EnterpriseDB Engineering Study

Performance Benchmarking

An EnterpriseDB White Paper

for DBAs, Application Developers, Enterprise Architects

February 2008

http://www.enterprisedb.com

EnterpriseDB™

Executive Summary

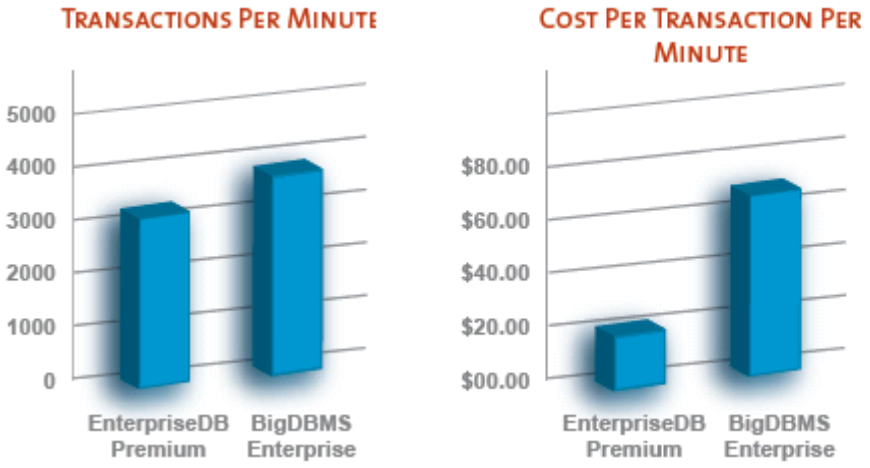
A key requirement of any enterprise-caliber database is high performance based on comparative benchmarks. EnterpriseDB has been an active member in, and sponsor of, the PostgreSQL community since the company was founded in March 2004 and has worked closely with the community to improve the performance of the PostgreSQL database.

To that end, EnterpriseDB has established a Performance Center dedicated to improving the performance of both PostgreSQL and Postgres Plus Advanced Server. As a result of these efforts and the efforts of the PostgreSQL community, native PostgreSQL is now considered twice as fast as previous versions.

In addition, EnterpriseDB continues to add advanced performance features to Postgres Plus Advanced Server to further enhance the product's performance. As part of this process, EnterpriseDB engineers routinely measure the performance and consequent price/performance of the database against other competing technologies in the market.

The results of the latest EnterpriseDB Engineering Study on performance confirm the price/performance leadership of Postgres Plus Advanced Server against a very large database rival and validate it as a viable alternative to the three major OLTP database vendors. The two key findings of this study are:

- Postgres Plus Advanced Server performs at 90% of BigDBMS.
- BigDBMS's price/performance results are more than 300% more expensive than Postgres Plus Advanced Server.



> Who is BigDBMS? License agreements for the traditional database vendors do not allow for comparative benchmark testing. BigDBMS does not refer to any particular database company, but represents a realistic approximation of the larger players, grouped into a single entity.

EnterpriseDB engineers utilize BenchmarkSQL for all performance testing in the lab because of its ability to run across multiple operating system/hardware platforms and multiple database technologies. In addition, BenchmarkSQL can be run with varying test lengths to simulate real-world loads.

This test is built on the open source JTPCC project combined with the ability to handle prepared SQL statements for testing consistency across all databases. Links for both tests can be found on page 5 in the section titled “Benchmarking Application and Methodology”.

An in depth discussion targeted specifically to your organization’s requirements can be scheduled with an EnterpriseDB domain expert by sending an email to sales@enterprisedb.com.

Benchmarking Application and Methodology

Benchmarking database performance is a difficult endeavor because of the number of tuning characteristics associated with each database. In addition, the types of performance will vary greatly depending on the type of application, the architecture of the database and an almost infinite number of other parameters.

In an effort to create an objective comparison in performance benchmarking, several companies formed a benchmarking association which produced scenario specifications to test different types of database application use cases.

Completing a formal public benchmark takes significant time and resources and includes a test driver application written from the ground-up specifically for the hardware and database software being tested. A cross-database benchmark requires something different - a test driver application that is database vendor and hardware platform neutral.

The benchmarking suite used by EnterpriseDB engineers in this test is modeled on an industry benchmarking association scenario. However, it is

not a formal implementation nor should this study be construed to be a formal public report of results.

For this test EnterpriseDB selected the open source database benchmark driver application JTPCC and enhanced it to push prepared SQL statements to all of the databases tested. This enhanced version of JTPCC has been released as an open source project called BenchmarkSQL.

