Oracle Database In-Memory
A Survey of In-Memory Databases from SAP, Microsoft, IBM and Oracle
Introduction

In-Memory Database (IMDB) technology is one of the most active data management software categories in 2014. In December 2013 a major technology research firm predicted that “Memory Optimized (‘In-Memory’) Database Technology is taking over Enterprise Databases”.¹ In this prediction, author Carl Olofson declared “This technology is sure to go mainstream in many enterprise data centers by the end of 2014”. In support of this prediction, between June 2013 and July 2014, IBM, Microsoft and Oracle released memory-optimized database technology as part of their mainstream relational databases, joining SAP’s HANA in-memory database released in late 2010. Thus, all four major database vendors had embraced IMDB technology as of mid-2014.

Databases have used in-memory data structures for many years to avoid the relatively slow performance of reading and writing directly to disk drives. As a result, many startup databases or caching products today claim to be part of the “in-memory” category and it would be near impossible to compare this broad spectrum of implementations. For the purposes of this paper, IMDB technologies are those that go further than just accessing data in memory. They apply formats and algorithms that take advantage of memory and modern processors and aren’t burdened by compatibility with disk-based approaches. The IMDB products from Oracle, IBM, Microsoft and SAP generally meet this definition, though to different degrees.

Performance is the rationale behind IMDB technology. More specifically, high-performance analytics is the most common objective, although faster OLTP might also be targeted. Analytics includes reporting, business intelligence and ad-hoc queries, often against a Data Warehouse and increasingly against transactional (OLTP) databases if possible. In the case of all four vendors, high-performance analytics is supported through data stores organized by columns rather than the traditional row format. The technologies and techniques employed are similar across all four vendors, such that the performance differences may not be substantially different. The major differences lay in other areas, such as scalability, application transparency and mixed workload support.

SAP HANA

Summary: The SAP HANA In-Memory Database has been available since November, 2010. There have been 9 subsequent HANA releases at 2-3 per year through December 2014. HANA is marketed as a way to perform real-time OLTP and OLAP against a column-format data store. SAP is the only vendor that promotes the use of a column-format data store for all database workloads.

SAP refers to HANA as a “platform” for real-time applications, rather than an in-memory database. However, the functional components of HANA are similar to those within a database such as Oracle, supporting many types of data, multiple ways to access and search for information, and a variety of programming interfaces and tools. The HANA programming interface is built on SQL but, like every database, has many unique constructs that render existing database applications (that use Oracle, IBM and Microsoft databases) incompatible with HANA without modifications. On closer inspection, one can also challenge the degree to which HANA can compare with mature enterprise databases, particularly in areas such as availability and disaster recovery.²

SAP has nurtured a community of third-party software vendors developing applications on HANA and has modified many SAP applications to use HANA, such as SAP Business Warehouse (BW) and SAP Business Suite. Organizations can deploy HANA on-premise after purchasing an SAP-certified hardware appliance, or on a public cloud from SAP, Amazon, Microsoft, IBM or several other cloud providers. According to SAP, there are thousands of SAP HANA customers.³ SAP has energized the IMDB category by being the first major vendor to emphasize the technology. Although SAP is early in the development cycle of a full-fledged database system and has yet to verify some of their claims, they have validated the ability to power real-time analytics with an IMDB and have attracted a respectable following considering the immaturity of the product. SAP has gone strategically “all in” on in-memory computing with HANA.

Technology: SAP HANA has traditionally been a "pure" IMDB, which limited HANA to databases that could be fully loaded into memory. This meant that HANA could not support very large databases where the size exceeded the amount of memory available. SAP partially addressed this problem with the dynamic tiering option and support for "extended tables" released in late 2014. Extended tables are disk-based structures that allow HANA to access data on disk as well as in memory. This solved the large database problem, but introduced other issues. Besides the issue that extended tables will be very slow to access, use of extended tables is also not transparent to the application. They must be

manually created and then SAP expects the application to define which data belongs where. Since the data cannot reside in both places simultaneously, data must be manually moved between a HANA in-memory table and a disk based “extended table” or vice versa. This is logic that may need to be added to every application that accesses data in a large database.

HANA does have support for row based tables, but the performance emphasis is on columnar based tables and this is the format that all of HANA’s performance features operate on. This is a significant issue in HANA, as a table is either row based or it is column based; it cannot be both simultaneously. So although row based tables are supported, HANA is primarily a columnar database and SAP emphasizes that most tables should be columnar based. This implies that a single data format is suitable for both OLTP and analytics. Let’s examine this implication.

