

# INSTITUTO DE COMPUTAÇÃO

## UNIVERSIDADE ESTADUAL DE CAMPINAS

### **MC437 - Grupo06 - Relatório 2**

*Gabriel Oliveira      João Fidélis      Lucas Moraes*  
*Matheus Figueiredo      Pedro Grijó*

Technical Report - IC-16-02 - Relatório Técnico

April - 2016 - Abril

The contents of this report are the sole responsibility of the authors.  
O conteúdo do presente relatório é de única responsabilidade dos autores.

# MC437 - Grupo06

Gabriel Bueno de Oliveira      João Guilherme Daros Fidélis  
Lucas Henrique Moraes      Matheus Yokoyama Figueiredo  
Pedro Rodrigues Grijó

## Resumo

Utilizamos o benchmark TPC-W, que modela uma livraria online, através de um ambiente controlado, para simular atividades num servidor WEB. Em conjunto com o simulador RBE, que gera três diferentes perfis de carga (Shopping, Ordering e Browsing), pudemos checar o desempenho do servidor instalado num cluster no IC, ao vermos o número de WIPS (WEB Interactions per Second) gerados por diferentes cargas. O relatório refere-se a segunda parte do projeto da disciplina de MC437 (Projeto de Sistemas de Informação) e tem como objetivo realizar testes com dois bancos de dados diferentes, sendo um primário que é replicado para um secundário em hot-standby, testando quais as melhores cargas que o primário consegue atender e depois medir o tempo de recuperação quando o primário é desligado e o secundário promovido.

## 1 Introdução

Este trabalho é um relatório da segunda parte do projeto da disciplina MC437 - Projeto de Sistemas de Informação. Essa parte foi constituída pela preparação de uma máquina remota com um servidor webApache Tomcat, utilização de duas máquinas remotas com instâncias de bancos de dados PostgreSQL, uma com um banco atuando como primário e outra com um banco secundário em hot-standby, utilizado para integrar todas essas funcionalidades e fazer um site de compras com dados gerados aleatoriamente.

Também foi instalado na máquina do web server o aplicativo TPC-W, um benchmark de transações web. Para fazer uso do TPC-W, foi utilizado o RBE (Remote Browser Emulator), que emula conjuntos de clientes que acessam o lado servidor do TPC-W, sendo que este implementa uma livraria digital.

O RBE é um simulador escrito completamente em Java que simula o tráfego HTTP gerado por um usuário que estivesse acessando o site através de um navegador.

Também utilizamos o HAProxy, que atua como um proxy para aplicações baseadas em TCP e HTTP. Ele oferece alta disponibilidade e balanceamento de carga para servidores web. Neste trabalho, sua função é de detectar falha do banco primário e redirecionar as requisições feitas para o banco secundário, que deve ser promovido a banco primário.

O TPC-W gera um número, o WIPS (Web Interactions per Second). O fluxo de trabalho é gerado pelo RBE e pode ser de três perfis. O perfil de compras (*shopping*), onde 80% das ações são de leitura e 20% de escrita no banco de dados. O perfil de navegação (*browsing*),

tem 95% das ações de leitura e 5% de escrita. Já o perfil de compras (*ordering*) tem 50% de suas operações de leitura e 50% de escrita.

Rodamos três experimentos diferentes, o primeiro apenas com o banco primário ativo, o segundo com o primário ligado e o secundário ligado em hot-standby e o terceiro onde na metade do experimento, o primário era desativado para que o secundário pudesse ser promovido.

## 2 Condições Experimentais

Nesta seção serão descritas as configurações de hardware e software utilizadas nos experimentos.

Nesta fase, contamos com quatro máquinas, todas iguais e fornecidas pelo Instituto de computação.

A primeira, chamada CBN6, é onde está instalado o HAProxy X.XX, Tomcat versão 7 e TPC-W. Ela que atua como servidor web.

Duas máquinas, a dbmaster2 e dbslave2, é onde ficam os banco de dados primário e secundário, respectivamente. Também tem instalados o PostgreSQL versão 9.5.1.

