

Table of Contents

Glue2 support in CREAM.....	1
1 Introduction.....	2
2 Target scenario.....	3
2.1 CREAM CE in no cluster mode.....	3
2.2 CREAM CE in cluster mode.....	3
2.3 Objectclasses.....	4
2.3.1 ComputingService.....	4
2.3.2 ComputingEndpoint.....	4
2.3.2.1 Policy for the ComputingEndPoint.....	5
2.3.3 ComputingShare.....	5
2.3.3.1 MappingPolicy for the ComputingShare.....	7
2.3.4 ComputingManager.....	7
2.3.5 Benchmark.....	7
2.3.6 ExecutionEnvironment.....	8
2.3.7 ApplicationEnvironment.....	10
2.3.8 ApplicationHandle.....	10
2.3.9 ComputingActivity.....	10
2.3.10 ToStorageService.....	10
3 Batch system dynamic information.....	12
3.1 Current Glue 1 scenario.....	12
3.1.1 Torque/PBS.....	12
3.1.2 LSF.....	12
3.1.3 SGE.....	12
3.1.4 Generic dynamic scheduler.....	12
3.2 Work to be done to support Glue2 publication.....	13
3.2.1 Work to be done in the PBS/Torque information provider (done).....	13
3.2.2 Work to be done in the LSF information provider (done).....	13
3.2.3 Work to be done in the SGE information provider (done).....	14
3.2.4 Work to be done in the generic dynamic scheduler (done).....	14
4 Relevant RFCs.....	15
5 Testbed.....	16
6 Raw notes (for Massimo use).....	17

Glue2 support in CREAM

CREAM CE provided with EMI-1 provides some initial support for glue2, that has been finalized in EMI-2

1 Introduction

The Glue 2.0 specification document is available here .

The Glue 2.0 specification can also be found here .

The GLUE v. 2.0 – Reference Implementation of an LDAP Schema document can be found here .

Glue 2.0 schema in SVN

2 Target scenario

2.1 CREAM CE in no cluster mode

We assume that a CREAM CE is configured in cluster mode if that is not the only CREAM CE available in the site. Sites with multiple CREAM CEs (submitting to the same batch system) should **always** have a cluster node and therefore be configured in cluster mode.

If a CREAM CE is configured in no cluster mode, all the Glue2 object classes are published by the resource BDII running on the CREAM CE.

These objectclasses are:

- ComputingService (done)
- ComputingEndPoint (done)
 - ◆ AccessPolicy for ComputingEndPoint (done)
- ComputingManager (done)
- ComputingShare (done)
 - ◆ MappingPolicy for ComputingShare (done)
- MappingPolicy (done)
- ExecutionEnvironment (done)
- Benchmark (done)
- ToStorageService (done)
- ApplicationEnvironment (done)
- EndPoint for RTEPublihser (done)
 - ◆ "Child" of ComputingService
- EndPoint for CEMon (done)
 - ◆ "Child" of ComputingService
 - ◆ Published only if CEMon is deployed

2.2 CREAM CE in cluster mode

Sites with multiple CREAM CEs (submitting to the same batch system) should **always** have a cluster node and therefore be configured in cluster mode.

If a CREAM CE is configured in cluster mode:

- The resource BDII running on the CREAM CE publishes just the following objectclasses:
 - ◆ ComputingEndpoint (done)
 - ◇ AccessPolicy for ComputingEndpoint (done)
 - ◆ EndPoint for CEMon (done)
 - ◇ "Child" of ComputingService
 - ◇ Published only if CEMon is deployed
- All the other objectclasses are published by the resource BDII running on the gLite-CLUSTER node:
 - ◆ ComputingService (done)
 - ◆ ComputingManager (done)
 - ◆ ComputingShare (done)
 - ◇ MappingPolicy for ComputingShare (done)
 - ◆ ExecutionEnvironment (done)
 - ◆ Benchmark (done)
 - ◆ ToStorageService (done)
 - ◆ ApplicationEnvironment (done)
 - ◆ EndPoint for RTEPublihser (done)

◇ "Child" of ComputingService

Actually it is necessary to publish also the ComputingService, since otherwise the resource bdi of the CREAM CE will not be able to publish the ComputingEndpoint

