



EUROPEAN MIDDLEWARE INITIATIVE

Compute Area Summary

Marco Cecchi
INFN

Topics

- Consolidation of the Parallel framework
- GLUE20 status for ARC and gLite
- EMI-ES implementation in server and clients
 - ARC
 - gLite
 - UNICORE
- CAR
- EMI clients harmonization

Framework for Parallel Jobs

- EMI-ES will provide a common interface for *submitting* jobs
- ***ParallelEnvironment*** in EMI-ES is the ***common framework*** for parallel jobs
- Modifications proposal by the MPI TF:
 - **Type** as a free string (e.g. MPI/OpenMP)
 - Change **ProcessesPerSlot** to **ProcessesPerHost**
 - **ThreadsPerProcess** with additional tag
useSlotsPerHost="true"
 - Any extra features in the **option** element →
fine, watch out

A12.1: Implementation of the proposal

- EMI-ES + mpi-start as basis for PE implementation
 - Already performed adaptation for ARC RuntimeEnvironments and UNICORE ParallelEnvironmnet.
- To be finished by M32, but should be fast once the EMI-ES implementations are ready

Parallel Framework JRA1.1.2 Tasks

Subtask	Name	Owner	Due
A11.1	define a proposal for a parallel execution framework within EMI	MPI TF	M18
A12.1	implementation of the proposal for a parallel execution framework within EMI	MPI TF	M32 In progress
A13.1	enable capabilities to support multi-core, multi-node execution in ARC	Arc CE	M36 In progress
A13.2	enable capabilities to support multi-core, multi-node execution in gLite	gLite JM	M36 In progress
A13.3	enable capabilities to support multi-core, multi-node execution in UNICORE	UNICORE -*	M36 In progress
A13.4	extend the capabilities to better address emerging architectures, multi-node execution on interconnected clusters and FPGAs, GPGPUs to support cross-middleware multi-core, multi-node execution	MPI TF	M36 ?

ARC GLUE2 Status (LDAP and XML)

- Florido Paganelli replaced Adrian Taga as developer/maintainer
- **Correctness**
 - Used schema schemas.ogf.org/glue/2009/03/spec/2/0
path to/GLUE2.xsd
 - Not the latest&greatest
- **Completeness**
 - Some information is missing, must be checked:
 - ARC CE has several endpoints, some are not published
 - Further investigation needed

ARC GLUE2 Plans

Mid of November:

- Fix information completeness

As soon as EMI updates LDAP schema

- Verify LDAP rendering
- Test GLUE2 LDAP client and server

Complete GLUE2 Support server-side will be provided for EMI2

GLUE20 in gLite CREAM

- Initial support for Glue2 in CREAM-CE provided with EMI-1, more or less static info
- Current work is addressing
 - 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 nodes
 - In gLite-cluster publishing of resources and RunTimeEnvironments related information
 - In no-cluster mode
 - Useful for small sites, it reduces to full GLUE20 for a CREAM NODE

GLUE20 in gLite CREAM

- Implementation status is almost done
 - Not clear whether GLUE20 support for gLite cluster will be provided by EMI2
- Dynamic information → EMI2?
 - It will rely on the usual dependencies (not the ARC modules)
 - 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 be able to manage that
 - Generic scheduler: J. Templon (J. Templon) agreed to do the work

gLite: 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 Benchmarks 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
 - They do publish such information as ACBRs in their resources publishing GLUE1.3

gLite: Comparison with ARC-CE (cont.ed)



- Different use of some objectclasses
 - For gLite a share is a Voview (since many sites do not dedicate queues to Vos)
 - for ARC a share is a queue
- For some attributes we publish in different ways (e.g. GLUE2EndpointTrustedCA)
- Scalability concerns with ApplicationEnvironment
 - We had hacks in GLUE1.3 to address this issue

GLUE20 in gLite 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
 - Then, new purchaser threads must be written to populate our cache in classad
 - backward compatibility (when possible)
 - mapping old names to GLUE2 ones using attribute references:
 - *GlueHostNetworkAdapterOutboundIP = GLUE2.ExecutionEnvironment.ConnectivityOut;*
 - *LRMSType = GLUE2.ComputingManager.ProductType;*

GLUE20 in gLite WMS

- We are striving to preserve backwards compatibility from the users point of view
 - Yet, the degree of complexity increases even more
 - The ISM footprint may explode
 - 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
 - To support fully meshed scenarios (and not all LRMSs do support this) with one share having more EEs and AEs very complex (and heavy) JDL extensions must be written. That would be phase II, in case and that would not be easy anyway.
 - Plans are now to support one EE per share

Compute area GLUE20

Conclusion:

none among A, G, U publishes GLUE20 information profiled for interoperability

Interoperability achieved between ARC and gLite in the past is at stake, with GLUE20

We need a working group on this. Making sure that UNICORE publishes usable info is important as well (COMPChem, DIRAC, etc.)

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

ARC EMI-ES implementation

- Of 15 operations 6 are implemented in both server and client
- “Hello World” level
- State model implementation incomplete
 - Internal states need to be more fine grained
 - 50% Job State Attributes/Flags
- Many features not implemented yet
 - Requires redesign of internal structures representing job description
 - Estimated time missing

ARC EMI-ES implementation (cont.ed)

- Implementation on track
 - Some issues need redesign/structure changes internally
 - No big obstacles
- Savannah EMI-ES implementation tasks
 - Service #22397 - deadline end Nov.
 - Client #22400 - deadline mid Dec.

