Advanced Databases Spring 2012

Project Milestone 3 Requirements Document

Due Wednesday May 30, 2012 at 1:15pm

Just like for Milestone 2 (M2), you have two choices

- Working on Squall, the open-source parallel stream engine or
- Working on another project (considered the default). For M3, it consists of building an OLAP engine.

You can choose either of the two, no matter if you did the mysql project or a Squall project for M2. If you did the mysql project in M2 **and** want to do the OLAP project in M3, go ahead. Otherwise, talk to us.

The project teams are the same as for M2.

The remainder of this document describes the OLAP project.

The goal of the OLAP project is to build a data cube and a data exploration frontend based on the system you developed in the M2 mysql project.

Data Cube:

- Start from the mysql-based query engine of M2.
- Create a data warehouse of TPC-H data.
  - Choose dimensions, hierarchical if possible, given the TPC-H schema. Try to support product, time, and location dimensions, or the best approximation to these that TPC-H supports. Support no finer resolution for time than days. The overall size of the data cube should not exceed the size of the base data in the database. Make appropriate choices.
  - Design the schema of the fact table and the dimension tables.
- Build a data cube from a suitable query aggregating sales grouped by (subsets of) the dimensions. You may hardwire that query using the operators developed in M2.
- All the views of the data cube have to be stored in the mysql databases. While the base TPC-H data that you construct the mysql database remain partitioned as in M2, it is your choice if and which of the data cube view to store in partitioned fashion. Your task is to maximize performance. If you think this requirement is ambiguous, document your thought processes and choices in the document you will submit with your solution.

Data Exploration Frontend:

- Implement a "currently displayed view" abstraction that allows exploratory data analysis. Implement roll-up, drill-down, slice, and dice operators. (Plus a step-back operator to undo slice/dice ops.) Remember

that these operations simply choose one of the materialized views and potentially apply a selection to show only part of it. The exploration ops (rollup, drill-down, etc) just navigate between these views.

- Conceptually, these operations support interactive data exploration, but developing an interactive (textual, keyboard-shortcut-based) user interface is optional. Should you want to create a graphical user interface (discouraged) that runs locally, the firewall settings may prevent it from connecting to the data cube on icdatasrv.
- Create either two java programs (one for constructing and storing the data cube and one for exploring it) or one which is in exploration mode if the database holds the data cube, with a command-line option to force data cube construction. You do not need to be able to automatically respond to database updates to refresh the data cube views. You construct the data cube – completely from scratch – when explicitly requested by the user.
- Code up a generator that creates and executes meaningful exploratory "walks" of given length – sequences of data cube operations. Example: Data cube on product, loc, time -> sales. Starting view: sales group by nothing (total sales). Example walk: drill down into product. Drill down into time. Dice Jan to April 2012. Roll-up loc (result is sales by time). Roll up time to months. (result is sales by month, for months Jan to April 2012). Drop dice on time (result is sales by month, for all months).

- Test the implementation, specifically correct cube construction and sequences of operators on the data cube.
- Try to design the components of the system as cleanly and generically as you can.
- Experiment with the performance of your system on icdatasrv, using the resources assigned to you there for M2.
- Report on the design and architecture of your system, and on the experimental results. Provide a guide how to install/set up your system and get it to run (on icdatasrv).

In those cases where this requirements document is vague, make your own choices and document them. Remember we are no robots, neither do we try to pose you a trap. Do something nice and it will be appreciated. (For example, choose the scaling factor accordingly.)

Submit your solution – source code and design/experiments document – zipped together in moodle by the deadline – precisely four weeks after the M2 deadline, by **Wednesday May 30, 2012 at 1:15pm**.
Submit the complete documented source code of your system – not just the files that are new compared to M2.