GRID COMPUTING – AN INTRODUCTION

Dr G SUDHA SADSAIVAM

Outline

- Introduction to Grid Computing
- Methods of Grid computing
- Grid Middleware
- Grid Architecture

Grid Computing

- **Grid computing** is a form of distributed computing whereby a "super and virtual computer" is composed of a cluster of networked, loosely coupled computers, acting in concert to perform very large tasks.

- Grid computing (Foster and Kesselman, 1999) is a growing technology that facilitates the executions of large-scale resource intensive applications on geographically distributed computing resources.

- Facilitates flexible, secure, coordinated large scale resource sharing among dynamic collections of individuals, institutions, and resource

- Enable communities ("virtual organizations") to share geographically distributed resources as they pursue common goals
- Ian Foster and Carl Kesselman

Criteria for a Grid:

- Coordinates resources that are not subject to centralized control.
- Uses standard, open, general-purpose protocols and interfaces.
- Delivers nontrivial qualities of service.

Benefits

- Exploit Underutilized resources
- Resource load Balancing
- Virtualize resources across an enterprise
  - Data Grids, Compute Grids
- Enable collaboration for virtual organizations

Grid Applications

Data and computationally intensive applications:

This technology has been applied to computationally-intensive scientific, mathematical, and academic problems like drug discovery, economic forecasting, seismic analysis back office data processing in support of e-commerce

- A chemist may utilize hundreds of processors to screen thousands of compounds per hour.
- Teams of engineers worldwide pool resources to analyze terabytes of structural data.
- Meteorologists seek to visualize and analyze petabytes of climate data with enormous computational demands.

Resource sharing

- Computers, storage, sensors, networks, …
- Sharing always conditional: issues of trust, policy, negotiation, payment, …

Coordinated problem solving

- distributed data analysis, computation, collaboration, …

Grid Topologies

- Intragrid
  - Local grid within an organisation
  - Trust based on personal contracts
- Extragrid
  - Resources of a consortium of organisations connected through a (Virtual) Private Network
  - Trust based on Business to Business contracts
- Intergrid
  - Global sharing of resources through the internet
  - Trust based on certification

Computational Grid

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

"The Grid: Blueprint for a New Computing Infrastructure", Kesselman & Foster

Example : Science Grid (US Department of Energy)

Data Grid

- A **data grid** is a grid computing system that deals with data — the **controlled sharing and management of large amounts of distributed data**.

- Data Grid is the storage component of a grid environment. Scientific and engineering applications require access to large amounts of data, and often this data is widely distributed. A data grid provides seamless access to the local or remote data required to complete compute intensive calculations.

Example :

Biomedical informatics Research Network (BIRN),

the Southern California earthquake Center (SCEC).

Methods of Grid Computing

- Distributed Supercomputing
- High-Throughput Computing
- On-Demand Computing
- Data-Intensive Computing
- Collaborative Computing
- Logistical Networking

Distributed Supercomputing

- Combining multiple high-capacity resources on a computational grid into a single, virtual distributed supercomputer.
- Tackle problems that cannot be solved on a single system.

High-Throughput Computing

- Uses the grid to schedule large numbers of loosely coupled or independent tasks, with the goal of putting unused processor cycles to work.

On-Demand Computing

- Uses grid capabilities to meet short-term requirements for resources that are not locally accessible.
- Models real-time computing demands.

Collaborative Computing

- Concerned primarily with enabling and enhancing human-to-human interactions.
- Applications are often structured in terms of a virtual shared space.

Data-Intensive Computing

- The focus is on synthesizing new information from data that is maintained in geographically distributed repositories, digital libraries, and databases.
- Particularly useful for distributed data mining.

