WS - PGRADE/gUSE Tutorial

MTA SZTAKI
Laboratory of Parallel and Distributed Systems (LPDS)

M. Kozlovszky
Senior Research fellow
m.kozlovszky@sztaki.hu

www.lpds.sztaki.hu
Contents

• History, family of products
  • P-GRADE Portal, WS-PGRADE, gUSE
• WS-PGRADE in a nutshell
• Architecture, high level structures, Workflow handling
• WS-PGRADE features
  • Parameter Study (PS) support
  • Repository
  • Internal Storage (home dir)
• Automated workflow submission & Timing
• Web services
• Embedded Workflows
• DB support
• Conditional job submission
• Job’s history, logging and statistics
• Howto use WS-PGRADE /Basic + Advanced tasks/
Family of P-GRADE Portal products

- P-GRADE portal
  - Creating (basic) workflows and parameter sweeps for clusters, service grids, desktop grids
  - [www.portal.p-grade.hu](http://www.portal.p-grade.hu)

- P-GRADE/GEMLCA portal (University of Westminster)
  - To wrap legacy applications into Grid Services
  - To add legacy code services to P-GRADE Portal workflows
  - [http://www.cpc.wmin.ac.uk/cpcsite/gemlca](http://www.cpc.wmin.ac.uk/cpcsite/gemlca)

- WS-PGRADE/gUSE (Web Services Parallel Grid Runtime and Developer Environment/Grid User Support Environment) architecture
  - Creating complex workflow and parameter sweeps for clusters, service grids, desktop grids, databases
  - Creating complex applications using embedded workflows, legacy codes and community components from workflow repository
  - [www.wspgrade.hu](http://www.wspgrade.hu)
P-GRADE Portal software family

- **P-GRADE Portal 2.4**
- **P-GRADE Portal 2.5**
- **P-GRADE Portal 2.8**
- **P-GRADE Portal 2.9.1**
- **P-GRADE Portal 2.10**
- **GEMLCA Grid Legacy Code Arch.**
- **NGS P-GRADE portal**

**Key Dates:**
- 2008
- 2009
- 2010
- 2011

- **Open source**
- **Basic concepts**
- **GEMLCA, storage concepts**

- **WS-PGRADE Portal 3.1**
- **WS-PGRADE Portal 3.2**
- **WS-PGRADE Portal 3.3**
References and WS-PGRADE/gUSE installations

- **WS-PGRADE Portal service is available for**
  - GILDA - Training VO of EGEE and other projects
  - SEE-GRID - South-Eastern European Grid
  - VOCE - Virtual Organization Central Europe of EGEE
  - HunGrid - Hungarian Grid VO of EGEE
  - NGS - UK National Grid Service
  - Desktop Grids: SZTAKI DG

- **Users and projects using WS-PGRADE/gUSE**
  - EDGeS, EDGI projects: Integrating EGEE with BOINC and XtremWeb technologies
  - ProSim project: In silico simulation of intermolecular recognition, JISC ENGAGE program
  - University of Westminster Desktop Grid: Using AutoDock on institutional PCs
  - CancerGrid project: Predicting various properties of molecules to find anti-cancer leads, eScience gateway for chemists
  - SHIWA (SHIWA Simulation platform v2.0)
WS-PGRADE/gUSE in a nutshell

• **General purpose, workflow-oriented portal.** Supports the development and execution of workflow-based applications

• **Based on Liferay**

• **Services** supported by the portal:
  
  • Web services
  • DB connectors
  • Embedded workflows
  • Job level Parameter Study
  • Conditional jobs
  • Recursive graph
  • Multi-generator
  • Multi-collector
  • CROSS product Parameter Study support
  • DOT product Parameter Study support

<table>
<thead>
<tr>
<th>DCIs</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>gLite</td>
<td>✓</td>
</tr>
<tr>
<td>ARC</td>
<td>✓</td>
</tr>
<tr>
<td>Unicore</td>
<td>✓</td>
</tr>
<tr>
<td>LCG2</td>
<td>✓</td>
</tr>
<tr>
<td>GT2, GT4</td>
<td>✓</td>
</tr>
<tr>
<td>Desktop Grids</td>
<td>✓</td>
</tr>
<tr>
<td>Clusters</td>
<td>✓</td>
</tr>
<tr>
<td>Clouds</td>
<td>✓</td>
</tr>
</tbody>
</table>

