NAREGI Middleware Objectives

• **Mid-range Project Target**
  – R&D on scalable middleware infrastructure for server grids and metacomputing resource management, deployable at large computing centers, based on Unicore, Globus, Condor

• **Final Project Target**
  – R&D Build resource management framework and middleware for VO hosting by the centers based on the OGSA standard
  – Distribute result as high-quality open source software
Server Grids and Metacomputing

Various (existing) applications and their workflow

Automated Allocation of resources and Scheduling of jobs

Metascheduler /Broker & Workflow Engine

Distributed Servers

National Research Grid Initiative
The Future: VO Hosting by Grid Centers

“Hosting” of Various VOs for Research Areas, Research Groups, etc. by a federation of centers

Dynamic provisioning of large resources to VOs

NanoGrid VO

Project X VO

Lab X Univ A
Lab Y Univ A
Grid Center Univ A
Various dedicated project machines

Grid Center
Research Lab B

Company D Division U

Lab V Univ C

Grid Center
Univ C

National Research Grid Initiative
Features of Grid Middleware

• **(Scalable) MetaScheduler**
  – Schedule large metacomputing jobs
  – “Scalable”, Agreement-based scheduling
  – Assume preemptive metascheduled jobs
• **(Scalable) Grid Information Service**
  – Support multiple monitoring systems
  – User and job auditing, accounting
  – CIM-based node information schema
• **GridVM (Lightweight Grid Virtual Machine)**
  – Metacomputing Support
  – Enforcing Authorization Policies, Sandbox
  – Checkpointing/FT/Preemption
• **Authentication Service**
  – Along with GGF defined assurance level
  – Authentication mechanism across policy domains

And more…
a NAREGI Grid Middleware

SuperScheduler

@SC2004
What is SuperScheduler?

Definition in GGF:

**Super-Scheduler**

The process that will
(1) discover available resources for a job
(2) select the appropriate system(s)
(3) submit the job.

**Meta-Scheduler**

A scheduler that allows to request resources of more than one machine for a single job.

(from GFD-I.11 “Grid Scheduling Dictionary of Terms and Keywords”)

NAREGI Super Scheduler is a **Super-Meta-Scheduler**
What is Super-Scheduler?

Phase 1  Resource Discovery
1. Authorization Filtering
2. Application requirement definitions
3. Minimal requirement filtering

Phase 2  System Selection
4. Gathering information (query)
5. Select the system(s) to run on

Phase 3  Run Job
6. (optional) Make an advance reservation
7. Submit job to resource
8. Preparation Tasks
9. Monitor progress (maybe go back to 4)
10. Find out Job is done
11. Completion tasks

(from GFD-I.4 “Ten Actions When SuperScheduling”)

Traditional Job Scheduler

National Research Grid Initiative
A Target Application and Brokering

RISM-FMO: A coupled simulation of molecular dynamics

Requirements:
- RISM requires one big SMP machine.
- Mediator requires multiple machines.
- FMO requires big cluster system.
- FMO consists of two processes at a node.
A Target Application and Brokering

RISM-FMO: A coupled simulation of molecular dynamics

 Requirements:
• RISM requires one big SMP machine.
• Mediator requires multiple machines.
• FMO requires big cluster system.
• FMO consists of two processes at a node.

Complex brokering result
Super Scheduler Architecture

Super-Scheduler in GGF OGSA-EMS

(from GWD-I “The Open Grid Services Architecture, Version 1.0”)

National Research Grid Initiative
Demonstration

Requirements
- CPU type
- # processors
- # hosts
  and more...

Which one?

Super Scheduler

Automatic Brokering

National Research Grid Initiative

SPARC 128CPU

Xeon 2CPU 128nodes Gbit Ether

Xeon 2CPU 128nodes InfiniBand

Power4 16CPU 32nodes SP switch

Power4 16CPU

Itanium2 32CPU
Demo: Can you find suitable resources for the job?

