Interoperability between
EGEE gLite and CNGrid GOS

Yaodong CHENG
IHEP, Chinese Academy of Sciences
ISGC 2008
Outline

- Major issues of interoperability between different grid infrastructures
- Status of our work in interoperability
- Gateway-based interoperability
Use of Grid

Write problem-solving code

“Adapt” to middleware

Submit to Grid

security

middleware

Publish

Dispatch to resources

Select resources

Stage data

Accounting

Steering and visualisation
Common issues in interoperability

- Job Description Language
- Job Submission
- Resource Discovery
- Resource Selection
- Data Staging
- Cross-domain Security
Job Description Languages

- Specify the job to run and how it will run
  - Different systems have their own job description languages
  - Choose to use the same description language or to do conversion
    - JSDL to JDL and JDL to JSDL conversion have been done in the gateway component in EUChinaGrid project
    - JSDL is a preferred job descriptor language, adopted by OGSA-BES

<table>
<thead>
<tr>
<th>Condor</th>
<th>Complex almost programming language (ClassAds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNGrid GOS</td>
<td>Job Submission Description Language (JSDL)</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Variation on the Condor ClassAds language (JDL)</td>
</tr>
</tbody>
</table>
Job Submission

- The way of submitting jobs to the Grid:
  - Different systems have different job submission mechanisms
  - In EUChinaGrid Project, we support interface similar to OGSA-BES and plan to provide a complete implementation of OGSA-BES;

<table>
<thead>
<tr>
<th></th>
<th>Command line, Web Service, port, Standard DRMAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condor</td>
<td></td>
</tr>
<tr>
<td>CNGrid GOS</td>
<td>Portal, Web Service</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Command line, API, (Some) Web Service</td>
</tr>
</tbody>
</table>
Resource Discovery

- Find availability of resources
  - Having a good knowledge of the current state of the resources helps in resource selection

- Three different schemas are widely used:
  - Glue Schema used by OSG, EGEE and Teragrid, mapped to LDAP, XML and the relational model and CNGrid GOS will support GLUE schema in the upcoming version
  - ARC schema used from NDGF
  - CIM schema used by NAREGI

- Use the same schema or perform necessary conversion for interoperability

<table>
<thead>
<tr>
<th>Condor</th>
<th>Resources publish themselves to the scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNGrid GOS</td>
<td>Resource register themselves to router service</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Resources publish themselves to an information service that the WMS can query</td>
</tr>
</tbody>
</table>
Resource Selection

- Select the best resources to run the job
  - Ensure that each job is placed on the most proper resource
  - A big problem for interoperability
    - Difficult to determine whether the received batch job should be dispatched to other grid middlewares or not
    - Usually resource selection is the core component of grid middleware and difficult to modify for interoperability

<table>
<thead>
<tr>
<th>Condor</th>
<th>Jobs and resources are “matched” together. Jobs will be launched when an idle resource matching the requirements is found</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNGrid GOS</td>
<td>Meta Schedule choose resource according to some predefined condition</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Workload Management Services are used to select the best CE to run the job</td>
</tr>
</tbody>
</table>
Data Staging

- Getting the data into and out of the resources

- **Data Staging interoperability** focuses on the following fields
  - Point to point movement of data between storage in different grids
    - For example: Grid-ftp interoperability or OGSA-ByteIO
  - Usage of managed resources and their APIs (SRM, SRB):
    - For example: SRM interoperability

<table>
<thead>
<tr>
<th>Condor</th>
<th>Jobs are given a virtual file space with read and write operations being passed back to the submission node</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNGrid GOS</td>
<td>Using FTP or HTTP as underlying transport protocols</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Jobs can be staged out or provided by streams. Storage elements can hold files</td>
</tr>
</tbody>
</table>
Security

- Three security issues involved in grid environment
  - Authentication
    - How do we positively identify users and resources?
  - Authorisation
    - How to do the authorization operation?
  - Accounting
    - How to do the accounting operation?
Security

- Protect underlying resources
  - Authentication and Authorisation are key points
  - Need to develop a level of trust for both users and the resource owners
  - Cross-domain security is a big challenge. We just made a first simple approach in EUChinaGrid Project.

