Dhirubhai Ambani University Technology

Formerly DA-IICT

AegisSecure

Comprehensive Performance Testing Report

Subject: Load, Spike, Soak, and Stress Testing Analysis

Platform: Apache JMeter & Hugging Face Deployment

Group Number: 35

Mevada Soham Meghalkumar [202301484]
Gohil Suryadeepsinh Hardevsinh [202301463]
Rana Neelabh Vijaykumar [202301476]
Hrithik B Patel [202301441]
Vadsmiya Pransu Pradipkumar [202301445]
Dhruv Jigneshkumar Patel [202301095]
Bhagiya Jenish Rameshbhai [202301480]
Akshat Bhatt [202301460]
Vrajkumar Makwana [202301436]
Chavda Mihirsinh Labhubhai [202301479]

IT314 - Software Engineering

Prof: Saurabh Tiwari | Mentor: Shyam Patel

Contents

1 Deployment Strategy & Infrastructure **2**

Deployment Strategy & Infrastructure **2**

2 Testing Methodology **2**

Testing Methodology **2**

3 Performance Test Results **3**

Performance Test Results **3**

3.1 Load Testing 3

3.2 Spike Testing 5

3.3 Soak Testing 7

3.4 Stress Testing 9

4 Some Optimizations and their Status: **10**

Optimizations **10**

1 Deployment Strategy & Infrastructure

The application was deployed using the **Hugging Face Free Tier**.

Infrastructure Decisions

Why Hugging Face?

- **Requirements:** The Machine Learning model exceeds 750MB. While Render's free tier is limited to 512MB, Hugging Face provides a generous 16GB of RAM and 1GB of storage, making it the superior choice for our resource-intensive model.
- **Google Cloud:** Google Cloud Free Tier was evaluated but rejected due to the mandatory security deposit requirement.
- **Latency Management:** Unlike Render, which forces a "cold start" of 6-7 minutes after inactivity, Hugging Face offers a significantly faster startup time, ensuring the application remains responsive even after idle periods.

2 Testing Methodology

Database State

During the performance of these tests, the database was contained data prior to testing, including approximately 438 emails, 166 SMS messages, and various linked user accounts.

Simulation Setup (Apache JMeter)

Tests were conducted using **Apache JMeter** via a mock user test case designed to mimic real-world behavioral patterns.

Defined User Flow:

1. Login → Dashboard (Both Views) → Dashboard (SMS View)
2. Dashboard (Mail View) → Account View → Dashboard
3. Email Fetch → Mail Account Manager
4. Inbox Change → Email Fetch → SMS List

Note: OAuth endpoints requiring real Gmail credentials and token-heavy Spam Prediction endpoints were excluded from this specific high-volume test suite to prevent API rate limiting.

3 Performance Test Results

3.1 Load Testing

Objective: To verify system stability under sustained normal load.

Test Configuration

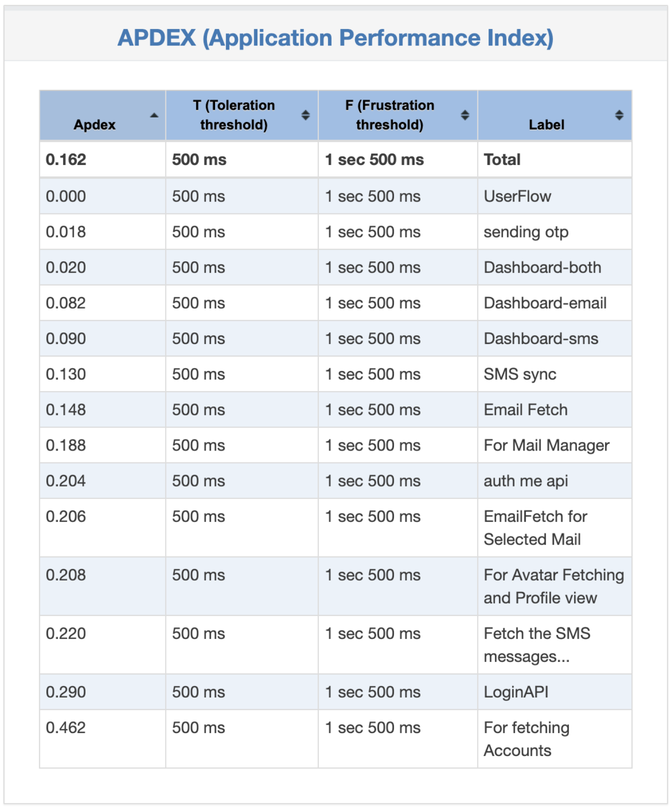
Number of Threads: 50
Number of Loops: 5

Load Test Analysis

Results Summary:

- **Pass Rate:** 99.63%
- **Average Response Time:** 350 seconds (Full User Flow)

Observation: The system maintained stability with a 99.63% pass rate. The failure rate of 0.37% was statistically insignificant and isolated to "Unauthorized requests." These errors occurred because a specific thread failed the initial login step, rendering subsequent requests in that loop unauthorized due to a missing JWT token.



APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.162	**500 ms**	**1 sec 500 ms**	**Total**
0.000	500 ms	1 sec 500 ms	UserFlow
0.018	500 ms	1 sec 500 ms	sending otp
0.020	500 ms	1 sec 500 ms	Dashboard-both
0.082	500 ms	1 sec 500 ms	Dashboard-email
0.090	500 ms	1 sec 500 ms	Dashboard-sms
0.130	500 ms	1 sec 500 ms	SMS sync
0.148	500 ms	1 sec 500 ms	Email Fetch
0.188	500 ms	1 sec 500 ms	For Mail Manager
0.204	500 ms	1 sec 500 ms	auth me api
0.206	500 ms	1 sec 500 ms	EmailFetch for Selected Mail
0.208	500 ms	1 sec 500 ms	For Avatar Fetching and Profile view
0.220	500 ms	1 sec 500 ms	Fetch the SMS messages...
0.290	500 ms	1 sec 500 ms	LoginAPI
0.462	500 ms	1 sec 500 ms	For fetching Accounts

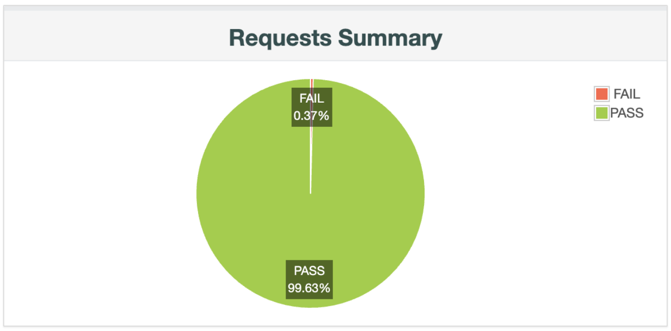


Figure 1: **Load Test Summary:** The dashboard confirms the 99.63% success rate. The visual distribution indicates that error rates remained negligible relative to the total transaction volume.

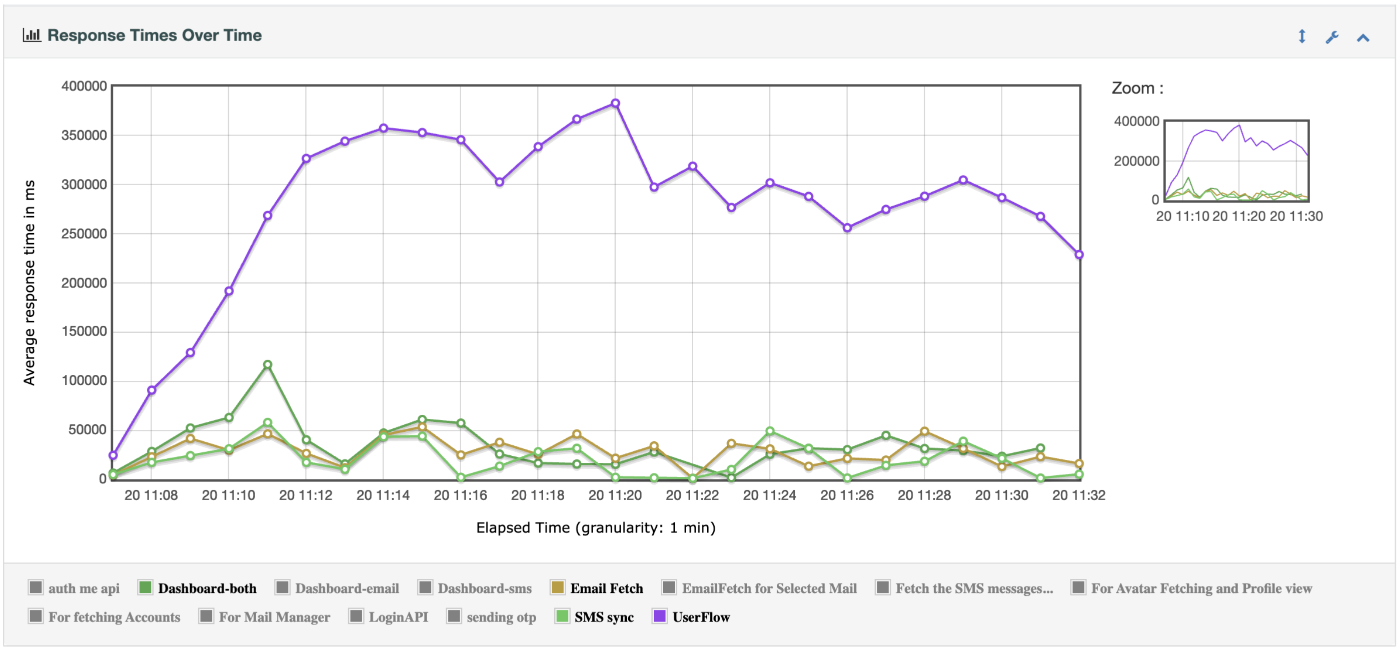


Figure 2: **Response Time Under Load:** The graph indicates that for an aggregate of 50 users repeating the cycle 5 times, the application handled the concurrency without system failure. However, the average response time was notably high due to the resource constraints of the hosting tier as well as system processing.

For full report use this link:
https://drive.google.com/drive/folders/1W-OBmV0J05YQeoWBhh4T3L8x-JGvwBKT?usp=sharing

3.2 Spike Testing

Objective: To validate system behavior during sudden surges in user activity (scaling from 0 to 200+ threads rapidly).

