The NERC Cluster Grid

Conference or Workshop Item

Presentation


It is advisable to refer to the publisher’s version if you intend to cite from the work. See Guidance on citing.

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the End User Agreement.

www.reading.ac.uk/centaur

CentAUR
Central Archive at the University of Reading
Reading’s research outputs online
The NERC Cluster Grid

Dan Bretherton, Jon Blower and Keith Haines

Reading e-Science Centre
www.resc.reading.ac.uk
Environmental Systems Science Centre
University of Reading, UK
Outline of presentation

• What is a grid?
• Running climate models on HPC clusters belonging to other institutes
  − Climate models: Challenges for grid middleware
• G-Rex grid middleware
  − The climate scientist’s view
  − The grid administrator’s view
• The NERC Cluster Grid
Some grid related organisations

- NERC e-Science Centres
  - Reading e-Science Centre (ReSC) - http://www.resc.reading.ac.uk/
  - National Institute for Environmental e-Science (NIEeS) - http://www.niees.ac.uk/
- GridInfo: http://www.niees.ac.uk/grid_info.shtml
- e-Research South - http://www.eresearchsouth.ac.uk/
- National Grid Service (NGS) - http://www.grid-support.ac.uk/
- National e-Science Centre (NeSC) - http://www.nesc.ac.uk/
A definition of “grid”

• From the NIEeS web site:
  - [A grid] “allows sharing of computing, application, data and storage resources”.
  - “Grids...
    • cross geographic and institutional boundaries
    • lack central control
    • are dynamic
      - (computers join and leave in an uncoordinated fashion).“
Wide scope of grid computing

- From Mike Mineter's presentation at NGS Application Developer's Course, NeSC Feb '07

Diagram derived from Ian Foster's slide
Wide scope of grid computing

- From Mike Mineter's presentation at NGS Application Developer's Course, NeSC Feb '07

Diagram derived from Ian Foster's slide
Computational challenges of climate models

- Typical requirements
  - Parallel processing (MPI) with large number of processors (usually 20-100)
  - Long runs lasting several hours, sometimes days
  - Large volumes of output
  - Large number of separate output files
NEMO Ocean Model

• Main parameters of a typical 1/4° Global Assimilation run for **one year**:
  - Run with 80 processors
  - 48 hours per model year on a typical cluster

• Outputs 4 GB in 1000 separate files as diagnostics every 40 minutes

• Output for a one year run is roughly 300 GB, a total of 75000 separate files
  - But, disk quota on remote cluster is only 250 GB

• 50-year `Reanalysis` = 15 Tb
NERC climate community’s grid middleware requirements

• Background
  - Many NERC institutes have their own HPC clusters
  - Scientific collaborations benefit from sharing cluster resources
    • Scientists already doing this quite happily in traditional way

• The scientist’s grid middleware requirements:
  - Deal with problem of small disk quotas on remote clusters
  - Minimal changes to scientific work-flow scripts

• The grid administrator’s middleware requirements
  - Easy to set up and maintain
  - Minimal involvement of remote cluster administrators
G-Rex (Grid Remote Execution)

• Successor to Styx Grid Services
• “Light weight” middleware implemented in Java
  – Platform independent (but only tested on Linux)
• G-Rex server is a Web application
  – Runs inside a servlet container (only tested Apache Tomcat)
  – Allows applications to be exposed as Web services
• G-Rex client is command line program G RexRun
  – Behaves as if remote model were actually running on user's own computer
    • Remote model's output becomes output from G RexRun
    • Waits until end of model run before exiting
Deployment of a NEMO G-Rex service

Client

NEMO model setup, including source code, work-flow scripts, input data and output from all runs

Input and output via HTTP

Server

G-Rex server

Apache Tomcat

Tomcat port open to client

NEMO launch scripts and forcing data (same every run)
NEMO service: SSH tunnel instead of open port

Client

- G-Rex client
- NEMO model setup, including source code, work-flow scripts, input data and output from all runs

Server

- Apache Tomcat
- G-Rex server
- NEMO launch scripts and forcing data (same every run)

Input and output via HTTP

SSH tunnel
G-Rex features important to scientists

• Output transferred back to user during model runs
  − Job can be monitored easily
    • Defective jobs identified early – avoids wasting CPU time
  − No data transfer delay at end of run

• Files deleted from server when transfer completed
  − Minimises accumulation of model output data

• GReXRun easily incorporated into existing scripts
  − GReXRun usually replaces mpirun
  − A typical GReXRun command to run NEMO model:

```bash
grexrun.sh http://user:passwd@host:port/GReX/nemo input.tar.gz ORCA025
--drm-walltime 7:00:00 --drm-procs 81
```
Important for grid administrator - easy server installation and setup procedure:

• Installation
  - Download tarball from Sourceforge and unpack
    http://grex.svn.sourceforge.net/viewvc/grex/trunk/G-Rex
  - Download and unpack Sun Java and Apache Tomcat
  - Copy G-Rex/code/dist/G-Rex.war to Tomcat’s webapps
  - Talk to cluster’s firewall admin. (SSH tunnel or open port?)

• Setting up a service
  - Write model launch script containing `mpirun` command
  - Add a section in `GRexConfig.xml` for each service; specifies:
    (1) model launch script (2) input & output file patterns
    (3) expected and optional arguments (4) flagged options
NERC Cluster Grid

• 1600 processors in 5 clusters
  - (1) ESSC - 64 processors  (2) BAS - 160  (3) PML - 344  (4) POL – 360  
    (5) NOC - 780

• G-Rex services
  - NEMO model: build and execution services
  - NEMO utilities: Data interpolation and aggregation
  - POLCOMS model: build and execution services
  - qstat (http://lovejoy.nerc-essc.ac.uk:8080/GridPortal/Portal)
  - qdel
  - Other services – requests & suggestions welcome

• Ganglia load and performance monitoring system
  - See Web frontend: http://www.resc.rdg.ac.uk/ganglia/
Acknowledgement & Summary

• Thanks to NERC cluster admins. for interest and support of NERC Cluster Grid project

• Climate models produce lots of data
  – Usually much more than quota on other institutes’ clusters

• G-Rex grid middleware has 3 key features:
  – Transfers output during runs, deletes from server
  – GRexRun easily integrated into scientific work-flow scripts
  – Web services easy to install and maintain

• NERC Cluster Grid – 1600 procs, 5 clusters
  – G-Rex services for NEMO and POLCOMS