<table>
<thead>
<tr>
<th>System</th>
<th>Security Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condor</td>
<td>Uses public key infrastructure x509 &amp; Proxy</td>
</tr>
<tr>
<td>CNGrid GOS</td>
<td>Uses public key infrastructure x509 &amp; Proxy</td>
</tr>
<tr>
<td>EGEE gLite</td>
<td>Uses public key infrastructure x509 &amp; Proxy + Annotations on the certificates</td>
</tr>
</tbody>
</table>
Overview of our work in interoperability

- **Our major work**
  - Design of a flexible gateway and proposal to a generic design for more complex scenarios
    - Use SEDA model as the task process tool
    - Use IoC model as the configuration and assembly tool
  - CNGrid GOS JobManager Framework extension
  - GLite LCG-CE JobManager Framework extension

- **Works achieved and going on**
  - First implementation of a testbed in IHEP (CAS) and in Catania (INFN)
    - running stably for about three months
    - Processed more than 1,500 batch jobs (including both GOS to GLite and GLite to GOS)
  - Focusing on data interoperability
Role of Gateway

- A logical component
  - Interface conversion
  - Function mapping
- Support the following features
  - Transparent to end users of different grid infrastructures
  - Easy to extend
  - Concurrency and high throughout
  - Standalone deployment or integrated underlying grid middleware
Gateway design

- Our Gateway design heavily depends on SEDA and IoC models

  SEDA model
  - SEDA--Staged Event Driven Architecture
  - Firstly proposed by Matt Welsh, David Culler, and Eric Brewer of UC Berkeley
    - Support massive concurrency, high throughout
    - Simplify the construction of well-conditioned Internet services
  - In our design, process is divided into independent basic stages of different pipelines for different purpose such as GLite-to-GOS batch job forwarding, and so on

  IoC model
  - IoC--Inversion of Control
    - Provide loose coupling among different modules and allow easy reuse of basic modules
    - Assemble new module easily and quickly
  - In our design, HiveMind 1.1 released under LGPL license is used as IoC container
Core components of our gateway

- Core components of our gateway
  - Pipelines for different purposes
    - Composed of different basic processing stages
    - Used for different purposes such as forwarding batch jobs from GOS to GLite and vice versa
  - Scheduler
    - Execute processing stages at fixed rate
    - One to one mapping between Pipeline and scheduler
  - Threads pool
    - Improves performance
    - One to one mapping between Thread pool and scheduler
  - Processing stages in the same pipeline perform different concrete functions such as StageIn, StageOut, and so on
Detailed description of gateway components

Extended GOS forwards batch job to gateway

GOS ➔ scheduler ➔ Thread Pool ➔ WMProxy

Thread Pool

Pipeline for GOS to GLite

Pipeline for GLite to GOS

1 Different colors in pipeline stand for different stages performing concrete functions such as data stageIn, data stageOut, and so on

2 Different pipelines use different thread pools and schedulers

Extended LCG-CE forwards batch job to gateway

Extended GOS

batch job

batch job

batch job

batch job

Scheduler executes stage in pipeline using idle thread from thread pool

idle threads pool used in schedule
Batch job level interoperability

- Extend JobManager in both GLite and GOS
- Extend GLite LCG-CE JobManager Framework
  - LCG-CE JobManager Framework is closely coupled with resource scheduling mechanism of GLite
  - Relatively difficult to extend, cost a lot time
- Provide Broker plugin for GOS JobManager framework
- Sandbox mode data transfer
- A fast approach for cross-domain security scenario
Testbed in Catania, INFN

- Portal or WS-Client
  - portal.ct.infn.it
  - gos.ct.infn.it
  - OpenPBS

- Gateway component
  - PipeLine4GLite
  - PipeLine4GOS
  - Extended LCG-CE

- Job wrapper
  - gLite-gos.ct.infn.it

- WMS
  - glite-rb2.ct.infn.it
  - WMProxy
  - GLite-UI
  - Command Line

- JDL
- RSL
- JSDL
- WMProxy
- Portal or WS-Client
Batch job level interoperability process