Logistical Networking

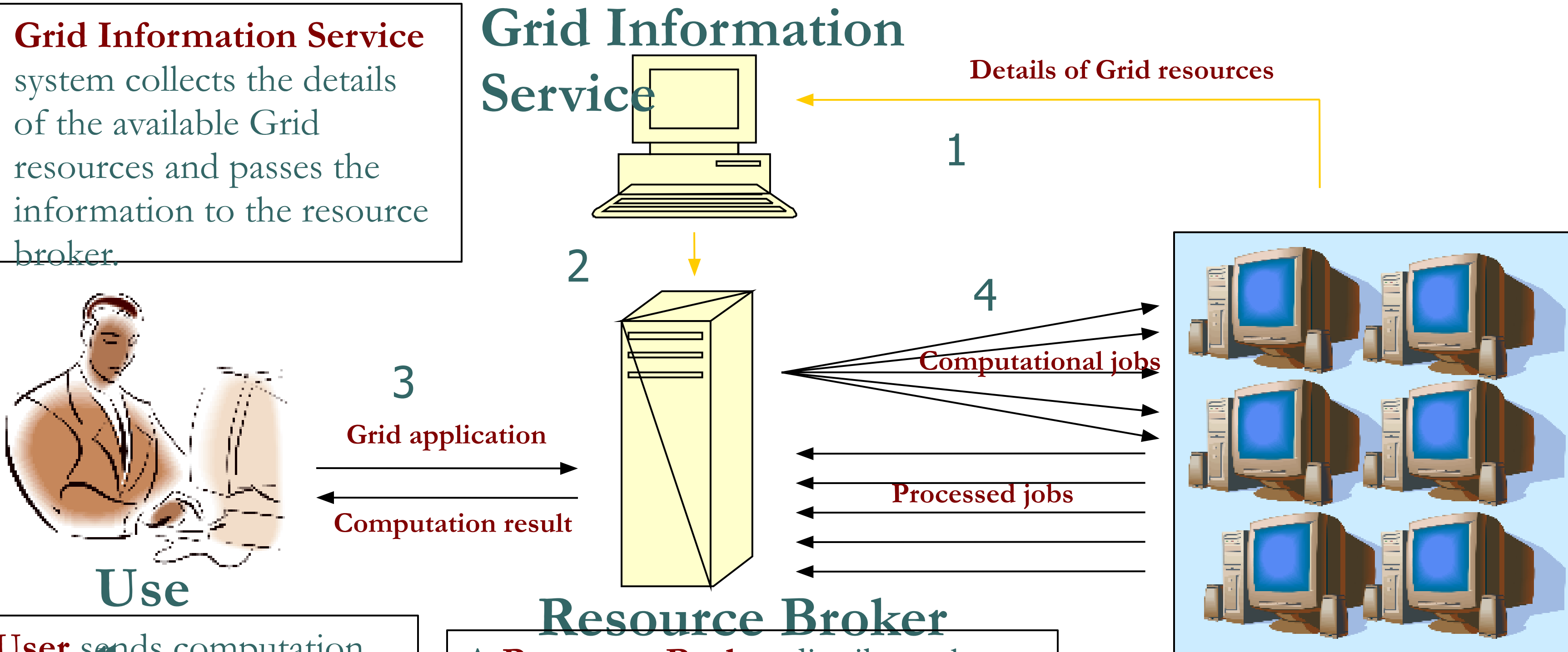
- Logistical networks focus on exposing storage resources inside networks by optimizing the global scheduling of data transport, and data storage.
- Contrasts with traditional networking, which does not explicitly model storage resources in the network.
- high-level services for Grid applications
- Called "logistical" because of the analogy it bears with the systems of warehouses, depots, and distribution channels.

P2P Computing vs Grid Computing

- Differ in Target Communities
- Grid system deals with more complex, more powerful, more diverse and highly interconnected set of resources than P2P.
- VO

A typical view of Grid environment

Grid Information Service system collects the details of the available Grid resources and passes the information to the resource broker.



A **User** sends computation or data intensive application to Global Grids in order to speed up the execution of the application.

A **Resource Broker** distribute the jobs in an application to the Grid resources based on user's QoS requirements and details of available Grid resources for further executions.

Grid Resources (Cluster, PC, Supercomputer, database, instruments, etc.) in the Global Grid execute the user jobs.

Grid Middleware

- Grids are typically managed by grid ware -

a special type of middleware that enable sharing and manage grid components based on user requirements and resource attributes (e.g., capacity, performance)

- Software that connects other software components or applications to provide the following functions:
  - Run applications on suitable available resources
    - Brokering, Scheduling
  - Provide uniform, high-level access to resources
    - Semantic interfaces
    - Web Services, Service Oriented Architectures
  - Address inter-domain issues of security, policy, etc.
    - Federated Identities
  - Provide application-level status
  - monitoring and control

Middlewares

- Globus –chicago Univ
- Condor – Wisconsin Univ – High throughput computing
- Legion – Virginia Univ – virtual workspaces- collaborative computing
- IBP – Internet back pane – Tennesse Univ – logistical networking
- NetSolve – solving scientific problems in heterogeneous env – high throughput & data intensive

Two Key Grid Computing Groups

The Globus Alliance (www.globus.org)