Já o RBE foi rodado a partir de outra máquina, a CBN 7.

Todas as máquinas tem a mesma configuração: sistema operacional Ubuntu 14.04, CPU Intel(R) Core(TM)2 Quad CPU Q8400 2.66GHz e memória RAM de 4GB e 1333 MHz.

## 3 Metodologia de Pesquisa

Foi necessário executar três diferentes experimentos. Cada experimento consiste em rodar o RBE para simular uma carga no servidor e avaliarmos as medições feitas.

O primeiro experimento, foi utilizar o RBE com apenas o banco primário ligado. No segundo experimento, estavam ligados os bancos primários e o secundário em hot-standby.

Esses dois experimentos, são feitos para caracterizar nosso servidor. O objetivo é medir a maior carga para a qual ele se comporta de forma consistente.

Para isso, variamos os valores de carga entre 2000 e 4000 clientes, variando de 1000 em cada iteração, isso para cada perfil de usuário descrito anteriormente. Assim, podemos fazer gráficos que nos mostram até qual carga o servidor manteve um bom nível de WIPS ao longo de todo o teste (que dura 100s no total).

Com esse valor achado, utilizamos esse valor de carga e utilizamos o RBE para simular essa "carga ótima" no servidor, causando um failover manualmente (isto é, matando o banco de dados primário), enquanto promovemos o banco de dados secundário, para ver em quanto tempo nosso servidor consegue se recuperar e voltar a ficar ativo.

É importante ressaltar que sempre rodamos o RBE de outra máquina remota (CBN7), diferente de nosso servidor, e que os parâmetros utilizados foram Ramp-Up Time: 5s, Ramp-Down Time: 5s e Measurement Interval: 90s. O valor de máximo número de erros tolerados foi colocado em 0. Todos os outros valores são os padrões do RBE.

## 4 Análise e Resultados

Seguem os gráficos gerados pela execução do RBE para cada experimento.

