Database Systems Performance Evaluation Techniques

Subharthi Paul subharthipaul@gmail.com (A project report written under the guidance of Prof. Raj Jain)

Download 

Abstract

The last few decades has seen a huge transformation in the way businesses are conducted. There has been a paradigm shift from product portfolio based marketing strategies to customer focused marketing strategies. The growth and diversity of the market has greatly profited consumers through higher availability, better quality and lower prices. The same factors however has made it more difficult for businesses to maintain their competitive edge over one another and hence has forced them to think beyond their product portfolio and look at other means to gain higher visibility and customer satisfaction, maintaining all the while their core advantages on pricing and product through improved and more efficient methods of manufacturing and distribution. The advent and spread of computers and networking has been one of the single largest factors that has spurred and aided this enormous movement. More specifically, database management systems now form the core of almost all enterprise logic and business intelligence solutions. This survey tries to emphasize the importance of database systems in enterprise setups and looks at the methods and metrics that are used to evaluate the performance of these database systems.

Keywords

Database Systems, Transaction Processing, Performance Evaluation Techniques, Database Benchmarking

Table of Contents

---

1. Introduction

The last few decades has seen a huge transformation in the way businesses are conducted. There has been a paradigm shift from product portfolio based marketing strategies to customer focused marketing strategies. The growth and diversity of the market has greatly profited consumers through higher availability, better quality and lower prices. The same factors however has made it more difficult for businesses to maintain their competitive edge over one another and hence has forced them to think beyond their product portfolio and look at other means to gain higher visibility and customer satisfaction, maintaining all the while their core advantages on pricing and product through improved and more efficient methods of manufacturing and distribution. The advent and spread of computers and networking has been one of the single largest factors that has spurred and aided this enormous movement. More specifically, database management systems now form the core of almost all enterprise logic and business intelligence solutions.

Database Systems are one of the key enabling forces behind business transformations. Apart from supporting enterprise logic they also enable business intelligence. Information is the key to success in today's businesses. However, maintaining information in logically consistent and feasibly retrievable format is a daunting task. More so with the added complications of transaction consistency management, synchronization across multiple repositories spread geographically across the globe, failover management and redundancy management, today's database systems are truly state-of-the-art high performance software systems.

Apart from managing a plethora of complicated tasks, database management systems also need to be efficient in terms of storage and speed. Businesses have a tendency to store un-required historical data often as a result of poor data planning or less frequently owing to federal obligations or consumer law. Dynamic addition and deletion of data from the database also pose a challenge to maintaining an efficient data retrieval mechanism. Though, limited in speed by some sense due to hardware limitations, database systems nonetheless need to achieve full throttle through efficient storage and retrieval techniques. Another factor that often hurt database performance is ill-written computationally expensive queries. Often, such situations are beyond control of the database system and require external performance tuning by experts.

As is true for most systems, reliability, availability and fault-tolerance is a huge concern for database systems. Reliability of a system is generally improved through redundancy. Modern businesses cannot afford to loose data or present wrong data. Modern business activities are highly centered around and dependant on electronic data. Modern database systems thus need to build in high reliability mechanisms in their designs. Availability is another issue that concerns a lot of businesses. As an example, "Netflix" an online DVD rental business receives online requests for 1.9 million DVD's everyday. An outage for 10 minutes cost huge losses to these businesses. Similar is the case for a lot of other businesses. Amazon.com stands to loose approximately $31000 per minute for a global outage [1]. Clearly, availability is a key metric for measuring the performance of database systems.

Another metric, which is more qualitative in nature, but extremely important is security. It is generally difficult to measure the level of security of any system unless its security vulnerabilities are exposed.

Database systems often store sensitive enterprise and customer data that if inappropriately used may cause huge monetary and business losses for the organization. Hence, though difficult to quantify through standard testing procedures, database systems strive for continual upgrade of their security features.