BenchmarkSQL allows tests to be executed against many different databases. It is a Java application, which is operating system and hardware-neutral, and uses database-neutral drivers – in this case JDBC – to communicate with the database. This effectively eliminates outside performance-influencing factors such as proprietary interfaces so that the end-result comparison is more on the core SQL processing and transaction handling capabilities of the tested databases.

The open source BenchmarkSQL project is available at http://sourceforge.net/projects/benchmarksql/.

The JTPCC benchmark, upon which BenchmarkSQL is based, is also available on SourceForge at http://sourceforge.net/projects/jtpcc.

The BenchmarkSQL OLTP (Online Transaction Processing) scenario models a wholesale supplier managing orders. The test is designed to impose a transaction load on a database and to then count how many new orders can be placed and completed under this load.

In addition to transaction processing, the benchmarking suite strings operations together into large transactions. A transaction history is maintained during the execution of the test and this history is compared with actual results to ensure that transactional and referential integrity is maintained throughout the term of the test. Non-transactional database engines will fail this test outright.

Benchmark Configuration

Setting the testing parameters

In this comparison study all database schemas were set-up identically. The test was driven by BenchmarkSQL with a setting of 100 Warehouses. BenchmarkSQL was hosted on a Dell™ Dimension E521 computer with 1 x AMD Athlon™ 64 3200+ processor, 1 GB of RAM, and 80GBs of storage.

The database was hosted on a Dell™ PowerEdge 6850 computer with 4 x 2.6GHz Dual Core Intel Xeon processors, 4 GB of RAM, and a PowerVault MD1000 Storage unit providing a total of 576 GBs of storage.

This test models a set of five transactions that are being driven by a group of simulated operators. The transactions modeled are:

- New-Order
- Payment
- Order Status
- Delivery
- Stock Level

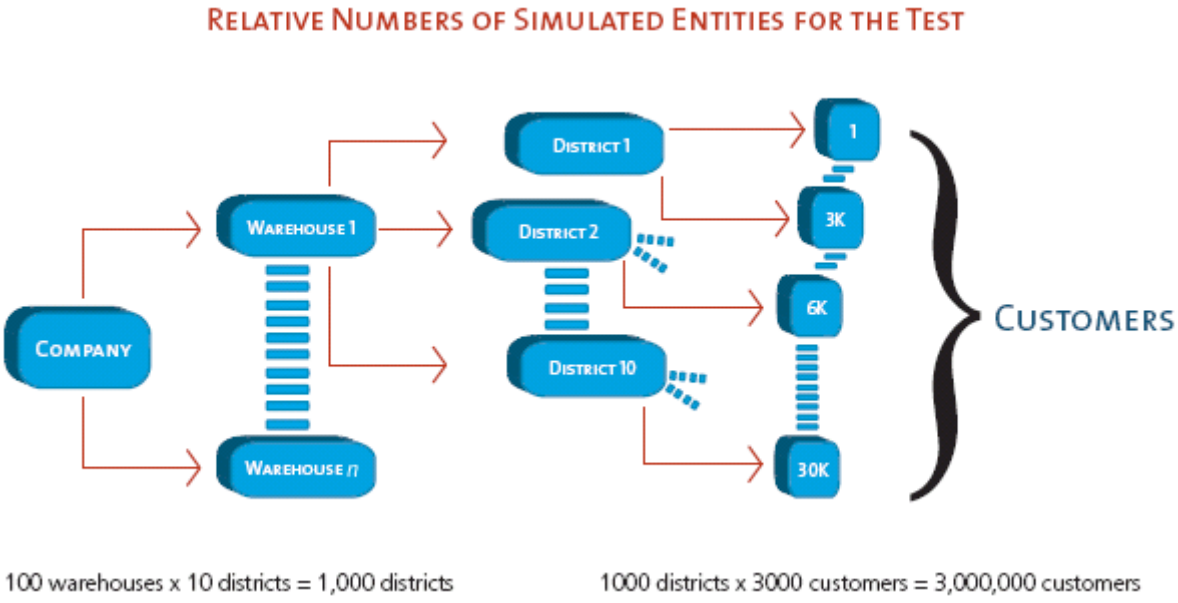
The data set exercised by these operators emulates the structural data requirements of a real business. In this example, a fictitious company has multiple warehouses and each of these warehouses has 10 districts.

Each district has its own sequential system for numbering order transactions. Each district has its own operator who creates new orders, books payments, checks status of existing orders, issues delivery tickets, and checks stock level. Each district has 3000 customers.

