

Classifying E-commerce Workloads under Dynamic Caching

Fujian Liu Dwight Makaroff

Department of Computer Science
University of Saskatchewan
110 Science Place
Saskatoon, SK, S7N 5C9, Canada
{ful113, makaroff}@cs.usask.ca

Said Elnaffar

College of Information Technology
UAE University
Al-Ain, UAE
elnaffar@uaeu.ac.ae

Abstract

In an E-commerce system, the database server performance is crucial. Dynamic caching is often used to reduce the load on the database server, which reduces the need for scalability within the server itself. A good understanding of the workload characteristics of the database server in an E-commerce environment is important to the design, tuning, and capacity planning. Using dynamic caching can dramatically alter the characteristics of requests that the database server receives, changing the dominant operations, even the workload type.

The two major database workload classifications are online analytical processing and online transaction processing. Allocations for resources such as main memory can be very different, depending on the workload type. We investigate the E-commerce activity profiles resulting from dynamic caching. We find that dynamic caching does tend to alter the behavior towards transaction processing.

1. Introduction

The use of the Internet to conduct E-commerce activities has shown a steady increase and is a major component of the retail industry [1]. A typical corporation-sized E-commerce system is composed of front-end servers and a back-end database server (see Figure 1). The front-end servers include web servers and application servers that interact with users through a web interface and execute business logic respectively. The database server stores business information and processes queries.

In an E-commerce system, the front-end servers can be replicated easily because they store read-only data (e.g., images, scripts, and static web pages). Low-end commodity machines are often used for this purpose. Alternatively, the database server stores frequently updated data. It is resource intensive to replicate the database server and maintain data consis-

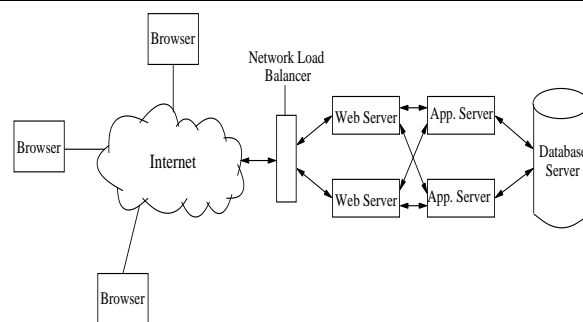


Figure 1. E-commerce System

tency. A single large database server is often used in such systems. Since most user requests invoke database queries, tuning the database to specifically support the workload type may provide better performance.

Understanding the workload characteristics of the database server is a prerequisite for studying its performance. The Database Management System (DBMS) is designed and tuned for traditional database workloads, i.e. On-line Transaction Processing (OLTP) and On-line Analytical Processing (OLAP).

E-commerce workloads may have different characteristics. Moreover, *dynamic caching* is often employed to reduce the load on the database server. Various dynamic caching technologies may be used in the system, and new caching approaches may emerge in the future. Dynamic cache may affect the database server workload and change workload classification, by pushing it further towards either an OLAP or OLTP workload. Database administrators, with the help of a workload classification tool [2] can use this information to adopt effective tuning strategies for the DBMS.

In this paper, we study the workload characteristics of the database server in a large E-commerce system, and investigate the impact of dynamic caching on the workload profiles. Specifically, we focus on classifying E-commerce workloads under dynamic caching.

We took snapshots of the database to determine the

active set of queries, the corresponding table accesses, and the transaction status of different workload profiles. The main finding is that dynamic caching does change the behavior of E-commerce activity profiles towards OLTP. Furthermore, *hybrid caching* has a more dramatic impact on this profile shifting than *query result caching*.

The remainder of this paper is organized as follows. We first introduce the background and related work of this study. We then describe the methodology and experiment setup used. After that, we present the classification of workload profiles of the database server and discuss the results. Finally we conclude the paper and outline possible future research directions.

2. Background and related work

2.1. The TPC-W benchmark

TPC-W is an industry-standard transactional Web E-commerce benchmark from the Transaction Processing Performance Council (TPC) [3]. The TPC-W benchmark simulates a breadth of activities of a business-oriented server system. It uses a web browser interface. The specific operational model of the system is an on-line bookstore. The store size is expressed by the number of items and emulated browsers (EBs).

