International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Enhancing Cloud System Resilience: A Case Study of Amazon DynamoDB with Active - Active Compute Writer and Active - Passive Data Layer Strategies

Akshay Prabhu

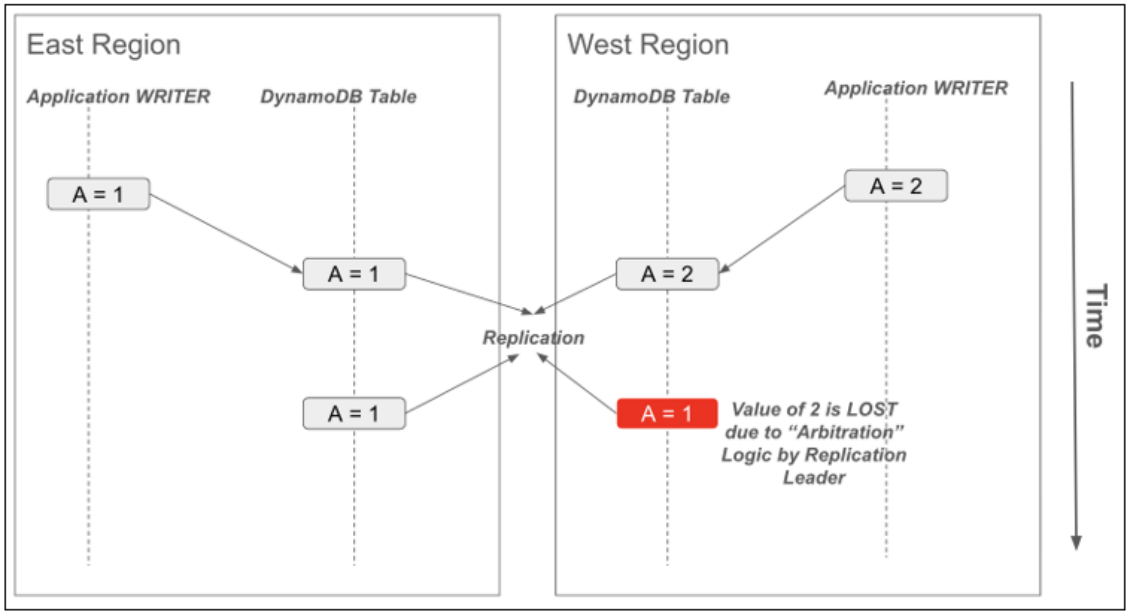
Abstract: ***Amazon DynamoDB, a NoSQL cloud database service, is renowned for delivering consistent performance at any scale and offers global replication capabilities that facilitate the development of resilient systems. This paper investigates the application of DynamoDB in constructing highly resilient cloud systems, focusing on the strengths and weaknesses of its global tables and data replication strategies. The study particularly addresses issues such as dirty or stale reads associated with multi - region writes into DynamoDB and proposes practical solutions, including alternative technologies and architectural approaches, to improve data consistency and system resilience. By evaluating these strategies, this paper aims to provide insights into optimizing DynamoDB's deployment to ensure data integrity for real - time use cases and optimize data consistency across multiple regions. This study is significant for cloud architects and engineers who aim to leverage Amazon DynamoDB's capabilities while ensuring high levels of data consistency and system resilience for real - time use cases and data reads.***

Keywords: Amazon DynamoDB, cloud resilience, multiregion writes, active replication, data consistency

1. Problem Statement

Each DynamoDB table consists of many items distributed across numerous partitions. Each partition is managed by a group of clusters. DynamoDB's Global Table Replication provides built - in replication across AWS regions using an internal policy that designates a single partition leader per region for each table. During write operations, requests are directed to the partition leader within the same region. This setup guarantees strong consistency for reads within a single region. However, a gap emerges when enabling writes across multiple regions, as data replication across regions relies on random "arbitration" logic to determine which write is considered final.

For example, the value of "A" from a writer in the West may be ignored even though it occurred simultaneously with a write operation in the East. This is due to the random "arbitration" logic employed by the partition leader to decide which write wins. As a result, any read at that point in time may return stale data. These side effects of enabling multi - region writes in DynamoDB can significantly impact overall data integrity, especially for applications requiring strongly consistent reads for real - time use cases.



2. Implications

The key implication of multi - region writes in DynamoDB is the compromise on strong read consistency. Applications that require strongly consistent reads must avoid enabling multi - region writes to ensure reliability. The risk of stale or inconsistent reads arises because replication across regions does not guarantee immediate consistency due to inherent

replication delays. Consequently, while DynamoDB offers global services, leveraging multi - region writes can lead to significant challenges in maintaining consistent data across regions.

Options for Enhancing Resilience

1) Active - Passive Configuration: Utilizing DynamoDB with an active - passive configuration involves a single active region where writes occur, while other regions serve as passive read replicas. This approach ensures strong consistency for reads and reduces the complexity associated with multi - region writes. It is advisable to implement this configuration if your application depends on strongly consistent reads, conditional expressions, or multi - item transactions.
2) Parallel Stacks Approach: Creating separate DynamoDB stacks in different regions, each with its own writer, table, and reader, can enhance resilience. Although this method incurs higher costs and maintenance efforts, it provides the highest level of system resiliency by isolating potential points of failure.
3) Alternative Technologies: For applications that require robust active - active configurations with strong consistency guarantees, exploring alternative technologies such as Apache Cassandra may be beneficial. Cassandra supports active - active configurations with tunable consistency levels, which can be more suitable for certain use cases.

3. Conclusion

This paper demonstrates that while Amazon DynamoDB offers powerful features for global data replication, its replication strategies present specific challenges in maintaining strong read consistency across regions. Multi - region writes in DynamoDB can lead to inconsistent reads due to replication lag and synchronization issues. Therefore, applications requiring strong read consistency should carefully consider adopting an active - passive write approach or exploring alternative technologies like Cassandra. By understanding these limitations and options, organizations can make informed decisions to enhance the resilience of their cloud systems.

Acknowledgements

This research was informed by various sources, including practical case studies and technical documentation on DynamoDB. Special thanks to the contributions from cloud infrastructure specialists and database engineers who provided insights into DynamoDB's replication mechanisms.

References

[1] Amazon Web Services. ‘DynamoDB Customer Success Stories’. Available at: <https: //aws. amazon. com/dynamodb/customers/> [Accessed 26 August 2024].
[2] Elhemali, R. (2022). ‘Replicated Consistency: A Case Study of Cloud Data Systems’. *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI) *. Available at: <https: //www.usenix. org/system/files/atc22 - elhemali. pdf> [Accessed 26 August 2024].
[3] Amazon Web Services. (n. d.). ‘Working with Items in DynamoDB’. Available at: <https: //docs. aws. amazon. com/amazondynamodb/latest/developerguide/Working WithItems. html#WorkingWithItems. ConditionalUpdate> [Accessed 26 August 2024].