DynamoDB-Like System with Replication

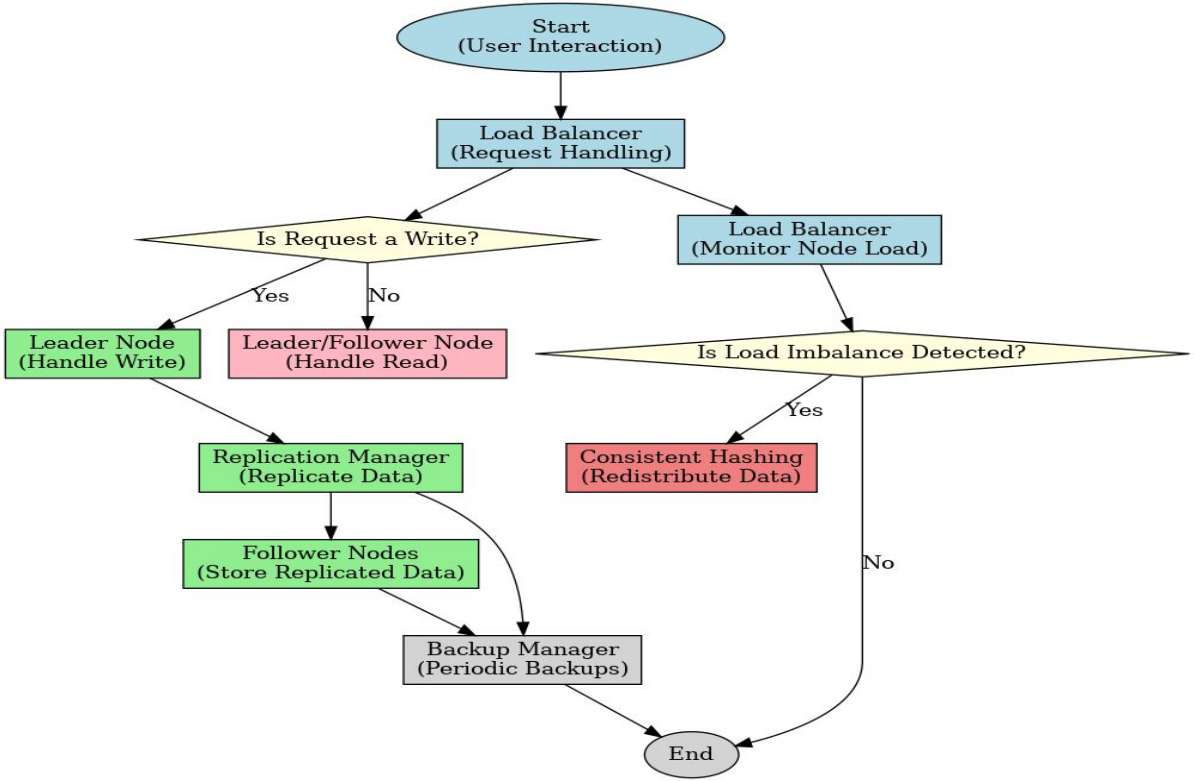
Comprehensive Overview of Project Components and Workflow

BY
Shreyansh Shrivastava(2023201006)
Rajeev Reddy Thumma(2023201014)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Introduction

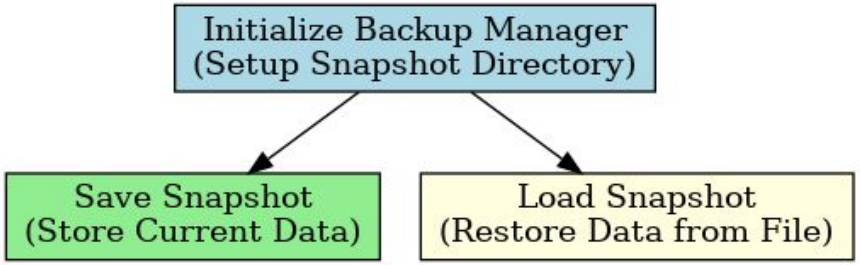
- This project is a simulation of a DynamoDB-like system designed to provide high availability, fault tolerance, and efficient data querying through features such as consistent hashing, replication, load balancing, and secondary indexing.
- It includes backend components that contribute to the overall scalability, reliability, and efficiency of the system. Each module is responsible for a critical aspect of data management, ensuring that the system is robust and capable of handling distributed data operations.



