

SQL Server In-Memory OLTP and Columnstore Feature Comparison

Technical White Paper

Authors: Jos de Bruijn (Microsoft), Sunil Agarwal (Microsoft), Darmadi Komo (Microsoft), Badal Bordia (ASPL), Vishal Soni (ASPL)

Technical reviewers: Bill Ramos (Indigo Slate)

Published: March 2016

Applies to: Microsoft SQL Server 2016 and SQL Server 2014

Summary: The in-memory features of Microsoft SQL Server are a unique combination of fully integrated tools that are currently running on thousands of production systems. These tools consist primarily of in-memory Online Transactional Processing (OLTP) and in-memory Columnstore.

In-memory OLTP provides row-based in-memory data access and modification capabilities, used mostly for transaction processing workloads, though it can also be used in data warehousing scenarios. This technology uses lock- and latch-free architecture that enables linear scaling. In-memory OLTP has memory-optimized data structures and provides native compilation, creating more efficient data access and querying capabilities. This technology is integrated into the SQL Server database engine, which enables lower total cost of ownership, since developers and database administrators (DBAs) can use the same T-SQL, client stack, tooling, backups, and AlwaysOn features. Furthermore, the same database can have both on-disk and in-memory features. In-memory OLTP can dramatically improve throughput and latency on transactional processing workloads and can provide significant performance improvements.

In-memory Columnstore uses column compression for reducing the storage footprint and improving query performance and allows running analytic queries concurrently with data loads. Columnstore indexes are updatable, memory-optimized, column-oriented indexes used primarily in data warehousing scenarios, though they can also be used for operational analytics. Columnstore indexes can be created in both the clustered and nonclustered varieties. Columnstore indexes organize data into row groups that can be efficiently compressed, which improves performance. Queries that utilize a Columnstore index can use batch-mode processing, which is optimized for in-memory performance. Columnstore indexes can be particularly useful on memory-optimized tables.

The SQL Server in-memory solutions lead to dramatic improvements in performance, providing faster transactions, faster queries, and faster insights—all on a proven data platform architecture. This white paper examines the key components of these tools and compares them with solutions from other providers, demonstrating how in-memory technology in SQL Server outpaces other solutions.

Copyright

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, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2016 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Microsoft Azure, Excel, SharePoint, SQL Server, Windows, and Windows Server, are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

- Overview of SQL Server in-memory advances 3
 - Drive real-time business with real-time insights.....3
 - OLTP transactions4
 - Analytics queries.....4
 - SQL Server in-memory OLTP overview4
 - Memory-optimized tables4
 - Indexes on memory-optimized tables.....4
 - Concurrency improvements.....5
 - Natively compiled stored procedures.....5
 - In-memory OLTP customer success story5
 - SQL Server in-memory Columnstore overview6
 - Clustered and nonclustered Columnstore indexes.....6
 - Batch-mode query processing.....7
 - In-memory Columnstore customer success story.....8
- Overview of in-memory technologies for Oracle, and IBM 8
 - Oracle Database In-Memory.....8
 - IBM DB2 BLU Acceleration.....8
- Comparison of Features..... 9
 - Oracle.....9
 - Oracle TimesTen: In-memory OLTP9
 - Comparing Oracle and SQL Server in-memory OLTP.....9
 - Oracle 12c: In-memory Columnstore10
 - Comparing Oracle 12c and SQL Server in-memory Columnstore.....11
 - IBM.....12
 - IBM DB2 10.5: Analytics with BLU Acceleration12
 - Comparing IBM DB2 BLU Acceleration and SQL Server in-memory13
- Myths and reality: SQL Server in-memory OLTP and in-memory 14
 - Latch-free architecture14
 - Separate database engine14
 - Not suitable for OLTP14
 - Response to competitors' in-memory offerings.....15
 - In-memory OLTP is the same as the old SQL feature DBCC PINTABLE.....15
- Conclusion 15
- Key resources..... 16

Overview of SQL Server in-memory advances

Drive real-time business with real-time insights

In-memory OLTP and in-memory Columnstore of Microsoft SQL Server leap a generation of OLTP and data warehousing solutions to show typical performance improvements of 10–100x. With in-memory OLTP, organizations can accelerate their transactional applications by up to 30x and can deliver faster business insights with more than 100x faster queries and reports. In-memory Columnstore indexes, with

scan rates of tens of billions of rows per second on typical industry hardware, provide users the ability to interact with and explore an unprecedented amount of data at the speed of thought. SQL Server provides organizations with a robust data platform architecture that can advance your business goals with the ability to transact and analyze data in near real-time.

OLTP transactions

In-memory OLTP is a memory-optimized database engine integrated into the SQL Server engine, optimized for OLTP workloads. The OLTP engine uses latch-free and lock-free data structures and multi-version concurrency control to support extremely high concurrency rates. The result is high throughput with linear scaling for database transactions. The actual performance gain depends on many factors, but a 2–20x improvement in performance is very common. In fact, a database can see transactional performance gains as high as 30x over traditional table and database engines.

Analytics queries

The SQL Server in-memory Columnstore index stores and manages data by using column-based data storage and column-based query processing. Columnstore indexes can transform the data warehouse experience for users by enabling faster performance for common data warehouse queries such as filtering, aggregating, grouping, and star-join queries. Columnstore indexing can be used to achieve up to 100x query-performance gains over traditional row-oriented storage and significant (typically 10x) data compression for common data patterns.

SQL Server in-memory OLTP overview

Memory-optimized tables

The SQL Server in-memory OLTP engine allows you to create in-memory optimized OLTP tables within your existing relational database. “Memory-optimized” tables (as opposed to standard, “disk-based” tables) reside completely in-memory. A key difference of memory-optimized tables over disk-based tables is that memory-optimized tables store the data as rows. No pages need to be read into memory when the memory-optimized tables are accessed. A set of checkpoint files (data and delta file pairs) is created in a memory-optimized file group for data persistence, similar to the data files used for disk-based tables. However, these checkpoint files, unlike data files used for disk-based tables, are append-only, allowing SQL Server to leverage the full I/O bandwidth of the storage. These checkpoint files, along with the transaction log, provide full durability and are used for recovery at restart or for database backup/restore.

There are two main types of in-memory-optimized OLTP tables: `SCHEMA_AND_DATA` (i.e. durable tables) and `SCHEMA_ONLY` (i.e. non-durable tables). The first type provides full durability guarantee just like disk-based tables for your OLTP workload. It is the default setting when creating memory-optimized tables. The second type persists the schema of the table but not the data. `SCHEMA_ONLY` would be used in scenarios where OLTP workloads do not require data persistence, such as session state management for an application, for staging tables in an ETL scenario, or as a replacement for temporary tables in TempDB that use a table type.

Indexes on memory-optimized tables

Memory-optimized tables support two types of nonclustered indexes: hash and range indexes. Hash indexes provide optimal access paths for equality searches, while range indexes are used for queries involving range predicates or for ordered retrieval of the data.

Every memory-optimized table must have at least one index. For durable memory-optimized tables, a unique index is required to uniquely identify a row when processing transaction log records change during

recovery. Indexes on in-memory tables reside only in-memory. They are not stored in checkpoint files nor are any changes to the indexes logged. The indexes are maintained automatically during all modification operations on memory-optimized tables, just like B-tree indexes on disk-based tables, but if SQL Server restarts, the indexes on the memory-optimized tables are rebuilt as the data are streamed into memory.

Concurrency improvements

Applications whose performance is affected by engine-level concurrency, such as latch contention or blocking, improve significantly when the application is migrated to in-memory OLTP. Memory-optimized tables do not have pages, so there are no latches and hence no latch wait. If your database application encounters blocking issues between read and write operations, in-memory OLTP removes the blocking issues because it uses optimistic concurrency control to access data. The optimistic control is implemented using row versions, but unlike disk-based tables, the row versions are kept in-memory. Since data for memory-optimized tables are always in-memory, the waits due to I/O path are eliminated. Also, there will be no waits for reading data from disks and no waits for locks on data rows.

Natively compiled stored procedures

SQL Server can natively compile stored procedures that access memory-optimized tables. A natively compiled stored procedure optimizes TSQL statements, transforms them into Visual C code, and then generates a DLL. This enables SQL Server to execute the business logic in the stored procedure at an order of magnitude of better efficiency using fewer instructions compared to traditional stored procedures. SQL 14 allows commonly used TSQL statements in OLTP workloads to be used inside natively stored procedures. The SQL Server team continues to expand the TSQL surface area in new releases.

In-memory OLTP customer success story



“Every second counts for a player waiting to place a bet, and by using in-memory OLTP in SQL Server 2014, we provide a faster-loading site and a faster overall experience, so players can place more bets and play games more smoothly.”

Rick Kutschera

bwin.party was formed from a merger between two gaming giants, bwin Interactive Entertainment and PartyGaming, each of which had high-traffic gaming websites. Consolidation of their websites resulted in huge overloads on their infrastructure. bwin.party needed to overcome its scalability issues and wanted to improve the performance of its gaming website to support rapid business growth.

Solution:

Using the Microsoft in-memory OLTP solution in SQL Server 2014, their gaming systems can now handle 250,000 requests per second (almost 20 times the original load) and offer players a faster, smoother gaming experience.

Benefits:

- **Scalability:** bwin.party gaming systems can scale to 250,000 requests per second, close to a 20x improvement.
- **Faster, smoother playing experience:** The standard system response time improved from 50 milliseconds to 2–3 milliseconds, resulting in faster and better performance.

Manager of Database
Engineering,
bwin.party

- **Reduced costs and increased revenue:** bwin.party is able to run in-memory OLTP on less hardware, which could save it as much as \$100,000 a year.

For more details, see <https://customers.microsoft.com/Pages/CustomerStory.aspx?recid=12362>.

SQL Server in-memory Columnstore overview

In-memory Columnstore uses columnar storage format to reduce the storage footprint significantly, typically 10x, while still delivering up to 100x better analytics query performance. You can obtain an additional 30% compression by using COLUMNSTORE_ARCHIVE compression for data that are not frequently referenced. Columnar storage format provides significant data compression by storing each column separately. Since the data within a column are similar and often repeated, SQL Server can achieve a very high level of data compression. Higher compression rates improve query performance by using a smaller in-memory footprint. Analytic queries often select only a few columns from the FACT table. With columnar storage, only the required columns are read into memory, reducing I/O even further, unlike row-based storage format where all columns get loaded into memory as part of the rows.

Clustered and nonclustered Columnstore indexes

SQL Server 2014 supports updatable clustered Columnstore indexes, which replace the traditional rowstore tables. The clustered Columnstore index allows users to modify data and load data concurrently for data warehouse and Decision Support System (DSS) workloads. Improved query performance of up to 100x speed up is provided with reduced I/Os and optimized query execution using techniques such as applying predicates in compressed format, pushing down predicates to storage layer when possible, leveraging new processor architectures, and a new BATCH execution mode.

SQL Server also allows you to create nonclustered Columnstore indexes on rowstore tables primarily for operational analytics, such as the ability to do analytics on operational store. The nonclustered Columnstore index is updatable in SQL Server 2016.

The rows in Columnstore indexes are generally grouped in a set of 1 million rows to achieve optimal data compression. This grouping of rows is referred to as a rowgroup. Within a rowgroup, the multiple values for each column are compressed and stored as LOBs. These LOB units are referred to as segments. The column segments are the unit of transfer between disk and memory.

Batch-mode query processing

Batch-mode query processing is basically a vector-based query execution mechanism, which is tightly integrated with the Columnstore index. Queries that target a Columnstore index can use batch-mode to process up to 900 rows together, which enables efficient query execution, providing 3-4x in query performance improvement. In SQL Server, batch-mode processing is optimized for Columnstore indexes to take full advantage of their structure and in-memory capabilities.

In-memory Columnstore customer success story



"A query that took a half hour to run could delay a report by a day or more. With SQL Server 2014, we do not expect that to happen."

Anatoly Ternov

Database Administrator and
ETL Team Leader,

Clalit Health Services

Part of the mission of Clalit Health Services, which provides healthcare to 60 percent of the Israeli population, is to continually improve clinical outcomes for its members. With a large, complex data infrastructure, it faces a real challenge to manage more users, more data, and more complex queries. The query response time is the real productivity killer for business analysts, and they needed a more effective way to conduct analyses on complex queries.

Solution:

Clalit conducted a proof of concept using the Columnstore index feature of Microsoft SQL Server 2014. It ran the "problem queries" on each database version without implementing any code changes or tuning.

Benefits:

- **Supports greater analyst productivity:** With query time reduced from 20 minutes to three seconds, the 300 analysts who work with the data warehouse daily see their productivity soar.
- **Reduces disk needs by 40 percent:** Queries ran in 40 percent less disk space than queries to earlier data software.
- **Delivers deeper data insights:** Analysts and researchers will not just run more analyses in less time, they will also run more complex analyses in less time.

For more details, see <https://customers.microsoft.com/Pages/CustomStory.aspx?recid=4166>.

Overview of in-memory technologies for Oracle, and IBM

At present, several organizations offer in-memory databases for a variety of tasks. We are going to examine in-memory technologies from Oracle and IBM.

Oracle Database In-Memory

Oracle Database In-Memory is a new option for Oracle database 12c. It is an optional add-on to Oracle Database 12c that enables the flagship relational software of Oracle to function as an in-memory database. Oracle Database In-Memory works on any platform running Oracle Database 12c. Oracle requires a separate product—Oracle TimesTen—for in-memory OLTP, which is not fully integrated into their main database.

IBM DB2 BLU Acceleration

IBM DB2 BLU Acceleration, developed by the IBM Research and Development Labs, is a new in-memory feature integrated into IBM DB2 10.5. It uses the same storage and memory constructs (storage groups, table spaces, buffer pools, etc.); SQL language interfaces; and administrative tools as traditional DB2. IBM DB2 BLU Acceleration is an in-memory Columnstore implementation suitable for data warehouse and analytics applications.

Comparison of Features

Oracle

At present, Oracle Database does not provide native support for in-memory capability for OLTP workload but supports in-memory as an optional add-on with Oracle Database 12c for analytics workloads. Oracle also offers standalone in-memory capability with a separate product called "TimesTen" that has been integrated with Oracle Database, but these are two different products that need to be installed/managed separately.

Oracle TimesTen: In-memory OLTP

TimesTen, an in-memory relational database system, started off as a research project in HP in 1995 and it later was moved into a separate company. TimesTen was acquired by Oracle in 2005, and since then, they have been working on integrating it with Oracle stack, including PL/SQL and OCI stack, and in their MAA (max availability architecture).

TimesTen can be deployed in the following ways:

- **Client/server interface:** Traditional client/server interfaces are supported, enabling scenarios like reporting and access to common in-memory databases for a large number of application-tier platforms.
- **Directly linked applications:** Applications can be directly linked to the TimesTen address space, eliminating the IPC overhead involved and streamlining query processing for optimized query performance.
- **Cache for Oracle Database (TimesTen cache):** When a dedicated Oracle database already exists for a workload, TimesTen can be deployed as an additional layer of cache database between the application and the Oracle database. The cache tables in this middle layer can be read-only or updatable. Applications can access the TimesTen cache tables using a standard SQL interface, and the synchronization between these cache tables and the Oracle database is performed automatically.

Comparing Oracle and SQL Server in-memory OLTP

Oracle TimesTen is a relatively old product. In contrast, the in-memory OLTP of SQL Server leverages a number of technology advancements, making it a superior toolset in many ways. Table 2 provides a feature comparison between Oracle TimesTen and SQL Server 2014 in-memory OLTP and SQL Server 2016 in-memory OLTP, which adds an increased SQL language surface area for native compilation and additional durability features.

Feature	TimesTen	SQL Server 2014	SQL Server 2016
SQL language	Supports most of PL/SQL (including DW)	InterOp supports most OLTP	Increase T-SQL surface area
Native compile	No	Yes. T-SQL surface area targeting OLTP workloads	Increased T-SQL surface area
Lock-based	Yes; row, table, database locks; choose at connection level	No. Uses optimistic concurrency control	No new changes
Integration	Loose integration with Oracle Database	Fully integrated with SQL	No new changes
Durability	At database level (permanent and temporary database)	At table level. All tables are durable by default but individual tables can be marked as non-durable.	2 TB of durable memory optimized tables in the database
Security		Auditing and Permissions to control access to tables and for Admin operation	Transparent data encryption

Table 1. In-memory OLTP: Oracle TimesTen vs. SQL Server

- **SQL language:** Both TimesTen and TimesTen Cache support PL/SQL, including data warehouse. The key advantage of this capability is that any PL/SQL application running on an Oracle server can be easily migrated to TimesTen with little change. In a similar fashion, SQL Server in InterOp mode supports most OLTP.
- **Native compile:** Native compilation for in-memory OLTP is not supported by Oracle TimesTen. In SQL Server, the operations (stored procedures) can be natively compiled on the in-memory OLTP tables to achieve maximum business processing performance. In future releases of SQL Server, the surface area will be further expanded to further enhance native compilation capabilities.
- **Lock-based:** Oracle provides locking mechanisms at row, table, and database levels, which can be configured at the time of connection. This method often leads to concurrency bottlenecks. SQL Server has no locks because it provides optimistic concurrency. Thus, it provides a friction-free scale-up.
- **Integration:** Because Oracle TimesTen is a separate product, it needs a mechanism to integrate it with Oracle Database. Microsoft in-memory OLTP is not a separate product, but a part of SQL Server. This makes it more efficient from backup, restore, and management perspectives. Also, because SQL Server in-memory OLTP is integrated with the database, you can choose to migrate only performance-critical tables into memory-optimized tables. SQL Server includes some useful reports that help you identify which tables should be migrated to memory optimization.
- **Durability:** In Oracle, durability is at the database level. In SQL Server, durability is at the table level, which allows flexibility to configure tables within an application appropriately. For example, you can create a non-durable memory-optimized table to stage data.

Oracle 12c: In-memory Columnstore

Oracle provides an in-memory Columnstore as an optional add-on with Oracle Database 12c. When this option is enabled for the table or the tablespace, Oracle internally creates a copy of the subset of the data in columnar storage format. This allows the database administrators to choose and populate only the

most performance-critical data in-memory to speed up analytics. The changes to in-memory data are not persisted and need to be rebuilt on the fly when Oracle DB is restarted. Oracle in-memory is designed to be transparent to applications and tools (Figure 1).

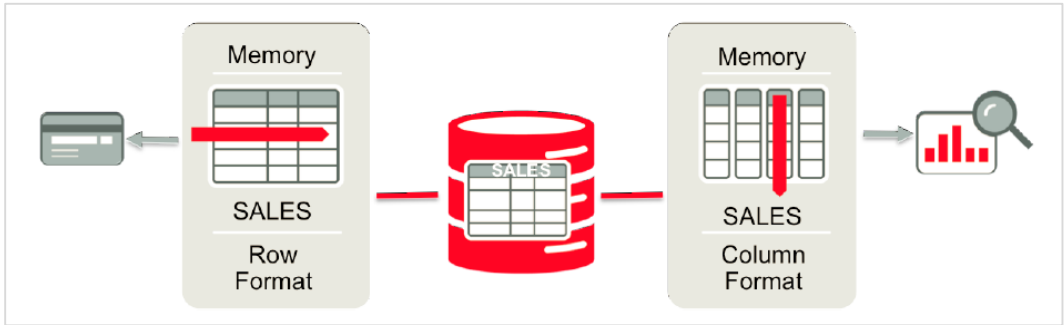


Figure 1: The dual-format architecture of Oracle¹

Comparing Oracle 12c and SQL Server in-memory Columnstore

Table 3 presents a feature comparison between Oracle 12c in-memory Columnstore and SQL Server in-memory Columnstore.

Feature	Oracle 12c	SQL Server 2014	SQL Server 2016
Persistence of Columnstore	No	Yes	Yes
Aggregate pushdown	Yes	No	Yes
Query on secondary replica	No	No	Yes
Materialized views with Columnstore	Yes	No	No
Integration with in-memory OLTP	No	No	Yes
Batch mprocessing	No	Yes	Yes
Integration with R	Yes	No	Yes
Configuration and operations	Provides many knobs making the configuration and operations lot more complex	Fully automated after creating Columnstore indexes	Fully automated after creating Columnstore indexes, including management of deleted rows

¹ <http://www.oracle.com/technetwork/database/in-memory/overview/index.html>

Table 2. In-memory: Oracle 12c vs. SQL Server

- **Persistence of Columnstore:** Data warehouses are typically large, and everything cannot be expected to be kept in-memory at all times. When Oracle Database is restarted, there are no data in the Columnstore index, and it is built in the background. Until then the analytics query cannot leverage columnar storage. SQL Server persists the Columnstore index; therefore, there is no need to rebuild it or provision memory to keep it all in-memory. SQL Server brings data in and out of memory based on queries run.
- **Aggregate pushdown:** This refers to having a pre-calculated aggregation (sum, average, etc.) of database values and pushing it down to the storage layer. Oracle 12c supports aggregate pushdown. In SQL Server 2014, sum and aggregates are calculated outside the storage engine in the execution layer. In the upcoming release of SQL Server, this will be pushed down to the storage layer, leading up to 4x or better performance.
- **Query on secondary replica:** In the upcoming version of SQL Server, users will be able to run data warehouse queries on the secondary replica. Oracle includes a configuration similar to AlwaysOn, but it does not allow users to allow the data warehouse queries on the secondary replica. This is a huge win for SQL Server as data warehouse workloads are mission-critical and are configured for high availability.
- **Materialized views with Columnstore:** Materialized views have a very peculiar use case. If a user can create a materialized view on an OLTP table and store that materialized view in a Columnstore format, this could result in almost cube-like functionality. There would be pre-aggregated data, which are always kept updated, like OLTP. This is quite expensive to maintain, however, and although it has value, it will impede OLTP and data-load performance.
- **Integration with In-Memory OLTP:** At present, neither SQL Server 2014 nor Oracle 12c provides support for integration with in-memory OLTP. However, SQL Server 2016 does integrate Columnstore index transparently with the OLTP workloads. To enable this, an updatable Columnstore index is created on one or more tables in an OLTP workload. This allows users to run the OLTP workload on the table and perform queries on the same table because Columnstore is updatable.
- **Batch mode processing:** Another major advantage of SQL Server 2014 is its batch mode processing capability, which significantly improves (typically 2-4x) query performance. Oracle currently does not have any such technology.

IBM

IBM DB2 10.5: Analytics with BLU Acceleration

IBM DB2 BLU Acceleration was released as an integrated in-memory computing solution with IBM DB2 10.5. It has several optimizations, but it is primarily driven by a concept called a “shadow table” that stores and maintains a copy of the data in columnar storage. Both tables remain in sync automatically. OLTP transactions are performed directly on the relational tables, but any analytical queries on those tables are redirected toward the column-based shadow tables, which provides faster analytical processing.

IBM DB2 BLU Acceleration encompasses “Seven Big Ideas,” or key functionalities:

- **Simple to use:** IBM claims that this feature is ready to use as soon as it is enabled. A user only needs to load the data and query. No other indexes or fine-tuning are required. Related operations like load, backup, and restore operations are also simplified.
- **Actionable compression:** Aside from using an optimized compression mechanism, IBM also claims to allow operations to be performed on the compressed data directly. In fact, BLU Acceleration can perform joins or aggregates and also apply predicates directly to the compressed data without having to decompress the data first.

- **Multiplied CPU power:** IBM uses a new concept, called Single Instruction Multiple Datasets (SIMD), to apply a single instruction to multiple data elements. Using SIMD, many routines can be executed simultaneously, resulting in faster query execution.
- **Core-friendly parallelism:** If the workload is running on a multi-core machine, IBM claims to leverage all of the processing power to parallelize the process. This is possible because the software is written from the ground up to take advantage of multiple cores.
- **Columnar storage:** Data are organized as columns, which provides benefits like efficient storage, faster query performance, and simplified fine-tuning.
- **Scan-friendly caching:** IBM claims that it uses optimized memory and cache-management techniques that are separately optimized for OLTP and data warehouse workloads. IBM claims that by using scan-friendly caching, it can minimize the effect on I/O performance.
- **Data skipping:** Data skipping ignores large segments of data that do not qualify for any query. This provides performance savings at the CPU, RAM, and I/O levels, thus enabling faster queries with no fine-tuning. Data skipping is the same mechanism as the segment elimination concept in SQL Server.

Comparing IBM DB2 BLU Acceleration and SQL Server in-memory

Table 4 provides a feature comparison between IBM DB2 BLU and SQL Server.

Feature	DB2 BLU Acceleration	SQL Server 2014	SQL Server 2016
“Seven Big Ideas”	Yes	Yes (mileage will vary)	Yes (mileage will vary)
Query performance	Yes	Yes (batch mode)	Yes (batch mode, aggregate pushdown, lookup/short-range queries)
Concurrent DML	No	Yes	Yes (improved concurrency with row-level locking and non-blocking reads)
Automatic index maintenance	Yes	No	Yes
Operational analytics	Yes (using shadow table)	No (could achieve by manually moving to CCI)	Yes (fully integrated)

Table 3. In-memory: IBM DB2 BLU Acceleration vs. SQL Server

- **“Seven Big Ideas”:** IBM DB2 BLU Acceleration is built around “Seven Big Ideas,” which include concepts and technologies like Columnstore tables, data compression, hardware-level processing-related optimizations, and memory and cache management. Microsoft uses the same concepts and technologies, but at different levels of implementation. These same ideas are also being propagated in the upcoming version of SQL Server.
- **Query performance:** IBM DB2 BLU Acceleration has implemented a mechanism for improving query performance for analytics. Microsoft also has this, but with a special mode called batch mode that delivers better performance. DB2 makes no mention of batch mode. In upcoming versions of SQL Server, batch mode execution will be added for more operators. For example, it is not possible to perform order-by queries in batch mode with SQL Server 2014, but it will be possible in upcoming versions. Microsoft is investing more in batch mode so more data warehouse queries may run much faster.
- **Concurrent DML:** IBM DB2 BLU Acceleration workloads do not behave well with concurrent DML because of blocking-related issues. Although it is not available in SQL Server 2014, the upcoming

version of SQL Server includes an implementation for running concurrent DMLs on Columnstore at a row-level locking. This implementation will be more concurrent in light of DML operations.

- **Automatic index maintenance:** IBM DB2 BLU Acceleration provides automatic index maintenance. In general, when data are deleted, corresponding rows are not removed from the clustered Columnstore index immediately. Those rows are marked with a delete label or flag, which signifies that the rows are deleted. Over time, when many rows have been deleted, they still occupy space in the Columnstore. One way to remove those deleted rows is to rebuild the indexes after a certain period. DB2 provides an automated index maintenance that removes the deleted rows automatically from the index. This feature is not currently available with SQL Server 2014, but it is available in the upcoming version of SQL Server.
- **Operational analytics:** DB2 uses the concept of shadow tables, which create new Columnstore tables based on an existing relational table. From an application perspective, the user has an OLTP table and a shadow table linked to it. The user can run workloads on the OLTP table, and analytics queries will be redirected toward the shadow table automatically. In SQL Server 2014, this needs to be done manually. The upcoming version of SQL Server will allow users to create a non-clustered Columnstore index, with which they can run OLTP and analytics on same table. There will not be a shadow table, but there will be an index that will be updatable.

Myths and reality: SQL Server in-memory OLTP and in-memory

Latch-free architecture

Myth: Since there is no locking, latching, or blocking in SQL Server in-memory OLTP, it can cause inconsistencies and data corruption.

Reality: SQL Server in-memory OLTP has full ACID support—it ensures Atomicity, Consistency, Isolation, and Durability of the data. The innovative lock-free architecture using optimistic concurrency control for in-memory improves performance by eliminating the need to take locks (i.e. no lock manager) and friction-free scaling by removing blocking. To prevent corruption of the links between the indexes and the row versions, SQL Server uses atomic test/set instruction that guarantees the operation is atomic across all the processors in the system.

Transaction logging ensures durability of data in SQL Server. A transaction is marked as committed only when its log has been flushed to the disk to ensure durability of memory-optimized tables. Logging for memory-optimized tables is very efficient, and the amount of log data generated is minimal.

Separate database engine

Myth: SQL in-memory OLTP is a separate product, and you need to redesign or rebuild your database.

Reality: Unlike the other major in-memory database products available today, in-memory OLTP is fully integrated into SQL Server. This means that it requires no separate installation and there is no need to learn different tools. It also allows an incremental investment strategy, where the user can selectively move tables to the most appropriate storage for the data represented by each table. Since it is built into the core SQL Server, other SQL Server functionalities can also be leveraged in addition to in-memory OLTP. SQL Server is the only mainstream DBMS with an integrated in-memory solution optimized for OLTP. Oracle TimesTen is a separate product.

Not suitable for OLTP

Myth: Columnstore in SQL Server 2014 is not suitable for OLTP.

Reality: Clustered Columnstore in SQL Server 2014 is positioned as a data warehouse technology, not OLTP. SQL Server 2014 adds options for clustered and non-clustered Columnstore indexes. The clustered Columnstore index permits data modifications and bulk load operations. Future releases of SQL Server will allow an updatable non-clustered Columnstore index both on memory-optimized and disk-based tables for real-time analytics on operational workload.

Response to competitors' in-memory offerings

Myth: In-memory OLTP is a recent response to competitors' in-memory offerings

Reality: The project, code-named "Hekaton," was started five years ago in response to business and hardware trends. Microsoft started an incubation project in partnership with Microsoft Research to imagine what a database engine designed from scratch for today's hardware realities would look like. The in-memory OLTP feature is the outcome of that incubation.

In-memory OLTP is the same as the old SQL feature DBCC PINTABLE

Myth: In-memory OLTP is the same as the old SQL 7.0 feature DBCC PINTABLE, which allowed pinning buffer pool pages or tables in memory.

Reality: In-memory OLTP uses a completely new design built from the ground up to optimize for efficient in-memory data operations. Data in memory-optimized tables are not organized in pages, and do not use the buffer pool. By dispensing with data structures and other infrastructure intended to facilitate paging subsets of data between disk and memory, in-memory OLTP provides a much leaner and more efficient data engine while still retaining the essential characteristics of the data engine.

Conclusion

Hardware trends such as declining memory costs, multi-core processors, and stalling CPU clock rate increases prompted the architectural design of in-memory computing. In-memory computing is one of the fastest-growing trends in the technology industry. Most technology vendors, including Microsoft, SAP, IBM, and Oracle, offer in-memory solutions to speed query performance.

The Microsoft in-memory processing capability built into SQL Server 2014 delivers breakthrough performance to enable new transformational scenarios to accelerate your business. Microsoft offers comprehensive in-memory technologies for OLTP, data warehouse, and analytics built directly into SQL Server. SQL Server 2016 continues to enhance in-memory technologies to provide improved functionality and performance.

The Microsoft in-memory OLTP solution not only accelerates transactions but also increases concurrency, giving you true scale-up without requiring the entire database to be loaded into memory. Likewise, its in-memory Columnstore solution for data warehouse accelerates queries and significantly reduces storage costs with high data compression.

Only SQL Server has built in in-memory technology optimized for OLTP, which means faster transactions. Plus, its enhanced in-memory Columnstore gives you faster queries and insights while minimizing total cost of ownership. For example, using the in-memory technology in SQL Server, bwin.party, a leader in online gaming, was able to boost performance gains by 17x and queries by 340x.)

The in-memory design in SQL Server removes database contention with lock-free and latch-free table architecture while maintaining 100-percent data durability. Neither Oracle nor IBM provide this. This means that you can take advantage of all your computing resources in parallel for more concurrent users. Unlike other offerings, SQL Server provides built-in in-memory capabilities, so there is no need to learn

new development tools or APIs. Finally, there is no additional cost to use the in-memory OLTP feature, unlike Oracle and IBM.

Key resources

To learn more, visit:

Microsoft SQL Server 2016

<http://www.microsoft.com/en-us/server-cloud/products/sql-server-2016/>

Microsoft SQL Server OLTP and database management

<https://www.microsoft.com/en-us/server-cloud/solutions/oltp-database-management.aspx>

In-Memory OLTP documentation

[https://technet.microsoft.com/en-us/library/dn133186\(v=sql.120\).aspx](https://technet.microsoft.com/en-us/library/dn133186(v=sql.120).aspx)

Download and evaluate SQL Server 2014

<http://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2014>

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>