Table of Contents

SFT Test Suite in Pre-Production

Overview

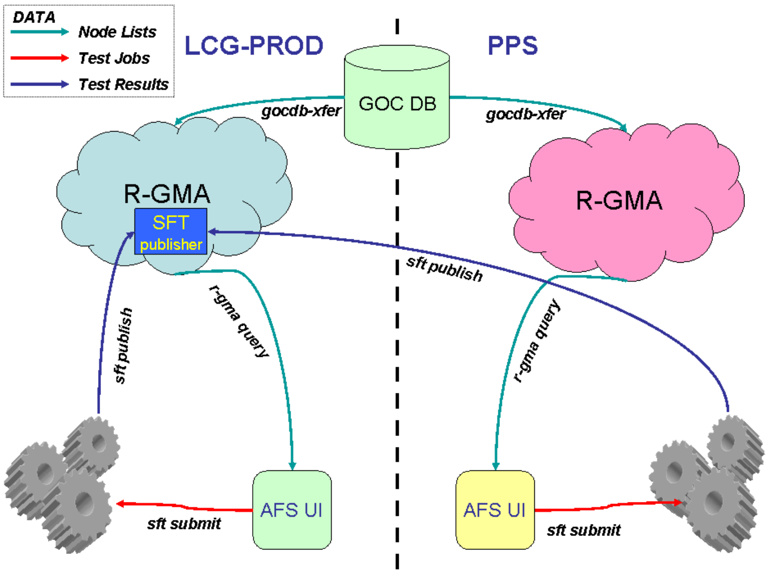
SFT (Site Functional Tests) is a test application used to run basic functionality tests of a site.

Sites in PPS have to register their CEs in the GOC db, in order to have them monitored.

Basically the flow of information from the GOC db to the SFT works as follows:

1. Sites and nodes are inserted by the site administrators into the GOC db.
2. A script is used to query this information and populate appropriate tables on the R-GMA db.
3. SFT queries the R-GMA and creates its own list of nodes.
4. SFT runs the test on the nodes in the list

The results of the SFT tests are published by the SFT publisher in the production framework as displayed in the picture below:



Installation and configuration

The "official" installation and configuration reference for the Site Functional Tests suite is ... no more there. Please contact operational-documentation@mailmanNOSPAMPLEASE.egi.eu for comments or questions.

That guide, maintained by Piotr Nyczyk is addressed to site administrators in the LCG production system.

It contains all the instructions needed to install and configure the SFT client and server.

For PPS purposes anyway, in order to implement the connections described in the picture above, only a subset of the installation procedure has to be done, this is due to the fact that the SFT web publisher in use is the production one and that the gLite R-GMA installation tool installs an archiver by default.

In the following procedure, for the sake of convenience, we will reproduce the command lines we used to set up the SFT in PPS. Nevertheless the links to original procedures, when provided, indicate that those should be considered the **real** reference. Therefore, if you find that the original procedures have been changed and that the proposed steps are obsolete, please feel free to update them

The overall configuration steps you need to do to implement such an infrastructure are:

- Install and configure a gLite R-GMA server (also known as MONBOX)
- Set-up the query to gather the list of PPS CEs from the GOC db
- Install and configure the SFT client on the AFS UI
- Run SFT and publish the results

Install and configure a gLite R-GMA server

You should follow the gLite Installation instructions of a MON node in

http://grid-deployment.web.cern.ch/grid-deployment/documentation/LCG2-Manual-Install/

Set-up the query to gather the list of PPS CEs from the GOC db

NOTE: The reference procedure for the following steps is ...no more there. Please contact operational-documentation@mailmanNOSPAMPLEASE.egi.eu for comments or questions.

If you find inconsistencies in the steps below please correct them on this page.

In order to set-up the regular query to the GOC db you have to:

- create the **GocSite_v0_4** and **GocNode_v0_4** tables.
- populate the R-GMA with the data in the GOC db.
- configure the archiver on the RGMA server.
- Set-up a cron job to repeat regularly the export.

