In-Memory Technique: High Performance Analytics with SAP HANA

Track E- DATA Warehousing and DATA Mining

***Sunitha Watts**
Professor, City College, Bangalore
E-Mail: jsw.watts@gmail.com

****Aliyas Shaik**
Assistant Professor, City College, Bangalore
E-Mail: aliyasshaik@gmail.com

ABSTRACT

Increasingly sophisticated business decision models depend on extremely fast access and manipulation of massive data stores. Insight into business operations often demands data volumes that are beyond the capabilities of traditional disk-based systems to process them in real time, which can limit access to benefits such as Efficiency provided by the ability to respond in real-time to the changing needs of the business [1, 3], Flexibility based on insight that accurately directs quick action and Empowerment of business users to make and act on smart decisions. This reality often causes a separation between on-going business needs and the analytic applications that support them. The purpose of this research is to identify and avoid delays caused by the data replication and the lag time between data gathering and interpretation, which delays the ability to benefit from business information, thereby limiting its value.

SAP HANA in-memory database is a hybrid in-memory database that combines rowbased, column-based, and object-based database technology, optimized to exploit the parallel processing capabilities of current hardware in cloud environment [9]. The study shows how to Gain real-time business insights with near-zero latency.

Keywords : In-Memory Technique, BI, SAP High Performance Analytics Appliance (HANA), RDBMS, Cloud, High Availability.

1. INTRODUCTION

In-memory computing can be used to significantly reduce the time required to access database content and to increase the processing speed of transactions considerably by storing the data in the main memory of the database system. As a result, large data volumes can be processed immediately and complex analyses can be performed extremely quickly and with upto-the-minute data. SAP HANA, in-memory processing has greatly shifted away from pure OLTP or OLAP, and has evolved into a converged platform that transforms information processing and energizes application development. SAP HANA is optimized for both transactional and analytical processing [1]. Instantly access and analyze transactional and analytical data. Dramatically improve business planning, forecasting, trend analysis and financial close processes while reducing TCO with less hardware, maintenance and testing. Explore and analyze vast quantities of data from virtually any data source – at the speed of thought.

With SAP Business Suite on HANA, the use of SAP HANA is supported in addition to regular databases in an integrated scenario. That means, in this case, the scenarios that used to be performed on a relational database of system are now performed exclusively on an SAP HANA database. In this way, SAP enables IT to use existing and optimized functions on an SAP HANA database [4].

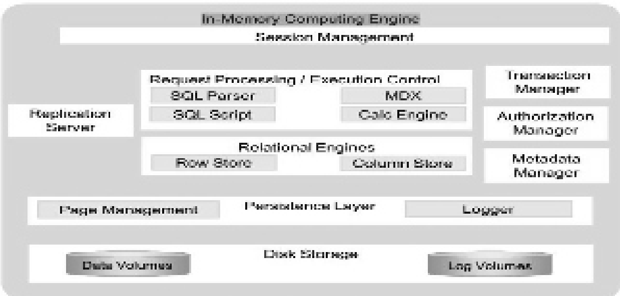

Fig. 1: SAP HANA Architecture Overview

3. BACKGROUND OF STUDY

Hardware factors

The speed limit of SAP HANA import is limited by hardware configuration. No matter what we have done in software level, the hardware is the most important factor. There are 3 kind of hardware factor which affect the speed limit of importing.

- **Disk type** - Since SAP HANA importing is always being with transaction log writing and delta log writing, the disk write speed is significant. SSD as log are and data area of SAP HANA is recommended.
- **Number of CPU cores** - SAP HANA is able to make full use of multiple cores to import data. So number of CPU cores decides the speed of importing.
- **Size of memory** - SAP HANA is an in-memory database which data is to stay in memory. If the size of memory is not big enough, data importing will lead to lack of memory. Then HANA will unload other data which is not used recently. It will reduce the speed of importing data. Besides, in the period of reading csv files, the size of cache will increase rapidly. When size of cache is too large, operation system will release some space of cache.

