

Table of Contents

The WMS configuration file.....	1
Common configuration.....	1
WorkloadManager configuration.....	1
JobController configuration.....	3
LogMonitor configuration.....	4
NetworkServer configuration.....	5
WorkloadManagerProxy configuration.....	6
ICE configuration.....	8

The WMS configuration file

The behavior of most of the processes running on the WMS node is driven by parameters set in a common configuration file, usually `/opt/glite/etc/glite_wms.conf`. Currently the syntax is based on the ClassAd language. The parameter names are case insensitive.

The file contains multiple sections, one per service plus a common one:

```
[
  Common = [...];
  JobController = [...];
  LogMonitor = [...];
  NetworkServer = [...];
  WorkloadManager = [...];
  WorkloadManagerProxy = [...];
  ICE = [...]
]
```

The value of a parameter can be expressed in terms of environment variables, with the typical UNIX shell syntax: a \$ sign followed by the name of the variable in brackets (e.g. `${HOME}`).

The following paragraphs describe the parameters available for each component. If a default value exists, this is shown in parenthesis at the end of the description.

Common configuration

HostProxyFile ()

the host proxy certificate file

LogFile

the name of the file where messages are logged

DGUser

the user under which a WMS process runs

The parameters in the Common section may be overridden in the specific component sections. This usually happens with LogFile and LogLevel.

WorkloadManager configuration

The attributes available in this section, in alphabetical order, are:

PropagateToLRMS ()

JDL attributes to be propagated to the batch system through CREAM and BLAH \$
SbRetryDifferentProtocols (false)

BrokerLib

the library implementing the brokering functionality. What is specified here is loaded with dlopen().
("libglite_wms_helper_broker_ism.so")

CeForwardParameters

the parameters forwarded by the WM to the CE

CeMonitorAsynchPort

the port used to listen to notification arriving from CEMon's. A value of -1 means that listening is disabled (-1)

CeMonitorServices

the list of CEMon's the WM listens to

DisablePurchasingFromGris

??? (false)

- DispatcherType**
the WM can read its input using different mechanisms. Currently supported types are "filelist" and "jobdir" ("filelist")
- DliServiceName**
??? ("data-location-interface")
- EnableBulkMM**
specifies if bulk matchmaking, i.e. matching multiple similar jobs in a collection in one shot, should be applied (false)
- EnablePurchasingFromRgma**
purchase CE information from RGMA (false)
- EnableRecovery**
specifies if at startup the WM should perform a special recovery procedure for the requests that it finds already in its input. This parameter should always be set to true (false)
- EnableStatusCheck**
specifies if, for each request the WM reads from input, it should check the status of the request (e.g. for a submit the only acceptable status is WAITING). As for the recovery, this check is not very reliable, so it should be disabled (false)
- ExpiryPeriod**
the maximum time, expressed in seconds, a submitted job is kept in the overall system, from the time it arrives for the first time at the WM (86400, i.e. one day)
- Input**
the input source of new requests. If DispatcherType is "filelist" the source is a file; if DispatcherType is "jobdir" the source is the base dir of a JobDir structure, which is supposed to be already in place when the WM starts. A JobDir structure consists of a base dir under which lie other three subdirectories, named tmp, new, old ("\${EDG_WL_TMP}/workload_manager/input.fl")
- IsmBlackList**
a list of CEs that have to be excluded in the ISM
- IsmCEMonAsynchPurchasingRate**
??? (30)
- IsmCEMonPurchasingRate**
??? (120)
- IsmDump**
if the ISM dump is enabled, the dump, in ClassAd format, will be written to this file. In order to avoid file corruptions, the contents of a dump are built in a temporary file, whose name is the same value of this parameter with the prefix ".tmp", which only at the end of the operation is renamed to the specified file ("\${GLITE_WMS_TMP}/workload_manager/ismdump.fl" - but it's not in filelist format!)
- IsmIiPurchasingRate**
the period between two ISM purchases from the BDII, in seconds (240)
- IsmRgmaPurchasingRate**
the period between two ISM purchases from RGMA, in seconds (120)
- IsmThreads**
(def=true) All the threads related to the ISM management are taken from the thread pool (false) or created separately (false)
- IsmUpdateRate**
the period between two updates of the ISM. The ISM updater is not the purchaser thread, it only checks for expired entries. The default value is too short (50)
- JobWrapperTemplateDir**
the job wrapper sent to the CE and then executed on Worker Node is based on a bash template which is augmented by the WM with job-specific information. This is the location where all the templates - one at the moment - are stored ("\${GLITE_WMS_LOCATION}/etc/templates")
- LogFile**
the name of the file where messages are logged
- LogLevel**

each logging statement in the code specifies a log level. If that level is less than or equal to LogLevel the message is actually logged, otherwise it is ignored. The levels go from 1 (minimum verbosity) to 6 (maximum verbosity)

