Microsoft SQL Server

Microsoft

Mission-Critical Performance

Technical White Paper

Published: May 2015

Applies to: SQL Server 2016 CTP2 and SQL Server 2014

Summary: As the volume and complexity of data continue to increase, organizations require a new approach to their mission-critical capabilities. In this white paper, we examine how capabilities built into Microsoft SQL Server—including enterprise-grade performance, security, availability, and scalability—define a “new mission critical” that answers what kinds of capabilities organizations need to compete in a dynamic global landscape. We also compare the cost impact of solutions offering mission-critical functionality built into the core database with those offering separate features that organizations can include for additional expense.

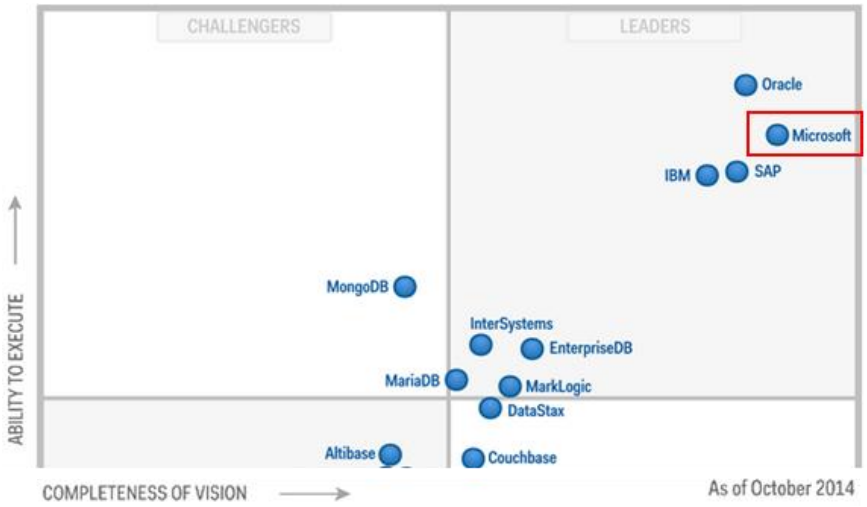
Copyright

Contents

SQL Server evolution

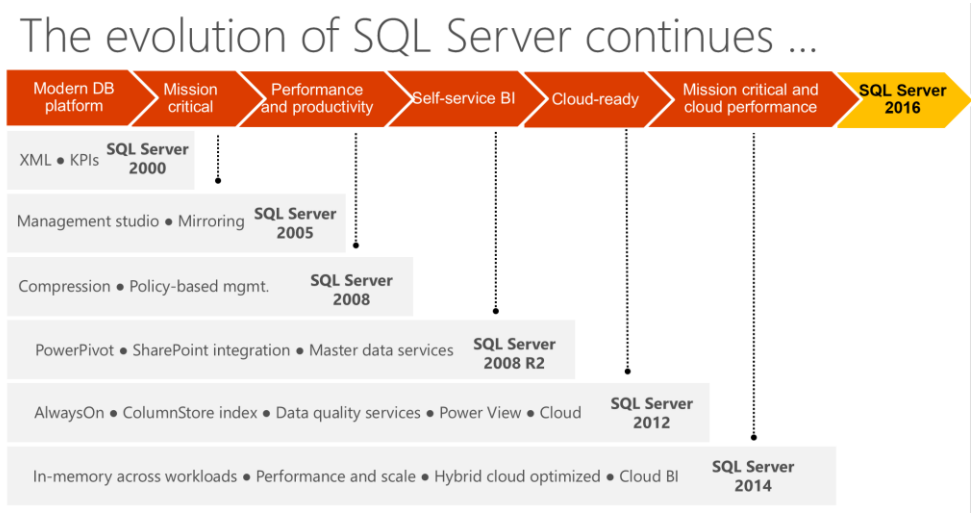
Microsoft SQL Server continues to evolve, developing and enhancing new features to stay ahead of organizational data needs. Customers have responded to this evolution by continuing to show confidence in using SQL Server to manage their mission-critical data. Industry analysts have also responded positively. Gartner recently rated SQL Server as having the most complete vision of any operational database management system (Figure 1).

Figure 1: Gartner Magic Quadrant for operational database management systems[1]



In addition, SQL Server has consistently added groundbreaking functionality over the last 15 years (Figure 2).

Figure 2: SQL Server functionalities added across releases



SQL Server 2016 introduces many new features and enhancements, including:

- Real-time insights on operational data and in-memory performance for more of your applications

[1] Gartner, http://www.gartner.com/technology/reprints.do?id=1-237UHKQ&ct=141016&st=sb, October 2014.

- Data security, from on-premises to cloud, with encryption at rest and in motion as well as greater compliance with row-level security
- Even higher availability with AlwaysOn, including more replicas, load balancing, and automation
- Enhanced database caching across multiple cores and enhanced Windows Server support

Mission-critical performance with SQL Server

SQL Server's evolution mirrors the growing critical nature of data for enterprises. Data is the new currency, and it has become a major competitive differentiator. Companies with a data-centric culture are harnessing increasingly diverse data. This data is not just relational or internal, but includes both relational and non-relational, as well as internal and external data sources. These companies are also using new analytical models to not just look at historical data but to use historical data to predict the future. Insights from the data are not limited to a few, but shared more broadly in the organization and much of this analysis is at speed: near real-time in many cases. Companies with a data-centric culture are more productive, more efficient with their operations, and innovate faster, leading to improvements in their bottom line.

More than ever, organizations need mission-critical operations that are easy to deploy and are balanced with faster time-to-solution. In this white paper, we review four main areas in which SQL Server continues to deliver in this regard: performance, security, availability, and scalability. We conclude with information about how to compare the total cost of ownership between SQL Server and other databases.

Performance

SQL Server consistently leads in performance benchmarks, such as TPC-E, TPC-H, and real-world application performance. With SQL Server 2014 and the upcoming SQL Server 2016, performance is enhanced with a number of new technologies, including in-memory, query store, JSON, and temporal support, to name a few.

SQL Server's integrated in-memory toolset goes far beyond isolated features and provides support for improving performance dramatically in a wide range of scenarios. These technologies include in-memory online transaction processing (OLTP), primarily for transactional workloads, and in-memory columnstore, primarily for decision support workloads (this last one is discussed in the *Deeper Insights* white paper).

The new query store feature allows you to monitor query plans to optimize them for particular application scenarios over time, providing additional performance-tuning opportunities. Native JSON support in the core database engine is another new feature that provides support for working with schema-free data within SQL Server. The new temporal database features allow you to record, audit, and query data changes over time.

In-memory online transaction processing

In-memory technology for SQL Server dramatically improves the throughput and latency of SQL Server OLTP capabilities. It is designed to meet the requirements of the most demanding transaction processing applications, and Microsoft has worked closely with a number of companies to prove these gains.

The feature set of in-memory OLTP includes the following:

- **Memory-optimized tables:** There are two types of memory-optimized tables. Durable tables are fully logged and persisted over server restarts. Non-durable tables are not persisted and are most commonly used in scenarios such as use in lieu of global temp tables in the user database, or where persistence is not needed, such as staging tables in an Extract Transform Load (ETL) process.
- **Natively compiled stored procedures:** SQL Server can natively compile stored procedures that access memory-optimized tables. Native compilation allows faster data access and more efficient query execution than interpreted (traditional) Transact-SQL. Natively compiled stored procedures are parsed and compiled when they are loaded to native DLLs. This is in contrast to other stored procedures which are compiled on first run, have an execution plan created and reused, and use an interpreter for execution.