- Composed of people from:

  Argonne National Labs, University of Chicago, University of Southern California Information Sciences Institute, University of Edinburgh and others.
- OGSA/I standards initially proposed by the Globus Group

The Global Grid Forum (www.ggf.org)

- Heavy involvement of Academic Groups and Industry
  - (e.g. IBM Grid Computing, HP, United Devices, Oracle, UK e-Science Programme, US DOE, US NSF, Indiana University, and many others)
- Process
  - Meets three times annually
  - Solicits involvement from industry, research groups, and academics

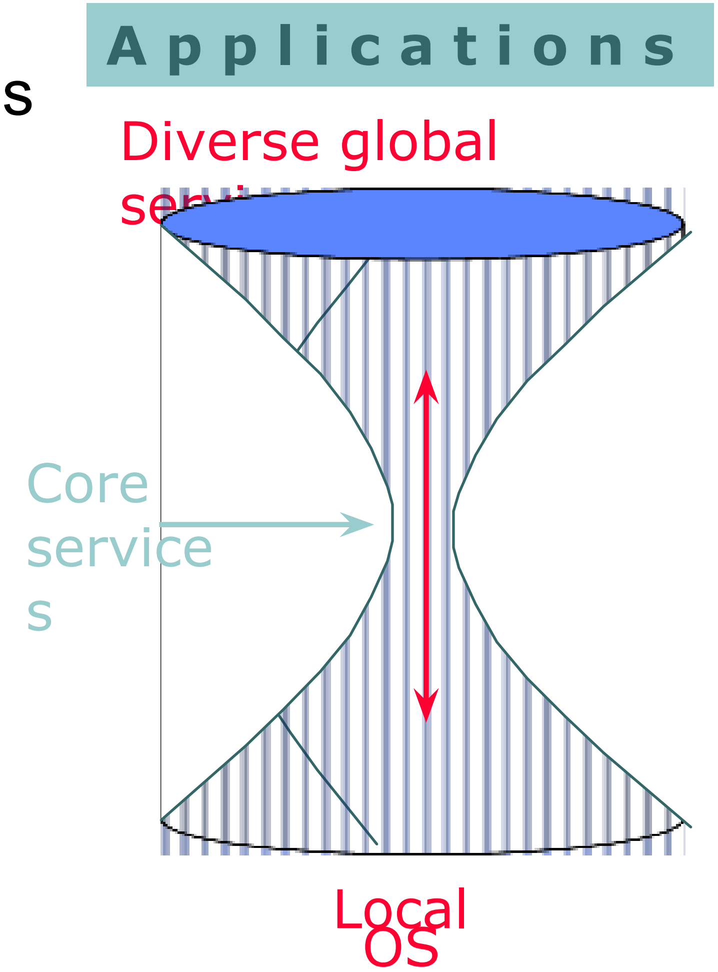
Some of the Major Grid Projects

Name	URL/Sponsor	Focus
EuroGrid, Grid Interoperability (GRIP)	eurogrid.org European Union	Create tech for remote access to super comp resources & simulation codes; in GRIP, integrate with Globus Toolkit™
Fusion Collaboratory	fusiongrid.org DOE Off. Science	Create a national computational collaboratory for fusion research
Globus Project™	globus.org DARPA, DOE, NSF, NASA, Msoft	Research on Grid technologies; development and support of Globus Toolkit™; application and deployment
GridLab	gridlab.org European Union	Grid technologies and applications
GridPP	gridpp.ac.uk U.K. eScience	Create & apply an operational grid within the U.K. for particle physics research
Grid Research Integration Dev. & Support Center	grids-center.org NSF	Integration, deployment, support of the NSF Middleware Infrastructure for research & education

Grid Architecture

The Hourglass Model

- Focus on architecture issues
  - Propose set of core services as basic infrastructure
  - Used to construct high-level, domain-specific solutions (diverse)
- Design principles
  - Keep participation cost low
  - Enable local control
  - Support for adaptation
  - “IP hourglass” model



