

GetLB: Balanceamento de Carga Eficiente para o Escalonamento de Transações Eletrônicas Financeiras

Rodrigo da Rosa Righi, Cristiano André da Costa, Alexandre Andrade, Lucas Graebin
Programa Interdisciplinar de Pós-Graduação em Computação Aplicada – UNISINOS, Av.
Unisinos, 950 - São Leopoldo, RS.

{rrrighi,cac}@unisinos.br, {alexandre.andrade, lgraebin}@gmail.com

Abstract. *This article aims at presenting the ideas for developing a framework for load balancing called GetLB. Considering the electronic funds transfer (EFT) context, GetLB offers a new way of organizing the interactions between the Dispatcher and Processing Machines in processing centers. This organization enables the Dispatcher to combine up-to-date information for executing a dynamic scheduling algorithm instead of using the Round-Robin approach among the Processing Machines. GetLB's scheduling algorithm splits the transactions in different types, combining their requirements of CPU, memory, disk and network data to offer an efficient load balancing. We implemented a prototype with Java RMI and tests were conducted on homogeneous and heterogeneous environments with real tracing data from an EFT-oriented service provider. The evaluation is focused on the scheduling part of the proposed framework, presenting advantages of adopting GetLB algorithms instead of the traditional Round-Robin approach.*

Resumo. *Este artigo tem como objetivo apresentar as ideias para o desenvolvimento de um arcabouço para balanceamento de carga chamado GetLB. Considerando o contexto de transferência eletrônica de fundos (TEF), GetLB oferece uma nova forma de organizar as interações entre o chaveador, ou despachante de transações, e as máquinas processadoras. Esta organização permite que o chaveador combine informações atualizadas para a execução de um algoritmo de escalonamento dinâmico para o mapeamento de transações, ao invés de usar a abordagem Round-Robin para tal. O algoritmo de escalonamento de GetLB divide as transações em diferentes tipos, combinando as necessidades de CPU, memória, disco e rede de cada um deles para oferecer um balanceamento de carga eficiente. Implementou-se um protótipo com Java RMI e testes foram conduzidos em ambientes homogêneos e heterogêneos com dados reais de uma empresa prestadora de serviço na área de TEF. A avaliação está focada na parte de escalonamento do arcabouço proposto, mostrando as vantagens da adoção de algoritmos de GetLB em vez da abordagem tradicional Round-Robin.*

1. Introdução

Atualmente, percebe-se o alto crescimento da adoção de meios de pagamento eletrônicos ao invés do uso de dinheiro em cédulas de papel ou cheque. Além da conveniência para os consumidores, o uso de cartões de crédito e débito representa um fator de competitividade para as empresas e facilita o acesso a serviços disponibilizados na Internet [1]. Normalmente, uma transação eletrônica está relacionada com operação de compra ou requisição de saldo e percorre um caminho de ida e volta desde o terminal até um centro de processamento. Um terminal pode ser representado por um ponto de venda (POS - *Point of Sale*), Transferência Eletrônica de Fundos (TEF), ATM (*Automatic Teller Machine*), bem como por dispositivos

móveis [2]. Depois da chegada na companhia de processamento, as transações são recebidas por um comutador (também chamado no contexto desse artigo de chaveador) que atua como um escalonador que as repassa para máquinas processadoras, ou MPs.

O gerenciamento de recursos e o escalonamento são serviços-chave para que se obtenha eficiência no processamento das transações. O mecanismo padrão para mapeamento de transações para MPs é o Round-Robin, o qual emprega uma lista de recursos e atribui transações de maneira circular [1]. Como ilustrado na Figura 1 (a), esse método de escalonamento representa uma maneira rápida e fácil para que se atinja um escalonamento ótimo quando ambos conjuntos de transações e MPs são caracterizados por um sistema homogêneo. Por exemplo, todas as transações devem ser estritamente de um único tipo, com os mesmos requerimentos de CPU e E/S entre si, e as MPs devem ser construídas com os mesmos componentes de hardware e software. Claramente, Round-Robin não é a melhor alternativa quando sistemas dinâmicos e/ou heterogêneos estão presentes. Sistemas de transações eletrônicas apresentam ambas características, uma vez que o intervalo entre chegadas das transações não é constante e que devem lidar com diferentes tipos de transações como depósito, retirada de dinheiro, saldo, reversão (*undo*) e telefonia pré-paga.

As transações eletrônicas podem ser organizadas em diferentes tipos, cada qual possuindo requisitos próprios de uso de CPU, acesso a banco de dados, consumo de rede e memória. Tais peculiaridades podem fazer com que o escalonamento Round-Robin sobrecarregue algumas máquinas e deixe outras ociosas. Ainda assim, a maioria das companhias de processamento de transações ainda usam o escalonamento Round-Robin porque ele é rápido na tomada de decisões, embora muitas vezes ela não seja a mais adequada [1], [2]. A Figura 1 (b) ilustra essa situação, onde todas as máquinas processadoras são homogêneas e localizadas na mesma sala de servidores (data center). Essa estrutura representa uma organização tradicional hoje em dia nos centros de processamento [6]. Nessa configuração, tipicamente os subsistemas requeridos pelas MPs, como segurança da informação, banco de dados e prevenção de fraudes, também estão presentes na mesma rede local.

Nesse contexto, o presente artigo apresenta uma proposta de arcabouço de balanceamento de carga chamada GetLB. GetLB atua como uma alternativa ao método Round-Robin no âmbito de transações eletrônicas financeiras e consiste das seguintes partes: (i) arquitetura de comunicação; (ii) escalonamento de transações; (iii) notificação. A comunicação diz respeito a eficiente interação entre o balanceador e as MPs para a coleta de dados de escalonamento. O tópico escalonamento, por sua vez, trabalha de acordo com o tipo das transações, bem como a carga e configuração das máquinas MPs. A terceira parte, notificações, é pertinente para aumentar ou diminuir o número de MPs sem interromper a disponibilidade do serviço transacional. A Figura 1 (c) mostra o funcionamento desejado de GetLB. Quanto ao desenvolvimento, um protótipo de GetLB envolvendo as partes de comunicação e escalonamento foi implementado usando Java e RMI (*Remote Method Invocation*). Quanto a avaliação, ela está focada na parte de escalonamento e contempla uma comparação de desempenho entre o método de mapeamento tradicional RR e aquele proposto por GetLB. Em adição, tais métodos foram avaliados com rastros de execução reais obtidos com a empresa GetNet (<http://www.getnet.com.br>) sobre dois tipos de *clusters*, um homogêneo e outro heterogêneo. Os testes revelaram que, apesar de GetLB ter um algoritmo de escalonamento mais custoso que Round-Robin, a qualidade do mapeamento faz com que GetLB apresente melhor desempenho final na execução de transações.

Esse artigo é organizado da seguinte forma. A Seção II mostra o funcionamento de um sistema típico de transações eletrônicas. A Seção III descreve GetLB, enquanto que a Seção IV apresenta seu protótipo. A metodologia de avaliação e os resultados são mostrados nas Seções V e VI, respectivamente. Trabalhos relacionados são apresentados na Seção VII. Por fim, a Seção VIII ressalta os principais pontos do artigo e sua contribuição científica.

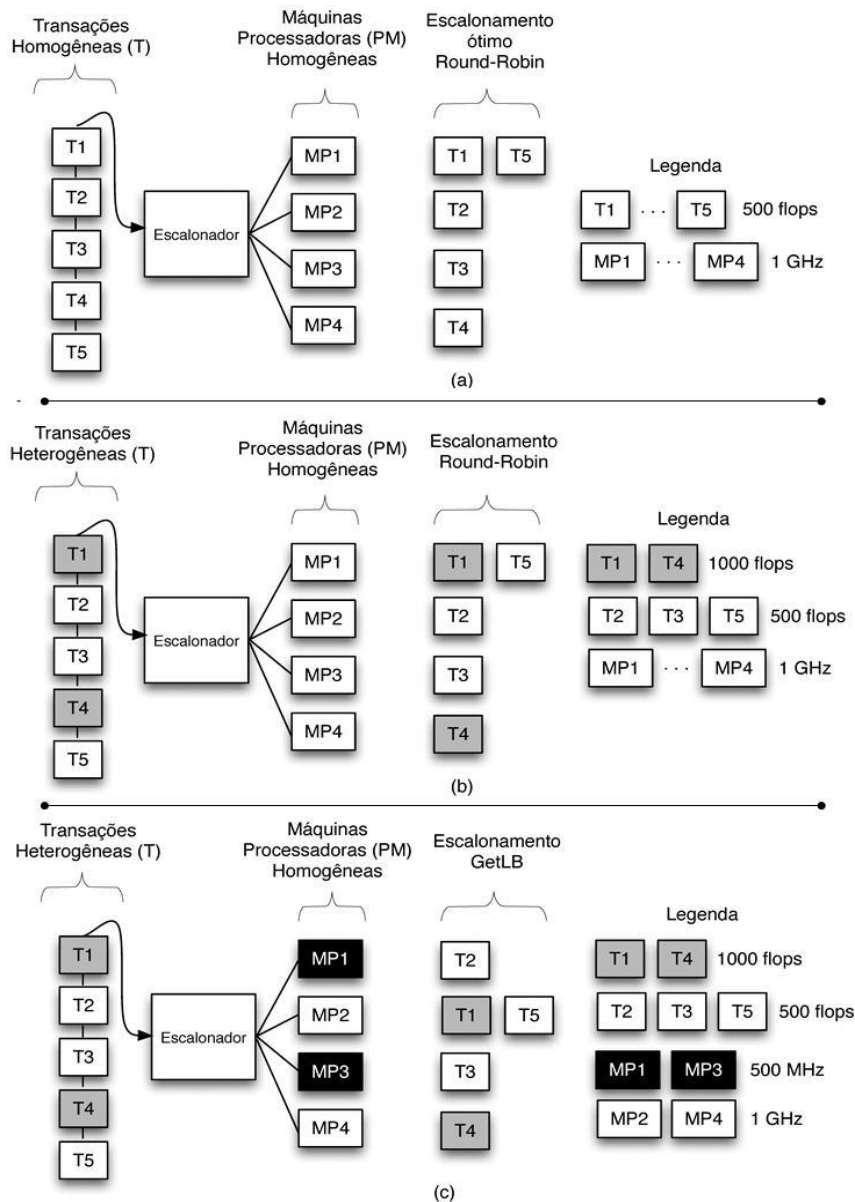


Figura 1. Escalonamento de transações: (a) sistema homogêneo; (b) sistema heterogêneo com Round-Robin; (c) estimativa de funcionamento da proposta GetLB.

2. Sistemas de Transações Eletrônicas

A Figura 2 ilustra a organização típica de uma empresa prestadora de serviço na área de transações eletrônicas. O comutador recebe as transações e as repassa para as máquinas MP. Essas, por sua vez, fazem uso de subsistemas internos para validar os dados transacionais e gerar arquivos de controle. Na sequência, é acionado um dos sistemas de saída, o qual

representa uma instituição responsável pela autorização da transação. Como já mencionada, a abordagem tradicional de mapeamento de transações para máquinas MP é o uso de Round-Robin. Em decorrência do elevado tempo de resposta de um subsistema, esse algoritmo pode fazer com que várias filas em MPs fiquem cheias enquanto outras tenham carga moderada [3]. Considerando que cada transação possui um tempo máximo (timeout) para permanecer dentro do sistema, essa abordagem pode levar a perda de transações [4].

Uma vez que o fluxo de chegada de transações não é constante (tipicamente picos ao meio-dia e às 19 horas), o Round-Robin é mantido por representar um sistema funcional e de fácil manutenção [1], [2]. Além dessa técnica de escalonamento, outra peculiaridade da organização tradicional diz respeito a organização de rede entre os componentes. O comutador, as máquinas MP e os subsistemas internos estão na mesma rede local e são acessíveis por endereços IP privados. O comutador, logicamente, também possui interface pública de rede para a recepção de transações dos terminais de acesso.

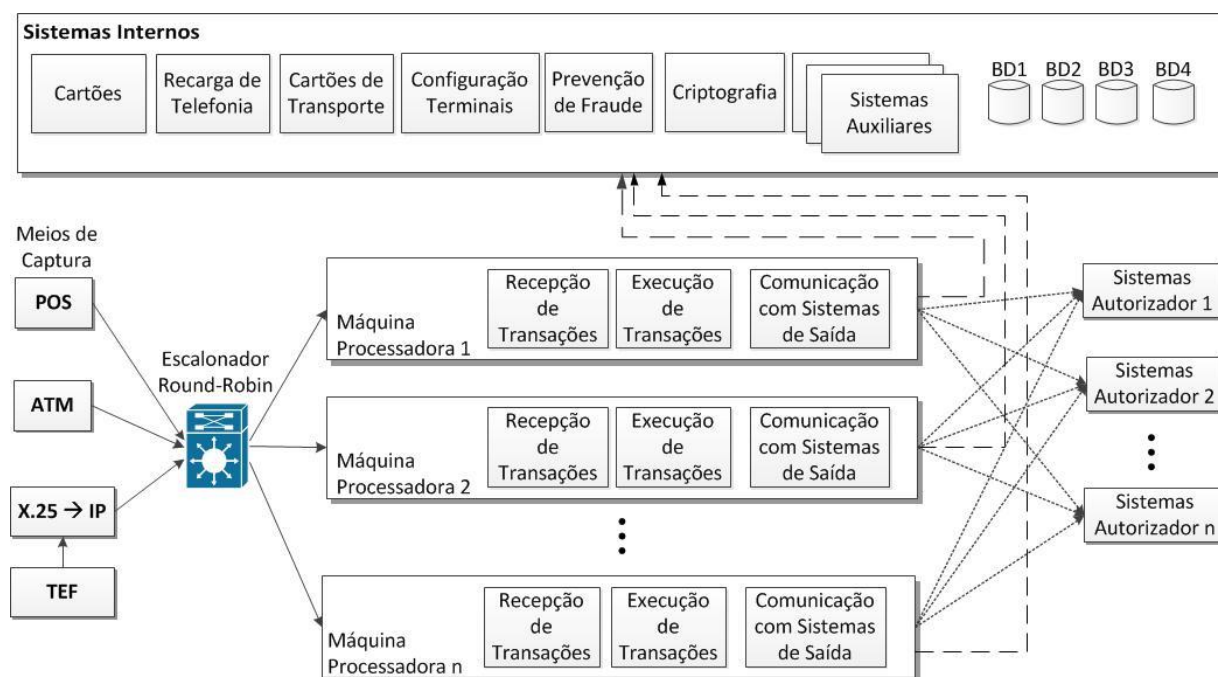


Figura 2. Arquitetura típica de um sistema de transações eletrônicas

3. GetLB: Arcabouço para Balanceamento de Carga em Transações Eletrônicas

GetLB é um arcabouço para balanceamento de carga para prestadoras de serviço de transações eletrônicas. Ele atua sobre os tópicos rede de comunicação e máquinas MP. Quanto ao primeiro, GetLB suporta o desacoplamento de componentes, que agora não precisam estar mais na mesma rede local. Quanto ao segundo tópico, GetLB suporta MPs heterogêneas e captura dinamicamente dados delas, das próprias transações e de comunicação para prover um escalonador eficiente. GetLB foi construído com base nas seguintes decisões de projeto: (i) o escalonamento deve combinar diferentes informações para compor a noção de carga; (ii) máquinas MP devem ser capazes de notificar o comutador na ocorrência de eventos que impactam nas decisões de escalonamento; (iii) o arcabouço deve lidar com recursos heterogêneos em ambos níveis de comunicação e computação.

Máquinas MP periodicamente atualizam dados de CPU e E/S para o nó comutador. Concorrentemente, esse último pode receber transações e usa os dados mais recentes sobre o conjunto de máquinas MP armazenados em sua memória para as decisões de escalonamento. O cálculo do escalonamento não dispara comunicação pela rede, o que é importante para melhorar a relação de seu custo benefício. Ainda assim, o tempo transcorrido para que seja encontrado um mapeamento foi uma questão-chave para o desenvolvimento da heurística de escalonamento de GetLB. Um bom escalonamento pode ter seu desempenho comprometido pelo tempo de sua obtenção.

A. Arquitetura de Comunicação

A Figura 3 ilustra a arquitetura de comunicação de GetLB. Com exceção do comutador, que deve estar presente na sede da prestadora de serviço, os demais componentes (máquinas MP e subsistemas internos) podem operar a partir de diferentes domínios administrativos da Internet. O único pré-requisito é que cada um deles deva ser acessado via endereço IP. Nesse contexto, o tempo de acesso do comutador para as máquinas MP pode não ser uniforme. O mesmo acontece quanto à comunicação de uma MP para cada componente dos subsistemas internos. Apesar da comunicação pela Internet apresentar uma latência entre 10 a 1000 vezes maior que a presente em redes locais [5], a flexibilidade da arquitetura de GetLB traz os seguintes benefícios: (i) dado que os recursos de uma instituição são limitados, pode-se agregar outros de empresas parceiras para suprir uma determinada demanda, caracterizando uma ação conhecida como *scaling out* [5]; (ii) certas instituições financeiras do Chile, por exemplo, impõem que transações de comércio eletrônico sejam computadas dentro do território nacional e, portanto, a expansão de componentes faz com que a empresa prestadora consiga ampliar a sua área de atuação. Para que seja usado um ou mais centros de dados para a computação das transações, é sempre necessário que o ambiente tenha o certificado PCI (*Payment Card Industry Security Standards*)¹. PCI apresenta as normas que regem o comportamento de uma transação desde o terminal até uma máquina processadora para que se tenha integridade e confidencialidade. Em adição, PCI também apresenta algumas boas práticas, como o uso de máquinas redundantes e ligações de rede estabelecidas com contingência.

A Figura 3 mostra trocas de mensagem na computação de uma transação eletrônica, bem como as mensagens relativas a escalonamento e notificação. Apesar de somente a MPI mostrar mensagens desses dois últimos tipos, o mesmo se repete para as demais MPs e foi abstraído do desenho para deixá-lo mais claro. Quanto aos canais de comunicação, GetLB contempla:

* Canal 1, de caráter bidirecional, para o fluxo de transações (envolvendo a chegada através dos terminais, o despacho, interação com os subsistemas e o retorno do resultado);

* Canal 2, de caráter unidirecional, para atividades de controle (envolvendo tráfego para atualizar dados de escalonamento e viabilizar as notificações).

