Grid Log Retention Guidelines

Introduction

The current minimal Log Retention policy states that log information must be kept for at least 90 days. This page intends to details a sample implementation of this policy.

Job submission will normally progress from a User Interface (UI) machine, through a Resource Broker (RB) to a Computing Element (CE) and hence to the compute resource (usually a batch system). In some cases the RB is not used and the UI submits the job directly to the CE. Data access is through a Storage Element (SE) service and may be initiated directly from the UI or from a task executed on the compute resource.

Issues

In practice, it is necessary to:

- Gather centrally the log information from the distributed Grid systems
- Parse this information in order to extract job details
- Store job details and raw logs for at least 90 days

Many sites already use a central syslog service, which gathers log information that is exported from every node on the farm. Unfortunately, some middleware components **do not** use the system syslog facility. As a result, even if the entire syslog information from all systems is sent to a central facility, **it will not be possible to obtain all the information about a job**.

Objectives

We need first to gather centrally all the necessary Grid information from the nodes. Then, it is suggested that, for each job, the following information should be retained:

- About the resources:
  - CE name
  - User's identity on the CE (UID and DN)
  - UI who submitted the job (UI or RB)
  - RB name
  - User's identity on the RB (UID and DN)
  - Hosts contacted using GridFTP
- About the job management system:
  - EDG id
  - JM-Contact String
  - GATEKEEPER_JM_ID
  - GRAM_SCRIPT_JOB_ID
  - Condor ID
  - Myproxy server used
- About the job executed:
  - Name of the executable, including arguments
  - Job requirements
  - Submission time

With this information, it is possible to answer questions such as:

- What are the 10 last jobs executed on CE foo.bar?
- When have been executed the 5 last jobs from DN /C=CH/O=CERN/OU=GRID/CN=John Doe 0001?
- Where have been executed the 10 last jobs scheduled by RB foo.bar
- What executables have been run on CE foo.bar since yesterday?
- How many distinct certificates have been using myproxy server foo.bar (and list them)?
- What DNs have been mapped to UID 11111 on CE foo.bar in the past week?
- When and where did we first see this DN (from where and what did to do)?

Obtaining the log information from the nodes

1. Install a central server service. Configuration of such service is outside the scope of this document, it is a common, easy to install/maintain service.
2. On most of the nodes (WNs and CEs), the log information is sent to the local syslog facility. This means that from the CEs/WNs all the job information will be automatically sent to the central syslog service.
3. Retrieve the information from the RB:

Since it is very difficult to predict when all the middleware components will make use of the system syslog facility, a simple script called *lcg-logger* has been implemented to tail these stand alone Grid log files and inject them in the local system's syslog facility. It is used here to retrieve information from the RBs.

lcg-logger is available as an RPM from http://grid-deployment.web.cern.ch/grid-deployment/apt-cert/SECURITY/sl3/en/i386/RPMS.lcg_sl3/.

It is suggested that the node installs *lcg-logger* using a package management tool such as *apt* or *yum*. For instance, adding the following source to your *apt* configuration will enable your machine to install and update easily the package:

```
rpm http://grid-deployment.web.cern.ch/grid-deployment/apt-cert/ SECURITY/sl3/en/i386 lcg_sl3
```

Once the *lcg-logger* RPM is installed, it is necessary to edit */etc/lcg-logger.conf*. It is recommended to configure it as followed:

```
# lcg-logger configuration file

#syslog_flag = glite-

# The following file will be monitored
# YOU **MUST** RESTART LCG-logger FOR CHANGES TO BE APPLIED
# By default nothing is monitored, but here are a few suggestions:

log_file = /var/log/edg-fetch-crl-cron.log

# LSF
log_file = /var/log/lsfjobs.log

# RB
log_file = /var/edgwl/logmonitor/log/events.log
log_file = /var/edgwl/networkserver/log/events.log
log_file = /var/log/edg-wl-in.ftpd.log
log_file = /root/RB-sandbox-cleanup.log
log_file = /var/edgwl/logging/status.log
log_file = /var/edgwl/jobcontrol/log/events.log
log_file = /var/edgwl/workload_manager/log/events.log
log_file =  /mnt/raid/rb-state/opt/condor/var/condor/log/SchedLog
```

The *lcg-logger* service has then to be restarted:

```
/etc/init.d/lcg-logger restart
```

lcg-logger sends the information from the tailed log files into the *user.notice* syslog facility. It is recommended that you exclude this facility from you local machine to avoid log files redundancy on the local system.

