

Internet and Cloud Computing

Task: Cloud based Collaborative Whiteboard

Deadline: November 19th , 2020

Cloud based Collaborative Whiteboard

There are over 250 members in a class. You need to design and develop a Cloud based Collaborative Whiteboard that can be shared between multiple users of this class over the network. You need to run this whiteboard in a distributed environment where people can join in either from their own machines or from distributed class rooms. Your whiteboard should support for basic drawing features such as line, circle, and rectangle as well as text anywhere in the whiteboard.



The Cloud based Collaborative Whiteboard should have the following features:

1. **Distributed System Design (6 Marks)**

You should propose a suitable architecture for your distributed whiteboard and then propose an implementation plan to build the proposed system architecture.

You should consider mechanisms to achieve consistency, scalability, fault tolerance, concurrency and performance in your architecture. You should also propose a suitable network model to manage traffic and provide fault tolerance.

You should produce (and draw) an architecture of your distributed whiteboard system so that it can be deployed in a cloud environment at later stages.

You should evaluate available cloud and distributed server architectures before you design and implement your own distributed whiteboard system. You should have convincing reasons to justify the technology choices that you have made for your architecture.

In this task you should have a clear understanding and justification of the tools and technologies that you will use to implement your Cloud based Collaborative whiteboard.

2. Distributed State Management (6 Marks)

The Cloud based Collaborative Whiteboard should allow multiple users to draw simultaneously on a shared interactive canvas.

When a new person joins the system the user should obtain the current state of the whiteboard so that the same objects are always displayed to every active user.

The system should support a single whiteboard that is shared between all of the clients.

All the users should see the same image of the whiteboard and should have the privilege of doing all the drawing operations.

When displaying the whiteboard, the user interface should show the usernames of other users who are currently editing the same whiteboard. Users must provide a username when joining the whiteboard.

3. Consensus and Leadership (4 Marks)

Users should vote to elect a leader who then has the admin rights. A new leader should be elected using a consensus protocols (Paxos, Raft etc). This node/user essentially takes charge of the system,

we call this node the “leader”. When this node fails, then the system detects this and users should elect another node to take the leadership role.

The leader should allow only a selected group of users/peers to write/draw on the whiteboard at the same time. Other users who want to participate should wait in a queue to be given permission.

All the peers will see the identical image of the whiteboard, as well as have the privilege of doing all the operations.

Only the leader of the whiteboard should be allowed to create a new whiteboard, open a previously saved one and close the application.

4. Replication (4 Marks)

The copy of the whiteboard should be saved when all users have agreed to the whiteboard state that meets their requirements. This should be achieved through implementing a consensus protocol.

The users/nodes/servers should respond with a “yes” or “no” vote regarding the transaction. If more than 50 percent servers respond with a “yes” vote, the leader issues a commit operation to the servers to make the transaction durable; i.e., the whiteboard state has been committed to all participating servers (strongly consistent vs loosely consistent).

5. Performance and concurrency (4 Marks)

Users should be able to work on a drawing together in real time, without appreciable delays between making and observing edits.

You can have multiple servers that communicate with each other or a single central one that manages all the system state.

You have to consider when/what messages are sent across the network because sending/receiving too many messages across users will have performance implications. This will require designing an exchange protocol that establishes which messages are sent in which situation and the replies that they should generate.

6. Docker Service (6 Marks)

You should create a Docker container for your cloud based whiteboard system and deploy it in the Docker registry or another registry of your choice. You should share the client details on how to access your Docker container for the system so that we can test your implementation. All details accessing

and running your service should be written in step by step so that it only takes **two minutes (max)** to test the complete functionality.

Submitting your Task

Please submit description of your architecture, salient features of your design, technology choices made, service implementation details, screen shots of your implementation/execution and code as a [single pdf document](#). This document should also contain URLs as well as other details in accessing and running your service. You should also provide a login and password for accessing the system.

WARNING

Please do not use the code from publicly available sources. You may get a zero mark if it is found that you have copied code from publicly available sources. The same applies to the documentation and design. This should be your own work and you should not share it with anyone else.