

Data Virtualization with SAP HANA Smart Data Access

Abani Pattanayak

SAP HANA Center of Excellence, SAP America, Dallas, USA

Email: abani.pattanayak@sap.com

How to cite this paper: Pattanayak, A. (2017) Data Virtualization with SAP HANA Smart Data Access. *Journal of Computer and Communications*, 5, 62-68.
<https://doi.org/10.4236/jcc.2017.58005>

Received: April 18, 2017

Accepted: June 23, 2017

Published: June 26, 2017

Copyright © 2017 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Digital transformation has been corner stone of business innovation in the last decade, and these innovations have dramatically changed the definition and boundaries of enterprise business applications. Introduction of new products/services, version management of existing products/services, management of customer/partner connections, management of multi-channel service delivery (web, social media, web etc.), merger/acquisitions of new businesses and adoption of new innovations/technologies will drive data growth in business applications. These datasets exist in different sharing nothing business applications at different locations and in various forms. So, to make sense of this information and derive insight, it is essential to break the data silos, streamline data retrieval and simplify information access across the entire organization. The information access framework must support just-in-time processing capabilities to bring data from multiple sources, be fast and powerful enough to transform and process huge amounts of data quickly, and be agile enough to accommodate new data sources per user needs. This paper discusses the SAP HANA Smart Data Access data-virtualization technology to enable unified access to heterogenous data across the organization and analysis of huge volume of data in real-time using SAP HANA in-memory platform.

Keywords

SAP HANA, In-Memory Computing, Smart Data Access (SDA), Data Virtualization & Data Federation, Virtual Data Model And Big Data

1. Introduction

SAP HANA [1] is a high-performance analytics platform based on an in-memory columnar database developed and marketed by SAP. In SAP HANA, data is stored in memory (RAM) which gives the CPUs quick access to data for processing and it can process around 3b scans/second/core and 20 m aggrega-

tions/core/second with Intel Haswell based processors [2]. Therefore, with a typical 120-core system, SAP HANA can perform 360 B scan/second and 2.5 B aggregations/sec, which is quite ground-breaking.

SAP HANA supports the paradigm of computing at the source and provides several techniques to move the application logic into the database. Since SAP HANA is a columnar database, only the requested table columns are scanned and selection constraints or filters on table columns are pushed to the table level, so only the requested rows are accessed and aggregated, resulting in a significant reduction in data transfer to the application layer. SAP HANA also offers specialized libraries at the database layer for advanced analytics (for text processing, geo-spatial, predictive, machine learning and graph analysis).

Using Smart Data Access (SDA) [3] technology, SAP HANA supports virtual real-time data access from multiple heterogeneous systems on premise and in cloud. External database objects can be accessed as virtual tables within SAP HANA database and sophisticated virtual data models can be built using SAP HANA calculation views to provide unified data access across the organization. These virtual models are easy to create and provide significant transformation & localization capabilities to model complex business requirements. SAP HANA views can be accessed using open standard interfaces like ODBC/JDBC, SOAP/REST, Cloud API etc.

2. SAP HANA Architecture

The key components of SAP HANA architecture are illustrated in **Figure 1**.