This operation can be done using the following */etc/syslog.conf* log file:

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
kern.*                     /var/log/messages

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;user.!=notice;mail.none;authpriv.none;cron.none      /var/log/messages

# The authpriv file has restricted access.
authpriv.*                  /var/log/secure

# Log all the mail messages in one place.
mail.*                     -/var/log/maillog

# Log cron stuff
cron.*                     /var/log/cron

# Everybody gets emergency messages
*.emerg                     *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                  /var/log/spooler

# Save boot messages also to boot.log
local7.*                  /var/log/boot.log
# Additional config for LCG audit server
*.*                               @<your syslog server>
```

If for some reason you **only** want to export Grid log information, change the last line to

```
user.notice              @<your syslog server>
```

Note: *user.notice* is the default syslog facility used by lcg-logger. It is possible to specify another facility in */usr/sbin/logme*.

Once the Grid log information is tailed and sent the local syslog facility, it will automatically be exported to the syslog server specified in _/etc/syslog.conf.

As a result, the Grid log information should now be available on the central syslog server for the CEs, WNs and RBs. 4. Retrieve the information from external CEs:

In some environment, it might be inappropriate to retrieve the entire syslog information to extract Grid job details. In this case, information from the RBs can be obtained using the previous steps and by exporting **only** the syslog facility that is used by *lcg-logger*. For the CEs, the job information is stored using the main syslog facility, but also in the Gatekeeper log file. As a result, *lcg-logger* can be installed on the CEs, with the following configuration file:

```
# lcg-logger configuration file

#syslog_flag = glite-

# The following file will be monitored
# YOU **MUST** RESTART LCG-logger FOR CHANGES TO BE APPLIED
```

```
# By default nothing is monitored, but here are a few suggestions:

log_file = /var/log/globus-gridftp.log
log_file = /var/log/globus-gatekeeper.log
```

Then, exporting the only syslog facility used by *lcg-logger* will be sufficient to export Grid job information.

Extract the job information from the log files

On the central syslog server, it is assumed that the incoming log information is stored in /data/log/. Once the log files are received on the system, a filtering mechanism is enabled to filter this information and extract details about all the jobs.

This filtering mechanism is based on a package called *lcg-logs*.

lcg-logs is available as an RPM from http://grid-deployment.web.cern.ch/grid-deployment/apt-cert/SECURITY/sl3/en/i386/RPMS.lcg_sl3/.

It is suggested that the node installs *lcg-logs* using a package management tool such as *apt* or *yum*. For instance, adding the following source to your *apt* configuration will enable your machine to install and update easily the package:

```
rpm http://grid-deployment.web.cern.ch/grid-deployment/apt-cert/ SECURITY/sl3/en/i386 lcg_sl3
```

It is *strongly advised *to maintain *lcg-logs* up-to-date with the latest version available as any schema change in the middleware may require an update of *lcg-logs*.

To setup this activity, the following steps should be followed:

1. Install a MySQL server on the system:

```
# apt-get install mysql-server
```

2. Configure the */etc/my.cnf* in order to remove the network service of MySQL:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-networking
[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

3. Configure MySQL to start at boot:

```
# chkconfig --level 235 mysqld on
```

4. Change the MySQL root password:

```
# /usr/bin/mysql mysql
~ update mysql.user set Password=PASSWORD( strong_root_password ) where User= root
and Host= localhost ;
~ flush privileges;
~ quit;
```

5. Create an lcglogs database:

```
mysqladmin -u root -p create lcglogs
```

6. Create the lcg-logs user:

```
# /usr/bin/mysql -u root -p mysql
~ GRANT SELECT,INSERT,UPDATE,DELETE ON lcglogs.* TO  lcg-logs @ localhost
IDENTIFIED BY  mysql_password ;
~ GRANT SELECT ON lcglogs.* TO  lcg-logs-reader @ localhost
IDENTIFIED BY  mysql_password2 ;
~ flush privileges;
~ quit;
```

The choice of *mysql password* and *mysql password2* are up to the service manager. 7. Install the lcg-logs RPM. The package is available from the CERN CVS software repository and part of the autobuild.

