MySQL Performance Benchmarks

Measuring MySQL's Scalability and Throughput

A MySQL® Technical White Paper

March 2005

Table of Contents

Executive Summary

To meet the service levels demanded by your users, your database-based application needs to deliver high performance and scalability. In addition, it requires complete data availability, which includes fault tolerance, service uptime, and throughput. In short, performance and service uptime are the two most important criteria to ensure an application operates at expected levels.

This paper discusses the performance of MySQL. It highlights the database elements that can be measured and adjusted during a benchmark. In addition it delivers a set of benchmark results obtained from third parties and internal tests run at the MySQL Test Labs that demonstrate MySQL's ability to meet the most demanding service levels.

The benchmark results show that:

- MySQL demonstrated the best overall performance in eWeek's ecommerce benchmark
- MySQL delivered the best cost per transaction value in the SPECjAppServer2002 benchmark
- MySQL shows near linear scalability in a multi-CPU environment with SysBench benchmarks.

Measuring Performance in a Database

What's the best way to measure the performance of a database? The most common technique for measuring performance is to take a black box approach that measures the Transactions Per Second (TPS) an application is able to execute against a database. In this scenario a "Transaction" is a unit of execution that a client application invokes against a database. This could be a simple read query or a grouping of updates done in a stored procedure. In this context, the term "Transaction" does not necessarily refer to an ACID-compliant transaction, such as those executed by MySQL's InnoDB storage engine, but may involve ACID-compliant transactions depending on how the test is structured.

The value of a benchmark that measures TPS is directly tied to the type of transactions being performed. A test that measures the number of simple read queries that can execute will yield different results than a test that measures the number of multi-table inserts. So, the absolute value of a TPS metric is not as useful as a TPS metric that is compared against many tests that vary a small number of variables. Finally, it's important to note that it's not useful to measure the TPS value of one type of test against a TPS value of a completely different test. For example, the TPS values from SPEC tests are not comparable with those of TPC tests.

Variables that can be altered and affect performance include, but are not limited to:

Variable	**Impact**
Hardware	The speed of the CPU, the CPU architecture, number of CPUs or CPU cores, bus interconnect speed, memory access speed, raw I/O capability, disk hardware and configuration including seek speed, network hardware and interface speed, are all items that a database depends upon and can impact the performance of a database.
Operating System	Variability in the performance of native API file, threading, locking, and memory can impact the performance of a common benchmark run on different operating systems.

Number of Client Connections	The number of clients concurrently attempting to connect to a database can measure its ability to handle increasing amounts of burden.
Level of DB Concurrency	The number of threads inside the database (controlled in MySQL by the total number of connections since there is a direct correlation) can impact the parallelism of the system. Too few parallel threads means that the system will under perform and too many parallel threads may cause the system to spend too much time managing thread overhead instead of executing transactions.
Data Schema	The structure of the database schema used in the test. The type of test, or workload, that is defined would vary between a test that uses a database with a single table versus a test on a database with 1000s of tables with complex foreign key relationships. There is a difference between the “workload” of a test which corresponds to the actions applied to a database versus the “logical” schema (structure of the data). In most pre-defined independent tests, the logical schema is defined along with a set of transactions to execute against that schema to define the workload pattern.
Volume of Data	A test on a database with a large volume of data will be a stronger measure of the effectiveness of the index implementation than a test on a database with a low amount of data.
Type of Application	Read-only, read-mostly, and read-write applications are a good reflection of data warehousing, eCommerce, and financial applications, respectively. Varying tests on the type of application will provide a good measure of how a system handles read operations versus data modification operations. TPS is a good measure used in OLTP workloads and you will typically see “length of time” measurements on workloads that represent a read-mostly / batch bulk update pattern (even though TPS and length of time measurements can be calculated form one another). Additionally, the type of application chosen will have a direct impact on the use (or not) of ACID-transactions in a test, which can have a huge impact on TPS.
Data Access Pattern	Since most applications access less than 5% of the data in a database more than 90% of the time, a test that follows this standard distribution will measure the effectiveness of a system’s caching abilities. A uniform test that accesses different data equally is a better measure of the database’s raw performance when most operations are executed against a raw disk partition. Also the data access pattern is an important aspect in how the workload will be defined (focus on transactional throughput or speed of a batch update?).
Database Configuration	MySQL allows a DBA or developer to modify parameters including the maximum number of client connections, the size of the query cache, the execution style of different logs, index memory cache size, the network protocol used for client-server communications, and dozens of others.

