Testes de Escalabilidade de Serviços Web

Diego de Araújo Martinez Camarinha
Prof. Fabio Kon
Paulo Bittencourt Moura (Doutorando)

11 de novembro de 2013

Serviços Web
O que são e por que usar?

- Componentes de software distribuídos
- Visam a promover comunicação interoperável entre sistemas
- Ex: Google Drive

Escalabilidade

Uma das muitas definições:

- Uma aplicação é escalável se atinge o mesmo desempenho quando a capacidade da infraestrutura aumenta na mesma proporção que o tamanho do problema[1]

Escalabilidade também leva em conta a infraestrutura onde o software está rodando!!

Testes de Escalabilidade

Normalmente difíceis de realizar:

- Infraestrutura
- Carga
- Emulação de clientes

Além disso, quais métricas escolher?

Ferramenta para os testes
Scalability Explorer[2]

Arcabouço de software livre para testes de escalabilidade que:

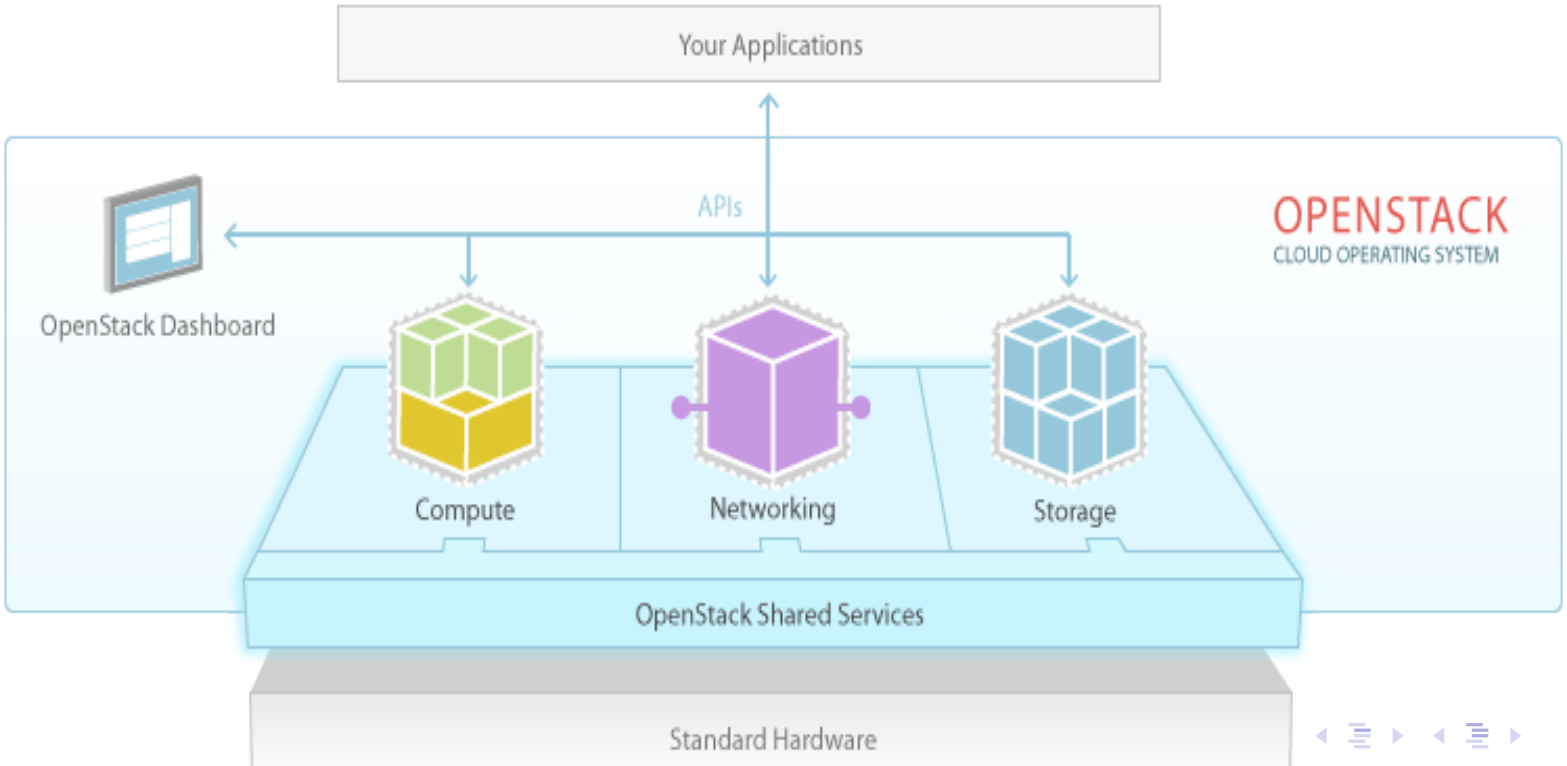
- Possibilita coletar três tipos de métricas
  - **Degradação**
  - ***Speedup***
  - **Performance agregada**
- Fornece suporte à emulação de clientes (Rehearsal[3])
- Consegue lidar com aumentos na carga de requisições

Porém...

E a infraestrutura???

OpenStack

- Sistema operacional de nuvem
- Gerencia grandes quantidades de recursos em uma nuvem
- *Dashboard* que facilita seu uso
- OpenStack do CCSL oferecia algumas configurações de máquinas virtuais



Escolha do Serviço
Kalibro Web Service

Motivos:

- Mezuro[4] - plataforma para monitoramento de código-fonte
- Kalibro[5] - software que realiza coleta e análise de métricas de código-fonte
- Com o Mezuro prestes a entrar em produção, é preciso que algumas perguntas sejam respondidas:
  - Como o Kalibro se comportará quando forem feitas muitas requisições simultâneas?
  - Quais são os seus limitantes? Se houverem, como melhorá-los?

Escolha do Serviço
Kalibro Web Service

- C, C++, JAVA e python
- Faz uso de coletores auxiliares (Analizo, Checkstyle e CVSAnaly)
- Utiliza configurações existentes ou personalizadas
- Fornece uma interpretação padrão para os resultados obtidos

Preparação para os testes

- Instalação do Scalability Explorer
- Entendimento do funcionamento do arcabouço
- Escrita dos testes
- Execução, primeiramente, no localhost
- Depois, preparação das máquinas virtuais no OpenStack...

Preparação para os testes
Problemas...

- Criação das máquinas virtuais manualmente
- Processo demorado
- Instabilidade do OpenStack
- Instalação de dependências e do serviço uma máquina virtual por vez...

Preparação para os testes
Problemas...

Uma...

Preparação para os testes
Problemas...

Por...

Preparação para os testes
Problemas...

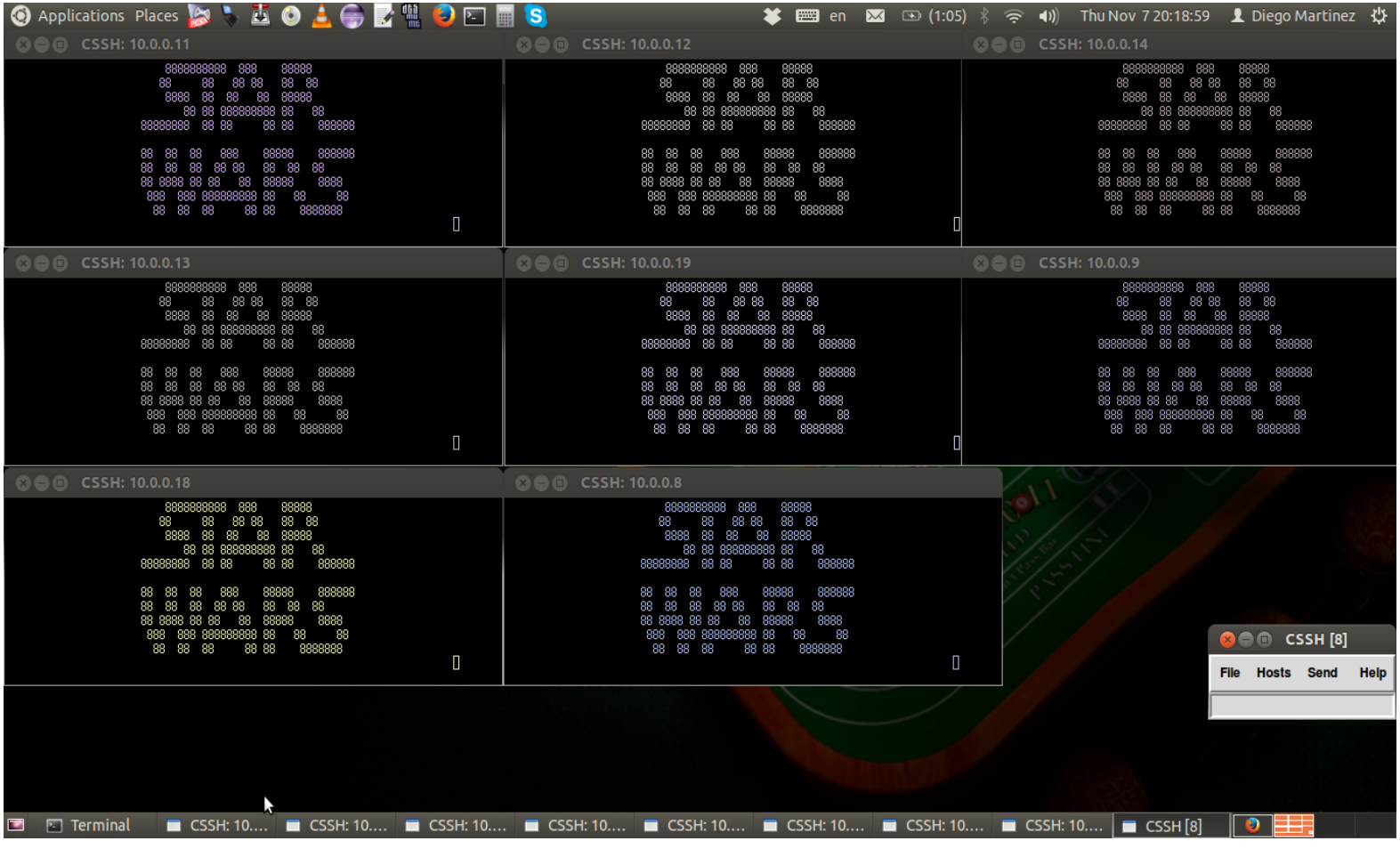
Vez...

Preparação para os testes
Solução!

ClusterSSH

- Controla diversos terminais através de um único console
- Permite enviar comandos simultaneamente a diversos servidores através de uma conexão ssh
- Realmente me salvou muitas horas de trabalho!

Preparação para os testes
Solução!



Bom uso do ClusterSSH.

Preparação para os testes
Configuração das máquinas virtuais

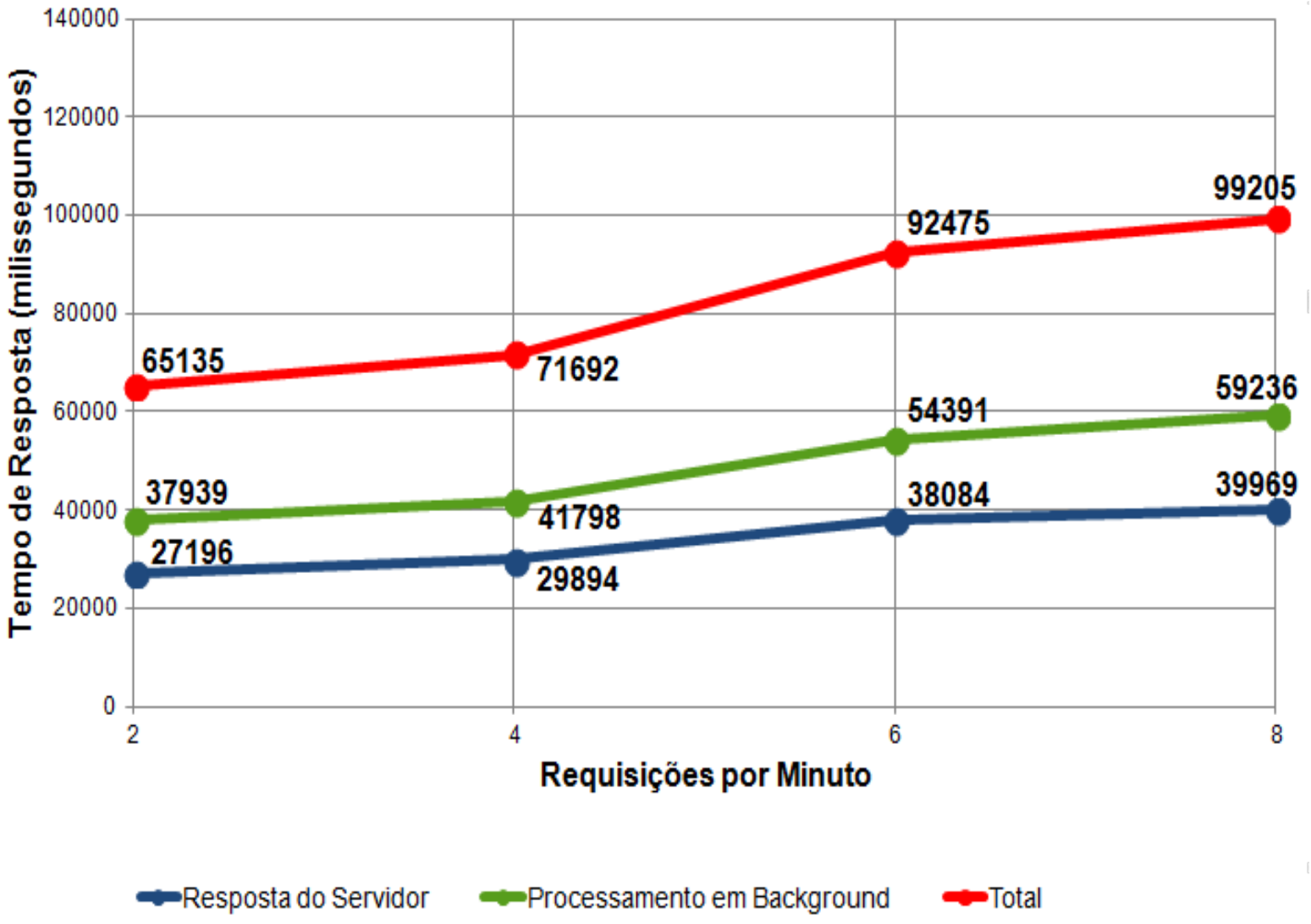
Tamanho	nº de VCPUs[1]	RAM	HD
Médio	2	4	40
Grande	4	8	80
XGrande	8	16	160
XXGRANDE	12	24	6

