MJRET
Open Access
Multidisciplinary Journal of Research in Engineering and Technology
Journal homepage: www.mjret.in
ISSN:2348 - 6953

FINANCIAL ARCHITECTURE MANAGEMENT FOR GLOBAL BANKING USING APACHE CASSANDRA

Somesh Biswas, Sahadev Bite, Abhishek Bhavsar, Harshal Kolhe

Genba Sopanrao Moze College of Engineering
Savitribai Phule Pune University
Pune, India

Abstract: *Over decades, Oracle has been the top relational database (RDBMS) use to support key global banking systems. However, today's always online global banking world has bought a substantive change in how banking sectors and IT professionals must manage data and use it to achieve maximum data integrity. This paper looks forward at why NOSQL technology like Apache Cassandra is becoming the first and best database choice over Oracle for online banking application providing guidelines for when a legacy RDBMS like Oracle should be used and when NOSQL is required.*

Keywords: *Apache Cassandra, Big data, CQL, Survey*

1. INTRODUCTION

Literally every IT professional is acquired with Oracle RDBMS for banking sectors. Oracle owns more than 48% of database market. While Oracle being a solid RDBMS that performs well for which it was designed (ERP and Banking applications) even its strongest supporters admit that it is not architecture to tackle the new wave of big data over online applications developed today. The mainframe to client-server to web/mobile has resulted in the generation of countless application and exploiting data volumes. Modern banking business sectors today need to manage big/fast data and always online application that necessitate a different set of technologies (NOSQL) that are replacing Oracle in many situations and becoming the first choice for database management. This paper demonstrates the application requirement of yesterday's banking application database with todays and takes a look at difference between Oracle and Cassandra

2. LITERATURE SURVEY

It is usually found that often working with big/fast application for banking sector & IT professional becomes complex for data handling and processing. One reason modern businesses switch from Oracle to NoSQL is because the underlying RDBMS does not often support the architectural requirements of modern online applications. Today's online system necessitate a divide-and-conquer configuration for processing both big and fast data, which comes in from millions of user interfaces from many different locations. The scale- up, master-slave, non-distributed architecture of Oracle was never designed for such use cases and therefore falls short of what modern online applications need. It becomes hectic for banking sectors and IT professionals integrate large amount of data from multiple sources and process records in a distributed manner. Cassandra extends the concept of "eventual consistency" in NoSQL databases by offering tunable consistency. For any given read or write operation, the client application decides how consistent the requested data should be. [1]

3. COMPARISON:CONSIDERATIONS AND FEATURES

Business Considerations: The question business leaders should ask when it comes to deciding whether a NoSQL database like Cassandra is suited for a particular application/use case over a traditional database like Oracle are the following

- Do you need to keep the application always online and serving customers?
- Do you need to serve customers with multiple interfaces ad in multiple locations?
- Do you need to consume and deliver lots of data very quickly?
- Do you need to easily add database capacity to handle increasing customers?
- Do you need to manage different type of data?

If the application being considered delivers multiple affirmatives, the NoSQL should be considered for all/part of solution.

Technical Considerations: The technical considerations for determining whether NoSQL should be used for an application reflect the business questions:

- Do you need continuous availability with redundancy in both data and function across one or more locations?
- Do you need a database that runs over multiple data centers/ cloud availability zones?
- Do you need to handle high velocity data coming in via sensors, mobile devices and the like and have extremely right speed and low latency query speed?

- Do you need to run different workloads (e.g. online, analytics, search) on the same data?
- Do you need to manage a widely distributed system with minimal staff?

Apache Cassandra Features:

- Massively scalable architecture – a master less design where all nodes are the same.
- Linear scale performance – online node additions produce predictable increases in performance.
- Continuous availability – redundancy of both data and function mean no single point of failure.
- Transparent fault detection and recovery – easy failed node recovery.
- Flexible, dynamic schema data modeling – easily supports structured, semi-structured, and unstructured data.
- Guaranteed data safety – commit log design ensures no data loss.
- Active everywhere design – all nodes may be written to and read from.
- Tunable data consistency – support for strong or eventual data consistency.
- Multi-data center replication – cross data center and multi-cloud availability zone support for writes/reads built in.
- Data compression – data compressed up to 80% without performance overhead.
- CQL (Cassandra Query Language) – an SQL – like language that makes moving from an RDBMS very easy.

4. REPLACEMENT &CO-EXISTENCE

Once online businesses realize they need to move from Oracle, they experience two basic implementation scenarios:

1. New online applications – these involve completely new applications whose requirements dictate the use of NoSQL over a legacy RDBMS like Oracle. They may or may not involve polyglot persistence approach (i.e. one that uses a combination of relational and NoSQL technology).
2. Existing online applications – these are legacy applications that are morphing into big/fast data systems and require NoSQL in either a singular or polyglot persistence manner.

The first situation normally doesn't require a rip/replace approach of both existing Oracle code and data to NoSQL; however the second often times, if not always, does. In that case,

how does an IT shop practically go about making the move from Oracle to something like Cassandra and DataStax Enterprise?

Customer Examples Who Have Moved From Oracle: DataStax has numerous customers who have either moved existing applications from Oracle to DataStax Enterprise or who use DataStax Enterprise in conjunction with Oracle. Some of these customers include the following.

- ***Netflix:*** Netflix stores 95% of their data on Cassandra in the cloud (AWS), having moved much of it from Oracle. Please see the Netflix case study for more information.

- ***eBay:*** eBay uses DataStax Enterprise for many use cases including fraud detection, time series data management, messaging and more, with a number of systems being moved from Oracle. eBay stores more than 250TB of data in DataStax Enterprise across three data centers, and sees 9 billion writes and 5 billion reads per day flow through their DataStax Enterprise clusters. Please see the eBay case study for more information.

- ***OpenWave:*** OpenWave moved its messaging store architecture from Oracle to DataStax Enterprise to save money and so its platform could scale and perform better.

5. CONCLUSION

There is no argument that Oracle is a strong RDBMS that well serves the use cases for which it was originally designed. But for IT professionals who are either planning new big/fast data applications or have existing Oracle systems that have begun to break down under big data workloads, a move to DataStax Enterprise and Cassandra makes both business and technical sense. Switching to a modern, big data platform like DataStax Enterprise will future-proof any application, and provide confidence that the system will scale and perform well now and into a demanding future.

REFERENCES

[1] *Artem Chebotko, Andrey Kashlev and Shiyong Lu, A Big Data Modeling Methodology For Apache Cassandra, IEEE International Congress on Big Data, 2015.*

[2] *F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone, "Database design for NoSQL systems, "in Proceedings of the 33rd International Conference on Conceptual Modeling, 2014, pp. 223–231.*

[3] *[Cassandra Query Language, https://cassandra.apache.org/doc/cql3/ CQL.html.*

[4] *Comparison Oracle with DataStax/Apache Cassandra,https://www.datastax.com/wp-content/uploads/2013/11/WP-DataStax-Oracle.pdf.*