High Performance Computing on UK and European Grids

Jon MacLaren

Manchester Computing
University of Manchester

j.maclaren@man.ac.uk

The GRID at Manchester: Why?

CSAR service includes diverse HPC architectures

- 512-processor SGI O3000, 128-processor O2000
- 816-processor Cray T3E
- 8-processor Fujitsu VPP300
- 16-processor Compaq Alphaserver
- 8-processor SGI Intel Itanium Cluster

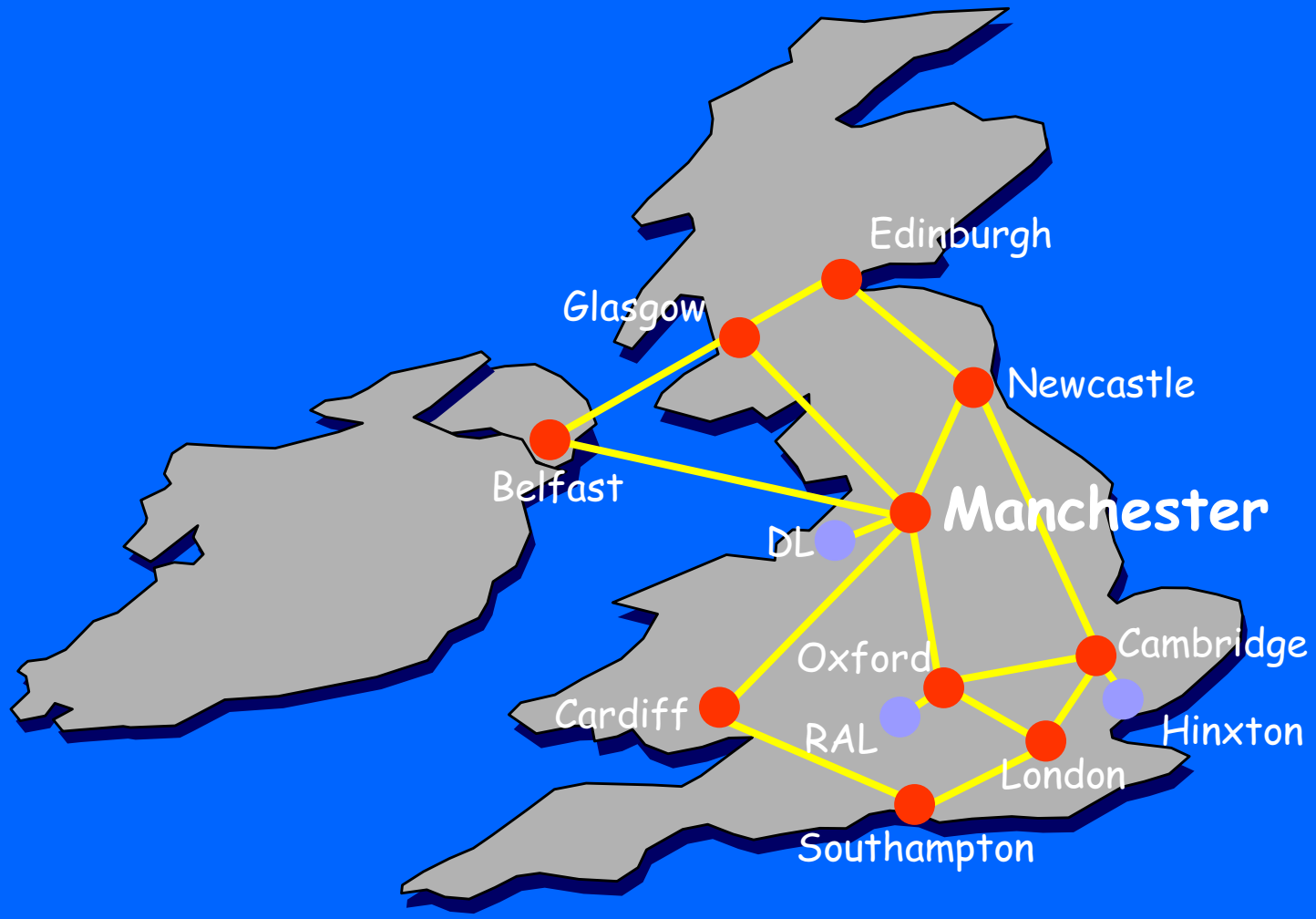
Have a user base which is divided into physically distributed project groups

Users access machines remotely, mainly with the UNIX command line – very primitive