[1] Cada VCPU era de 2.4 GHz

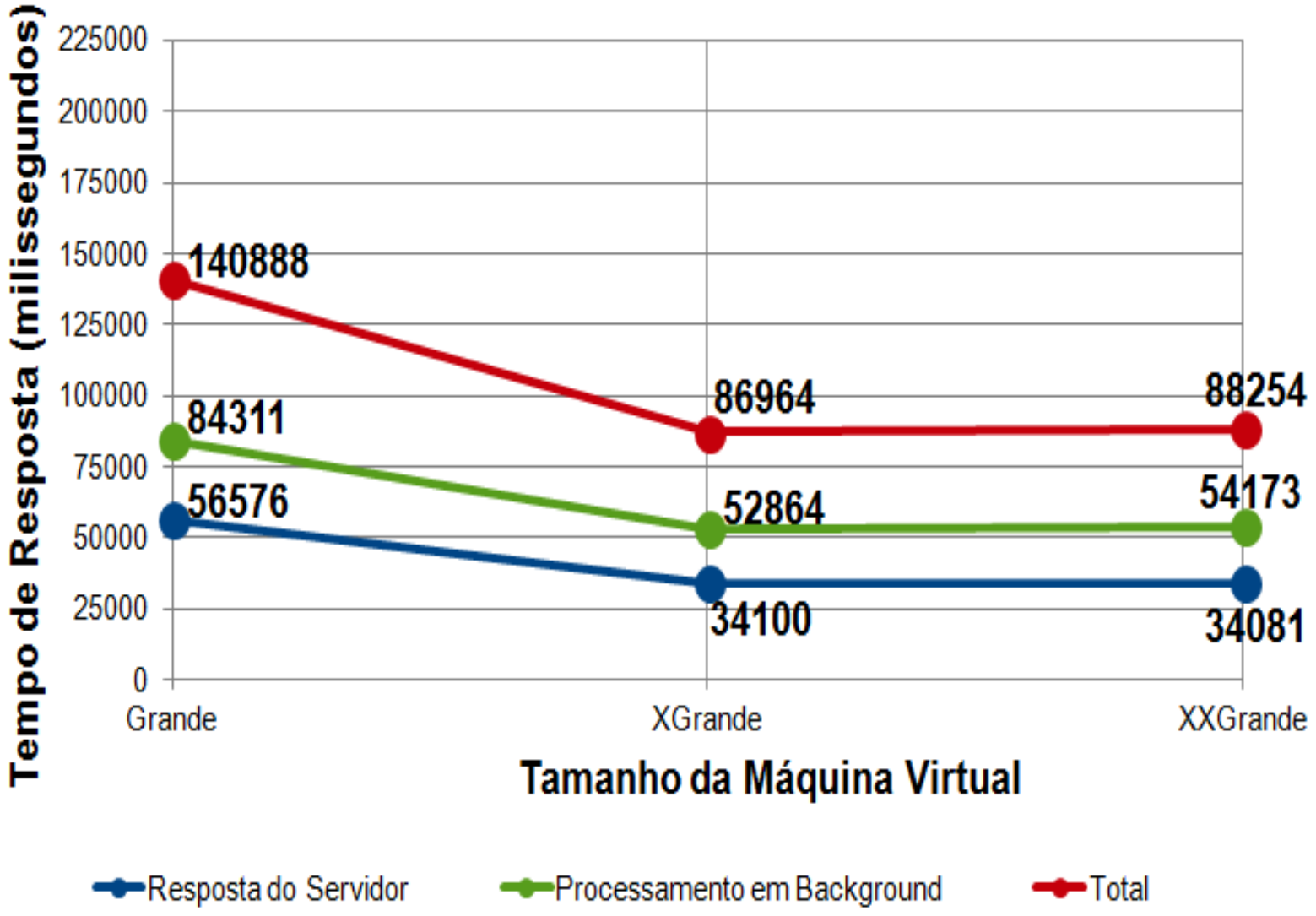
Identificação da operação mais problemática
Processamento de repositório de código-fonte

- Kalibro possui 50 operações
- Ao todo, 150 testes
- Dentre as operações problemáticas, escolhemos tentar melhorar a de processamento de repositório de código-fonte