System Workflow Diagram

Backup Manager Overview

The Backup Manager module handles point-in-time recovery of data using snapshots.

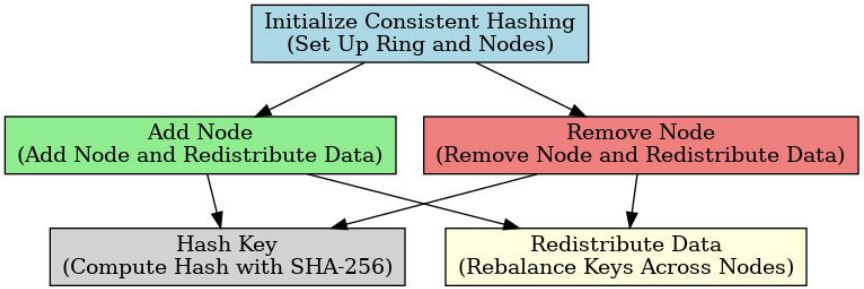


Key Features:

- Manages the creation of snapshots to save the current state of data.
- Enables recovery by loading data from previously saved snapshots.
- Ensures data integrity and resilience by providing a backup and restore mechanism.

Consistent Hashing Overview

The Consistent Hashing module ensures efficient data distribution across nodes.

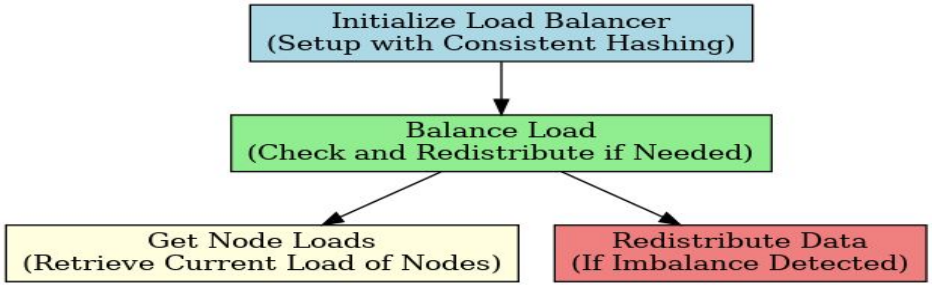


Key Features:

- Ensures even data distribution by adding multiple replicas for each node.
- Handles the addition and removal of nodes dynamically, redistributing data accordingly.
- Helps maintain balanced load across all nodes, contributing to scalability.

Load Balancer Overview

The Load Balancer module ensures even load distribution across all nodes to prevent overload.

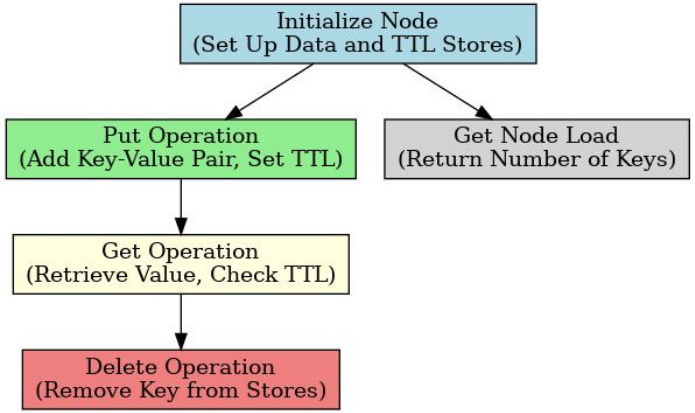


Key Features:

- Continuously monitors the load on each node in the system.
- Redistributes data to maintain balanced loads, avoiding bottlenecks.
- Works closely with consistent hashing to dynamically rebalance data.

Node Overview

The Node represents a single storage unit in the system and manages key-value pairs.

