
# DSA1.7 REPORT ON IMPLEMENTATION OF LOGGING SYSTEM AND ACCOUNTING POLICIES

| | |
|---|---|
| Document Filename: | **BG-DSA1.7-v2.1-Logging_Systems_and_Accounting.doc** |
| Activity: | **SA1** |
| Partner(s): | **EENet, NICPB, PSNC, IFJ PAN** |
| Lead Partner: | **EENet** |
| Document classification: | **PUBLIC** |

Abstract: Logging and accounting are of crucial importance when creating a production-level computing Grid. This document provides an overview of BalticGrid logging and accounting systems available currently and also describes the future plans

**Document review and moderation**

| | Name | Partner | Date | Signature |
|---|---|---|---|---|
| Released for moderation to | Marcin Radecki | IFJ PAN | 21/11/2006 | |
| Approved for delivery by | | | | |

**Document Log**

| Version | Date | Summary of changes | Author |
|---|---|---|---|
| 0.1 | 16/11/2006 | First version | Lauri Anton, Hardi Teder, Michal Jankowski, Mario Kadastik |
| 0.2 | 17/11/2006 | Information about logging added | Lauri Anton, Michal Jankowski |
| 1.0 | 20/11/2006 | Future plans added | Lauri Anton, Hardi Teder, Michal Jankowski, Mario Kadastik, Marcin Radecki |
| 2.0 | 21/11/2006 | Final version | Lauri Anton, Hardi Teder, Michal Jankowski, Mario Kadastik, Marcin Radecki |
| 2.1 | 22/11/2006 | Final version with comments from reviewers | Lauri Anton, Hardi Teder, Michal Jankowski, Mario Kadastik, Marcin Radecki |

# Contents

## GLOSSARY

| | |
|---|---|
| APEL | Accounting Processor for Event Logs |
| BG | Baltic Grid |
| CE | Computing Element |
| CESGA | Supercomputing Center of Galicia |
| DGAS | Distributed Grid Accounting System |
| GGF | Global Grid Forum |
| GMA | Grid Monitoring Architecture |
| JRA | Joint Research Activity |
| LBS | Local Batch System |
| LCG | LHC Computing Grid |
| LHC | Large Hadron Collider |
| LSF | Load Sharing Facility |
| PBS | Portable Batch System |
| R-GMA | Relational Grid Monitoring Architecture |
| SGE | Sun Grid Engine |
| SQL | Structured Query Language |
| SLA | Service Level Agreement |
| TORQUE | Terascale Open-source Resource and QUEue manager |
| VO | Virtual Organization |
| WN | Worker Node |

# 1. INTRODUCTION

The grid services offered for science and development are becoming more and more mature. Still, the services are for free, partly because grids are mostly used in academic environments as well as due to the fact that the accounting systems are not ready yet.

This situation has to and will change, because there are costs associated with keeping computing resources running, employing grid administrators, etc. In the future, the grid usage will be charged as any other service.

There is need for systems, which logs all the events occurring and enable the accounting of activities performed using grid resources.

The information collected by logging system can be used for several reasons:

- Bookkeeping
- Proving the resource usage
- Solving stability problems
- Solving security incidents

Accounting extracts information from logged data and gives overview of the usage of the systems.

This report gives overview of the logging system and accounting tools used in BalticGrid:

- Section 2 gives overview of logging data harvesting systems in grids and status of logging facilities in BalticGrid
- Section 3 describes the accounting tools used in BalticGrid
- Section 4 gives directions for future development on logging and accounting area

# 2. LOGGING SYSTEMS

## 2.1. LOGGING AND ACCOUNTING FACILITIES

Any production grid needs the accounting feature. Before issuing any bills, however, the accounting data must be collected (possibly from several locations) and tied to users and VOs. To do this the accounting system utilizes local logging facilities.

In general case of logging one can also take a look from the point of view of security, as it is important to track user activities (e.g. by analyzing system logs) in order to detect any rule breaking. It must be possible to identify the user who has performed a particular action.

Both of the above require proper encapsulation of jobs (e.g. running jobs of different users on the same account should be not allowed) and the history of mappings: global user identity (Distinguished Name) – local identity (Unix account) must be stored.

