

# Replicação de Dados em Servidores Paralelos de Comércio Eletrônico

A. Pereira B. Gontijo T. Cançado W. Meira Jr.  
{adrianoc, gontijo, tassni, meira}@dcc.ufmg.br

<sup>1</sup> e-SPEED – Departamento de Ciência da Computação  
Universidade Federal de Minas Gerais  
Belo Horizonte – MG – Brasil

## Resumo—

Comércio eletrônico, a exemplo de outras aplicações da WWW, tem crescido a taxas sem precedentes, resultando em servidores sobrecarregados e que oferecem serviços de má qualidade. No contexto de arquiteturas tradicionais de comércio eletrônico, o custo de consultas aos servidores de banco de dados é comumente uma das principais causas da degradação de desempenho dos servidores de comércio eletrônico. Esses custos são resultado da complexidade inerente às consultas e da contenção associada à manutenção da integridade dos dados. Uma estratégia para minimizar o custo de acesso a dados é a replicação destes em outros componentes do servidor de comércio eletrônico. A replicação deve ser coerente com os vários tipos de serviços providos pelo servidor e explorar a localidade de referência a dados de cada tipo de serviço. Por outro lado, um servidor de comércio eletrônico sequencial que atenda uma única operação por vez também seria ineficiente, motivando a sua paralelização. Neste artigo, discutimos estratégias de replicação de dados em servidores paralelos de comércio eletrônico e descrevemos a implementação destas estratégias no âmbito de uma livraria virtual paralelizada, validada através da utilização de cargas de trabalho reais. Os resultados mostram que a estratégia da replicação de dados, aliada a um servidor paralelo, permite melhorias significativas no tempo de resposta às requisições de um cliente e na performance do servidor de comércio eletrônico.

## I. INTRODUÇÃO

A Internet e, em particular, a *World Wide Web* (WWW) têm presenciado um enorme crescimento nos últimos anos. As pessoas têm utilizado a grande rede para se comunicar, buscar informações variadas e se divertir. Além disso, uma nova tendência começa a fazer parte do cotidiano dos usuários: o comércio eletrônico.

De forma abrangente, podemos definir comércio eletrônico como o uso de recursos de rede e tecnologia da informação para facilitar processos centrais ao funcionamento de uma organização. Existe um consenso de que o tráfego intenso e o baixo desempenho são alguns dos grandes empecilhos para um maior crescimento do comércio eletrônico.

Pesquisas revelam que a maioria dos consumidores, ao se deparar com um *site* cujo tempo de resposta não corresponde ao que considera razoável, costuma desistir e procurar serviços em outros *sites* que atendam à sua necessidade com qualidade [6]. Logo, torna-se extremamente importante minimizar esse tempo, evitando a perda de clientes.

No entanto, minimizar o tempo de resposta a uma requisição efetuada por um cliente não é uma tarefa fácil, haja vista a grande quantidade de dados processados ao atender a mesma. O armazenamento desses dados, em geral, é feito em SGBDs, pois estes garantem três propriedades fundamentais: coerência, consistência e persistência. Por outro lado, o custo de acesso a SGBDs é normalmente alto e isso resulta em um aumento da latência observada pelos clientes [1].

A fim de otimizar o tempo de resposta, algumas estratégias aplicadas no servidor de comércio eletrônico têm sido utilizadas e têm apresentado resultados promissores. Dentre elas, a replicação de dados e utilização de servidores paralelos merecem destaque.

A primeira consiste na replicação dos dados entre componentes dos servidores de comércio eletrônico, diminuindo o número de consultas submetidas ao SGBD e, conseqüentemente, o tempo de resposta. Em [8], é apresentado um mecanismo de *cache* semântica voltado para a *Web* e, em [2], uma análise detalhada sobre *cache* em servidores de informação. Já a segunda estratégia baseia-se na utilização de servidores paralelos, possibilitando assim o processamento simultâneo de requisições independentes.

Este artigo está dividido em sete seções. Na próxima seção, descrevemos a arquitetura de servidores de comércio eletrônico, seguido de um estudo sobre modelos de replicação de dados no contexto de comércio eletrônico. Na seção IV, é analisada uma carga de trabalho real de um servidor de comércio eletrônico e discutidas as oportunidades de replicação. A implementação da nossa estratégia de replicação é descrita na Seção V. Na Seção VI, são apresentados os resultados obtidos em experimentos utilizando uma carga real. Por fim, a Seção VII mostra as conclusões e os trabalhos futuros.