Test Configuration

Thread Group 1 (Aegis70): 70 threads, 2 loops
Thread Group 2 (Aegis100): 100 threads, 2 loops
Thread Group 3 (Aegis50): 50 threads, 4 loops

Spike Test Analysis

Results Summary:

- **Pass Rate:** 99.97%
- **Fail Rate:** 0.03%

Observation: The system demonstrated elasticity. Despite a rapid ramp-up to over 200 concurrent threads, the failure rate was negligible (0.03

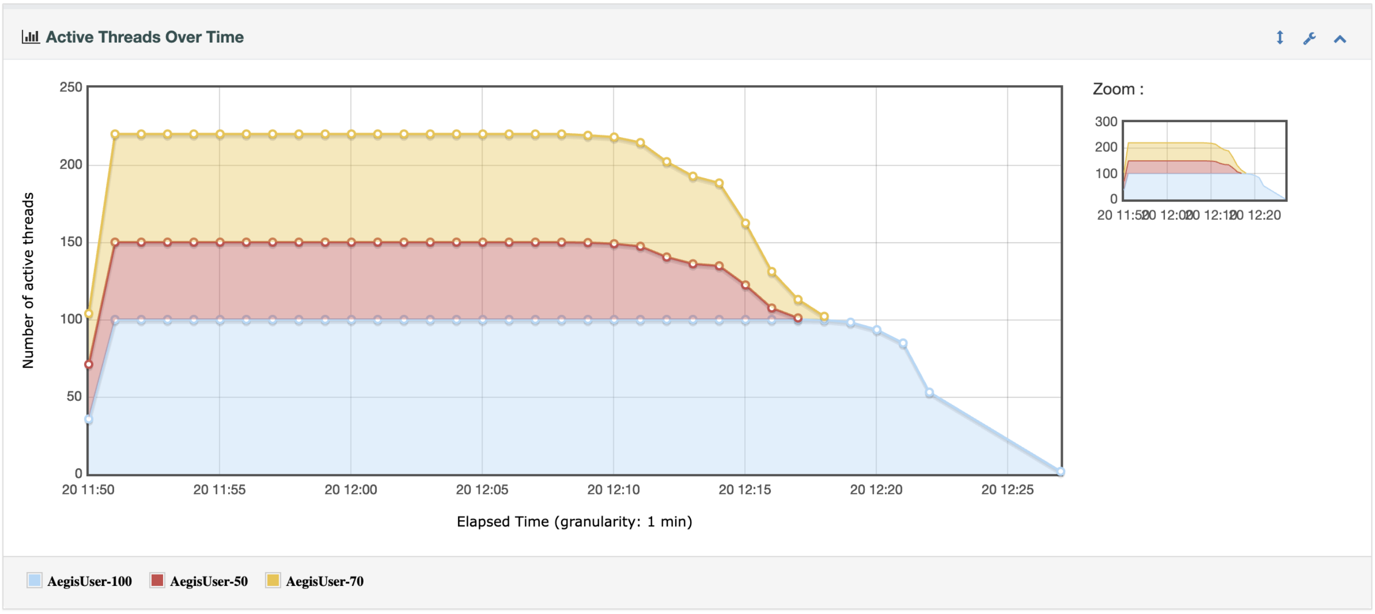
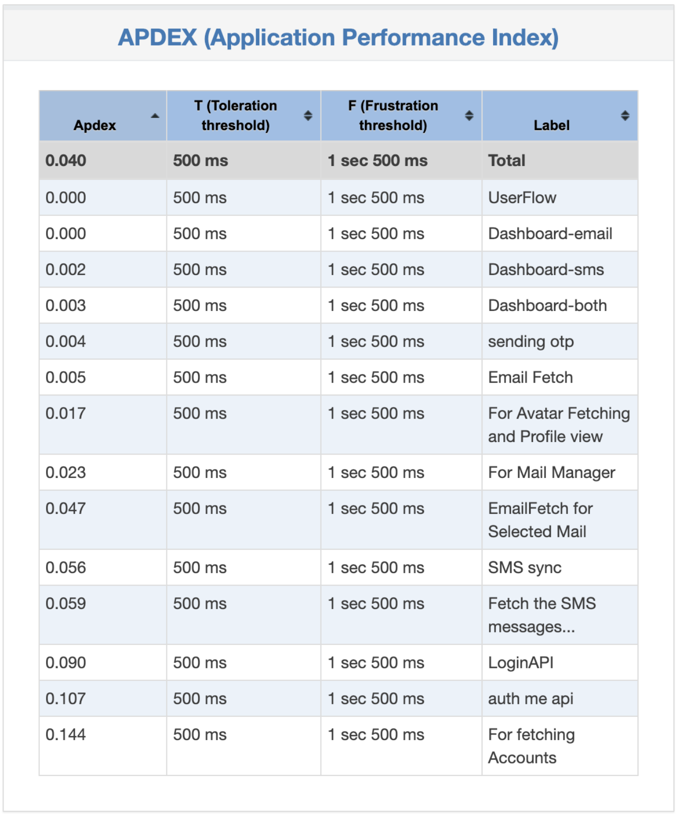


Figure 3: **Thread Ramp-Up:** This graph visualizes the "Spike" profile. The active thread count escalates rapidly to 200+ concurrent users across the three distinct thread groups, sustains the peak, and then descends.



APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.040	**500 ms**	**1 sec 500 ms**	**Total**
0.000	500 ms	1 sec 500 ms	UserFlow
0.000	500 ms	1 sec 500 ms	Dashboard-email
0.002	500 ms	1 sec 500 ms	Dashboard-sms
0.003	500 ms	1 sec 500 ms	Dashboard-both
0.004	500 ms	1 sec 500 ms	sending otp
0.005	500 ms	1 sec 500 ms	Email Fetch
0.017	500 ms	1 sec 500 ms	For Avatar Fetching and Profile view
0.023	500 ms	1 sec 500 ms	For Mail Manager
0.047	500 ms	1 sec 500 ms	EmailFetch for Selected Mail
0.056	500 ms	1 sec 500 ms	SMS sync
0.059	500 ms	1 sec 500 ms	Fetch the SMS messages...
0.090	500 ms	1 sec 500 ms	LoginAPI
0.107	500 ms	1 sec 500 ms	auth me api
0.144	500 ms	1 sec 500 ms	For fetching Accounts

