SQL Server 2008 Performance and Scale

White Paper

Published: February 2008

Updated: July 2008

Summary: Microsoft SQL Server 2008 incorporates the tools and technologies that are necessary to implement relational databases, reporting systems, and data warehouses of enterprise scale, and provides optimal performance and responsiveness. With SQL Server 2008, you can take advantage of the latest hardware technologies while scaling up your servers to support server consolidation. SQL Server 2008 also enables you to scale out your largest data solutions.

For the latest information, see Microsoft SQL Server 2008.

Contents

Introduction

Today’s organizations need easily accessible and readily available business data so that they can compete in the global marketplace. In response to this need, relational and analytical databases continue to grow in size, embedded databases ship with many products, and many companies consolidate servers to ease management concerns. Companies must maintain optimal performance while their data environment continues to grow in size and complexity.

This white paper describes the performance and scalability capabilities of Microsoft® SQL Server® 2008 and explains how you can use these capabilities to:

- Optimize performance for any size of database with the tools and features that are available for the database engine, analysis services, reporting services, and integration services.
- Scale up your servers to take full advantage of new hardware capabilities.

Scale out your database environment to optimize responsiveness and to move your data closer to your users.

Optimizing Performance with SQL Server 2008

Because your corporate data continues to grow in size and complexity, you must take steps to provide optimal data access times. SQL Server 2008 includes many features and enhancements to optimize performance across all of its areas of functionality, including relational Online Transaction Processing (OLTP) databases; Online Analytical Processing (OLAP) databases; reporting; and data extract, transform, and load (ETL) processes.

Relational Database Performance

In most business environments, relational databases are at the core of business-critical applications and services. As volumes of data increase, and the number of users and applications that are dependent on relational data-stores grows, organizations must be able to ensure consistent performance and responsiveness from their data systems. SQL Server 2008 provides a robust database engine that supports large relational databases and complex query processing.

Measurable, Real-World Performance

SQL Server 2008 builds on the industry-leading performance of previous versions of SQL Server to provide the highest possible standard of database performance to your organization. Having demonstrated the high performance capabilities of SQL Server in the past with the Transaction Processing

Performance Council's TPC-C benchmark, Microsoft was the first database vendor to publish results for the newer TCP-E benchmark, which represents more accurately the kinds of OLTP workloads that are common in modern organizations.

Additionally, SQL Server demonstrates its performance capabilities for large-scale, data warehousing workloads through TPC-H results in the 3-terabyte and 10-terabyte categories. (For current benchmark results, see the TPC Web site at www.tpc.org.)

High Performance Query Processing Engine

The high performance query processing engine of SQL Server helps users to maximize their application performance. The query processing engine evaluates queries and generates optimal query execution plans that are based on dynamically maintained statistics about indexes, key selectivity, and data volumes. You can lock these query plans in SQL Server 2008 to ensure consistent performance for commonly executed queries. The query processing engine can also take advantage of multi-core or multi-processor systems and generate execution plans that take advantage of parallelism to further increase performance.

Usually, the most costly operation in terms of query performance is disk I/O. The dynamic caching capabilities of SQL Server reduce the amount of physical disk access that is required to retrieve and modify data, and the query processing engine can significantly improve overall performance by using read-ahead scans to anticipate the data pages that are required for a given execution plan and preemptively read them into the cache. Additionally, the SQL Server 2008 native support for data compression can reduce the number of data pages that must be read, which improves performance on I/O-bound workloads.

SQL Server 2008 supports partitioning of tables and indexes, which enables administrators to control the physical placement of data by assigning partitions from the same table or index to multiple file groups on separate physical storage devices. Optimizations to the query processing engine in SQL Server 2008 enable it to parallelize access to partitioned data, which significantly enhances performance.

Performance Optimization Tools

SQL Server 2008 includes SQL Server Profiler and the Database Engine Tuning Advisor. By using SQL Server Profiler you can capture a trace of the events that occur in a typical workload for your application, and then replay that trace in the Database Engine Tuning Advisor, which generates and implements recommendations for indexing and partitioning of your data, so you can optimize the performance of your application.

After creating the indexes and partitions that best suit the workload of your application, you can use the SQL Server Agent to schedule an automated database maintenance plan. The automated maintenance periodically reorganizes or rebuilds indexes, and updates index and selectivity statistics, to ensure consistently optimized performance as data inserts and modifications fragment the physical data pages of your database.

