

Improving Performance of Data Analytical Queries Using In-Memory Database Systems

Syed Ateeq Ahmed

Dept. of Computing, Middle East College, Oman

Abstract - Improving performance of data analytical queries needed a different approach than the online transaction processing queries. Traditional database systems which store data in row format in memory are more relevant for transaction processing requirements. To achieve significant performance gains of data analytical queries, columnar format is more relevant. In-memory database systems support storing data in columnar format and are more relevant for improving performance of data analytic queries. In this paper, architecture of in-memory database systems is discussed and compared with traditional database systems. This paper provides a solution for performance enhancements of data analytical queries and the need for using in-memory database systems in decision-making.

Key Words: Database Systems, In-memory database (IMDB), Random access memory (RAM), Online transaction processing systems (OLTP), Solid-state drives (SSD).

1. INTRODUCTION

Database management systems (DBMS) are software's used to create and work with databases. These database systems are very popular and used in almost all organizations where data storage, retrieval and processing of data are one of the key requirements of the organization.

Main memory plays an important role in the performance of the database. Database system caches the information in memory and to preserve persistency changes must be stored on disk. Compared to the memory access, reads from disk take considerably more amount of time around in the order of 10 milliseconds. Physical input and output also has an impact on CPU resources. It is highly recommended that repeatedly accessed objects if cached in memory will have a positive performance impact.

1.1 Type of memory and its impact

One of the reasons for usage of type of database system is based on which of type of storage is used to keep the data. Though there are various types of storage available, they can be classified in the three groups as solid-state drives (SSDs), hard-disk drives (HDDs) and random access memory (RAM). Each of these types of memories vary in access time, speed and price.

Over the years, as the size of memory is growing continuously and in contrast the price is falling drastically by a factor of 10 every 5 years [7]. The memory types considered are random access memory (RAM), solid-state drive and hard disk drive.

In case of in-memory databases, RAM is used to store copy of data and solid-state or hard disk can be used for persistent storage whereas traditional database systems uses RAM for caches and persists data is stored on SSD or HDD. RAM is faster than the SSD and HDD, whereas SSD is faster than the HDD. In terms of price, HDD is cheapest among the indicated types of memories. SSD is around ten times more expensive than an HDD and RAM is ten times more expensive than SDD. If the required access time is less than 1ms, RAM is the preferred choice. However, SSD can be an option if the access time is up to 100 MS. If the access time is not critical, and requirement is such that more than 100 ms is suffice, storing data in in hard disk drive can be considered.

Type of access, sequential or random also plays a role in choosing type of storage. Cost is directly proportional to the storage speed, cost will be higher if the storage is slower. Slow storage are useful for sequential access of data. If the acceptance level of sequential access of large data volumes is up to 100 MB/s, hard-disks will be an cost effective storage option.

Data that should be accessed online with less than 1 ms, it must be stored in the primary storage i.e. RAM. This kind of data access needs to be stored in an in-memory database as it is the most efficient solution for memory-resident data. Comparatively, a relation database, is several times slower. One of the main disadvantages of in-memory databases is their relatively poor performances in the multiple CPU core environments.

In any organization there will be data which need not be accessed online and the requirement is such that data access speed is acceptable up to 100 ms can be stored in a relational database or an in-memory database depending on certain factors. If the storage is not having any effect on the total cost, it can be stored in an in-memory database or vice versa. In-memory database systems sometimes are also used to store offline data as they will utilize the CPU more efficiently than the relational database systems. To avoid any further workload on CPU and disks, unstructured data in the organization for example, user profiles etc. can stored using the regular file systems. Usage of file systems in such a situations can help in improving the performance of the database systems. The historical data for analytical purpose can be stored in hard disks as the access type is mostly sequential.

1.2 Pros and Cons of Choosing an in-memory database system for an organization

The following factors must be considered while choosing the in-memory database systems for any organization [5].

- In-memory database systems are used to improve the performance of underlying queries which access data in the applications. Fewer activities and I/O operations are required to access the data due to the reduced instruction set.
- As the price of memory continue to decline and most of the servers are having 32 TB or more of memory, to achieve performance gains in-memory database systems can be considered.
- Most of the organizations are providing real-time interfaces and web-based applications, the need of speed is the key requirement. In such situations, high-end performance provided by the in-memory database systems can be useful.
- The current in-memory processing technology, issue of persistence is solved. The data in primary memory persists even in case of power failures.
- With increased data reliability and persistence, in-memory database systems are feasible for both analytical and transactional processing needs of organizations. Some of the examples can be networking and telecom, capital markets, defence and intelligence, gaming, real-time analytics etc.
- Part of the reason for not migrating to in-memory database is cost. Although memory is getting cheaper every year, it is still costlier than disk.
- Lack of in-memory database expertise and use of interfaces other than pure SQL can be barrier in migrating to in-memory database systems.
- Many still believe that the size of the in-memory database must be limited in size, however, some of the in-memory database systems support up to terabytes of data.
- Tests should be conducted to ensure that in-memory database systems are a viable option to obtain performance gains.

2. In-memory database systems and persistence

In case of in-memory database systems, complete dataset is kept in the primary memory. User requests for data access and updates to the existing data are done in the primary memory without involving the relatively slow disks. Persistence is a key requirement of a database system. To achieve persistence, all data is kept in memory, however, the transactions which changes the database state, are send to disk and stored in a transaction log. Transaction log can be seen in the figure 1.

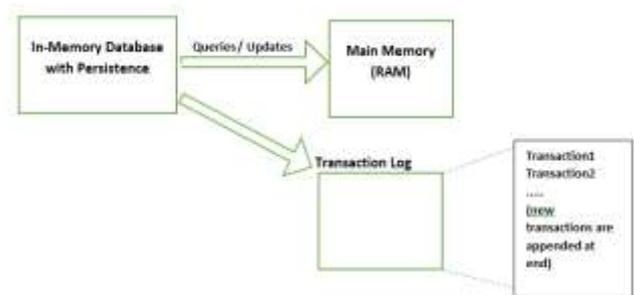


Fig -1: In-memory database with persistence

Even though the disk is used for storing transaction logs there will be no major impact on the performance as the transactions are appended at the end of the transaction log file which works in an append-only way. The hard disks work pretty faster when working in this fashion (append-only). Hard disks can be written with as fast as 100 MB/s when the data is appended at the end of the file. If used sequentially, hard disks can work at a very high speed, however, the disks will be slower when data is accessed randomly.

In case of solid-state drives, data access speeds will be much higher as there will be no moving parts like disks. In these devices, sequential access can be between 200-300 MB/s. Considering these, if a transaction size is of 100 bytes, then using solid-state drives, it can be a million transactions per second. This indicates, disks can never be a hindrance in for in-memory database systems. In-memory database systems, disk is not used when there no changes in the data and hence in case of changes in data, disk is used in the fastest possible way.

3. The Cloud and In-memory database systems

Cloud can be used to provide an excellent environment for efficient use of in-memory database systems. Organizations need not spend on purchasing large amount of primary storage, instead they can use the services provided in the cloud environment. In a cloud based environment failures can be prevented due to the availability of the redundant hosts and virtual machines. In such an environment, any failure due to RAM will not lead to a data loss. To achieve maximum performance gains, can combine in-memory database systems with cloud based environment.

3.1 Oracle Database In-Memory

Performance enhancements of analytical queries are achieved using the in-memory database aspect in Oracle 18c. This feature is very useful in taking real-time business decisions in an organization. Using this aspect, organizations improve productivity and increase competitiveness. It can useful in accelerating the data warehouse as well as OLTP databases. Oracle 18c supports dual-format architecture wherein data is maintained in a row-format for OLTP operations and in a column-format for analytical processing. The four key aspects of Oracle in-memory database which enables orders of magnitude faster analytic query processing are [2]:

- Compressed columnar storage
- Vector processing
- In-memory storage indexes
- In-memory optimized joins and reporting

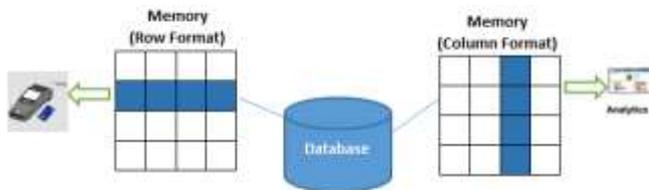


Figure: Oracle dual-format architecture

3.2 In-Memory Column Store

One of the key aspects in the in-memory database is in-memory column store. Classical relational database systems store data in a row or column format. In these systems data in both the memory and the disk are stored in the same format. Data stored in a row format gives optimum performance for transaction processing applications wherein updating a some of the columns in a few rows needs to access and modify small number of blocks. Using the same approach for analytical processing results in a poorer performance as the analytical workloads need to access few columns, however, scans the entire data set.

Using columnar format is efficient for analytical workloads. In a columnar database, columns but not rows are stored continuously. In a columnar format, as the columns are stored separately, analytical query needs to access only the required columns thereby avoiding the scan of data which is not required. For example, a report to compute and show the sales total by city can quickly process large number of rows, however, accesses only a few columns.

4. The Oracle Database In-Memory Solution

Most of the database vendors give one option to the customers to either choose the columnar or the row format. In such cases, if the data format is chosen as columnar, then the columnar format is applicable in both the disk as well as the memory. This indicates advantages of both the columnar and row formats cannot be obtained at the same time. In case of databases which have both OLTP and data analytical needs, performance problems will arise due to the single format.

The Oracle in-memory aspect provides a complete solution for both the analytical queries and mixed-use databases. Oracle's in-memory aspects like IM column store, improvised query optimization techniques, and availability provides the solution for faster execution of data analytical queries of data warehouses without sacrificing the performance requirements of OLTP applications. The feature of compressed columnar format enables faster scan, queries,

aggregates and joins. The IM column store is for improving performance of analytical queries, the dual-memory format architecture can indirectly improve OLTP performance. The in-memory feature in Oracle database is easy to implement and requires no changes in the application.

Scans done using the in-memory columnar format are faster than the row-format due to the following reasons [1]:

- Avoids the overhead due to buffer cache.
- Only the required columns are scanned avoiding to scan entire rows of data.
- In the in-memory column format, compression is done to speed-up the scans.
- Vector scans by a CPU core are orders of magnitude faster than row scans.

For example suppose a user executes the following ad hoc query:

```
select inv_no, cust_name, supp_no
from sales_invoice
where item_no between 101 and 151;
```

In case of traditional database systems using the buffer cache, data is fetched using index to find the item_no, database uses the rowids to fetch and transfer the rows from disk into the buffer cache and then discard the unneeded data. Data scans in row formats require multiple CPU instructions which may have an impact on the performance. However, data scan using columnar format pipelines only the required columns to the CPU, increasing the performance.

5. CONCLUSION

In this research paper, traditional and in-memory database systems are compared. To achieve performance gains of data analytical queries and decision making requirements of the organization, Database Administrators (DBA) can use in-memory database systems. In-memory database systems also work efficiently for OLTP systems, DBAs must implement relevant data format i.e. row or columnar as per the requirements.

REFERENCES

- [1] Ashdown, L. (2019) Oracle ® Database: Database In-Memory Guide. Oracle Corporation
- [2] Lahiri, T. (2015) 'When to Use Oracle Database In-Memory'. Oracle Corporation [online]
- [3] Plattner, H. (2013) A Course In-Memory Data Management. Springer
- [4] D'Souza, S. (2012) In-memory database technology gains ground, but challenges remain [online] available from < <http://www.computerweekly.com/feature/In-Memory>

memory-database-technology-gains-ground-but-challenges-remain [2 June 2019]

- [5] Mullins, C.S. (2015) How to determine if an in-memory DBMS is right for your company [online] available from <
<https://searchdatamanagement.techtarget.com/feature/How-to-determine-if-an-in-memory-DBMS-is-right-for-your-company>> [25 June 2019]
- [6] Mullins, C.S. (2017) What is an In-Memory Database System? [online] available from <
<http://www.dbta.com/Columns/DBA-Corner/What-is-an-In-Memory-Database-System-119241.aspx>> [2 July 2019]
- [7] Anikin, D (2017) Choosing Between an In-Memory and a Traditional DBMS [online] available from <
<https://dzone.com/articles/when-and-why-i-use-an-in-memory-database-or-a-trad>> [13 July 2019]
- [8] Anikin, D (2017) What an In-Memory Database is and How it Persists Data Efficiently [online] available from <
<https://dzone.com/articles/what-an-in-memory-database-is-and-how-it-persists>> [25 July 2019]