CREAM EMI-ES implementation

- Source code porting from Axis-1.4 to Axis2
 - CEMon service: ongoing
 - CREAM service (legacy interface): almost done
 - performance/functionality/stability testing: TBD
- New deployment configuration based on the Axis2 guidelines needed for both the services: ongoing
- Some changes needed in the YAIM configuration for the CREAM-CE module: TBD

CREAM EMI-ES implementation (cont.ed)

- Implementation shifted due to:
 - some issues found in both EMI-ES specification and schema;
 - our requests for changes approved by the standardization group only recently (Sept 2011); They implied the revision of the EMI-ES specification and of the related schema finalized with the last version 1.03
 - https://twiki.cern.ch/twiki/pub/EMI/EmiExecutionService/EMI-ES-Specification_v1.03.odt
- As soon as the Axis2 migration of the CREAM legacy interface is completed, we can start the implementation of the EMI-ES
 - some work already done
 - java stub/skeleton generated by Axis2
 - first prototype of the EMI-ES Delegation PortType: ready
 - to be completely tested

CREAM-UI EMI-ES implementation

- The low level one is essentially strictly connected to the bindings to SOAP and exposes several low-level details. The high-level one is comparable to the SAGA API, but not following the SAGA specification; it exposes very simple and “abstract” interface.
- The work done till now:
 - **SOAP bindings for:** `ListActivities`, `GetActivityStatus`, `GetActivityInfo`, `PauseActivity`, `ResumeActivity`, `CancelActivity`, `WipeActivity`, `RestartActivity`, `NotifyService`, `GetActivityStatus`, `GetActivityInfo`, `CreateActivity`
 - A mock service that returns meaningless info, but it is useful to test the client-side correct working
- To be done:
 - `GetResourceInfo`, `QueryResourceInfo`, `*Delegation`, AUTHN communication (i.e. inclusion and setup of gsoap-plugin fwk)
 - High level API
 - Unit test (using cppunit fwk)
 - ICE adaptation to be able to talk also to the new ES
- ET: End of Oct. for low level API

UNICORE EMI-ES implementation

- Implementation tasks
 - create XMLBeans from XML
 - create web service interface classes and implementation classes
 - extend/refactor XNJS to be able to handle the EMI-ES job description
 - create client classes wrapping the raw web service interfaces
 - (?) create client (UCC) module allowing to use EMI-ES from UCC

UNICORE EMI-ES implementation (cont.ed)

- Status
 - XNJS refactoring: done
 - Core EMI-ES service implementation: mostly functional, but degree of completeness ~30%
 - Client classes: development has started
 - UCC: same problem, not as straightforward, to be addressed by the harmonization TF
 - The idea is to provide a basic client from client classes that drop out of service development and use the JAVA API in HiLA

UNICORE EMI-ES implementation (cont.ed)

- Obstacles and Plans
 - There are some issues that still need to be sorted out (might not be easy)
 - GetActivityInfo uses GLUE20
 - Can't re-use UNICORE security layer for AC
 - Activities are not individually addressable (like UNICORE jobs are)
- EMI-ES implementation to be included in EMI-2
 - Probably not same level of “production ready” as existing services

CAR TF

- Goal: define a common compute accounting Usage Record
 - Requirements:
 - Start from OGF UR V1.0.
 - Consider already existing Use Cases (such as those coming from EGI).
 - Use the OGF UR extension framework just for local (to an NGI or a specific accounting implementation) use.
 - Identify a set of extension to the existing standard as a proposal to OGF.

CAR TF (cont.ed)

- Work has been divided in the following steps:
 - Existing records survey
OK

From the survey, it turned out that DGAS, ARC, UNICORE are already capable to produce and consume OGF UR V1.0 Usage Records, yet they are all incompatible, because of different interpretation of the same fields

- Semantic definitions:
 - Base properties
OK
 - Differentiated properties

CAR TF (cont.ed)

- Semantic of the record is mainly clear and defined.
- Syntactical aspects are being investigated, but it is just a matter of properly defining correct fields attributes and their needed cardinality.
- XSD schema needs to be produced.

CAR TF (cont.ed)

- Record aggregation is mainly an open issue given the fact OGF record do not enters into the topic.
- Our proposal is to stick to what APEL is already doing in their SSM activity, just defining a proper XML based document instead of the plain text key/value pair that has been used so far
- CAR/StAR unification
 - Is that really worth?
 - Not if it requires breaking OGF UR compliance and the opportunity to give our agreement a bigger audience

Proposed Plan after parallel meeting

- Focus on developing C and JAVA APIs for job submission based on EMI-ES
 - A general design document will specify how these high-level APIs should look like, addressing general aspects and JAVA and C++
 - JAVA API: UNICORE in charge of development
 - C++ API: ARC, possibly working in cooperation with gLite
 - gLite security people will peer review the ARC library that deals with web services without high level deps (gSOAP, Globus libs, etc.)
- Work in parallel on implementation/integration of EMI-ES into the existing clients when it comes with little effort
 - ARC is implementing EMI-ES in libarcclient already
 - gLite in gSOAP or from the final APIs, to be discussed
 - UNICORE can develop a basic client out the EMI-ES server-side development
 - We'll try to uniform the look and feel, for what can be done
- A unified client could be built on top of one of our APIs anyway
 - So why not using Y3 to try and get a unified client out JAVA APIs?



Thank you!

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

Conclusions

- EMI-ES
 - Is going to become the real glue
 - Parallel framework
 - Starting point for harmonization/interoperability enabled in existing clients
 - Let's have it ASAP :-)
- CAR
 - Not sure how unification of CAR and StAR is important
- GLUE20
 - More specific profiling
 - Make sure we can really interoperate