SAP HANA – SAP HIGH-PERFORMANCE ANALYTIC APPLIANCE

MOGÎLDEA Andrei
Technical University of Moldova

Abstract: *SAP HANA is an in-memory platform for processing high volumes of data in real-time, which is deployable as an on premise appliance, or in the cloud. It is a revolutionary platform that's best suited for performing real-time analytics, and developing and deploying real-time applications. At the core of this real-time data platform is the SAP HANA database which is fundamentally different than any other database engine in the market today. SAP HANA can be deployed on-site as an appliance or purchased as a managed cloud or hybrid- cloud service. SAP HANA was previously called SAP High-Performance Analytic Appliance.*

Key words: *SAP HANA, Storage, Compression, SQL, Engine, SAP HANA Studio, in-memory, Dilemma, architecture.*

1. Introduction

SAP had in mind The Innovator's Dilemma, and the only way to solve it out was to innovate around the database industry entirely. None of the existing database vendors had any incentive to change the status quo, and SAP couldn't afford to sit by and watch these problems continue to get worse for their customers. SAP needed to engineer a breakthrough innovation in in-memory databases to build the foundations for a future architecture that was faster, simpler, more flexible, and much cheaper to acquire and operate. It was one of those impossible challenges that engineers and business people secretly love to tackle, and it couldn't have been more critical to SAP's future success.

2. In-Memory Basics

Thus far, we've focused on the transition to in-memory computing and its implications for IT. With this information as background, we next "dive into the deep end" of SAP HANA. Before we do so, however, here are a few basic concepts about in-memory computing that you'll need to understand. Some of these concepts might be similar to what you already know about databases and server technology. There are also some cutting-edge concepts, however that merit discussion.

Storing data in memory isn't a new concept. What is new is that now you can store your whole operation and analytic data entirely in RAM as the primary persistence layer. Historically database systems were designed to perform well on computer with limited RAM. As we have seen, in these systems slow disk I/O was the main bottleneck in data throughput. Today, multi-core CPU's – multiple CPU's located on one chip in one package-are standard, with fast communication between processor cores enabling parallel processing. Currently server processors have up to 64 cores, and 128 cores will soon be available. With increasing number of cores, CPU's are able to process increased data volumes in parallel. Main memory is no longer a limited resource. In fact, modern servers can have 2TB of system memory, which allows them to hold complete databases in RAM.

In a disk-based database architecture, there are several levels of caching and temporary storage to keep data closer to the application and avoid excessive numbers of round-trips to the database (which slows things down). The key difference with SAP HANA is that all of those caches and layers are eliminated because the entire physical database is literally sitting on the motherboard and is therefore in memory all the time. This arrangement dramatically simplifies the architecture.

2.1 Pure In-Memory Database

With SAP HANA, all relevant data are available in main memory, which avoids the performance penalty of disk I/O completely. Either disk or solid-state drives are still required for permanent persistency in the event of a power failure or some other catastrophe. This doesn't slow down performance, however, because the required backup operations to disk can take place asynchronously as a background task.

2.2 Parallel Processing

Multiple CPU's can now process parallel requests in order to fully utilize the available computing resources. So, not only is there a bigger "pipe" between the processor and database, but this pipe can send a flood of data to hundreds of processors at the same time so that they can crunch more data without waiting for anything.

2.3 Columnar and Row-Based Data Storage

Conceptually, a database table is a two-dimensional data structure with cells organized in rows and columns, just like a Microsoft Excel spreadsheet. Computer memory, in contrast, is organized as a linear structure. To store a table in linear memory, two options exist: row based storage and column storage. A row-oriented storage system stores a table as a sequence of records, each of which contains the fields of one row. Conversely, in column storage the entries of a column are stored in contiguous memory locations. SAP HANA is a "hybrid" database that uses both methods simultaneously to provide an optimal balance between them.

The SAP HANA database allows the application developer to specify whether a table is to be stored column-wise or row-wise. It also enables the developer to alter an existing table from columnar to row-based and vice-versa.

The decision to use columnar or row-based is typically a determined by how the data will be used and which method is the most efficient for that type of usage.

Column-based tables have advantages in the following circumstances:

- Calculations are typically executed on a single column or a few columns only;
- The table is searched based on values of a few columns;
- The table has a large number of columns;
- The table has a large numbers of rows, so that columnar operations are required (aggregate, scan, etc.);
- High compression rates can be achieved because the majority of the columns contain only few distinct values (compared to the number of rows).

Row-based tables have advantages in the following circumstances:

- The application needs to only process a single record at one time;
  (This applies to many selects and/or updates of single records.)
- The application typically needs to access a complete record (or row);
- The columns contain primarily distinct values so that the compression rate would be low;
- Neither aggregations nor fast searching is required;
- The table has a small number rows (configuration tables).

2.4 Compression

Because of the innovations in hybrid row/column storage in SAP HANA, companies can typically achieve between 5x and 10x to compressions ratios on the raw data. This means that 5 TB of raw data can optimally fit onto an SAP HANA server that has 1 TB of RAM. SAP typically recommends that companies double the estimated compressed table data to determine the amount of RAM needed in order to account for real-time calculations, swap space, OS and other associated programs beyond just the raw table data.

3. SAP HANA Architectural Overview

Conceptually SAP HANA is very similar to most databases you’re familiar with. Applications have to put data in and take data out of database, data sources have to interface with it, and it has to store and manage data reliably. Despite these surface similarities, however, SAP HANA is quite different “under the hood” that any database in the market. In fact, SAP HANA is much more than just a database. It includes many tools and capabilities “in the box” that make it much more valuable and versatile than a regular database. In reality, it’s a full-featured database platform.

