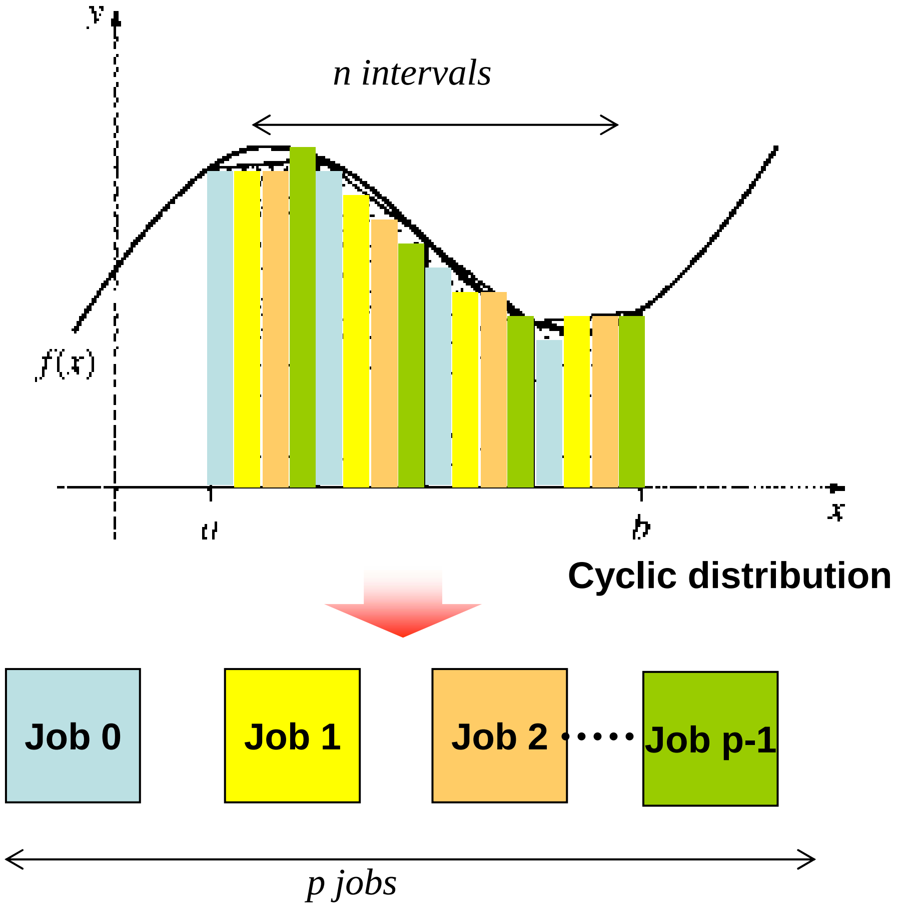
Area in Interval *i*



$$A_i = f(x_i + \frac{h}{2})h$$

3. Sample Application: Computing π on the Grid

Interval Scheduling



n intervals

Cyclic distribution

Job 0

Job 1

Job 2

Job p-1

p jobs

3. Sample Application: Computing π on the Grid

The Job

Template

```
ON_FAILURE = reschedule
EXECUTABLE_FILE=pi
EXECUTABLE_ARGUMENTS=“${GW_TASK_ID} ${GW_TOTAL_TASKS} 1000000000"
STDOUT_FILE=stdout.${GW_TASK_ID}
STDERR_FILE=stderr.${GW_TASK_ID}
RESOURCE_SELECTOR  = scripts/rs_round_robin.sh
GW_HOST_LIST=$HOME/DRMAA_Tutorial/pi_script/host.list
```

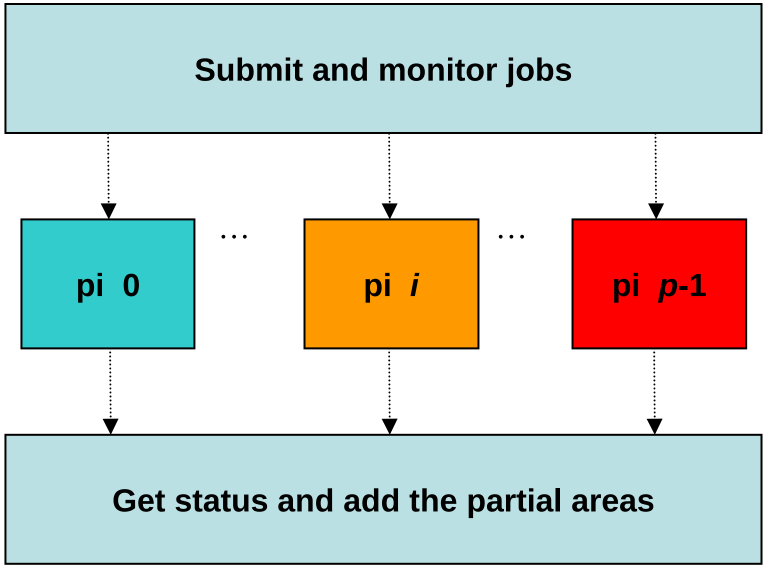
```
pi rank p n
…
int main (int nargs, char** args)
{
    …
    rank  = atoi(args[1]);
    total = atoi(args[2]);
    n     = atoi(args[3]);
        h = 1.0/n;
    l_sum = 0.;
    for (i = rank; i < n; i += total)
    {
        x = (i+0.5)*h;
        l_sum += 4.0/(1.0+x*x);
    }
    l_sum *= h;
    printf("%g\n",l_sum);
    return 0;
}
```

Command Interface

```
gwsubmit –n 20 –t job_template
```

```
gwps –cd 1
```

```
gwwait –A 0
```



```
my $i = 0;
my $content;
my $pi = 0;
while ($i < $num_of_files)
{
        open (FILE,"stdout.$i");
        $content = <FILE>;
        close (FILE);
        $pi = $pi + $content;
        $i++;
}
print "Pi is: $pi\n";
```

4. DRMAA Implementation

About DRMAA

- **Homogenous interface** to different Distributed Resource Management Systems (DRMS)
- A suitable and portable framework to **express distributed computations**.