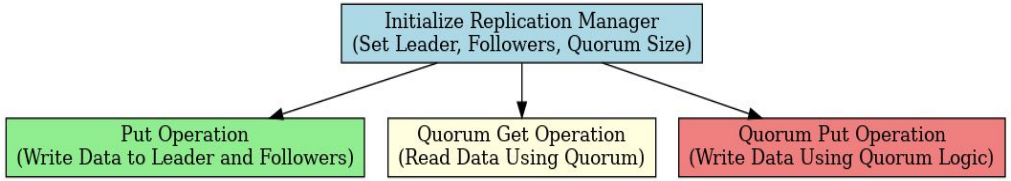


Key Features:

- Stores data and handles TTL for automatic data expiration.
- Supports essential operations such as adding, retrieving, and deleting data.
- Provides load statistics to support load balancing decisions.

Replication Manager Overview

The Replication Manager ensures data availability and consistency across nodes by replicating data.

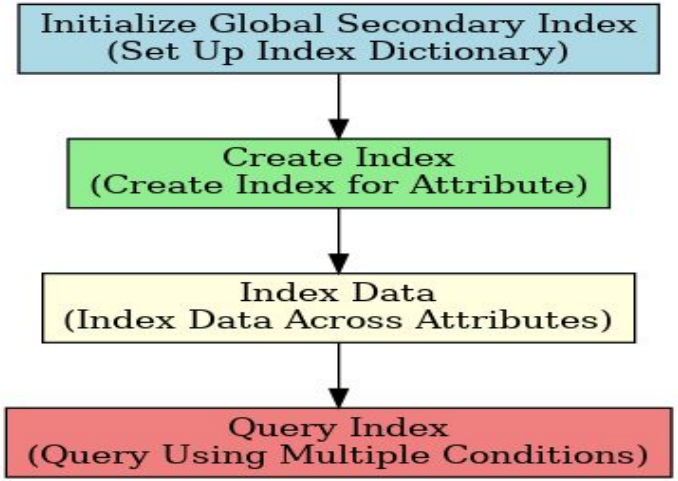


Key Features:

- Handles replication for write operations, ensuring data is available on multiple nodes.
- Uses quorum-based reads and writes to maintain consistency.
- Guarantees high availability of data in case of node failures.

Global Secondary Index Overview

The Global Secondary Index provides advanced querying capabilities based on non-primary attributes.

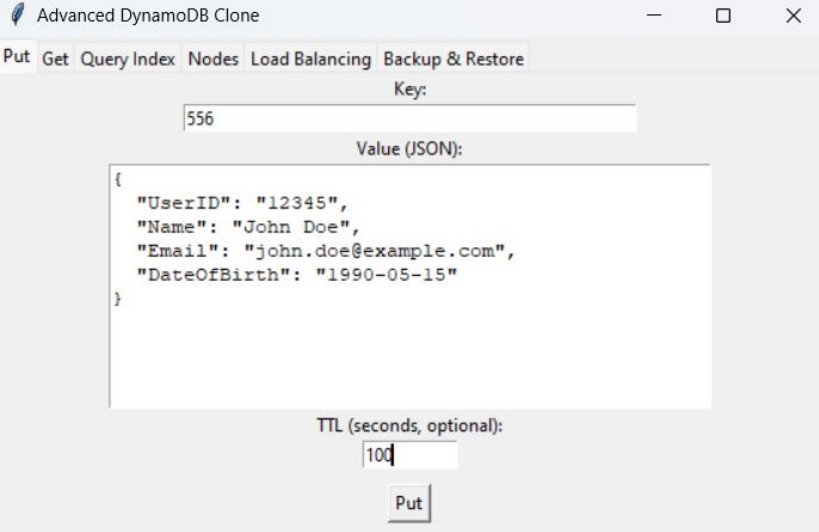


Key Features:

- Allows indexing on multiple non-primary attributes for efficient queries.
- Supports complex queries involving multiple conditions.
- Enhances data accessibility and flexibility in data retrieval.

User Interface

- The app uses Tkinter to create a GUI for interacting with a DynamoDB-like system.
- It supports operations like Put, Get, secondary index queries, and node management etc.



Conclusion

- The DynamoDB-like system we have developed includes features that ensure scalability, reliability, and efficient data management. Modules such as Consistent Hashing, Load Balancer, Replication Manager, and Global Secondary Index work cohesively to create a resilient distributed database system.
- The architecture presented ensures high availability, data redundancy, and efficient querying, making it a robust solution that can serve as an inspiration for distributed systems in real-world applications.

THANK YOU !!