No canal 1 trafegam dados relativos as transações. Todas elas possuem um cabeçalho, identificando o seu tipo (compra, saldo, cancelamento, entre outras), um número único e dados do terminal que a originou (com código do estabelecimento, tempo da saída e número

¹ Detalhes em: <https://www.pcisecuritystandards.org/documents/PCI%20SSC%20-%20Overview.pdf>

único do terminal). Quanto ao formato das mensagens do Canal 1, tanto a interação terminal para chaveador (ou comutador), quanto máquinas processadoras para sistemas de autorização acontecem no formato ISO 8583². Já, internamente a prestadora de serviço, usa-se arquivos XML para capturar os campos de forma mais rápida. Quanto ao retorno de uma transação, segundo norma ISO 8853, um terminal pode esperar no máximo 30 segundos. Caso a resposta não ocorra, dar-se-á um *timeout* e a operação é refeita. Desses 30 segundos, de acordo com a empresa GetNet, calcula-se que 2 segundos são estimados no tráfego de rede, enquanto que uma máquina processadora tem até 28 segundos para fazer o cômputo da transação. A mensagem de retorno possui o código do terminal, o código do estabelecimento, um timbre de data e hora e um número único que identifica a transação. Quando uma transação é realizada com sucesso, o terminal recebe o retorno do resultado da transação e sinaliza o chaveador para que ocorra uma desconexão do canal de comunicação.

O Canal 2 é uma inovação de GetLB para o tratamento de ambientes dinâmicos e heterogêneos. Essa última somente atua no sentido MP para comutador, onde cada MP periodicamente atualiza seus dados de escalonamento para o comutador, enquanto que as notificações são eventos assíncronos. Diferente de Round-Robin, cada MP precisa guardar e atualizar dados em sua memória referentes a CPU, memória principal e secundária, bem como tempos de entrada/saída para acesso ao comutador e subsistemas. Tais informações são passadas para o comutador que usa as mais recentes para inferir o desempenho de carga de uma máquina MP.

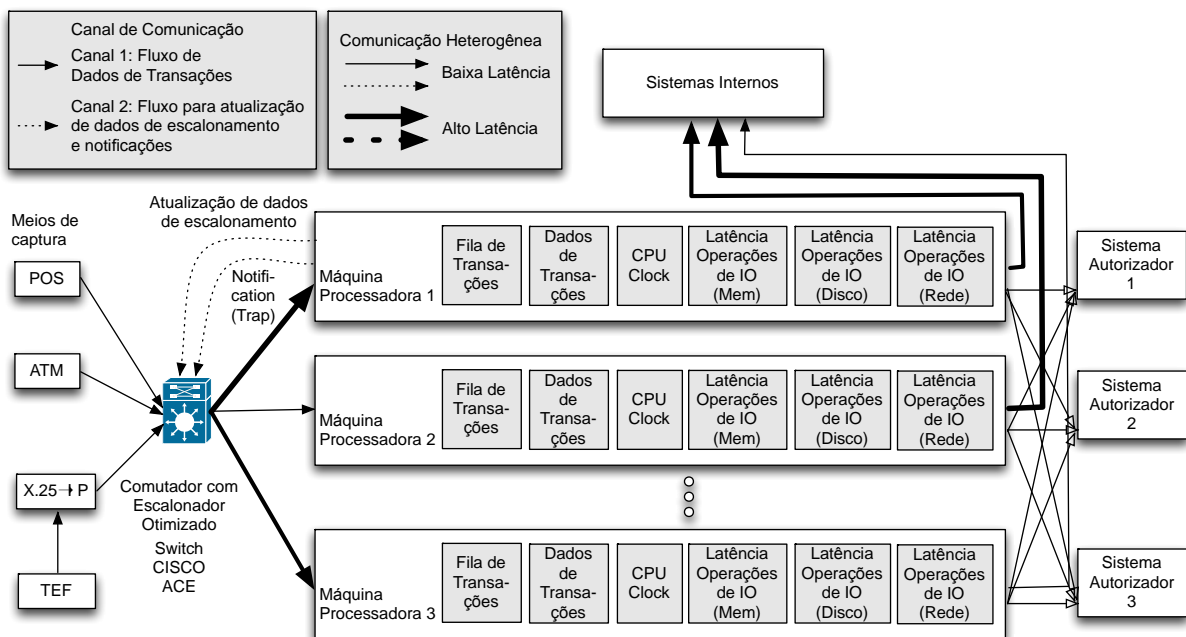


Figura 3. Arquitetura de comunicação de GetLB com desacoplamento de rede, máquinas MP potencialmente heterogêneas e interações MP-comutador para envio de dados de escalonamento e notificações.

² Detalhes em: http://www.iso.org/iso/catalogue_detail.htm?csnumber=31628

B. Escalonamento de Transações

Considerando que ambos os tipos de transações e máquinas MP podem ser modeladas como um sistema heterogêneo, foi desenvolvida uma heurística de escalonamento chamada LL (*Load Level*). LL pode ser vista como uma função de decisão $LL(i, j)$ com dois parâmetros de entrada: i representa um tipo de transação e j uma máquina MP candidata a processar a referida transação. Assim, o comutador irá calcular n equações LL para cada transação de entrada, onde n significa o número de máquinas MP. Neste caminho, o menor resultado informa a máquina que irá receber a transação analisada. $LL(i, j)$ pode ser obtido através da Equação 1.

$$LL(i, j) = Recv(i, j) + Proc(i, j) \quad (1)$$

$$Recv(i, j) = bytes(i) \cdot transfer(j) \quad (2)$$

$$Procv(i, j) = transaction(i, j) + \sum_{z=m}^{m-1} transaction(i, j) \quad (3)$$

$$transaction(i, j) = \frac{instructions(i)}{clock(j) \cdot (1 - load(j))} + \frac{RAM(i)}{serviceRAM(j) \cdot freeRAM(j)} + \frac{HD(i)}{serviceHD(j) \cdot freeHD(j)} + sub(i, j) \quad (4)$$

$$sub(i, j) = \sum_{y=0}^{x-1} [(2 \cdot sub_a(y, j)) + sub_t(j)] \cdot sub_r(i, y) \quad (5)$$

$LL(i, j)$ é obtido através do cálculo que leva em consideração o tempo de recepção e do próprio processamento da transação i na máquina j . O termo $Recv(i, j)$ considera o tempo final para transmitir todos os bytes da transação em questão desde o comutador até a MP. A Equação 2 mostra os dados necessários para a obtenção desse tempo, onde $transfer(j)$ denota o tempo para transferência de 1 *byte* do comutador para a máquina j . A medida obtida pela Equação 2 pode ter um impacto significativo na avaliação da Equação 1, uma vez que redes LAN e WAN podem ser consideradas.

