SQL Server Megaservers: Scalability, Availability, Manageability

Jim Gray, Microsoft Research

Richard Waymire, SQL Server Development

March 2003

Summary: Microsoft® SQL Server™ has evolved to support huge databases and applications, including multiterabyte databases used by millions of people. SQL Server achieves this scalability by supporting scale up on symmetric multiprocessor (SMP) systems, allowing users to add processors, memory, disks and networking to build a large single node, as well as scale out on multinode clusters, allowing a huge database to be partitioned into a cluster of servers, each server storing part of the whole database, and each doing a part of the work, while the database remains accessible as a single entity. Using scale out, SQL Server 2000 achieved the top Transaction Processing Council Benchmark C (TPC-C) performance results of any database system on any platform.

.NET servers and SQL Server clusters provide high availability and automated management. SQL Server supports high availability through built-in failover and replication technologies. SQL Server also provides a powerful management model based on a user interface, wizards, a job scheduler for repetitive tasks, and SQL-DMO for scripting application-specific operations. SQL Server architecture accommodates modular growth, automated configuration, maintenance, and programming of large server farms.

Copyright

Table of Contents

Introduction

Many successful companies are expanding their online applications as their businesses explode with the growth of e-business, line of business applications, and business intelligence. Now that every Internet and intranet user is a potential client, applications face huge user and transaction loads. Most companies are now building megaservers, managing terabytes of information and supporting millions of customers and users. Database systems are at the core of these megaservers.

Scalable systems provide a way to grow the network, servers, database, and applications by simply adding more hardware. Scalable computer systems can grow an application's client base, database, and throughput without application reprogramming. The expanded server is as easy to manage on a per-user basis as the smaller system.

As Figure 1 shows, systems can grow by:

- Adding hardware to a single node or upgrading to a larger node. This is known as **scale up**.
- Adding more nodes and spreading the data and workload among them. This is known as **scale out**.

Figure 1: Scale up and scale out.

A scalable system allows the designer to start small and grow the system as large as necessary. Some applications—for example customer relationship management—need small nodes, perhaps as small as portable computers that can store parts of the database and perform parts of the application. Ideally, all the nodes of this distributed system present the same operations and programming interfaces.

To date, most scalability has been achieved thorough symmetric multiprocessor (SMP) scale up, that is, by adding more processors, memory, disks, and network cards to a single server. Several vendors have shown that SMP servers can offer a 10-fold scale-up over uniprocessor systems on commercial workloads. Eventually, however, a single-node architecture hits a bottleneck and cannot feasibly grow any further. This bottleneck appears as diminishing returns or prohibitively expensive hardware.

To grow much beyond a factor of 10, application designers have gravitated to a cluster scale-out architecture in which the workload and database are partitioned among an array of SMP nodes. Scale-out systems grow by adding more nodes to the cluster. Although it is in fact an array of nodes, the cluster is programmed and managed as a single system. Ideally, this partitioning is transparent to the clients and to the application. All truly large systems are built as scale-out clusters: the IBM MVS Geoplex and SP2, the HP VMScluster and NonStop Himalaya, and the NCR Teradata system are just a few examples. Clusters are also appearing in the form of storage area networks from EMC, HP, IBM, and others.

Unlike increasingly larger SMP systems, clusters can grow in small increments using commodity components, and the relative independence of cluster nodes gives a natural failover and high-availability design. However, clustering poses management challenges, because more components must be managed.

SQL Server 2000 and Windows Server 2003: SMP and Clustered Megaservers

Microsoft Windows® Server 2003 and SQL Server™ 2000 support both SMP scale-up and cluster scale-out architectures. SQL Server scales down to run on portable computers and even runs on Windows CE, and scales up to run on huge servers. It delivers impressive peak performance for both transaction processing and data warehouse applications.

Although the most common SMP hardware systems are 2-way, 4-way, and 8-way, SQL Server 2000 and Windows Server 2003 can run on SMP hardware systems with up to 64 nodes. These systems can have up to 64 gigabytes (GB) of memory with the 32-bit Intel architecture, and up to 4 terabytes of memory with Intel's new 64-bit Itanium architecture. To date, the largest configurations supported by SQL Server 2000 running on Windows Server 2003 are 32 processors with 512 GB of memory. These systems demonstrate excellent SMP scalability, both on audited benchmarks and in real applications. Today, a single CPU can support 14,000 users accessing a 1-terabyte database, an 8-processor node can support more than 92,000 concurrent users accessing a SQL Server managing billions of records on a 8-terabyte disk array, and a 32-CPU node can support 290,000 users accessing a SQL Server database hosted on 24-terabyte disk array. The largest of these servers are capable of processing more than 1 billion business transactions per day.

A cluster of SQL Server SMP nodes can do even more. In one benchmark, HP demonstrated a 32-node cluster of 8-way Xeon processors supporting more than 575,000 concurrent users, and a 53-terabyte database processed 709,220 Transaction Processing Council Benchmark C (TPC-C) transactions per minute (tpmC), at a cost of less than $15 per tpmC. Based on the TPC-C metrics, SQL Server 2000 has the best peak performance and best price/performance of any database system in the world.

SQL Server is also excellent at decision-support and data-mining tasks, demonstrating outstanding performance and price/performance on the popular TPC-H query set.

SQL Server performance has been more than doubling each year since 1995, as measured by the TPC-C benchmark. Price/performance has been improving at a similar rate. The combination of continuing hardware and software improvements will continue this trend in the near future.

SQL Server 2000: Scalability, Availability, Manageability

SQL Server 2000 Enterprise Edition uses Windows 2000 Server and Windows Server 2003 features to build megaservers. SQL Server uses the extra processors to run extra execution threads and uses the extra memory to store more of the database in memory. The SQL Server relational engine supports high-speed transaction processing, as well as demanding data warehouse applications. The query execution engine exploits multiprocessor and multidisk systems, through parallel hybrid-hash joins and merge joins. The query processor has many innovations including hash teams, joining covering indexes, bit-vector filtering within hash joins, and transparent application access to views of value-based partitioned tables within a cluster. The query executor uses large main memories (up to 512 GB), large asynchronous I/O, and intraquery parallelism for good SMP performance on decision support queries. The optimizer has many special techniques for star schemas and other richly indexed databases. It optimizes batch updates by sorting them before they are applied to the base tables and the indices. The query processor exposes and uses native OLE DB, so it can integrate data from heterogeneous data sources. Using these techniques, SQL Server has the best TPC-C performance and best SMP scalability on nonclustered Intel systems, and the best peak performance on a cluster of SMPs. SQL Server 2000 supports indexed views that are important for report-oriented applications. SQL Server also includes powerful analytical (OLAP) tools to build and process data cubes. It also includes, data mining tools and a text-indexing and retrieval component.

Distributed transactions allow partitioning of SQL Server databases among servers running Windows Server 2003, as well as servers running Windows 2000 Server, Windows XP , and Windows CE. Distributed transactions also allow SQL Server to participate in transactions that span DB2/MVS, UNIX, and Windows nodes, including databases from IBM and Oracle. Microsoft Distributed Transaction Coordinator supports the XOpen XA interfaces, and automatically manages the work of transactions that span these nodes. Microsoft and HP built clusters of 32-node and 45-node clusters that could process one billion transactions per day using Microsoft Distributed Transaction Coordinator. This cluster contained 32 servers running SQL Server, each storing part of the database. Microsoft COM+ managed the application and coordinated transactions among the servers.

Scalability and Availability Features

SQL Server 2000 has powerful scalability and reliability features including:

- Log shipping for hot standby servers.
- Updateable partitioned views among cluster nodes.
- Large memory support (up to 16 terabytes).
- SMP support (up to 64 processors).
- Support for large Windows Server 2003 Data Center Server clusters.
- Support for multiple instances of SQL Server 2000 on a single server.
- Integration with Active Directory to provide location-transparent access to servers running SQL Server.
- Improved parallelism in data and database management operations.
- Indexed views and snowflake schema to support large-scale data warehouses.
- Native XML support for Internet and data interchange operations.
- Notification Services to support client caching and messaging applications.