About DRMAA for GridWay

- **Partial implementation** of the C language binding 0.9.5.
- **Missing functionality**
  - Non-blocking and timeout waits, and information retrieval functions `drmaa_w*`
  - Some required template attributes: `JOB_CATEGORY`, `START_TIME`…
  - Hold/Release states
  - All auxiliary functions
- It is **not thread-safe**

4. DRMAA Implementation

Session Handling

```
drmaa_init
drmaa_exit
```

Buffer Size Defines

```
#define DRMAA_ATTR_BUFFER   1024
#define DRMAA_ERROR_STRING_BUFFER   1024
#define DRMAA_GW_JOBID_BUFFER   5
```

Job Template Management

```
drmaa_allocate_job_template
drmaa_delete_job_template
drmaa_set_attribute
drmaa_set_vector_attribute
drmaa_get_attribute
drmaa_get_vector_attribute
```

Job Template Attributes

Job Template is created as **DRMAA_WD/DRMAA_JOB_NAME**

Required Attributes	GridWay Attributes	Job Template
		# Scheduling Variables
	DRMAA_GW_DISCOVERY_TIME	DISCOVERY_TIME = 60
	DRMAA_GW_DISCOVERY_TIMEOUT	DISCOVERY_TIMEOUT = 1200
		# Performance Variables
	DRMAA_GW_POLL_TIMEOUT	POLL_TIMEOUT = 60
	DRMAA_GW_SUSPENSION_TIMEOUT	SUSPENSION_TIMEOUT = 3000
		# Behaviour in case of failure
	DRMAA_GW_ON_FAILURE	ON_FAILURE = reschedule
	DRMAA_GW_NUMBER_OF_RETRIES	NUMBER_OF_RETRIES = 3
		# Executable Variables
DRMAA_REMOTE_COMMAND		EXECUTABLE_FILE = /bin/ls
DRMAA_V_ARGV		EXECUTABLE_ARGUMENTS = -la
	DRMAA_V_GW_INPUT_FILES	INPUT_FILES =
	DRMAA_V_GW_OUTPUT_FILES	OUTPUT_FILES =
	DRMAA_V_GW_RESTART_FILES	RESTART_FILES =
		# Standard Streams
DRMAA_INPUT_PATH		STDIN_FILE = /dev/null
DRMAA_ERROR_PATH		STDOUT_FILE = stdout_file.${GW_JOB_ID}
DRMAA_OUTPUT_PATH		STDERR_FILE = stderr_file.${GW_JOB_ID}
		# Execution Modules
	DRMAA_GW_RESOURCE_SELECTOR	RESOURCE_SELECTOR= **scripts/rs_round_robin.sh**
Those are explicitly written in the template		# Driver specific information and user-defined and module specific variables ...

4. DRMAA Implementation

Keywords in Attributed Values

```
#define DRMAA_PLACEHOLDER_INCR   "${GW_DRMAA_TASK_ID}“
#define DRMAA_GW_TASK_ID   "${GW_TASK_ID}"
#define DRMAA_GW_TOTAL_TASKS   "${GW_TOTAL_TASKS}"
#define DRMAA_GW_START_TASK   "${GW_DRMAA_START_TASK}"
#define DRMAA_GW_INCR   "${GW_DRMAA_INCR}"
#define DRMAA_GW_JOB_ID   "${GW_JOB_ID}"
#define DRMAA_GW_ARCH   "${GW_ARCH}"
#define DRMAA_GW_DEV_NULL   "/dev/null"
```