Performance evaluation of database systems is thus an important concern. However, easier said than done, performance evaluation of database system is a non-trivial activity, made more complicated by the existence of different flavors of database systems fine tuned for serving specific requirements. However performance analysts try to identify certain key aspects generally desired of all database systems and try to define benchmarks for them. In the rest of this survey, we shall provide a formal definition of database systems followed by a few methods to categorize or classify database systems. This shall be followed by a look at the various performance evaluation techniques that are employed to benchmark database systems, some of the key benchmarking techniques used in practice in the industry and some open source benchmarking schemes available for use in the public domain.

---

2. Database Management Systems: Formal Definition and Classification

A Database Management System (DBMS) is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMS' are categorized according to their data structures or types. It is a set of prewritten programs that are used to store, update and retrieve a Database [2].

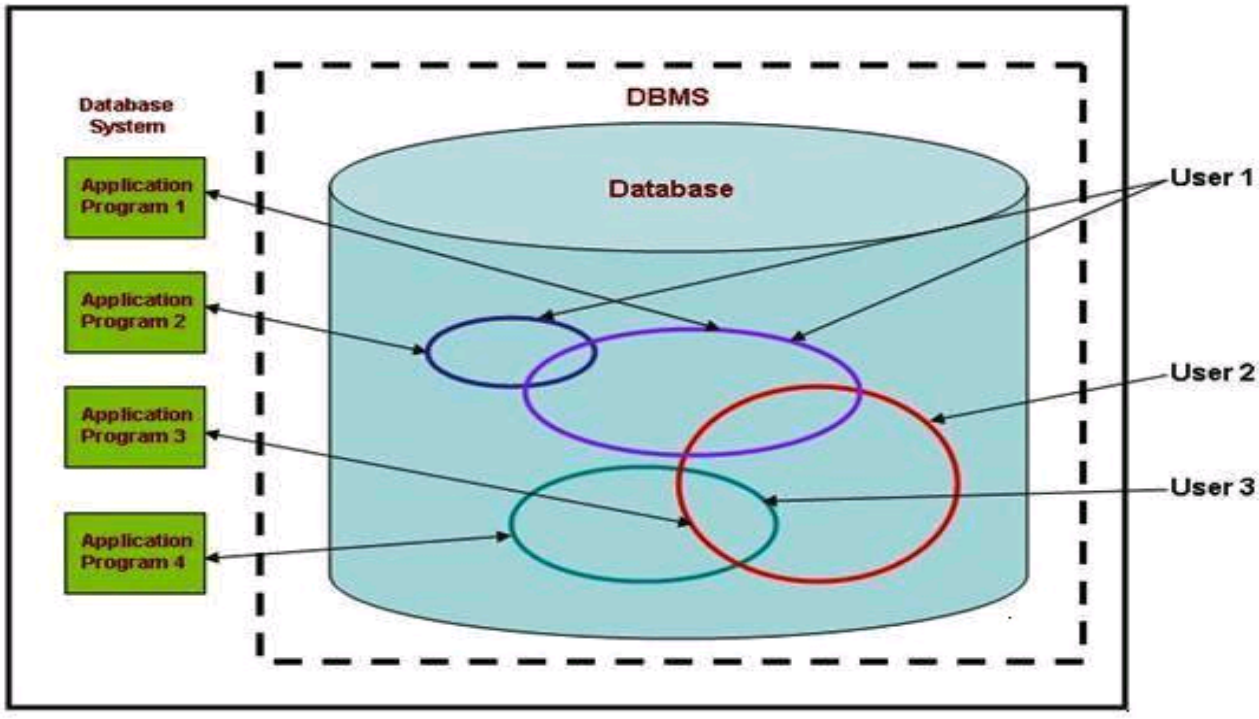

Figure 1 - A generic high level view of a Database Management System

Figure 1 presents the generic high level view of a database management system. The "Database Management System is a collection of software programs that allow multiple users to access, create, update and retrieve data from and to the database and the "Databae" is a shared resource that is at the centre of such a system. The database's functionality is optimal storage and retrieval of data, maintain correctness of the data, and maintaining consistency of the system at all times.

