

“A Comparative Analysis between EGEE and GridWay Workload Management Systems”

J.L. Vázquez-Poletti

E. Huedo

R.S. Montero

I.M. Llorente

**Universidad Complutense
de Madrid**

- Spain -



Index

- ⌘ What is a Grid Metascheduler?
- ⌘ Some Metaschedulers
- ⌘ GridWay vs. EGEE WMS
 - Architectures
 - Comparison
 - Experiments
 - Results
- ⌘ Conclusions

Meta...?

- ⌘ Grid Middleware
- ⌘ What?
 - Resource discovery
 - Resource evaluation
 - Resource (local schedulers) assignation to grid jobs
- ⌘ How?
 - Scheduler activity (local/cluster level) coordination

Some Metaschedulers

- ⌘ **CSF:** Round Robin based scheduling and non-advanced resource reservation
- ⌘ **Nimrod/G:** Auction mechanisms
- ⌘ **Condor-G:** Helper Mechanisms (*ClassAd* and *DAGMan*)
- ⌘ **GrADS y AppLeS:** Application and System level environments are considered

GridWay vs. EGEE WMS?

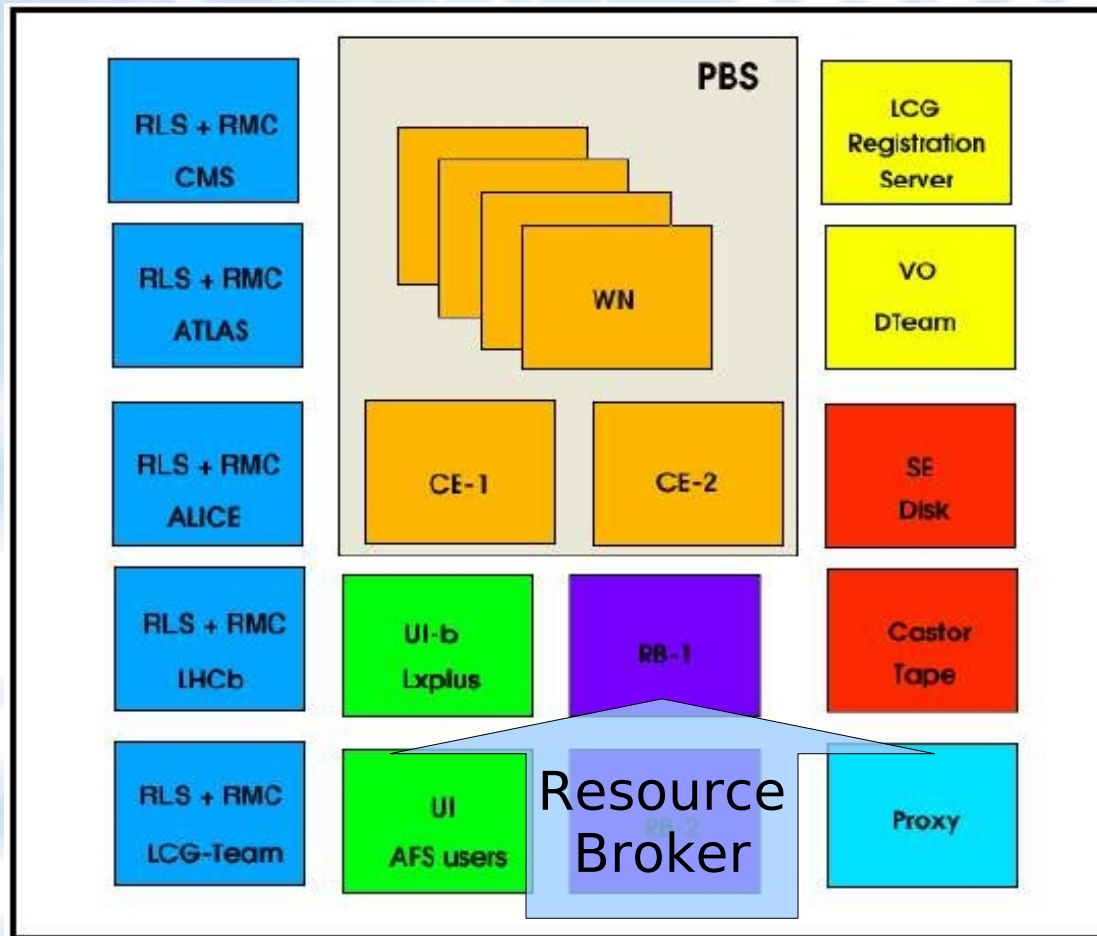
⌘ Before...