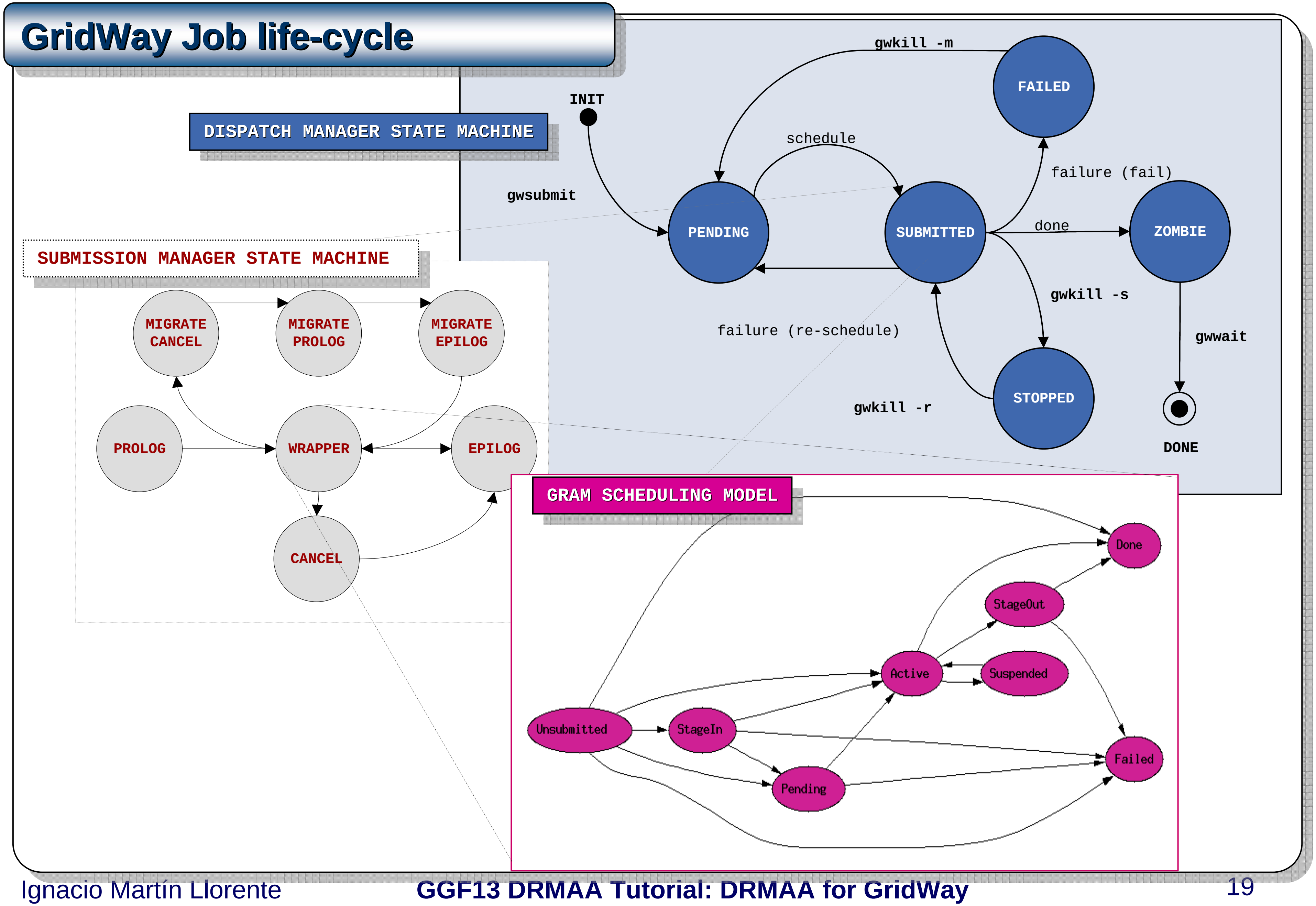
Job Submission

```
drmaa_run_job
drmaa_run_bulk_job
```

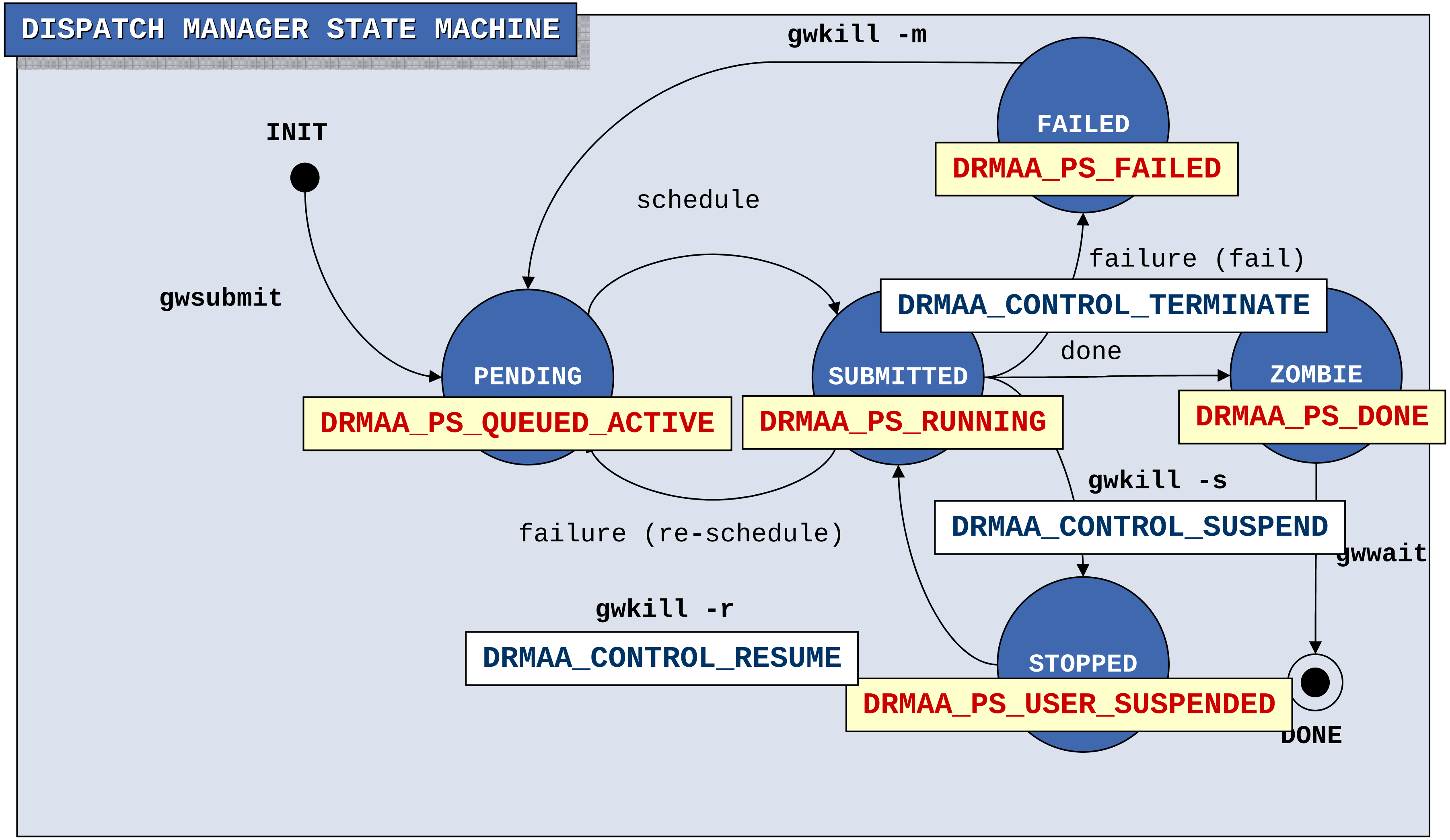
Job Control and Monitoring

```
drmaa_control
drmaa_synchronized
drmaa_job_ps
drmaa_wait
```

