

Table of Contents

The Workload Management System Admin Guide.....	1
Introduction.....	1
Service Overview.....	1
Installation and configuration.....	1
Hardware Requirements.....	1
Install & Configure.....	1
Daemons and services running.....	1
File Systems/Directories.....	2
Log files locations.....	2
Configuration Guide.....	3
Tuning of some configuration parameters.....	3
Select only specific VO resources.....	4
Cron Jobs.....	4
Troubleshooting.....	4
WMS Monitoring.....	4
Service Monitoring Guide.....	4

The Workload Management System Admin Guide

Introduction

Service Overview

The Workload Management System (WMS) comprises a set of Grid middleware components responsible for the distribution and management of tasks across Grid resources, in such a way that applications are conveniently, efficiently and effectively executed. Following the list of sub-services the WMS is composed of:

- **Workload Management – WM:** Core component of the Workload Management, its purpose is to accept and satisfy requests for job management coming from its clients
- **WMProxy:** Web service interface to submit jobs to the WM
- **Job Controller – JC:** Acts as an interface to condor for the WM
- **Log Monitor – LM:** Directly connected to JC acts as a job monitoring tool parsing condor log files
- **Local Logger:** copy events to be sent to the LB server into a local disk file
- **LBProxy:** keeps a local view of the job state to be sent to the LB server
- **Proxy Renewal:** Service to renew the proxy of a long-running job
- **ICE:** (Interface to CREAM Environment) is the WMS service dealing when interacting with CREAM based CEs.

Installation and configuration

Hardware Requirements

- 4 GB RAM is the minimum suggested for memory
- quad-core processor is recommended to better handle parallel matchmaking and all the different sub-services running on a WMS
- Min disk space: depends on load and type of jobs submitted.
 - ◆ under '`{GLITE_LOCATION_VAR}/sandboxdir`' 30-40GB is the min on several wms used in production with the cron job to purge job sandboxes once a week enabled in order to accomodate submitted job sandbox dirs.

Install & Configure

- Install and configure OS and basic services according to the <https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide310>
- Install the glite-WMS metapackage from the appropriate gLite software repository
- Configure the WMS node by running `'/opt/glite/yaim/bin/yaim -c -s site-info.def -n WMS'`
Following a list of the WMS specific variables that can be set in the '`site-info.def`' file:
 - ◆ `$WMS_HOST` -> the WMS hostname, ex. : 'egee-rb-01.\$MY_DOMAIN'
 - ◆ `$PX_HOST` -> the hostname of a server myproxy, ex.: 'myproxy.\$MY_DOMAIN'
 - ◆ `$BDII_HOST` -> the hostname of the site bdii to be used, ex: 'sitebdii.\$MY_DOMAIN'
 - ◆ `$LB_HOST` -> the hostname of the LB server to be used, ex:
'lb-server.\$MY_DOMAIN:9000' This variable is set as a service specific variable in the file `services/glite-wms`, located one directory below the one where the '`site-info.def`' file is located

Daemons and services running

Scripts to check the daemons status and to start/stop are located in the `{GLITE_WMS_LOCATION}/etc/init.d/` directory (i.e. `{GLITE_WMS_LOCATION}/etc/init.f/glite-wms-wm start/stop/status`). Glite production installation also

provide a more generic service, called gLite, to manage all of them simultaneously, try service gLite status/start/stop On a typical WMS node the following services must be running:

- **glite-lb-locallogger:**

```
glite-lb-logd running
glite-lb-interlogd running
```

- **glite-lb-proxy:**

```
glite-lb-proxy running as 4137
```

- **glite-proxy-renewald:**

```
glite-proxy-renewd running
```

- **globus-gridftp:**

```
globus-gridftp-server (pid 3107) is running...
```

- **glite-wms-jc:**

```
JobController running in pid: 10008
CondorG master running in pid: 10063 10062
CondorG schedd running in pid: 10070
```

- **glite-wms-lm:**

```
Logmonitor running...
```

- **glite-wms-wm:**

```
/opt/glite/bin/glite-wms-workload_manager (pid 9957) is running...
```

- **glite-wms-wmproxy:**

```
WMPProxy httpd listening on port 7443
httpd (pid 22223 22222 22221 22220 22219 22218 22217) is running ....
===
WMPProxy Server running instances:
UID      PID  PPID  C  STIME TTY          TIME CMD
```

- **glite-wms-ice:**

```
/opt/glite/bin/glite-wms-ice-safe (pid 10103) is running...
```

File Systems/Directories

- **\${GLITE_LOCATION_VAR}/sandboxdir:** it is where the job sandboxes are located, they are automatically purged when a get-output of a job is done. If a VO does not take care of getting the job output back from the WMS node SB dir can become a serious problem for the HD occupancy. Another situation in which the SB dir can become problematic is when a job (or a certain number of jobs) has a huge output sandbox and the control on the OSB is not enabled on the glite_wms.conf file (see WMPProxy configuration parameter section). In any case it is a good habit to purge periodically the Sandbox Dir.
- **\${GLITE_LOCATION_VAR}/workloadmanager/:** it is where the input file list or the jobdir are located depending on the value of 'DispatcherType' attribute in the 'glite_wms.conf' file
- **\${GLITE_LOCATION_VAR}/logmonitor/:** this directory contains mainly condor log files
- **\${GLITE_LOCATION_VAR}/jocontrol/**

Log files locations

Log files are located under \${GLITE_LOCATION_LOG}, typically being '/var/log/glite'. This directory on a heavily used WMS can become quite big, on the order of tens of GB. Old rotated log files should be manually removed. Following the default log files that can be found on a typical WMS:

- **wmproxy.log** Used in case of authentication or submission error
- **workload_manager_events.log** Used to check the status of the matchmaking process (from Waiting to Ready status) and the query to the information system to fill in the InformationSuperMarket
- **ice.log** used to check jobs that matched a CREAM based CE and are sent to it via ICE
- **jobcontoller_events.log** Used to check the jobs events once arrived on condor
- **httpd-wmproxy-errors.log** Used in case of problems in contacting the WMPProxy service
- **httpd-wmproxy-access.log**
- **logmonitor_events.log** Aggregate information about each job coming from various log files
- **glite-wms-wmproxy-purge-proxycache.log**
- **lcmaps.log** Used when there are problems in the mapping of remote users to local pool accounts

Other log files that can be useful in case of trouble are the condor log in:

- **/var/local/condor/log/**
- **/var/glite/logmonitor/CondorG.log/**

Configuration Guide

The general configuration file for the WMS is located in `${GLITE_LOCATION}/etc/glite_wms.conf`. This file is organized in sections, one for every running service plus a Common section. For a general description of the `glite_wms.conf` configuration file, the configuration parameters and their default values see here: <https://twiki.cnaf.infn.it/cgi-bin/twiki/view/EgeeJra1It/WMSConfFile>

Other configuration files useful to know for troubleshooting:

- `${GLITE_LOCATION}/etc/glite_wms_wmproxy.gacl`: this file is used for authorization purposes in order to check for requesting user rights.
- `${GLITE_LOCATION}/etc/glite_wms_wmproxy_httpd.conf`: this file is a WMPProxy specific configuration file configuring the HTTP daemon and Fast CGI
- `{GLITE_LOCATION}/etc/wmproxy_logrotate.conf`: this file configures the logrotate tool, that performs the rotation of `httpd-wmproxy-access` and `httpd-wmproxy-errors` HTTPD daemon log files.
- `${GLITE_LOCATION_VAR}/.drain`: this file is used to put the WMS in draining mode, so that it does not accept new submission request but allows any other operation as the output retrieval.

Tuning of some configuration parameters

- **II_timeout**: default by yaim is set to 30, increase it for low-memory machines. 4 GB is the minimum suggested for memory
- **MatchRetryPeriod**: once a job becomes pending, meaning that there are no resources available, this parameter represents the period between successive match-making attempts, in seconds. The default value is '1000', in order to decrease the number of periodic retries of unmatched jobs the value of this parameter should be increased. A suggested value (used on several production wms) is several hours, '14400'
- **RuntimeMalloc**: in the WM section, allows to use an alternative malloc library (i.e. nedmalloc, google performance tools and many more), run-time redirecting with LD_PRELOAD. Possible values are, for example, `RuntimeMalloc? = "/usr/lib/libtcmalloc_minimal.so"` if you use Google malloc.

Select only specific VO resources

Sometimes it could be usefull forcing the WMS to select only resources specific to given VO. This would obviously reduce the matchmaking time and can be achieved providing an additional ldap clause which will be added in the search filter at purchasing time. The default search filter used is:

```
( | (objectclass=gluecesebind) (objectclass=gluecluster) (objectclass=gluesubcluster) )
```

The idea is to supply system administrators with the possibility to specify an additional ldap clause which will be added in logical AND to the latest two clauses of the default filter in order to match gluece/gluevoview objectclasses specific attributes. To such an aim the configuration file supply users with:

- `IsmIILDAPCEFilterExt` for handling the additional search filter while purchasing information about CE from the BDII

As an example, by specifying the following:

```
IsmIILDAPCEFilterExt =
" ( | (GlueCEAccessControlBaseRule=VO:cms) (GlueCEAccessControlBaseRule=VOMS:/cms/*
```

the search filter during the purchasing would be:

```
( | (objectclass=gluecesebind) (objectclass=gluecluster) (objectclass=gluesubcluster)
(& ( | (objectclass=gluevoview) (objectclass=gluece) )
( | (GlueCEAccessControlBaseRule=VO:cms) (GlueCEAccessControlBaseRule=VOMS:/cms/*) )
```

and thus the WMS would select only resources (i.e. CE/Views) belonging to CMS.

Cron Jobs

Several cron jobs are installed by yaim:

- `/etc/cron.d/glite-wms-purger.cron`: it periodically purges the sandbox dirs performing a check on the job status
- `glite-wms-wmproxy-purge-proxycache.cron`: expired proxies are purged by this cron job
- `fetch-crl`: cron job that retrieves the CRLs periodically

Troubleshooting

Some common error messages and troubleshooting operations that can be performed on a WMSLB instance are described here

WMS Monitoring

A tool to monitor the glite-WMSLB instances is available, it has been developed and currently maintained by INFN. For an extensive description of the tool go here

Service Monitoring Guide

-- ElisabettaMolinari - 11 Apr 2008

This topic: EgeeJra1It > WMS_guide

Topic revision: r11 - 2010-03-05 - ElisabettaMolinari



Copyright © 2008-2023 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback