Glue2 support in gLite CREAM-CE and WMS

Massimo Sgaravatto – INFN Padova

Salvatore Monforte – INFN Catania

Glue2 in CREAM-CE

- Initial support for Glue2 in CREAM-CE provided with EMI-1
- But still many things to finalize/fix
- Work tracked in Savannah task #21912
  - Planned to be finished on December 2011
  - The “Should be finished on” was August 2011
- http://wiki.italiangrid.org/twiki/bin/view/CREAM/CreamGlue2

Deployment scenarios

- CREAM CE can be deployed
  - In cluster mode
    - To be used when there are multiple CE head nodes and/or in the site there are multiple disjoint sets of worker nodes
    - gLite-cluster node and one or more CREAM-CE node
    - In gLite-cluster publising of resources and RunTimeEnvironments related information
  - In no-cluster mode
    - Useful for small sites

Deployment scenarios and Glue2

- No cluster mode
  - All Glue2 objectclasses are published by the resource BDII running on the CREAM-CE
- Cluster mode
  - The resource BDII running on each CREAM-CE just publishes
    - The EndPoint objectclass for CREAM (along with the AccessPolicy objectclasses)
    - The Endpoint objectclass for CEMon (if it is deployed)
  - The resource BDII running on the gLite-cluster publishes all the other objectclasses

Implementation status

- ComputingService
  - Basically done
  - Just to address a problem with Glue2ServiceComplexity in cluster mode
- ComputingEndpoint
  - Basically done
  - Just to enhance the publication of Glue2EndpointServingState
- Policy for the ComputingEndpoint
  - Done

Implementation status (cont.ed)

- ComputingShare
  - Still to do
    - Dynamic batch system information (see next slides)
    - Links to EndPoints and ExecutionEnvironments
    - Finalize for cluster mode
- MappingPolicy for ComputingShare
  - Done for no cluster mode
  - To be finalized for cluster mode

Implementation status (cont.ed)

- ComputingManager
  - Missing dynamic batch system specific information (see next slides)
- Benchmark
  - Done
- ExecutionEnvironment
  - Done
- ApplicationEnvironment
  - To do: see next slides

Implementation status (cont.ed)

- ApplicationHandle
  - Not done, and we don't plan to implement it (for application so far we only published the RTE tags)
- ComputingActivity
  - Not done, and we don't plan to implement it (we never published information about jobs in the infosystem)
- ToStorageService
  - Done
- Endpoint for CEMon
  - Done
- Endpoint for RTEPublisher
  - Done

Batch system dynamic information

- Not too much excitement in the harmonization ideas presented at last AHM
- So Glue2 support on the existing implementation
- Identified and documented in detail what is needed to be done
- Apart from SGE (CESGA), the relevant providers have been implemented and are currently maintained by people who are not part of EMI project
  - LSF: U. Schwickerath (Cern) agreed to do the work
  - Torque: J. Templon (J. Templon) agreed to do the work for the python part
    - Not willing to do the job for the Perl part, but we should we able to manage that
  - Generic scheduler: J. Templon (J. Templon) agreed to do the work

ApplicationEnvironment

- In Glue1
  - We just had to publish a list of RTE tags
  - The default publication generates a Glue Location object for every RTE tag
  - Data volume issue:
    - Some VOs have a huge number of tags, so those objects take a lot of space
    - → We had to drop these Location objects and we just publish the RTEs in the GlueSubcluster (GlueHostApplicationSoftwareRunTimeEnvironment)

ApplicationEnvironment (cont.ed)

- We have the same issue in Glue2 since ApplicationEnvironment is a separate object
- What should we do ?
  - Ignore the issue and use a ApplicationEnvironment objectclass for each RTE
  - Consider a different solution, e.g. to group all the tags for a given VO in a single ApplicationEnvironment object
- We need to find and implement a common solution

Some implementation details

- For RTEPublisher and CEMon endpoints using BDII GIP provider
- For the other objectclasses LDIF files plus BDII GIP plugins for dynamic info
  - LDIF files created by Perl scripts which read the relevant values from a configuration file
  - The configuration file is filled by yaim
  - Scripts generating the ldif files run by yaim
  - Less dependent on yaim then before
    - For glue1, yaim writes directly the ldif files

Comparison with ARC-CE

- Checked what is published by a ARC-CE running EMI-1 Update 8 (pgs03.grid.upjs.sk) to see the main differences
  - Some objectclasses are published by gLite but not by ARC, and viceversa
    - E.g.gLite doesn't publish ComputingActivity
    - E.g. ARC doesn't publish Benchmarsks and policies for ComputingEndPoint and for ComputingShare
      - The missing policies is a problem for submission through WMS, since the WMS needs to know who is authorized

Comparison with ARC-CE (cont.ed)

- Different use of some objectclass
  - For gLite a share is a Voview (since many sites do not dedicate queues to Vos)
  - for ARC a share is a queue
- For each objectclass some attributes are published in ARC but not in gLite and viceversa
- For some attributes we publish in different ways (e.g. GLUE2EndpointTrustedCA)

Support for Glue2 in WMS

- The first step is acquiring information from GLUE2 enabled BDII
  - code is almost complete as far as querying computing resources is concerned
    - ObjectClasses involved GLUE2:
      - ComputingService,ComputingManager, ComputingShare, ComputingEndPoint, ToStorageService, MappingPolicy, ExecutionEnvironment, ApplicationEnvironment, Benchmark
  - Coding involving retrieving of StorageService info and MM engine not started yet
    - coding delay also because a storage disaster wiping all the work performed until July 2011 along with 98TB of data at INFN-CT

Information SuperMarket

- ClassAd representation of GLUE2 resources
  - GLUE2 prefix stripped from both objectclass and attributes names and define the first-level Ad in the structured representation
    - a nested classad is created for each objectclass at the lowest hierarchy LDAP level
      - attributes form inherited objectclasses are flattened within nested Ads
  - backward compatibility (when possible)
    - mapping old names to GLUE2 ones using attribute references:
      - *GlueHostNetworkAdapterOutboundIP = GLUE2.ExecutionEnvironment.ConnectivityOut;*
      - *LRMSType = GLUE2.ComputingManager.ProductType;*

Information SuperMarket

```
[ GLUE2 = [
                ComputingShare =
                    [
                        CreationTime = "2011-06-13T05:53:23Z";
                        ServingState = "production";
                        ID = "creamtest1_alice_cream-38.pd.infn.it_ComputingElement";
                        ComputingServiceForeignKey = "cream-38.pd.infn.it_ComputingElement";
                        MappingQueue = "creamtest1";
                        ResourceForeignKey =
                           {
                              "cream-38.pd.infn.it"
                           };
                        ServiceForeignKey = "cream-38.pd.infn.it_ComputingElement";
                        ExecutionEnvironmentForeignKey =
                           {
                              "cream-38.pd.infn.it"
                           };
                        OtherInfo =
                           {
                              "CREAMCEId=cream-38.pd.infn.it:8443/cream-pbs-creamtest1",
                              "InfoProviderName=glite-ce-glue2-share-static",
                              "InfoProviderVersion=1.0",
                              "InfoProviderHost=cream-38.pd.infn.it"
                           }
                    ];
                ExecutionEnvironment =
                    [
                        [...]
                    ];
    [...]
]]
```

GLUE1 and Glue2 together …

- Handling resources exposing both GLUE1.3 and GLUE2.0
  - prioritized ISM purchasing
    - precedence to G2.0 purchasing
    - only G1.3 resources not yet inserted in ISM by G2.0 purchaser considered
      - For that we need to know the glue1 entity mapped to this Glue2 share
      - Possible solution: Shares **should** publish the G1.3 ID under OtherInfo eg:
        - » OtherInfo : GLUE13ID=cream-38.pd.infn.it:8443/cream-pbs-creamtest1-voview1
- JDLs containing attribute specification compliant with
  - GLUE2
    - match **ONLY** GLUE2 resources
  - GLUE1.3
    - match GLUE13 and **might match** GLUE2 resources
      - due to restrictions on 1-1 mapping

Many to many relations

- In Glue 2 several relations are explicitly many to many
  - e.g. :different EEs may relate different sets of AEs
  - The canonical bilateral ClassAdMatch might not be generic enough
    - users should explicitly use GangMatch expressions
      - requirements= anyMatch(other.ExecutionEnvironment, target.LogicalCPUs >= 2) && ...;
    - But some limits in the current GangMatch extension
      - works only at the first nesting level
        - » i.e. it is not possible to reference attributes belonging to the j-th AE relevant to the i-th EE of a given Share

Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611