GridWay Job life-cycle



DRMAA Job Status and Control Actions



4. DRMAA Implementation

Other Control Operation Defines

```
#define DRMAA_TIMEOUT_WAIT_FOREVER   -1
#define DRMAA_JOB_IDS_SESSION_ALL  "DRMAA_JOB_IDS_SESSION_ALL"
#define DRMAA_JOB_IDS_SESSION_ANY  "DRMAA_JOB_IDS_SESSION_ANY"
```

Helper Functions

```
drmaa_get_next_attr_value
drmaa_get_next_job_id
drmaa_release_attr_values
drmaa_release_job_ids
```

Job Template Setup for pi Job

```
void setup_job_template( drmaa_job_template_t **jt)
{...
rc = drmaa_allocate_job_template(jt, error, DRMAA_ERROR_STRING_BUFFER);
rc = drmaa_set_attribute(*jt,DRMAA_WD,cwd, error, DRMAA_ERROR_STRING_BUFFER);
rc = drmaa_set_attribute(*jt, DRMAA_JOB_NAME,"pi_template", error, DRMAA_ERROR_STRING_BUFFER);
rc = drmaa_set_attribute(*jt, DRMAA_GW_ON_FAILURE,"reschedule", error,DRMAA_ERROR_STRING_BUFFER);
rc = drmaa_set_attribute(*jt, DRMAA_REMOTE_COMMAND,"pi", error,DRMAA_ERROR_STRING_BUFFER);
rc = drmaa_set_vector_attribute(*jt, DRMAA_V_ARGV, args, error, DRMAA_ERROR_STRING_BUFFER);
...
snprintf(attr_buffer,DRMAA_ATTR_BUFFER,"stdout."DRMAA_GW_TASK_ID);
rc = drmaa_set_attribute(*jt, DRMAA_GW_STDOUT_FILE, attr_buffer, error,DRMAA_ERROR_STRING_BUFFER);
snprintf(attr_buffer,DRMAA_ATTR_BUFFER,"stderr."DRMAA_GW_TASK_ID);
rc = drmaa_set_attribute(*jt, DRMAA_GW_STDERR_FILE, attr_buffer, error,DRMAA_ERROR_STRING_BUFFER);
snprintf(attr_buffer,DRMAA_ATTR_BUFFER,"%s/host.list",cwd);
rc = drmaa_set_attribute(*jt, "GW_HOST_LIST", attr_buffer, error, DRMAA_ERROR_STRING_BUFFER);
...}
```

