

Problem Statement 2:

Building a Task Management Application with Raft Consensus Algorithm and MySQL

Overview:

The objective of this project is to develop a task management application that utilises the Raft consensus algorithm to ensure consistency and fault tolerance across multiple nodes. MySQL will be employed as the backend database to store task data. The application will enable users to create, update, delete, and manage tasks across the distributed system.

Objectives:

- Implement a task management system with Raft consensus algorithm for distributed coordination.
- Utilise MySQL as the backend database to store task data.
- Ensure fault tolerance and consistency across multiple nodes using Raft.
- Enable users to perform CRUD operations on tasks via a user-friendly interface.
- Demonstrate distributed system principles and fault tolerance mechanisms through the application.

Pre-requisites:

- Raft Consensus Algorithm: Basic knowledge of Raft consensus algorithm and its implementation.
- MySQL Database: Good understanding of relational databases and experience with MySQL.
- Proficient in backend development using Python, with a basic understanding of frontend technologies for building user interfaces.

Weekly Deliverables:

Week1:

1. Designing the Application Architecture
 - Define the architecture of the task management application.
 - Determine the role of each node in the Raft consensus algorithm, such as leader, follower, and candidate.
 - Plan the schema for storing task data in MySQL, considering factors like table structure, relationships, and indexing.
2. Implementing Raft Consensus Algorithm
 - Implement Raft consensus algorithm to ensure fault tolerance and consistency across multiple nodes.
 - Develop communication protocols between nodes for achieving consensus.
 - Test and validate the Raft implementation under different scenarios.

Week 2:

3. Integrating with MySQL Database
 - Set up a MySQL database to store task data.
 - Implement CRUD operations for task management using MySQL queries.
 - Ensure data consistency between Raft nodes and MySQL database.
4. Backend Development
 - Develop backend services for handling user requests and interactions.
 - Integrate Raft consensus algorithm and MySQL database with backend services.

Week 3:

5. Frontend Development
 - Design and develop a simple user-friendly interface for task management.
 - Implement features for creating, updating, deleting, and viewing tasks.
6. Deployment and Testing
 - Deploy the application across multiple nodes to test Raft consensus algorithm.
 - Conduct comprehensive testing to ensure fault tolerance and data consistency.
 - Monitor application performance and troubleshoot any issues.

General Guidelines for the project:

1. Create a GitHub Repository.
2. Weekly progress according to the problem statement assigned must be pushed to the repo and this will be considered while evaluating.
3. Each team's weekly progress update in the repository must be shown to the teachers in Class.
4. Naming convention: **<SRN1>_<SRN2>_<SRN3>_<SRN4>_Project title** (srn in ascending order).