There are two basic criterions on which database systems are generally classified. One of them is based on Data Modeling while the other is Functional Categorization. Here, we briefly discuss each of these classification techniques. A few functional categorization shall be discussed in the next section when we discuss "Database Performance Evaluation Techniques for specialized Databases" in Section

2.1 Classification based on Data Modeling

2.1.1 Hierarchical Model:

In a hierarchical model data is represented as having a parent-child relationship among each other and is organized in a tree-like structure. The organization of data enforces a structure wherein a parent can have many children but a child can have only one parent. Thus, this model inherently forces repetitions of data at the child levels. The records have 1:N or more generally a "one-to-many" relationship between them. This was one of the first data base models to be used and implemented in IBM's Information Management System(IMS) [3] . Hierarchical model was the most intuitive way to represent real-world data but was not the most optimal one.This model was later replaced by a more efficient and optimal model called the "Relational Model" that we shall discuss next.

2.1.2 Relational Model:

The Relational Model [4] was formulated by Edgar Codd and is one of the most influential model that has governed the implementation of some of the best known Database systems. The model is based on "first order predicate logic". In a representation such as A = {x| P(x) }, P(x) is the predicate that makes a descriptive statement about the elements of set A, such as "All positive integers less than 1000" results in a set A = { 1,..,999}. So, a predicate maps an "entity" to a "truth table". In the above example, 'x' is an entity and P(x) is a mapping which determines whether 'x' belongs to set A or not. Now, suppose that we have a huge domain and we need to make some statements that apply selectively to some or all members in the domain. The formal language that allows us to make such a statement is called "first order logic". In first order logic statements, a predicate either takes the role of a defining the "property" of an entity or the "relationship between entities". Further and in-depth description on the mathematical foundations of the relational model is beyond the scope of the present work.

So, in the relational model, data is represented using a set of predicates over a finite set of variables that model the belongingness of certain values to a certain sets and the constraints that apply on them. The relational model, owing to its strong foundations in mathematical formal logic is extremely powerful in representation and at the same time efficient in terms of storage requirements (by removing redundancies due to repeated data) and also speed of retrieval/storage.

2.1.3 Network Model:

The network model [5] can be seen as a generalization of the hierarchical model. In this model, each data object can have multiple parents and each parent object can have multiple children. This the network model forms a "lattice-type" structure in contrast to the "tree-like structure" of the hierarchical model. The network model represents real-world data relationship more naturally and under less constrained environment than the hierarchical model.

2.1.4 Object-Relational Model:

The Object-Relational Model [6] is similar to the relational model except for additions of object-oriented concepts in modeling. The data modeling is done using relational concepts while the object-oriented concepts

facilitate complex modeling of the data and its relationship with the methods of manipulation/retrieval and storage. This is one of the newer and more powerful models of all the models discussed in this section.

There are lots of other models classifying databases based on the object-models such as Semi-structure Model, Associative Model, Entity-Attribute-Value Model etc., but a detailed study of each of these is beyond the present scope. The aim of the above classification was to attain a minimal understanding that would help us appreciate the design of performance analysis methods for data bases.

---

3. A General Approach to Database Performance Evaluation

In this section we discuss some of the more "general" methods that can be used for database performance evaluation. The word "general" is binding to systems, meaning that the approaches mentioned here are generally true for "systems" with a special focus on database systems.

According to [7], performance analysis of database systems serve two basic purposes:

1. For the evaluation of the best configuration and operating environment of a single database system, and

2. Studying two or more database systems and providing a systematic comparision of the systems.

Accordingly, some of the analytical modeling methods for evaluating systems that are applicable for database systems too are:

1. Queuing Models: Queuing models are effective to study the dynamics of a database system when it is modeled as a multi-component system with resource allocation constraints and jobs moving around from one component to another. Examples of such dynamic studies are concurrent transaction control algorithms, data allocation and management in distributed database systems etc.

2. Cost Models: Cost Models are useful in studying the cost in terms of Physical storage and query processing time. The cost model gives some real insight into the actual physical structure and performance of a database system.

