

Mini Prometheus-Grafana Progress Report

Yunus Emre Özdemir
Hasan Kerem Şeker

Completed Tasks

Week 1 – Load Generation and Metric Exporter

Implement Python Load Generator
Develop Java server with CPU-bound mock tasks
Integrate real-time metric collection

Week 2 – Metric Monitoring and Communication

Design and implement custom UDP protocol
Build Prometheus-like Java server that logs incoming metrics
Develop in-memory data store and web socket communication

Week 3 – Dashboard and Integration

Build React dashboard and consume web socket data
Display time-series charts and thresholds

Incomplete Tasks

Make Prometheus-like Java server store incoming metrics
Finalize, test, create final report, and prepare presentation

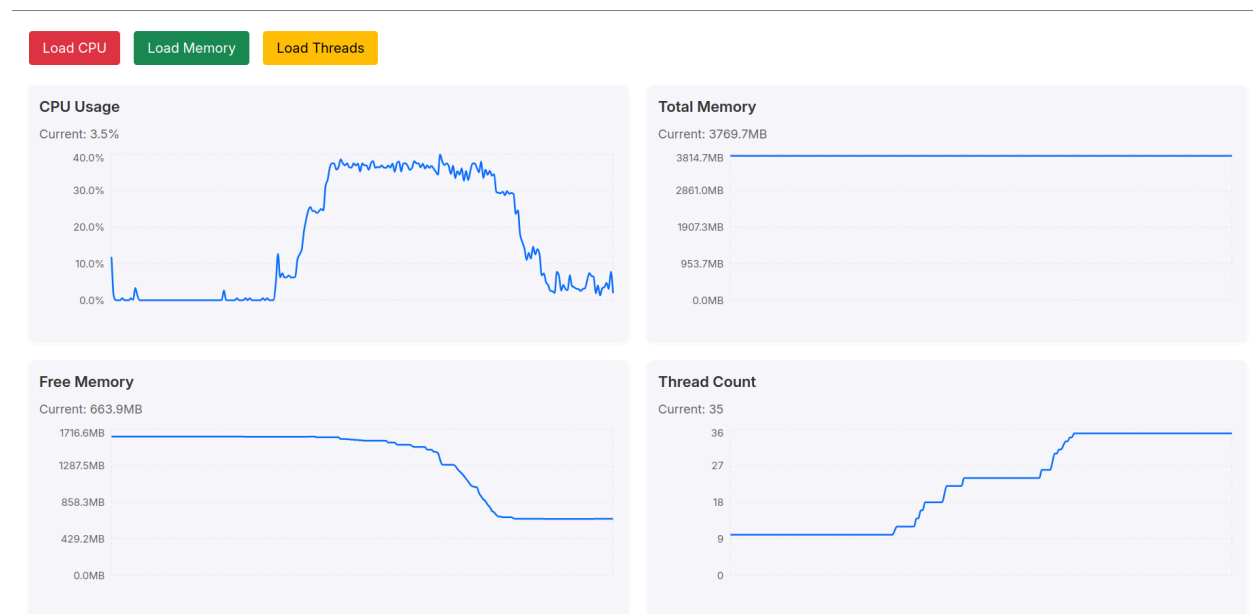
Extra Refinements

Create an utility to free up the resources after loading the system

Fine-tune memory loading parameters to reduce the number of button clicks needed for effective system load testing

Set alarm thresholds

Screenshot



This screenshot is from merged load generator and dashboard.

Plan Changes

We decided to merge the load generator and dashboard to create a seamless process for loading the system and observing its effects. This eliminates the need for users to switch between two windows when loading the system and monitoring its responses.