

Table of Contents

VO software installation and maintenance.....	1
Basic Requirements.....	1
Grid middleware integration.....	1
Available options.....	1
LJSFI.....	2
General information.....	2
Architecture.....	2
Installation.....	2
Server.....	2
Client.....	2
Usage.....	2
Known issues.....	4

VO software installation and maintenance

Basic Requirements

While many individual grid users supply the executables, libraries and application files required by their grid jobs during submission, specialized VOs have teams that use a common set of applications. For consistency reasons (ie all VO users use the same programming code) and practical reasons, as sometimes the applications can have very large size, grid sites support a special shared storage area for each supported VO where they can install software which should be available to all their members using a grid site. The benefits of this practice include optimal usage of network/storage resources, as the software needs to be deployed only once, instead for each job individually (and with some sites supporting thousands of jobs this could be a serious problem) which also results in reduction of time consumed for file transfers and instalaltions.

Depending on the Linux distribution being used, software can be installed via RPM packages, .tgz packages, .debs, or directly from source code compilation either manually or via an automatic system. However most of these methods are designed towards installing system wide available software in the OS file system which means they require increased, typically root, privileges. The simplest method to overcome this is to install the software from a compressed file (typically a .tar.gz file) to the shared VO software area.

Grid middleware integration

As far as glite middleware is concerned, a VO software installation task is no different than any other grid job. They payload is executed by a special pool account designed as software manager. The glite middleware also supplied the lcg-tags command, which allows the VO software manager to add to the sites information system special tags which indicate the presence of a certain software. Grid users can then adjust their jobs to match only sites containing specific VO software required.

Available options

Installing a software across grid sites can be a very challenging task, especially if this software evolves during time, with newer version coming out, old ones becoming obsolete, while there may be a requirement for multiple versions to be installed at the same time. Large VOs like the LHC VOs have pioneered in this area, due to the nature of their work and being the first ones to use the grid. Various methods have been developed to perform grid scale software installations. The Atlas VO in particular has developed a very attractive tool, called LJSFI (Light Job Submission Framework for Installation), which recently has been expanded to support VOs other than atlas.

LJSFI

General information

The LJSFI is described as "VO-independent framework for job tracking and task management in LCG/EGEE. The framework is a thin layer over the Grid middleware, built partially of shell scripts, to wrap the Grid commands, and python scripts to interface to the database. More information can be found here <http://iopscience.iop.org/1742-6596/119/5/052013/pdf?ejredirect=.iopscience> . It is developed and maintained by Alessandro De Salvo.

Architecture

The LJSFI software consists of a server and a client part. The server part is consisted a database and web interface. The web interface is being used to define and request tasks such as installation, removal, patching, or anything that a software manager can deem necessary. It is also used to define software releases that are to be installed and their target machine architecture as well as groups of sites where operations such as installation/removal are to take place.

The client part contains a number of scripts that queries the database about pending requests, launches the actual software installation job against the sites using a set of scripts and jdl templates supplied by the software manager in a special directory. It also updates the database with the status of the tasks requested.

Some more information can be found here:

<https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiArchitecture>

Installation

During the tests of the NA4 VO support group, installation was performed with the instructions available at <https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiInstallation> . The client system was installed as glite 3.2 UI on a SL 5.4 x86_64 while the server was installed on SL 4.8 x86_64.

Server

The server requires a host certificate. A normal Scientific Linux installation, with apache, php, mod_ssl, php-ldap is sufficient.

When the above are available installation can proceed according to the instructions at <https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiServerInstallation> .

Client

For the client a Scientific Linux 4 User Interface with host certificate is sufficient. Step by step instructions are available at <https://atlas-install.roma1.infn.it/twiki/bin/view/Main/LJSFiClientInstallation> . Note that while not mentioned, the installation of the client should be done with a normal user account and not root.

