Setup Desktop Grids and Bridges

Tutorial

Robert Lovas, MTA SZTAKI
Outline of the SZDG installation process

1. Installing the base operating system
2. Basic configuration of the operating system
3. Installing the SZTAKI LDG packages from eCom4Com
4. Creating a BOINC project
5. Managing the BOINC project
6. Deploying the application on the LDG
Step 0: Installing the OS

- SZTAKI Desktop Grid is supported on Debian/GNU Linux stable (Lenny) i386 and amd64 platforms
- Hardware mainly depends on the apps, for the infrastructure itself with simple apps up to a few thousand clients any current system is sufficient
- See Debian install documentation for details at http://www.debian.org/releases/stable/installmanual
- SZTAKI Desktop Grid distribution and documentation http://www.desktopgrid.hu/
Step 1: Basic OS configuration

- Make sure /etc/hosts contains your FQDN

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Hostname</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>localhost.localdomain</td>
</tr>
<tr>
<td>192.168.192.193</td>
<td>boinc.lpds.sztaki.hu</td>
</tr>
</tbody>
</table>

- Make sure e-mail works (MTA installed)

```bash
apt-get install postfix
apt-get install exim; eximconfig
```

- Set up package repositories:
  - Add to /etc/apt/sources.list (as one single line):

```
deb http://www.desktopgrid.hu/debian/ etch szdg
```

- Run `apt-get update` to update the repository cache
Step 2: Install SZTAKI Desktop Grid

- Install dependencies:
  
  ```
  apt-get install apache2-mpm-prefork libapache2-mod-auth-plain
  apt-get install libapache2-mod-php5 php5-cli
  apt-get install mysql-server-5.0 php5-mysql
  apt-get install pwgen
  ```

- To install BOINC with the standard web interface, type:
  
  ```
  apt-get install boinc boinc-skin-standard
  ```

- To install the SZTAKI Local Desktop Grid interface, type:
  
  ```
  apt-get install boinc boinc-skin-ldg
  ```

- To install the DC-API development files for BOINC, type:
  
  ```
  apt-get install libdcapi-boinc-dev
  ```

- To install the DC-API devel. files for local execution, type:
  
  ```
  apt-get install libdcapi-local-dev
  ```
Step 3: Creating a project (as root)

- Creating a new project takes just a single command:

  ```
  boinc_create_project --name=test --long-name="Test Project"
  ```

  - This will create a UNIX user named `boinc-test` and a MySQL database/user named `boinc_test`
  - All files belonging to the project are under the directory `/var/lib/boinc/test` which is the same as `~boinc-test`

- Create a normal user account (this will be a project admin)

  ```
  adduser pradmin
  ```

- Make the `pradmin` user a project administrator:

  ```
  boinc_admin --name=test --add pradmin
  ```

  - Password set here is for admin web interface accessible as `http://<host name>/<project short name>/ops`
Step 4: Managing the BOINC project

- The project is now ready to use
- Root privileges are not needed anymore
  - so log out as root and log in as pradmin
- You can assume project administrator role by running
  ```
sudo su - boinc-test
  ```
- After the above command the environment is set up so that you can issue BOINC administrative commands directly, such as: start, stop, etc.
- You can start the project now typing
  ```
  start
  ```
Step 5: Deploying the application on the LDG

- In plain BOINC, application deployment is a many step process
- With SZDG applications can be packaged, deploying a package is just one command:
  
  boinc_appmgr --add primesearch.tar.gz

- boinc_appmgr uses application descriptors in the package
  - client.xml
  - master.xml
3G Bridge

BOINC

Job source
BOINC client

Job source
gLite GRAM

Job Handler Interface

Job Database

Queue Manager

grid Handler interface

DC-API Plugin

gLite Plugin

XtremWe b Plugin

Xtrem Web

gLite

gUSE/WS-
PGRADe
portal

gUSE
job submitter

RI-261556  6th IDGF Tutorial
The Generic Grid-Grid Bridge (3G Bridge) is a software component used within the EDGeS project that provides the core component of the Service Grid - Desktop Grid interoperability solution.

