Grid Middleware

Vu Chi Hieu
Tran Thi Thanh Nga
Huynh Nguyen Tin
Outline

- Introduction
- Architecture
- Software
- Challenges
- Standardization
- Services
- Grid Middleware Toolkits
Introduction

“A grid is a system that:

- Coordinates resources that are not subject to centralized control...
- Using standard, open, general-purpose protocols and interfaces...
- To deliver nontrivial qualities of services”

Ian Foster (What is the Grid? A Three Point Checklist)
Grid Middleware

- An evolving layer of software infrastructure residing between the Grid Fabric and Applications.
  - **Fabric layer**: protocols, application interfaces, toolkits that allow development of services and components to access locally controlled resources (computers, storage resources, networks, sensors)
  - **Application layer** comprises the users’ applications that are used within a virtual organization.
Architecture
Architecture

- Grid Middleware:
  - **Connectivity layer**: include the necessary Grid-specific core communication and authentication support to perform secure network transactions with multiple resources with the Grid fabric.
  - **Resource layer**: contains protocols that enable secure access and monitoring by collective operations.
  - **Collective layer**: is concerned with the coordination of multiple resources and defines collections of resources that are part of a virtual organization (ex: directories for resource discovery and brokers for distributed task and job scheduling).
Software

• Requirement:
  • shared, reused and extended by others in a multitude of higher-level frameworks.

Includes protocols, data structures, and objects that can be accessed through convenient APIs and classes.
Challenges

- Wide user community ranging from application developers to system administrators.
- Wide range of infrastructure.
  
  → must identify a path that leads to an integral software environment developed by the community.
- The maintenance, management and policy issues.
  
  ➔ social challenge.
Standardization

- Global Grid Forum (GGF):
  - To promote and support the development, deployment, and implementation of Grid technologies and applications through the creation and documentation of “best practices” – technical specifications, user experiences and implementation guidelines.
Services

- Elementary Grid Middleware Services:
  - Job execution services, security services, information services and file transfer services
- Advanced Grid Management Services
  - File Management, task Management, information management
Services
Grid Middleware Toolkits

- Globus Toolkit 4
- UNICORE
- gLite
Globus Toolkit 4

- An open source toolkit organized as a collection of loosely coupled components.
  - Services
  - Programming libraries
  - Development tools designed for building Grid-based applications.
Globus Toolkit 4
Globus Security - Grid Security Infrastructure

- **Authentication and Authorization:**
  - libraries
  - tools for controlling access to services and resources
  - Framework that enables the use of different authorization methods.

- **Delegation:** includes a service that delegates credentials to a container.
Globus Security - Grid Security Infrastructure

- **Community Authorization**: virtual organizations can use the Community Authorization Service (CAS) to manage authorization policies for VO resources.

- **Credential Management**:
  - SimpleCA
  - A simple Certificate Authority for users without access to a full-blown CA
  - MyProxy
  - An online credential repository.
Data Management

- Provide for the discovery, transfer and access of large data.

- **GridFTP:**
  - A functional GridFTP server
  - Several client-side utilities.
  - The GridFTP protocol is specially optimized to transfer large amounts of data between hosts.

- **RFP (Reliable File Transfer):**
  - Is a WSRF-enabled service that use GridFTP to move large amounts of data.
Data Management (cont)

- **Replica Location Service (RLS):** allows users to keep track of where different replicas of a dataset are located in a virtual organization.

- **Data Replication Service (DRS):** uses RLS and RFT to guarantee that local copies of replicas are available to the hosts that need them.