## II. SERVIDORES DE COMÉRCIO ELETRÔNICO

Servidores de comércio eletrônico e servidores WWW tradicionais atuam em contextos bem distintos [5]. Conseqüentemente, existem muitas diferenças entre essas classes de servidores. No entanto, podemos diferenciá-los pelas funciona-

lidades adicionais suportadas e as informações armazenadas por um servidor de comércio eletrônico. Servidores WWW apenas recebem e respondem requisições, enquanto que servidores de comércio eletrônico mantêm um conjunto de mensagens transmitidas entre o usuário e o servidor, e suas ações associadas. Esse conjunto de informações, o qual denominamos sessão de usuário, é guardado para efeitos de integridade transacional e suporte aos serviços oferecidos.

As subseções seguintes descrevem de forma objetiva a arquitetura de um servidor de comércio eletrônico e o seu modelo de armazenamento de dados.

## A. Arquitetura

### A.1 Componentes

Um servidor de comércio eletrônico pode ser dividido em três componentes integrados (Figura 1):

1. **Servidor WWW:** É o gerente das tarefas, sendo responsável pela interface com os usuários (clientes), interagindo diretamente com eles, recebendo as solicitações de consultas ao banco de dados, disparando-as e repassando os resultados. Provê a interface entre a ferramenta de acesso do cliente (normalmente um *browser*) e o servidor de comércio eletrônico.
2. **Servidor de Transações:** Realiza o processamento das requisições submetidas ao servidor de comércio eletrônico, como a adição/remoção de um novo produto à cesta de compras.
3. **Banco de Dados:** Armazena todas as informações da loja virtual, como descrição do produto e nível de estoque. Mais que um simples repositório, agrega uma série de funcionalidades que permitem o acesso padronizado, seguro e eficiente aos dados, através, por exemplo, da criação de índices e controle de acesso utilizando autenticação por usuário.

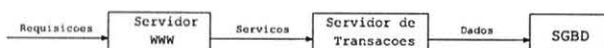


Figura 1. Arquitetura de 3 níveis

### A.2 Requisitos

Há quatro requisitos essenciais para a implementação de servidores de comércio eletrônico:

1. **Gerenciamento do estado da aplicação:** O estado da aplicação é o conjunto de informações relativas aos usuários e às suas interações acessando o servidor. Particularmente, o gerenciamento do estado da aplicação possibilita a autenticação de usuários, controle de sessão do usuário e o uso de cursores para efeitos de desempenho junto ao SGBD.

2. **Integridade transacional:** Diz respeito às transações que satisfazem certas características tradicionalmente agrupadas em quatro propriedades sob o acrônimo ACID (Atomicidade, Consistência, Isolamento, Durabilidade). Estas características possibilitam, entre outros, o controle de concorrência sobre a base de dados e mecanismos de recuperação em caso de falhas.
3. **Segurança:** Os requisitos de segurança dizem respeito às restrições de acesso aos dados gerenciados pelo servidor. A *Web* é reconhecidamente um ambiente não seguro e a disponibilização de aplicações de comércio eletrônico deve levar em consideração os requisitos básicos de restrições de acesso aos objetos do banco de dados.
4. **Desempenho:** O desempenho de servidores de comércio eletrônico é um fator crucial para a satisfação dos clientes e consequente efetivação de transações comerciais.

### A.3 Métricas de Desempenho

A qualidade do serviço prestado pelos servidores de comércio eletrônico é medida, geralmente, utilizando métricas como tempo de resposta e taxa de serviço [1].

Tempo de resposta é o intervalo de tempo entre o recebimento da requisição e o envio total da resposta. Esse tempo é usualmente medido no próprio servidor, uma vez que o tempo de resposta, observado pelos clientes, é muito difícil de ser medido pela variância das condições de rede e conectividade dos mesmos.

A taxa de serviço é o número de requisições atendidas por unidade de tempo. Unidades de medição comuns de taxa de serviço são requisições por segundo e transações por segundo. Essa métrica é bastante importante do ponto de vista da eficiência do(s) servidor(es), uma vez que uma baixa taxa de serviço pode levar um comprador em potencial a desistir de suas compras.