**Project Home**
edges-3g-bridge.sf.net

**Develop**
sf.net/projects/edges-3g-bridge/devel

**Support**
sf.net/projects/edges-3g-bridge/support

**Last Update**
2010-11-01

**License**
GNU General Public License (GPL)

**Registered**
2009-07-05

**Release Date**
2010-11-01

**Operating System**
All POSIX (Linux/BSD/UNIX-like OSes)

**Programming Language**
C++
Scenario 1 – DG to gLite via bridge

User entry point is DG – using gLite is completely transparent from user’s point of view.
Outline (Part I)

• What this HOWTO is about?
• BOINC → gLite bridge in detail
• Prerequisites
• What this HOWTO is about?
  • Prerequisites
Aim of this HOWTO

• You are: a BOINC project admin
• You want to:
  – improve the computation performance of your Grid
  Use DesktopGrid

• With technology: 3G Bridge

• In a nutshell: run your BOINC workunits on the DesktopGrid VO
System overview

BOINC project server

- Scheduler
- Work gen.
- Assim.
- WU/App/Result Database

DesktopGrid VO

- Computing Element\textsubscript{1}
- Computing Element\textsubscript{2}
- BOINC to EGEE Bridge

WMS LB BDII …
BOINC \rightarrow gLite bridge details

• Task to be solved:
  • Process BOINC workunits
  • In a gLite-based infrastructure

• Using a bridge that:
  • Is able to handle BOINC workunits
  • And is able to create gLite jobs from the workunits, and run them on gLite-based Grid
BOINC → gLite

Bridge solution concept

- Wrapped workunit execution:
  - Fetch BOINC workunits
  - Parse the workunits’ contents instead of starting them, and wrap them into a package
  - Send the package to the 3G Bridge
  - An gLite plugin of the 3G Bridge arranges the package execution on gLite
  - The result of the gLite execution (output package) is unpacked, and results are sent back to the BOINC project
BOINC → gLite bridge
Using 3G Bridge concept

- Collect jobs originating from BOINC:
  - Place them in a queue
  - New jobs in the queue are periodically handled by an gLite plugin, that
  - Uses Collection possibilities of gLite to submit many jobs in one request
- This way the usage of the WMS is reduced
Prerequisites

• A BOINC project
• A DesktopGrid VO
• An gLite User Interface machine with:
  • BOINC jobwrapper client installed
  • BOINC jobwrapper installed
  • 3G Bridge with gLite plugin support installed
  • DesktopGrid VO configured
Tasks of the BOINC project admin I.

• Get a grid certificate from your national CA
  • Certificates are essential for accessing gLite services
  • Consists of two parts:
    • Public key
    • Private key protected by a password
  • Usually are valid for a year, can be extended
  • Are used to identify you within the gLite grid infrastructure
Tasks of the BOINC project

admin II.

• Upload a long-term proxy to the EDGeS MyProxy server
  • Proxies are generated from your certificate by decoding its key (using the password) and offering usually a short lifetime (few hours)
  • Long-term proxies are stored on trusted entities (MyProxy servers), are used to generate short-term proxies in a trustworthy manner
• Use this command:
  \[\text{GT\_PROXY\_MODE=old myproxy-init -s myproxy.grid.edges-grid.eu -d -n}\]
Tasks of the BOINC project admin III.

- Send your certificate’s subject to the Bridge Admin
- Create a new BOINC user on the BOINC project
- Send the BOINC project’s URL to the Bridge Admin
- Send the new BOINC user’s account key to the Bridge Admin
Tasks of the Bridge admin I.

- Wait for the info provided by the BOINC admin
- Update 3G Bridge config file:

```plaintext
[new_boinc]
name = desktopgrid.vo.edges-grid.eu
myproxy_host = myproxy.grid.edges-grid.eu
myproxy_port = 7512
myproxy_user = /C=HU/...
myproxy_authcert = /etc/grid-security/bridge.cert
myproxy_authkey = /etc/grid-security/bridge.key
jobbaseurl = #http://fn2.hpcc.sztaki.hu/dpm/hpcc.sztaki.hu/home/desktopgrid.vo.edges-grid.eu/3g-bridge_u11.grid.edges-grid.eu/
joblogdir = /var/log/3g-bridge/joblogs
joblogs = error

[jobwrapper-newproject]
grid = new boinc
jobwrapper-path = /var/log/3g-bridge/jobwrapper-newproject-wrapper.log
```

Certificate subject
Tasks of the Bridge admin

II.

• Create a new algorithm queue in the 3G Bridge database for the ‘new_boinc’ plugin:
  mysql> insert into cg_algqueue(grid, alg, batchsize) values('new_boinc', '', 10);
  • The above command adds a new queue for the ‘new_boinc’ plugin using any executable and using collection size 10 during job submission

• Restart the bridge, so the new plugin will be initialized
Tasks of the Bridge admin III.

