

## Managing a CREAM CE

*Alvise Dorigo (alvise.dorigo@pd.infn.it)*

*Sara Bertocco (sara.bertocco@pd.infn.it)*

*Luigi Zangrando (luigi.zangrando@pd.infn.it)*

**EGEE'07**

**Budapest, October 1-5, 2007**

[www.eu-egEE.org](http://www.eu-egEE.org)



Information Society  
and Media



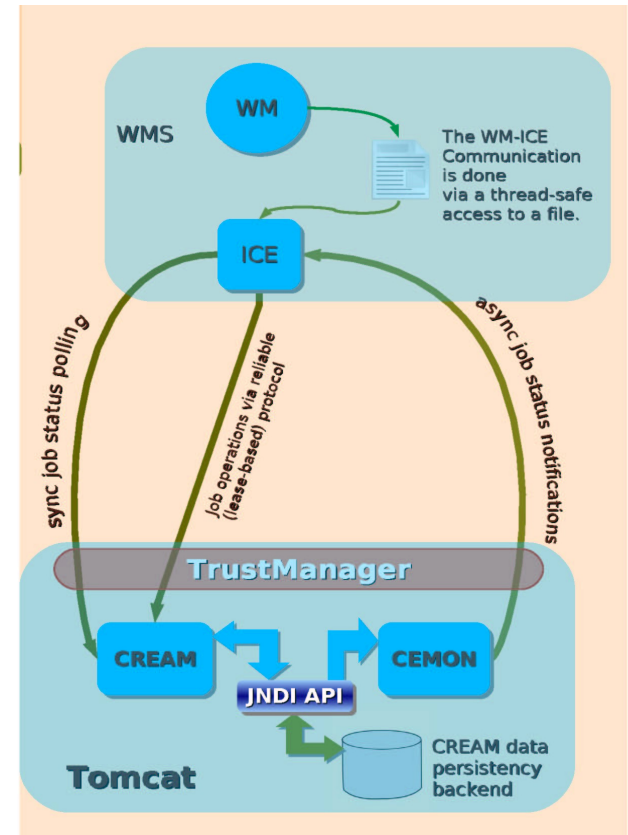
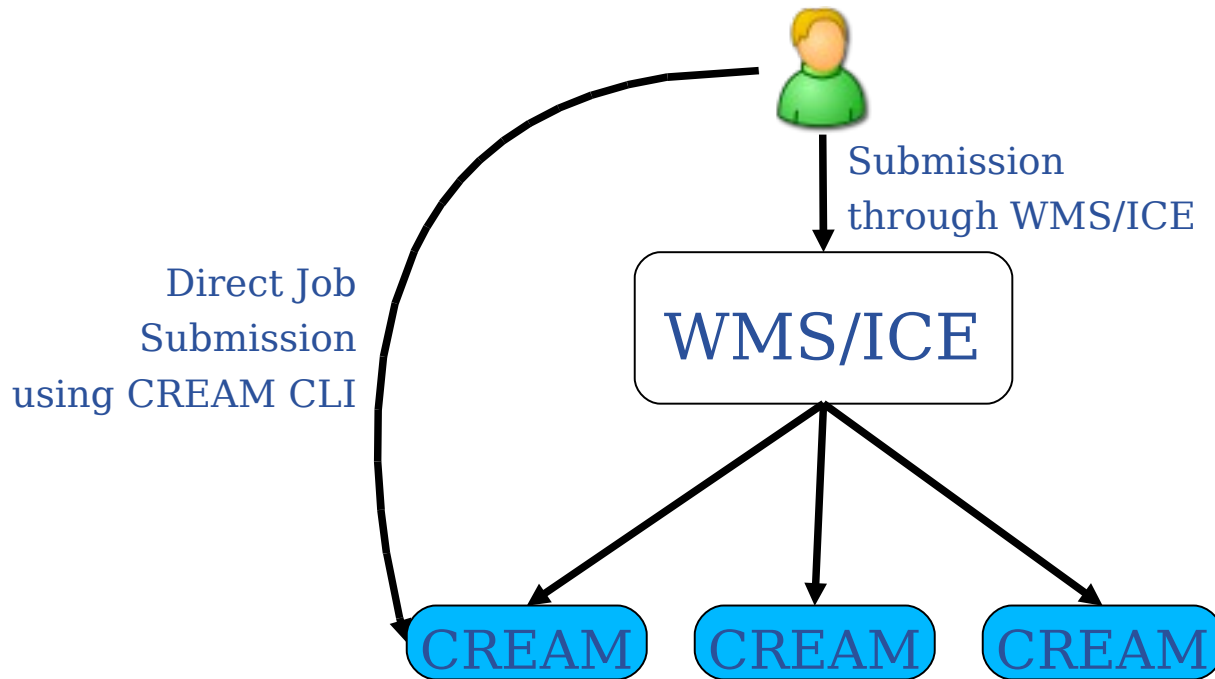
*CREAM (Computing Resource Execution and Management) service is a system for job management and access to computational resources at Computing Element (CE) level developed for gLite by INFN (Padova, Italy).*