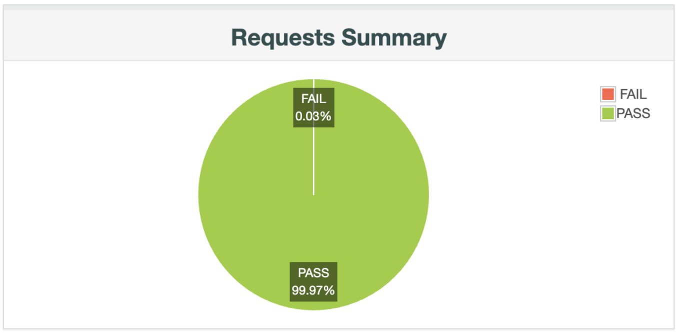


Figure 4: **Spike Performance Overview:** The summary table indicates that APDEX (Application Performance Index) scores remained within tolerable limits, verifying that the system availability was not compromised during the surge.

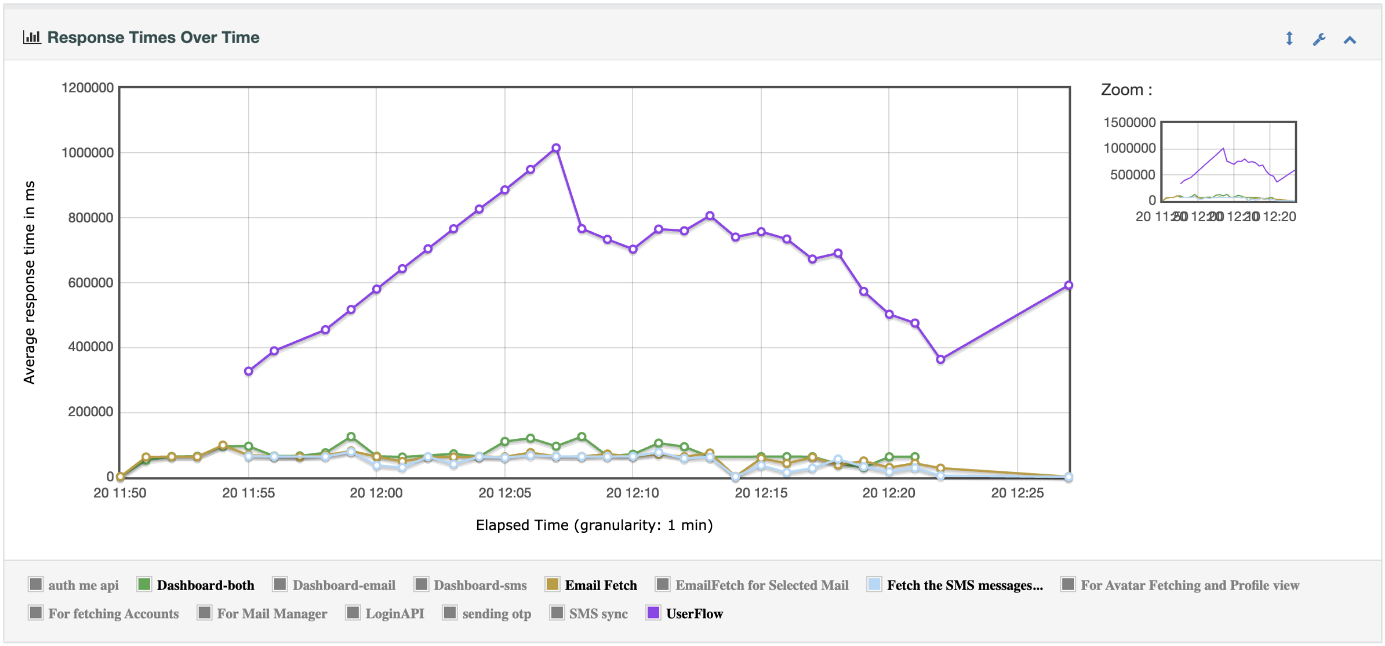


Figure 5: **Latency Response to Spikes:** Response times correlated with thread count, peaking during the maximum load window. The subsequent immediate recovery in response time indicates the system clears the request queue efficiently once load decreases.

For full report use this link:
https://drive.google.com/drive/folders/1X3N43Q42oACxpzkLE8ssvhLUXT9sjDYf?usp=sharing

3.3 Soak Testing

Objective: To verify system stability and check for memory leaks over an extended period.

Test Configuration

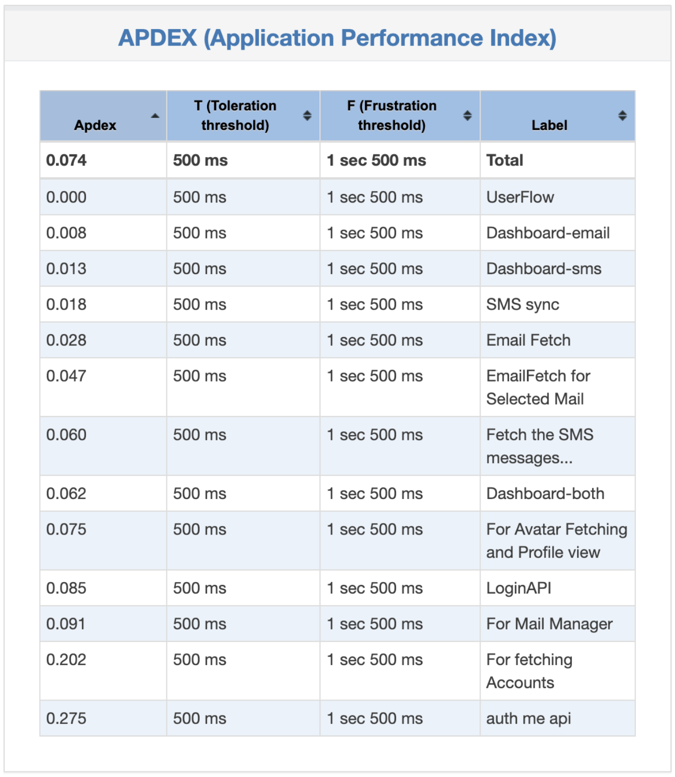
Number of Threads: 100
Number of Loops: 2

Soak Test Analysis

Results Summary:

- **Pass Rate:** 97.67%
- **Degradation:** Throughput degradation observed over the test duration.

