OpenTURBO Performance Report

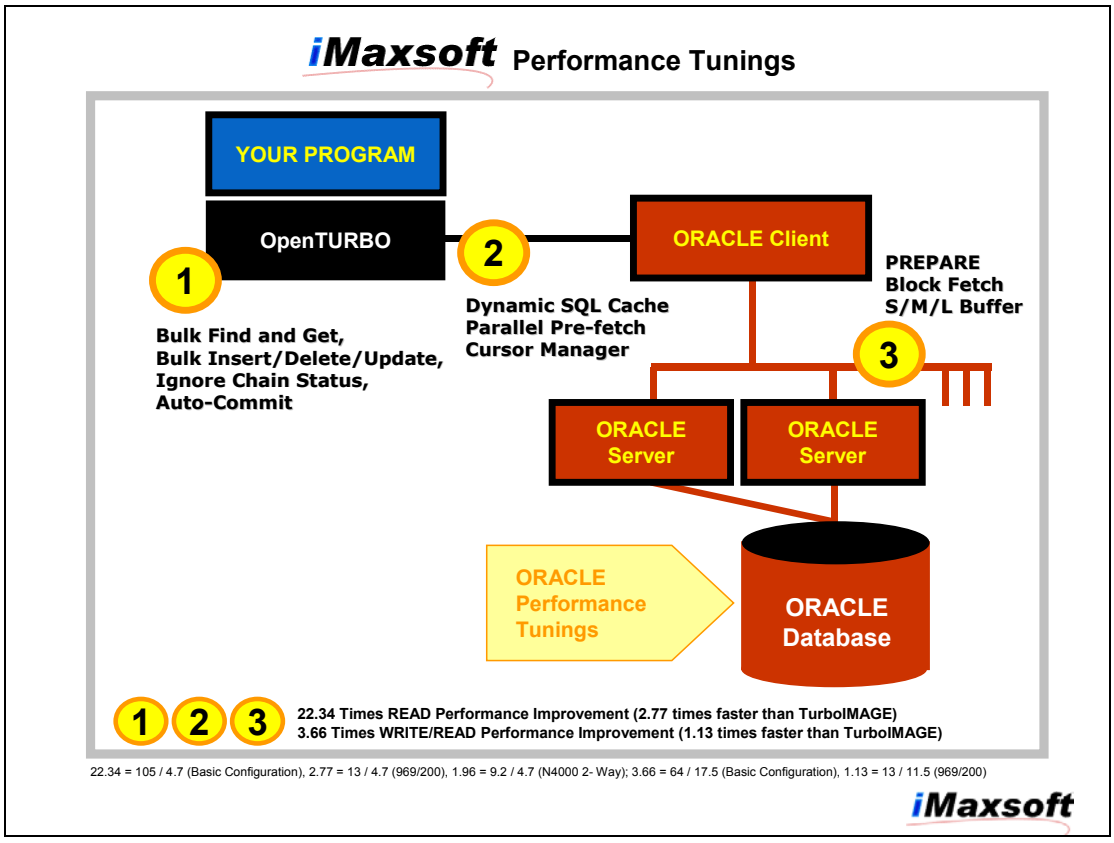
Performance Power Release A.02.00
January 8, 2003
by iMaxsoft OpenTURBO Core Development Team

Abstract:

OpenTURBO Release A.02.00 is a performance release. Our OpenTURBO SQL Core has been re-written and re-architect for efficient data retrieval and robust dynamic SQL statement management. Our new SQL Core provides 2 modes, one is our current standard n-tier client-server mode, which supports cross-machine concurrent TurboIMAGE and ORACLE dual-access, and the other is a performance driver direct mode, which is a dynamic library that can be linked directly to your program and pays no extra network transport overhead. The pre-fetcher and cursor manager are integrated to effectively communicate with ORACLE server process for small, medium or large block and single or multi-threaded data pre-fetching based upon our query optimizer's cost analysis and data access pattern statistics. The SQL Statement Caching is an intelligent resource manager and uses the lease-used-first-out algorithm for resource overflow control.

All tests conducted did not include any ORACLE configurable performance tuning factors, a static and basic ORACLE configuration is used throughout the entire performance benchmark process.

There are 3 areas that causes major performance blockage, 1) the TurboIMAGE emulation in ORACLE, 2) the preparation of ORACLE dynamic SQL statements, and 3) the data transmission between ORACLE client and server. Each area is responsible for 1/3 of overall 95.52% performance improvement according to our test results.



