Generating Benchmark Databases for My Applications (PI: Dr. Lo Chi Lik Eric; 2009/10)

Performance is a key factor in evaluating database management systems (DBMSs). To evaluate the performance of a DBMS, it is necessary to execute a set of benchmark queries on a number of benchmark databases and measure the query execution time.

Benchmarking requires the generation of a number of benchmark databases. To thoroughly benchmark a DBMS, we usually generate a variety of benchmark databases in different data sizes (e.g., 10G, 100G, ...) and with different data characteristics (e.g., skew data distribution, uniform distribution). Nowadays, the Transaction Processing Council (TPC) benchmarks are the most widely accepted tools for benchmarking DBMSs. Although TPC benchmarks are comprehensively designed, they are confined to a predefined set of database schemas and queries. As a result, they may not reliably predict the performance of a DBMS with respect to a customer's real application. For example, in one of our experiments, TPC-H (a benchmark to evaluate the decision-support capability of DBMSs) concludes that DBMS A outperforms DBMS B but TPC-W (a benchmark to evaluate the transactional processing ability of DBMSs) concludes that DBMS B outperforms DBMS A. Imagine that we have a database application containing both complex decision support queries and simple transactional queries. It would be very difficult for us to follow the benchmark results and decide the best DBMS for our database applications.

Of course, we can abandon the use of standard benchmarks and use general-purpose database generators to generate our own customized benchmark databases. Unfortunately, general-purpose database generators (e.g., IBM DB2 Database Generator) generate databases based on the schema only (and possibly scaling factors

and value distributions) and *do not consider the queries during data generation*. Thus, it is not unusual for most application (benchmark) queries to finish quickly and return empty results if they are posed on such generated databases and these kinds of benchmark results are indeed not useful in benchmarking. To generate useful benchmark databases for our applications, this project proposes an "application-aware" benchmark database generator, namely, MyDBGen (a DataBase Generator for My applications). The breakthrough of MyDBGen is that it will allow users to control not only the characteristics of the generated data, but also *the characteristics (e.g., the result size) of the application (benchmark) queries*. This way, the generated benchmark databases can provide reliable predictions of the performance of various DBMSs with respect to some real customer database applications.

As an "application-aware" database generator, MyDBGen has a lot of applications. Other than helping customers to find the best DBMS for their database applications, application developers can use MyDBGen to generate "application-aware" benchmark databases which guarantee the application queries return very large query results (e.g., 1 million rows). That can help developers to stress testing the limit of their database applications. A DBMS vendor can also use MyDBGen to test the capability of the DBMS by generating a lot of simulated workloads (e.g., a large database with highly selective queries). Note that, no current database generators are able to generate data that provides guarantees on the query characteristics. MyDBGen will thus complement TPC benchmarks, traditional database generators, and real workloads (new applications do not have real data) in benchmarking and provide the database industry (including both DBMS vendors and customers) a useful and innovative "customer-oriented" benchmarking tool. The goal of this project is to investigate the various issues concerning the (1) design, (2) implementation, (3) performance, and (4) applicability of MyDBGen.