The Anatomy and The Physiology of the Grid

B.Ramamurthy

Based on The Anatomy and The Physiology papers about the "grid"

Introduction

- Two papers that give an overview of the components (anatomy) and the functionality (physiology) of the grid. These are:
  1. The Anatomy of a grid: Enabling Virtual Organizations by I. Foster et al.
  2. The Physiology of the Grid By I. Foster et al.
- We will discuss the "problem Space" that the grid addresses.

Grid Technology Problem Space

- Grid technologies and infrastructures support the sharing and coordinated use of diverse resources in dynamic, distributed “virtual organizations”.
- Grid technologies are distinct from technology trends such as **Internet, enterprise, distributed and peer-to-peer computing**. But these technologies can benefit from growing into the “problem space” addressed by grid technologies.

Virtual Organization: Problem Space

- **An industrial consortium formed to develop a feasibility study for a next generation supersonic aircraft undertakes a highly accurate multidisciplinary simulation of the entire aircraft.**
- **A crisis management teams responds to a chemical spill by using local weather and soil models to estimate the spread of the spill, planning and coordinating evacuation, notifying hospitals and so forth.**
- **Thousands of physicists come together to design, create, operate and analyze products by pooling together computing, storage, networking resources to create a Data Grid.**
- **A data grid + a compute grid to support cure/vaccine for any number of diseases “plaguing” the earth.**

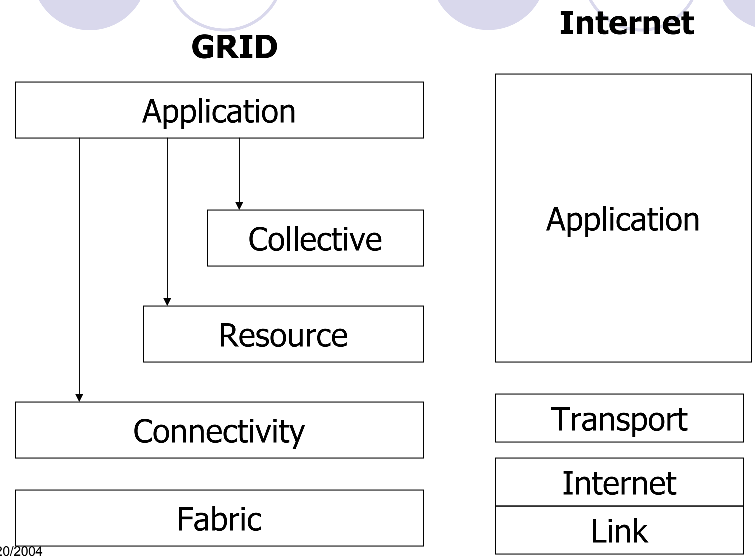
Resource Sharing Requirements

- Members should be trustful and trustworthy.
- Sharing is conditional.
- Should be secure.
- Sharing should be able to change dynamically over time.
- Need for discovery and registering of resources.
- Can be peer to peer or client/server.
- Same resource may be used in different ways.
- All these point to a well defined architecture and protocols.

Grid Definition

- **Architecture identifies the fundamental system components, specifies purpose and function of these components, and indicates how these components interact with each other.**
- **Grid architecture is a protocol architecture, with protocols defining the basic mechanisms by which VO users and resources negotiate , establish, manage and exploit sharing relationships.**
- **Grid architecture is also a services standards-based open architecture that facilitates extensibility, interoperability, portability and code sharing.**
- **API and Toolkits are also being developed.**

Architecture



Fabric Layer

- Fabric layer: Provides the resources to which shared access is mediated by Grid protocols.
- Example: computational resources, storage systems, catalogs, network resources, and sensors.
- Fabric components implement local, resource specific operations.
- Richer fabric functionality enables more sophisticated sharing operations.
- Sample resources: computational resources, storage resources, network resources, code repositories, catalogs.

Connectivity Layer

- Communicating easily and securely.
- Connectivity layer defines the core communication and authentication protocols required for grid-specific network functions.
- This enables the exchange of data between fabric layer resources.
- Support for this layer is drawn from TCP/IP's IP, TCL and DNS layers.
- Authentication solutions: single sign on, etc.

