

UE21CS351B – Cloud Computing

Miniproject Report

Problem Statement 2: Building a Task Management Application with Raft Consensus Algorithm and MySQL

Section: 6 'H'

Team Members:

- 1) PES1UG21CS478 – Pranavi Rasamsetty
- 2) PES1UG21CS487 – Ria R Kulkarni
- 3) PES1UG21CS492 – Riya Jayakumar
- 4) PES1UG21CS929 – Riya Bansal

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 is employed as the backend database to store task data. The application enables users to create, update, delete, and manage tasks across the distributed system.

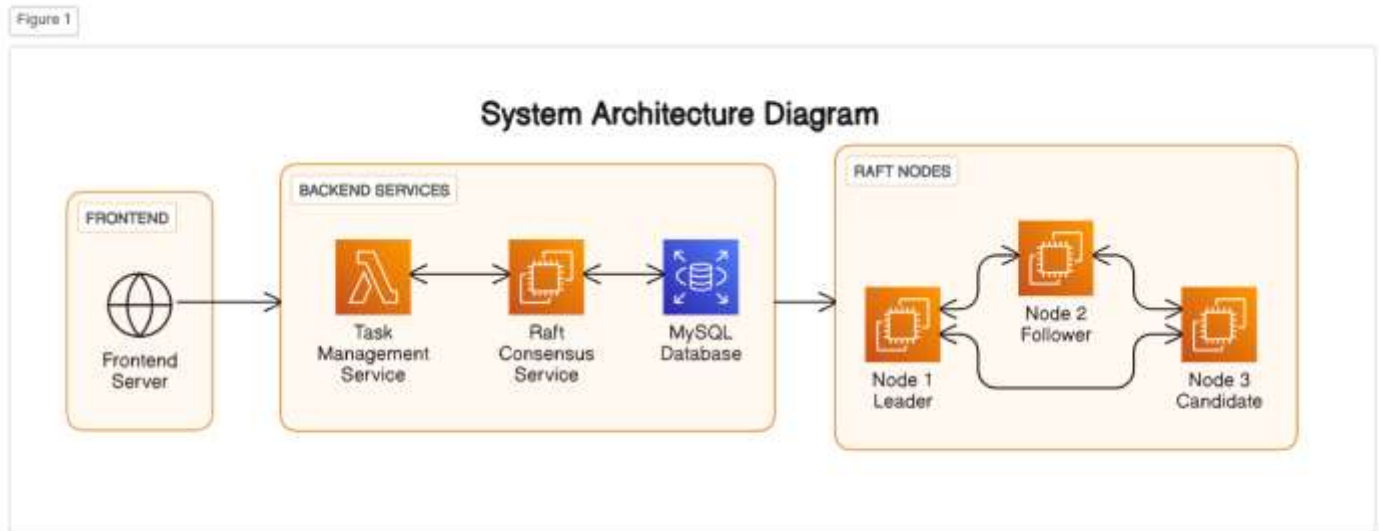
Objectives:

- 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.

WEEKLY DELIVERABLES

WEEK-1 PROGRESS:

1) System architecture design:



2) MySQL schema design:

```
mysql> use task_manager;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_task_manager |
+-----+
| tasks                    |
+-----+
1 row in set (0.01 sec)

mysql> desc tasks;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| description | varchar(255)  | NO   |     | NULL    |                |
| status     | varchar(50)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

WEEK-2 PROGRESS:

1) Implementing Raft Consensus Algorithm:

- Leader: Manages all write operations (CREATE, UPDATE, DELETE) on the task data, ensuring data consistency by replicating logs to followers. The leader also handles heartbeat messages to maintain authority within the cluster.
- Follower: Stores replicas of the leader's log and participates in elections. Followers may serve read-only queries if they have up-to-date information, reducing the load on the leader.
- Candidate: A node becomes a candidate in the case of a leader failure, initiating an election process among the nodes to elect a new leader.

```
Last login: Mon Apr 22 11:39:43 on ttys000
riya@Riyas-MacBook-Air 929_487_492_478_CC_P2 % go run raft_sql.go
Connected to MySQL database!
Initiating leader election from node 1
Node 1 sending vote request to node 2
Node 1 received 1 votes, status: alive
Re-electing leader...Initiating leader election from node 2
Node 2 received 1 votes, status: alive
Server is running on port 8080
riya@Riyas-MacBook-Air 929_487_492_478_CC_P2 %
```

2) Communication protocols implemented:

- Heartbeat Mechanism: Implemented via periodic AJAX calls from the leader to followers, ensuring continuous monitoring and leader authority.
- Log Replication Protocol: Utilizes HTTP POST requests for the leader to replicate logs to followers, with JSON payloads containing log entries.
- Election Process: Involves a combination of HTTP GET requests for voting and POST requests for announcing election results with timeouts.

3) Developed backend:

```
Last login: Mon Apr 22 11:35:07 on ttys001
riya@Riyas-MacBook-Air ~ % mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use task_manager;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from tasks;
+----+-----+-----+
| id | description | status |
+----+-----+-----+
| 1  | Task 1     | Pending |
| 4  | Task 2     | Pending |
+----+-----+-----+
2 rows in set (0.00 sec)

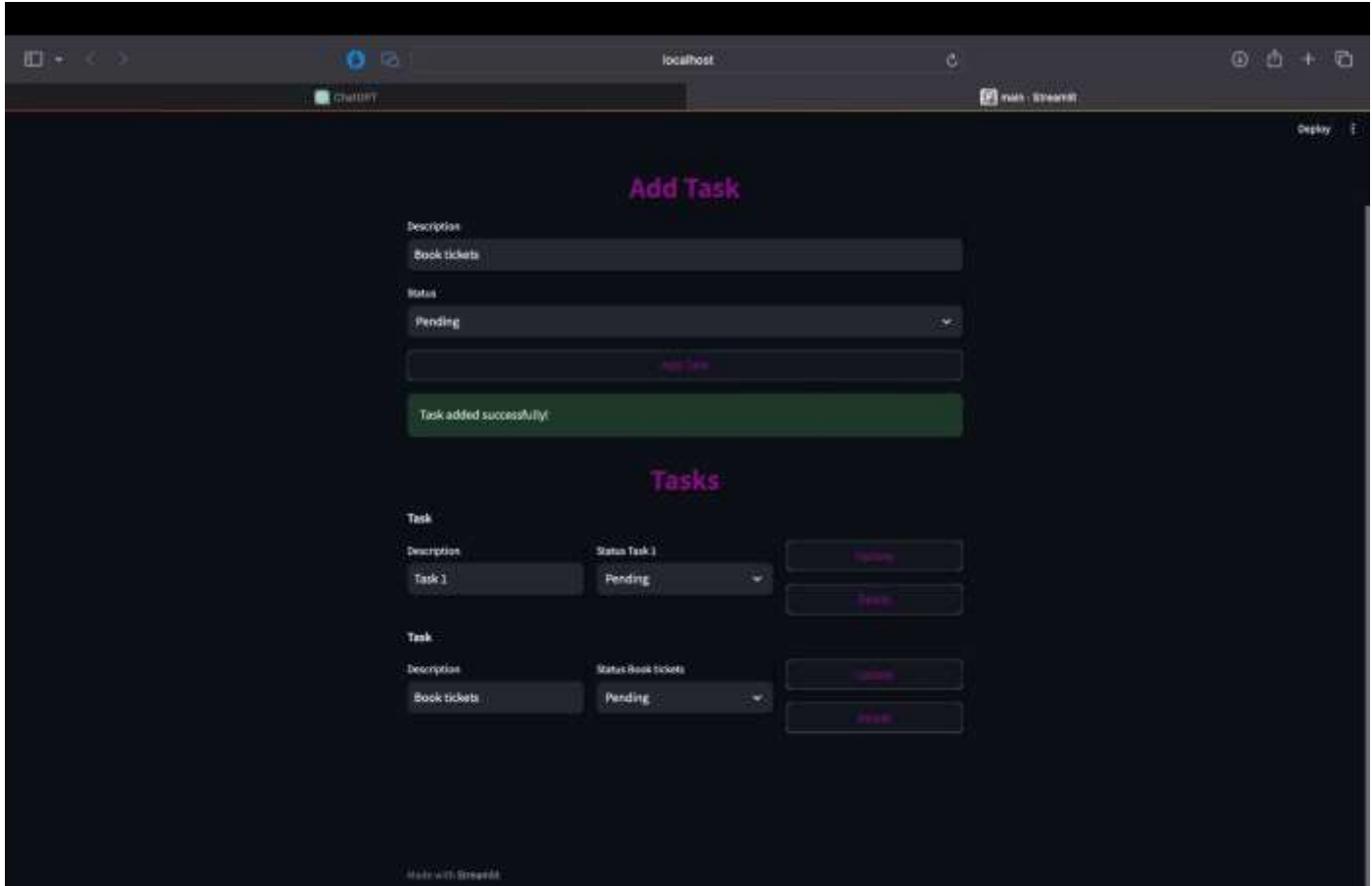
mysql> select * from tasks;
+----+-----+-----+
| id | description | status |
+----+-----+-----+
| 1  | Task 1     | Pending |
+----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

WEEK-3 PROGRESS:

1) Frontend Design:

Designed frontend using Streamlit and CSS.



localhost

Deploy

Add Task

Description

Book tickets

Status

Pending

Add Task

Task added successfully!

Tasks

Task	
Description	Status Task 1
Task 1	Pending
Update	
Delete	

Task	
Description	Status Book tickets
Book tickets	Pending
Update	
Delete	

Made with Streamlit