**JSDL**

**PipeLineForGOS**
1. Data Transfer
2. Submit job to GOS

**GOS Node**
- portal.ct.infn.it
- gos.ct.infn.it

**PipeLineForGLite**
1. Convert JSDL to JDL
2. Data Transfer
3. Submit job to WMProxy

**WMProxy**
- glite-rb2.ct.infn.it

**Extended LCG-CE**
1. Extend Globus JobManager used in LCG-CE
2. Convert RSL to JSDL
3. Submit batch job to PipeLineForGOS

**PipeLineForGOS**
1. Data Transfer
2. Submit job to GOS

**GOS Node**
- portal.ct.infn.it
- gos.ct.infn.it

**WMS**
- Dispatch batch job

**GLite-UI**
- Submit JDL

**Extended LCG-CE**
1. Convert JSDL to JDL
2. Data Transfer
3. Submit job to WMProxy
Data Transfer

- Data transfer between CNGrid and EGEE supports two different modes:
  - **Sandbox-Based Data Transfer**, for small scale data transfer:
    - All data transfer operations pass the batch job gateway and suitable for small scale data transfer scenario
    - Batch job gateway acts as data transfer center and has two different roles at the same time
      - GridFTP client: Gateway upload/download necessary data to/from GLite WMS;
      - FTP server: GOS upload/download data to/from gateway component;
  - **SRM-Based Data Transfer**, for large scale data transfer:
    - There is a separate data interoperability gateway which supports SRM specification and can be interacted through multiple protocols including GridFTP and FTP;
    - CNGrid GOS/EGEE gLite interact directly with data interoperability gateway;
      - gLite WN upload/download data files using gridFTP protocol;
      - CNGrid GOS and Batch job interoperability Gateway upload/download using SRM Specification which is based on FTP protocol;
Sandbox-Based Data Transfer Scenario

FTP protocol is used in CNGrid, so we use FTP protocol to transfer data between GOS node and the gateway.

GridFTP protocol is used in gLite environment, so we use GridFTP protocol to transfer data between gLite RB and the gateway.

Roles of gateway in data transfer scenario:
- FTP Client
- FTP Server
- GridFTP Client
- GridFTP Server
- Gateway Component

FTP Client

FTP Server

GridFTP Client

GridFTP Server

FTP Protocol

GridFTP Protocol

GOS node

gLite SE or gLite WMS

yaodong cheng, ihep, cas

isgc 2008 taipei, 04.2008
SRM-Based Data Transfer Scenario

- Client
- Batch-Job Interoperability GW
  - GridFTP Protocol
  - SRM Specification
- SRM-Based Multiple Protocol SE
  - GridFTP Protocol
  - SRM Specification
- gLite CE
- gLite WN
- gLite WMS (WMSProxy)
- gLite-UI

• GridFTP Protocol
• SRM Specification
Security issues

- A first simple approach for cross-domain security
  - Some users in GOS and GLite are predefined for interoperability purpose, a static approach
    - Requests from GOS to GLite use predefined voms proxy
    - Requests from GLite to GOS use predefined name
  - User Management module is designed to keep mapping of relationships

- Security token service
  - Used to keep, distribute, exchange and verify security tokens between GOS and GLite and provide dynamic approach
  - MyProxy Server is currently used to store temporary security token. Plan to replace it with newly developed security token service
Role of security token service

Roles of security token service in cross-domain security scenario

Security token Service

Gateway

WS-Trust Client

security token

GOS

obey the way that GOS used for service invocation

GLite

obey the way that GLite used for service invocation

obey the way that GOS used for service invocation
What to do next

- Cross-domain Security
  - Security token service based token distribution in cross-domain scenario:
    - More generic solution for cross-domain security token distribution
    - Comply with WS-Trust specification
  - Comply with some work of OGF GIN Group
    - OGSA-BES
    - SRM
  - Support real grid application interoperability between CNGrid and EGEE
    - POSIX application is supported now
    - Choose from applications supported by EUChinaGrid project
Thanks to Yongjian WANG, Diego Scardaci, BINGHENG YAN, GANG CHEN, Giuseppe Andronico and other EUChinaGrid members for their contribution to this work!
Thanks for your attentions

Any Questions?