Usage

Let's assume we want to install a software called analyzer, in version 1.0 which we have build for noarch and packaged in a file call analyzer-1.0.tar.gz. In the web interface , first we visit the Releases menu. There we need to:

1. Define the release name, in our case analyzer-1.0
2. Define the release type, for most cases that should be stable release.
3. Define the release architecture id and the target machine architecture id. For our example this is noarch. We can define additional architectures from the Architectures drop down menu on the left.
4. Define the defaults tasks related to software management. For our example we only need to install software so we define only the default installation task as release installation.
5. Auto installation targets: Usually, that should be All targets. That will install the software in all the sites supporting the VO.
6. Tag name: The tag to be added to the grid site information system when software is installed. In our case, that should be VO-see-analyzer.
7. The rest of the fields are self explained. Help is always available from the question mark next to each choice

Next we need to create the scripts and jdl's to be used for every task we defined, in our case the installation task. This is done in the client machine with the following steps:

1. Inside the directory where we installed the client software (usually VOname/ljsfi), there is a templates/VOname directory. It contains template jdl's for various tasks. For installation we need to edit the install-release.jdl.template. The default points to a "fake-install" script located in the "scripts" folder. For our example we create an install-analyzer script with the following contents:

```
#!/bin/sh
RELEASE=$1
HOST_CE=$2
echo $1
echo $2
echo "Sample installation script"
echo "This script tries to install a software called analyzer from a tarball"
echo "This job ran on `hostname -f` on `date`"
echo "The install.xml file, required for the short description of the job,"
echo "will be created now."

cat > install.xml <<EOD
<?xml version=1.0 encoding=UTF-8?>
<install>
<fakeinstall type="INFO" datetime="`date`">Running on `hostname -f`</fakeinstall>
</install>
EOD
WKDIR=`pwd`
wget http://tassadar.physics.auth.gr/analyzer.tar.gz
cd /opt/exp_soft/see/
tar -xzf $WKDIR/analyzer.tar.gz
echo lcg-tags --ce $HOST_CE --add --tags VO-see-analyzer-${RELEASE}
lcg-tags --ce $HOST_CE --add --tags VO-see-analyzer-${RELEASE}
exit
```

1. Next we need to create the corresponding jdl for this task. In the ljsfi folder templates/VONAME there are some sample jdl's available. Our install-release.jdl.template contains the following:

```
Executable = "install-analyzer";
InputSandbox = {"@SCRIPTPATH@/install-analyzer"};
OutputSandbox = {"stdout", "stderr", "install.xml"};
stdoutoutput = "stdout";
stderrerror = "stderr";
Arguments = "@RELEASE@ @SITE_CENAME@";
Environment = {"LFC_HOST=@LFC_HOST@"};
VirtualOrganisation = "@VO@";
Requirements = other.GlueCEUniqueId=="@SITE_CS@";
```

All should be set now but nothing is executed until the installation agent is launched in the client machine. To

do this we:

1. Obtain grid credentials
2. Launch the daemon that will monitor the installation database and launch the tasks: In the bin directory, execute "autoinstall start"
3. From the web interface, select a site where the software is to be installed from the "Request an installation" menu.
4. Logging information is stored in the var/log folder.

Status of the installations requests is available in the web interface via the "Show requests menu"

After some time, depending on the grid workload, the installed should have finished.

In a similar fashion as for install, we can define and create arbitrary tasks such as removal of release we don't want any more. The ljsfi comes with template tasks for patching or verifying installed software. A VO software manager can define his own arbitrary tasks to suit his requirements.

Known issues

1. The proxy renewal mechanism from myproxy does not to work. That means when our initial proxy expires (after 12 hours by default), the agent will get stuck and increase machine load to 3-4 Use long lived proxies instead till this is fixed.
2. Automatic deployment in all sites supporting the VO is not working in the current version due to a problem still under investigation.

-- DimitrisZilaskos - 15-Jan-2010

This topic: EGEE > VOSoftwareInstallation

Topic revision: r7 - 2010-04-28 - DimitrisZilaskos



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback