In-Memory-Databases: An In-Depth Dive Into SAP HANA

Bachelorarbeit zur Erlangung des akademischen Grades
Bachelor of Science
im Studiengang Informatik
an der Fakultät 10 für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

vorgelegt von: Fatima Sabra
Matrikel-Nr.:
Adresse:

eingereicht bei: Prof. Dr. Birgit Bertelsmeier
Zweitgutachterin: Prof. Dr. Heide Faeskorn-Woyke

Köln, 29.04.2024

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder der Verfasserin/des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Köln, 29.04.2024	Fatima Sabra
Ort, Datum	Unterschrift

Abstract

In the contemporary era, many organizations and companies are confronted with a significant surge in data volumes. This has led to the challenge of capturing, storing, managing, and analyzing terabytes of data, which are stored in diverse formats and originate from numerous internal and external sources. Furthermore, the emergence of novel applications, such as trading, and artificial intelligence, has made the processing of vast amounts of data in real time an absolute necessity. These requirements exceed the processing capacity of traditional on-disk database management systems, which are ill-equipped to manage this data and to provide real-time results. Therefore, data management requires new solutions to cope with the challenges of data volumes and processing data in real time. An in-memory database system (IMDB- or IMD system) is a database management system that is emerging as a solution to these challenges, with the support of other technologies. IMDBs are capable of processing massive data distinctly faster than traditional database management systems. This work examines the approach of IMDBs, with a particular focus on SAP HANA, and compares it with other IMDBs.

Key words: In-memory database, IMDB, IMD, SAP HANA.

Table of Contents

List of Tables

List of Figures

Introduction

In the early days of computing, the primary focus was optimizing the use of main memory (RAM), which was both expensive and limited in capacity[1]. For on-disk databases, the main bottleneck is their input/output (I/O) operations, which are very slow and can't be optimized beyond a specific limit due to their mechanical nature. Traditional on-disk Database management systems (DBMSs) have tried to improve this by introducing various caching techniques of frequently accessed data, however, it came at the cost of synchronizing the cache with the disk and vice versa, furthermore, implementing various complex transaction and resource management logic, is itself a performance constraint[2]. The continued development of semi-conductor memory, specifically non-volatile memory, has been a significant advancement as it ensures that data stored in memory remains intact even in the event of a power failure[3].

Domdouzis *et al.*[4] defined in-memory databases systems (IMDBS) (also main memory databases (MMDBs)[5]) as: “database management systems where the data is stored entirely in main memory.”, this implies that each time a database is queried, or updated, the only component accessed is the main memory. Thus, disk involvement is avoided, which is beneficial as the main memory is considerably faster than any disk. However, to prevent data loss in the event of a power outage, for instance, these databases are designed persistent by keeping data in memory all while recording each operation in a transaction log, which is then saved on a disk.

Some of the features and advantages of using IMDBs include speed, high performance, reduced complexity, data persistence and scalability. However, all this comes with the increased requirement for memory compared to the disk-based database variant which might lead to issues in managing large datasets that exceed available capacities.

In practice, there is no dominant technology, with multiple options available such as H2, IBMs solidDB, Oracles TimesTen, Microsoft SQL Server, and SAP HANA, the latter being the focus of this work.

---

[1] Domdouzis *et. al.*, 2021, P. 37

[2] Gupta *et. al.* 2014, P. 333

[3] Domdouzis *et. al.*, 2021, P. 37

[4] Domdouzis *et. al.*, 2021, P. 189

[5] Garcia & Salem, 1992, P. 509

1 Databases and Memory Systems

Behind every website that provides any kind of information is a database, whether it’s Google, Amazon, or your local department store, if they provide information on their website, then they are operating a database.

In-Memory Database managements system is a database that stores data on the main memory, while the traditional DBMS moves data from disk to memory using cache or buffer whenever it is accessed[6].

In this section, database storage techniques mainly used in relational IMDBs will be discussed.

1.1 Database Management System

Alvarez *et. al.* described a database as “a collection of information that exists over a long period of time, often many years.” [7]. According to Alvarez *et. al.* a DBMS is expected to allow users to create new databases, gives them the ability to query the data, provides a magnitude of storage space, enables durability, and allows controlled access of many users simultaneously[8]. An outline of a complete DBMS is seen in Figure 1.

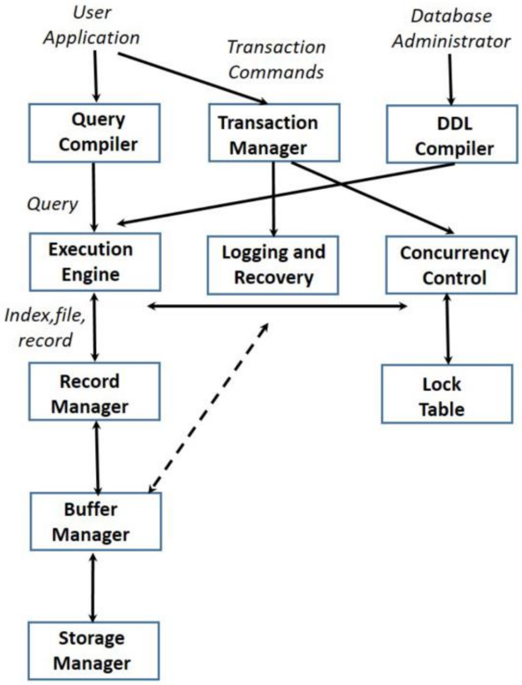


Figure 1: Outline of a DBMS (source: Alvarez *et. al.* (2022), P. 59)

---

[6] Domdouzis *et. al.*, 2021, P. 37

[7] Alvarez *et. al.* 2022, P. 57

[8] Alvarez *et. al.* 2022, P. 57-58

1.2 In-Memory Databases (IMDBs)

In this section we will have a closer look at IMDBs, their history, functions, and their pros and cons.

1.2.1 What are IMDBs?

An IMDB is a database that has its data stored in main physical memory[9]. While all operations are performed in memory, it is possible to have a backup copy of the database saved on a disk, and a log of the transactions is kept, ensuring data integrity in case of power failure[10].

1.2.2 History of IMDBs

Figure 2 shows a timeline of databases.

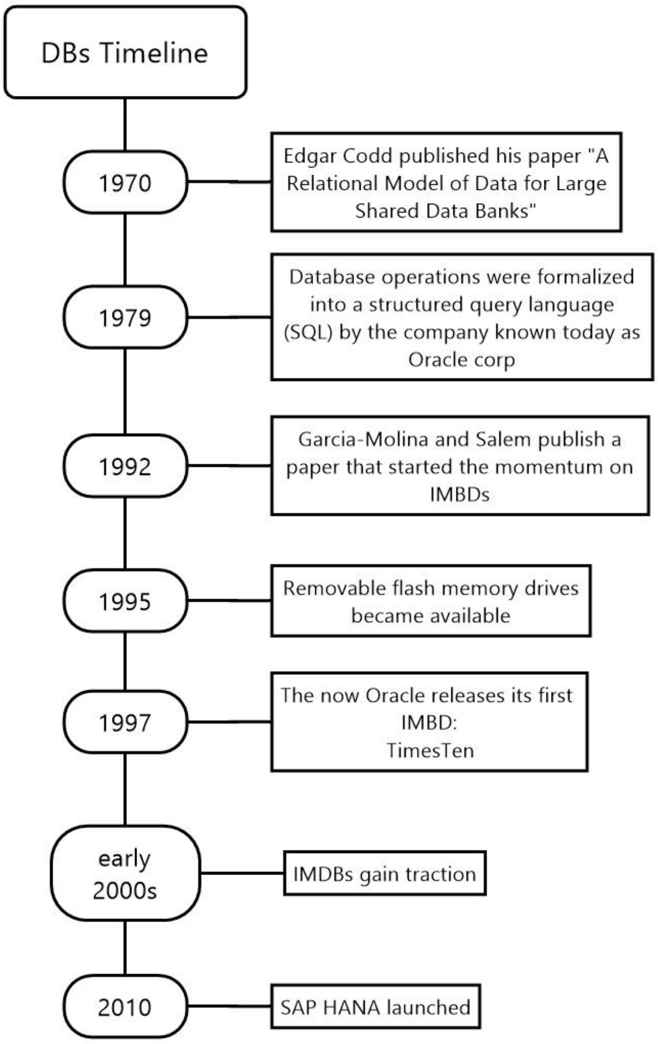


Figure 2: timeline of databases (source: own illustration, based on: Lake & Crowther (2013) and SAP website (What is SAP HANA?))

---

[9] Domdouzis *et. al.*, 2021, P. 37

[10] Alvarez *et. al.* 2022, P. 62

1.2.3 Main Memory vs. Magnetic Disks

The main memory of a computer has different properties than that of magnetic disks, these differences have several implications on the design and performance of the database system[11,12]:

1) The access time for main memory is significantly lower than that of disk storage.
2) Main memory is typically volatile, while disk storage is not. However, it is possible to construct nonvolatile main memory.
3) Disks have a high, fixed cost per access that does not depend on the amount of data that is retrieved during the access. For this reason, disks are block-oriented storage devices. Main memory is not block oriented.
4) The layout of data on a disk is critical compared to the layout in main memory since sequential access to a disk is faster than random access. Sequential access is not as important in main memories.
5) Main memory is typically accessible by processors directly, while disks are not. This may make data in main memory more susceptible to software errors than disk resident ones.

1.2.4 Uses of IMDBs

The use of IMBDs is dependent on whether a system requires to perform online analytical processing (OLAP) and online transaction processing (OLTP) simultaneously and in real-time. A study performed in 2011, found that IMBDs are used by companies for the following applications[13]:

- Business Analytics.
- Web-based transactions.
- Reporting.
- Finance (trading, market data, etc.).
- Billing and Provisioning.
- Embedded/Mobile applications.

---

[11] Garcia & Salem, 1992, P. 509

[12] Alvarez *et. al.* 2022, P. 62-63

[13] Lake & Crowther, 2013, P. 193

- Supply chain management.
- Sensor data management.
- Geo-spatial applications.
- Network information management.
- Product information management.
- Online advertising and materials management.

