An Integrated Database Benchmark Suite

Hoe Jin Jeong and Sang Ho Lee
School of Computing, Soongsil University, Seoul, Korea
bqangel@gmail.com and shlee@comp.ssu.ac.kr

Abstract

This paper presents an integrated database benchmark suite, which is named SIMS. SIMS offers generic benchmarks, custom benchmarks, and hybrid benchmarks to users on a unified Web interface. Users can run benchmarks in realistic environments by performing the workload generation facility of SIMS, which generates composite workloads similar to those of the real world. Using SIMS, users can easily implement new custom and generic benchmarks in SIMS. An illustrative demonstration to add a new custom benchmark to SIMS is presented.

1. Introduction

As new database systems are developed or new functions are added to existing database systems, developers and users would like to evaluate new database systems or new functions systematically in different environments. As such, a number of database benchmarks [2][3][6][7][8] have been proposed in the literature.

This paper describes an integrated database benchmark suite, the Soongsil Integrated Benchmark Suite (SIMS), which provides environments convenient to run generic benchmarks, custom benchmarks, and hybrid benchmarks all together and a facility for realistic workload generation. SIMS offers a number of generic benchmarks, a framework to develop custom benchmarks, and hybrid benchmarks for database systems on a unified Web user interface.

2. Design Philosophy

The objective of SIMS is to provide users with a benchmark environment that helps evaluate the performance of database systems in various ways. Recognizing the difficulties most users can face during benchmarking database systems, we would like to build a framework that helps users effectively meet a variety of benchmark requirements.

One important consideration in the design phase of SIMS is extensibility. Our suite is extensible in that without changing the underlying architecture of SIMS users can: (1) add any generic benchmarks to SIMS, (2) extend pre-specified test queries in generic benchmarks to model new features of database systems, (3) add new custom benchmarks to SIMS.

We designed SIMS to simulate real-world environments as much as possible. To provide users with realistic benchmark environments, we developed a workload generation facility that puts a synthesized workload into the system under test. Workload generation can be performed in two ways [1]. One is an analytic approach, and the other is a trace-based approach. In the analytic approach, only a limited number of mathematical distributions are applied for workload generation. It may be unsuitable to use the analytic approach for the generation of realistic workloads, because real-world workloads are often too complicated to be simulated by mathematical models and workloads are subject to dramatic changes in unexpected ways. Hence, we adapt a trace-based approach to generate workloads in SIMS.

3. An Integrated Database Benchmark Suite

3.1. Modules and Parameter Files

SIMS consists of five independent modules: the unified interface module, the database creation module, the data generation module, the query execution module, and the workload generation module. Each module has exactly one parameter file, and it is implemented as a single process. The parameter files contain three components: section, entry, and value. The parameter file must be made in a pre-determined way.

Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG 2005)
0-7695-2534-2/05 $20.00 © 2006 IEEE

We completely separate the data generation from the query specification, so that any combination of data generation and query specification can be possible in SIMS. This formation of modules allows us to implement the most important feature of SIMS, extensibility.

3.2. Data Generation

This section describes the data generation module, a component of SIMS, by which users can generate a large volume of text data as well as XML data. SIMS totally supports eight distributions on three output data types (i.e., integer, float, and character string): ordinal, uniform, random, constant, Poisson, normal, negative exponential, and Zipfian distributions. We support the normal, negative exponential, Poisson, and Zipfian distributions by resorting to the algorithms in [4].

The data generation module of SIMS offers three different methods for users to generate XML data. The first method is to use a text file and a data structure definition file. The second method is to use a text file and a parameter file. The parameter file has elementary information for the data structures to be generated. Users can use the third method if they do not have any text files or data structure files.

3.3. Workload Generation

SIMS generates memory-bound, I/O-bound, and CPU-bound workloads. The workloads can be generated independently or in chorus by forking user processes to directly consume the resources of an operating system. It is difficult for a user process to precisely control the operating system resources. Hence, we use the notion of tolerance margin that represents the gap between the user's intended workload and the generated actual workload.

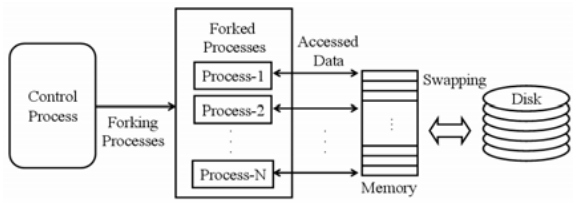

Figure 1. Memory-bound workload generation

As shown in figure 1, the workload generation module generates the memory-bound workload by causing the operating system to exchange data between the physical memory and the swap memory due to the virtual memory scheme. The workload generation module forks a number of processes that compete against each other to secure space in the physical memory of which the operating system is in charge.

Users can express the memory-bound workload by using the "iowait" value in a system command (for example, "top", "iostat", etc). The control process tries to achieve a workload as close as possible to the requested workload by controlling the number of processes, the amount of memory to be requested, and the sleep time of the processes. Given the user's requested workload, we are able to generate the workload with a 5% tolerance margin. See [5] for the detailed description.

4. Illustrative Demonstration

In this section, we describe how to easily add a new custom benchmark to SIMS and how to execute the benchmark in realistic environments with the workload generation facility. The new custom benchmark we want to add evaluates join operations and contains 42 queries. Users are expected to create five parameter files for adding the new custom benchmark into SIMS.

