JMeter, Prometheus & Grafana

Gustavo Gomides

CE-237

Sumário

- Introdução
- Objetivos
- Motivação
- Teste Performance x Carga x Estresse
- JMeter
- Prometheus
- Grafana
- Exemplo Prático
- Conclusões e Recomendações

Introdução

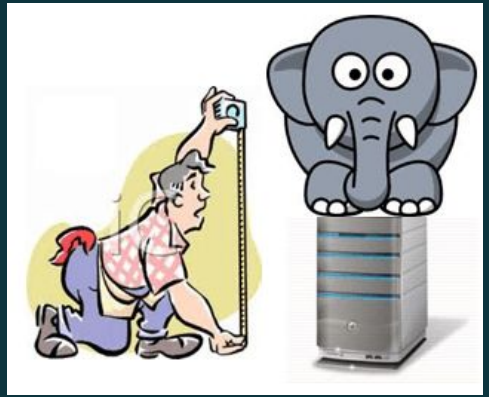

Figura 1: Teste de Performance

Fonte: https://guru99.com

Introdução

- Crescente aumento no uso de testes automatizados
- Enfoque na realização de testes durante todas as fases do projeto
- Testes que validem a estabilidade e robustez da arquitetura projetada

Objetivos

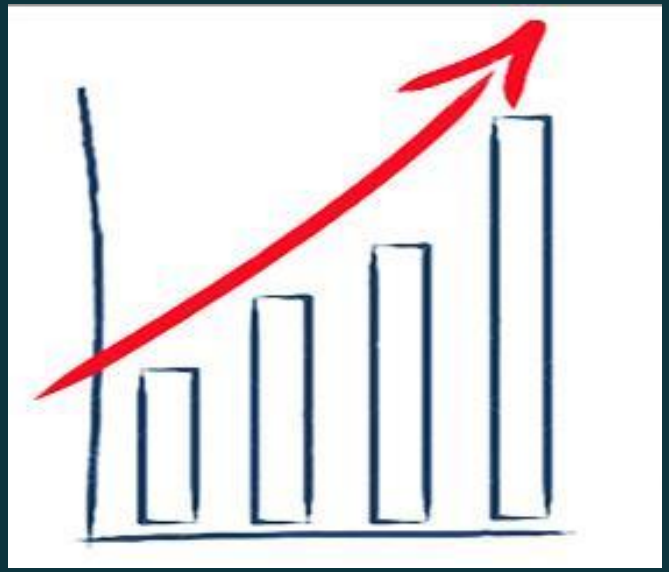

Figura 2: Detalhamento do objetivo

Fonte: http://old.secovi.com.br

Objetivos

- Entender a diferença entre Teste de Performance x Carga x Estresse
- Conhecer as ferramentas JMeter, Prometheus e Grafana
- Desenvolver exemplo prático integrando as três ferramentas

Motivação

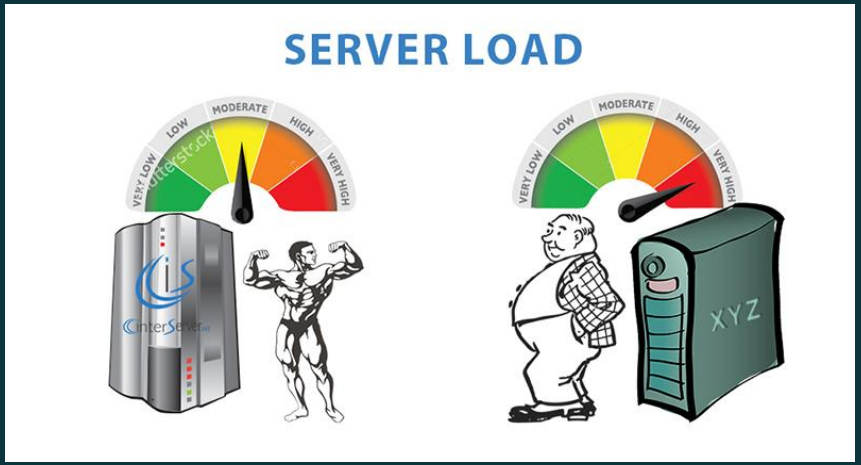

Figura 3: Server Load
Fonte: https://www.interserver.net

“Aquilo que não se pode medir, não se pode melhorar” *Lord Kelvin*

- Quantos usuários simultâneos sua aplicação suporta?
- Quantas requisições por segundo sua aplicação suporta?
- Como monitorar sua aplicação e servidores em tempo real?