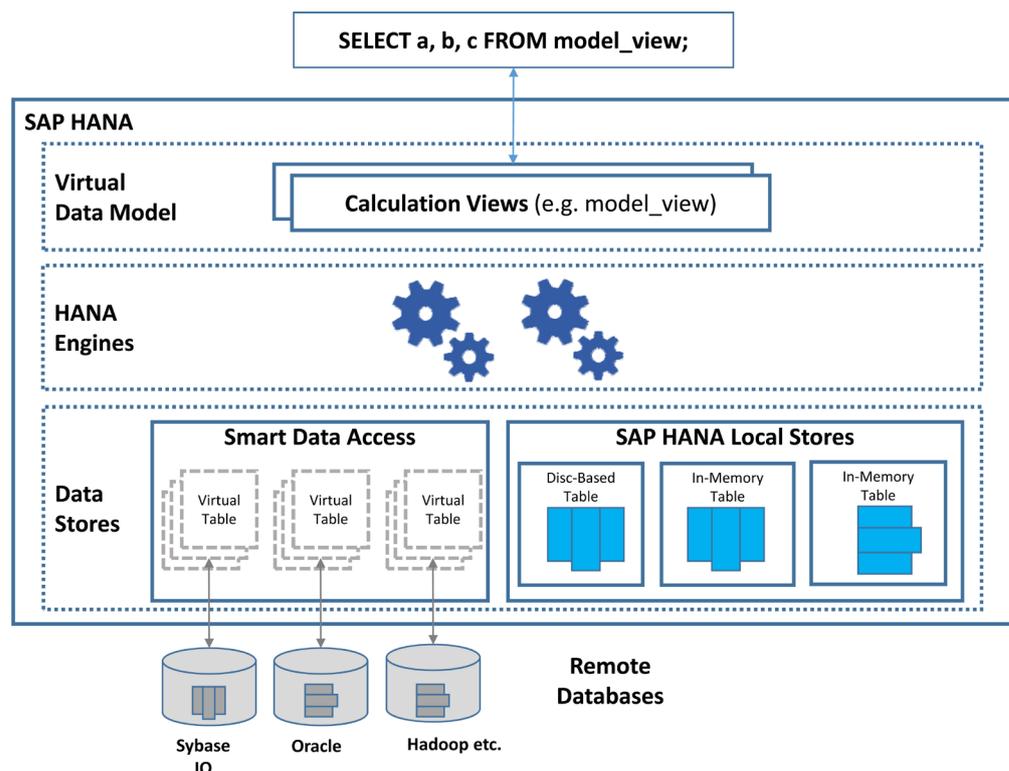


Figure 1. SAP HANA architecture.

SAP HANA Local Stores: This includes all data stores which physically reside in the SAP HANA database *i.e.* in-memory column store, in-memory row store tables and disc-based (dynamic tiering) column tables. The most active and most critical dataset is stored as in-memory column store tables and offers the best query performance. Data in the in-memory column store tables may be from a business application running on the SAP HANA database or may be replicated from an external business application. The less active and less critical dataset (typically older dataset) may be stored in the disc-based column tables, which offers reduced cost of storage.

Smart Data Access: Smart Data Access technology was introduced with SAP HANA SP06. The Smart Data Access layer includes virtual connection to remote database objects (tables, views and calculation views) in an external supported database like SAP HANA, Sybase IQ, Oracle, SQL Server, Teradata, DB2, Hadoop (Hive and Spark SQL) etc. Only metadata (connection and structure) of the remote database objects are stored as virtual table in the SAP HANA database schema. Data in the virtual table continue to stay in the remote database and is accessed in real-time irrespective of the location of the remote database. The speed of data access will depend on the capability of the remote database, network connection to the remote database and amount of data transferred across the network.

HANA Engines: SAP HANA has various database engines (Row Engine, OLAP Engine, Join Engine, Calculation Engine and SQL Engine) to perform various operations efficiently. The SQL engine is responsible for processing SQL queries in SAP HANA and will engage other engines based on the type of operations and calculation. The SQL engine is also responsible for forwarding appropriate SQL sub-queries to the remote database for execution.

Virtual Data Model: The Virtual Data Models in SAP HANA support set-based operations, enable real-time analytics and on the fly calculations without the use of persistent aggregates. Virtual data models are build using SAP HANA information views (*i.e. Attribute Views, Analytic Views, Graphical Calculation Views and Scripted Calculation Views*) [4]. However, only calculation views are supported to model virtual data models on SDA virtual tables. SAP also recommends using graphical calculation views for both local and virtual tables with introduction of SAP HANA SP09.

3. Smart Data Access Components

The key components of Smart Data Access [5] are *Persistence Layer, Query Processing and Federation Adapter Framework* as illustrated in **Figure 2**.

3.1. Administration

SAP HANA Studio or SAP HANA Web Workbench is used to configure, manage and monitor SDA connections. Certified JDBC/ODBC SDA driver for the remote database must be installed before configuring connection to the remote data sources. The user must have *CREATE REMOTE SOURCE* or *DATA*