Além do tempo para a transferência da transação i , $LL(i, j)$ também leva em consideração uma predição do tempo de processamento de todas as transações já mapeadas para j além do cômputo da própria transação i nessa máquina. GetLB trabalha com dados estáticos quanto aos requerimentos (CPU, memória, acesso a subsistemas internos) de cada tipo de transação, bem como dinâmicos que dizem respeito a tempos de comunicação e carga de CPU e E/S. Esses últimos são atualizados pelas máquinas MP para o comutador periodicamente. Na Equação 3, a variável m usada no somatório representa a quantidade de transações já mapeadas para a máquina j . A Equação 4 soma dados relativos ao uso da CPU, memória principal e secundária e tempo total gasto com subsistemas internos. Nessa equação, $serviceRAM(j)$ e $serviceHD(j)$ representam o tempo de latência de escrita (pior caso) para as operações de memória e disco na máquina j , respectivamente. $RAM(i)$ e $HD(i)$ tratam a quantidade de operações de memória e disco atrelados ao tipo da transação i .

A Equação 5 apresenta o tempo total despendido no acesso e processamento síncrono de subsistemas. Nessa equação, a variável x denota a quantidade de subsistemas que o tipo da transação i deve acessar. A Equação 5 é modelada com três subfunções: $sub_a(y, j)$, $sub_t(y)$ e $sub_r(i, y)$. O tempo de acesso ao subsistema y e o seu tempo de processamento são encontrados pelas equações $sub_a(y, j)$ e $sub_t(y)$, respectivamente. Em especial, a primeira é multiplicada por dois para que seja obtido o tempo de ida e volta de uma mensagem. $sub_r(i, y)$

denota a quantidade de requisições que é feita ao mesmo subsistema para o cômputo de uma transação. Os tipos de transações eletrônicas e os parâmetros para a Equação 5 podem ser capturados com a empresa prestadora de serviço que irá executar o protótipo de GetLB.

4. Protótipo

Para a avaliação de GetLB, foi desenvolvido um protótipo que emula o processamento de transações eletrônicas financeiras. No entendimento dos autores, o protótipo é pertinente para avaliar a parte de escalonamento (métricas usadas e combinação delas) do middleware proposto e fazer uma comparação com a abordagem tradicional para essa tarefa, que usa o algoritmo Round-Robin. A implementação do protótipo foi desenvolvida com o *middleware* RMI pertencente ao kit de desenvolvimento do Java, versão 1.7. Apesar dessa linguagem não ser voltada para alto desempenho, o seu emprego nesse contexto foi pertinente para avaliar os possíveis ganhos de desempenho e a sobrecarga de GetLB se comparado com a abordagem tradicional para processamento de transações eletrônicas. A ideia do arcabouço passa pela criação de objetos remotos tanto na máquina comutador quanto em cada MP. Tal organização está ilustrada na Figura 4. O comutador tem objetos remotos que guardam informações sobre cada uma das máquinas MP. Cada máquina MP, por sua vez, cria um objeto remoto para tratar o enfileiramento e processamento de transações. A máquina ACE possui procuradores (*proxies*) para os objetos remotos presentes nas máquinas MP e deve decidir qual deles acionar no momento da chegada de uma nova transação.

No comutador será criado um vetor de objetos remotos, os quais serão atualizados através de procuradores em cada uma das máquinas MP. Cada objeto remoto é responsável por guardar informações como CPU, disco, memória e tempo de rede sobre uma máquina MP em particular. Um novo fluxo de execução chamado *MachineThread* é criado no construtor na classe que define uma máquina MP. Ele tem o papel de periodicamente coletar dados atualizados do estado corrente da máquina e chamar os métodos de atualização do objeto remoto. Por fim, cada máquina MP implementa um mecanismo para consumir as transações que estão em sua fila de chegada. Tal consumo segue o regime FIFO (*First In First Out*). O protótipo desenvolvido não se comunica com sistemas de saída até o momento. Mesmo assim, ele é útil para analisar o mapeamento em relação a dados de rede, subsistemas e máquinas MP.

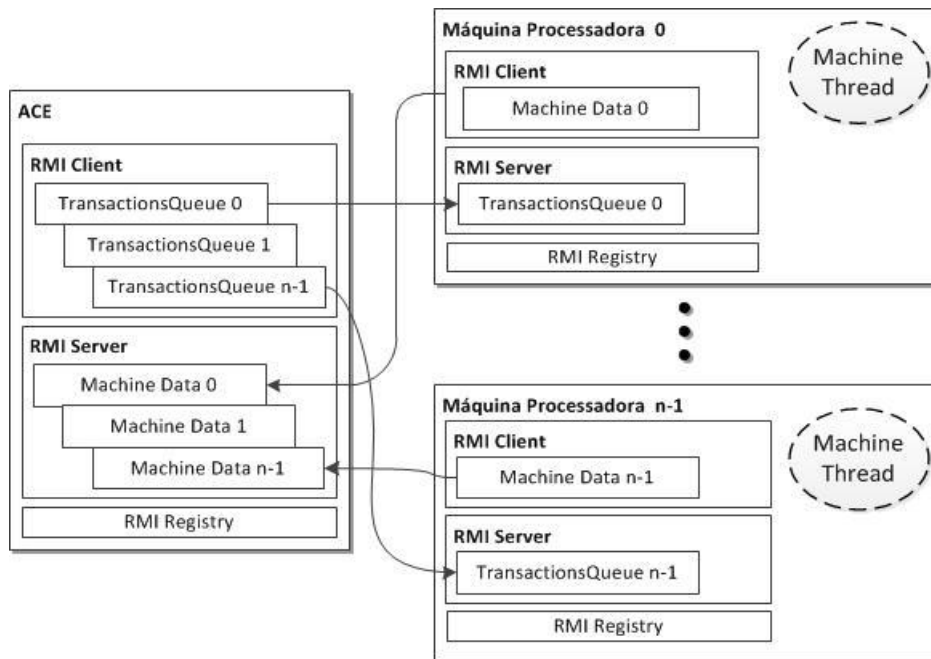


Figura 4. Protótipo GetLB em Java RMI.