The detailed requirements are as follows:

- It must be possible to identify the user and context (VO, role, capabilities, time) of any action that is subject to local logging or accounting. Thus it should be possible to fetch all the accounting and logging information from the standard system mechanisms (such as e.g. Unix accounting). In many applications it may be a sufficient amount of data.
- To further extend the system with possibilities of future use, the system should be capable of storing non-standard accounting data. This is especially useful when the node provides access to non-standard resources, e.g. specific equipment in a virtual laboratory with remote access through grid. The data may be put by any external application through a special API. However this is not a priority at this point.
- The accounting data must be easily accessible for all parties involved in a particular action. The resource provider wants to access all information connected with resources it provided to a grid. A virtual organization would like to see all information on resources used by its members so that VO can make internal decisions according to that information. The end user who submitted some jobs would like to access all information on resources consumed by those jobs.
- The accounting mechanism must be able to gather the accounting data for the VO as well as for the groups of resources.
- If the accounting system is not used only for gathering some statistics, but for charging the users and institutions, reliability and security issues are of the highest importance. [2]

## 2.2. STANDARDS

The architecture of any kind of logging and accounting system should be modular and flexible. It should give a chance for easy integration with the existing solutions and standards. The modularity embraces plug-in based authorization, and possibility of easy integration with existing accounting services and tools.

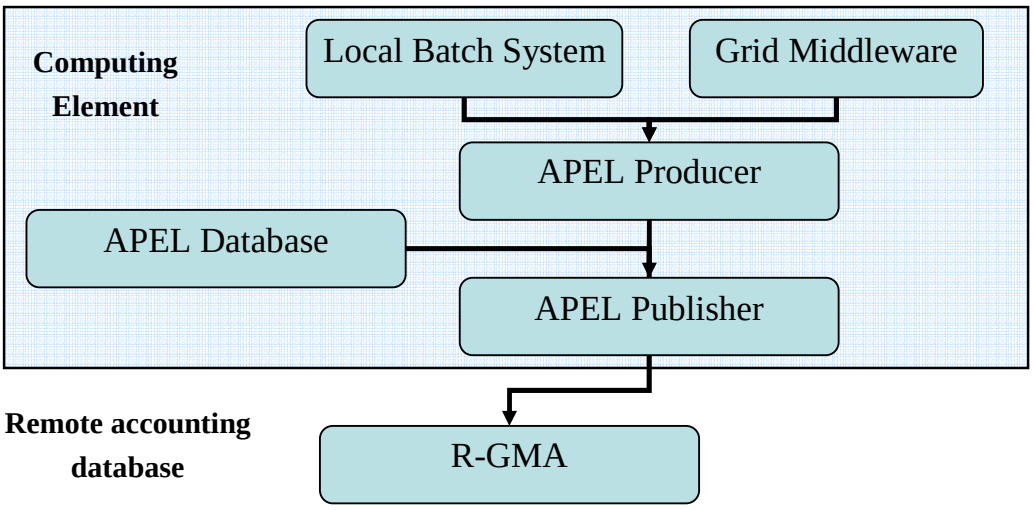
A special group of standards widely used by Grid computing are web services (WS). Web services provide standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks [8]. This high interoperability is achieved by wide usage of XML-based languages both for interface description (WSDL) and communication (SOAP). Another virtue is extensibility characteristic to XML. Due to the textual character, the WS communication may use (and

uses) popular standards as the transport layer (HTTPS), which allows reusing the already implemented parts and simplifies security configuration. Thus using the WS interface is recommended.

For resources to be shared, sites must be able to exchange basic accounting and usage data in a common format. This format must encompass both job level accounting, and aggregate accounting. The GGF Usage Record Working Group [7] focuses on that problem. Usage Record standard describes a common format for exchanging resource usage data, but doesn't affect the way the data is gathered and stored. The Usage Record format is based on XML, so it naturally fits into SOAP messages. There are also services that will accept this format (e.g. DGAS).

## 2.3. CURRENT LOGGING SYSTEM IN BALTICGRID

Current logging and accounting system in use in BalticGrid is a passive system which allows only to review the usage of resources after the usage has already occurred. There is no active component on a global scope to determine and enforce accounting policies and limits. Current logging system in BalticGrid consists of following components: *Local Batch System logs, Grid Middleware logs, APEL* and *R-GMA.*




### 2.3.1. Logging in Local Batch System and in Grid Middleware

Job scheduling and management in clusters is provided by some local batch system (LBS). In BalticGrid, most widely used LBS is OpenPBS/TORQUE. In general, the LBS takes care of job scheduling, execution, output management and also logging of job's activities. It also allows for user or VO prioritization as well as access control to resources based on different queues and limits. For example a site dedicated mostly to one specific VO might allow this VO users access to all of the resources while providing best-effort access to other VO-s on a smaller scale. Usually such policies are enforced on the LBS level. This is also currently the only active part in accounting enforcement, which allows sites to scale the resource usage according to local or pre-agreed terms with VO-s. However as this is only a locally configurable system, then its efficiency on a global scale is small.