TPC-W defines two classes of web interactions: *browse* and *order*. *Browse* interactions involve browsing the web site and searching the database (e.g., querying new products). *Order* interactions update the database (e.g., loading shopping carts). We are interested in the database transactions that result from these interactive web activities. TPC-W simulates three kinds of workload profiles by varying the ratio of the *browse* to *order* web interactions: *Browsing*, *Shopping*, and *Ordering*. Table 1 summarizes the percentage of the *Browse* and *Order* web interactions in each workload profile.

| Workload Profile | <i>Browse</i> Interactions | <i>Order</i> Interactions |
|------------------|----------------------------|---------------------------|
| Browsing | 95% | 5% |
| Shopping | 80% | 20% |
| Ordering | 50% | 50% |

Table 1. TPC-W workload profiles

2.2. Dynamic caching

Dynamic web pages are widely used in E-commerce systems. Generating these pages often requires executing queries at the database server and cannot be cached by traditional static caches such as web proxies.

Dynamic caches, which reside on separate machines other than the database server are designed to cache dynamic content. They can be roughly classified as *query result caching* and *table caching*.

Query result caching. In query result caching, the results of a query (or the web page segments generated from these results) are saved such that subsequent identical queries are served directly from the cache [4, 5]. A query result cache does not process database queries, providing only storage and simple lookup facilities.

Table caching. A table cache server contains some replicated portions of the database server [6, 7] and processes queries involving the cached tables. If a query involves other tables, the table cache server partitions the query into sub-queries, some of which will be processed locally. Infrequently changed tables are suitable caching candidates to minimize the update propagation cost. A table cache server is more complicated and costly to implement than a query result cache.

2.3. Workload classification under dynamic caching

OLAP and OLTP are two typical DBMS workload categories. In general, an OLAP workload performs complex queries on large amounts of data, while an OLTP workload performs intensive data-update transactions for a large user population. Fundamentally different DBMS configurations are required [8, 9] to support particular workload types.

Database workload characterization/classification has received some recent attention. Elnaffar [2, 10] classified the TPC-W Browsing profile and Ordering profile as light OLAP and OLTP respectively. Wasserman et al. [11] revealed there are four different groups of queries in a typical OLAP workload. Unfortunately, dynamic caching was not considered in either study. Our previous study [12] has shown that dynamic caches located between web users and a DBMS in E-commerce systems, are effective at reducing the system response time and the page reference locality of the TPC-W workload profiles. However, the effect of dynamic caching on the classification of the TPC-W workload profiles remains unanswered. This is of interest because changes in the workload type (i.e. the proportion of OLTP vs. OLAP in the workload mix) require DBMS configuration changes to optimize overall performance.

3. Methodology and experiment setup

3.1. Methodology

The classification of DBMS workloads can be viewed as a data mining process, which includes two steps: training a classifier and applying it to target classification candidates [2, 10]. In this study, database snapshots collected during the execution of TPC-W workload profiles without caching are used to train the classifier. Then snapshots of the workload profile execution with caching are used for classification purposes.

Each snapshot can be viewed as a vector whose dimensions are the attributes that summarize the activities in the system during the execution as follows:

- **Queries Proportion:** The proportion of SELECT statements to UPDATE/INSERT/DELETE statements. The Queries Proportion is generally higher in OLAP than in OLTP.
- **Pages Read:** OLAP usually accesses larger portions of the database than OLTP.
- **Rows Selected:** OLAP tends to select more rows than OLTP.
- **Number of Sorts:** OLAP typically performs a larger number of sorts than OLTP.
- **Ratio of Index Usage:** The ratio of data pages obtained from indexes to the pages obtained from tables, in order to serve a query. This ratio is expected higher in an OLTP workload than in a OLAP workload.
- **Logging:** The number of pages accessed of the log file of the database. OLTP tends to have relatively more UPDATE/INSERT/DELETE transactions, thus more logging activity.

These attributes reflect the behavioural differences between OLAP and OLTP. In the first step, we choose TPC-W Browsing and Ordering profiles to be representative of OLAP and OLTP workloads, respectively, to train the classifier (i.e. establish its rules for the classification). In the second step, the trained classifier is applied to the database snapshots from the TPC-W Browsing, Shopping, and Ordering profiles under different dynamic caching mechanisms.

The decision tree model J48 from Weka [13] is used in this study as the classifier because it can extract the classification rules easily, enabling more straightforward justification the results in comparison with other techniques such as neural networks.

