Benchmarking intelligent information integration – A generic construct-based model

Jia-Lang Seng [a,*], M.I. Lin [b]

[a] Department of Graduate School of Accountancy, College of Commerce, National Chengchi University, Taipei, Taiwan
[b] Senior Systems Engineer, IT & EC Sector, COMPAL Electronics, Taipei, Taiwan

ARTICLE INFO

ABSTRACT

Benchmarks are vital tools in the performance measurement and evaluation of computer hardware and software systems. Standard benchmarks such as the TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS³AP, OO1, OO7, XOO7 benchmarks have been used to assess the system performance. These benchmarks are domain-specific in that they model typical applications and tie to a problem domain. Test results from these benchmarks are estimates of possible system performance for certain pre-determined problem types. When the user domain differs from the standard problem domain or when the application workload is divergent from the standard workload, they do not provide an accurate way to measure the system performance of the user problem domain. System performance of the actual problem domain in terms of data and transactions may vary significantly from the standard benchmarks. In this research, we address the issue of domain boundness and workload boundness which results in the ir-representative and ir-reproducible performance reading. We tackle the issue by proposing a domain-independent and workload-independent benchmark method which is developed from the perspective of the user requirements. We present a user-driven workload model to develop a benchmark in a process of workload requirements representation, transformation, and generation. We aim to create a more generalized and precise evaluation method which derives test suites from the actual user domain and application. The benchmark method comprises three main components. They are a high-level workload specification scheme, a translator of the scheme, and a set of generators to generate the test database and the test suite. The specification scheme is used to formalize the workload requirements. The translator is used to transform the specification. The generator is used to produce the test database and the test workload. In web search, the generic constructs are main common carriers we adopt to capture and compose the workload requirements. We determine the requirements via the analysis of literature study. In this study, we have conducted ten baseline experiments to validate the feasibility and validity of the benchmark method. An experimental prototype is built to execute these experiments. Experimental results demonstrate that the method is capable of modeling the standard benchmarks as well as more general benchmark requirements.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

1.1. Research motivation

A benchmark is a standard by which something can be measured or judged. A computer system benchmark is a set of executable instructions to be enforced in controlled experiments to compare two or more computer hardware and software systems. Hence, benchmarking is the process of evaluating different hardware systems or reviewing different software systems on the same or different hardware platforms. A web search service benchmark is therefore a standard set of executable instructions which are used to measure and compare the relative and quantitative performance of two or more systems through the execution of controlled experiments. Benchmark data such as throughput, jobs per time unit, response time, time per job unit, price and performance ratio, and other measures serve to predict price and performance and help us to procure systems, plan capacity, uncover bottlenecks, and govern information resources for various user, developer, and management groups (Anon et al., 1985).

Examples are the TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS³AP, OO1, OO7, XOO7 standard benchmarks that have been used to assess the system performance. These benchmarks are domain-specific in that they model typical applications and tie to a problem domain. Test results from these benchmarks

* Corresponding author.
  E-mail addresses: seng@nccu.edu.tw (J.-L. Seng), jljan2@gmail.com (M.I. Lin).

are estimates of possible system performance for certain pre-determined problem types. When the user domain differs from the standard problem domain or when the application workload is divergent from the standard workload, they do not provide an accurate way to measure the system performance of the user problem domain. System performance of the actual problem domain in terms of data and transactions may vary significantly from the standard benchmarks. Performance measurement and evaluation is crucial in the development and advance of web search technology. A more open and generic benchmark method is needed to provide a more representative and reproducible workload model and performance profile.

1.2. Research problem

Domain boundness and workload boundness are the research problem we try to tackle in this research. As described above, standard benchmarks model certain application types in a pre-determined problem domain. They represent a fixed problem set presented to the proposed system. When the user domain differs from the standard domain or when the user workload deviates from the standard workload, the test results vary significantly in the real setting and under the actual application context. Users cannot reproduce the test results and predict the performance. The reason is because benchmark results are highly dependent upon the real workload and the actual application. The standard test workload cannot represent the real workload and the test suite cannot accommodate the application requirement. Standard benchmarks cannot measure the effects of the user problem on the target system nor generate the realistic and meaningful test results.

In this research, we address the issue by proposing a domain-independent and workload-independent benchmark method which is developed from the perspective of the user requirements. We propose to develop a more generalized and more precise performance evaluation method from the perspective of the common carriers of workload requirements. We create a user-driven approach which models the benchmark development in a process of workload requirements representation, transformation, and generation.

1.3. Research approach

Benchmarks can be synthetic or empirical. Synthetic benchmarks model the typical applications in a problem domain and create the synthetic workload. Empirical benchmarks utilize the real data and tests. Though real workloads are ideal tests, the costs of re-implementation of the actual systems usually outweigh the benefits obtained. Synthetic benchmarks are therefore the common approach chosen by developers and managers. Further, benchmark experiments are composed of the experimental factors and the performance metrics. Experimental factors represent the variables which can affect the performance of the systems. Performance metrics are the quantitative measurements to be collected and observed in the benchmark experiments. They represent the set of independent variables and dependent variables to be modeled and formulated in the benchmark.

A workload is the amount of work assigned to or performed by a worker or unit of workers in a given time period. The workload is the amount of work assigned to or performed by a system in a given period of time. The loads are best described by the amount of work, the rate at which the work is created, and the characteristics, distribution, and content of the work. Conventionally, workload modeling and characterization start with the domain survey, observation, and data collection, and continue with a study of the main components and their characteristics. In general, the workload components consist of the data, operations, and control.

In specific, workload analysis involves the data analysis and the operation analysis. We analyze the size of the data, the number of records, the length of records, the types of attributes, the value distributions and correlations, the keys and indexing, the hit ratios, the selectivity factors. We investigate the complexity of operations, the correlation of operation, the data input into the operation, the attributes and objects used by the operation, the result size, and the output mode. These are further examined with the control analysis of the duration of test, the number of user, the order of test, the number of repetition, the frequency and distribution of test, and the performance metrics.

In the web search context, we develop a benchmark method that comprises a workload requirements specification scheme, a scheme translator, and a set of benchmark generators. We adopt the common carrier of generic constructs. We analyze the key web search algorithms and formulate the generic constructs. The generic constructs describe the page structure and the query structure of web search that is not tied to a per-determined search engine.

1.3.1. Workload specification scheme
The workload specification scheme is designed to model the application requirements. It is a high-level generic construct concept to describe requirements concerning data, operation, and control. A generic construct is the basic unit of operand. An operation is the basic unit of operator. The collection of a generic construct and an operation formulate a workload unit. Each workload unit becomes a building block to compose a larger workload unit.

1.3.2. Scheme translator
The scheme translator is created with a set of lexical rules and a set of syntactical rules to translate the workload specification. It performs the code generation and produces three output specifications. One is the data specification. The other is the operation specification. Another is the control specification.

1.3.3. Data generator
The data generator is made up of a set of data generation procedures which are used to create the test database according to the data distribution specification.

1.3.4. Operation generator
The operation generator is made up of a set of operation generation procedures to generate the search operations. These procedures select operations, determine operation precedence, schedule arrivals, prepare input data, issue tests, handle queues, gather and report time statistics.

1.3.5. Control generator
The control generator is made up of a set of control generation procedures to generate the control scripts which are used to drive and supervise the experiment execution.

2. Literature review

2.1. XML query capability

Benchmarking the XML data management systems should consider many factors. Designing a set of comprehensive queries to test the XML databases' performance is an important point. XML query languages should capture the whole characteristics of a XML document, and the functionalities they provide would influence the query performance. The W3C XML Query Language work-