- **OGAS-DAI (OGSA Data Access and Integration):** provides a framework to access and integrate datasets on a Grid which might be available in different formats (plain text, databases, XML,…).
Execution Management

- Deal with the deployment, scheduling and monitoring of executable programs \( \rightarrow \) jobs.
- Grid Resource Allocation & Management (GRAM):
  - the heart of GT Execution Management
  - Providing services to deploy and monitor jobs on a Grid.
- Community Scheduler Framework (CSF): provides a single interface to different resource schedulers.
- Workspace Management: allows users to dynamically create and manage workspaces on remote hosts.
- Grid Telecontrol Protocol: provides a WSRF-enabled service interface for telecontrol (control of remote instruments).
Information Services

- Referred to as the Monitoring and Discovery System (MDS).
- Includes a set of component to monitor and discover resources in a virtual organization.
- Index Service: is used to aggregate resources of interest to a VO.
- Trigger Service: collects data from resources, but is configured to perform certain actions based on that data.
- WebMDS: provides a web browser-based view of data collected by GT4 aggregator services.
Common Runtime

- Provide a set of fundamental libraries and tools for hosting existing services as well as developing new services.
- C Runtime: tools, libraries and a WS hosting environment for C developers.
- Python Runtime: tools + libraries + WS hosting environment for Python developer.
- Java Runtime: tools + libraries + service hosting environment for Java developers.
Grid Middleware Toolkits

- Globus Toolkit 4
- UNICORE 6
- gLite
WHY UNICORE?

- Define job workflows in abstract manner.
- Immediate portability of job definitions for other systems with other architecture.
- Applications across multiple supercomputer/cluster
UNICORE 6

- Integrated, complete Grid middleware stack:
- Easy to install, configure, administrate and monitor
- Excellent application and workflow support
- Suited for both high performance and high-throughput usage.
- Wide variety of clients: GUI, command line, APIs
- Supports common operating and batch systems
  - UNIXes, MacOS X, Windows.
  - TSIs: embedded, Torque, LoadLeveller, LSF, SGE,...
  - Existing UNICORE 5 TSI installations can be re-used.
- Active developers, responsive to user wish, quick and efficient support.
UNICORE 6

• UNICORE 5 plus ...
  • Openness, extensibility, interoperability
    • Service-oriented
    • Web services foundation
    • Communications are no longer tied to the Java platform.
    • Highly extensible core engine.
UNICORE 6

- ... added value
  - Performance and scalability
    - Fast default file transfer
    - Tested and proven scalability of core components
  - State of the art software
    - Uses Java 5 features (generics, JMX)
    - Jetty 6 web server
    - XFire SOAP stack
    - HSQLDB embedded database
    - ...

[26]
UNICORE 6 Architecture

- Client layer
- Service layer
- System layer
UNICORE 6 Architecture
Client layer

- **UCC (UNICORE command line client):**
  - Allows users to access all features of the UNICORE service layer in a shell or scripting environment.
  - Allows to run jobs, monitor their status and retrieve generated output, both in single job mode or in a powerful and flexible batch mode for multiple jobs.
  - Workflows can be submitted and controlled with the UCC.
Client layer

- URC (UNICORE Rich Client):
  - Offers the full set of functionalities to the users in a graphical representation.
  - Targets a wide range of users with varying Grid and IT experience.
  - Provides a useful graphical view of the Grid, which can be filtered in order to find specific resources, services or files.
Client layer

- HiLA (High Level API for Grid Applications):
  - Allows simple development of clients with just a few lines of code for otherwise complex functionality.
  - Provides a single interface with multiple implementations for UNICORE 5, UNICORE 6, OGSA-BES.

- Portal: external component, not developed and not under control by the UNICORE community.
  - UNICORE 6 services can also be accessed from portals.
Service layer

- Comprises all services and components of the UNICORE Service-Oriented Architecture (SOA) based on WS-RF 1.2, SOAP and WS-I standards.

- Gateway:
  - Acts as the entry point to a UNICORE site and performs the authentication of all incoming requests.
Service layer

- **XNJS:**
  - is the job management and execution engine; the heart of a UNICORE 6 site.
  - Provides storage resources, file transfer services and job management services.

- **IDB (Incarnation Data Base):**
  - Performs the job incarnation, namely the mapping of the abstract job description in JSDL to concrete job description for a specific resource.
  - Information about available applications and resource characteristics has to be defined in this database.
Service layer

- **XUUDB:**
  - is a Web service ➔ can be used from multiple UNICORE installations.

- **UVOS (UNICORE VO Service):**
  - An alternative to the XUUDB, a Virtual Organization (VO) service can be for user authorization.
  - Use the SAML standard and offers a wide variety of features.