SAP has yet to validate that a column-format data store can process high-volume OLTP. It is intuitive that a single row insert into a row-format table is more efficient than inserting values into potentially hundreds of columns in a column-format table. Not surprisingly then, although SAP claims over 1,450 users of SAP Business Suite applications (OLTP workloads) on HANA, no such references detailing OLTP performance results can be found.

While the concept may sound appealing, computer science fundamentals do not support SAP’s claims for HANA performing high-volume OLTP from a columnar data store. Even SAP researchers have admitted to the challenge and offered scant solutions. The two big areas of concern are high volume DML operations (insert, update and deletes) and HANA’s ability to merge these changes into the main column store without delays, and the ability to process high volumes of “row” based queries where all or most of the columns of a row are requested. Both of these workloads are typical of high volume OLTP databases. While HANA may be suitable for SAP-designed applications, it’s not clear whether

---

4 [https://www.youtube.com/watch?v=ae1qIViv5sk](https://www.youtube.com/watch?v=ae1qIViv5sk)
5 [http://sites.computer.org/debull/A12mar/hana.pdf](http://sites.computer.org/debull/A12mar/hana.pdf)
HANA will provide a good general purpose database solution, and it is precisely the mixed workload environment for which SAP has said HANA is well suited.\textsuperscript{3}

SAP also makes claims about HANA's ability to scale out, but HANA uses a shared-nothing, scale-out (cluster) architecture that is unproven for large clusters and has never been demonstrated for OLTP workloads. Previous attempts at a shared-nothing architecture for OLTP, by major vendors, have been abandoned as infeasible.

Perhaps another sign of expediency is revealed when one looks under the covers of HANA. Far from being a grounds-up “pure” development, HANA is the combination of several database and search products that SAP acquired or developed; MaxDB, P*Time, and TREX.\textsuperscript{7} SAP has not disclosed the manner in which these three separate databases have been integrated into HANA, though it stands to reason that considerable complexity exists.

**Strategy:** Although SAP has traditionally been known as an applications company, its corporate strategy has shifted significantly toward building the HANA platform and integrating current and new SAP and third-party applications on top of HANA, preferably in the cloud.\textsuperscript{3} Yet the reshaping of an enterprise as large as SAP is by no means certain, particularly given the technical complexities of an enterprise database platform.

Bending to the reality that a “pure in-memory” database store is generally impractical for all but the smallest environments, SAP has attempted to address the issue in its last two HANA releases, SPS 08 and 09. First a near-line storage solution based upon the old Sybase IQ platform for archived data was introduced and more recently the dynamic tiering option that enables the administrator to create disk based tables to contain older, less frequently accessed data.

For SAP’s existing customers, this “bet the farm” strategic shift may be viewed as adding risk and diverting resources from applications, and imposing frequent product updates for early adopters.

**Validation:** SAP has successfully evangelized HANA and in-memory computing and described new categories of analytics applications that take advantage of HANA. Many hardware vendors have introduced HANA appliances in support of SAP’s agenda, and a community of ISVs is slowly forming.

\textsuperscript{3} [http://blogs.sapguides.com/2013/01/03/oltp-bi-in-a-single-hana-instance/]
\textsuperscript{7} [http://en.wikipedia.org/wiki/SAP_HANA]
\textsuperscript{8} [http://www.sap.com/corporate-en/factsheet]
SAP’s main challenge is to complete the development of HANA and catch up to decades of database development. The architectural choices that underlie HANA may also need rethinking in order to fulfill SAP’s vision of a single platform for OLTP and analytics, with all data resident in a column-format data store.
Microsoft SQL Server 2014

Summary: Microsoft claims in-memory support in SQL Server 2014 (released April, 2014) through three features; In-Memory OLTP for speeding up transactional workloads, columnstore and clustered columnstore indexes for read-only or read-mostly Data Warehouses.

Columnstore and clustered columnstore indexes are actually disk-based structures that are organized into compressed columns but still use the regular buffer cache for memory, which makes the “in-memory” label somewhat misleading. Data Warehouse applications should experience performance increases using these indexes because columnar formats and related compression reduce the amount of data that must be accessed for analytic queries. In this way, the use of memory is optimized but the approach is not fully “in-memory”. On the other hand, this approach is not limited by the amount of memory available as would be the case with a pure in-memory database.

In-Memory OLTP was designed to improve SQL Server scalability for high-volume transactional workloads. Tables and indexes are loaded into memory maintaining their row format and multi-version concurrency control, latch-free data structures and compiled stored procedures are employed. Similar techniques have been available for years in other database systems such as Oracle Database, for which scalability is not a weakness, nor are tables required to fit entirely into memory - a requirement of Microsoft’s In-Memory OLTP feature.