SQL Server uses the Microsoft Cluster Service to support symmetric virtual servers: each SQL Server cluster node acts as a hot standby for up to three others, while still performing useful work. For disaster recovery, SQL Server supports log shipping from one server to a remote server; in case of catastrophic failure at the primary server, the second server can recover within minutes and continue offering service to customers.

SQL Server has a well-earned reputation for easy installation and management:

- Enterprise Manager allows an operator to monitor and manage multiple instances of SQL Server from a single console embedded within the Windows Server 2003 master console.
- The database is largely self-tuning. As memory is added, SQL Server uses it; as memory pressure from other applications increases, SQL Server releases it. Similarly, SQL Server dynamically grows and shrinks the database and log space based on demand.
- The system computes database statistics and performs other housekeeping tasks automatically, freeing the database administrators and operators to focus on higher-level issues.
- SQL Server provides an extensive collection of wizards to help administrators automate standard tasks. There is a job scheduler to perform these tasks on a regular basis. The alert system records events in the Windows event log, and alerts the operator by e-mail or page. It also optionally invokes a user-defined database procedure for each event class.
- SQL Server 2000 supports multiple instances on one SMP node. A single large SMP node can host multiple servers each serving multiple databases.

SQL Server supports exabyte-sized databases; the only practical limit is how long it takes to backup, recover, and reorganize the database. In addition, the SQL Server product has made enormous strides in this area over the last few years: backup and restore are now

incremental and restartable. Computer Associated backed up 2.6 terabytes per hour and restored at 2.2 terabytes per hour with only a 25 percent degradation in online throughput. Even more impressively, SQL Server can recover a 2.5-terabyte database from a disk shadow copy in 11 minutes, using Windows Server 2003 shadow-disk technology. These rates are improving with improved storage and network technologies. For more information, search on "Multi-Terabyte Backup Benchmarking," at the Computer Associates website at http://ca.com/.

SQL Server is designed to monitor and tune itself automatically as the load or hardware environment changes. Tools have been added to help design the database, to watch the system, to display system status and query plans graphically, to recommend reorganization, and to help the operator manage routine tasks. A sophisticated built-in workflow system orchestrates data scrubbing, data transformation, and data loading typical of most data marts and data warehouses. Finally, a wizard examines the system workload and recommends better physical designs.

Clusters allow SQL Server to scale out to arbitrarily large databases. Windows Server 2003 clusters provide modular growth, allowing customers to buy just what they need, and grow the system by adding processing, storage, and network modules to the server as demand rises. Microsoft simplifies building and managing these megaservers; indeed, Microsoft wants to bring the ease of Plug and Play to enterprise clusters, and is automating much of the work of configuring and managing a cluster. SQL Server 2000 offers transparent access to data partitioned in such clusters. Partitioned clusters potentially allow a multi-hundred-terabyte database. Such a system should be big enough for almost any application; moreover, if current price trends continue, by 2005, such a cluster could be built from commodity components for a few million dollars.

The following table documents the scalability status of SQL Server. The numbers cited are not hard limits, but instead indicate how far Microsoft expects the tools to scale.

SQL Server 2000 Scalability as of March 2003			
Technologies	**Active Users**	**Throughput**	**Database Size**
SMP, failover, parallel query, distributed transactions, SQL Server Enterprise Manager	400,000	300,000 transactions per minute250 million transactions per day	40 terabytes
Clusters of SMPs, failover, parallel query, distributed transactions, SQL Server Enterprise Manager	700,000	500,000 transactions per minute 1 billion transactions per day	60 terabytes
Data warehousing and decision support, star schemas, complex query optimization, data cubes, data mining technologies	100	2,000 queries per hour	3 terabytes

At the other end of the scalability spectrum, SQL Server 2000 has scaled down to small Windows systems, and offers disconnected operation for these systems to support mobile applications. A Windows CE version of SQL Server is available.

Scalability Metrics

Kinds of Growth

As organizations grow and acquire more data, they must deal with increased transaction workloads and larger databases. Each major increment presents new scalability challenges.

Scalability covers several kinds of growth:

- **Growing user population and network loads.** If the user population doubles, the network and database workload probably does too.
- **Growing database size.** With databases commonly reaching hundreds of gigabytes, operations such as backup, restore, and load can become bottlenecks.
- **Growing transaction complexity.** Application designers are building more intelligence into applications, relieving users of tedious tasks. Increasingly, applications are used for data mining and data analysis.
- **Growing applications.** As applications become easier and less expensive to build, organizations are using computers in new ways. These new applications increase the load on existing databases and servers.
- **Growing numbers of servers.** Clustered and distributed applications involve many nodes. As desktops and portable computers grow in power, they acquire local data stores and replicas of key data sources. Scalable systems allow a large number of nodes to be managed easily from a single location.

Scale Up, Scale Out, and Speed Up

An ideal system's performance scales linearly; that is, if you double the number of processors and disks, throughput doubles, or response time is cut in half. These results are known as linear scale up and linear speed up, respectively. Linear scaling is rarely achieved in practice, however, because it requires all aspects of the system to be perfectly scalable.

It is tempting to simplify scalability to a single metric, such as the number of processors a system can support. However, many database applications are very I/O intensive, so adding CPUs to an I/O-bound system will not make it faster. Microsoft SQL Server running on today's typical 4-processor servers can achieve performance comparable to other software running on hardware with 10 processors.

SQL Server 2000 and Windows Server 2003 support 64 GB of main memory on Intel 32-bit architectures, and up to 4 terabytes of main memory on the Intel Itanium 64-bit architecture. Huge main memory reduces I/O traffic and gives SQL Server a substantial performance boost. SQL Server 2000 also supports 32-way SMPs and large clusters built from these SMPs.

There is no universally accepted measure of scalability. However, useful information can be gained from Transaction Processing Performance Council (TPC) benchmarks (see http://www.tpc.org). The TPC is a nonprofit organization that defines industry-standard transaction processing and database benchmarks. Members of the council today include all

major database vendors and suppliers of server hardware systems. They have defined a series of benchmarks, TPC-A, TPC-B, TPC-C, TPC-D, TPC-H, TPC-R, and TPC-W.

TPC-C is the industry-standard benchmark for measuring the performance and scalability of OLTP systems. TPC-C tests a broad cross-section of database functionality including inquiry, update, and queued minibatch transactions. The specification is strict in critical areas such as database transparency and transaction isolation. Many consider TPC-C a good indicator of real-world OLTP system performance. Independent auditors audit the benchmark results, and a full disclosure report is filed with the TPC. These reports contain a lot of information about how easy the various systems are to use and how much the systems cost.

The audited results in the following table show that SQL Server 2000 and Microsoft Windows Server 2003 deliver excellent SMP scalability for up to 32 processors. Indeed, it has better TPC-C performance and price-performance than any SMP result (or cluster result) reported by Oracle or DB2 on any platform.

SMP Performance and Price Performance (8 to 64 cpus) on the TPC-C Benchmark* SQL Server vs. DB2 vs. Oracle

Database	Hardware	CPUs	tpmC	$/tpmC	System Cost	Availability
SQL Server 2000 Enterprise	HP Proliant DL760-G2 8P	8	115,025	$7.69	$ 884,216	3/31/2003
Oracle 9i R2 Enterprise Edition	IBM eServer pSeries 660 - 6M1	8	105,025	$23.45	$2,462,401	9/21/2001
DB2/AS400 V4 R5	IBM eServer iSeries 400 - 840-2420-1	24	163,776	$51.58	$8,448,137	12/15/2000
Oracle 9i R2 Enterprise Edition	IBM eServer pSeries 690	32	427,761	$17.75	$7,591,038	05/31/03
Oracle 9i R2 Enterprise Edition	HP 9000 Superdome	64	423,414	$15.64	$6,621,072	8/26/2002
SQL Server 2000 Enterprise 64-bit	NEC Express 5800/1320Xc C/S	32	433,108	$12.98	$5,619,528	6/30/03

*Best SMP results from each database vendor as of March 6, 2003.

