EGI-EDGI Accounting

Adam Visegradi, Jozsef Kovacs, Sandor Acs, Zoltan Farkas
Computer and Automation Research Institute
MTA SZTAKI
H-1518 Budapest, P.O. Box 63, Hungary
{visegradi.adam,kovacs.jozsef,acs.sandor,farkas.zoltan}@sztaki.mta.hu

1 Introduction

In order to fully integrate Service Grid infrastructure (gLite [1], ARC [2], UNICORE [3]) with the Desktop Grid (BOINC [4], XtremWeb [5]) three EU FP7 projects are collaborating: EDGI [6], EGI-Inspire [7], EMI [8]. a short description of these 3 projects are following.

The European Desktop Grid Initiative (EDGI) project aims to deploy desktop grid (DG) and cloud services for EGI user communities that are heavy users of DCIs and require extremely large multi-national e-infrastructure. In order to achieve this goal software components of ARC, gLite, UNICORE, BOINC, XWHEP, ADICS, 3G Bridge [9], OpenNebula [10], Eucalyptus [11] will be integrated into SGDGCloud platforms for service provision and as a result EDGI will extend ARC, gLite and UNICORE grids with volunteer and institutional DG systems. EDGI will create novel QoS support for the DG systems and will explore new service provision models in order to ensure harmonized DGCloud interfaces to ARC, gLite, UNICORE resources. EDGI will provide a workflow-oriented science gateway to enable user communities to more easily access the EDGI infrastructure. EDGI will establish the IDGF [12] organization to coordinate DG-related activities in Europe both for solving technical issues as well as to attract volunteer DG resource donors by disseminating results of the EDGI and EGI projects. IDGF and EDGI will work in strong collaboration with EGI, EMI, NorduGrid, UNICORE Forum and interested NGIs.

The Stichting European Grid Initiative Foundation (hereafter referred to as EGI.eu) has been created under the Dutch law with the mission to create and maintain a pan-European Grid Infrastructure in collaboration with its Participants, i.e. the National Grid Initiatives (NGIs), and Associated Par-

ticipants (e.g. European International Research Organisations - EIROs) in order to guarantee the long-term availability of a generic e-infrastructure for all European research communities and their international collaborators. In its role of coordinating grid activities between European NGIs EGI.eu will: 1) operate a secure integrated production grid infrastructure that seamlessly federates resources from providers around Europe; 2) coordinate the support of the research communities using the European infrastructure coordinated by EGI.eu; 3) work with software providers within Europe and worldwide to provide high-quality innovative software solutions that deliver the capability required by our user communities; 4) ensure the development of EGI.eu through the coordination and participation in collaborative research projects that bring innovation to European Distributed Computing Infrastructures (DCIs).

The European Middleware Initiative (EMI) project is a close collaboration of the four major middleware providers, ARC, dCache [13], gLite and UNICORE. It plans to deliver a consolidated set of middleware components for deployment in EGI, PRACE and other DCIs, extend the interoperability and integration between grids and other computing infrastructures, strengthen the reliability and manageability of the services and establish a sustainable model to support, harmonize and evolve the middleware, ensuring it responds effectively to the requirements of the scientific communities relying on it.

EDGI is working on integrating the SG and DG infrastructure together with EMI and EGI.eu. The SG - DG integration has three main logical parts:

1. Seamless transfer of jobs from gLite, ARC or UNICORE to BOINC or XtremWeb-based Desktop Grid sites. This task belongs to EDGI and this integration regarding job transfer has been successfully done by EDGI. As a result modified computing elements has been developed and became part of the EMI software distribution. SLA and OLA has been signed between EDGI and EMI for further software support.

2. Monitoring Desktop Grid sites within the EGI monitoring infrastructure. This work has been done in the first half of 2012 within the framework of the EDGI - EGI MoU. EDGI developed and maintains probes for monitoring Desktop Grid. Probes are provided as RPM packages which follow EGI probe development guidelines. Currently, the Hungarian NGI is operating the nagios probes for the DG sites.