With nearly an infinite number of variables that can alter the results of a benchmark, what is the best technique to use in measuring performance? We recommend a series of tests, following a scientific method that alters a small number of variables on each test. These benchmarks will demonstrate value in two areas:

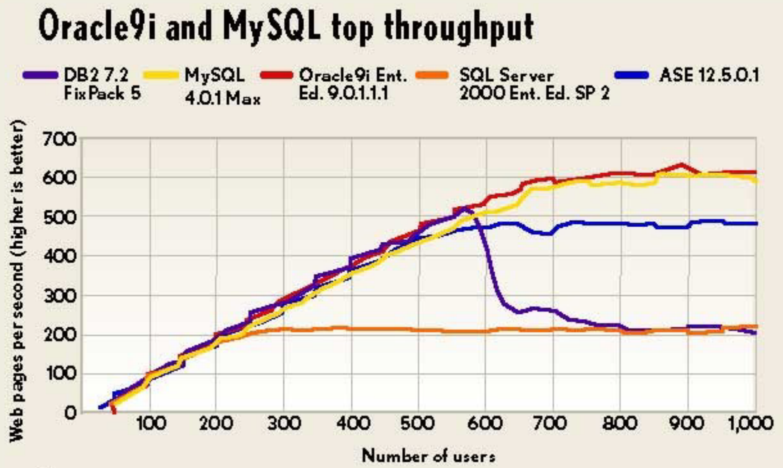
- The absolute TPS number achieved in various tests will provide a baseline metric that can be used for planning, estimation, and a "likely" result that will be seen in a production environment using a similar configuration.
- The changes in TPS values between tests will document the type of impact changing a variable has on the overall performance.

A Sampling of Benchmark Results

We have included some benchmark results that have been run against MySQL. This section includes tests run by third-parties and some internal tests run by MySQL in our Test Labs. These results provide a relative comparison of how MySQL performs against other databases and also how MySQL scales with additional CPUs.

eWeek Database Benchmark

The eWeek Database Benchmark is an eCommerce application that has a Web interface that combines read and write queries. This test measures the performance of an overall system in terms of Web Pages per Second, which is equivalent to TPS since each Web page invokes a single transaction. Their findings show that MySQL has the best overall performance and that MySQL scalability matches Oracle. Additionally, their tests showed that MySQL excelled in stability, ease of tuning, and connectivity options. Finally, they also found that MySQL had the highest throughput, even exceeding the numbers generated by Oracle.



Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2GB of RAM, a Gigabit Ethernet Intel Corp. Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.

From the eWeek Database Benchmark Test

"Of the five databases we tested, only Oracle9i and MySQL were able to run our Nile application as originally written for 8 hours without problems."

"The Oracle and MySQL drivers had the best combination of a complete JDBC feature set and stability."

"SQL Server and MySQL were the easiest to tune, and Oracle9i was the most difficult because it has so many separate memory caches that can be adjusted."

Timothy Dyck, eWeek
"Clash of the Titans: SQL Databases"

The following table shows the actual results of the test measuring pages per second at different thresholds of users.

Users	MySQL	Oracle	SQL Server	DB2	Sybase ASE
100	94.735	100.700	99.724	102.382	95.753
200	186.594	199.653	181.406	203.859	191.071
300	270.824	293.318	208.900	302.783	280.535
400	361.812	384.671	205.335	398.688	370.335
500	443.559	476.241	206.676	484.065	448.888
600	523.524	544.665	204.429	322.888	475.829
700	578.082	594.806	207.359	237.747	471.294
800	599.612	615.624	200.518	215.476	471.118
900	601.788	631.388	203.829	207.247	478.718
1000	590.488	624.386	203.870	204.682	475.934

