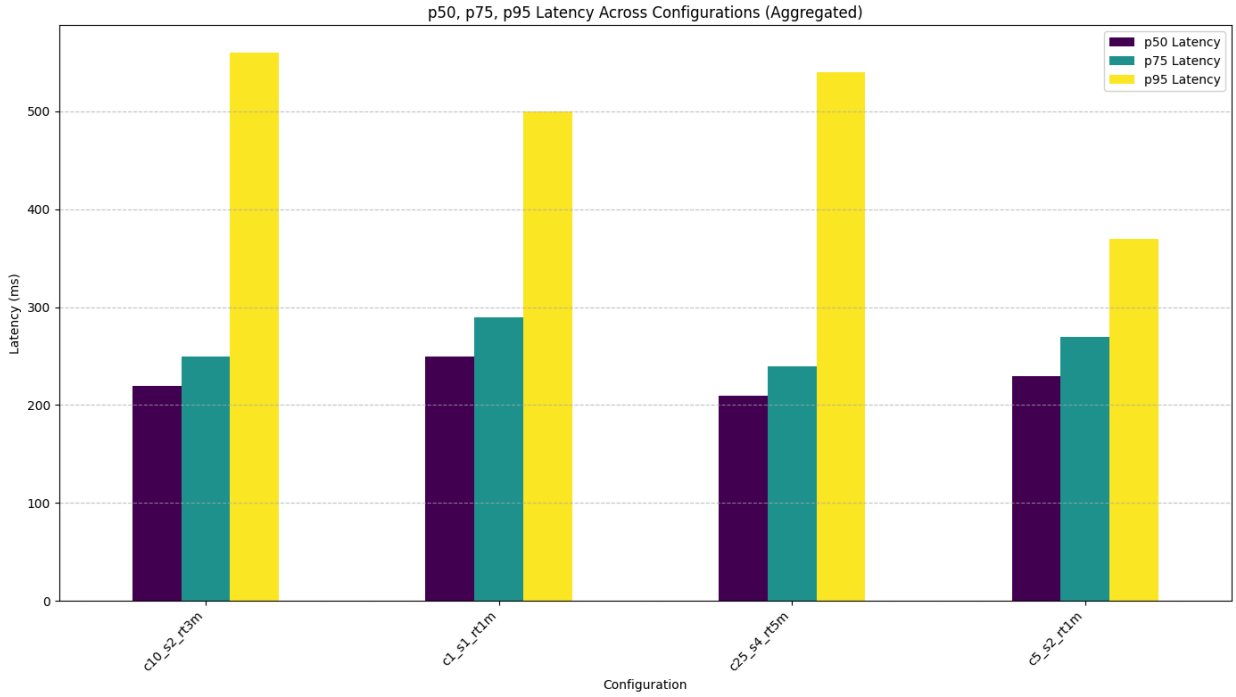
Performance Interpretation Report

This report summarizes the performance analysis of the SarvamAI's transliteration API under various load conditions, focusing on latency, throughput, and error rates across different configurations.

1. Executive Summary

The load tests reveal that the API generally maintains acceptable aggregated latency (p95=492.5ms) up to a certain point. However, a significant increase in load (specifically the `c25_s4_rt5m` configuration) introduces a notable error rate, indicating a bottleneck based on RPS and Error rate charts. Furthermore, certain language models, particularly Malayalam, Marathi, and Kannada, exhibit severe latency spikes under moderate to high load (`c10_s2_rt3m`), which is a critical area for immediate investigation.