Resources Layer

- Resource layer defines protocols, APIs, and SDKs for secure negotiations, initiation, monitoring control, accounting and payment of sharing operations on individual resources.
- Two protocols information protocol and management protocol define this layer.
- Information protocols are used to obtain the information about the structure and state of the resource, ex: configuration, current load and usage policy.
- Management protocols are used to negotiate access to the shared resource, specifying for example qos, advanced reservation, etc.

Collective Layer

- Coordinating multiple resources.
- Contains protocols and services that capture interactions among a collection of resources.
- It supports a variety of sharing behaviors without placing new requirements on the resources being shared.
- Sample services: directory services, coallocation, brokering and scheduling services, data replication service, workload management services, collaboratory services.

Applications Layer

- These are user applications that operate within VO environment.
- Applications are constructed by calling upon services defined at any layer.
- Each of the layers are well defined using protocols, provide access to useful services.
- Well defined APIs also exist to work with these services.
- A toolkit Globus implements all these layers and supports grid application development.

Open Grid Services Architecture

- **Builds on concepts and technologies from the Grid and Web services communities.**
- **Defines a uniform exposed service semantics.**
  - **Defines standard mechanisms for creating, naming, and discovering transient grid service instance;**
  - **Provides location transparency and multiple protocol bindings**
  - **Supports integration with underlying native platform facilities.**
  - **Defines WSDL definition for creating sophisticated distributed system including lifetime management, change management, and notification. It also supports authentication, authorization, and delegation.**

An Open Grid-services Architecture

- Service orientation and virtualization:
  - A service is a network-enabled entity provides some capability.
  - Virtualization allows the composition of services to form lower-level resources.
  - WSDL allows for multiple bindings of a single interface, including distributed communication protocols.

Service Semantics

- **OGSA defines the semantics of a Grid Service instance: how it is created, how its lifetime is determined, how to communicate and so on.**
- **WSDL** is used to define **standard interfaces** that address discovery, dynamic service creation, lifetime management, notification, and manageability.
- **Transient services** along with the conventional persistence services. Example: video conferencing, where is QoS is important.
- **Upgradeability**: services within a complex distributed system must be independently upgradeable. Needs reliable service invocation and authentication.

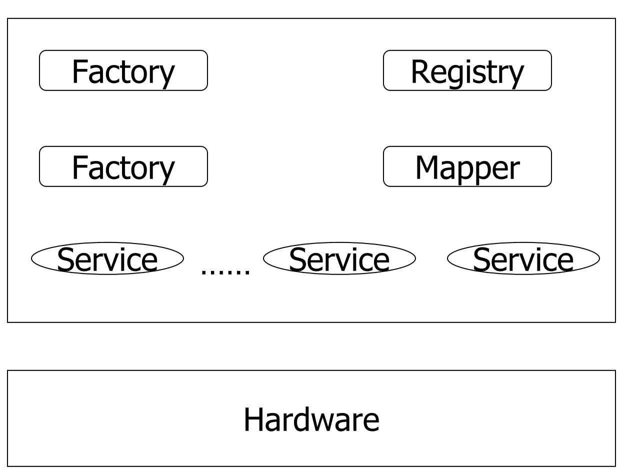
Building Virtual Organization

- Applications create transient services to discover and determine the properties of available services.
- OGSA's Factory, Registry, GridService, and HandleMap interfaces support the creation of transient service instances and the discovery of services associated with a VO.

Possible Environments

- Simple hosting environment: Set of resources located within a single administrative domain.
  - Example: J2ee application server, MS's .net system, or a Linux cluster.
- Virtual Hosting environment: VO span heterogeneous, geographically distributed "hosting environments", a combinations several simple environments.
- Collective operations: A "virtual hosting environment" that provides VO participants with more sophisticated, virtual "collective" or "end-to-end" services.

VO Organization



We will look into more "physiology" of the grid next lecture.