MPI and Parallel Code Support

Alessandro Costantini,

Isabel Campos, Enol Fernández, Antonio Laganà, John Walsh

Core Objectives

- Improved end-user documentation, addressing MPI application development and job submission in ARC, gLite and UNICORE
- Feedback from User community, NGI and site
- Outreach and dissemination at EGI events and workshops
- Participation in selected standardisation activity and task force

MPI Application / Libraries

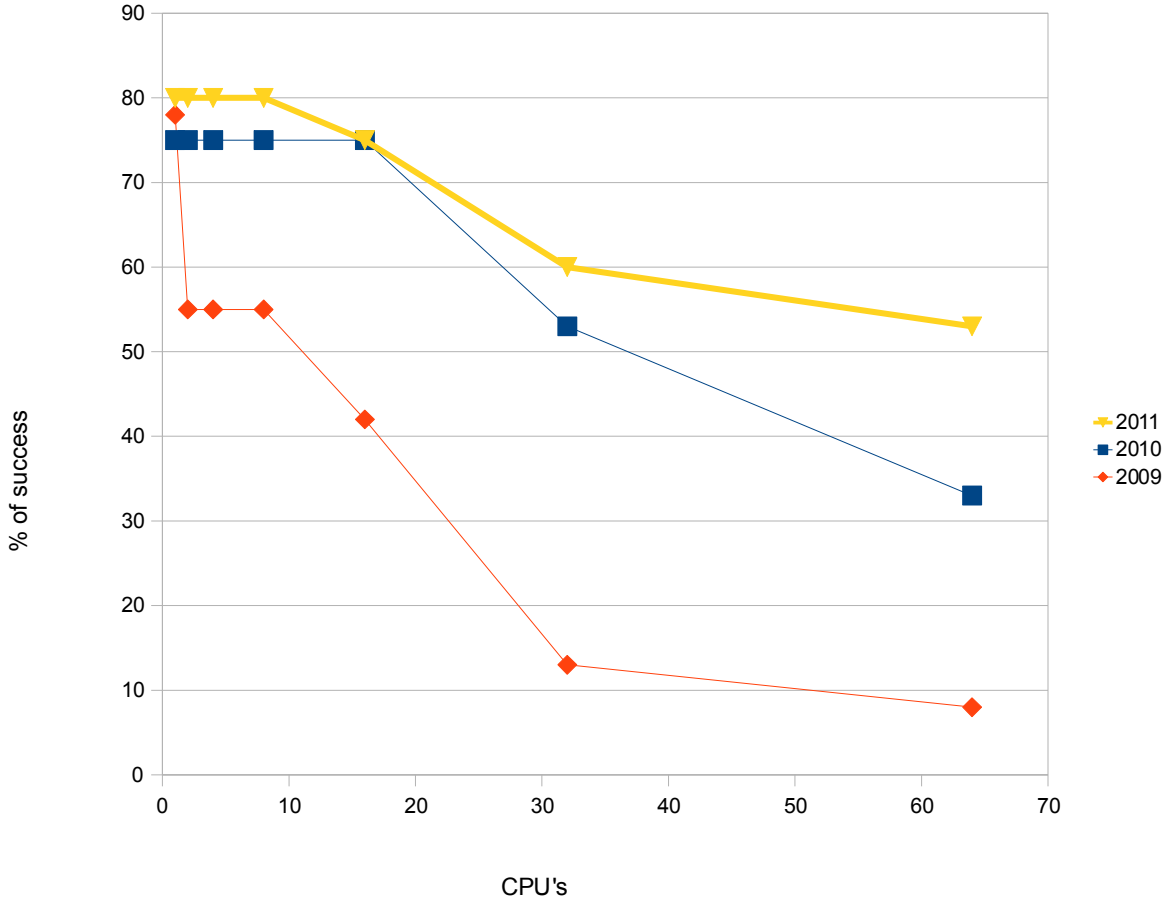
- Most Common Application / Libraries ported
  - DL_POLY
  - NAMD
  - VENUS96
  - GROMACS
  - GAMESS
  - MUFTE
  - mpiBLAST
- Easy compilation in UI (may be problems for local compilation in WN)

MPI Status

- 119 clusters publish MPI-START tag
  - Very little change since last year
  - However, big change in reliability!
    - Sites now tested every hour via SAM (NAGIOS)
    - NGIs/Sites must follow-up on MPI failures
- Compchem VO performed wide scale testing
    - Uses UNIPG production codes of DL_POLY
    - 16 sites of 25 support both CompChem and MPI
    - Tested sequentially on one node, then parallel on 2, 4, 8, 16, 32, 64 nodes