3. Simulation Modeling: A simulation Modeling is more effective for obtaining better estimates since it not only analyses the database system in isolation but can effectively analyze the database system with the application program running on top of it and the database system itself operating within the constrained environment of an operating system on a real physical hardware.

4. Benchmarking: Benchmarking is the best method when multiple database systems need to be evaluated against each other but suffer from the inherent setback that it assumes all systems to be fully installed and operational. Benchmarking relies on the effectiveness of the synthetic workloads. Real workloads are non repeatable and hence not good for effective benchmarking.

[7] presents a 3 step process for benchmark evaluation

a. Benchmark design

b. Benchmark Execution

c. Benchmark analysis

Our study makes a delves deeper into the various database system evaluation benchmarks and we leave the

more detailed analysis of Database systems benchmark studies to a later section where we see different real-life benchmarking techniques.

[8] proposes a set of methods for database performance evaluation assuming the system to be operating in a multi-user environment. Accordingly the three factors that effect the performance of a database system in a multi-user environment are:

1. Multi-programming level

2. Query Mix

3. Extent of data Sharing

Data sharing is the condition of concurrent access of a data object by multiple processes. The interesting factor here is that of the query mix. A proper query mix needs to test the appropriate levels of CPU and disk utilization required to serve a particular query. The query mix needs to properly represent a true multi-user environment. Also, the query mix may be designed to represent a certain percentage of data sharing. Once these have been figured out, the query-mix forms a representative benchmark program and multiple copies of the bench-mark program are issued concurrently to simulated multi-programming effects. Also, different query-mixes allow diversity in the experimental design conditions. The response variable studied is system throughput and response time.

Summarizing, in this section, the focus was primarily to study the performance evaluation techniques considering the "general system criterion" of a database system. In the next section we look at the performance evaluation techniques more specialized for particular database system types.

---

4. Database Performance Evaluation Techniques for specialized Databases

In the last section we discussed about a few performance evaluation techniques that are extremely general and apply to almost all database systems and as such to most generic systems. In this section, we refine our discussion to the techniques developed specially for the performance evaluation of databases of various types or in special application specific areas.

4.1 Real-time Database Systems performance Evaluation

Simulation studies on real time database systems are limited by the unavailability of realistic workloads and hence fail to test the system in real dynamic situations where they have to operate. Real-time database systems are generally mission-critical and has high QoS requirements. Proper evaluation techniques for the performance real-time systems is thus extremely important and at the same time extremely challenging. [10] develops "Chronos", a real-time database test-bed to evaluate performance of such systems.

Unlike non-real time test-beds, chronos does not evaluate the average throughput and average response time. These are not the representative metrics to test real-time database systems which have added constraints of "data freshness" , "deadlines" and "time bound". Chronos provides methods to vary transactions and workloads applied to a real-time database system and studies the timeliness and freshness of data metrics under multiple load conditions.

Chronos also implements a QoS management scheme which incorporates overload detection, admission control and adaptive policy update. As mentioned previously, QoS requirements of real time databases are

generally high and so it is extremely important to be able to quantify the system state based on such QoS parameters. A good real-time system should be one that can correctly evaluate its state and provide QoS guarantees in terms of timeliness, transaction time bounds and delays.

4.2 Web- Database Systems performance Evaluation

Web databases serve the back-ends of Web-Servers. Web databases are the classic examples of high load, high performance database systems. The most common performance metrics for susch database systems are average response time and throughput, fault tolerance, scalability etc. Availability metrics such as MTTF(Mean-time to failure), MTTR(Mean time to repair) etc are also extremely important metrics for such systems. The challenges in performance evaluation of web databases is the proper workload selection representative of the typical load as well as highly loaded situations. [11] proposes a workload that consists of a query mix as shown in Table 1. The objective behind the query design is to provide fundamental queries that can be elementarily grouped to provide different load and resource utilization condidtions.