### A.4 Limitações

Existem alguns fatores que podem prejudicar significativamente o desempenho dos servidores em razão de contenção por recursos. Dentre os quais destacam-se:

- **Comunicação:** O tráfego na Internet está cada vez mais intenso, podendo elevar o tempo de execução de algumas operações. Outra fonte de contenção pode ser o grande número de requisições a serem tratadas por um servidor, que pode sobrecarregar a interface de rede do mesmo.
- **Processamento:** Tanto o tempo gasto em processamento de tarefas (CPU), quanto em acesso a disco podem comprometer as operações entre servidor e cliente, desde que o número de requisições seja maior que a capacidade de execução do servidor.

- **Acesso a Dados:** A velocidade de acesso bem como a taxa de serviço dos servidores de bancos de dados utilizados são fatores críticos para o bom desempenho do servidor como um todo.

### B. Armazenamento de Dados

Discute-se, nesta seção, alguns dos fatores associados aos serviços oferecidos pelos servidores de comércio eletrônico e algumas características dos dados utilizados para descrição dos produtos vendidos. Toda a discussão está focada em torno de uma classe de serviços de comércio eletrônico que segue o modelo conhecido como *empresa-consumidor*, cujo exemplo mais comum são as lojas virtuais. No entanto, todos os conceitos e definições aqui expostos podem ser facilmente adaptados e generalizados para outras formas de comércio eletrônico, por exemplo, negociações *on-line* e leilões eletrônicos.

Qualquer loja de comércio eletrônico, voltada para o consumidor final, contém dados sobre os produtos disponibilizados. Por sua vez, esses dados podem ser classificados, de acordo com sua taxa de atualização, em três categorias distintas:

1. **Estáticos:** Incluem o conjunto de páginas estáticas mantidas pela loja e o conjunto de todas as descrições estáticas sobre os produtos (como, por exemplo, o título do livro, autor, editora, ano de publicação, sumário e outros).
2. **Pouco voláteis:** Incluem os dados cujo conteúdo varia a uma taxa muito baixa (p.ex., a colocação de um livro no *ranking* de uma livraria virtual, ou as opiniões a respeito de um produto que foram postadas por outros consumidores). A ausência de coerência temporal entre o servidor original e os dados replicados pode ser tolerada seguramente para o caso de dados pouco voláteis. Este tipo de informação pode incluir até versões dos resultados das consultas realizadas por usuários, na base de dados da loja, armazenadas previamente. De modo geral, os dados pouco voláteis são modificados em função das ações tomadas pelos administradores do *site*, e não da interatividade entre os clientes e a loja.
3. **Muito voláteis:** Incluem os dados modificados com grande rapidez, normalmente em função das interações entre os clientes e a loja virtual. O melhor exemplo é provavelmente a disponibilidade dos produtos. A informação de disponibilidade de um produto deve ser atualizada toda vez que um consumidor o compra. A manipulação de dados muito voláteis é sempre mais complexa e, portanto, exige maior atenção.

## III. DESCRIÇÃO DE TÉCNICAS

### A. Replicação de Dados

Há dois importantes fatores que motivam a replicação de dados em servidores de comércio eletrônico. O primeiro é o número limitado de operações providas pelo servidor, as quais são bem definidas. No caso de uma loja virtual, por exemplo, podemos distinguir três operações que podem se beneficiar da replicação de dados: busca por termos (*search*), navegação utilizando classes de produtos pré-definidos (*browse*) e verificação de detalhes sobre um dado produto (*select*). O segundo fator é a redução do custo de execução destas operações pelo acesso a informações replicadas, dispensando a consulta ao SGBD, cujo custo de acesso é relativamente alto. Assim sendo, replicam-se dados com o objetivo de otimizar o tempo de resposta da requisição de um cliente e, assim, aumentar o *throughput* do servidor tendo em vista o maior número de requisições processadas em um dado intervalo de tempo.

Idealmente, os dados podem ser replicados em quaisquer componentes entre o SGBD e a máquina do cliente, incluindo o servidor WWW e servidores *proxy cache*[2]. Os servidores de transação são os melhores candidatos a armazenarem os dados replicados do SGBD por várias razões. A principal delas é ser o servidor de transações o responsável pela execução das mesmas, normalmente explorando a localidade temporal dos dados replicados. Adicionalmente, o servidor de transações já interage com o SGBD, podendo utilizar os mesmos mecanismos para obter dados a serem replicados. Por fim, já que ambos (servidor de transações e SGBD) são componentes do servidor de comércio eletrônico, não há necessidade de medidas de segurança para acesso e armazenamento dos dados replicados, sendo que o servidor de transações satisfaz os requisitos de segurança, por ser parte do próprio servidor.