The table shows recent SMP TPC-C benchmarks of SQL Server and the best comparable results of the other database vendors on 8-way and 32-way SMP servers. It shows that SQL Server on an 8-way SMP server supporting over 100,000 tpmC and accessing an 8-terabyte database, SQL Server has the best 8-way SMP performance—better than any 8-way UNIX systems running DB2 or Oracle. The Microsoft solution is also three times less expensive. On 32-processor systems, SQL Server achieves slightly better performance than the best reported Oracle result. Overall, SQL Server has better peak performance than DB2 or Oracle, and is typically much less expensive than the UNIX solutions.

The previous table showed the best TPC-C results for single-node SMPs – the scale-up metric. In fact, most large servers are actually farms of Web servers that front-end clustered database servers. This is the scale-out design, for which performance results are shown in the following table.

Clustered Performance and Price Performance (8 to 64 cpus) on the TPC-C Benchmark* SQL Server vs. DB2 vs. Oracle Clustered						
Database	**Hardware**	**CPUs**	**tpmC**	**$/tpmC**	**System Cost**	**Availability**
SQL Server 2000 Enterprise Edition	HP ProLiant DL760-900-256P	272 (34x8)	709,220	$14.96	$ 10,603,803	10/15/01
Oracle 9i R2 Enterprise	HP ProLiant DL580-PDC 32P	32 (8x4)	138,362	$17.38	$2,404,503	3/5/03
Oracle 9i R2 Enterprise	HP ProLiant DL580-PDC 32P	32 (8x4)	137,261	$18.46	$2,533,095	9/6/02
SQL Server 2000 Enterprise Edition**	HP Proliant DL760-G2 8P	8 (1x8)	115,025	$7.69	$ 884,216	3/31/03

*Best cluster results as of March 6, 2003.

**This nonclustered result is shown for comparison purposes.

SQL Server has long dominated the scale-out category of the TPC-C benchmark. As expected, these scale-out results are substantially higher than those scale-up results possible with SMPs. SQL Server 2000 holds the top performance spots. SQL Server 2000 delivers the best performance and the best price/performance of any database system. Its performance is 60 percent higher than the performance of its closest competitor. Two Oracle results are reported here, one on Windows 2000 Server and one on Linux, using nearly identical hardware (although the benchmarks are done about 6 months apart, so the Linux hardware is slightly cheaper). The best SQL Server result is more than five times better than Oracle's best cluster result; indeed, Oracle's best cluster result is more comparable to a single node, 8-CPU SQL Server result. Also, the Oracle price/performance is uniformly worse.

In summary, it is fair to say that the TPC-C results show that SQL Server has the best peak performance and the best price/performance of any product on any platform.

SQL Server 2000 scales nearly linearly on a cluster of HP nodes running Windows 2000 Server. Figure 2 above shows the thousands of transactions processed as a cluster grows by adding groups of 8-processor SMPs. The cluster starts with 16, then 24, and then 32 SQL Server nodes. (The actual node counts are 17, 26, and 34; the extra nodes act as transaction coordinators.) The largest system consists of 272 CPUs, serving a 58-terabyte database spread across more than 3,000 disks. The largest system should be enough for even the largest e-commerce sites, but if not, the cluster can be grown by adding more nodes

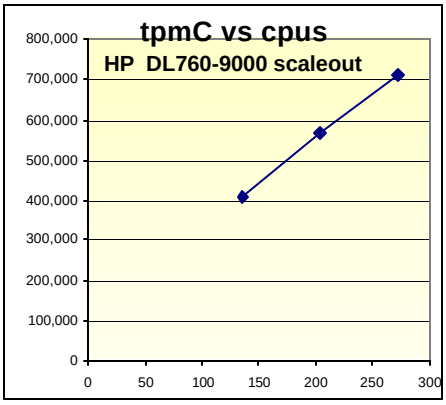


Figure 2: Throughput vs. CPUs

to the cluster. These numbers are beyond anything reported by any other system on any platform.

SQL Server on a cluster of HP DL76-9000 servers shows near-linear scale out as nodes are added (once the initial cost of the scale-out design is paid). This shows the scalability from 17 to 26 to 34 8-processor SMP nodes (272 CPUs in all). As the system scales up, disks and network bandwidth are added along with processors and memory.

The TPC-W benchmark models a Web server application with much more complex transactions. The primary metrics are the performance metric of Web Interactions Per Second (WIPS) and the price-performance metric of $/WIPS. To date, only IBM/DB2 and SQL Server have reported results. The best results so far are shown in the table below.

Best Results on TPC-W Benchmark (100,000 Items)*				
Database	**Hardware**	**WIPS**	**$/WIPS**	**Availability**
SQL Server 2000 Enterprise Edition	Unisys ES 70000	10,440	$106.73	7/10/01
IBM DB2 UDB 7.2	IBM eServer xSeries 430	7,554	$136.80	6/8/01
Oracle	No entry	-	-	-

*Best SMP results of each vendor as of March 6, 2003.

Turning to decision support and reporting workloads, the TPC has defined a widely reported workload, TPC-H, The TPC Benchmark™H (TPC-H) is a decision support benchmark that defines a suite of business oriented ad-hoc queries and concurrent data modifications. The benchmark illustrates decision support systems that analyze large data volumes with complex queries. The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size). SQL Server has reported TPC-H results in the 100-GB and 300-GB categories.

The best results for 300 GB are shown in the table below. No clear pattern emerges from this table. However, SQL Server is among the least expensive solutions and has respectable performance on a 16-processsor SMP server.

SQL Server vs. SMP UNIX Solutions on the TPC-H Benchmark (300 GB)						
Database	**Hardware**	**CPUs**	**QphH @300 GB**	**$/QphH @300 GB**	**System Cost**	**Availability Date**
Informix XPS 8.31 FD1	HP Alpha Server ES40 Model 6/667	16	2,832	$1,058	$2,995,034	02/14/01
SQL Server 2000 Enterprise Edition. 64-bit	Unisys ES7000 Orion 130	16	4,774	$219	$1,043,153	03/31/03
Oracle 9i R2 Enterprise Edition	HP Alpha Server ES45 Model 68/1000	16	5,976	$453	$2,706,063	06/01/02
IBM DB2 UDB 7.2	HP ProLiant DL760 x900-64P	64	12,995	$199	$2,573,870	06/20/02

*Best performance results of each vendor as of March 6, 2003.

TPC results are confirmed by many other benchmarks tests from the likes of SAP, PeopleSoft, and others. The following table summarizes these benchmarks. They show that SQL Server ranks first in several significant end-user benchmarks.

Best Results on Application Benchmarks		
Benchmark	**World Record**	**Winner**
TPC-C	709,220 tpmC $14.96 per tpmC	SQL Server
TPC-W	21,139 WIPS @ 10,000 $32.62 per WIPS	SQL Server
TPC-H	27,094 QphH @ 3 TB $240 per QphH	Oracle
SAP R/3 Sales & Distribution	47,528 concurrent users	IBM
SAP R/3 Retail	3.2 billion sales data line items/hr	SQL Server
Great Plains Software	2,400 concurrent users	SQL Server
Onyx	57,000 concurrent users	SQL Server
Pivotal eRelationship	20,000 concurrent users	SQL Server
CA Brightstor Backup	2.6 terabytes per hour	SQL Server
PeopleSoft eBill Pmt	191,694 payments per hour	SQL Server
PeopleSoft CRM 8.4	25,400 concurrent users	SQL Server
PeopleSoft Financials	15,000 concurrent users	IBM
J.D. Edwards OneWorld	9,000 concurrent users	Oracle

The best performance metric, of course, is how well the system scales for your application. That is impossible to know in advance, but you can estimate scalability by examining standard benchmarks and by looking at applications in related industries.