Class	Description	Load on DB Servers		Load Web Servers		Load on Network
		CPU	DISK	CPU	DISK	
1	Complex query with small result size	High	Low	Low	N/A	Low
2	Complex query with large result size	High	High	High	N/A	High
3	Simple query with small result size	Low	Low	Low	N/A	Low
4	Simple query with large result size	Low	High	High	N/A	High
5	No query, small file size	N/A	N/A	Low	Low	Low
6	No query, large file size	N/A	N/A	High	High	High

Table 1: Workload classification of web database systems [11]

Another architectural component typical of web-database systems are various input and output queues. Transactions pile up at both these queues based on the arrival rate and response time of the database. Modeling a simulation or test bench for a web-database system requires considering this architectural element for proper representation of the system in an operational environment.

4.3 Enterprise Data Mining System Performance Evaluation

Enterprise database systems form the heart of enterprise operations as well as enterprise intelligence. They are huge database systems (often called :data mining systems") often storing historical and redundant data. They are the basis of all business intelligence processes and decision support systems. Efficient storage and retrieval from suce huge databases ("data-warehouses") require special techniques of aggregation and classification of data while storing. The process of enterprise decision making involves high level canned queries by managers, mapping of these high level queries into specific predicate rules and retrieval of the datasets that match those rules. Also, often the result of a query involves data retrieval from multiple distributed databases. The performance evaluation of such systems entail mostly the performance evaluation of data mining algorithms. [13] proposes a test bed for the performance evaluation of frequent pattern matching algorithms. It incorporates a synthetic data generator that can generate all kinds of data following a

wide variety of frequency distribution patterns, as set of classical pattern matching techniques and a performance database that records the performance of the algorithm on each generated data sets. The main idea is to test the data mining system with reference to specific algorithms under stress conditions of randomly generated data.

4.4 Object Oriented Systems Performance Evaluation

[14] presents a comprehensive study on the comparison of object oriented database systems. The standard benchmarks to evaluate object oriented database systems are OO1, OO7, HyperModel, and ACOB. We shall discuss more about these benchmarks in later section. For the current discussion we try to isolate the issues specific to OODBMS's performance evaluation and the techniques employed to study them. The chief factors governing the performance of such database systems are evaluated using object operations. The object oriented database schema consists of different types of objects(simple and complex) and the relationships(direct, hierarchical etc) between them. The evaluation technique for such object oriented schema involves the development of operations on these objects such as dense traversal, sparse traversal, string search, join etc. Also other system level characteristics such as number of users, load etc are varied just like techniques for evaluating other database systems. The main idea is to generate queries on object oriented schema to test the effectiveness and design optimization of the object design and then testing for system level parameters at the same time.

The specific techniques discussed for each type of database systems in this section warrant a deeper study into each of these evaluation methods and also just presents a preliminary survey to the great diversity that is hidden behind the seemingly simple task of storing and retrieving data.

---

5. Database Systems: Performance Evaluation Benchmarks

Having established the importance of database system in the last few sections, it is needless to say that there is a plethora of commercial benchmarks to evaluate database systems. Such benchmarks are extremely important since a lot of business activities in the real world dwell around database systems. Also, such benchmarks are important pointers towards the selection of the right package and the right configuration for a particular operating environment. The most famous are the TPC (Transaction Processing Performance Council) family of benchmarks[15] . Another set of benchmarks evaluating security in database systems are CIS Benchmarks for SQL Server presented by Center for Internet Security [17].

Benchmarking is generally referred to the process of evaluating a system against some reference to determine the relative performance of the system. As already discussed in the paper, though the basic primitive job of the database system remains generic, yet database systems exhibit different flavors and requirements based on the environment in which they operate. Also, database systems contribute hugely to the proper and efficient functioning of organizational and business information needs. Hence, selecting the right database with the right features is often a very critical decision. To aid such decisions, a bunch of bench-marking techniques have been designed, both commercially and in the open source platform. In this section we shall look at some of these benchmarking techniques in more detail.

