Microsoft SQL Server 2000 Basic Capacity Scalability

By Man Xiong

Microsoft SQL Server 2000 running on a Dell enterprise eight-way server can support multiple-terabyte databases and handle heavy workload and administrative tasks. The tests described in this article demonstrate the scalability of SQL Server 2000 running on Microsoft Windows 2000 and Dell hardware.

The growth of e-commerce places enormous user and transaction loads on database applications. As a result, many companies are building systems that can support millions of customers and users as well as manage terabytes (TB) of information. Scalable systems give architects a way to expand the network, servers, database, and applications by simply adding more hardware.

Microsoft® SQL Server 2000 can scale up on symmetric multiprocessing (SMP) systems and scale out on multiple-node clusters. This article focuses on the scale-up scenario and demonstrates the scalability of a single SQL Server database running on a large SMP system, the Dell® PowerEdge® 8450. A 1 TB online transaction processing (OLTP) database, a 2 TB decision support system (DSS) database, and a 4 TB DSS database provided the test scenarios.

The results demonstrate that without special tuning or unrealistic benchmark hardware configurations, SQL Server 2000 can manage very large OLTP and DSS databases by:

- Maximizing the utilization of additional hardware resources (processors and memory) to provide predictable performance as the workload on the system increases
- Minimizing performance degradation while performing online management in the background

Hardware components that affect database performance

The hardware environment is a key consideration to achieve performance in scalable database environments. These specific tests used a static hardware environment: an eight-processor Dell PowerEdge 8450 system with 5 TB of storage on 160 physical disks running on Microsoft Windows® 2000 Datacenter Server. For OLTP-centric solutions, rigorous testing and participation in industry-standard benchmarks have shown that the following hardware considerations can significantly affect the application's performance:

Processors and memory. Database applications can be processor and memory intensive, depending on the database type, the application's focus, and usage patterns. To achieve optimal processor performance in scalable database environments, both processor speed and the size of the level 2 (L2) cache are important. If the application is configured to maximize the available processing power, then the more L2 cache available on the processor, the better the application performance. Additionally, if the available amount of memory creates a bottleneck in the application, adding memory to the system can improve performance.

Disk subsystem. Input/output (I/O) becomes a common bottleneck when scaling database applications. When access to the application data is more random than sequential (as in OLTP), increasing the number of disk drives can mitigate an I/O bottleneck. The application can spread the I/O load across multiple paths and put the performance load on the system processor, rather than the disk subsystem. If the application data access is more sequential than random (as in DSS), additional drives will not provide as much performance benefit.

Networking. The bandwidth available over the network for the application affects performance. High-performance network protocols, such as the Virtual Interface Architecture-based storage area network (SAN) support included in SQL Server 2000, reduce networking bottlenecks in scalable database solutions.

SERVER CONFIGURATION FOR TESTS

Server

- 1 Dell PowerEdge 8450, with 8 Intel® Pentium® III Xeon™ 700 MHz processors and 32 GB of RAM
- 8 QLogic® QLA2200 PCI host bus adapters
- 4 Dell PowerVault® 650Fs; each has 10 disks with 10,000 rpm, 32.9 GB capacity, and 512 KB write-read cache
- 12 Dell PowerVault 630Fs; each has 10 disks with 10,000 rpm and 32.9 GB capacity
- 5 TB total disk space: 160 disks with 10,000 rpm and 32.9 GB capacity

Operating system

- Microsoft Windows 2000 Datacenter Server, Build 5.00.2195, Service Pack 1

Database server

- Microsoft SQL Server 2000 Enterprise Edition, Build 2000.80.194.0

SQL Server and a 1 TB OLTP database

This test uses underlying components and activities representative of typical OLTP systems. The test application models a wholesale supplier managing orders and inventory. Transactions are provided to submit an order, process a payment, record deliveries, and check order status and stock levels. The test examines SQL Server 2000 performance under various CPU and memory combinations as well as workload degradation during online index reorganization.