Resource Governor

Often, a single server is used to provide multiple data services. In some cases, many applications and workloads rely on the same data source. As the current trend for server consolidation continues, it can be difficult to provide predictable performance for a given workload because other workloads on the same server compete for system resources. With multiple workloads on a single server, administrators must avoid problems such as a runaway query that starves another workload of system resources, or low-priority workloads that adversely affect high-priority workloads. SQL Server 2008 includes Resource Governor, which enables administrators to define limits and assign priorities to individual workloads that are running on a SQL Server instance. Workloads are based on factors such as users, applications, and databases. By defining limits on resources, administrators can minimize the possibility of runaway queries as well as limit the resources that are available to workloads that monopolize resources. By setting priorities, administrators can optimize the performance of a mission-critical process while maintaining predictability for the other workloads on the server.

Performance Studio

SQL Server 2008 provides Performance Studio, an integrated framework that you can use to collect, analyze, troubleshoot, and store SQL Server diagnostics information. Performance Studio provides an end-to-end solution for performance monitoring that includes low overhead collection, centralized storage, and analytical reporting of performance data. You can use SQL Server Management Studio to manage collection tasks, such as enabling the data collector, starting a collection set, and viewing system collection set reports as a performance dashboard. You can also use system stored procedures and the Performance Studio application programming interface (API) to build your own performance management utilities based on Performance Studio.

Performance Studio provides a unified data collection infrastructure that consists of a data collector in each SQL Server instance you want to monitor. The data collector is flexible and provides the ability to manage the scope of data collection to fit development, test, and production environments. You can easily collect both performance and general diagnostic data with the data collection framework.

The data collector infrastructure introduces the following new concepts and definitions:

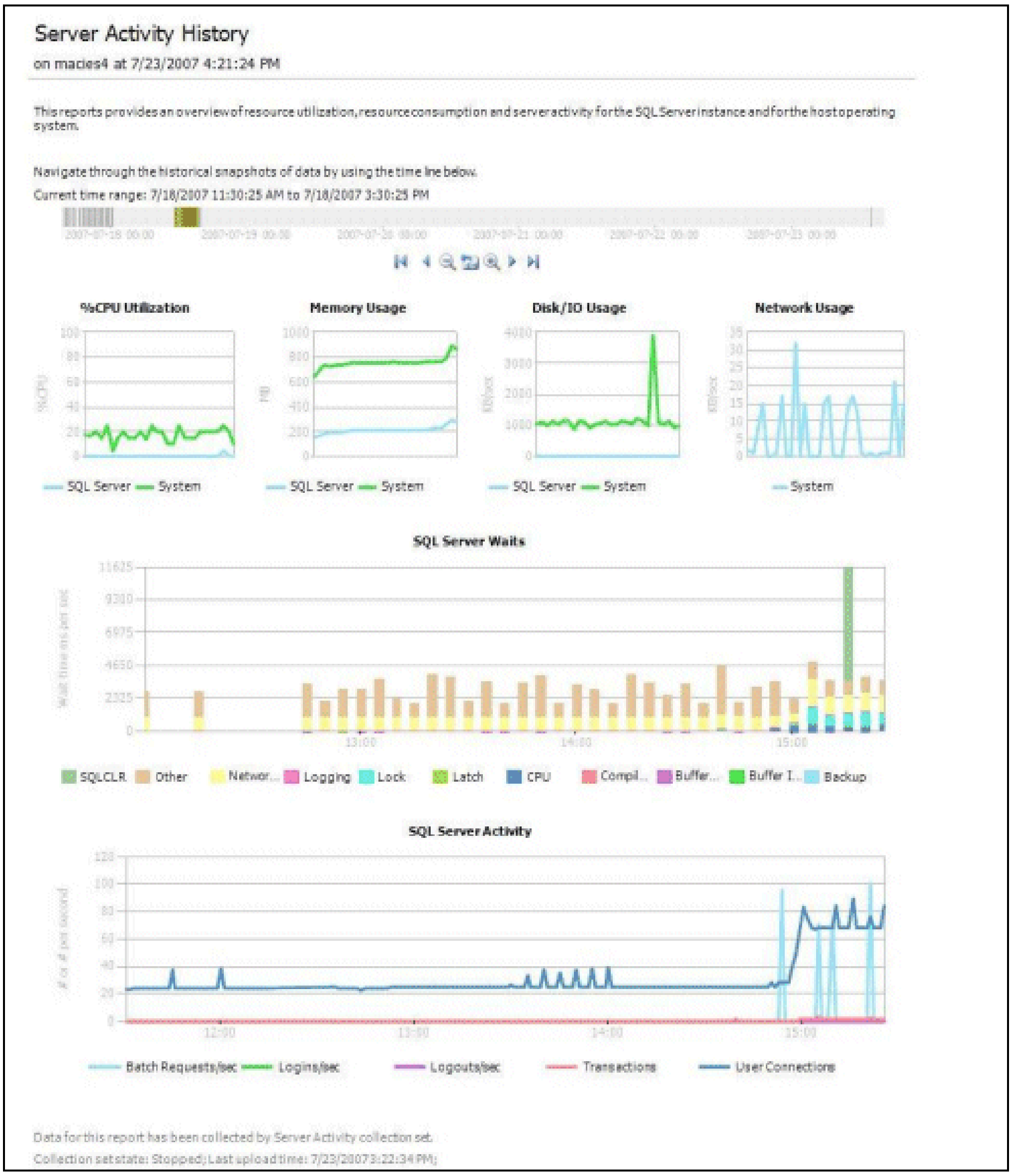
- **Data Provider**. Sources of performance or diagnostic information that can include SQL Trace, performance counters, and Transact-SQL queries (for example, to retrieve data from distributed management views).
- **Collector Type**. A logical wrapper that provides the mechanism for collecting the data from the data provider.
- **Collection Item**. An instance of a collector type. When you create a collection item, you define the input properties and collection frequency for the item. A collection item cannot exist on its own.
- **Collection Set**. The basic unit of data collection. A collection set is a group of collection items that are defined and deployed on a SQL Server instance. Collection sets can run independently of each other.
- **Collection Mode**. The manner in which the data in a collection set is collected and stored. The collection mode can be set to cached or non-cached. The collection mode affects the type of jobs and schedules that exist for the collection set.

