Design and Implementation of Condor-UNICORE Bridge

Hidemoto Nakada ¹, ⁴, Jaime Frey ², Motohiro Yamada ³, Yasuyoshi Itou ³, Yasumasa Nakano ³, Satoshi Matsuoka ¹, ⁵

¹.National Institute of Advanced Industrial Science and Technology, Japan
².University of Wisconsin
³.Fujitsu Limited
⁴.Tokyo Institute of Technology
⁵.National Institute of Informatics
Condor Overview (1)

- Job scheduling system developed by Univ. Wisconsin, from '80
- The initial goal was to utilize computational resources in campus
Condor Overview (2)

Now, it is a meta-scheduler for the Grid

Can utilize Globus-managed clusters as a computational resource pool
The problem

- We already have several Grid Middlewares except for Globus
  - UNICORE, NORDGRID, ...
- The number is increasing

- Generic way to handle them is required
The Goal

- To give a generic framework that enables
  - Easy implementation of the bridge to Grid Middlewares
  - Without modifying the Condor main modules

- Validate the framework with implementing a bridge to UNICORE
Outline of the talk

- Condor architecture
  - Existing bridging mechanism

- Proposed bridging mechanism

- Implementation of a bridge to UNICORE

Summary
Condor Architecture

Central Manager

Periodic Status Report

Schedd

User

Submit Machine

Startd

Process

Execute Machine

Periodic Status Report

Match Making
Condor ClassAd: source of flexibility

- Data format used in whole condor system
- Attribute Key and Value pair
  - Expressions are supported
- All the information and requirements on jobs and machines are represented in classAd form
- Central manager ‘matchmakes’ them, assigning job to machine

```
MyType = "Job"
TargetType = "Machine"
ClusterId = 292
User = "nakada@a02.aist.go.jp"
In = "/dev/null"
Out = "A.out"
Err = "/dev/null"
```
Existing bridging mechanism: Condor-G

Condor Submit Machine

- User
- Grid Manager
- Globus GAHP
- Globus GAHP Server

Schedd

GLOBUS WORLD

GRAM Protocol

- GateKeeper
- JobManager
- Batch System
- Job

GLOBUS WORLD

12/2/2005

HPC ASIA 2005 @ Beijing
Existing bridging mechanism: Condor-G

- Talks Condor Protocol
- Handles Globus specific logic

System Neutral protocol

- Talks Globus Protocol
Condor-G concept

Clear separation between Condor and Globus specific codes

- It is really hard to Link them together.
- Connect them with System neutral protocol

Great?, but in reality,

- The Globus specific logic is handled within the GridManager, not in the GAHP Server
- GAHP protocol itself was system neutral, but the command set used on the protocol was very Globus Specific
GAHP command set for Globus

- INITIALIZE_FROM_FILE
- INITIALIZE_FROM_MYPROXY
- VERSION
- ASYNC_MODE_ON
- ASYNC_MODE_OFF
- QUIT
- RESULTS
- GRAM_CALLBACK_ALLOW
- GRAM_ERROR_STRING
- GRAM_JOB_REQUEST
- GRAM_JOB_CANCEL
- GRAM_JOB_STATUS
- GRAM_JOB_SIGNAL
- GRAM_PING
- GRAM_JOB_CALLBACK_REGISTER
- GASS_SERVER_INIT
- REFRESH_PROXY_FROM_FILE
- MYPROXY_REFRESH
- MYPROXY_RETRIEVE
- PROXY_INFO
- MYPROXY_DESTROY
- MYPROXY_DELEGATE
Proposed architecture

Condor Submit Machine

Schedd

User •Talks Condor Protocol
Grid Manager •Handles XXX Specific logic

XXX GAHP •System Neutral protocol

XXX GAHP Server •System Neutral command set

XXX Server •Talks XXX Protocol
•Handles XXX logic
Proposed architecture

To support a new Grid Middleware system

- Only the GAHP server is required to be re-implemented
  - Java classes are provided to talk GAHP protocol
- GridManager can be remain untouched
  - It is not easy to modify it for third party
What we did

- Redesign the GAHP command set
  - Simple and Generic as much as possible
  - Note: The GAHP protocol is not changed

- Implement some logic in the Gridmanager to support Generic commands
  - Can be reused for other systems
Design principle of the Generic GAHP command set

- **Simple and Generic**
  - High level commands
    - Just 5 commands - c.f. 23 commands for Globus

- **Use ClassAd as a command argument and a return value**
  - To ensure the genericity of the command set
  - System specific things are encapsulated in the ClassAd
  - You can extend the functionality by just extending the ClassAd attribute, without touching the Command set itself
Generic GAHP command set

- **Job Create**
  - Create a Job
  - Input: `ClassAd`
  - Output: Job Handle

- **Job Start**
  - Invoke the Job
  - Input: Job Handle

- **Job Status**
  - Query the Job status
  - Input: Job Handle
  - Output: Status `ClassAd`

- **Job Destroy**
  - Destroy information stored in the GAHP server
  - Input: Job handle

- **Job Recover**
  - Reconnect already running job. Used when GridManager recovers from crash
  - Input: Job handle, `ClassAd`
Condor-U

- The bridge to invoke Condor job via UNICORE

- We implemented it to validate the architecture

Implementation steps

- Design ClassAd for UNICORE
- Implement the GAHP Server which understand UNICORE specific ClassAd and talks with UNICORE
The UNICORE