CPU and memory considerations for OLTP

Adding server CPUs can help scale the system to handle more users or additional workload. OLTP workloads are often very I/O intensive due to simple transactions requiring random disk reads and disk writes. These workloads benefit from increased memory because a larger SQL Server buffer cache greatly reduces data and index page faults, thereby enhancing the workload performance.

SQL Server 2000 on Windows 2000 Datacenter Server can manage up to 64 GB of memory by using the Address Windowing Extensions (AWE). Standard 32-bit addresses can map a maximum of 4 GB of memory, which limits the standard address space of 32-bit Windows 2000 processes to 4 GB. By default, the operating system reserves 2 GB, making 2 GB available to the application.

TEST CONFIGURATION FOR THE 1 TB OLTP DATABASE

Disk configuration

The table in Figure A shows the disk configuration used for the 1 TB OLTP database (DB) test. Both disk write-cache and read-cache are set to 249 MB.

Database space configuration

- Data size: 1046 GB
- Index size: 244 GB
- Space allocated for data and indexes: 1500 GB
- Log space: 60 GB
- Total disk space used (backup excluded): 1560 GB

Database server settings

- The lightweight pooling server option was used for the test.
- The affinity mask option was used to allocate the appropriate number of processors for SQL Server during the CPU tests.
- Address Windowing Extensions (AWE) was enabled for SQL Server to use more memory during memory tests.
- Maximum memory was set to leave 1 GB of memory to the operating system when AWE was enabled.
- All other SQL Server configuration parameters were left at their default values.

Logical drives	Number of physical disks	RAID configuration	Capacity (GB)	Comment
F	14	10	320	For OLTP DB data file and backup space
G	12	10	320	For OLTP DB log file
H	12	10	320	For OLTP DB data file and backup space
I	14	10	320	For OLTP DB data file and backup space
J	12	10	320	For OLTP DB data file and backup space
K	12	10	320	For OLTP DB data file and backup space
L	14	10	352	For OLTP DB data file and backup space
M	12	10	352	For OLTP DB data file and backup space
N	12	10	352	For OLTP DB data file and backup space
O	14	10	320	For OLTP DB data file and backup space
P	12	10	64	For OLTP DB data file and backup space
Q	12	10	320	For OLTP DB data file and backup space

Figure A. Disk configuration for 1 TB OLTP database

By specifying a /3GB switch in the Windows 2000 Datacenter Server Boot.ini file, the operating system reserves only 1 GB of the address space, and the application can access up to 3 GB.

AWE, a set of extensions to the Microsoft Win32® application programming interface (API) memory management functions, allows applications to address more memory than the 4 GB available through standard 32-bit addressing. Applications can acquire physical memory as nonpaged memory and then dynamically map views of the nonpaged memory to the 32-bit address space. Although the 32-bit address space is limited to 4 GB, the nonpaged memory can be much larger. This capability enables memory-intensive applications that support AWE, such as SQL Server 2000 Enterprise Edition, to address more memory than a 32-bit address space can support.

On systems with 16 GB or more of physical memory, it is recommended that the maximum memory for SQL Server be set to at least 1 GB less than the total available physical memory, thereby leaving at least 1 GB for the operating system.

OLTP workload test results. The test runs the OLTP workload on systems with the following CPU and memory configurations:

- Two CPUs, 4 GB of total physical memory
- Four CPUs, 8 GB of total physical memory
- Eight CPUs, 32 GB of total physical memory

OLTP workload performance improves when the number of CPUs is increased along with the memory (see Figure 1). The increase in transaction throughput is nearly linear, demonstrating an excellent return on hardware investment.

Additional tests with different combinations of CPUs and memory show that more memory helps only when CPU cycles are available (see Figure 2). For this database size and OLTP workload, using the full processing power and memory of the Dell eight-way server significantly increases transaction throughput.

Index reorganization

Microsoft SQL Server 2000 includes online index reorganization, a new feature that effectively maintains optimal performance despite the effects of index fragmentation common to OLTP databases. Because this feature can be used online, it supports today's 24×7 requirement for high availability.