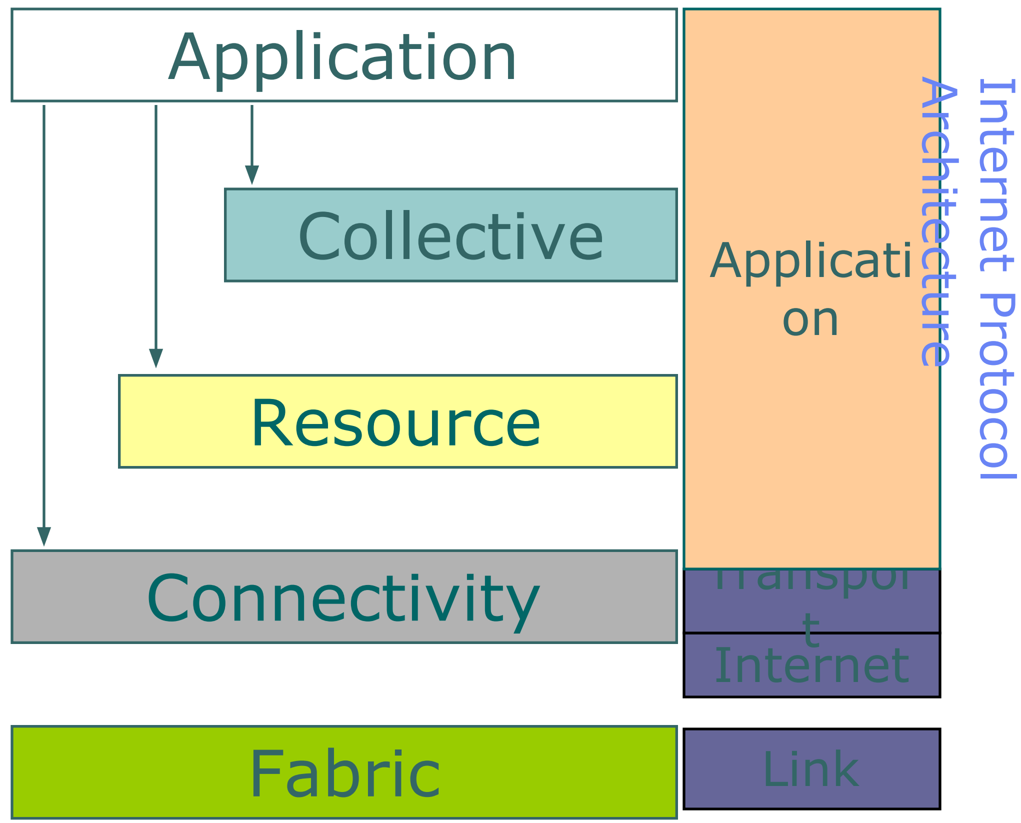
Layered Grid Architecture
(By Analogy to Internet Architecture)

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

"Sharing single resources": negotiating access, controlling use

"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources



Example:
Data Grid Architecture

App	Discipline-Specific Data Grid Application
Collective (App)	Coherency control, replica selection, task management, virtual data catalog, virtual data code catalog, …
Collective (Generic)	Replica catalog, replica management, co-allocation, certificate authorities, metadata catalogs,
Resource	Access to data, access to computers, access to network performance data, …
Connect	Communication, service discovery (DNS), authentication, authorization, delegation
Fabric	Storage systems, clusters, networks, network caches, …

Simulation tools

- GridSim – job scheduling
- SimGrid – single client multiserver scheduling
- Bricks – scheduling
- GangSim- Ganglia VO
- OptoSim – Data Grid Simulations
- G3S – Grid Security services Simulator – security services

Simulation tool

- GridSim is a Java-based toolkit for modeling, and simulation of distributed resource management and scheduling for conventional Grid environment.
- GridSim is based on SimJava, a general purpose discrete-event simulation package implemented in Java.
- All components in GridSim communicate with each other through message passing operations defined by SimJava.

Salient features of the GridSim

- It allows modeling of **heterogeneous** types of resources.
- Resources can be modeled operating under **space- or time-shared mode.**
- Resource capability can be defined (in the form of **MIPS** (Million Instructions Per Second) benchmark.
- Resources can be located in **any time zone**.
- **Weekends and holidays** can be mapped depending on resource's local time to model non-Grid (local) workload.
- Resources can be **booked** for advance reservation.
- Applications with different **parallel application** models can be simulated.

Salient features of the GridSim

- Application tasks can be heterogeneous and they can be CPU or I/O intensive.
- There is no limit on the number of application jobs that can be submitted to a resource.
- Multiple user entities can submit tasks for execution simultaneously in the same resource, which may be time-shared or space-shared. This feature helps in building schedulers that can use different market-driven economic models for selecting services competitively.
- Network speed between resources can be specified.
- It supports simulation of both static and dynamic schedulers.
- Statistics of all or selected operations can be recorded and they can be analyzed using GridSim statistics analysis methods.