2. Overall Latency Across Configurations (p50, p75, p95)

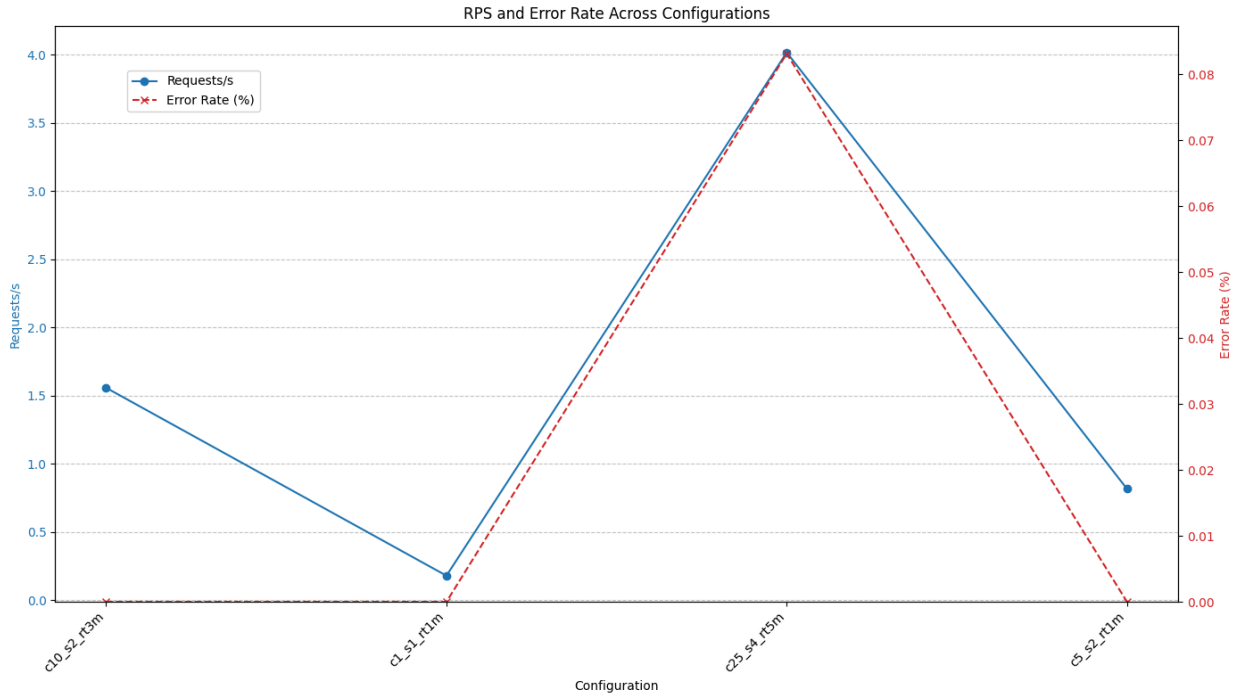


The "p50, p75, p95 Latency Across Configurations (Aggregated)" chart illustrates the overall response time percentiles:

- **c1_s1_rt1m (1 User, 1 Spawn Rate, 1 Min Run Time):** Shows a p95 latency of 500ms.
- **c5_s2_rt1m (5 Users, 2 Spawn Rate, 1 Min Run Time):** P95 latency equals to 370ms, which is lower than the baseline `c1_s1_rt1m`.

- **c10_s2_rt3m (10 Users, 2 Spawn Rate, 3 Min Run Time):** The p95 latency is similar to `c1_s1_rt1m`, equals to 560ms.
- **c25_s4_rt5m (25 Users, 4 Spawn Rate, 5 Min Run Time):** This configuration also maintains a p95 latency equals to 540ms.

3. Requests per Second (RPS) and Error Rate Across Configurations



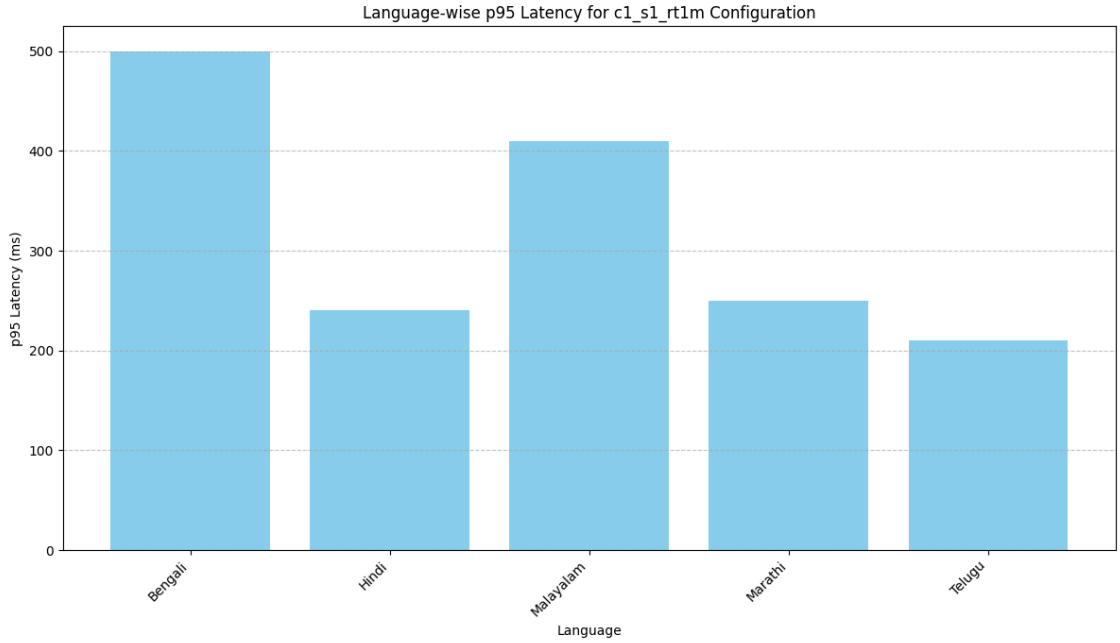
The "RPS and Error Rate Across Configurations" chart provides insight into the system's throughput and stability:

- **c1_s1_rt1m:** Achieves a very low RPS of about 0.15 requests/second with a 0% error rate.
- **c5_s2_rt1m:** Shows an RPS of approximately 0.8 requests/second with a 0% error rate.
- **c10_s2_rt3m:** Achieves an RPS of about 1.5 requests/second with a 0% error rate.
- **c25_s4_rt5m:** This configuration reaches the highest RPS at just over 4 requests/second, but crucially, experiences a significant spike in Error Rate to over 0.08%.

4. Language-wise p95 Latency Comparisons

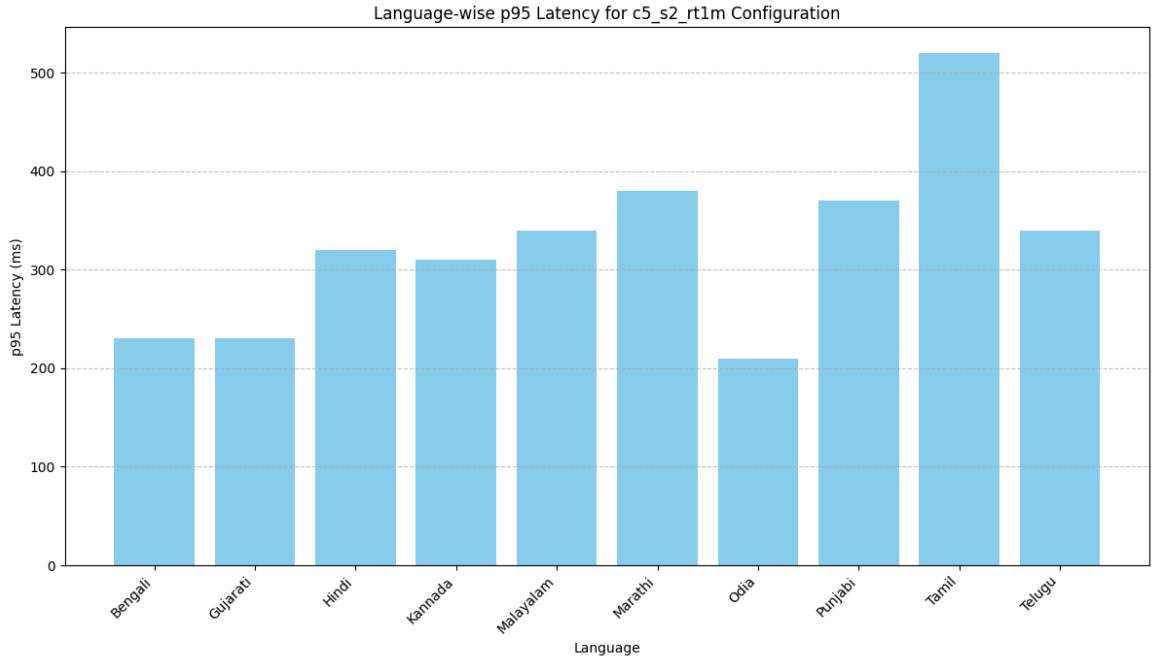
Individual language performance varies significantly across configurations, especially under increasing load.

4.1. c1_s1_rt1m Configuration



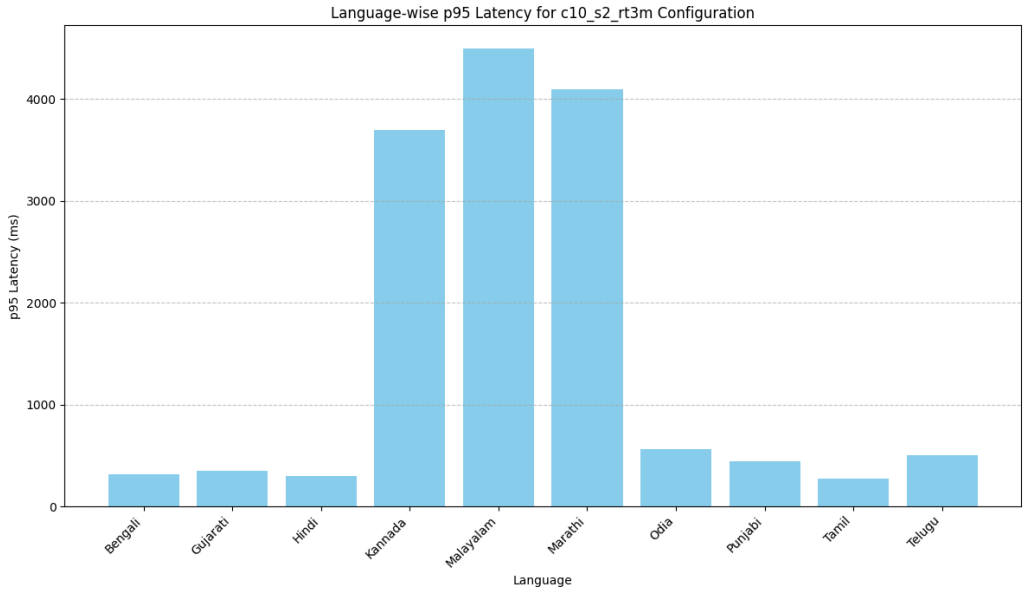
- **Bengali** shows the highest p95 latency at 500ms.
- Hindi, Marathi, and Telugu are around 200-250ms, while Malayalam is over 400ms.