1.2.5 In-memory databases vs. traditional databases

As previously discussed, an IMDB is a database that stores data on the main memory, while the traditional DBMS moves data from disk to memory using cache or buffer whenever it is accessed[14]. Compared to traditional DBs, IMDBs are faster, providing access in real-time to multiple clients simultaneously, however, since the data is stored on main memory it makes it more susceptible to loss[15].

1.2.6 Advantages and Disadvantages of IMDBs

Some of the advantages of using IMDBs is the improved access speed that is resulted from the use of RAM. This also allows for faster data analysis, IMDBs enable the evaluation of structured and unstructured data from any system. The use of distributed data infrastructures enables the storage of unstructured data in an IMDB, where several processing units work in parallel to perform a single task and distribute it to different server clusters. This results in a higher storage capacity, faster processing, and improved transfer speed for unstructured data[16].

While the use of RAM provides a faster access to data, it also brings another set of issues along, for example should the system suffer from power outage all data stored would be lost, this is solved by persisting the data to a non-volatile storage or a battery powered main memory, another design approach to address this problem is high availability implementation, which involves replicating the data to other available nodes in real time; which will in turn reduce the probability of complete data failure[17] another option is using

---

[14] Domdouzis *et. al.*, 2021, P. 37

[15] Gupta *et. al.* 2014, P. 334

[16] Jadhav *et. al.*, 2020

[17] Gupta *et. al.* 2014, P. 335

transaction logging, this means only operations of the updated data are logged, and then combined with group commits[18].

1.2.7 IMDBs Misconceptions

1.2.7.1 Populating the Databsase

A popular misconception is that populating a very large in-memory database is a slower process than populating an equivalent database on disk. On disk systems employ memory caches to accelerate the process; however, the cache eventually becomes full and must be written to disk. This necessitates transferring data from the cache to I/O buffers. Subsequently, data is physically written to disk, which is a relatively slow process compared to other operations. In addition, data is written to disk indexes, which map the data's storage location. As more data is added to the system, these indexes grow and become more complex. Furthermore, as the size of the database increases, the percentage of data that can be held in cache memory decreases, further reducing performance. Finally, as the database grows, the physical size on the disk increases, resulting in higher seek times. In essence, as the size of the database increases, there is a consistent decline in performance. Conversely, the performance of an in-memory database remains relatively stable as more data is added, as the on-disk issues do not apply[19].

1.2.7.2 Single User, Single System

This is a huge misconception, as IMDBs are not restricted to a single user and the database is held in shared memory which can also be operated on remote servers, meaning, multiple databases and multiple users[20].

1.2.7.3 An In-Memory Database Is the Same as an Embedded Database

Not completely false, as there are databases that are similar to embedded databases, if not the same, however IMBDs can use the client server model with the use of shared memory[21].

---

[18] Tan *et. al.*, 2015

[19] Lake & Crowther, 2013, P. 187

[20] Lake & Crowther, 2013, P. 187-188

[21] Lake & Crowther, 2013, P. 188

1.2.8 Popular IMDBs

Several IMDB solutions have risen in the last couple of decades, these include both commercial and open-source databases, this section will only touch on them briefly.

1.2.8.1 Commercial IMDBs

1. Oracles TimesTen: it is a completely persistent database, it has full ACID (atomicity, consistency, isolation, and durability) properties[22].
2. IBMs SolidDB: it is a relational in-memory database server that contains two optimized engines; a disk- and main-memory based engines[23].
3. SAP HANA: supports both column and row oriented physical representations in a hybrid engine[24].

1.2.8.2 Open-source IMDBs

1. SQLite: a small (lite), fast and reliable IMDB, it is often deployed in embedded systems, such as mobile phones[25].
2. CSQL: a single table database, that offers high performance on simple SQL queries and DML statements[26].

The next chapter will cover SAP HANA IMDB in detail.

---

[22] Lahiri *et. al.*, 2013

[23] Lindström *et. al.*, 2013

[24] Lee *et. al.*, 2013

[25] Gupta *et. al.* 2014, P. 335

[26] Gupta *et. al.* 2014, P. 335-336

2 SAP HANA

This section will contain an in-depth look at SAP HANA and its features.

2.1 What is SAP HANA?

SAP HANA is an IMDB developed and marketed by SAP SE. There are multiple components of SAP HANA which are include SAP HANA Database; which refers to the core database, and SAP HANA Studio (S/4HANA); which is the modelling tool offered by SAP[27]. SAP HANA integrates both OLTP and OLAP, and unifies structured, semi-structured, and unstructured data processing. Data is kept in memory under the condition that there is enough space, otherwise the entire data is unloaded until the time they are needed again[28].

The structure of the HANA In-memory database is shown in **Error! Reference source not found.**.

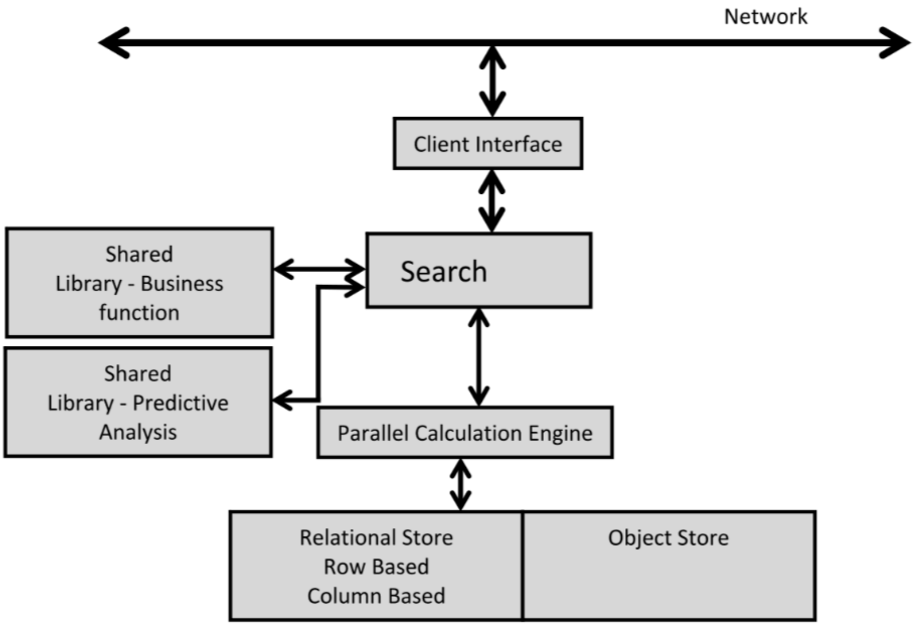


Figure 3: SAP HANA In-memory database (source: Lake & Crowther (2013))

---

[27] Jadhav *et. al.*, 2020

[28] Alvarez *et. al.* 2022, P. 98

2.2 What is SAP?

SAP is a German software company founded in 1972, initially called “**S**ystem **A**nalysis **P**rogram Development” or “**S**ystem**A**nalyse **P**rogrammentwicklung” in German, now simply abbreviated to SAP[29].

2.3 History of SAP HANA

A brief timeline of SAP HANA is illustrated in Figure 4.

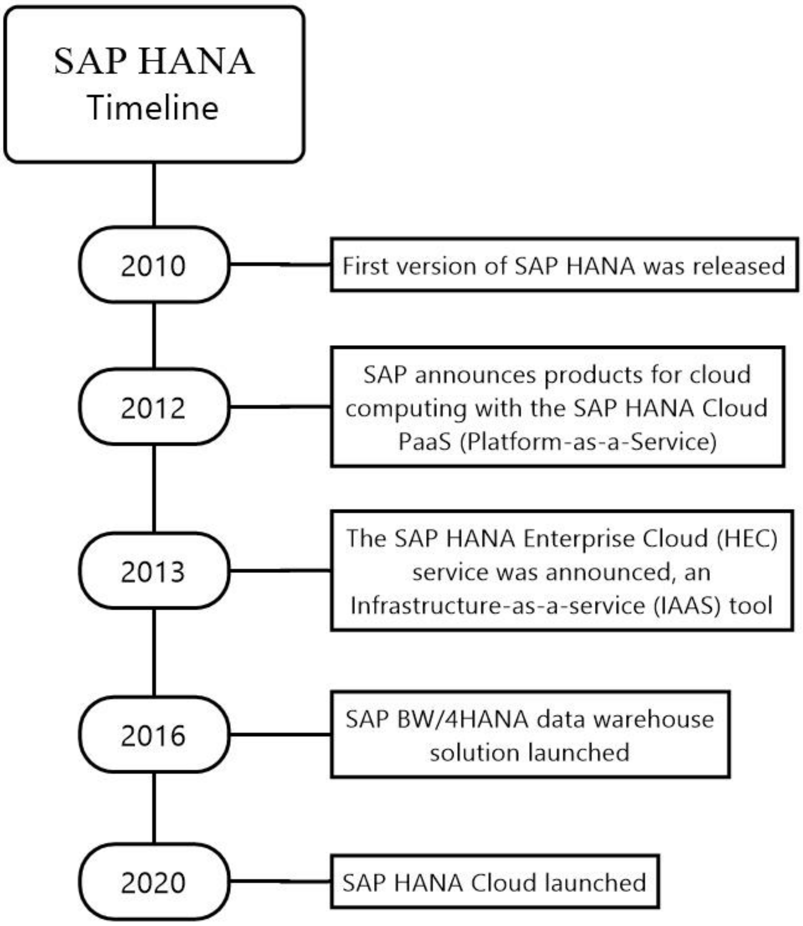


Figure 4: timeline of SAP HANA (source: own illustration, based on: SAPs official website (What is SAP HANA?))

---

[29] What is SAP? (SAPs official website)

2.4 Key Features of SAP HANA

SAP HANA combines a full database management system with both a standard SQL interface and ACID, it also exploits multicore processors to optimize native SQL commands[30]. Some of SAP HANAs key features include[31]:

1. The support of both row- and column-oriented stores for relational data.
2. Offers multiple query language interfaces, allowing for a high data analysis functionality, which reduces high data transfer cost.
3. It provides support for temporal queries based on the timeline index, as data is versioned in HANA.
4. It provides snapshot isolation based on multi-version concurrency control.
5. It provides transaction semantics based on optimized two-phase commit control.
6. It provides fault-tolerance by periodically logging and checkpointing into the general parallel file system (GPFS).

2.4.1 Relational Stores

While SAP HANA supports both row- and column-oriented physical representations of relational tables, each has its own merit; row store is advantageous for heavy updates and inserts, as well as point queries that are common in OLTP, while the column store is optimal for OLAP applications, as they typically access all values of a column together, and few columns at a time. Additionally, column-oriented representation can more effectively and efficiently utilize compression techniques. In SAP HANA, a table or partition can be configured to be either in the row store or in the column store, and it can also be restructured from one store to the other[32].

Column Versus Row-Based Storage:

The use of column or row storage is recommended based on the needs of the client; row-based storage is recommended when the table has a small number of rows, records are singerly processed, columns have distinct values, fast operations are not necessary or when the full record is needed. Column-based storage is recommended when there is a restricted number of columns required for calculations or searches, a table has many

---

[30] Lake & Crowther, 2013, P. 190

[31] Alvarez *et. al.* 2022, P. 98

[32] Alvarez *et. al.* 2022, P. 98-99

columns or records where column-operations are required, and if most columns have distinct values in contrast to number of rows[33].

In SAP HANAs column store, data is stored in read- and write-optimized main and delta fragments respectively[34]. For token generation, the transaction manager keeps track of: (i) unique transaction IDs (TID), (ii) the state of each transaction, i.e., open, aborted, or committed, and (iii) once the transaction is committed, a commit ID (CID) for all write transaction[35], as seen in Figure 5.

Since single updates are expensive each table has delta storage, which balances high update rates and good read performance[36]. Whether inserting new rows or updating existing ones, both operations are regarded as changes. Changes do not physically modify existing rows but rather append new rows into the delta fragments. All committed row changes from the delta fragment are moved into a newly constructed main fragment during the delta merge operation. Each query on a column is evaluated independently on the main and on the delta fragment of the column. The two result sets are then united and returned, with some rows removed after applying proper row visibility rules. In the main fragment, a data vector is used per column to store which row position contains what value identifier from the column's dictionary.[37]

---

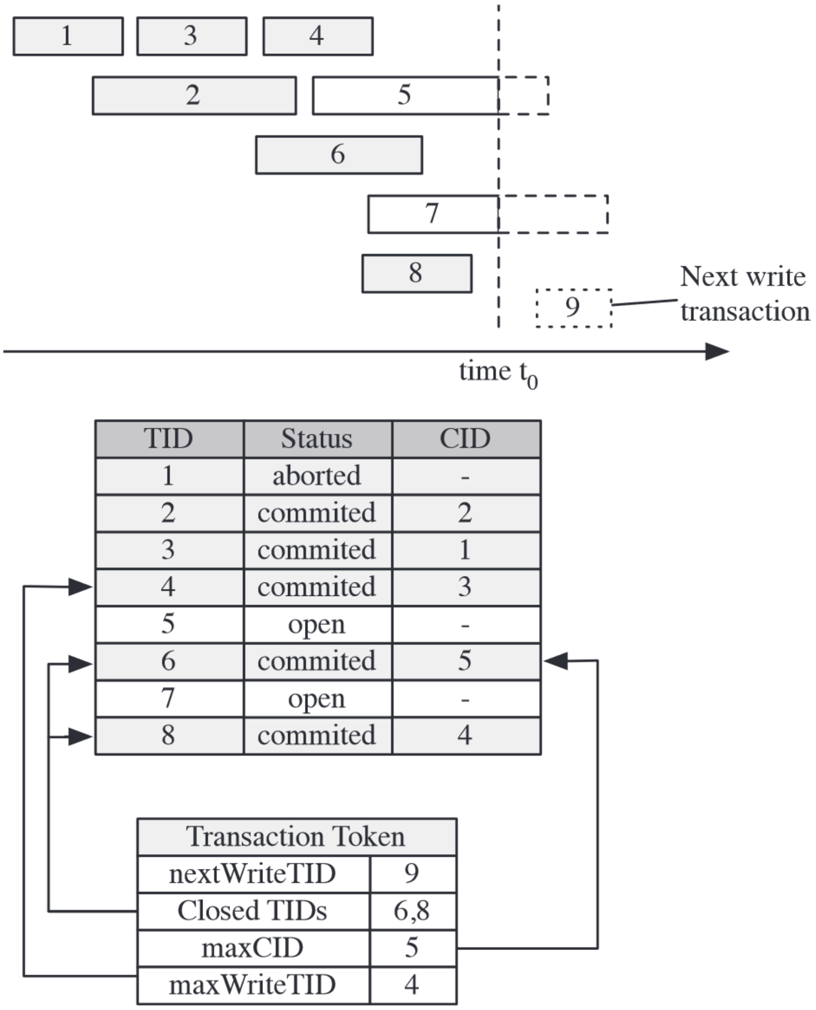
[33] Lake & Crowther, 2013, P. 191-192

[34] Nica *et. al.* 2017, P. 1959

[35] Lee *et. al.* 2013, P. 29

[36] Färber *et. al.* 2012, P. 4

[37] Nica *et. al.* 2017, P. 1959



TID	Status	CID
1	aborted	-
2	commited	2
3	commited	1
4	commited	3
5	open	-
6	commited	5
7	open	-
8	commited	4

Transaction Token	
nextWriteTID	9
Closed TIDs	6,8
maxCID	5
maxWriteTID	4

Figure 5: Data Consolidation with Transaction Token (source: Lee *et. al.* (2013))

2.4.2 Analytical Query Processing

Column-stores are well suited for analytical queries on massive amounts of data, the SAP HANA DB column store uses efficient compression to provide high read performance in combination with cache and parallel algorithms[38].

2.5 Advantages of SAP HANA

One key asset of the SAP HANA DB is its ability to execute business and application logic inside the database kernel. For this purpose, the calculation engine provides an abstraction of logical execution plans, called calculation models[39].

---

[38] Färber *et. al.* 2012, P. 4

[39] Färber *et. al.* 2012, P. 3

SAP HANA supports the representation of application-specific business objects and logic directly inside the database engine, it is optimized to communicate efficiently between the data management and application layers[40].

2.6 Disadvantages of SAP HANA

One of the downsides of using SAP HANA is the need to invest time and money updating to newer versions as newer versions and updates are released frequently and older version this becoming outdated[41].

---

[40] Sikka *et. al.* 2012, P. 2

[41] Wagh, 2020

S/4HANA

S/4HANA is the next-generation SAP Business Suite application created for the SAP HANA Platform, its main function is to simplify business processes, improve transaction efficiency and provide faster analytics. [42]

Figure 6 shows the components of the SAP HANA Appliance.

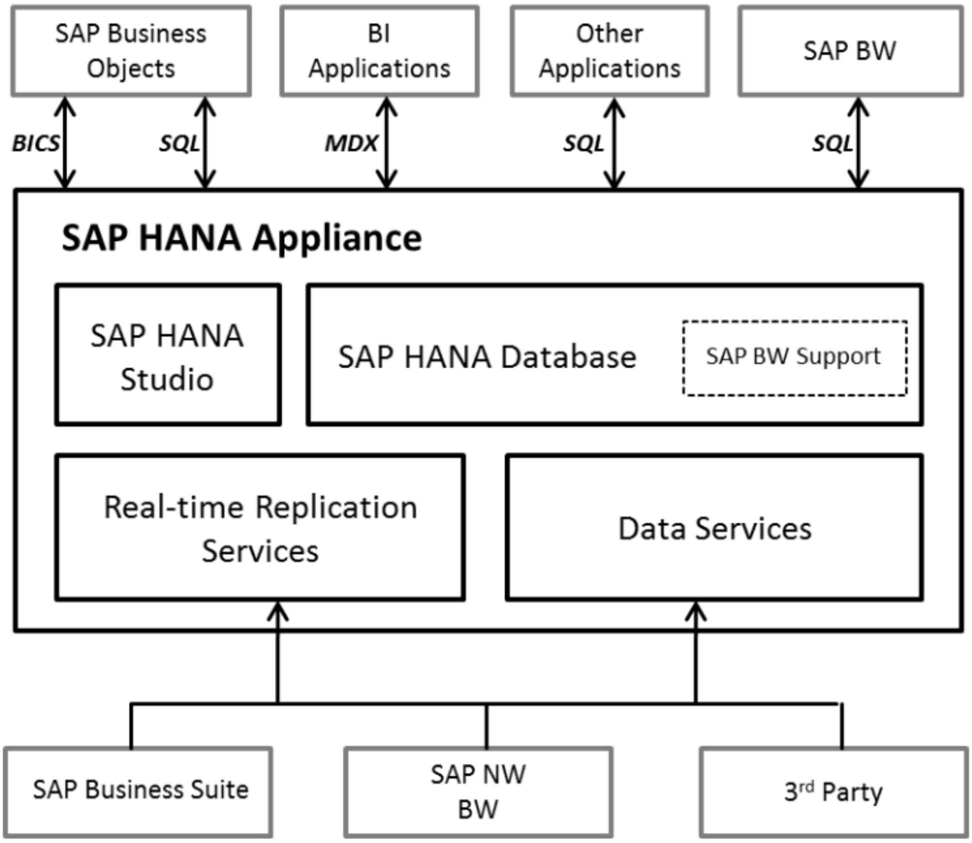


Figure 6: Components of the SAP HANA Appliance (source: Sikka *et. al.* (2012))

---

[42] Pattanayak 2017, P. 559

3 SAP HANA vs. Oracles TimesTen

Shah & Patel[43] performed a comparison between SAP HANA and Oracles TimesTen, they examined four major criteria: data loading, logical data layout, data processing, and implementation infrastructure.

Table 1: Data loading (source: Shah & Patel. 2016)

Feature	Oracle TimesTen	SAP HANA
support bulk loading of data	Yes	Yes
Allow for a streaming load of data from an API or other sources	Yes	Yes
Data load facility via a GUI	No	Yes

Table 2: Logical Data Layout (source: Shah & Patel. 2016)

Feature	Oracle TimesTen	SAP HANA
Organize data with a row-based storage strategy	Yes	Partial
Organize data with a column-based storage strategy	Partial	Yes
Data definition / Schema information configured via a GUI	No	Yes

Table 3: Data Processing (source: Shah & Patel. 2016)

Feature	Oracle TimesTen	SAP HANA
Transaction consistency based on ACID concept	Yes	Yes
utilizes MPP	Partial	Yes
Works with RDBMSs	Yes	Partial

Table 4: Implementation Infrastructure (source: Shah & Patel. 2016)

Feature	Oracle TimesTen	SAP HANA
Unix/Linux installation is supported by solution	Yes	Yes
Windows installation is supported by solution	Yes	Yes
Public cloud implementation is supported by solution	No	Yes
Private cloud implementation is supported by solution	Yes	Yes

---

[43] Shah & Patel. 2016 P.928-929

A further comparison is between Figure 3: SAP HANA In-memory database (source: Lake & Crowther (2013)) and seen below.

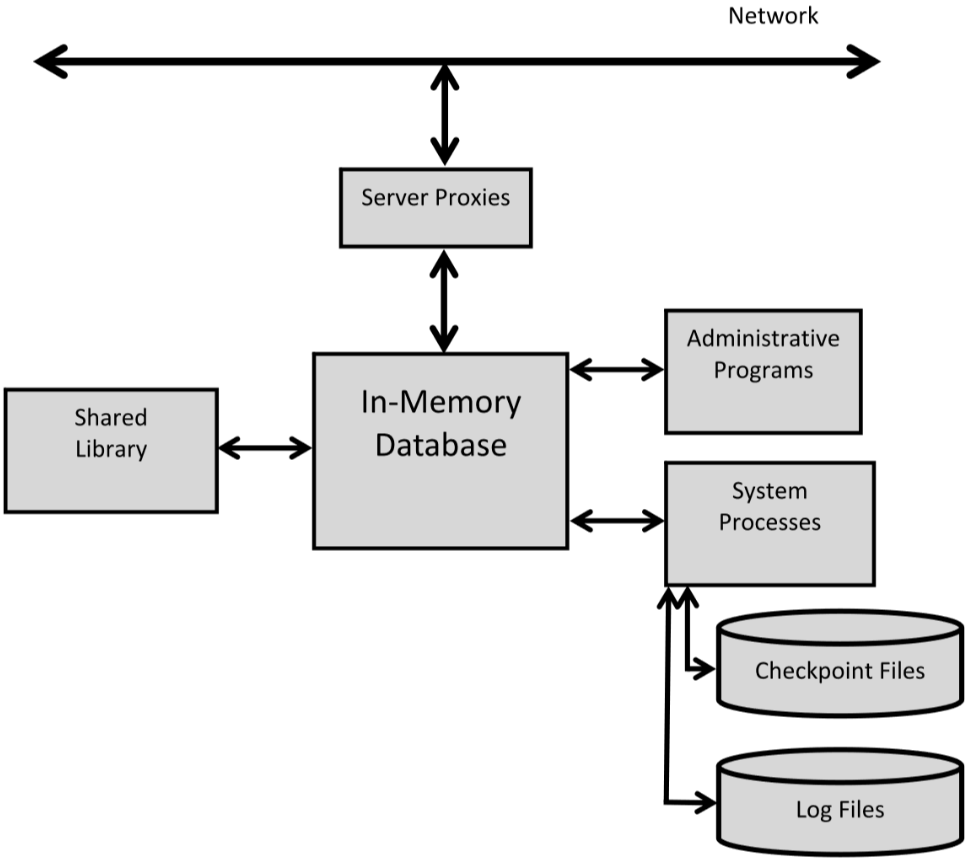


Figure 7: Oracle TimesTen in-memory database (source: source: Lake & Crowther (2013))

4 SAP HANA vs. IBMs SolidDB

In this section we used Shah & Patel[44] comparison criteria (data loading, logical data layout, data processing, and implementation infrastructure.) between SAP HANA and Oracles TimesTen, to do a comparison between SAP HANA vs. IBMs SolidDB.