The LBS logs various events in the system for every scheduled job: user name, job start time, job end time, used resources, etc. This is the information which is useful for passive accounting.

Grid Middleware logs are used for connecting remote grid user ID-s to local jobs.

### 2.3.2. APEL – Accounting Processor for Event Logs

In BalticGrid the centralized accounting is done using APEL. APEL is a log processing application which is used to interpret grid middleware and batch system logs to produce accounting records. In APEL the local APEL Producer is used to gather information from various batch systems. Currently supported batch systems are PBS/Torque, LSF, Condor and SGE. After the data is collected from LBS, APEL Publisher sends the data to a central R-GMA database.

The basic functionality of APEL is controlled through periodic execution of scripts, which run APEL Producer once per day. During that execution APEL Producer parses the LBS system logs to collect information of jobs which have started/completed within the last period (usually 24 hours) and converts it to a standardized format. It also queries the site local GIIS for information about the resource specifics on which the respective jobs ran. The information gathered from LBS and also Grid middleware logs like gatekeeper logs with also the resource specifications is kept in a site local database.

By default the producer parses each log file only once. But that way, it could not account for jobs which start time is logged in one log file and the end time in the next day's one. In BalticGrid, we require from site administrators that they configure the Producer to run through all log files every night.

Usually once per day the APEL Publisher is used to synchronize this information in local database to a central database, which then propagates this information to R-GMA. This publishing also includes basic integrity checking to validate that all the accounting records are propagated to the central repository. It also has the option to republish all of the locally available accounting data to compensate for situations where the central repository was unavailable at the time of publishing thus eliminating lost records.

### 2.3.3. R-GMA – Relational Grid Monitoring Architecture.

R-GMA stands for Relational Grid Monitoring Architecture. It is a simple implementation of GMA which is defined by the GGF (Global Grid Forum) as a way to monitor Grids. The basic operations of R-GMA is through the use of producers of information (like the APEL Producer) and consumers who require such information. In the general form of GMA the location of producers who can provide the information to consumers is done through a registry. In R-GMA this is hidden and done behind the scenes by providing an SQL like interface to the consumers to query for the information they require.

The consumers are typically systems which provide different views to the end users of the information stored in R-GMA. In the context of this document the information of interest is that of accounted use of resources. There is a multitude of ways to access such information. The most basic and robust interface is the direct access R-GMA client, which is installed with every R-GMA client installation (e.g. APEL client/producer) on clusters. It is a command line tool, which provides direct SQL like access to the information. For a sample session of R-GMA command line tool look into section 3.1.1.

# 3. ACCOUNTING

Accounting extracts information from logging data and gives overview of the usage of the systems. At this moment accounting is only used for reporting the usage of the systems, but in the future the grid users will be charged according to the cost of used resources, which implies a system capable of both logging the user activities through accounting as well as enforcing policies set by the user or some other central system.

The need for a billing system and to bill users of resources in general comes from the simple fact that there are costs associated with CPU, storage, bandwidth, software usage as well as with support actions. There is thus an urgent need to develop models for a Grid economy that support the allocation and accounting of these resources and that can be deployed on Grids of varying sizes.

Today we use for accounting both EGEE tools by CESGA and also an accounting portal for BalticGrid provided by Polish partners, which was made in the BalticGrid framework. In the near future it is expected that the systems of APEL and DGAS will join. If that happens, then BalticGrid will also migrate to that system, as it should provide in addition to passive accounting also the option to enforce accounting policies actively.

## 3.1. ACCESS TO ACCOUNTING INFORMATION

To access the accounting information one has to query the central R-GMA repository. To do that a number of tools and views have been developed by a multitude of collaborations and projects. A few samples are listed below.