Resources:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Available</th>
<th>Suitable</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPTU</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>7TJUF</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>QOH</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

This table shows the availability and suitability of different resources for the job. The green check mark indicates availability, and the red check mark indicates suitability.
GridVM Overview

- Virtual Machine Layer Tailored for the Grid
- Provides unified and secure execution environment for Grid applications
- Copes with underlying OS/HW heterogeneity

Upper-level Grid Middleware

Grid Service

Metacomputing
Co-scheduling, Gang Scheduling and other inter-node synch.
Coallocation reservation support

Security
Fine-Grain Access Control
Authorization Enforcement
Sandboxed Execution

Fault Tolerance
Checkpointing,
Fault Detection/
Prevention/Injection

Virtualizing underlying OS/HW

Cluster Software
Local Scheduler

Node OS, e.g., Linux
CPU
NW
Mem
HDD

National Research Grid Initiative
Connecting NJS and GridVM

- Replace TSI with GridVM
  (For the time being, TSI will be used for file staging ...)
- GridVM Java API for local NJS job management

Unicore Client → Job submit → Super Scheduler

WS-Agreement

NJS
- reservation
- submit
- ...

GridVM
(Java instance)

JSD L

Local Scheduler

GridVM
(deamon)

Job Process

Agreement

NJS
- reservation
- submit
- ...

GridVM
(Java instance)

JSD L

Local Scheduler

GridVM
(deamon)

Job Process
GridVM Resource Reservation

• **Making reservations ("makeReservation" method)**
  - Makes a reservation with "JSDL" as input
  - NJS assigns "SubjobID" in JSDL used for subsequent operations
  - Properties that are reserved are:
    • Start time, End time, Number of nodes, Node names (if necessary)

• **Cancel Reservation ("cancelReservation" method)**
  - Cancels a specified reservation

• **Query Reservation ("queryReservation" method)**
  - Queries information about specified reservation
  - Returns a XML including the following information
    • Start time, End time, Number of nodes, Node names (if necessary)
### Sample NAREGI JSDL Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
<th>Note about JSDL</th>
</tr>
</thead>
<tbody>
<tr>
<td>/naregi-jsdl:SubJobID</td>
<td>Job ID assigned by NJS</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>/naregi-jsdl:CPUCount</td>
<td>Number of needed CPUs</td>
<td>8</td>
<td>Deleted from 0.4.3</td>
</tr>
<tr>
<td>/naregi-jsdl:TasksPerHost</td>
<td>Number of Tasks per host</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>/naregi-jsdl:TotalTasks</td>
<td>Number of MPI tasks</td>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td>/naregi-jsdl:NumberOfNodes</td>
<td>Number of Nodes used</td>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td>/naregi-jsdl:CheckpointablePeriod</td>
<td>Interval for checkpoint</td>
<td>100</td>
<td>N/A</td>
</tr>
<tr>
<td>/jsdl:PhysicalMemory</td>
<td>Amount of needed Physical Memory</td>
<td>1GB</td>
<td>JSDL 0.4.3</td>
</tr>
<tr>
<td>/jsdl:ProcessVirtualMemoryLimit</td>
<td>Amount of needed Virtual Memory</td>
<td>100GB</td>
<td>JSDL 0.4.3</td>
</tr>
<tr>
<td>/naregi-jsdl:JobStartTrigger</td>
<td>Reserved job start time</td>
<td></td>
<td>Deleted from 0.4.3</td>
</tr>
<tr>
<td>/jsdl:WallTimeLimit</td>
<td>Max amount of wall time required</td>
<td>100h</td>
<td>JSDL 0.4.3</td>
</tr>
<tr>
<td>/jsdl:CPUTimeLimit</td>
<td>Max amount of cpu time required</td>
<td>200h</td>
<td>JSDL 0.4.3</td>
</tr>
<tr>
<td>/jsdl:FileSizeLimit</td>
<td>Max file size created</td>
<td>10GB</td>
<td>JSDL 0.4.3</td>
</tr>
<tr>
<td>/jsdl:Queue</td>
<td>Name of queue to which job is submitted</td>
<td>grid_01</td>
<td>JSDL 0.4.3</td>
</tr>
</tbody>
</table>
GridVM Demo@SC2004
@Pittsburgh, USA

Job Submit

Cluster A
OGSI
GridVM
SCore

Cluster B
OGSI
Synchronized Start
GridVM
SCore

Cluster C
OGSI
SCore

Cluster D
OGSI
SCore

Job running over multiple clusters

@NII, Japan

National Research Grid Initiative
Distributed Information Service

NAREGI WP1
& Hitachi, Ltd.
Overview

NaReGI Middleware

Viewer

Info. Service for VO

Virtual Org.

Resource
Log

Performance
User
Account

Budget
Policy

Generate, Control

R.W. Management

SQL Query, Data Service

Local Mgmt. system

GT3 Info. Service

RDB

Site
Center
Inst.