3. Accounting for Desktop Grid sites for the EGI infrastructure. This

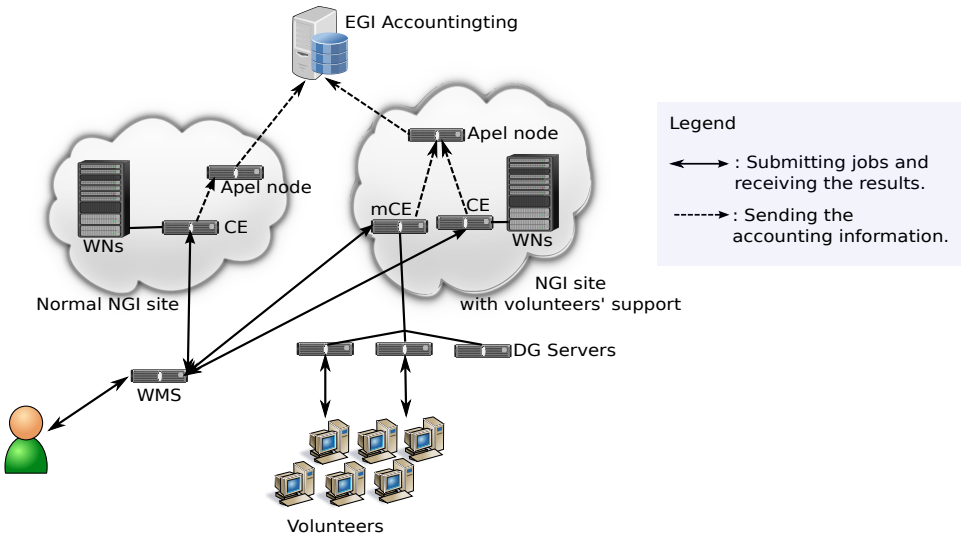

Figure 1: The big picture

work is being done in the second half of 2012 within the framework of the EDGI - EGI MoU. The goal is to create a design plan how the accounting information is going to be transferred for the EGI accounting infrastructure.

The goal of this document is to give an overview, how the accounting information can be provided by the modified computing element for gLite based onthe information provided by the Desktop Grid sites. Currently, the Desktop Grid sites are not returning accounting related information on the jobs they executing. This document as a design plan will detail alternative(s) how this accounting retrieval could be implemented. The document is organised as follows: Chapter 2 is giving the big picture i.e. details the whole infrastructure including the gLite and Desktop Grid part. It also details, how the gLite part should interface with the accounting infrastructure behind. Chapter 3 is detailing how a Desktop Grid site could extract the accounting information generated on the DG site and how it could return the required information for the modified computing element about a certain job.

2 Big Picture

Figure 1 presents the normal and the designed accounting architecture in high abstraction level. The registered user of the NGI can submit jobs into normal Service Grid (SG) resources via the Workload Management System (WMS).

- In case of normal SG (e.g gLite):

  The WMS sends the tasks to Computing Elements (CE). The CE pushes the jobs into the Worker Nodes (WN). The WNs process the jobs and send back the results to the CE. The CE makes the results available for the WMS and generates and stores the related accounting informations.

- In case of EDGI-like infrastructure:

  The WMS sends the tasks to modified Computing Elements (mCE). The mCE pushes the jobs into the Desktop Grid (DG) servers. The desktop machines of the volunteers fetch the Work Units (WU), process and send back the results through the mCE.

The user fetch the outputs via the WMS. Periodically, the (m)CE checks the accounting information and synchronizing with the site level Apel [14] node. The Apel node regularly sends the accounting information of the site to the central EGI accounting server.

Table 1 shows the required information from individual jobs by the accounting system.

3 Integration of Desktop Grid accounting

3.1 Required information

Most of the information about jobs, required by the APEL infrastructure, is available at the modified computing element. The actual accounting information about them, however, can only be provided by the desktop grid. The accounting metrics the desktop grid has to provide to the modified computing element are summarized in Table 2.

Most of the metrics specified are readily available in the desktop grid database; however, there are two exceptions.

Key	Value	Description	M.
Site	str	GOCDB sitename	Yes
SubmitHost	str	Head node	Yes
LocalJobID	str	Batch System Job ID	Yes
LocalUserID	str	Local username	No
GlobalUserN.	str	User's X509 DN	No
VO	str	User's VO	No
GridGroup	str	Users VO Group	No
Role	str	Users VO Role	No
WallDuration	int	Wallclock time for the job (s)	Yes
CpuDuration	int	CPU time for the job (s)	Yes
Processors	int	Number of processors	No
NodeCount	int	Number of nodes	No
StartTime	int	Start time of the job (epoch time)	Yes
EndTime	int	Stop time of the job (epoch time)	Yes
MemoryReal	int	Memory consumed by job (kbytes)	No
MemoryVirt.	int	Virt. mem. consumed by job (kb)	No
ScalingFactorU.	str	HepSpec06, SpecInt or custom	Yes
ScalingFactor	doub	Value of HepSpec, SpecInt or custom	Yes

Table 1: Required information from individual jobs

Memory

The memory consumed by the job is not recorded by the desktop grid. On the other hand, for each application, an upper limit for memory consumption must be estimated. This information is stored as an attribute for each workunit, and can be used by clients to filter out workunits that would require too much memory. If a running workunits exceeds its limit, it will be immediately terminated.

Because of the termination policy, it is guaranteed, that the memory consumed by the workunit will be at most that estimated. Because of the filtering policy, application developers are forced to provide tight estimates. Therefore, the limit associated with the workunits is a suitable estimate for the memory consumption of the job.

Start/stop times	Time when the job has been started, and when it has finished
Wallclock time	Total time the job has spent running (= stop time − start time)
CPU time	Consumed CPU time measured by the kernel on the executing host
Memory	Real and virtual memory consumed by the job
Benchmark values	Constant factors describing the performance of the executing host
Number of CPUs	Number of CPUs on the executing host

Table 2: Accounting metrics provided by the desktop grid

Benchmark values

Only the number of floating point operations per second (FLOPS) is provided as information about hosts' performance. The current APEL records store SpecInt2K and SpecFloat2K factors (although this seems to be changing[1]). To integrate the desktop grid with the APEL accounting system, we have to find a reliable mapping between these two values.

As the specification of APEL seems to be changing, the simplest and best solution is that the desktop grid will only provide the raw FLOPS value to the computing element. Then, the computing element will be able to map the raw value to a suitable number as necessary.

The parameters listed in Table 1, but not listed in Table 2 are filled up by the computing element as it is performed by any non-modified gLite CREAM CE.

3.2 Accessing information

The modified computing element communicates with the desktop grid via a web-service interface of the 3G Bridge (Figure 2). The current interface

[1] https://twiki.cern.ch/twiki/pub/EMI/APELClient/APEL-Messaging-v2.2.pdf

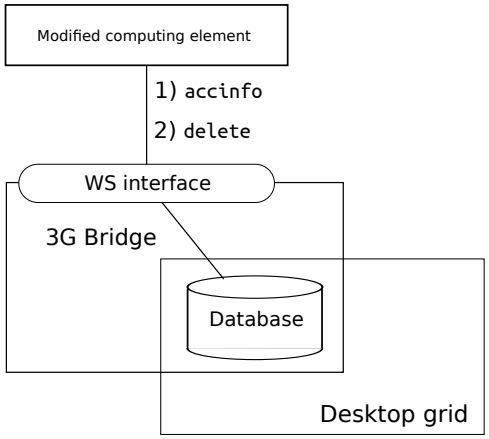

Figure 2: The EDGI Desktop Grid architecture

does not define a way to access detailed information about a job; therefore, a new method (`accinfo`) must be defined in the interface.

The 3G Bridge itself cannot provide the accounting information needed, only its back-end desktop grid can. However, the desktop grid database and the 3G Bridge database are physically the same; that is, the 3G Bridge web-service can access it and extract the needed information from it *directly*. This means, that the interface between the 3G Bridge and the back-end desktop grid can be left intact.

The computing element must call `accinfo` with a job identifier as an argument. As a result, the information specified in Table 2 is returned as key–value pairs, in a clean text format.

The 3G Bridge does not employ a garbage collection system, the client side—here, the modified computing element—has to explicitly delete a job after execution. After deleting the job, no information about it will be accessible—including the accounting information. Because of this, the specification of this interface must include that the `accinfo` must always be called *before* `delete`.

4 Adding accounting handling to mCE

4.1 The current mCE

The goal of the mCE was to create a solution that requires only very few modification in the existing CREAM code. Fortunately, we were able to find

