Enhancement of Database Mining Strategies in SAP HANA

Suraj U Rasal
Assistant Professor
Department of Computer Engineering
Bharati Vidyapeeth Deemed University's College of Engineering, Pune

Kuldeep Prakash
Department of Computer Engineering
Bharati Vidyapeeth Deemed University's College of Engineering, Pune

Varsha Thanaji Mulik
Department of Computer Engineering
Nehru College of Engineering & Research center, Kerala

Shraddha T Shelar
Department of Information Technology
D Y Patil College of Engineering Akurdi, Pune

Abstract

Three main layers of SAP-HANA shows data mining approach and the communication approach with the outside environment. These three layers are appliance layer, database layer and user connectivity layer. Data mining approach is applied between appliance and database layer [1]. And same approach is applied for the third layer also. SAP-HANA modules works with data mining approaches to access and transfer data [6]. In this paper data mining approach is improved by adding mediator concept to manage database access. This mediator approach will improve the performance level while processing communication between appliance and database layer of the SAP-HANA architecture in the respective module.

Keywords: (Software update (S_U), Hosting(H), Client info (C_I), Work (W_A)

I. INTRODUCTION

In the ever growing industrialization, the changes in business strategies and analysis techniques have changed rapidly. Due to which there is a need felt to avoid the massive adaption breakdown due to the accelerative thrust and the unprecedented impacts of it. The business applications require a large number of functionalities and data management techniques. These requirements have increased with the passage of time and the classic techniques can no longer stand alone to fulfill the broadness of the functionalities and flexibilities required. Applications now demand immediate availability of data in order to make quick analysis with proper insight for an efficient decision making for which large amount of data is needed to be kept online for analytics and queries. Some applications require the graphical data models whereas some require text retrieval in the data management in order to associate the structured and unstructured data [8]. As the spectrum of required application support has a huge range of interaction pattern and inhabits heterogeneous nature, the classical SQL based data management engines are draw limitations for these application requirements. Hence, the SAP-HANA database has come up as a comprehensive way of database management providing efficient database management services for the various complex business processes of the modern time [3]. SAP has publicized HANA as the modern platform for real-time analytics and applications. SAP HANA is improved stage as compare to SAP including all its modules. SAP HANA architecture shows three measure interactive layers, appliance, database and outside environment. To make processing with outside world, SAP HANA has to interact with appliance layer which is keen to improve its strategies to show high efficiency and performance. These strategies have been developed so that SAP HANA architecture can return high performance and efficiency [3]. Hence in this paper mediator are allocated to manage all segments in appliances.

II. EXISTING APPROACHES OF SAP-HANA

The SAP HANA is a database that has main focus on the memory of the system and leverages a large amount of low cost main memory, multi-core processor and SSD storage for its fast transactional applications and analytical performances [2]. These leverages qualify SAP HANA for the capabilities of modern hardware. Existing datamining approach shows generalized data access structure where appliance layer delivers its information in the regular approach. In this approach relational database concepts are applied [4]. Some where some BI and accelerators are also used [5].

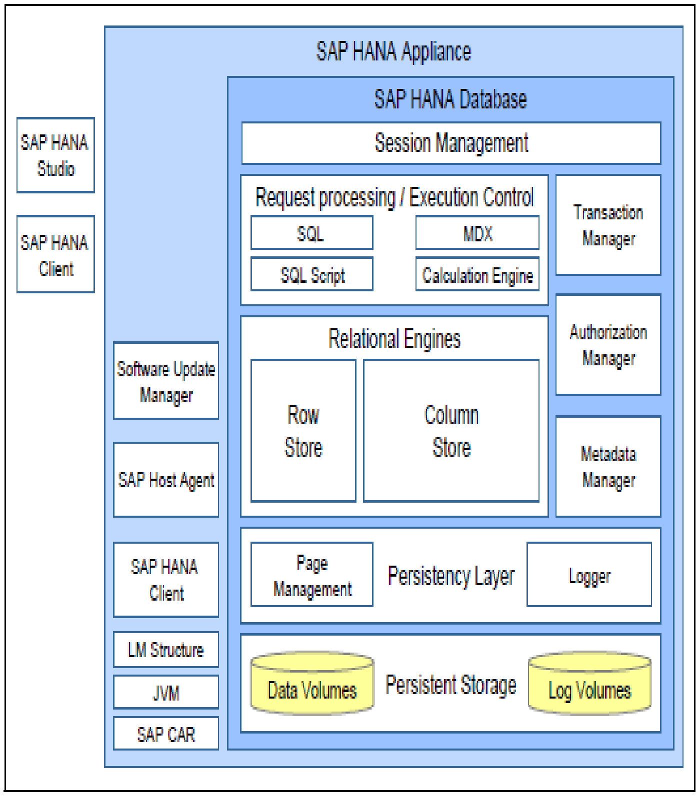

Fig. 1: SAP-HANA Architecture [1]

Following are the architectural components of SAP HANA:

A. Connection and Session Management

It component has the control over creating and managing sessions and connections. After the establishment of connection and session, it can be used by the client for SQL, SQL Script, MDX or other domain specific languages. For application specific calculations inside the database the SQL Script is used which provides optimization and parallelization for better efficiency and are translated by their specific compliers and are represented as Calculation Model which are further executed by the Calculation Engine [1],[3].

B. Transaction Manager

It ensures the ACID properties of the database transactions and maintains the track of running and closed transactions. It implements the MVCC for the concurrency control within the database that allows uninterrupted long running read transactions along with the update transaction.
It also allows temporal queries inside Relational Engine with combination of time-travel mechanism [6].

C. Authorization Manager

It has control over the unauthorized access of users by checking for the required privileges for the execution of requested operations. The privilege permits the execution of specified operations. The task of authentication is either performed by the database or an external authorization provider such as LDAP directory [1].

D. Metadata Manager

It does the task of managing the metadata and the definition of SQL Script functions. The different types of metadata are stored in one common catalogue for all underlying storage engines [1].

E. Relational Engines

Engines are classified as two database engines:

- The column-based store, which store the data into columns and are optimized for holding large amount of data which are aggregated and used in analytical operations. The tables present here are loaded in the memory at the time of start up or during the normal operations of database [6].

- The row-based store, which store data in rows and are optimized for write operations and have a lower compression rate and query performance in comparison to column-based store. The tables present here are loaded in the memory at the time of startup [6].

F. *Persistency Layer*

This layer ensures the durability and the atomicity of transaction. The relational engines share the persistency layer for the data persistency. It provides an interface for the manipulation of data leveraged by the storage engines and maintain the durability and atomicity of the transactions. By the persistency it means that the transactions are either completely executed or are completely rolled back and also that the database is restored to the most recent committed state. It uses write a head logs, save points and shadow paging for an efficient persistency [6].

G. *Persistent Storage*

It persists all the transactions committed in the database by the logger of the persistency layer in a log entry which are written in synchronization t the log volumes on the persistent storage [6]

III. RESEARCH METHODOLOGY

SAP HANA is most reliable system to complete business tasks. According to architecture, when some task or work has to accomplish its request is loaded into SAP HANA appliance segment.

A. *Task Completion with System Calling*

For completion all individual segments have to work to result the task. Indirectly SAP HANA database is interlinked with SAP appliance segments. These segments work according to request generated by SAP HANA appliance. Here for these individual request made by appliance segment goes to the proper environments through proper channel. If any work requires information from host agent, client info or SAP CAR, these are contacted to the appliance manager through individual system calls [1], [7].

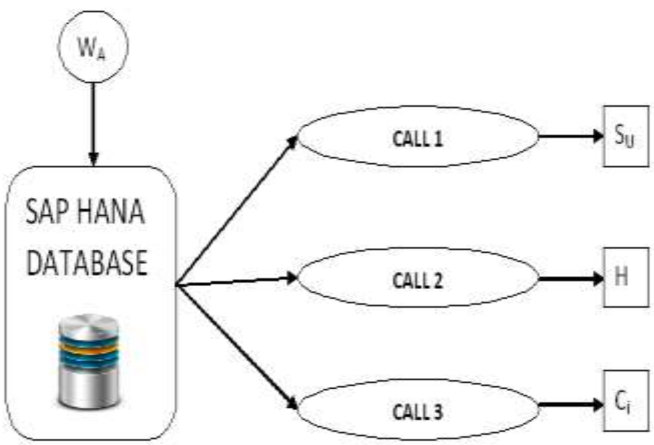

Fig. 2: Task completion with calling function

Where,
S_U- Software update
H- Hosting
C_I- Client info
W_A- Work has to complete under SAP HANA (appliance)

In the fig.2, work W_A has to be accomplish which requires information from three environments update S_U hosting (H) and client info fetch (C_I). Total calls (T_C) will be sum of individual calls required.

$W_A = S_U + H + C_I$
$T_C = C1+C2+C3$
$T_C = \sum C$

These individual tasks require system calls generated within SAP HANA environment. Numbers of calls are allocated with respect to number of request made by the appliance layer to the appliance segments. These calls execute individually which takes individual time also. System calls are considered for communication purpose only. After completion of these calls only communication process will be executed successfully. To execute further processes, first step is to complete communication process.

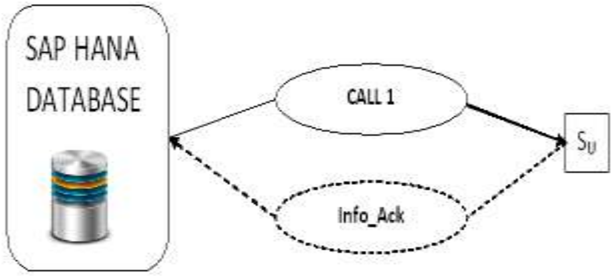

Fig. 3: Individual Communication calling

In fig.3 it's shown that how individual calls are executed with execution time. This time depends on the computing environment. Computing environment includes terms like types of computing algorithms and process execution techniques are applied. Maximum probability is, if more calls are generated more timing is required. Here efficiency is considered as E_F.

$T_C \alpha E_F$

$\sum C \alpha E_F$

It shows that till some pick level SAP HANA environment will work with proper efficiency. But after some peak level its performance will be reduced.

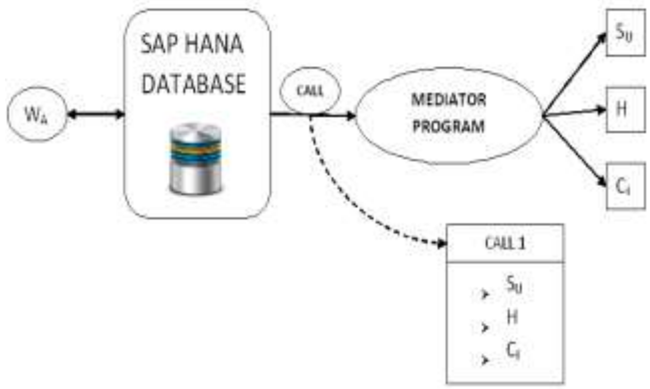

Fig. 4: Mediator concept in SAP-HANA architecture

B. Task Completion with Mediator Program

Existing environment can support till some extent. After reaching peak level time component increases. This paper is defined the term mediator which is one program considered to reduce call generation. Here single mediator program will include all tasks link within single cal, from SAP HANA appliance layer. As above mentioned, SAP HANA appliance layer completes its task by interlinking with SAP HANA database.

In fig. 4 it's shown that how mediator program is acting as middle channel to complete W_A work. This mediator program will include all list of information like S_U, H, C_I and so on. By this concept all communication execution will be done is single program only which will require less amount of time.

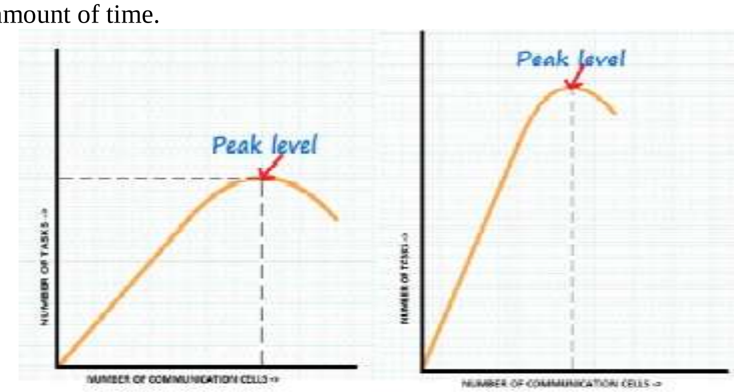

Fig. 5: (a) Program without mediator & Program with mediator

$$\frac{1}{C} \alpha M_c$$

It is shown in fig. 5, graph represents how efficiency level in SAP-HANA architecture can be improve by using mediator concept. So number of mediators will be inversely proportional to the number of calls. If the more mediator programs will have added for reserved segments for particular work, total execution process time will be reduced which will improve efficiency. Also process and tasks ongoing between SAP-HANA appliance and SAP-HANA database layer will be done with less processing time.

IV. CONCLUSION

Existing data mining approach in SAP-HANA was generic approach while making contact between appliance layer and database layer. For individual interaction between appliance segments and database layer, system generated calls are initiated. As the rate of number of calls generation increases, data mining approach was keen to be added with multiple function calling approach for better performances. After allocating mediator program for multiple segments, it can manage all database calling functions which increases peak level of data mining approach between appliance layer and database layer.

REFERENCES

[1] An Oracle White Paper. (2014). Analysis of SAP HANA High Availability Capabilities. Available: http://www.oracle.com/technetwork/database/availability/sap-hana-ha-analysis-cwp-1959003.pdf. Last accessed 22nd March 2016.
[2] Intel Real-Time Business Intelligence White Paper. (2014). Security in the Cloud for SAP HANA*. Available:http://www.intel.in/content/dam/www/public/us/en/documents/white-papers/cloud-security-xeon-e7-v2-sap-virtustream-paper.pdf. Last accessed 22nd March 2016.
[3] Sikka, V., Färber, F., Lehner, W., Cha, S. K., Peh, T., & Bornhövd, C. (2012). Efficient transaction processing in SAP HANA database. SIGMOD Conference (p. 731).
[4] Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., & Lehner, W. (2011). SAP HANA Database - Data Management for modern Business Applications. SIGMOD Record, 40(4), 45-51.
[5] Legler, T., Lehner, W., & Ross, A. (2006). Data mining with the SAP NetWeaver BI accelerator, 1059-1068.
[6] Marcus Paradies, Wolfgang Lehner, Christof Bornhövd. GRAPHITE: An Extensible Graph Traversal Framework for Relational Database Management Systems. SSDBM 2015, San Diego, USA, June 29 - July 1, 2015.
[7] Florian Wolf, Iraklis Psaroudakis, Norman May, Anastasia Ailamaki, Kai-Uwe Sattler. Extending Database Task Schedulers for Multi-threaded Application Code. SSDBM 2015, San Diego, USA, June 29 - July 1, 2015.
[8] Timur Mirzoev,Craig Brockman. (Feb2013). SAP HANA AND ITS PERFORMANCE BENEFITS. i-manager's Journal on Information Technology. 2 (1), p13-22.