

GRid-aware Optimal data Warehouse design (GROW)

Wednesday, February 13, 2008 3:00 PM (20 minutes)

The application is implemented as a Java framework for executing genetic algorithms in a distributed fashion using a Grid. The framework consists of two parts: genetic algorithm framework and grid tools. The first part enables researchers to easily implement new optimization problems by simply extending several classes. The second part enables researchers to make their application Grid-aware. In other words it enables easy Grid job submission, job status and output retrieval. The GROW application is a VIS optimization. The chromosomes are bit sequences, each bit representing weather particular view or index is materialized in the database. The chromosomes are evaluated on a set of database queries, where for each query we estimate the time and memory usage for its execution. The parameters for the GA optimization influence both per population GA execution and grid execution workflow. Some parameters are: mutation and crossover probability, islands, epochs, seasons, migration width.

3. Impact

The framework uses the following Java grid features: WMPProxy job submission, VOMS proxy init, DAG (Workflow) execution and LBProxy. Because the framework is implemented in Java, it makes the applications implemented in it portable on all operating systems supporting java 1.5. Also by using Java implementation of the Grid job management functions, the developed applications does not need an installed UI machine. For the Java Grid tools to work the application user needs to have: his certificate in p12 format, CA certificates, VOMS certificates and specification. For the implemented GROW application, the user needs to put the formerly mentioned files in different folders and specify their location in the application properties file. When the application loads, and the user wants to submit a job, he first must generate a VOMS proxy. For this he provides a password for the p12 file, VOMS name and FQAN. After this the other functionalities for the Grid tools are available.

4. Conclusions / Future plans

The porting process was in two phases. The first phase was the implementation of the Genetic algorithm framework. This was mainly to enable researchers reuse the already implemented GA structures. The second phase consisted of implementation of tools for automatic generation of JDL workflows, job submission, job status reporting and job output retrieval. Further development should enable automatic retrieval of CA certificates, VOMS configuration and infrastructure information (BDII).

Provide a set of generic keywords that define your contribution (e.g. Data Management, Workflows, High Energy Physics)

Workflow, Java WMPProxy, Java LBProxy, Genetic Algorithms

1. Short overview

GRid-aware Optimal data Warehouse design uses Gridified genetic algorithm to solve the problem of optimal data warehouse design. The main problem is to select the optimal set of physical objects (Views and Indexes) materialization (VIS) of a data warehouse for a specified database design, considering specified queries and additional parameters. This can significantly increase the performance of any large database. The Grid is used for parallelization of genetic algorithm optimizations.

Primary author: Mr JAKIMOVSKI, Boro (University of Sts. Cyril and Methodius)

Co-authors: Mr CEREPNALKOSKI, Darko (University of Sts. Cyril and Methodius); Mr VELINOV, Goran (University of Sts. Cyril and Methodius)

Presenter: Mr JAKIMOVSKI, Boro (University of Sts. Cyril and Methodius)

Session Classification: Workflow and Parallelism

Track Classification: Application Porting and Deployment