- The ServiceId is the the one specified by the yaim variable `COMPUTING_SERVICE_ID` which is such scenario is mandatory. This variable must have the same value in all the relevant nodes (in the cluster node and in all the CREAM CEs)

2.3 Objectclasses

This section reports the most significant information concerning the implementation of the Glue 2 objectclasses wrt the CREAM CE

2.3.1 ComputingService

- `GLUE2ServiceID` is the value `ComputingServiceId` in `/etc/glite-ce-glue2/glite-ce-glue2.conf`. This value is the one specified by the yaim variable `COMPUTING_SERVICE_ID`, if specified (this variable is mandatory in cluster mode). Otherwise it is `#{CE_HOST}_ComputingElement`
- `GLUE2EntityCreationTime` is the timestamp when the ldif file was created
- `GLUE2EntityName` is "Computing Service on <host>"
- `GLUE2EntityOtherInfo` includes information concerning the info provider
- `GLUE2ServiceCapability` is `executionmanagement.jobexecution`
- `GLUE2ServiceType` is `org.glite.ce.CREAM`
- `GLUE2ServiceQualityLevel` is `production`
- `GLUE2ServiceComplexity` indicates the number of endpoints, the number of shares, and the number of resources
 - ◆ For a CREAM CE in no-cluster mode: endpoints (2 or 3: one for CREAM, one for the RTEPublisher and one for CEMon, if deployed),
 - ◆ For a gLite-cluster TODO: endpoints should consider the total number of CREAM endpoints (how to find this ?) the total number of CEMon (how to find this ???) and the RTEPublisher
- `GLUE2ServiceAdminDomainForeignKey` is the value of `SiteId` in `/etc/glite-ce-glue2/glite-ce-glue2.conf`. This value is the one specified by the yaim variable `SITE_NAME`

2.3.2 ComputingEndpoint

- `GLUE2EndpointID` is `<hostname> + "_org.glite.ce.CREAM"`
- `GLUE2EntityCreationTime` is the timestamp when the ldif file was created. This value is then overwritten by the dynamic plugin
- `Glue2EndpointStartTime` is the timestamp when the ldif file was created. This value is then overwritten by the dynamic plugin
- `GLUE2EntityName` is the `EndPointId`
- `GLUE2EntityOtherInfo` includes the host DN, the EMI middleware version, information if the CREAM CE is using Argus, and information concerning the info provider
- `GLUE2EndpointURL` is the endpoint URL of the CREAM CE, that is: `"https://" + <host> + "8443/ce-cream/services"`
- `GLUE2EndpointCapability` is `executionmanagement.jobexecution`
- `GLUE2EndpointTechnology` is `webservice`
- `GLUE2EndpointInterfaceName` is `org.glite.ce.CREAM`
- `GLUE2EndpointInterfaceVersion` is read from the CREAM configuration file (attribute `cream_interface_version`)

- GLUE2EndpointWSDL is got from the service itself, i.e. "https://" + <host> + ":8443/ce-cream/services/CREAM2?wsdl";
- GLUE2EndpointSupportedProfile is `http://www.ws-i.org/Profiles/BasicProfile-1.0.html`
- GLUE2EndpointSemantics is the link to the CREAM user guide
- GLUE2EndpointImplementor is `gLite`
- GLUE2EndpointImplementationName is `CREAM`
- GLUE2EndpointImplementationVersion is read from the CREAM configuration file (attribute `cream_service_version`)
- GLUE2EndpointQualityLevel=`s` is `=production`
- GLUE2EndpointHealthState is unknown in the ldif static file; it is overwritten by the `glite-ce-glue2-endpoint-dynamic` plugin (which check the `glite-info-service-test` output and the status of the tomcat service)
- GLUE2EndpointHealthStateInfo is N/A in the ldif static file; it is overwritten by the `glite-ce-glue2-endpoint-dynamic` plugin (which check the `glite-info-service-test` output)
- GLUE2EndpointServingState is a value provided by a dynamic plugin; it is checked if submissions are disabled (by the limiter, or explicitly by the admin). If so, it is published `draining` (see also <http://savannah.cern.ch/bugs/?69854>). Otherwise a static value is used. It is read from `/etc/glite-ce-glue2/glite-ce-glue2.conf` (attribute `ServingState`), which is initially filled by yaim considering the yaim variable `CREAM_CE_STATE`. The admin can edit this file if he wants to publish a specific value without reconfiguring.
- GLUE2EndpointIssuerCA is found with a "openssl x509 -issuer -noout -in" on the host certificate
- GLUE2EndpointTrustedCA is `IGTF`
- GLUE2EndpointDowntimeInfo is "See the GOC DB for downtimes: <https://goc.egi.eu>"
- GLUE2ComputingEndpointStaging is `staginginout`
- GLUE2ComputingEndpointJobDescription is `glite:jdl`
- GLUE2EndpointServiceForeignKey is the value `ComputingServiceId` in `/etc/glite-ce-glue2/glite-ce-glue2.conf`. This value is the one specified by the yaim variable `COMPUTING_SERVICE_ID`, if specified (this variable is mandatory in cluster mode). Otherwise it is `=${CE_HOST}_ComputingElement`
- GLUE2ComputingEndpointComputingServiceForeignKey is the same as `GLUE2EndpointServiceForeignKey`

