Achieving superlative performance for India’s largest E-commerce platform

Flipkart is a leading e-commerce platform in India, offering a wide range of products and services for both sellers and customers. Known for its annual Big Billion Days sale, the platform sees millions of users shopping across multiple categories. Ensuring a seamless shopping experience, even during high-traffic periods, is critical for the business to maintain customer satisfaction and driving revenue growth.

The situation before

Client was facing scalability and performance issues, resulting in loss of revenue and customer dissatisfaction. These issues risked poor user experience during high-traffic periods, directly impacting business growth.

1. **Order API Bottleneck:** Performance flatlining at 1000-1500 users due to autoscaling issues.

2. **Warehouse Queue Slowness:** Order processing queue consuming only 30-40 orders/second.

3. **CPU Bottlenecks:** 10% fewer impressions per page on the front end due to CPU constraints

Key Highlights

30%
overall backend performance improvement

10%
increase in impressions per page.

2X
increase in rate of processing

5X
increase in scalability

Solution

To address these challenges, a two phased approach was implemented, focusing on system analysis and realistic scenario-based testing.

- Leveraged client's custom in-house PaaS/IaC tool built on Locust, automated load tests, and simulated traffic surges that exposed inefficiencies in autoscaling.
- Monitored downstream systems to assess application's performance. Automatically scaled additional systems when server usage exceeded its limits and redirected users via load balancers resulting in smoother mixed item order processing.
- Analyzed GPU overdraw to fix performance lags and conducted a detailed network analysis to identify and eliminate any unnecessary data in the request and response payloads.
- Monitored system performance under high load through spike and volume tests. CPU bottlenecks were identified and resolved, boosting front end efficiency and improving impressions per page.
- Introduced efficient test data management by loading only the necessary data into memory before execution.
- Developed a reusable framework for backend and frontend API calls, minimizing latency and optimizing system resources for better load handling.
- Optimized the order confirmation API response body for faster processing and enhanced the infrastructure by adding load balancers.

Business outcome

- The platform can now handle larger user volumes, increasing from 1,000 to 5,000, without incurring additional infrastructure scaling costs.
- The order processing rate doubled from 30 to 60 orders/sec, improving operational efficiency, while optimizing the order confirmation API led to faster response times, enhancing the UX.
- Because of the optimized infrastructure, client can now make efficient use of their resources leading to reduced latency and cost savings.
- Better user engagement was achieved through faster page load times and increased impressions per page. Optimizing CPU usage also improved battery efficiency, particularly for mobile users.