**Reddit Clone: Engine and Simulator**

**Introduction**

This project implements a Reddit Clone engine and a client tester/simulator. The engine provides the foundation for functionalities such as user registration, subreddit management, posting, commenting, and voting, similar to the Reddit platform. The simulator tests the engine's performance and scalability under realistic conditions. It also includes a REST API built using the Gin framework to interact with the engine.

The implementation separates the engine and client processes, ensuring modularity and scalability. The simulator mimics thousands of users interacting with the engine to generate meaningful metrics.

**Video Submission**

**https://www.youtube.com/watch?v=touC6qHoHCU&list=PPSV&t=3s&ab\_channel=KingdomMutalaAkugri**

**Engine and Simulator Implementation**

1. **Engine**:
   * **User Management**: Supports user registration, connection/disconnection handling, and active user tracking.
   * **Subreddit Features**: Allows creation, joining, and leaving of subreddits.
   * **Post and Comment Management**: Enables text-based posts and hierarchical comments.
   * **Voting Mechanism**: Tracks upvotes, downvotes, and karma computation.
   * **Direct Messaging**: Facilitates private message exchange between users.
2. **Simulator**:
   * **User Behavior Emulation**: Simulates thousands of users performing actions like posting and commenting.
   * **Dynamic Connections**: Models connection/disconnection cycles.
   * **Activity Distribution**: Implements a Zipf distribution to simulate varying subreddit popularity.
3. API Endpoints
   * POST /register: Register a new user.
   * POST /subreddit: Create a new subreddit.
   * POST /post: Create a new post.
   * POST /comment: Create a new comment.
   * POST /vote: Vote on a post or comment.
   * POST /message: Send a direct message.

**Code Structure**

1. **Core Modules**:
   * engine/: Contains the core logic for user, subreddit, post, and comment management.
   * simulator/: Manages user simulation and activity generation.
   * proto/: Defines Protobuf messages for communication between components.
2. **Main Components**:
   * **Engine**: Processes actions like creating posts, voting, and generating metrics.
   * **Simulator**: Generates user activity and measures system performance.

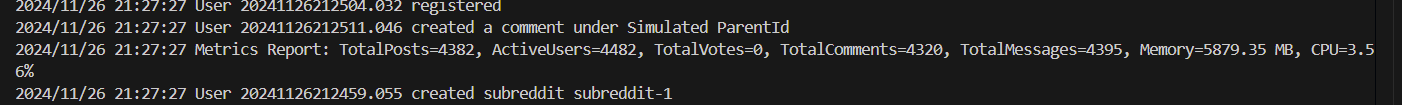
**Output**

The system outputs periodic performance metrics, including:

* Total posts, comments, votes, and messages.
* Number of active users.

**Experimental Findings for 2-minute runs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Users** | **Active Accounts** | **CPU Usage (%)** | **Memory (MB)** | **Posts** | **Comments** | **Messages** |
| 10 | 38 | 0.86 | 46.91 | 26 | 24 | 25 |
| 100 | 237 | 1.21 | 297.04 | 229 | 226 | 217 |
| 500 | 946 | 1.77 | 1415.22 | 1019 | 994 | 981 |
| 1000 | 2072 | 3.47 | 2800.17 | 2101 | 2008 | 1952 |
| 1500 | 2869 | 4.03 | 4191.21 | 2867 | 2873 | 2973 |
| 2000 | 4215 | 5.06 | 5601.22 | 4312 | 4204 | 4210 |
| 2100 | 4482 | 5.36 | 5879.35 | 4382 | 4320 | 4395 |
|  |  |  |  |  |  |  |



**Graphs**

Users vs. Memory Usage

A graph with a line

Description automatically generated

Users vs CPU

A graph with a line

Description automatically generated

**Conclusion**

This project demonstrates a Reddit clone with a backend engine, a simulator for user interactions, and a REST API for external interactions. The engine handles core functionalities, while the simulator generates user activity, and the API provides endpoints for various operations.