Pervasive.SQL Client/Server Performance
Windows NT and NetWare

Debit/Credit Transaction Benchmark – TPC-B transaction profile
Debit/Credit Transaction Benchmark – TPC-B transactions with think times
Database primitives test – Update, Select and Insert performance

Pervasive Software System Architecture
2/18/98

Executive Summary

Pervasive.SQL on both Windows NT and NetWare consistently outperforms similar configurations of Btrieve 6.15 by margins of from 50% to 10 times. Pervasive.SQL also significantly outperforms both Sybase SqlAnywhere and Microsoft SQL Server on tests of important database primitives such as selects updates and inserts. The performance results presented in this report were generated primarily from a debit / credit style benchmark with a TPC-B like transaction profile.

Overview of Performance Comparisons

This performance comparison consists of three suites of tests:

1. Debit/Credit transaction benchmark with TPC-B like transaction profile.

This benchmark simulates a distributed banking environment with 100,000 Accounts 10,000 branches and 10,000 tellers. The TPC-B transaction profile has become a defacto standard for determining database sustainable transaction throughput. The results are expressed as transactions per second. As defined by the TPC-B spec, this benchmark has multiple clients which make continuous transaction requests. In other words there is no "think-time" between the requests. As a result there is not really a concept of users but instead of drivers. The results of this test are itemized for Windows NT in graph #1 and for NetWare in graph #2. In all cases results are presented for configurations ranging from one to 60 drivers.

2. Debit/Credit benchmark with TPC-B like transaction profiles and 2 Sec. Think times.

While a TPC-B style debit/credit benchmark provides a good indicator of peak sustainable database throughput, it does not provide information on how performance evolves in the more realistic case of hundreds of users submitting requests with several seconds of "think-time" between the requests. Our benchmark in this case inserts a 2 Second think-time between each request. The results of this test are itemized for Windows NT in graph #3. As with test #1, the results are expressed as transactions per second. In all cases results are presented for configurations ranging from 5 to 220 users.

3. Database Primitives

While TPC-B like debit/credit transaction benchmarks give a good picture of overall database system performance, they provide little information on the individual database operations upon

which system database performance is built. This test suite measures fundamental database primitives such as selects updates and inserts in a variety of conditions. The tests presented are as follows:

Cached selects	A single record is repeatedly selected. This gives insights into engine speed in delivering records out of the database cache.
Random selects	Selects are performed on a 150MB database. This gives insights into engine performance when physical I/O is required in order to satisfy the request.
Cached updates	A single record is repeatedly updated. This gives insights into engine performance when writing records into the database cache.
Random selects	Updates are performed on a 150MB database. This gives insights into engine performance when writes are forced through the database cache and physical I/Os are required.
Inserts	10 MB of TPC-B style database records are inserted.

A few words about TPC style benchmarks

The Transaction Processing Performance Council is a benchmark standards organization made up of 35 prominent computer vendors. It was formed with the intent to standardize a set of database benchmarks. The first result of this effort was TPC-A, a debit credit benchmark addressing large WAN and LAN based client/server environments. The focus of TPC-A on large environments results in typical system test environments requiring thousands of clients. While this is a useful test for large environments, there is certainly a strong argument that the Small to Medium Business computing environment requires a benchmark which address the 10 to 300 user installation.

The TPC-B benchmark was designed to quickly determine the peak sustainable database throughput achievable by a given configuration. As such, a TPC-B benchmark is useful for determining a given configuration's capabilities without the necessity of configuring thousands of clients.

The TPC-C benchmark simulates a distributed warehouse management infrastructure with point of sale and inventory management. While this could be a useful environment to test, the implementation of TPC-C is such that systems costing hundreds of thousands to millions of dollars are the more typical test beds.

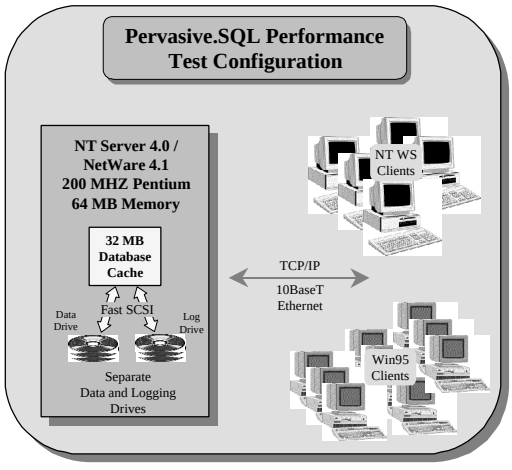

Figure 1

Pervasive.SQL Performance Results

Debit/Credit benchmark with TPC-B like transaction profile

The configuration tested attempts to measure "pure engine performance". The database occupied approximately 15 MB while the database cache was configured to 32 MB. As a result the throughput reflects as clearly as possible pure database engine performance unperturbed by data read I/O loading. Of course, as larger databases are addressed, increasing amounts of data read I/Os will increasingly mask these performance differences. In addition, data writes are still periodically flushed through to disk and in the case of durable transactions, logging I/Os occur to the logging drive. An important point to note, as indicated in figure 1 above, is that data I/Os and logging I/Os were targeted to separate disk drives.