Scalable Hardware Architectures

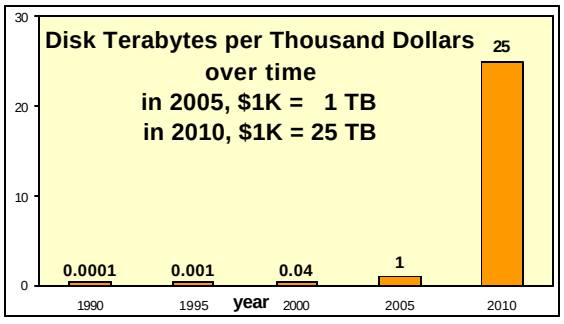
Technology Trends Encourage Building Scalable Systems

Today, commodity components make up the fastest and most reliable computers, networks, and storage devices. The entire computer industry uses the same family of RAM chips that go into computers. Memory prices for commodity computers are often three to ten times less than the price of the same memory for a proprietary computer. Meanwhile, the intense competition among microprocessor vendors has created incredibly fast processing chips. Indeed, the most powerful supercomputers are built from such chips. Conventional water-cooled mainframes are moving to this higher-speed technology.

Still, commodity workstations and servers rival and often outstrip the performance of mainframes. The pace of change in this commodity market is so rapid that the low end is years ahead of the installed base of conventional mainframe and minicomputer architectures.

On the networking side, commodity computer interconnects are also making extraordinary progress. Ethernet speeds have graduated to 120 MB per second. Switched Ethernet gives a 100-fold speed up in local networks at commodity prices. Ten-gigabit and 40-gigabit Ethernet networks are on the horizon. Switched Ethernet, fiber-channel, and other new interconnection technologies offer inexpensive high-speed system area networks (SANs) that will be the basis of clustered architectures in the next few years.

As for storage, the highest-performance and most reliable disks are 3.5" SCSI disks. They have doubled in capacity every year, and are rated at a 50-year mean time to hardware failure. Today, 74-GB disks are standard, and 300-GB disks are available for high-capacity applications. By the end of 2005, disks of 1-terabyte capacity are predicted. In early 2003, $1,000 buys 300 GB of disk capacity. This is a 10,000-fold improvement over disks of 20 years ago. These low prices explain why today's typical servers are configured with several terabytes of disk capacity: such disks cost about $3,000 per terabyte. The chart also suggests that by 2005, servers will typically have 50 terabytes of disk storage, and that by the end of the decade, petabyte-sized disk stores will be common. These will contain very large databases.



Computer Architecture Choices for SMPs and Clusters

The proliferation of many processors, disks, and networks poses an architectural challenge: What hardware architecture best exploits these commodity components? No single winner has emerged, but there is broad agreement that three generic architectures can provide scalability: shared memory, shared disk, and shared nothing. Shared memory is used by SMP systems, while shared disk and shared memory are used by clusters. Windows 2000 Server and Windows Server 2003 support all these architectures, and will evolve in parallel as these architectures evolve.

SMP Systems

SMP grows a server by adding multiple processors to a single shared memory. The system grows by adding memory, disks, network interfaces, and processors. SMP is the most popular way to scale beyond a single processor. The SMP software model, often called the shared-memory model, runs a single copy of the operating system with application processes running as if they were on a single-processor system. SMP systems are relatively easy to program, and they leverage the benefits of industry-standard software and hardware components. SQL Server 2000 is designed to scale well on SMP systems.

The practical limits for general-purpose use today on a single SMP node are:

- 64 processors.
- 512 gigabytes of main memory.
- 30 terabytes of protected storage (400 74-GB disk drives configured as 60 hardware RAID sets and 10 logical volumes) per node.
- 400,000 active clients accessing a SQL Server via the IIS Web server or some transaction monitor.

These are the maximum sizes Microsoft has seen. Typical large servers are half this size or less. With time, SQL Server, Windows, and hardware technology will improve to support even larger configurations.

SMP Scalability

Today, SMP is by far the most popular parallel hardware architecture, and with good reason. SMP servers based on industry-standard Intel microprocessors deliver astonishing performance and price/performance for database platforms. Intel markets an 8-way SMP board based on the Xeon processor that is incorporated in servers from many hardware vendors. Intel 8x Xeon servers have been the workhorses of client-server and e-commerce computing for the last few years. Now 32-processor Xeons and Intel's 64-bit Itanium architecture are also maturing.

The most impressive SMP numbers take advantage of the massive main memory enabled by these Itanium processors. SQL Server achieves nearly linear SMP scalability on the TPC-C online transaction processing benchmark. Throughput (transactions per minute) increases as CPUs are added.

SMP systems become increasingly expensive to build as microprocessor speeds increase. The price steps are modest as a system scales up from 1 processor to 4 processors. Going to 8 processors is also relatively easy. Going beyond 32 processors, however, is a challenge, prices rise steeply, and the returns diminish.

At the software level, multiple processors concurrently accessing shared resources must be serialized. This serialization limits the practical scalability for individual applications in shared-memory SMP systems. These software bottlenecks in operating systems, database systems, and applications are as significant as the hardware limitations.

Nonetheless, SMP systems are the most common form of scalability and will remain so for many years to come. Intel Xeon, and Itanium processors give very powerful and inexpensive SMP nodes.

Diminishing Returns on SMP Performance with Added CPUs

Database	Hardware	CPUs	tpmC	$/tpmC	Sys Cost	Available
SQL Server 2000 Standard Edition	Dell PowerEdge 2650/2.4/1P	1	16,256	$2.78	$46,502	9/11/02
SQL Server 2000 Enterprise Edition	HP ProLaint ML530G3 2P	2	26,725	$3,72	$99,211	3/31/03
SQL Server 2000 Enterprise Edition	HP ProLiant DL580-G2/2GHz 4P	4	77,905	$5.32	$413,764	12/31/02
SQL Server 7.0 Enterprise Edition	HP ProLiant DL760-G2 8P	8	115,025	$7.69	$884,216	3/31/03
SQL Server 2000 Enterprise Edition	Unisys ES7000 Orion 230	32	234,325	$11.59	$2,715,310	3/31/03
SQL Server 2000 Enterprise 64-bit	NEC Express 5800/1320Xc C/S	32	433,108	$12.98	$5,619,528	6/30/03
Oracle 9i Enterprise	IBM eServer pSeries 690	32	427,760	$17.75	$7,179,598	5/31/03

The table above shows that SQL Server delivers good performance on standard and widely available SMPs. Using commodity hardware SQL Server delivers very cost-effective database support. Comparing the 32-processor SQL Server system to a 32-processor Oracle UNIX computer shows that the UNIX system costs 1.6 times as much, but delivers only 18 percent better performance.

Windows Server 2003 clusters provide scale out, allowing customers to add processing, storage, and network services to a pre-existing configuration. The picture to the right shows a cluster that grew from one 8-cpu node –to a six-node 48-cpu cluster by adding one node at a time. In this case, each node is an SMP megaserver Clusters are often built with commodity components and commodity interconnects. Dual interconnects give fault tolerance.

