Announcing

Agenda

Introduction

Audience

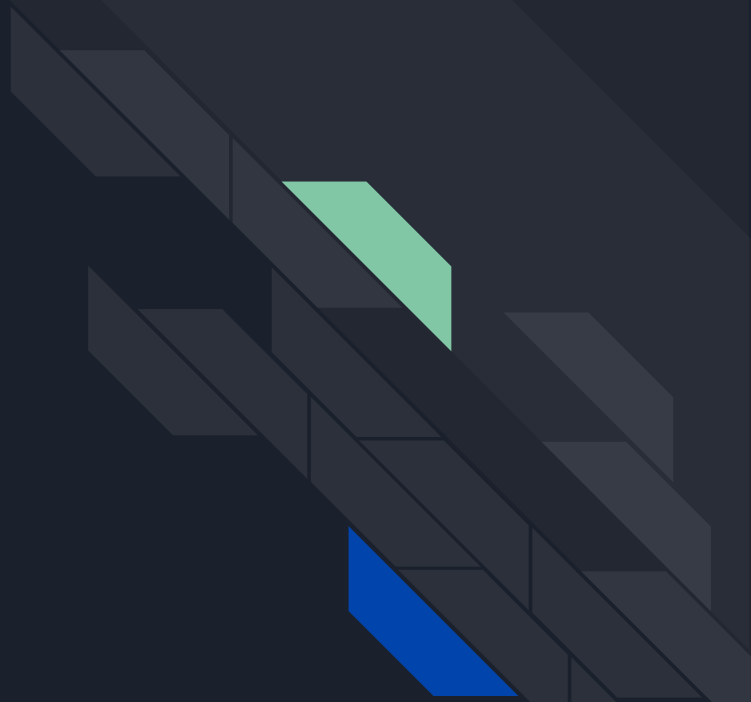
Protocol Support

Test Recorders

Custom Binaries

Results Output

Examples



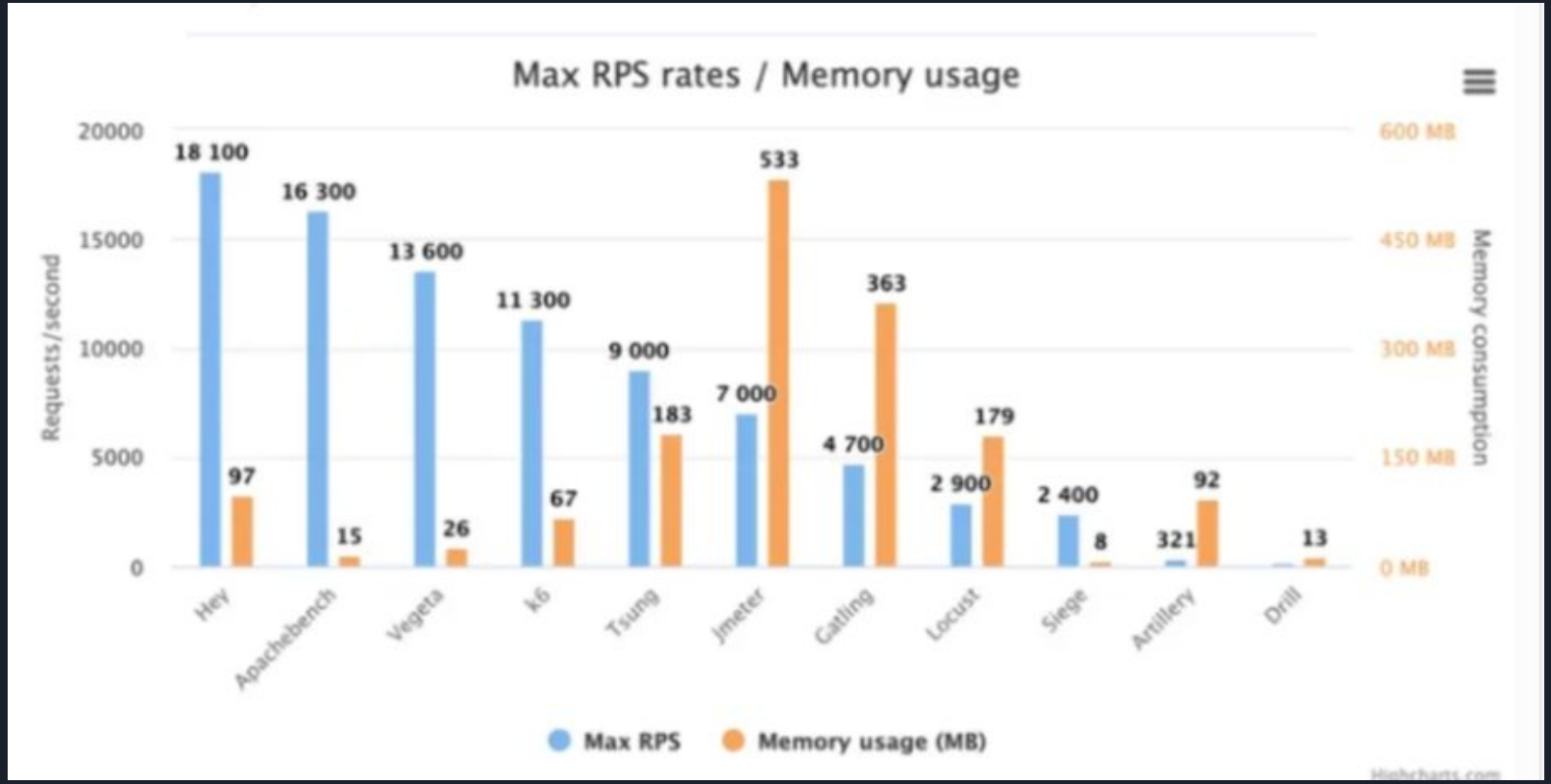
Introduction

k6 is a developer-friendly, open-source performance testing tool built on Go. It simulates concurrent virtual users to load test applications and infrastructure.

Key Features:

- User-friendly: Easy to learn and use.
- Open-source: Free and customizable.
- Versatile: Can test various performance aspects.
- Efficient: Offers memory optimization options.

By using k6, engineering teams can ensure application reliability, prevent errors, and maintain high performance.


Max RPS rates / Memory usage
Requests/second
Memory consumption
20000
15000
10000
5000
0
600 MB
450 MB
300 MB
150 MB
0 MB
18 100
97
16 300
15
13 600
26
11 300
67
9 000
183
7 000
533
4 700
363
2 900
179
2 400
8
321
92
13
Hey
Apachebench
Vegeta
k6
Tsung
Jmeter
Gatling
Locust
Siege
Artillery
Drill
Max RPS
Memory usage (MB)

Audience

Engineering teams, including Developers, QA Engineers, SDETs, and SREs, commonly use k6 for:

- Load Testing
  - Running a high load performance testing with **minimal resources**
- Browser Automation Testing
  - Can run browser based performance and automation test and **collect browser metrics**
- Synthetic Monitoring
  - Schedule a browser automation to simulate the user actions
- Chaos and resilience Testing
  - With help of custom binaries you can **simulate traffic and inject fault** on Kubernetes
- Infrastructure Testing
  - With k6 extension you can support new protocols within your infrastructure

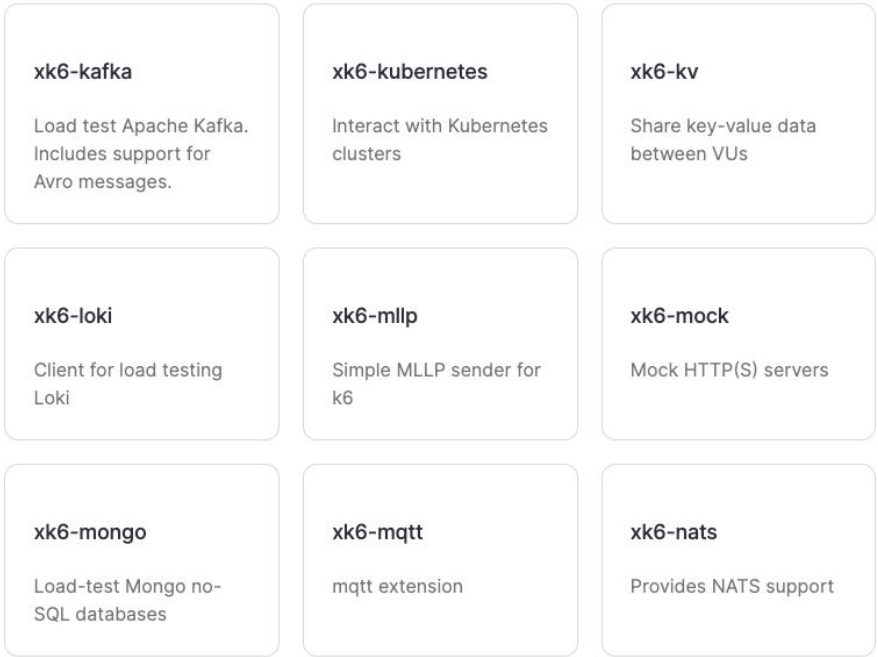
Reference : Grafana k6 | Grafana k6 documentation

Protocol Support

- HTTP /1.1
- HTTP /2
- Websocket
- gRPC

You can use k6 on more protocol with xk6

Reference : Extensions | Grafana k6 documentation



Test Recorders

A recording stores the sequence of requests and parameters of a user session or API interaction. You can use this recording to auto-generate your test logic.

- Browser Recorder
- DevTools Recorder
- HAR Converter

Results Output

k6 emits metrics with timestamps at every point of the test. You can output the metric results as either aggregated statistics or individual data points.

1. For a top-level test overview, use the end-of-test summary.
2. For granular output of all metrics (with timestamps), you stream metrics in real time.

If you stream your metrics, you can either write them to a file, like JSON, or stream them to a service, like InfluxDB, Prometheus, and etc...

Reference : https://grafana.com/docs/k6/latest/results-output/

Custom binaries

Create tailored extensions: Address specific reliability-testing needs.

Leverage the xk6 framework: Build custom k6 binaries in Go.

Two development options:

- Go and xk6:

Write extensions in Go. Bundle them with xk6 to create custom binaries.

- Docker:

Utilize a Dockerfile to build custom binaries. Incorporate extensions and dependen

```
xk6 build --with github.com/grafana/xk6-kubernetes
```

Extensions | Grafana k6 documentation

Some Examples

- Load testing
- API Testing with BDD Style
- Browser Automation
- Build custom k6 binary
- Kubernetes

Reference

- K6 — Swiss Army Knife for Dev Teams | by Sheron Gerard | Medium
- Grafana k6 | Grafana k6 documentation
- K6 Environment Setup - Projects - Wiley Global Confluence
- K6 Scripting and Execution Guide - Projects - Wiley Global Confluence
- HTTP Authentication | Grafana k6 documentation
- Integrations & Tools | Grafana k6 documentation
- GitHub - lgjsherond/PerfGuildDemo: Demo purpose

Thank you!