Middleware Architecture (IS3108 / SCS3203) - Assignment

Task 4 Documentation

Proposed new Distributed Architecture for improved Availability and Reliability

Limitations of Existing Implementation

The current architecture is built with a single Server and therefore has several limitations,

- Server is single point of failure and can cause the failure of entire system
- Issues regarding scalability due to a single server
- Lack of load distribution

Proposed Distributed Architecture

Our proposed distributed architecture aims for improvement and to minimize the limitations of the existing architecture and provide a new architecture that is available, reliable and scalable.

The proposed architecture will contain

- Multiple Servers handling connections from clients.
- Service Discovery which will enable the automatic detection and registration of servers instances
- Load balancers that will distribute Client connections across multiple Server instances
- A centralized message broker for the reliable delivery of messages(Examples include RabbitMQ or Kafka)
- Replication mechanisms to reduce failures.

Architecture Diagram of Proposed Distributed Architecture

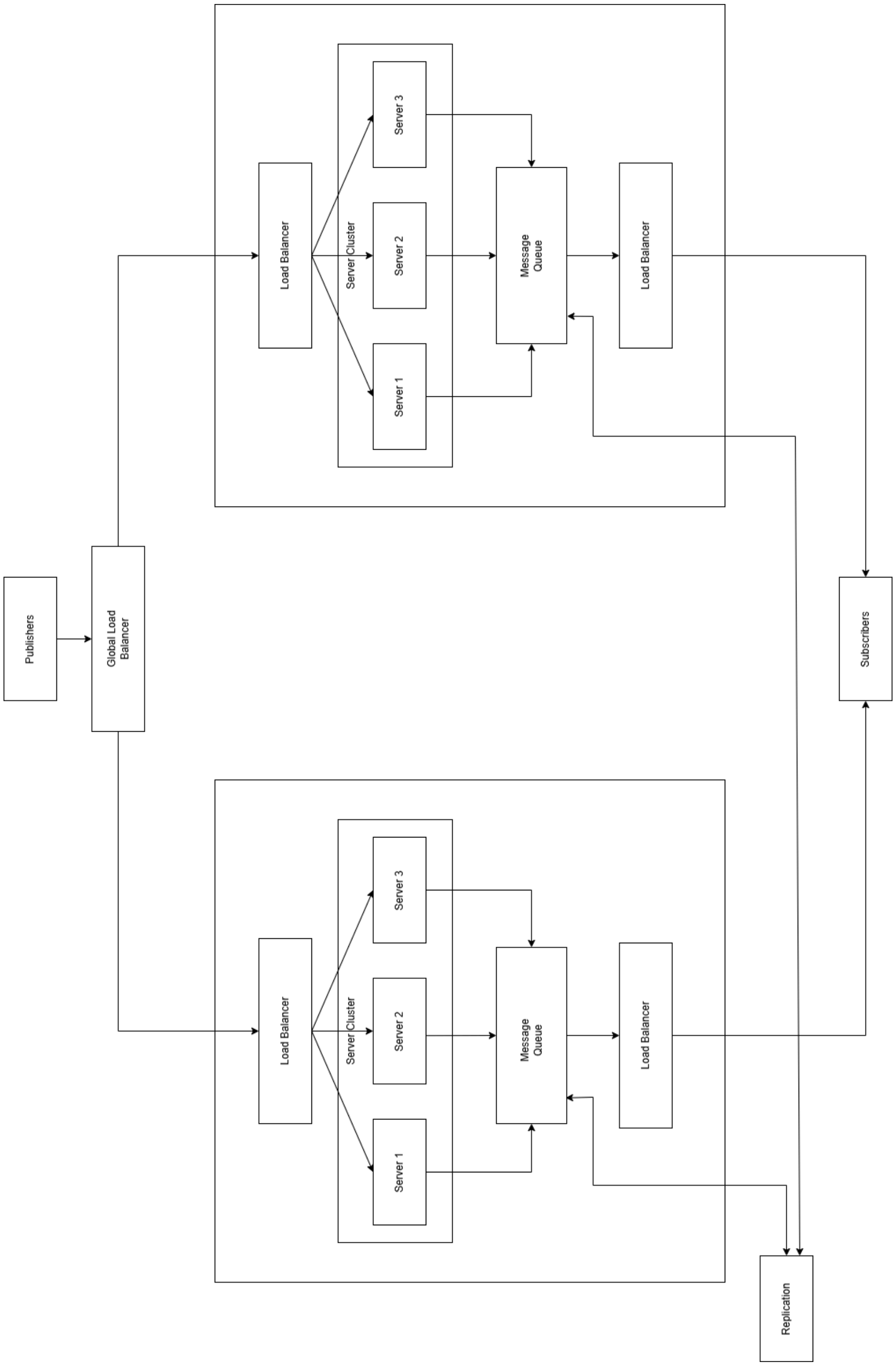

Diagram Link

https://drive.google.com/file/d/12dz5XoL0LhoEnhi-jPmXWoCcXDBqKCNd/view?usp=sharing

Improvements Provided by Proposed Architecture

- Clustering of servers by the deployment of several server instances which each instance handling a set of clients witch communication handled by a distributed Queue
- Implementation of a Message Queue Distribution ensuring consistency and reliability
- Load balancing with the distribution of connections across servers
- Failover prevention through replication mechanisms
- Overall improved availability, scalability reliability and fault tolerance through the distributed system

Challenges of the Proposed Architecture

- Higher complexity and cost of distributed system implementation

Final Conclusion

By adopting the proposed distributed architecture, we are able to eliminate the server as a single point of failure. This will ensure the high availability, reliable delivery, and scalability of the middleware Pub/Sub architecture.