Index reorganization test results. Using default parameters, DBCC INDEXDEFRAG runs concurrently with the OLTP workload. First, the workload is run normally at 50 percent CPU utilization to establish a baseline for transaction throughput. Next, an online reorganization is performed with the workload running. In this test, the online reorganization causes transaction throughput to degrade by only one percent.

This test demonstrates that SQL Server 2000 can perform online reorganization with minimal transaction throughput degradation. Results vary with hardware configuration and characteristics of the OLTP workload.

Performance measurements for 2 TB and 4 TB decision-support systems

These tests model a decision-support system that loads data or refreshes data from an OLTP database and produces computed and refined data to support sound business decisions. Supplied queries assist decision makers in five domains of business analysis: pricing and promotions, supply and demand management, profit and revenue management, customer satisfaction, market share, and shipping management. The application's underlying components and activities represent a typical DSS system.

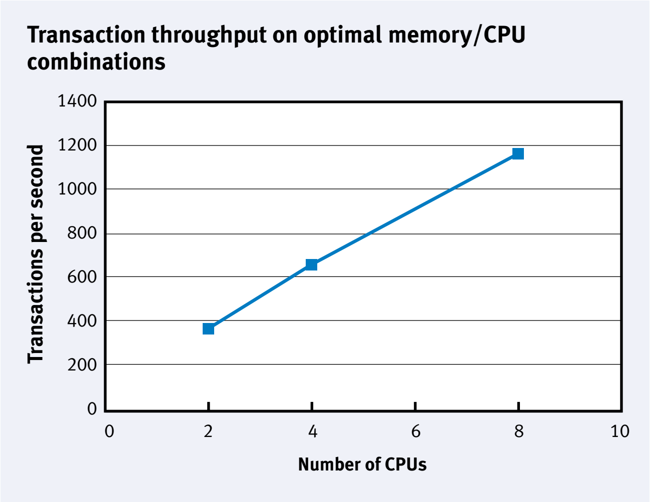

Figure 1. Transaction throughput on optimal memory/CPU combinations

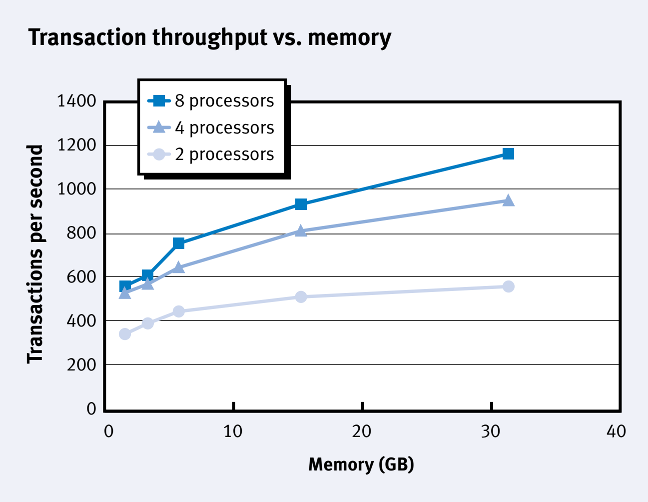

Figure 2. Transaction throughput vs. memory

TEST CONFIGURATION FOR THE 2 TB DSS DATABASE

Disk configuration

The table in Figure B shows the disk configuration used for the tests run against the 2 TB DSS database. Both disk write-cache and read-cache are set to 249 MB.

Database space configuration

- Data size: 1096 GB
- Index size: 566 GB
- Space allocated for data and indexes: 2680 GB (extra space needed for clustered index creation and sorting)
- Log space: 60 GB
- tempdb space: 360 GB

Database server settings

- The affinity mask option was used to allocate the number of processors for SQL Server during the CPU tests.
- All other SQL Server configuration parameters were left at their default values.

