Job Invocation Interoperability between NAREGI Middleware Beta and gLite

Hidemoto Nakada (AIST), Hitoshi Sato (Titech),
Kazushige Saga (NII), Masayuki Hatanaka (Fujitsu),
Yuji Saeki (NII), Satoshi Matsuoka (Titech, NII)
Background

Recent development of Grid middleware stacks

- Globus, UNICORE, NAREGI Middleware, gLite
- Some of them are used in production grids
- Resources cannot be shared by grids operated by different middleware stacks

→ Interoperation is required
Background (2)

- OGF (Open Grid Forum) GIN-CG
  - Grid Interoperation Now Community Group

- Try to make grid middleware stacks interoperable using currently available technologies
Goal

As a part of GIN-CG, perform interoperation experiments between the following two grid middleware stacks:

- NAREGI Middleware Beta
- gLite from EGEE

Interoperability

- Security Mechanisms
- Information Service
- Job Submission
- Large-scale Data Transfer
Outline

Architecture of the Grid middleware stacks
  - NAREGI Middleware beta
  - gLite

Strategies for interoperation and implementation

Measurement Results
What are ‘grid middleware stacks’

Assumptions
- Each ‘grid’ involves several ‘sites’.
- Each ‘site’ has several computers managed by some kind of ‘local scheduler’

Grid middleware stacks
- Get job execution request from users and dispatch them to ‘proper’ site, securely.
  - ‘Proper’ - load distribution, Virtual Organization Management
  - ‘Secure’ - Authentication, Authorization
- Local schedulers are responsible for load distribution inside the sites.
General configuration of Grid Middleware Stacks

- **User submit jobs**
- **Gathers info.**
- **Manage Each site**
- **Batch queuing system**
  - SGE, PBS, Condor

**Client**
- Determines site to execute

**Broker**
- **Information Service**

**Job Manager**
- **local scheduler**

**Site**
NAREGI Middleware beta

The second generation of the grid middleware developed by NAREGI

- alpha: developed in 2004
  - Based on UNICORE
- beta: developed 2005 -
  - Based on WSRF
  - Conforms OGF standards

Outstanding features

- Workflow management
- Parallel job execution over multiple sites
  - Automatic job partitioning and resource allocation
NAREGI Middleware beta overview

- **SS (Super Scheduler)**
  - Broker
  - Workflow engine

- **IS (Information Server)**
  - Information aggregation
  - DB wrapped by OGSA-DAI

- **GridVM**
  - Cluster management
  - Based on GT4
  - Note: not the ‘real virtual machine’
Overview of NAREGI Information Service

- **CIM scheme based**
  - Stores in a DB

- **Information Collection**
  - LRPS (Local Resource Provider Service)

- **Information Aggregation**
  - Aggregator Service

- **Lookup**
  - OGSA-DAI

  - WSRF based Data base access protocol

Diagram:
- DB
- WSRF Container
- OGSA-DAI Service
- Aggregator Service
- SS
- LRPS for GridVM
- GridVM
- NAREGI-Middleware-β
Overview of EGEE gLite

- Grid middleware stack from EGEE (Enabling Grids for E-Science in Europe)

- Employs Condor modules in several ways
  - Condor
    - Batch queuing system developed by Wisconsin Univ.
  - Brokering based on Condor ‘Match making’
  - Job submission by Condor-C
Overview of gLite

- **Client**
  - **WMS**
    - Workload Management System
    - Brokering based on classad
  - **BDII (Berkley Directory Information Index)**
    - LDAP based information repository

- **gLite-CE**
  - local scheduler

- **LCG-CE**
  - local scheduler

- **CE (Compute Element)**
  - gLite-CE
    - Complicated module that use Condor-C
  - LCG-CE
    - Globus GRAM2
    - Carried over from LCG (LHC Computing Grid) project
Outline

- Architecture of the Grid middleware stacks
  - NAREGI Middleware β
  - gLite

- Strategy for interoperation and implementation

- Measurement Results

- Conclusion
Requirements for mutual job submission

- **Authentication and Authorization Interoperation**
  - Security Infrastructure
  - All the other components rely on it
    - Crucial for interoperation

- **Information Service Interoperation**
  - Look up the resources on the other middleware stack

- **Job Submission Interoperation**
Authentication, Authorization Interoperation

- **Authentication**
  - ‘Who are you’
  - PKI based authentication is generally used

- **Authentication**
  - ‘What can you do’
  - Virtual Organization Management

Fortunately, we did not have any issues on this.

- Authentication – GSI is used
- Virtual Organization Management - VOMS
Interoperability for Information Service

Diagram showing relationships between different components, including DB, WSRF Container, OGSA-DAI Service, Aggregator Service, LRPS for GridVM, GridVM, NAREGI-Middleware, BDII, LRPS for BDDI, translator, CE, and gLite.
3 ways for mutual job submission

- **Broker -> JobManager**
  - (relatively) faster
  - The callee grid policies might be ignored
  - Information service interoperability is mandatory

- **Broker -> Broker**
  - (relatively) slower
  - Easy to enforce callee grid policies

- **JobManager -> Broker**
  - Slowest
  - Easy to enforce callee grid policies
Design of mutual job submission

- Where to have bridges?
- Points that have standard interface are preferable

Points that have standard interface are preferable

Client

SS
GridVM SC

GridVM
local scheduler

WMS
qLite-CE

LCG-CE
local scheduler

Library is available
Interface is defined and published
Proprietary interface

NAREGI-Middleware-β

gLite
NAREGI→gLite

Diagram showing the integration between NAREGI and gLite. The diagram illustrates the flow of communication and services involved in this integration, including Client, SS, GridVM, GridVM SC, GRAM2 SC, WMS, LCG-CE, BDII, gLite-CE, and local scheduler.
Developed a SC that calls LCG-CE(GRAM2) instead of GridVM

- SCs are designed as dynamically loadable independent modules
- Problem: GRAM2 does not provide reservation capability
  - Solution: SC just pretend to make reservation

Automatic selection of SC, based on information provided by the information service
- Hidden from users
gLite → NAREGI
Implementation details

Client → WMS
1

WMS
2

WMS
3

WMS
4

grid ftp

local scheduler

script

Job

1. WMS
2. Condor-C
3. gLite-CE
4. grid ftp
Implementation details

Client

WMS

GRAM client
Condor schedd

SS

Condor-C

grid ftp

GridVM

local scheduler

script

Job

Implementation details
BLAHP Protocol

- Text-based protocol for intermediate processes
  - Based on GAHP, with command set
  - GAHP (Globus Ascii Helper Protocol) - initially designed to call Globus modules from Condor

- Based on UNICORE GAHP (Nakada '04) Command set
  - BLAH_JOB_SUBMIT
  - BLAH_JOB_STATUS
  - BLAH_JOB_CANCEL

We could 'reuse' UNICORE GAHPD codes
Problems solved (1)

File staging to NAREGI failed because gLite-CE uses virtual users on the node

Create a readable temporary directory for each job and copy the files there

Diagram:
- Client
- WMS
- GRAM client
- Condor schedd
- Condor-C
- GridManager
- BLAHPD
- gLite-CE
Problems solved (2)

- Limitation for proxy certificates delegation times
  - Proxy certs. - uses intermediated CA mechanism internally
  - Theoretically, there is no limitation for delegation times
  - Gridftp implementation by Globus has a bug
    - openssh library used in Globus had the default limitation number of intermediate CAs
    - Can be easily fixed

Solution

- Patched the gridftp
Outline

- Architecture of the Grid middleware stacks
  - NAREGI Middleware β
  - gLite

- Strategies for interoperation and implementation

Measurement Results

Conclusion
Experiments

- Measured elapsed time for mutual job submission.
  - Also measured job submission with in each middleware stacks
  - Average time of 10 measurements

Environment

- All the nodes are located in a NAREGI campus
Experimental results

Setups
- Pentium 4 Xeon 3GHz dual, Mem. 1Gbyte, RedHat 8
- Network 1000base-T
Outline

- Architecture of the Grid middleware stacks
  - NAREGI Middleware β
  - gLite

- Strategy for interoperation and implementation

- Measurement Results

Conclusion
Conclusion

- Performed job submission interoperation experiments between NAREGI Middleware beta and EGEE gLite
  - No issues on certs. and VO management
  - Differences in information service layer could be managed
  - Mutual job submission could be successfully performed with proper bridging modules
Future Work

- Precise measurement and analysis
- Experiments on Production systems
  - Confirm interoperability using VOMS in production
  - Investigate effects of latency between Japan and Europe
- More sophisticated mutual job submission
  - Having NxN bridges are not good idea
  - To have standardized Job submission interface will be the best
Thank you

Acknowledgement:
A part of this research was supported by a grant from the Ministry of Education, Sports, Culture, Science, and Technology (MEXT) of Japan through the NAREGI (National Research Grid Initiative) Project.