- Coordinated harnessing of EGEE and non-EGEE resources with GridWay

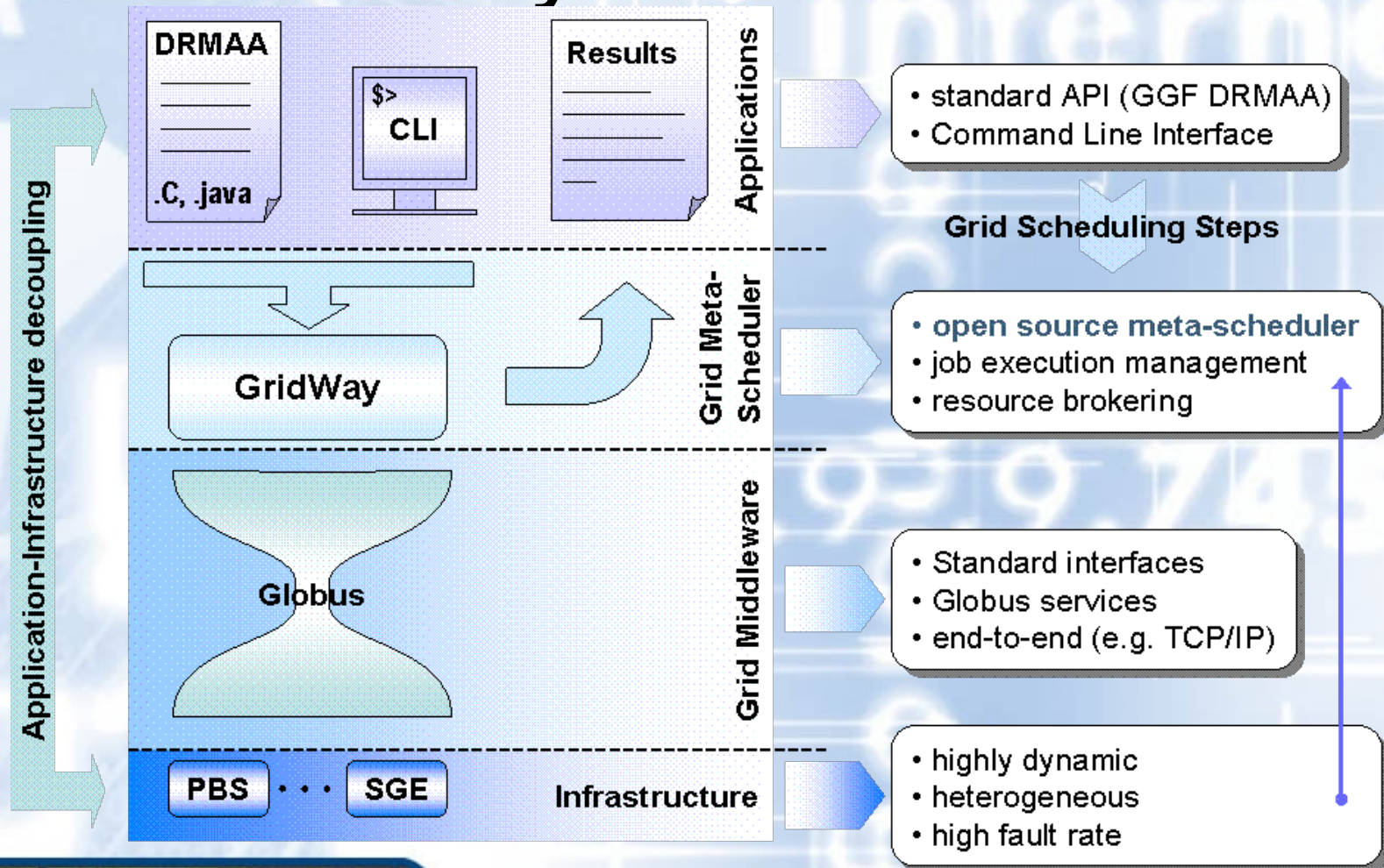
⌘ Now...

