University of Colombo School of Computing

SCS3203 Middleware Architecture

Assignment 1

Team KID^{2}S

20000103 - D.A. Amarasinghe
20001207 - K.K.S Nayanathara
20000723 - A.A.D Harshani
20000766 - P.A.I Heshan
20000804 - J.K.K.K. Jagoda

July 19, 2023

Task 4

Enhance the Architecture to Gain Improvement in Availability and Reliability

Our system's existing Pub/Sub application relies on just one server node, which has a big impact on its availability and dependability. We suggest a strong, distributed design to overcome this issue, ensuring increased fault tolerance and constant communication even in the event of server failure. A central broker serves as an intermediary between publishers and subscribers in the new architecture, which is built on a broker-based Pub/Sub paradigm. Copies of the broker are hosted by numerous server nodes, enabling redundancy and load balancing. The broker effectively transmits communications that publishers give to it to the right subscribers. We can eliminate the single point of failure by decentralizing the system and having numerous broker instances, ensuring continued communication even if one of the servers fails. To achieve high availability and data integrity, we can employ fault detection techniques and data replication across servers.

The proposed distributed design gives our current system a number of significant upgrades. The addition of several broker instances, in the first place, ensures fault tolerance and reduces the possibility of communication breakdown due to server failures. The demand is distributed equitably across the system since each server node hosting a broker may manage client connections and topic subscriptions. The implementation of asynchronous communication methods further improves the performance and responsiveness of our Pub/Sub system. Utilizing a load balancer also makes it possible to utilize resources effectively and avoid overloading particular nodes. We will be able to accomplish data redundancy and better resource usage by sharding topics and replicating broker state, which increases reliability. We can include monitoring and auto-scaling systems, enabling the system to automatically modify its capacity based on the load, to maintain the system's health and scalability. Overall, the distributed Pub/Sub architecture that is being suggested offers a reliable messaging system that can resist server outages and maintain constant user communication.

GitHub Repository: https://github.com/dinukaashan77/middleware_architecture.git

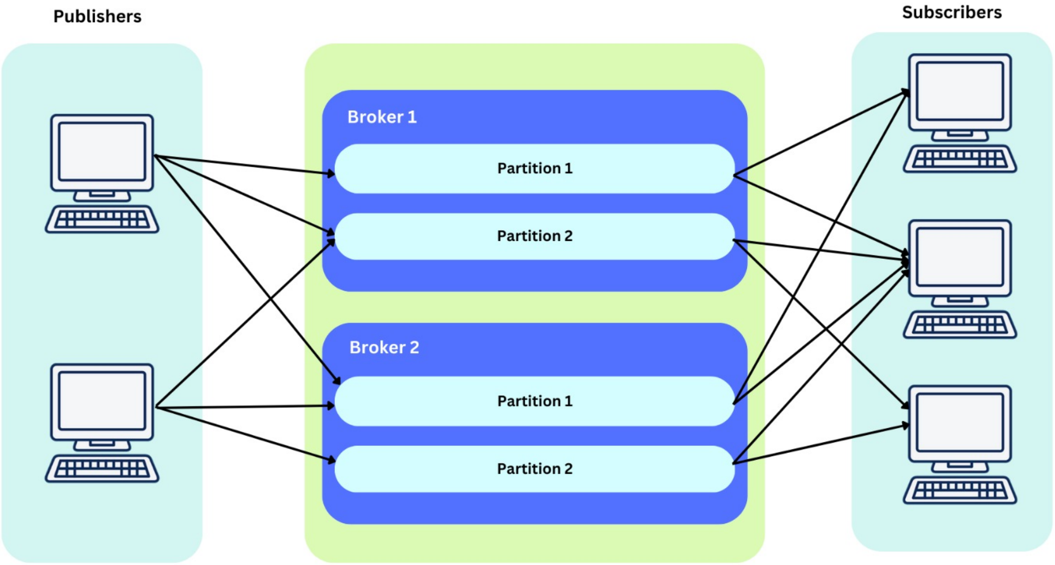

Figure 1: Distributed pub-sub model link