Implementation of Version Stamps from NOSQL Databases using Java

Dipali Meher, PhD
Assistant Professor
Modern College of Arts, Science and Commerce, Ganeshkhind, Pune 16

Sakshi Pawar
Student
Modern College of Arts, Science and Commerce, Ganeshkhind, Pune 16

Akanksha Gore
Student
Modern College of Arts, Science and Commerce, Ganeshkhind, Pune 16

ABSTRACT
In 1990, due to the emergence of cluster technology, RDBMS databases failed and NOSQL databases emerged. NOSQL databases provide scalability and performance, and due to their speedy development, they were used by all database administrators. In today's cloud computing environment, NOSQL databases are used instead of relational databases. Due to their semi-structured and unstructured properties, they can handle many critical applications. Relational databases use table structure, which is the backbone of structured query language, whereas query language used by NOSQL databases does not have any standard. NOSQL databases handle structured, semi-structured, and unstructured data. By using fixed schemas, tables can be created, but in NOSQL databases, there is no fixed schema. The CRUD operations can be done on the fly. RDBMS uses ACID properties on transactions, but in NOSQL databases, there is the CAP theorem. In any kind of database, you think that consistency is the main feature that it should provide. But it is very difficult to maintain this feature due to the schema less nature of NOSQL databases. Every database is identified by or works well due to its consistency feature. The databases should be consistent if the data written to them must be valid according to all defined rules, including tables, primary keys, foreign keys, constraints, cascades, triggers, or any combination thereof. *The CAP* theorem in NOSQL databases states that database administrators can only achieve at most two out of three guarantees for a database. (either or and & consistency/availability/partition tolerance). Version stamps are the most important topic used in NoSQL. Version stamps help you detect concurrency conflicts in NOSQL databases. In between read and write operations performed by the user, it has to check the version stamp so that data between read and write is not updated. to ensure nobody updated the data between your read and write. There are various methods to implement version stamps like counters, GUIDs, content hashes, timestamps, or a combination of these. Version stamps provide a mechanism for detecting concurrency conflicts. This paper focused on various methods used to create the version stamps and their advantages and disadvantages. Among these counter methods is simple and widely used. The author had tried to implement the counter method of the version stamp using the Java language.

General Terms
Version stamps

Keywords
NOSQL, Version Stamp, ACID, GUID, timestamp

1. INTRODUCTION
NoSQL, or "Not Only SQL," is a modern approach to database management that offers flexible and scalable solutions for handling large and diverse data like unstructured or semi-structured data and can adapt to changing data requirements. They are often used in machine learning, various web applications, and real-time analytics, providing high performance and horizontal scalability. NOSQL databases has various types, such as document-based, key-value, column-oriented, and graph-databases. Each database is tailored for its specific use. In essence, NOSQL databases provide versatile options for storing and retrieving data in a dynamic and data-intensive digital landscape. Such NOSQL databases are now used in machine learning, artificial intelligence, and deep learning systems [9].

Types of NOSQL Databases:

It is a nonrelational database. Unlike storing the data in rows and columns (tables), it uses the documents to store the data in the database. It uses JSON, BSON, or XML documents to store the data.

Key value store is the simplest form of NOSQL databases. Every data element in the database is stored in key-value pairs. The key is unique and it is associated with some value in database. Simple data types like strings and numbers or complex objects can be used to represent value.

A column-oriented database stores the data in columns instead of rows. It allows to run analytics on a small number of columns, so user can read those columns directly without consuming memory with the unwanted data. Data retrieval speed is high in columnar databases.

Graph-based databases— It focuses on the relationship between the elements. Nodes are entities and relationships are cardinalities.

CAP Theorem—The CAP theorem, an essential concept in the world of NoSQL databases, revolves around three crucial properties: consistency, availability, and partition tolerance. It essentially says that in a distributed database system, you can't have all three properties at their highest levels simultaneously. You must make trade-offs.

Consistency (C) means that all nodes or replicas in the database will have the same data, and any query will return a consistent view.