/home/nacho/DRMAA_Tutorial/pi_DRMAA/pi_template

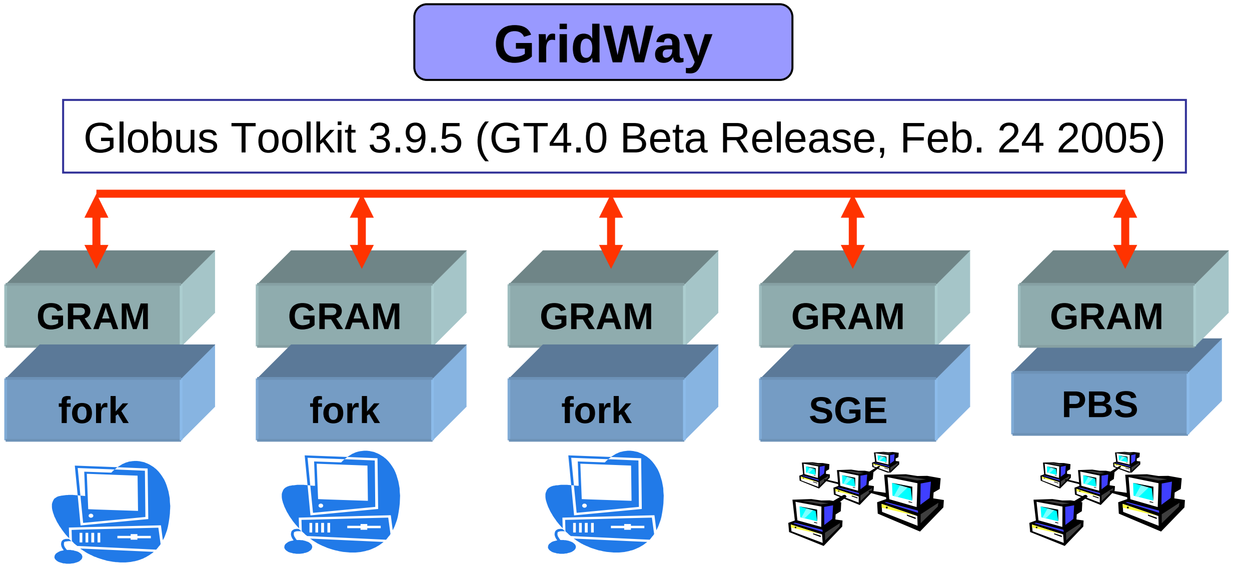
```
#This file was automatically generated by the DRMAA library
GW_DRMAA_START_TASK=0
GW_DRMAA_INCR=1
GW_DRMAA_TASK_ID="`expr ${GW_DRMAA_START_TASK} + ${GW_DRMAA_INCR} \* ${GW_TASK_ID}`"
WORKING_DIRECTORY=/home/nacho/DRMAA_Tutorial/pi_DRMAA
JOB_NAME=pi_template
ON_FAILURE=reschedule
EXECUTABLE_FILE=pi
EXECUTABLE_ARGUMENTS="${GW_TASK_ID} ${GW_TOTAL_TASKS} 1000000000"
STDOUT_FILE=stdout.${GW_TASK_ID}
STDERR_FILE=stderr.${GW_TASK_ID}
GW_HOST_LIST=/home/nacho/DRMAA_Tutorial/pi_DRMAA/host.list
```

The DRMAA Code for Computing π on the Grid

```
rc = drmaa_init(NULL, error, DRMAA_ERROR_STRING_BUFFER-1);
setup_job_template(&jt);

rc =drmaa_run_bulk_jobs(&jobids,jt,0,end,1,error,DRMAA_ERROR_STRING_BUFFER);
fprintf(stderr,"Bulk job successfully submitted IDs are:\n");
do {rc = drmaa_get_next_job_id(jobids, value, DRMAA_ATTR_BUFFER);
}while (rc == DRMAA_ERRNO_SUCCESS );

fprintf(stderr,"Waiting for bulk job to finish... \n");
rc = drmaa_synchronize(job_ids,DRMAA_TIMEOUT_WAIT_FOREVER,0,error,DRMAA_ERROR_STRING_BUFFER);
fprintf(stderr,"All Jobs finished \n");

pi = 0.0;
for (i=0; i < end + 1 ; i++)
{
  snprintf(attr_buffer,DRMAA_ATTR_BUFFER,"stdout.%i",i);
  fp = fopen(attr_buffer,"r"); fscanf(fp,"%f",&pi_t);
  fprintf(stderr,"Partial computed by task %i = %f \n",i,pi_t);
  fclose(fp); pi += pi_t;
}
fprintf(stderr,"\nPI=%f\n",pi);

for (i=0; i < end + 1 ; i++)
{
  rc = drmaa_get_next_job_id(jobids, value, DRMAA_ATTR_BUFFER);
  drmaa_wait(value, NULL, 0, &stat, DRMAA_TIMEOUT_WAIT_FOREVER, &rusage, error, DRMAA_ERROR_STRING_BUFFER);
  fprintf(stderr,"Rusage for task %i (exit code %i)\n", i, stat);
  do {
    rc = drmaa_get_next_attr_value(rusage, value, DRMAA_ATTR_BUFFER);
    if ( rc == DRMAA_ERRNO_SUCCESS ) fprintf(stderr,"\t%s\n", value);
  }while (rc == DRMAA_ERRNO_SUCCESS );
}

drmaa_exit(NULL, 0);
```

A Research TestBed

					DRMS	
Name	**Processor**	**Speed**	**O.S.**	**Nodes**	**pre-WS**	**WS**
ursa	Intel P4 HT	3.2 Ghz	Linux 2.4	1	fork	fork
draco	Intel P4 HT	3.2 Ghz	Linux 2.4	1	fork	fork
cygnus	Intel P4	2.5 Ghz	Linux 2.4	1	fork	fork
hydrus	Intel P4 HT	3.2 Ghz	Linux 2.4	4	PBS	PBS
aquila	Intel P3	600 Mhz	Linux 2.4	2	**SGE**	**fork**



GridWay

Globus Toolkit 3.9.5 (GT4.0 Beta Release, Feb. 24 2005)

DRMAA Demonstration on pre-WS Components

Resource Selector: Round Robin
Number of Jobs: 20
Number of Intervals: 10^9
Execution Time on hydrus: 12 minutes

