ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 6, June 2013

Distributed SAP HANA Database for Efficient processing

Tusal Patel[1], Preeti Gupta[2], Nishant Khatri[3]

M.Tech (CSE), Amity School of Engineering & Technology, Amity University Rajasthan, Jaipur, India [1]
Asst. Professor, CSE Department, Amity School of Engineering & Technology, Amity University Rajasthan, Jaipur, India [2]
M.Tech (CSE), Amity School of Engineering & Technology, Amity University Rajasthan, Jaipur, India [3]

Abstract: SAP HANA is the main constituent of SAP HANA appliance for modern business processes in combination with OLAP and OLTP. SAP HANA database is the database that supports both OLAP and OLTP transactions. The paper highlights the effectiveness that can be achieved through distributed SAP HANA database at different servers, with multiple data processing engines and a distributed query environment for data processing as compared to centralized database approach**.**

Keywords: Distributed SAP HANA database, OLTP, OLAP, HANA database

I. INTRODUCTION

SAP HANA: High-Performance Analytic Appliance (HANA) is an In-Memory database from SAP to store data and analyze large volumes of non-aggregated transactional data in real-time with unprecedented performance, ideal for decision support & predictive analysis.[1]

The In-Memory computing engine is a next generation innovation that uses cache-conscious data-structures and algorithms leveraging hardware innovation as well as SAP software technology innovations. It is ideal for Real-time OLTP and OLAP in one appliance i.e. E-2-E solution, right from transactional to high performance analytics [2]. SAP HANA can also be used as a secondary database to accelerate analytics on existing applications.

In real world we have variety of data sources, e.g. unstructured data, operational data stores, data marts, data warehouses, online analytical stores, etc. To do analytics or knowledge mining from such big data at real time, number of hurdles like Latency, High Cost and Complexity were identified. Disk I/O was yet another performance bottleneck in the past. Though in memory computing was always much faster than disk I/O earlier, the cost of in-memory computing was prohibitive for any large scale implementation. Now with Multi-Core CPU and high capacity of RAM, the hosting of the entire database in memory is possible. So now in the changing scenario CPU waits for data to be loaded from main memory into CPU cache – which may create performance bottleneck .

Confidentiality Issues in Centralized HANA database:-

Big organizations treat data as an important asset. Most of the organizations using ERP, work with centralized database which may create bottlenecks due to high data traffic. Due to this some parameters like query processing, data availability, response time efficiency cannot be achieved as desired. Moreover data recovery after a case of failure also becomes costly and time consuming. The same applies to centralized HANA database.

Through this paper the advantage of incorporating distributed SAP HANA database to improve factors like data availability, query processing, data recovery, response time in big organization is brought forth.

.

II. MOTIVATION BEHIND DISTRIBUTED DATABASES

Over time, it became easier for IT to add hardware to the data center rather than to focus on making the data center itself more effective. And this plan worked. By adding more and more resources into the data center, IT ensured that critical applications wouldn't run out of resources. At the same time, these companies built or bought software to meet business needs. The applications that were built internally were often large and complex. They had been modified repeatedly to satisfy changes without regard to their underlying architecture. Between managing a vast array of expanding hardware resources combined with managing huge and unwieldy business software, IT management found itself under pressure to become much more effective and efficient. This tug of war between the needs of the business and the data center constraints has caused friction over the past few decades. Clearly, need and money must be balanced. To meet these challenges, there have been significant technology advancements including virtualization, service-oriented architecture, and service management. Each of these areas is intended to provide more modularity, flexibility, and better performance.

When an organization is geographically dispersed, it may choose to store its databases on a central database server or to distribute them to local servers (or a combination of both). A distributed database is a single logical database that is spread physically across computers in multiple locations that are connected by a data communications network. It should be noted that a distributed database is truly a database, not a loose collection of files. The distributed database is still centrally administered as a corporate resource while providing local flexibility and customization. The network must allow the users to share the data; thus a user (or program) at location A must be able to access (and perhaps update) data at location B. The sites of a distributed system may be spread over a large area (e.g., the United States or the world) or over a small area (e.g., a building or campus). The computers may range from PCs to large-scale servers or even supercomputers.

A distributed database requires multiple instances of a database management system (or several DBMSs)[1], running at each remote site. The degree to which these different DBMS instances cooperate, or work in partnership, and whether there is a master site that coordinates requests involving data from multiple sites distinguish different types of distributed database environments. It is important to distinguish between distributed and decentralized databases. A decentralized database is also stored on computers at multiple locations; however, the computers are not interconnected by network and database software make the data appear to be in one logical database. Thus, users at the various sites cannot share data. A decentralized database is best regarded as a collection of independent databases, rather than having the geographical distribution of a single database.

Various business conditions encourage the use of distributed databases:

• **Distribution and autonomy of business units:** Divisions, departments, and facilities in modern organizations are often geographically distributed, often across national boundaries. Often each unit has the authority to create its own information systems, and often these units want local data over which they can have control. Business mergers and acquisitions often create this environment.

• **Data sharing:** Even moderately complex business decisions require sharing data across business units, so it must be convenient to consolidate data across local databases on demand.

• **Data communications costs and reliability:** The cost to ship large quantities of data across a communications network or to handle a large volume of transactions from remote sources can still be high, even if data communication costs have decreased substantially. It is in many cases more economical to locate data and applications close to where they are needed .Also, dependence on data communications always involves an element of risk, so keeping local copies or fragments of data can be a reliable way to support the need for rapid access to data across the organization.

• **Multiple application vendor environments:** Today, many organizations purchase packaged application software from several different vendors. Each “best in breed” package is designed to work with its own database, and possibly with different database management systems. A distributed database can possibly be defined to provide functionality that cuts across the separate applications.

• **Database recovery:** Replicating data on separate computers is one strategy for ensuring that a damaged database can be quickly recovered and users can have access to data while the primary site is being restored. Replicating data across multiple computer sites is one natural form of a distributed database.

• **Support for both transaction and analytical processing:** The requirements for database management vary across OLTP and OLAP applications. Yet, the same data are in common between the two databases supporting each type of application. Distributed database technology can be helpful in synchronizing data across OLTP and OLAP platforms.

III. ADOPTING CENTRALIZED SAP HANA DATABASE ARCHITECTURE IN DISTRIBUTED SCENARIO

It is proposed to have the following architecture for distributed SAP HANA database which consists of multiple servers as shown in Figure 1.The most important component is the Index Server. Distributed SAP HANA database can be consists of Index Server, Name Server, Statistics Server, and Preprocessor Server.

1. **Index Server** contains the actual data and the engines for processing the data. It also coordinates and uses all the other servers.
2. **Name Server** holds information about the Distributed SAP HANA database topology. This is used in a distributed system with instances of HANA database on different hosts. The name server knows where the components are running and which data is located on which server.
3. **Statistics Server** collects information about Status, Performance and Resource Consumption from all the other server components. From the SAP HANA Studio we can access the Statistics Server to get status of various alert monitors.
4. **Preprocessor Server** is used for Analyzing Text Data and extracting the information on which the text search capabilities are based**.**

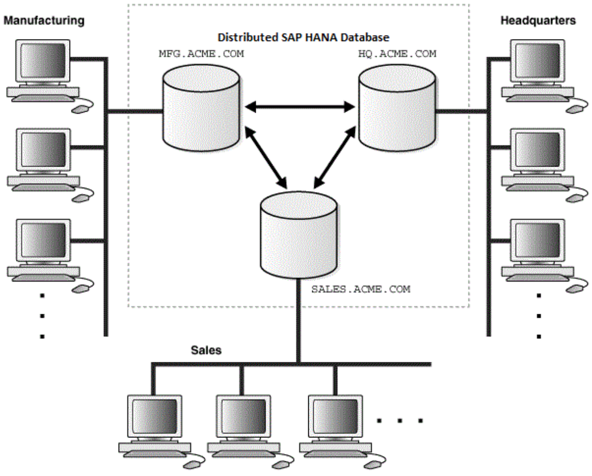

Fig.1 Distributed SAP HANA Architecture

The SAP HANA Appliance software supports high availability. SAP HANA scales systems beyond one server and can remove the possibility of single point of failure. So a typical distributed scale out cluster landscape will have many server instances in a cluster. Therefore a large table can also be distributed across multiple servers. Again queries can also be executed across servers. SAP HANA Distributed System also ensures transaction safety.

Some of the key features that can be further identified are

- "A" number of **Active** Servers or **Worker** hosts in the cluster.
- "B" number of Standby Server(s) in the cluster.
- **Shared file system** for all Servers. Several instances of SAP HANA share the same metadata.
- **Each** Server hosts an **Index** Server & **Name** Server.
- Only **one Active** Server hosts the **Statistics** Server.
- During startup one server gets elected as **Active Master**.
- The Active Master assigns a volume to each starting Index Server or no volume in case of cold Standby Servers.

IV. METHODOLOGY ADOPTED

Depending on the way data is distributed, most requests for data by users at a particular site can be satisfied by data stored at that site. This speeds up query processing since communication and central computer delays are minimized. It may also be possible to split complex queries into sub queries that can be processed in parallel at several sites, providing even faster response.

In this paper for exhibiting the effect of distributed database, one particular database is subjected to centralized HANA environment and the same database in subjected to a Distributed HANA environment by creating its two replicas on different server.

On applying the same query the response time is measured for both centralized SAP HANA and distributed SAP HANA database and on comparison it is found that distributed SAP HANA database returns better results in terms of response time for the query.

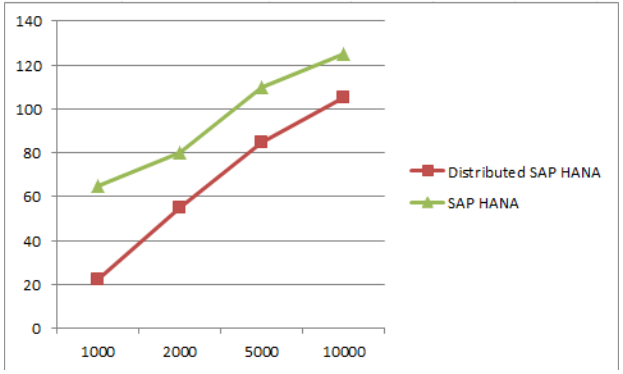

Fig-2, Response Time Graph

The graph in Figure 2 denotes the response time of distributed SAP HANA database compared to centralized SAP HANA database.

V. OTHER BENEFITS OF DISTRIBUTED HANA DATABASE COMPARED TO CENTRALIZED HANA DATABASE

1) **In case of any System fails.**

- High Availability enables the failover of a node within one distributed SAP HANA. Failover uses a cold Standby node and gets triggered automatically. So when a Active Server A fails, Standby Server N+1 reads indexes from the shared storage and connects to logical connection of failed server A.
- If the Distributed SAP HANA system detects a failure situation, the work of the services on the failed server is reassigned to the services running on the standby host. The failed volume and all the included tables are reassigned and loaded into memory in accordance with the failover strategy defined for the system. This reassignment can be performed without moving any data, because all the persistency of the servers is stored on a shared disk. Data and logs are stored on shared storage, where every server has access to the same disks.

- The Master Name Server detects an Index Server failure and executes the failover. During the failover the Master Name Server assigns the volume of the failed Index Server to the cold Standby Server. In case of a Master Name Server failure, another of the remaining Name Servers will become Active Master.
- Before a failover is performed, the system waits for a few seconds to determine whether the service can be restarted. Standby node can take over the role of a failing master or failing slave node.

2) Local control :-

- Distributing the data encourages local groups to exercise greater control over "their" data, which promotes improved data integrity and administration. At the same time, users can access nonlocal data when necessary. Hardware can be chosen for the local site to match the local, not global, data processing work.

3) **Modular growth :-**

- Suppose that an organization expands to a new location or adds a new work group. It is often easier and more economical to add a local computer and its associated data to the distributed network than to expand a large central computer. Also, there is less chance of disruption to existing users than is the case when a central computer system is modified or expanded.

4) **Lower communication costs :-**

- With a distributed system, data can be located closer to their point of use. This can reduce communication costs, compared to a central system

VI. CONCLUSION

The paper introduces the Distributed SAP HANA database architecture for optimizing query processing for reducing the overall query execution times. There are numerous advantages to distributed SAP HANA databases that were identified. The most important of these are increased reliability and availability of data, local control by users over their data, modular (or incremental) growth, reduced communications costs, and faster response to request for data.

SAP HANA database is already an efficient database but by converting it to distributed SAP HANA database further enhancement in the efficiency can be witnessed.

REFERENCES

[1] F. Farber, S. K. Cha, J. Primsch, C. Bornhovd, S. Sigg, and W. Lehner." SAP HANA database: data Management for Modern business applications." SIGMOD Rec., 40(4):45-51, Jan. 2012.

[2] H. Plattner. A common database approach for OLTP and OLAP using an in-memory column database. In Proceedings of the 35th SIGMOD international conference on Management of data, SIGMOD '09,

[3] F. Faerber, S. K. Cha, Vishal Sikka, and W. Lehner "Efficient Transaction Processing in SAP HANA Database –The End of a Column Store Myth" , SIGMOD '12

[4] Joos-Hendrik Boese, Christian Mathis, Cafer Tosun,Franz Faerber "Data Management with SAPs In-Memory Computing Engine" EDBT'12, March-2012, Berlin, Germany