A Modular Architecture for GridSim Platform and Components.

Application, User, Grid Scenario's input and Results

Appn Conf | Res Conf | User Req | Grid Sc | ... | Output

Grid Resource Brokers or Schedulers

GridSim Toolkit

Appn modelin | Res entity | Info serv | Job mgmt | Res alloc | Statis

Resource Modeling and Simulation

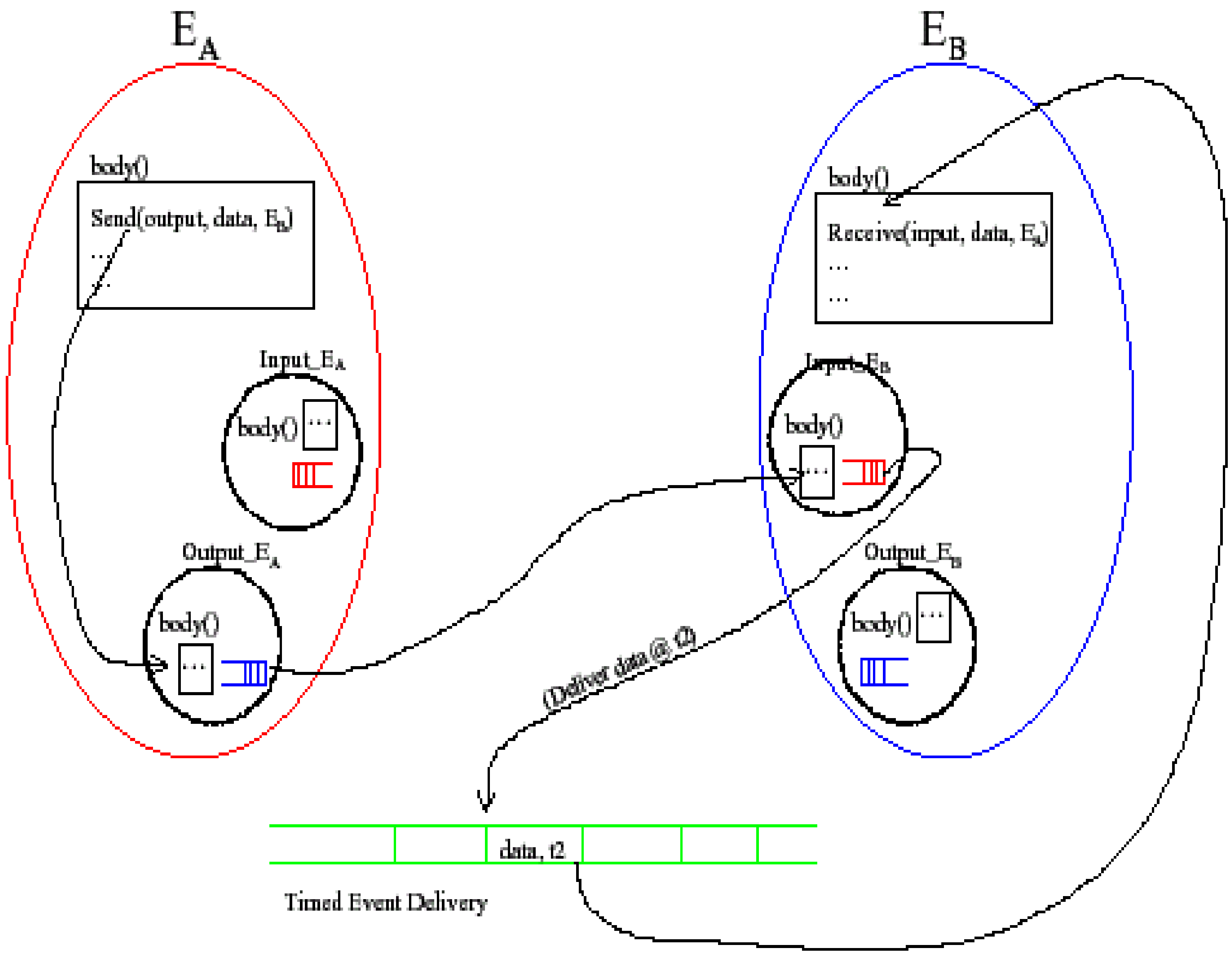
Single CPU | SMPs | Clusters | Load | Netw | Reservati on