Each warehouse has inventory from a list of 100,000 parts so a stock-level of up to 100,000 parts must be maintained for each warehouse. The amount of information grows very quickly with the addition of every warehouse.

For example, in a test run on 100 warehouses there will be 1000 districts, each with an operator (meaning there will be 1000 terminal connections pushing transactions), and with 3000 customers for a total of 3,000,000 customers to track along with the status of any orders generated for each customer.

As a result, a benchmark simulating 10,000 warehouses requires significantly more underlying hardware than a test simulating 100 warehouses.



The test is designed to measure not just the raw throughput of a database but the throughput of New-Order transactions while under a heavy load from the other four transactions listed above. These transactions not only place load but also exercise the ability of the database to effectively and efficiently maintain the integrity of information as it is being accessed and changed from multiple points.

The database is responsible for processing concurrent transactions on the same information and giving results that are accurate for the specific point in time in which they are relevant. Checking status of an order, for example, tests the multi-version concurrency control of a database – the value that is returned from a query on an order's status should reflect the state at the exact time of the request. This is true even if milliseconds after the query was issued an update is performed that would change the state of that order.

A minimum ratio of the other four transaction types is maintained to ensure a healthy load is placed on the database at the same time New-Order transactions are being processed. This ratio is based on a goal of at least one Payment transaction for every New-Order and at least one Order-Status, Delivery and Stock-Level transaction for every ten New-Orders. This order is maintained by the testing application.

Ratio of Transactions for the Test	
Transaction Type	**% of mix**
New-Order	Up to 45.0
Payment	43.0 Minimum
Order-Status	4.0 Minimum
Delivery	4.0 Minimum
Stock-Level	4.0 Minimum

Ramp-up is an amount of time where the test is running just to allow the database to reach a steady transaction rate. The test Ramp-up period was 10 minutes. During this time the caches are filled and prioritized. The database is given time to adjust to the load produced by the test. All of the tested databases reached their stable state during the 10 minute Ramp-up time.

After the ramp-up time the measurement interval was 60 minutes. This measurement interval is the time during which transactions per minute are tracked.

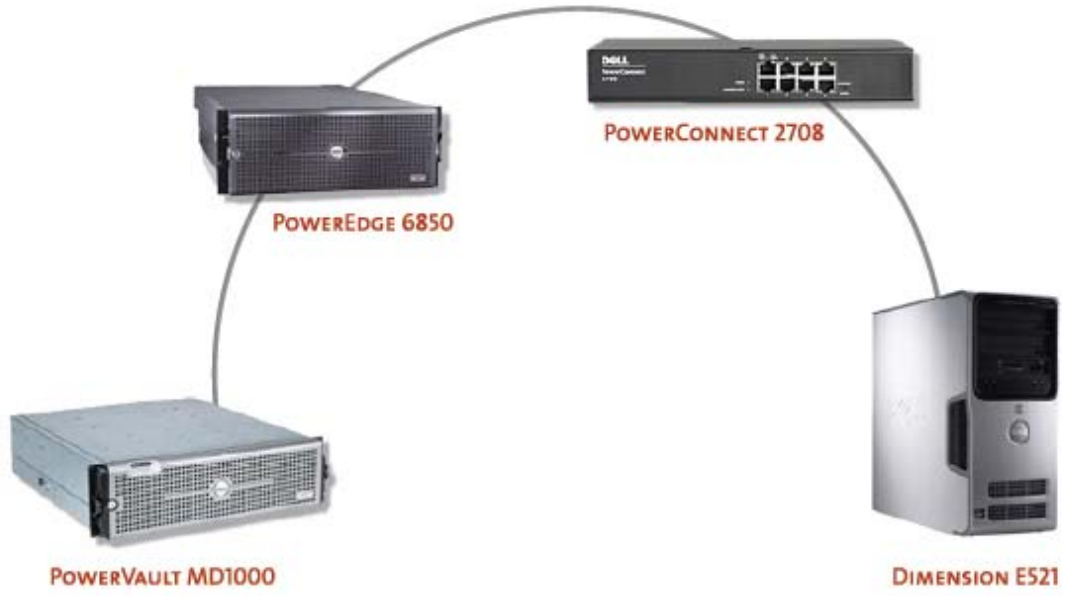
The test driver application, BenchmarkSQL, was configured to push as many transactions as possible at each of the tested databases. Further, in this scenario it was configured to skip the wait times of a standard benchmarking association-style test in order to create the most intense load and update contention possible. The effective rate of New-Order transactions is roughly equivalent to the maximum throughput of 300 warehouses and 3000 terminals.