In-Memory OLTP, columnstore indexes and clustered columnstore indexes were implemented as independent projects, each with a significant list of restrictions and no integration across these features. As a result, a given table cannot be used for both In-Memory OLTP and columnstore processing in the same database. Although these features may improve the performance of specific use cases, broad adoption will likely require the removal of numerous restrictions on their usage as well as enabling the features to be used together.

Technology: In-Memory OLTP (code-named Hekaton) was a multi-year project at Microsoft culminating in the first release within SQL Server 2014. The In-Memory OLTP feature includes memory-optimized tables and native compilation of Transact-SQL stored procedures for accessing those tables. The overall objective of In-Memory OLTP is to maximize the performance of transactional workloads that access tables that are entirely resident in memory and running on modern multi-core processors. This required new data structures, optimistic locking algorithms and multi-version
concurrency control. Microsoft claims performance improvements up to 30x compared to SQL Server performance without In-Memory OLTP.

The constraints placed upon the use of In-Memory OLTP in SQL Server 2014 can be substantial. For instance, a database cannot use more than 250 GB of In-Memory OLTP data and the database server hardware should not exceed 4 processors and 60 compute cores, a modest-sized system in today’s world.\(^1\) If using compiled stored procedures, the list of restrictions is significant, including no triggers or constraints, such as foreign key or unique constraints.\(^2\) This means referential integrity must be enforced in application logic and could require significant code changes.

SQL Server 2012 introduced the columnstore index feature. It is a disk-based index in a compressed column format that renders the entire table read-only. The clustered columnstore index was added in SQL Server 2014 to add support for update, delete and insert operations. However, while they are technically updateable, Microsoft still does not recommend their use in transactional or mixed workload environments. The goal of running Data Warehouse queries directly against a high-volume production OLTP database is not possible in SQL Server 2014.

“Use a clustered columnstore index to improve data compression and query performance for data warehousing workloads that primarily perform bulk loads and read-only queries. Since the clustered columnstore index is updateable, the workload can perform some insert, update, and delete operations.”


**Strategy:** Microsoft’s strategy for IMDB technology has been erratic. The company first introduced in-memory technology into SQL Server 2008 R2, by way of the VertiPaq columnstore engine, which later became the xVelocity in-memory analysis engine and then simply a feature called columnstore indexes in SQL Server 2012. That evolved into clustered columnstore indexes in SQL Server 2014 and a completely separate but parallel effort for OLTP, called In-Memory OLTP aka In-Memory Optimization.\(^3\) The appearance is of separate teams with different agendas and no common strategy. As of today, Microsoft’s in-memory features have numerous restrictions and aren’t well integrated.

---

\(^3\) [http://searchsqlserver.techtarget.com/feature/In-memory-from-xVelocity-to-the-memory-optimized-columnstore-index](http://searchsqlserver.techtarget.com/feature/In-memory-from-xVelocity-to-the-memory-optimized-columnstore-index)
Validation: Microsoft’s In-Memory OLTP feature was recently released in April, 2014. As such, the volume of customers and references is understandably small, but the performance gains described are significant. These include companies doing online gaming, smart meter collection and currency trading.

Microsoft’s columnstore index technology has been available since the SQL Server 2012 release, supporting analytics. There is a scarcity of customer reference material on the use of columnstore indexes, which explains the frequent user commentary regarding the restrictions they imposed, leading to clustered columnstore indexes in SQL Server 2014.
IBM DB2 10.5 with BLU Acceleration

Summary: IBM released DB2 10.5 in June, 2013. A major feature of that release was BLU Acceleration (DB2 BLU), implemented as column-organized tables for high-performance analytics. Similar to Microsoft, column-organized tables are actually disk-based and not strictly “in-memory”, but can thus exceed the capacity of memory. Column-organized tables are updatable, but intended for read-mostly environments and employ the main technologies associated with in-memory databases.

In August, 2014, IBM added functionality to DB2 BLU (release “Cancun”) that enabled a table to be represented in both row and column formats, so that the same data could be used for both OLTP and analytics workloads. IBM’s data replication product is used to transfer changes from the row format to the column format. The release also added support for a variety of data types and DB2 features not previously supported, though a number of restrictions still remain. Notably, clustering via IBM pureScale is still not supported in DB2 BLU, so you cannot spread a column store across more than one server.

Overall, IBM’s approach with DB2 BLU covers many of the IMDB bases. It employs the principal in-memory technologies, supports both row and column formats for mixed workloads, and isn’t constrained by the size of memory. However, there are still many incompatibilities with existing DB2 features plus the use of replication to synchronize row and column stores adds contradictory performance overhead and excessive administration.

“To implement shadow tables in your environment, you must set up and configure all software components and replication, create the shadow tables, start replication to populate the shadow tables with data, and enable query routing to shadow tables.”

Bringing BLU Acceleration performance to OLTP environments with Shadow Tables: IBM Documentation

Technology: The key in-memory concepts that are embodied in DB2 BLU are column-organized tables, compression, SIMD vector processing and data skipping. Similar approaches are used in all major IMDB products for high-performance analytics. What separates DB2 BLU from SAP HANA and Microsoft SQL Server 2014 is the ability to represent the same table in both row and column formats.
within a common database. DB2 BLU does this with a “shadow table”; a new type of column-organized table that is derived from and synchronized with a corresponding row table. Shadow tables combine the pre-existing Materialized Query Table (MQT) with new column-organized table technology.

This approach is conceptually similar to that of Oracle’s Database In-Memory option, with its “dual-format” architecture. In the case of DB2 BLU, “shadow tables” are columnar representations of a row format table that are updated via IBM’s InfoSphere® Data Replication product. Shadow tables may contain a subset of the columns from the row format. Using a replication tool to synchronize the two formats can add significant latency, making it more likely that stale data could be returned from the column store. Users can set a latency threshold, above which a query is directed to the row store instead of the column store version. This architecture, though workable in some situations, is not sufficient to support real-time analytics against a high-volume OLTP database if the queries must return current data.

**Strategy:** DB2 BLU is not the first attempt at an IMDB from IBM. IBM Smart Analytics Optimizer for DB2 z/OS (2010) and Informix Warehouse Accelerator (2011) incorporated in-memory database features. Such features evolved from a project code-named Blink, and DB2 BLU is reputed to be a further evolution of the same project.

IBM Smart Analytics Optimizer has since been renamed DB2 Analytics Accelerator for z/OS (IDAA). It combines DB2 mainframe features with the Netezza data warehouse appliance (renamed IBM PureData System for Analytics) that uses a modified version of the open source database PostgreSQL. Evidence (or lack thereof) suggests IDAA has had minimal acceptance by IBM DB2 mainframe customers and could be replaced by DB2 BLU technology, or some variant, in the future. Data type incompatibilities between DB2 z/OS and the Netezza PostgreSQL database are notable issues. IBM has yet to rationalize its Netezza acquisition for Data Warehousing, as compared to the use of DB2 LUW (Linux, Unix and Windows), and DB2 on the mainframe.

Despite multiple attempts at an IMDB strategy, DB2 BLU is clearly IBM’s way forward for DB2 LUW, and may be the unifying approach across IBM’s DB2 variations in the future. Future releases of DB2 BLU are likely to focus on closing the remaining compatibility gaps with DB2, with the most notable issue being the lack of clustering support. For now it appears that Netezza (PureData System for Analytics) will not embrace IMDB technology in the near term and Informix databases will continue on a separate path.
Validation: IBM appears to be putting significant effort into popularizing the use of DB2 BLU and building an active user community. A reasonable amount of customer validation is beginning to appear for DB2 BLU and this critical mass may help to unify the many (sometimes competing) data management agendas at IBM. The outlier in IBM’s IMDB plans appears to be their acquisition of Data Warehousing appliance vendor Netezza, now renamed IBM PureData for Analytics. Once a highly visible competitor in Data Warehousing, the Netezza technology has largely stagnated and fallen behind alternatives, with the lack of IMDB support another example.
Oracle Database In-Memory

Summary: Oracle’s mainstream IMDB product was released in July, 2014 as the Oracle Database In-Memory option (In-Memory). Like other IMDB implementations, It features a column-format in-memory data store with compression, data skipping and SIMD vector processing to support real-time analytics. Additional features, such as in-memory aggregation and in-memory fault tolerance, distinguish Oracle Database In-Memory. Unlike other IMDB implementations, 100% of the Oracle SQL and data types are supported, allowing all existing applications to use Oracle Database In-Memory without any changes. In addition, all the functionality in the Oracle Database is inherited by the implementation of Oracle Database In-Memory, such as logging and recovery, data replication, security and clustering. Oracle Database In-Memory is available as a separately-priced option to Oracle Database 12c Enterprise Edition.16

Oracle Database In-Memory features a “dual-format” architecture to represent the same data in both row and column format, allowing the optimizer to determine the best format to satisfy each SQL request. Only specific columns from specific tables or just partitions of tables that require real-time analytics need to be defined for column-format representation. Real-time analytics can run directly against current production data in the column store while transactions continue to use the row store. Changes to the row store are consistently applied to the column store to ensure analytics queries are accessing current data. In addition, the in-memory store allows analytical indexes to be removed from the row store, speeding OLTP by removing the overhead of those index updates.