2.3.2.1 Policy for the ComputingEndPoint

For each ComputingEndPoint, there are as many as policies as the policy rules that must be defined

- GLUE2EntityCreationTime is the timestamp when the ldif file was created
- "GLUE2EntityName is "Access control rules for Endpoint EndPointId"
- GLUE2EntityOtherInfo includes information concerning the info provider
- GLUE2PolicyScheme is `org.glite.standard`
- GLUE2PolicyRule is an element of ACBR in `/etc/glite-ce-glue2/glite-ce-glue2.conf` (e.g. `VO:cms`). It is `ALL` if there are no policies
- GLUE2PolicyUserDomainForeignKey is an element of Owner in `/etc/glite-ce-glue2/glite-ce-glue2.conf` (e.g. `cms`)
- GLUE2AccessPolicyEndpointForeignKey is the `EndpointId` (<hostname> + "_org.glite.ce.CREAM")

2.3.3 ComputingShare

A ComputingShare corresponds to a Glue1 VOView.

If needed, besides the VOViews we will also represent batch system queues as ComputingShares (this will have some impact on the WMS matchmaker)

- GLUE2ShareID is concatenation of queue name + owner + ServiceId
- GLUE2EntityCreationTime is the timestamp when the Ldif file was created
- GLUE2EntityOtherInfo includes the CEId(s) (read from the `/etc/glite-ce-glue2/glite-ce-glue2.conf` = conf file) and information concerning the info provider
- GLUE2ShareDescription is "Share of <queuename> for <VO>"
- GLUE2ComputingShareMappingQueue is the batch system queue name. It is read from the `/etc/glite-ce-glue2/glite-ce-glue2.conf` = conf file)
- GLUE2ComputingShareMaxWallTime is 999999999 in the Ldif static file; it is supposed to be overwritten by the batch system specific dynamic plugin
- GLUE2ComputingShareMaxCPUTime is 999999999 in the Ldif static file; it is supposed to be overwritten by the batch system specific dynamic plugin
- GLUE2ComputingShareMaxRunningJobs is 999999999 in the Ldif static file; it is supposed to be overwritten by the batch system specific dynamic plugin
- GLUE2ComputingShareServingState is production in the Ldif static file; it is supposed to be overwritten by the batch system specific dynamic plugin
- GLUE2ComputingShareTotalJobs is 0 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ComputingShareRunningJobs is 0 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ComputingShareWaitingJobs is 444444 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ComputingShareEstimatedAverageWaitingTime is 2146660842 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ComputingShareEstimatedWorstWaitingTime is 2146660842 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ComputingShareFreeSlots is 0 in the Ldif static file; it is supposed to be overwritten by the generic dynamic scheduler plugin
- GLUE2ShareResourceForeignKey
 - ◆ for a CREAM CE in no-cluster mode this is the link to the the first (and unique) element of the attribute ExecutionEnvironments in `/etc/glite-ce-glue2/glite-ce-glue2.conf` (which is the hostname of the CREAM-CE)
 - ◆ for a gLite cluster: this is the link to the first ExecutionEnvironment (i.e. first element of ExecutionEnvironments in `/etc/glite-ce-glue2/glite-ce-glue2.conf`)
- GLUE2ComputingShareExecutionEnvironmentForeignKey the same as GLUE2ShareResourceForeignKey
- GLUE2ShareEndpointForeignKey is the link to the EndPoints. The value is read from the `/etc/glite-ce-glue2/glite-ce-glue2.conf`.
 - ◆ For a CREAM CE in no cluster mode this is the hostname + `_org.glite.ce.CREAM`
 - ◆ For a gLite cluster this is the list of hostname + `_org.glite.ce.CREAM` considering which cluster is associated to that queue, and which CEs are defined for that cluster
- GLUE2ComputingShareComputingEndpointForeignKey is the same as GLUE2ComputingShareComputingEndpointForeignKey
- GLUE2ShareServiceForeignKey is the value ComputingServiceId in `/etc/glite-ce-glue2/glite-ce-glue2.conf`. This value is the one specified by the yaim variable `COMPUTING_SERVICE_ID`, if specified (this variable is mandatory in cluster mode). Otherwise it is `=${CE_HOST}_ComputingElement`
- GLUE2ComputingShareComputingServiceForeignKey is the same as GLUE2ShareServiceForeignKey