1) the TurboIMAGE emulation in ORACLE; the price for emulating TurboIMAGE in ORACLE is quite high, good fundamental design has major impact to the throughput as well as performance.
2) the preparation of ORACLE dynamic SQL statements; it costs 2 physical roundtrips between ORACLE client and server processes per PREPARE a dynamic SQL statement, a dynamic SQL statement manager with caching capability is developed for managing re-usable ORACLE resources and objects.
3) the data transmission between ORACLE client and server processes; how to effectively trigger ORACLE server pre-fetching and parallel-fetching capabilities are the key to improve overall throughput; and how to properly allocate data transmission block is crucial to the performance.

OpenTURBO READ only performance:

READ only performance tests cover sequential read (DBGET mode 2 and 3), chain find (DBFIND by exact key and b-tree wildcard key), chain read (DBGET mode 5 and 6), direct read (DBGET mode 1 and 4) and hash read (DBGET mode 7 and 8).

OpenTURBO READ performance tuning factors:

OT_BULKCHAINGET - supports forward or backward BULK chain get, but does not support concurrent forward and backward chain get.

OT_IGNORE_CHAINSTATUS - supports forward or backward BULK chain get returning NO Chain Info: chain length and forward and backward chain pointers.

OpenTURBO-ORACLE (READ Only):

	HP9000 RP5470 2-Ways 2G MM
OpenTURBO Standard C/S Mode	Performance Base = 100
OpenTURBO OT_BULKCHAINGET = ON	24.76 (75% Improvement*)
OpenTURBO OT_IGNORE_CHAINSTATUS = ON	15.24 (38% Improvement*)
OpenTURBO Network Overhead	10.67 (30% Improvement*)
OpenTURBO Pre-fetching Manager	7.47 (30% Improvement*)
OpenTURBO SQL Caching Engine	4.48 (40% Improvement*)
Net Result:	**22.5 Times Faster Than Base 95.52% Improvement**

* Improvement = 100% - current line / immediate preceding line * 100%

OpenTURBO-ORACLE vs. TurboIMAGE (READ Only):

	HP e3000 969KS/200 2-Ways 0.5G MM	HP e3000 N-Class 2-Ways 1.5G MM	HP9000 RP5470 2-Ways 2G MM
MPE-TurboIMAGE	14.5 Minutes	9.4 Minutes	
OpenTURBO-ORACLE			4.7 Minutes

OpenTURBO WRITE/READ performance:

WRITE/READ performance tests cover sequential read, chain read, direct read, hash read, delete, put, update, lock and auto-commit (DBBEGIN and DBEND).

OpenTURBO WRITE performance tuning factors:

OT_IGNORE_DBPUTSTATUS - ignores DBPUT returned record number.

DBBEGIN and DBEND - DBBEGIN begins an ORACLE transaction, DBEND issues COMMIT WORK and ends a transaction; this allows you to mimic SQL's AUTO-COMMIT. For example, you can call DBEND for every 1000 inserts, which reduces ORACLE overhead by having 999 less COMMIT WORK, but these 1000 rows will NOT available to other processes until a DBEND, COMMIT WORK, is called.

OpenTURBO and ORACLE (WRITE/READ):

	HP9000 RP5470 2-Ways 2G MM
OpenTURBO Standard C/S Mode	Performance Base = 100
OpenTURBO OT_BULKCHAINGET = OFF	100
OpenTURBO OT_IGNORE_CHAINSTATUS = OFF	100
OpenTURBO OT_IGNORE_DBPUTSTATUS = ON	50.00 (50% Improvement*)
OpenTURBO Network Overhead	35.00 (30% Improvement*)
OpenTURBO SQL Caching Engine	28.00 (20% Improvement*)
Net Result:	**3.6 Times Faster Than Base 72% Improvement**

* Improvement = 100% - current line / immediate preceding line * 100%

OpenTURBO-ORACLE vs. TurboIMAGE (WRITE/READ):

	HP e3000 969KS/200 2-Ways 0.5G MM	HP e3000 N-Class 2-Ways 1.5G MM	HP9000 RP5470 2-Ways 2G MM
MPE-TurboIMAGE	13 Minutes	12 Minutes	
OpenTURBO-ORACLE			11.5 Minutes