The TPC-W Java implementation [14] is used to generate the workload profiles. The system is composed of *emulated browsers* (EBs), *application logic*, *dynamic cache emulator*, and *database server*. Although we use a benchmark E-commerce system rather than a real system, the analysis methodology employed in this paper

is general, providing a controlled environment to emulate most key features affecting system performance.

In previous work [12], we described and emulated three kinds of dynamic caches: *query-result cache*, *table cache*, and *hybrid cache*. We designed the emulators to make sure that the workloads to the database server using these emulators are the same as those using the corresponding real caches. In this study, we used the query-result cache and the hybrid cache. The experimental results of the table cache are not shown because our previous study showed that hybrid cache has very similar performance characteristics as table cache.

3.2. Configurations and parameters

The machine we used for the experiments is an IBM eServer xSeries 255 with four hyper-threaded Intel Xeon MP 1.5GHz processors, 8GB RAM, and twelve 34.7GB IBM U320 disks attached to two IBM ServeRAID-4Lx Ultra160 SCSI controllers, implementing RAID-0 (15,000RPM with 3.6 ms average seek time). The operating system is Windows 2000 Advanced Server, and the DBMS is IBM DB2 8.1. The TPC-W database is built with 10,000 items and 3,000 EBs for a total size of 17.6GB, and a page of 4KB.

3.3. Snapshot collection and processing

We developed a light-weight snapshot-collecting tool. With a snapshot interval of one second, we observed that many SQL statements complete within that size interval. This is not always the case, especially for the workloads that contain complex queries. We therefore coalesce the snapshots until we encompass at least one statement completion. We then normalize the consolidated snapshots with respect to the number of SQL statements executed such that each normalized snapshot describes the characteristics of a single SQL statement.

During the snapshot collection phase, we run a workload for about 25 minutes. Snapshots are collected after 5 minutes of warm-up and recorded during the next 15 minutes, producing about 700 raw snapshots.

4. Workload classification

4.1. Attributes without caching

Initially, the snapshots of the TPC-W Browsing and Ordering profiles without caching are used to train the classifier. Figure 2 shows the attribute values of Browsing, Shopping, and Ordering. All the values for each profile are normalized with respect to the Browsing profile. The values used in this figure are the average of those attribute values with respect to the snapshots obtained from running the benchmark.

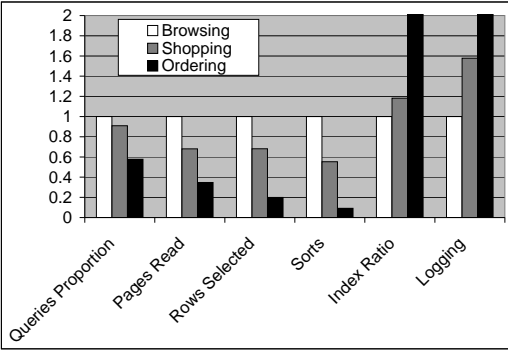


Figure 2. Snapshot Attributes

The figure shows that the snapshots attributes of the Browsing profile are quite different from those of the Ordering profile. The Browsing profile always gets higher values for Queries Proportion, Pages Read, Rows Selected, and Sorts, while the ordering profile has much higher values for Index Ratio and Logging. The attributes of Shopping profile are always between the Browsing profile and the Ordering profile. This makes sense because the ratio of request types is between the Browsing and Ordering profile by definition [3].

To further understand the attributes shown in Figure 2, we also plot the distributions of attribute values in Figure 3 in three pairs, in which 50 random attribute values are plotted. The left column contains the results without caching. There are 12 other pairwise comparisons that all show similar trends. The Shopping profile is not shown, as it always falls in between the Browsing profile and Ordering profile.

This figure indicates that the attributes of Browsing profile and Ordering profile span a relative large range, but they are clearly in two distinct groups. The Ordering profile also has several attributes whose values are more tightly clustered at the extremes of the range of observed values for those attributes. This indicates a more uniform and predictable use of the database resources. An exception is the Queries-Proportion attribute, where the Browsing profile is more homogeneous. Nevertheless, it seems as though the Ordering profile is overall more homogeneous.

In Figure 4, we show the CDF of two of the attributes: Pages Read and Index Ratio. These are the attributes for which caching has the most significant effect. The left column shows the values without caching. There is a clear distinction between the browsing and ordering workload types for both attributes, and between all workload types for the Pages Read metric.