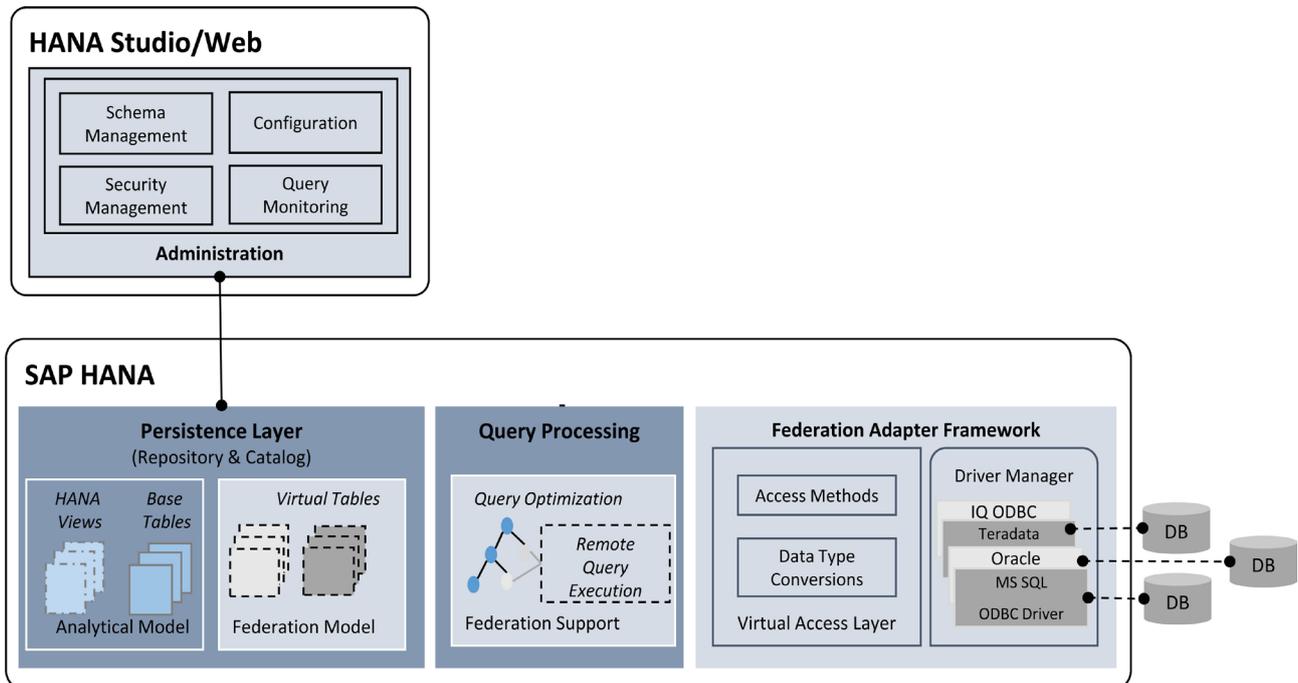


Figure 2. Smart data access components.

ADMIN system privilege and *CREATE VIRTUAL TABLE* object privilege to work with virtual tables.

Virtual tables can be created in any schema in the SAP HANA database. It is recommended to create separate schema for each remote data source to secure them using standard SAP HANA security framework.

The *Query Monitor* of the Smart Data Access administration displays statistics of all the queries executed or in process in the remote data sources.

3.2. Persistence Layer

Virtual tables can be created in any schema in SAP HANA database like local tables. The persistence layer includes local tables, activated calculation views and metadata (structure of the object and remote connection details) of the virtual table. Only the metadata of the virtual table is persisted in SAP HANA, the data in the virtual table continue to stay in the remote database. Virtual table can be accessed in SQL script and used in Calculation views like local tables in SAP HANA database.

3.3. Query Processing

The SQL engine is responsible for processing the SQL queries on virtual data models using both local and virtual table. The SQL parser of the SQL engine validates the syntax and semantics of the SQL query. The SQL optimizer then creates a single acyclic execution model by decomposing the calculation views [6]. A rule-based optimizer is engaged to push the filter to the table level, combine multiple projection/aggregation/join steps into one projection/aggregation/join step and identify query fragments that need to be executed in the remote

SDA system. A cost-based optimizer estimates cost of each operation to derive the most cost-effective execution plan and the SQL executor then delegates the operations to SAP HANA internal engines (for local tables) and forwards the SQL fragments to remote database for execution.

The SQL optimizer is aware of the capabilities (and shortcomings) of the remote database and uses statistics of the virtual table to optimize remote execution. It also uses the following techniques to reduce data transfer across the network.

Predicate Push down: The filters/constraints part of the calculation view and SQL query are automatically added to SQL fragment to be executed in the remote server to reduce the data selection from the remote database.

Join Relocation: The statistics of the local tables and virtual tables (e.g. no of records and size of columns) are used to determine the most effective place to perform the join. Data movement can occur in either direction to perform the join efficiently. For example—join between two virtual table on the same remote database will be executed in the remote database, while join between two tables (one smaller & one bigger) will occur at location of the bigger table (SAP HANA or remote database) and the smaller table will be copied over (to the location of bigger table) to perform the join.

Functional compensation: If the remote database does not support a SQL operation which is supported in SAP HANA, that SQL operation on the remote dataset will be performed in SAP HANA during SQL execution.

Remote Caching: If result materialization/caching is supported by the remote sources (like Hadoop Hive data sources), SAP HANA can trigger the caching process on the remote source to avoid the repetitive execution of the same query in the remote system. This behavior is easily managed by including a hint clause (e.g., *WITH HINT (USE_REMOTE_CACHE)*) part of the SQL query.

