Chapter 3

TOWARDS A SIMPLIFIED DATABASE WORKLOAD FOR COMPUTER ARCHITECTURE EVALUATIONS

Kimberly Keeton and David A. Patterson
University of California at Berkeley
Computer Science Division
384 Soda Hall #1776
Berkeley, CA 94720-1776
{kkeeton,patterson}@cs.berkeley.edu

Abstract We propose and evaluate a simplified technique for studying the architectural behavior of database workloads. This “microbenchmark” technique poses simple queries of the database to generate the same dominant I/O patterns exhibited in more complex, fully-scaled workloads. The potential benefits from this microbenchmark approach include smaller hardware requirements, less extensive workload parameter tuning, and simpler database parameter tuning. We demonstrate that the microbenchmark workload exhibits processor and memory system behavior relatively similar to that of the more complex standardized benchmarks. We also enumerate several factors that impact the representativeness of these microbenchmark workloads.

Keywords: Database, transaction processing, decision support, microbenchmark, and performance evaluation.

1. INTRODUCTION

In the last five to ten years, several studies have explored the architectural characteristics of online transaction processing (OLTP) database workloads [3] [7] [8] [9] [16] [17] [18] [19] [22] [23] [24] [26] [27]

1 This work was performed as part of the author’s dissertation research. The author's present address is: Storage Systems Program, Hewlett-Packard Laboratories, 1501 Page Mill Road, M/S 1U-13, Palo Alto, CA 94304-1126. Her current email address is kkeeton@hpl.hp.com.

L. K. John et al. (eds.), Workload Characterization for Computer System Design
and decision support (DSS) database workloads [3] [5] [15] [17] [18] [23] [32]. These studies used standard workloads defined by the Transaction Processing Performance Council (TPC), namely TPC-B and TPC-C for OLTP [10] and TPC-D, TPC-H and TPC-R for DSS [10] [30] [31]. Although these benchmarks specify well-defined workloads, they pose several challenges for the computer architects who attempt to use them in performance evaluations.

First, studying full-scale TPC database performance requires large hardware configurations, including at least tens or hundreds of disk drives and gigabytes of main memory [28] [29]. Researchers hoping to measure the performance of a real system must construct such a system, costing hundreds of thousands to millions of dollars. If, instead, the researcher wishes to simulate a full-scale system, he or she must simulate a large and complex I/O subsystem, requiring considerable computational resources and time.

Second, the complexity of the TPC benchmarks requires that researchers make many configuration choices, including whether data is accessed through the file system or through the raw disk device interface, the layout of data on disk to avoid access hot spots, and the choice of index creation to improve performance. Running the benchmark workloads requires tuning additional configuration parameters, such as database size (e.g., TPC-C’s number of warehouses or TPC-D’s scale factor) and the number of simulated OLTP clients.

Third, the database servers themselves pose both hardware and software configuration challenges. Large hardware systems force the researcher to decide on the number and speed of disks, I/O controllers, and I/O buses, and the amount and configuration of physical memory. Commercially available database servers have 75 to 200 initialization parameters to control runtime management issues such as the buffer pool size and management strategy, the degree of multithreading/processing, logging, and disk read-ahead. The operating system also presents numerous configuration alternatives. Although default values are often provided for these configuration parameters, they do not necessarily match the requirements of the intended workloads.

Few researchers have the resources to construct the large hardware systems required for fully scaled TPC benchmark workloads, nor the expertise to tune workload-specific and database-specific parameters so that the workloads run efficiently. Although some computer architecture researchers have found methods for working around many of the difficulties, the barriers to studying database workloads still exist. Essentially, one must have close industrial ties to study interesting configurations. As a result, database workloads have been used in fewer than ten percent of the performance evaluations reported in ISCA and ASPLOS over the last five