An APEL Tool Based CPU Usage Accounting Infrastructure for Large Scale Computing Grids

Ming Jiang, Cristina Del Cano Novales, Gilles Mathieu, John Casson, William Rogers, John Gordon

{ming.jiang, cristina.del-cano-novales, gilles.mathieu, john.casson, will.rogers, john.gordon}@stfc.ac.uk

e-Science Centre, Science and Technology Facilities Council, United Kingdom

Abstract The APEL (Accounting Processor for Event Logs) is the fundamental tool for the CPU usage accounting infrastructure deployed within the WLCG and EGEE Grids. In these Grids, jobs are submitted by users to computing resources via a Grid Resource Broker (e.g. gLite Workload Management System). As a log processing tool, APEL interprets logs of Grid gatekeeper (e.g. globus) and batch system logs (e.g. PBS, LSF, SGE and Condor) to produce CPU job accounting records identified with Grid identities. These records provide a complete description of usage of computing resources by user's jobs. APEL publishes accounting records into an accounting record repository at a Grid Operations Centre (GOC) for the access from a GUI web tool. The functions of log files parsing, records generation and publication are implemented by the APEL Parser, APEL Core, and APEL Publisher component respectively. Within the distributed accounting infrastructure, accounting records are transported from APEL Publishers at Grid sites to either a regionalised accounting system or the central one by choice via a common ActiveMQ message broker network. This provides an open transport layer for other accounting systems to publish relevant accounting data to a central accounting repository via a unified interface provided an APEL Publisher and also will give regional/National Grid Initiatives (NGIs) Grids the flexibility in their choice of accounting system. The robust and secure delivery of accounting record messages at an NGI level and between NGI accounting instances and the central one are achieved by using configurable APEL Publishers and an ActiveMQ message broker network.

1. INTRODUCTION

A computational Grid is a large scale virtual information processing infrastructure, which is spanning technologies, platforms, and organisations. In this infra-

structure, various distributed information processing resources are “shared together and a common collaboration is established across multiple administrative organisations” [1].

In any computational Grid, a reliable and efficient accounting mechanism is the key to measure its resource usage and consumption accurately and provide accountable and valuable information for Grid resource providers and their users to optimise the provision and usage of these resources respectively [2].

In this paper, an open and flexible distributed CPU usage accounting infrastructure for the WLCG and EGEE Grids is introduced. The paper is organised as follows: In Section 2, the WLCG and EGEE Grids are introduced and the accounting mode of CPU usage by Virtual Organisation (VO) users of the Grids are explained briefly. In Section 3, APEL, a CPU usage accounting information collection, publication and storage tool is introduced and explained. In Section 4, the analysis design, implementation, and evaluation of CPU usage accounting infrastructure for the WLCG and EGEE Grids is reported. Finally, Section 5 presents future research topics and concludes the paper.

2. WLCG AND EGEE GRIDS

2.1. Grid Infrastructure

Historically the birth and development of Grid computing technology was largely driven by the High Energy Physics (HEP) community who demands huge amount of storage and computing resources for experimental data storage and analysis [3]. Most recently, the Worldwide LHC Computing Grid (WLCG) is built for the “entire” HEP community to use the Large Hadron Collider (LHC), the largest scientific instrument on the planet, at CERN to discover new fundamental particles and fields and analyse their properties [4]. It is estimated that around 15 Petabytes (15 million Gigabytes) of data will be produced by LHC experiments and its detectors. This data will be distributed around the globe, accessed and analysed by thousands of scientists worldwide.

Similar to the development history of Grid computing technology, the Enabling Grids for E-sciencE (EGEE) project started as a European infrastructure to open Grid facilities to multi-disciplinary applications [5]. The European part of the WLCG relies on the EGEE Grid as the underlying infrastructure provider [6]. The EGEE Production Service infrastructure federates 250 resource centres worldwide, providing some 40,000 CPUs and several Petabytes of storage.

2.2. The Role of Accounting in the Grids

In WLCG/EGEE Grids, jobs are submitted by users to computing resources via a Grid Resource Broker (e.g. gLite Workload Management System). The usage of these resources is measured by recording the CPU usage of each user's jobs to provide a complete description of usage of resources. Note that apart from the CPU usage accounting, there is also another purpose of accounting, such as the storage usage accounting and more general "Grid Services" accounting for the usage of core services, middleware and software licenses rather than CPU usage [7].

