GRAAP-WG

Grid Resource Allocation Agreement Protocol Working Group

GGF6

Chicago, IL

October, 2002

Agenda

1. Discuss the revised GRAAP-WG milestones.
2. Discuss web-page clarifications about the group and its purpose (2.1), including links to other GGF groups (2.2).
3. Discuss working definition of Advance Reservation.
4. Discuss revised document/additions to Advance Reservation State-of-the-Art Document (4.1). Get named contacts for each scheduler, so we can contact them directly to get this document into a useful state (4.2)...
5. Discussion of Use Cases. Construct initial requirements for the protocol, including a discussion on whether we need to do an exemplar Resource Description Language or not.
6. Anything else?

1. Revised Milestones

- **Milestones (including past):**

End of May: First draft of the charter ready **(done, charter approved)**

GGF-5: Discussion of the charter, SchedWD 12.2, SNAP; next Steps **(done)**

GGF-6: Grid RAA Protocol: Discussion of Use Cases to procure requirements

GGF-7: Grid RAA Protocol: Description of Requirements ready

GGF-8: Grid RAA Protocol: Description of Operations ready

GGF-9: Description of Leverage/Interaction with other Grid Service Standards

GGF-10: Grid RAA Protocol: First Description of Bindings ready

GGF-11: Final Grid RAA Protocol specification ready

2.1. Clarifications of Purpose and Scope

- **Charter already approved**
- **Wanted to clarify purpose of the group**
- **So we have added the following clarification to our Web Page:**

"We noticed at GGF5 that there was some confusion over the scope of the WG. It seemed that it would be helpful for us to make some clarifications to our scope. In particular, we would like to make it clear that we will NOT be working on a co-scheduler, although our protocol is intended to be used by a co-scheduler... As part of this scoping, we will also work out which groups we are going to have a relationship with, and state these explicitly. These links will be discussed on the list, and then posted here"

2.2. Other GGF Groups to Link To

- **Existing Groups:**
  - Scheduling Dictionary (SD-WG)
  - Distributed Resource Management Application API (DRMAA-WG)
  - CIM-based Grid Schema (CGS-WG): for Resource Description purposes, also Job Submission Interface Model (JSIM)
  - Open Grid Service Infrastructure (OGSI-WG)
  - New Productivity Initiative (NPI-WG)
  - Grid Protocol Architecture (GPA-WG)
  - Something in the security area? Choices are:
    - Grid Security Infrastructure (GSI-WG)
    - Grid Security Policy (GCP-WG)
    - Open Grid Services Architecture (OGSA) Security (OGSA-SEC-WG)

- **Proposed Groups:**
  - Scheduling Architecture Research Group

3. Definition of Advance Reservation

- **Current Definition stands as:**
  - **Advance Reservation** is the process of negotiating the (possibly limited or restricted) **delegation** of particular resource capabilities over a defined time interval from the **resource owner** to the requester.
- **This definition has been discussed on the list, in particular regarding the use of the word "Delegation".**
- **This has led to a clarification:**
  - The **delegation** process is rooted with the **Resource Owner** who delegates the resource capabilities to a different delegation domain, i.e. to some management system which is accessible through GRAAP (e.g. a scheduler) which has received its control by some other delegation mechanism (i.e. the administrator has installed and activated it).

3. Definition of Advance Reservation

- **The IETF RAP group’s RFC 2753 defines:**
  - **Administrative Domain:** A collection of networks under the same administrative control and grouped together for administrative purposes.

- **For the same token we could define “resource owner" as something like:**
  - A **Resource Owner** controls a collection of resources from a particular **administrative domain**.
- **Example Resource Owners:**
  - A broker that owns resources delegated from a number of different resource owners
  - A department owning a sub-set of a set of corporate resources.

- **Do we have any requirements in this area, i.e. regarding ownership or delegation?**

4.1. State-of-the-Art: Current state

- **Document now in HTML format at:**

  http://people.man.ac.uk/~zzcgujm/GGF/sched-graap-2.0.html

- **Document to be rapidly updated, periodically "baselined" as a fixed version, and kept as both HTML and Acrobat PDF, e.g.**

  http://people.man.ac.uk/~zzcgujm/GGF/sota-04.10.02/sched-graap-2.0.html

  http://people.man.ac.uk/~zzcgujm/GGF/sota-04.10.02/sched-graap-2.0.pdf

- **Other key changes this time:**
  - Basic clarifications, incl. replacing "Advanced Reservation" with "Advance Reservation", consistent with Scheduling Dictionary.
  - Added contributions on Maui and Catalina.
  - Added list of contributors/editors.
  - Clarified what negotiation means in the table.
  - Added document change control section.
  - Updated document into correct GGF format.

4.2. State-of-the-Art: Contacts and Future

- **Contacts we have:**
  - LSF – Bingfeng Lu (Platform)
  - Maui – Dave Jackson (Supercluster)
  - EASY/COSY – Wolfgang Ziegler (SCAI)
  - Catalina – Kenneth Yoshimoto (SDSC)

- **Contacts we need:**
  - OpenPBS/PBS –
  - Paderborn CCS –
  - LoadLeveler – Gareth Bestor (IBM)?
  - Sun GridEngine –

- **More schedulers?**
  - Libra (budget-based Economic Scheduler which plugs into PBS) – Rajkumar Buyya (University of Melbourne)?

5. Use Cases and Requirements

- **Thanks to all who sent in Use Cases for GRAAP:**
  - Jim Pruyne (HP Labs)
  - Volker Sander (FZ-Jülich) and Keith Jackson (Lawrence Berkeley National Laboratory)
  - Stephen Pickles (University of Manchester)
  - J P Giddy (Welsh e-Science Centre - WeSC) – no co-scheduling required.

Classes of Grid Applications

- **Distributed Supercomputing**

  Coupling of multiple potentially distributed supercomputers to solve a single, tightly coupled problem

- **High-Throughput Computing**

  Massive acquisition of available and accessible computing resource to solve an autonomous fraction of a large scale problem problem

- **On-Demand Computing**

  Ad-Hoc access to advanced capabilities to fulfill a short term demand

- **Data-Intensive Computing**

  Synthesis of new information from geographically-distributed data repositories, digital libraries, and databases

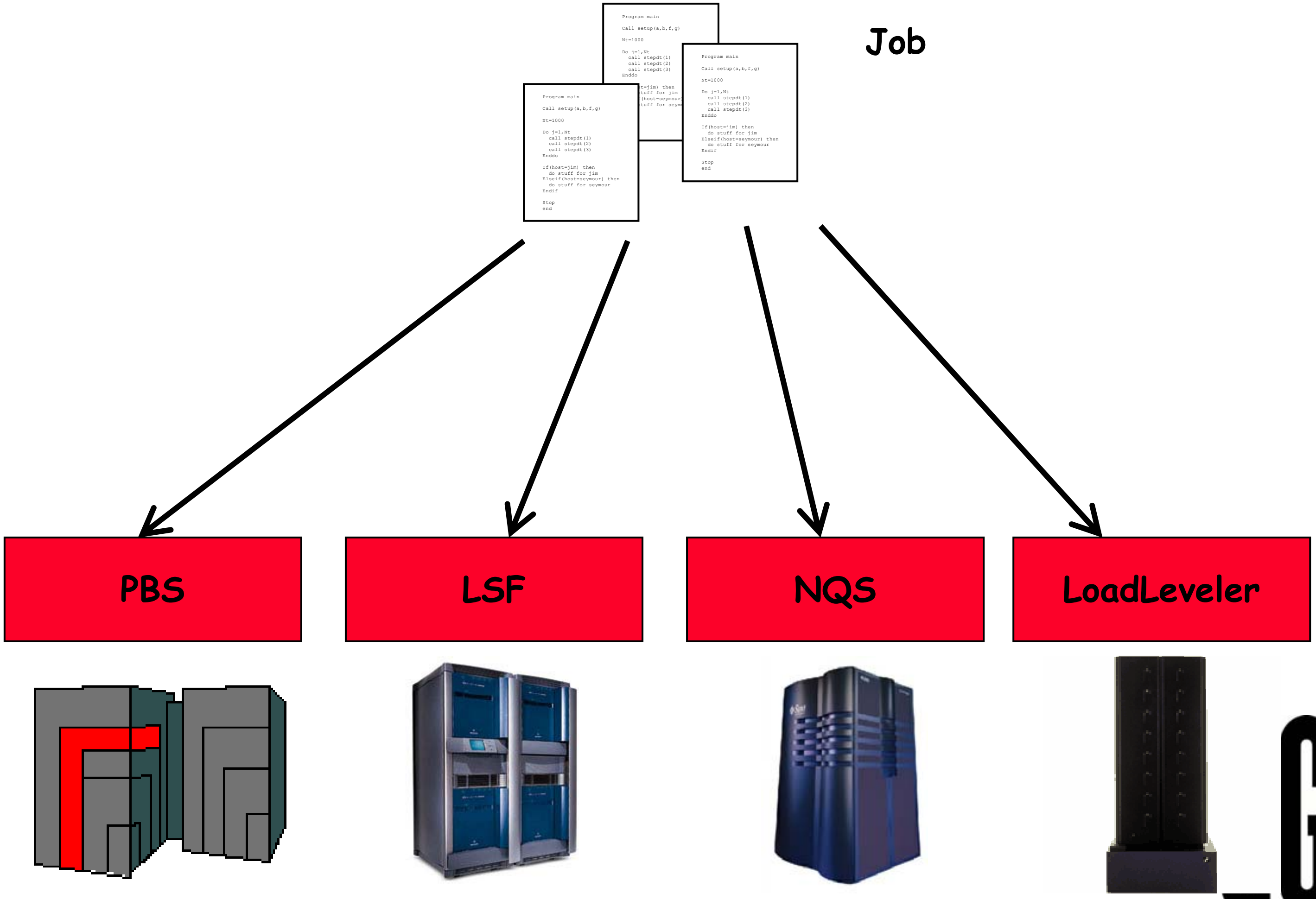
- **Collaborative Computing**

  Enabling and enhancing human interactions

The Advance Reservation Case

- **Distributed Supercomputing**
  - "Co-scheduling" and co-allocation of scarce, expensive resources
  - Some local schedulers have to exclusively assign subtasks to nodes
  - Advance Reservation can improve usage economy
- **High-Throughput Computing**
  - Trade-off between allocation overhead and benefit
  - Remote checkpointing
  - Win might be improved by relying on particular network capabilities
- **On-Demand Computing**
  - Increased Trade-off demand
  - Co-allocation and network capabilities often required
- **Data-Intensive Computing**
  - Workflow dependencies
  - Deadline file staging
- **Collaborative Computing**
  - Co-allocation and network capabilities often required

The Root: Job-Scheduling on the Grid



```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

Job

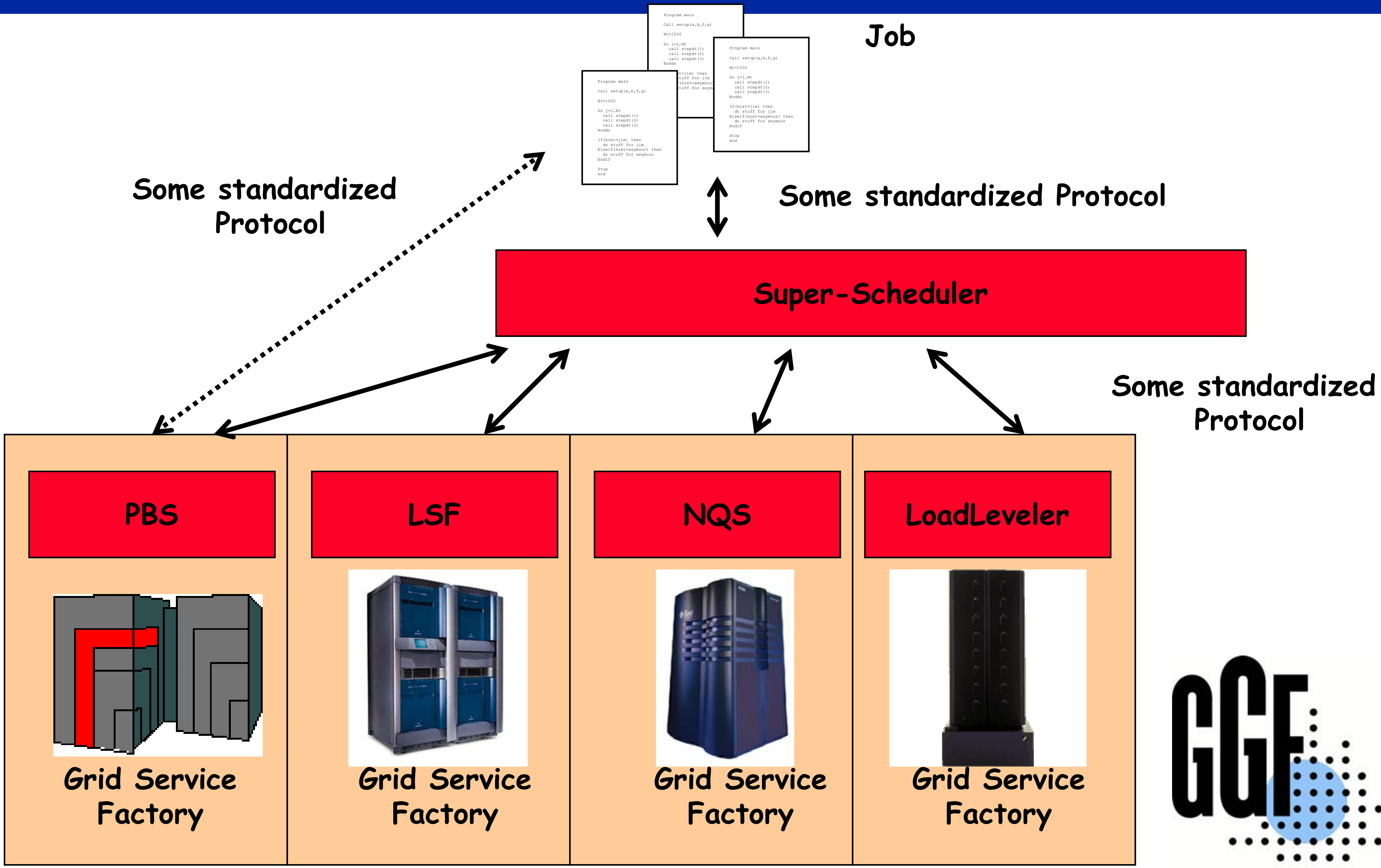
PBS

LSF

NQS

LoadLeveler

Distributed Supercomputer Application



Job

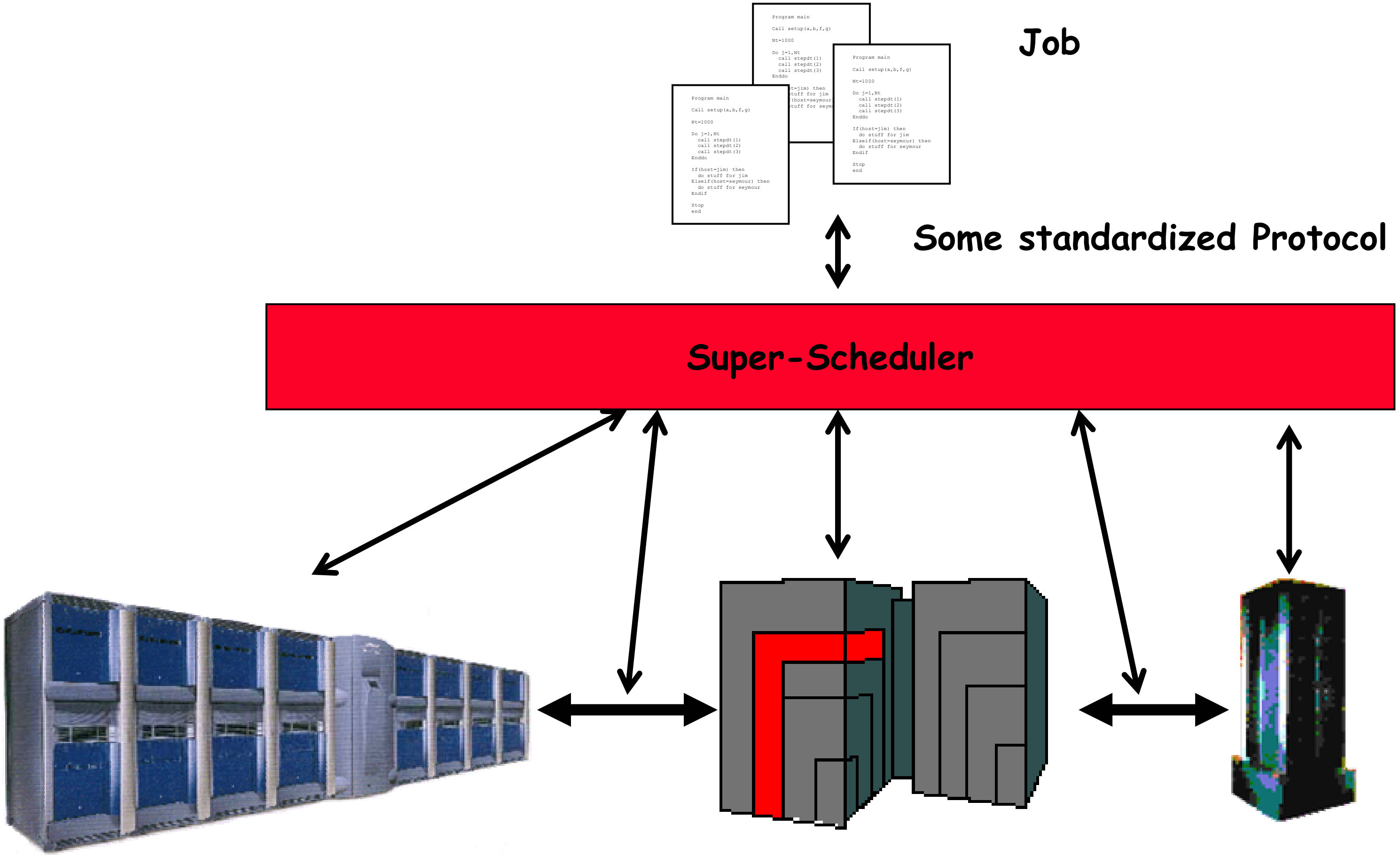
```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

Some standardized Protocol

Some standardized Protocol

Some standardized Protocol

Application Example: Distributed Supercomputing

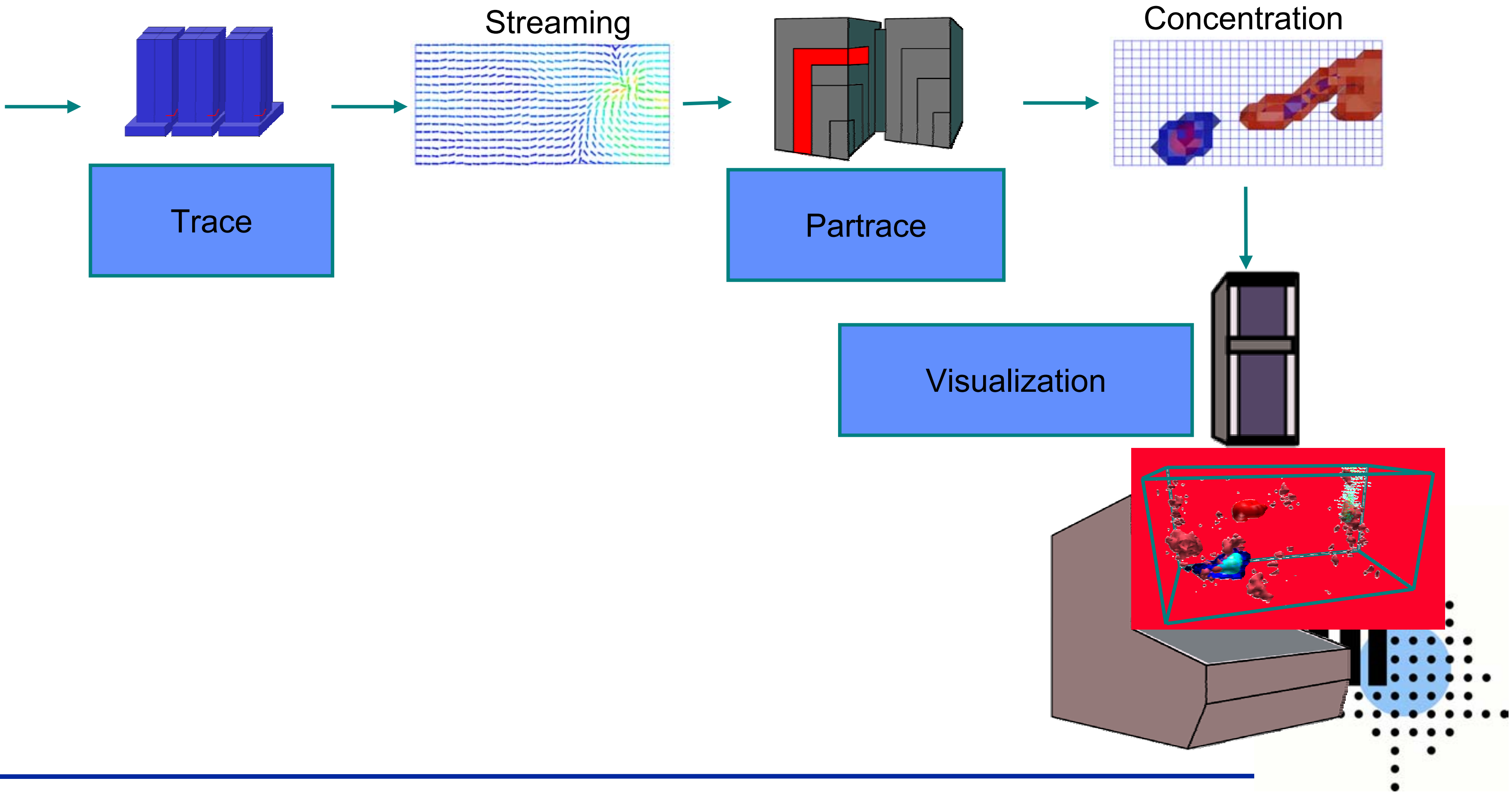


```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

Example

Pollution Penetration in the Ground



Streaming

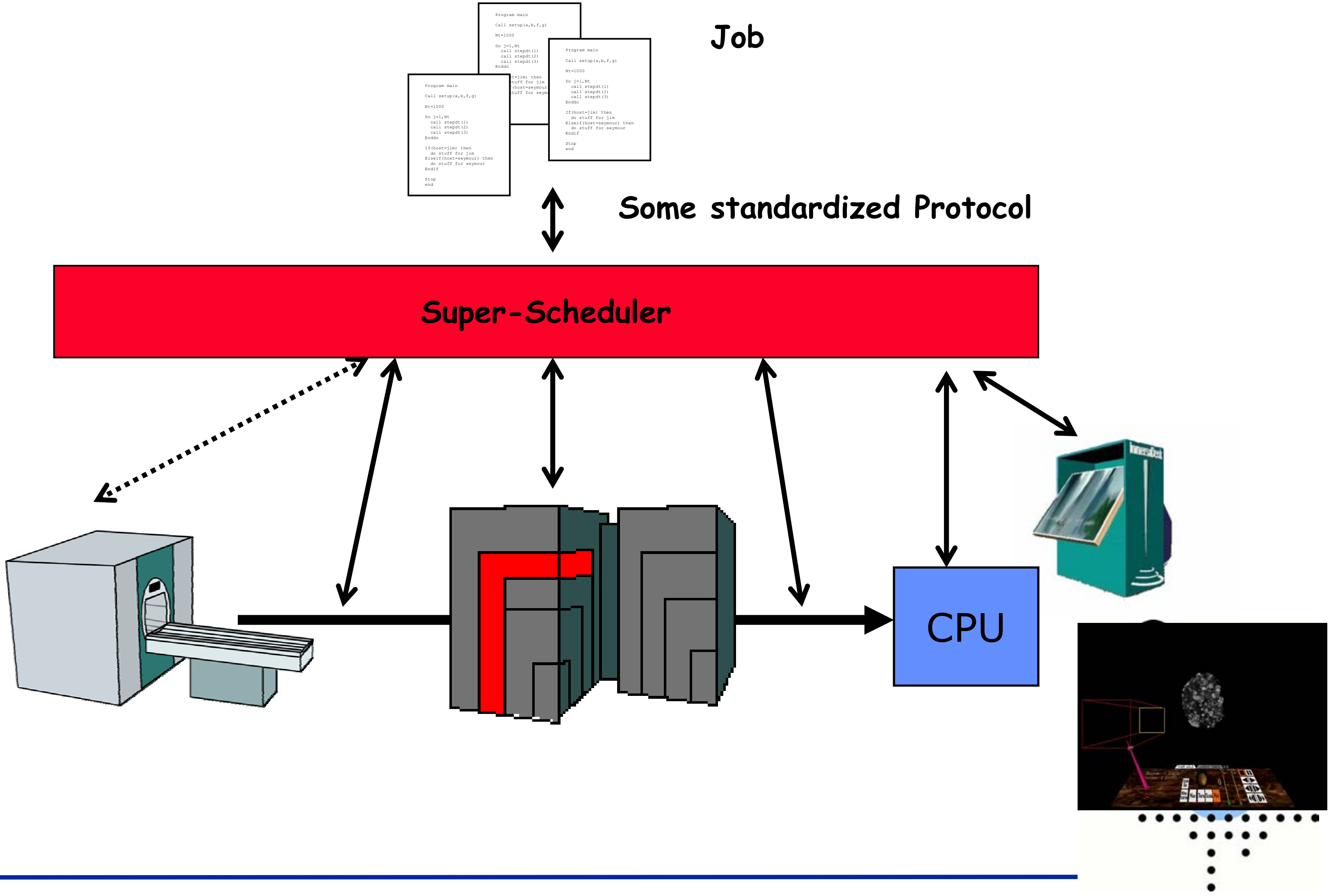
Concentration

Trace

Partrace

Visualization

Application Example:
Smart Instruments

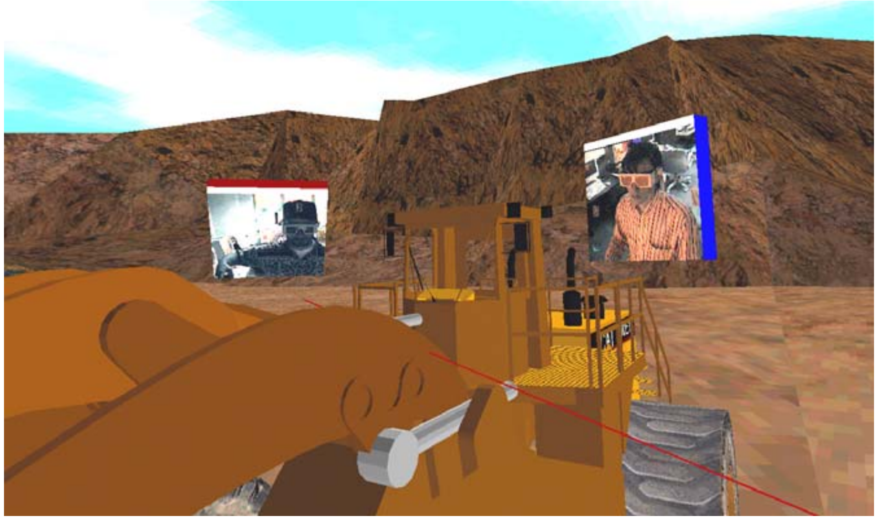


```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

st=jim) then
stuff for jim
(host=seymour)
stuff for seym
```

```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

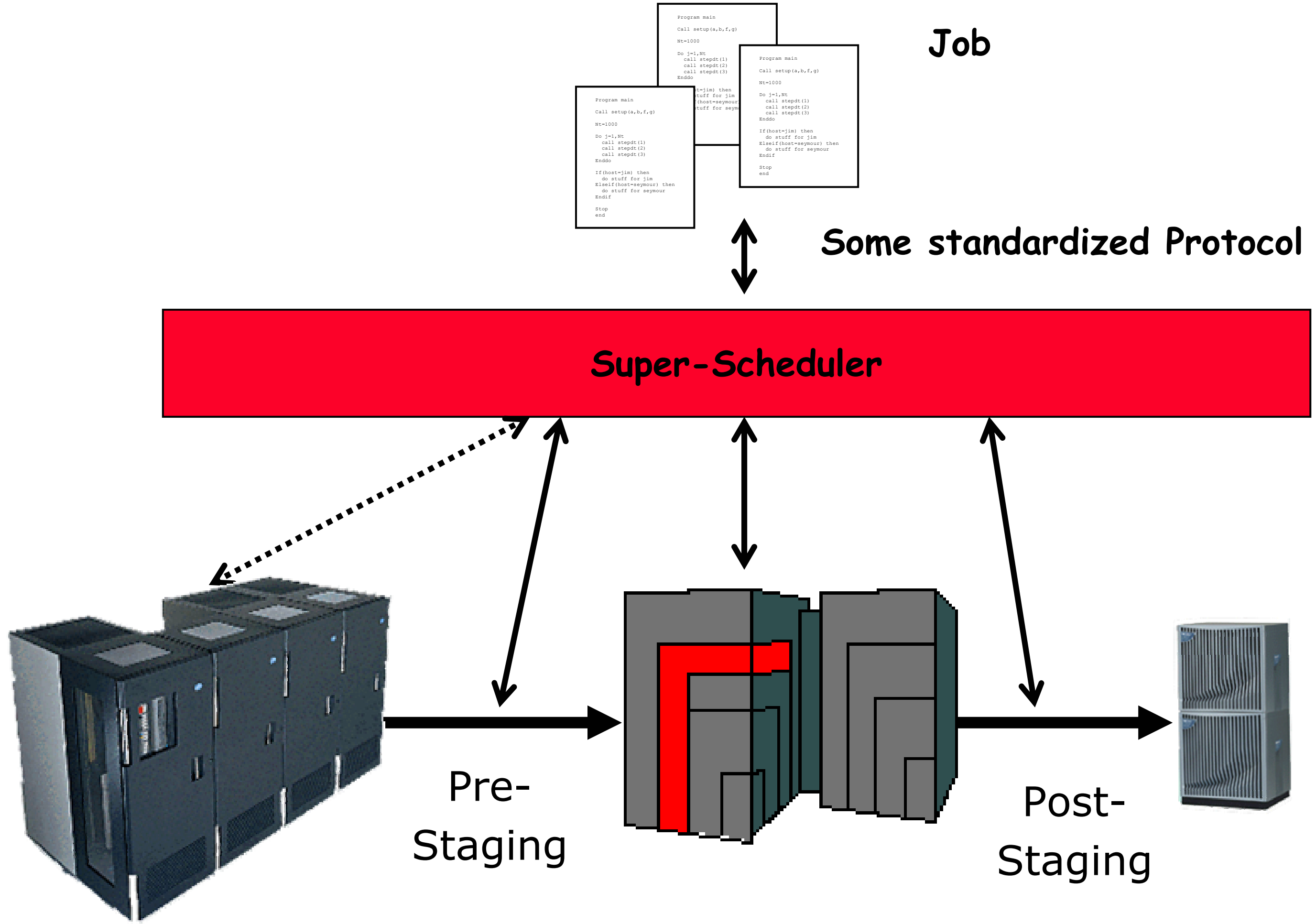
```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

Collaborative Development Environments



Graphics: Valerie Lehner, NCSA, Tom DeFanti, UIC/EVL

Application Example:
Simulation using Data from some Archive



```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

```
Program main

Call setup(a,b,f,g)

Nt=1000

Do j=1,Nt
  call stepdt(1)
  call stepdt(2)
  call stepdt(3)
Enddo

If(host=jim) then
  do stuff for jim
Elseif(host=seymour) then
  do stuff for seymour
Endif

Stop
end
```

Job

Some standardized Protocol

Super-Scheduler

Pre-Staging

Post-Staging

archival storage

A use case from the RealityGrid project (1/9)

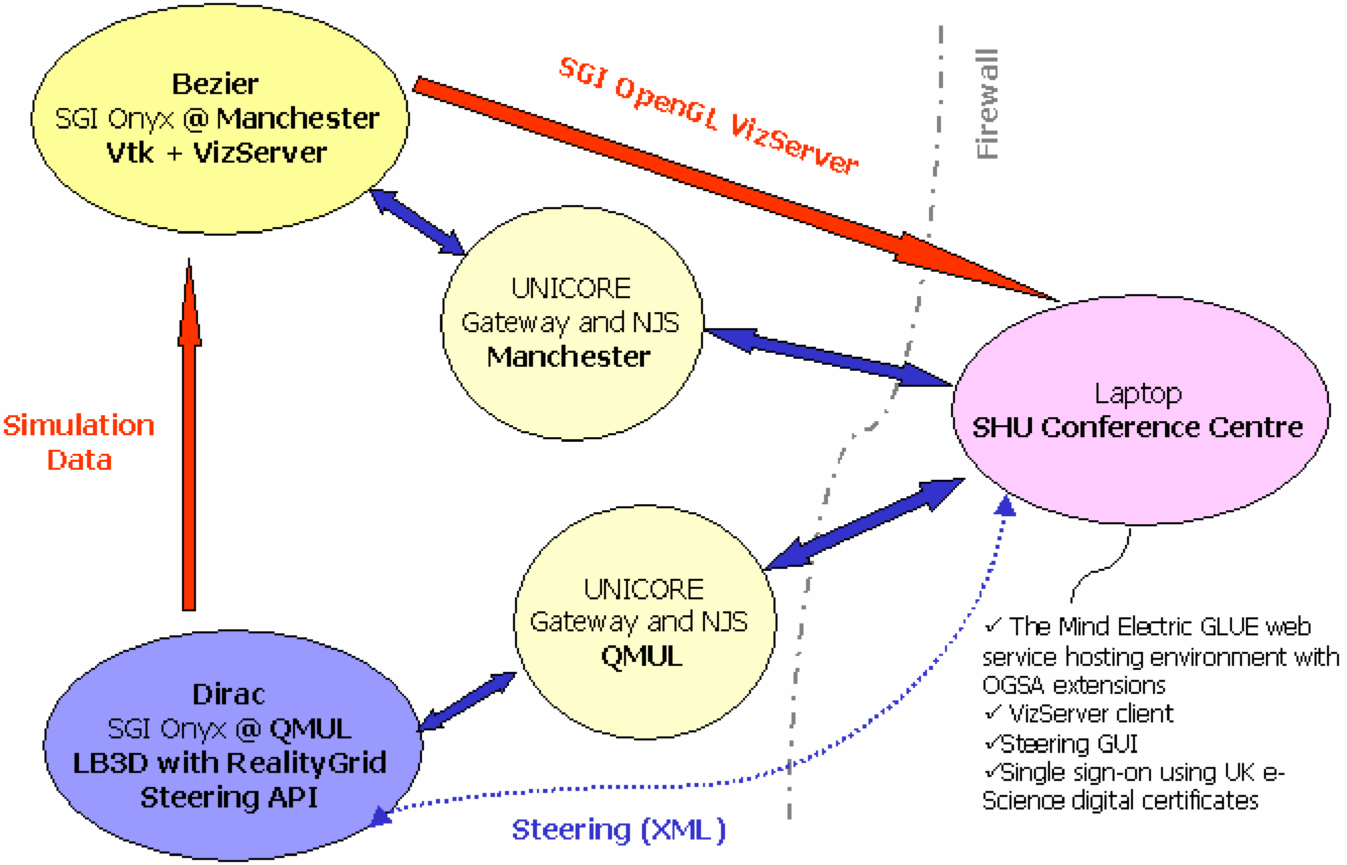
Sent in by Stephen Pickles, Software Infrastructure Manager, RealityGrid, 13 October 2002

Introduction (1/3)

The RealityGrid project (http://www.realitygrid.org) aims to predict the realistic behaviour of matter based on the properties of the microscopic components using diverse simulation methods (Lattice Boltzmann, Molecular Dynamics and Monte Carlo) spanning many time and length scales and the discovery of new materials through integrated experiments. A central theme of RealityGrid is the facilitation of distributed and collaborative exploration of parameter space through computational steering and on-line, high-end visualization.

A use case from the RealityGrid project (2/9)

Introduction (2/3)



Bezier
SGI Onyx @ Manchester
Vtk + VizServer

SGI OpenGL VizServer

Firewall

UNICORE
Gateway and NJS
Manchester

Laptop
SHU Conference Centre

Simulation
Data

UNICORE
Gateway and NJS
QMUL

Dirac
SGI Onyx @ QMUL
LB3D with RealityGrid
Steering API

Steering (XML)

✓ The Mind Electric GLUE web service hosting environment with OGSA extensions
✓ VizServer client
✓Steering GUI
✓Single sign-on using UK e-Science digital certificates

Figure 1. One experimental configuration employed in the RealityGrid "Fast Track" Computational steering demonstration at the UK e-Science All Hands Meeting, Sheffield, September 2002

A use case from the RealityGrid project (3/9)

Introduction (3/3)

- A typical RealityGrid scenario involves a large-scale simulation running on a massively parallel system at on site coupled to a high-end visualization system at another site with the steering and display interfaces running at one or more remote sites.
- The simulation component periodically (or as demanded by the steerer component) emits “samples” for consumption by the visualization component, while grid middleware is responsible for the transfer of data between components.
- The RealityGrid “fast track” prototype (see figure 1) relies heavily on UNICORE and web services with OGSA extensions, but RealityGrid is not locked in to any particular middleware; indeed the RealityGrid “deep track” is able to demonstrate similar functionality using ICENI and Globus.

A use case from the RealityGrid project (4/9)

Immediate Requirements for Advance Reservation (1/3)

- The most pressing requirement for advance reservation in RealityGrid arises out of the need to co-allocate (or co-schedule)
  - (a) multiple processors to run a parallel simulation code and
  - (b) multiple graphics pipes and processors on the visualization system
- Co-allocation may be required now (either by a RealityGrid developer or by a scientist engaged in routine investigations) or at some more distant time in the future (for a scheduled collaborative session). We expect advance reservation to subsume both co-allocation scenarios.
- The visualization resources (b) will usually be located on a different system to the computational resources (a).
- The two sets of resources ((a) and (b)) will often be located on systems owned and administered by different organisations, and the administration teams within the two organisations, if aware of each other's existence at all, are unlikely to have established comprehensive Service Level Agreements.

A use case from the RealityGrid project (5/9)

Immediate Requirements for Advance Reservation (2/3)

- It is assumed that the end-user(s) will be able to access resources on both systems by presenting a single credential, through a single sign-on mechanism based on digital certificates such as GSI or the UNICORE security model.
- It is anticipated that the system (a) running the simulation will typically be a massively parallel system with a workload characterised by sustained heavy demand and therefore the allocation of resources is likely to be entrusted to a batch scheduling system. The characteristics of the visualization system on the other hand are likely to vary, with demand for graphics and CPU ranging independently from low to high, and there may or may not be a batch scheduling system in place.
- The resources (b) required on the visualization system include both graphics pipes and processor (CPU+memory) resources. In general, whatever system may exist for booking the graphics pipes is unlikely to be integrated with whatever system may exist for booking the processors.

A use case from the RealityGrid project (6/9)

•Immediate Requirements for Advance Reservation (3/3)

- We may therefore distinguish two cases:
  1. co-allocation of processors on the simulation system, graphics pipes on the visualization system, and processors on the visualisation system;
  2. where the visualization system does not run a batch scheduling system, co-allocation of processors on the simulation system and graphics pipes on the visualization system, relying on chance (or external arrangement) to acquire a sufficient share of CPU resource for visualization purposes.
- The ability for a RealityGrid user to reserve processors and graphics pipes manually *without involving system administrators* would be useful now, and would remove a significant barrier to the routine use of computational steering. The ability for an agent to do the same will be important for the resource broker that will be developed in the later stages of the RealityGrid project.

A use case from the RealityGrid project (7/9)

- **Future Requirements for Advance Reservation (1/3)**
  - Based on current projections, the largest computationally-steered simulations that RealityGrid is likely to undertake will require bandwidth between the simulation and visualization systems of order 5 Gbps in order to achieve satisfactory interactivity. The bandwidth requirements between visualization systems are less demanding – 100 Mbps will be adequate for most purposes – but reasonably good latency and jitter characteristics are desirable. Thus the ability to make advance reservations of network bandwidth with certain quality of service characteristics and using the same protocols as for the reservation of processors are seen as desirable by RealityGrid.

A use case from the RealityGrid project (8/9)

•Future Requirements for Advance Reservation (2/3)

- RealityGrid's design philosophy is component based. Although the componentisation in the "fast track" example of figure (1) is coarse-grained (the simulation and visualization components are deployed on different systems), the RealityGrid "deep track" is investigating finer-grained componentisation in which the simulation is composed out of a number of smaller communicating components, each of which must be deployed onto (possibly remote) computational resources at run-time. Thus RealityGrid will need robust mechanisms for co-allocating much more complex sets of resources than indicated in figure (1).

A use case from the RealityGrid project (9/9)

- **Future Requirements for Advance Reservation (3/3)**
  - RealityGrid has a significant “deep track” work package devoted to performance control. The goal of this work package is to optimise the collective performance of the components comprising the RealityGrid application based on performance information collected at run time. Initially, the set of resources will be assumed to be fixed during execution, and it is by redistributing components across this set of resources that the performance control system hopes to achieve performance improvement. Ultimately, however, the ambition is to adapt the application to utilize new resources that become available during execution; this is likely to require rather specialist functionality of the advance reservation system such as the ability to renegotiate an existing reservation.

5. Requirements: Do we write an RDL?

- The GRAAP protocol will carry a description of the resources that are being negotiated, specified in some Resource Description Language or RDL.
- We want GRAAP to be RDL-neutral, i.e. to be able to handle any well-defined where the resource description can be encoded as some sort of byte-stream, e.g. string.
- We also want to be able to use and test GRAAP.
- So do we need to write a simple exemplar RDL, e.g. for blocks of supercompter time?
- Or can we wait for a while to see if the Grid community will do this for us?
- Must stay within the scope of our group…