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 (1000)

MaxOutputSandboxSize

the maximum size of the output sandbox, in bytes. The limit is currently enforced by the job wrapper running on the Worker Node, which doesn't upload more data than what specified here. If the value is -1 there is no limit. Currently the mechanism doesn't work well, so it is suggested to set this parameter to -1 (100000000)

MaxRetryCount

the system limit to the number of deep resubmissions for a job. The actual limit is the minimum between this value and the one specified in the job description (10)

MaxShallowRetryCount

the system limit to the number of shallow resubmissions for a job. The actual limit is the minimum between this value and the one specified in the job description (10)

QueueSize

(def=1000) Size of the queue of events "ready" to be managed by the workers thread pool

RgmaConsumerLifeCycle

??? (30)

RgmaConsumerTtl

??? (300)

RgmaQueryTimeout

??? (30)

RuntimeMalloc

(def="") allows to the use an alternative malloc library (examples are nedmalloc, google performance tools, ccmalloc), specifying the path to the shared object, to be loaded with LD_PRELOAD. Example: RuntimeMalloc = "/usr/lib/libtcmalloc_minimal.so".

SiServiceName

??? ("org.glite.SEIndex")

TokenFile

the shallow resubmission mechanism works by removing an empty file on the gridftp server running on the WSM machine from the job wrapper running on a Worker Node. This parameter specifies the name of that file ("token.txt")

WorkerThreads

the number of request handler threads (10)

JobController configuration

The attributes available in this section, divided by significant groups are:

LogFile

String attribute containing the path of the JobController log file

LogLevel

Log verbosity level. If that level is less than or equal to LogLevel the message is actually logged, otherwise it is ignored. The level goes from 1 (minimum verbosity) to 6 (maximum verbosity)

LogRotationMaxFileNumber

Log files that are rotated this numnber of times before being removed (5)

LogFileMaxSize

Log files are rotated when they grow bigger then this size bytes (100000000)

LogRotationBaseFile

Base name of the rotated log files (LogFile value)

LockFile

Path of the lock file for the service.

CondorSubmit

Path of the "condor_submit" command

CondorRemove

Path of the "condor_remove" command

CondorDagman

Path of the "condor_dagman" command

DagmanMaxPre

Sets the maximum number of PRE scripts within the DAG that may be running at one time; it is the "-maxpre" parameter of the condor_dagman command. (10)

DagmanLogLevel

Log verbosity level for Dagman; it is the "-Debug" parameter of the condor_dagman command. The level goes from 0 (minimum verbosity) to 7 (maximum verbosity)

DagmanLogRotate

It is the value for the environment variable used by condor: _CONDOR_MAX_DAGMAN_LOG

MaximumTimeAllowedForCondorMatch

Sets the number of seconds that a job can wait in the condor queue to be matched before to be resubmitted. (900)

ContainerRefreshThreshold

Number of jobs that JobController can take in memory before to resynchronize its container with the one physically saved in the file "IdRepositoryName" (see LM section). (1000)

InputType

The JobController can read its input using different mechanisms. Currently supported types are "filelist" and "jobdir". ("filelist")

Input

The input source of new requests. If InputType is "filelist" the source is a file; if InputType is "jobdir" the source is the base dir of a JobDir structure, which is supposed to be already in place when the JobController starts. A JobDir structure consists of a base dir under which lie other three subdirectories, named tmp, new, old.

SubmitFileDir

Path of the directory where the submit files for condor are stored.

OutputFileDir

Path of the directory where the standard error/output files of the jobs (e.g. the JobWrapper) are stored.

UseFakeForProxy

This parameter tells to JobController don't use a real Proxy client but a fake one; used only for testing.

UseFakeForReal

This parameter tells to JobController don't use a real Client but a fake one; used only for testing.