2.3.3.1 MappingPolicy for the ComputingShare

- GLUE2PolicyID is ComputingShareId plus "_policy"
- GLUE2EntityCreationTime is the timestamp when the ldif file was created
- GLUE2PolicyScheme is org.glite.standard
- GLUE2PolicyRule is the list of ACBRs for this share (read from /etc/glite-ce-glue2/glite-ce-glue2.conf)
- GLUE2PolicyUserDomainForeignKey is the owner (read from /etc/glite-ce-glue2/glite-ce-glue2.conf)
- GLUE2MappingPolicyShareForeignKey is the ComputingShareId

2.3.4 ComputingManager

- GLUE2ManagerID is ServiceId + "_Manager"
- GLUE2EntityCreationTime is the timestamp when the ldif file was created
- GLUE2EntityName is: "Computing Manager on <host>"
- GLUE2EntityOtherInfo includes information concerning the info provider
- GLUE2ManagerProductName is the value CE_BATCH_SYS in /etc/glite-ce-glue2/glite-ce-glue2.conf. This value is the one specified by the yaim variable CE_BATCH_SYS
- GLUE2ManagerProductVersion in the ldif file is the value BATCH_VERSION in /etc/glite-ce-glue2/glite-ce-glue2.conf. This value is the one specified by the yaim variable BATCH_VERSION. It is supposed to be overwritten by the batch system specific dynamic plugin

2.3.5 Benchmark

For each ExecutionEnvironment, a Benchmark objectclass is created for each benchmark that must be represented.

In /etc/glite-ce-glue2/glite-ce-glue2.conf (filled by yaim) the following is defined:

```
ExecutionEnvironment_<ExecutionEnvironmentId>_Benchmarks = (Benchmark1, Benchmark2, ..., BenchmarkN)
```

where the format of Benchmark_i is: (Type Value)

This is then used to produce the ldif file with the Benchmark objectclasses.

The benchmark that are now represented are:

- specfp2000 (using the yaim variable CE_SF00)
- specint2000 (using the yaim variable CE_SIO)
- HEP-SPEC06 (if the yaim variable CE_OTHERDESCR reports the value for this benchmark)
- GLUE2BenchmarkID is the concatenation of ResourceId and type of benchmark
- GLUE2EntityCreationTime is the timestamp when the ldif file was created
- Glue2EntityName is "Benchmark" + the type of bechmark
- GLUE2EntityOtherInfo includes information concerning the info provider
- GLUE2BenchmarkType is the type of benchmark (specfp2000, specint2000, ..)
- GLUE2BenchmarkValue is the value for that benchmark

2.3.6 ExecutionEnvironment

For a CREAM CE configured in no cluster mode there is a single ExecutionEnvironment.

For a gLite-Cluster there is one ExecutionEnvironment for each subcluster. However please note that, because of the current behavior in the WMS matchmaking, a ComputingShare can refer to a single ExecutionEnvironment. If there are multiple ExecutionEnvironments, the first one is chosen for such association.

