Load Testing Report - Sarvam Transliteration API

Name: Abhinav Singh

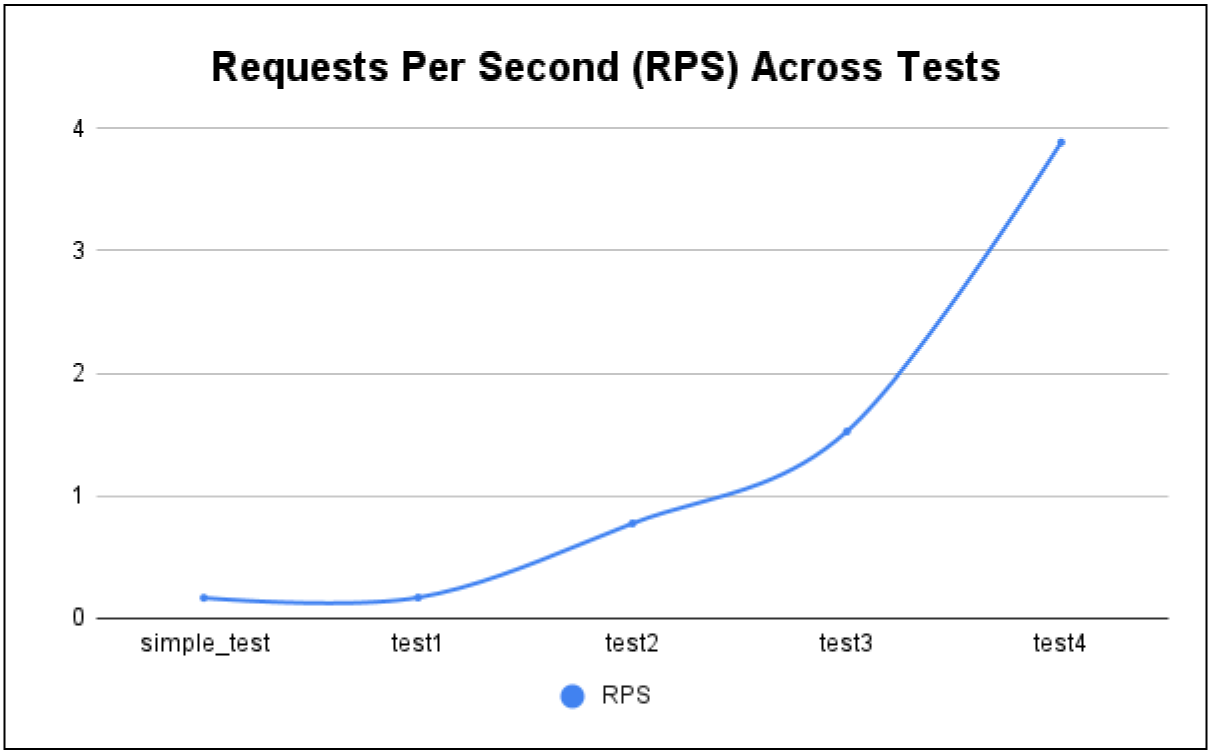
Email: 22ucs004@lnmiit.ac.in

Summary

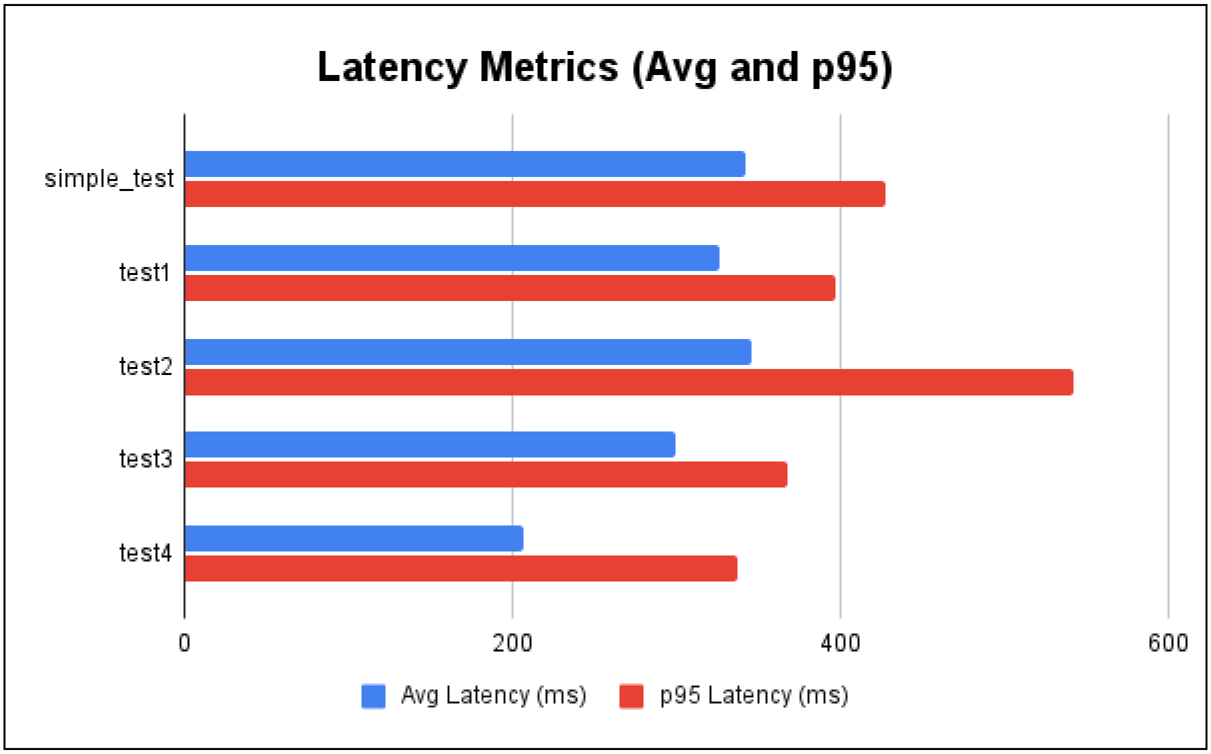
This report analyzes the performance of the Sarvam Transliteration API under various load conditions, tested with configurations ranging from 1 to 25 concurrent users. The API demonstrates robust performance up to 10 concurrent users but shows scalability limitations at higher loads, with a notable error rate increase. Language-specific latency across Hindi, Tamil, and Bengali remains consistent.

Visualizations

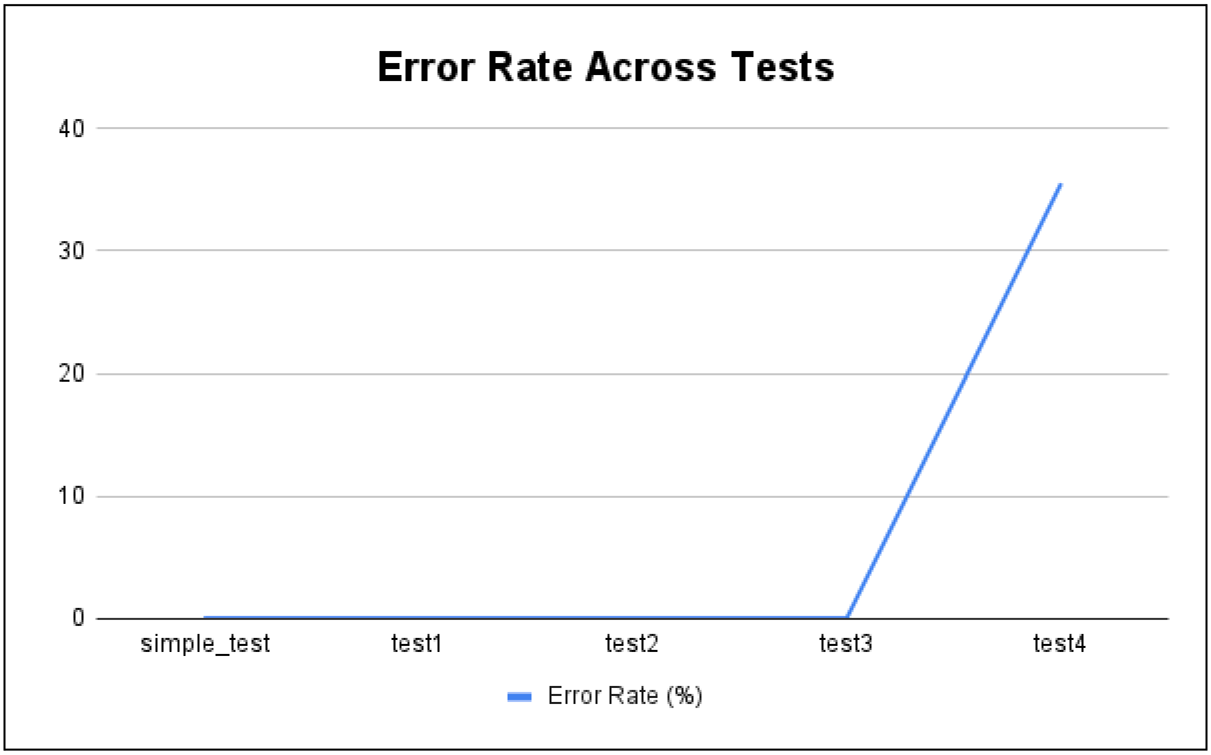
Requests Per Second (RPS) Across Tests



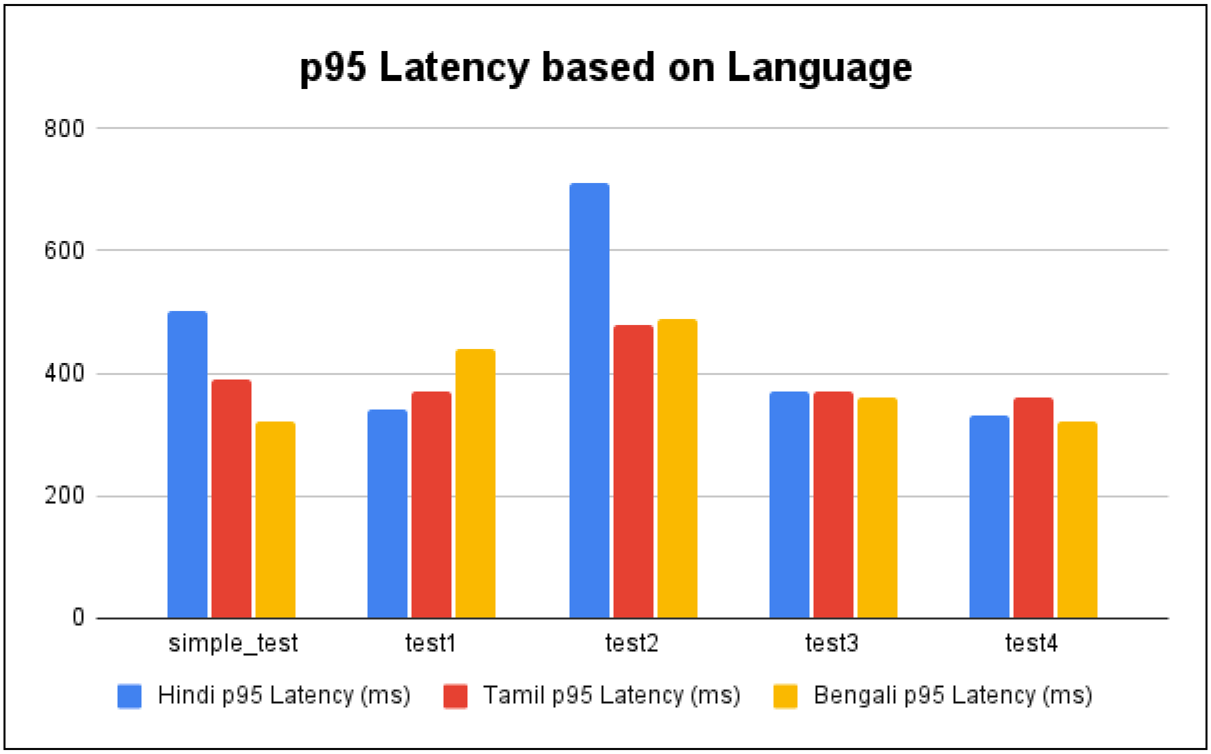
Latency Metrics (Avg and p95)



Error Rate Across Tests



p95 Latency by Language



Key Findings

- **Scalability:** The API handles up to 10 concurrent users (test3) with 0% error rate, processing 1,092 requests in 3 minutes. At 25 concurrent users (test4), the error rate spikes to 35.5%, with 827 failures out of 4,654 requests.
- **Latency:** Average latency decreases from 342 ms (simple_test) to 206 ms (test4) for successful requests, likely due to failed requests being excluded. Maximum response time reaches 2,507 ms in test4, indicating occasional delays.
- **Language Performance:** No consistent latency disparity across Hindi, Tamil, and Bengali. Hindi shows a peak p95 latency of 710 ms in test2, but values align closely in other tests (e.g., test4: 330-360 ms).
- **Interesting Fact:** Despite a 35.5% error rate in test4, successful requests were faster than in lighter tests, suggesting efficient handling or quick rejection of excess load.

Recommendations

- Optimize the API to handle >25 concurrent users, addressing the high error rate in test4.
- Investigate error causes (e.g., rate limiting, resource limits) using server logs or response codes.
- Monitor and mitigate occasional latency spikes, as seen with max response times in test4.

Conclusion

The Sarvam Transliteration API excels under moderate loads but faces challenges at higher concurrency, evidenced by a significant error rate increase. Language-specific performance is consistent, with no notable disparities. Scaling improvements and error analysis are recommended to enhance reliability under heavy load.