So GRID software should make life easier for all of us…

Also, through links with Jodrell Bank, we are interested in MetaComputing – have been involved in Demonstrations at SC99 and SC2000 (and SC2001)

ESNW: E-Science North West



Current GRID Activities

ACCESS GRID

1st Access GRID node in UK

SC-Global Constellation Site

Current GRID Projects

UK funded: EPSRC e-Science Pilot Projects

- **The RealityGrid: A Tool for Investigating Condensed Matter and Materials**
  - QMW, Manchester, Edinburgh, IC, Loughborough, Oxford
- **MyGrid: Personalised Extensible Environments for Data Intensive *in silico* Experiments in Biology**
  - Manchester, EBI, Southampton, Nottingham, Newcastle, Sheffield
- **GEODISE: Grid Enabled Optimisation and Design Search for Engineering**
  - Southampton, Oxford, Manchester

EC Funded:

- **EUROGRID Project (more later…)**
- **GRIP: GRID Interoperability Project (in final negotiation)**
  - FZ-Jülich, Pallas GmbH, Manchester University, ICM Warsaw, Deutscher Wetterdienst (DWD), Southampton University, and Argonne National Laboratories

EUROGRID

Vision: Researchers in Europe should be able to access and use High Performance Computing systems from any location as easily and as securely as they use their own workstation

EUROGRID

- High Performance Computing (HPC) systems are a critical resource for scientific and industrial development
- Operation of HPC systems requires specialized centres
- European HPC centres operate a variety of HPC architectures from several vendors with frequent changes
- Users want to focus on their science rather than becoming HPC specialists
- European HPC centres and users are connected by high-bandwidth networks

EUROGRID

HPC Centres

CSCS Manno (CH)
FZ Jülich (D)
ICM Warsaw (PL)
IDRIS Paris (F)
Univ Bergen (N)
Univ Manchester (UK)

Users

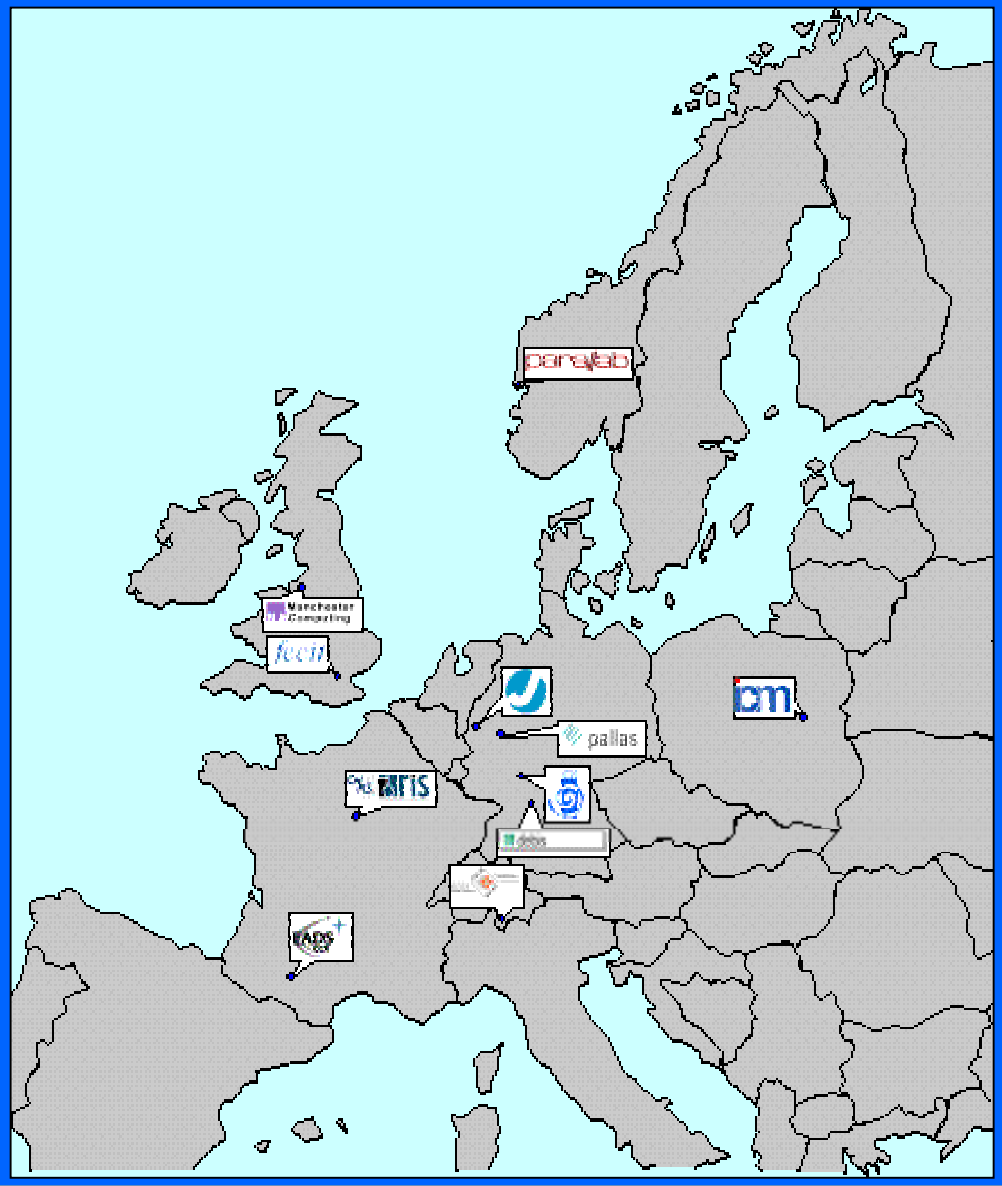
Deutscher Wetterdienst
EADS
debis Systemhaus
(Assistant Partner)