- *We consider a computing resource (defined as a Computing Element) typically as a cluster of computational nodes managed by a Local Resource Management System (LRMS)*
  - **LSF, PBS/Torque, Condor, etc.**

- **There are two kind of CREAM's users:**
  - simple user:
    - able to operates only to its jobs
  - administrator
    - can handle the jobs of any users
    - it is enabled to perform management tasks
      - *e.g. enable/disable acceptance of new job submission*

CREAM should be invoked:

- by the Workload Management System (WMS), via the ICE (Interface to CREAM Environment) service
- by a generic client, e.g. an end-user willing to directly submit jobs to a CREAM CE. A C++/Java CLIs are available



## □ Job submission

- Submission of jobs to a CREAM based CE
- Support for direct staging of input sandboxes files (split in two operations: job register and job start)
- Job characteristics described via JDL (Job Description Language) expression
  - CREAM JDL is basically the same JDL used by the EGEE gLite Workload Management System with CREAM-specific extensions
- Supported job types
  - Simple jobs
  - MPI jobs
  - Support of bulk jobs in progress (DAG jobs, parametric jobs, job collections)

## □ Proxy delegation

- Possibility to automatically delegate a proxy for each job submission
- Possibility to use a previously delegated proxy for multiple job submissions
  - recommended approach wrt performance, since proxy delegation can be “expensive”

## □ Job status

- Retrieval of job status and other info (e.g. job creation/submission/start/execution/completion time, worker node, failure reason, etc.) of submitted jobs

## □ Job list

- To get the identifiers of all your jobs (CREAM Job IDs)

## □ Job cancellation

- To cancel previously submitted jobs

## ❑ **Job suspension and resume**

- To hold and then restart jobs

## ❑ **Job lease**

- Mechanism for cancelling all jobs for which the lease time is expired

## ❑ **Accounting**

- For each job, proper information is logged in a log file which is then “managed” by the EGEE accounting service (done by blah)

## □ Disabling of new job submissions

- Can be done only by the CE administrator
- useful if the CE has to be shut-down for maintenance
- Other operations still allowed (job cancel, job info, etc.)
- two ways:
  - explicit: operations `enableAcceptJobSubmissions / disableAcceptJobSubmissions`
    - Specific command line provided for this operation
  - implicit: it is also possible to define **policies** (in classad format) on waiting/pending/running jobs to disable new job submissions
    - e.g. disable new submissions if the number of active jobs is  $> 10000$
    - useful to avoid the CE overloading
    - policy cyclically executed by CREAM (default: every 10 minutes)



- **Job purge**

- To clear a terminated job from a CREAM based CE (that means removing all persistent info about the job and its ISB/OSB)
- A CREAM job is purged only when the user explicitly invokes the purge method.
- At any rate the administrator can purge old jobs that have been forgotten on the CREAM CE host. For this purpose a specific tool is provided and is installed with the CREAM RPM.
  - **this program could be used regularly, e.g. via a cron job**

- The job purger script (*/opt/glite/bin/glite-cream-purger.sh*) requires a configuration filename as argument; this file has the following structure:

```
cream_scratch_dir=<dir>  
provider_factory=<value>  
<status1>=<time1>  
<status2>=<time2>  
...
```

- The values for `cream_scratch_dir` and `provider_factory` must match the values of the attributes with the same name in the CREAM configuration file.
- `<status>` represents one of the possible CREAM job states.
- `<time>` represents a time period (e.g. 2 days, 1month, etc): the keywords `days`, `months` and `years` (whose meaning doesn't need to be explained) can be used.
- The job purger will purge jobs:
  - \* which are in status `DONE-OK` since more than 2 days
  - \* which are in status `ABORTED` since more than 1 month
  - \* ...