Logical drives	Number of physical disks	RAID configuration	Capacity (GB)	Comment
F	10	0	320	For DSS DB data file and backup space
G	10	0	320	For DSS DB log file and backup space
H	10	0	320	For DSS DB data file and backup space
I	10	0	320	For DSS DB data file and backup space
J	10	0	320	For DSS DB data file and backup space
K	10	0	320	For DSS DB data file and backup space
L	10	0	352	For DSS DB data file and backup space
M	12	0	352	For tempdb
N	12	0	352	For tempdb
O	10	0	320	For tempdb
P	4	10	64	For logs of DSS DB and tempdb
Q	10	0	320	For DSS DB data file and backup space
R	10	0	320	For DSS DB data file and backup space
S	10	0	320	For DSS DB data file and backup space
T	10	0	320	For DSS DB data file and backup space
U	10	0	320	For DSS DB data file and backup space

Figure B. Disk configuration for 2 TB DSS database

TEST CONFIGURATION FOR THE 4 TB DSS DATABASE

Disk configuration

The table in Figure C shows the disk configuration used for the tests run against the 4 TB DSS database. Disk read-cache and write-cache are set to 249 MB, except where noted in the test results described in the article.

Database space configuration

- Data size: 2356 GB
- Index size: 1359 GB
- Space allocated for data and indexes: 4096 GB
- Log space: 100 GB
- tempdb space: 800 GB

Database server settings

- The affinity mask option was used to allocate the number of processors for SQL Server during the CPU tests.
- All other SQL Server configuration parameters were left at their default values.

Logical drives	Number of physical disks	RAID configuration	Capacity (GB)	Comment
F	10	0	320	For DSS DB data file and backup space
G	10	0	320	For DSS DB log file and backup space
H	10	0	320	For DSS DB data file and backup space
I	10	0	320	For DSS DB data file and backup space
J	10	0	320	For DSS DB data file and backup space
K	10	0	320	For DSS DB data file and backup space
L	10	0	352	For DSS DB data file and backup space
M	12	0	352	For tempdb
N	12	0	352	For tempdb
O	10	0	320	For tempdb
P	6	1	192	For logs of DSS DB and tempdb
Q	10	0	320	For DSS DB data file and backup space
R	10	0	320	For DSS DB data file and backup space
S	10	0	320	For DSS DB data file and backup space
T	10	0	320	For DSS DB data file and backup space
U	10	0	320	For DSS DB data file and backup space

Figure C. Disk configuration for 4 TB DSS database

These tests examine SQL Server 2000 performance under a standard DSS workload, workload degradation during online management operations, and scalability with increasing database size.

CREATE DATABASE statement

The `CREATE DATABASE` statement initializes the 2 TB and 4 TB databases and allocates the space for the databases.

Scalability on the size of the database. The database loading time scales linearly from a 2 TB database to a 4 TB database. (The average throughput is 730 GB per hour for the 2 TB DSS and 722 GB per hour for the 4 TB DSS.)

Bulk data loading

Microsoft SQL Server 2000 supports bulk database loading from multiple data streams in parallel. This technology makes optimal use of multiple CPUs and available I/O bandwidth.

Effect of multiple streams. The SQL Server 2000 `BULK INSERT` statement is used to load data into the 2 TB and 4 TB DSS databases. Data-loading throughput increases predictably with the number of input streams (see Figure 3).

Effect of increased data volume. The database loading time scales linearly from a 2 TB database to a 4 TB database.

Index creation

Usually, decision-support systems require very complex indexes to speed up query processing. In this test, indexes are built after the data is loaded.

SQL Server 2000 uses clustered and nonclustered indexes. A clustered index determines the physical order of data in a table. A nonclustered index stores the data in one location and the index in another, with pointers to the storage location of the data. The indexed items are stored in the order of the index key values, but the information in the table is stored in a different order (which can be dictated by a clustered index).