4.2. c5_s2_rt1m Configuration



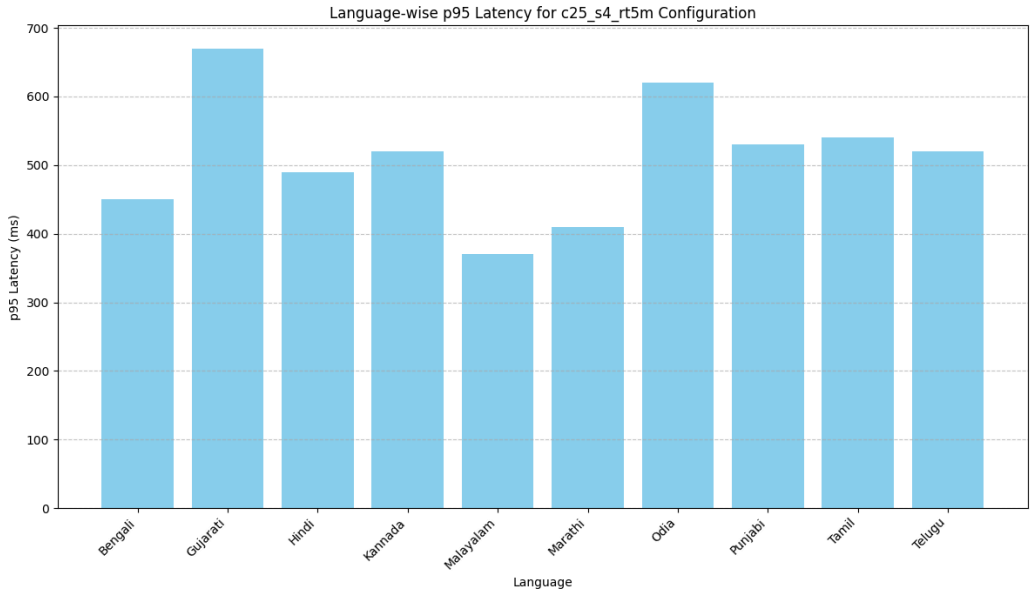
- **Tamil** shows the highest p95 latency at approximately 510ms.
- Hindi, Malayalam, and Marathi are in the 300-400ms range. Odia, Gujrati and Bengali are lowest around 210-230ms.

4.3. c10_s2_rt3m Configuration



- This configuration reveals **extreme p95 latencies** for **Malayalam (over 4500ms)** and **Marathi (over 4000ms)**.
- **Kannada** also experiences a high latency of over 3500ms.
- Other languages like Bengali, Gujarati, and Hindi remain relatively low (under 500ms) in this scenario.

4.4. c25_s4_rt5m Configuration



- **Gujarati** shows the highest p95 latency at approximately 670ms.
- **Odia** also has a high latency around 620ms.
- Most other languages fall within the 400-550ms range.

5. Identified Bottlenecks and Thresholds

- **High Concurrency Bottleneck:** The most significant bottleneck appears at the **`c25_s4_rt5m` configuration**. At 25 concurrent users with a spawn rate of 4 users/second, the system experiences a clear increase in **Error Rate (over 0.08%)**, indicating a limit to its stability and capacity under these conditions.
- **Language-Specific Performance Degradation:** Prior to outright errors, the **`c10_s2_rt3m` configuration** reveals severe performance degradation for **Malayalam, Marathi, and Kannada**. While the overall error rate for this configuration is still 0%, the p95 latencies for these specific languages soar into the thousands of milliseconds, signaling a critical per-language bottleneck.
- **NOTE: `c25_s4_rt5m` and `c10_s2_rt3m` configuration** was having a very higher error rate when the wait_time was between(0, 5), after performing 3-4 trials, I found that wait_time=between(4, 8) will be most suitable one.

6. Languages Showing Higher Latency

While not consistently highest across *all* configurations, the following languages demonstrate concerning high latencies under increased load:

- **Malayalam, Marathi, and Kannada:** Exhibit extreme latency spikes in the `c10_s2_rt3m` configuration, making them critical areas for investigation.
- **Gujarati and Odia:** Show elevated p95 latencies in the `c25_s4_rt5m` configuration.

7. Performance Test Dashboard

I've built a simple web application using streamlit with customizable concurrency, spawn rate, and run time. This dashboard allows users to configure and trigger Locust tests without needing to interact directly with the command line.

Key Features:

- **Adjustable Load Parameters:** Input fields allow you to define the number of concurrent users, the spawn rate(users per second), and the total run time of the test.
- **One-Click Test Trigger:** A dedicated button initiates the Locust test with specified parameters.
- **Results Saved Automatically:** All test results(CSV reports, Charts) are automatically saved to a dedicated time-stamped folder within the results directory.
- **Multiple Test Configurations:** Now, you can add and manage multiple distinct test configurations within a single session, allowing for comprehensive testing across various load scenarios.

Made by Aditya Chaudhary(22DCS002)