In-memory OLTP is designed on the following architectural principles:

- **Optimize for main memory data access:** Storage-optimized engines (such as the current OLTP engine in SQL Server) will retain hot data in a main memory buffer pool based on frequency of access. The data access and modification capabilities, however, are built on the viewpoint that data may be paged in or paged out to disk at any point. With in-memory OLTP, you place tables used in the extreme transaction processing portion of an application into memory-optimized main-memory structures. The remaining application tables, such as reference data details or historical data, are left in traditional storage-optimized structures. This approach lets you optimize hotspots for memory use without having to manage multiple data engines. Main memory structures for in-memory OLTP eliminate the overhead and indirection of the storage-optimized view while still providing the full atomicity, consistency, isolation, and durability (ACID) properties you expect of a database system.
- **Analyze, migrate, report (AMR) Tool:** To identify the appropriate tables to take advantage of in-memory OLTP, the new AMR tool is integrated into the SQL Server 2014 Management Studio to assist in transition to in-memory OLTP. It consists of a set of data collectors, leveraging the existing data collection framework in SQL Server, as well as a set of management data warehouse analysis reports. The reports will provide a set of recommended tables or stored procedures for in-memory OLTP to speed the overall performance.
- **Accelerate business-logic processing:** In-memory OLTP, queries, and procedural logic in procedures that are stored in T-SQL are compiled directly into machine code through aggressive optimizations that are applied at compilation time. Consequently, the stored procedure can be executed at the speed of native code.
- **Provide frictionless scale-up:** In-memory OLTP implements a highly scalable concurrency control mechanism and uses a series of lock-free data structures to eliminate traditional locks and latches while guaranteeing the correct transactional semantics that ensure data consistency.
- **Built into SQL Server:** The most impressive thing about in-memory OLTP is that it achieves breakthrough improvement in transactional processing capabilities without requiring a separate data management product or a new programming model. It is still SQL Server! This allows for an integrated developer and DBA experience with the same T-SQL, client stack, tooling, backup and restore, and AlwaysOn capabilities. By offering in-memory functionality within SQL Server, your total cost of ownership is lower than if you had to purchase, manage, and maintain a separate system for handling in-memory processing.

In-memory OLTP enhancements in SQL Server 2016:

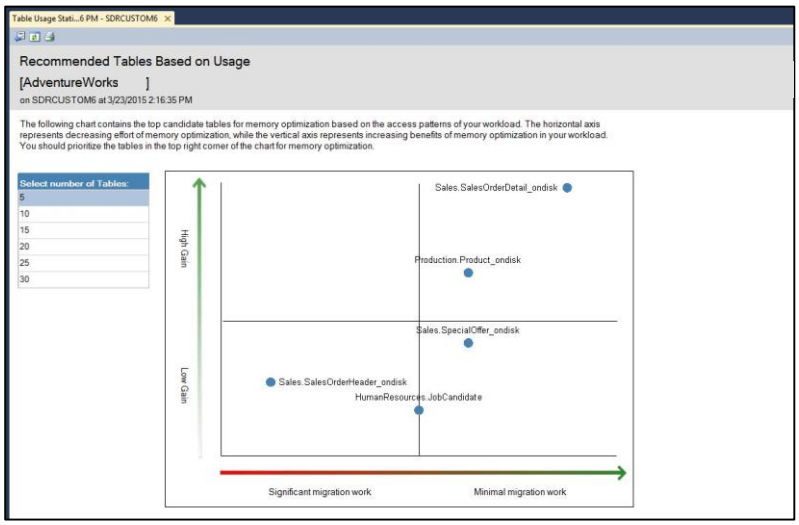
In SQL Server 2014, once a memory-optimized table was created, it needed to be dropped and recreated if a change was required. With SQL Server 2016, ALTER TABLE statements on index and schema changes are now supported.

```
ALTER TABLE Sales.SalesOrderDetail_inmem
    ALTER INDEX imPK_SalesOrderDetail_SalesOrderDetailID
        REBUILD WITH (BUCKET_COUNT=100000000)
```

SQL Server 2016 now has almost full coverage of T-SQL query surface area, adding support for left outer joins, union all, and distinct syntax. It has improved scaling features, including a large increase to the total size of durable tables and improved throughput, with increased socket availability and larger log generation capacity for AlwaysOn configuration.

SQL Server 2016 also has enhanced migration reports that help you identify tables and stored procedures that you might consider migrating to in-memory (Figure 3). These reports are now available directly in SQL Server Management Studio (SSMS), without having to set up a management data warehouse.

Figure 3: Enhanced migration reports in SQL Server 2016



Other improvements include support for multiple active result sets (MARS) and improved auto-update and sampled statistics for performance analysis.

Query-processing enhancements

Cardinality estimator

The cardinality estimator improves the querying process and offers the following benefits:

- **Consistent, predictable query performance:** Different operator trees that represent the same relation have the same cardinality estimates.
- **New model for better performance:** Significant changes to the model result in more accurate cardinality estimates and better plan choices.

- **Easier to support:** The querying process separates querying into two steps, *decision-making* and *execution*. It also produces tracing output to facilitate troubleshooting.

Improvements to trigger the automatic update of statistics enable better query results because statistics invalidation occurs earlier in SQL Server 2014. Faster and more frequent refreshes of statistics are possible because the invalidation threshold has been set to 20 percent of a single partition. In addition, data loading is significantly faster because data insertion into a table can occur in parallel through the SELECT INTO operation.

Reduced database size and increased performance: Data and backup compression

Many organizations want to increase speed and reliability by putting more data onto specialized disk arrays or a SAN, but often they are prohibited by the cost of these high-end disk resources. Backup and data compression in SQL Server can free up space by dramatically reducing the size of databases. Reduced data size also can increase performance. With additional space, more data can be stored on the SAN. And because storing data on the SAN is more reliable, it also increases availability.

Additionally, SQL Server enables data compression for people who use Unicode UCS-2. This capability enables organizations that have global language sets in their data storage to take advantage of data compression and experience the benefits of compression.

Proactive troubleshooting and diagnostics: Performance Data Collector and Management Studio

Organizations want to proactively manage the health of their systems and quality of queries across their environments to ensure the best possible performance. SQL Server delivers a suite of diagnostics and tuning tools built in at no extra cost. Performance Data Collector allows administrators to view SQL Server diagnostics from performance counters, dynamic management views (DMVs), SQL Trace, and other sources for baseline and historical comparisons. They can view performance data with built-in reports on topics such as server activity, disk usage, and query activity. Additionally, SQL Server Profiler can capture server events for real-time diagnosis, and it can correlate traces with performance counters to analyze events and diagnose issues. Dynamic management views and functions that relay server state information help IT administrators monitor the health of server instances, diagnose problems, and tune performance. The Database Engine Tuning Advisor helps administrators select and create an optimal set of indexes, indexed views, and partitions without requiring an expert understanding of the structure of the database or the internal workings of SQL Server. Simply select the databases to tune, and the advisor generates indexing and partitioning recommendations.

New in SQL Server 2016:

Query store

Query store is a new SQL Server component that captures queries, query plans, runtime statistics, etc. in a persistent store inside the database. You can think of it as a flight recorder, or black box, for your database. It can also enforce policies to direct the SQL Server Query Processor to compile queries to be executed in a specific manner, such as forcing plans. It is a database-scoped persistent store of query workload history. Query store is primarily targeting administrative scenarios for performance troubleshooting and identifying regressed workloads.

Query store collects query texts and all relevant properties, as well as query plan choices and performance metrics. This collection process works across restarts or upgrades of the server and across

recompilation of indexes. It provides many configurable options for customization. Query store integrates with existing query execution statistics, plan forcing, and manageability tools.

Query store lowers the bar dramatically for performance troubleshooting and allows you to perform several kinds of scenarios:

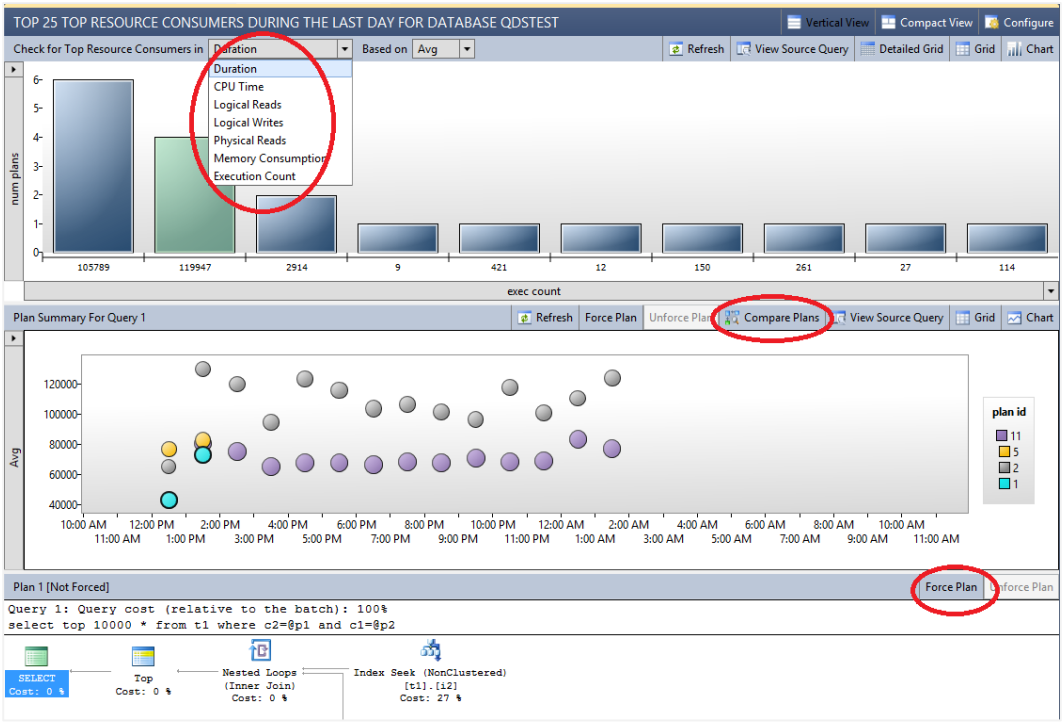
- Conduct system-wide or database-level analysis and troubleshooting
- Access the full history of query execution
- Quickly pinpoint the most expensive queries
- Get all queries whose performance regressed over time
- Easily force a better plan from history
- Safely conduct server restarts or upgrades
- Identify issues with upgrades