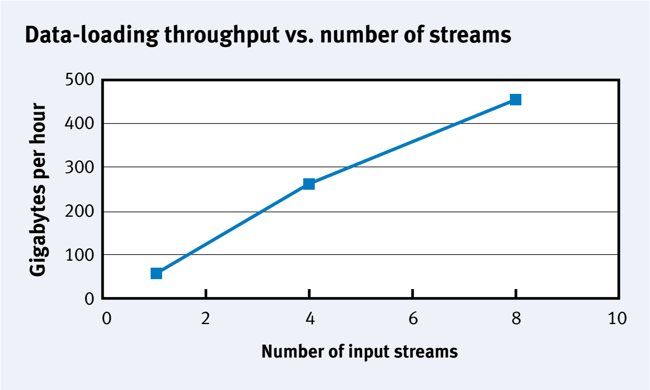

Figure 3. Data-loading throughput vs. number of streams

Clustered vs. nonclustered index creation times. Creating the clustered index on column A of a 0.65 TB table takes 25 percent more time than creating the nonclustered index on the same column, primarily due to the additional time required to move the data into key order.

Effect of multiple CPUs on index creation. A clustered index is built on column A of a 0.65 TB table and a nonclustered index on column B of the same table. All columns are of `int` data type. For testing purposes, the database with unsorted data is backed up once before building these indexes, and then this backup is restored prior to building the desired index.

Increasing the number of CPUs increases index creation throughput for all indexes. Again, throughput increases predictably with additional CPUs.

Effect of table size on index creation time. The index creation time scaled linearly from a 0.65 TB table to a 1.30 TB table, demonstrating that SQL Server 2000 index creation scales predictably.

Disk space required by index creation. Creating a clustered index requires extra disk space—approximately 1.2 times the data volume beyond the size of the existing table data. When a clustered index is created, the table is copied, the data in the table is sorted, and then the original table is deleted. Therefore, enough empty space must exist in the database to hold a copy of the data, the B-tree structure, and some space for sorting.

This test requires 0.85 TB extra disk space so the data files can create a clustered index on a 0.7 TB table. In this test, the recovery mode of the database was set to `BULK_LOGGED`. In this mode, the system requires log space sufficient to record the allocation of storage for the new clustered index and the deletion of the original table. In this configuration, 50 GB of log space is adequate to create a clustered index on a 0.65 TB table.

DSS workload

The test DSS workload comprises complex, read-only ad hoc queries as well as inserts and deletes to simulate the refresh of the database, perhaps from a source OLTP system. Queries are submitted during multiple, concurrent user sessions to simulate a real-world DSS environment. This requires SQL Server to balance system resource allocation among the simultaneously running OLTP and DSS queries, which is done without any human intervention.

Effect of the number of CPUs. This test on the 2 TB DSS database runs eight `osql` connections. Each executes more than 20 representative queries, plus insert and delete activity. Figure 4 shows relative query performance as a function of number of CPUs. The results for two and four processors are given relative to the eight-processor result. This graph illustrates nearly perfect linear scaling of query performance with the number of CPUs using SQL Server 2000.