Observation: A cumulative error rate of 2.33% was observed. Analysis of the logs indicates this was primarily driven by a login failure in a single thread; this failure cascaded, causing all subsequent requests in that specific thread's loop to return "Unauthorized," thereby inflating the error count.



APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.074	**500 ms**	**1 sec 500 ms**	**Total**
0.000	500 ms	1 sec 500 ms	UserFlow
0.008	500 ms	1 sec 500 ms	Dashboard-email
0.013	500 ms	1 sec 500 ms	Dashboard-sms
0.018	500 ms	1 sec 500 ms	SMS sync
0.028	500 ms	1 sec 500 ms	Email Fetch
0.047	500 ms	1 sec 500 ms	EmailFetch for Selected Mail
0.060	500 ms	1 sec 500 ms	Fetch the SMS messages...
0.062	500 ms	1 sec 500 ms	Dashboard-both
0.075	500 ms	1 sec 500 ms	For Avatar Fetching and Profile view
0.085	500 ms	1 sec 500 ms	LoginAPI
0.091	500 ms	1 sec 500 ms	For Mail Manager
0.202	500 ms	1 sec 500 ms	For fetching Accounts
0.275	500 ms	1 sec 500 ms	auth me api

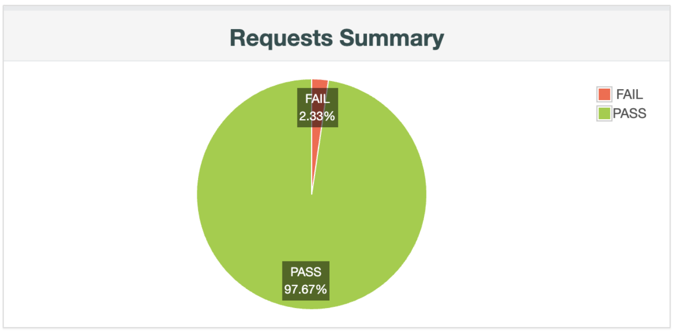


Figure 6: **Soak Performance Overview:** The dashboard highlights the breakdown of passed vs. failed transactions. While the majority of requests were successful, the non-zero error rate suggests potential session token expiration or resource limits during extended usage.

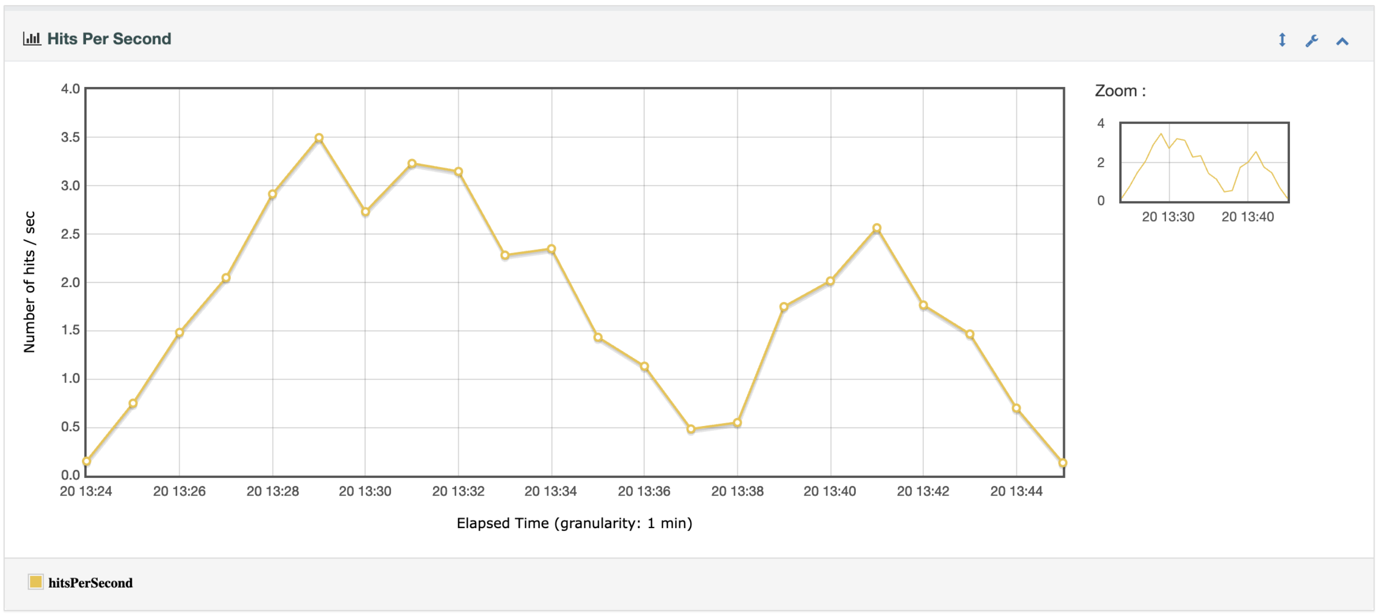


Figure 7: **Throughput Consistency:** The Hits Per Second graph shows a relatively stable request rate initially, followed by a slight downward trend. This behavior is consistent with resource exhaustion typical in Free Tier deployments under sustained load.