The data collector is extensible and supports the addition of new data providers.

When the data collector is configured, a relational database with the default name **MDW** is created as a management data warehouse in which to store the collected data. This database can reside on the same system as the data collector or on a separate server. Objects in the management data warehouse are grouped into the following three preconfigured schemas, each of which has a different purpose:

- The **Core** schema includes tables and stored procedures for organizing and identifying the collected date.
- The **Snapshot** schema includes data tables, views, and other objects to support the data collected from the standard collector types.
- The **Custom_Snapshot** schema enables the creation of new data tables to support user-defined collection sets that are created from standard and extended collector types.

Performance Studio provides a robust set of preconfigured system collection sets, including **Server Activity**, **Query Statistics**, and **Disk Usage**, to help you to quickly analyze your collected data. You usually start your monitoring and troubleshooting with the **Server Activity** system collection set. A set of reports associated with each system collection set is published in SQL Server Management Studio, and you can use these reports as a performance dashboard to help you to analyze the performance of your database systems as shown in the following figure.



Server Activity History

on macies4 at 7/23/2007 4:21:24 PM

This reports provides an overview of resource utilization, resource consumption and server activity for the SQL Server instance and for the host operating system.

Navigate through the historical snapshots of data by using the time line below.

Current time range: 7/18/2007 11:30:25 AM to 7/18/2007 3:30:25 PM

Data for this report has been collected by Server Activity collection set.

Collection set state: Stopped; Last upload time: 7/23/20073:22:34 PM;

Figure 1: A Performance Studio report

Data Warehousing and Analysis Performance

Data warehouse environments must keep up with growing volumes of data and user requirements and maintain optimal performance. As data warehouse queries become more complex, each part of the query must be optimized to maintain acceptable performance. In SQL Server 2008, the query optimizer can dynamically introduce an optimized bitmap filter to enhance query performance for star join queries.

Analysis Services applications typically require large and complex computations. Precious processor time is wasted by computing aggregations that resolve to NULL or zero. Block computations in SQL Server 2008 Analysis

Services use default values, minimize the number of expressions that must be computed, and limit cell navigation to once for the entire space, rather than once for each cell, which significantly improves computation performance.

Although Multidimensional OLAP (MOLAP) partitions provide greater query performance, organizations that require write-back capabilities were previously required to use Relational OLAP (ROLAP) partitions to maintain the write-back tables. SQL Server 2008 adds the ability to perform write-back operations to MOLAP partitions, which removes the performance degradation that is caused by maintaining ROLAP write-back tables.

Reporting Services Performance

The SQL Server 2008 Reporting Services engine has been re-engineered to add greater performance and scalability to Reporting Services with on-demand processing. Reports are no longer memory bound because report processing now uses a file system cache to adapt to memory pressure. Report processing can also adapt to other processes that consume memory.

A new rendering architecture removes memory usage problems from previous versions of renderers. These new renderers also provide improvements, such as a true data renderer added to the CSV renderer, and support for nested data regions and nested sub-reports in the Microsoft Office Excel® renderer.