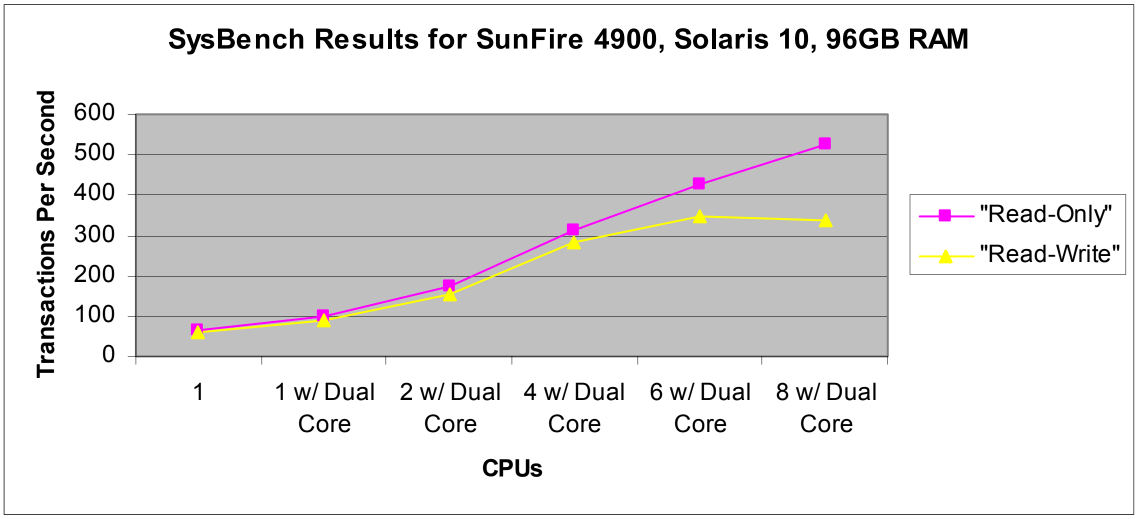
SysBench Scalability Benchmark (Sun Solaris)

In March of 2005, the MySQL Test Labs conducted tests with MySQL running on a SunFire 4900 with 96GB of RAM using Solaris 10. The SunFire 4900 system has a dual core system that can be enabled or disabled when there is a single CPU. We did our tests with the dual core enabled and disabled. This test was done using SysBench 0.3.1, an open source MySQL benchmarking tool that is hosted at Sourceforge. The SysBench test is designed to simulate a generic OLTP workload, which results in a low level database access pattern similar to an eCommerce application such as a bookstore.

The MySQL Test Labs ran the tool with this configuration:

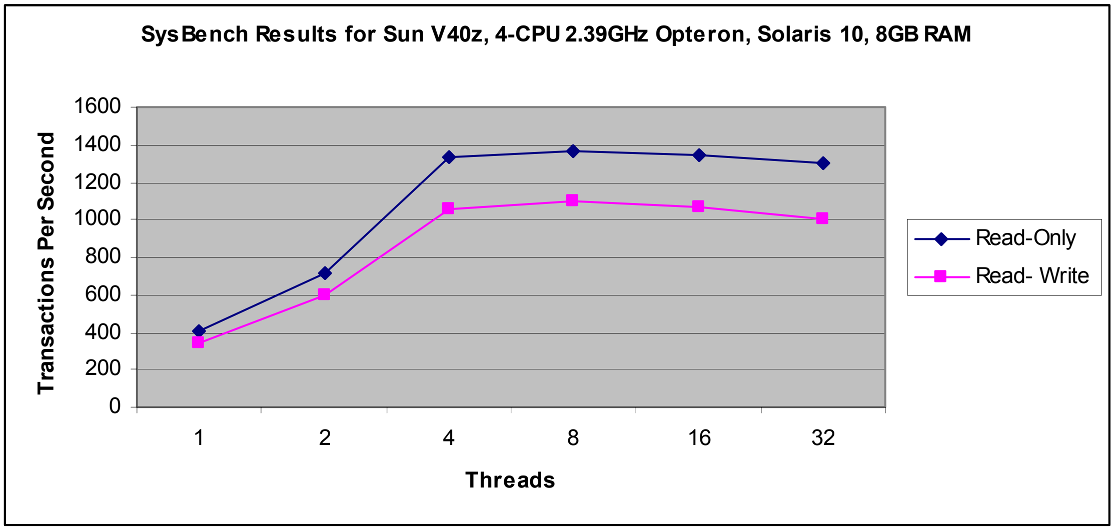
- MySQL 4.1.9 using InnoDB tables with ACID-compliant transactions
- 10 million records (~2.5 GB of data)
- Standard access distribution (the bulk of read queries are targeted at 1% of the data) used as opposed to a uniform distribution (all data is accessed evenly)
- The read-only run was done with 32 threads (concurrent connections) and the read-write run was done with 16 threads (concurrent connections). There is tapering performance in the read-write scenario when the system has more threads than CPUs. This is expected due to various concurrency issues, thread scheduling overhead, and cache wiping that consumes extra CPU cycles. The read-only scenario scales better and reaches its maximum at a higher number of threads because there are less lock concurrency issues.

