Reading Assignment

Name: Aliyah Inshaal

Roll# 20SW026

Course Name: Data Science and Analytics

# **INTRODUCTION**

This paper introduces Dynamo, a highly available key-value storage system designed and implemented by Amazon.com for reliability at scale. Serving millions of customers worldwide, Amazon's e-commerce platform is built on a distributed, service-oriented architecture with stringent requirements for performance, scalability, and reliability. This paper emphasizes the importance of managing application state to achieve system reliability and scalability.  
  
Dynamo sacrifices consistency in certain failure scenarios to ensure high availability. Object versioning and application-assisted conflict resolution are used to achieve this. The system relies on well-known techniques such as consistent hashing for partitioning and replication, gossip-based distributed error detection and membership protocols. Dynamo is fully decentralized, so storage nodes can be added and removed dynamically without manual intervention.  
  
This paper highlights the success of using Dynamo as the storage technology for key services in Amazon's e-commerce platform. During peak loads, Dynamo was able to efficiently scale to handle tens of millions of requests and millions of checkouts per day, proving its reliability and availability. This paper also shares insights and experiences gained using Dynamo in production. The main contribution of this paper is to evaluate various combinations of techniques for building highly available systems and finally demonstrate the feasibility of consistent storage systems in demanding production environments. It also provides valuable insight into tuning techniques to meet performance requirements.  
The article concludes with a discussion of system background, prerequisites, and requirements. Dynamo provides a simple key/value interface, efficient resource usage, and a scale-out scheme to handle increasing datasets and request rates. While some details have been omitted to protect business interests, this document provides an overview of Dynamo's design, implementation, and suitability for applications that primarily require primary key access to data. is explained. Overall, this paper highlights the importance of reliability and scalability when managing application state and shows how Dynamo effectively addresses these challenges in a highly available and scalable manner. increase.

# **OVERVIEW**

This text discusses the importance of service level agreements (SLAs) in guaranteeing the performance of dependencies within the platform. An SLA is a formal contract that specifies system-related characteristics such as the expected distribution of request rates and service delays. Amazon's distributed service-oriented infrastructure is based on SLAs that guarantee that each service in the call chain delivers its functionality within agreed limits.  
  
It shows the architecture of the Amazon platform, where dynamic web content is generated by a page rendering component that queries multiple services. SLAs play an important role in maintaining clear boundaries for site delivery, as each service must meet performance agreements to ensure a good customer experience. Amazon found the SLA expressed and measured at 99.9. Response time distribution percentiles improve overall performance compared to SLAs based on mean or median values.  
Storage systems have a big impact on his SLA of service, especially in scenarios with simple business logic. State management has become a key component of SLAs, and Dynamo is designed to allow services to control system characteristics such as durability, consistency, functionality, performance, and trade-offs between cost efficiency.  
  
The text also discusses design considerations for data replication and conflict resolution. Dynamo is designed to be an ultimately consistent data store, with conflict resolution occurring at read time, not write time. The choice of who performs conflict resolution is left to the application, giving you the flexibility to decide which resolution method best suits your customer's experience. Dynamo's design principles include incremental scalability, symmetry, decentralization, and heterogeneity. The system should be able to scale one node at a time, give all nodes equal responsibility, avoid centralized control to improve scalability and availability, and leverage the capabilities of individual servers.  
  
In summary, this paper highlights the role of his SLAs in ensuring in-platform performance, especially in the context of storage systems. Dynamo's design considerations and principles are aimed at providing availability and scalable storage so that the service can meet his SLAs and provide a reliable and efficient user experience.

**CONCLUSION**

This post described Dynamo, a highly available and scalable data store used to store the state of many core services of Amazon.com's e-commerce platform. Dynamo provided the desired level of availability and performance and successfully handled server outages, data center failures, and network partitions. Dynamo is incrementally scalable, so service owners can scale up and down based on current Table 2.  
Implementation of client-controlled and server-controlled coordination approaches. 99.9. Percentile Read Latency (ms) 99.9. Percentile Write Latency (ms) Average Read Latency (ms) Average Write Latency (ms) Server Driven 68.9 68.5 3.9 4.02 Client Driven 30.4 30.4 1.55 1.9 208 218 Read Requests. Dynamo allows service owners to tune their storage systems to meet their required performance, durability, and consistency SLAs by adjusting the N, R, and W parameters. Dynamo's production deployment over the past year has shown that distributed technologies can be combined to provide a single highly available system. His success in one of the most demanding application environments shows that eventually consistent storage systems can be building blocks for high availability applications.