CS455: Homework 4

Performance Testing and Reliability Enhancements

Due Date: 16 Nov, 2024 11:59 pm

Submission Format:

1. A report (single PDF per team) containing team members roll numbers and name, GitHub Repository link (including README, code, performance test scripts, and documentation).

What to do:

Part 1: Performance Testing

1. Client side
   - Load time: Measure and document the time it takes for the game client to load in the browser. Identify any assets (e.g., images, scripts) that significantly impact load time and suggest optimizations. This requires you to use something like selenium (actually load the game and then see the time; remember perf tests should be as close to actual as possible). Do this for the leaderboard page load as well.
2. Server-Side Performance Testing
   - Load Testing + Response time: Simulate concurrent users accessing the server to understand how well it handles multiple requests. Use tools like Apache JMeter, k6, or Artillery to simulate up to 10000 concurrent users. Ensure you do this for game page and the leaderboard page. Measure the server's response time to requests from the client under varying load conditions. Document results.
3. Ensure these are added to your GitHub and test suite, and is run nightly every night, or on triggering the action manually.

Part 2: Reliability and Failover Mechanisms

1. Load Balancing
   - Implement load balancing between multiple server instances to evenly distribute incoming requests.
   - Use a round-robin approach or set up a simple load balancer to manage traffic across at least two instances of your game server.
2. Automatic Failover
   - Set up a primary and backup server. Implement a mechanism that checks the health of the primary server.
   - If the primary server goes down, the backup server should automatically take over and handle incoming requests.
   - Consider using tools like PM2 (for Node.js) or a simple script to manage server monitoring and automatic switching.

Grading Criteria: Individual contributions, quality and accuracy of performance tests, implementation of reliability features, and clear documentation.