The Anatomy of the Grid
Enabling Scalable Virtual Organizations

Carl Kesselman
USC/Information Sciences Institute

Ian Foster, Steve Tuecke
Argonne National Laboratory

Some slides based on those by B.Ramamurthy

Grid Computing – Making the Global Infrastructure a Reality
Fran Berman, Geoffrey Fox, and Tony Hey
2002 John Wiley & Sons, Ltd.
Chapter 6

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. - this is the next reading assignment

First we review 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 SARS.**

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 well defined architecture and protocols.

Programming & Systems Problems

- The programming problem
  - Facilitate development of sophisticated apps
  - Facilitate code sharing
  - Requires programming environments
    - APIs, SDKs, tools
- The systems problem
  - Facilitate coordinated use of diverse resources
  - Facilitate infrastructure sharing
    - e.g., certificate authorities, information services
  - Requires systems
    - protocols, services

The Systems Problem: Resource Sharing Mechanisms That …

- Address security and policy concerns of resource owners and users
- Are flexible enough to deal with many resource types and sharing modalities
- Scale to large number of resources, many participants, many program components
- Operate efficiently when dealing with large amounts of data & computation

Aspects of the Systems Problem

1) Need for interoperability when different groups want to share resources
   - Diverse components, policies, mechanisms
   - E.g., standard notions of identity, means of communication, resource descriptions
2) Need for shared infrastructure services to avoid repeated development, installation
   - E.g., one port/service/protocol for remote access to computing, not one per tool/appln
   - E.g., Certificate Authorities: expensive to run
- A common need for protocols & services

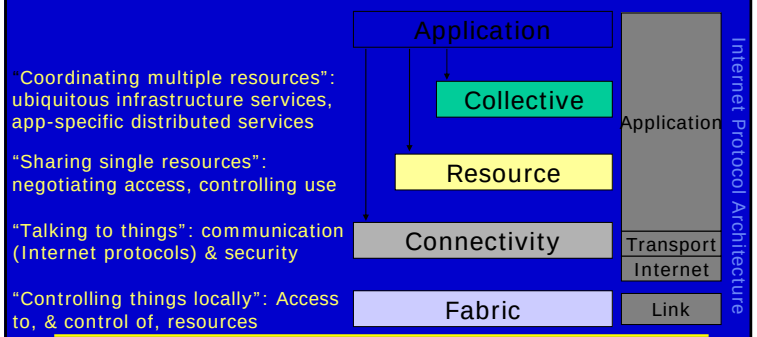
Hence, a Protocol-Oriented View of Grid Architecture, that Emphasizes …

- Development of Grid protocols & services
  - Protocol-mediated access to remote resources
  - New services: e.g., resource brokering
  - "On the Grid" = speak Intergrid protocols
  - Mostly (extensions to) existing protocols
- Development of Grid APIs & SDKs
  - Interfaces to Grid protocols & services
  - Facilitate application development by supplying higher-level abstractions
- The (hugely successful) model is the Internet

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 (Application Programing Interface) and SDKs – (Software Development Toolkits) are also being developed.**

Layered Grid Architecture (By Analogy to Internet Architecture)

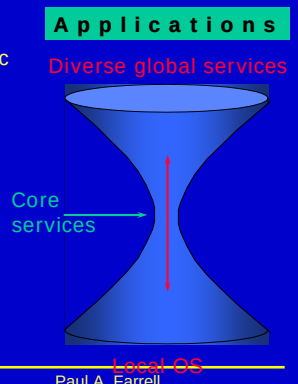


Important Points

- Built on Internet protocols & services
  - Communication, routing, name resolution, etc.
- "Layering" here is conceptual, does not imply constraints on who can call what
  - Protocols/services/APIs/SDKs will, ideally, be largely self-contained
  - Some things are fundamental: e.g., communication and security
  - But, advantageous for higher-level functions to use common lower-level functions

The Hourglass Model

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



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.

Globus Toolkit 2 (GT2) - Fabric

- Designed to use vendor supplied protocols & interfaces
- Supplies version in case these are absent e.g. enquiry software for discovering structure and state information
- Exception: Resource management – must be supplied externally
- Local resource managers
- Option if absent : GARA (General-purpose Architecture for Reservation and Allocation), enhancements of PBS, Condor

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 protocol stack (IP/ICMP, TCP/UDP and DNS/OSPF/RSVP etc).
- Authentication solutions: single sign on, delegation, integration with local security solns, user-based trust relationships

GT2 Connectivity Layer Protocols & Services

- Communication
  - Internet protocols: IP, DNS, routing, etc.
- Security: Grid Security Infrastructure (GSI)
  - Builds on TLS (Transport Layer Security)
  - Uniform authentication, authorization, and message protection mechanisms in multi-institutional setting
  - Single sign-on, delegation, identity mapping
  - Public key technology, SSL, X.509, GSS (Generic Security Service)-API
  - Stakeholder control of auth via GAA (Generic Authorization and Access) interface
  - Supporting infrastructure: Certificate Authorities, certificate & key management, …

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.

GT2 Resource Layer - Protocols & Services

- Grid Resource Allocation Management (GRAM)
  - Remote allocation, reservation, monitoring, control of compute resources
- GridFTP protocol (FTP extensions)
  - High-performance data access & transport
- Grid Resource Information Service (GRIS)
  - Access to structure & state information
  - GRIP (GRI Protocol) information protocol
  - GRRP (Grid Resource Registration Protocol) for registering
- Others emerging: Catalog access (LDAP based), code repository access, accounting, etc.
- All built on connectivity layer: GSI & IP

GT2 Resource Layer - Protocols & Services

- GT defines client-side Cand Java APIs and SDKs for each protocol
- Also server-side SDKs and servers
  - GRIS implements server-side LDAP
  - Gatekeeper : GSI authenticated *inetd* that speaks GRAM
  - GSS (Generic Security Services) API used for authentication credentials and to provide integrity/privacy within SDKs/servers enabling security services at Connectivity layer

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; monitoring/disgnostic; data replication service; grid-enabled programming systems; workload management services; software discovery services; community authorization servers; community accounting/payment services; collaboratory services.

GT2 Collective Layer - Protocols & Services

- Index servers aka metadirectory services
  - MDS (Meta Directory Services) : (GRIS/GIIS)
    - Based on LDAP protocol
  - Custom views on dynamic resource collections assembled by a community
- Resource brokers (e.g., Condor Matchmaker)
  - Resource discovery and allocation
- Replica catalogs
- Replication services
- Co-reservation and co-allocation services : DUROC
- Workflow management services
- Online credential repository (MyProxy)

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.

Protocols, Services, and APIs Occur at Each Level