All data that query store stores is available through the following views:

- sys.query_store_query_text
- sys.query_store_query
- sys.query_store_plan
- sys.query_context_settings
- sys.query_store_runtime_stats_interval
- sys.query_store_runtime_stats

By combining these views in various queries, you will gain necessary insights about user workload for a period of time when query store was active. Query store also has a rich user-interface component that uses several of these same DMVs (Figure 4). Familiarizing yourself with the UI can be an excellent introduction to the kind of information you can obtain by querying the DMVs.

Figure 4: Dynamic management views in query store



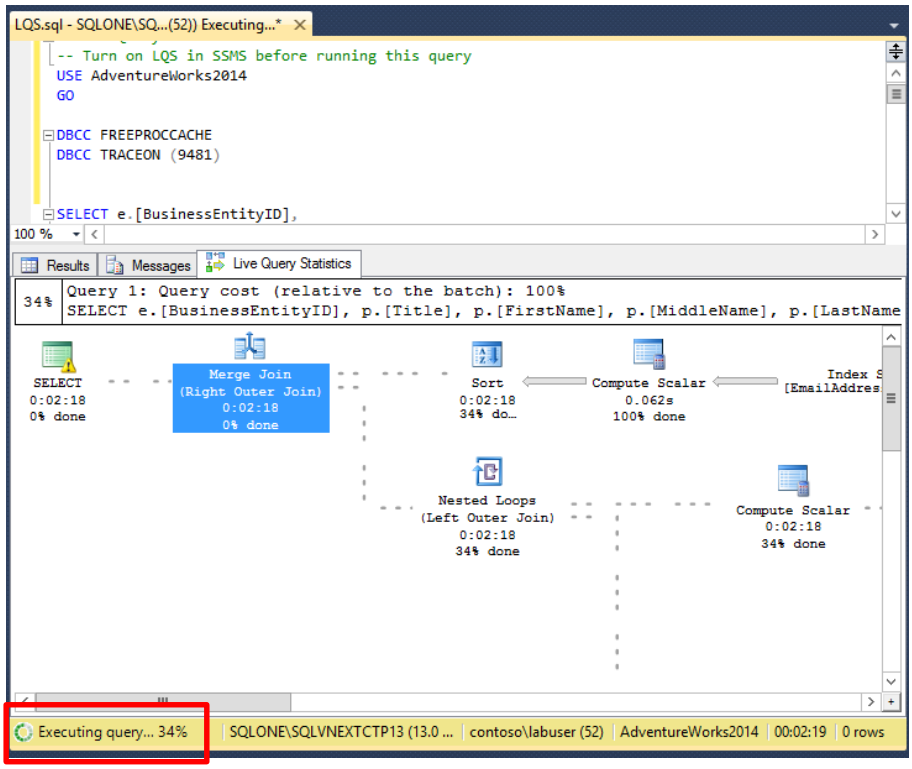
New in SQL Server 2016:

Live query statistics

Before SQL Server 2016, developers and DBAs could troubleshoot query performance with Showplan at query runtime. The runtime execution plan, often referred to as query execution statistics, enables you to collect actual metrics about the query that occurred during its execution, such as its execution time and actual cost, after the query finishes running.

SQL Server 2016 has a new feature, called Live Query Statistics (LQS), that allows you to view what is happening during the query execution (Figure 5). LQS lets you view a list of active queries and associated statistics such as current CPU/memory usage, execution time, query progress, and so on. This enables rapid identification of potential bottlenecks for troubleshooting query performance issues. LQS also allows you to drill down into a query plan of an active query, and view live operator level statistics such as the number of generated rows, elapsed time, operator progress, and live warnings. This facilitates in-depth troubleshooting of query performance issues, without having to wait for query completion. You can watch the statistics change during the query execution in real time.

Figure 5: Live Query Statistics in SQL Server 2016



New in SQL Server 2016:

Native JSON support

JSON is a popular, language-independent data-interchange format used in modern web and mobile applications, as well for storing unstructured data. JSON is an alternate to XML and is more compact than that format, and as such has become the first choice for many applications that need to move data around on the web. One of the biggest reasons JSON is becoming more important than XML is that XML

has to be parsed with an XML parser, while JSON can be parsed by a standard JavaScript function. This makes it easier and faster than working with XML.

SQL Server 2016 supports several new T-SQL constructs to facilitate working with JSON data; however, JSON is not treated as a separate data type by SQL Server, like XML is. For example, appending FOR JSON AUTO to a standard SELECT statement returns the result set in a JSON format. JSON syntax is simple and human-readable. JSON values consist of name/value pairs, and individual values are separated by commas. Objects are containers of one or more name/value pairs and are contained within curly brackets. JSON arrays can contain multiple objects, and arrays are contained within square brackets.

```
SELECT TOP 7 M.ProductModelID, M.Name AS [ProductModel.Name],
ProductID, P.Name AS [Product.Name], ProductNumber, MakeFlag,
FinishedGoodsFlag, Color, Size, SafetyStockLevel, ReorderPoint,
SellStartDate
FROM Production.Product P
INNER JOIN Production.ProductModel M ON P.ProductModelID =
M.ProductModelID
FOR JSON PATH, ROOT('ProductModel')
```

JSON is the storage format used in several NoSQL engines, including Azure DocumentDB. DocumentDB uses Azure Blob storage to store schema-less documents, but provides a rich SQL query dialect that allows you to conduct SQL queries over the data contained in the documents. In addition to DocumentDB, Azure Search also utilizes JSON. Azure Search is a fully managed search solution that allows developers to embed sophisticated search experiences into web and mobile applications without having to worry about the complexities of full-text search and without having to deploy, maintain, or manage any infrastructure.

The combination of SQL Server's new support for JSON with these other Microsoft tools enables many scenarios for moving data back and forth between relational and schema-less storage and the applications that access such data (Figure 6). For example, these tools would allow you to set up periodical extractions of relational data in SQL Server, transforming the data into JSON and loading it to a JSON-based Azure DocumentDB storage that is searchable from a mobile device, along with data from many other sources, utilizing Azure Search.

Figure 6: Data set returned as JSON

New in SQL Server 2016:

Temporal database support

Data is rarely static. Sometimes it can be very valuable to see how data has evolved over time, or to query data as of a particular point in time. While traditional databases store the data that is considered to be valid at the current time only, SQL Server 2016 temporal features provide correct information about stored facts at any point in time.

Each temporal table (or "system-versioned" temporal table) consists of two tables actually—one for current data and the other for historical data. Additional querying constructs are provided to hide this complexity from users.

SQL Server 2016 maintains validity periods for current and old records and ensures an immutable history of data. It also provides a convenient way to query table snapshots from any point in time, or data current over a range of time between two points.

```
SELECT * FROM Person.BusinessEntityContact
FOR SYSTEM_TIME BETWEEN @Start AND @End
WHERE ContactTypeID = 17
```

The temporal features of SQL Server 2016 can also help with the following:

- Providing regulatory compliance and performing auditing
- Reverting a table to "last good known state" without downtime
- Implementing slowly changing dimensions
- Performing time-based data analysis

Since maintaining temporal consistency (between existing and new facts) can be very complex, SQL Server 2016 creates a parallel table structure for current and historical data that works with the datetime2 data type.

Security

Secure by default: Lowering vulnerability

Microsoft and the SQL Server team take security seriously. More than 10 years ago, Microsoft implemented the Trustworthy Computing initiative. The initiative requires SQL Server engineers to take regular security training and carry that responsibility for security across their job duties, regardless of the group in which they reside. This company-wide discipline to protect security and privacy was developed to create software that is secure by design—and to reduce by default the overall risks related to security. To that end, according to the National Institute of Standards and Technology (NIST) public security board, SQL Server reportedly has the lowest number of security vulnerabilities across the major database vendors. In addition, SQL Server has been deemed "the most secure database" by the Information Technology Industry Council (ITIC).[2]

SQL Server 2016 introduces several security innovations. Always Encrypted adds a unique capability to have your data encrypted while at rest and in motion with the ability to query that data while it is

[2] Information Technology Intelligence Corp. (ITIC), SQL Server Delivers Industry-Leading Security, September 2012.

encrypted. This is optimal for internal compliance, especially in regulated industries or for handling very sensitive data, and is accomplished with minimal overhead.

Row-Level Security is another new feature that provides fine-grained access control over rows in a table based upon conditions you set up. A few other enhancements like Transparent Data Encryption for In-Memory OLTP and enhanced auditing capabilities give you several new options to meet your security needs. SQL Server is in the clear leadership position when it comes to security for your mission-critical applications.

Transparent Data Encryption

SQL Server's Transparent Data Encryption (TDE) allows organizations to encrypt data when it is stored on a disk, and decrypt it when it is read into memory. TDE uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. TDE protects data at rest, meaning the data and log files. This enables software developers to encrypt data by using AES and 3DES encryption algorithms, without changing existing applications. Encryption and decryption operations are handled by the database engine in the background. Therefore, organizations do not have to make changes to their applications for SQL Server to secure their data. Because encryption is built into the database engine, it is transparent to applications and users—and it is included in SQL Server Enterprise edition.

Additionally, extensible key management works with TDE to store encryption keys outside of the database. With extensible key management, organizations can use a hardware device or a third-party encryption tool to create encryption keys. Storing the keys separately from the encrypted data makes it even harder for unauthorized users to gain access to encrypted data.

For databases protected by TDE, backups of those databases are also encrypted.

Enhanced in SQL Server 2016:

TDE now supports storage of memory-optimized OLTP Tables. This allows for greater security along with the performance enhancements provided by memory-optimization.

Dynamic Data Masking limits exposure to sensitive data by obfuscating it for non-privileged users. This feature enables you to set up policies at the table and column level that provide multiple masking functions, such as obfuscating the first eight digits and displaying the last four digits of an ID or credit card number. Once the policies have been set up, these masks are applied in queries. You can allow certain privileged logins to see the data unmasked.

Backup Encryption

SQL Server has the ability to encrypt the data while creating a backup. By specifying the encryption algorithm and the encryptor (a certificate or asymmetric key) when creating a backup, you can create an encrypted backup file. On-premises and Window Azure storage locations are supported. In addition, encryption options can be configured for SQL Server Managed Backup to Windows Azure operations, a new feature introduced in SQL Server 2014. To encrypt during backup, you must specify an encryption algorithm, and an encryptor to secure the encryption key. The following are the supported encryption options:

- Encryption algorithm: The supported encryption algorithms are: AES 128, AES 192, AES 256, and Triple DES.
- Encryptor: A certificate or asymmetric key.

Enhanced in SQL Server 2016:

Backup encryption is now supported with compression.

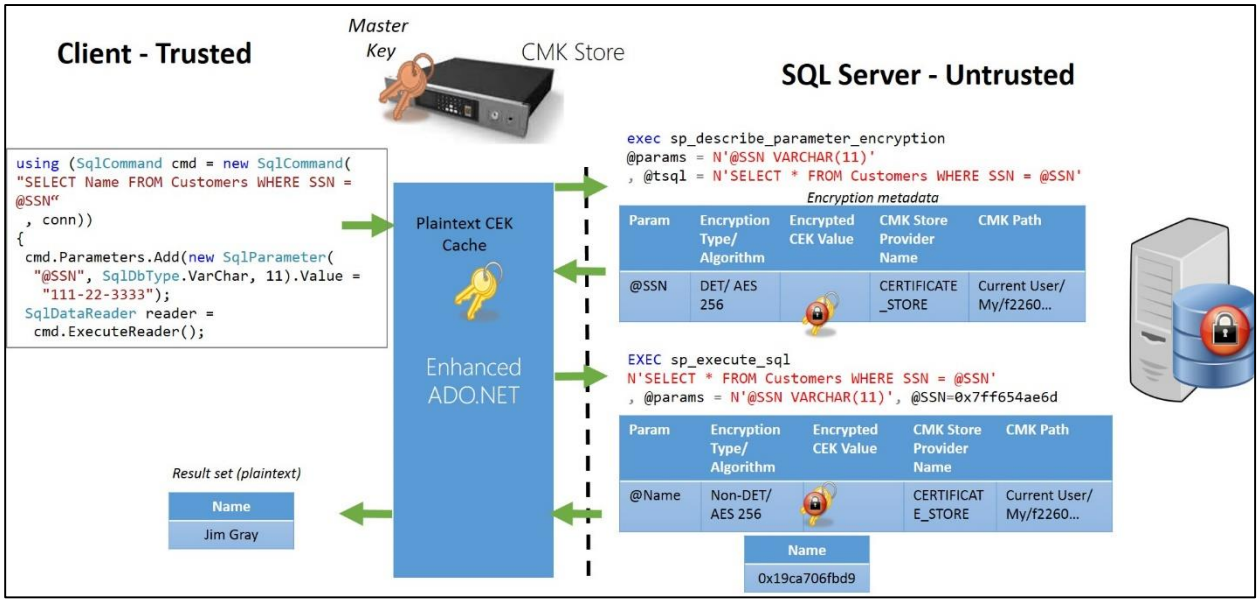
New in SQL Server 2016:

Always Encrypted

TDE does not prevent a security administrator or DBA from accessing the encrypted data. The new Always Encrypted feature in SQL Server 2016 protects sensitive data stored in a SQL database from DBAs and other high-privileged yet unauthorized users. Always Encrypted transparently encrypts data in an Always Encrypted-enabled client driver, before the encrypted data is uploaded to the database (or transparently decrypts data prior to returning it to the application). SQL Server guarantees that the data and the corresponding keys are never seen in plain text on the server, yet SQL Server can process queries against the encrypted data.

Always Encrypted is the first data platform solution on the market providing queryable encryption. SQL Server 2016 supports deterministic encryption, which allows equality comparisons on encrypted columns. Equality operations include joins, group by, and distinct operators. This will allow encryption of sensitive data such as identification or credit card numbers, which are typically only involved in lookup operations. All other operations will fail gracefully as unsupported when executed on encrypted columns. This means, for example, that an application could run a SELECT statement against a particular credit card number, without any credit card numbers being visible to unauthorized users (even users with SA credentials) (Figure 7).

Figure 7: Queryable encryption with Always Encrypted



Using Always Encrypted with client applications requires very few changes either on the client or the server, so the development costs to realize the benefits of this feature are minimal. It does require an Always Encrypted-enabled client driver to sit between the client and the database though. Whether you are setting up Always Encrypted with a new or existing application, implementation is straightforward using SSMS or SQL Server Data Tools (SSDT) and is supported with tools to make the process as easy as possible.

With new applications, you select the columns to be encrypted and encryption settings, set up the content master key (CMK) and the content encryption key (CEK) using the key setup tool, and identify any impacts on the schema or application queries using the schema analysis tool (SSDT only).

With existing applications, the setup additionally requires encrypting the (previously) plain text data in the selected columns. This can be accomplished in two ways: 1) Creating new, encrypted columns and copying the data from the unencrypted columns, then swapping the old columns for the new in the schema and recreating any dependencies from the old to the new. This process is facilitated by the encryption tool. 2) If you are migrating the database to a new target server, you can use the encryption tool in conjunction with Import/Export to migrate the data into an encrypted column.

Controlled access to data for administrators: User-Defined Server Roles

User-Defined Server Roles increases flexibility and manageability, and it facilitates compliance through clearer separation of duties. It allows creation of server roles to suit different organizations that separate administrative duties according to roles. Roles also can be nested to allow more flexibility in mapping to hierarchical structures in organizations. User-Defined Server Roles also helps prevent organizations from using sysadmin for database administration. For example, a special database administration role can be created for common database administration, without the ability to access user data.

New in SQL Server 2016:

Row-Level Security

Row-Level Security (RLS) enables developers and DBAs to implement fine-grained access control over rows in a table. Using RLS, you can store data for different customers, departments, or tenants in the same table, while restricting access to rows based on a query's execution context. For example, you could filter rows returned by "SELECT * FROM myTable" according to the identity of the logged-in user or the value of a session-scoped variable like CONTEXT_INFO.

RLS works transparently at query time, with no application changes required. It uses a centralized security logic that resides inside the database and is schema-bound to the table it protects, providing greater security. Implementing RLS in the database can greatly reduce client application maintenance and complexity.

In RLS, you first create a predicate function that encapsulates your access logic. This is simply a user-defined, inline table-valued function that implements the security logic. It can be as simple or complex as you need it to be and it can reference as many tables as needed. The predicate function filters the rows that any given user can access. For example, imagine a function allows staff to access rows in a patient table only where there is a match between the staff member's assigned hospital wings and the dates that they were assigned to each wing with the patient's wing and admission dates.

```
CREATE FUNCTION [Security].fn_securitypredicate(@wing int,
@startTime datetime, @endTime datetime)
RETURNS TABLE
WITH SCHEMABINDING
AS
     RETURN SELECT 1 as [fn_securitypredicate_result]
     FROM dbo.StaffDuties d
     INNER JOIN dbo.Employees e ON (d.EmpId = e.EmpId)
```

```
WHERE e.databasePrincipalId = DATABASE_PRINCIPAL_ID()
AND @wing = d.Wing
AND ((d.endTime >= @startTime)
     AND
     ((d.startTime <= isnull(@endTime, getdate()))))
```

The second step in implementing RLS is to create a security policy that binds the predicate function to one or more tables. Once bound to the table, all access to the table is routed through the security policy. So, for example, a staff member who runs SELECT * FROM patients would only see those patients who were in her wing during the time she was assigned to that wing.

```
CREATE SECURITY POLICY [Security].[PatientsSecurity]
ADD FILTER PREDICATE [Security].[fn_securitypredicate]([wing],
[startTime], [endTime])
ON [dbo].[Patients]
```

In this example, the predicate function might use the logged-on user's DATABASE_PRINCIPAL_ID or SUSER to identify the rights of the user. In the context of middle-tier apps, where application users share the same SQL login (for instance, via connection pooling), the application could specify the currently logged in user using CONTEXT_INFO upon opening the database connection.

RLS is implemented through the query optimizer. Once the security policy has been created, all queries on the table are automatically rewritten by the optimizer to apply the predicate function. For example, "SELECT * FROM myTable" might be rewritten as "SELECT * FROM myTable WHERE StaffId = DATABASE_PRINCIPAL_ID()".

Built-in tools for enabling compliance: SQL Server audit tools

Database auditing is built into SQL Server to make auditing easy because it is continually available, and to help organizations audit database activities, including database reads, with minimal impact to performance. As compliance policies get increasingly tighter, organizations can use built-in tools such as the following:

- **SQL Server Audit (all editions):** Enables organizations to extend the benefits of SQL Server Audit from Enterprise edition to all SQL Server editions. This extensibility allows for more thorough auditing practices across SQL Server databases, and it enables standardization, better performance, and richer features.
- **User-Defined Audit:** Allows the middle-tier application to write custom events into the audit log, which enables more flexibility to store audit information.
- **Audit Filtering:** Provides greater flexibility to filter unwanted events in an audit log.
- **Audit Resilience:** Gives the ability to recover auditing data from temporary file and network issues to help ensure that audit logs are not lost during failover.

Enhanced in SQL Server 2016:

Audit enhancements

SQL Server 2016 supports important new functionality for auditing. When a user attempts to perform an operation, the user must first have permission to perform that operation. Prior versions of SQL Server could track and audit whether a permissions check succeeded or failed. Now with SQL Server 2016, you can also audit whether the underlying operation was successful. So, even if a user has permission, it is still possible that the desired operation might fail due to referential integrity rules or another constraint. SQL Server 2016 can now track the success or failure of the operation, in addition to the permission check. This applies to both explicit as well as implicit transactions.

Controlled access to data across business intelligence tools: Microsoft SharePoint and Microsoft Active Directory

As Microsoft continues to deliver business intelligence tools that are used by a broader set of users, security concerns also increase because of broader implications if security is compromised. SQL Server helps organizations secure end-user data analytics with built-in IT controls, including new SharePoint and Active Directory security models for end-user reports that are published and shared in SharePoint. Enhanced security models provide control at row and column levels.

Availability

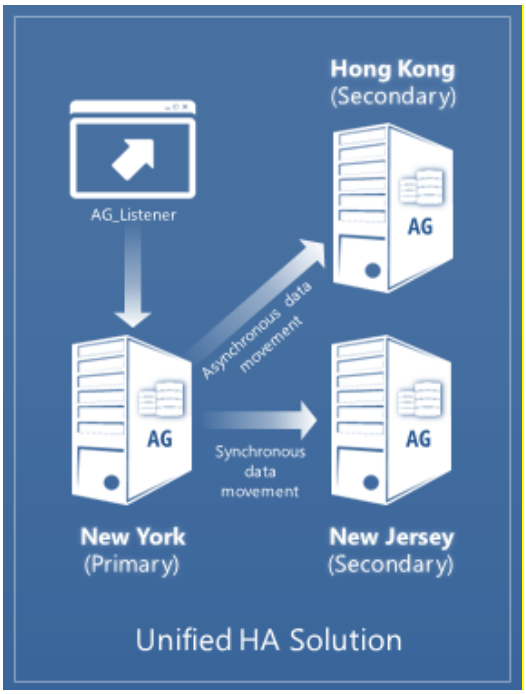
High availability of mission-critical systems

SQL Server high availability solutions provide mission-critical uptime, fast failover, improved manageability, and better use of hardware resources. AlwaysOn continues to get better and more powerful with every release. SQL Server 2016 adds significant new enhancements to AlwaysOn. You can now have up to three synchronous replicas, as well as readable asynchronous replicas. These replicas can exist in different domains and be on-premises and on Azure virtual machines. SQL Server 2016 also adds better load balancing of replicas using a round-robin methodology. This version has also resolved compatibility issues with Distributed Transaction Coordinator (DTC) and SSIS when using AlwaysOn. You now also have greater uptime with Enhanced Online Operations when conducting ALTER or TRUNCATE operations on tables. SQL Server 2016 also the high availability and scalability capabilities of SQL Server standard edition by adding AlwaysOn basic for two-node failover and removing core and memory restrictions on standard edition.

AlwaysOn

AlwaysOn was first introduced in SQL Server 2012, and SQL Server 2014 continues to deliver an enhanced high availability solution (Figure 8). This integrated high availability and disaster recovery solution provides redundancy within a data center and across data centers to help enable fast failover of applications during planned and unplanned downtime. AlwaysOn delivers a suite of capabilities rolled into a single solution.

Figure 8: Unified high availability solution



SQL Server AlwaysOn Availability Groups is a high availability and disaster recovery solution that provides an enterprise-level alternative to database mirroring. Availability Groups are an integrated set of options that include automatic and manual failover of a group of databases, support for as many as eight secondary replicas (“*secondaries*”), faster failover for applications, and automatic page repair. Each availability group is a container for a discrete set of user databases known as *availability databases* that fail over together. An availability group can have many possible failover targets (secondary replicas). Moreover, organizations can easily configure secondary replicas to support read-only access to secondary databases and back up secondary databases. The addition of Availability Groups removes the requirement of shared disk storage such as storage area network (SAN) or network-attached storage (NAS) for deployment of a Failover Cluster Instance.

SQL Server AlwaysOn Failover Cluster Instances enhance SQL Server Failover Clustering and support multisite clustering across subnets, which helps enable failover of SQL Server instances across data centers. Faster and more predictable failover of instances is another key benefit that helps ensure faster recovery of applications. By supporting Windows Server Cluster Shared Volumes, AlwaysOn further improves use and management of SAN storage through increased resilience of storage failover and avoidance of the drive-letter limitation in SAN.

SQL Server AlwaysOn Multiple, Active Secondaries enables use of as many as eight secondary instances for running report queries (many times faster than replication) and backup operations, even in the presence of network failures—which helps in repurposing idle hardware and improving resource utility. It also helps to dramatically improve performance for both primary and secondary workloads because they are no longer competing for resources.

SQL Server AlwaysOn Availability Groups Listener enables faster failover in client connections for AlwaysOn in scenarios that employ multiple subnets. Now, client applications can achieve failover across multiple subnets (as many as 64) almost as fast as they can achieve failover within a single subnet. Meanwhile, the ability to set the connection from within applications to read-only (instead of read and

write) empowers organizations to control the type of workloads that run on their high availability servers, so they can more efficiently manage their resources.

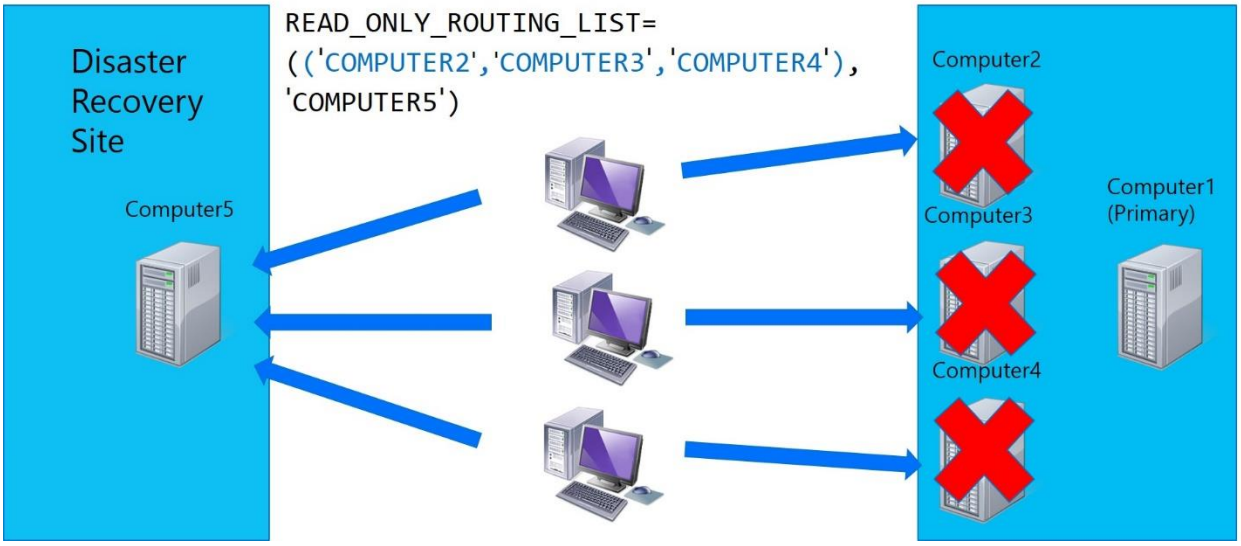
SQL Server AlwaysOn to Azure Virtual Machine enables organizations to add secondary replicas in an Azure Virtual Machine through the Add Azure Replica Wizard. They can then use this replica for disaster recovery, reporting, and backup operations. This configuration can lower capital expenses by eliminating the need to purchase additional hardware for AlwaysOn secondaries.

Enhanced in SQL Server 2016:

Enhanced AlwaysOn

SQL Server 2016 makes major improvements in AlwaysOn in the areas of scalability and manageability. For scalability, SQL Server 2016 adds in load balancing of readable secondaries. This allows you to define one or more groups of readable secondaries to load balance. Connections are assigned round-robin to members of the group. This is set up by specifying an extension of the READ_ONLY_ROUTING list (Figure 9).

Figure 9: AlwaysOn—readable secondary load balancing



This version also increases the number of auto-failover targets from two to three. Finally, log transport performance has been improved. With in-memory OLTP and other technologies pushing database speeds, the high availability pipeline becomes more critical. SQL Server 2016 offers improvements in both throughput and latency.

Manageability has also been improved in several areas with SQL Server 2016, including support for Distributed Transaction Coordinator (DTC)–enrolled transactions for Availability Group (AG) databases. DTC resources are tied to the database instead of the instance so, on failover, the DTC sees the same resource on the new primary, and transaction outcomes can be resolved.

Another manageability improvement is database-level health monitoring. In SQL Server 2014, AG health only monitors the health of the instance. A database can be offline or corrupt, but as long as the instance itself is healthy, SQL Server won't trigger a failover. SQL Server 2016 allows you to optionally change the health monitoring to also consider the health of the databases in the AG.

Group managed service accounts (GMSA) are another manageability improvement. GMSAs are domain-level accounts that are automatically managed. These are similar to the default service account, but with a domain scope. This enables setting permissions for network resources across the AG instances without requiring user accounts.

Online database operations

SQL Server continues to enable organizations to achieve high availability during resource-intensive operations. For example, the ability to rebuild online indexes in a single partition provides partition-level control for users who need continual access to the database. This approach also requires fewer resources (CPU and memory), so it minimizes the impact of rebuilding indexes. Specifically, the ability to manage priorities for locking tables gives organizations greater control over the impact of maintenance operations on running transactions—from table switching to online index rebuild operations—by allowing database administrators to specify whether or not to terminate processes that block their ability to lock tables.

Enhanced in SQL Server 2016:

Enhanced online operations

SQL Server 2016 now provides 100-percent uptime, with enhanced online database operations when conducting ALTER or TRUNCATE operations on tables.

Predictable, efficient, and flexible data backups

Recovery Advisor provides an enhanced user experience for database administrators to restore databases using SQL Server Management Studio. SQL Server offers a variety of backup types, so creating the right recovery sequence for any point in time can get tricky. To help streamline this process, SQL Server Recovery Advisor helps database administrators create a more predictable and optimal restore sequence.

Capabilities include a visual timeline that shows the backup history of the database and the available points in time to which the user can restore the database; algorithms that help streamline the process of identifying the right sets of backup media to get the database back to a specific point in time; and a page restore dialog box in SQL Server Management Studio to do page-level restores of the database.

Scalability

The interaction between SQL Server and Windows Server is an area that can lead to large improvements in scalability. SQL Server can take advantage of the full feature set in Windows Server in a number of areas. With SQL Server and Windows Server, physical processing now scales up to 640 logical processors, and virtual machines scale up to 64 logical processors. SQL Server also utilizes storage spaces and network virtualization to optimize your resources. It can run on Windows Server Core to lower the surface area of attack. From advancements in compute, to storage and networking, all have a direct impact on mission-critical SQL Server workloads. A number of other tools and techniques can help boost scalability as well.

Setting up a private cloud can be a complex undertaking. SQL Server has several enhancements that can make this job easier. Additionally, SQL Server is integrated into Microsoft's Cloud Platform System, discussed below, which provides new possibilities for scaling a private cloud very quickly.

Better together with Windows Server

Support for Windows Server Core

SQL Server is supported on Windows Server Core—the Windows Server edition with the smallest footprint. Because Windows Server Core requires less maintenance and fewer OS patches, planned downtime is greatly reduced when you run SQL Server on Windows Server Core. The percentage reduction in patching and OS reboots can be as much as 50 to 60 percent in certain environments, depending on the server roles that are enabled and the type of patches that are applied.[3]

Support for Windows Server ReFS

SQL Server 2014 supports the usage of Windows Server ReFS (Resilient File System) to provide maximum availability, scalability, and integrity. ReFS gives organizations a cost-effective platform that maximizes data availability, scales efficiently to very large data sets across diverse workloads, and guarantees data integrity by providing resilience to corruption (regardless of software or hardware failures).

Faster live migration

Windows Server allows simultaneous migration of as many SQL Server virtual machines as you need, which helps organizations maintain availability of SQL Server while decreasing planned downtime. Faster live migration also helps organizations decrease planned downtime by allowing migration of many SQL Server virtual machines (using priority settings) in a clustered environment, and by using as much as 10 GB of network bandwidth.

Live migration for non-clustered virtual machines

Windows Server allows live migration of SQL Server virtual machines in a non-clustered environment, both in centrally shared and non-shared virtual machine storage scenarios. This practice helps organizations reduce the cost and complexity of SQL Server deployments in virtualized environments while maintaining availability during planned downtime.

Cluster-Aware Updating

With Cluster-Aware Updating, organizations can apply updates automatically to the host operating system, or other system components in a clustered SQL Server environment, while maintaining availability. This approach can significantly help to increase SQL Server availability during the update process in both virtualized and non-virtualized environments.