Teste Performance x Carga x Estresse

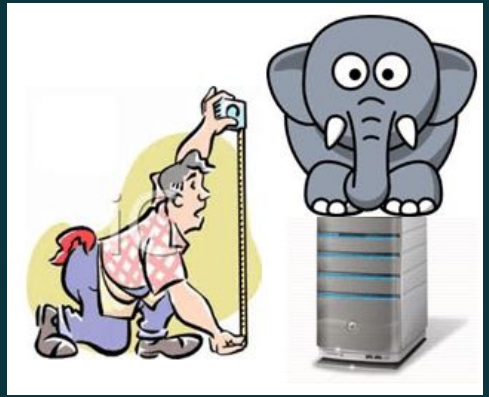

Figura 4: Teste de Performance

Fonte: https://guru99.com

Teste de Performance

- Identificar gargalos e mensurar se a arquitetura projetada suportará a aplicação

- Necessário definir o que é cenário excelente, aceitável e ruim
  - Quanto de memória poderá ser consumido pelo servidor?

- Executar teste -> mensurar performance -> incrementar performance até atingir o nível de performance desejado

Teste de Carga

- Processo de executar a aplicação sob testes, incrementando as tarefas de maior carga

- Tem como objetivo evitar riscos não detectados em testes superficiais, como estouro da memória

- Verificar que a aplicação atenderá os níveis definidos no teste de performance

Teste de Estresse

- Derrubar a aplicação que está sendo testada, consumindo todos os recursos

- Verificar se a aplicação consegue se restabelecer após as quedas

- Observar o comportamento da aplicação em cenários adversos

JMeter

Figura 5: Apache JMeter

Fonte: https://jmeter.apache.org/

JMeter

- Ferramenta que possibilita a realização de teste de performance, carga e estresse, e outros tipos de teste
- Parte do projeto Jakarta, da Apache Software Foundation
- 100% desenvolvida em Java e open-source
- Primeira versão lançada em 1998

JMeter

- **Number of Threads (users):** número de threads que serão criadas

- **Ramp-up Period:** número de segundos que o JMeter levará para criar todas as threads configuradas
  - 18 requisições por segundo:
    - number of threads: 360
    - ramp-up period: 18