Integration Services Performance

ETL processes are frequently used to populate and update data in data warehouses from business data in source databases throughout the enterprise. Traditionally, many companies required only historical data with infrequent data refreshes to the data warehouse. Now, many organizations want near real-time data to be available through the data warehouse. As greater amounts of data and more frequent data warehouse refreshes are required, the ETL process time and flexibility becomes more important.

Data refreshes require SQL Server Integration Services to use lookups to compare source rows to data that is already in the data warehouse. Integration Services includes greatly improved lookup performance that decreases package run-times and optimizes ETL operations. As well, in SQL Server 2008 SQL Server Integration Services, several threads can work together to do the work that a single thread is forced to do in SQL Server 2005 SQL Server Integration Services. This can give you a several-fold speedup in ETL performance.

Another problem with traditional ETL processes is determining what data has changed in the source database. Administrators had to be extremely careful to avoid duplication of existing data. Some administrators chose to remove all of the data values and reload the data warehouse rather than manage data that had been changed. This added a great deal of overhead to the ETL process. SQL Server 2008 includes *change data capture* functionality to log updates to change tables, which helps to track data changes and ensure consistency in the data warehouse when data refreshes are scheduled.

Scaling Up with SQL Server 2008

Server consolidation, large data stores, and complex queries require physical resources to support the various workloads running on a server. SQL Server 2008 has the capability to take full advantage of the latest hardware technologies. Multiple database engine instances and multiple analysis services instances can be installed on a single server to consolidate hardware usage. As many as 50 instances can be installed on a single server without compromising performance or responsiveness.

Hardware Support

SQL Server 2008 takes full advantage of modern hardware including 64-bit, multi-core, and multi-processor systems. To support increased reporting, analytical, and data access loads, SQL Server can address up to 64 GB of memory and supports dynamic allocation of AWE-mapped memory on 32-bit hardware, and can address up to 8 terabytes of memory on 64-bit hardware.

When a large number of processors are added to a server, memory access can be slowed down if processors must access memory that is not local to the processor. Hardware built to the non-uniform memory access (NUMA) architecture overcomes these memory access limitations by enabling processors to access local memory. SQL Server is aware of NUMA hardware, so provides companies with greater scalability and more performance options. You can take advantage of NUMA-based computers without application configuration changes. SQL Server 2008 supports both hardware NUMA and soft-NUMA.

Hot-Add Hardware

Although you can easily scale up a SQL Server instance by adding memory or CPUs, scheduling downtime to add hardware to scale up your mission critical applications and twenty-four-hour-a-day, seven-day-a-week operations can be difficult. With SQL Server 2008, you can scale up your server by adding CPUs and memory to compatible machines without having to stop your database services.

The following requirements must be met to hot-add memory:

- SQL Server 2008 Enterprise
- Windows Server® 2003 Enterprise Edition or Windows Server 2003 Datacenter Edition
- 64-bit SQL Server or 32-bit SQL Server with AWE support enabled
- Hardware from your hardware vendor that supports memory addition, or virtualization software
- SQL Server started with the –h option

The following requirements must be met to hot-add CPUs:

- SQL Server 2008 Enterprise
- Windows Server® 2008 Enterprise Edition for Itanium Systems or Windows Server 2008 Datacenter Edition for x64 bit systems
- 64-bit SQL Server
- Hardware that supports CPU additions, or virtualization software

Advanced Concurrency Features

The purpose of scaling up your database server is to support increasing numbers of users or applications. As the number of users increases, responsiveness can be affected by concurrency issues when multiple transactions attempt to access the same data. SQL Server 2008 provides numerous isolation levels to support a variety of solutions that balance concurrency with read integrity. For row-level versioning support, SQL Server 2008 includes a read committed isolation level that uses the READ_COMMITTED_SNAPSHOT database option and a snapshot isolation level that uses the ALLOW_SNAPSHOT_ISOLATION database option. Additionally, the Lock Escalation setting on a table enables you to improve performance and maintain concurrency, especially when querying partitioned tables.

Scaling Out with SQL Server 2008

In addition to scaling up individual servers to support growing data environments, SQL Server 2008 offers tools and capabilities to scale out databases to increase performance of very large databases and to move the data closer to the users.

Scalable Shared Databases

Data warehouses are typically used by multiple consumers of read-only data, such as analysis and reporting solutions, and can become overloaded with data requests, which reduces responsiveness. To overcome this issue, SQL Server 2008 supports scalable shared databases, which provide a way to scale out read-only reporting databases across multiple database server instances to distributes the query engine workload and isolate resource-intensive queries. The scalable shared database feature enables administrators to create a dedicated read-only data source by mounting copies of a read-only database on multiple reporting servers. Applications access a consistent copy of the data, independent of the reporting server to which they connect.

