DISTRIBUTED DATABASE SYSTEM FOR A GLOBAL E-COMMERCE PLATFORM

DEPARTMENT OF INFORMATION TECHNOLOGY

SABARISH R 7376222IT237

INTRODUCTION:

The system requirements for a Distributed Database System for a global e-commerce platform include both functional and non-functional needs. Functionally, it must handle high-volume transactions, support data partitioning across multiple regions, and enable real-time data synchronization to provide accurate inventory, pricing, and user information globally. Non-functional requirements emphasize scalability, ensuring the platform can expand seamlessly as the user base grows, and performance, where low-latency access and quick response times are essential. Security is paramount, requiring robust encryption, access control, and compliance with regulations like GDPR and CCPA. Additionally, high availability and fault tolerance are essential to minimize downtime, ensuring a consistent user experience worldwide.

SYSTEM REQUIREMENTS:

The system requirements for a Distributed Database System for a global e-commerce platform include both functional and non-functional needs. Functionally, it must handle high-volume transactions, support data partitioning across multiple regions, and enable real-time data synchronization to provide accurate inventory, pricing, and user information globally. Non-functional requirements emphasize scalability, ensuring the platform can expand seamlessly as the user base grows, and performance, where low-latency access and quick response times are essential. Security is paramount, requiring robust encryption, access control, and compliance with regulations like GDPR and CCPA. Additionally, high availability and fault tolerance are essential to minimize downtime, ensuring a consistent user experience worldwide.

ARCHITECTURE DESIGN

The architecture design for a Distributed Database System in a global e-commerce platform is centered around a robust, scalable, and fault-tolerant structure. This includes a system architecture overview that defines the distributed nature of the system, where databases are spread across multiple regions to ensure low latency and meet global demands. The architecture leverages distributed database models like sharding, replication, and partitioning to manage data effectively across locations, balancing load and reducing bottlenecks.

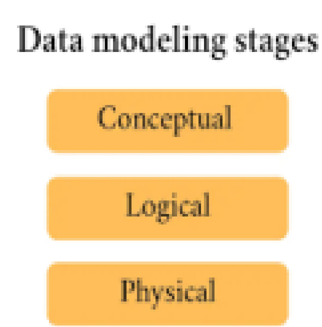
To further improve performance and reliability, data flow and network topology are designed to optimize data access routes, minimize delays, and handle regional traffic spikes. High availability and redundancy are achieved through failover mechanisms and data replication strategies, ensuring uninterrupted service during failures. This architecture aims to meet the platform's requirements for scalability, security, and real-time data processing, while allowing for easy expansion as demand grows.

DATA MODEL AND SCHEMA DESIGN

The Data Model and Schema Design for a Distributed Database System in a global e-commerce platform focuses on structuring data to support scalability, efficiency, and responsiveness. This begins with creating a detailed Entity-Relationship Diagram (ERD) that captures key entities like users, products, orders, inventory, and transactions, defining their attributes and relationships to represent the business logic comprehensively.

The table structures and relationships are designed to support rapid querying and updates, with key data organized in a way that supports easy partitioning for distribution across regions. To accommodate the high volume and complexity of e-commerce data,

the schema includes a blend of normalization and denormalization, balancing data integrity with performance. Primary and foreign



keys are carefully assigned to establish clear relationships between entities while supporting quick retrieval of relevant data. The schema design also considers indexing and caching strategies to optimize query performance, ensuring that users can access accurate, real-time information, regardless of location.

DATA DISTRIBUTION STRATEGIES

The Data Distribution Strategies for a global e-commerce platform's Distributed Database System are critical to ensuring efficient data access, minimized latency, and optimal load balancing across regions. Key strategies include horizontal and vertical partitioning, where data is split by either rows (horizontal) or columns (vertical) to distribute workload evenly among servers and improve access speeds. Additionally, geographic data partitioning is used to localize data closer to regional user bases, reducing latency by keeping frequently accessed data closer to users in different global locations.

Load balancing mechanisms are employed to manage data requests dynamically, rerouting traffic to ensure no single node becomes a bottleneck. To support global operations, multi-region and multi-data center distribution provides data redundancy and fault tolerance, ensuring high availability and resilience. These strategies are designed to balance performance, data integrity, and redundancy, allowing the platform to meet the demands of a growing and widely distributed user base.

CONSISTENCY, AVAILABILITY, AND PARTITION TOLERANCE (CAP):

The Consistency, Availability, and Partition Tolerance (CAP) principle is central to the design of a Distributed Database System for a global e-commerce platform, helping guide choices for handling data access, distribution, and reliability across geographically dispersed servers. The CAP theorem asserts that in a distributed data system, it's impossible to achieve all three properties — consistency, availability, and partition tolerance — simultaneously under network failure; thus, a balance must be struck based on system priorities. For a global e-commerce platform, partition tolerance is essential, as the system must continue to function despite network partitions or node failures to maintain service continuity across regions. Availability is also crucial, ensuring that every request receives a response, though it may not always reflect the latest state of data. Consistency, which requires that all nodes have the same data at any given time, is often relaxed to eventual consistency to maintain performance and availability in such a dynamic environment.

This trade-off typically leads to adopting a flexible consistency model (e.g., eventual or causal consistency), particularly suited for e-commerce, where minor delays in reflecting updates may be tolerable. Careful tuning of these factors allows the platform to provide a seamless user experience, maintaining high availability and reliable data access across global users.

REPLICATION AND DATA SYNCHRONIZATION

Replication and Data Synchronization are crucial to maintaining data consistency, availability, and fault tolerance in a global e-commerce platform's Distributed Database System. **Replication** involves creating copies of data across multiple nodes or regions, ensuring that users can access information from the closest available server, which minimizes latency and supports seamless, high-speed transactions. There are two primary types of replication: **synchronous** and **asynchronous**. Synchronous replication updates all replicas in real time, maintaining data consistency but potentially increasing latency. Asynchronous replication, on the other hand, allows updates to be delayed, which improves performance but may create temporary inconsistencies.

Data synchronization protocols coordinate updates across replicas to maintain data integrity and address conflicts. Conflict resolution strategies, such as **last-write-wins** or

version-based reconciliation, are used to handle discrepancies when replicas are out of sync due to network delays or concurrent changes. To further support real-time updates and high availability, **data latency management** ensures that replicas synchronize data within acceptable timeframes, minimizing delays without overburdening the network.

CONCLUSION:

In conclusion, a well-designed Distributed Database System is fundamental for supporting the complex requirements of a global e-commerce platform. By implementing strategic data distribution, replication, and synchronization across geographically dispersed nodes, the system can deliver high availability, low latency, and a seamless user experience. Balancing CAP theorem constraints ensures the platform meets scalability needs while maintaining data integrity and resilience against failures. Through thoughtful architecture design, data model structuring, and security and compliance measures, the platform is equipped to handle high transaction volumes, dynamic data access, and evolving regulatory requirements. This robust distributed database framework not only supports current demands but is scalable and adaptable to future growth, driving efficient, secure, and user-centric e-commerce operations globally.