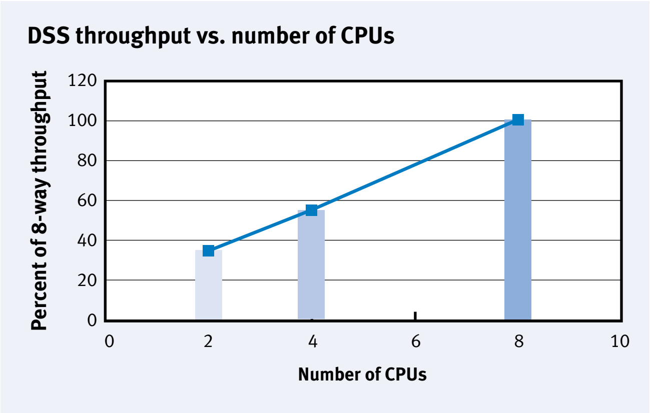

Figure 4. DSS throughput vs. number of CPUs

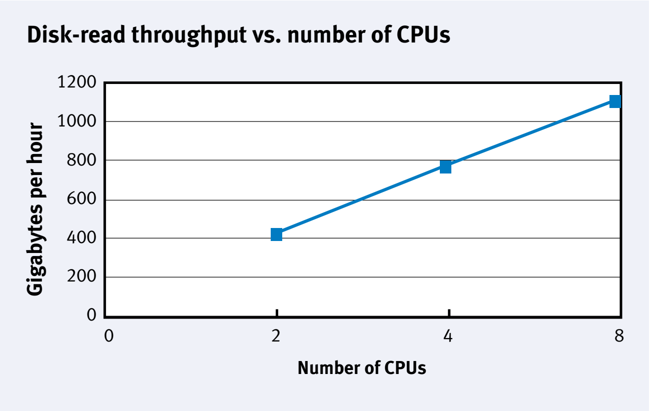

Figure 5. Disk-read throughput vs. number of CPUs

Disk I/O throughput. DSS queries often require sequential disk reads to perform range scans of tables and indexes. SQL Server 2000 automatically uses the maximum available I/O capacity. Disk-read throughput increases as a function of the number of CPUs, which illustrates that SQL Server uses processor power effectively to drive query processing and related I/O activity (see Figure 5).

tempdb space required. All DSS tests used a 300 GB tempdb database. This space is preallocated to prevent variations in the testing because of spontaneous growth of tempdb.

Database verification

`DBCC CHECKDB` performs an online analysis of the integrity of all the database's storage structures. Online verification, like other maintenance operations in SQL Server, is designed to have minimal effect on the production workload.

By default, `DBCC CHECKDB` performs a detailed check of all data structures. Running this check in parallel makes effective use of large systems with many processors.

Usually, `DBCC CHECKDB` should be used with the `PHYSICAL_ONLY` option to check the physical structure of the page and record headers. This operation performs a quick check designed to detect hardware-induced errors.

Effect of the database size. Execution time of `DBCC CHECKDB` scales linearly from the 2 TB to the 4 TB DSS database, demonstrating predictable scaling of verification with database size.

Minimal workload performance degradation. While `DBCC CHECKDB` is run with `PHYSICAL_ONLY` during a heavy concurrent DSS workload, only 3 percent performance degradation is measured on the workload.

Estimating temporary space required by database verification. `DBCC CHECKDB` requires additional space in tempdb. `DBCC CHECKDB WITH ESTIMATE ONLY` is run before `DBCC CHECKDB` to determine the tempdb space needed. Then sufficient physical disk space is allocated to tempdb. `DBCC CHECKDB WITH ESTIMATE ONLY` on the 2 TB DSS database takes four minutes and reports that 38.5 GB of space in tempdb is needed, very close to the actual space used (37 GB). The same command on the 4 TB DSS database takes eight minutes and reports that 81 GB of space in tempdb is needed, very close to the actual space used (83 GB).

The estimation process is a fast and accurate way to predict temporary space requirements for database verification. In addition, the test shows that the time required to obtain the estimate is predictable based on database size.

Database solution offers scalability

The combination of Microsoft SQL Server 2000, Microsoft Windows 2000 Datacenter Server, and the Dell PowerEdge 8450 server scales predictably as database size increases, allowing administrators to deploy mission-critical very large database systems confidently. Excellent performance is achieved at database sizes of 1 TB, 2 TB, and 4 TB. The server handles heavy OLTP and DSS workloads, and it requires no manual tuning of SQL Server parameters. During heavy workload tests, normal management tasks complete in the background with minimal degradation to production work.

SQL Server 2000 maximizes return on investment in SMP systems, allowing users to add processors, memory, and disks to build large, centrally managed enterprise servers.

Man Xiong *(manx@microsoft.com) is a software design engineer in test on the SQL Server Scalability team at Microsoft Corporation.*

FOR MORE INFORMATION

For more information about SQL Server and hardware scalability, visit:

SQL Server home page:
http://www.microsoft.com/sql