- `GLUE2ResourceID`
 - ◆ for a CREAM CE in no-cluster mode, this is the first (and unique) element of the attribute `ExecutionEnvironments` in `/etc/glite-ce-glue2/glite-ce-glue2.conf` (which is the hostname of the CREAM-CE)
 - ◆ for gLite-cluster this is an element of the attribute `ExecutionEnvironments` in `/etc/glite-ce-glue2/glite-ce-glue2.conf`, Each element of `ExecutionEnvironments` is a subcluster-identifier
- `GLUE2EntityCreationTime` is the timestamp when the ldif file was created
- `GLUE2EntityName` is the `ResourceID`
- `GLUE2EntityOtherInfo` includes information concerning
 - ◆ the `smpsize`: this is read from `config_cream_gip_glue2` (attribute `ExecutionEnvironment_xxx_SmpSize`)
 - ◇ For a CREAM CE in no-cluster mode this is the value of the yaim variable `CE_SMP_SIZE`
 - ◇ for a gLite-cluster this is the value of the yaim variable `SUBCLUSTER_<subcluster-identifier>_HOST_ArchitectureSMPSize`
 - ◆ the number of cores: this is read from `config_cream_gip_glue2` (attribute `ExecutionEnvironment_xxx_Cores`)
 - ◇ For a CREAM CE in no cluster mode this is taken from the yaim variable `CE_OTHERDESCR`
 - ◇ For a gLite-cluster this is taken from the yaim variable `SUBCLUSTER_xxx_HOST_ProcessorOtherDescription`
 - ◆ the info provider
- `GLUE2ExecutionEnvironmentPlatform` is read from `config_cream_gip_glue2` (attribute `ExecutionEnvironment_xxx_ArchitecturePlatformType`)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable `CE_OS_ARCH`
 - ◆ For a gLite-cluster this is taken from the yaim variable `SUBCLUSTER_xxx_HOST_ArchitecturePlatformType`
- `GLUE2ExecutionEnvironmentTotalInstances` is `GLUE2ExecutionEnvironmentLogicalCPUs` (see below) divide `SmpSize` (published in `GLUE2EntityOtherInfo`, see above)
- `GLUE2ExecutionEnvironmentPhysicalCPUs` is read from `config_cream_gip_glue2` (attribute `ExecutionEnvironment_xxx_PhysicalCPUs`)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable `CE_PHYSCPU`
 - ◆ For a gLite-cluster mode this is the value of the yaim variable `SUBCLUSTER_xxx_SUBCLUSTER_PhysicalCPUs`
- `GLUE2ExecutionEnvironmentLogicalCPUs` is read from `config_cream_gip_glue2` (attribute `ExecutionEnvironment_xxx_LogicalCPUs`)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable `CE_LOGCPU`
 - ◆ For a gLite-cluster mode this is the value of the yaim variable `SUBCLUSTER_xxx_SUBCLUSTER_LogicalCPUs`
- `GLUE2ExecutionEnvironmentCPUMultiplicity` is: `+ "cpu" + "-" + "core"`
 - ◆ is "single" if (`SmpSize == (LogicalCPUs / PhysicalCPUs)`). It is "multi" otherwise
 - ◆ is "single" if (`PhysicalCPUs == LogicalCPUs`). It is "multi" otherwise

CreamGlue2 < CREAM < TWiki

- GLUE2ExecutionEnvironmentCPUVendor is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_ProcessorVendor)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_CPU_VENDOR
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_ProcessorVendor
- GLUE2ExecutionEnvironmentCPUModel is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_ProcessorModel)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_CPU_MODEL
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_ProcessorModel
- GLUE2ExecutionEnvironmentCPUClockSpeed is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_ProcessorClockSpeed)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_CPU_SPEED
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_ProcessorClockSpeed
- GLUE2ExecutionEnvironmentMainMemorySize is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_MainMemoryRAMSize)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_MINPHYSMEM
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_MainMemoryRAMSize
- GLUE2ExecutionEnvironmentVirtualMemorySize is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_MainMemoryVirtualSize)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_MINVIRTMEM
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_MainMemoryVirtualSize
- GLUE2ExecutionEnvironmentOSFamily is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_OperatingSystemFamily)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_OS_FAMILY
 - ◆ For a gLite-cluster mode this is the value of the yaim variable CE_OS_FAMILY
- GLUE2ExecutionEnvironmentOSName is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_OperatingSystemName)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_OS
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_OperatingSystemName
- GLUE2ExecutionEnvironmentOperatingSystemRelease is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_OperatingSystemRelease)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_OS_RELEASE
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_OperatingSystemRelease
- GLUE2ExecutionEnvironmentConnectivityIn is read from config_cream_gip_glue2 (attribute ExecutionEnvironment_xxx_NetworkAdapterInboundIP)
 - ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable CE_INBOUNDIP
 - ◆ For a gLite-cluster mode this is the value of the yaim variable SUBCLUSTER_xxx_HOST_NetworkAdapterInboundIP)
- GLUE2ExecutionEnvironmentConnectivityOut is read from