No caso das três operações mencionadas, os dados replicados são normalmente armazenados em duas estruturas: listas de produtos e informação de produto. Listas de produtos são conjuntos de identificadores que satisfazem um critério que caracteriza as listas. Exemplos de critérios são os termos de uma busca ou uma categoria de navegação. Essas listas contêm referências para as informações de produto. Ambas as estruturas de dados podem ser visualizadas na Figura 2.

As operações de busca e navegação verificam a lista de produtos associada ao critério da operação e geram como resposta os produtos que o satisfazem. Já a seleção de produtos para verificação detalhada utiliza apenas as informações de produto.

Embora as estruturas de dados, contendo listas de produtos e informações de produto, sejam utilizadas originalmente para a construção de respostas aos clientes, elas também se prestam para a replicação de dados. Para tanto, basta que

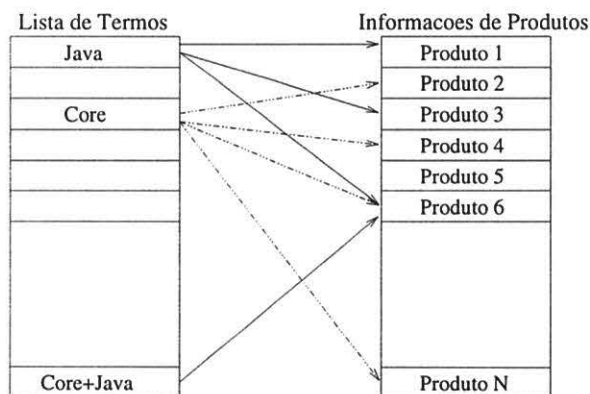


Figura 2. Estrutura de dados do servidor de transações

essas estruturas sejam persistentes, ou seja, não sejam deslocadas após o fim da consulta que causou a sua replicação, e que sejam implementados mecanismos para controle da utilização de memória no servidor de transações. Desta forma, operações de busca e navegação seriam implementadas verificando-se a existência das listas de produtos necessárias e das informações dos produtos referenciados nestas listas, enquanto operações de verificação de detalhes dos produtos acessariam diretamente a tabela de produtos. Os detalhes da implementação são descritos na Seção V.

### B. Processamento Paralelo

Utilizando-se a técnica de replicação de dados, a contenção provocada pelos acessos ao SGBD tenderá a ser minimizada. O desempenho do sistema passa então a ser comprometido pelo acesso serial aos dados replicados e pelo número de clientes atendidos simultaneamente. Em um servidor de transações sequencial, apenas um cliente pode ser atendido a cada momento, o que representa um desperdício de recursos, pois o processamento envolve um grande número de operações de entrada e saída. Tornar o servidor de transações paralelo é uma alternativa interessante para aumentar o número de requisições atendidas por unidade de tempo.

Desta forma, a paralelização demanda a análise dos tipos de dados e dos tipos de acessos comumente realizados em transações de comércio eletrônico. Como estamos tratando de informações estáticas ou pouco voláteis sobre os dados, a maioria das operações realizadas pelo servidor de transações é de leitura. Neste caso, apenas aquelas que alteram o estado do servidor e das transações efetuando escritas exclusivas (como alteração de estoques, pagamento e efetivação de compra) necessitam ser executadas de forma serial para manter a integridade e consistência dos dados.

As três operações citadas, *browse*, *search* e *select*, são, portanto, favorecidas pela paralelização do servidor de

transações. As duas primeiras são mais fortemente beneficiadas, pois retornam informações estáticas sobre os produtos indicados. Estando essas informações em uma *cache*, o acesso concorrente a ela possibilita um *throughput* maior. Para as operações de *select*, esse ganho não seria tão significativo, já que informações detalhadas como preço e quantidade em estoque são voláteis e podem estar desatualizadas na *cache*.

Um ponto crucial na implementação associada das técnicas de replicação de dados e de processamento paralelo consiste em manter a consistência dos dados armazenados na *cache*, problema este acentuado pelo acesso concorrente a eles. Na Seção V, abordaremos em profundidade essa situação.

## IV. CARACTERIZAÇÃO DA CARGA DE TRABALHO

Nesta seção, é apresentada uma caracterização da carga de trabalho utilizada na execução dos testes. Essa carga é formada pelo *log* real de uma livraria virtual contendo 10.000 sessões de usuário, cada uma composta em média por 25,3 requisições. Há 85.083 requisições de busca contendo, em média, 2,03 termos. Essas requisições de busca possuem 46.268 termos distintos e 52.569 consultas distintas (ou seja, consultas cujo conjunto de termos é único). A carga também contém 17.823 requisições de navegação para 127 categorias de produtos. Finalmente, há 75.288 requisições de verificação detalhada por 1.740 dos 4.309 produtos na base de dados. Para avaliar a localidade de referência desta carga de trabalho, foram simuladas *caches* infinitas para as listas de produtos e informações de produtos. Avaliou-se a taxa de acerto na *cache* de listas de produtos para buscas considerando, ambas as organizações, termos e consultas. A taxa de acerto para termos é 73% enquanto a taxa de acerto para consultas é 38%. Já a taxa de acerto para navegação é 99%, enquanto a taxa de acerto para verificação de detalhes é 97%.

## V. IMPLEMENTAÇÃO

Descreve-se aqui a implementação de uma estratégia de replicação de dados em uma loja virtual paralela. Esta implementação busca explorar as características da carga de trabalho, apresentadas na Seção IV, e verificar os ganhos obtidos através da paralelização da loja virtual em conjunto com a replicação de dados. O servidor de transações da loja virtual é um programa composto de várias *threads* que executam o mesmo código. Cada *thread* pode atender uma requisição de um cliente por um serviço provido pela loja. O assinalamento de requisições a *threads* é feito utilizando um algoritmo *round-robin* e o acesso a estruturas compartilhadas é controlado por semáforos.

A *cache* implementada é organizada em três níveis, conforme mostrado na Figura 3. No primeiro nível (*Cache 1*) residem as listas de produtos utilizadas para a indexação dos dados armazenados no segundo nível da *cache*. Por sua vez,

o segundo nível (*Cache 2*) contém apenas as informações de produto a serem incluídas nas respostas a operações de busca e navegação. O terceiro nível da *cache* (*Cache 3*) armazena as informações de produto restantes, que não são armazenadas no segundo nível. As informações dos segundo e terceiro níveis são utilizadas para construir a resposta para operações de verificação de detalhes de produtos. Essa divisão das informações de produto em dois níveis é motivada pelo grande número de operações de busca, as quais demandam apenas uma fração daquelas informações.

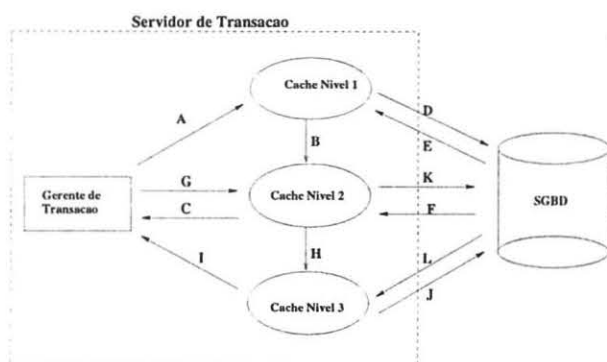


Figura 3. Modelo de Replicação de Dados

A Figura 3 ilustra o funcionamento do mecanismo de *cache* implementado, para as operações de busca, navegação e verificação. O ponto de partida da utilização da *cache* é o momento em que a requisição de serviço é recebida pelo servidor de transação e identificada como uma operação que pode se beneficiar da *cache*, pelo gerente de transações.

As operações de busca e navegação são idênticas, porque a *cache 1* é organizada como uma lista de consultas. A operação se inicia com a verificação da existência da lista para a consulta na *cache 1* (A). Se a lista não estiver na *cache*, a consulta é submetida ao SGBD (D), causando a replicação, em paralelo, dos dados pertinentes nas *caches 1* e *2* (E e F, respectivamente). Estando as *caches* preenchidas, as informações de produto demandadas (B) são utilizadas pelo gerente de transações (C) para a construção da resposta.