This process will affect importing data.

SAP HANA factors

In SAP HANA, data are stored not only in memory but also in disk and log files. To get the speed limit of importing data, we need abandon some configurations for security reasons.

- **Partition** - The partition of table can contribute to improve the parallel degree. Through my experiments, hash partition is the best method of partition. And the numeric field is the best type of partition value.
- **Auto merge** - By default, the data imported into table are stored in delta are. Then the delta area is merged into main area automatically. And this process will not import data into database. So we can disable auto merge to ensure the process of importing will not do merge operation.
- Delta log - To column store, SAP HANA will write delta log into disk when importing data. This process will reduce the speed of importing data. Our aim of

importing is to put the data in to memory, and that process is to make sure the imported data will lose. So we can disable delta log to improve the speed of importing.

- Number of threads - To make full use of multiple cores, we can set the number of threads when importing data. Through experiment, number of threads= number of CPU cores is the best setting.
- Number of tuples in a batch - SAP HANA imports data in batches. We can set the number of tuples in a batch.

Hardware configuration	Importing speed
CPU : 16 cores Memory : 256GB Disk type : SSD	100 M/s
CPU : 80 cores Memory : 1 TB Disk type : SSD	1. 308.8M/s

Fig. 2: Results of DATA Importing Speed based on different H/W Configurations

Request Processing and Execution Control (Calculation Model)

The easiest way to think of Calculation Models is to see them as dataflow graphs, where the modeler can define data sources as inputs and different operations (join, aggregation, projection) on top of them for data manipulations. The Calculation Engine will break up a model, for example some SQL Script, into operations that can be processed in parallel (rule based model optimizer). Then these operations will be passed to the database optimizer which will determine the best plan for accessing row or column stores like algebraic transformations and cost based optimizations based on database statistics.