config_cream_gip_glue2 (attribute
 ExecutionEnvironment_XXX_NetworkAdapterOutboundIP)

- ◆ For a CREAM CE in no-cluster mode this is the value of the yaim variable
 CE_OUTBOUNDIP
- ◆ For a gLite-cluster mode this is the value of the yaim variable
 SUBCLUSTER_XXX_HOST_NetworkAdapterOutboundIP
- GLUE2ResourceManagerForeignKey is the ManagerId (i.e. "ServiceID" + "_Manager")
- GLUE2ExecutionEnvironmentComputingManagerForeignKey is the same as
 GLUE2ResourceManagerForeignKey

2.3.7 ApplicationEnvironment

For each ExecutionEnvironment, a single objectclass is created for each software tag of that ExecutionEnvironment. These software tags are the ones defined during configuration (yaim variables CE_RUNTIMEENV (for no cluster mode) and SUBCLUSTER_XXX_HOST_ApplicationSoftwareRunTimeEnvironment for gLite cluster) plus the ones published by VO admins in /opt/edg/var/info and /opt/glite/var/info.

These are published using a gip provider.

- GLUE2ApplicationEnvironmentId is concatenation of the name of the software tag and the ExecutionEnvironment ID
- GLUE2EntityCreationTime is the timestamp when the provider script is run
- GLUE2EntityValidity is 3600
- GLUE2EntityOtherInfo includes information concerning the info provider
- GLUE2ApplicationEnvironmentAppName is the name of the software tag
- GLUE2ApplicationEnvironmentComputingManagerForeignKey is the name of the ComputingManager

2.3.8 ApplicationHandle

We don't implement the ApplicationHandle objectclass

2.3.9 ComputingActivity

We don't implement the ComputingActivity objectclass, since we don't publish information regarding jobs

2.3.10 ToStorageService

There is a ToStorageService objectclass for each SE close to the considered CE

In the configuration file config_cream_gip_glue2 this is represented by the attribute CloseSEs which has the following format:

```
# Format: CloseSEs = (closeSE1, closeSE2, ..., closeSEn)
# Format of closeSEi: (StorageServiceid LocalPath RemotePath)
```

- Each StorageServiceid is an element of the yaim variable SE_LIST
- For LocalPath and RemotePath the values of SE_MOUNT_INFO_LIST are used
- GLUE2ToStorageServiceID is the concatenation of ServiceId and StorageServiceId (the latter is read from config_cream_gip_glue: first attribute of a CloseSEs)
- GLUE2EntityCreationTime is the timestamp when the ldif file was created
- Glue2EntityName is the same as GLUE2ToStorageServiceID

CreamGlue2 < CREAM < TWiki

- `GLUE2EntityOtherInfo` includes information concerning the info provider
- `GLUE2ToStorageServiceLocalPath` is read from `config_cream_gip_glue` (second attribute of a `CloseSEs`)
- `GLUE2ToStorageServiceRemotePath` is read from `config_cream_gip_glue` (third attribute of a `CloseSEs`)
- `GLUE2ToStorageServiceComputingServiceForeignKey` is the `ServiceId`
- `GLUE2ToStorageServiceStorageServiceForeignKey` is the `StorageServiceId` (read from `config_cream_gip_glue`: first attribute of a `CloseSEs`)

3 Batch system dynamic information

3.1 Current Glue 1 scenario

3.1.1 Torque/PBS

The PBS dynamic plugin for Glue1 publishes for each batch system queue something like:

```
dn: GlueCEUniqueID=cream-38.pd.infn.it:8443/cream-pbs-creamtest1,mds-vo-name=resource,o=grid
GlueCEInfoLRMSVersion: 2.5.7
GlueCEInfoTotalCPUs: 5
GlueCEPolicyAssignedJobSlots: 5
GlueCEStateFreeCPUs: 5
GlueCEPolicyMaxCPUTime: 2880
GlueCEPolicyMaxWallClockTime: 4320
GlueCEStateStatus: Production
```