### 3.1.1. Command line view

The command line view can be accessed from any system, which has a R-GMA client installed. For example all computing elements have it installed and hence they provide the site admin access to the R-GMA repository. A sample session follows:

```
> rgma
Welcome to the R-GMA virtual database for Virtual Organisations.
================================================================
Your local R-GMA server is:
  https://titan.hep.kbfi.ee:8443/R-GMA
You are connected to the following R-GMA Registry services:
  https://lcgic01.gridpp.rl.ac.uk:8443/R-GMA/RegistryServlet
You are connected to the following R-GMA Schema service:
  https://lcgic01.gridpp.rl.ac.uk:8443/R-GMA/SchemaServlet
Type "help" for a list of commands.
```

```
rgma> describe LcgRecords
+--------------------+--------------+-------------+-------------+
| Column name        | Type         | Primary key | Can be NULL |
+--------------------+--------------+-------------+-------------+
| RecordIdentity     | VARCHAR(255) | Yes         | No          |
| ExecutingSite      | VARCHAR(50)  | No          | Yes         |
| LocalJobID         | VARCHAR(50)  | No          | Yes         |
| LCGJobID           | VARCHAR(50)  | No          | Yes         |
| LocalUserID        | VARCHAR(50)  | No          | Yes         |
| LCGUserID          | VARCHAR(255) | No          | Yes         |
| LCGUserVO          | VARCHAR(50)  | No          | Yes         |
| ElapsedTime        | VARCHAR(30)  | No          | Yes         |
| BaseCpuTime        | VARCHAR(30)  | No          | Yes         |
| ElapsedTimeSeconds | INTEGER      | No          | Yes         |
| BaseCpuTimeSeconds | INTEGER      | No          | Yes         |
| StartTime          | VARCHAR(30)  | No          | Yes         |
| StopTime           | VARCHAR(30)  | No          | Yes         |
| StartTimeUTC       | VARCHAR(30)  | No          | Yes         |
| StopTimeUTC        | VARCHAR(30)  | No          | Yes         |
| StartTimeEpoch     | INTEGER      | No          | Yes         |
| StopTimeEpoch      | INTEGER      | No          | Yes         |
| ExecutingCE        | VARCHAR(50)  | No          | Yes         |
| MemoryReal         | INTEGER      | No          | Yes         |
| MemoryVirtual      | INTEGER      | No          | Yes         |
| SpecInt2000        | INTEGER      | No          | Yes         |
| SpecFloat2000      | INTEGER      | No          | Yes         |
| EventDate          | DATE         | No          | No          |
| EventTime          | TIME         | No          | No          |
| MeasurementDate    | DATE         | No          | No          |
| MeasurementTime    | TIME         | No          | No          |
+--------------------+--------------+-------------+-------------+
```

```
 rgma> select ExecutingSite from LcgRecords where LCGUserVO='balticgrid' group by
ExecutingSite;

+--------------------+
| ExecutingSite      |
+--------------------+
| AMD64.PSNC.PL      |
| CYF-CERTIF-TB      |
| CYFRONET-IA64      |
| CYFRONET-LCG2      |
| EENet              |
| egee.man.poznan.pl |
| IMCSUL             |
| IMCSUL-INF         |
| ITPA-LCG2          |
| KTU-ELEN-LCG2      |
| PDC                |
| RTUETF             |
| T2_Estonia         |
| VDU-IF-LCG2        |
| VU-MIF-LCG2        |
+--------------------+
15 rows
```

The first query explains the information available in the virtual table LcgRecords which contains all of the accounting records. It is a useful way of seeing what information is available and in what format it is stored. The second query is a basic SQL query, which asks for all sites which have executed jobs from users who belong to BalticGrid VO.

As can be seen the basic interface is very robust and requires extensive knowledge of underlying information as well as knowledge of SQL and is in such a form not acceptable to the common public or for more complex views. To allow for higher-level views a number of web portals have been devised within different projects.

### 3.1.2. Accounting portal for BalticGrid provided by IFJ PAN.

For a period of time partners from Poland deployed and customised a web based portal to query complex information from R-GMA on accounting through a simple web forms interface. However this portal did not provide accurate information due to many binding problems with R-GMA and has since been removed from operations back to development. It is the wish of the collaboration to have such a personalized tool for BalticGrid resources to better view and display the information available in the accounting portal for our specific region. It is hoped that this specific view will be restored within a reasonable amount of time to be used actively in the second reporting period of this project.

### 3.1.3. CESGA view

The one most commonly used accounting view in EGEE is the CESGA view. It is a web-based portal, which contains static information of accounting information, which is updated once per day. It provides historic information of resource usage by sites, by VO-s. It shows the information with a minimum granularity of one month per site per VO. It also provides historic evolution graphs to see the behaviour and utilization of the resources as well as evolution in resources used (for example the

scaling factor used which best describes the power of CPU-s used at any site). A few sample views can be seen in Figures 1 and 2.