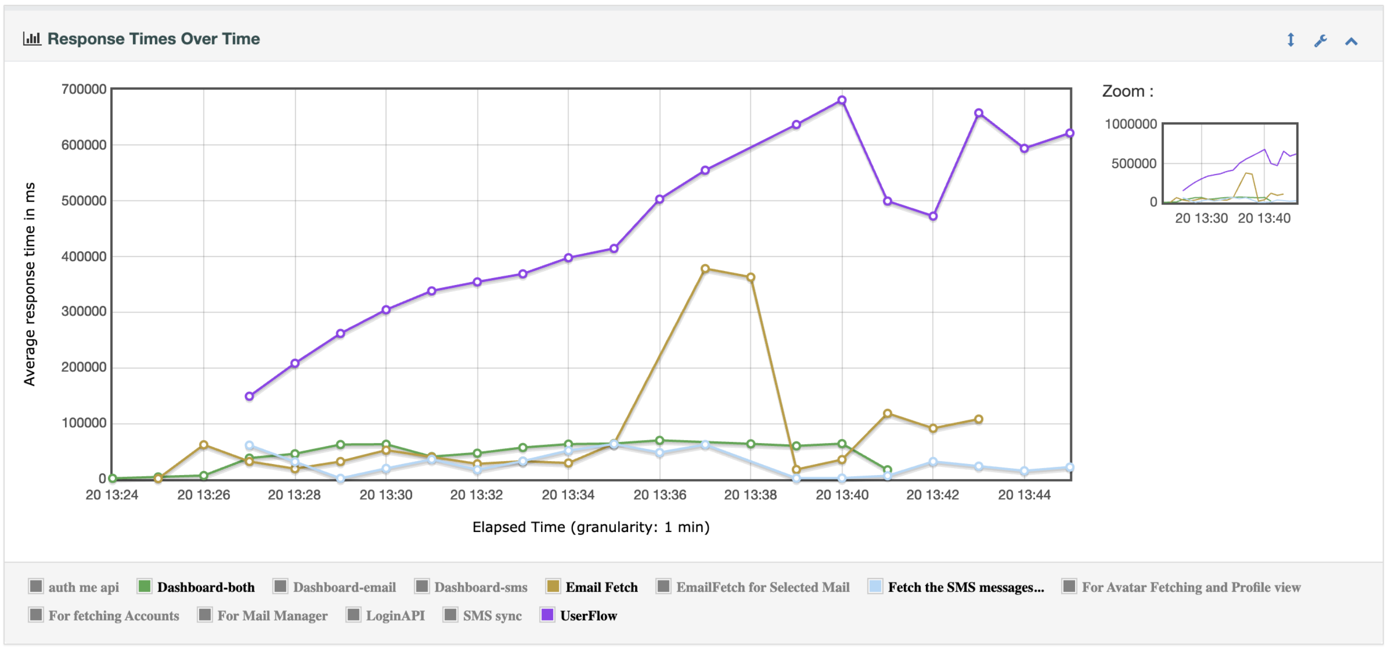


Figure 8: **Long-Duration Latency:** The upward trajectory of the response time graph toward the end of the test cycle indicates a gradual degradation in performance, likely due to accumulated memory usage or connection pool saturation.

For full report use this link:
https://drive.google.com/drive/folders/1vMcG1m_jHN-mSQKMq1BHJFv8aEr617GQ?usp=sharing

3.4 Stress Testing

Objective: To determine the system's breaking point by overwhelming it with requests beyond expected capacity.