  Compiled using IFC, MPICH (static compiled on the UI)

Performances over the Grid



- SAM-MPI tests enabled
- Parallel applications run properly on 12 sites up to 16 CPUs

Success rates and issues

2 to 8 Cores

Job Status (Percent)	2009	2010	2011
Successful	53	75	80
Unsuccessful	47	25	20

Unsuccessful	2009	2010	2011
Aborted by CE	52	0	80
Scheduler Error	39	100	20
MPI-START	9	0	

16 to 64 Cores

Job Status (Percent)	2009	2010	2011
Successful	21	54	62
Unsuccessful	79	46	38

Unsuccessful	2009	2010	2011
Aborted by CE	73	0	50
Scheduler Error	23	93	0
Proxy Expired	4	7	50

- Clearly a need to isolate outstanding issues!

OpenMP/User defined allocation support

- New M/W features requested by users
  - OpenMP support added to MPI-START
  - User defined allocation of processes/node (SMP tag)
- OpenMP advantages
  - Most sites now use >= 4 cores per machine
  - OpenMP is lightweight, easy to use, fast
- Accounting issues being investigated
  - EGI Accounting Workshop (EGI-TF 2011)
  - Expected release in UMD 1.3

GP-GPU support

- CUDA/OpenCL has a steep learning curve
  - High-end units offer ECC/better precision
  - Especially double precision calculations
- Large scale growth at HPC centres
  - HPC Top 500
- Increasing number of Applications
  - Across all scientific domains

GP-GPU support

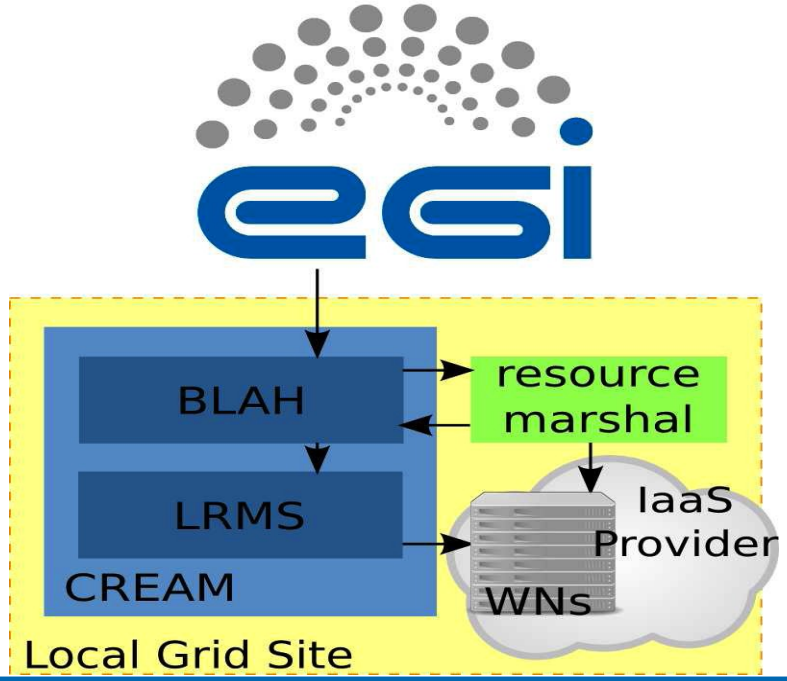
- GP-GPU resource schedulers
  - Basic support in Torque 2.5.4
  - No support in MAUI (MOAB yes)
  - SLURM supports GPGPU resources

Problems using GP-GPU

- OpenMPI (CUDA support must be explicitly configured)
- All user a/c have R/W access to resource
  - Most nodes now MultiCore
    - Multiple job slots per physical machine
    - Distinct pool a/c may access same GP-GPU
    - User code needs guaranteed exclusive access

Current GPU Scheduling Solutions

- Innovative mixed Grid/Cloud approach
  - Need Grid standardisation (#GPU cores, tags...)
- Exploits new features in:
  - WMS + H/W virtualization + PCI pass-through of GPU to VM
- Compchem & Theophys VOs



Conclusions

- SAM-MPI tests solve usual site problems
  - Easier to detect source of failure
  - MPI now more reliable for large jobs
  - Waiting times at sites can be prohibitive
    - Works best when as many free nodes at site as job size.
- Need wider deployment of UMD WMS 3.3
  - Improved generic parallel job support
- Exciting time ahead with GP-GPU/Virtualisation