The installation and configuration steps described in this section have been done once in PPS. They are reported in this document for future reference, but likely most of them will not need to be done again, **with few exceptions** (e.g. changes to be made into the UI setenv scripts). Please just go through the list and check that everything is exactly as it is supposed to be.

Create the GocSite_v0_4 and GocNode_v0_4 tables

If the tables **GocSite_v0_4** and **GocNode_v0_4** do not exist on the R-GMA schema server, you need to create them. You can check R-GMA to see if the tables have been created before.

e.g. https://lxb2093.cern.ch:8443/R-GMA/

Alternatively, in order to verify that the tables GocSite_v0_4 and GocNode_v0_4 are there, you can also run

```
> rgma -c "show tables" | grep GocSite_v0_4
| GocSite_v0_4                              |
> rgma -c "show tables" | grep GocNode_v0_4
| GocNode_v0_4                              |
```

You need the RGMA client installed so that the following commands can be run on a UI or on whatever machine has an RGMA client installed. You could install the client on the RGMA server itself (e.g the machine you are using as SFT server) **BUT**, in order to interact with the registry (see later in the procedure) against a secure RGMA server (as the gLite one is) you need also a proxy, so it is definitely better to run on a UI.

- log into a UI

- Assure that you can connect to rgma and everything looks fine
    - `> voms-proxy-init -voms dteam` (or equivalent)
    - `> rgma`

```
Welcome to the R-GMA virtual database for Virtual Organisations.
================================================================

Your local R-GMA server is:

  https://lxb2093.cern.ch:8443/R-GMA

You are connected to the following R-GMA Registry services:

  https://pps-rgma-server.egee.cesga.es:8443/R-GMA/RegistryServlet

You are connected to the following R-GMA Schema service:

  https://pps-rgma-server.egee.cesga.es:8443/R-GMA/SchemaServlet

Type "help" for a list of commands.

rgma>
```

- create the tables
    - `rgma> CREATE TABLE GocSite_v0_4 (siteID integer primary key, officialname varchar(100), sitename varchar(100), friendlyname varchar(100), domain varchar(50), homeURI varchar(255), country varchar(50), tier integer, giisUrl varchar(250), inMonitoring varchar(1), status varchar(30), type varchar(30), region varchar(50), inMaintenance varchar(1))`
    - `rgma> CREATE TABLE GocNode_v0_4 (nodeID integer primary key, siteID integer, nodetype varchar(50), nodetype2 varchar(50), hostname varchar(50), domain varchar(200), ip varchar(15), grp varchar(20), hostdn varchar(255), monitor varchar(1))`

WARNING: R-GMA, starting from version 1.5, allows the tables to be created with a sql-like statement, with no more need to use the *create_table* utility. Unfortunately the definition of the SQL is not complete. So pay attention to use the type **'integer'** and **NOT** 'int' in your *create* statement, otherwise the creation will fail (with no error messages) and you will get a run-time unhandled exception querying the table.