- **Loop Count:** número de vezes que o teste será repetido

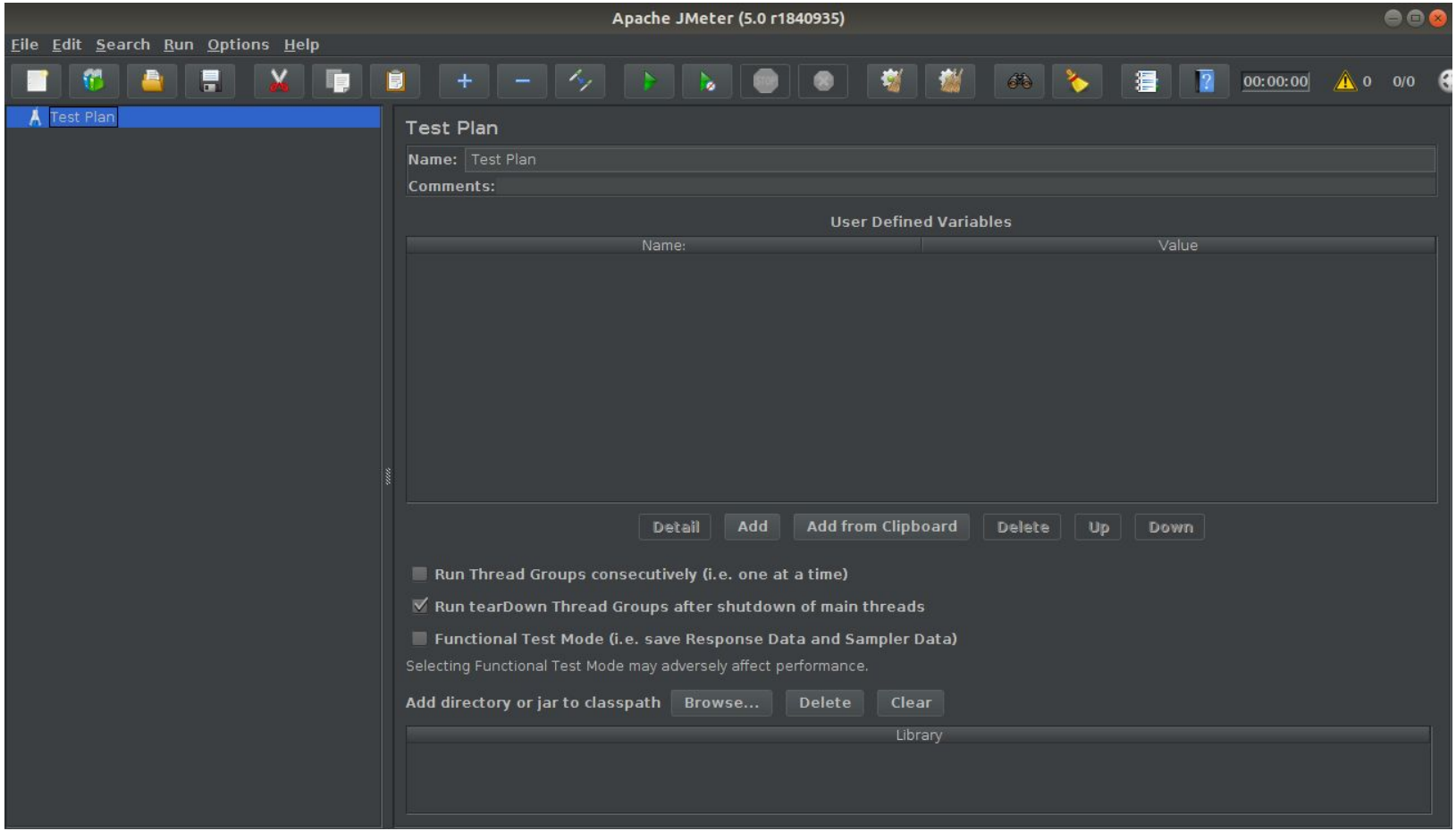

Figura 6: Tela Inicial Apache JMeter

Fonte: https://medium.com

Prometheus

Figura 7: Prometheus

Fonte: https://prometheus.io

Prometheus

- Sistema de monitoramento para serviços e aplicações
- Coleta as métricas dos hosts periodicamente -> avalia as regras criadas anteriormente -> exibe os resultados -> emite alertas se necessário
- Não necessita de armazenamento externo
- Primeira versão lançada em 2012