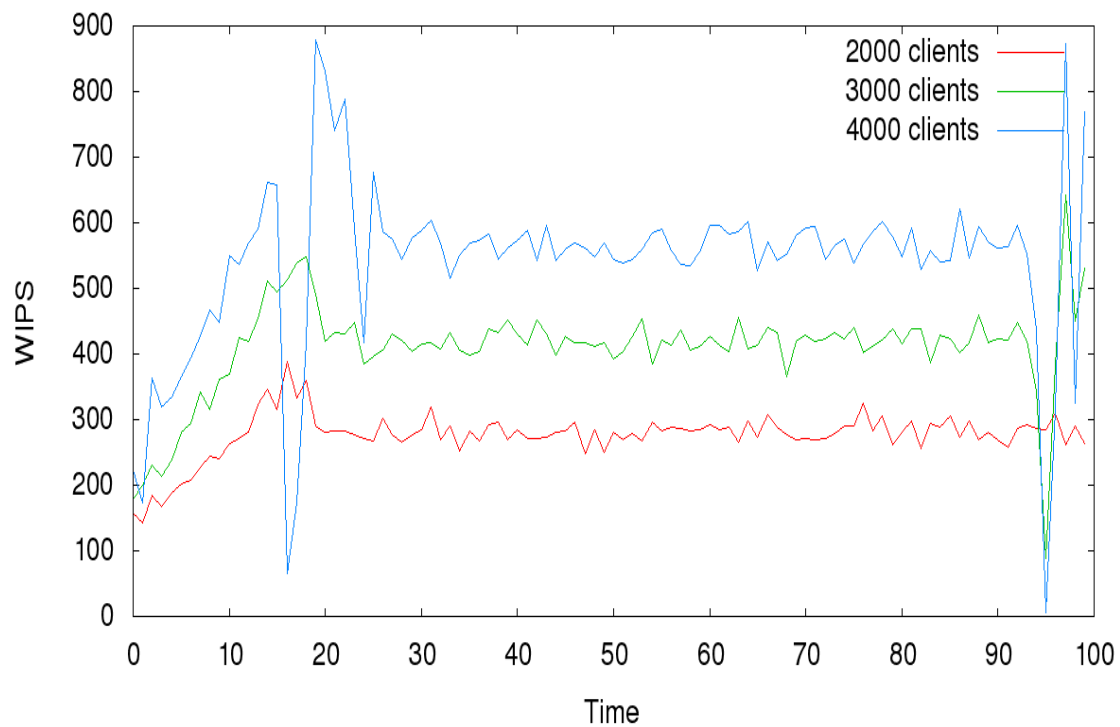


Figura 1: WIPs por tempo para cada carga simulada pelo RBE no perfil Browsing no experimento 1

Vemos que 4000 usuários teve uma boa consistência durante o meio do experimento mas uma grande queda no final, assim como o de 3000. O de 4000 também houve uma variação grande por volta dos 15 segundos.

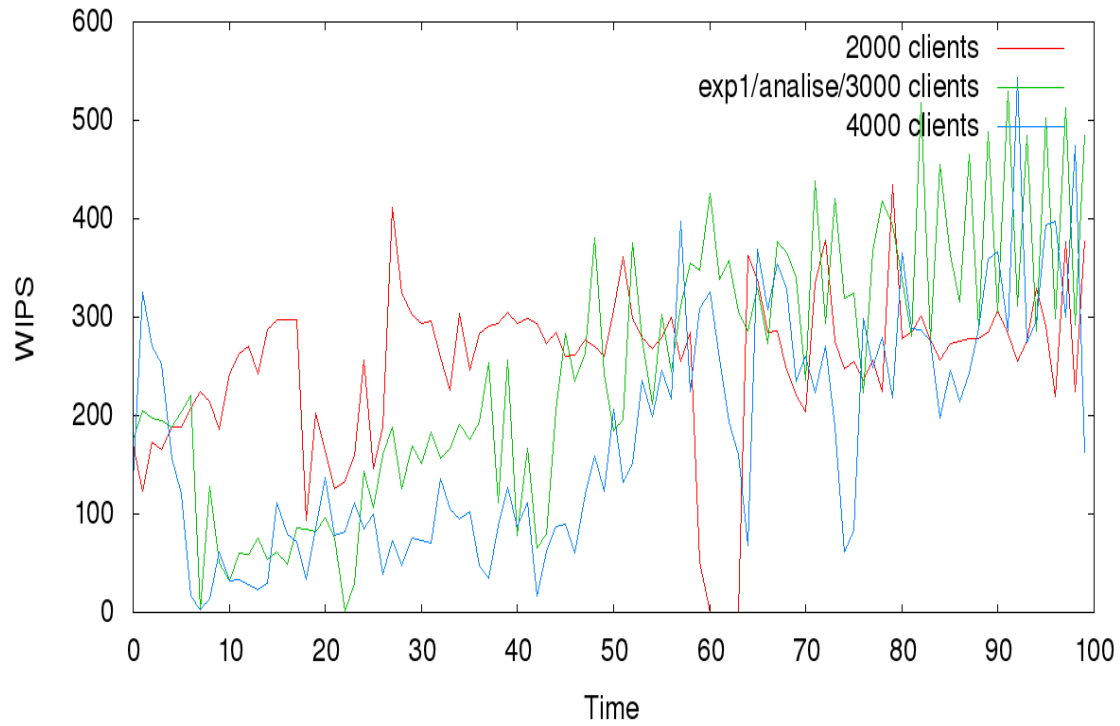


Figura 2: WIPSo por tempo para cada carga simulada pelo RBE no perfil Ordering no experimento 1

Pelo gráfico, podemos ver que o teste com 3000 clientes teve o melhor desempenho no geral. Até a metade, 2000 usuários estava melhor mas depois 3000 foi crescendo e manteve uma boa taxa até o final.