3. THE APEL ACCOUNTING TOOL

3.1. CPU Usage Accounting

The APEL (Accounting Processor for Event Logs) is a CPU usage accounting tool designed and deployed for the WLCG/EGEE Grids [8]. As a log processing application, it interprets logs of Grid gatekeeper (e.g. globus) and batch system logs (e.g. PBS, LSF, SGE and Condor) to produce CPU job accounting records identified with Grid identities. APEL publishes accounting records into a centralised repository at a Grid Operations Centre (GOC) for access from a GUI web tool. The functions of log files parsing, record generation and publication are implemented by the APEL Parser, APEL Core, and APEL Publisher component respectively.

3.1.1. Accounting Records Schema

APEL describes accounting data using two different schemas: an individual Grid job record, based on the Open Grid Forum Usage Record (OGF-UR) v.1 specification, and an aggregated record, based on a proposal for the OGF-UR v.2 specification. An individual Grid job accounting record describes the resources consumed by a single executing job. It contains information about the submitting user, the executing Site, the CPU usage amongst other job information (Table1). An aggregated accounting record describes the resource usage by a collection of Grid jobs (Table2 and 3).

APEL distinguishes between two different sets of accounting summaries. Anonymous data is public and describes resources consumed per site/VO/month (Table 2). User level data contains resource usage information for individual users (Table3). Access to this data must be restricted as it contains personal information such as userDN and VOMS authorization information.

Column name	Type
RecordIdentity	VARCHAR(255)
ExecutingSite	VARCHAR(50)
LocalJobID	VARCHAR(50)
LCGJobID	VARCHAR(50)
LocalUserID	VARCHAR(50)
LCGUserID	VARCHAR(255)
LCGUserVO	VARCHAR(50)
ElapsedTime	VARCHAR(30)
BaseCpuTime	VARCHAR(30)
ElapsedTimeSeconds	INTEGER
BaseCpuTimeSeconds	INTEGER
StartTime	VARCHAR(30)
StopTime	VARCHAR(30)
StartTimeUTC	VARCHAR(30)
StopTimeUTC	VARCHAR(30)
StartTimeEpoch	INTEGER
StopTimeEpoch	INTEGER
ExecutingCE	VARCHAR(50)
MemoryReal	INTEGER
MemoryVirtual	INTEGER
SpecInt2000	INTEGER
SpecFloat2000	INTEGER
EventDate	DATE
EventTime	TIME
MeasurementDate	DATE
MeasurementTime	TIME

Table 1. Individual Job Records

Column name	Type
ExecutingSite	varchar(50)
LCGUserVO	varchar(255)
Njobs	int(11)
SumCPU	decimal(10,0)
NormSumCPU	decimal(10,0)
SumWCT	decimal(10,0)
NormSumWCT	decimal(10,0)
Month	int(11)

Year	int(11)
RecordStart	date
RecordEnd	date

Table 2. Anonymous Summary Record

Column name	Type
ExecutingSite	varchar(50)
LCGUserVO	varchar(255)
UserDN	varchar(255)
PrimaryGroup	varchar(255)
PrimaryRole	varchar(255)
Njobs	int(11)
SumCPU	decimal(10,0)
NormSumCPU	decimal(10,0)
SumWCT	decimal(10,0)
NormSumWCT	decimal(10,0)
Month	int(11)
Year	int(11)
RecordStart	date
RecordEnd	date

Table 3. User-level Summary Record

3.1.2. Encryption of UserDN

As part of the accounting data, APEL gathers and publishes the X.509 certificate DN of the submitting user. This information is considered to be personal data and therefore there is a requirement to protect it during transportation and when it is stored.

APEL encrypts user DN information based on a public/private key pair and a randomising function that reduces the likelihood of repeatable patterns arising in the encrypted string.

Before encryption of the DN string, two random numbers n1 and n2 are generated by the Bouncy Castle Crypto pseudo-random number generator and added to the input string according to the following format: *n1*DN_string*n2*.