Integration

Pallas (Project Coordinator)
Fecit (Assistant Partner)

Volume: 33 person years, 2 MEuro funding
by European Commission Grant No. IST-1999-20247

EUROGRID Geography



EUROGRID Objectives

To establish a European GRID or e-Science Testbed

To operate and support the EUROGRID software infrastructure

To develop important GRID software components and to integrate them into EUROGRID

To demonstrate the effectiveness of the e-Science Testbed using distributed simulation codes

To contribute to the International GRID development

To productise the EUROGRID software components

Software Components and Simulation Codes for the e-Science Testbed

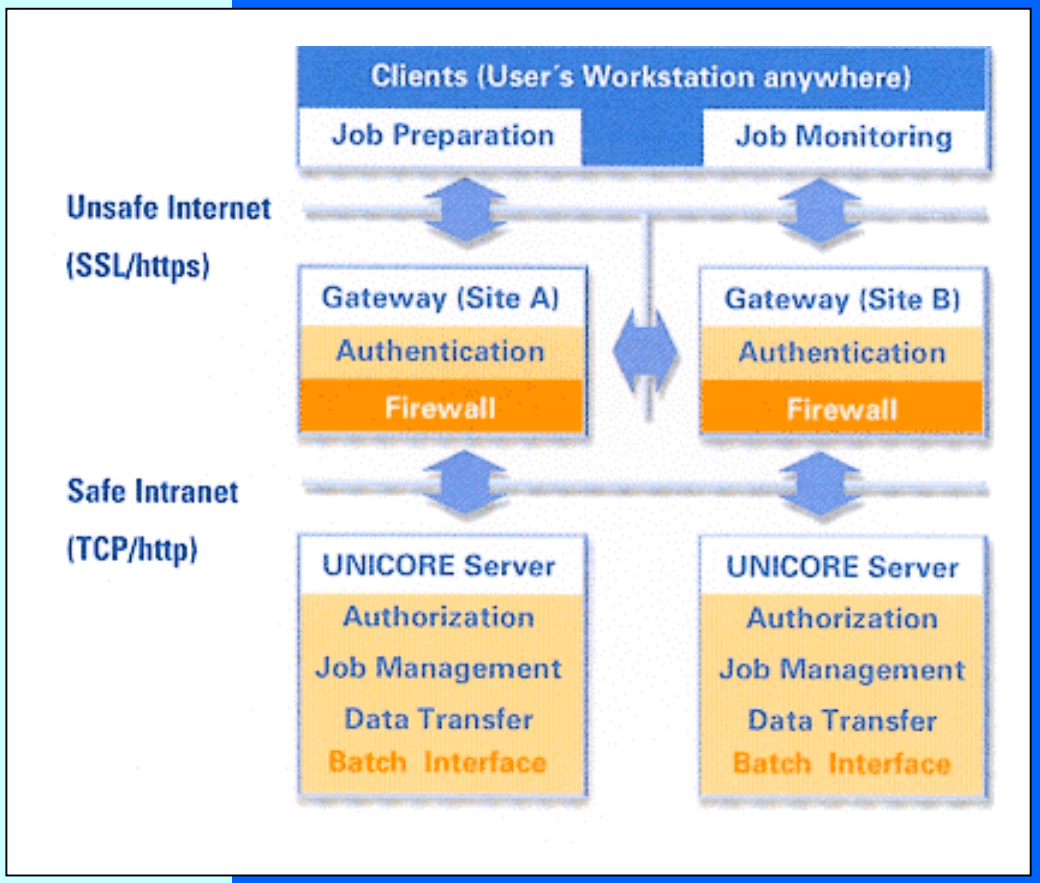
