

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 do trabalho contamos com quatro máquinas, todas com a mesma configuração de hardware, fornecidas pelo Instituto de Computação.

A primeira, denominada CBN6, é onde estão instalados o HAProxy 1.6, Apache Tomcat versão 7 e TPC-W. A máquina CBN6 atua como servidor web.

Duas máquinas, a dbmaster2 e dbslave2, é onde ficam os bancos de dados primário e secundário, respectivamente. Nelas está instalado o PostgreSQL versão 9.5.1.

Já o RBE foi rodado a partir de outra máquina, a CBN7.

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

Foram executados três diferentes experimentos. Cada experimento consistiu em rodar o RBE para simular uma carga no servidor, e após isso foram avaliadas as medições realizadas pelo RBE.

No primeiro experimento, o RBE foi utilizado com apenas o banco primário ligado. No segundo experimento, estavam ligados o banco primário e o banco secundário em hot-standby.

Esses dois experimentos foram feitos para caracterizar o servidor. O objetivo foi 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 1000 para cada iteração para cada perfil de usuário descrito anteriormente. Assim, gráficos foram gerados para demonstrar até qual carga o servidor se manteve num bom nível de WIPS ao longo de todo o teste (que dura 100s no total).

O valor encontrado foi utilizado no RBE para simular essa "carga ótima" no servidor, causando um failover manualmente (isto é, matando o banco de dados primário), enquanto o banco de dados secundário era promovido, para ver em quanto tempo o servidor conseguia se recuperar e voltar a ficar ativo.

É importante ressaltar que o RBE sempre foi RBE de outra máquina remota (CBN7), diferente de da máquina do servidor (CBN6), 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 o experimento 1.

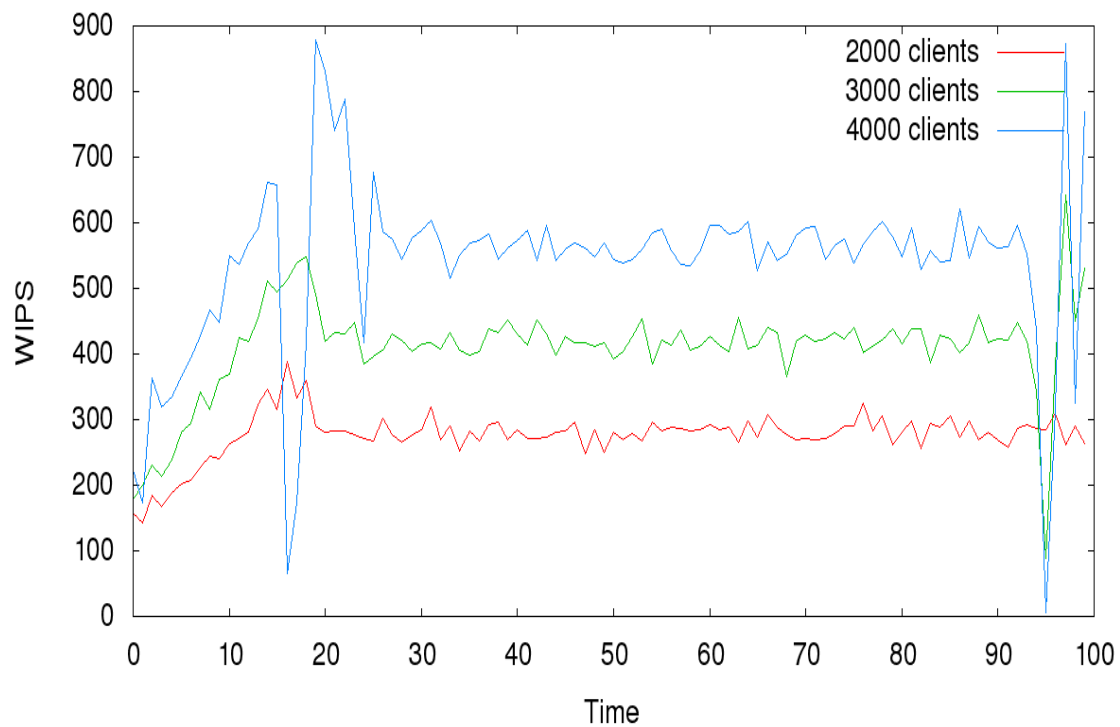


Figura 1: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Browsing do experimento 1

Nota-se que o experimento com 4000 clientes teve uma boa consistência entre 30 e 90 segundos, porém houve uma grande queda por volta dos 15s e a partir dos 90s. O mesmo ocorreu no caso para 3000 clientes a partir de 90s.

