WLCG Installed Capacity information

Short Version

Editor: Flavia Donno

Date: 11/24/2008

Version: 1.0

Contributors/Authors:

Stephen Burke (GLUE/RAL), Greig Cowan (Storage Accounting/Edinburgh), Flavia Donno (WG Coordinator/CERN), Laurence Field (Glue/CERN) Jens Jensen (RAL), Michel Jouvin (DPM/GRIF), Miguel Marques Coelho Dos Santos (CASTOR/CERN), Luca Magnoni (StoRM/CNAF), Paul Millar (dCache/DESY), Jason Shih (CASTOR/Taiwan), Jeff Templon (NIKHEF), Steve Traylen (WN WG/CERN), Ron Trompert (dCache/SARA), Jan Van Eldik (CASTOR/CERN), Riccardo Zappi (StoRM/CNAF), Tanya Levshina, Burt Holzman, Brian Bockelman (OSG)

Purpose of this document

The goal of this document is to detail how to use the Glue Schema version 1.3 in order to publish information to provide the WLCG management with a view of the total installed capacity and resource usage by the VOs at sites. This information can also be used by VO operations and management in order to monitor the VO usage of the resources:

This document is organized in two main sections: the first one covers the details concerning the CPU installed capacity, while the second one tackles the issues concerning storage resources.

1. WLCG Computing Resources

In what follows, we detail the meaning of the several Glue Classes concerning computing resources. We specify how the various attributes must be interpreted.

1.1 The Glue Cluster, SubCluster, CE Classes

We now describe the individual Glue classes attributes needed in order to report on the Computing Installed Capacity at a site.

The class ***Cluster*** represents an aggregation of heterogeneous computing resources managed by a batch system. The *Cluster* class has a set of attributes described in section 3.1 of [1]. A *Cluster* has references to its *SubCluster*s, to the Computing Elements or queues that it manages, and to the site it is part of.

A ***SubCluster*** is a set of (homogeneous) computing resources forming part of a Cluster. The attributes of a subcluster are described in section 3.3 of [1].

> ***Note:*** *Please note that although a SubCluster is defined as a homogeneous set of WNs, in practice the way the WMS works limits us to publishing one SubCluster per Cluster (basically the current WMS can't tell the LRMS which SubCluster it wants to use). The result is that in general* ***SubClusters are heterogeneous****. Sites have to publish some kind of average WN specification as SubCluster attributes. The Worker Nodes Working Group proposes to split the system and have separate queues (CEs), e.g. for large memory nodes. Splitting the system to have separate homogeneous queues is the solution that is proposed to be adopted in WLCG.*

> ***Note:*** *It is mandatory for WLCG sites to publish Glue SubCluster objects and in particular the attributes outlined in what follows.*

The *SubCluster* attributes relevant for making public available information about the installed capacity at sites are:

PhysicalCPUs - defined as the "*Total number of real CPUs/physical chips in the SubCluster*".

Please note that by the number of real CPUs we **do not** intend the number of cores [3] in a SubCluster, but indeed the number of processors chips [2] installed in the Worker Nodes comprising a SubCluster.

Please note that the PhysicalCPUs ***must be a static number***, i.e. it is configured by the admin and does not change if a few nodes are down temporarily.

The PhysicalCPUs numbers should be summed over all WNs in a SubCluster to obtain the total number of CPUs in that SubCluster.

Another WLCG mandatory attribute is:

LogicalCPUs - defined as the "*Total number of cores/hyperthreaded CPUs in the SubCluster*"

In other words, LogicalCPUs counts the number of computing units seen by the OS on the WNs of a SubCluster.

Please note that again LogicalCPUs is a static number manually configured by the system administrator at a site and does not reflect the dynamic state of the WNs.

A ***Host*** entity is attached to the SubCluster. It is meant to describe each node in the (homogeneous) SubCluster. The Host attributes are described in section 3.3 of [1]. The Host attributes relevant for this work are:

BenchmarkSI00- defined as the "*SpecInt2000 provided by a typical SubCluster Logical CPU*"

For heterogeneous SubCluster we propose the following definition for BenchmarkSI00:

BenchmarkSI00 defined as the "*Average SpecInt2000 rating per LogicalCPU*"

In the case of a homogenous SubCluster, the only change is that the SpecInt rating is provided per core, instead of per chip. In the case of a heterogeneous SubCluster, the associated BenchmarkSI00 MUST be calculated as follows. If A, B and C are sets of homogeneous machines in a SubCluster SC, if we indicate with:

p_A = the power per core for set A

n_A == number of cores in for set A

and we use a similar symbology for sets B and C, the BenchmarkSI00 for the heterogeneous SubCluster SC is:

(1) $BenchmarkSI00_{SC} = (p\_A * n\_A + p\_B * n\_B + p\_C * n\_C) / (n\_A + n\_B + n\_C)$

Note: This attribute is normally a static value filled in by hand by the system administrator of the site. This is one source of possible errors, since it depends on a manual procedure to ensure the correctness of this information.

1.2 Computing the total installed computing capacity at a WLCG site

In order to calculate the ***total installed capacity at a site*** the following formula will be used in WLCG:

$$\textbf{Total Installed Computing Capacity}_{Site}\ (KSI00) = \left(\sum_{WLCG\ Subclusters} GlueHostBenchMarkSI00 * GlueSubClusterLogicalCPUs\right)/10^3$$

where the sum will be executed over all SubClusters used by queues (CEs) which support WLCG VOs.

2. WLCG Storage Resources

In what follows, we outline the Glue Classes attributes concerning installed storage resources.

2.1 Glue Storage Element, Storage Area

The Storage Element (SE) Class describes storage resources at a site. There MAY be more than one SE at one given site. A Storage Element represents a convenient partition of the storage resources of one or more storage systems as a single Grid entity.

The attributes of the Storage Element Class are described in section 4.1 of [1]. Here are the relevant attributes to be considered when publishing the storage installed capacity at a site:

- The *SizeTotal, SizeFree, TotalOnlineSize and UsedOnlineSize* (in GB) SHOULD be published. They SHOULD be aggregated from what's in the SAs and they SHOULD summarize the space in the entire SE.

We now describe the GlueSA and GlueVOInfo classes that are the most relevant for the publication of storage resource installation and usage.

The Glue ***Storage Area*** (SA) class describes a logical view of a portion of space that can include disks and tape resources. Storage Areas map to ***physical*** portions of storage. ***SAs MUST NOT overlap***. Shared portions of storage MUST be represented with a single GlueSA object, with multiple GlueSAAccessControlBaseRule attributes and optionally with multiple VOInfo objects pointing to it.

Normally a Storage Area is used to represent SRM-reserved/used space. It is RECOMMENDED that a Storage Area object be published for portions of storage configured but yet unreserved. In this case the SA MUST publish ReservedOnlineSize=0 and ReservedNearlineSize=0. A special Capability attribute MUST be used to describe this situation (the InstalledCapacity attribute described later):

- *AccessControlBaseRule* This attribute SHOULD be set. Formally it is allowed to publish an SA with no ACBRs, although it would be inaccessible - e.g. to publish unallocated space. If this attribute is set, its value MUST be one of the following:

  a. <DN>

  b. <VO NAME> - deprecated

  c. VO:<VO NAME>

  d. VOMS:<FQAN>

  Clients MUST be able to accept multi-valued AccessControlBaseRule. Multiple ACBRs are ORed, i.e. access is assumed to be allowed if any of them match. Clients SHOULD ignore ACBR schemes they do not understand to allow for future expansion. There is NO negative ACBR. Clients SHOULD be prepared to accept incorrectly formatted AccessControlBaseRule attribute values containing the VO name only, with no qualifying scheme identifier. Wildcards in ACBRs are currently not allowed, but there is an EGEE proposal [6] to support limited wildcards.

- *Reserved[Online|Nearline]Size* (in GB=10^9 bytes) is a portion of available storage physically allocated to a VO or to a set of VOs. The value of this attribute MUST be 0 for a Storage Area representing an unreserved space. The Reserved Online [Nearline] Size MUST NOT be negative. For WLCG usage *Online* refers to space on disk while *Nearline* refers to space on tape. For tapes, sizes MUST be reported publishing the actual size on tape after compression.

- *Total[Online|Nearline]Size* (in GB=10^9 bytes) is the total online or nearline space available at a given moment (it does not include broken disk servers, draining pools, etc.). In the absence of unavailable pools the Total Size is equal to the Reserved Size. The Total Online [Nearline] Size MUST NOT be a negative number and MUST NOT exceed the Reserved Online [Nearline] Size.

- *Used[Online|Nearline]Size* (in GB=10^9 bytes) is the space occupied by available and accessible files that are not candidates for garbage collection. For CASTOR, since all files in T1D0 are candidates for garbage collection, it has been agreed that in this case UsedOnlineSize is equal to GlueSATotalOnlineSize. For T0D1 classes of storage this is the space occupied by valid files. Size MUST NOT be a negative number and MUST NOT exceed the Total Online [Nearline] Size. For a definition of *TnDm*, please refer to [5].

- *Free[Online|Nearline]Size* (in GB=10^9 bytes) is equal to Total – Used

- *Capability* is a string that publishes various characteristics of a Storage Area. At the moment the agreed capabilities of a Storage Area are:

    a. *Installed[Online|Nearline]Capacity*=<size> (in GB=10^9 bytes). This attribute has been added to track unavailable but configured space for accounting purpose. This attribute MUST always be published. This expresses the size of the space configured but not yet reserved to any specific VO, or space which is reserved for a VO but not installed in an active space token, or simply space already reserved and assigned to a VO. In order to collect the information about storage installed capacity at a site, clients can add up the value of this attribute for every SA at a site. To give an example of publishing unavailable space in CASTOR, an SA with disk servers online in a service class not associated to a specific space token description for a given VO will be published with the InstalledOnlineCapacity=<size> capability.

    b. *scratch*. This capability indicates that this Storage Area is of the type T0D0, not supported according to the WLCG SRM Usage Agreement. In this space, files MUST be created by users as VOLATILE and MAY be removed by the system as soon as their lifetime expires.

    c. *stage*. This is a Storage Area used for staging operations only. This SA has no associated VOInfo object. The SAAccessControlBaseRules list all FQAN that can use this SA.

The Glue ***VOInfo*** class describes VO specific attributes of a Storage Area, or better it gives a view of a Storage Area from a VO perspective. One Storage Area can be associated to zero or more VOInfo objects.

- *VOInfoTag*. If an SA contains resources allocated to a VO and the VO can access such an SA via a space token description, then the SA MUST have a VOInfo object associated with it which publishes the space token description. The *VOInfoTag* attribute MUST be used for this.

- *VOInfoAccessControlBaseRule* MUST be published. . This MUST correspond to a logical subset of the rules published in the corresponding SAAcessControlBaseRules, and the ACBRs for VOInfo objects associated with the same SA MUST NOT overlap. The value MUST be one of the following:

  a. <DN>

  b. <VO NAME> - deprecated

  c. VO:<VO NAME>

  d. VOMS:<FQAN>

  Clients MUST be able to accept multi-valued AccessControlBaseRule. Multiple ACBRs are ORed, i.e. access is assumed to be allowed if any of them match. Clients SHOULD ignore ACBR schemes they do not understand to allow for future expansion. There is NO negative ACBR. Clients SHOULD be prepared to accept incorrectly formatted AccessControlBaseRule attribute values containing the VO name only, with no qualifying scheme identifier. Wildcards in ACBRs are currently not allowed, but there is an EGEE proposal [6] to support limited wildcards.

- *VOInfoPath* is a string that describes the Path to be used in constructing a SURL when writing to the associated Storage Area. A VOInfo object that does not publish a VOInfoPath indicates that the associated Storage Area cannot be accessed in write mode by the associated VO/FQAN. Please note that the SAPath SHOULD NOT be published if the VOInfoPath is published. If both path attributes are published, the client MUST use the one in the VOInfo object.

2.2 Computing the Total Installed Online and Nearline Storage Capacity at a Site

In order to calculate the total storage installed capacity at a site the following formula will be used in WLCG:

$$\textbf{\textit{Total Installed Online Storage Capacity}}_{Site}\ \textbf{\textit{(GB)}} =$$

$$\left(\sum_{WLCG\ GlueSA} \textbf{\textit{GlueSACapability(InstalledOnlineCapacity)}}\right.$$

$$\textbf{\textit{Total Installed Nearline Storage Capacity}}_{Site}\ \textbf{\textit{(GB)}} =$$

$$\left(\sum_{WLCG\ GlueSA} \textbf{\textit{GlueSACapability(InstalledOnlineCapacity)}}\right.$$

where the sum will be executed over all Storage Areas used by WLCG VOs.

REFERENCES

[1] Glue Schema Specification version 1.3, Final – 16 Jan 2007, http://forge.cnaf.infn.it/plugins/scmsvn/viewcvs.php/*checkout*/v_1_3/spec/pdf/GLUESchema.pdf?rev=48&root=glueschema
[2] Wikipedia definition of CPU: http://en.wikipedia.org/wiki/CPU
[3] Wikipedia definition of Multi-core CPU: http://en.wikipedia.org/wiki/Multi-core
[4] http://goc.grid.sinica.edu.tw/gocwiki/How_does_GStat_count_CPUs
[5] Storage Element Model for SRM 2.2 and GLUE schema description, F. Donno et al., v. 3.5, 27 October 2006, https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.glue-wg/docman.root.background.specifications/doc14619;jsessionid=58E33DC10A69FABED90ACD4C8EFE6E1F
[6] Recommendations for Changes in gLite Authorization, C.Witzig, 14 April 2008, https://edms.cern.ch/document/887174/1