Simultaneous use of various clusters, service and desktop Grids at workflow level
WS-PGRADE architecture

Graphical User Interface: WS-PGRADE

Workflow storage

Workflow Engine

Meta-broker

gUSE

Workflow information system

Submitters

File storage

Application repository

Logging

Local resources, service grid VOs, Desktop Grid resources, Web services, Databases

Liferay/Gridsphere portlets

Autonomous Services: high level middleware service layer

Resources: middleware service layer
Important high-level graph structures in WS-PGRADE

Graph
Jobs, Edges, Ports

Concrete Workflow
 Algorithms, executable
 Resource references, Inputs

Workflow Instance
Running state, Outputs

Template
Constraints, Comments, Form Generators

Repository Item
Application OR Project OR, Workflow part (G,T,CW)

Legend:
a → b   a must reference b
a ←→ b   a may reference b
1. Create and edit the Graph structure
2. Add name to the graph and configure it (job, port, resources, etc.)
3. Submit the Concrete Workflow, observe its status and fetch its result
4. Restrict some parameters/features of the workflow and create a template
5. Add name to the template and configure it (job, port, resources, etc.)
6. Export workflow, template, graph, to the repository (reuse it later by you or by end users)
7. Reuse an Application, Template, graph or Workflow from repository
8. Redefine the originally used graph within the workflow
Workflow handling by end user

1. Import an Application/Template

2. Set the available parameters on the end-user View

3. Submit the Concrete Workflow, observe its status and fetch its result
• History, family of products
  • P-GRADE Portal, WS-PGRADE, gUSE
• WS-PGRADE/gUSE in a nutshell
• Architecture, high level structures, Workflow handling
• WS-PGRADE features
  • Parameter Study (PS) support
  • Repository
  • Internal Storage (home dir)
  • Workflow activation & Timing
  • Web services
  • Embedded Workflows
  • Cloud jobs
  • DB support
  • Conditional job submission
  • Job’s history, logging and statistics
• Howto use WS-PGRADE /Basic + Advanced tasks/
Repository usage & services

• Share with others the developed
  • Graphs
  • Workflows - Configured graphs
  • Templates - Restricted workflows
  • Applications - Semantically tested, trusted Workflows containing the definitions of the eventually referenced Embedded Workflows (including their transitive closures)
  • Projects - Applications which are not ready/fully functional yet

• Build up a science/application gateway for end-users
  • Restrict certain workflow parameters
  • Provide easy workflow configuration/parameter setup interface
  • Repository is acting like an interface between developers and end users

Note:
Exported/imported “Projects” are not checked for completeness and correctness “Applications” are checked. Name collisions are checked and handled by the system.
Repository views

- Materials are classified by their type
  - Latest available lists
    - Application
    - Project
    - Concrete
    - Template
    - Graph

- To import
Internal Storage (user workspace)

<table>
<thead>
<tr>
<th>Workflow name</th>
<th>Size (Bytes)</th>
<th>Instances</th>
<th>Identifier</th>
<th>Size</th>
<th>Action</th>
<th>Download actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asb_coll_seegrid_2011-06-06-123009</td>
<td>302900</td>
<td>0</td>
<td>2011-6-6 12:30</td>
<td>191796</td>
<td>get Outputs</td>
<td>get All, get All but Logs, get Inputs, get Outputs</td>
</tr>
</tbody>
</table>

Columns of **individual** instances, please note, that outputs can be downloaded separately.

Columns of **bulk download**: All or proper parts of all instances of a given WF can be downloaded.