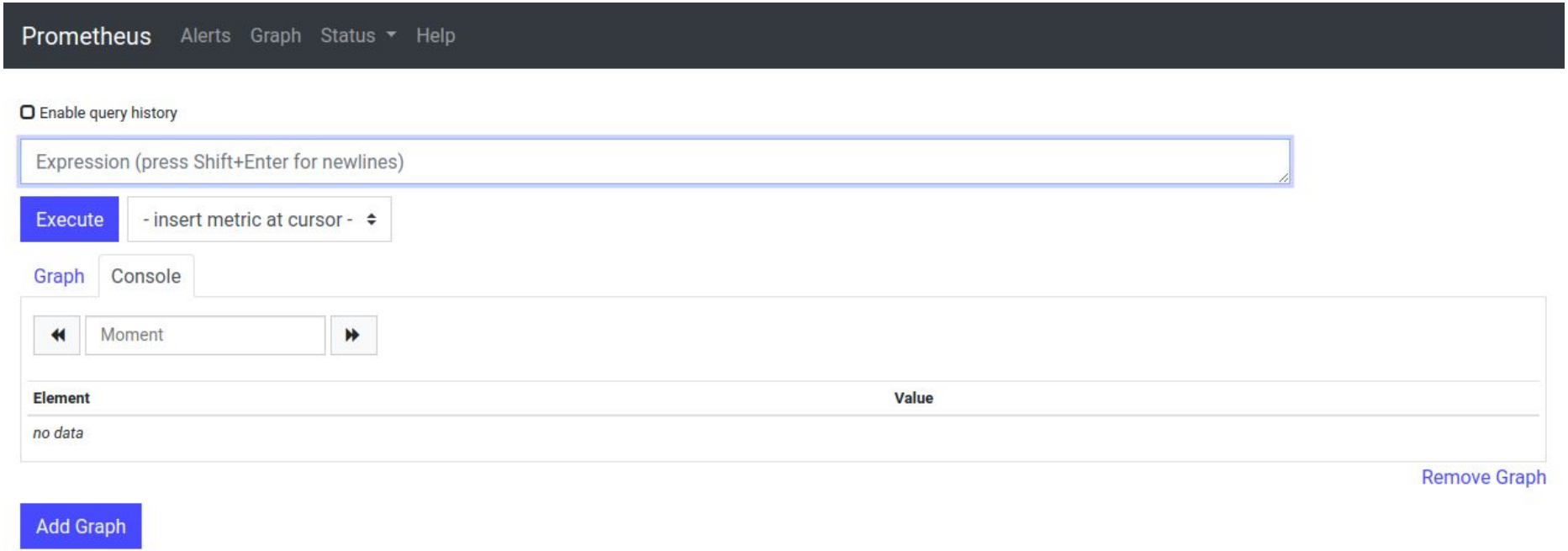

Figura 8: Tela Inicial Prometheus

Fonte: próprio autor

Grafana

Figura 9: Grafana

Fonte: https://grafana.com

Grafana

- Plataforma para visualizar e analisar métricas por meio de gráficos
- Suporte para diferentes tipos de bancos de dados
- Permite criar dashboards dinâmicos
- Primeira versão lançada em 2014

Figura 10: Dashboard Grafana

Fonte: https://grafana.com

Exemplo Prático

Figura 11: Docker

Fonte: https://docker.com

Aplicação

- Multiplicação de matrizes
- API rest desenvolvida em python
- Input de entrada é o tamanho da matriz

Metodologia

- Monitoramento da máquina e containers utilizando prometheus e grafana

- Teste de performance utilizando JMeter
  - 6 requisições por segundo com matriz de dimensão 100
    - Cenário excelente: utilizar menos que 70% CPU
    - Cenário aceitável: utilizar entre 70 e 80% CPU
    - Cenário ruim: utilizar > 80% CPU

- Deploy do prometheus, grafana e aplicação utilizando docker

Conclusões e recomendações

Recapitulando

- Teste de Performance x Carga x Estresse
- JMeter, Prometheus e Grafana
- Exemplo prático integrando as três ferramentas

Conclusões e Recomendações

- Pôde-se verificar as diferenças entre teste de performance, carga e estresse

- Foi apresentado um exemplo prático envolvendo as ferramentas JMeter, Prometheus e Grafana, utilizando docker

- O monitoramento do host antes e durante a realização da bateria de teste é de extrema importância

- Recomenda-se a execução do teste de performance no STAGIOP-TR

Referências

- Grafana: https://grafana.com/
- JMeter: https://jmeter.apache.org/
- Prometheus: https://prometheus.io/
- How To Use Apache JMeter To Perform Load Testing on a Web Server. Disponível em:<https://www.digitalocean.com/community/tutorials/how-to-use-apache-jmeter-to-perform-load-testing-on-a-web-server>. Acesso em: 01 out. 2019
- NUNES, Rafael. Sobrecarregando sua aplicação com JMeter. Disponível em: <http://www.univale.com.br/unisite/mundo-j/artigos/23Jmeter.pdf>. Acesso em: 30 set. 2019

Dúvidas?

Gustavo Gomides
gustavo.gomides7@gmail.com