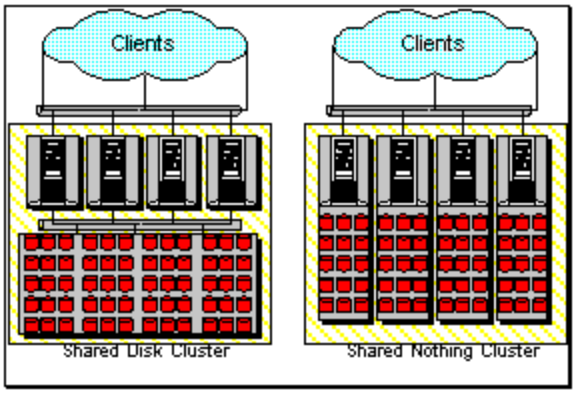
Cluster Architectures

A cluster is a set of loosely coupled, independent computer systems that behave as a single system. The cluster nodes may be single-processor systems, or they can be SMPs. The nodes may be connected by a commodity network or by a proprietary, very high-speed communications bus. The computers in a cluster cooperate so that clients see the cluster as if it were a single very high-performance, highly reliable server. Because the cluster is modular, it can be scaled out incrementally and at low cost by adding servers, disks and networking.

Microsoft believes that the cluster is the most economical way to achieve scalability beyond 8 processors. Cluster architectures allow scale out to systems more powerful than any single SMP node. When demand exceeds the capacity of commodity SMP nodes, or when fault tolerance demands a second failover server, forming a cluster from multiple nodes is an attractive option.

SQL Server 2000 and Windows clusters bring scalability and fault tolerance to the commodity marketplace. Microsoft has built clustering technology directly into the Windows 2000 Server and Windows Server 2003 operating systems; this technology works well with commodity servers and interconnects, and it leverages special hardware accelerators from vendors like HP, Dell, HP, IBM, and Unisys. Microsoft BackOffice® products such as SQL Server, Internet Information Server, and Exchange take advantage of this clustering support. Many third-party products have also been ported to this architecture.

Shared-Disk and Shared-Nothing Clusters



There are two basic models for clusters: shared disk and shared nothing. In a shared-disk cluster, all processors have direct access to all disks (and data), but they do not share main memory. An extra layer of software, called a distributed cache or lock manager, is required to manage cache concurrency globally among processors. IBM's DB2/OS390 SysPlex and Oracle Parallel Server are common examples of the shared-disk parallel database architecture. Because the lock or cache manager serializes

access to data, the shared-disk cluster has some of the same scalability limitations that shared-memory SMP systems have.

A shared-nothing cluster, by contrast, is a federation of database systems. Each node in a shared-nothing cluster is a free-standing computer with its own resources and operating system. Each node has its own memory and disk storage; nodes communicate with one another by exchanging messages across a shared interconnect. Each is a unit of service and availability. Each node owns and provides services for some disks, tapes, network links, database partitions, or other resources. In case of node failure, the disks of one node may fail over to an adjacent node, but at any instant, only one node is managing each disk. It is easier to build shared-nothing clusters from commodity components.

SQL Server 2000 Clusters

SQL Server 2000 supports the shared-nothing cluster model with distributed partitioned views and with distributed transactions. A table can be partitioned by primary key values into disjoint member tables, each stored at one node of the cluster. A distributed partition view defined at each node unifies the member tables into a location-transparent view at each node. Applications can use this view to access the union of the member tables as a virtual table.

The clustered SQL Server TPC-C benchmark results described earlier used distributed partition views in a shared-nothing cluster of up to 34 nodes. The resulting performance exceeds the performance reported for any other database system by a wide margin. Indeed, Figure 3 below shows the performance of SQL Server on this growing cluster, as well as the price/performance ($/tmpC) against performance (tpmC) of all reported systems over 100,000 tpmC as of December 2002. SQL Server has the best performance and the best price/performance by a wide margin. Notice that price/performance for SQL Server is about $13/tpmC for systems ranging from 100,000 tpmC to 700,000 tpmC. The UNIX systems are more expensive and deliver less throughput.

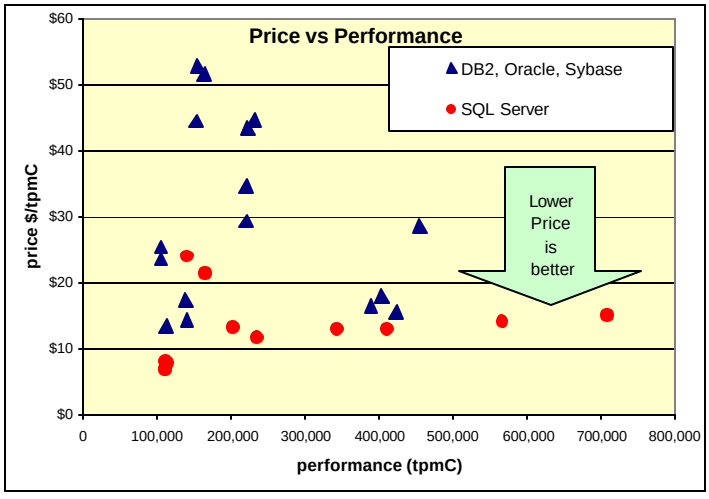


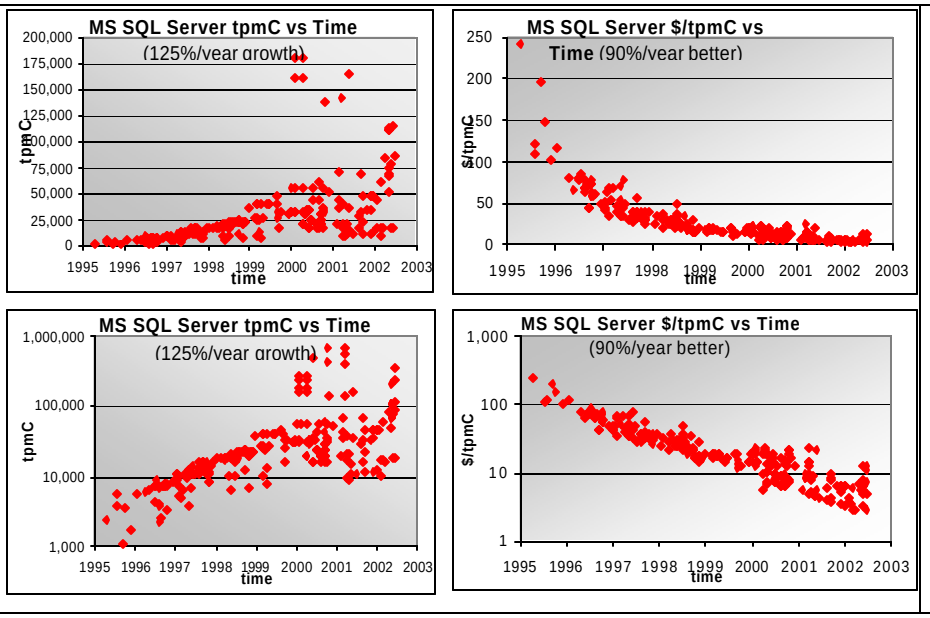
Figure 3: Price/performance vs. performance.

SQL Server Performance Improvements

In round numbers, performance has doubled and price has dropped in half every year for the seven years 1995-2002.

The performance improvements over time of SQL Server on the TPC-C benchmark are impressive. The graphs below show the peak throughput and peak price/performance reports of SQL Server since early 1995. The performance has improved from 2,455 tpmC to 709,220 tpmC, a 290-fold improvement in 7 years. The price has meanwhile dropped from about $240/tpmC to less than $3/tpmC. This is approximately a 90-fold price improvement. Annualized, this represents a 125 percent improvement in performance and a 65 percent improvement in price each year. In round numbers, performance has doubled and price halved every year for the past 7 years. With the introduction of clustering in SQL Server 2000, there is no practical limit on how large a transaction processing database you can build with SQL Server.

Figure 4: SQL Server performance gains since 1995



The performance and price/performance data for SQL Server have improved dramatically over the last 7 years. These graphs show progress on the TPC-C benchmark from 1995 to 2003. The upper graphs are linear, and the lower are semi-log to show the growth rate.