Information about the quota of the user allotted storage capacity in the Portal server.
Upload Implementation to the internal Storage

Step 1:
Select the compressed file in the client machine containing the requested Workflow

Step 2(option):
Check the kind of name(s) you want to redefine

Step 3 (If Step 2 performed):
Enter the new name(s) which will not collide

Step 4:
Confirm the operation
Workflow submission

A workflow can be submitted by 3 different way:

1. Interactively started by the user hitting the button **Submit** belonging to the given concrete workflow on the portlet **Workflow/Concrete (default)**

2. Started by a **-crontab** like - predefined time schedule. The corresponding timetable can be set in the portlet **Workflow/Timing**

3. Started by an external event. The corresponding event to be waited for and the name of the Workflow can be defined in the portlet **Workflow/Remoting**
Workflow Submission(1)
Interactively by the user

Step 1: The workflow is selected by button “Submit”

Step 2: The submission can be confirmed or refused after the optional filling of a free description field identifying Workflow Instance for the user
Workflow Submission (2) (on HP-SEE portal yet not enabled)

Start automatically by an internal timer

- Start a workflow at a predefined time
- Valid certificate should be available (if grid is used)
- Crontab like working
- Single run

Checklist to select an existing Workflow to be added
Definition of a new element to be added
List of scheduled Workflows

Tool to define submission time
Confirm button to append a new element to the list
Item can be revoked deleting it from the list

The items will leave the list on schedule time expiration even if the Workflow submission has failed by any cause
Workflow Submission(3) (on HP-SEE portal yet not enabled)
Start by external event

Two parts must be defined:

• **On the Portal Server Side**
  - the name of the Workflow and
  - the identifier “Remote Key”

• **On the Client Side inside of a WSDL description**
  - the URN of service call
  - the name of the service
  - the owner of the workflow (user)
  - the identifier “Remote Key”
  - a free string to be associated to Workflow Instance

Note, that a Service call referencing a common Remote Key used in more than one Portal Server Side description submits all the associated Workflows
Workflow Submission (3)
Start by external event - Portal Side

Name of WF to be appended to the list of remotely callable WF-s
Definition of Keyword to identify the call
Append button
List of listening Workflows
Button to see a stored Keyword
Button to hide the stored Keyword and the button Delete
Any Workflow can be deleted from the listeners. The delete button can accessed only in the not hidden state
Check list can select each of the available workflows
Workflow Submission(3)
Start by external event – Client side: WSDL

The client should provide a Service call conforming the WSDL.
Web service support

Principle:

- **Job is a web service**
  user can reach an existing remote service with the following attributes:

1. **Type:**
   Base standard type of the Web Service. The administrator of the portal server sets the list of standards the portal can understand. The default value is Axis.

2. **Services:**
   Selection list of services of the given *type* having been explored by the Portal Server

3. **Methods:**
   Selection list functions the selected *service* implements
Web service support (contd.)

Step 1. Select Service as Job interpretation class

Step 2. Select type of Service to be understood

Step 3. Select one from the found services

Step 4. Select a Method among the interface routines of Service
Embedded workflow support

Principle:
Job is an embedded workflow

**Embedded Workflow**
To ensure the compatibility of interfaces the embedded workflow must be defined by a Template

**Original Workflow**
The dummy job whose execution will be substituted by the call of the embedded one
Embedded workflow support

Focus on the caller Job

Step 1. Select (embedded) Workflow as Job interpretation class

Step 2. Embedded (called) Workflow is selected by the check list showing the possible templated Workflows

Focus has been set by selecting a job
Cloud supported job

Step 1. Select Cloud as Job interpretation class

Step 3. Select one from the found clouds
DB support (Datasource: SQL Database)

Step 1 (offline)
Create Database on a remote site

Step 2 (online)
Port configuration: set Query

Step 3 (online)
Port configuration: File generation from result set
Conditional job submission

Rule:
To any input port a boolean conditional expression may be attached. They will be evaluated upon the value(s) of the referenced port(s). The false value inhibits the execution of the actual and subsequent jobs. The state of the failed job is indicated as “Term_false” the eventual subsequent jobs will remain in “Init” state.

Expression Syntax:

- `<File value> <Operator> <Constant string>` |
- `<File value> <Operator> <reference of a foreign port>`

where

- `<Operator>`: `==` |
- `!=` |
- `contain`

The comparison of the file value (regarded as a string) associated to the current port and of other operand which may be a string or the string file content of a different port results true or false depending on the string operation, where `==` means equal, `!=` means not equal, and `contain` means, that the left operand contains the right operand.