A operação de verificação de detalhes de um produto é similar. Inicialmente o gerente de transações verifica se o produto desejado está replicado na *cache 2* (G). Se este não for o caso, todas as informações de produto são requisitadas (K) e as *caches 2* e *3* atualizadas em paralelo (F e L). Se apenas a *cache 2* contém as informações de produto (H), é feita uma consulta ao SGBD (J) e a *cache* é atualizada (L). Estando as *caches 2* e *3* aptas a atender a requisição, as informações de produto são acessadas pelo gerente de transações (C e I).

O gerenciamento de espaço [4] das *caches* é feita de forma hierárquica. Assim, todas as referências em listas da *cache 1* são para dados válidos na *cache 2*, e todos os produtos cujos detalhes estejam armazenados na *cache 3* também

Operações	Sem Cache	Cache		Cache aquecida	
	Tempo Resposta	Tempo Resposta	Hit Ratio	Tempo Resposta	Hit Ratio
Busca	9.11	7.48	13.65	6.80	27.25
Navegação	17.99	7.44	79.59	7.17	79.17
Verificação	4.93	1.92	54.09	1.78	54.24

Tabela I

Hit ratio e TEMPO DE RESPOSTA DAS VÁRIAS CONFIGURAÇÕES

têm suas informações na *cache 2*. A política de reposição de dados[7] utilizada é a LRU e também é implementada de forma hierárquica. Quando o limite de armazenamento da *cache 1* é atingido, as listas de consultas acessadas menos recentemente são descartadas. A mesma estratégia é empregada na *cache 3*. Já o descarte de entradas na *cache 2* é mais complexo, uma vez que as informações de produtos são referenciadas por listas na *cache 1*. Assim, sempre que a *cache 2* atinge o seu limite de armazenamento, remove-se as entradas na *cache 1* e suas referências exclusivas nas *caches 2* e *3*, para que a consistência de toda *cache* seja mantida.

## VI. RESULTADOS

Nesta seção apresenta-se os resultados preliminares obtidos a partir dos testes feitos em um servidor de comércio eletrônico, utilizando a carga descrita na Seção IV. A carga de trabalho em todos os experimentos foi gerada por uma versão modificada do Surge [3] que é capaz de gerenciar sessões, ou seja, sequências de requisições realizadas por um cliente. Para os vários experimentos utilizou-se uma loja virtual cujo servidor de transações foi implementado em linguagem C. Como mencionado, o servidor de comércio eletrônico é organizado em três níveis: servidor WWW, servidor de transações e servidor de banco de dados. O servidor WWW foi o Apache versão 1.3.9 executando em um Pentium 166MHz. O servidor de banco de dados foi o MySQL versão 3.22.22 e executando em um computador Pentium 233MHz. Tanto o servidor de transações quanto os clientes foram executados em um Intel Celeron de 300 MHz. Em todas as máquinas foi utilizado o sistema operacional Linux 2.2.12.

Foram realizados experimentos com três configurações para avaliar estratégias de paralelização e replicação de dados: servidor sem *cache*, com *cache* e com *cache* aquecida. A *cache* foi inicializada utilizando-se as 2.000 expressões mais frequentes em operações de busca da carga de trabalho. A motivação para a inicialização da *cache* é a baixa localidade de referência exibida pelas operações de busca. A inicialização permite verificar o desempenho da *cache* em situações mais realistas.

A análise da Tabela I permite concluir que a inicialização da *cache* traz melhoras significativas, tendo em vista o aumento do *hit ratio* em operações de busca. Observando o *hit*

	Número de <i>threads</i>				
	1	3	6	9	12
Com Cache Aquecida	1.44	2.21	3.11	4.89	6.96
Sem Cache	1.43	2.15	2.85	3.71	4.77

Tabela II

NÚMERO DE REQUISIÇÕES ATENDIDAS POR SEGUNDO

*ratio* de ambas as configurações que utilizam *cache*, verifica-se que os valores estão aquém dos obtidos na caracterização de carga, o que pode ser explicado com base na duração dos testes, que foi de uma hora, sem considerar o tempo de inicialização.