Basic Discrete Event Simulation Infrastructure

SimJava | Distributed SimJava

Virtual Machine

PCs | Workstati on | SMPs | Clusters | Distributed Resources



E_A
body()
Send(output, data, E_B)
…
…
Input_E_A
body()
…
Output_E_A
body()
…
E_B
body()
Receive(input, data, E_A)
…
…
Input_E_B
body()
…
Output_E_B
body()
…
(Deliver data @ t2)
data, t2
Timed Event Delivery

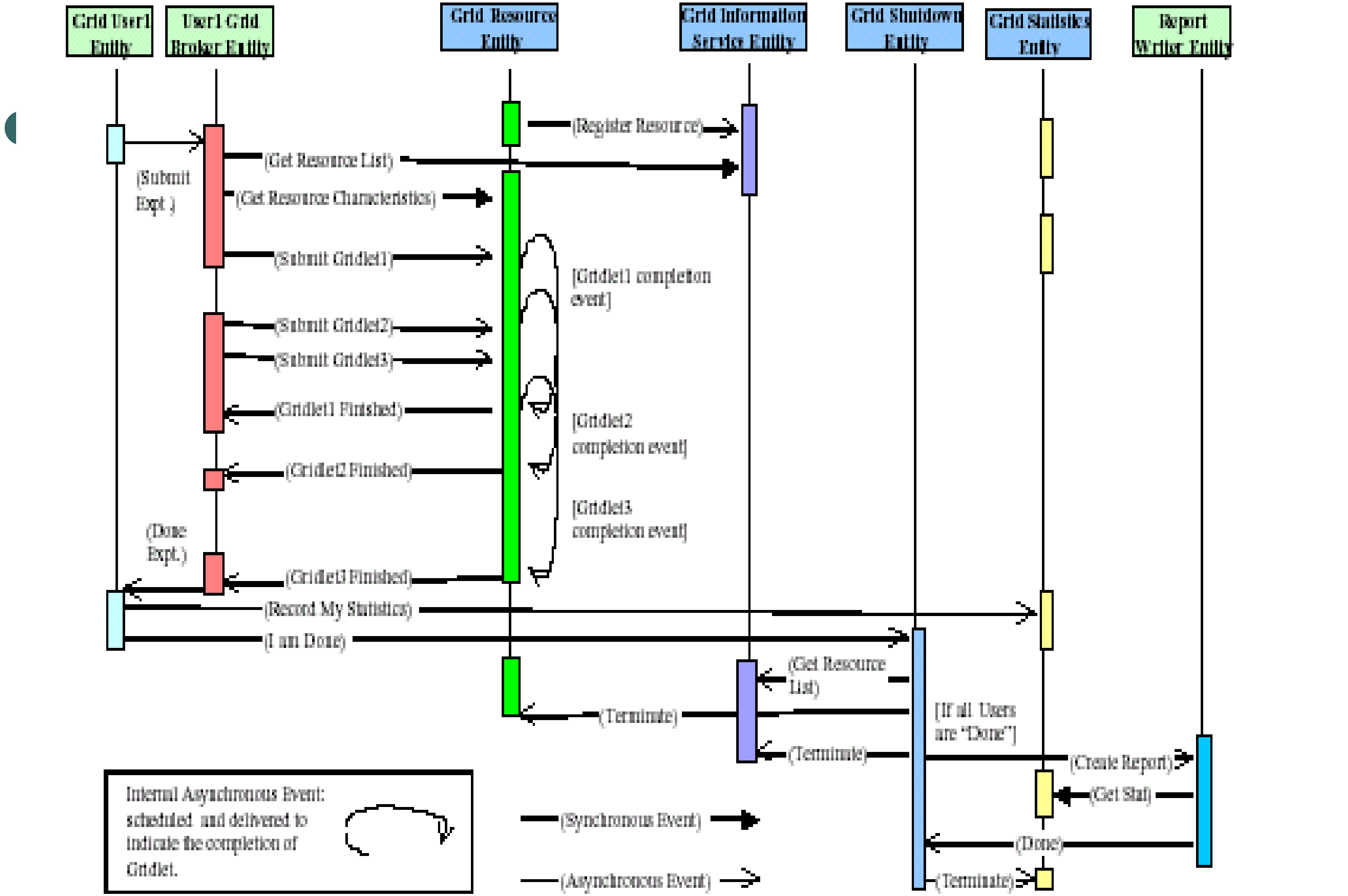


Figure 6: An event diagram for interaction between a space-shared resource and other entities.