SLA4D-GRID

Documentation of the SLA4D-Grid Negotiation Manager (Globus)

Release date : 02/07/2010

Author: Martin Raack, TU Berlin

Version: 1.0

Status: Final

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Inhaltsverzeichnis

1 Introduction

2 Installation

The Negotiation Manager requires a Globus Toolkit installation. The version currently used in development is 4.0.8. Further, we require an installation of the Torque/Maui Scheduler.

In the following **$INSTALL** will refer to the directory where the NegMgr installation packet has been extracted. **$GLOBUS_LOCATION** will refer to the location of the local GT installation.

2.1 ZIB ARS Module

The Negotiation Manager uses the ARS module (Advance Reservation Service) developed by Zuse Institut Berlin (ZIB) to make reservations using Globus. Since GTs WS-GRAM does not natively support reservations, two files need to be modified:

- $GLOBUS_LOCATION/share/globus_gram_job_manager/pbs.rvf

  need to be extended by:

  ```
  Attribute: reservation_id
  Description: "Target the job to an advance reservation as defined
          by the scheduler at the defined (remote) resource."
  ValidWhen: GLOBUS_GRAM_JOB_SUBMIT
  ```

- $GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager/pbs.pm

  the function submit needs to be extended by:

  ```
  if (defined $description->reservation_id())
  {
  print JOB '#PBS -W x=FLAGS:ADVRES:', $description->reservation_id(), "\n";
  }
  ```

Now the ZIB ARS Modul can be installed:

- > cd $INSTALL/ZIBARS/
- >./install.sh

For user authentication a grid-mapfile needs to be created (i.e. linking to /etc/grid-security/grid-mapfile):

- $GLOBUS_LOCATION/etc/maui_crm/grid-mapfile

In general, these steps should suffice to prepare GT for the installation of the ARS module. In case you encounter any problems, more detailed instructions can be found in the ARS install guide [1].

2.2 NegMgr

The installation is started by calling the installation script:

```
> cd $INSTALL/NegMgr
> ./install.sh
```

The NegMgr authenticates users using a grid-mapfile, which needs to be created in:

- $GLOBUS_LOCATION/etc/negmgr-server-1.0-SNAPSHOT/users.gridmap

We suggest that you create a symbolic link to /etc/grid-security/grid-mapfile.

Now the hostname and -port of the GT instance that should be used for job submission needs to be configured (JOB_SUBMISSION_HOST and JOB_SUBMISSION_PORT) :

- $GLOBUS_LOCATION/etc/negmgr-domain-server-1.0-SNAPSHOT/server-config.wsdd

After that, the installation is complete.

3 Usage

3.1 Application Database

The NegMgr-Domain-Server provides agreement templates, which are generated from an application database located in:

- $GLOBUS_LOCATION/etc/negmgr-domain-server-1.0-SNAPSHOT/sample.idb

This file can be easily extended to provide custom applications. To add a new application, you basically only need to provide a unique application name, version and a path to an executable.

The database provides two sample applications:

- testtouch 1.0 – creates an empty file /tmp/testtouch
- cattest 1.0 – creates a copy /tmp/output.txt of /tmp/input.txt

We will use these applications in the following usage examples.

3.2 Target System Information

The application database file also contains information about the offered local resources, like CPU architecture, CPU count and memory size. This information will be included in generated templates and also automatically validated upon agreement creation.

3.3 Custom service description terms

If you would like to extend templates by custom terms, you will need to alter the file:

$SRC/negmgr/negmgr-domain-server/src/main/resources/samples/compute.xml.vm

which contains the actual validation constraints. Afterwards, you will need to rebuild and re-deploy the NegMgr module.

3.4 Sample agreement lifecycle

The NegMgr provides a sample client to test the complete lifecycle of an agreement:

- > wsag-example https://<host>:8443/wsrf/services/DomainAgreementFactory

The client first requests templates from the server, then chooses the „cattest“ 1.0 template and creates an offer which it sends to the server to create an agreement. The offer also contains the file-staging description.

After successfully creating an agreement, the client will periodically monitor the state of the agreement. Once it reaches the state “complete”, the client will exit.

The corresponding sources can be found in

- $SRC/negmgr/negmgr-client/src/main/java/de/dgrid/sla4dgrid/clients/example/

Note:

The file staging (out) does not work correctly yet.
In case the user executing the client (uC) is different to the user running the server (uS), the staged out files will belong to the latter user (uS) and might not be readable to the client user (uC). In that case, the above sample client will report the following error: „Error: /tmp/output2“.

3.5 Command Line Clients

The NegMrg provides various command line clients:

- wsag-get-templates
- wsag-create-agreement
- wsag-monitor
- wsag-terminate

We will now describe each client in more detail. The provided examples, executed one after another, represent a complete lifecycle example.

3.5.1 wsag-get-templates

This client requests available templates from the server. The option „*-c <name> <version>*“ can be used to convert the template of a specific application into an offer.

Example:

- > wsag-get-templates
  -s https://<host>:8443/wsrf/services/DomainAgreementFactory
  -o ./offer
  -c testtouch 1.0

Afterwards, some specific parameters in the offer need to be edited:

- wsag4jt:StartTime : needs to be set to a valid start time
- wsag4jt:Duration : needs to be adjusted to a reasonable duration (i.e. 120s)

Offers that have been created this way can be used to create an agreement with the help of the following client.

3.5.2 wsag-create-agreement

This client creates an agreement based on a file containing an agreement offer.

Example:

- > wsag-create-agreement
  -s https://<host>:8443/wsrf/services/DomainAgreementFactory
  -i ./offer-1.xml
  -o ./epr.xml

Upon successful agreement creation the client will create an Endpoint-reference that can be used as input to the following clients.

3.5.3 wsag-monitor

This client allows to monitor the state of an agreement, as well as ist ServiceTermStates and GuaranteeTermStates. Options “-p” and “-d” can be used to monitor these properties periodically.

Example:

- > wsag-monitor
  -e ./epr.xml
  -p 2
  -d 6

The above example will monitor the agreement two times (-p 2) and wait six seconds (-d 6) after each monitoring. Once the state of the agreement equals “Completed” the client will exit.

3.5.4 wsag-terminate

This client can be used to terminated an agreement.

Example:

- > wsag-terminate
  -e ./epr.xml

4 References

[1] ***http://www.zib.de/cluster-usr/twiki/pub/Main/AdvanceReservations/d5_6_rsv.pdf***