4.2. Attributes under caching

The distributions of attribute values of TPC-W Browsing and Ordering profiles under query-result cache and hybrid cache are also shown in Figure 3. This shows that query-result cache (center column) and hybrid cache (right column) effectively reduce the Queries Proportion, Pages Read, Rows Selected, and Sorts, while increase Index Ratio and Logging, both OLTP characteristics. This is because the dynamic caching mechanisms filter out a large portion of queries destined for the database server. The distributions of most of the Shopping profile attributes still remain between the Browsing profile and the Ordering profile under caching, and are not shown. The only attribute which moves Shopping outside Browsing and Ordering is the Index Ratio. We also see that hybrid caching is more effective at moving the Browsing profile towards OLTP.

The effect of caching on the distribution of individual parameter values is also shown in the center (Query Result Cache) and right (Hybrid Cache) columns of figure 4. With regard to Pages Read, both types of caching completely blur the distinction between the workloads, so that they are both more similar to OLTP than without caching. The hybrid cache has a dramatic effect on the Index Ratio as both workloads show a very high Index Ratio and there is very little difference between the workload types.

4.3. Classification under caching

Figure 5 shows the prediction results from the trained classifier. Those results with over 90% confidence are used. The y-axis is the percentage of “OLTP” predicted by the classifier over all the predicted results from the target snapshots of a certain workload profile. It is clearly shown that the classifier predicts that all the workload profiles (Browsing, Shopping, and Ordering) under hybrid cache are OLTP-like workloads. Query-result cache also pushes TPC-W Browsing and Shopping profiles towards OLTP workloads.

5. Conclusions and future work

The performance of the database server is crucial to an E-commerce system. The workload seen by the database server is dramatically changed by the use of dynamic caching. Understanding these changes is important to design, tuning and capacity planning.

We studied the classification of workloads on the database server in a benchmark E-commerce system. Different kinds of dynamic caching mechanisms were used, including query result caching, table caching, and hybrid caching.

The experiment results show that dynamic caching dramatically pushes the TPC-W Browsing and Shop-