- The CEMon service is a general purpose notification framework responsible for providing information about the CE asynchronously. Users/services can also synchronously get information from it.
  - publish/subscribe model (kind of info, notif. frequency, duration, etc.)
  - extensible and pluggable sensors based architecture
    - 3 levels of visibility of the published info (user, group, all)
  - main sensors available:
    - CESensor: The “default information” the CEMon service is able to provide is related to the CE itself: this information is represented according to the Glue Schema (1.1 and 1.2 version)
    - GridICE sensor: publishing GridICE information
    - OSG sensor: publishing information useful for OSG business
    - CREAM\_JOBS sensor: publishing CREAM job status information

## Administration tasks:

- set-up the visibility of the published information
- add/remove sensors (hot pluggable)
- add/remove predefined subscriptions (restart of service not needed)

- *ig-yaim installation scripts are available*
  - *used for testing and development environment*
- *almost ready for integration with the official glite-yaim*
  - *waiting for glite-yaim cvs module*

- **Detailed description available at INFN GRID wiki page**

<http://igrelease.forge.cnaf.infn.it/doku.php?id=doc:guides:install-cream31-cert>

- **Summarizing, 4 steps:**
  - **Download and install glite-yaim-core and ig-yaim**
  - **Setup properly variables in file site-info.def**
  - **Installation:**
    - `/opt/glite/yaim/bin/ig-yaim -i -s <site-info.def> -m <cream>`
  - **Configuration:**
    - `/opt/glite/yaim/bin/ig-yaim -c -s <site-info.def> -n <cream>`

- *CREAM and CEMon configuration based on xml files*
  - ❑ */opt/glite/etc/glite-ce-cream/cream\_config.xml*
  - ❑ */opt/glite/etc/glite-ce-monitor/cemonitor\_config.xml*

*The administrator can fine tune the CREAM parameters like:*

- ❑ *max number of internal threads*
- ❑ *max value of the internal cache size*
- ❑ *scratch and proxy dirs, sandbox area, etc*
- ❑ *classad policy for disabling the job submission*
- ❑ *the LRMS connector (the default is based BLAH)*
- ❑ *...*

```

<service id="main-service"
  description="cream service"
  cream_scratch_dir="/opt/glite/var/cream"
  cream_sandbox_dir="/opt/glite/var/cream_sandbox"
  lrms_connectors_dir="/var/lib/tomcat5/webapps/ce-cream/connectors"
  cream_journal_num_threads="50"
  glxec_path="/usr/share/tomcat5/glexec-wrapper.sh"
  cream_cache_size="400"
  disable_submission_policy="false"

```

```

....
/>

```

- Possibility to define a set of administrators adding their DN in the `/etc/grid-security/admin-list` file

Example:

```
# /etc/grid-security/admin-list
"/C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Massimo
Sgaravatto/Email=massimo.sgaravatto@pd.infn.it" .egee

"/C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Paolo
Andreetto/Email=paolo.andreetto@pd.infn.it" .egee
```

- *Log files to check*
  - **CREAM specific:**
    - */var/log/tomcat5/glite-ce-cream.log (log4j)*
    - */var/log/tomcat5/glite-ce-cemon.log (log4j)*
    - **not yet in /opt/glite/var/log standard location (very soon)**
    - **new common logging format adoption (soon)**
  - **trustmanager**
    - */var/log/tomcat5/glite-security-trustmanager.log (log4j)*
  - **Tomcat:**
    - */var/log/tomcat5/catalina.out*
  - **Blparser:**
    - */var/log/glite-blparser.log*
  - **glexec:**
    - */var/log/glexec/glexec\_log*
    - */var/log/glexec/glexec\_lcas\_lcmaps.log*

- *Log rotation is granted by:*
  - **Logging mechanisms adopted (e.g. log4j)**
  - **Cron jobs defined during the configuration scripts by yaim procedure**
- *Log tuning:*
  - *verbosity levels tunable via log4j configuration*
  - *log4j enables logging*
    - *on SYSLOG file*
    - *a remote SYSLOG daemon (by log4j's SyslogAppender)*



- **CREAM port list:**
  - 8443 (CREAM service)
  - 9090 internal port for receiving notifications from jobWrapper about the job status change
- **CEMon port list:**
  - 8443 (CEMon service)
- **blParser:**
  - pbs/torque
    - GLITE\_CE\_BLPARSERPBS\_PORT1=33332
    - GLITE\_CE\_BLPARSERPBS\_CREAMPORT1=56565
  - lsf
    - GLITE\_CE\_BLPARSERLSF\_PORT1=33333
    - GLITE\_CE\_BLPARSERLSF\_CREAMPORT1=56566
- **gridFTP standard port: 2811**