SCS 3203 – Middleware Architecture

Assignment 01

Group Number :27

Group Name: Ninja Turtles

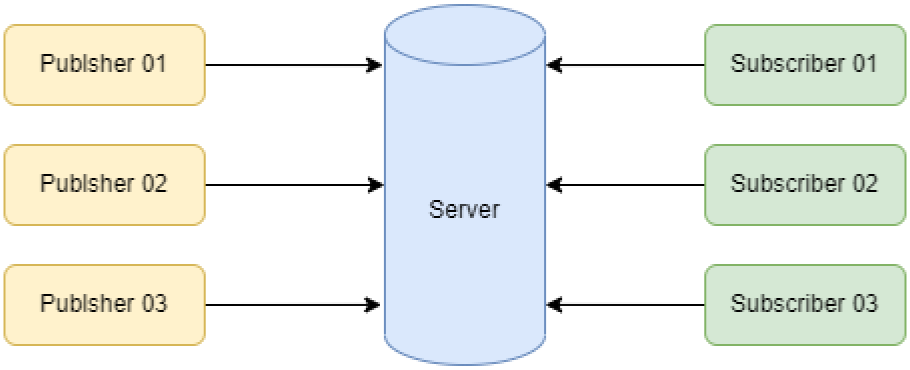
Team Members:

- 20000634 – W.A.K. Gimhani
- 20000219 – J.V.S. Chamari
- 20020066 – S.C.H. Arachchi
- 20001568 - P.B.G. Samarasekara
- 20002203 – M.G.N.D. Wimalarathna

Task 04

Enhance the Architecture to Gain Improvement in Availability and Reliability

Single server-based Pub/Sub Architecture



To enhance the architecture and gain improvements in availability and reliability, we can introduce two key components. Replication and load balancing. These enhancements address the potential drawbacks of a single server-based approach and result in a more robust and fault tolerant Pub/Sub system.

01) Replication

- *Drawback* – In a single server-based architecture, the server acts as a single point of failure. If the server goes down, the entire Pub/Sub system becomes unavailable, and messages might be lost. This scenario can lead to severe disruptions in communication and data dissemination. For instance, if a critical system relies on real-time updates through the Pub/Sub system, a server failure can result in delays in data delivery and processing, impacting the accuracy and responsiveness of the application. Additionally, any messages in transit or awaiting delivery to Subscribers are at risk of being lost, leading to potential data loss and inconsistencies between connected components.
- *Improvement* – By implementing data replication across multiple servers, the system gain fault tolerance, allowing it to remain operational in the presence of failures or issues, while also ensuring data redundancy by storing multiple copies of data in different locations.
- *Explanation* – In the enhanced architecture, multiple broker nodes, which are responsible for message distribution, are deployed, and each broker node maintains a copy of the same set of messages and subscription information. When a Publisher sends a message to a topic, the message is replicated and distributed to all broker nodes responsible for that topic. Similarly, when a subscriber subscribes to a topic, the subscription information is replicated across relevant broker nodes. This way, even if one broker node fails, the data and subscription information

remain accessible from other nodes, ensuring high availability and minimizing the risk of data loss.

- *Example* – Suppose we have three broker nodes, A, B and C. A publisher sends a message to Topic X, and the message is replicated and stored on all three nodes. Similarly, when a Subscriber subscribes to Topic X, the subscription information is replicated across all three nodes. If node A fails, the Publisher and Subscriber can still interact with nodes B and C to publish and receive messages related to Topic X.

02) Load Balancing

- *Drawback* – In a single server-based architecture, all Publishers and Subscribers connect to a single server. As the number of clients increases, the server may become overloaded, leading to performance issues and reduced availability.
- *Improvement* – Implement Load Balancing mechanisms to distribute Publishers and Subscribers evenly across multiple broker nodes.
- *Explanation* – Load balancing algorithms are used to distribute incoming Publisher and Subscriber connections across the available broker nodes based on factors such as node capacity, network conditions, or round-robin distribution. This ensures that the workload is evenly distributed among broker nodes, preventing any single node from becoming a performance bottleneck. Additionally, load balancing improves scalability allowing the system to handle a higher number of connected clients.
- *Example* – Let's say we have three broker nodes, A, B, and C, and ten publishers and ten subscribers connecting to the system. With load balancing, the first Publisher connects to node A, the second Publisher connects to node B, and so on, spreading the load across the nodes. This ensures that no single node becomes overwhelmed with connections, leading to a balanced and efficient system.

So, adding replication and load balancing, the Pub/Sub system achieves high availability, fault tolerance, scalability, redundancy, and improved performance.