- **Registry:**
  - build-up and operate a distributed UNICORE infrastructure.
  - is contacted by the clients in order to “connect to the Grid”.
  - runs in UNICORE’s WS-RF hosting environment.
Service layer

- CIS (Common Information Service):
  - is the information services of UNICORE 6.
  - gathers both static and dynamic information from all connected XNJS.

- Workflow Engine
  - Two –layered architecture:
    - workflow engine
    - The service orchestrator layers.

- Service Orchestrator:
  - execute the individual tasks in a workflow
  - handle job execution
  - monitor on the Grid.
System layer

- **TSI (Target System Interface):**
  - In the TSI component the abstracted commands from the Grid are translated to system-specific commands.
  - is performing the proper setting of the userID and invocation of his/her environment.

- **Uspace:**
  - Is UNICORE’s job directory. A separate directory exists for every job, where the XNJS and TSI stores all input data and where stdout and stderr are written to.

- **External Storage:**
  - For a transfer of data from and to external storage, the GridFTP transfer protocols can be used.
Grid Middleware Toolkits

- Globus Toolkit 4
- UNICORE
- gLite
gLite

- is an integrated set of components designed to enable resource sharing.
- was produced by the EGEE project and it is currently being developed by the EMI project.
- contributions from many other projects, including LCG and VDT.
- is to construct different services ('node-types') from these components and then ensure easy installation and configuration on the chosen platforms (currently Scientific Linux versions 4 and 5, and also Debian 4 for the WNs).
gLite

• provides the user with high level services for
  • scheduling and running computational jobs
  • accessing and moving data
  • obtaining information on the Grid infrastructure as well as Grid applications

• Follow Service Oriented Architecture
  • Easily interactive of services
  • Easily adaptive to open standard
  • Services can work independently
  • Services can work together

• Introduce a framework for in application development in Grid environment
gLite

- A layer between services and grid resources
gLite - Architecture
gLite

- All grid services are accessible via:
  - API - Application Programming Interface
  - CLI - Command Line Interface
gLite

Security Services:

- **Authentication**
  - identify entities (users, systems, and services)
  - Based on personal X.509 Certificates (issued by trusted CAs).
- **Authorization**
  - allows or denies access to specific services and resources
  - three basic authorization models, classified as *agent*, *push* and *pull*.
- **Auditing**
  - monitoring and providing information for post-mortem analysis of security related events.
- **Dynamic Connectivity Service**
  - tackle the problems that arise when connectivity is restricted by the resource owner.
gLite

Information & Monitoring Services

- R-GMA Components: provides a uniform method to access and publish distributed information and monitoring data.
- Job Management Services (Application Service)
  - Java logging service, log4j, Apache/Chainsaw (for other languages).
- Discovery Service
  - Provides a standard set of methods for locating Grid services
- Network Performance Monitoring
  - define a standard interface to monitoring frameworks.
gLite

Job Management Services

- Accounting
  - accumulates information about the usage of Grid resources
- Computing element
  - represent a computing resource (a cluster, supercomputers, single workstations).
- Workload Management
  - comprises a set of Grid middleware components responsible for the distribution and management of tasks across Grid resources
- Job Provenance
  - keep track of the definition of submitted jobs, execution conditions and environment.
- Package Manager
  - is a helper service that automates the process of installing, upgrading configuring, and removing software packages from a shared area
gLite

Data Services

- **Storage Element (SE):**
  - represent a storage resource

- **Metadate catalog services:**
  - keep track of the data location and relevant metadata (filesize,...)

- **data movement services:**
  - allow for efficient managed data transfers between SEs.
gLite

Helper Services

- Configuration and Instrumentation Service – Queries service state.
- Agreement Service – Implements a communication protocol for the Service Level Agreements.
- Bandwidth Allocation & Reservation service (BAR) – Controlling, Balancing and Manage Network flows.
Reference

- [http://www.unicore.eu/](http://www.unicore.eu/)
- [http://www.globus.org/toolkit/](http://www.globus.org/toolkit/)
- [http://glite.cern.ch/](http://glite.cern.ch/)
- Qusay H. Mahmoud, Middleware for Communications, Wiley, 2004