Primeiramente, o computador recebe como entrada dois arquivos de texto: (i) com as transações a serem processadas; (ii) descrevendo as máquinas do *Cluster*. O arquivo com as transações possui o tipo da transação e o tempo em que esta transação chegou ao sistema. O arquivo que descreve as máquinas do *Cluster* possui o nome da máquina, o endereço IP, a frequência do processador, o total de memória e o tamanho máximo da fila. Além de RMI, o protótipo usa o retorno do comando `top` do Linux para recuperar informações sobre o percentual de CPU e memória livre nos computadores. A versão corrente não recupera informações sobre consumo de disco, sendo assim, a parte da equação que considera esta métrica não é calculada no protótipo. As máquinas MP reportam dados para o computador com uma frequência que é parametrizada. No protótipo, a implementação dos algoritmos GetLB e Round-Robin está no computador. As máquinas processadoras não possuem informação sobre o algoritmo de escalonamento em uso. Elas apenas recebem um objeto do tipo *Transaction* que possui as informações necessárias para emular a execução da transação.

5. Metodologia de Avaliação

A avaliação está focada na parte de escalonamento de transações do arcabouço GetLB, avaliando tanto a sobrecarga do cálculo do escalonamento quanto a qualidade do mapeamento transações-MPs no âmbito de RR (Round-Robin) e LL (Load Level). Para tal, utilizou-se 5 tipos de transações, cada qual com particularidades quanto ao uso de CPU e memória RAM, tamanho e tempo de acesso aos subsistemas internos. A Tabela 1 apresenta detalhes sobre cada um desses tipos e seus requerimentos. Os dados foram obtidos da empresa GetNet, que atua no ramo de processamento de transações eletrônicas. O tamanho da transação é a quantidade em bytes que trafega até computador e depois é despachada para uma máquina MP. Cada MP possui um repositório local que armazena a transação em si. Devido a necessidade de aplicação de filtros, necessários para converter dados no padrão ISO 8583 para o padrão XML aceito em sistemas internos, uma determinada área da RAM é ocupada para cada transação. Tal parâmetro é apresentado na terceira coluna da Tabela I. Esses dados são somente desalocados da memória no término da transação. O parâmetro instruções deve ser

multiplicado por 10^6 e corresponde à quantidade de ciclos de CPU necessários para execução de cada tipo de transação.

Tabela 1. Tipos de transações e suas características.

Tipo de transação	Tamanho médio (bytes)	Uso de memória RAM	Ciclos de máquinas (Mega)	Acesso aos subsistemas (milissegundos)	Descrição de subsistemas acessados
Recarga de telefonia	1146	2000	42	1150	Sistema de Recarga (3 acessos), Sistema de Prevenção à Fraude (1 acesso), Sistema de Gerenciamento (1 acesso)
Compra com cartão	1322	12300	107	1140	Sistema de Cartões (3 acessos), Sistema de Criptografia (1 acesso), Sistema de Prevenção à Fraude (1 acesso), Sistema de Gerenciamento (1 acesso)
Cancelamento de compra com cartão	1212	12250	102	1100	Sistema de Cartões (3 acessos), Sistema de Prevenção à Fraude (1 acesso), Sistema de Gerenciamento (1 acesso)
Reversão de transação	348	1140	83	400	Sistema de Cartões ou Sistema de Recarga (1 acesso), Sistema de Gerenciamento (1 acesso)
Administrativa	6700	7000	30	1350	Sistema de Configuração de Terminais (1 acesso), Sistema de Cartões (1 acesso), Sistema de Prevenção à Fraude (1 acesso)

Quanto maior a frequência de relógio de uma máquina, mais rápido o processamento da transação. Apesar da heurística de escalonamento de transações de GetLB trabalhar com os parâmetros tempo de acesso a subsistemas e número de chamadas e tempo de processamento de cada um, os dados fornecidos pela GetNet dizem respeito ao tempo de acesso (contando requisição e retorno) e processamento a todos os subsistemas necessários para a execução de um tipo de transação. Assim, o campo tempo total mostrado na Tabela 1 representa a função $sub(i, j)$ presente na Equação 4.

As transações usadas nos testes foram capturadas pela empresa GetNet, preservando dados sigilosos, números de cartão de crédito e valores. Para fins de escalonamento, simplesmente a sequência e o tipo delas são pertinentes para análise. Os dados foram obtidos no dia 23 de dezembro de 2011 no intervalo entre 13h30min e 18h30min, totalizando 8168 transações. Esse montante está classificado da seguinte forma: 1916 são de Compra, 41 são de Cancelamento de Compra, 36 são de Reversão de Transação, 5303 dizem respeito a Recarga de Telefonia e 872 são do tipo Administrativa. A Figura 5 mostra a ordem de chegada das 8168 transações.

Os testes contam com uma máquina que age no papel de comutador (Xeon 2.4 GHz) que despacha transações para um cluster de máquinas MP homogêneo ou heterogêneo. O homogêneo é formado por seis máquinas Xeon 2.4 GHz com 2 GB de RAM, enquanto que o heterogêneo possui quatro máquinas com essa configuração e outras duas com 1.2 GHz e 2 GB de RAM. A máquina que distribui as transações também está conectada ao mesmo chaveador usado em cada um dos clusters. Em ambos, o conjunto de transações foi avaliado com o protótipo GetLB e com o mecanismo de escalonamento padrão Round-Robin (RR). É importante salientar que o algoritmo RR também foi implementado em Java, versão 1.7.



Figura 5. Chegada e classificação por tipo das 8168 transações. Tipos: 1-Administrativa; 2-Recarga de telefonia; 3-Compra; 4-Cancelamento de compra; 5-Reversão de transação.

O algoritmo de despacho das transações está ilustrado na Figura 6. Essa sequência de passos é executada pelo comutador, que utiliza um arquivo de rastros que descreve uma transação em cada linha. A linha 17 do algoritmo aguarda pela conclusão de todas as transações em cada máquina. Cada máquina gera um tempo final de operações que é a diferença entre o seu tempo de término e o tempo t_1 capturado no início do programa. O tempo de conclusão da carga de trabalho (*workload*) é obtido na linha 18. Na literatura de escalonamento, tal medida também é conhecida como *makespan* [6]. O método despacha (linha 15) possui um caráter assíncrono, sendo responsável pelo envio de uma mensagem pela rede até a fila de entrada da máquina MP alvo. Por outro lado, o método apresentado na linha 17 é síncrono, ou seja, tem comportamento bloqueante até a conclusão de todas as transações. A diferença entre os tempos t_3 e t_2 informa o tempo gasto no algoritmo de escalonamento, RR ou LL. O algoritmo RR simplesmente captura a próxima máquina em uma fila circular, enquanto que LL usa o tipo da transação como parâmetro de entrada para escolher a máquina mais apropriada para processá-la.

```

01. t1 = captura_tempo();
02. Enquanto existirem transações
03. {
04.   transação = captura_transacao(arquivo);
05.   t2 = captura_tempo();
06.   se algoritmo for Round-Robin
07.   {
08.     maquina = captura_maquina_rr ();
09.   }
10.   senão
11.   {
12.     maquina = captura_maquina_ll (transação);
13.   }
14.   t3 = captura_tempo();
15.   despacha(transação, maquina);
16. }
17. vetor[] v = aguarda_conclusao_transacoes(t1);
18. makespan = captura_maior_tempo(v);

```

Figura 6. Algoritmo para despacho das transações no comutador e cálculo do tempo total para a computação das transações (*makespan* entre as linhas 01 e 18) e da sobrecarga na tomada do escalonamento (entre as linhas 05 e 14).

6. Análise dos Resultados

Os testes que mostram o tempo de funcionamento das máquinas MP e a distribuição das transações no cluster homogêneo estão apresentados na Tabela 2. O tempo apresentado nessa tabela está na notação horas: minutos: segundos e denota a diferença de tempo desde o início do programa até o processamento da última transação. Percebe-se um equilíbrio na distribuição das transações com RR, porém isso não significa melhor balanceamento sob a perspectiva de tempo de processamento. Apesar das máquinas serem homogêneas, o conjunto de transações avaliado não é, e RR as mapeia ciclicamente para os recursos sem observar as suas características. Tal situação foi ilustrada previamente na Figura 1(b). Já GetLB foi responsável pela obtenção de um tempo total de processamento da carga de trabalho, também chamado de *makespan*, menor que aquele obtido no RR. As medidas 00:05:21 e 00:06:26 são responsáveis por essa observação.

Tabela 2. Tempo de processamento e distribuição em máquinas homogêneas.

Máquina	GetLB			Round-Robin		
	Transações	Percentual de transações recebidas	Tempo total execução	Transações	Percentual de transações recebidas	Tempo total execução
M0	1470	18%	00:04:45	1362	16,66%	00:05:51
M1	1192	15%	00:04:47	1362	16,66%	00:05:52
M2	1467	18%	00:05:21	1361	16,66%	00:06:26
M3	1365	17%	00:04:46	1361	16,66%	00:05:52
M4	1377	17%	00:04:38	1361	16,66%	00:05:44
M5	1297	16%	00:04:06	1361	16,66%	00:05:27
	Total 8168	Total 100%	Maior 00:05:21	Total 8168	Total 100%	Maior 00:06:26

A Tabela 3 apresenta a relação de tempo de processamento gasto em cada máquina para a computação de um determinado tipo de transação ao usar GetLB. O balanceamento de carga atua de modo a equilibrar a soma de processamento efetuado em cada máquina. Ao analisar os 6 valores na última linha da tabela, chega-se a uma média de 228.78 segundos e um desvio padrão de 20.12. O mapeamento proposto por GetLB é eficaz ao ponto de amortizar o gasto para a computação do algoritmo para cada transação. Para cada uma, são sempre computadas n equações $LL(i, j)$, onde n é o número de máquinas MP candidatas, i denota a transação em questão e j uma máquina MP em particular. O menor índice LL informa a máquina que receberá a transação. Para cada transação, GetLB leva em média 6.96 milissegundos para o cálculo das equações, que é feito com dados locais atualizados periodicamente pelas máquinas MP, e escolha do alvo para despacho. Por outro lado, o algoritmo RR é computado em 1.12 milissegundos. O fato da computação do algoritmo RR ser 6 vezes mais rápida que a do LL é explicada pela sua simplicidade, que ao invés do cômputo de equação faz somente um deslocamento de uma posição numa fila circular de máquinas alvo. Com uma carga de trabalho de 8168 transações, gaste-se 9.14 segundos para escalonar as transações no algoritmo RR e 56.84 segundos para essa tarefa com o algoritmo LL. Entretanto, como já ilustrado na Figura 6, o lançamento das transações é um processo assíncrono em relação ao processamento delas.

Os resultados com o cluster heterogêneo são mostrados na Tabela 4. RR penaliza o tempo final para execução das transações porque parte delas são igualmente enviadas para máquinas com menor capacidade. Sendo assim, além do conjunto de transações ser caracterizado como um sistema heterogêneo, o fato dos recursos também serem degradados ainda mais o desempenho de RR. GetLB por sua vez, despacha mais de 90% das transações para as quatro máquinas com maior capacidade. Essa estratégia faz com que GetLB execute a carga de transações em 551 segundos, enquanto RR faz essa tarefa em 621 segundos. Apesar de mostrar um ganho de 11.27% em favor de GetLB, percebe-se que as máquinas MP0 e MP1 são subutilizadas. Em outras palavras, GetLB escolhe uma máquina mais poderosa com transações em fila a outra ociosa com menor capacidade de processamento.

Tabela 3. Tempo de processamento em segundos de cada máquina segundo a distribuição de GetLB de acordo como tipo de transação. Tipo 1 = Administrativa, Tipo 2 = Recarga, Tipo 3 = Compra com cartão, Tipo 4 = Cancelamento de compra, Tipo 5 = Reversão de transação.