Finalmente, observou-se uma considerável melhora do tempo médio de resposta às requisições quando utilizada a estratégia de replicação de dados, resultando em uma melhor qualidade de serviço para os clientes da loja virtual. Basta observar que o tempo médio de resposta para as três operações, no modelo sem *cache*, é 32 segundos, enquanto que a versão com *cache* equivale à metade desse tempo.

Avaliou-se a escalabilidade da loja virtual variando o número de *threads* que são mantidas pelo servidor. Para isso, foram realizados testes variando também o número de clientes que submeteram requisições simultaneamente à loja virtual de 1 a 12. A Tabela II apresenta os resultados obtidos nesses testes.

A partir da análise destes resultados pode-se constatar que a estratégia de paralelização adotada atingiu o objetivo desejado, ou seja, o tempo de resposta diminuiu com a utilização de mais clientes no servidor de transação. Isso pode ser verificado a partir do aumento do número de requisições atendidas por segundo com o aumento da carga de trabalho. Similarmente, à medida que o número de clientes aumenta, também observa-se que o ganho obtido com a abordagem que emprega *cache* torna-se mais significativo.

## VII. CONCLUSÃO E TRABALHOS FUTUROS

Neste artigo, discutiu-se o modelo, a implementação e a validação de uma arquitetura para maximizar o desempenho obtido pelos servidores de comércio eletrônico. Esta arquitetura foi proposta tendo em vista a crescente utilização dos *sites* de comércio eletrônico. É sabido que a performance apresentada pelos servidores desses *sites* é fator determinante de seu sucesso.

Por sua vez, o desempenho obtido por esses servidores está diretamente ligado ao dos Sistemas Gerenciadores de Banco de Dados, onde são armazenados os dados a serem processados. A solução adotada para contornar o problema foi a replicação de dados, que elimina a latência de acesso ao SGBD. No entanto, como servidores de comércio eletrônico devem atender a múltiplas requisições de maneira eficiente, apenas a replicação de dados não proporciona o aumento

de desempenho pretendido. Para atingir o patamar desejado, paralelizou-se o processamento das transações no servidor de comércio eletrônico. Dessa forma, obteve-se um ganho significativo, conforme mostrado nos testes realizados, onde o número de requisições e clientes atendidos por servidor cresceu e o tempo de resposta diminuiu.

Dando continuidade ao trabalho aqui apresentado, esforços estão voltados para o estudo e a implementação de uma estratégia de replicação de dados indexada por termos, e uma análise comparativa entre as abordagens de replicação estudadas. Além disso, analisa-se a distribuição de carga entre vários servidores de transação, pois essa distribuição, associada à paralelização efetuada, provavelmente proporcionará uma maior escalabilidade aos servidores de comércio eletrônico.

## REFERÊNCIAS

- [1] Goedson Paix ao, Wagner Meira Jr., Virgílio Almeida, Daniel Menascé, and Adriano Pereira. Design and implementation of a tool for measuring the performance of complex e-commerce sites. Performance Tools 2000 - Motorola University, Illinois - USA.
- [2] Yigal Arens and Craig A. Knoblock. Intelligent caching: Selecting, representing, and reusing data in an information server. Proceedings of the Third International Conference on Information and Knowledge Management, 1994.
- [3] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In Proc. 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, June 1998.
- [4] M. Carey, M. Franklin, M. Livny, and E. Shekita. Data caching tradeoffs in client-server dbms architectures. Proceedings of ACM-SIGMOD 1991 International Conference on Management of Data, Denver, Colorado, pages 357-366, May 1991.
- [5] D. Krishnamurty and J. Rolia. Predicting the performance of an e-commerce server: Those mean percents. In Proc. First Workshop on Internet Server Performance - ACM SIGMETRICS, July 1998.
- [6] Rodrigo Fonseca Daniel A. Menascé, Virgílio A. F. Almeida and Marco A. Mendes. A methodology for workload characterization of e-commerce sites. 1999. In Proc. of ACM's Electronic Commerce Conference.
- [7] S. Dar, M. Franklin, B. Jonsson, D. Srivastava, and M. Tan. Semantic data caching and replacement. Proc. VLDB Conf., Bombay, India, September, 1996.
- [8] D. Lee and W. Chu. A semantic caching scheme for web sources. UCLA-CS-TR-990004, 1999.
- [9] Daniel A. Menascé and Virgílio A. F. Almeida. Capacity Planning for Web Performance: metrics, models and methods. Prentice Hall, 1998.