Reddit-like Engine and Simulator in Go

Nirvisha Soni **Neel Malwatkar**

This project implements a simplified Reddit-like engine and simulator using Go and the *Proto Actor* framework. The system simulates user activities, subreddit interactions, and content creation in a concurrent and distributed manner. The simulator uses a Zipf distribution to model the popularity of subreddits, which helps create a more realistic simulation of user behavior and content distribution.

Overview

The project consists of five main components:

1. **`main.go:`** The entry point of the system.
2. **`messages.go:`** Defines the message structures used for communication between actors.
3. **`models.go:`** Contains the data models for users, subreddits, posts, and comments.
4. **`engine.go:`** Implements the core logic of the Reddit-like system.
5. **`simulator.go:`** Simulates user actions and interactions with the engine.

Implemented Features

1. Register User.
2. Create, Join & Leave a SubReddit.
3. Post to any SubReddit, being a subscriber is not necessary, just like real world.
4. Comment on Posts and reply/comment on comments.
5. Posts can be Upvoted and Downvoted.
6. Whenever a post is created, it gets 1 upvote by default (Author's).
7. Karma Rules.-
    1. +1 Karma per post.
    2. +1 Karma after commenting on a post.
    3. -1 Karma for each downvote on the post made.
    4. Karma from posted content = Total Upvotes - Total Downvotes
8. Send direct message to an user.
9. Reply to direct message sent by another user.
10. Connection/Disconnection of users, basically an user will perform an action(Post, create sub, comment, leave or join sub) only when its connected to the simulator even though simulator might choose a user to do some task.
11. Get feed for joined SubReddits.

Algorithm

The simulation follows these main steps:

1. Initialize the actor system with an Engine actor and a Simulator actor.
2. The Simulator registers initial users and creates initial subreddits.
3. The Simulator then randomly generates actions (e.g., creating posts, commenting, voting) and sends them to the Engine.
4. The Engine processes these actions, updating the system state accordingly.
5. The simulation runs for a specified duration or number of actions.
6. Finally, the system prints out statistics and user actions.

Usage

To run the simulation, use the following command:

```
go run . -users 10 -subreddits 3 -actions 100 -time 3
```

Available flags: - `-users`: Maximum number of users (default: 30) - `-subreddits`: Maximum number of subreddits (default: 6) - `-actions`: Number of simulation actions (default: 200) - `-time`: Simulation time in seconds (default: 5)

The simulation will run for the specified time or number of actions, generating a variety of user activities and interactions within the simulated Reddit-like system.

Note: To see all the implemented features, run the simulation for default (or lower) values using `go run .`

Largest Network

```
Maximum Users     : 100K
Total Actions     : 250K
Total SubReddits : 600
Time  (mm:ss)     : 2:40
Machine           : 12-Core M3 Pro
Memory            : 18 GB
```

`Note:` More users can be simulated, however the simulation time increases with Actions, SubReddits and Users.

Engine Methods

The Engine actor handles the core functionality of the Reddit-like system. Here's an overview of its main methods:

`Receive(context actor.Context):`
Handles incoming messages and routes them to appropriate methods.

registerUser(username string):
Creates a new user with the given username.

createSubreddit(name, creator string):
Creates a new subreddit with the given name and creator.

joinSubreddit(subredditName, username string):
Adds a user to a subreddit's member list.

leaveSubreddit(subredditName, username string):
Removes a user from a subreddit's member list.

createPost(postID, subredditName, author, title, content string):
Creates a new post in a specified subreddit.

createComment(postID, parentID, commentID, author, content string):
Adds a comment to a post or as a reply to another comment.

vote(postID, userID string, isUpvote bool):
Records a user's vote (upvote or downvote) on a post.

sendDirectMessage(from, to, content string):
Sends a direct message from one user to another.

getFeed(username string):
Generates a feed of posts from subreddits the user is subscribed to.

getSimulationStats():
Prints out statistics about users, subreddits, and posts.

Simulator Methods

The Simulator actor generates random actions to simulate user behavior. Here are its main methods:

Receive(context actor.Context):
Handles incoming messages and initiates the simulation.

runSimulation(context actor.Context):
Runs the main simulation loop, generating random actions.

registerInitialUsers(context actor.Context):
Creates a set of initial users at the start of the simulation.

createInitialSubreddits(context actor.Context):
Creates a set of initial subreddits at the start of the simulation.

`simulateAction(context actor.Context):`
Randomly selects and executes a simulated user action.

`simulateConnection():`
Simulates users connecting to or disconnecting from the system.

`simulateRegisterUser(context actor.Context):`
Simulates a new user registration.

`simulateCreateSubreddit(context actor.Context):`
Simulates the creation of a new subreddit.

`simulateJoinSubreddit(context actor.Context):`
Simulates users joining subreddits.

`simulateLeaveSubreddit(context actor.Context):`
Simulates a user leaving a subreddit.

`simulateCreatePost(context actor.Context):`
Simulates a user creating a new post.

`simulateCreateComment(context actor.Context):`
Simulates a user commenting on a post or replying to a comment.

`simulateVote(context actor.Context):`
Simulates a user voting on a post.

`simulateSendDirectMessage(context actor.Context):`
Simulates a user sending a direct message to another user.

`simulateGetFeed(context actor.Context):`
Simulates a user requesting their personalized feed.

Authors

- Nirvisha Soni : 47638268
- Neel Malwatkar : 68517665