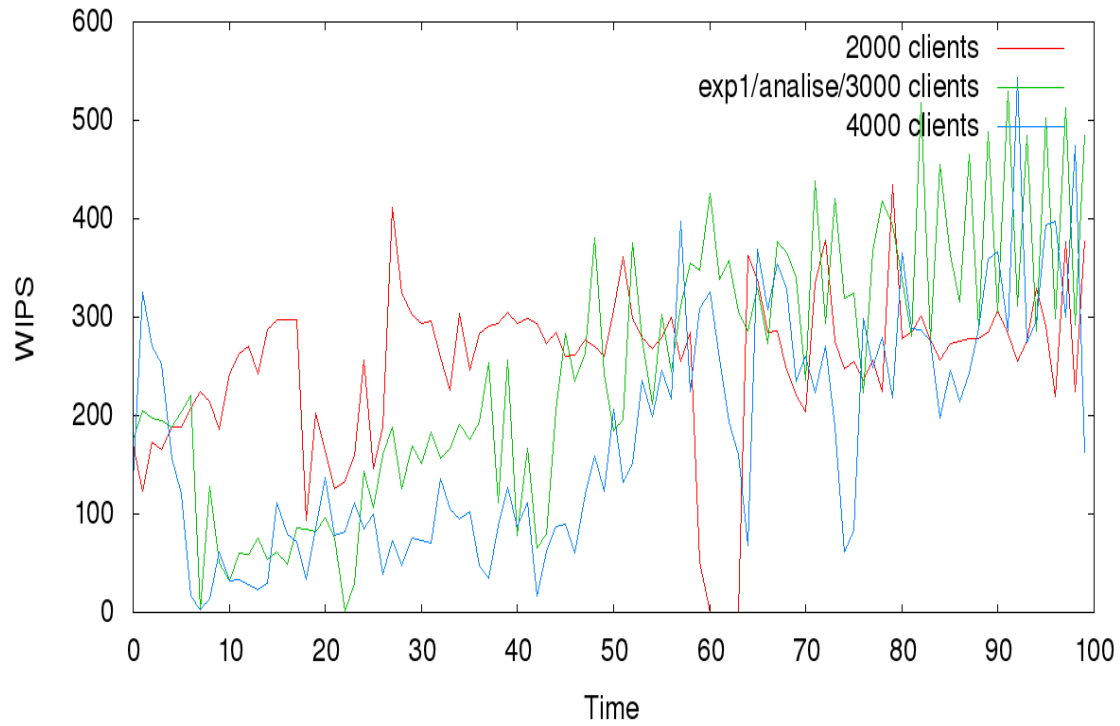


Figura 2: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Ordering do experimento 1

Pelo gráfico, é visível que o teste com 3000 clientes obteve o melhor desempenho no geral. Até 50s, o desempenho era melhor com 2000 clientes, porém após esse tempo o teste com 3000 clientes obteve uma quantidade maior de WIPS e se manteve consistentemente melhor que os testes com 2000 e 4000 clientes.

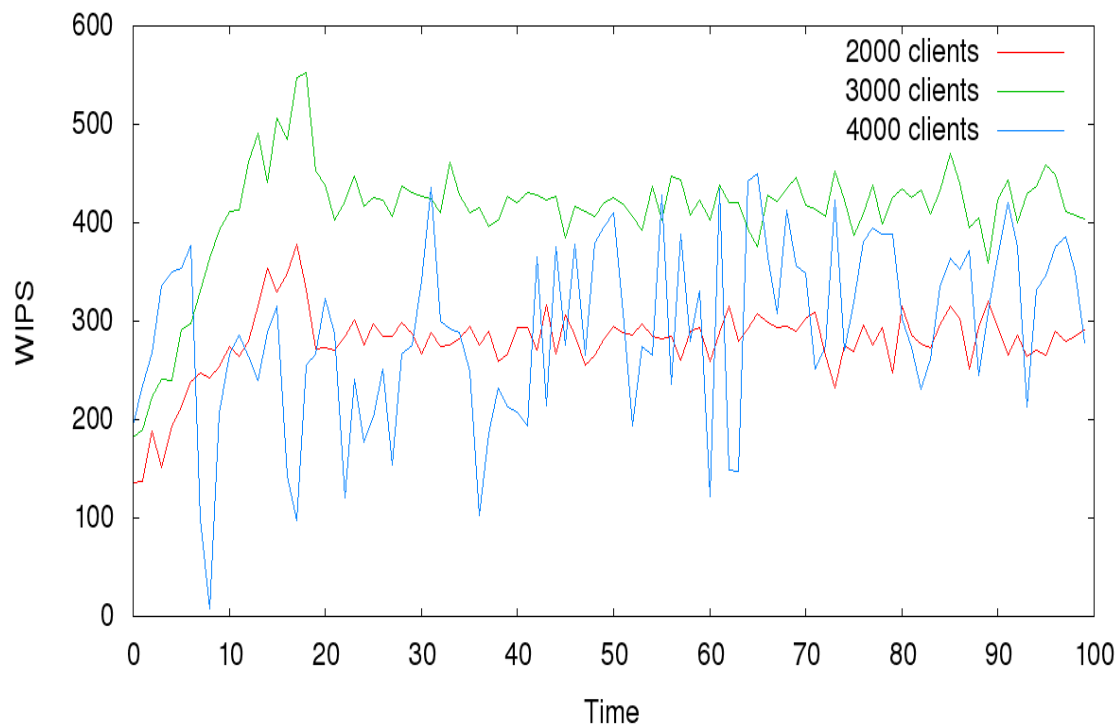


Figura 3: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Shopping do experimento 1

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.