The new string is then encrypted using a 1024-bit RSA key with PKCS1 v.1.5 padding. The size of the encryption key and the type of padding determines the length of the data that can be encrypted; in this case, the limit is 117 bytes. Doubling the size of the RSA key to 2048 bits would allow an input string of 245

bytes, but at the cost of greatly increasing the time needed to generate and operate the strings.

Although the 1024-bit key is enough to encrypt most of the user DNs in the EGEE project, in some occasions a longer DN is encountered and an algorithm is applied to shorten this input string before the encryption is applied. The random numbers are removed and common patterns in the DN (i.e. /OU=personal certificate/) can be reduced to simple strings (/ou=pc/).

To identify the encryption scheme used for decryption, the algorithm version header is added as a prefix to the encrypted cipher string. In the current version of APEL, the version header is *APEL V.0.2*.

Once the encrypted data has been published into the APEL server, it is stored in an offline database with restricted access. Only then the data can be decrypted using the header information together with the APEL private key. The random information is then removed using regular expression analysis.

3.1.3. CPU Time Normalization

As the CPU performance varies greatly between different resources, even within a single site, a reference is needed to provide a fair comparison of resource usage consumption.

APEL scales CPU time to a reference benchmark of 1K.SI2K hours. Each Grid site publishes a value for the CPU speed (described by the SpecInt2000 performance benchmark) for each site cluster as part of the site's GLUE schema. When generating accounting records, APEL interrogates the site Grid Index Information Service (GIIS) to obtain this data. Each individual record will then contain the CPU speed equivalent from the worker node where the job was executed. Once the record has been published into the APEL Accounting Server, the CPU time can then be normalized to the reference value (1K.SI2K Hours) by applying the following calculation:

*NormCPUTime=GlueHostBenchmarkS100/SI2K*BaseCPUTime*,

where SI2KRef is 1000. This procedure should be treated as an approximation as most sites are not homogeneous and describing a cluster by the performance of an average CPU is not entirely sufficient.

3.2. Scalable and Secure Transportation Mechanism

The APEL tool is at the time of writing in a transition phase. Current production system still uses of R-GMA (Relational Grid Monitoring Architecture) [9] as the transport mechanism for moving accounting records generated on each Grid client site to a centralised repository at a GOC. R-GMA Primary Producers for publishing records from each Grid site and a Secondary Producer for aggregating records into a centralised repository. A general topic publication and subscription messaging model enables distributed components in a system to publish and sub-

scribe messages to/from a well defined topic that can be viewed as a virtual destination and source of messages. The definition of topics and low level reliable delivery of messages among components can be achieved by using a concrete message broker implementation to this model.

In the work reported here, such a model is investigated and implemented with a view to construct a distributed accounting infrastructure, which will includes a large number of NGIs and support flexible queries on accounting records generated by VOs across multiple NGIs.

3.3. Integration with External Accounting Systems

Some regions within EGEE, as well as some partner projects, have their own accounting infrastructure deployed. They are namely: INFN-Grid, which are using DGAS [10], NorduGrid with SGAS [11] and the OSG with GRATIA [12]. Getting accounting information from these regions is done by these systems publishing already processed data and summaries to the APEL front end.

4. AN OPEN AND FLEXIBLE ACCOUNTING INFRASTRUCTURE

4.1. New Requirements from EGI and NGI

The third phase of EGEE, started in May 2008, brings dramatic changes to the project's operational model compared to EGEE-I and EGEE-II. These changes are proposed in order to achieve a successful transition from a central, project-based model to a sustainable infrastructure built on top of each EGEE region, possibly breaking down to country level. This final requirement is a result of a European Grid Initiative (EGI) [13] design study where each participating country is given the responsibility of maintaining its own National Grid Infrastructure (NGI). Ideas for a general evolution in the years to come are discussed in the EGEE-III OAT strategy document [14].

This redefinition of the European Grid landscape induces many changes in all tools and services running on top of it, and APEL is no exception.

One of the first requirements of the system is to allow for each future NGI to have their own accounting repository should they wish to. This resulted in the design of a multi-level accounting system that will be described below. Along with changes in the architecture, distributing APEL massively across EGEE regions

also implies a new design where automation, simplicity and scalability are key factors.

