Accelerated SQL Server 2008

Copyright © 2008 by Robert Walters

PART 1

■■■

Overview of SQL Server

SQL Server 2008 Overview

The previous release of SQL Server, SQL Server 2005, was a major release. It contained a ton of new functionality, including the revision of major query-processing components within the database engine. With SQL Server 2008, the development cycle was much shorter, and the changes are not quite as dramatic. Even though the overall quantity of changes is less than those in SQL Server 2005, the improvements that were made in this new release are specific and significant, and they will absolutely be of value to your organization.

In recent years, one of the trends has been the explosion of data. This massive increase in the quantity of data can be attributed to changes in behavior by consumers and businesses. For consumers, we need to look no further than digital cameras for a great example. With digital cameras, it is now possible to take a thousand different perspectives of your favorite landmark without worrying about film or development costs. All of these digital photos take up cheap disk space and need to be managed for easy searching. As another example, consider the amount of data generated by the medical industry in its daily operations. X-rays are being created and stored in a digital format for easier portability and quicker viewing time (doctors don't need to wait for someone to fetch a patient record from the vault).

The increase of data in recent years comes with a price, and it's not the cost of hardware, as some may think. Although the hardware industry has done a great job of providing low-dollar-per-gigabyte ratios year after year, the increase in digitally born data has sweetened the deal for potential data thieves. This increased security risk, combined with the influx of regulatory compliance laws, such as the Sarbanes-Oxley Act of 2002, has influenced the security features within SQL Server. Security design has been at the core of every feature implementation in SQL Server, and the SQL Server 2008 release is no different. Most of the security-specific improvements, such as auditing and transparent data encryption, are discussed in Chapters 6 and 7 of this book.

The Vision of SQL Server 2008

SQL Server 2008 is driven by a vision consisting of four key pillars: enterprise data platform, beyond relational, dynamic development, and pervasive insight. Although some might think of these catchy names as merely marketing hype, they actually meant something to the product development team.

Microsoft changed the way SQL Server is designed and developed. Specific user scenarios were derived from these core themes, and in the end, actual improvements to the product were made relating to those scenarios. The result is a high-quality release that focuses on the key pain points and industry trends. Here, we'll take a look at each of the pillars and discuss some key features within them. These features, as well as many others, are discussed in more detail in the subsequent chapters of this book.

Enterprise Data Platform

To be classified as an enterprise-ready database means much more than being able to formulate query results really fast. An enterprise database must meet the strict service level agreements established by the organizations using SQL Server. SQL Server 2008 has made improvements in supporting high service level agreements, like the ability to hot-add CPUs. Administrators will also find installing and managing the setup of cluster nodes to be much easier. More information about SQL Server setup and configuration can be found in Chapter 2.

Being an enterprise data platform also means the data that is stored inside the database is secure. SQL Server 2008 continually builds upon its security features. For example, it enables database files to be automatically encrypted with transparent data encryption. The importance of encryption is prominent with Microsoft, opening the door for Hardware Security Module (HSM) and Enterprise Key Management (EKM) vendors to integrate natively with the encryption support in SQL Server 2008. Encrypting data with SQL Server and storing the encrypted keys within the database provide security, but a more secure solution is to store the encryption keys separately from the actual data, and that is where HSM and EKM solutions add value.

Performance has always been a key attribute of an enterprise-ready database. The Data Collector is a feature within SQL Server that gives database administrators (DBAs) the ability to collect performance-related data and store it within a database. This data can be practically anything, such as Performance Monitor counters, results from database management views, and specific queries. Having performance data reside in a database allows for easy data mining and reporting, and that is the key benefit of the Data Collector. Details on this feature are discussed in Chapter 5.

SQL Server 2008 has many new features that validate SQL Server as an enterprise data platform. These features are discussed in detail throughout Part 2 of this book.

Beyond Relational

Data managed within a data platform is more than just relational data. As data growth increases, the types of data stored are no longer the traditional integer, character, and binary values we know and love. New data structures that are important to users are movie files, audio files, and medical images, to name a few. And we not only need to store these new types of data, but we also want to be able to perform useful operations on them, such as indexing and metadata searches.

With SQL Server 2008, investments were made in a feature called *filestream*, which allows files of arbitrary size to be stored in the file system and managed from the database. This capability enables database applications to exceed the 2GB limit. We can essentially place volume-sized binaries directly into SQL Server and obtain the same seek performance we would if we were querying the file system directly, instead of through Transact-SQL (T-SQL). Filestream is discussed in detail in Chapter 11.

SQL Server 2008 also includes support for spatial datatypes. The support conforms to the OpenGIS standards and allows for easy management of global positioning system (GPS) and geographic information system (GIS) data. Having native spatial support also makes it easy and fun to work with spatial applications like Microsoft's Virtual Earth. Spatial data support is also discussed in Chapter 11.

Dynamic Development

Without developer support, platform products such as SQL Server would have died off a long time ago. Microsoft has always made developer productivity a high priority in all of its products. SQL Server 2008 is no exception.

SQL Server 2008 contains improvements in the T-SQL language (discussed in Chapter 12), as well as new date and time datatypes (discussed in Chapter 11) to fill the void that the existing ones created. SQL Server has also integrated itself with the LINQ effort within Microsoft. LINQ provides

a higher level of data abstraction, making it really easy to code against disparate data sources. LINQ, as it's related to SQL Server, is discussed in detail in Chapter 18.

Pervasive Insight

Since the inception of Online Analytical Processing (OLAP) services in SQL Server 7.0, Microsoft has continually strived for a self-service business intelligence model. The idea is to allow the average employee to easily ask a business intelligence question and get the results, without needing to go through various layers of DBAs and report developers. Gradually, throughout the releases of SQL Server, we have seen more tools and features that promote this behavior.

At the core of business intelligence is SQL Server Analysis Services. New in Analysis Services are enhanced cube, dimension, and attribute designers. These designers, as well as core improvements related to the monitoring, analysis, and performance tuning of Analysis Services, continually push Microsoft's business intelligence engine further into the leader category of various industry analysts' charts. Analysis Services is discussed in Chapter 20.

In order for Analysis Services to effectively mine data, it needs a great Extract, Transform, and Load (ETL) tool. Investments with SQL Server Integration Services (the replacement for Data Transformation Services in SQL Server 2000) have continued, with the addition of capabilities such as caching transformations, enhanced lookup transformations, data profiling, and a set of expanded data sources. These exciting enhancements are discussed in Chapter 21.

Over the past few years, Microsoft has acquired a few companies in the reporting market. Some of these acquisitions, like Dundas and its graphical reporting controls, have shown up in Reporting Services in SQL Server 2008. The Reporting Services engine has also been upgraded to release its dependency on Internet Information Server (IIS), among other well-anticipated features. Chapter 19 covers Reporting Services in SQL Server 2008.

SQL Server 2008 Editions

At the time of this writing, the SQL Server 2008 editions are essentially the same as those that were available for SQL Server 2005. Five main SQL Server 2008 editions are available:

Enterprise Edition: This is primarily used for business-critical, large-scale online transaction processing (OLTP), large-scale reporting, data warehousing, and server consolidation requirements. Enterprise Edition comes with more than 60 features that are not found in Standard Edition. Some of these features are significant enough to entice those who have always said that Standard Edition was good enough. Features found only in Enterprise Edition are data and backup compression, audits that use extended events, and Resource Governor, to name a few. The gap of features between Standard Edition and Enterprise Edition is far greater in SQL Server 2008 than it was in SQL Server 2005.

Standard Edition: This edition is primarily used for departmental applications and small to medium-sized OLTP loads. Standard Edition comes with most of the powerful reporting capabilities found in SQL Server Reporting Services and makes a great reporting and analytics server as well.

Workgroup Edition: This edition includes the basic SQL Server relational database capabilities, as well as some of the replication technologies. This makes Workgroup Edition good for running branch office applications and performing remote synchronization with other geographically separated servers. Workgroup Edition is considerably less expensive than Standard Edition.

Express Edition: This is the free version of SQL Server. It's ideal for learning and building desktop and small server applications, and for redistribution. Although a lot of the Enterprise Edition functionality is intentionally disabled, the actual SQL Server runtime binary (`sqlservr.exe`) is created with the exact same code base as that of SQL Server Enterprise Edition (and all other editions, for that matter). This makes Express Edition a stable, high-performance database engine for a great price.

Compact Edition: This is the other free SQL Server version, designed to be an embedded database for applications. The ideal use case for Compact Edition is building stand-alone and occasionally connected applications for mobile devices, desktops, and clients.

■**Note** Developer and Evaluation Editions expose the same functionality as Enterprise Edition, but have special licensing restrictions. For example, Microsoft does not allow you to run Developer Edition on your production server.

A Word on Server Consolidation

Currently, many of us are working in companies where the latest cost-saving initiative is that of *server consolidation*. This new buzzword is getting a lot of attention lately. You may be happy to learn that SQL Server 2008 has some features that will help if your organization decides to consolidate.

If we look at why companies consolidate, we find two key motivations behind the effort:

Reduce costs: Consolidation reduces costs through reductions in software licensing fees, technical support costs, and, ultimately, hardware expenses. By using fewer servers, we will generate less heat and consume less electricity, and in the end, we are a bit friendlier to the environment.

Reduce server management complexity: Since SQL Server is a low-cost, enterprise-ready database platform, a lot of application vendors have included SQL Server as part of their solution. This trend, in addition to the proliferation of SQL Server in the enterprise as the organization's critical database, has led to SQL Server sprawl. When the sprawl gets out of hand, administrators do not know how many SQL Server implementations are actually deployed or where a specific database actually resides. Sprawl is much more difficult to administer, secure, and update. Server consolidation can reduce this sprawl.

Now that you are sold on server consolidation, realize that you should not implement server consolidation just for the sake of doing so. If done wrong, consolidation can make things worse. Some applications will not work well in a consolidated environment, and you must ensure that proper testing has been performed before moving consolidated servers into production.

When we talk about server consolidation and SQL Server, it is important to separate the physical hardware and operating system from the software aspect of a SQL Server implementation. From a physical aspect, depending on the hardware vendor, the server can be set up to host different operating systems by physically separating resources like memory and CPU utilization. Certain operating systems have native virtualization capabilities, like Windows Server 2008 and its Hyper-V technology. Hyper-V allows you to create separate virtual machines (VMs) running within a single physical machine. With Hyper-V, you can also efficiently run multiple different operating systems—Windows, Linux, and others—in parallel, on a single server. This exciting new technology in Windows Server 2008 does give VMWare a run for the money in the virtualization market. If you are interested in consolidating via server virtualization on Windows, check out the plethora of information found at `http://www.microsoft.com/virtualization/default.mspx`.

From a database application perspective, consolidation to SQL Server means either consolidating multiple databases on a single server instance or consolidating multiple server instances across multiple physical servers, but fewer than the number of servers the organization currently has.

Generally, in consolidation, we are reducing the number of physical servers and/or software licenses needed for our organization. What an organization actually ends up with as far as how many databases live within each instance and on how many instances live on each physical server is totally dependent on the organization's architecture and the requirements and restrictions for each database.

Consolidation may have some unwanted side effects. Consider the case where we are consolidating two server instances into one. If we have different users for each of these instances, we may have inadvertently given an elevation of privilege to some user accounts with this merger. Consider the case where the sysadmin of server instance 1 has no access to server instance 2, and these two instances are merged. The sysadmin of server instance 1 now has access to the contents that were contained on server instance 2. In security lingo, this is a prime example of an elevation of privilege. Given this example, it is important to emphasize that consolidation will not only put a strain on performance with respect to disk I/O, but it also may create some security issues if not properly designed.

Realize that the final architecture and design of consolidation is only one part of the overall consolidation effort. There are a few key issues that you may need to address, depending on where you fall in terms of responsibility in your organization. Perhaps one of the most important issues is that consolidation needs a buy-in from all parts of the organization. It needs an executive sponsor to help push the monetary considerations through the upper level of management. It also needs a buy-in from the actual workers who will be performing the consolidation. To some people, consolidation means potential job loss, and in some rare circumstances, this may be true. However, in the majority of cases, consolidation will give DBAs more time to work on the more challenging issues facing their organization. With fewer servers to manage, less time needs to be devoted to the mundane tasks of database maintenance.

In addition to buy-ins, if you are a manager, you need to make sure you allocate enough time for your resources. Proper planning and execution takes time, and most DBAs don't usually have any spare free time to dedicate to consolidation.

If you are interested in or are currently involved in a consolidation effort, take a look at the Resource Governor and Policy Management features in SQL Server 2008. Resource Governor, discussed in detail in Chapter 5, gives DBAs the ability to restrict users' or applications' use of SQL Server resources like memory and CPU utilization. Resource Governor was designed to prevent runaway queries from taking over the SQL Server instance, but it also has direct benefits in the SQL Server consolidation scenario.

Policy Management (PM) is another must-consider feature to use in a consolidation effort. Discussed in detail in Chapter 3, PM allows DBAs to define policies and enforce these policies against the SQL Server instances they are managing. For example, imagine a policy that would enforce the Resource Governor settings across all the servers in your organization. Suppose that this enforcement is necessary to maintain the service level agreement that your company's information technology department has with the organization. If another administrator wanted to tweak his server to gain more CPU bandwidth for his own queries, he would be denied. PM enables a plethora of scenarios, not just for consolidation, but in security, naming consistency, and many other areas. The time invested in learning PM is well worth it.

Summary

SQL Server 2008 contains a series of improvements that target scalability, performance, availability, security, and manageability. Keep in mind that some of the improvements mentioned in this book target and benefit multiple scenarios. Data compression is a prime example. This feature, which is discussed in Chapter 5, allows users to compress the storage of data within their tables. When using data compression in the business intelligence scenario, the primary benefit is disk-space savings,

since databases used for analytical processing tend to have many repeating and null values. Thus, this feature could have also been discussed in the chapter that covers Analysis Services (Chapter 20).

This book is based on the feature-complete February community technology preview (CTP) release of SQL Server 2008. As with any product that is not shipped, there may some minor changes in the user interfaces shown in this book's screenshots by the time the final version is released.