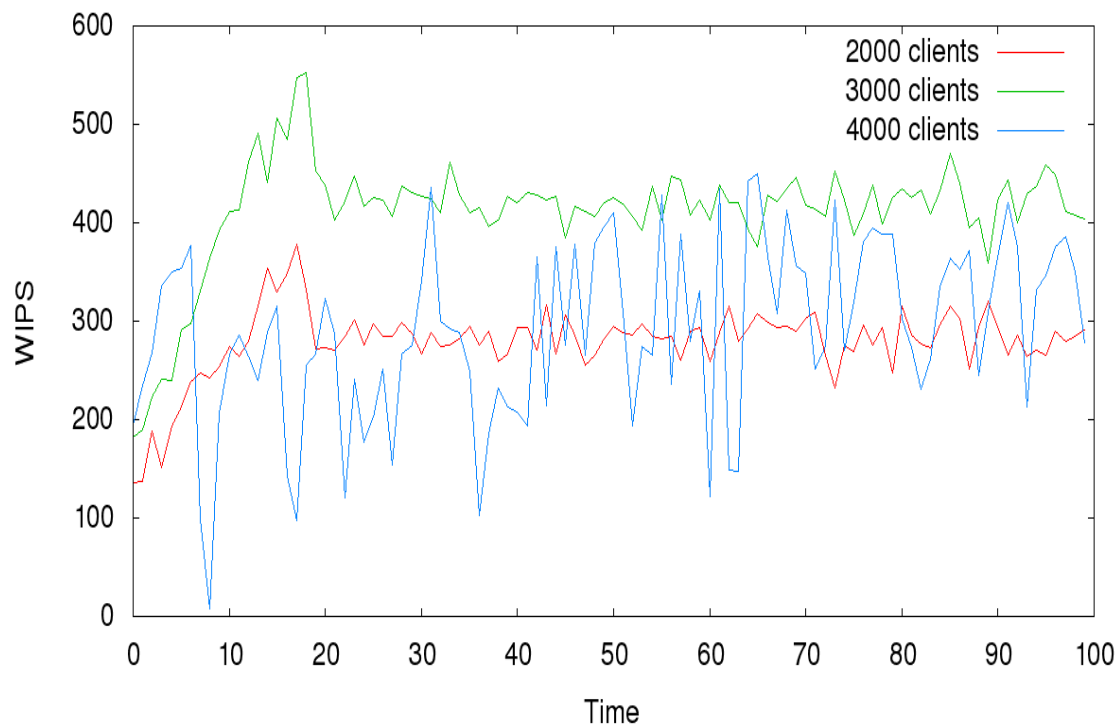


Figura 3: WIPS por tempo para cada carga simulada pelo RBE no perfil Shopping no experimento 1

Neste gráfico, ficou evidente o melhor desempenho da simulação com 3000 clientes sobre as duas outras. Uma melhor taxa de WIPS foi obtida durante praticamente todo o experimento.

Esses três gráficos combinados, nos dá indícios de que 3000 usuários talvez seja o melhor número a ser utilizado, pois tem taxas de WIPS mais altas e estáveis que as outras testadas.

Agora, analisaremos os dados do experimento 2.

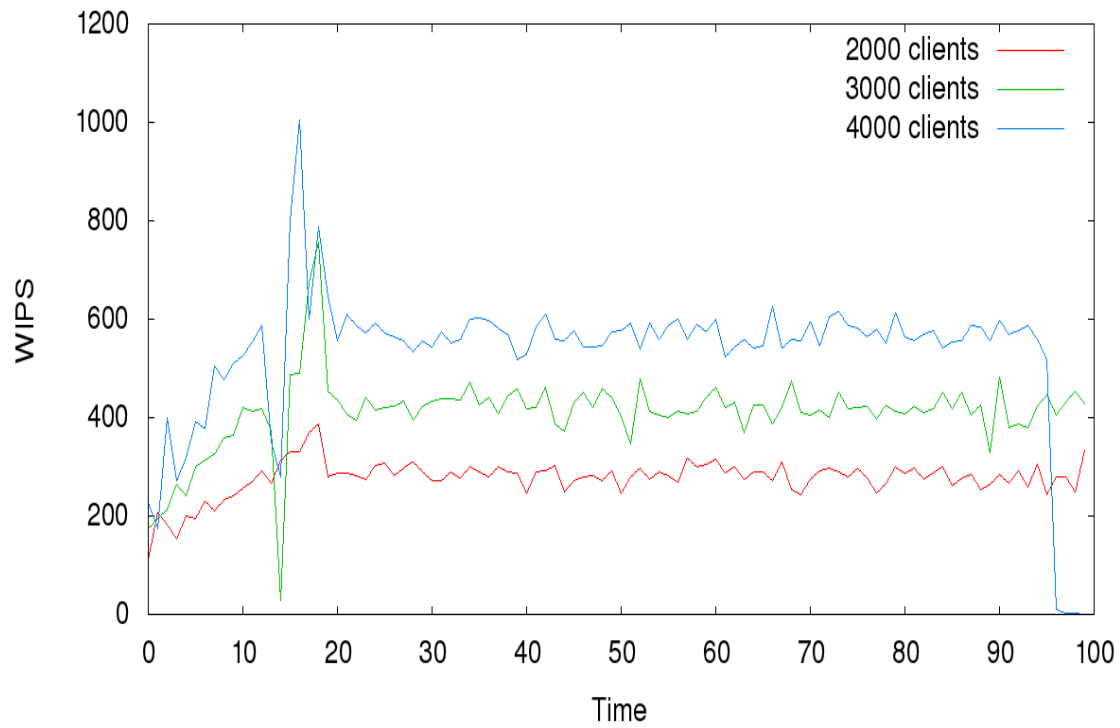


Figura 4: WIPs por tempo para cada carga simulada pelo RBE no perfil Browsing no experimento 2

O gráfico indica que um melhor desempenho no geral para 4000 clientes, porém, há uma grande queda no final, que pode indicar problemas devido a alta carga.

