Table of Contents

LCG Computing Element (LCG CE)

Functional description

LCG CE is a native computing resource access service with Globus Gatekeeper and GRAM protocol. LCG has modified some of its component to improve its performance. It is no longer recommended for new installations - sites should be using the CREAM Computing Element instead. No new development is taking place on the lcg-CE.

Daemons running

- globus-gatekeeper — must be started
- globus-gridftp — must be started
- globus-job-manager-marshal — must be started
- globus-gass-cache-marshal — should be started, but the client is able to work in fall-back mode with stopped daemon
- globus-gma — must be started if GLOBUS_GMA is enabled in site's config
- glite-lb-logd — should be started for L&B to work properly
- glite-lb-interlogd — should be started for L&B to work properly (this daemon ignores SIGTERM, use SIGKILL instead)

Init scripts and options (start|stop|restart|reload|status|...)

- globus-gatekeeper, globus-gridftp (start, stop, restart, status)
- gLite (start, stop, restart, status) *glite-lb-interlogd does not stop properly*
- globus-job-manager-marshal, globus-gass-cache-marshal, globus-gma (start, stop, restart, reload, status)

Configuration files location with example or template

- **/opt/globus/etc/globus-gass-cache-marshal.conf**, **/opt/globus/etc/globus-job-manager-marshal.conf**
  - **`logf`** (srting) — location of the log file (default is relative to GLOBUS_LOCATION)
  - **`dgaspath`** (string) — *[only for globus-job-manager-marshal]* path to DGAS directory (default is /opt/edg/var/gatekeeper/jobs/)
  - **`maxproc`** (numeric) — maximum number of parallel requests *[this is the most useful variable for tuning]* (5 by default)
  - **`rrobin`** (0, 1 or 2) — enables round-robin queue mode for users(1) or groups (2) (disabled (0) by default)
  - **`groups`** (0 or 1) — if set, supplementary groups will be applied to the job-manager processes (disabled (0) by default)
  - **`tick`** (numeric) — hung child processes are killed every this number of seconds (if no other events are happening) (300 by default).
  - **`reqtout`** (numeric) — client should send a complete request in this number of seconds after connection (10 by default)
  - **`proctout`** (numeric) — each request (child process) is allowed to run for this number of seconds (600 by default)
  - **`reqlimit`** (numeric) — maximum size of a request in bytes (16384 by default). One should increase this limit if environment is very large.
  - **`window`** (numeric) — data block for recv/send in bytes *[never change this]* (default value is 4096 (x86 page size)).

- **debug** (0, 1 or 2) — debug level. There are three of them: 0 — only warnings (default), 1 — all messages, 2 — stderr is being redirected to the log file (bad for log parsers, but good for catching problems in Perl jobmanagers)

- **/opt/globus/etc/globus-gma.conf**
  - **logf** (srting) — location of the log file (default is relative to GLOBUS_LOCATION)
  - **gridservices** (string) — path to the gridservices directory *[never change this]* (default is relative to GLOBUS_LOCATION)
  - **agentpath** (string) — path to the directory with agent files *[never change this]* (default is relative to GLOBUS_LOCATION)
  - **groups** (0 or 1) — if set, supplementary groups will be applied to the poll process (disabled (0) by default)
  - **condorfix** (0 or 1) — enables a Condor work-around for not distinguishing VOMS attributes (disabled (0) by default)
  - **tout** (numeric) — sets a limit in seconds for a single job state poll to finish (30 by default)
  - **toutlim** (numeric) — sets a limit for a number of consecutive poll timeouts for a given user, after which all remaining jobs for that user will be skipped till the next poll cycle (4 by default)
  - **tick** (numeric) — number of seconds between poll cycles.This parameter defines granularity for **stateage**, **fileage** and adaptive state refresh interval below (300 by default).
  - **stateage** (numeric) — number of seconds for which a job state is considered 'fresh' (600 by default)
  - **statefact** (numeric) — division factor for calculating adaptive state refresh interval for short jobs *[refresh_interval = min(stateage, job_run_time / statefact)]* (disabled (0) by default)
  - **fileage** (numeric) — number of seconds before a job file is considered 'stale' and gets removed (86400 by default)
  - **fileretry** (numeric) — number of retries to read a job file (2 by default)
  - **filesleep** (numeric) — delay in milliseconds between retries above (10 by default)
  - **debug** (0, 1 or 2) — debug level. There are three of them: 0 — only warnings (default), 1 — all messages, 2 — stderr is being redirected to the log file (bad for log parsers, but good for catching problems in Perl jobmanagers)

