Integrating Streams of Monitoring Data using Flexible View Hierarchies

Andy Cooke, Alasdair Gray, Lisha Ma, and Werner Nutt

School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh
nutt@macs.hw.ac.uk

Computational Grids are complex distributed systems to provide access to computational resources in a transparent fashion. A Grid constantly produces information about its own state, for instance about loads of CPUs, network traffic, running applications etc. The task of collecting such information and making it available to other Grid components for querying is called Grid monitoring.

DataGrid is the largest European Grid development project. We are involved in building DataGrid's monitoring system R-GMA (Relational Grid Monitoring Architecture).

R-GMA provides an infrastructure that allows arbitrary Grid components to play the roles of producers or consumers of information. This is achieved by specialized producer and consumer agents that support components in playing these roles. Information can take the form of static data sets or streams of data. Producers essentially describe their supply as views on a global relational schema while consumers pose queries against it. Thus, R-GMA implements a "local as view" approach to information integration.

An essential requirement for a Grid monitoring system like R-GMA is scalability. The system must be able to answer queries based on data collected from hundreds or thousands of producers without a decline in performance.

To achieve this we are currently adding so-called stream republishers to R-GMA. Republishers realize a mechanism similar to views in a database management system. A stream republisher is a component that is set up by posing a query over data streams. It then publishes the stream of answers to that query and makes it available to further querying. In particular, republishers can consume data from other republishers.

By creating republishers for a hierarchy of increasingly general queries one can create a mechanism that collects data in stages, starting with small trickles of data and combining them successively to larger streams. In this way, republishers allow one to collect data in a highly distributed environment without overburdening a single component.

Hierarchies of republishers are being maintained automatically. For each query defining a republisher, a query plan is generated that specifies from which producers and republishers to consume data and how to combine those data to an answer stream. Query planning for republishers has to address two issues:

1. Redundancies among the available data: Such redundancies exist because a republisher makes available data that it receives from its own sources.

When planning how to answer a query, one has to decide whether to use a republisher or to use its sources.

2. Changes among the set of producers and republishers: While a stream query is being executed, new producers may come into existence or existing republishers may fail and therefore no longer be usable. In such a situation a query plan may have to be altered so that no answers are missed. Moreover, if a new republisher is created, it may offer the possibility to improve a plan by replacing several existing producers and republishers with the new one.

We have defined a number of desirable properties of query plans, among them soundness, completeness, and duplicate freeness. For simple types of queries, we have developed algorithms that produce query plans with these properties. Our algorithms can also respond to changes among the the set of data sources by changing query plans in such a way that the properties mentioned above are maintained.

Although R-GMA is specifically developed for Grid monitoring, its architecture and mechanisms are applicable to other areas where there is a need for publishing and querying information in a distributed fashion. R-GMA can be downloaded from

`http://hepunx.rl.ac.uk/edg/wp3/`