The address of the CESGA view is:

http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee.php

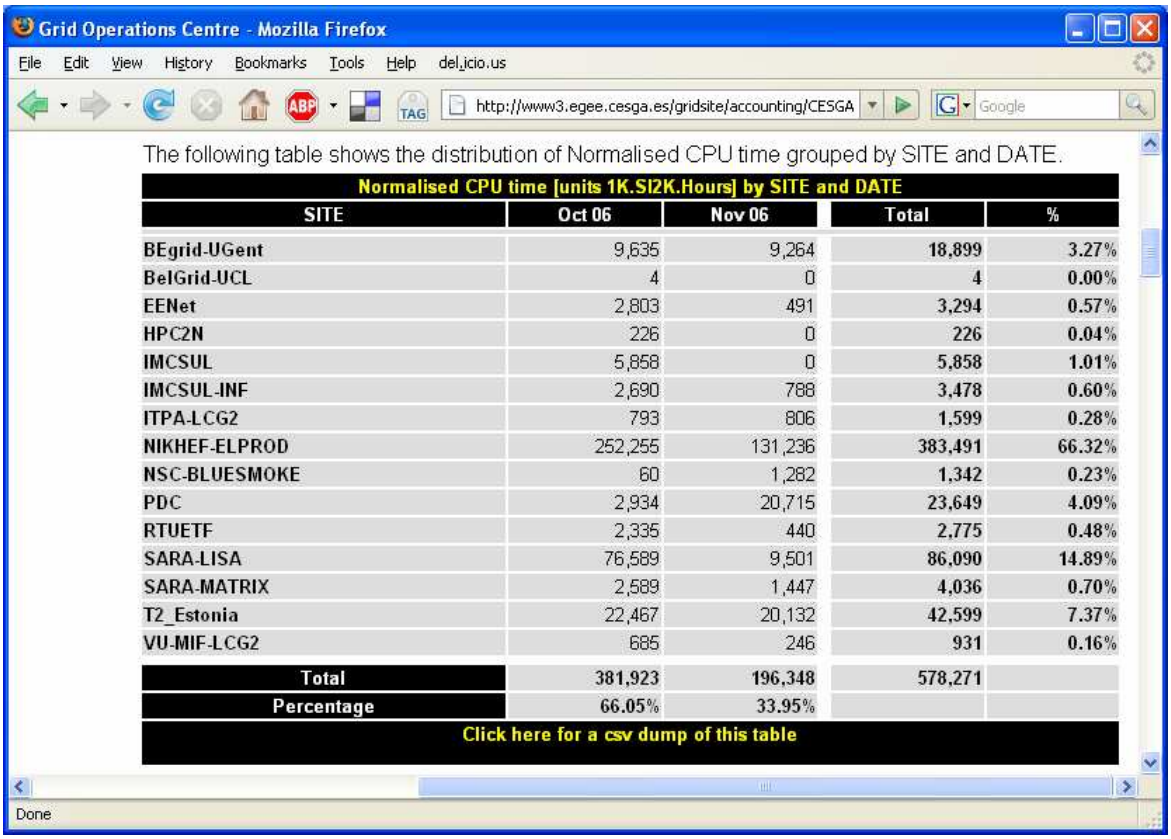



**Figure 1 CESGA view, Normalized CPU time per site and date, October – November 2006 in NorthernEurope federation.**

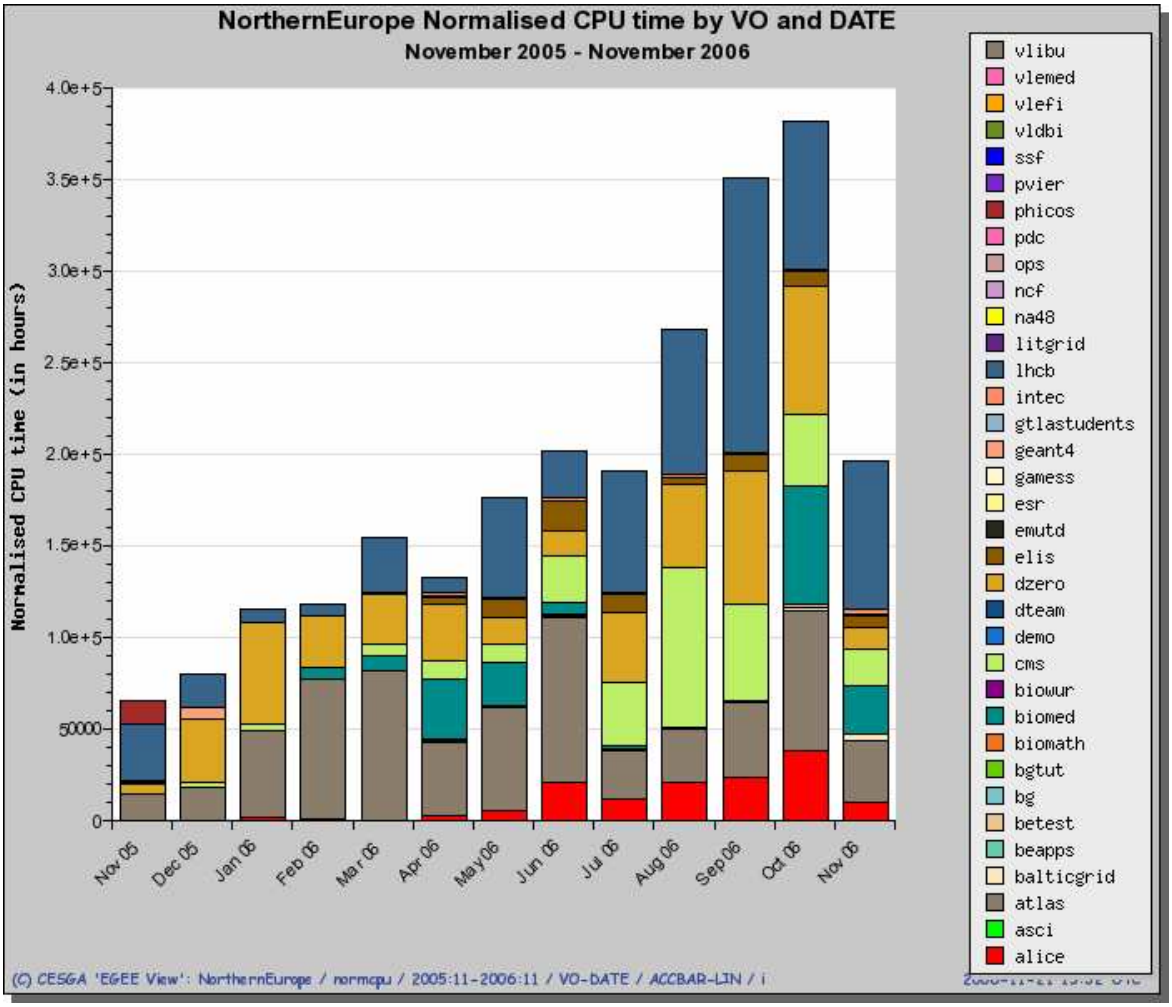



**Figure 2 CESGA view, NortherEurope Normalized CPUtime by VO and DATE for November 2005 – November 2006**

## 3.2. ACCOUNTING POLICIES

Accounting is one of the current key issues when deploying Grids for production. So far, Grids have primarily supported homogeneous user groups with dedicated resources who have joined forces in larger collaborations to exchange and share resources in an open and free manner. Grids connecting general purpose resources and thus requiring accounting policies, procedures and reporting have been much less common. One of the main reasons for this is the lack of general allocation and accounting policies enabling the development of standardized mechanisms for an economy for the Grid [3].

An accounting policy is a set of rules or guidelines to allow certain groups of people access to resources in a controlled manner. That means in a way where the site administrator can limit access on some level of granularity as well as the group itself. The main need for such policies and their enforcement is the lack of resources for everyone to perform their computations as well as the actual costs of operating such resources.

Standard scheduling systems as well as operating systems commonly have built-in accounting systems to track resource usage. However, they often assume a homogeneous run-time environment, and they lack standard and uniform ways to obtain and represent information from several heterogeneous resources. Thus there has been a strong need for Grid accounting systems that integrate local accounting solutions similarly to the way Grid meta-schedulers and co-allocation managers coordinate, and administer job submissions across several schedulers [3].

A prerequisite to accounting policies is the knowledge of used resources. So for any kind of active policy enforcement there must be a way first to collect the existing records and provide a uniform and standard way to present and verify these numbers. As long as Grids remain for specific use only by centres, which come with their own resources and join forces, there is no need to be able to certify what the resource usage numbers are. But as soon as a price tag is associated with a resource usage number, users will want proof of use.

Also in case of actual billing for use of resources users will want the ability to pre-emptively limit the maximum amount of resources spent on different granularity levels to make sure they are actually capable of paying for the resources used. It is namely very easy to accidentally submit a huge number of jobs which take up a lot of resources while the whole incident could be just a small typo in the original submission process (f.e. setting number of jobs to 10 000 instead of 1000).

Considering the current experience with Grid accounting, which is already included with the currently used middleware, we can conclude that accounting and metering as passive tools are already there and functional although there are many areas of potential improvements. One example is enabling of user level accounting instead of VO level accounting to allow for VO-s to better assess and control their members. However any kind of accounting system which goes down to user level with fully identifiable person and his behaviour pattern must also assess all of the implications with privacy laws throughout the region where this kind of logging and accounting is enabled as well as where the collected information is kept. Also access control must be enabled on all of the possible views to this kind of information to only allow authorized users to see the usage records on a per person level.

As at this moment there are no tools implemented to actually allow for policy enforcement, except the very basic tools of limiting access to resources on a queue level at LBS, then there is also no way to currently enforce any kind of policies in BalticGrid. Once tools become available to enable policy enforcement in the Grid middleware or as add-on toolkits, we will try implementing them.

# 4. FUTURE IMPROVEMENTS

There is a multitude of possible improvement options in improving both the current logging system as well as improving the accounting system up to the level of active policy enforcement.

## 4.1. LOGGING AND ACCOUNTING SYSTEM IMPROVEMENTS

The main technical improvements can come from improving the current logging and accounting system. Some of the possible future improvements listed below.

### 4.1.1. Per-user basis logging

The current logging and accounting system does not exploit the per-user basis granularity. It only supports up to the level of VO-s. The Producer side does support a per user based logging, but to to enable it and start distributing it needs also intervention from the central database system part as well as serious work on the privacy issues to make sure no private information is available to the general public. Either in transit or stored centrally.

It also requires creation of new views to take into account the new granularity and these views must also allow access based on a stricter policy. Probable access levels include:

- The user himself should be allowed to view his usage records
- The VO to which the user belongs should be able to see the records of all of its members in detail
- A site administrator should be able to see who have been the users using his site (he could get that information from the local log files anyway)

### 4.1.2. APEL and DGAS converging

APEL is the currently used accounting system in EGEE, but does not support the concepts of Grid economy, e.g. it is not possible to account and charge the cost of usage per-user basis. It does on the other hand provide logging facilities, which are already in place.

DGAS is a research project funded through EGEE-II JRA1 activity and is focusing on creating an accounting system, which incorporates all of the necessary components for a production quality Grid economy. It already takes into account the per-user level logging as well as signing of every log entry with the user proxy to make sure that the records are really created for that user and by that users jobs, it also has the ability to limit the amount of resources used through a centralized system where a users credits are checked before general job submission by the Workload Management System (WMS).

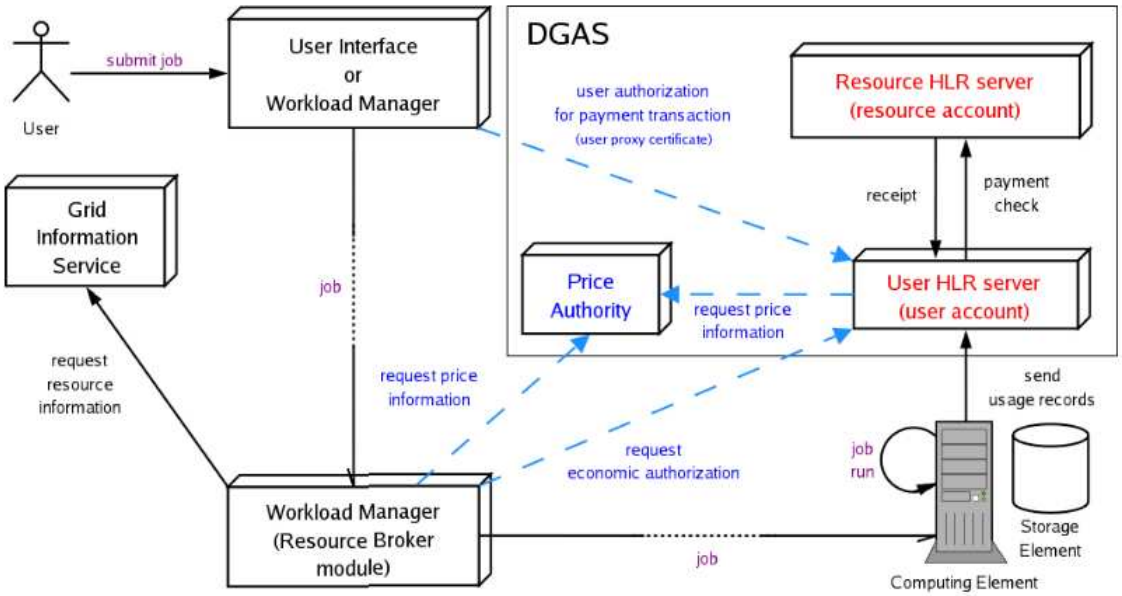



**Figure 3 - Job submission procedure with DGAS accounting in place, which confirms availability of credits before job submission [9].**

