



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Performance Analysis and Architectural Overview of SAP HANA

Ayushi Chhabra¹, Shwetank Sharma², Sanjay Ojha³
Centre for Development of Advanced Computing, Noida, India

Abstract: - The traditional database systems drawback from the feature of greater access time as the data is to be read from the disk due to limited memory structures. Also, the transactional and analytical data could not be used together on the same database management system. With the introduction of the in memory computing technology, the data can be stored on memory through compression techniques and can be accessed directly from the memory without any intervention of the disk. This concept is the core of SAP HANA Database, which is a component of the overall SAP HANA Appliance. Also, the SAP HANA Database supports both the transactional and analytical processing as it gives the representation of data both row-wise and column-wise.

Key-Words: - SAP HANA, ERP, HANA Architecture, HANA Database Architecture, Database, DBMS, HANA Studio, OLAP and OLTP.

I.Introduction

Traditional database management systems were designed such that they optimize the performance of the disk, however, main memory was still a constraint on the overall performance of the database management systems. The reason lies in the fact that the memory could only be used to store the frequently used data, however if the data is in large amounts, disk reads are generally required. Simply accessing and reading the data from the disk can take a significant amount of time. Fig.1 [1] shows the access and read times of disk and memory:

Action	Time
Main Memory Access	100 ns
Read 1 MB Sequentially from Memory	250,000 ns
Disk Seek	5,000,000 ns
Read 1 MB Sequentially from Disk	30,000,000 ns

Fig.1 Access and Read times of Disk and Memory

By optimizing the disk access, the number of disk pages to be read into main memory during processing could be minimized, however the amount of data that could be stored on the disk was still limited, as a disk cannot store much bigger data. It has been analysed [2] in early 2011 that there would be variety in data, with 80% enterprise data being unstructured, increase in the volume of the data by 1200 Exabyte's in 2011 and the velocity with which the digital data content will double would be tremendously higher.

The data could be regarded as transactional or analytical. Transactional data would be, for instance related to searching, inserting into or updating the database. The analytical data would be called so if analysis is done to obtain a certain data, for instance, sales analysis across a specific region. The On-Line Transaction Processing (OLTP) uses row-based database systems while column-stores have become more popular for On-Line Analytical Processing (OLAP).

Ideally, a database should be such that it can manage both the OLTP and OLAP type of workloads into a single system, which the traditional databases aren't able to do. Besides the processing of analytical and transactional applications in one database management system, the other major problem with the databases is read consistency, which implies that the user should see the data as it was when the query was executed even if the row got changed in the meantime.

The SAP HANA database is designed from the ground up around the idea that memory is available in abundance, considering that roughly 18 billion gigabytes or 18 Exabyte's are the theoretical limits of memory capacity for 64-bit systems, and that I/O access to the hard disk is not a constraint. SAP HANA Database is the component of SAP HANA Appliance that provides a foundation for data management for renovated and newly developed SAP applications. SAP HANA Platform Architecture is discussed in Section 2. Now, instead of optimizing I/O hard disk access, SAP HANA optimizes memory access between the CPU cache and main memory [3]. Not only this, SAP HANA follows a more holistic data management approach by integrating OLTP and OLAP functionality in a single system and by adding features beyond traditional database management systems, such as graph or text processing for semi- and unstructured data [4]. SAP HANA includes

multiple data processing engines and languages for different application domains. These characteristics make HANA a promising platform to meet the data management requirements.