Com a combinação dos resultados desses gráficos, temos 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 quantidades de usuários testadas.

Agora, analisaremos os dados do experimento 2.

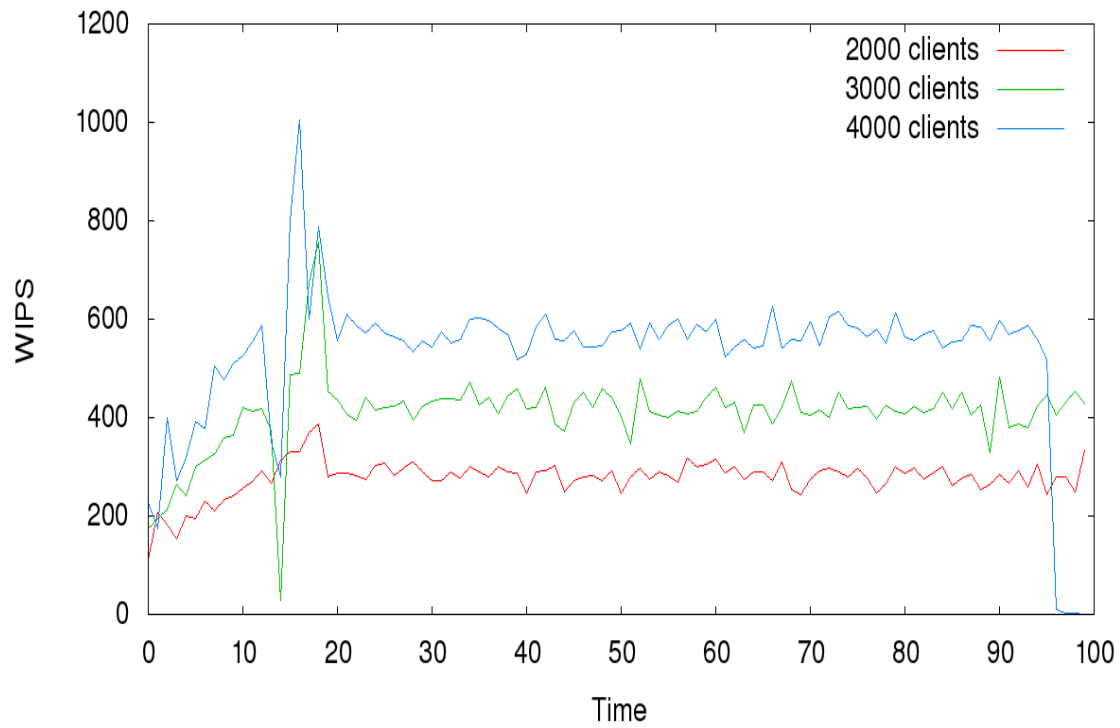


Figura 4: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Browsing do experimento 2

O gráfico indica um melhor desempenho geral para 4000 clientes. Entretanto, no final do experimento há uma grande queda, que indica problemas devido a alta carga. Nesse caso, 3000 clientes apresenta melhor estabilidade em relação a 4000 e maior quantidade de WIPS em relação a 2000 clientes