Execution Time on the Testbed: 3 minutes

```
gwps –cd 1
```

```
JID AID TID DM   SM   EM   STIME    ETIME    EXETIME XFRTIME EXIT TEMPLATE       HOST
0   0   0   zomb done done 13:25:41 13:26:58 0:00:22 0:00:28 0    pi_template    draco.dacya.ucm.es
1   0   1   zomb done done 13:25:41 13:26:59 0:00:24 0:00:27 0    pi_template    ursa.dacya.ucm.es
2   0   2   zomb done done 13:25:41 13:27:14 0:00:38 0:00:28 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
3   0   3   zomb done done 13:25:41 13:27:04 0:00:26 0:00:30 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
4   0   4   zomb done done 13:25:41 13:27:02 0:00:25 0:00:29 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
5   0   5   zomb done done 13:25:41 13:27:03 0:00:25 0:00:30 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
6   0   6   zomb done done 13:25:41 13:27:20 0:00:38 0:00:34 0    pi_template    cygnus.dacya.ucm.es
7   0   7   zomb done done 13:25:41 13:28:14 0:01:33 0:00:33 0    pi_template    aquila.dacya.ucm.es/jobmanager-sge
8   0   8   zomb done done 13:25:41 13:27:58 0:01:21 0:00:29 0    pi_template    aquila.dacya.ucm.es/jobmanager-sge
9   0   9   zomb done done 13:25:41 13:27:54 0:00:21 0:00:25 0    pi_template    draco.dacya.ucm.es
10  0   10  zomb done done 13:25:41 13:27:54 0:00:22 0:00:24 0    pi_template    ursa.dacya.ucm.es
11  0   11  zomb done done 13:25:41 13:28:00 0:00:24 0:00:28 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
12  0   12  zomb done done 13:25:41 13:27:59 0:00:23 0:00:28 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
13  0   13  zomb done done 13:25:41 13:27:57 0:00:24 0:00:25 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
14  0   14  zomb done done 13:25:41 13:28:28 0:00:22 0:00:28 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
15  0   15  zomb done done 13:25:41 13:28:39 0:00:32 0:00:29 0    pi_template    cygnus.dacya.ucm.es
16  0   16  zomb done done 13:25:41 13:28:52 0:00:22 0:00:22 0    pi_template    draco.dacya.ucm.es
17  0   17  zomb done done 13:25:41 13:28:53 0:00:22 0:00:23 0    pi_template    ursa.dacya.ucm.es
18  0   18  zomb done done 13:25:41 13:28:56 0:00:24 0:00:24 0    pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
19  0   19  subm eplg actv 13:25:41 --:--:-- 0:00:23 0:00:25 --   pi_template    hydrus.dacya.ucm.es/jobmanager-pbs
```

DRMAA Demonstration on WS Components

Resource Selector: Round Robin
Number of Jobs: 20
Number of Intervals: 10^9
Execution Time on hydrus: 12 minutes

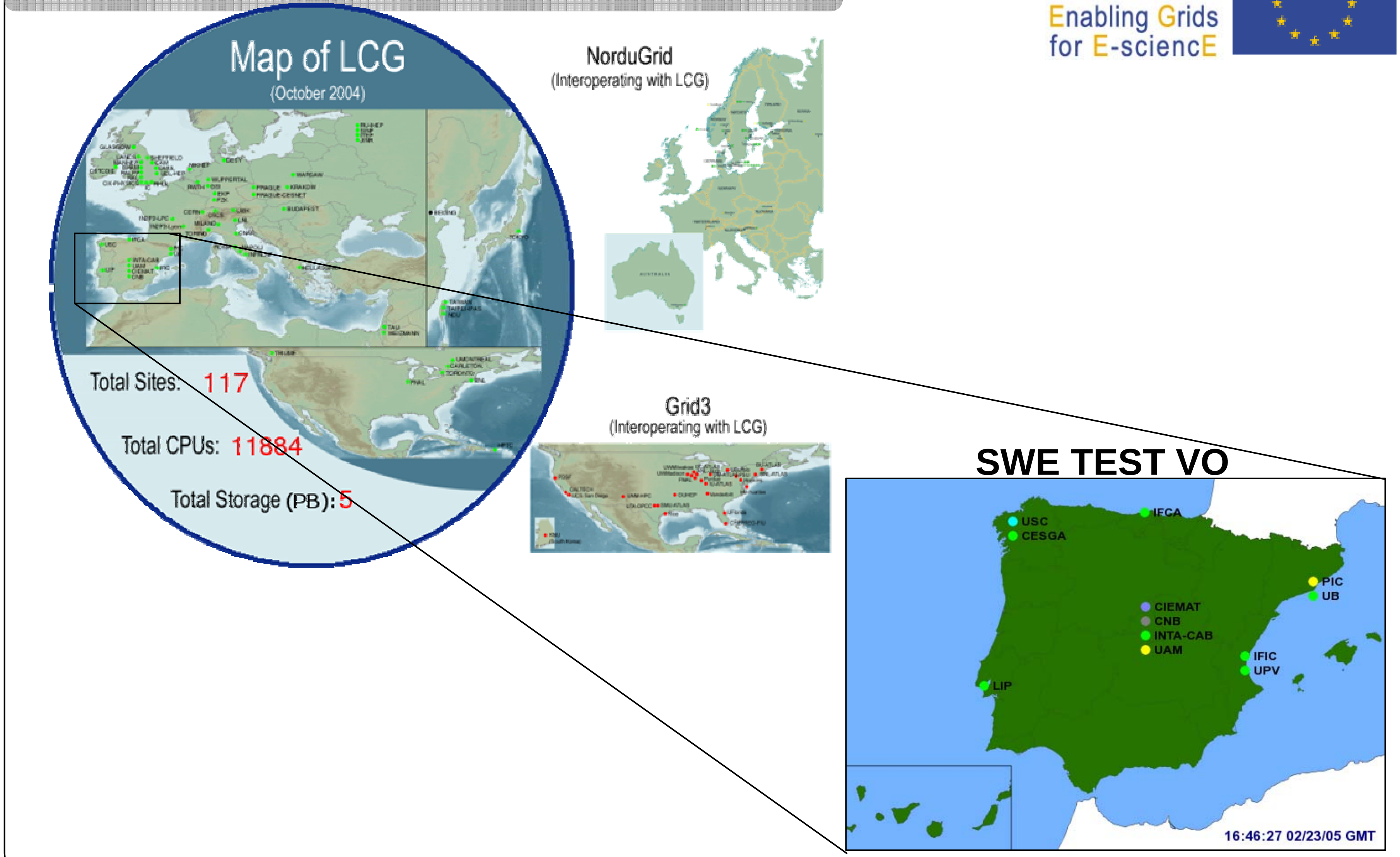
Execution Time on the Testbed: 7 minutes
(fewer resources than pre-WS execution)

```
gwps –cd 1
```

```
JID AID TID DM   SM   EM   STIME    ETIME    EXETIME XFRTIME EXIT TEMPLATE        HOST
0   0   0   zomb done done 20:16:14 20:18:08 0:00:42 0:00:55 0    pi_template     draco.dacya.ucm.es
1   0   1   zomb done done 20:16:14 20:18:13 0:00:37 0:01:05 0    pi_template     ursa.dacya.ucm.es
2   0   2   subm wrap actv 20:16:14 --:--:-- 0:06:33 0:00:56 --   pi_template     hydrus.dacya.ucm.es/PBS
3   0   3   zomb done done 20:16:14 20:18:50 0:00:50 0:01:29 0    pi_template     hydrus.dacya.ucm[illegible]
4   0   4   zomb done done 20:16:14 20:18:50 0:00:50 0:01:28 0    pi_template     hydrus.d[illegible]
5   0   5   zomb done done 20:16:14 20:18:50 0:00:54 0:01:25 0    pi_template     [illegible]
6   0   6   zomb done done 20:16:14 20:18:36 0:00:46 0:01:19 0    pi_templ[illegible]
7   0   7   zomb done done 20:16:14 20:20:14 0:01:54 0:01:49 0    [illegible]a.ucm.es
8   0   8   zomb done done 20:16:14 20:20:37 0:00:34 0:01[illegible]aco.dacya.ucm.es
9   0   9   zomb done done 20:16:14 20:19:51 0:00[illegible]  ursa.dacya.ucm.es
10  0   10  zomb done done 20:16:14 20:20[illegible]template     hydrus.dacya.ucm.es/PBS
11  0   11  zomb done done 20:16[illegible]0    pi_template     hydrus.dacya.ucm.es/PBS
12  0   12  zomb done do[illegible]0:01:02 0    pi_template     hydrus.dacya.ucm.es/PBS
13  0   13  zomb[illegible]0:00:41 0:00:52 0    pi_template     cygnus.dacya.ucm.es
14  0   1[illegible]20:21:13 0:00:35 0:00:37 0    pi_template     ursa.dacya.ucm.es
1[illegible]20:16:14 20:23:11 0:01:31 0:01:07 0    pi_template     aquila.dacya.ucm.es
1[illegible]ne done 20:16:14 20:22:19 0:00:36 0:00:42 0    pi_template     draco.dacya.ucm.es
17[illegible]zomb done done 20:16:14 20:22:26 0:00:36 0:00:49 0    pi_template     hydrus.dacya.ucm.es/PBS
18  0   18  zomb done done 20:16:14 20:22:41 0:00:33 0:01:04 0    pi_template     ursa.dacya.ucm.es
19  0   19  zomb done done 20:16:14 20:23:25 0:00:50 0:01:13 0    pi_template     cygnus.dacya.ucm.es
```

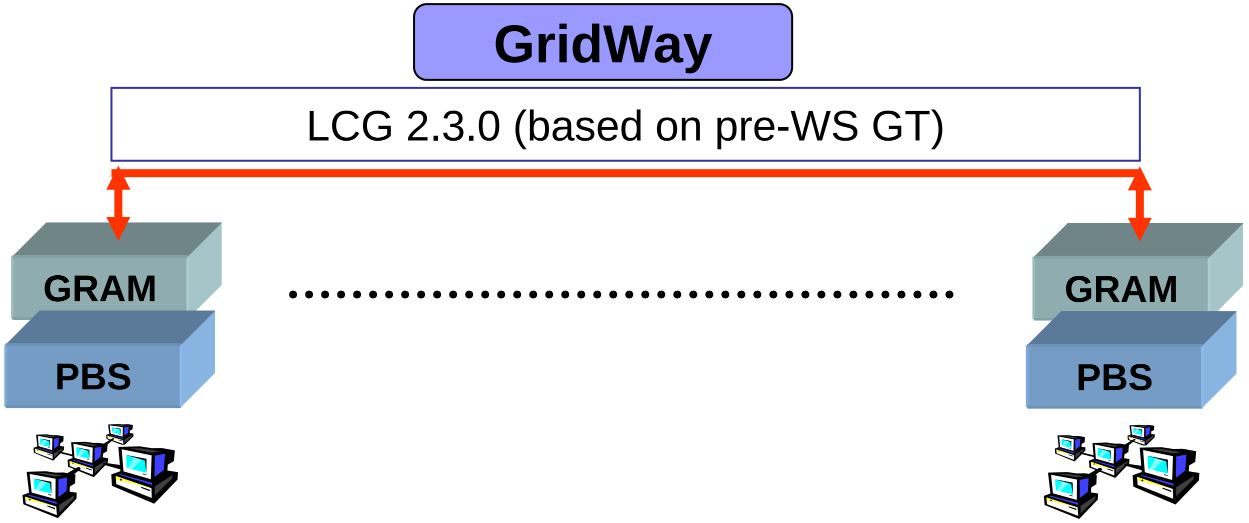
GridWay is able to simultaneously harness pre-WS and WS

A Production TestBed



A Production TestBed

Name	Site	Node	Processor	Speed	O.S.	Nodes	DRMS
ramses	UPV	Valencia	2 x Intel PIII	900 Mhz	Linux 2.4	12	PBS
ce00	CAB-INTA	Madrid	Intel P4 HT	2.8 Ghz	Linux 2.4	8	PBS
mallarme	CNB	Madrid	4 x Xeon	2.0 Ghz	Linux 2.4	16	PBS
ce2	CESGA	Santiago	2 x Intel PIII	1.0 Ghz	Linux 2.4	6	PBS
ce01	LIP	Lisboa	2 x Intel PIII	800 Mhz	Linux 2.4	8	PBS
gtbcg12	IFCA	Santander	2 x Intel PIII	1.2 Ghz	Linux 2.4	34	PBS
lcg2ce	IFIC	Valencia	AMD Athlon	1.2 Ghz	Linux 2.4	117	PBS
ce01	PIC	Barcelona	Intel P4 HT	3.4 Ghz	Linux 2.4	20	PBS



GridWay

LCG 2.3.0 (based on pre-WS GT)

GRAM

PBS

GRAM

PBS

DRMAA Demonstration on a Production Testbed

Resource Selector: Round Robin
Number of Jobs: 20
Number of Intervals: 10^9
Execution Time on `hydrus`: 12 minutes

Execution Time on the Testbed: 6 minutes

```
gwps –cd 1
```

```
JID AID TID DM   SM   EM   STIME    ETIME    EXETIME XFRTIME EXIT TEMPLATE        HOST
40  2   0   zomb done done 16:01:52 16:05:02 0:01:57 0:00:50 0    pi_template     ce00.inta.es/jobmanager-torque
41  2   1   zomb done done 16:01:52 16:05:00 0:01:36 0:01:09 0    pi_template     ce00.inta.es/jobmanager-torque
42  2   2   zomb done done 16:01:52 16:05:04 0:01:48 0:01:01 0    pi_template     ce00.inta.es/jobmanager-torque
43  2   3   zomb done done 16:01:52 16:05:04 0:01:30 0:01:19 0    pi_template     ce00.inta.es/jobmanager-torque
44  2   4   zomb done done 16:01:52 16:05:49 0:02:39 0:00:55 0    pi_template     lcg2ce.ific.uv.es/jobmanager-lcgpbs
45  2   5   zomb done done 16:01:52 16:05:56 0:02:42 0:00:59 0    pi_template     lcg2ce.ific.uv.es/jobmanager-lcgpbs
46  2   6   zomb done done 16:01:52 16:05:56 0:02:45 0:00:56 0    pi_template     lcg2ce.ific.uv.es/jobmanager-lcgpbs
47  2   7   zomb done done 16:01:52 16:05:53 0:02:45 0:00:53 0    pi_template     lcg2ce.ific.uv.es/jobmanager-lcgpbs
48  2   8   zomb done done 16:01:52 16:04:11 0:01:08 0:00:48 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
49  2   9   zomb done done 16:01:52 16:03:58 0:00:57 0:00:46 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
50  2   10  zomb done done 16:01:52 16:04:19 0:01:06 0:00:58 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
51  2   11  zomb done done 16:01:52 16:04:04 0:00:52 0:00:57 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
52  2   12  zomb done done 16:01:52 16:05:58 0:01:00 0:00:43 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
53  2   13  zomb done done 16:01:52 16:05:57 0:01:00 0:00:42 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
54  2   14  zomb done done 16:01:52 16:05:58 0:01:03 0:00:40 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
55  2   15  zomb done done 16:01:52 16:06:28 0:01:04 0:00:39 0    pi_template     ramses.dsic.upv.es/jobmanager-torque
56  2   16  zomb done done 16:01:52 16:07:59 0:01:54 0:00:50 0    pi_template     ce00.inta.es/jobmanager-torque
57  2   17  zomb done done 16:01:52 16:07:56 0:01:54 0:00:47 0    pi_template     ce00.inta.es/jobmanager-torque
58  2   18  zomb done done 16:01:52 16:07:58 0:01:49 0:00:54 0    pi_template     ce00.inta.es/jobmanager-torque
59  2   19  subm eplg actv 16:01:52 --:--:-- 0:01:47 0:00:57 --   pi_template     ce00.inta.es/jobmanager-torque
```

5. Future Work

About GridWay

- Modify GridWay to provide **full DRMAA support**
- Provide a **thread-safe implementation**

About DRMAA

- DRMAA implementation for **EGEE Resource Broker**

Thank you
for your attention!

More information…
http://www.GridWay.org
llorente@dacya.ucm.es