



Maximizando o desempenho: Explorando Testes de Performance com K6

João Martins

Breve Apresentação



- Formado em Ciência da Computação - UTFPR - CM;
- Moro a 5 anos em Maringá;
- 5 anos na área de desenvolvimento de software
 - 1 ano como fullstack;
 - 3 anos como QAOps;
 - 1 ano como QA Tech Lead;
- Atualmente trabalho como QA Especialista de Performance no Grupo Casas Bahia;
- Sócio proprietário Green Coworking;
- Fã de Filosofia, livros e empreendedorismo;



@joaomartinsfilho



joaomartinsfilho.substack.com



@jfilhogn

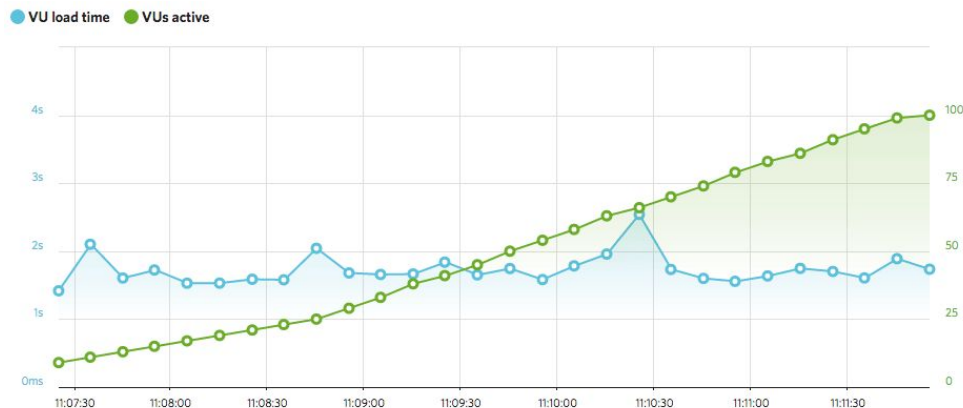
Conteúdo

- O que é teste de performance?
- Importância dos testes de performance!
- Como começar os testes?
- E quais métricas devo verificar?
- Ferramentas do mercado;
- Conhecendo o K6;
- Instalação;
- Conhecendo a documentação e API do K6;
- Prática K6 para teste de performance;
- Prática K6 para teste de API;
- Plus → Extensões



O que é teste de performance?

- O teste de performance é um subconjunto de testes de software que avalia a capacidade de resposta, estabilidade, escalabilidade e velocidade de um aplicativo sob várias condições.
- O objetivo principal é identificar gargalos, identificar problemas de desempenho e garantir que o software atenda às expectativas de desempenho antes de chegar aos usuários finais.



Importância dos testes de performance!

- Satisfação do usuário!
- Confiabilidade e Estabilidade;
- Escalabilidade;
- Economia;



Como começar os testes?

1. **Definir cenários:** identifique jornadas e interações do usuário para replicar nos testes.
2. **Scripting:** Desenvolva scripts para simular o comportamento do usuário, incluindo navegação e entrada de dados.
3. **Geração de carga:** use ferramentas de teste de desempenho para simular a carga e as interações do usuário.
4. **Monitoramento:** implemente o monitoramento para capturar métricas de desempenho em todo o fluxo de trabalho.
5. **Análise:** Analise as métricas para identificar gargalos e áreas de melhoria.

E quais métricas devo verificar?

- **Request:** A partir do número de request conseguimos estipular o quão nossos serviços conseguem suportar sem perder a performance com inúmeros acessos simultâneos. Contudo, metrificando sozinho não garante a eficiência do serviço.
- **Response time:** o tempo de resposta é uma medida da qualidade do serviço que o sistema está fornecendo aos usuários. Quanto menor o tempo de resposta, melhor a experiência do usuário.
- **Erros:** Monitorar os erros durante os testes de desempenho ajuda a identificar problemas no sistema, como falhas de rede, erros de codificação, problemas de banco de dados, entre outros. Isso permite que a equipe de desenvolvimento resolva esses problemas antes que afetem os usuários reais.

Ferramentas do mercado;

- JMeter;
- Grafana K6;
- Gatling;
- Locust;
- Vegeta.