Knowing that some regions or partner projects use their own accounting system and will continue to do so, integration and interoperation is also a major requirement. Since there is a need for a central accounting repository for EGI, all participating NGIs should be able to report to it, whatever tool they use. This involves opening the system in such way that accounting data can transit from external tools to APEL in an easy way.

Another requirement lies in the generalized use of a shared communication mechanism between EGEE tools, which took the shape of a backbone message bus [15]. The replacement of the APEL transport layer has become one of its key evolutions, as explained in previous section.

Finally and in the long term, the general move towards standardization of accounting records format and the opening of data flows might result in a requirement for a standard infrastructure, which will also be described later in this paper.

4.2. Infrastructure Design

Based on the analysis in the previous section, the design of an open and flexible distributed accounting infrastructure aims to achieve these goals:

1) Multiple levels of publication should be supported (country -> region -> central),

2) Different regions should use a unified publication interface,

3) No significant operational changes to the existing Grid client side publication should arise,

4) A region should optionally be able to set up its local Accounting Server or use the Central Accounting Cache, and

5) The system should allow for potential interoperability and integration with other Grid middleware (e.g. Monitoring).

The infrastructure supports three major accounting publication modes: regionalized, non-regionalized and integration with the third-part accounting systems. However, due to the lack of available efficient distributed database query mechanism, a super central records cache of NGI or local Grid accounting instances will be set up to support accounting records queries across VOs.

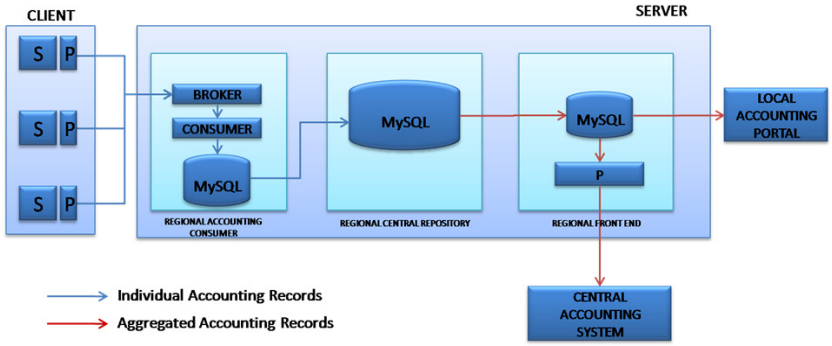

Fig. 1. A Regionalised Accounting System Connected to the Central Accounting System

Figure 1 illustrates the architecture of a single regionalised accounting system. The system may optionally connect to a central accounting system to publish an aggregated summary of accounting records. The central accounting system is illustrated in Figure 2, in which, Region A is "non-regionalised" direct publication of Grid sites; Region B is "regionalised" as in Figure 1; and Region C is a third party accounting system that is integrated into the infrastructure via a unified publication interface. An important design feature of the infrastructure is that the scalability of it can be increased by replicating the function components of a central accounting system into regional accounting systems that only republish aggregated accounting record summaries into the central accounting system.