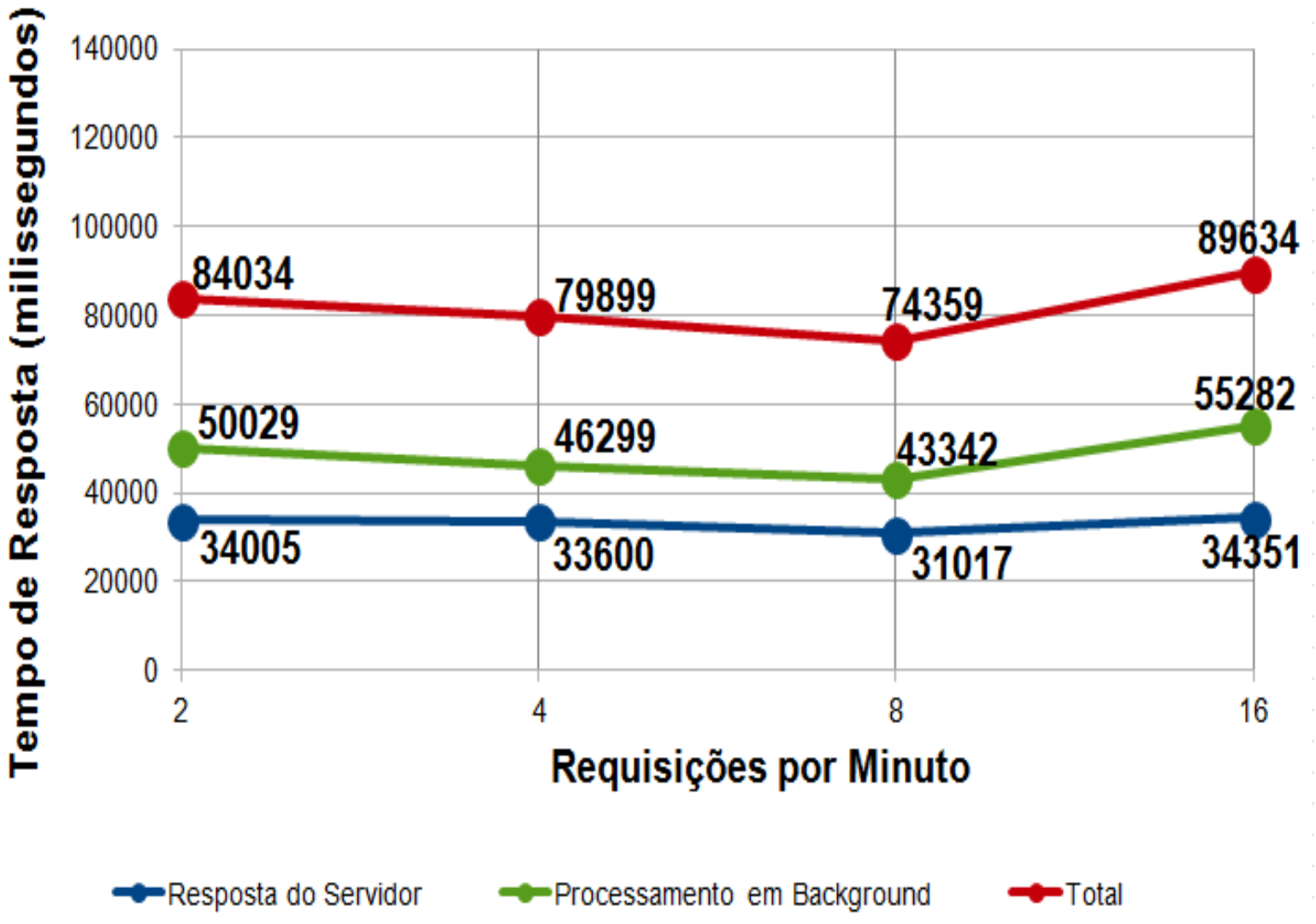
Teste de degradação



Teste de speedup

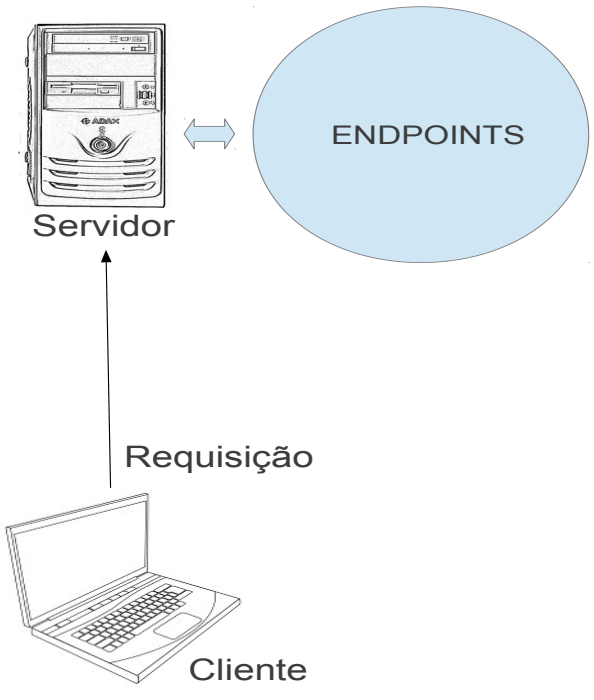


Teste de performance agregada

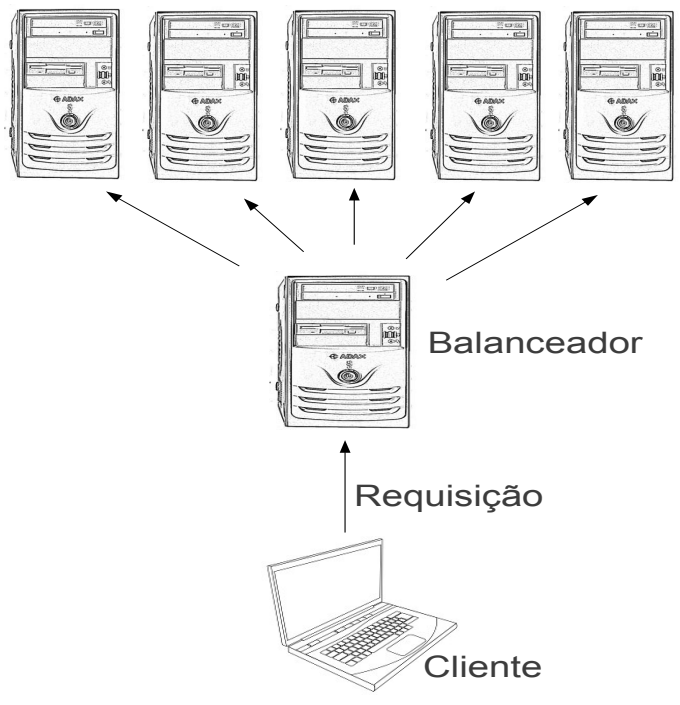


Primeira solução
Balanceamento de carga

Antes:

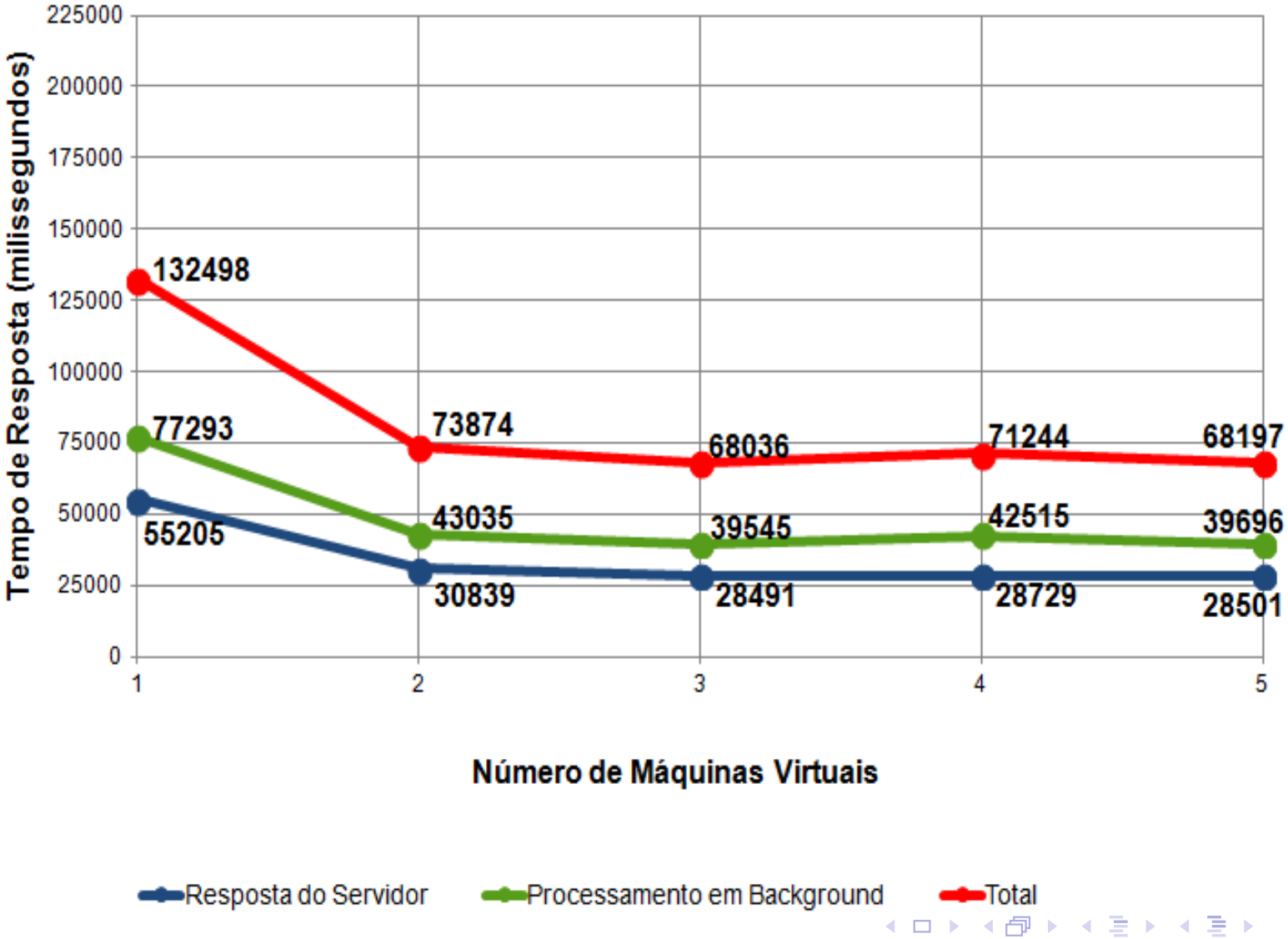


Depois:



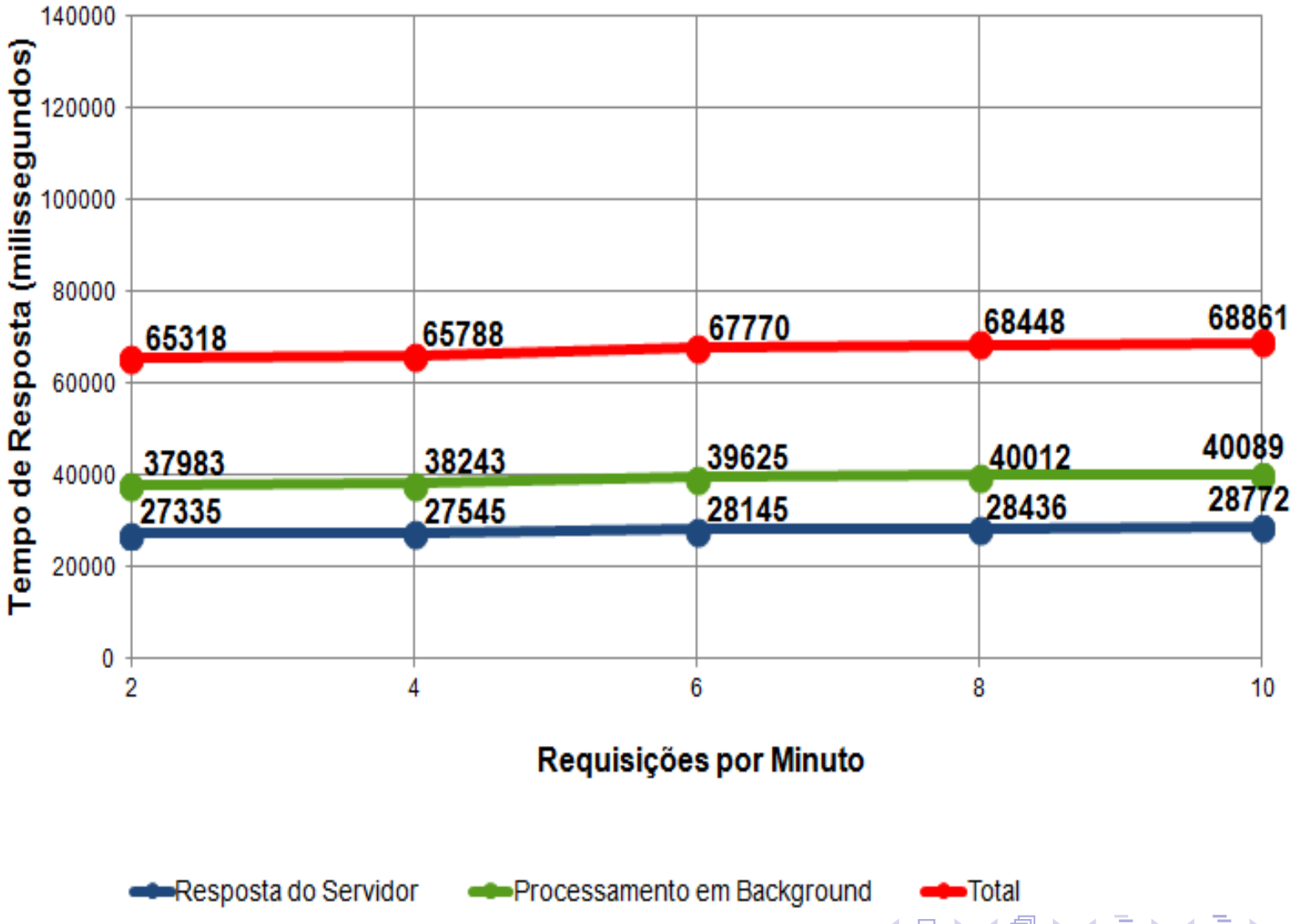
Primeira solução

Teste de speedup com balanceamento de carga



Primeira solução

Teste de performance agregada com balanceamento de carga

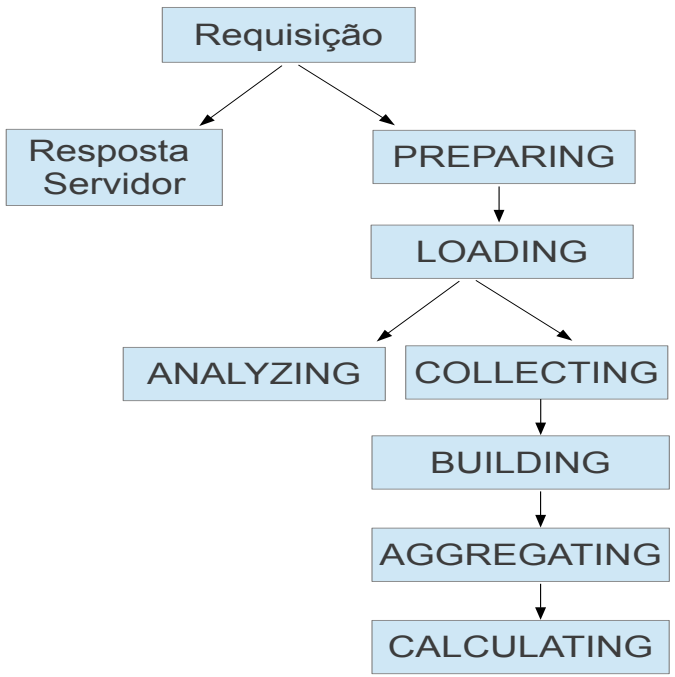


Segunda solução
Mudança na infraestrutura do Kalibro

Antes:



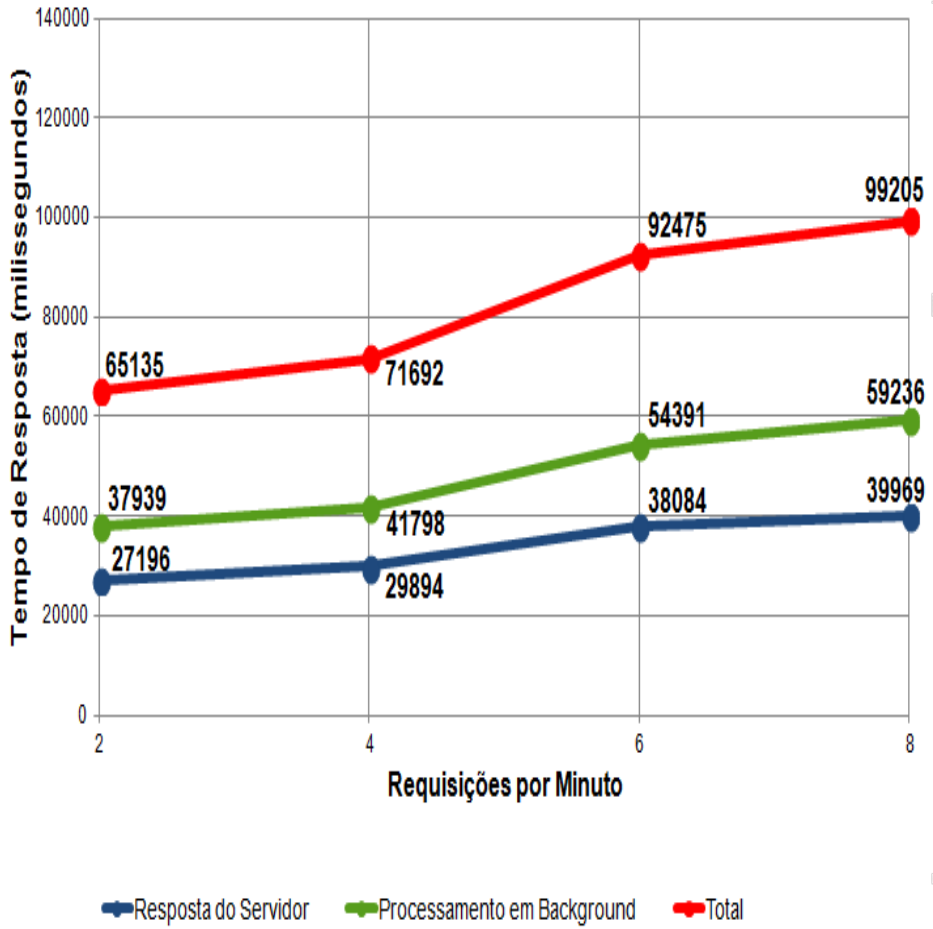
Depois:



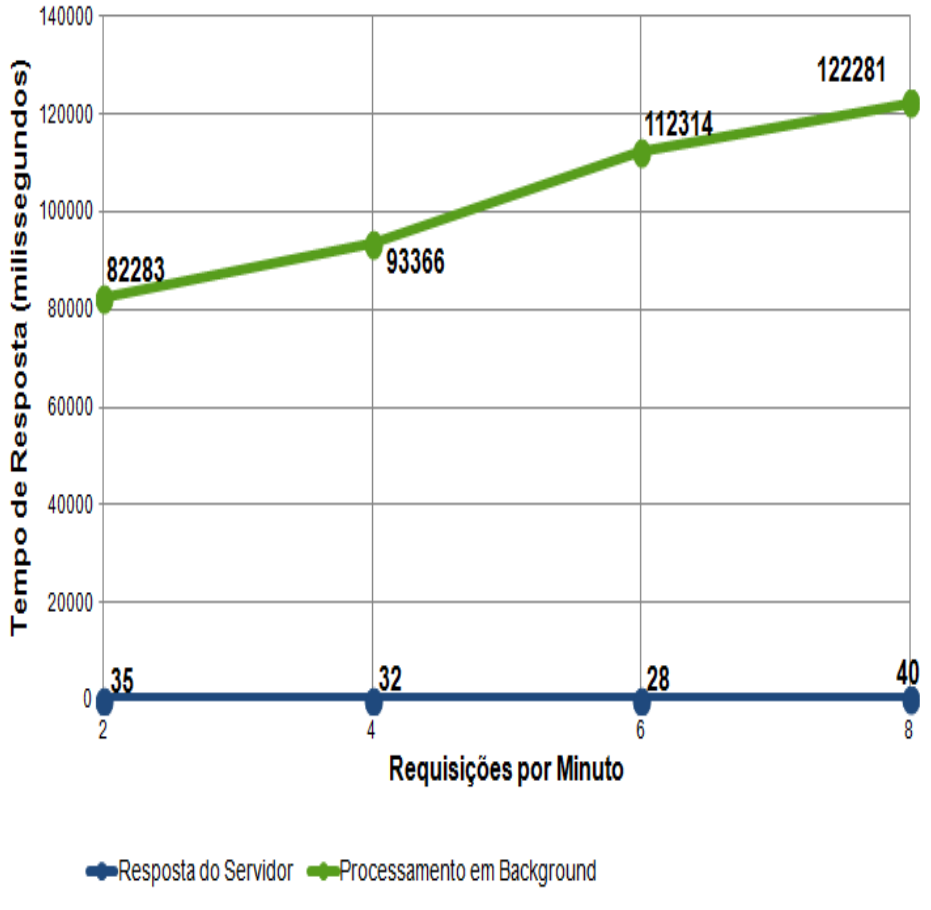
Segunda solução

Teste de degradação com a infraestrutura nova

Antes:



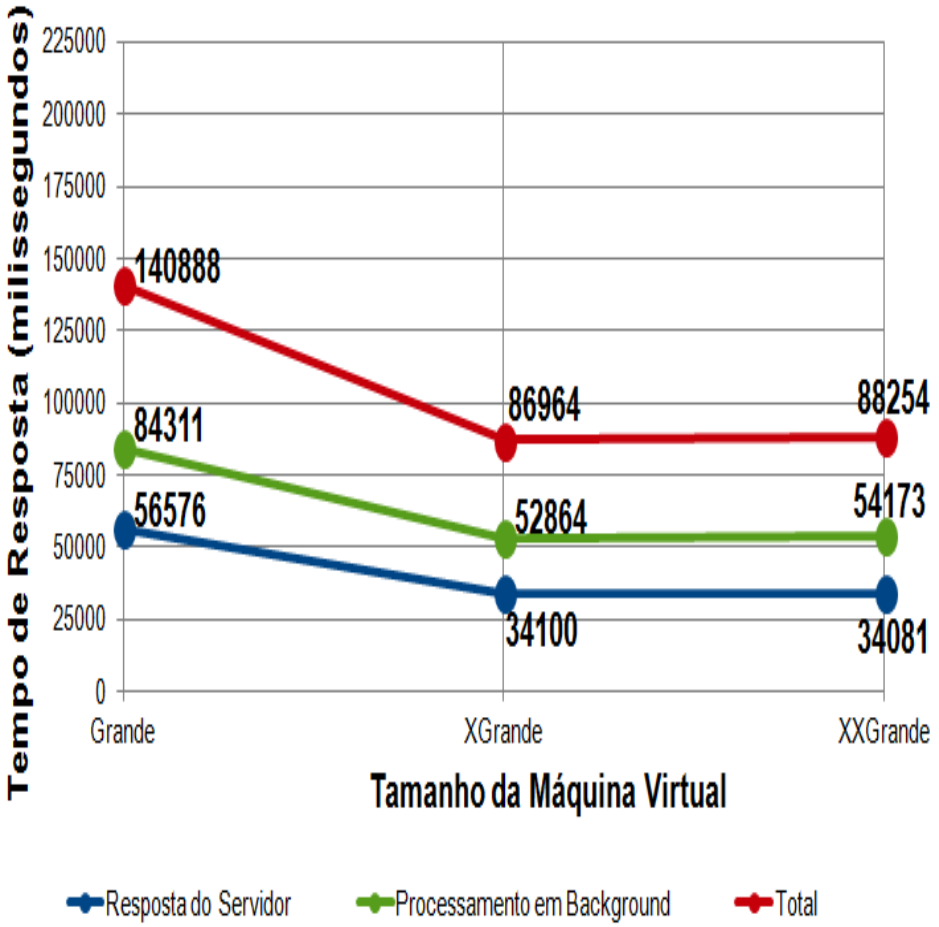
Depois:



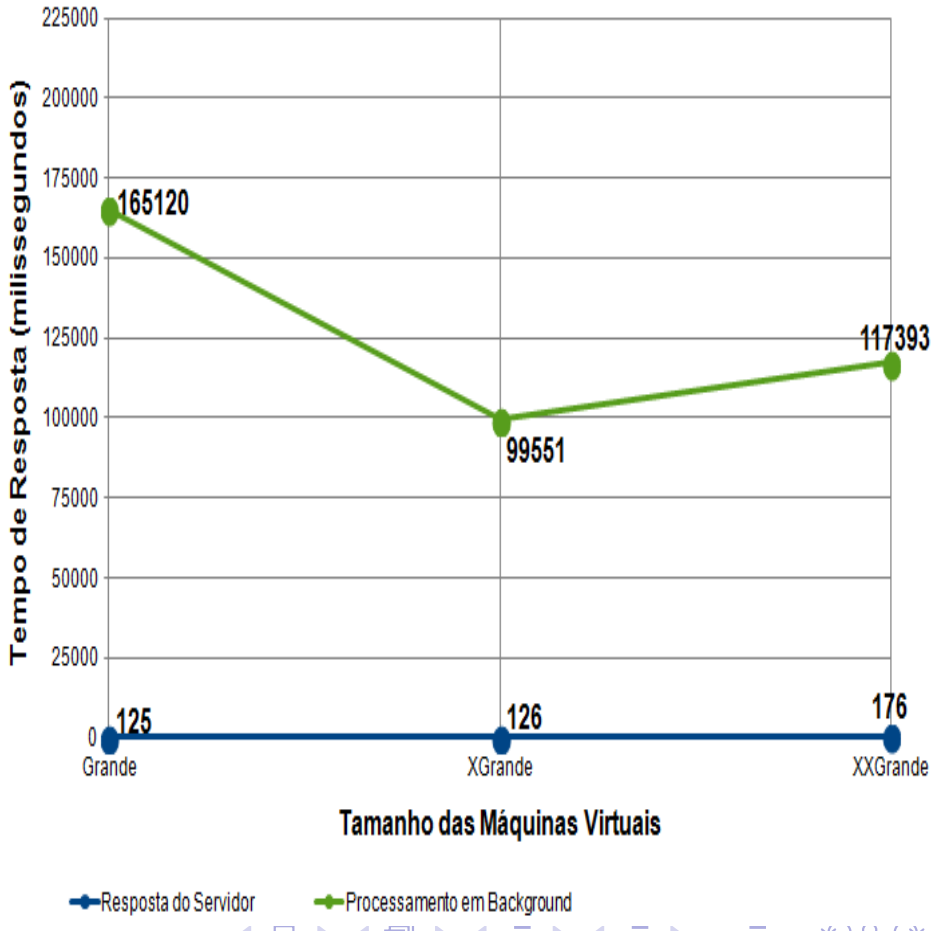
Segunda solução

Teste de speedup com a infraestrutura nova

Antes:



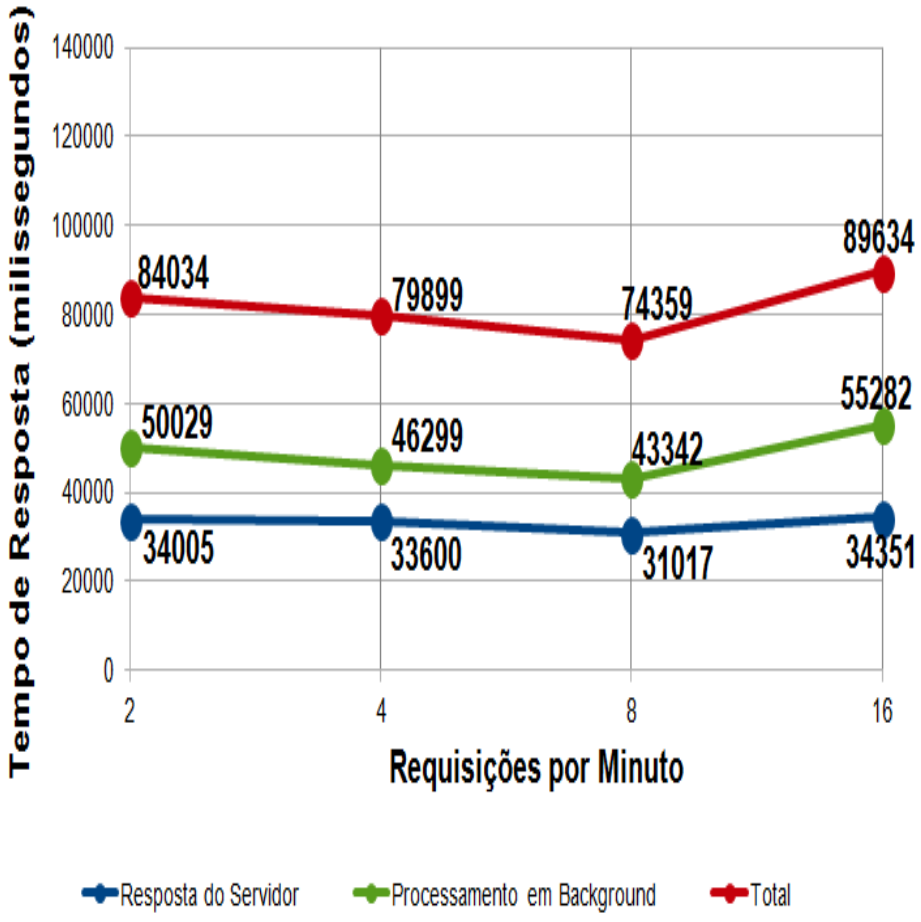
Depois:



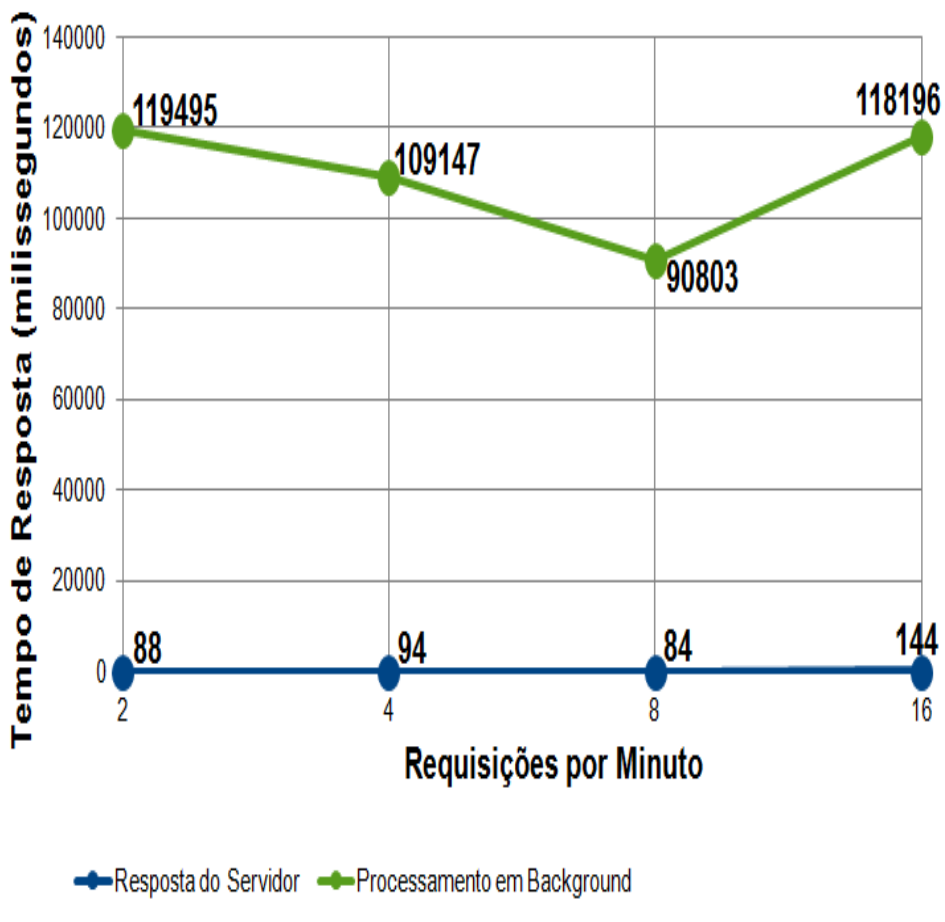
Segunda solução

Teste de performance agregada com a infraestrutura nova

Antes:



Depois:



Conclusões
Scalability Explorer

- Manipulação de clientes
- Escrita dos testes
- Várias opções de parâmetro
- Paulo Moura

Conclusões
Balanceamento de carga

Vantagens:

- Estabilidade do sistema em horários de picos de requisição
- Elasticidade da nuvem

Desvantagens:

- Infraestrutura cara
- Mais difícil de manter

Conclusões
Mudança na infraestrutura

Vantagens:

- *Feedback* muito rápido
- Criação de mais estados permite melhor identificação de erros

Desvantagens:

- O tempo total de execução piorou
- Mais complexidade no Kalibro

Conclusões

Portanto, para melhorar a escalabilidade do Kalibro:

- Combinar as duas soluções
- Dessa forma, o *feedback* é rápido e, em horários de pico de requisição, é possível manter o tempo de resposta quase inalterado

Agradecimentos







Perguntas?

Referências bibliográficas

[1] M. Quinn. Parallel Computing: Theory and Practice. McGraw-Hill, 2nd edição, 1994.

[2] Moura and Kon, "Automated Scalability Testing of Software as a Service", *8th International Workshop on Automation of Software Test*, 2013.

[3] Besson et al., "Rehearsal: Um arcabouço para teste automatizado de coreografias de serviços web". Disseração de mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo, 2012.

[4] Meirelles et al., "Mezuro Plataform: Source Code Tracking Network", *Sessão de Ferramentas do III Congresso Brasileiro de Software: Teoria e Prática*, 2012.

[5] Morais et al., "Kalibro: Interpretação de métricas de código-fonte". Dissertação de mestrado, Instituto de Matemática e Estatística, Univesidade de São Paulo, 2013.