A single instance of SQL Server on Windows can support thousands of users accessing a database containing billions of records. Such systems are capable of supporting a user community exceeding 250,000 or a much larger community of Internet users who are occasionally connected to the server. Just to give a sense of scale, the largest American banks have about 10,000 tellers, and the largest telemarketing organizations have fewer than 10,000 active agents; these systems could support the traditional front office of huge corporations.

As a demonstration, the SQL Server team built a large multimedia database, called TerraServer. It stores several terabytes of satellite images of the earth, 10 million square kilometers in all. The images are stored in 350 million database records on 324 HP StorageWorks™ disks. The server has been on the Internet since June 1998. In that time, it has served 6 billion queries to several million visitors.

A second demonstration, also using SQL Server, partitioned a 1 billion-record banking database among 20 servers running SQL Server, each running on one of 20 nodes. The

database partitioning was managed by the application. COM+ and Microsoft Distributed Transaction Coordinator were used to coordinate transactions that involved more than two servers.

Using partitioned views, SQL Server 2000 demonstrated extraordinary performance: a peak of 709,220 tpmC at $14.96/tpmC on a 34-node cluster. This is a nearly 6-fold increase over the single-node performance, and involves a 54-terabyte database. By adding more nodes, that system could scale to even higher performance.

These examples show that SQL Server can handle huge transaction volumes (millions per day), huge user communities (tens of thousands) and huge databases (terabytes). The product has moreover been improving by a factor of 2 to 3 each year. The improvements are partly due to faster and less expensive hardware, but SQL Server itself has been improving very rapidly. We believe that both the hardware and software improvements will continue at this blistering pace for the foreseeable future.

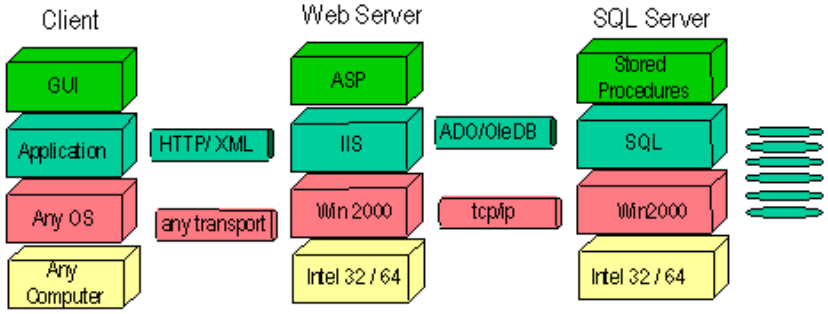
SQL Server Software Scalability Architecture

SQL Server 2000 Application Architecture

SQL Server 2000 is designed to fit into the .NET architecture. Its tools and support system help customers build Active Server Pages and COM+ objects that implement the business logic and access data managed by SQL Server.

Each Windows node typically has a single SQL Server address space managing all the SQL databases at that node. SQL Server runs as a main address space with several pools of threads. Some threads are dedicated to housekeeping tasks like logging, buffer-pool management, servicing operations requests, and monitoring the system. A second, larger pool of threads performs user requests. These threads execute stored procedures or SQL statements requested by the clients.

Typically, SQL Server is used in a client/server environment where clients on other computers connect to the server and either issue SQL requests or, more commonly, run stored procedures typically written in the Transact-SQL language. Clients may also be colocated at the server node. SQL Server uses a built-in transaction processing (TP) monitor facility, Open Data Services, to support large numbers of clients. In practice, this configuration has scaled to 5,000 concurrent clients. Beyond that size, it makes sense to either partition the application into a cluster of nodes, or use a Web server or TP monitor to connect clients to SQL server. Common TP monitors such as CICS, Tuxedo, and Top End have been ported to Windows and interface with SQL Server. Increasingly, applications are being built using Internet Information Services (IIS) and Active Server Pages (ASPs) for presentation. These ASPs use Microsoft Visual Basic® Scripting Edition or JavaScript to invoke business-logic COM objects that in turn call SQL Server using the Active Data Objects (ADO) interface. An object request broker and distributed transaction manager are built into Windows.



Cluster Transparency

Cluster transparency allows applications to access data and objects anywhere in the cluster as though the data were local; data can move from one partition to another without affecting behavior of the application programs. Being able to add nodes to the system and move data to those new nodes without changing the applications, makes transparency the key to modular growth. Transparency is also key to high availability, allowing data to fail over from one node to another when the first node fails.

Distributed Systems Techniques

Distributed systems techniques are the key to building transparency in clusters. By structuring applications and systems as modules that interact via remote procedure calls, applications become more modular, and they can be distributed among nodes of the cluster.

The client calls upon a service by name. The procedure call either invokes the service locally or uses a remote procedure call if the service is remote.

Microsoft has invested heavily in structuring its software as components that interact via remote procedure calls. The resulting infrastructure is variously called OLE (object linking and embedding), COM (component object model), DCOM (distributed COM), Microsoft ActiveX® (the Internet-centric extensions of COM), and most recently COM+. Many aspects of COM+ are in place today and more are coming soon. In particular, with Windows and SQL Server 2000, Microsoft delivers the following:

- COM+ is a core part of Windows Server. It allows any object to be safely and efficiently invoked, so that one program can invoke other programs running anywhere in the network. It combines the features of a transaction manger with the features of an object request broker and a TP monitor. It is the core of Microsoft's distributed object mechanisms.
- Distributed transactions allow applications to do work at many SQL Server database partitions, as well as other resource mangers, and automatically and transparently get the ACID distributed-transaction semantics.
- OLE DB allows SQL Server and other data integrators to access data from any data source. OLE DB interfaces are being built for almost all data sources. Most Microsoft data storage components have an OLE DB interface (for example, Exchange Server, Active Directory, Word, and Excel), and OLE DB interfaces are rapidly appearing for legacy data stores like VSAM and RMS.
- Database Management Objects (DMO) is a COM externalization of all the management interfaces to SQL Server. Using DMO, customers and ISVs can build tools to manage local and remote computers running SQL Server.
- Windows also includes a reliable queuing mechanism, Message Queuing, that allows applications to issue deferred invocations. These queues work for disconnected nodes as well, allowing them to submit work that will be processed when the node reconnects to the main network.
- SQL Server 2000 now includes a notification service that provides a robust and scalable way to signal clients when the state of the database changes. This can be used to manage stock levels, delivery schedules, and other situations where timely action is required when the database state changes.

Windows already supports these features along with many other cluster facilities including cluster security (domains), cluster software management (System Management Server), cluster naming (Distributed Name Service and Active Directory), and cluster performance monitoring (System Monitor). SQL Server complements these facilities with management tools built into Enterprise Manger that allow it to manage and monitor an array of servers running SQL Server.

A key goal of Windows and SQL Server is to make the cluster as easy to manage and use as a single large system. Windows and SQL Server have facilities to manage and balance application servers across a cluster of many nodes. Active Directory tracks objects and gives clients transparent access to them. Windows clusters have a common console and a simple management model for all the components. Clusters have the added benefit that fault tolerance masks many system faults and provides highly available services.

Distributed Partitioned Views

SQL Server allows users to partition a table into many member tables, and then create a partitioned view over the set of member tables. Such views are called distributed partition views. Each member table can be stored at a different node of the cluster. If the view definition is replicated at each cluster node, applications can access the whole table from any node as though the table were stored locally.

Distributed partitioned views are created as follows. First, the table is partitioned on the prefix of the primary key field. These subtables, called member tables, all have the same schema as the original table, except that they have an additional integrity constraint on the partitioning attributes that limit the range of values of the partitioning field (for example, `customer_ID between 1000000 and 2000000`). A union view is created that combines all these tables. Then, at each of the members of the cluster, the administrator performs these steps:

1. The administrator defines the member table and populates it with the data for that member table.
2. The administrator defines a link to all the other members of the federation, so that SQL Server 2000 can access the other nodes as part of its distributed query processing.
3. The administrator defines the distributed partitioned view as a union of the member tables.

Thereafter, the applications can efficiently access the distributed partitioned view and the underlying member tables as though all the data were local. The SQL Server 2000 query optimizer and Microsoft Distributed Transaction Coordinator (MS DTC) assure that programs execute efficiently, and that they get ACID properties.

Partitioned Data and Data Pipes

SQL Server has always allowed customers to partition their databases and applications among instances of SQL Server running on several nodes. Clients connect to an application at one of the servers. If a client requests access to data at another node, the application can either access the data through Transact-SQL, or it can make a remote procedure call to the instance of SQL Server at the other node.

For example, each warehouse of a distributed application might store all the local orders, invoices, and inventory. When an item is backordered or a new shipment arrives from the factory, the local system has to perform transactions that involve both the warehouse and the factory nodes. In this case, the application running on a server at the warehouse can access the factory data directly, or it can invoke a stored procedure at the factory server. MS DTC and SQL Server automatically manage the data integrity between the factory and the warehouse.

After data and applications are partitioned among multiple servers in a cluster, there needs to be a convenient and high-performance way to move data among these servers. Data pipes make it easy to ship data between servers by capturing result sets returned by remote procedure calls directly in node-local tables. This approach can be used by many applications as an alternative to distributed query.

Distributed Transactions

Distributed transactions are an integral part of Windows Server and one more step toward a full Windows Server cluster facility. To create a distributed transaction, applications declare BEGIN DISTRIBUTED TRANSACTION, and MS DTC then manages the transaction automatically.

MS DTC also connects SQL Server to the open transaction standard X/Open XA. Clients can connect to TP monitors like CICS, Encina, and Tuxedo, which in turn route requests to the servers. The use of TP monitors to route transactions to the appropriate servers is another approach to distributing the application. A TP monitor also allows SQL Server to participate in transactions distributed across many nodes.

Lastly, Windows Server 2003 includes Microsoft COM+, an embedded transaction monitor. COM+ dispatches client requests to application servers.

All these approaches make it relatively easy to partition data and applications among servers in a cluster.

Transparent Partitioning and Parallel Database Techniques

Everything described so far exists; you can acquire and use it today. Indeed, many customers are installing SQL Server on their servers, scaling up to 8-way SMP hardware systems, and then scaling out beyond that by partitioning their databases and applications. Often, these partitions are placed close to the actual data users: data collection is placed in the retail outlets, a data mart of recent activity is placed in the accounting group, and a data warehouse is placed in the planning and marketing group. Each of these applications is fairly separate and partitions naturally. In addition, the data flows among the groups are well-defined. Replication and data pipes make it easy to move data among servers. The systems are very scalable. The graphical and operations interfaces, combined with Microsoft Visual Basic® scripting, are used to automate the operation of many computers running SQL Server.

Over the next few years, Microsoft expects to add transparent partitioning of data among computers running SQL Server, allowing partitioned data without requiring that the application be made aware of the partitioning.

After partition transparency is in place, the SQL Server group expects to provide parallel query decomposition. That is, large data queries typical of decision-support applications will be decomposed into components that can be independently executed by multiple nodes of a partitioned database.

Data Partitioning Example: TPC-C and 1 Billion Transactions per Day

Distributed partitioned views in SQL Server make possible the very high TPC-C performance discussed in the previous sections, almost twice the peak performance of any other DBMS. The great part about that design is that it can be grown to manage twice as much data and process even more transactions by just adding more nodes and more federated SQL Servers to the cluster.

Microsoft, Intel, and Compaq cooperated to build a large (140-CPU, 45-node, 4-terabyte) system running the DebitCredit transaction in a classic three-tier DCOM+ SQL Server 7.0

application. Twenty front-end nodes simulated 160,000 connected users submitting transactions at the rate of 14,000 transactions per second. Twenty server nodes each stored equal portions of the database in an instance of SQL Server. That system did not use distributed partitioned views; rather, the application had to manage the data partitioning manually. The client nodes simulated the network and made DCOM calls to objects on the client that in turn made ODBC calls to the servers. The servers collectively stored 1.6 billion account records and were sized for 30 billion history records. The servers ran 85 percent of the transactions locally, and 15 percent of the transactions were remote. MS DTC coordinated the distributed transactions. Five nodes were dedicated to this function.

The system was run continuously at the Microsoft Executive Briefing Center for a year, and processed about 1.1 billion transactions each day, approximately five times more transactions than the travel industry, 10 times more than all the credit card transactions in a day, and one thousand times larger than the largest stock exchange. Today, we would do that application with partitioned distributed views on SQL Server 2000.

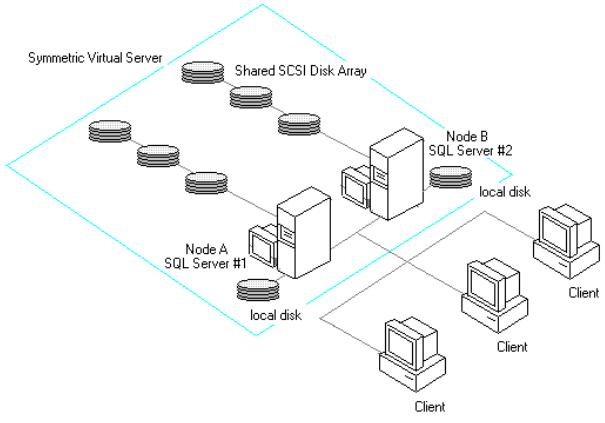
High-Availability Databases with Microsoft Cluster Service

SQL Server Enterprise Edition offers fault tolerance and high availability by providing failover from one server to another when the first server fails or needs to be taken out of service for maintenance. The failover mechanism works as follows. Two Windows Server 2003 servers are configured as a cluster. These two servers can support two instances of SQL Server. Each server manages a partition of the application database. So far, this is just standard SQL Server technology.

With SQL Server Enterprise Edition, each server can be made a virtual server that continues to offer service, even if the underlying node is taken offline by a fault or for system maintenance. To achieve this, the SQL Server databases are placed on shared SCSI disks accessible to both servers. If one server fails, the other server takes ownership of the disks and restarts the failed server on the surviving node. The restarted server recovers the database and begins accepting client connections. For their part, clients reconnect to the virtual server when the primary server fails. A Windows 2000 Server feature, Microsoft Cluster Service, allows the virtual server name and IP address to migrate among the nodes, so that clients are unaware that the server has moved. Microsoft Cluster Service is available in Windows 2000 Server, as well as Windows Server 2003 Enterprise Edition and Windows Server 2003 Datacenter Server.

SQL Server Enterprise Edition provides native setup support to set up virtual servers. After configuration, the virtual servers look just like any other server in the network, except that they can tolerate hardware failures.

SQL Server failover is completely automatic. Detecting and recovering from failure takes only a few minutes. When the failed node is repaired, it is restarted and it becomes the new backup server. If desired, the node can fall back to the original server when it is repaired.



Data Replication for Data Marts and Disaster Recovery

Data replication helps configure and manage partitioned applications. Many applications naturally partition into disjoint sections. For example, hotel, retail, and warehouse systems have strong geographic locality. The applications and databases can accordingly be broken into servers for geographic areas. Similarly, customer care, sales force automation, and telemarketing applications often have strong partitioning. Nonetheless, all these applications need some shared global data. They also need periodic reporting and disaster recovery via electronic vaulting.

Data replication helps solve these problems by propagating changes automatically from one database to another. Replication can propagate changes to a SQL Server system at a remote site for disaster recovery. Then, if the primary site fails, the backup site can recover and offer service.

The same mechanism can be used to allow one site to act as a data warehouse for data-capture OLTP systems. The data warehouse in turn may publish its data to many data marts that provide decision support data to analysts. Some applications dispense with the data warehouse and have the operational systems publish updates to the data marts directly.

SQL Server has a data replication system that is both powerful and simple to use. A graphical user interface in Enterprise Manager allows the administrator to tell operational databases to publish their updates, and allows other nodes to subscribe to these updates. This publish-distribute-subscribe metaphor allows one-to-one and one-to-many publications. Cascading distributors can scale the replication mechanism to huge numbers of subscribers. Replication is in transaction units, so each subscriber sees the database from a consistent point in time.

SQL Server applications routinely publish tens of megabytes of updates per hour. Publication can be immediate, periodic, or on demand. Replication is fully automatic and easy to administer.

SQL Server and Windows Server 2003 Manageability

Because Microsoft provides easy installation of the operating system and database using graphical tools and wizards, it is relatively easy to build huge systems with SQL Server. SQL Server also includes wizards to set up operational procedures.

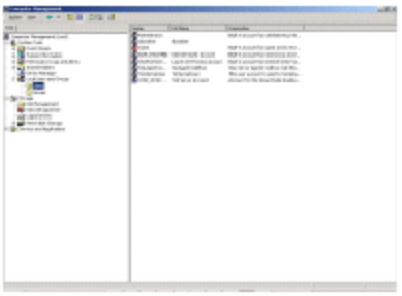
These huge systems involve thousands of client systems, and huge databases; therefore, manageability is the challenge. Loading, dumping, and reorganizing 100-GB databases, at the 3 MB/sec data rate of most tape drives, takes 10 hours with one tape drive. Defining and managing the security attributes of 10,000 different users is also a daunting task. Configuring the hardware and software for 10,000 clients is another time-consuming task.

Microsoft recognizes that manageability is the largest barrier to scalability. Microsoft's solution to these problems is described in the product documentation. This section summarizes the administrative facilities of Windows Server 2003 and SQL Server 2000.

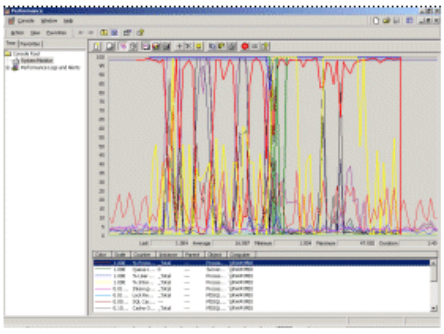
Scalable Windows Server 2003 Management

Managing the software and hardware configurations of thousands of clients is probably the most challenging task of operating large client-server systems. Windows Server 2003 and Microsoft System Management Server automate many of these tasks.

First, Windows security provides a domain concept and a single logon for each application running on Windows. Windows security provides user groups. Large user populations can be managed by authorizing groups and adding users to groups. The Windows 2000 Server security mechanism works as a collection of security servers (domain controllers) spread among the network nodes. This distribution provides both scalability and availability. Individual domains have been scaled to more than 40,000 users. Windows security scales beyond that size by partitioning into a multidomain architecture with trust relationships among domains. The security system has both a programmatic and an intuitive graphical interface that allows any node to manage the network security.



Second, Microsoft System Management Server allows a single administrator to manage the software configuration, licensing, and upgrades of tens of thousands of clients. System Management Server automates most tasks, and tries to minimize exception situations that only an expert can solve. In addition, the Windows DHCP protocol automatically assigns TCP/IP addresses to nodes on demand, thereby eliminating time-consuming and error prone tasks, allowing node mobility, and conserving the address pool.



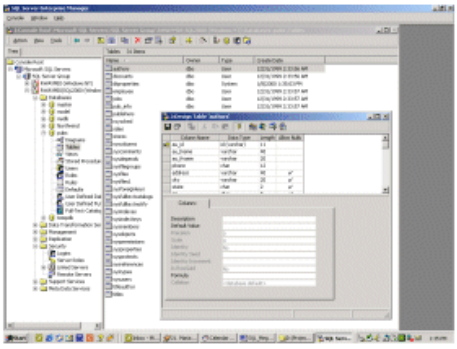
Finally, Windows Server 2003 has built-in tools to log errors, manage disk space, set priorities, and monitor system performance. All these tools can manage a cluster of client and server nodes. The previous illustration shows System Monitor at one node tracking the

CPU and network utilization of several nodes. Each Windows node has more than 500 performance counters for its internal behavior. SQL Server, Microsoft Exchange, and many other products ported to Windows integrate with System Monitor. Indeed, SQL Server adds more than 75 performance counters and integrates with the Windows event log to announce events.

Scalable SQL Server Management

SQL Server Enterprise Manager is a breakthrough in managing database servers. It gives administrators a visual way to manage and operate many SQL systems from a single console. The key features of SQL Server Enterprise Manager are:

- A graphical administration interface to control and monitor the operation of many servers and the clients accessing them.
- A job scheduler that runs periodic administrative tasks such as dumps, reorganizations, and integrity checks.
- A Data Management Objects mechanism that allows administrators to automate exception handling and to automate tasks, either by writing Visual Basic scripts or letting a wizard write the script. These procedures can use e-mail and a Telephony API (TAPI)-based beeper system to report results or notify operators.
- An extension mechanism that allows third parties to add new administrative tools.
- A fully graphical interface to configure and manage database replication.
- Integration with Active Directory, which registers servers and databases by name in a location-transparent way.



SQL Server Enterprise Manager also includes wizards to set up routine operations. It provides a wizard to set up automatic backup, reorganization, and operations tasks. It provides another wizard to publish Web pages routinely from the database to the Internet or to an intranet. It also provides wizards to help set up data replication. Utilities to load, backup, recover, check, and reorganize large databases are key to operating a system with huge databases. Backing up a multiterabyte database using a single high-performance tape drive will take several hours, or even days. By using multiple disks and tapes in parallel, SQL Server and Windows 2000 Server have shown multiterabyte rates: 2.6 terabytes/hour online backup with a 26 percent additional CPU load, and 2.2 terabytes/hour online restore with a 12 percent additional CPU load. Backup and restore from disk is even faster.

The SQL Server Enterprise Manager job scheduler, using commodity tape robots, orchestrates the backup/restore process. Backups can be done either at full speed or in the background at a slower rate. By doing incremental backups and by increasing the degree of parallelism, huge databases can be backed up in a few hours. Using the shadow-copy technology of Windows Server 2003, backups and restores of multiterabyte databases can be done in minutes.

Summary

Windows Server 2003 and SQL Server scale up to huge databases on a single SMP server and scale out to multiple servers each executing a part of the application and storing a partition of the database. SQL Server Enterprise Manager makes it easy to configure and manage these servers. OLE transactions, replication, and data pipes make it easy to move data and requests among them.

Today, a single instance of SQL Server can support more than 250,000 active users connected to a single server via IIS and ADO/ODBC. These servers can process several million transactions in an 8-hour day. They support databases with billions of records spread across a hundred disks holding terabytes of data.

A Windows Server 2003 cluster of a 32 such servers can process well in excess of 700,000 business transactions per minute—more than a billion transactions per day—against a database of more than 50 terabytes. By using clustering, there is virtually no limit to the size and throughput of a SQL Server 2000 system.

The addition of automatic failover, COM+, .NET XML Web services, and large main memory Itanium processors are the most recent steps toward Windows Server 2003 clusters running SQL Server that can scale both up and out.

SQL Server performs these tasks with unrivaled price/performance and ease of use. SQL Server 2000 has the best performance of any product on the standard TPC-C benchmark. Additionally, the performance of SQL Server, Windows Server 2003, and the underlying hardware has more than doubled each year for the last three years. Microsoft expects this trend to continue for several more years.

SQL Server has impressive scalability today. Even more impressive performance is expected in future releases.

For more information about SQL Server, see http://www.microsoft.com/sql, or visit the SQL Server Forum on the Microsoft Network (GO WORD: MSSQL).