MySQL generated more than 500 transactions per second (TPS) in an 8 CPU environment for a read-only test. These results show a near-linear increase in performance as CPUs were added to the system. Here are the results that were generated.



SysBench Scalability Benchmark (Sun V40z)

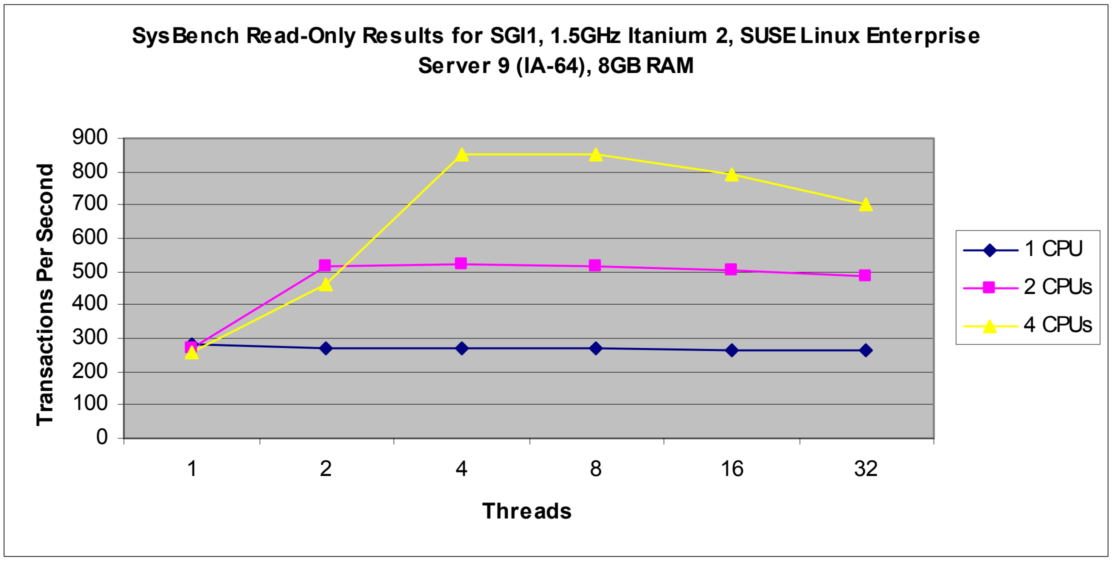
Additionally, the MySQL Test Labs performed the same SysBench test on a Sun 40z 4 2.39GHz CPU machine running Solaris 10 with 8GB RAM. This system showed great absolute performance with the read-only test achieving over 1350 TPS and the read-write test nearing 1100 TPS when there are 8 threads.



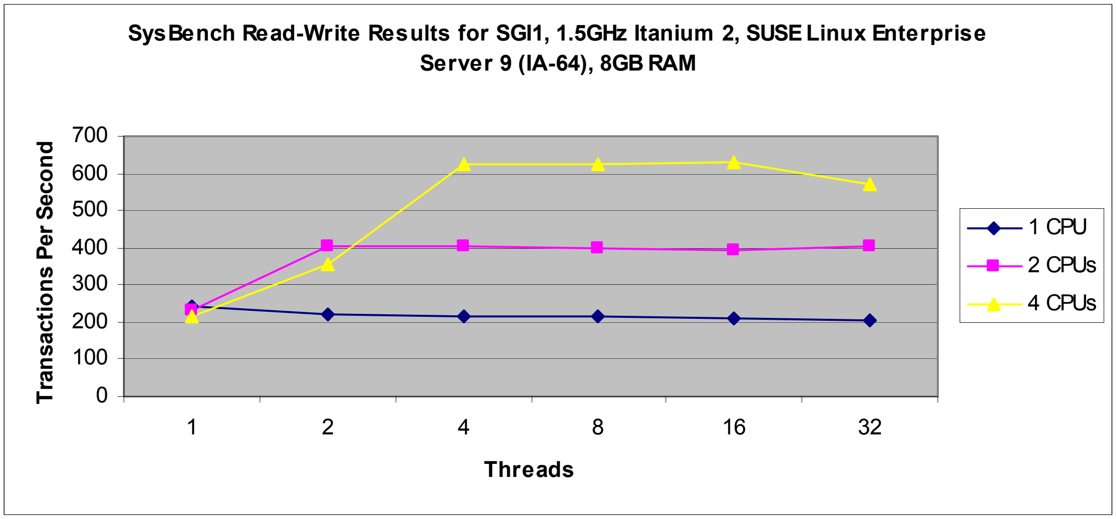
SysBench Scalability Benchmark (SGI Linux Itanium 2)