Oracle Database In-Memory works with any size database, regardless of how much data will fit in memory. Data larger than one server’s memory can span a storage hierarchy of memory, flash and disk, and can also be spread across a cluster of servers, each with its own memory and storage hierarchy. When used on an Oracle Engineered System platform, such as Oracle Exadata or Oracle SuperCluster, a unique fault tolerant feature automatically duplicates in-memory data on more than one server in the cluster for continual access to in-memory data should a cluster node fail.

Although Oracle Database In-Memory is compatible with all existing applications, yielding “out of the box” performance gains, even greater performance is possible when applications take full advantage of the potential of IMDB technology. For example, applications that perform set processing and parallelization can exploit in-memory technologies to the fullest. To that end, Oracle has announced a series of new “in-memory” modules for performance-critical functions in applications such as Oracle E-

Business Suite, Oracle PeopleSoft, Oracle JD Edwards, and Oracle Siebel. End user and partner experiences with Oracle Database In-Memory were also described at the June 2014 launch of the product.

In addition to Oracle Database In-Memory, Oracle TimesTen In-Memory Database has been available since 2005 for embedded real-time OLTP applications. Such applications are typically used inside communications networks, financial trading and risk management systems. A key differentiator for TimesTen is the elimination of network latency by embedding the IMDB within the application. By doing so, response times for applications using TimesTen can be measured in microseconds, whereas the addition of a network adds a millisecond or more to the response time.

**Technology:** Oracle Database In-Memory is pure IMDB technology seamlessly integrated into Oracle Database 12c using Oracle’s “dual-format” architecture. The Oracle optimizer decides when to use the in-memory column store and when to use the traditional Oracle row store. Changes to the data are applied to the row store and immediately reflected in the column store via lightweight mechanisms. The in-memory column store also implements in-memory compression, data skipping via Oracle in-memory storage indexes, in-memory scans, joins (via bloom filters) and aggregation, and SIMD vector processing. In short, the latest technologies and techniques for real-time analytics are contained in Oracle Database In-Memory.

The dual-format architecture enables the same production database to service both high-volume transactions and real-time analytics, avoiding the need to copy the production database to another server. The time and overhead of creating the copy is eliminated and the reports and analytic queries access the very latest production data. Since the in-memory column store now services the analytical requests, it’s possible to drop previous analytic indexes from the row store and improve OLTP performance as well. All of this is accomplished with no application changes.

**Strategy:** Oracle has been in the IMDB business longer than any major database vendor by virtue of the TimesTen In-Memory Database acquired by Oracle in 2005. It can also be argued that the Oracle Database In-Memory option is the first mainstream IMDB product that is a complete database implementation. As was detailed, the other implementations have significant restrictions, incompatibilities or incomplete functionality, relative to a mature database system. The strategy Oracle

has taken, making ease of use and 100% compatibility a priority, should result in rapid and broad adoption across Oracle’s large customer base.

**Validation:** A number of customers and partners joined in the launch of Oracle Database In-Memory in June, 2014, after an extensive beta test period. Oracle applications teams have already announced new in-memory application modules and the Oracle PartnerNetwork’s Oracle Database Ready certification now includes Oracle Database In-Memory. End-user deployment references will soon appear.

For Oracle Database customers, testing the impact of Oracle Database In-Memory is trivial. Install Oracle Database 12c, enable in-memory data and identify the data to be populated in the column store. Existing production databases can be tested for real-time analytics and concurrent OLTP. On Oracle Engineered Systems platforms, such as Oracle Exadata, in-memory fault tolerance is available to ensure uninterrupted real-time processing.

---

## IMDB Product Comparison Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Oracle Database In-Memory</th>
<th>SAP HANA</th>
<th>Microsoft SQL Server 2014</th>
<th>IBM DB2 BLU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial release date</td>
<td>July 2014</td>
<td>November 2010</td>
<td>April 2014</td>
<td>June 2013</td>
</tr>
<tr>
<td>Columnar</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compression</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SIMD vector processing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data skipping</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>In-memory aggregation</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-Memory OLTP optimizations</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Size not limited by DRAM</td>
<td>✓</td>
<td>4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cluster (scale-out) support</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Dual-format in one database</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Consistent updates</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>100% application transparency</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comprehensive availability support</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>