3.1.2 LSF

The LSF dynamic plugin for Glue1 publishes for each batch system queue something like:

```
dn: GlueCEUniqueID=cream-29.pd.infn.it:8443/cream-lsf-creamcert2,mds-vo-name=resource,o=grid
GlueCEInfoLRMSVersion: 7 Update 5
GlueCEInfoTotalCPUs: 216
GlueCEPolicyAssignedJobSlots: 216
GlueCEPolicyMaxRunningJobs: 216
GlueCEPolicyMaxCPUTime: 999999999
GlueCEPolicyMaxWallClockTime: 999999999
GlueCEPolicyPriority: -20
GlueCEStateFreeCPUs: 6
GlueCEStateFreeJobSlots: 216
GlueCEStateStatus: Production
```

3.1.3 SGE

The SGE dynamic plugin for Glue1 publishes for each batch system queue something like:

```
dn: GlueCEUniqueID=sa3-ce.egee.cesga.es:8443/cream-sge-ops,mds-vo-name=resource,o=grid
GlueCEInfoLRMSVersion: 6.1u3
GlueCEPolicyAssignedJobSlots: 1
GlueCEPolicyMaxRunningJobs: 1
GlueCEInfoTotalCPUs: 1
GlueCEStateFreeJobSlots: 1
GlueCEStateFreeCPUs: 1
GlueCEPolicyMaxCPUTime: 4320
GlueCEPolicyMaxWallClockTime: 9000
GlueCEStateStatus: Production
```

3.1.4 Generic dynamic scheduler

The generic dynamic scheduler plugin, for Glue1 publishes for each VOView something like:

```
dn: GlueVOViewLocalID=alice,GlueCEUniqueID=cream-38.pd.infn.it:8443/cream-pbs-creamtest1,mds-vo-n
GlueVOViewLocalID: alice
GlueCEStateRunningJobs: 0
GlueCEStateWaitingJobs: 0
GlueCEStateTotalJobs: 0
GlueCEStateFreeJobSlots: 5
GlueCEStateEstimatedResponseTime: 0
```

GlueCEStateWorstResponseTime: 0

and for each queue publishes something like:

```
dn: GlueCEUniqueID=cream-38.pd.infn.it:8443/cream-pbs-creamtest1,mds-vo-name=resource,o=grid
GlueCEStateFreeJobSlots: 5
GlueCEStateFreeCPUs: 5
GlueCEStateRunningJobs: 0
GlueCEStateWaitingJobs: 0
GlueCEStateTotalJobs: 0
GlueCEStateEstimatedResponseTime: 0
GlueCEStateWorstResponseTime: 0
```

3.2 Work to be done to support Glue2 publication

3.2.1 Work to be done in the PBS/Torque information provider (done)

- The value published in Glue1 as GlueCEInfoLRMSVersion should be published in Glue2 as GLUE2ManagerProductVersion (ComputingManager objectclass)
- The value published in Glue1 as GlueCEPolicyMaxCPUTime should be published in Glue2 as GLUE2ComputingShareDefaultCPUTime (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEPolicyMaxObtainableCPUTime should be published in Glue2 as GLUE2ComputingShareMaxCPUTime (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEPolicyMaxWallClockTime should be published in Glue2 as GLUE2ComputingShareDefaultWallTime (ComputingShare objectclass) * For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEPolicyMaxObtainableWallClockTime should be published in Glue2 as GLUE2ComputingShareMaxWallTime (ComputingShare objectclass) * For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEStateStatus should be published in Glue2 as GLUE2ComputingShareServingState (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue

3.2.2 Work to be done in the LSF information provider (done)

- The value published in Glue1 as GlueCEInfoLRMSVersion should be published in Glue2 as GLUE2ManagerProductVersion (ComputingManager objectclass)
- The value published in Glue1 as GlueCEPolicyMaxCPUTime should be published in Glue2 as GLUE2ComputingShareDefaultCPUTime (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEPolicyMaxWallClockTime should be published in Glue2 as GLUE2ComputingShareDefaultWallTime (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEPolicyMaxRunningJobs should be published in Glue2 as GLUE2ComputingShareMaxRunningJobs (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue
- The value published in Glue1 as GlueCEStateStatus should be published in Glue2 as GLUE2ComputingShareServingState (ComputingShare objectclass)
 - ◆ For all the ComputingShares referring to that batch system queue

3.2.3 Work to be done in the SGE information provider (done)

- The value published in Glue1 as `GlueCEInfoLRMSVersion` should be published in Glue2 as `GLUE2ManagerProductVersion` (`ComputingManager` objectclass)
- The value published in Glue1 as `GlueCEPolicyMaxCPUTime` should be published in Glue2 as `GLUE2ComputingShareDefaultCPUTime` (`ComputingShare` objectclass)
 - ◆ For all the `ComputingShares` referring to that batch system queue
- The value published in Glue1 as `GlueCEPolicyMaxWallClockTime` should be published in Glue2 as `GLUE2ComputingShareDefaultWallTime` (`ComputingShare` objectclass)
 - ◆ For all the `ComputingShares` referring to that batch system queue
- The value published in Glue1 as `GlueCEPolicyMaxRunningJobs` should be published in Glue2 as `GLUE2ComputingShareMaxRunningJobs` (`ComputingShare` objectclass)
 - ◆ For all the `ComputingShares` referring to that batch system queue
- The value published in Glue1 as `GlueCEStateStatus` should be published in Glue2 as `GLUE2ComputingShareServingState` (`ComputingShare` objectclass)
 - ◆ For all the `ComputingShares` referring to that batch system queue

3.2.4 Work to be done in the generic dynamic scheduler (done)

- The value published in Glue1 as `GlueCEStateRunningJobs` for the `VOView` objectclass should be published in Glue2 as `GLUE2ComputingShareRunningJobs` (`ComputingShare` objectclass)
- The value published in Glue1 as `GlueCEStateWaitingJobs` for the `VOView` objectclass should be published in Glue2 as `GLUE2ComputingShareWaitingJobs` (`ComputingShare` objectclass)
- The value published in Glue1 as `GlueCEStateTotalJobs` for the `VOView` objectclass should be published in Glue2 as `GLUE2ComputingShareTotalJobs` (`ComputingShare` objectclass)
- The value published in Glue1 as `GlueCEStateFreeJobSlots` for the `VOView` objectclass should be published in Glue2 as `GLUE2ComputingShareFreeSlots` (`ComputingShare` objectclass)
- The value published in Glue1 as `GlueCEStateEstimatedResponseTime` for the `VOView` should be published in Glue2 as `GLUE2ComputingShareEstimatedAverageWaitingTime` (`ComputingShare` objectclass)
- The value published in Glue1 as `GlueCEStateWorstResponseTime` for the `VOView` should be published in Glue2 as `GLUE2ComputingShareEstimatedWorstWaitingTime` (`ComputingShare` objectclass)

4 Relevant RFCs

- <http://savannah.cern.ch/bugs/?69318>
- <http://savannah.cern.ch/bugs/?83297>
- <http://savannah.cern.ch/bugs/?83314>
- <http://savannah.cern.ch/bugs/?83338>
- <http://savannah.cern.ch/bugs/?83474>
- <https://savannah.cern.ch/bugs/?86609>

5 Testbed

The following machines are being used for testing

- cream-38.pd.infn.it (Torque)
- cream-29.pd.infn.it (LSF)
- cream-18.pd.infn.it (gLite-cluster node)

6 Raw notes (for Massimo use)

Dynamic batch system info:

- For a CE in no cluster mode: both glue1 and glue2
- For a CE in cluster mode with cluster not deployed in that cream ce host: yes glue1 but not glue2
- For a CE in cluster mode with cluster deployed in that cream ce host: yes glue1 but not glue2 (glue2 is printed by cluster)
- For a cluster (doesn't matter if on the same node there is also a cream ce): yes glue2 but no glue1

-- MassimoSgaravatto - 2011-06-21

This topic: CREAM > CreamGlue2

Topic revision: r48 - 2012-04-02 - MassimoSgaravatto



Copyright © 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback