

# The glite-CLUSTER node type

---

*Stephen Burke, Flavia Donno, Maarten Litmaath, Maria Alandes Pradillo,  
David Smith*

## ***Executive Summary***

*The glite-CLUSTER node type allows the configuration of information related to the batch system environment to be separated from the configuration related to the job submission interface, whether LCG-CE or CREAM. This in turn permits a site to publish such information easily, consistently and without workarounds when the configuration of the batch system is non-trivial, which is normally the case for larger sites. An important benefit of interest to the WLCG Management Board is the clean publication of installed capacities, avoiding double-counting by design. In particular the glite-CLUSTER can properly deal with sites having multiple CE head nodes and/or multiple sub-clusters (disjoint sets of worker nodes, each set having sufficiently homogeneous properties). Sites with simple configurations can still co-host the glite-CLUSTER on the CE itself.*

## **Cluster and SubCluster Publication**

The LCG CE as configured by the current yaim module has various information providers to publish GLUE objects to the information system:

- A GlueService object for the gatekeeper endpoint.
- A GlueService object for the gridftp server (see below).
- One GlueCE object per batch queue accessible via the gatekeeper, each with one dependent GlueVOView per VO allowed to submit to the queue.
- One GlueCluster and one dependent GlueSubCluster, describing the hardware of WNs in the batch system and with a set of RunTimeEnvironment strings to identify software installed on the WNs (see below). There is no information provider for the Cluster and SubCluster attributes (apart from the RTE tags), the LDIF is simply published verbatim from a text file, which has to be created from information supplied by the sysadmin, using YAIM or by some other method. In the past we attempted to collect the information in an automated way but no robust method was found; in any case sysadmins are now used to providing the information as part of their YAIM configuration.

In GLUE 1 these objects are all published independently in the LDAP tree but are related to one another via foreign keys or other connections.

For the purposes of this note a CREAM CE is essentially identical to an LCG CE, it just has a different type of endpoint.

GLUE defines a SubCluster to be homogeneous, i.e. it should represent a set of identical, or effectively identical, WNs. The schema relates the GlueCE to the GlueCluster; if a batch queue can submit to more than one type of WN the idea was to have multiple SubClusters within the Cluster. However, the RB and WMS have never supported that, they require publication of exactly one SubCluster per Cluster (and hence per batch queue). The new ability of CREAM to pass requirements to the batch system does not affect this restriction, although it may ameliorate some of the consequences.

Thus sites with heterogenous hardware (the usual case after several procurement cycles) have two choices: either publish a SubCluster with a representative/minimum hardware description, e.g. the minimum memory on any node, or define separate batch queues for each hardware configuration, e.g. low/high memory queues, and attach the corresponding GlueCE objects to separate Cluster/SubCluster pairs. For attributes with discrete values, e.g. SL4 vs SL5, the second option is the only one which makes sense.

However, the current production version of YAIM (for both CE types) is only capable of configuring a single Cluster per CE head node. This implies that sites which want to use multiple Clusters are forced to have multiple head nodes if they want to use YAIM, which imposes an extra administrative load. The alternative is to modify the YAIM-generated LDIF by hand to publish multiple Clusters from a single node, but this requires a good knowledge of the information system and is error-prone. Sites with limited sysadmin effort/knowledge are therefore likely to find both of these options unsatisfactory.

A second problem arises for larger sites which install multiple CE headnodes submitting to the **same** batch queues for redundancy and load-balancing. Sites with both LCG-CE and CREAM nodes submitting to the same queues face the same problem. In this case YAIM generates a separate Cluster/SubCluster pair for each head node even though they all describe the same hardware. This causes no problems for job submission, but by default would overcount the installed capacity at the site by a multiple of the number of SubClusters. The current recommended solution is to publish zero values for the installed capacity from all but one of the nodes. However, this makes one node "special", needing a different configuration to the others, and means that if that node is down the installed capacity is reported as zero. The multiple publication of identical information also increases the total volume of data in the information system unnecessarily.

The problems described above are multiplicative: a site which wanted, say, 3 head nodes per queue for redundancy, two SubCluster types (e.g. SL4 and SL5) and installed both the LCG CE and CREAM would need a total of 12 nodes if it used YAIM, and would overcount the installed capacity by a factor 6 unless all nodes but two (one SL4 and one SL5) published zeroes. Again it is possible to hand-configure the publication, e.g. the RAL Tier-1 has a very complex home-grown solution, but this is only feasible for the largest and most experienced sites.

## Installed Software Publication

This is described separately as it forms a fairly complex subsystem in its own right. The components and their functions are as follows:

- Sites give each VO an area of disk space in which they can install VO-specific software (semi-) permanently. This is typically NFS-mounted on each WN. A VO-specific environment variable on the WN points to the area. The areas are readable by any user in the VO, but writeable only by a user mapped to a special unix group (a so-called "sgm account"). This requires a corresponding role to be assigned correctly in VOMS and mapped correctly to an sgm pool account on the site. Sites may also give sgm accounts priority or other special treatment in the batch system - the role should only be used for the purpose of installing software.
- Once an sgm user has installed some software and verified that it works, she needs to publish a RunTimeEnvironment tag in the SubCluster so that the WMS can match to sites with the right software installed. This can (for historical reasons) be done using either of two commands, lcg-tags or lcg-ManageVOTag. These contact a GridFTP server running on the CE head node and use it to write to a specific, hardcoded file containing all the tags for the VO. Again this must be writeable only by an sgm user.
- The CE node runs an information provider which appends the tags for all VOs to the static SubCluster information described above. This implies that if a site has multiple head nodes the tag area needs to be mounted to be visible on all of them (and should also be backed up since a VO may be significantly impacted if the tags are lost - in the worst case all software may need to be reinstalled). A second, more significant, implication is that the same tags appear in all SubClusters on a given node, which caused some problems during the SL4/SL5 migration as SL4 software often doesn't work on SL5. The only way to avoid that is by having separate head nodes for SubClusters which need different tags.
- The information provider also publishes a GlueLocation object for each installed software tag. This was designed to allow the location of the software to be discovered externally, as opposed to using the environment variables mentioned above which are only visible to running jobs. However, with the very large number of software versions published in particular by atlas and cms this generates a very large data volume in the information system, despite the fact that the Location objects are not used by those VOs. The system as it stands has no way to control whether those objects are published. In addition, GLUE 1.3 obsoleted the Location object in favour of a new GlueSoftware object with additional information (basically to allow software-specific setup information to be published, e.g the name of a setup script), but this is not supported by the current system.

## The Cluster Node

In 2007 the EGEE TCG set up a working group to look at various aspects of the way Worker Nodes are used, chaired by Steve Traylen:

<http://egee-intranet.web.cern.ch/egee-intranet/NA1/TCG/wgs/wn.htm>

The working group produced a plan consisting of several small pieces and one fairly big one (the cluster node), which was endorsed by the TCG and has now largely been implemented. One important constraint was that the changes could be implemented gradually without breaking any existing functionality. Some information about the plan is available on the following wiki page, and the main steps are summarised below:

<https://twiki.cern.ch/twiki/bin/view/EGEE/WNWorkingGroup>

- A command `glite-wn-info` is installed on each WN to return the UniqueID of the SubCluster to which the WN belongs. In production since May 2009.
- The endpoint of the GridFTP server used to write the RunTimeEnvironment tag files for a given SubCluster is published as a GlueService object. This replaces the hardwired assumption that the server is on the CE head node, allowing it to be relocated to another node. It also allows the use of something other than GridFTP, e.g. `apache+gridsite`, to write the files, since the protocol is implied by the endpoint URL scheme. It also publishes the location of the tag file directory, potentially removing the need for the location to be hardwired. In production since July 2009.
- A new information provider is provided to publish the RunTimeEnvironment tags for the SubCluster. The previous structure allowed only one set of tags to be stored on a given node. The new provider (`lcg-info-dynamic-software`) reads both the old-style tags and a new structure which has separate tag files for each SubCluster, allowing SubClusters requiring different software installations, e.g. SL4/SL5, to be published from the same node. The location of this new area is published in the GlueService object described above. The information provider has been in production since June 2009. YAIM also needs to create the new directories with the right permissions - in production since May 2009.
- The tag management commands `lcg-tags` and `lcg-ManageVOTag` need to be enhanced to support publication of tags for a specific SubCluster, and to discover the relevant GridFTP (or http) endpoint from the information system. For `lcg-tags` this is in production since July 2009, but for `lcg-ManageVOTag` the change has only been made in January 2011 and is not yet in production (bug #46726).
- A new `glite-CLUSTER` node type needs to be created. This contains a subset of the functionality previously incorporated in both the LCG-CE and CREAM node types: the publication of the GlueCluster and its dependent objects, the publication of the GlueService object for the GridFTP endpoint, and the directories which store the RunTimeEnvironment tags, together with the YAIM functions which configure them. Those YAIM functions also need to be enhanced to allow the creation of multiple SubClusters to represent sets of WNs with different properties, which allows much greater flexibility to tailor the configuration to the needs of sites. This work was started in February 2008, but for various reasons has only just (November 2010) been certified.
- The LCG-CE and CREAM node types need to have the functionality removed which is now incorporated in `glite-CLUSTER`. The previous situation with everything installed on one node is still available simply by installing both node types on a single node. The YAIM functions which configure the publication of the GlueCE objects also need to be adapted to allow them to point to the GlueClusterUniqueID(s) defined in the `glite-CLUSTER` configuration. This modification has now been certified for the LCG-CE, but has not yet been started for CREAM.

It is however relatively simple to implement. The release of the modified LCG-CE is currently on hold until the use of the glite-CLUSTER functionality has been made optional, so that a site will not have to change its YAIM configuration if a glite-CLUSTER would bring little benefit (e.g. because it only has a single head node and a single type of WNs). This is expected to be done by the end of January 2011.

## GLUE 2

The GLUE 1 computing schema has grown by accretion and has various defects, in particular that much of the published information is duplicated and that the information which can be published is limited relative to our current needs. GLUE 2 aims to solve these problems: it has a logical structure which is shared by all services, it has many more attributes to help satisfy all current use-cases, and it is much more extensible to allow new use-cases to be supported without needing a schema revision.

For the publication of information about computing systems the main GLUE 2 entities are as follows:

- **ComputingService** - a single object which summarises the properties of the entire system, and which is the parent of all the other objects both in the abstract schema and in the published LDAP tree. There is no equivalent in GLUE 1.
- **ComputingManager** - one object to represent the batch system as a whole. To a large extent this is new in GLUE 2, although it contains some information currently in the GlueCE.
- **ComputingShare** - one object per batch queue or per scheduling group within a batch system, i.e. it represents a set of jobs which are scheduled in some uniform way according to a policy. This contains information present in the current GlueCE and GlueVOView objects.
- **ExecutionEnvironment** - one object per set of homogeneous WNs to represent their hardware properties. This is largely the same as the current GlueSubCluster object.
- **ApplicationEnvironment** - represents the properties of a particular installed software application, i.e. one object per application. This is analogous to the current GlueLocation/GlueSoftware objects, and also absorbs the functionality of the RunTimeEnvironment attribute in the GlueSubCluster.
- **ComputingEndpoint** - one object per head node, to represent the external interface(s) to the system. This contains information present in the current GlueService and GlueCE objects. This object would also be used to publish the endpoint for the GridFTP server used in the software installation system.

From the description above it can be seen that the majority of these objects should be published from a single node, with access to the dynamic state of the batch system and to information about the WN hardware and installed software. The natural way to achieve this is to extend the functionality of the glite-CLUSTER node, which already publishes the equivalent hardware and software objects for GLUE 1. It would also be natural to publish the ComputingEndpoint objects from the corresponding head nodes and collect them on the central node, and in fact the production version of CREAM already publishes a prototype Endpoint to the GLUE 2 BDII.