- Two philosophies comparison
 - EGEE WMS -> Middleware LCG-2
- Over EGEE resources

LCG-2 Architecture



GridWay Architecture



Scheduling Capabilities

LCG-2	GridWay
Jobs are treated in FIFO mode	
Dynamic scheduling using dynamic requirements and attributes	
Same attribute names as in Information Service	Attribute names independent from Information Service
Access only to BDII Process only GLUE schema	Access to many Information Services (using Information MADs)
Globus PreWS	Globus PreWS and WS
	Opportunistic Migration (best resource)
	Performance slowdown detection (queue waiting, less CPU than expected)
Static requirements and ranking	Dynamic requirements and ranking
Checkpointing through an API	Checkpointing must be implemented
Accounting using APEL	Accounting using Berkeley DB
Jobs dependencies (DAGMan)	Job dependencies (arguments)

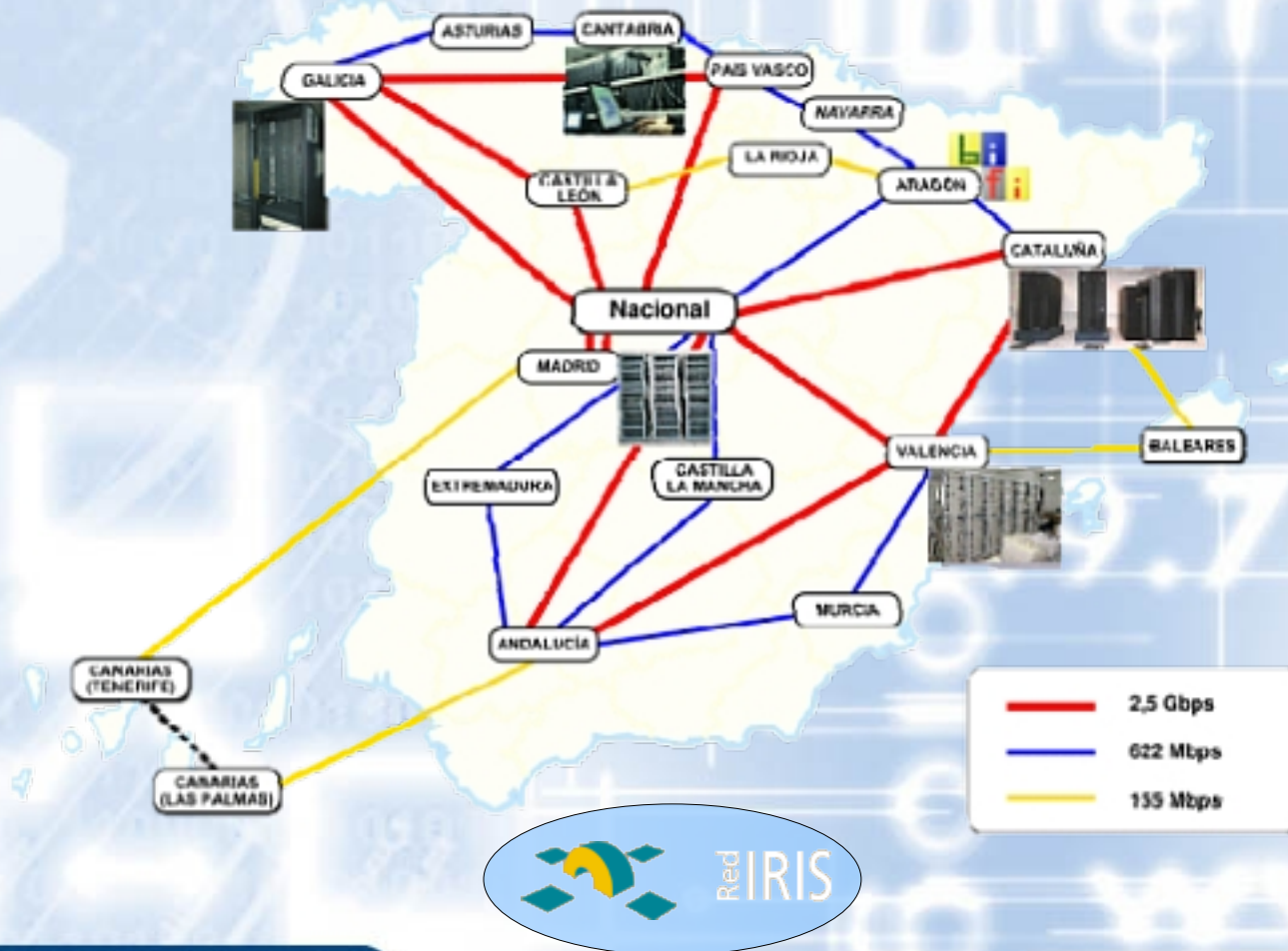
Failure Detection and Recovery

LCG-2	GridWay
Error detection (Condor-G mechanisms)	Detection of: <ul style="list-style-type: none">- Job cancelation- Remote system crash- Network disconnection
Status saved in case of metascheduler crash	

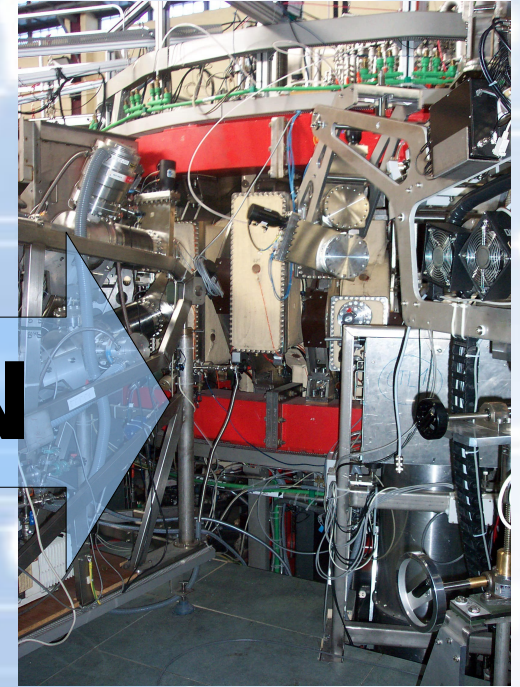
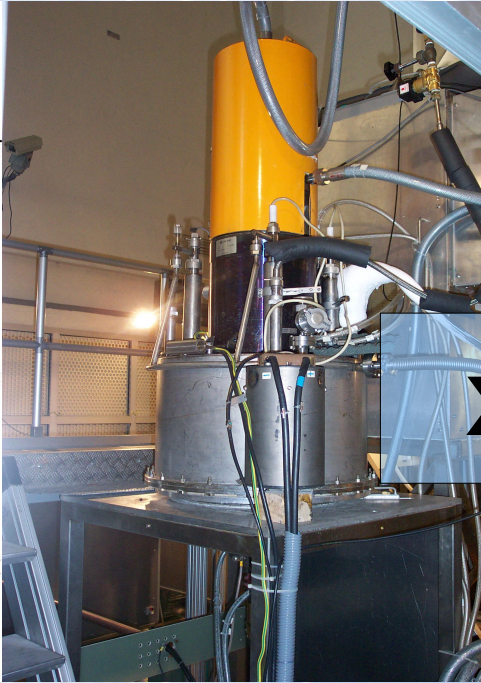
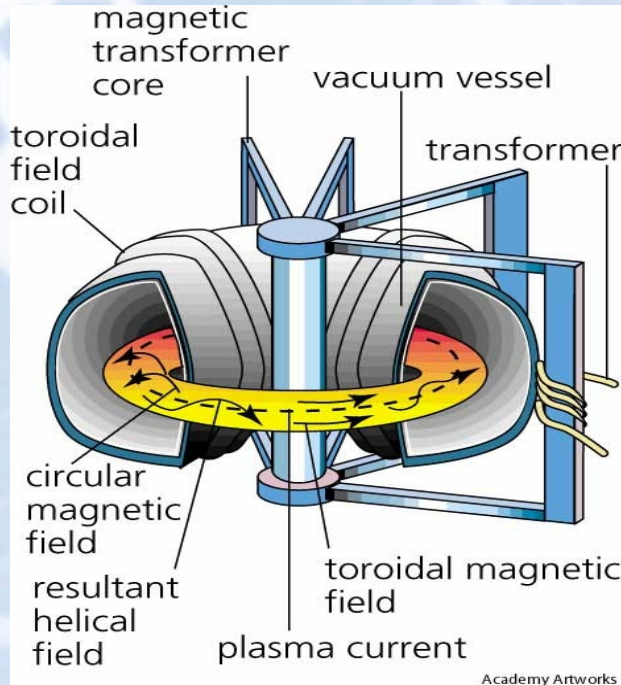
User Interface Functionality

LCG-2	GridWay
Simple jobs (also with dependencies)	
	<ul style="list-style-type: none">- Array jobs- Complex jobs
“Explicit Synchronization” (active polling)	Real Synchronization
EDG WMS API (C++ and Java)	DRMAA API (C and Java) – GGF Standard
	Command line interface similar to that found in LRMS (SGE, PBS)

Grid Infrastructure



MASSIVE RAY TRACING in Fusion Plasmas

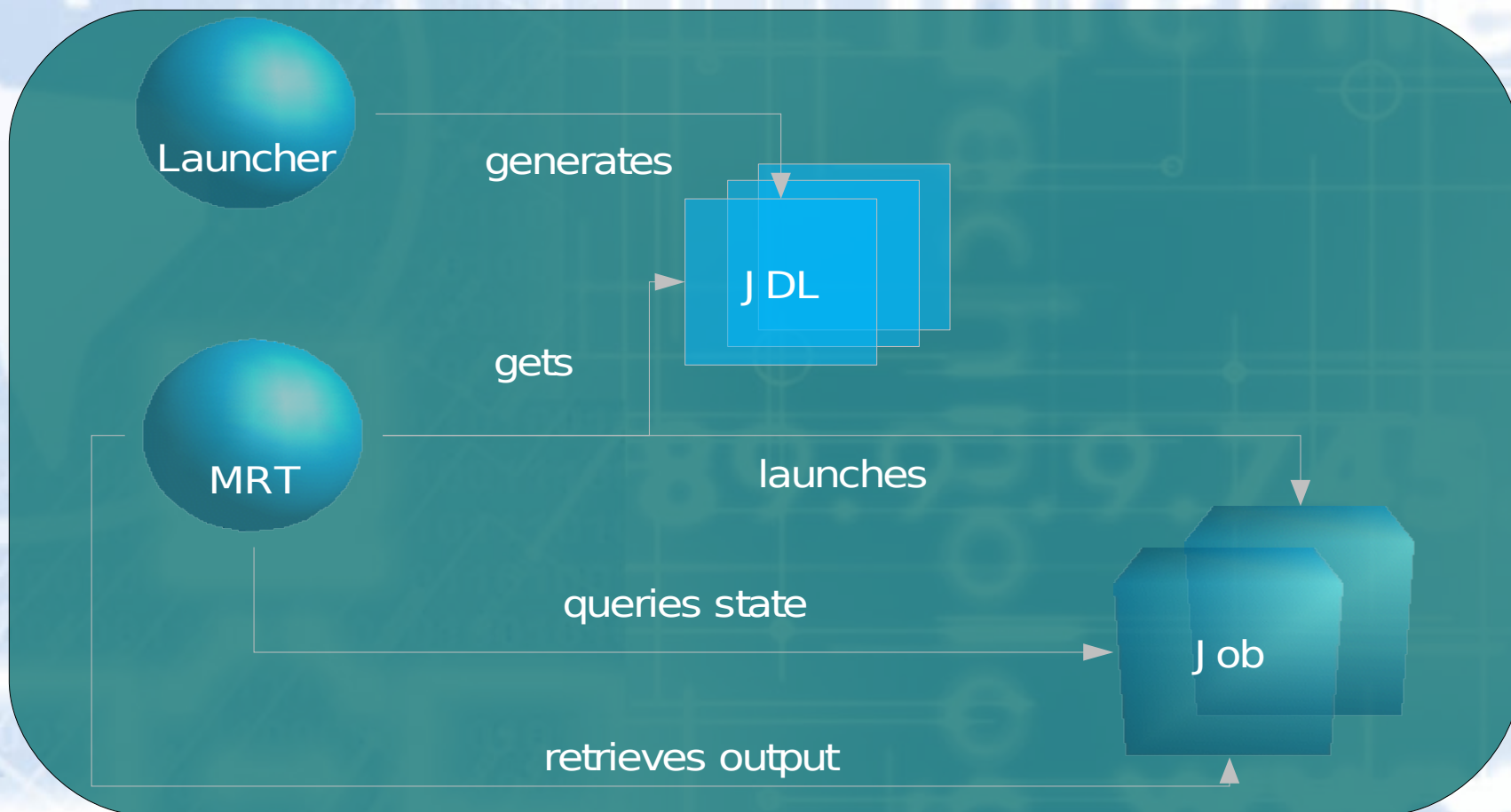


x N

- ⌘ Executable: *Truba*
 - 1,8 MB – 9' (Pentium 4 3,20 Ghz) – 50 Executions
- ⌘ Input files = ~ 70 KB
- ⌘ Output files = ~ 549 KB

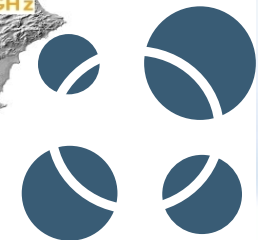
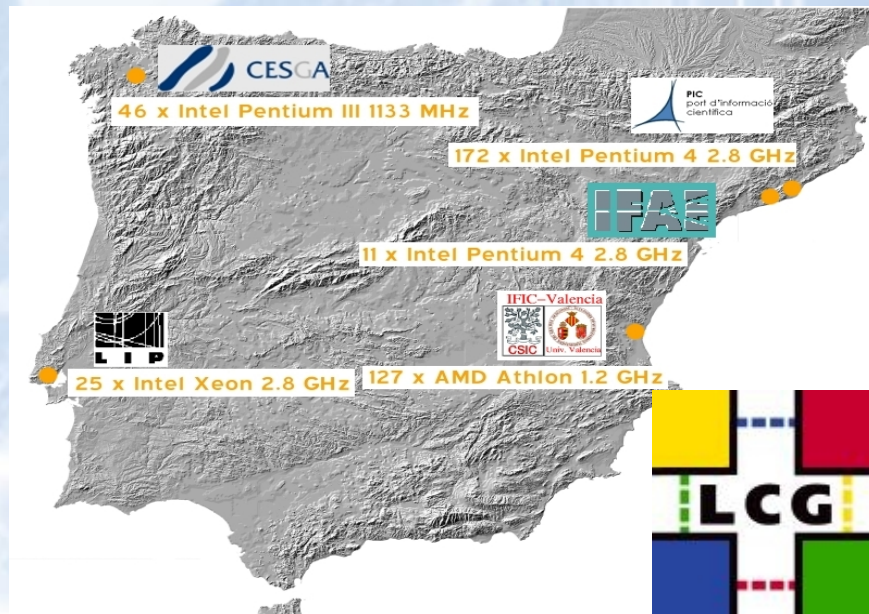
02/11/2006

lcg2.1.69 UI C++ API



Experimental Results

(minutes)	Exec./Job		Transf./Job		Total	Product. (per hour)	Overhead / Job
	Mean	Dev.	Mean	Dev.			
LCG-2	10,33	11,38	0,42	0,06	195	15,38	1,82
GridWay	36,80	16,23	0,87	0,51	120	25,00	0,52



Some Details

- ⌘ GridWay takes more advantage of available resources
 - Scheduling capabilities over dynamic resources
- ⌘ LCG-2 problems:
 - Lack of Opportunistic Migration
 - Lack of Performance Slowdown Detection
 - Jobs assigned to busy resources

Level of Parallelism

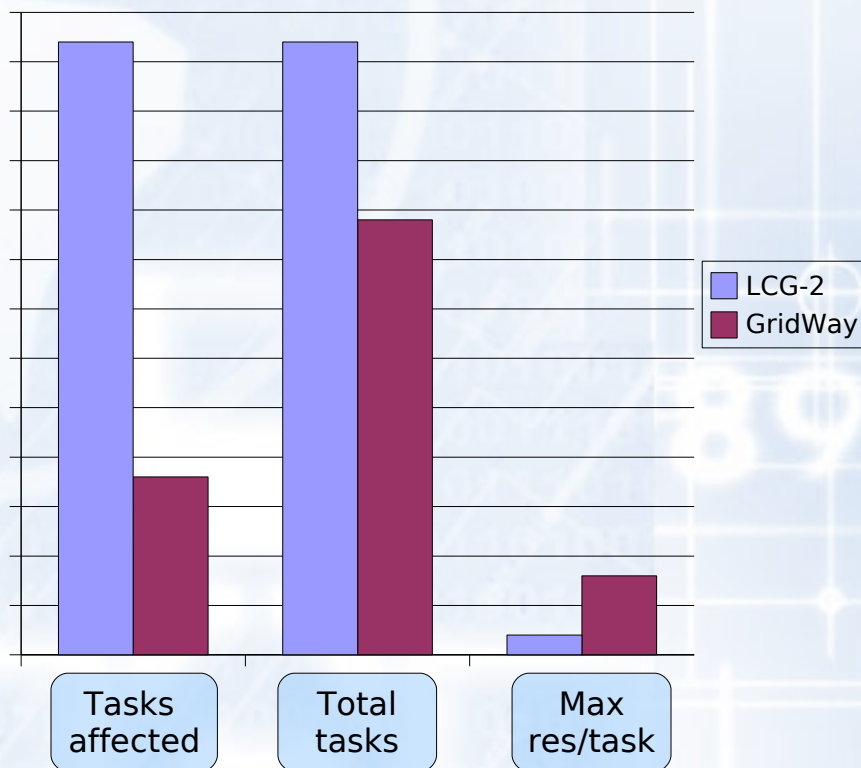
$$U = \frac{T_{exe}}{T}$$

- ⌘ T_{exe} : Sum of all execution times
- ⌘ T : Total time

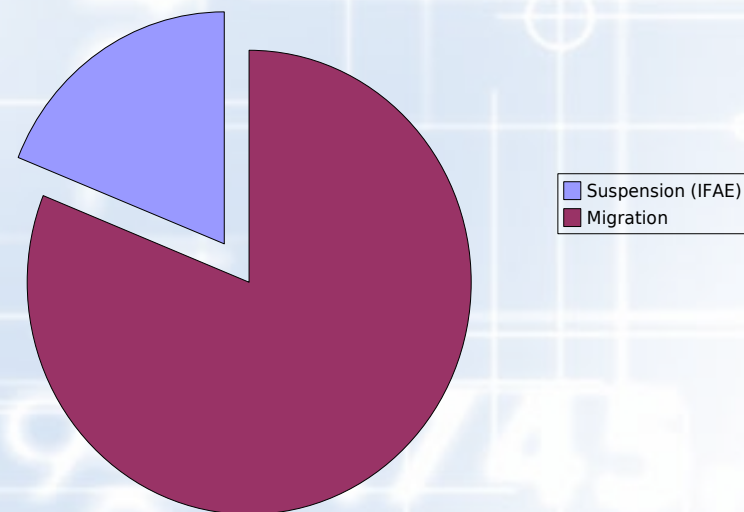
GridWay	14,91
LCG-2	6,89

As real as...

Resubmissions / Reschedules



GridWay: Rescheduling reasons



During the experiment with GridWay 1 task failed (and was rescheduled)

Performance Analysis

- ⌘ r_{∞} : **Asymptotic performance** – Maximum rate of performance (Tasks/s)
- ⌘ $n_{1/2}$: **Half-performance length** – Number of required to obtain half of the performance

$$n(t) = r_{\infty} t - n_{1/2}$$

System performance

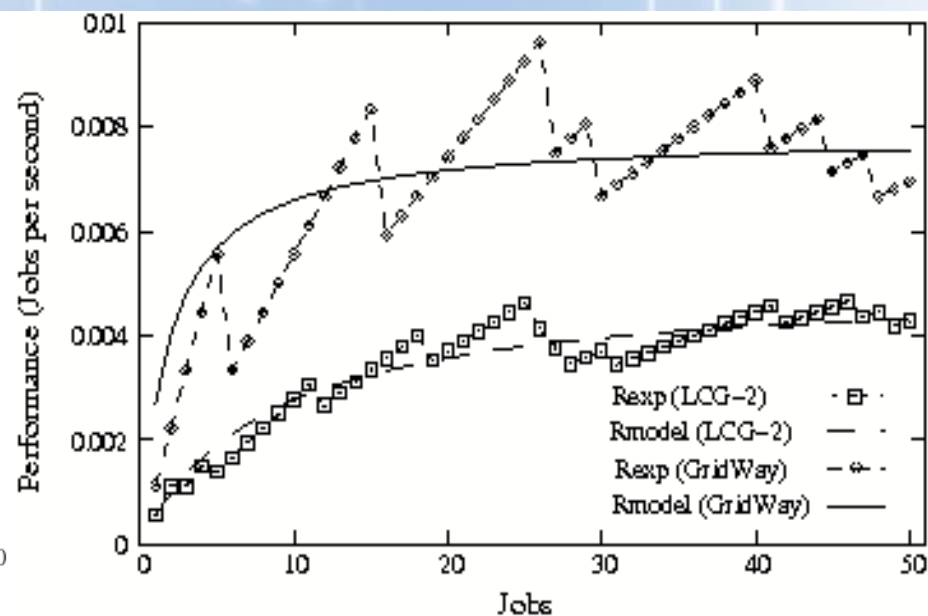
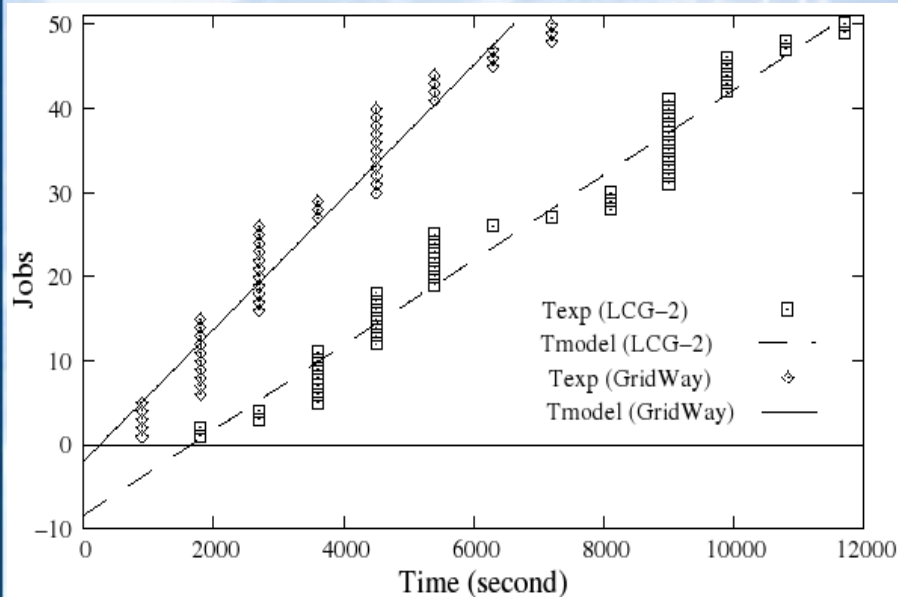
Linear Model

$$r(n) = n(t)/t = \frac{r_{\infty}}{1 + n_{1/2}/n}$$

n = number of tasks

Performance Values

	r_{∞}	$n_{1/2}$
LCG-2	0,0051 Tasks/s (18,19 Tasks/h)	8,33
GridWay	0,0079 Tasks/s (28,26 Tasks/h)	1,92



So?

- ⌘ GridWay fits in the LCG-2 picture
 - High productivity with EGEE resources
 - Additional mechanisms:
 - Opportunistic migration
 - Performance slowdown detection
- ⌘ And... LCG-2 has other interesting components!

Thank you very much!



02/11/2006