Dynamic Quorum

Windows Server Failover Clustering Dynamic Quorum enables the SQL Server AlwaysOn cluster to dynamically adjust the number of quorum votes that are required to keep the system running. This adjustment can simplify setup by as much as 80 percent. It also helps increase availability of a SQL Server cluster in failover scenarios in both virtualized and non-virtualized environments—with the ability to recalculate a quorum as needed and still maintain a working cluster.

---

[3] "Why Is Server Core Useful," Microsoft TechNet, http://technet.microsoft.com/en-us/library/dd184076.aspx, accessed May 15, 2013.

Enhanced in SQL Server 2016:

Hardware Acceleration for TDE Encryption/Decryption

SQL Server 2016 now implements Microsoft's next-generation cryptography (CNG) API to take advantage of specialized microprocessor instructions (e.g., INTEL's AES-NI, which is an extension to the x86 instruction set available from Intel and AMD) in order to speed up encryption/decryption of data for TDE using hardware. In the case of INTEL, the hardware acceleration is advertised to improve performance of an implementation of AES (Advanced Encryption Standard) by three to 10 times over a pure software implementation.

Parallelizing the Decryption Built-in Function to Improve Read Performance

Before SQL Server 2016, the decryption function was marked as sequential and hence not parallelizable by the optimizer when SQL Server is running on a multi-core machine. The lack of parallelism could be an issue in some situations. SQL Server 2016 marks the decryption function as parallelizable, which improves the performance of read operations on encrypted data (e.g., SQL Server column encryption). Allowing the decryption functions to run in parallel should result in dramatically better response times for queries with encrypted data columns.

Enhancements working with Windows Server 2016

SQL Server 2016 is able to take advantage of several enhancements in Windows Server 2016, such as increased upward limits on memory available to the operating system.

Buffer pool extension

SQL Server enables improvement to query performance by allowing the use of non-volatile devices such as solid-state drives (SSDs) to reduce SQL Server memory pressure with no risk of data loss. The configuration is simple and can greatly improve query performance.

Tier-1 partitioning: Scale to 15,000 partitions

SQL Server supports as many as 15,000 table partitions. This support enables large *sliding window* scenarios, which means that applications such as SAP, which take tens of thousands of snapshots of data in daily or hourly partitions, can significantly extend the length of time data is held before it's "pushed out" to allow for new data to enter—generally making it easier to manage these large amounts of data. This capability also helps administrators streamline maintenance of large data sets within file groups that need data switched in and out according to the needs of the data warehouse.

Scalable real-world application testing: Distributed Replay

Organizations need a way to apply real-world application loads to their applications within test environments. Previously, they could use SQL Server Profiler, which only allowed simulation of a workload from a single computer. This limitation made it difficult to test large-scale workload simulation.

Distributed Replay helps organizations to simplify application testing and minimize errors, with application changes, configuration changes, and upgrades. This multithreaded replay utility enables simulation to test production workload scenarios after upgrades or configuration changes—ultimately leading to protected performance during changes. Additionally, integration with SQL Server Upgrade Assistant can help organizations assess the impact of future SQL Server upgrades.

Enhanced in SQL Server 2016:

TempDB optimization

SQL Server also allows you to scale up your database with enhanced data caching, using support for multiple TempDB files per instance for multi-core environments. This reduces metadata- and allocation contention for TempDB workloads, improving performance and scalability.

Private cloud

Resource Governor enhancements

Resource Governor enables organizations to further ensure consistent performance for concurrent and mixed workloads across different SQL Server applications and within private clouds. Database administrators can define which workloads can take what percentage of performance on any given CPU, memory, and I/O resource. The Resource Governor in SQL Server allows scaling with the maximum number of resource pools at 64. Also, it enables the use of minimum and maximum capacity settings in the CPU, memory, and input/output operations per second (IOPS). Scalability is also improved through an affinity of resource pools with CPU schedulers and Non-Uniform Memory Access (NUMA) nodes, as well as governance of I/O resources allowing administrators to control physical I/O for users by adding a setting for maximum and minimum IOPS per volume.

Sysprep for SQL Server

SQL Server supports the preparation of virtual machine templates through SQL Server Sysprep. Administrators can prepare images with the desired features and then deploy them later in private and public cloud environments. SQL Server Sysprep supports SQL Server Database Engine, SQL Server Reporting Services, SQL Server Analysis Services, SQL Server Integration Services, and shared features. With the addition of cluster support, SQL Server Sysprep can be used in a wider variety of image-preparation scenarios.

Cloud Platform System

The Microsoft Cloud Platform System (CPS) is designed specifically to reduce the complexity and risk of implementing a self-service cloud. Because CPS includes all of the required software and hardware, service providers and enterprises can give customers the self-service offerings they are demanding. As a result, these providers can respond quickly to business opportunities—without worrying about having the ability to manage dynamic, highly virtualized workloads.

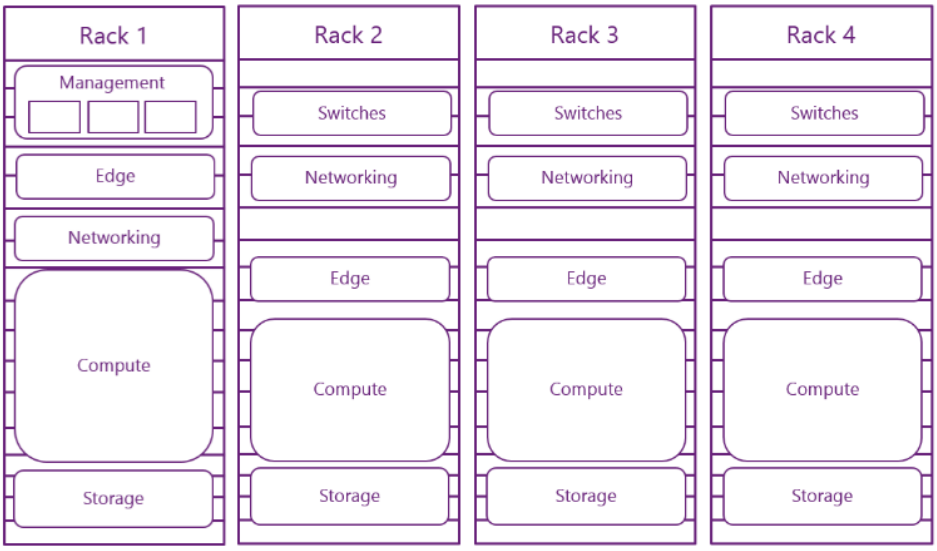
CPS can go from delivery to live within days—not months—and lets service providers and enterprises move up the stack to focus on delivering services to users. The Cloud Platform System converges the hardware and software needed to create the agile datacenter of the future—specifically, Windows Server 2012 R2, Microsoft System Center 2012 R2, and Windows Azure technologies. Preconfigured hardware and software working together speeds the ability to offer customers the infrastructure as a service (IaaS) and platform as a service (PaaS) resources they want, whether that includes self-provisioned virtual machines, web applications, or other resources.

A major part of the appeal of the Cloud Services model is the variety of prebuilt templates that can accelerate productivity for the end user. With the Cloud Platform System, service providers and enterprises can build a robust portfolio of potential workloads that they can make available to customers and end users. To help providers quickly start delivering highly scalable workloads, Microsoft has established workload templates for CPS for Active Directory, Microsoft Exchange, Microsoft SharePoint,

and Microsoft SQL Server. These templates (known as VMRoles) are co-designed with each of the Microsoft product teams to deliver highly available, performant, and protected deployments. Many non-Microsoft workloads are available, such as CentOS 6, Oracle, and others. Each of these workloads can be offered to tenants.

The solution scales from half-rack proof-of-concept deployments to single-rack and multi-rack production deployments, with transparent upgrade paths. Therefore, as customer needs change over time, the hardware can expand while services remain continuously available. As the infrastructure grows from one rack up to potentially four racks, additional elements are included to enable connectivity—treating the whole infrastructure as a single pool (stamp). The management cluster that drives the services is the only cluster that remains on the first rack only. A four-rack stamp provides 128 compute nodes across all different compute scale units (Figure 10).