LogMonitor configuration

The attributes available in this section, divided by significant groups are:

LogFile

String attribute containing the path of the JobController log file

LogLevel

Log verbosity level. If that level is less than or equal to LogLevel the message is actually logged, otherwise it is ignored. The level goes from 1 (minimum verbosity) to 6 (maximum verbosity)

LogRotationMaxFileNumber

Log files that are rotated this number of times before being removed (5)

LogFileMaxSize

Log files are rotated when they grow bigger then this size bytes (100000000)

LogRotationBaseFile

Base name of the rotated log files (LogFile value)

LockFile

Path of the lock file for the service.

CondorLogDir

Path of the directory where LogMonitor stores the CondorG log files.

CondorLogRecycleDir

Path of the directory where old CondorG log files are saved.

JobsPerCondorLog

Max number of job logged in the same CondorG log file. (1000)

GlobusDownTimeout

Log monitor waits this number of seconds before to consider a condor job, which has lose contact with the CE, failed (and so resubmits it if possible). (600)

AbortedJobsTimeout

A condor job could stay in the "removed" state forever (state "X"), so after this number of seconds from the previous remove command Logmonitor sends to condor a new condor_remove if the job is again in the condor queue. (600)

ForceCancellationRetries

Number of retries to remove a job from the condor queue before to forget it definitively. LogMonitor wait AbortedJobsTimeout seconds between two tries. (10)

MainLoopDuration

LogMonitor loops between the CondorLog files every this number of seconds. (3)

MonitorInternalDir

Path of the directory where LogMonitor stores its own files.

IdRepositoryName

Name of the file contains informations about the jobs used by LogMonitor and JobController.

ExternalLogFile

Path of the directory where extra log files are stored.

UseMaradonaFile

The user job exit status is written into an extra "Maradona" file that is copied to the WMS with globus-url-copy as a sort of backup. (True)

MaradonaTransportProtocol

The transport protocol used to transfer the "Maradona" file. ("gsiftp")

RemoveJobFiles

If sets to true all files used to submit jobs to condor are removed when they are no more necessary. Set it to "false" only for debug purpose. Files are stored in the SubmitFileDir directory as it is set in the JobController section. (True)

ContainersCompactThreshold

Number of jobs stored in the private container before to compact it to improve performances. (1000)

NetworkServer configuration

II_Contact, II_Port, II_Timeout

TOP BDII hostname, port and connection timeout

MaxInputSandboxSize

Maximum dimension of Input Sanbdox

EnableQuotaManagement

enable Sandbox quotas per user

WorkloadManagerProxy configuration

The attributes available in this section, divided by significant groups are:

Logging:

LogFile

String attribute containing the path of the WMPProxy log file (Optional)

LogLevel

Integer attribute containing a value from 0 to 6 (Optional). The integer value represents the WMPProxy log file verbosity level: from 0 (fatal) to 6 (debug: maximum verbosity)

Sandbox:

SandboxStagingPath

Root directory where job sandboxes are stored. It **MUST** be in the form: /, where DocumentRoot is set as inside glite_wms_wmproxy_httpd.conf configuration file. The directory **MUST** be accessible by the user under which WMPProxy is running (usually it is the "glite" user). The user running WMPProxy is determined by the value of the environment variable GLITE_USER, if not differently set with User directive inside glite_wms_wmproxy_httpd.conf configuration file

MaxInputSandboxSize

Maximum number of bytes for input sandboxes on a per-job basis (Optional - default value is 10000000). This attribute, even if optional, **SHOULD** be properly set (if quota are not set for users on the WMS node) according to the storage capacity of WMS node in order to avoid filling up of the WMS disk. **NOTE:** this value is a per job one.

List Match:

\$ ListMatchRootPath - Directory path where temporary pipes for list-match operations are created. The directory **MUST** be accessible by the user under which WMPProxy is running (usually it is the "glite" user). The user running WMPProxy is determined by the value of the environment variable GLITE_USER, if not differently set with User directive inside glite_wms_wmproxy_httpd.conf configuration file (Optional - default value is /tmp)

Ports:

HTTPSPort

The HTTPS port where the WMPProxy service is listening

GridFTPPort

Port number where gridFTP server is listening (Optional - default value is gridFTP standard port 2811)

DefaultProtocol

The protocol used for input sandbox file transferring. Currently supported protocols are gsiftp and https. (Optional - default value is gsiftp).

