dQSQ User Guide

Noam Pettel

February 28, 2006

Contents

1 Introduction

dQSQ (short for distributed Query-Sub-Query) is a distributed system that optimizes evaluation of distributed Datalog (dDatalog) queries in a peer-to-peer environment. dQSQ implements a distributed version of the Query-Sub-Query technique that was designed to optimize evaluation of (centralized) Datalog queries. The dQSQ system is implemented on top of the Active XML (AXML) system (`http://activexml.net`).

2 Installing dQSQ

The dQSQ system is provided as an extension of the AXML framework. The steps needed to install dQSQ are as follows.

2.1 Install Active XML

The first step is to install AXML as a web application. For details refer to the AXML user guide in `http://activexml.net/AXML%20Guide.pdf`. If you want to simulate n dQSQ peers running on the same computer, install n AXML web applications.

2.2 Download dQSQ

The latest version of dQSQ, containing binaries and source code, is available for download at `http://www.cs.tau.ac.il/~milo/DQSQ/dqsq.html`. Unzip the downloaded zip file to your dQSQ home directory.

2.3 Edit dQSQ configuration file

Edit the file `config.xml` in your dQSQ home directory as follows. The `dQSQConfig` root element should have a `peer` element for every dQSQ peer installed on this computer. For each peer state the AXML home directory and point to the input Datalog file.

2.4 Deploy dQSQ

On Windows, run the script `installdQSQ.bat` on the dQSQ home directory.
On other operating systems run the command
`java -classpath setup/WEB-INF/lib/dqsq.jar;setup/WEB-INF/lib/jdom.jar;bin il.ac.tau.cs.dqsqenv.DQSQEnv install.`
After installing dQSQ, the file `dQSQHomePage.html` should be created in the current directory.

3 Running dQSQ

3.1 Generate dDatalog input

Before running the dQSQ system, you should define the dDatalog input program. Each peer should define its own part of the dDatalog program in an XML file. This file contains a definition of the extensional relations as well as the rules that are located on this peer. A sample `datalog.xml` file is available in the dQSQ home directory.

3.2 Run dQSQ

On Windows, run the script `rundQSQ.bat` on the dQSQ home directory.
On other operating systems run the command
```
java -classpath setup/WEB-INF/lib/dqsq.jar;setup/WEB-INF/lib/jdom.jar;bin
il.ac.tau.cs.dqsqenv.DQSQEnv run.
```

4 Web User Interface

The dQSQ peer has a Web User Interface, where you can perform several operations such as submitting a query, viewing the input Datalog program, viewing the program after the dQSQ rewriting and viewing the log. To enter the dQSQ home page, double-click on the `dQSQHomePage.html` on your dQSQ home directory. This file is created automatically after the installation of dQSQ. As an alternative, you can open a browser and go the URL `http://localhost:8080/axml/jsp/dQSQ/dQSQ.jsp`.

4.1 Submitting Queries

A dQSQ query has the following format: `q@P(x):-ancestor@P("Lois", x)`. This query computes all the ancestors of "Lois", assuming the ancestor relation is defined on peer P, and that ancestor("a", "b") means that "a" is an ancestor of "b". The result of the query is added to relation q.

- q is the query name, which is actually a relation that accumules the query result.

- P is the name of the current peer.

- Constants should be quoted. ("Lois" in the above example.)

- The relation in the body of the query (ancestor in the above example) must be located on the current peer. A query such as q@P(x) :- parent@R("Lois", x) cannot be posed on peer P because the parent relation is located on peer R.

5 Known Problems & Limitations

1. After several queries (depends on the load that the queries crearte) the system may become very slow and unusable. This is due to unresolved problems in Active XML. The only solution is to stop Tomcat and restart the dQSQ system.

2. Due to unresolved problems in Active XML, the result of a query may be incomplete.

3. Due to unresolved problems in Active XML, a query may never terminate. In such a case press the 'Stop Query' button to stop the query.