As a follow on benchmark, the MySQL Test Labs performed the same set of tests with variability in the number of threads to get a measure of how an SGI Linux system performed and how increasing concurrency impacts performance. The machine used was an SGI1 with 1.5GHz Itanium 2 CPUs run on SUSE Linux Enterprise Server 9 (IA-64) with 8GB RAM. The SysBench configuration was exactly the same as the one used on the SunFire 4900 benchmark.

The first set of tests was a read-only test done using a standard distribution. The number of transactions per second increases based upon the number of CPUs that were available. The benchmark tops out at **851 TPS** when using 4 CPUs with 4 threads! This information can help your organization achieve higher levels of throughput at a lower cost with a scale out strategy.



The second set of tests was a read-write test done using a standard distribution. The number of transactions per second increases based upon the number of CPUs that were available, but the absolute number of transactions executed per second is not as high as those in the read-only scenario. This happens because read-only transactions are read-write transactions with 4 write operations disabled. So a read-only transaction will have approximately 16 total operations instead of 20 operations for a read-write transaction. Read-write load does not scale as much as read-only due to its read-write nature but also the absolute number is different because read-write transactions are different types of transactions. This pattern can change based upon the ratio of read queries to write queries. If you implement a similar type of benchmark tailored to your application, it's critical that you resemble this ratio as accurately as possible so that the data you generate can be confidently used in planning and estimation activities.



SPECjAppServer2002 Benchmarks

At the end of 2004, MySQL teamed with BEA and Sun to post new results for the SPECjAppServer2002 benchmark. A MySQL database running with BEA's WebLogic Server on Sun Microsystems' Sun Fire hardware set world records for speed and price/performance as published by the Standard Performance Evaluation Corp. (SPEC). SPECjAppServer2002 is an industry benchmark, administered by SPEC used to measure the performance of J2EE application server platforms connecting with a database. It is designed to model a real-world manufacturing, supply-chain management and order/inventory application.

The competitive benchmark results listed below were performed on standard x86 processors including AMD's Opteron, Intel's Xeon and Itanium. They reflect results posted as of September 2, 2004. SPECjAppServer2002 measures based upon TOPS (Total Operations per Second) and also on US$/TOPS, which is the total cost of the entire system (hardware + licensing + support fees) divided by TOPS.

Price/Performance value (Lower numbers demonstrate better price/performance)				
Database	**App Server**	**Database Hardware**	**App Server Hardware**	**US$/TOPS**
MySQL	BEA WebLogic	Sun Fire V40z	Sun Fire V20z	$82.74
SQL Server	BEA WebLogic	Unisys ES7000 Aries 410	Unisys ES3020L Server	$139.84
Oracle	Oracle	HP ProLiant DL580 G2	HP ProLiant ML370 G3	$150.68
SQL Server	BEA WebLogic	Unisys ES7000 Aries 420	Unisys ES3020L Server	$157.66
Oracle	Oracle	HP ProLiant DL580 G2	HP ProLiant ML370 G3	$168.62
SQL Server	BEA WebLogic	HP ProLiant DL580 G2	HP ProLiant DL360 G3	$200.34
IBM DB2	IBM Websphere	IBM eServer pSeries p650	IBM eServer xSeries x335	$330.07

Performance of entries with databases running on 4-processor hardware
(Higher numbers demonstrate faster performance)

Database	App Server	Database Hardware	App Server Hardware	TOPS
MySQL	BEA WebLogic	Sun Fire V40z	Sun Fire V20z	1363.88
Oracle	Oracle	HP ProLiant DL580 G2	HP ProLiant ML370 G3	1165.06
SQL Server	BEA WebLogic	HP ProLiant DL580 G2	HP ProLiant DL360 G3	1037.02
Oracle	Oracle	HP ProLiant DL580 G2	HP ProLiant ML370 G3	863.40

Conclusion

Performance is one of the three design tenets of MySQL (Performance, Reliability, and Ease of Use). Our unique multiple-storage engine architecture has enabled us to stay true to that objective. To meet the service levels demanded by your users, it’s critical that your application has the availability and performance expected regardless of the type of application or the work load that is placed upon the system.

As demonstrated in this paper, MySQL has repeatedly achieved top performance and scalability in a wide range of industry recognized benchmarks by third parties and in the MySQL Test Labs making it the right choice for high-volume business-critical transaction systems.

About MySQL

MySQL AB develops, markets and supports a family of high performance, affordable database servers and tools. The company's flagship product is MySQL, the world's most popular open source database, with more than six million active installations. Many of the world's largest organizations, including Google, Sabre Holdings, The Associated Press, Suzuki and NASA, are realizing significant cost savings by using MySQL to power Web sites, business-critical enterprise applications and packaged software. MySQL AB is a second generation open source company, and supports both open source values and corporate customer needs in a profitable, sustainable business. For more information about MySQL, please go to www.mysql.com.

Appendix: Benchmarking Tools

There are a number of packages and utilities available to let you run a benchmark against a MySQL database. Short of writing a custom application to test the performance of your system, these packages can provide out-of-the-box sample data, test scripts, and configurable parameters to create a custom test environment that approximates your production environment. Please refer to the documentation contained with each of these benchmarks on their Web site for more information.

The eWeek Nile eCommerce Benchmark

eWeek's Nile benchmark test was designed to compare the results of several relational database management systems in a "real world" ecommerce application. eWeek tested several databases including MySQL, Microsoft SQL Server, IBM DB2, Oracle and Sybase. Test results show that MySQL is able to meet the highest levels of transaction performance and scalability.

The Nile benchmarks have been updated to version 1.1 and are available with full source code, data and instructions. More information can be found at www.eweek.com/article2/0,4149,293,00.asp

SysBench

SysBench is a modular, cross-platform and multi-threaded benchmark for evaluating operating system parameters that are important for a system running a database under intensive load. SysBench is a flexible testing utility that allows a variety of different test modes to be plugged-in for use with the utility. SysBench was originally designed to test parameters such as file I/O performance, scheduler performance, memory allocation and transfer speed, and POSIX thread implementation performance.

SysBench allows a tester to configure the number of threads, the amount of data in a database, the access pattern, and whether the database is read-only, read-mostly, or read-write.

MySQL performance tests with Sysbench demonstrate our ability to scale with CPUs and threads on a variety of platforms. More on SysBench can be found here: http://sysbench.sourceforge.net/

SPEC and TPC

The Standard Performance Evaluation Corporation (SPEC) and the Transaction Processing Council (TPC) provide proprietary transaction processing and database benchmarks that are used to test Web servers, application servers, databases, and high performance computers in a variety of configurations and scenarios. Both SPEC and TPC provide a suite of different tests designed to benchmark individual usage scenarios. These tests are comprehensive and complicated, often requiring numerous resources to execute successfully. SPEC and TPC results are typically performed by software and hardware vendors under a set of strict set of rules and processes administered by SPEC and TPC. These benchmarks can provide a relative comparison of an application used in different environments from different vendors. These tests have a fee associated with them and are not usually applied to individual applications.

MySQL has been benchmarked with the SPECjAppServer2002 benchmark in a partnership with BEA where we demonstrated the best cost value per transaction and, in many scenarios, the best transaction throughput. MySQL has not tested with TPC benchmark because the tests are often skewed by how much high-performance hardware and disk arrays are used. So while TPC is a good measure of overall system performance, it is not a real world benchmark and does not take into account the total system costs.

More information on the SpecJ test can be found here: www.spec.org/jAppServer2002/

More information on the Transaction Processing Council can be found here: www.tpc.org/.

Other Benchmark Tests

If you're looking for other database benchmark tests, the resources may be helpful.

- **MySQL Benchmark Suite.** Designed to identify which operations embedded in the database are performing well or poorly: http://dev.mysql.com/doc/mysql/en/mysql-benchmarks.html

- **The Open Source Database Benchmark (OSDB).** A flexible database benchmarking utility for Unix platforms developed by a consortium of developers. http://osdb.sourceforge.net/

- **BenchW.** A tool to compare the capabilities of different database managers for data warehouse activities. http://benchw.sourceforge.net/

- **Open Source Development Labs' (ODSL) Database Test Suite (DBT).** Designed to help Linux developers benchmark the performance of an applications that relates to real-world usage. http://www.osdl.org/lab_activities/kernel_testing/osdl_database_test_suite/