GRID Software Components

- An efficient data transfer mechanism
- **A resource broker**
- An interface for coupled applications
- An interface for "computing-on-demand"
- An interface for interactive use

Distributed Simulation Codes for e-Science Testbed

- Biomolecular simulations
- Weather prediction
- Coupled CAE simulations
- ASP-type services
- Real-time data processing
- More applications to be added by partners

EUROGRID Structure and Components



Based on UNICORE system

Develop additional GRID components

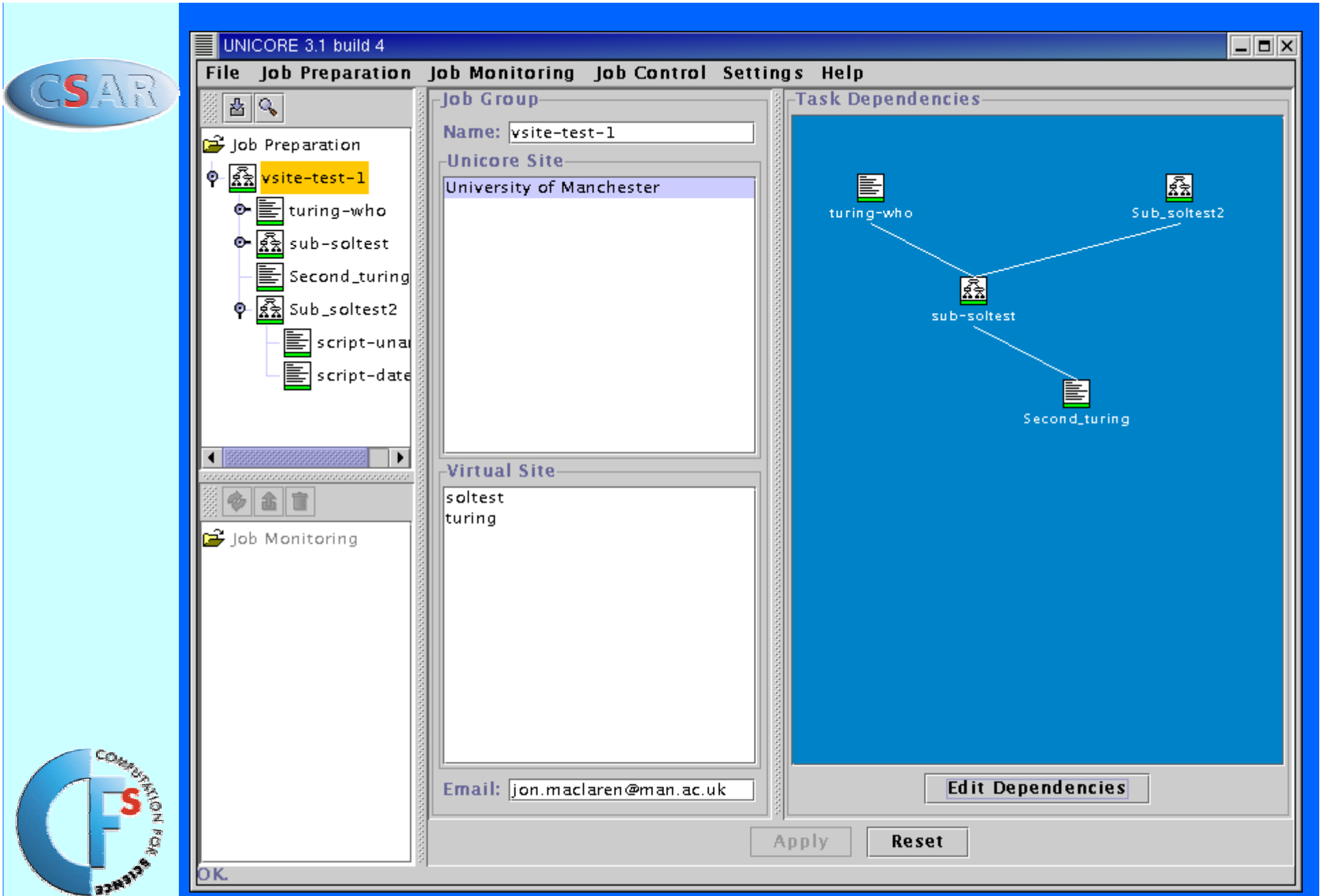
- Efficient Data Transfer
- ASP infrastructure
- **Resource Broker**
- application Coupling
- Interactive Access

Integration of new components by Pallas and Fecit

UNICORE and Globus

- Java and Perl
- Packaged Software with GUI
- Not completely open-source
- Build to handle firewalls
- Easy to customise
- No delegation
- Abstract Job Object (AJO) Framework

- ANSI C
- Toolkit approach
- Open Source
- Firewalls are a problem
- Easy to build on
- Supports Delegation
- ???



Abstract Job Object (AJO) Framework

An Abstract Job Object is an object which represents the job you want to submit. The AJO is submitted by a user to a computational resource, or Vsite.

The AJO contains all the components of the job, and a DAG representing the dependences between the components.

A component may be a script, a compilation or run of an executable, a file transfer task. *This list can be extended…*

Components can also contain locations of files to be imported and exported, at pre- and post-execution

A component may also be a SubAJO, associated with a different computational resource, or Vsite.

Seamlessness

- UNICORE knows the location of the compilers, what the optimisation flags should be, how the Batch Queue System works, and what the queues are like
- The NJS (Network Job Submission) software managing the computational resource instantiates the components of the Job for execution by a TSI (Task Submission Interface) - a lightweight Perl script
- The AJO framework provides a way of abstracting away from explicit UNIX commands
- Provides "Seamless Computing" for the users
- Must be UNIX, but this is not a problem for HPC
- Only the TSI script runs on the computational resource, so there is negligible additional load

Extending UNICORE

UNICORE can be extended via a system of plug-ins

The plug-in has a GUI front end (JPanel) which takes information from the user in a Domain-specific context

The plug-in then constructs an AJO from the user's requirements, which the user submits

Such plug-ins may only be compatible with sites with specific applications, e.g. CPMD, Gaussian

In UNICORE 3.5, application software is a resource:

- Users specify applications/versions as requirements
- Sites advertise applications as resources they can supply
- Runtime licences not yet supported – how best to do this?

Need to match resources and requirements…

Resource Brokering

- Want the user to describe the resources they require, then discover where they can run their job
- Part of the EUROGRID project is to develop a Resource Broker (Manchester and FECIT)
- The Resource Broker will match user requirements with the resources available on various sites
- Will initially work only on statically available data, i.e. will not take account of machine load, etc.
- At first, the user will be provided with a list of possible sites, from which they will manually choose one
- As the project proceeds, the brokering process will become more automatic, selecting based on user specified criteria (or even code)

What we need right now…

For the Resource Broker to become really interesting for EUROGRID/UNICORE users, several things must happen…

Firstly, the task dependence graph in the AJO framework will be made more general, and will include co-scheduling

BUT!!! The middleware must also be able to:

- Obtain estimates from resources concerning availability and cost;
- Reserve resources on behalf of the users for a certain time.

This requires:

- Batch subsystems that can estimate when a job with certain properties will run if submitted now;
- Batch subsystems that can provide advanced reservation;
- A system for advanced reservation of network bandwidth;
- A system for advanced reservation of software licences.

…and in the future

The GRID is not, and will not, be free – users must pay for the resources that they use

Users should be able to buy resources from sites they have never user previously (with suitable authentication), with tokens, e.g. money

Will strengthen supercomputer market, as they become easier to justify financially for smaller institutions

Brokers should also be able to charge for their services; they can attract business by doing loss-leading, etc.

Will lead to a **GRID-economy**

UK (and rest of Europe) ahead of US in thinking about this (CSAR already has token-based economy)

May eventually want a resource futures market…

Relevant URLs

EUROGRID Home Page:

- http://www.eurogrid.org

UNICORE Test Site:

- http://www.fz-juelich.de/unicore-test/

Access GRID at Manchester:

- http://www.man.ac.uk/mrccs/global_supercomputing/access_grid.html

Access GRID and SC2001 Sites:

- http://www-fp.mcs.anl.gov/fl/accessgrid/
- http://www-fp.mcs.anl.gov/scglobal/