- check the tables to have been correctly created

```
rgma> describe GocSite_v0_4

+-----------------+--------------+-------------+-------------+
| Column name     | Type         | Primary key | Can be NULL |
+-----------------+--------------+-------------+-------------+
| siteID          | INTEGER      | Yes         | No          |
| officialname    | VARCHAR(100) | No          | Yes         |
| sitename        | VARCHAR(100) | No          | Yes         |
| friendlyname    | VARCHAR(100) | No          | Yes         |
| domain          | VARCHAR(50)  | No          | Yes         |
| homeURI         | VARCHAR(255) | No          | Yes         |
| country         | VARCHAR(50)  | No          | Yes         |
| tier            | INTEGER      | No          | Yes         |
| giisUrl         | VARCHAR(250) | No          | Yes         |
| inMonitoring    | VARCHAR(1)   | No          | Yes         |
| status          | VARCHAR(30)  | No          | Yes         |
| type            | VARCHAR(30)  | No          | Yes         |
| region          | VARCHAR(50)  | No          | Yes         |
| inMaintenance   | VARCHAR(1)   | No          | Yes         |
| MeasurementDate | DATE         | No          | No          |
| MeasurementTime | TIME         | No          | No          |
+-----------------+--------------+-------------+-------------+
rgma> describe GocNode_v0_4

+-----------------+--------------+-------------+-------------+
| Column name     | Type         | Primary key | Can be NULL |
+-----------------+--------------+-------------+-------------+
| nodeID          | INTEGER      | Yes         | No          |
| siteID          | INTEGER      | No          | Yes         |
| nodetype        | VARCHAR(50)  | No          | Yes         |
| nodetype2       | VARCHAR(50)  | No          | Yes         |
| hostname        | VARCHAR(50)  | No          | Yes         |
| domain          | VARCHAR(200) | No          | Yes         |
| ip              | VARCHAR(15)  | No          | Yes         |
| grp             | VARCHAR(20)  | No          | Yes         |
| hostdn          | VARCHAR(255) | No          | Yes         |
| monitor         | VARCHAR(1)   | No          | Yes         |
| MeasurementDate | DATE         | No          | No          |
| MeasurementTime | TIME         | No          | No          |
+-----------------+--------------+-------------+-------------+
```

Populate the R-GMA with the data in the GOC db

To perform this step you need to run a script called **gocdb-xfer.py** developed by Min Tsai.

NOTE: The script needs the **MySQLdb** python module on the UI, so please check that the **MySQL-python** rpm is installed on the UI running `rpm -q MySQL-python`. If not installed, you should install it.

- Download the rpm (a copy is present on the certification website) and install it:
    - `> wget http://www.cern.ch/egee-middleware-certification/extra-rpms/MySQL-python-0.9.1-6.i38`
    - `> rpm -ivh MySQL-python-0.9.1-6.i386.rpm`

- **If you cannot run "rpm"** (e.g. because you are using an AFS UI) to manage to have the package installed is a little bit more tricky. In PPS we use an AFS UI, therefore we realized a shared installation of the package MySQL-python. Steps from 1 to 4 have already been made, so if you are following this procedure (e.g. beacause you are enabling a new UI to use SFT) presumably you need to start only from step 5:
    1. `> cd /afs/cern.ch/project/gd/egee/vn`
    2. `> wget http://www.cern.ch/egee-middleware-certification/extra-rpms/MySQL-python-0.9.1-6.i38`
    3. `> rpm2cpio MySQL-python-0.9.1-6.i386.rpm | cpio -iumd`
    4. `> rm MySQL-python-0.9.1-6.i386.rpm`
    5. `> vi /afs/cern.ch/project/gd/egee/glite/ui_PPS14/glite_setenv.csh`

       ```
       [...]
       # Python module Needed for gocdb-xfer (only PPS)
       setenv PYTHONPATH ${PYTHONPATH}:/afs/cern.ch/project/gd/egee/vn/usr/lib/python2.2/si
       ```
    6. `> vi /afs/cern.ch/project/gd/egee/glite/ui_PPS14/glite_setenv.sh`

       ```
       [...]
       # Python module Needed for gocdb-xfer (only PPS)
       export PYTHONPATH=${PYTHONPATH/afs/cern.ch/project/gd/egee/vn/usr/lib/python2.2/site
       ```

- From http://goc.grid.sinica.edu.tw/cvs/SER/GOC/gocdb-xfer/bin/ download into the directory */afs/cern.ch/project/gd/egee/gocdb-xfer* the following scripts:
    - GocDB.py
    - MyRgma.py

- ♦ gocdb-xfer.py

- Edit the file *gocdb-xfer.conf* and insert username and password of the GOC db. Ask Antonio Retico or Min Tsai if you don't know them.

```
####################################
# Config file for gocdb-xfer
####################################
# Default values: do not modify this section
[DEFAULT]

[gocdb]
dbhost = goc.grid-support.ac.uk
dbuser = <username>
dbpass = <password>
dbname = goc-2_0
```