3.4. Federation Adapter Framework

The federation adapter framework is responsible for establishing communication & coordinating data exchange with remote sources and aggregating results from all remote sources involved in query execution.

The *Driver manager* is responsible loading database-specific JDBC/ODBC drivers to establish connection to the remote data source. Please note certified SDA driver must be installed and configured before configuring a remote data source.

Access Methods enables integration with the SQL-optimizer and other SAP HANA query processing components.

Data Type Conversion: For seamless data access across multiple heterogeneous remote systems, all remote data types are converted to SAP HANA data types (e.g. VARCHAR2 in Oracle database is converted to NVARCHAR in SAP HANA).

4. Building Virtual Data Model

Building virtual data models with SDA virtual tables is like that of local tables

with the following exceptions.

- 1) SDA virtual tables are only supported in Calculation Views (no support for Attribute and Analytic views).
- 2) One or more virtual tables from the same remote source must be isolated into one or more (virtual table exclusive) calculation views. These calculation views must not have SAP HANA native objects (*i.e.*, tables, analytic views, attribute views, etc.) or virtual tables from another remote source. These calculation views must be set to execute in *SQL engine*.
- 3) One more calculation views may be used to combine results (union, join etc.) from virtual table exclusive calculation views (from #2) and calculation views based on local tables to provide unified access to data from multiple heterogeneous data sources.

5. Reporting and Visualizations

SAP HANA virtual models can be used in SAP Business Objects BI tools and Business Objects Cloud for analysis and visualizations. SAP HANA virtual data models can also be consumed by third-party BI tools (Tableau, MicroStrategy, Cognos, QlikView etc.) supporting JDBC/ODBC data access. SAP HANA also support development of HTML5 based applications & visualizations using the in-built XS Server and XS Advanced framework.

6. Conclusions

As discussed SAP HANA provides a sophisticated platform to provide unified real-time data access across the entire organization irrespective of the physical location, type of database and technology platform. The Smart Data Access technology significantly enhances the value of business intelligence solutions with a complete view (from the entire cross section of business applications on premise and cloud) of the information to make business decisions. In addition, data virtualization using Smart Data Access provides the following benefits

- 1) It offers better/faster access to data than physically copying/moving data, and faster/less costly than batch data movement (ETL).
- 2) Simplifies data usage. No special syntax is needed to access heterogeneous data sources.
- 3) Since data is centrally accessed, it provides improved access control, compliance and policy enforcement.
- 4) Since data sources are loosely coupled, independent development/ maintenance possible at each source system with no impact on other systems. Also, new source systems can be added or existing one can be taken offline with no impact on end users.
- 5) Significantly reduces the risk of data exposure with fewer physical copies of data and provides reverse proxy access to data keeping physical data sources anonymous.
- 6) Improves the use of data assets and maximizes return from existing hardware and storage investments.

- 7) Provides big data integration, enabling access to data stored in Hadoop and NoSQL systems as SQL tables.

Acknowledgements

I want to express my gratitude for my supervisor Greg Smith, V.P.—Digital Business Services, SAP America & Dhirendra Gehlot, Director—Digital Business Services, SAP America for their encouragement and support in publishing this paper.

References

- [1] SAP (2017) What's SAP HANA?
<https://www.sap.com/product/technology-platform/hana.html>
- [2] SAP (2017) What's SAP HANA?
<http://discover.sap.com/hana/en-gb/what-is-hana.html>
- [3] SAP Note 1868209 Version 14 (2017) SAP HANA Smart Data Access: Central Note.
<https://launchpad.support.sap.com/#/notes/1868209>
- [4] SAP Help Portal (2017) SAP HANA Modeling Guide.
<https://help.sap.com/viewer/fc5ace7a367c434190a8047881f92ed8/2.0.00/en-US>
- [5] Pattanayak, A. and Ruter, W. (2016) Smart Data Access: Data Virtualization in SAP HANA. Rheinwerk Publishing, Quincy.
- [6] SAP Note 2291812 Version 4 (2016) SAP HANA DB: Disable/Enable Calculation Engine Feature—CalcView Unfolding.
<https://launchpad.support.sap.com/#/notes/2291812>



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.
A wide selection of journals (inclusive of 9 subjects, more than 200 journals)
Providing 24-hour high-quality service
User-friendly online submission system
Fair and swift peer-review system
Efficient typesetting and proofreading procedure
Display of the result of downloads and visits, as well as the number of cited articles
Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact jcc@scirp.org