Data Dependent Routing

When a company decides to scale out its database structure into a federated database, it must determine how to divide the data logically between the servers and how to route requests to the appropriate server. With SQL Server 2008, you can implement data dependent routing as a service by using Service Broker to route queries to the appropriate locations.

Peer-to-Peer Replication

Peer-to-peer replication can provide an effective scale-out solution in which identical copies of a database are distributed to locations throughout the organization, so that modifications made to the local copy of the data are propagated automatically to the other replicated copies. SQL Server 2008 helps you to reduce the time taken to implement and manage a peer-to-peer replication solution with the new Peer-to-Peer Topology wizard and visual designer. By using peer-to-peer replication you can enable applications to read or modify data in any of the databases that are participating in replication. While previous versions of SQL Server required administrators to stop activity on published tables on all nodes before attaching a new node to an existing node, SQL Server 2008 enables new nodes to be added and connected, even during replication activity.

Query Notifications

Most enterprise applications are based on a three-tier architecture in which data is retrieved from the database server by one or more application servers (often a Web farm), which is in turn accessed by client computers. To improve performance, many application servers cache data to provide quicker response times to users. One limitation of cached data is the need to refresh the data, because if the data is not refreshed frequently enough, users can receive stale data that is no longer accurate. Refreshing data more frequently adds overhead which can ultimately slow down the performance on the application server. SQL Server 2008 helps applications to use application cache more efficiently by using query notifications to automatically notify middle tier applications when the cached data is outdated. The application server can subscribe to query notification so that it is informed when updates that affect the cached data are performed on the database. The application server can then dynamically refresh the cache with the updated data.

Scalable Shared Databases for Analysis Services

Although SQL Server 2005 Analysis Server cubes are usually read-only databases; each instance maintains its own data directory. Although you can create multiple copies of an Analysis Services database by synchronizing cubes across multiple servers, the cube synchronization process introduces latency that may be unacceptable in many business environments. SQL Server 2008 Analysis Services overcomes these issues by supporting a scale-out Analysis Services deployment in which a single, centralized read-only copy of the Analysis Services database is shared across multiple instances and accessed through a single virtual IP address as shown in Figure 2.

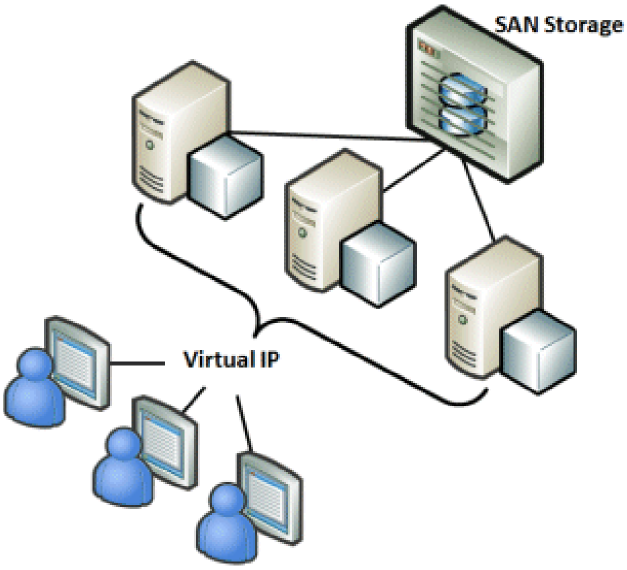


Figure 2: Scale-out Analysis Services

Conclusion

SQL Server 2008 provides a rich and powerful environment for all of your data needs. It is optimized to provide the best performance through its advanced relational database engine, Analysis Services, Reporting Services, and Integration Services. New and well-established tools and techniques work together to help companies to deploy any size of data environment and to maintain optimal performance. SQL Server 2008 supports the latest technology trends to give companies the flexibility to scale up individual servers and consolidate workloads onto these servers. It also provides companies with the technologies that are necessary to scale out the largest databases.

For more information:

Microsoft SQL Server 2008
http://www.microsoft.com/sqlserver/2008/en/us/default.aspx

SQL Server Developer Center
http://msdn2.microsoft.com/sqlserver

SQL Server TechCenter
http://technet.microsoft.com/sqlserver

Please give us your feedback:

Did this paper help you? Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you giving it a high rating because it has good examples, excellent screenshots, clear writing, or another reason?
- Are you giving it a low rating because it has poor examples, fuzzy screenshots, unclear writing?

This feedback will help us improve the quality of white papers we release. Send feedback.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.