

## LCG-RUS: Aggregative Accounting Service Enabling Economic Modelling for Commercial Grid

---

**Xiaoyu Chen**\*

*School of Design and Engineering*

*Brunel University, UK*

*E-mail: Xiaoyu.Chen@brunel.ac.uk*

**Akram Khan**

*School of Design and Engineering*

*Brunel University, UK*

*E-mail: Akram.Khan@brunel.ac.uk*

### Abstract

Grid computing has evolved into a standard for resource sharing across different communities to realise Internet-based high performance computing. With adoptions to popular Web service technologies, a stream of commercial Grid initiatives are ready to offer Grid computing for commercial usages based on Globus and customised proprietary architecture. In this paper, we describe an extensive work on Global Grid Forum (GGF) Resource Usage Service (RUS), the LCG-RUS, which is a partially EU-funded project and provides high-level aggregative view of accounting information across Grid infrastructures in LCG. With LCG-RUS, heterogeneous economic models could be adapted to perform a Grid-enabled economic analysis and simulation. The result of economic activities in turn contributes to decision making for managers of commercial Grid.

*Grid Technology for Financial Modeling and Simulations*  
*Palazzo Steri, Villa Zito, Palermo, Italy*  
*February 3 – 4, 2006*

---

\* Speaker

## 1. Introduction

Grid accounting provides an accurate view of the resource and service usage for each job utilising the Grid. The emergence of Grid accounting makes it possible for various economic models to make use of accounting information for whatever economic analysis and simulation activities. Supply and demand model, for example, could make use of the accounting information to predict the changes in price and quantity of Grid resources as commodities. However, the accounting information at present is stored at the job level and does not leave the site, which then provides limited information for economic activities. In this paper, we present the LCG-RUS project that collects, refines and aggregate job accounting information from sites visually.

## 2. Overview of LCG-RUS

The LCG-RUS project is motivated by providing a Virtual Organisation (VO)[1] view of accounting information consumed by jobs running across Grid infrastructures, (i.e. Open Science Grid in U.S, NorduGrid in Netherlands, and EGEE in Europe). At the moment, the main accounting information is stored conforming to Usage Record (UR) [2] format recommended by the Grid Global Forum (GGF) standard organisation as well as existing SQL schema (APEL lcgrecords Schema for instance). These job-level accounting information is accessible via the Resource Usage Service (RUS) [3], an operational specification also defined by GGF for monitoring and auditing purposes. However, implementations of GGF-UR and –RUS expose several problems for application consuming accounting information. First of all, the job-level accounting information as with the UR specification includes verbose details about jobs and leaves lots of workload for information consumers to organise the most interested parts. Secondly, the huge amount of jobs would result in poor performance of operation upon accounting records. There are, for example, almost 170 sites producing five million job accounting records annually in LCG [4]. It consequently would take minutes for a consumer endpoint, for example an economic model, to fetch monthly job accounting records from each site. Similar lessons are also learned from the MCS-RUS [5] project, a WS-I implementation of RUS.

The main idea behind The LCG-RUS is to provide middleware that supplies optimised operation performance for potential consumers of Grid accounting information. As illustrated in figure 1, the LCG-RUS accepts job-level accounting information from sites; refines its contents for different consumer parties (Grid users, Site/Grid Centre Operator, VO manager); and aggregates information for different usage parties or applications. The flexibility and extensibility of the aggregative schema allow LCG-RUS consumers to customise their aggregative preferences. The supply and demand model at the diagram for CPU time requires quantity of available CPU times and prices of different providers to predicate the future trends of CPU usage. Then the application could specify the required information as pricing and CPU usage information within a fixed period.

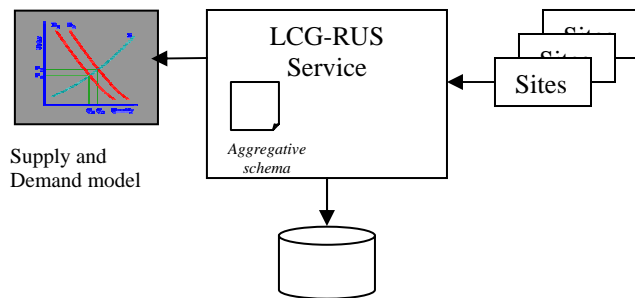


Figure 1: LCG-RUS Diagram

### 3. Sumur Schema

The aggregative schema, “Sumur” short for Summary Usage Record, is the most important part of the LCGRUS specifying the format of aggregative accounting records that are fed into the LCGRUS operations. By dividing the usage level in hierarchy and providing extensible framework consumer endpoints are allowed to defining customised levels.

#### 3.1 Aggregative Levels and Extensive Framework

The aggregative accounting record should indicate collective information in which endpoint applications are mostly interested. For instance, Grid job users only care about monthly accounting records with collective view of charges and usage information, while VO managers more concern how many jobs run and resource consumed for a specific user group this month this year. Two properties contribute to a hierarchical classification of aggregative levels. The naming property takes effort as the first level, which divides aggregative accounting records into user level and anonymous level. At the user level, application endpoints aggregate user-specific accounting records. The anonymous level aggregates accounting records according to the second property, the granularity. It would be further divided into four sub-levels according to the fine-to-coarse granularity, site level, Grid level, and VO level. This classification of aggregative levels satisfies the data input of different economic model simulators either for microeconomics and macroeconomics. Each level defines a set of required properties and is abstracted into a group. The head group is the “level” property without any elements defined. The new levels would either defined by extend the head level or starts at any level with a “substituteGroup” attribute pointing to an existing level.

#### 3.2 Structure

An aggregative usage record instance is consisting of three blocks: the root element, aggregative usage information, and charging block. As with GGF-URWG specification, an aggregative accounting record should start with the “<aggregate>” element indicating the beginning of an aggregative accounting record. The attribute, “level” is mandatory to declare the level of this accounting record explicitly. The usage information block contains aggregative usage information with that specific level. The charge information block contains the site specific (every site has its own pricing and charging policies) pricing and charge information as well as aggregative charge information of that level, such as total charge amount.

#### 3.3 Properties

According to GGF-UR standard, an aggregative accounting record MAY include any number of basic properties defined in the GGF-UR standard as well as customised aggregative properties. The Sumur schema introduces two main properties, the usage information property and the charge information property. Most of the sub properties of usage information block refer to GGF-UR’s base properties. The charge information property defines the basic properties about the charge agents and charge information of the endpoint with that level.

### 4. LCG-RUS Prototype

GGF-RUS working group defines RUS specification based on GGF-UR specification to store accounting information. Accordingly, LCG-RUS defines Web service port types based on

operation requirements of LCG-RUS summary schema. The LCG-RUS validates the incoming aggregative accounting records with the Sumur schema, before extracting aggregative property information and storing them into relational database system. An identity is automatically produced during parsing. Combining four mandatory properties, the executing site, year, month, and level type, makes the identity global unique. An authorised site could update monthly data by sending a monthly resource record twice. The LCG-RUS will detect a duplicated identity when trying to “insert” the record into database. Alternatively, a compulsive port is provided only for LCG-RUS administrator to update records explicitly on unexpected occasions. Deleting a record is also allowed only for LCG-RUS administrator. Every Grid user has authority to view user-level accounting, while only some group of peoples have the right to perform extraction operations of other level types.

### 5. Implementation

The implementation of LCG-RUS has been deployed into Apache Tomcat with JAX-RPC [6] Web service implementation supported. XMLDB [7] APIs, providing the facility storing XML fields into MySQL database are integrated into LCG-RUS core accompanying with aggregative accounting record validator module based on XML parser. Security is ensured by the GridSite [8], which based on Apache web service and provides mutual authentication and role-based access control.



Figure 1: LCG-RUS Prototype

### 6. Conclusion and Future work

Grid computing has received attentions from enterprise. A stream of commercial Grid initiatives, such as IBM, SUN and HP, provides Grid solutions for enterprise. Economic model and commercial Grid are mutual beneficial in this sense. For one side the Economic model of commercial Grid provides valuable information for commercial decision making. On the other hand, the Grid provides an unbeatable platform for economists to perform economic analysis and simulations.

The LCG-RUS project exposes a bridge linking the Grid communities and economic world. However, the LCG-RUS exposes some restrictions at the meantime. The site has to formalise a well-formed accounting record compatible to summer schema, the functionality which LCG-RUS does not provides. There is an increasing risk of large number of records with more and more Grid peers getting involved. A possible solution would be distributed LCG-RUS in Peer-to-Peer model. All those issues would be within our next-step scope.

## References

- [1] I. Foster, C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International J. Supercomputer Applications, 15(3), 2001.
- [2] R. March, R. Lepro-Metz, and S. Jackson, *Usage Record-Format Recommendation*, Global Grid Forum, 2003.
- [3] S. Newhouse, *Resource Usage Service*, Global Grid Forum, June, 2003.
- [4] LHC Computing Grid Project.
- [5] J. D. Ainsworth, J. MacLaren, J. M. Brooke, *Implementing a Secure, Service Oriented Accounting System for Computational Economics*, CCGrid, 2005.
- [6] JSR-101 expert group, *Java API for XML-based RPC*, Sun Microsystem, Inc, October, 13, 2003.
- [7] Kimbro Staken, *XML Database*, XML:DB working group, September, 2001.
- [8] A. McNab and S. Kaushal, *The GridSite Security Framework*, AHM 2005 (417).