Search, Set, Notify

Real World

Resource
User
Log

Performance
Policy

Budget
Account

Information Provider

NaReGI m/w, Scheduler, etc.

National Research Grid Initiative
Support for Resource Brokering and Accounting ...

- GridVMs provide information about Job Queue and Job Usage.
- Resource Brokers consume the information using SQL query.

Ⅰ General schema for resource description based on CIM Schema.
  Ⅱ can satisfy requirement of other middleware.
  Ⅲ can include existing / standard schema.
      (Ⅳ GGF / JSIM, UR Schema )

Ⅰ Aggregated CIM objects are accumulated to RDB.
  Ⅰ Resource discovery by using SQL query,
      Analysis of time-series data.

Ⅰ Implemented as Grid Service (on GT3.2 with OGSA-DAI).

Ⅰ Stable, Scalable and Secure access to distributed large DB.
Cell Domain

Node A

Resource
Accounting
Log
...

Node B

Node C

Viewer
Super Scheduler

CIMOM (Pegasus)
CIM Providers

PostgreSQL

CIMOM (Pegasus)
CIM Providers

GT3/ Index Service (aggregates information to Node C)

NAREGI schema

CI MOM (Pegasus)
CI Providers

CI MOM (Pegasus)
CI Providers

CI MOM (Pegasus)
CI Providers

Decode Service
(1) Resource information
User can get information about Computer Systems to which the user has access right.

Proxy cert. → ViewService ↔ CIMQueryService ↔ PostgreSQL

Get user’s DN from proxy and makes SQL to find information about his accessible resources.

OS, Processor, FileSystem, Account, + Job Queue, Job Usage(2004), Association: user – account – system … Each class and attribute is defined in schema based on CIM.

(2) Accounting information
User can get information of Resource Usage on computer systems to which the user has access right.

makes SQL to find accumulated resource usage and budget of his account on computer systems.

(3) Log analysis
System administrators can find Logs with his specifying conditions;
{Category, Severity, Source, Time stamp, Message}.
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUStatus</td>
<td>1</td>
</tr>
<tr>
<td>Caption</td>
<td>Processor 1</td>
</tr>
<tr>
<td>CurrentClockSpeed</td>
<td>2794</td>
</tr>
<tr>
<td>Description</td>
<td>This is an instance of a CIM_Processor on...</td>
</tr>
<tr>
<td>Family</td>
<td>2</td>
</tr>
<tr>
<td>LoadPercentage</td>
<td>59</td>
</tr>
<tr>
<td>MaxClockSpeed</td>
<td>2794</td>
</tr>
<tr>
<td>Status</td>
<td>OK</td>
</tr>
<tr>
<td>Stepping</td>
<td>9</td>
</tr>
<tr>
<td>Distinguished Name (GlobalID)</td>
<td>CPU Time (Budget)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>/O=ST1G/O=STC/OU=srg.hitachi-to.co.jp/CN=glo</td>
<td>1010.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>HostName</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>png2042.naregi.org</td>
</tr>
<tr>
<td>png2044.naregi.org</td>
</tr>
<tr>
<td>png2044.naregi.org</td>
</tr>
</tbody>
</table>
Autonomous Configuration of
Grid Monitoring System (Background)

• In general, autonomous management is necessary on the Grid
  – (Automated) distributed component setup, dynamic reconfiguration and recovery
  – For long deployment of the Grid with a little engineers’ and operators’ costs
• Especially monitoring system is necessary for effective management and resource provisioning, and distributed all around the Grid

Need Grid Monitoring System with autonomous management

National Research Grid Initiative
A prototype of self-configuration framework for Network Weather Service (NWS) [`99 Wolski et al.]
- There is a meta-component for each component which controls the component and exchanges messages between meta-components
- Meta-components act based on some rules and distributed algorithm
- We will support many management type such as complete, partial and embedded (In this time, We demonstrate complete type)
Autonomous Configuration of Grid Monitoring System –Future Issues–

- Support for many grid monitoring systems in general
- Autonomic deployment of the Grid not only monitoring system with little intervention of administrators or users
- Adaptation to large environment

Application example for the Grid environment

- Configure NWS on the Campus Grid Testbed (Titech Grid) in Tokyo Institute of Technology
- Show assignment the NWS component to the nodes by complete management feature