Perusal:

MinPerusalTimeInterval

Integer value representing the time interval (in seconds) between two savings of job partial execution output. This attribute affects the WMPProxy and other componets behaviour only if perusal functionality are explicitly requested by the user via the JDL, see EnableFilePerusal JDL attribute (Optional - default value is 10 seconds)

LB:

LBProxy

Boolean attribute to switch from LB and LBProxy. If the value of this attribute is true, LBProxy is used by WMPProxy for logging and query operations about jobs (Optional - default value is true)

LBServer

Address or list of addresses of the LB Server[s] to be used for storing job's information in the format of [:] (default value for port is 9000). This attribute is needed only if LB Server is not running in the WMPProxy server host, or if more than one LB Servers must be used. Selection of the LB Server to use is made randomly from the list by the WMPProxy, for any different service request. WMPProxy maintains a list of weights associated to the available LB Servers so that failing LB Servers have decreasing probability of being selected. If the Service Discovery is enabled, the LB Servers found using the Service Discovery are added in the list.

Note that the following lines have same meaning:

```
LBServer = "ghemon.cnaf.infn.it:9000"; LBServer = {"ghemon.cnaf.infn.it:9000"};
```

WeightsCacheValidity

Time in seconds (n) indicating the validity of the weights (i.e. probability to be selected) associated to the available LB Servers. When last weights update (i.e. last received request) has occurred more than n seconds ago then the weights are restored to the same value for all LB Servers (Optional - default value is 21600 seconds)

WeightsCachePath

Location of the directory on the WMPProxy node where the LB Servers weights file is stored (Optional - default value is directory /var/glite/wmproxy)

LBLocalLogger

Address of LB Local Logger in the format of [:] (default value for port is 9002). This attribute is needed only if LB Local Logger runs on another host and LBProxy is not enabled

Job Start Options:

AsyncJobStart

Boolean attribute used to switch from synchronous/asynchronous job start behavior. When set to true, during job start operation the control is returned to user immediately after the request has been received, while the actual execution of the operation (that could be quite time consuming) is performed asynchronously

Requirements:

WMSRequirements

This attribute contains an expression to be appended to the user JDL requirements coming from the UI

Service Discovery:

EnableServiceDiscovery

Boolean attribute to enable Service Discovery. If the value of this attribute is true, the Service Discovery is enabled, i.e. WMPProxy invokes Service Discovery for finding available LB Servers

ServiceDiscoveryInfoValidity

Time in seconds (n) indicating the validity of the information provided by the Service Discovery. A call to Service Discovery for updated information is done every n seconds.

LBServiceDiscoveryType

Type key for LB Servers to be discovered by Service Discovery (Optional - default value is org.glite.lb.server)

Served Requests:

MaxServedRequests

Long attribute limiting the number of operation served by each WMPProxy instance before exiting and releasing possibly allocated memory. This value is overridden by `GLITE_WMS_WM_PROXY_MAX_SERVED_REQUESTS` environment variable, if set. This feature can be disabled by setting a lower-or-equal to zero value. (Optional - default value is 100 requests, minimum allowed value is 40)

Load Scripts:

OperationLoadScripts

ClassAd type attribute where an internal attribute can be specified for any WMPProxy provided operation. The names of these attributes are equal to the names of the server operations (e.g. for jobSubmit operation the attribute name to use is "jobSubmit"). This internal attributes are used to provide the path and the name of the script to be executed to verify the load of the WMPProxy server for any provided operation. If the server load is too high the requested operation is refused. The path and the name of the script can be followed by user defined options and parameters depending on the specific script needs for arguments.

WMPProxy provide a load script that can be used for any of the provided operations. The template load script `glite_wms_wmproxy_load_monitor.template` is installed by the rpm file `glite-wms-wmproxy` in the directory `${GLITE_LOCATION}/sbin`.

To call the script `glite_wms_wmproxy_load_monitor`, when the operation `jobSubmit` is requested, with the options:

```
--load1 10 --load5 10 --load15 10 --memusage 95 --diskusage 95 --fdnum 500
```

add the attribute:

```
OperationLoadScripts [
  jobSubmit = "${GLITE_LOCATION}/sbin/glite_wms_wmproxy_load_monitor
    --oper jobSubmit --load1 10 --load5 10 --load15 10
    --memusage 95 --diskusage 95 --fdnum 500";
]
```

Any kind of load script file can be used. If a user custom script is used, the only rule to follow is that the script exit value must be 0 in the case the operation can continue the execution, 1 in the opposite case (operation refused - Server load too high).

The script files must be executable and must have the proper access permissions.

ICE configuration

The parameters for ICE are available in the ICE Configuration Guide

-- FrancescoGiacomini - 29 Oct 2007

This topic: EgeeJra1It > WMSConfFile

Topic revision: r15 - 2011-11-15 - MarcoCecchi



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