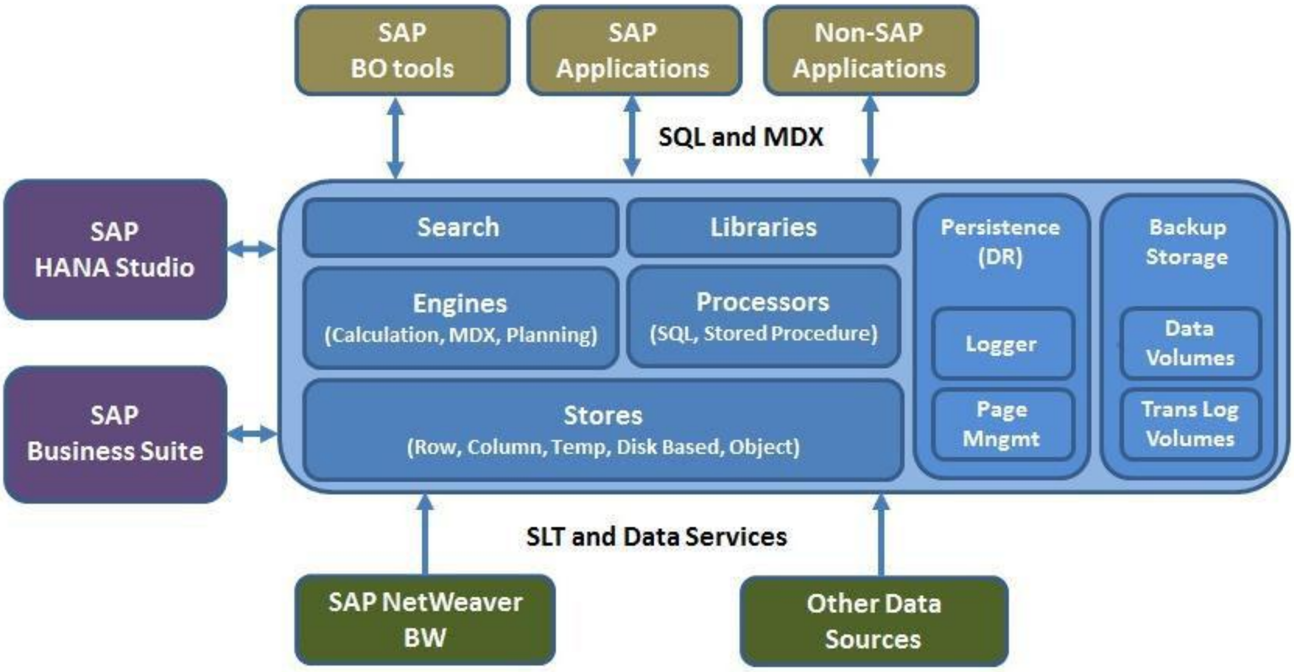

Figure 1 – Architectural overview of SAP HANA in-memory appliance

In what ways is SAP HANA unique? First, it is delivered as a pre-configured, pre-installed appliance on certified hardware. This eliminates many of the typical activities and problems you find in regular databases. Second, it includes all of the standard application interfaces and libraries so that developers can immediately get to work using it, without re-learning any proprietary API’s.

Finally, SAP HANA comes with several ways to connect easily to nearly any source system in either real-time or near real-time. These features are designed to make SAP HANA as close to “plug-and-play” as it can be and to make it a non-disruptive addition to your existing landscape (visualize Figure 1).

3.1 SQL

SQL is the main interface for client applications. The SQL implementation of the SAP HANA database is based on SQL 92 entry-level features and core features of SQL 99. However, it offers several SQL extensions on top of this standard. These extensions are available for creating tables as both row-based and column based tables and for conversion between two formats. For most SQL statements it is irrelevant whether the table is column-based or row-based. However, there are some features – for example, time-based queries and column-store specific parameters – that are supported only for columnar tables.

3.2 SQLScript

SAP HANA database has its own scripting language, named SQLScript, which offers scripting capabilities that allow application specific calculations to run inside the database. SQLScript is similar conceptually to "stored procedures" but it contains several modern innovations that make it much more powerful and flexible.

3.3 MDX Interface

The SAP HANA database also supports MDX (Multidimensional expressions), the *de facto* standard for multidimensional queries. MDX can be used to connect a variety of analytics applications like SAP Business Objects products and clients such as Microsoft Excel.

3.4 Libraries

The technical details of communicating with the SAP HANA database are contained in a set of included client libraries for standard platforms and clients. The following client libraries are provided for accessing the SAP HANA database via SQL or MDX:

- JDBC driver for Java clients;
- ODBC driver for Windows/Unix/Linux clients, especially for MS Office integration;
- DBSL (Database Shared Library) for ABAP.

SAP has leveraged its deep application knowledge from the ABAP stack to port specific functionality as infrastructure components within SAP HANA to be consumed by any application logic extension. Examples of common business functions are "currency conversion" and "calendar functionality".

4. Conclusion

What truly makes SAP HANA unique is that, in addition to its being a standard SQL database, it also natively supports data calculation inside the database itself. By incorporating procedural language support like C++, Python, and ABAP – directly into the database kernel through a dedicated calculation engine, it can achieve exceptional performance because the data do not need to be moved out of the database, processed, and then written back in.

Bibliography

1. Basic Sap Production Guidelines and Maple Syrup Grading and Instrumentation Staats, L.J., and Hollen, G. 1993. Vol 7. (30 minute educational video) Department of Natural Resources, Cornell University.

2. Faerber, F., May, N., Lehner, W., Große, P., Mueller, I., Rauche, H. and Dees, J.The SAP HANA Database – An Architecture Overview 2012 – IEEE In-text: (Faerber et al.)Bibliography: Faerber, Franz et al. The SAP HANA Database – An Architecture Overview. 1st ed. IEEE, 2012. Web. 15 Jan. 2015.

3. Hana.sap.com HANA Technology 2015 In-text: (Hana.sap.com) Bibliography: Hana.sap.com,. "HANA Technology". N.p., 2015. Web. 16 Jan. 2015.