- **If your CE suffers form a very high load, try to decrease the `maxproc` parameters of globus-*-marshal. On the other hand if you have a CE with modern hardware, lots of CPUs and a very fast disk subsystem, consider increasing it.**
- **If your site is running short jobs consider decreasing the `tick` parameter to 60 and set `statefact` to a non-zero value (e.g. 3 or 4, check the formula above). In this case `stateage` may be increased up to 900 and more in order to keep batch system load low.**
- **All parameters (except debug 1 2) could be changed online (modify config file and send a HUP signal). All daemons create pidfiles in /var/run/ (not configurable).**

Logfile locations (and management) and other useful audit information

- /opt/globus/var/log/*.log — configurable with **logf** options above.
- /var/log/globus-gridftp.log
- /var/log/globus-gatekeeper.log
- /var/log/message
- /opt/edg/var/gatekeeper/

Open ports

- 2811 — Gridftp Server
- 2119 — Globus Gatekeeper
- 9002 — Locallogger Daemon
- Ports from $GLOBUS_TCP_PORT_RANGE should be open

Possible unit test of the service

Submitting jobs to it through both WMS and globus-job-run

Where is service state held (and can it be rebuilt)

Staged files are held under home directory of pool account

Job state files are in $GLOBUS_LOCATION/tmp/gram_job_state

Cron jobs

The cron jobs can be found in:

- /etc/cron.d/

and are:

- bdii-proxy
- edg-mkgridmap
- lcg-expiregridmapdir
- cleanup-grid-accounts
- edg-pbs-knownhosts
- cleanup-job-records
- edg-pbs-shostsequiv
- edg-apel-pbs-parser
- fetch-crl

Security information

Access control Mechanism description (authentication & authorization)

Authorisation is done with lcmaps and lcas. First its being checked whether or not a user is banned by verification of his or her certificate. If there is no ban, proxy data such as VO, group and role indicate the appropriate Unix group. With gridmapdir a freely available account is then beeing matched. In case there is no free account left, the mapping fails and so does the authorisation. It is planned for the future to introduce a central authorisation mechanism per site (SCAS). Then those steps will get executed on this host through a network request.

How to block/ban a user

- If it is necessary to ban a user on a CE, the following step:

- Add the user(s)'s DN into the "ban_users.db" file, which in default can be found at /opt/edg/etc/lcas/ or /opt/glite/etc/lcas/ if it is glite CE, as follow:
    - "User1's DN"
    - "User2's DN"
    - ... ...
    - "UserN's DN"

- If there are multiple DNs to be banned, each DN name should be in separated lines and must be quoted with the double quote mark (""), otherwise LCAS will not be able to block the user. At the moment, LCAS does not support wild mark, therefore you can not use "/C=UK/O=eScience/OU=CLRC/L=RAL/*" to ban a group of users. To verify that the user has indeed been banned, in the log there should be something like "LCAS failed authorization" if the job of the banned user landed on the CE.

- Nothing needs to be restarted

- If it is necessary to ban a VO reconfigure the service without that VO
    - Will also adapt the information system

Network Usage

Communication is taking place between the CE and RB/WMS, worker nodes, batch server and occasionally the UI.

Firewall configuration

(tcp) port 2119: globus-gatekeeper port 2170: ldap port 2811: globus-gridftp

port 20000 - 25000 (tcp + udp): gridftp-server

Security recommendations

If there are more than one computing elements at a site, that access the same worker nodes, it`s recommended to set up individual, unique accounts for all CEs (such as "user001" ... "user100" on CE1 and "user101" ... "user200" on CE2 and so on) or to centrally mount gridmapdir, in order to guarantee, that no user is able to read the data of any other users. Releasing accounts for, i.e., sharing them with other users should be an exception to the rule and should happen only then, if there's other way round. However, if this happens, it should be taken care of, that really all jobs of that account are finished. This is currently hardly to check, though. So after the starting time of the last job for that account you have to wait at least for the maximum time jobs are allowed to run on this computing element. If there is a big maintenance and naturally no job is running, the opportunity arises to clean the mapping in the gridmapdir.

See also EGEE'08 presentation.

Security incompatibilities

Nothing reported.