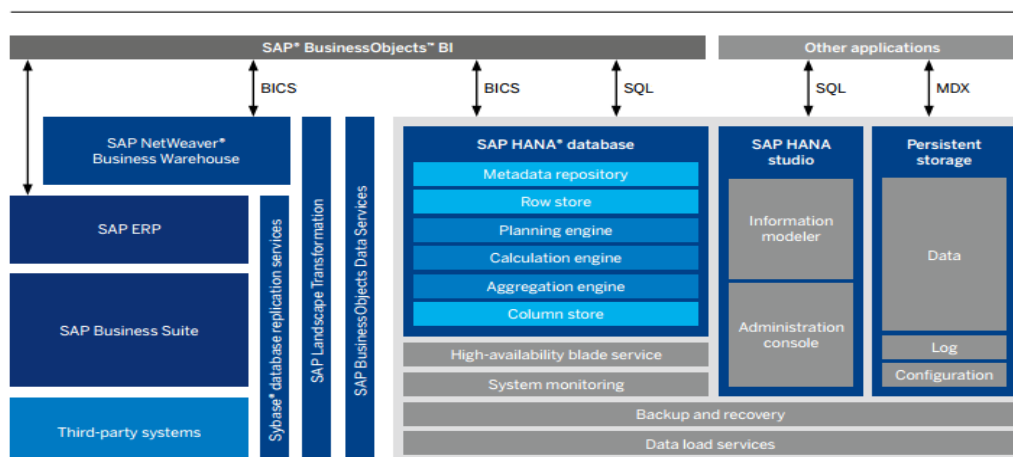


Fig.2 [3] SAP HANA Platform Overview

II.SAP HANA Platform Architecture

SAP HANA is a platform for real time analytics and applications. It enables organizations to analyse business operations based on a large volume and variety of detailed data in real time, literally at the speed of thought, from a human perspective. Initial deployments of SAP applications on SAP HANA have shown that business users can act on sub-seconds system response times, which opens the door to application possibilities that may not yet have been imagined. The platform can be deployed as an appliance or delivered via a cloud. SAP in-memory computing is the core technology underlying the SAP HANA platform [3].

SAP HANA appliance is a flexible, multi-purpose, data-source-agnostic in-memory appliance that combines SAP software components optimized to run on Intel-based hardware delivered by SAP's leading partners. SAP HANA is a combination of well-defined stack of hardware and software components. A number of components are included in SAP HANA Appliance, such as SAP HANA Database, data and lifecycle management applications, support for multiple interfaces based on industry standards, and the SAP HANA studio, an easy-to-use data modelling and administration tool.

SAP Landscape Transformation software, with real-time replication services, and SAP Business Objects Data Services software integrate with SAP HANA but are not preinstalled. SAP Landscape Transformation Replication Server is used by Trigger-Based Data Replication and is based on capturing the database changes at a high level of abstraction in the source ERP system. This method can be used to parallelize the database changes on multiple tables or by segmenting large table changes. SAP Business Objects Data Services is used by the Extraction-Transformation-Load (ETL) based Data Replication to specify and load the relevant business data in defines periods of time from an ERP system into the SAP HANA Database. ETL-based method also offers options for the integration of the third party data providers. Another type of Data Replication is Transaction Log-Based Data Replication which uses Sybase Replication and is based on capturing the table changes from low –level database log files. This method is database-dependent. The database changes are propagated on a per database transaction basis and then they are replayed on the SAP HANA Database.

SAP HANA Clients are provided for various operating systems. These database clients are connected to SAP HANA via interfaces such as JDBC (Java Database Connectivity), ODBC (Open Database Connectivity) or BICS (Business Intelligence User Services). These interfaces, for example, enable customers to use Microsoft Excel as a client tool directly from SAP HANA, as well as to employ other third-party tools that use MDX and SQL. The replication and connectivity to data source systems and business intelligence (BI) clients is established by the customer. This also includes the deployment of additional replication components on source systems and client components from the SAP Business Objects BI suite.

The SAP HANA Database is the core of SAP HANA Appliance and has been discussed in Section 4. SAP HANA main function lies in delivering the real-time analytic insight on vast data volumes. For the real-time aspect, data acquisition in real time is required, which is done by Sybase Replication Server. Tables from SAP ERP system are initially loaded into SAP HANA. All subsequent changes to these ERP tables are immediately replicated into the HANA server. To this end Replication Server makes use of the database logs in the ERP system. The tool that helps selecting the tables to be loaded and replicated is integrated into the In-Memory Computing Studio. Once the tables are created in HANA and loaded from the source system, the semantic relationships between the tables need to be modelled. At the end, the data is to be showcased for which a variety

of reporting tools are used. These reporting tools can be SAP Business Objects Explorer, Microsoft Excel, SAP Business Objects BI 4.0 Suite (Web Intelligence, Dashboards and Crystal Reports).

The SAP HANA Studio [5] is a collection of applications for the SAP HANA appliance software. It enables technical users to manage the SAP HANA database, to Create and Manage User Authorizations, and to Create new or Modify existing Models of data in the SAP HANA database. It is a client tool, which can be used to access local or remote SAP HANA databases. There are four perspectives of SAP HANA Studio as discussed below in section 2.

A. Modeller Perspective

It provides views and menu options that help to define the analytical model, for example, attribute, analytic and calculation views of SAP HANA data [6]. These views are optimized for SAP HANA Engines and Calculation Operators.

Attribute views are used to describe a transaction or other quantifiable measures in a set of data in a table. But before the attribute views can be matched to a transaction table, transaction table must be configured in an analytic view. Analytic views are the SAP HANA multi-dimensional modelling components that are used to define both measures and the relationship between their attributes. Calculation views are used in situations where an analytic or an attribute view are unable to express a calculation or when business requirements dictate a more advanced layering of processing logic. As they are considered read-only views, they can't be used to manipulate the physical data. These views express the same functionality as an analytic view, but they are more appropriately used in situations where an analytic view is unable to facilitate a desired query.

B. Development Perspective

The development perspective of SAP HANA Studio enables all HANA-related development scenarios and workflows. It is done by supporting all the development artefacts necessary for building a HANA application, covering development, testing, debugging, supportability and life cycle management. It also provides read access to the complete data model to back-end provided by service APIs. It not only helps to manage the application-development projects, but also displays the content of application packages and browse the SAP HANA Repository.

C. Debug Perspective

The debug perspective of the SAP HANA Studio provides views and menu options that help to test the applications developed by the application developer, for example, to view the source code, monitor or modify variables, and set break points.

D. Administration Perspective

This perspective provides views that enable to perform administrative tasks on SAP HANA instances. Administrators can use SAP HANA Studio to start and stop the services, to monitor the system, to configure system settings, and to manage users and authorizations. Database administration and monitoring features are primarily contained within the Administration Console perspective. Besides providing administration and monitoring features, it provides support for backup and recovery, logging, and lifecycle management.

III. SAP HANA Database Features

The distinctive features of SAP HANA can be outlined below [7, 8]:

A. Multi Engine Query Processing Environment

There are various engines to operate the different components of the database. The Relational Engine is used for storing the data in a row or column oriented manner. The Text Engine facilitates the searching of both structured and semi-structured data. The graph Engine provides the capability to run graph algorithms on networks of data entities to support business applications. The full spectrum of data engines is based on a common table abstraction as the underlying physical data representation to allow for interoperability and the combination of data of different types.

B. Representation of application specific business objects

In contrast with the relational databases, SAP HANA not only provides support to the complex enterprise-scale applications and data intensive business processes, but it also helps to understand and directly work with the business objects stored in the database. The SAP HANA database makes it possible to register "semantic models" inside the database engine to push down more application semantics into the data management layer. It supports the representation of application specific business objects (like OLAP cubes) and logic (domain specific function libraries) directly into the database engine. This permits the exchange of application semantics with the underlying data management platform that can be exploited to increase the query expressiveness and to reduce the number of individual application-to-database roundtrips and the amount of data transferred between database and the application.

C. Exploitation of current hardware development

In order to efficiently leverage modern hardware resources and to guarantee good query performance, the database management systems must consider current developments with respect to the availability of large amount of main memory, the number of cores per node, cluster configurations and SSD/flash storage characteristics. The SAP HANA Database incorporates scalable parallelism for both system level and application level algorithms.

D. Efficient communication with the application layer

HANA database is optimized to efficiently communicate between the data management and the application layer. For example, the HANA database natively supports the data types of the SAP application server. Furthermore, plans are to integrate novel application server technology directly into the SAP HANA database cluster infrastructure to enable an interweaved execution of application logic and database management functionality.

IV. Overview of SAP HANA Database Architecture

The SAP HANA data architecture is shown in Fig.3 [9]. The SAP HANA database is also known as SAP's IN-Memory Computing Engine (IMCE). The goal of IMCE is to provide a main memory data-centric platform to support pure SQL for classical applications as well as a specific interaction model between SAP applications and the database system [10].

Connections and sessions need to be created so that the database clients can interact with the database. These connections and sessions are created and managed by the Connection and Session Management component of the SAP HANA Database. Once a connection has been established, the clients can interact with the database using languages such as SQL, SQL Script, MDX or domain specific languages such as FOX, for planning applications and WIPE. A JDBC or ODBC connection needs to be maintained for interacting via SQL. The syntax and semantics of the SQL statements is checked by the SQL Parser, and then the SQL queries are translated into an execution plan by the plan generator. This execution plan is later optimized and executed by the execution engine. SQLScript is a scripting language owned by SAP HANA Database that is designed for optimization and parallelization. Functionality that cannot be expressed in SQL or other domain specific languages can be implemented using SQLScript. It is a flexible programming language with imperative elements such as loops and conditions allowing for defining the control flow of applications. Multidimensional Expressions or MDX is a language for querying and manipulating the multidimensional data stored in OLAP cubes.

The client requests are analysed by the Request Parser and dispatched to the respective component. The Execution Engine acts as a controller that invokes the different engines and routes intermediate results to the next execution step. For example, Data Manipulation statements are forwarded to the Optimizer which creates an Optimized Execution Plan that is subsequently forwarded to the execution layer. A component called Planning engine is also available in the SAP HANA Database, for executing the basic planning operations in the database layer. SQLScript, planning operations and other domain specific languages are implemented using the Calculation Engine. These languages are converted by the compilers into the Calculation Model. The calculation Engine creates a logical execution plan for Calculation Models. The Calculation Model is broken down by the Calculation Engine into operations that can be processed in parallel. Connection and Session Management component is responsible for creating and managing sessions and connections for the database clients.

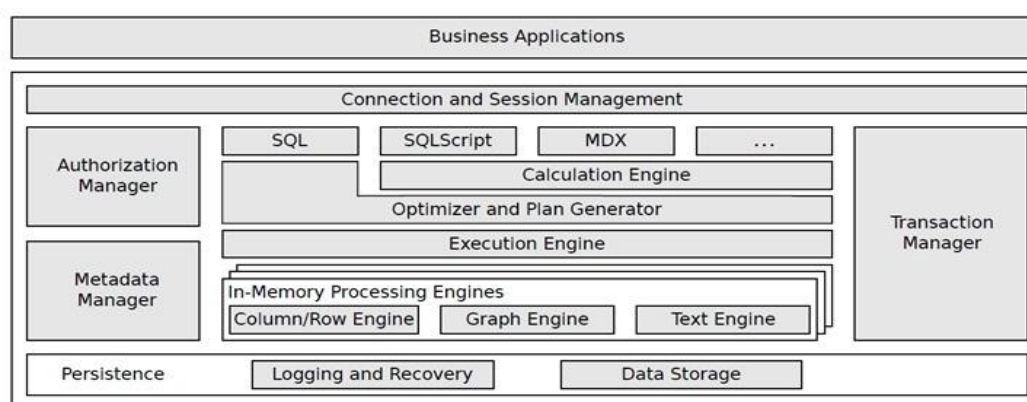


Fig. 3 SAP HANA Database Architecture

Once a session is established, clients can communicate with the SAP HANA database using SQL statements. For each session a set of parameters are maintained like, auto-commit, current transaction isolation level etc. Users are Authenticated either by the SAP HANA database itself (login with user and password) or

authentication can be delegated to an external authentication providers such as an LDAP directory. The SAP HANA database also consists of Authorization Manager, Transaction Manager and Metadata Manager performing different functionalities. In SAP HANA Database, each SQL statement is processed in context of a transaction. The transaction Manager coordinates the database transactions, by maintaining the ACID (Atomicity, Consistency, Isolation and Durability) properties of the transactions. This is achieved by cooperating with the persistence layer. It also keeps record of the running and closed transactions. Also, when a transaction is rolled back or committed, the transaction manager informs the involved engines about this event so that necessary actions can be executed. The next is the Metadata Manager, which is used to access the metadata.

The SAP HANA database metadata comprises of a variety of objects such as relational tables, columns, views and indexes. All this metadata is stored in one common catalogue for all SAP HANA storage engines. The Authorization Manager is used to authorize the user against the various operations that the user requests to perform. It checks whether the user has a privilege to execute the operations or not and it is also used to grant privileges to the user or revoke the same from the user. The heart of the SAP HANA database consists of a set of in-memory processing engines [4]. There are three engines available for processing: Relational Engine, Graph Engine and Text Engine. The relational Engine supports both row-based and column based physical representations of relational table.

The Relational Engine combines both row-oriented and column-oriented representations of relational database tables. The Relational Engine combines SAP's P* Time Database Engine and SAP's TREX (Text Retrieval and Extraction) Engine currently being marketed as SAP BWA to accelerate Business Intelligence queries in the context of SAP Business Warehouse. Column-oriented data is stored in a highly compressed format in order to improve the efficiency of memory resource usage and to speed up the data transfer from storage to memory or from memory to CPU. A system administrator specifies at definition time whether a new table is to be stored in a row- or in a column-oriented format. Row- and column-oriented database tables can be seamlessly combined into one SQL statement, and subsequently, tables can be moved from one representation form to the other. As a rule of thumb, user and application data is stored in a column oriented format to benefit from the high compression rate and from the highly optimized access for selection and aggregation queries. Metadata or data with very few accesses is stored in a row-oriented format [7].

The Graph engine is used to store the graph data and to access the graph structures, as required in SAP's planning and supply chain applications. The processing of data graphs is done with the special typing system. WIPE is used for managing the graph traversal manipulation with BI-like data aggregation. WIPE stands for "Weakly-structured Information Processing and Exploration". It is a data manipulation and query language built on top of the graph functionality in the SAP HANA Database [11]. Like other domain specific languages provided by SAP HANA Database, WIPE is embedded in transactional context, which means that multiple WIPE statements can be executed concurrently, guaranteeing the atomicity, consistency, isolation and durability. With the help of this language, multiple graph operations such as inserting, updating or deleting a node and other query operations can be declared in one complex statement. It is the graph abstraction layer in the SAP HANA Database that provides interaction with the graph data stored in the database by exposing graph concepts directly to the application developer. The application developer can create or delete graphs, access the existing graphs, modify the vertices and edges of the graphs, or retrieve a set of vertices and edges based on their attributes [6]. Besides retrieval and manipulation functions, a set of built-in graph operators are also provided by the SAP HANA Database. These operators, such as breadth-first or depth-first traversal algorithms, interact with the column store of the relational engine to execute efficiently and in a highly optimum manner.

The Text Engine is used to store the text data. As already mentioned, the data used by SAP HANA can be structured or unstructured, therefore a fuzzy search or exact words and phrases search is required by the database system. Text Engine in the SAP HANA provides these capabilities. Although virtually, all the data is stored in the main memory, however, the data also need to be stored for backup and recovery in case of system restart, shutdown or a failure.

The persistence Layer offers interfaces for reading and writing data. The backup is done by the Logger and Backup of the persistence layer by ensuring that the database is restored to the most recent committed state after a restart and that transactions are either completely executed or completely undone. For this, a combination of write-ahead logs, shadow paging and save points are used. Thus, atomicity and durability of database transactions is maintained by the persistence layer.

V. Conclusion

It can be concluded that SAP HANA Database is an important and core component of the SAP HANA Appliance. It is the SAP HANA Database, with which the connections are established by the clients and interaction is done by the clients by passing the queries through JDBC or ODBC. These queries are passed to the corresponding engines for processing. The database tables can be stored in both row and column oriented formats, as opposed to the row-oriented representation in traditional relational database management systems.

The main features highlighted with SAP HANA Database are compression of data, column stores which has led to an increased query execution. Not only this, the multi core CPUs and large memory footprints have decreased the access time.

VI. References

- [1] Hasso Plattner, Alexander Zeier, In-Memory data Management: An Inflection Point for Enterprise Applications, Springer, Berlin, 2011.
- [2] David Marks, Keith Johnson, SAP HANA Understanding the In-Memory Computing, DCS Federal, 2011.
- [3] Erich Schneider, Raghav Jandhyala, SAP HANA Platform- Technical Overview: Driving Innovations in IT and in Business In-Memory Computing Technology, SAP HANA.
- [4] Joose-Hendrik Boese, Cafer Tosun, Christian Mathis, Franz Faerber, Data Management with SAP's In-Memory Computing Engine, EDBT 2012, March 26-30, 2012, Berlin, Germany.
- [5] SAP HANA Studio- Overview, June 2013, SAP AG.
- [6] Jonathan Haun, Chris Hickman, Don Loden, and Roy Wells, Implementing SAP HANA, Galileo Press, September 1, 2013, Bonn, Boston.
- [7] Franz Farber, Sang Kyun Cha, Jurgen Primsch, Christof Bornhovd, Stefan Sigg, Wolfgang Lehner, SAP HANA Database- Data Management for Modern Business Applications, Vol, 40, No. 4, SIGMOD Record, December 2011.
- [8] Vishal Sikka, Franz Farber, Wolfgang Lehner, Sang Kyun Cha, Thomas Peh, Christof Bornhovd, Efficient Transaction Processing in SAP HANA Database- The End of a Column Store Myth, SIGMOD Record, May 20-24, 2012, Scottsdale, Arizona, USA.
- [9] Franz Farber, Norman May, Wolfgang Lehner, Philipp Grobe, Ingo Muller, Hannes Rauhe, Jonathan Dees, The Sap HANA Database- An Architecture Overview, IEEE Data Eng. Bull, 2012.
- [10] P. Grosse, W. Lehner, T. Weichert, F. Farber, and W.-S. Li. Bridging two worlds with RICE, Vol. 4, No 12, VLDB Conference, 2011.
- [11] Michael Rudolf, Marcus Paradies, Christof Bornhovd, and Wolfgang Lehner, The Graph Story of the SAP HANA Database, BTW, 2013.

Acknowledgments

We would like to thank all the faculty members of School of Management CDAC-Noida for sharing their knowledge and experiences towards completion of this. Without their constant feedback this would have been a distant reality.