BenchmarkSQL Parameters for the Test	
Parameter	**Value**
Warehouses	100
Entry Terminals	100
Districts [1]	1,000
Customers [2]	3,000,000
Test Ramp-up	10 minutes
Test Length	60 minutes

[1] Ratio of Warehouses to Districts is fixed at 1:10
[2] Ratio of Districts to Customers is fixed at 1:3000

EnterpriseDB Advanced Server 8.2 Premium		Benchmark SQL 5.2 - Report Date April 16, 2007		
Database Vendor	**Database Edition incl. 24X7 Support**	**Total System Cost including DB License**	**Transactions per Minute**	**Price/ Performance**
EnterpriseDB	Premium	$82,468.00	3,634.0	$22.69
BigDBMS	Enterprise	$288,068.00	4,019.0	$71.68

Processors	Operating System
4 x 2.66 GHz Dual Core Intel® Xeon® Server	Novell's SUSE Linux Enterprise Server 10
1 x AMD Athlon™ 64 3200+Client	Red Hat Enterprise Linux 4 Workstation



	Server		Client	
System Components	**QTY**	**Description**	**QTY**	**Description**
Processor	2	2.66 GHz Dual Core Intel® Xeon®	1	AMD Athlon™ 64 3200+
Memory	4	1GB	1	1GB
Disk Controllers	1	PERC 5/1 x4 Backplane	1	On-board SATA
Disk Drives	1	36GB SAS 15K	1	80GB SATA 7200RPM
	15	36GB SAS 15K		
Total Storage	2	576GB		80GB
Total Hardware System Price				**$22,468.00**

Software Configuration for All Tested Databases

The products listed below are compliant with each database vendor's licensing model for the hardware on which they all execute i.e. 8 CPU cores (4 sockets) and include the cost of 3 years of Maintenance and 24x7 Support. The prices reflect the advertised prices from each of the tested vendors.

Postgres Plus Advanced Server is a subscription model that is licensed by the socket. BigDBMS Enterprise follows a traditional license and maintenance model.

	EnterpriseDB Advanced Server	BigDBMS Enterprise
Processors	4	4
Initial License	N/A	$40,000/processor
Maintenance	N/A	$8,800/processor
Subscription	$5,000/proc	N/A
3-Year Cost	$60,000.00	$265,600

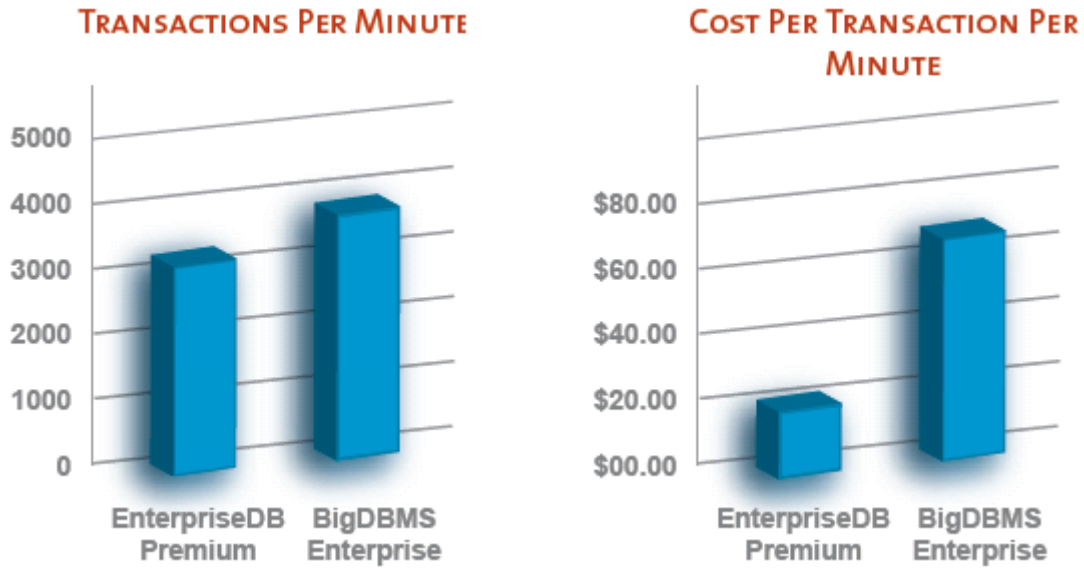
Final Benchmarking Results

90% of the speed of BigDBMS - 1/3rd the cost of BigDBMS

For this test, database-specific DBAs tuned each of the databases to their optimal level for the benchmark. Postgres Plus Advanced Server however, required very little performance tuning because the database was near-optimally tuned out-of-the-box by DynaTune™.