- A Grid middleware developed mainly by Fujitsu Lab. Europe
  - Owned by UNICORE Forum
  - Designed to utilize several supercomputers installed distributed supercomputer centers
  - SSH based security model
    - No PROXY CERT.
  - Firewall Aware (c.f. Globus)
    - Connection is one-way:
      - Can be used from private addressed network
      - Submit and Disconnect!
  - Totally in Java (except for small perl scripts)
The UNICORE architecture

**Gateway**
- Application level Router
- Runs on a Firewall
- Relay all communications
- SSL based security

**NJS (Network Job Supervisor)**
- Workflow engine
- Interpret AJO and execute

**TSI (Target System Interface)**
- Wrap batch sub system
- Implemented in Perl

Diagram:
- Client
  - Gateway
    - NJS
      - TSI
        - Batch Subsystem
          - Vsite
          - Usite
    - Firewall
### ClassAd Attributes (1) generic

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cmd</strong></td>
<td>pathname of the command to execute</td>
<td><code>/home/foo/a.exe</code></td>
</tr>
<tr>
<td><strong>Args</strong></td>
<td>arguments for the executable</td>
<td><code>arg1, -a</code></td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>environment variables</td>
<td><code>LANG=en_US</code></td>
</tr>
<tr>
<td></td>
<td>current directory</td>
<td><code>home/nakada/condor</code></td>
</tr>
<tr>
<td><strong>In</strong></td>
<td>standard input</td>
<td><code>input.dat</code></td>
</tr>
<tr>
<td><strong>Out</strong></td>
<td>standard output</td>
<td><code>output.dat</code></td>
</tr>
<tr>
<td><strong>Err</strong></td>
<td>error</td>
<td><code>Error.dat</code></td>
</tr>
<tr>
<td><strong>TransferInput</strong></td>
<td>Stage in files</td>
<td><code>a.exe, input.dat</code></td>
</tr>
<tr>
<td><strong>TransferOutput</strong></td>
<td>Stage out files</td>
<td><code>output.dat</code></td>
</tr>
<tr>
<td><strong>JobStatus</strong></td>
<td>Condor job status</td>
<td><code>jobClassAdAttributes</code></td>
</tr>
<tr>
<td><strong>ErrMessage</strong></td>
<td>Error messages</td>
<td><code>Script reported no errors</code></td>
</tr>
<tr>
<td><strong>RemoteWallClockTime</strong></td>
<td>Execution wall clock time</td>
<td><code>123.0</code></td>
</tr>
<tr>
<td><strong>ByteSent</strong></td>
<td>Bytes sent</td>
<td><code>1023004</code></td>
</tr>
<tr>
<td><strong>ByteRecvd</strong></td>
<td>Bytes received</td>
<td><code>1023004</code></td>
</tr>
<tr>
<td><strong>ExitBySignal</strong></td>
<td>The job process quitted by signal?</td>
<td><code>TRUE</code></td>
</tr>
<tr>
<td><strong>ExitCode</strong></td>
<td>Exit code of the process</td>
<td><code>1</code></td>
</tr>
<tr>
<td><strong>ExitSignal</strong></td>
<td>Exit signal of the process</td>
<td><code>9</code></td>
</tr>
</tbody>
</table>
ClassAd attributes (2) Unicore Dependent

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridResource</td>
<td>- gaoh server name</td>
<td>Unicore fujitsu.com:1234 NaReGI</td>
</tr>
<tr>
<td></td>
<td>- FQDN and port number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for the Unicore Usite gateway</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Unicore Vsie Name</td>
<td></td>
</tr>
<tr>
<td>KeystoreFile</td>
<td>Keystore file name</td>
<td>/home/foo/key</td>
</tr>
<tr>
<td>KeyStorePassphraseFile</td>
<td>Pass phrase file name</td>
<td>/home/foo/passwd</td>
</tr>
<tr>
<td>UnicoreJobId</td>
<td>Job ID used as the handle</td>
<td>fujitsu.com:1234/NaReGI/1374036929</td>
</tr>
<tr>
<td>UnicoreJobStatus</td>
<td>UNICORE job status</td>
<td></td>
</tr>
<tr>
<td>UnicoreLog</td>
<td>UNICORE log filename</td>
<td>/var/log/unicore.log</td>
</tr>
</tbody>
</table>
Submit file sample for Condor-U

```
Universe      = grid
GridResource  = unicore fujitsu.com:1234 ghost
Executable    = a.out
output        = tmpOut
error         = tmpErr
log           = tmp.log

KeystoreFile  = /home/foo/key
KeystorePassphraseFile =
                     /home/foo/passwd
```

Declaration to use unicore
V-site

Queue

U-site

Key management

12/2/2005
Implication of the Condor-U

- Condor users can use resources managed by the UNICORE
  - From outside of the sites
  - Users can use Condor as a job-scheduler for UNICORE managed resources

- Condor GlideIn might be used on it
  - Note: Communication between nodes have to be assured - it is not common for UNICORE setup
Current Status

Merged into the main trunk of the Condor CVS

Will be shipped as a part of the next development release
Summary and Future Work

We defined a generic and simple command set for Condor external interface

- Unicore GAHP server implementation showed feasibility of the command set

Future work

- Implements GAHP servers for other Grid systems with the command set
  - UNIGRID ?
  - NAREGI Middleware ?
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.