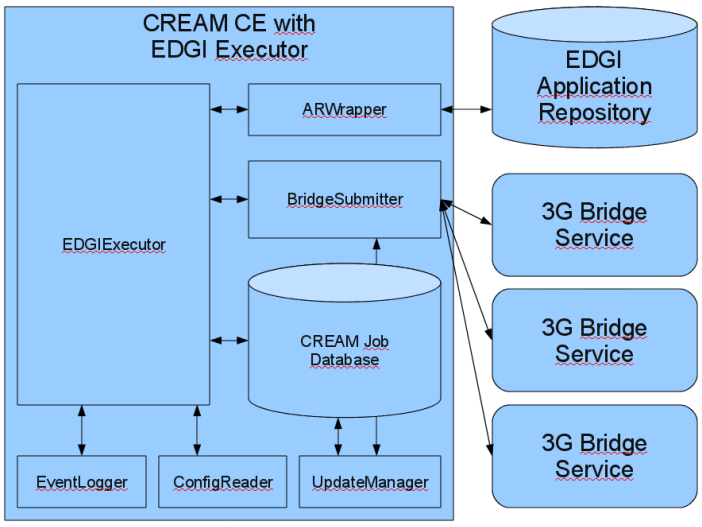

Figure 3: EDGI Executor and related components overview

a solution that needs no modification at all. The idea behind the solution is to create a new EDGI connector beside the BLAH connector, whose task is to intercept gLite jobs, and send them to a target 3G Bridge service after parsing incoming gLite jobs. Thus, from higher level CREAM component's point of view the new connector (EDGI Executor) behaves like a batch system implementation with the difference that the job is not run on a Worker Node belonging to the CREAM CE, but is sent to a 3G Bridge service for execution by a grid plugin.

The EDGI Executor consists of a number of components as shown in Figure 3: a ConfigReader, an EDGIExecutor, an ARWrapper, a BridgeSubmitter, an EventLogger and an UpdateManager.

4.2 The new component in mCE

In order to seamlessly integrate EGI accounting, the mCE functions should be extended. The enumeration below shows the current and the new functions of mCE in nutshell:

Receive job during this phase the mCE receives the job either from the WMS or directly from the user. A worker thread is invoked to manage the job

Prepare inputs this stage prepares the input files belonging to the job. All

the local input files are exposed through a web server for the desktop grid

Submit to LRMS once the input files are ready, the job is submitted to the desktop grid through the 3G Bridge component

Get the status an update thread of the mCE periodically updates the status of jobs submitted to the desktop grids and reports any job status changes

Get the output if a job has finished, its outputs are fetched from the desktop grid server and are published as requested (are either sent to the WMS or are placed under a GridFTP location specified by the user). Additinally, the job is cleared from the desktop grid

Get accounting informations and insert it site Apel node (as Section 3 describes)

Clear after the job's outputs have been handled, the job is cleared from the mCE

The gathered accounting information can be stored in a prepared database or in XML file format. In the mCE, the Apel client periodically checks the changes and synchronizes the database with the site Apel node. The site Apel node regularly sends the accounting information to the central EGI accounting system. (As we described in Section 2).

5 Summary

In this document we gave a rough overview of the planned accounting system for jobs executed on DG resources. This design plan has the following logical blocks:

- Exracting the account information about the workunits on the DG sites. This requires an extra fuctionality in the 3G Bridge component. See section 3.
- Providing an interface for the mCE to be able to query the account information from the DG sites. This can be done through an updated 3G Bridge web service interface. See section 3.
- Integrating the account query into the job handling flow of the EDGI Cream Computing element. See section 4.

- Forwarding the account information of the jobs to the APEL accounting system. See section 4.

References

[1] gLite grid middleware. http://glite.cern.ch/, 2012.

[2] ARC grid middleware. http://www.nordugrid.org/arc/, 2012.

[3] Unicore grid middleware. http://www.unicore.eu/, 2012.

[4] Boinc desktop grid middleware. http://boinc.berkeley.edu/, 2012.

[5] XWHEP desktop grid middleware. http://www.xtremweb-hep.org, 2012.

[6] European Desktop Grid Initiative (EDGI). http://edgi-project.eu/, 2012.

[7] European Grid Infrastructure (EGI). http://www.egi.eu/, 2012.

[8] European Middleware Initiative (EMI). http://www.eu-emi.eu/, 2012.

[9] Generic Grid-Grid (3G) Bridge. http://www.lpds.sztaki.hu/products/3g-bridge, 2012.

[10] OpenNebula cloud. http://opennebula.org, 2012.

[11] Eucalyptus cloud. http://www.eucalyptus.com/, 2012.

[12] International Desktop Grid Federation (IDGF). http://desktopgridfederation.org, 2012.

[13] dCache. http://www.dcache.org/, 2012.

[14] Apel. https://wiki.egi.eu/wiki/APEL, 2012.