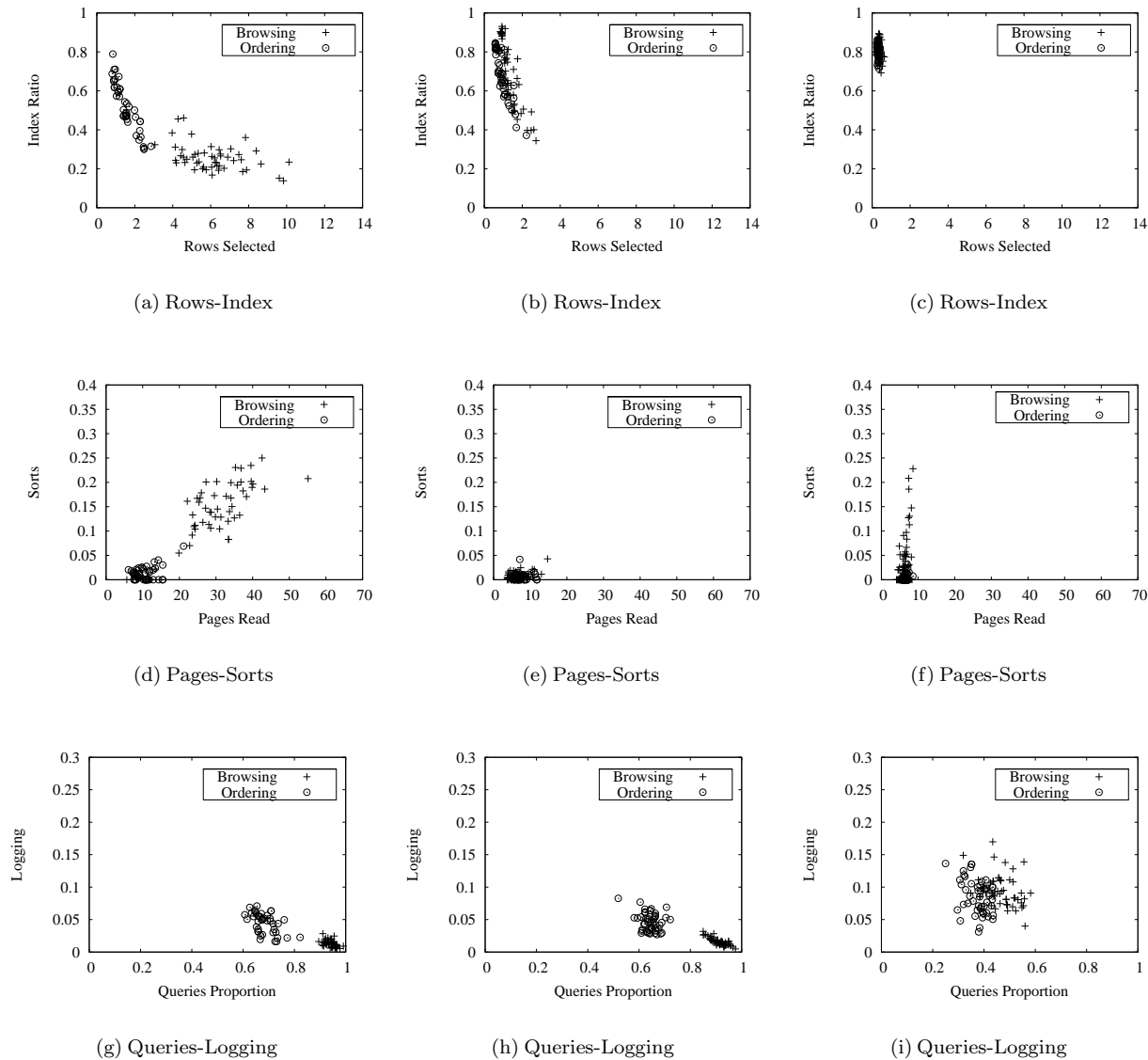


Figure 3. Workload Attributes (no-cache, Query Result Cache, Hybrid Cache)

ping profiles towards OLTP workloads. This suggests that if caching is deployed, a DBMS tuned for OLAP workloads similar to TPC-W Browsing should be adjusted to an OLTP-only DBMS. As well, if hybrid caching is used, the benchmarking of E-commerce workload can just focus on the performance of the OLTP (Ordering) transactions. As indicated in all the graphs, this profile has a more homogeneous use of the majority of the database resources.

As E-commerce Web sites provide better searching and data mining services, the proportion of cache-able queries may increase in future E-commerce workloads. Using dynamic caching may bring more benefits to such systems.

Further work will include investigating the caching effects in a real E-commerce environment.

Acknowledgments The authors would like to thank Wenguang Wang for discussions regarding the simulation framework.

References

- [1] United Nations Conference on Trade and Development Team. E-Commerce and Development Report 2002, 2002. UNCTAD/SDTE/ECB/2.
- [2] S. Elnaffar. A Methodology for Auto-recognizing DBMS Workloads. In *Proc. of CASCON 2002*, pages 74–88, Toronto, ON, October 2002.
- [3] TPC BenchmarkTM W. <http://www.tpc.org/tpcw/>, 2004.
- [4] A. Datta, K. Dutta, H. Thomas, D. VanderMeer, Suresha, and K. Ramamritham. Proxy-based Acceleration of Dynamically Generated Content on the World Wide