Figure 2 shows how to create a data generation parameter file that is one of the five parameter files. There are three tables in the "Tables" section, and information on "tableB" table is currently shown. The "Number of Records" entry shows that the number of records of "tableB" table was set as 208,000. The "Columns" section shows that the "tableB" table has three columns ("col1B", "col2B", and "chpadB") and the "col2B" column has an integer data type, a uniform distribution, no shuffling, and no null value. The "Minimum Number" and the "Maximum Number" entries are only used for the uniform distribution.

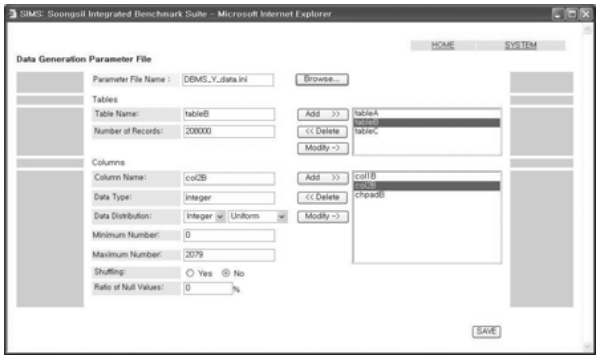

Figure 2. Data generation parameter file

After completing the five parameter files using the Web user interface, users can start the custom

benchmark by clicking the "Run" button in figure 3. Figure 3 shows the main Web page of SIMS.

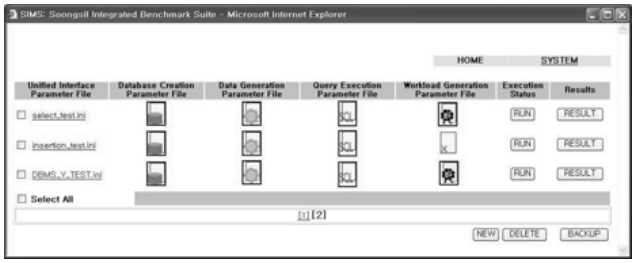

Figure 3. Main Web page of SIMS

We used a Sun Enterprise E3500 server and a commercial database system to execute the new custom benchmark. Figure 4 shows the elapsed times of the new custom benchmark's queries with the two workloads: the real TPC-C workload and the synthesized workload similar to the TPC-C workload. The shapes of the two graphs in figure 4 are similar to each other. The fact that the two graphs are similar means that SIMS generated a workload similar to the TPC-C workload. In summary, it is easy to add a new custom benchmark to SIMS and it is possible to execute the benchmark in realistic environments with the workload generation facility of SIMS.

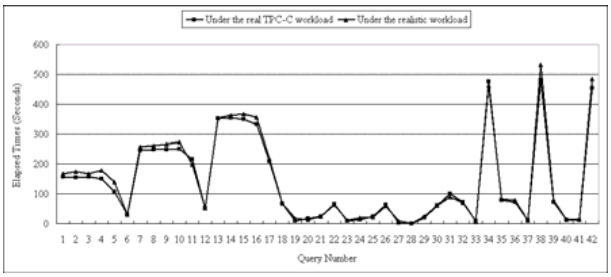

Figure 4. Results of the new custom benchmark

5. Conclusions

This paper describes an integrated database benchmark suite, SIMS, which offers generic benchmarks, custom benchmarks, and hybrid benchmarks to users in a unified Web interface. The extensibility of SIMS easily allows new benchmarks to be included in SIMS. SIMS supports facilities of data generation as well as workload generation.

We executed many internal experiments to show that SIMS can generate workloads users want to produce. The experimental results showed us that SIMS can generate workloads users want as well as the TPC-C workload.

SIMS was developed over several years, and implemented with two commercial database systems. Initially, we started to develop SIMS to fulfill the requirements of a local start-up company that is developing and marketing an object-relational database system. SIMS is currently in use at a local database company. An extension of SIMS for a multi-user environment is recommended for future work.

6. Acknowledgements

This work was supported by Korea Research Foundation Grant. (KRF-2004-005-D00172)

7. References

[1] Barford, P. and Crovella, M., Generating Representative Web Workloads for Network and Server Performance Evaluation, Proceedings of the 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 1998, pp.151-160.

[2] Cattell, R.G.G. and Skeen, K., Object Operations Benchmark, ACM Transactions on Database Systems, Vol. 17, No. 3, 1992, pp.1-31.

[3] DeWitt, D.J., The Wisconsin Benchmark: Past, Present, and Future, Gray, J. Ed., The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., Morgan Kaufmann, 1993, pp.269-316.

[4] Gray, J., Sundaresan, P., Englert, S., Baclawski, K., and Weinberger, P.J., Quickly Generating Billion-Record Synthetic Databases, Proceedings of the 1994 ACM SIGMOD Conference on Management of Data, 1994, pp.233-242.

[5] Jeong, H.J. and Lee, S.H., A Workload Generator for Database System Benchmarks, Proceedings of the 7th International Conference on Information Integration and Web-based Applications & Services, 2005, pp.813-822.

[6] Lee, S.H., Kim, S.J., and Kim, W., The BORD Benchmark for Object-Relational Database, Proceedings of the 11th International Conference on Database and Expert Systems Applications, 2000, pp.6-20.

[7] O'Neil, P.E., The Set Query Benchmark, Gray, J. Ed., The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., Morgan Kaufmann, 1993, pp.359-395.

[8] The TPC home page, http://www.tpc.org/.