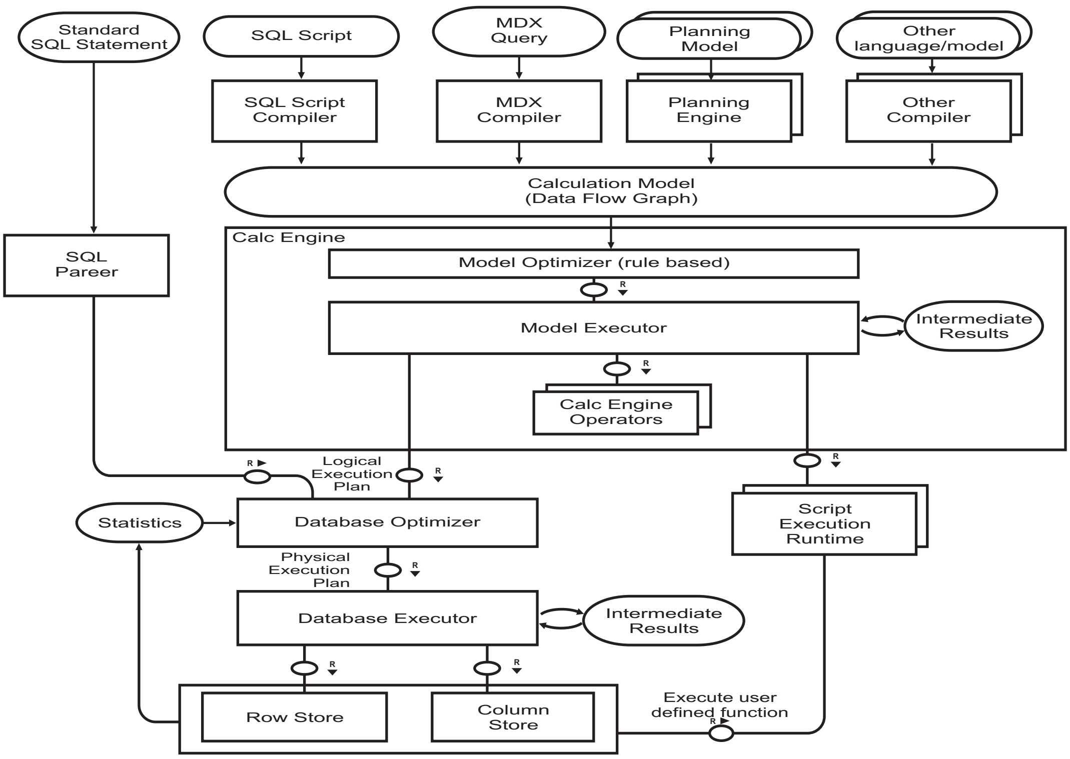

Fig. 2: Conceptual View - Request Processing and Execution Control

Overall we moved data into a 2 TB on HANA environment and tested almost 100 of the most used queries. We also tested HANA data load performance of over 500 jobs. This was a test to see if it could be done and to get some performance benchmarks from testing a real system with real queries.

HANA Query Performance – 16.6 times faster

Tested almost 100 queries from different business areas and found that had an average of 16.6 times faster query run-times. This meant that a query that took on average 64 seconds on Oracle now ran in 4 seconds on HANA. Actually, queries that predominantly used simple sub-totals, group-buy and large data reads, performed the best in HANA.

HANA Load Performance– 43% faster

When executed over 530 data load jobs in our test. Many were executed multiple times. We did not make any changes to the SAP on HANA loads, even when we could have got even faster data loads by adding hints on slow ABAP routines.

4. METHODOLOGY

Every technology has limits, and HANA is constrained by the same laws of physics as any other computer system. The major differences with HANA are:

1. Data is kept entirely in-memory and therefore accessing all data has the same latency.
2. Data is compressed in-memory and only uncompressed in the CPU itself.
3. Data may be loaded into all cores simultaneously and a single instruction executed against all at the same time (massively parallel).

The major challenge with HANA is getting optimized data in fast enough. Networks bottom out at 300MB/sec, disks are often even slower and if organization need to load HANA data fast, then need to think carefully about design.

5. LITERATURE REVIEW

Inefficient, with a Shared-Nothing Model

HANA suffers from the classic deficiencies of a shared-nothing partitioned model. Unless the application itself is partition aware (which would be a poor app design anyway), to make this data partitioning work, the data access has to be a multi-hop process, in which the MASTER Name Server needs to do a lookup and forward incoming connections to the right Index Server hosting the corresponding data partition. This increases data access latency, and also leads to poor load balancing, especially when a local data set is accessed heavily. This makes HANA a poor choice for most OLTP workloads, adding prohibitive workload for short transactions.

Synchronous in-memory

The primary system commits the transaction after it receives a reply that the log was received by the secondary system, but before it was persisted. The transaction delay in the primary system is shorter than the previous case, because it only includes the data transmission time and no disk I/O latency. This option provides better performance, but is slightly more vulnerable to data loss.

Standard Application Benchmark

The SAP BW Enhanced Mixed Load Benchmark (BW-EML Benchmark) meets the current demands of typical business warehouse customers [6]. These demands are mainly coined by three major requirements:

- **Near real-time reporting** – Getting instant results from analytical applications on last minute data nowadays is crucial for timely decision making. Typical examples for realtime data analysis include smart metering or trade promotion management.
- **Ad-hoc reporting capabilities** – During the last few years, data volumes in data warehouses have dramatically grown. One reason for this growth is the increased complexity and detail level of the data, which in turn requires much more sophisticated and complex analysis methods. As a result, analytical applications are supposed to facilitate navigating through huge amounts of data by providing extensive slicing and dicing functionality. This makes it inherently difficult to foresee frequent navigation patterns and pre-calculate intermediate results to speed up reporting performance. Ad-hoc type query capabilities are required to satisfy these demands.
- **Reduction of TCO** – Typical sizes of today's data warehouses comprise tens or hundreds of terabytes of data. It is therefore crucial to keep data redundancy at a low level, but still be able to maintain layered data models. SAP NetWeaver BW 7.30 helps reduce the total cost of ownership by allowing reports to run directly on DataStore objects which are the core building elements of a layered warehousing architecture, often eliminating the need to maintain data redundantly in multi-dimensional Info Cube data structures.