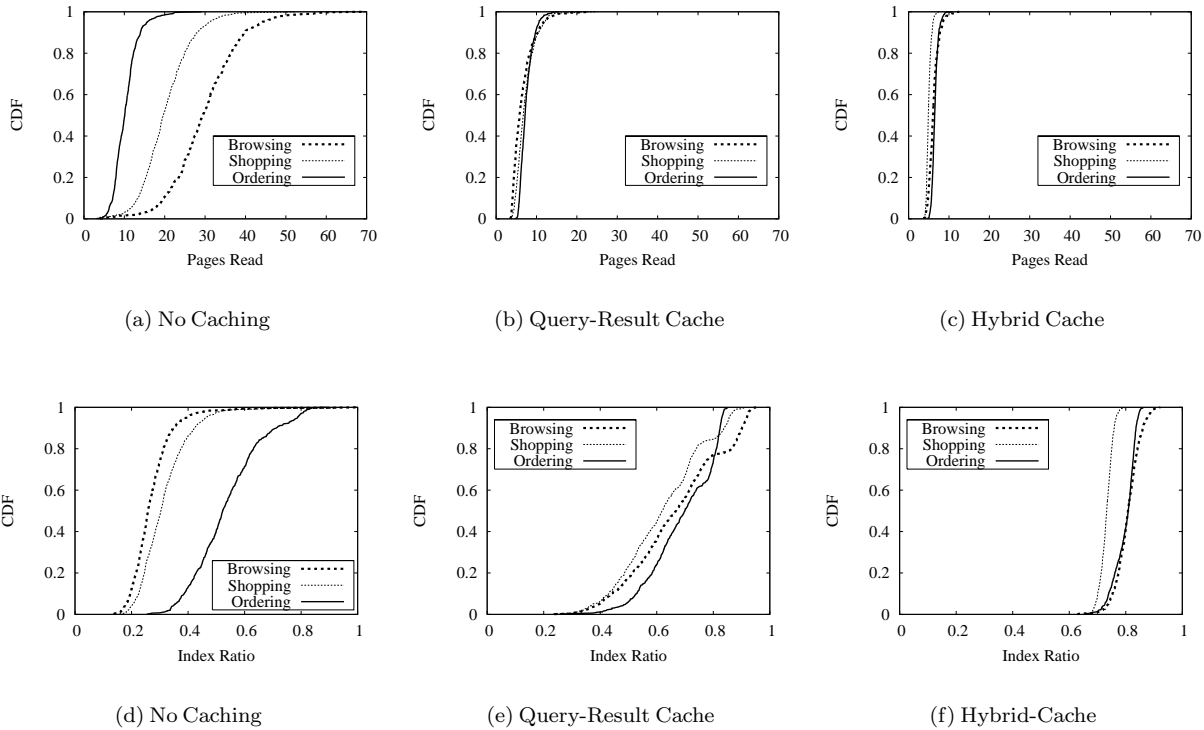


Figure 4. Cumulative Distributions of Attribute Values

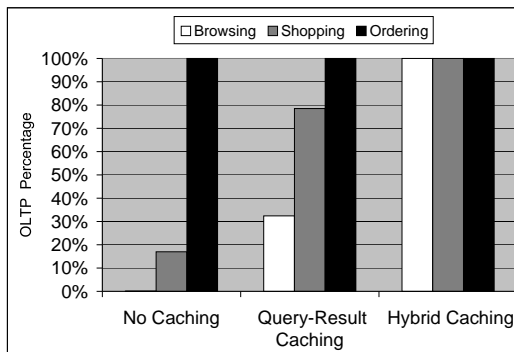


Figure 5. OLTP Percentage of TPC-W Profiles

- Web: An Approach and Implementation. In *Proc. of ACM SIGMOD'02*, pages 97–108, Madison, WI, 2002.
- [5] K. S. Candan, D. Agrawal, W. Li, O. Po, and W. Hsiung. View Invalidation for Dynamic Content Caching in Multitiered Architectures. In *Proc. of VLDB 2002*, pages 562–573, Hong Kong, China, August 2002.
- [6] The TimesTen Team. High Performance and Scalability through Application-Tier, In-Memory Data Management. In *Proc. of VLDB'00*, pages 677–680, Cairo, Egypt, 2000.
- [7] M. Altinel, C. Bornhövd, S. Krishnamurthy, C. Mohan, H. Pirahesh, and B. Reinwald. Cache Tables: Paving

the Way for an Adaptive Database Cache. In *Proc. of VLDB'03*, pages 718–729, Berlin, Germany, 2003.

- [8] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1):65–74, 1997.
- [9] L. A. Barroso, K. Gharachorloo, and E. Bugnion. Memory System Characterization of Commercial Workloads. In *Proc. of the 25th International Symposium on Computer Architecture*, pages 3–14, Barcelona, Spain, June 1998.
- [10] S. Elnaffar, P. Martin, and R. Horman. Automatically Classifying Database Workloads. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM ACM 02)*, pages 622–624, McLean, VA, November 2002.
- [11] T. J. Wasserman, P. Martin, D. B. Skillicorn, and H. Rizvi. Developing a Characterization of Business Intelligence Workloads for Sizing New Database Systems. In *Proc. of the 7th ACM International Workshop on Data Warehousing and OLAP*, pages 7–13, Washington, DC, November 2004.
- [12] F. Liu, Y. Zhao, W. Wang, and D. Makaroff. Database Server Workload Characterization in an E-commerce Environment. In *Proc. of MASCOTS 2004*, pages 475–483, Volendam, Netherlands, October 2004.
- [13] I. H. Witten, E. Frank, and M. Kaufmann. *Data Mining*. Morgan Kaufmann, San Francisco, 1st edition, 1999.
- [14] Java TPC-W Implementation Distribution. Website, <http://www.ece.wisc.edu/~pharm/tpcw.shtml>.