Contribution ID: **215** Type: **poster**

Multiple Virtual Databases to support multiple VOs in R-GMA

Monday 13 February 2006 11:00 (20 minutes)

R-GMA is a relational implementation of the GGF's Grid Monitoring Architecture (GMA). In some respects it can be seen as a virtual database (VDB), supporting the publishing and retrieval of time-stamped tuples. The scope of an R-GMA installation is defined by its schema and registry. The schema holds the table definitions and, in future, the authorization rules. The registry holds a list of the available producers and consumers. At present, while it is possible to have multiple installations, a user can only use one at a time and hence cannot access tables in another installation. We plan to introduce multiple VDBs, where each VDB is defined by its own registry and schema. In this paper we explain the basic idea of R-GMA, why we need multiple VDBs to support multiple VOs, and how we will implement them. We also discuss the possible need to create some VDBs not related to end-user VOs. We also explain why we do not plan to provide a catalogue of VDBs as a part of R-GMA.

Summary

R-GMA is a relational implementation of the GGF's Grid Monitoring Architecture (GMA). GMA defines producers of information and consumers of information and a registry which knows the location of all consumers and producers.

The scope of an R-GMA installation is defined by its schema and registry. The schema holds the table definitions and, in future, the authorization rules and the registry holds the information about available producers and consumers. An R-GMA installation looks similar to a single virtual database.

R-GMA currently offers a single schema to all users. There can be more than one R-GMA installation in the world but a site cannot easily belong to more than one installation. The client APIs currently connect to a single registry and schema. The registry and schema may be physically replicated for resilience and performance however this is not the subject of this paper.

The solution we plan to implement is to introduce multiple virtual databases. Each virtual database will be defined by its own registry and schema. In HEP, today, an experiment typically has just one VO but we imagine it would be responsible for one or more virtual databases. The virtual database (VDB) concept has no direct link with that of the VO except that someone must be responsible for the administration of a VDB, and this responsibility might well be assumed by a VO.

When data are inserted into a producer, the data are published into a specific VDB. Queries may include data from several VDBs. We use the normal SQL syntax of a database prefix before the table name to specify the VDB.

Some information may be of interest to a large number of VOs, for example the GLUE schema, which holds information on available resources. There would appear to be two ways of handling this. There could be a special VDB defined for resource providers. This could have authorization rules to ensure that only resource owners can publish information about their own resources. Alternatively each VO with an interest in GLUE, which would include all LCG VOs, defines the same

table in a VO defined VDB. The Resources then publish information to the VDBs of the VOs they serve.

A query of the form “SELECT * FROM VDB1.T, VDB2.T” would not have the effect of returning the union of the two queries: “SELECT * FROM VDB1.T” and “SELECT * FROM VDB2.T” but would rather return all pairs of tuples. To provide this union, additional syntax has been defined: “SELECT * FROM {VDB1,VDB2}.T” to indicate that the query should be evaluated over the union of the tuples from table T in VDB1 and VDB2. It requires that the tables T from the two VDBs are identical. If only certain columns are requested the constraint would only apply to that projection.

We do not plan to provide any general way to determine what VDBs exist globally as this would imply some kind of global registry of VDBs. If someone wished to use R-GMA to create such a global registry, it would be possible. A VDB could be created with a table or tables where other VDB creators are invited to publish their VDB names and registry and schema endpoints, but this will not be an R-GMA provided feature.

A single registry or schema service will host multiple VDBs, where each VDB will probably have its own database on the server. There would be a requirement that a VDB name would be “globally unique” - in fact the real constraint would be that a site would not be able to offer a service to different VDBs with clashing names. The user would need to ensure that the server he was using was supporting the set of VDBs in which he was interested.

We hope to implement this very soon to make the R-GMA system more scalable with multi-VDB support to meet the various needs of different VOs.

Authors: Mr DUNCAN, A. (Rutherford Appleton Laboratory); Mr WILSON, A.J. (Rutherford Appleton Laboratory); Mr WALK, J.A. (Rutherford Appleton Laboratory); Dr CORNWALL, L.A. (Rutherford Appleton Laboratory); Dr CRAIG, M.S. (Rutherford Appleton Laboratory); Mr BYROM, R. (Rutherford Appleton Laboratory); Dr MIDDLETON, R.P. (Rutherford Appleton Laboratory); Mr HICKS, S.J.C. (Rutherford Appleton Laboratory); Dr FISHER, S.M. (Rutherford Appleton Laboratory)

Presenter: Mr WILSON, A.J. (Rutherford Appleton Laboratory)

Session Classification: Poster

Track Classification: Grid middleware and e-Infrastructure operation