Availability (A) ensures that every request made to the database system gets a response, even if some nodes are temporarily unavailable.

Partition tolerance (P) relates to the system's ability to continue functioning when network partitions occur, which can temporarily isolate nodes from each other.

Clustering IN NOSQL— Clustering in databases involves

organizing data into groups based on specific criteria, making it easier to manage and retrieve information. These criteria can include time, content, dependencies, or other factors. Clustering helps users and systems navigate data efficiently and understand relationships between different data elements. It's especially valuable in large datasets and distributed systems for optimizing data retrieval, analysis, and maintaining data consistency.

Many NOSQL databases don't support transactions. The base for the consistency feature is transactions. As RDBMS strongly supports the transaction concept, they provide strong consistency. The transaction feature also helps database developers and programmers code for consistency maintenance in any DBMS application. With the help of aggregate concept NOSQL databases support atomic updates. It is seen that transactions also have limitations. Human interventions are also needed while doing atomic updates within transactions. Such human interventions will keep transactions open for a long time. Version stamps are used to handle such situations.

2. VERSION STAMPS

Clustering IN NOSQL— Clustering in databases involves organizing data into groups based on specific criteria, making it easier to manage and retrieve information. These criteria can include time, content, dependencies, or other factors. Clustering helps users and systems navigate data efficiently and understand relationships between different data elements. It's especially valuable in large datasets and distributed systems for optimizing data retrieval, analysis, and maintaining data consistency.

Many NOSQL databases don't support transactions. The base for the consistency feature is transactions. As RDBMS strongly supports the transaction concept, they provide strong consistency. The transaction feature also helps database developers and programmers code for consistency maintenance in any DBMS application. With the help of aggregate concept NOSQL databases support atomic updates. It is seen that transactions also have limitations. Human interventions are also needed while doing atomic updates within transactions. Such human interventions will keep transactions open for a long time. Version stamps are used to handle such situations.

Version stamps are used to detect concurrency conflicts. It is associated with each data item. When an item gets updated the version stamp gets updated. Before updating a data item, a process can check its version stamp to see if it has been updated since it was last read.

Implementation methods of version stamp:

1)Counter Method : Value of counter increments when cluster updates the database item. By looking at recent (the value is highest) value of counter database administrator will come to know that which update is recent. Servers should generate counter value. Single master is needed to decide that value of counter so that value should not be duplicated by replicas. For example. R1 to R6 are replicas and M1 is master (which replica R7)C is counter variable. Its value is 3. a=1(database item).All replicas are having database item value i.e. a= 1. Following figure shows all replicas are having value of database item a=1 and C=3. The above situation is shown in following figure 1.

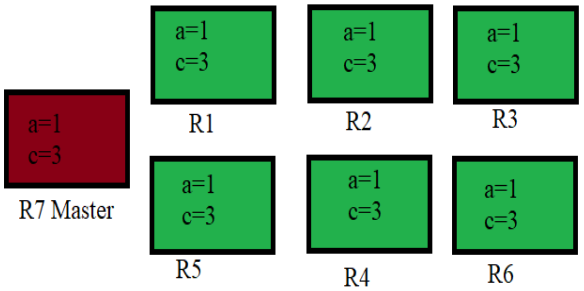

Fig 1: Situation of Replicas: Master slave Replication Before Updations(Counter Method)

Now, Replica R3 want to update database item value to 6 i.e., a=6. So, its counter value should be incremented first. Server at master side will increment the value of counter by 1. i.e. C=4. So, the values of database item and counter are reflected at replica R3 and Master. Now C=4 is the recent (highest counter value) value, so this update is recent. Master will propagate to all replicas this update.

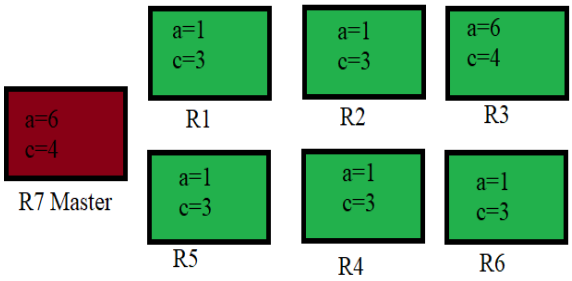

Fig 2: Situation of Replicas: Master slave Replication After updations (Counter Method)

Now if any other replica wants to update database item value, then its counter value should be updated first and communicated to master then master will communicate it to the other replicas. This is shown in figure 2.