The following diagram shows Pervasive.SQL performance Vs Btrieve 6.15 performance on NT Server. For durable configurations data and logs were placed on separate disks.

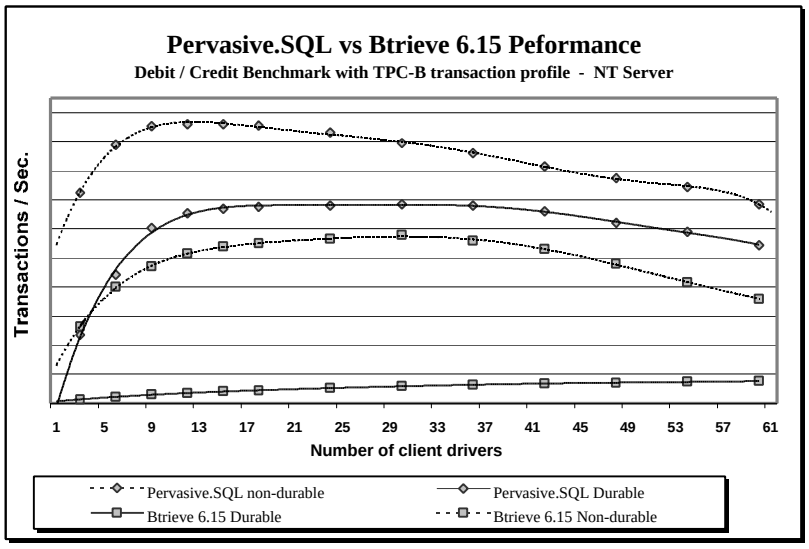

Figure 2

The following diagram shows Pervasive.SQL performance vs. Btrieve 6.15 performance running the Debit/Credit benchmark on NetWare 4.1. For durable configurations, logs and data were placed on separate disks.

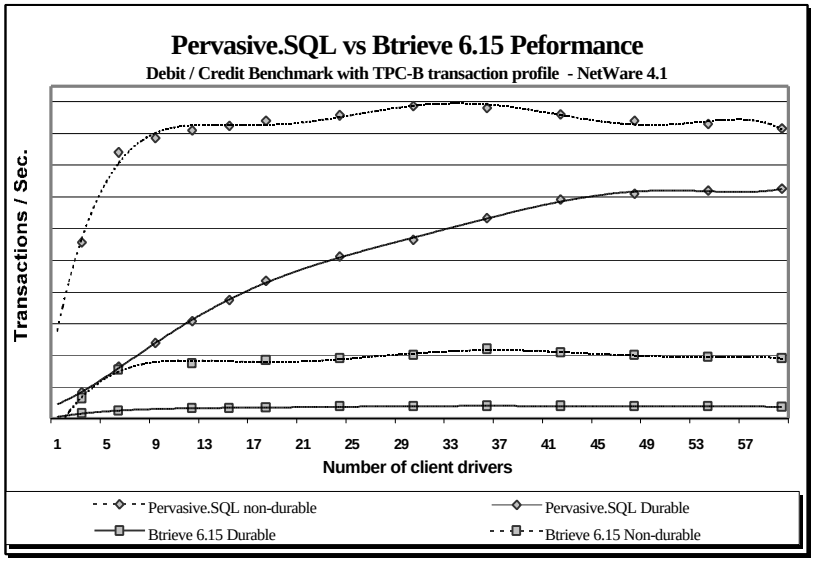

Figure 3

The following diagram shows Pervasive.SQL performance vs Btrieve 6.15 performance running the Debit/Credit benchmark with a 2 second think time. Both Pervasive.SQL and Btrieve 6.15 were configured for durable transactions. In this configuration the data and logs were placed on seperate disk.