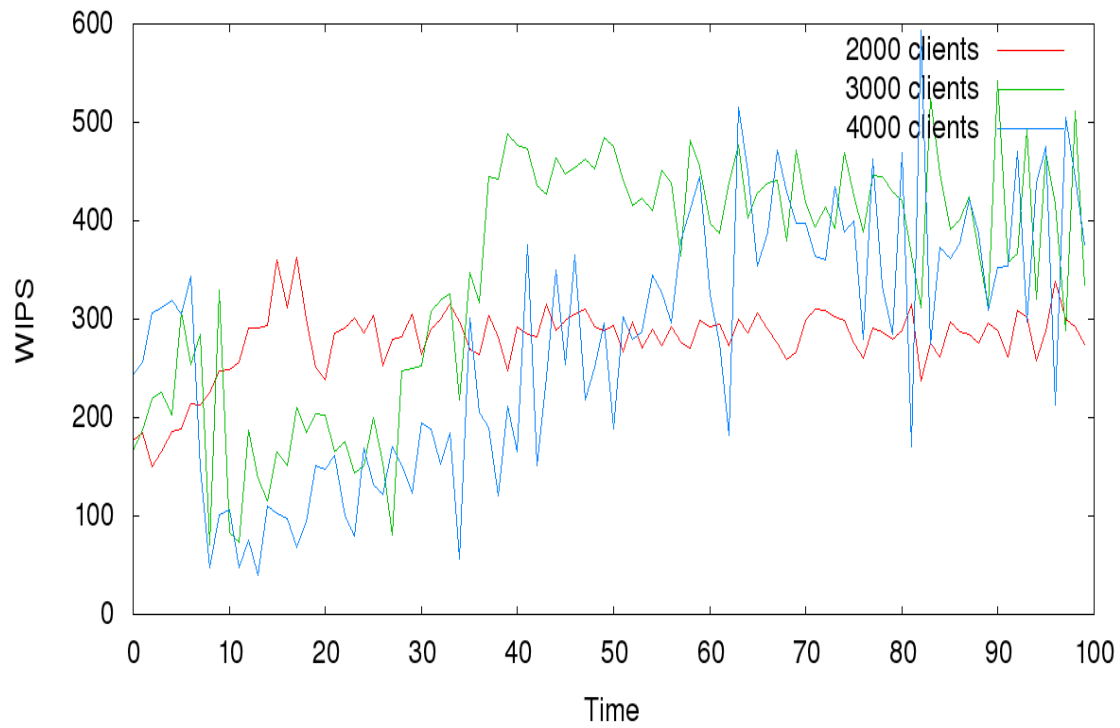


Figura 5: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Ordering do experimento 2

O melhor desempenho geral foi para 3000 clientes, pois essa carga manteve uma boa média durante todo o experimento e se manteve no mesmo nível de WIPS para 4000 clientes na segunda metade do experimento.

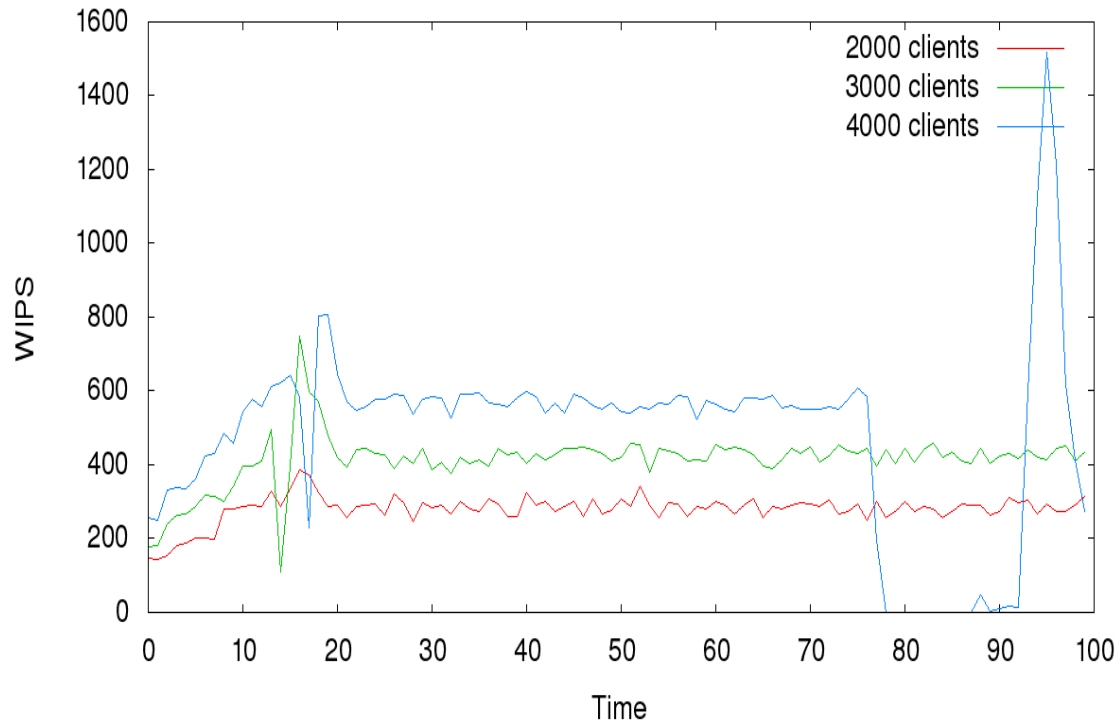


Figura 6: WIPS por tempo(s) para cada carga simulada pelo RBE no perfil Shopping do experimento 2

Este resultado se assemelha ao do perfil Browsing. 4000 clientes possui bom desempenho até 70s do experimento, quando sofre queda de WIPS, o que indica que esta alta carga sobrecarrega o servidor. Como no perfil Browsing, a carga mais estável é de 3000 clientes.

5 Conclusão