AND operation is assumed for more than one ports associated with conditional expressions:

```plaintext
B1  AND  B2 = FALSE
```
## Job’s history / Usage Logging

### Example of Job’s History Logging

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-01-12</td>
<td>Create new job</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2021-01-13</td>
<td>Modify the status of the job</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2021-01-14</td>
<td>Start the job</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2021-01-15</td>
<td>Complete the job</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Usage Logging Dashboard

- **Portal Statistics**: Displays the number of users, sessions, and activity over time.
- **User Statistics**: Shows the top users and their activity levels.
- **Overall Portal Statistics**: Highlights the most active users and their engagement levels.
- **Top Users**: Identifies the users with the highest number of sessions.
- **User Activity**: Tracks user engagement and session duration.

![Portal Statistics Dashboard](image1.png)

![User Statistics Dashboard](image2.png)
How to use WS-PGRADE

Basic tasks

• Create Graph (or reuse an existing one)
• Create Concrete Workflow from Graph
• Setup Concrete Workflow (ports, jobs)
• Submit and check the running Workflow
There are two ways to create a Workflow Graph (WfG):

- Opening the Graph Editor (with the proper button) in the portlet Workflow/Graph

- Clone an existing one (Saving the actual Graph with a new name)
Graph Editor detailed - Build up a WF Graph

1. **New Job**
2. **New Port**
3. **Right Click**
4. **Port Property**
5. **Select Output**
6. **Port name can be changed**
7. **Comment can be inserted**
8. **Close by OK**
9. **Press mouse...**
10. **and drag to Input Port**

- **New Port**
- **New Job**

End.
How to use WS-PGRADE

Basic tasks

- Create Graph (or reuse an existing one)
- Create Concrete Workflow from Graph
- Setup Concrete Workflow (ports, jobs)
- Submit and check the running Workflow
Development cycle - Creating the Concrete Workflow

- Create Concrete Workflow from Graph, from template, or from another workflow
- give a name to the Concrete Workflow
- give notes
How to use WS-PGRADE

Basic tasks

• Create Graph, or reuse an existing one
• Create Concrete Workflow from Graph
• Setup Concrete Workflow (ports, jobs)
• Submit and check the running Workflow
Setup Concrete Workflow

- Setup all the job properties
  - Execution model:
    - Workflow, Service, Binary
  - Type: gLITE, ARC, GEMLCA, GT-4, GT-2, Local
  - Grid resource
  - Type of binary
    - SEQ, MPI, Java
  - Number of MPI Nodes
  - Executable
  - Additional parameters

- Setup port properties
Workflow configuration Hierarchy

Concrete WF List

Selected WF

Selected Job

Job Executable

Job Inputs & Outputs

JDL/RSL

Job Config. History

Job is Workflow

Job is Service

Job is Binary
How to use WS-PGRADE
Basic tasks

• Create Graph, or reuse an existing one
• Create Concrete Workflow from Graph
• Setup Concrete Workflow (ports, jobs)
• Submit and check the running Workflow
In case of PS Workflows the list “Job Instance” may contain more than one elements with different “PID” -s
To become a user of HP-SEE’s P-GRADE/gUSE Portals

• Become user of SEE-GRID if you are still not...
• Request an account: http://ls-hpsee.nik.uni-obuda.hu:8080/web/guest/welcome
• Request an account on the SZTAKI’s multigrid portal: https://guse.sztaki.hu/
Thank you

Miklos Kozlovszky
m.kozlovszky@sztaki.hu

www.wspgrade.hu
www.guse.hu
www.lpds.sztaki.hu
Infrastructure

Client machines with browsers

ls-hpsee.nik.uni-obuda.hu

HP-SEE Portal
(Obuda University, Budapest)
based on WS-PGRADE/gUSE

MyProxy name:
ARC:summerathens

MyProxy password:
ARC:…

Proxy

Tutors generated and uploaded your proxy

ARC based HPC services
(NIIF’s infrastructure)

gLite based HPC services
(ACAD’s infrastructure)