THE UNIVERSITIES OF LEEDS, SHEFFIELD AND YORK

THE WHITE ROSE GRID

The White Rose Grid

Experiences and Wish List

YORKSHIRE FORWARD

Yorkshire Alive with Opportunity!

UNIVERSITY OF LEEDS

THE UNIVERSITY of York

Introduction

- Three sites – Leeds, Sheffield, York.

- e-Science Centre of Excellence.
- e-Science Test Bed (Leeds).

- Flagship project – DAME.
- Grid projects in BioInformatics area.
- Grid projects with govt. agencies.

Why SGEEE?

- A batch system needed on multiple platforms.
- Open source support meant the opportunity to look at code and contribute if possible.
- Simple to use, but flexible enough.
- SGEP/TCP support interesting.
- Good price.

Experience

- Easy to set up out of the box.
- Easy to get basic functionality to work.
- Problems:
  - Issues with a quiet system, and maxujobs, max_u_jobs, and suspension.
  - Problems with MPICH on a beowulf.
  - Some accounting irregularities (patched)
  - MPI and jobs not stopping, etc.

Queues (at York)

- Ideally want lots of short queues.
  - Easier to balance fair shares between users.
  - Easier to fill machine to capacity.
  - Use checkpointing
- But...
  - Users don't want to rewrite to support checkpointing, or don't have the source code.
  - End up needing a mix of queues.
  - Found it hard to set up interactive queues for login without breaking other tools.

Parallel Jobs

- Have had problems with jobs not accounting properly.
- Jobs not finishing properly.
- Jobs not even running (MPICH at York – possibly a York config problem and still ongoing).

Accounting

- Had some issues with double accounting
- Mostly works very well.
- Some issues with the way jobs are accounted back to a time period only on completion.

Wish List

- Queue trees.
- Per host user limits.
- Easier use of checkpointing.
- Improved management GUI.
- Separate job submission GUI for users without X.
- Visualisation of accounting information.
- Accounting functionality changes.
- Brokering tie ins – prediction, preemption, preallocation.
- Windows support.
- Ready made solutions for interoperability.

Queue Trees

- Would like to hide beowulfs as one resource, with indication of proportion used.
- Can do this with transfer queues, but this makes fair sharing difficult.
- Would like to be able to set a tree of queues.
  - User would see 'job is running on beowulf'.
  - User would see 'X of Y nodes being used'.
  - Config of tree affects all nodes in tree, etc.

Per Host User Limits

- maxujobs per host/queue tree
  - Useful in a mixed cluster.
  - Don't want to let one user dominate a machine of a particular architecture.
  - Want to allow a user to be able to run many jobs in total.
  - Want to fair share over whole system (i.e. Not use transfer queues, etc).

Checkpointing

- Users find this hard work.
  - Not all programmers.
  - Don't want to rewrite or recompile.
  - We don't have the resources to rewrite for them.
  - Want to qsub and forget.
- Kernel level checkpointing would be nice.
- Need more tutorials, help files, etc.

Management GUI

- Current tool has to be built carefully so it can work between 32 and 64 bit platforms.
- Has some idiosyncracies.
- Possible to break reliance on X, to allow more portable solutions?
  - E.g. I could tweak the config from my handheld while waiting at Munich airport.

New User GUI

- Non X (can be used easily over dialup, etc, for homeworking).
- Secure, with various forms of authentication (certs).
- Simple
- Configurable (i.e. Provide a subset of qmon, or allow users to remove those bits they don't care about)
- SGEP is hard work to install, dependencies on non-free elements.
- Could be done with a generic portal over Globus.

Visualisation of Accounting

- Nice for users to be able to see how much they have used in a web page.
- Good for funding bodies to be able to see this information.
- Tool built at York.
  - http://www.wrg.york.ac.uk/usage/stats
- Also tools being looked at by Sheffield.
- Will probably (very likely) contribute tools to OpenSource effort when working.

Accounting Functionality

- Currently a job is accounted to the period it was started in when it was finished.
- With long queues it makes creating monthly figures difficult.
- Need a true snapshot of a time period.
- Could use UNIX accounting – but no support for projects without Solaris 9+ only extensions.

Resource Brokering Tools

- Prediction
  - Handy for users to know too
  - Would allow users to request overrides.
- Preallocation
  - For workflows, especially interactively
- Preemption
  - For high priority tasks
  - Tie in to accounting, using override tickets?
  - Need checkpointing working perfectly

Windows Support

- Would allow windows-based resouces to be tied in.
- Some people want to use windows applications that don't run under wine, etc.
- Don't want to have to run Vmware etc to support Windows over Linux if possible.

More Integration

- E.g. Integrate SGEEE with:
  - Resources running Condor.
  - Resources running Mosix.
  - Globus (well under way).
  - Etc.
- Can conglomerate existing resources.
- Use attributes of other systems, but from a single point of entry under SGE.

Conclusion

- SGEEE mostly works well for us
  - Basic use easy for administration and users.
- Issues are (in a nutshell)
  - Simplifying things for users in some areas.
  - Extending access technologies for users.
  - Some niggles here and there – partly SGEEE, partly us.
- Very much looking forward to 6.0.