5.1 TPC Benchmarks

TPC or "Transaction Processing Performance Council" has defined a suite of benchmarks for transaction processing and database performance analysis. These benchmarks do not solely evaluate the database component of the system but rather evaluates the whole system of which the Database system is one of the key differentiating factors. The suite contains a mix of benchmarks for evaluating different performance

aspects of such systems.

5.1.1 TPC-C Benchmark [19]

The TPC-C benchmark is an On-line Transaction Processing (OLTP) Benchmark. TPC-C benchmark contains a mix of different types of concurrent transactions, a complex database, nine types of tables with different record and population sizes. It simulates the process of a multi-user environment making concurrent queries to a central database. The performance evaluation involves a thorough monitoring of all system state parameters for correctness of update as well as performance parameters such as service time etc. This benchmark is most suitable for businesses that need database to support online handling of orders, sell product and manage inventory.

5.1.2 TPC-E Benchmark [20]

The TPC-E benchmark is another OLTP Benchmark but is especially designed for evaluating database systems needed to be installed at brokerage firms. It is quite similar to the TPC-C benchmark in terms of setup and components differing only in the design of the transactions which are more relevant in a brokerage firm environment such as account inquiries, online trading and market research etc.

5.1.3 TPC-H Benchmark [21]

The TPC-H benchmark is fine tuned for decision support systems. The transactions in such environments are characterized by business intelligence intensive complex data mining queries and concurrent data modifications. The performance metric used to evaluate such systems is generally TPC-H composite query per hour.

5.2 Other Benchmarks

TPC is the largest and most popular benchmarking authority in databases and hence was dedicated a separate sub-section. The other benchmarking tools that are available are discussed in this sub-section.

5.2.1 Bristlecone [22]

Bristlecone is a java based database performance testing benchmarking utility. It provides knobs to vary the system parameters for a single scenario across different set of rules or environments. The standard run parameters for synthetic replication include number of users, number of tables, number of rows per table, number of rows returned by queries or size and complexity of queries etc. It is supposed to be lightweight and easily installable tool that can be used to serve as a load generation and performance evaluation tool against any database server.

5.2.2 Benchmark Factory [23]

Benchmark factory is a testing and performance evaluation tools for databases that can be used pre-deployment to gauge the effectiveness of the particular database system in meeting an organizations IT goals and needs. It simulates multi-user transaction environment on a database in a production environment or a synthetic workload in a non-production environment. Benchmark factory is thus meant to be a tool for cost effective pre-validation and pre-verification of a database deployment in a particular operating environment.

5.2.3 CIS Benchmark [17]

The CIS benchmark is a set of security benchmark for the SQL Server 2005 and SQL Server 2000. These benchmarks provide a testing tool to evaluate these database systems against common security vulnerabilities. Generally while installing databases most administrators focus on key operating performance issues such as scalability, load balancing, failovers, availability etc and let security settings to be default factory settings. Default security settings are prone to a number of security hazards and generally not advisable. Also, even if the security settings are changed from default, they may not fulfill the security requirement expected. The CIS benchmark test database system against security configuration errors in a non-intrusive environment that does not in any ways jeopardize the normal functioning of the database.

5.2.4 SPEC Benchmark [16]

SPEC (Standard Performance Evaluation Corporation) has a benchmark to evaluate Web Servers and is called SPECweb2005. It simulates multiple users sending tightly secure (https), moderately secure (http/https) and unsecure (http) requests to a web server. The dynamic content is the web server is implemented in PHP or JSP.

The SPECweb2005 is well representative of actual web-server scenarios since it simulates multi-user dynamic page request scenario over a mix of application layer protocols. Also, it simulates web-browser caching effects which is one of the chief optimization technique applied by all such systems.

Though not exactly a database benchmark, SPECweb2005 deserved a special mention because database systems serve the heart of all web server systems and hence one of the keys to the performance bottleneck of such systems.

5.2.5 PolePosition [24]

PolePosition is an open source database benchmark. It consists of a test-suite that compares the database engine and the object-relational mapping technology. It has a java based framework. Keeping up with the spirit of its name, each database test suite is called a "circuit" exercising a mix of different tests. Each of these circuits are intended to test a given database against a pre-canned set of criteria such as heavy-reads, frequent-updates etc. An administrator wanting to choose a database implementation for his requirements may try and map his requirement to a particular "circuit" and test databases in that particular circuit to get the "PolePosition" of databases relative to that "circuit".

5.2.6 Open Source Development Labs Database Test Suite

The OSDL DBT is also an open source database benchmarking tool. It is one of the most comprehensive open-source database benchmarking test-suite available in open source and it is based on the standards set by TPC. However, they differ in a few areas and hence it is not advisable to compare results from tests in this suite with standard TPC tests.

In this section, we have tried to discuss the most commonly used benchmarks that are actually used in the real-world. However, there are lots of other benchmarking suites available for database systems, but a discussion on all of those are beyond the scope of the current work An interested reader may follow-up his/her reading and may be directed to [26] for further pointers.

---

6. Summary

In this paper, we have tried to highlight the importance of Database systems in this IT enabled world, the

various flavors of database applications and production environments, the different requirements and configurations in these environments and lastly, pointers to be able to evaluate databases in accordance with its area of application. This paper is expected to be a preliminary reading exercise and hopefully an interested reader would have gain enough understanding from here to be able to follow up with a more specific are of interest.

References

[1] Amazon suffers U.S. outage on Friday, CNET News

[2] Wikipedia: Database Management Systems

[3] Rick Long, Mark Harrington, Robert Hain, Geoff Nicholls, IMS Primer, International Technical Support Organization, IBM Corporation, 2000

[4] Codd, E.F. , "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377-387, 1970

[5] Charles W. Bachman, The Programmer as Navigator. ACM Turing Award lecture, Communications of the ACM, Volume 16, Issue 11, 1973, pp. 653-658

[6] Stonebraker, Michael with Moore, Dorothy. Object-Relational DBMSs: The Next Great Wave. Morgan Kaufmann Publishers, 1996.

[7] Yao-S Bing, Alan R. Hevner, A Guide to Performance Evaluation of Database Systems, Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402,1984

[8] Haran Boral, David J DeWitt, A methodology for database system performance evaluation, ACM SIGMOD Record, Volume 14, Issue 2, June 1984

[9] S. H. Son and N. Haghighi, Performance evaluation of multiversion database systems, Sixth International Conference on Data Engineering, Issue 5-9, Pages 129-136, February 1990

[10] Kyoung-Don Kang and Phillip H. Sin and Jisu Oh, A Real-Time Database Testbed and Performance Evaluation, Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Pages 319-326 , 2007

[11] Yuanling Zhu and Kevin Lu, Performance analysis of Web Database systems, LNCS, Volume1873/2000, pp 805-814, 2000

[12] Peter G. Harrison and Catalina M. Llado, Performance Evaluation of a Distributed Enterprise Data Mining System, Vol 1786/2000,pp 117-131, 2000

[13] Jian Pei and Runying Mao and Kan Hu and Hua Zhu,Towards data mining benchmarking: a test bed for performance study of frequent pattern mining, SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp 592,2000

[14] Jia-Lang Seng, Comparing Object-Oriented Database Systems Benchmark Methods, Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, Volume 6, Page 455, 1998

[15] TPC benchmarks

[16] SPEC Benchmarks for Java Client Server applications

[17] The Centre for Internet Security, CIS Benchmarks for SQL Server 2005 and SQL Server 2000 Databases, 2005

[18] TPC benchmarks

[19] TPC-C benchmark

[20] TPC-E Benchmark

[21] TPC-H benchmark

[22] Bristlecone Benchmark, 2008

[23] Qwest Software, Benchmark Factory for Databases

[24] PolePosition

[25] Open Source Development Labs Database Test Suite

[26] Database Benchmarking, http://wiki.oracle.com/page/Database+Benchmarking

---

Last modified on November 24, 2008
This and other papers on latest advances in performance analysis are available on line at
http://www.cse.wustl.edu/~jain/cse567-08/index.html
SHARE Back to Raj Jain's Home Page