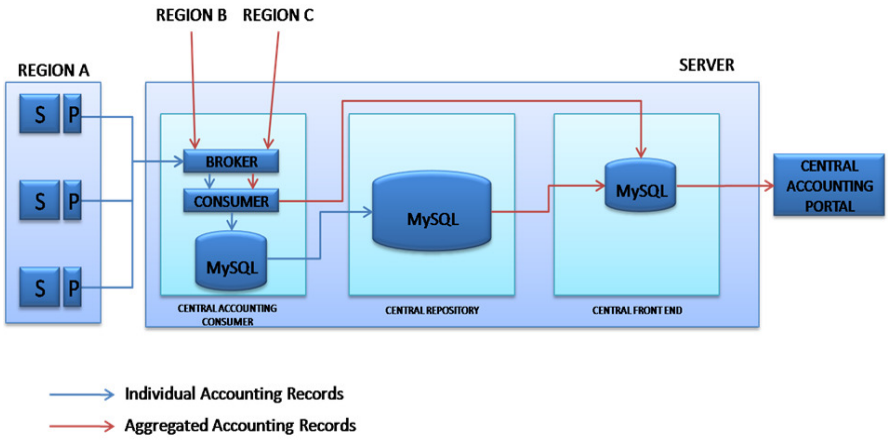

Fig.2. Three Accounting Publication Sources and the Central Accounting System

4.3. Implementation and Evaluations

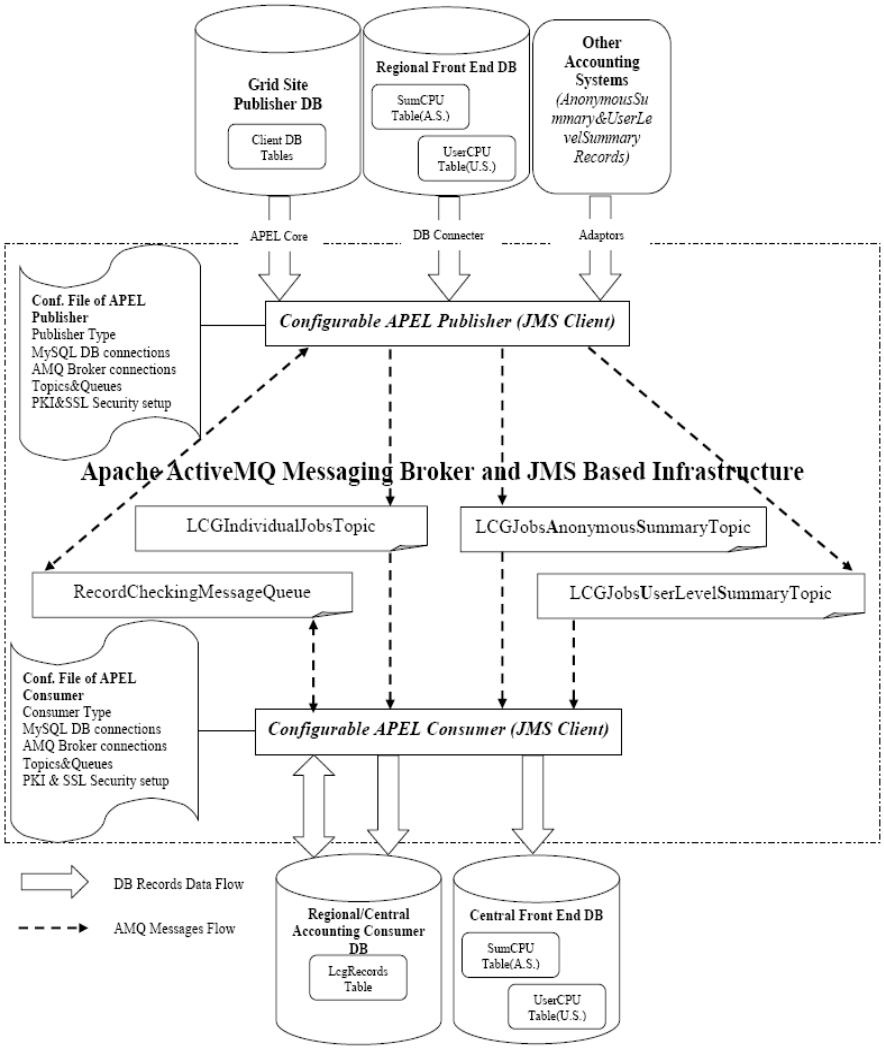

Fig. 3. An Accounting infrastructure based on Apache ActiveMQ Messaging Broker and Java Message Service APIs

The work reported here investigates the feasibility of adopting a general messaging model to implement a distributed accounting infrastructure and utilises the Apache ActiveMQ [16] message broker and Java Message Service (JMS) API based clients to implement the accounting records transport layer of APEL for robust delivery of accounting record messages. In this implementation, while the

ActiveMQ message brokers will manage the delivery of accounting records message with secure mechanisms (SSL and PKI based authentication and DN based authorization), at a NGI level and between NGI accounting instances and the central records cache, the original user interfaces for existing APEL clients will remain consistent.
As illustrated in Figure 3, from top to bottom, there are three key groups of components in the implemented infrastructure:

1) Accounting record sources (see Section 4.2),

2) A configurable APEL accounting publisher and consumer (JMS Clients), and

3) Accounting record destinations (a regionalised accounting system or the central accounting system).

Although the ActiveMQ broker itself and its security configuration is not explicitly presented in the figure, various topics and queues of the accounting records messaging system are highlighted as they are defined and supported by a concrete message broker. The connection and configuration information of a concrete broker is set up in configuration files of APEL accounting publisher and consumer.

Another feature of this implementation is that the accounting messages are sent in a predefined plain text format (using key-value pairs). This allows interoperability with consumers other than the APEL client; provided the other client is authorized to use the broker network. This feature allows integration with other Grid middleware for wider applications.

Preliminary testing of the implementation with production CPU usage records demonstrates that the transport mechanism of the distributed accounting infrastructure is reliable and promising. More large scale testing is designed and prepared as the paper is being prepared.

5. CONCLUSION AND FUTURE WORK

In this paper, an APEL tool based CPU usage accounting infrastructure for the WLCG/EGEE Grids is presented. The infrastructure is designed and implemented to be open and flexible to regionalised accounting requirements in future NGI environments. Within the infrastructure, accounting records are transported from Grid sites APEL publishers to either a regionalised accounting system or the central one by choice via a common ActiveMQ broker network. The record messages can be generated according to a common format and delivered as plain text messages so that it may enable another Grid middleware (e.g. monitoring) to consume accounting information, provided it is authorised to connect to the common broker network.

In the short term, further investigations on the scalability (e.g. network of brokers) and fault tolerance features (e.g. failover pair, master/slave backup) of ActiveMQ will be conducted to construct a robust and durable accounting service.

Within the first months of EGI, there will be a strong validation test for all operational tools and systems within the current infrastructure. At that time, further work on interoperations with other Grid infrastructure will potentially join the efforts on distributing the system across NGIs.

In the longer term, progressing towards an open, standard and interoperable accounting system leads the way to the deployment of a Resource Usage Service (RUS) [17] as defined within the Open Grid Forum (OGF) [18]. Discussions and collaborations have been started in this area between APEL developers and the OGF RUS working group. It is expected that further inspirations and proposals may be produced in the future.

6. REFERENCES

[1] I. Foster and C. Kesselman and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of Supercomputer Applications, 15(3), 2001.

[2] Martin Waldburger, Matthias Göhner, Helmut Reiser, Gabi Dreo Rodosek and Burkhard Stiller: Evaluation of an Accounting Model for Dynamic Virtual Organizations. Journal of Grid Computing, Springer, Vol. Online First, No. DOI: 10.1007/s10723-008-9109-9, pages 1-19, Netherlands, September 2008.

[3] The Large Hadron Collider (LHC), http://lhc.web.cern.ch

[4] The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG

[5] Enabling Grids for E-SciencE (EGEE), http://www.eu-egee.org

[6] Ian Bird, Bob Jones and Kerk F. Kee: The Organization and Management of Grid Infrastructures, pages 36-46, Computer, IEEE Computer Society, January 2009.

[7] M. A. Pettipher, A. Khan, T.W. Robinson and X. Chan: Review of Accounting and Usage Monitoring, Final Report, e-Infrastructure Programme, 17 September 2007, JISC.

[8] http://goc.grid.sinica.edu.tw/gocwiki/ApelHome

[9] http://www.r-gma.org

[10] http://www.to.infn.it/grid/accounting/main.html

[11] http://www.sgas.se

[12] The Gratia Accounting System, https://twiki.grid.iu.edu/bin/view/MonitoringInformation/WebHome#Gratia_Accounting

[13] European Grid Initiative, http://web.eu-egi.eu

[14] J. Casey, et al, Operations Automation Strategy, https://edms.cern.ch/document/927171

[15] J. Casey, et al, MSG - A messaging system for efficient and scalable grid monitoring, EGEE User Forum, March 09.

[16] Apache ActiveMQ, http://activemq.apache.org

[17] Resource Usage Service, http://www.ogf.org/gf/group_info/view.php?group=rus-wg

[18] OGF, www.ogf.org