## **Stress Test Report: /client\_login Endpoint**

**Date:** Dec 30, 2024

**Tester:** Aathiraja

**Application Under Test:** Task app

**Endpoint:** /client\_login

**Test Objective:** To evaluate the performance and identify any bottlenecks in the /client\_login endpoint under increasing load.

**Testing Methodology:**

* **Load Testing Tool:** k6
* **Testing Scenario:** Stress test using ramping-vus executor.
* **Test Parameters:**
  + **Start VUs:** 0
  + **Stages:**
    - Stage 1: Ramp up to 1000 VUs over 5 minutes.
    - Stage 2: Maintain 1000 VUs for 15 minutes.
    - Stage 3: Ramp down to 0 VUs over 5 minutes.
* **Metrics:**
  + Response time (p50, p90, p95, p99)
  + Throughput (requests per second)
  + Error rate
  + CPU and memory utilization (if monitored)

**Test Results:**

* The /client\_login endpoint exhibited a significant performance degradation starting at approximately **550 VUs with a constant arrival rate of 0.1 VUs/sec**.
* Observed issues:
  + **Increased response times:** Average response times increased significantly beyond acceptable thresholds.
  + **Rising error rates:** Error rates for the /client\_login endpoint increased substantially, indicating system instability.
  + **Potential bottlenecks:**
    - **Database:** Increased database load and slow query response times.
    - **Application Server:** High CPU or memory utilization on the application server.
    - **Network Congestion:** Increased network latency and packet loss.

**Analysis:**

The observed performance degradation suggests that the /client\_login endpoint has limitations in handling high concurrent user loads. Potential causes include:

* **Database bottlenecks:** Inefficient database queries, insufficient database resources (CPU, memory, disk I/O), or slow database connections.
* **Application server limitations:** Insufficient CPU, memory, or thread resources to handle the increased load.
* **Code inefficiencies:** Inefficient algorithms or blocking operations within the login logic.
* **Network limitations:** Network congestion or latency issues impacting communication between the client and the server.

**Recommendations:**

* **Conduct further investigation:**
  + Analyze server logs and application logs to identify specific error messages and pinpoint the root cause of the issues.
  + Monitor server resource utilization (CPU, memory, disk I/O) during the stress test.
  + Profile the application code to identify performance bottlenecks.
* **Optimize database queries:**
  + Review and optimize database queries for better performance.
  + Consider using database indexes and caching mechanisms.
* **Improve application code:**
  + Identify and address any performance bottlenecks in the application code.
  + Implement asynchronous operations and thread pooling to improve concurrency.
* **Scale infrastructure:**
  + Increase server resources (CPU, memory, RAM) if necessary.
  + Implement load balancing to distribute traffic across multiple servers.
* **Network optimization:**
  + Investigate and resolve any network bottlenecks or latency issues.

**Conclusion:**

The stress test identified performance limitations of the /client\_login endpoint under high load conditions. Further investigation and optimization are required to ensure the system's stability and performance under expected and peak loads.