```
# rpm -i lcg-logs-*.i386.rpm
# chkconfig --add lcg-logs
# chkconfig --level 235 lcg-logs on
```

8. Configure lcg-logs using */etc/lcg-logs.conf*, as:

```
# lcg-logs configuration file
# Location of all the log files
log_dir = /data/log/

# Location of the specific log files containing EDG id and CE information
log_file = /data/log/messages.log /data/log/daemon.log

# Output files for the job
output_file = /data/job_information.log

# MySQL User
mysql_db = lcglogs
mysql_user = lcg-logs
mysql_password = <mysql_password>
```

The choice of *mysql password* is up to the service manager.

9. Start the lcg-logs process:

```
# /etc/init.d/lcg-logs start
```

(Optional) Installing a Web interface

The Web interface is **entirely optional** to implement.

It is simply using phpMyAdmin, via Apache.

Due to the nature of the information, **under no circumstance the Apache server must be made available to the network, even for a short period of time**.

Access the Web server must be blocked at the firewall level and restricted in Apache s configuration.

The Web interface can be configured by:

1. Installing the relevant packages:

```
apt-get install php httpd php-mysql
```

2. Configuring */etc/httpd/conf/httpd.conf* so that Apache will only be listening on 127.0.0.1:

```
# cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.save
# sed -e  s/Listen 0.0.0.0/Listen 127.0.0.1/g  /etc/httpd/conf/httpd.conf.save |cat > /etc/httpd/
# /etc/init.d/httpd restart
```

3. Downloading phpMyAdmin from http://www.phpmyadmin.net

4. Copying the phpMyAdmin files to the Apache DocumentRoot (usually /var/www/html)

5. Configuring config.inc.php, with at least the following options:

```
$cfg[ PmaAbsoluteUri ] =  http://localhost:8080/ ;
$cfg[ Servers ][$i][ connect_type ] =  socket ;
$cfg[ Servers ][$i][ auth_type ] =  config ;
$cfg[ Servers ][$i][ user ] =  lcg-logs-reader ;
$cfg[ Servers ][$i][ password ] =  <mysql_password2> ;
```

Finally, access to the Web interface can be made available using SSH port forwarding. A remote, authorized user would simply need to:

1. Connect to lcg-audit with the appropriate SSH parameters (the localhost network interface must be enabled):

```
ssh -L 8080:localhost:80 <your syslog server>
```

2. Use a Web browser and go to http://localhost:8080/

Results

As shown in the screenshots attached to this page, once the job information is stored in the database, it is possible to search for jobs that match some specific criteria:

- **id** = Job ID specific to the database.
- **time** = Entry registration time
- **EDGiD** = Job EDG ID
- **JM1** - **JM2** = JM Contact String, minus the CE name and port
- **CE** = Name of the CE used by the job
- **CE_UID** = Local uid that has been assigned on the CE to the DN
- **CE_DN** = DN used by the user to access the CE
- **UI** = IP address of the service that submitted the job (could be a UI or an RB)
- **UI_FQDN** = Name of the service that submitted the job (could be a UI or an RB)
- **GATEKEEPER_JM_ID** = GATEKEEPER_JM_ID assigned to the job
- **GRAM_SCRIPT_JOB_ID** = GRAM_SCRIPT_JOB_ID assigned to the job
- **RB** = Name of the RB that handled the job
- **Executable** = Name of the executable run by the job
- **Job_Args** = Arguments specified with the executable
- **Job_Req** = Job requirements requested by the used
- **Submission_Time** = Job Submission Time
- **RB_UID** = Local uid that has been assigned on the RB to the DN
- **RB_DN** = DN used by the user to access the RB
- **My_Proxy** = Name of the MyProxy server used by the user
- **Target_CE** = Name and Job Manager type selected by the RB to submit the job
- **CondorID** = Condor ID assigned to the job

- **FTP_Dest** = Hosts that have been contacted using Grid FTP during the job submission process

While some data is probably still missing, these search fields should enable system administrators to answer a few basic questions.

-- Romain Wartel

This topic: LCG > LogRetention
Topic revision: r5 - 2006-03-14 - RomainWartel