- Edit the file *GocDB.py* and insert the correct values for *root_dir* and *config_path* , e.g.

```
...
root_dir = "/afs/cern.ch/project/gd/egee/gocdb-xfer/"
config_path = os.path.join(root_dir, "gocdb-xfer.conf")
...
```

- Run the script to export data from the GOC db (The script needs a proxy to be created)
    - ♦ `> chmod 700 gocdb-xfer.py`
    - ♦ `> ./gocdb-xfer.py` (It takes some minutes)

After the script finishes, the sites should be visible in the RGMA browser but you have to run a "Continuous & old" query. That means that you are querying directly the producer and the data are only temporarily available. In order to instantiate a secondary producer and have the data permanently stored in the mysql database, you need a secondary producer and the archiver so you need the flexible archiver to be configured.

Configure the archiver on the R-GMA server

Using LCG-2 MONBOX + flexible archiver

This option was used in the SFT client instance installed to monitor LCG-2 CEs in PPS. There were serious compatibility issues introduced by R-GMA version 1.5 and, in order for the sft join query and the GocNode_v0_4 and GocSite_v0_4 to be correctly done by the SFT, the flexible archiver had to be used. This is the way I did:

- configure the lcg-archiver as suggested in Piotr's instructions **except** for the tables.

```
> vi /opt/lcg/etc/lcg-archiver.conf

# Essential parameters
RGMA_HOME = /opt/glite
RGMA_PROPS = /opt/glite/etc/rgma

# Tables to be archived
#tables = userTable GlueCE:LATEST GlueCluster:HISTORY
tables = userTable GocSite_v0_4:LATEST GocNode_v0_4:LATEST


# Database settings
db_user = rgma
db_passwd = xxxx
db_latest = jdbc:mysql://localhost/latestProducer
db_hist= jdbc:mysql://localhost/dbProducer
```

```
# Possible values: CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET
logging_level = ERROR
```

- `> service lcg-archiver restart`
- `> chkconfig lcg-archiver on`
- configure the flexible archiver

```
> vi /opt/glite/etc/rgma-glue-archiver/glue.config

# glue archiver config file

Type=latest

DBName=jdbc:mysql://localhost/latestProducer
DBUsername=rgma
DBPassword=xxxx

HistoryRetentionPeriod=90

Tables=
GocNode_v0_4
GocSite_v0_4

GocNode_v0_4_HistoryRetentionPeriod=1440
GocSite_v0_4_HistoryRetentionPeriod=1440
```

- `> /etc/init.d/rgma-glue-archiver start`
- `> chkconfig rgma-glue-archiver on`

Set-up a cron job to repeat regularly the export.

You need to run the *gocdb-xfer.py* script regularly. A convenient solution is to schedule the run via a cron job.

If you are running on a secure R-GMA (as it is the case in PPS) you might want the needed user proxy to be atomatically created as well. The following example is similar to the *acron* job we run in PPS.

- `> touch mysecret`
- `> chmod go-rxw mysecret`
- Type your PEM passphrase into *mysecret*
- `> vi /afs/cern.ch/project/gd/egee/gocdb-xfer/launch-gocdb-xfer.sh`

```
#!/bin/sh

#!/bin/sh

source /afs/cern.ch/project/gd/egee/glite/ui_PPS14/glite_setenv.sh
cat mysecret | voms-proxy-init -voms dteam -pwstdin
sleep 2
/afs/cern.ch/project/gd/egee/gocdb-xfer/gocdb-xfer.py

exit
```

- `> acrontab -e`

```
11 * * * * lxplus062.cern.ch /afs/cern.ch/project/gd/egee/gocdb-xfer/launch-gocdb-xfer.sh> /afs/c
```

NOTE: The choice of lxplus062.cern.ch is for the time being mandatory since only connections coming from this node are currently authorized on the GOC db.

Install and configure the SFT client on the AFS UI

The client has been installed.

The installation directory is

/afs/cern.ch/project/gd/egee/sft-glite

The configuration file *defaults.glite* is

```
SFT_JOB_SUBMIT_CMD=glite-job-submit
SFT_JOB_STATUS_CMD=glite-job-status
SFT_JOB_OUTPUT_CMD=glite-job-output
SFT_JOB_LOGGING_CMD=glite-job-logging-info
SFT_JOB_LISTMATCH_CMD=glite-job-list-match
SFT_JOB_CANCEL_CMD=glite-job-cancel

SFT_PUBLISHER_PROXY=http://lcg-sft-publish.cern.ch:8083/sft/publishTuple

SFT_GOC_MAP_SELECT="select GocSite_v0_4.siteID,hostname,sitename,region,inMaintenance from GocSit

#SFT_GOC_MAP_URL=http://grid-deployment.web.cern.ch/grid-deployment/gis/sft2/glite-nodes.txt
SFT_LCG_VER_FILTER="LCG-[23]_[4567890123]"
```

The default flavour of the SFT tests used on PPS is "glite". The *defaults* file is

```
SFT_VO=dteam

# default definitions for status codes
SFT_OK=10
SFT_INFO=20
SFT_NOTICE=30
SFT_WARNING=40
SFT_ERROR=50
SFT_CRITICAL=60

SFT_TYPE=glite

SFT_LCG_CATALOG_TYPE=lfc
SFT_LFC_HOME=/grid/$SFT_VO/SFT

SFT_SAME_PUBLISHER_WSDL=http://gvdev.cern.ch:8080/gridview/services/WebArchiver?wsdl
```

The set of tests done for gLite has been re-defined and estended to the whole set previously used for the LCG CEs. The *tests.glite file* looks like:

```
sft-wn
sft-softver
sft-caver --conf data/ca_data.dat --web
sft-brokerinfo
sft-csh
sft-lcg-rm
sft-vo-tag
sft-vo-swdir
sft-rgma
sft-rgma-sc
sft-crl
sft-apel
```

The list of RB in *prefRB.lst.glite* is

```
lxb2059.cern.ch
```

NOTE: `lxb2059.cern.ch` is a **gLiteWMS**, which allows jobs to be sent both to LCG and gLiteCEs.

The list of SE in *prefSE.lst* is

```
grid007g.cnaf.infn.it
lxb2058.cern.ch
```

NOTE: The choice of `grid007g.cnaf.infn.it` as the destination for the RM tests was carefully made. This SE has the important characteristic to be known both by the **Production** and **pre-production** BDIIs. So it basically belongs to two grids. This allows also PPS CEs accessing the production WNs to pass the RM tests.

Details of the PPS SFT client instance

The PPS SFT client has been set-up to write on `lxb1908` in `/afs/cern.ch/project/gd/egee/sft-glite` as follows:

```
> cat /afs/cern.ch/project/gd/egee/sft-glite-workdir.cfg
SFT_WORK=$HOME/.sft-glite
```

To use it you need to specify the configuration file in the command line.

```
>  ./sftests -c sft-glite-workdir.cfg submit
>  ./sftests -c sft-glite-workdir.cfg status
>  ./sftests -c sft-glite-workdir.cfg publish
```

I created a script to run it directly from lxb1908

```
> cat /afs/cern.ch/project/gd/egee/sft-glite/submit-sft-glite-tests.sh
#!/bin/sh

source /etc/glite/profile.d/glite_setenv.sh

cat mysecret | voms-proxy-init -voms dteam -pwstdin
sleep 2
/afs/cern.ch/project/gd/egee/sft-glite/sftests -c /afs/cern.ch/project/gd/egee/sft-glite/sft-glit
sleep 2
/afs/cern.ch/project/gd/egee/sft-glite/sftests -c /afs/cern.ch/project/gd/egee/sft-glite/sft-glit

exit
```

Set-up a cron job to repeat regularly export and tests.

The *gocdb-xfer.py* and *submit-sft-glite-tests.sh* scripts need to be run regularly. A convenient solution is to schedule the run via a cron job.

```
> cat /afs/cern.ch/project/gd/egee/gocdb-xfer/launch-gocdb-xfer.sh

#!/bin/sh

source /etc/glite/profile.d/glite_setenv.sh
cat mysecret | voms-proxy-init -voms dteam -pwstdin
sleep 2
/afs/cern.ch/project/gd/egee/gocdb-xfer/gocdb-xfer.py

exit
```

- `> acrontab -e`

```
00 * * * * lxb1908.cern.ch /afs/cern.ch/project/gd/egee/gocdb-xfer/launch-gocdb-xfer.sh> /afs/cer
40 * * * * lxb1908.cern.ch /afs/cern.ch/project/gd/egee/sft-glite/submit-sft-glite-tests.sh > /af
```

NOTE: connections from lxb1908.cern.ch had to be previously authorized from the administrators of the GOC DB.

If your UI node does not accept acrontab jobs, you need some further configuration to do. In the next paragraph I give an example of the extra configuration I needed on lxb1908. Thes configuration are very specific of CERN nodes, so you probably need to customize them for your institute.

Set up ARC (Authenticated Remote Control) on the UI

The following steps have been done following the indications in SetUpAnLHCbUI

- Install the arcd daemon rpm:

```
apt-get install arc-server
```

- edit the file */etc/xinetd.d/arcd* and change the value on "disable line" to "no"

```
( disable                = no)
```

- be sure that the file */etc/krb.realms* contains the lines:

```
.wgs.cern.ch CERN.CH
.plus.cern.ch CERN.CH
```

- be sure that CERN.CH is the first line of the file */etc/krb.conf*

(if the files above do not exist please install krbafs-1.1.1-11 rpm)

- be sure that the line

```
   arc     4241/tcp        # Authenticated Remote Control
```

is included in the file */etc/services*

- Make also sure that your local firewall is configured to let

in traffic to port 4241 (arc).

```
(-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 4241 -j ACCEPT)
```

- check the existence of the file */etc/krb5.keytab*.

If not do the command

```
 cern-config-srvtab
```

to obtain the /etc/srvtab and /etc/krb5.keytab files

- add users to the file */usr/libexec/arcd/ACL* as following:

```
aretico@CERN.CH
```

- restart xinetd (/etc/init.d/xinetd restart)

more info about acrontab can be retrieved at:

link to arc

Usage

For generic SFT user instructions please consult

...

To use correctly the PPS-specific installation

- Log onto `lxb1908`
- Source the correct UI environment

```
> source /etc/glite/profile.d/glite_setenv.csh
```

- Create a proxy (voms-proxy-init or grid-proxy-init)

```
> voms-proxy-init -voms dteam
```

- sft tests syntax:

```
> cd /afs/cern.ch/project/gd/egee/sft-glite
> ./sftests -c sft-glite-workdir status
> ./sftests -c sft-glite-workdir cancel [ce-name]
> ./sftests -c sft-glite-workdir submit [ce-name]
> ./sftests -c sft-glite-workdir publish [ce-name]
```

Daily checks (PPS SFT Babysitting)

A short list of things to be checked daily (especially on Mondays), to make sure that everything is going fine.

Have a look at the display.

gLite : https://lcg-sft.cern.ch/sft-pps/lastreport.cgi

Go quickly through the status column and check the dates of the last change. Consider that normally for a site in green the last test should have run more or less an hour previously.

If you find something suspect have a look at the logs

The logs for the SFT suite are: /afs/cern.ch/project/gd/egee/gocdb-xfer/gocdb-xfer.log

/afs/cern.ch/project/gd/egee/sft-glite/sft-glite-cron.log

If the logs are not "normal" run a manual test

If you find something which seems not "normal", run a manual test. Please refer to the Usage section.

Quick rgma test

- Check the R-GMA status from the browser (https://lxb2093.cern.ch:8443/R-GMA/)

If the R-GMA browser does not work, quick rgma test on the UI

```
[aretico@lxb1908 sft-glite] rgma

Welcome to the R-GMA virtual database for Virtual Organisations.
================================================================

Your local R-GMA server is:

  https://lxb2093.cern.ch:8443/R-GMA

You are connected to the following R-GMA Registry services:

  https://pps-rgma-server.egee.cesga.es:8443/R-GMA/RegistryServlet

You are connected to the following R-GMA Schema service:

  https://pps-rgma-server.egee.cesga.es:8443/R-GMA/SchemaServlet

Type "help" for a list of commands.

rgma> show tables

+------------------------------------------+
| Table Name                               |
+------------------------------------------+
| bossJobExOutMessage                      |
| bossJobExOutStandardInfo                 |
| GlueCE                                   |

[ ... ]

| NetworkFileTransferThroughput            |
| GocNode_v0_4                             |
| GocSite_v0_4                             |
| LcgRecords                               |
+------------------------------------------+

rgma> describe GocNode_v0_4

+-----------------+--------------+-------------+-------------+
| Column name     | Type         | Primary key | Can be NULL |
+-----------------+--------------+-------------+-------------+
| nodeID          | INTEGER      | Yes         | No          |
| siteID          | INTEGER      | No          | Yes         |
| nodetype        | VARCHAR(50)  | No          | Yes         |
| nodetype2       | VARCHAR(50)  | No          | Yes         |
| hostname        | VARCHAR(50)  | No          | Yes         |
| domain          | VARCHAR(200) | No          | Yes         |
| ip              | VARCHAR(15)  | No          | Yes         |
| grp             | VARCHAR(20)  | No          | Yes         |
| hostdn          | VARCHAR(255) | No          | Yes         |
| monitor         | VARCHAR(1)   | No          | Yes         |
| MeasurementDate | DATE         | No          | No          |
| MeasurementTime | TIME         | No          | No          |
+-----------------+--------------+-------------+-------------+

rgma> select sitename from GocSite_v0_4 where type='PPS'

+------------------------+
| sitename               |
+------------------------+
| PPS-IFIC               |
| preprod.nikhef.nl      |
| PPS-CYFRONET           |
| PPS-SWITCH             |
| SCAI-PPS              |
| UKI-SOUTHGRID-BHAM-PPS |
| UKI-LT2-IC-HEP-PPS    |
| UKI-ScotGrid-Gla-PPS  |
| PPS-PADOVA            |
| Morpheus              |
| IN2P3-CC-PPS          |
| Taiwan-PPS            |
| CERN_PPS              |
| PPS-CNAF              |
| PPS-LIP               |
| CESGA-PPS             |
| FZK-PP                |
| PPS-PIC               |
| PreGR-02-UPATRAS      |
| EGEE-SEE-CERT         |
| UCM                   |
| PreGR-01-UoM          |
+-----------------------+
22 rows
rgma>
```

- If something is not working, restore R-GMA
  - Log as root on `lxb2093`
  - `> service tomcat5 restart`
  - `> service lcg-archiver restart`
  - `> service rgma-glue-archiver restart`
  - Log on `lxb1908` with your AFS account
  - Create a proxy (you should have already one)
  - `> /afs/cern.ch/project/gd/egee/gocdb-xfer/gocdb-xfer.py` (Downloads sites form the GOC db to R-GMA, it takes about 10 minutes)
  - re-run SFT tests (see Usage)

-- Main.aretico - 24 Oct 2005

This topic: LCG > PPSSFTInstallation
Topic revision: r26 - 2011-06-21 - AndresAeschlimann