2) GUID (Guaranteed Unique ID) Method:

The large random number which is to be used to be assured as unique. This random number is created by taking combination of dates, hardware information or any other way. These GUID i.e., random numbers will never be same. The disadvantage is that the random number is very large so comparison of then for their recentness will be difficult. For example. In this example GUIDs are generated with online GUID generator application. The link is as https://www.guidgenerator.com/online-guid-generator.aspx

3) Hashing: Hashing technique will be used to has the contents of resource. The hash key size will be big and content hashing will be done with globally unique GUID. GUIDs are deterministic. i.e., any replica can generate the same content of hashing for same resource of database item. For example, consider database item a=1, replicas R1 to R6. R7 replica is Master. Replica R1 wat to update database item a to value 43. With the help of simple hashing function of mod 10 (10 is hash key) it will generate bucket. i.e., 43%10= 3 Now it is easy to find out that replica R1 is having recent updates (as the word deterministic). The following figure 3 will explain this concept.

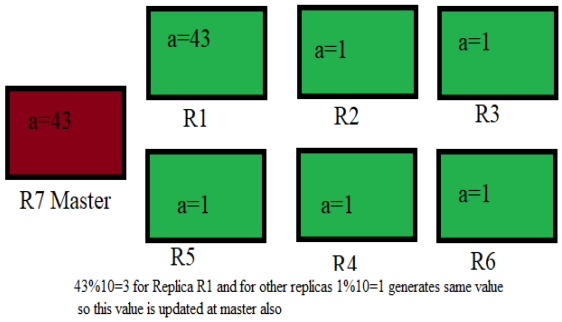

Fig 3 : Situation of Replicas: Master slave Replication (Hashing Method)

4) Timestamp Method: If any update is made timestamp will be checked. Their working is same as counters, and they can be compared for their recentness. In this situation many replicas(machines) can generate timestamps. One thing must keep in mind that all machines' clocks should be synchronized with each other. If any replica will have bad clock (or its clock is not working properly in synchronization manner) then data corruption problem will arise. Database administrators must check granularity of timestamps otherwise timestamps will also get duplicated. Timestamps are good if milliseconds precision will be used. The following figure 4 will show first situation that all timestamps are of all replicas are same.

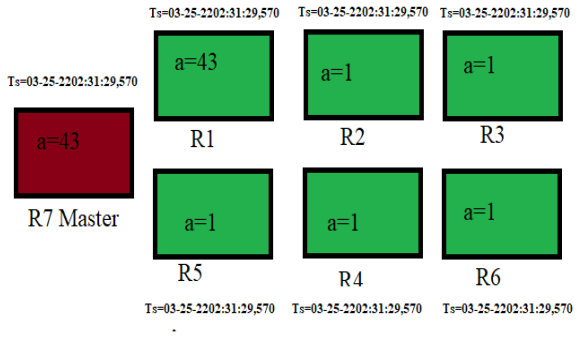

Fig 4: Situation of Replicas: Master slave Replication Before updations(Timestamp Method)

Now, Replica R2 want to update database item value to 4 i.e., a=4. R2s timestamps is by millisecond updated and it is now TS=-3-25-22:02:31:29,571 so it is compared with other replicas time stamp and 571>570 (with millisecond precision) so R2 contains most updated value. It is explained with following figure 5.

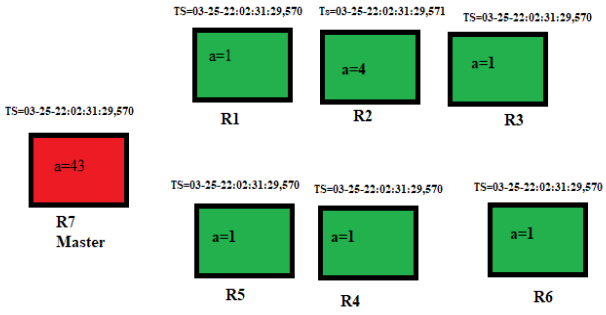

Fig 5: Situation of Replicas: Master slave Replication After updations(Timestamp Method)

A combination of above techniques can be done to make composite stamp. CouchDB database will be using combination of counter and content hashing.

3. LITERATURE REVIEW

Concurrency control means using same database at the same time by different transactions or clients. When different users try to update same database at the same time conflict in database may arise. To avoid these conflicts different methods such as serializability, two phase locking mechanisms were used in RDBMS. But due to lack of providing clustering concept NOSQL databases came into picture. They are 21st century web estate databases. As all know many NOSQL databases may not provide ACID properties, but they provide CAP theorem. As NOSQL databases are using CAP theorem in this Availability feature is provided using replication and sharding concepts. This theorem is proposed by Eric Brewer [1]. NOSQL databases also provides schema-free databases and scalability [2]. They are easily scalable for web applications. Replication is not useful for arbitrary partitions so P.S. Almeida; C. Baquero; V. Fonte et.al authors suggested instead of using counter method version tracking method is provided [3]. GUID method is used by author Paulo Sérgio Almeida, Carlos Baquero & Victor Fonte for version stamp creation and updations [4]. Version stamps were also used in object databases for object addressing, concurrency control, access authorization and other object management problems [5]. Author O'Neill, Melissa Elizabeth in their thesis version stamps for functional arrays and determinacy checking: two applications of ordered list for advanced programming languages mentioned that fat-elements method for providing functional arrays and the LR-tags method for determinacy checking[6]. The databases which are on cloud used logical clocks [7].

4. RESEARCH GAP

After literature review it has been found that there are various methods to maintain concurrency in NOSQL databases. From all these method counter methods are suitable and very useful.

5. RESEARCH METHODOLOGY

Authors have done partial programming to apply counter method for version stamps in Java programming. Array concept is used to implement this method. The zeroth index of an array-maintained version stamp of every replica. Number of arrays are created as replicas. When value in array is updated then its version stamp is also getting updated[8]. Following diagram 1.1 will give you clear ides of system architecture of version stamp counter method.

6. SYSTEM ARCHITECTURE

The following figure 6 explains the system architecture of counter method using Master Slave Replication.

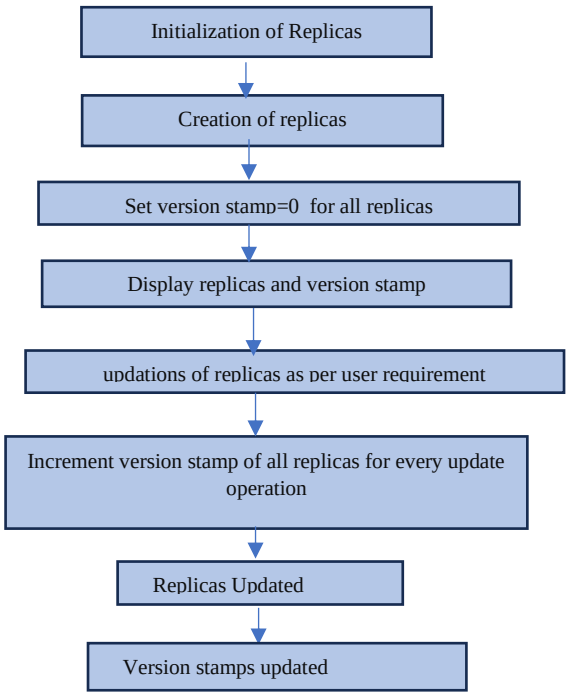

Fig 6: System Architecture

Partial implementation of counter method:

Java Language is used to code above scenario. Due to garbage collection in Java it will become easier to allocate and update values in replicas .Following figure 7 and 8 shows the output of our code snippet

```
Enter how many replicas you want : 5
Enter the size of replicas: 4
Enter  values for the  array :
10 3 6 -5
Array  elements : 0 10 3 6 -5
Array  elements : 0 10 3 6 -5
Array  elements : 0 10 3 6 -5
Array  elements : 0 10 3 6 -5
Array  elements : 0 10 3 6 -5
```

Fig 7: System Architecture(Running Mode)

```
3
enter which index you want to update=
2
which value u want to update=
44
replicas are updated 1times
Array  elements : 1 10 44 6 -5
Array  elements : 1 10 44 6 -5
Array  elements : 1 10 44 6 -5
Array  elements : 1 10 44 6 -5
Array  elements : 1 10 44 6 -5
which replica you want to update=
```

Fig 8: System Architecture(Running Mode)

Following table 1 shows the time complexity of all methods used in creation of version stamps. Counter method has high time complexity but it is easier to implement than other methods.

Table 1: Comparison of time complexities of different methods.

Method	Time complexity
Counter method	O(n2)
Hashing method	O(n)
Timestamp method	O(n)
GUID method	O(1)

7. CONCLUSION

Version stamps in NoSQL databases are essential for detecting concurrency conflicts, managing causal histories, and ensuring data consistency. They help identify conflicts, track how events are related, and control concurrent updates. With version stamps, databases can resolve conflicts, maintain data consistency, and optimize performance by organizing data versions effectively. Due to the simplicity of the counter method, this method is used worldwide to maintain version stamps in NOSQL databases. Authors have tired of code version stamps and their partial implementation in the Java language. The time complexity of an algorithm is $O(n^2)$. The Java language already has garbage collection facility so implantation of the version stamp concept becomes quite easy. While coding in Java author have used the concept of static memory allocation so in the future, the same concept can be applied using dynamic implementation using memory, i.e., linked lists. While designing NOSQL databases implementation of version stamps needs to be done as database updations is crucial part. New emerging NOSQL databases can use Java or any Object-Oriented language for version stamp implementation.

8. REFERENCES

[1] Brewer, Eric A.: Towards Robust Distributed Systems. Portland, Oregon, 2000, Keynote at the ACM Symposium on Principles of Distributed Computing (PODC) on 2000-07-19. http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pd

[2] S. Banerjee and A. Sarkar,(2016), "Logical level design of NoSQL databases," 2016 *IEEE Region 10 Conference (TENCON)*, Singapore, 2016, pp. 2360-2365, doi: 10.1109/TENCON.2016.7848452.

[3] P. S. Almeida, C. Baquero and V. Fonte, "Version stamps-decentralized version vectors," *Proceedings 22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 544-551, doi: 10.1109/ICDCS.2002.1022304.

[4] Almeida, P.S., Baquero, C., Fonte, V. (2007). Improving on Version Stamps. In: Meersman, R., Tari, Z., Herrero, P. (eds) On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. OTM 2007. Lecture Notes in Computer Science, vol 4806. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-76890-6_29

[5] Cellary, W., & Jomier, G. (1990). Consistency of Versions in Object-Oriented Database Version Approach. Proceedings of the 16th International Conference on Very Large Data Bases, 432–441

[6] ONeil Melissa, (2000),Version stamps for functional arrays and determinacy checking: Two applications of ordered list for advanced programming languages

[7] Ricardo Jorge Tomé Gonçalves,(2011),Logical Clocks for Could Databases, *ProQuest.*

[8] http://www.cs.gordon.edu/courses/cps352/2015-spring/lectures/13_NoSQL_Databases.pdf

[9] Dipali Meher, Baljeet Kaur,et.al, (2023)," Databases for machine learning: A journey from SQL to NOSQL", Recent Advances in Material, Manufacturing, and Machine Learning,(RAMMML-22), Volume 1, available at: https://www.taylorfrancis.com/chapters/edit/10.1201/9781003358596-26/databases-machine-learning-journey-sql-nosql-dipali-meher-baljeet-kaur-alaknanda-pawar-sheetal-parekh