Test Configuration

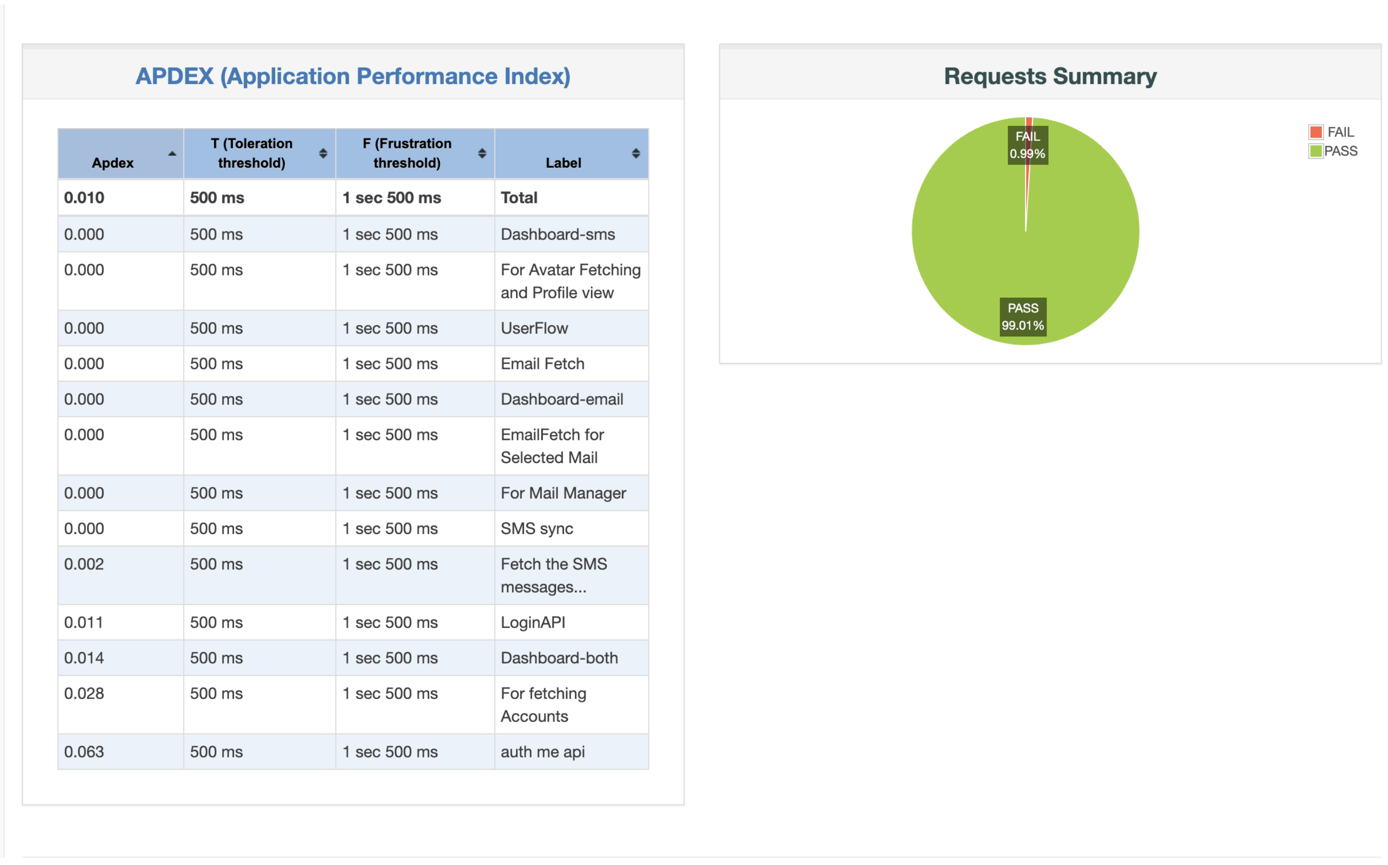
Thread Type 1: 200 threads (Start time: 0 min)
Thread Type 2: 100 threads (Start time: +10 min)
Thread Type 3: 70 threads (Start time: +5 min)

Stress Test Analysis

Results Summary:

- **Pass Rate:** 99.01%
- **Behavior:** High volatility in throughput and latency.

Observation: The system maintained a 99.01% success rate under stress conditions. However, the throughput metrics exhibited high variance, indicating that the server reached its processing capacity and was struggling to accept new connections at a consistent rate.



APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.010	**500 ms**	**1 sec 500 ms**	**Total**
0.000	500 ms	1 sec 500 ms	Dashboard-sms
0.000	500 ms	1 sec 500 ms	For Avatar Fetching and Profile view
0.000	500 ms	1 sec 500 ms	UserFlow
0.000	500 ms	1 sec 500 ms	Email Fetch
0.000	500 ms	1 sec 500 ms	Dashboard-email
0.000	500 ms	1 sec 500 ms	EmailFetch for Selected Mail
0.000	500 ms	1 sec 500 ms	For Mail Manager
0.000	500 ms	1 sec 500 ms	SMS sync
0.002	500 ms	1 sec 500 ms	Fetch the SMS messages...
0.011	500 ms	1 sec 500 ms	LoginAPI
0.014	500 ms	1 sec 500 ms	Dashboard-both
0.028	500 ms	1 sec 500 ms	For fetching Accounts
0.063	500 ms	1 sec 500 ms	auth me api

Requests Summary

Figure 9: **Dashboard Stress Metrics:** The dashboard visualizes the impact of the staggered thread start times. Despite the heavy load introduced by 370 cumulative threads, the system avoided a complete outage, maintaining a high percentage of successful transactions.