Conhecendo o K6



- Criado para ser fácil utilização para todos os membros do time;
- Desenvolvido em GO Lang;
- Baseado em código e não no-code, low-code;
- Scripts em Javascript ou Typescript;
- Para ser executado em qualquer lugar;
- Foco em teste de performance e API;
- Feito para ser integrado e extensível;
- Mantido pela Grafana Labs com possibilidade de utilização K6 Cloud;

Instalação

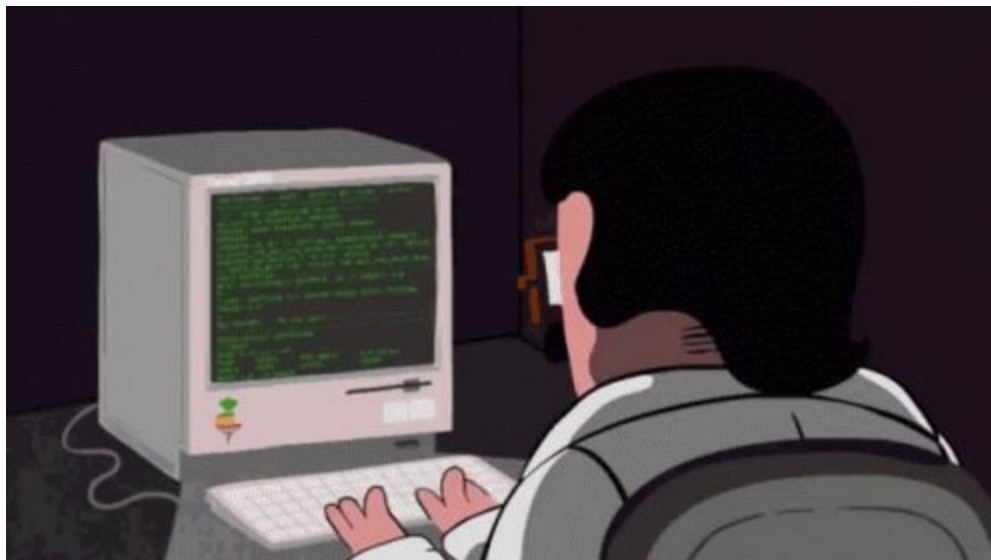
- Debian/Ubuntu
- Mac OS
 - brew install k6
- Windows
 - [the latest official installer](#)
- Utilizável em docker

```
sudo gpg -k
sudo gpg --no-default-keyring --keyring /usr/share/keyrings/k6-archive-keyring.gpg --keyserver hkps://keyserver.ubuntu.com:4001
echo "deb [signed-by=/usr/share/keyrings/k6-archive-keyring.gpg] https://dl.k6.io/deb stable main" | sudo tee /etc/apt/sources.list.d/k6.list
sudo apt-get update
sudo apt-get install k6
```

Conhecendo a documentação e API do K6

- HTTP Requests;
- Metrics;
- Checks;
- Thresholds;
- Options e Configuration;
- Test lifecycle.

Prática K6 para Teste de Performance



<https://k6.io/website-stress-testing/>

<http://ecommerce.test.k6.io/>

Prática de K6 para teste de API

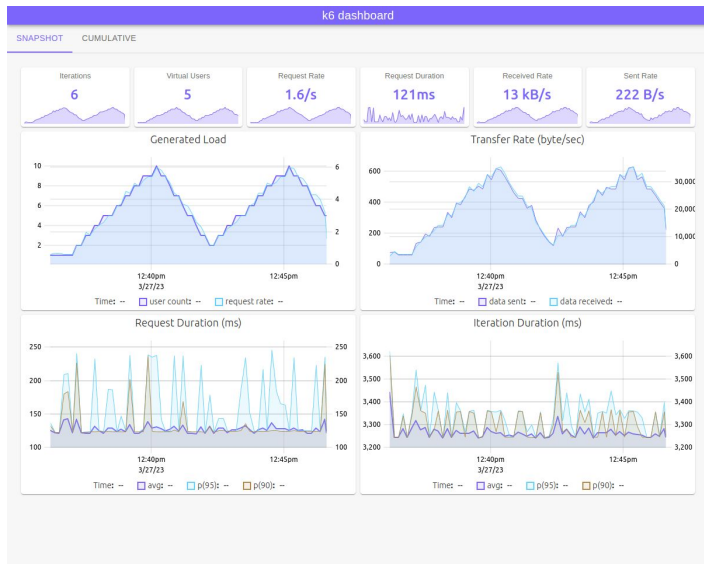


<https://serverest.dev/#/>

Plus → Extensões

- Por ser open source há muitas extensões consolidadas, entre elas:

→ Xk6-dashboard



→ Xk6-kubernetes

```
import { Kubernetes } from 'k6/x/kubernetes';

const manifest = `
apiVersion: batch/v1
kind: Job
metadata:
  name: busybox
  namespace: testns
spec:
  template:
    spec:
      containers:
        - name: busybox
          image: busybox
          command: ["sleep", "300"]
          restartPolicy: Never
`;

export default function () {
  const kubernetes = new Kubernetes();

  kubernetes.apply(manifest);

  const jobs = kubernetes.list("Job", "testns");

  console.log(`${jobs.length} Jobs found:`);
  pods.map(function(job) {
    console.log(` ${job.metadata.name}`);
  });
}
```

Dúvidas?



joaomartinsfilho.substack.com