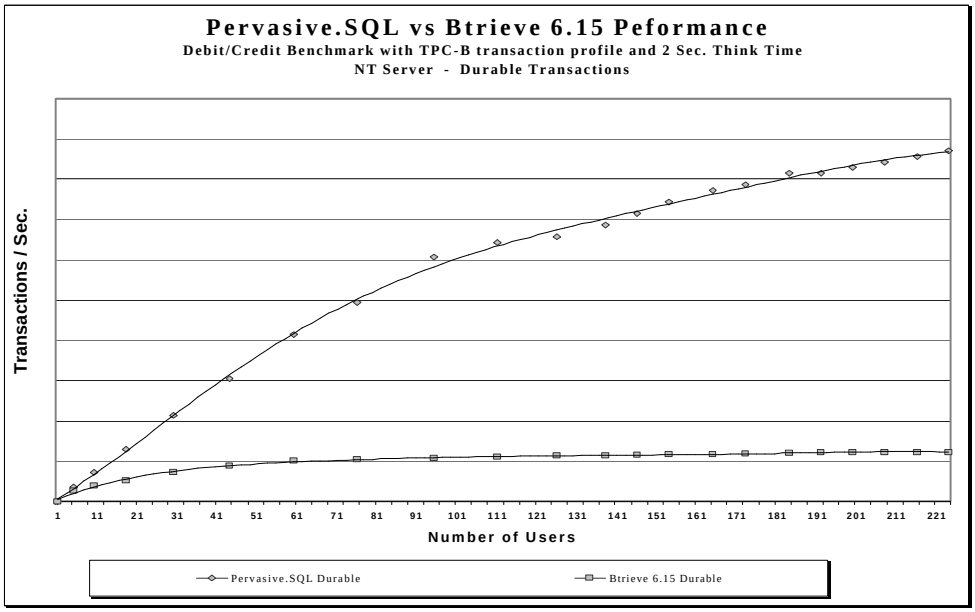

Figure 4

The following diagram shows Pervasive.SQL performance vs Btrieve 6.15 performance running the Debit/Credit benchmark with a 2 second think time. Both Pervasive.SQL and Btrieve 6.15 were configured for non-durable transactions:

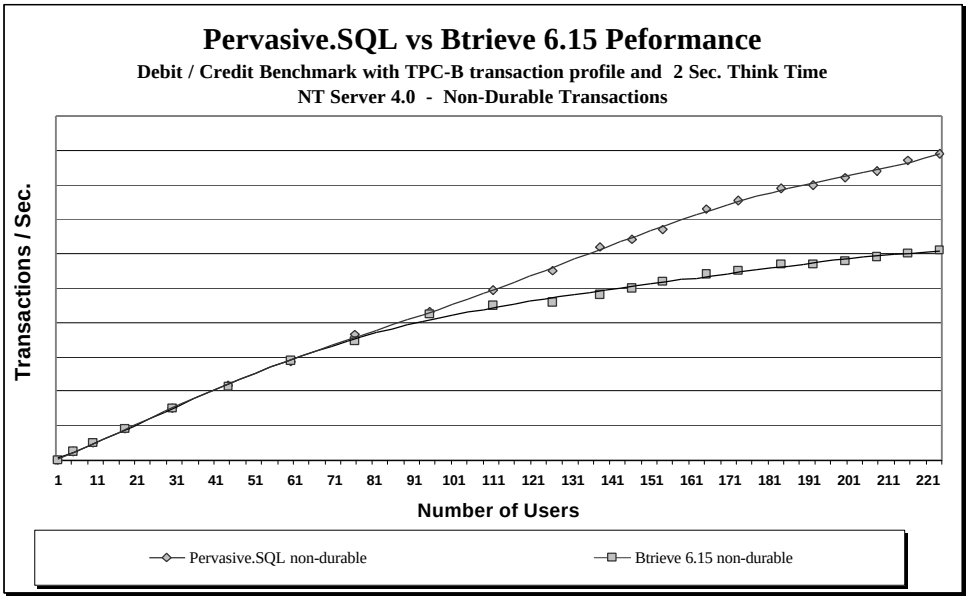

Figure 5

Hardware Configuration

The tested hardware configuration was as follow:

Server:

System:	Dell Gxi 200Mhz Pentium running Windows NT Server 4.0
Memory:	64 MB memory, 512 KB write-back cache
Disks:	2 Seagate 32550 Fast SCSI
SCSI Adapter:	1 Adaptec AHA-2940 PCI-to-Fast SCSI Adapter
Network:	3Com Fast EtherLink XL – 10baseT ethernet

Server and clients on single ethernet segment

Clients:

A Mix of:	200 Mhz Pentiums 64 MB running Windows NT workstation
	200 Mhz Pentiums 32 MB running Win95
	233 Mhz Pentiums 64 MB running Windows NT workstation
	233 Mhz Pentiums 32 MB running Win95
	133 Mhz Pentiums 32 MB running Win95
	90 Mhz Pentiums 32 MB running Win95
	100 Mhz 80486s 32 MB running Win95
	66 Mhz 80486s 32 MB running Win95