Figure 10: Scaling out the infrastructure of CPS to four racks



When it comes to management, the Cloud Platform System lets service providers and enterprises manage the entire stamp fabric. As a result, true datacenter agility can be achieved. Management services are predeployed, integrated, and customized for CPS in a highly available configuration. Administrative work is done from the same consoles that should be familiar to service providers or enterprise IT staff, using the System Center 2012 R2 suite to manage all of the underlying cloud fabric used by the tenants. Moreover, CPS provides a single source of support versus having to contact multiple vendors. Finally, Microsoft has made heavy investments in testing, with best practices applied. Ultimately, this means that prequalified patches, firmware updates, and driver updates are packaged for the entire solution.

If a service provider or enterprise owns multiple data centers, it can use Azure Site Recovery to create a disaster recovery plan between two CPS stamps.

Comparing the cost of mission-critical capabilities

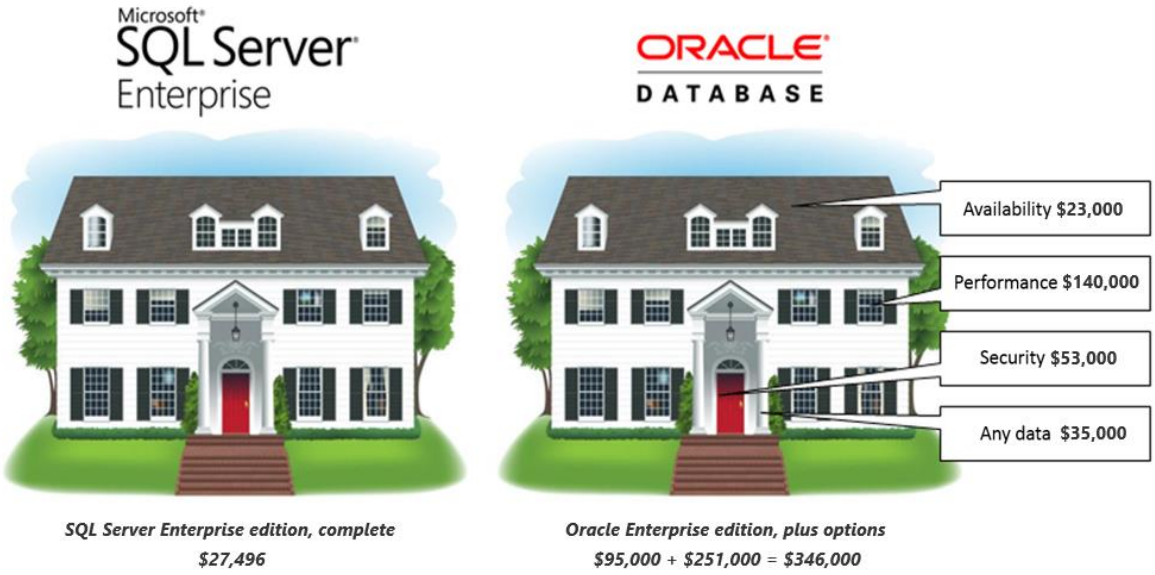
In a technology landscape where organizations expect vendors to intuitively know what "mission critical" means and to provide easy and cost-effective solutions, Microsoft answers these expectations with enterprise-class tools and abilities that are built into database technology—without the need to purchase costly add-ons. SQL Server delivers the mission-critical capabilities required by organizations to compete in a dynamic digital world. The features discussed in this paper are all included in SQL Server Enterprise Edition and don't require costly options to deliver a complete and modern database solution.

Expecting the fundamentals

Many legacy vendors deliver limited functionality in their premium editions and only provide mission-critical requirements through additional options or feature packs. Examples of additional options are security features, high availability, performance, and spatial capabilities. Organizations have evolved—and it's no longer optional for vendors to ensure enterprise-grade security features, availability, performance, or support for complex data types, to name a few.

Just like home buyers expect a roof, windows, and doors to be included in their purchase, an organization can expect an enterprise-class database to include built-in availability, performance, and security features. Figure 11 shows the difference between two similar database solutions, Microsoft SQL Server and Oracle Database, with the budget impact of adding options to arrive at a similar end state.

Figure 11: SQL Server and Oracle Database compared



Understanding options

Table 1 highlights the options required across the major database management system (DBMS) vendors to meet mission-critical needs in modern organizations. What used to be optional is more often required by organizations to meet the new standard in mission-critical operations. It is easy to see how achieving mission-critical readiness by adding options or feature packs can dramatically change the total cost of a database solution.

Table 1: Comparison of mission-critical solutions from Microsoft and Oracle

	Microsoft SQL Server	Oracle Database (all options not shown)
Enterprise Edition base license (includes 1-year support)	**$27,496**	**$95,000**
Data availability	Included	$11,500 (Active Data Guard) **$23,000 Total**
Performance and scale	Included	$11,500 (Advanced Compression) $23,000 (Database In-Memory) $11,500 (Partitioning) $11,500 (Real Application Testing) $7,500 (Diagnostics Pack) $5,000 (Tuning Pack) **$140,000 Total**
Enterprise security	Included Free Download	$15,000 (Advanced Security) $11,500 (Label Security) **$53,000 Total**
Any data, built-in	Included	$17,500 (Spatial & Graph) **$35,000 Total**
Total cost	**$27,496**	**$346,000**

Note Microsoft prices are based on estimated retail price. All Microsoft and Oracle prices are per-processor (based on a quad core Intel Xeon processor) database pricing for purchases within the United States and are in U.S. dollars. Pricing is based on information available on vendor websites. Oracle prices are based on the Oracle Technology Global Price List, April 9, 2015.

In addition to its mission-critical database functionality, SQL Server Enterprise includes a range of capabilities for data integration, data management, data warehouse, data cleansing, and end-to-end business intelligence. According to the Gartner Magic Quadrant for Data Warehouse and Data Management Solutions for Analytics[4], Microsoft is positioned as a leader in bringing enterprise data warehouse platform to companies by product innovation such as in-memory columnstore technology. In addition, according to the Gartner Magic Quadrant for Business Intelligence and Analytics Platforms,[5] Microsoft is positioned as a leader in helping organizations to enable broad end-user insight and productivity—balanced with IT oversight through managed self-service business intelligence tools that work both as standalones and within Microsoft SharePoint. SQL Server delivers access to these industry-leading business intelligence capabilities, without requiring costly add-ons. With SQL Server, business intelligence tools are built into the base Enterprise license and are also available in the new Business Intelligence edition. Organizations also can increase cost-savings through built-in data integration, management, and cleansing tools. These tools enable data quality managers to easily cleanse and manage data through SQL Server Integration Services, Master Data Management, and Data Quality Services. Similar business intelligence and data management tools with other vendors can add up to hundreds of thousands of dollars in additional costs.

[4] Gartner, Magic Quadrant for Data Warehouse and Data Management Solutions for Analytics, February 2015
[5] Gartner, Magic Quadrant for Business Intelligence and Analytics Platforms, February 2015.

Conclusion

SQL Server delivers a new standard in enabling mission-critical operations—with true enterprise-class availability, performance, and security features built into the solution. Integrated high availability solutions enable faster failover and more reliable backups—and they are easier to configure, maintain, and monitor, which helps organizations reduce the total cost of ownership (TCO). SQL Server also delivers mission-critical performance and scale, with predictable performance across server activities including complex queries, data integration, and analysis. Because SQL Server is designed to security standards, it has minimal total surface area and database software that is inherently more secure. Enhanced security, combined with built-in, easy-to-use tools and controlled data access, helps organizations meet strict compliance policies. SQL Server supports complex data types *and* non-traditional data sources, and it handles them with the same attention—so organizations experience seamless support for a variety of platforms and heterogeneous environments. Finally, SQL Server delivers mission-critical capabilities at low TCO—with full enterprise capabilities that are built into the solution, not provided as costly add-ons. Ultimately, organizations can rely on a comprehensive, integrated solution that helps to contain costs and manage compliance requirements while meeting the demands of the evolving digital world.

More information

For more information about topics discussed in this white paper, see the SQL Server website at http://www.microsoft.com/en-us/server-cloud/products/sql-server-2016/

Feedback

Did this paper help you? Please give us your feedback by telling us on a scale of 1 (poor) to 5 (excellent) how you rate this paper and why you have given it this rating. More specifically:

- Are you rating it highly because of relevant examples, helpful screen shots, clear writing, or another reason?
- Are you rating it poorly because of examples that don't apply to your concerns, fuzzy screen shots, or unclear writing?

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

Please send your feedback to: mailto:sqlfback@microsoft.com