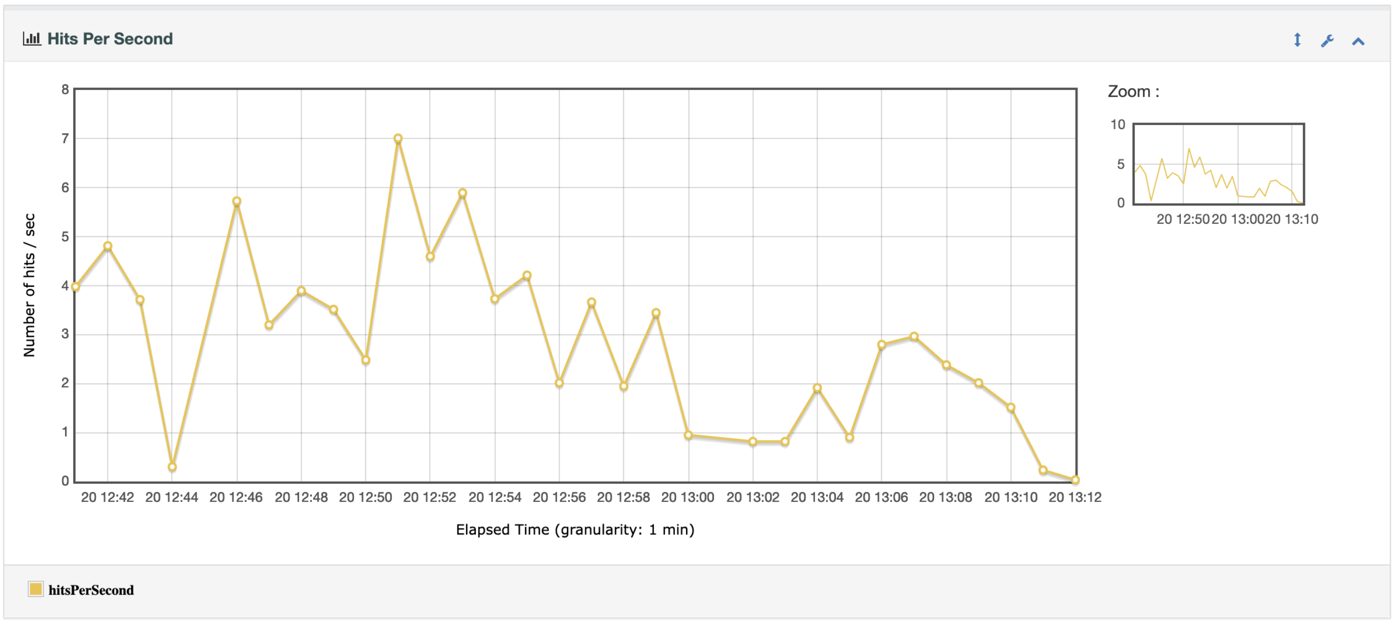


Figure 10: **Stress Volatility:** Unlike the smooth graphs observed in Load Testing, this chart displays sharp peaks and troughs. This instability confirms the server was operating at maximum capacity, resulting in intermittent request throttling.

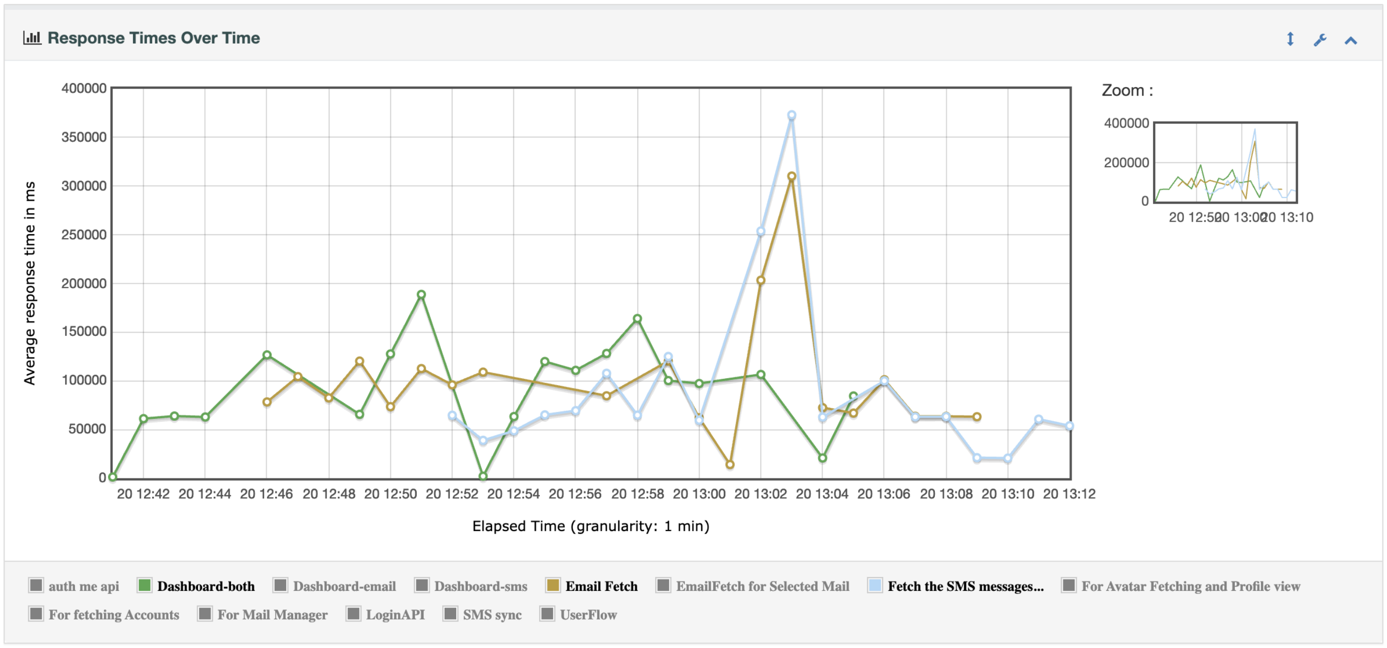


Figure 11: **Extreme Latency Spikes:** Significant outliers in response time (exceeding 300 seconds in some instances) confirm the stress conditions. The system processed requests significantly slower, but recovered without a total crash.

For full report use this link:
https://drive.google.com/drive/folders/16Uc_7dU8k_CpOMEClQPm7-3ehUxuGcIb?usp=sharing

4 Some Optimizations and their Status:

Based on the performance metrics gathered, the following optimizations were analyzed:

1. Pagination Implementation

Strategy: Implement pagination for fetching SMS and emails (batching 10-15 messages) to reduce payload size.
Status: Attempted but deferred. The implementation required extensive architectural changes to the database schema and frontend logic which could not be completed within the current sprint timeline.

2. Data Retention Policy

Strategy: Limit data storage to the last 100 messages per user.
Status: Rejected. This approach leads to data inconsistency in the Dashboard and negatively impacts user experience, as users often need to review analysis results for older messages.

3. Database Indexing

Strategy: Application of stronger indices to improve query speeds.
Status: Implemented. We optimized several database indices. While theoretically sound, the observable optimization in this specific test environment was minimal, likely due to the dataset size not yet reaching critical mass.

Conclusion

The application demonstrates resilience on the Hugging Face Free Tier, with pass rates consistently exceeding 97% across all testing scenarios. While response time latency increases significantly under concurrent load due to the inherent hardware limitations of the free tier, the system successfully handles spikes and sustained usage without catastrophic failure, validating the architecture for the initial release.