

Replicated key-value data store.

1. Introduction

Leader election is a fundamental concept in distributed systems, ensuring that only one node assumes the role of a leader at any given time. ZooKeeper, a distributed coordination service, provides the necessary primitives for implementing leader election algorithms. This report presents a detailed overview of the implementation of leader election using ZooKeeper, including the algorithm used, Docker setup, test script, execution logs, conclusion, and future work.

2. Leader Election Algorithm

The leader election algorithm implemented utilizes ZooKeeper's sequential znodes to determine the order of node creation. Each server participating in the election process creates a znode under a designated path (/election) with a sequential ID. The server with the lowest sequential ID is elected as the leader. This approach ensures that the leader is selected in a deterministic manner, regardless of the timing of node creation.

The Kazoo library is employed to interact with ZooKeeper and implement the leader election algorithm in Python. Kazoo provides a high-level interface for working with ZooKeeper, simplifying the process of creating znodes and handling leader election logic.

3. Docker Setup

To simulate a distributed environment with multiple ZooKeeper servers, Docker Swarm mode is utilized. Docker Compose is used to define the configuration for running multiple ZooKeeper servers as services. Each ZooKeeper server is configured with a unique ID and a list of all servers in the ensemble.

This Docker setup enables easy deployment and testing of distributed systems in a controlled environment. It allows for scalability testing and facilitates the evaluation of the leader election algorithm's performance under various conditions.

4. Test Script

The test script is designed to validate the behavior of the system under different scenarios, including leader failure and stale reads. It performs tasks such as killing the leader node and performing a stale read test to observe how the system responds to these events.

The test script utilizes the Kazoo library to interact with ZooKeeper and execute various operations such as creating znodes, retrieving data, and monitoring the state of the system.

Execution logs are generated during the test execution to provide insights into the observed behavior of the system.

5. Execution Logs

During the execution of the test script, logs are generated to capture the events and actions performed by the system. These logs include information about leader election, leader failure, stale reads, and other relevant events.

The execution logs provide valuable insights into the behavior of the system under different conditions. They help in identifying any issues or anomalies in the implementation and serve as a basis for further analysis and debugging.

6. Conclusion

The implementation of leader election using ZooKeeper demonstrates the effectiveness of ZooKeeper in managing distributed coordination tasks. The use of sequential znodes ensures a deterministic process for selecting the leader, while the Docker setup facilitates easy deployment and testing of distributed systems.

The test script validates the correctness and robustness of the leader election algorithm, providing confidence in its functionality under various scenarios. The execution logs offer valuable insights into the behavior of the system and its response to different events.

7. Future Work

Future work could involve further experimentation and optimization of the leader election algorithm. This could include stress testing the system under various scenarios to evaluate its performance and scalability. Additionally, additional features such as fault tolerance mechanisms and data replication strategies could be explored to enhance the system's resilience and reliability.

8. References

Consulted documentation and resources related to ZooKeeper, Kazoo library, and Docker Swarm mode during the implementation process. These resources provided valuable insights and guidance for developing and testing the leader election algorithm.