As illustrated in the left graph below, when the databases are fully tuned EnterpriseDB handles approximately 90% of the transactions per minute (TPM) as the leader BigDBMS.

Additionally, as illustrated in the right graph, Postgres Plus Advanced Server significantly beats BigDBMS in Dollars per Transaction (price/performance) requiring less than 1/3rd of the cost of BigDBMS.



Measuring Price/performance. Price/performance is measured by taking the total cost of the system (hardware and software and three years maintenance) and dividing it by the average transactions per minute measured over the run of the test. This result is the $/TPM or price/performance of the database environment and demonstrates how performance can impact the cost of doing business.

The results are dramatic. Postgres Plus Advanced Server and BigDBMS are almost equal in performance, with BigDBMS coming in at around 9% faster in this test. However, given the difference in the acquisition cost, Postgres Plus Advanced Server is $48.99 per transaction less expensive. That is more than a 300% difference.

Conclusions

Enterprise class databases must prove their mettle, in comparative performance benchmark testing.

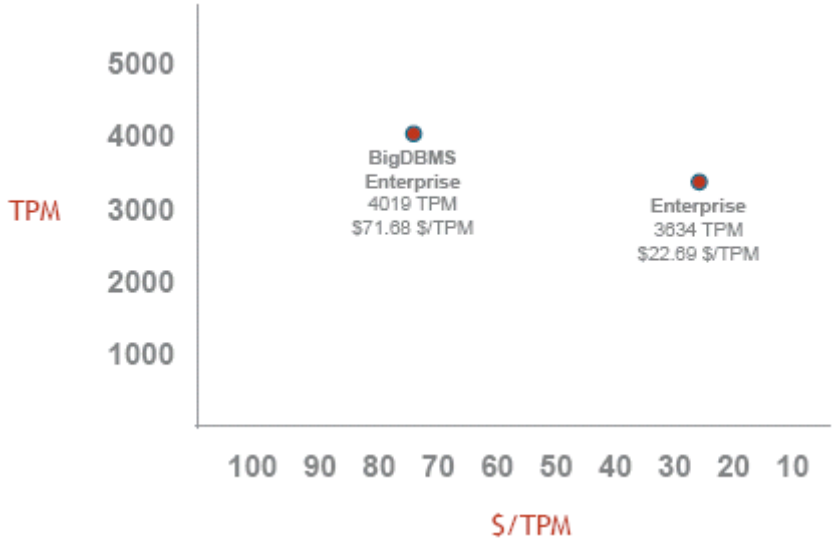
One measure of EnterpriseDB's seriousness as the open source leader for enterprise class databases is its dedicated Postgres Performance Center.

EnterpriseDB has contributed to efforts that have doubled the performance of PostgreSQL and continues to augment Postgres Plus Advanced Server with even more performance enhancements.

The mature OLTP database market is poised for disruption and Postgres Plus Advanced Server is a viable alternative.

This study shows that Postgres Plus Advanced Server delivers enterprise-class OLTP features and reliability combined with world-class performance and price/performance:

- Postgres Plus Advanced Server and BigDBMS have almost equal performance,
- BigDBMS price/performance is greater than 300% more expensive than Postgres Plus Advanced Server.



Additionally, Total Cost of Ownership is further reduced with advanced features such as DynaTune™. BigDBMS requires significant time and resources to tune the database to perform properly adding additional overhead and costs to an already expensive environment.

NOTE: The findings of this paper are based on EnterpriseDB studies and every effort has been made to optimize the performance of the databases tested in this study. However, these results are for illustrative purposes only and different tuning parameters may result in different findings.

For more information regarding Postgres Plus in your environment, please contact us at: https://www.enterprisedb.com/about/contact_us.do or contact the Sales department at: sales-us@enterprisedb.com (US), sales-intl@enterprisedb.com (Intl), or call +1-732-331-1315, 1-877-377-4352.

About EnterpriseDB

EnterpriseDB is the leading provider of enterprise class products and services based on PostgreSQL, the world's most advanced open source database. The company's Postgres Plus products are ideally suited for transaction-intensive applications requiring superior performance, massive scalability, and compatibility with proprietary database products. Postgres Plus also provides an economical open source alternative or complement to proprietary databases without sacrificing features or quality. EnterpriseDB has offices in North America, Europe, and Asia. The company was founded in 2004 and is headquartered in Edison, N.J. For more information, please call +1-732-331-1300 or visit http://www.enterprisedb.com .