To our knowledge APEL and DGAS are converging towards a uniform solution, where the finer grain accounting (user level, signed by user and confidential) data is kept in DGAS and more general aggregated accounting information in APEL to allow for a smoother migration.

### 4.1.3. Tycoon

A different approach to accounting is presented by Tycoon, a system for enforcing SLA-s, to be deployed in BalticGrid by the KTH partner:

- a central Bank pre-allocates credits (virtual money) for the users
- the client agent (kind of broker) selects resource and transfers some credits to the resource in the Bank
- the price for the resource is based on the market (more users compete for the resources, the price is higher)
- SLA is enforced by virtual machine mechanisms, that allows for using strictly defined fractions of the physical resources while creating the virtual ones
- the user who requested the job run will be mapped to the root accounts of the virtual machines of virtual cluster.

In this model, central Bank is needed for pre-allocation of funds and probably there is no sense for gathering the detailed accounting (separate CPU, memory, etc.), as the user is charged a-priori and may manipulate the local accounting (inside the virtual cluster) as the root. With Tycoon, the SLA and quota enforcement can be achieved.

Tycoon is a bit heavy-weight in deployment, as it requires Xen virtual manager on CE-s and WN-s as well as some services are duplicated with gLite (e.g. WMS). In fact, gLite software would be deployed inside Tycoon virtual clusters. The current potential future plan of the SLA infrastructure will be

probably to deploy only on a few BalticGrid sites, as some sites may not want to provide resources for Tycoon installation.

## 4.2. IMPLEMENTATION OF RECOMMENDATIONS FROM JRA1 ACTIVITY

The JRA1 activity is currently working on possible accounting systems to find the most suitable one for deployment in BalticGrid. The work is done in cooperation with NA4 activity to maintain a level of standardisation.

Once the recommendations and prototype software are available for deployment from the research activity, the Grid infrastructure will test it in its testbed. Upon a successful test the accounting system will be deployed also on production Grid depending on the accounting systems structure and the need of the community.

## 5. CONCLUSIONS

In this document we have presented the current situation with logging and accounting in BalticGrid. The basic result is that at this point no special additional software to that of EGEE provided middleware is used. The resulting logging and accounting infrastructure provides adequate results for passive accounting to review resources usage and plan for the future. All the sites support accounting and provide reliable data to R-GMA.

In the next phase, we will try to implement per-user accounting and also a system, which allows enforcement of SLAs and quotas. A more thorough report on possible accounting systems, which could be used by the BalticGrid, is being investigated by the JRA1 activity whose results will be presented in the deliverable DJRA1.5.

## 6. REFERENCES


| | |
|---|---|
| [1] | Grid Log Retention Guidelines , https://twiki.cern.ch/twiki/bin/view/LCG/LogRetention |
| [2] | BalticGrid DJRA1.1, http://www.balticgrid.org/Results/Deliverables_final/BG-DJRA1.1-v0.9.pdf |
| [3] | Grid Economy - Allocation and Accounting , http://wiki.e-irg.org/bin/view/Main/GridEconomyAllocationAndAccounting |
| [4] | Grid Log Retention Guidelines , https://twiki.cern.ch/twiki/bin/view/LCG/LogRetention |
| [5] | Using DGAS to Feed Accounting Information into APEL, http://goc.grid-support.ac.uk/gridsite/accounting/dgas.html |
| [6] | APEL: An implementation of Grid accounting using R-GMA, http://www.r-gma.org/pub/ah05_364.pdf |
| [7] | Usage Record Working Group, http://www.psc.edu/~lfm/PSC/Grid/UR-WG/ |
| [8] | *Web Services Description Requirements, W3C Working Draft*, http://www.w3.org/TR/ws-desc-reqs/, 28 October 2002 |
| [9] | DGAS, http://www.to.infn.it/grid/accounting/ |


**Table 1 References**