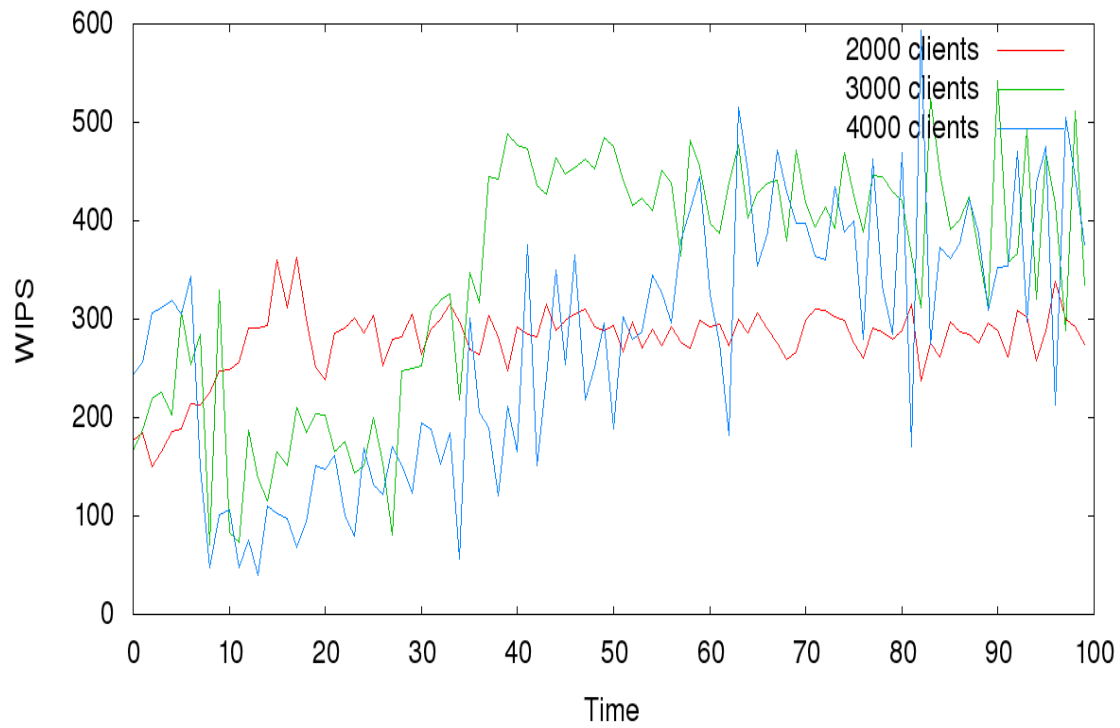


Figura 5: WIPSo por tempo para cada carga simulada pelo RBE no perfil Ordering no experimento 2

Vemos um melhor desempenho de 3000 clientes no geral, que manteve uma boa média durante todo o experimento e se manteve no mesmo nível de WIPs para 4000 clientes na segunda metade do experimento.



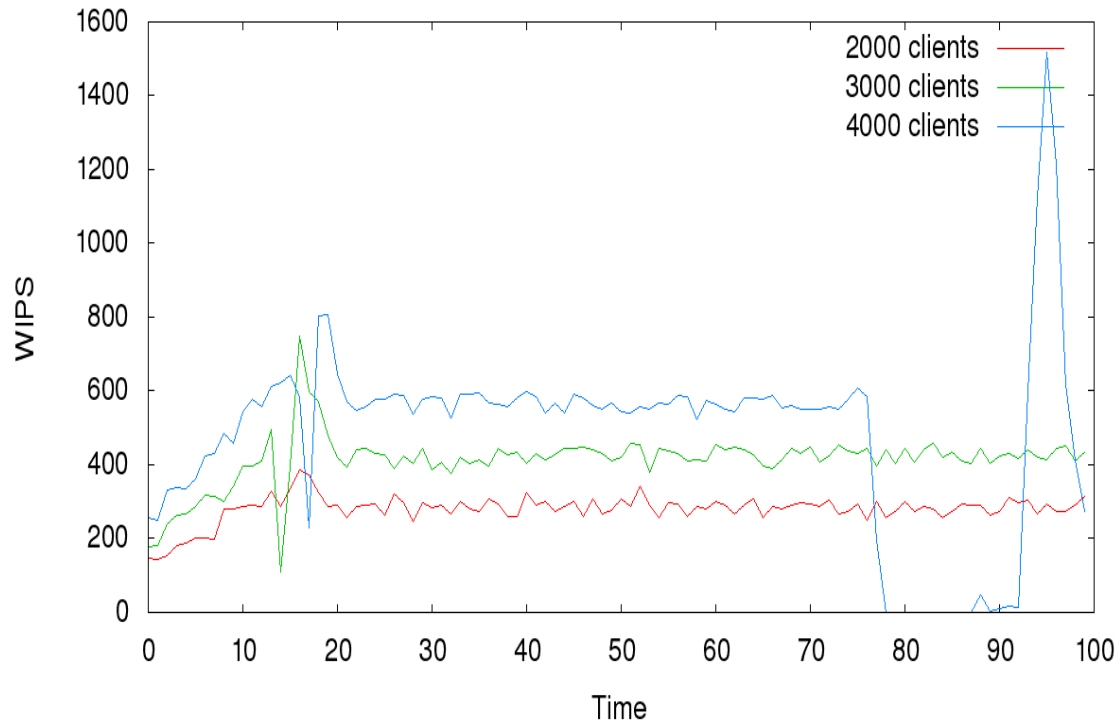


Figura 6: WIPS por tempo para cada carga simulada pelo RBE no perfil Shopping no experimento 2

Podemos ver que 4000 clientes mantiveram uma taxa de WIPS melhor durante boa parte do experimento, mas existiu uma grande queda dos WIPS na segunda metade, assim como no perfil de Browsing, o que pode indicar que essa carga já está alta e sobrecarregando o servidor.

## 5 Conclusão