Table 5: Data loading (source: Shah & Patel. 2016 and Lindström *et. al*. 2013)

Feature	**IBMs SolidDB**	**SAP HANA**
support bulk loading of data	Yes	Yes
Allow for a streaming load of data from an API or other sources	Yes	Yes
Data load facility via a GUI	No	Yes

Table 6: Logical Data Layout (source: Shah & Patel. 2016 and Lindström *et. al*. 2013)

Feature	**IBMs SolidDB**	**SAP HANA**
Organize data with a row-based storage strategy	Yes	Partial
Organize data with a column-based storage strategy	Yes	Yes
Data definition / Schema information configured via a GUI	No	Yes

Table 7: Data Processing (source: Shah & Patel. 2016 and Lindström *et. al*. 2013)

Feature	**IBMs SolidDB**	**SAP HANA**
Transaction consistency based on ACID concept	Yes	Yes
utilizes MPP	Yes	Yes
Works with RDBMSs	Yes	Partial

Table 8: Implementation Infrastructure (source: Shah & Patel. 2016 and Lindström *et. al*. 2013)

Feature	**IBMs SolidDB**	**SAP HANA**
Unix/Linux installation is supported by solution	Yes	Yes
Windows installation is supported by solution	Yes	Yes
Public cloud implementation is supported by solution	Partial	Yes
Private cloud implementation is supported by solution	Partial	Yes

---

[44] Shah & Patel. 2016 P.928-929

5 Conclusion

In memory databases have many applications and types, they are very useful tools that have concreted themselves in all aspects of business and the world in general. There are many tools that employ IMDBs, SAP HANA being one of them, this work has barely scratched the surface of the depths and magnitude of what SAP HANA can do, especially as the number of independent reviews and work is limited. SAP HANA is without doubt a powerful tool, but as mentioned there are many challenges facing it and other IMDBs.

References

Alvarez P.M., Ayala M.L., and Cisneros S.O. (2022): **Main Memory Management on Relational Database Systems.** Springer Nature.

Domdouzis, K., Lake, P., and Crowther, P. (2021): **In-Memory Databases. In: Concise Guide to Databases.** Undergraduate Topics in Computer Science. Springer.

Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., & Lehner, W. (2012): **SAP HANA database: data management for modern business applications.** ACM Sigmod Record, 40(4), 45-51.

Färber, F., May, N., Lehner, W., Große, P., Müller, I., Rauhe, H., & Dees, J. (2012): **The SAP HANA Database--An Architecture Overview.** IEEE Data Eng. Bull., 35(1), 28-33.

Garcia-Molina, H., & Salem, K. (1992): **Main memory database systems: An overview.** *IEEE* Transactions on knowledge and data engineering, 4(6), 509-516.

Gupta, M.K., Verma, V., & Verma, M.S. (2014): **In-memory database systems-a paradigm shift.** International Journal of Engineering Trends and Technology (IJETT), 6(6), 2013.

Jadhav, A., Dsouza, D., Khan, T., and Shaikh, P. (2020): **Design and Implementation of In-Memory Database System for Efficient Information Retrieval.** Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT) 2020, Available at SSRN: https://ssrn.com/abstract=3686068 or http://dx.doi.org/10.2139/ssrn.3686068

Lahiri, T., Neimat, M. A., & Folkman, S. (2013): **Oracle TimesTen: An In-Memory Database for Enterprise Applications**. IEEE Data Eng. Bull., 36(2), 6-13.

Lake, P., & Crowther, P. (2013): **Concise guide to databases: A Practical Introduction.** London: Springer.

Lee, J., Muehle, M., May, N., Faerber, F., Sikka, V., Plattner, H., Krueger, M., & Grund, M. (2013). **High-Performance Transaction Processing in SAP HANA.** IEEE Data Eng. Bull., 36(2), 28-33.

Lindström, J., Raatikka, V., Ruuth, J., Soini, P., & Vakkila, K. (2013): **IBM solidDB: In-Memory Database Optimized for Extreme Speed and Availability.** IEEE Data Eng. Bull., 36(2), 14-20.

Lindström, J., Raatikka, V., Ruuth, J., Soini, P., & Vakkila, K. (2013): **IBM solidDB: In-Memory Database Optimized for Extreme Speed and Availability.** IEEE Data Eng. Bull., 36(2), 14-20.

Nica, A., Sherkat, R., Andrei, M., Cheng, X., Heidel, M., Bensberg, C., & Gerwens, H. (2017): **Statisticum: data statistics management in SAP HANA.** Proceedings of the VLDB Endowment, 10(12), 1658-1669.

Pattanayak, A. (2017): **SAP S/4HANA embedded analytics: an overview**. Journal of computer and communications, 5(9), 1-7.

Shah, A., & Patel, N. M. (2016): **The Impact of In-memory Databases over Multitenant Architecture.** International Journal of Computer Science and Information Security (IJCSIS), 14(9).

Sikka, V., Färber, F., Lehner, W., Cha, S. K., Peh, T., & Bornhövd, C. (2012): **Efficient transaction processing in SAP HANA database: the end of a column store myth.** In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 731-742.

Tan K.L., Cai Q., Ooi B.C., Wong W.F., Yao C., Zhang H. (2015): **In-memory databases: Challenges and opportunities from software and hardware perspectives.** ACM Sigmod Record. 44(2), 35-40.

Wagh A. (2020): https://digitalconqurer.com/dev-corner/pros-and-cons-of-sap-hana/ (last access: 28.04.2024)

What is SAP HANA? https://www.sap.com/products/technology-platform/hana/what-is-sap-hana.html (last accessed: 28.04.2024)

What is SAP? https://www.sap.com/about/what-is-sap.html (last accessed: 28.04.2024)