The latest addition to the family of SAP BW Application Benchmarks – the BW-EML Benchmark – has been developed with these three customer requirements in mind.

Like its predecessor the BW-MXL Benchmark, the EML Benchmark focuses on a mix of multi user reporting load and delta data that is loaded into the database simultaneously to the queries.

Benchmark Data Model

The data model consists of three InfoCubes and seven DataStore objects. Each of these objects holds data of one particular year. The three InfoCubes hold the same data as the corresponding DataStore objects for the last three years. Both object types have the same set of fields. The InfoCube comes with a full set of 16 dimensions which comprise a total of 63 characteristics, having cardinalities of up to 1 million different values and one complex hierarchy. With its 30 different key figures, including key figures requiring exception aggregation, the InfoCube data model has been defined in close accordance with typical customer data models. In the DataStore object data model, the high cardinality characteristics have been defined as key members, while other characteristics have been modeled as part of the data members.

Data Volumes

SAP BW-EML Benchmark can be executed with various different data volumes. In its smallest configuration, the benchmark rules require an initial load of a total of 500 million records (i.e. 50 million records per InfoCube / DataStore object) coming from ASCII flat files. Further possible configurations include initial load volumes of 1000 million and 2000 million records and more. Even larger data volumes can be defined for distributed server landscapes. The total record length in the ASCII files is 873 bytes. In any case, the total number of records that needs to be loaded in addition to the initial load is one thousandth of the initial records. A single benchmark run is supposed to last at least one hour. During this time, the delta data have to be loaded in small chunks every five minutes. Each InfoCube and DataStore object has to be loaded with the same number of records.

Query Model

Eight reports have been defined on two Multi Providers – one Multi Provider for the three InfoCubes, and another Multi Provider for the seven DataStore objects. Since the InfoCubes and DataStore objects have the same set of fields, the respective reports on both Multi Providers are identical so that we actually have two sets of four queries each.

Reports select data for one particular year, picking the InfoCube or DataStore object containing the data randomly. Within one report further navigation steps are executed, each of them resulting in an individual query and a database access. Although the first three reports follow similar navigation patterns, the filter and drill-down operations have been randomized to address the demand for ad-hoc types of queries. While random values for filter parameters make sure that different partitions of data are accessed, a random choice of characteristic that are used for drill downs or other slice-and-dice operations makes sure that a huge number of different characteristics combinations is covered in a multi user reporting scenario. In order to guarantee a high degree of reproducibility of the reporting results, characteristics have been grouped by their respective cardinalities, and only characteristics of the same cardinality are considered for a randomized operation.

The key figure of this benchmark is the number of ad-hoc navigation steps/hour.

Date of Certifcation (mm/dd/yyyy)	Technology Partner	Number of records	Ad-Hoc Navigation Steps/Hour	Operating System - Release Database Server	RDBMS Release	SAP NetWeaver Release	Number & Type of Database/Application Servers	Certifica tion Number
2/6/2014	HP	2,000,000,000	111,850	SuSE Linux Enterprise Server 11 on VMWARE ESX 5.5	SAP HANA 1.0	SAP NetWeaver 7.30	1 database server: HP DL580 Gen8, 4 processors / 60 cores / 120 threads, Intel Xeon Processor E7-4880 v2, 2.50 GHz, 64 KB L1 cache and 256 KB L2 cache per core, 30 MB L3 cache per processor, 1024 GB main memory 1 application servers: HP BL680 G7, 4 processor / 40 cores / 80 threads, Intel Xeon Processor E7-4870, 2.40 GHz, 64 KB L1 cache and 256 KB L2 cache per core, 30 MB L3 cache per processor, 1024 GB main memory 1 application servers: HP BL680 G7, 4 processor / 40 cores / 80 threads, Intel Xeon Processor E7-4870, 2.40 GHz, 64 KB L1 cache and 256 KB L2 cache per core, 30 MB L3 cache per processor, 512 GB main memory	2014021
3/26/2014	HP	2,000,000,000	126,980	SuSE Linux Enterprise Server 11	SAP HANA 1.0	SAP NetWeaver 7.30	1 database server: HP DL580 Gen8, 4 processors / 60 cores / 120 threads, Intel Xeon Processor E7-4880 v2, 2.50 GHz, 64 KB L1 cache and 256 KB L2 cache per core, 30 MB L3 cache per processor, 1024 GB main memory 2 application servers: HP BL680 G7, 4 processor / 40 cores / 80 threads, Intel Xeon Processor E7-4870, 2.40 GHz, 64 KB L1 cache and 256 KB L2 cache per core, 30 MB L3 cache per processor, 1024 GB main memory"	2014009

Fig. 3: SAP BW Enhanced Mixed Load Standard Application Benchmark Results

6. CONCLUTION

SAP Landscape Transformation (LT) is trigger-based so there is a small overhead, and it is therefore suited to low-mid volume transactional tables like sales orders. It can be used to replicate tables from non-SAP databases like Oracle. Sybase Replication Server (SRS) has several benefits over LT, because it scrapes database logs, which has lower latency and overhead than the trigger-based LT. Stock Exchanges in Capital Markets use SRS to replicate trading data between continents, it's that fast. Sybase Event Stream Processing (ESP) is being integrated into the HANA database will be able to do all of this in-memory. ESP is low-latency and can shift a very large amount of messages a second, depending on the hardware and networks. It can store a buffer of data in memory for doing time stream processing. Each data set performs slightly differently with HANA because of the complexities of massive parallel processing, columnar compression and column widths and complexities, plus the need to transform data.

SAP HANA delivers leading performance and scale-out ability and enables real-time BI for businesses that must support a range of analytic workloads, massive volumes of data, and thousands of concurrent users.

REFRENCES

[1] Vishal Sikka, Juchang Lee, Michael Muehle, Norman May, Franz Faerber, Hasso Plattner, Jens Krueger, Martin Grund "High-Performance Transaction Processing in SAP HANA".

[2] Lalit Dole, Girish Talmale and Jayant Rajurkar "Analysis of SAP HANA High Achievability Competence" January - 2014, pp. 141-146.

[3] Philipp Grobe, Wolfgang Lehner and Norman May "Advanced Analytics with the SAP HANA Database".

[4] Franz Farber and Norman May and Wolfgang Lehner "The SAP HANA Database – An Architecture Overview".

[5] Dr. Berg, "Real HANA Performance Test Benchmarks" 2013,http://sapinsider.wispubs.com/

[6] Todd Muirhead, "Certified SAP BW-EML Benchmark on Virtual HANA" 2014.

[7] "SAP Standard Application Benchmark Results" 2013 and 2014, http://global.sap.com/

[8] H. Plattner. A, "Common Database Approach for OLTP and OLAP Using an In-Memory Column Database". In Proc. SIGMOD, pages 1–2, 2009.

[9] SAP HANA Cloud Documentation, https://help.hana.ondemand.com/

[10] TPC-H. http://www.tpc.org/tpch/, 2014.

[11] "Quest Benchmark Factory," Website, 2012, http://www.quest.com/benchmark-factory/

[12] IBM X6 enterprise servers, http://www-03.ibm.com/systems/in/x/x6/

[13] IBM Redbook: In-memory computing with SAP HANA on IBM eX5 Systems