- Create a new working directory for the BOINC jobwrapper client
- Create jobwrapper_config.xml in the dir:
  - Use 10 CPUs
  - Use GUI RPC port 10000
  - Specify the jobwrapper binary
  - Also specify the config section

```xml
<cpu>10</cpu>
<jobwrapper_binary>/usr/libexec/3g-bridge/jobwrapper</jobwrapper_binary>
<bridge_conf>/etc/3g-bridge.conf</bridge_conf>
<conf_section>jobwrapper-newproject</conf_section>
<gui_rpc_port>10000</gui_rpc_port>
```
Tasks of the Bridge admin IV.

- Attach to the BOINC project
- Restart the BOINC jobwrapper service on the gLite UI machine
DesktopGrid VO activities

- CPU > 1 Day/Week
- CPU > 1 Month/Week
- CPU > 1 Year/Week

Activity information by VO
January 2009 - May 2010
Scenario 2 – gLite to DG via bridge

Desktop Grid 1

Desktop Grid n

DEGISCO Services

gLite VO

WMS and other EGEE services

gLite user (using gLite UI machine or portal)
- User entry point is gLite
- using DG is transparent from user’s point of view
How to connect your SG system to EDGI/DEGISCO? (Part II)

- This HOWTO is about setting up the gLite→DG bridge
- The title assumes user view (i.e. you want your jobs in your SG to go to DGs)
- From the admin view it requires more work from the DG admin and may look more like adding DGs to an SG, but don't get confused by this
- In this session you will see how to prepare your DG set up earlier to accept gLite jobs (as a DG admin)
What can be bridged?

- Let there be a validated version of an application in the AR with executables for gLite and different DG systems (and on DGs for different platforms).
- This application (the client part) is deployed on a DG that is connected to the bridge and this DG is registered in the AR as supporting the application.
- An gLite VO is also registered in the AR as an allowed source of jobs for this application.
How does bridging work?

- When a job is submitted to a bridge CE it checks the following:
  - Executable matches the one in the AR for the source VO by MD5 hash
    1. The source VO must be allowed
    2. The application executable must be allowed
  - The target DG is registered as supporting the application (the DG version is deployed there)
- If the above are true the job is bridged if false then the job is rejected
What needs to be set up?

• On the DG side:
  – 3g-bridge queue manager
  – 3g-bridge wssubmitter service

• To get applications from the AR to be installed locally and to register installed applications:
  – gemlcaclli and gridftp clients

• On the gLite side
  – An lcg-CE with edges-BRIDGE

• Connecting the gLite CE to the wssubmitter(s)
Manual
Tutorial – BOINC and 3G-Bridge

Virtual gLite, BOINC and 3G-Bridge infrastructure
(provided by EDGI project)
Purpose of virtual infrastructure

The purpose of these VMs is to provide a base to easily set up local test or development infrastructure to be used for:

- getting to know these services or
- developing and testing applications and
- new middleware components

in a local usage scenario.
Virtual machine (VM) images are set up as a test infrastructure for EDGI project. Available “http://www.edgi-grid.eu/downloads/vmimages/”

Virtual machines are configured to function as:

- a minimal,
- self contained,

test infrastructure of the SG ⇒ DG (service grid to desktop grid) infrastructure.
Remarks

The components are similar to those used in the EDGeS/EDGI production infrastructure however, these VMs are not meant to be used for setting up a public production infrastructure.

The focus while creating these VMs were on:
✓ easy installation and
✓ simple local usage
as opposed to :
✗ performance and
✗ security
which in this setup do not meet the requirements of production usage.
Overview of Virtual machines

- testvoms.edgitest
  192.168.143.101
  VOMS, BDII_site, CA
  SL5 x86_64, gLite 3.2

- testui.edgitest
  192.168.143.100
  UI, BDII_top
  SL5 x86_64, gLite 3.2

- testwms.edgitest
  192.168.143.102
  WMS, LB
  SL4 i386, gLite 3.1

- testce.edgitest
  192.168.143.103
  Lcg-CE, bridge-CE
  SL4 i386, gLite 3.1

- testboinc.edgitest
  192.168.143.105
  BOINC, 3g-bridge
  Debian 5.0, SZDG 6.11
Available with documentation

http://www.edgi-grid.eu/downloads/vmimages/v1.0/