List of externals (packages are NOT maintained by Red Hat or by gLite)

Java (jdk, bouncycastle)

Other security relevant comments

- If you need to handle suspicious jobs, these the step tp follow:
  - Pause or stop the batch system queues
  - Suspend all active jobs, if the batch system supports it
  - Stop gatekeeper and gridftp-server while suspected DNs not yet identified
  - Ban suspected DNs or VO
  - Keep the active jobs submitted by the suspected accounts suspended if possible, to facilitate forensic investigations. Otherwise kill the jobs.
  - Follow the EGEE Incident Response Procedure: IncidentReporting

Utility scripts

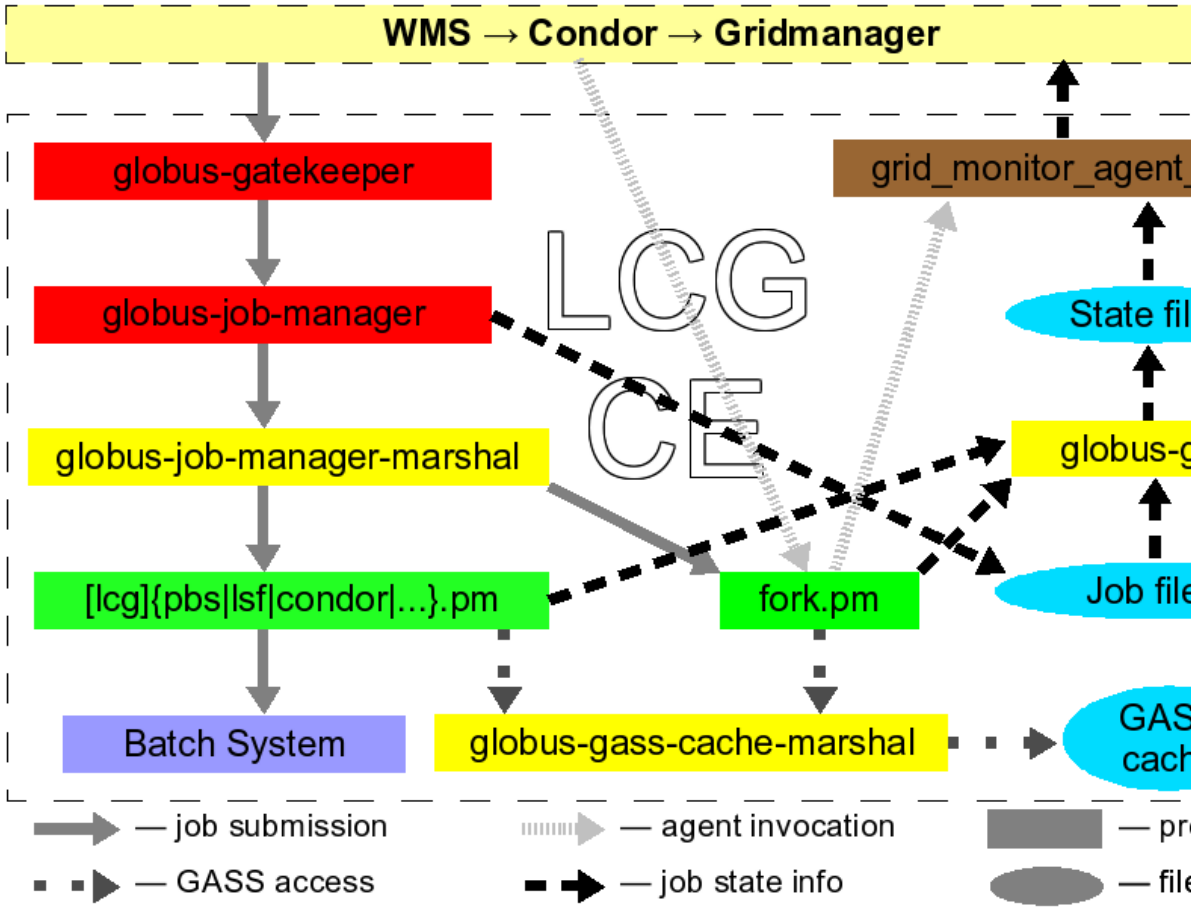
Nothing reported.

Location of reference documentation for users

- gLite 3.1 documentation

Location of reference documentation for administrators

- gLite 3.1 documentation
- Integration with BatchSystems
- LCG-CE Internals:



- On the image above: red boxes are Globus binaries, yellow boxes are Perl daemons from LCG, green boxes are Perl job-manager libraries.

Support Lifetime

As specified in SupportedServiceVersions, the normal support window for the lcg-CE is 6 months. Please check the linked page before reporting bugs.

Developer information

The lcg-CE is mostly deprecated now in favor of the glite-CREAM CE.

This topic: EGEE > LcgCE
Topic revision: r30 - 2011-03-25 - MaartenLitmaath