Tipo	MP0	MP1	MP2	MP3	MP4	MP5
1	14,77	10,68	15,86	11,7	12,2	12,64
2	117,3	96,51	121,99	115,75	108,78	103,59
3	113,12	91,62	112,99	96,23	97,94	96,3
4	0,87	0,68	2,9	2,45	1,31	2,48
5	2,07	2,47	2,22	1,9	1,22	2,05
	Total 248,13	Total 201,96	Total 255,19	Total 228,03	Total 221,45	Total 217,06

Tabela 4. Tempo de processamento e distribuição de transações em máquinas heterogêneas.

Máquina	GetLB			Roud-Robin		
	Transações	Percentual de transações recebidas	Tempo total execução	Transações	Percentual de transações recebidas	Tempo total execução
M0	653	7,99%	00:09:10	1362	16,67%	00:10:20
M1	5	0,06%	00:09:12	1362	16,67%	00:10:22
M2	1787	21,88%	00:09:59	1361	16,66%	00:11:08
M3	1865	22,83%	00:09:08	1361	16,66%	00:10:18
M4	1940	23,75%	00:09:01	1361	16,66%	00:10:10
M5	1918	23,48%	00:09:11	1361	16,66%	00:10:21
	Total 8168	Total 100%	Maior 00:09:59	Total 8168	Total 100%	Maior 00:11:08

7. Trabalhos Relacionados

Atualmente, percebe-se que meios eletrônicos de pagamento são cada vez mais adotados, em detrimento do emprego de dinheiro em papel moeda ou cheque [7], [8]. Além de conveniência para consumidores, o uso de cartões eletrônicos beneficia instituições de comércio e facilita o acesso a aplicações e serviços na Internet. Virnes et al. [7] afirmam que essa transição aparece tanto nos bancos e sistemas de e-commerce, quanto na governança eletrônica, entretenimento,

sistemas de saúde e dispositivos móveis. Um dos tópicos mais estudados em sistemas de transação eletrônica diz respeito a segurança da informação [9], [10], [11]. Vishik et al. [11] apresentam que ambas transmissão segura de dados e relação de confiança devem ser reanalisadas na medida que sistemas embarcados e smartphones são cada vez mais utilizados como fontes de transações eletrônicas. Em particular, Sastre, Bacon e Herrero [9] discutem algoritmos de segurança otimizados para atender diferentes meios de transmissão, como ADSL e GPRS.

Sousa et al. [3] apresentam um modelo estocástico para avaliação de desempenho e planejamento de recursos em sistemas de transferência eletrônica de fundos (EFT). Estes autores fazem um estudo do desempenho levando em consideração características de dependabilidade como disponibilidade, confiabilidade, escalabilidade e segurança. Segundo eles, uma análise de um sistema EFT sem esses critérios pode levar a resultados imprecisos. Ainda, Sousa et al. relatam que os critérios mostrados anteriormente devem guiar o uso eficiente de recursos para que seja mantido o SLA (*Service Level Agreement*) com os clientes. Araújo et al. [1] afirmam que a análise de desempenho deve observar o pior caso de volume de chegada de transações para ser verossímil com a realidade da empresa de processamento de dados. Para tal, eles adotaram Redes de Petri e fazem uso de informações de acesso e armazenamento em disco, além do volume transacional.

A análise formal de desempenho é aplicada nos mais diversos segmentos da computação paralela e distribuída. Desnoyers et al. [12] desenvolveram um sistema chamado Modellus, que permite modelar de forma automática o uso de *datacenters* na Internet. Modellus usa teoria de filas para derivar modelos de predição do uso de recursos pelas aplicações. Seu diferencial está concentrado na combinação de dados de diversas aplicações para inferir sobre o estado de um *datacenter*. Nessa mesma linha, a teoria de filas também é aplicada a redes sem fio de sensores em [13]. Outro trabalho contempla a avaliação de estratégias para escalonamento em grades computacionais usando rastros reais de cargas de trabalho [14]. Análise formal permite avaliar a eficiência dos algoritmos e o tempo médio de espera para a conclusão de cada trabalho (*job*).

Luan et al. [15] trabalham com grades computacionais e migração de jobs. Eles propuseram um arcabouço matemático para modelar e analisar o tempo de migração. Assim como a abordagem feita no fluxo transacional da GetNet, o artigo desses autores também faz uma estimativa de atuação para o futuro. Eles observaram os possíveis ganhos com migração na execução de *jobs* com longa duração. A temática do pagamento móvel é discutida em [16]. Os autores desenvolveram uma plataforma baseada na camada SaaS (*Software as a Service*) da computação em nuvens para viabilizar os negócios. Esse trabalho comenta a importância do dimensionamento de sistemas de processamento de transação para evitar perdas e fidelizar clientes.

A análise de predição de desempenho de um ambiente EFT passa pela definição da taxa média de chegada de transações. Para tal, são coletadas amostras com uma periodicidade t que irão guiar o cálculo da taxa acima. O artigo escrito por Tchakian, Basu e Mahony [4] apresenta uma predição do fluxo de tráfego de veículos com base em séries temporais. Para tal, eles usam um intervalo de coleta de dado de 15 minutos. O mesmo passo não pode ser aplicado a sistema EFT, uma vez que pode desprezar um determinado pico de chegada de transações. O presente artigo leva essa análise em consideração e trabalha com intervalos de 1

segundo para a coleta de informações, tanto na chegada quanto no atendimento de transações eletrônicas.

8. Conclusão

Com o aumento do uso da Internet e melhorias dos recursos de comunicação nos últimos anos, houve um aumento exponencial do volume de transações eletrônicas que já preocupa especialistas na área. Essa análise vai ao encontro da era corrente de Big Data, que além do volume, também considera aspectos como a variedade e a velocidade de captura dos dados [17]. Alinhado a essa realidade, é necessário que as empresas prestadoras de serviço estejam preparadas para suportar o crescente fluxo transacional, uma vez que brasileiros adotam cada vez mais cartões de pagamento para efetuar suas compras e o que Brasil está numa ascendente no recebimento de turistas estrangeiros, que efetuam aqui suas transações, nos últimos 3 anos³. Nesse contexto, o presente artigo apresentou um modelo de balanceamento de carga para transações eletrônicas chamado GetLB. A **contribuição técnica** de GetLB está na organização de seu arcabouço, que considera tanto comunicação no sentido máquina MP para o comutador para o envio de dados de escalonamento, quanto a possibilidade de empregar diferentes redes para compor o parque de máquinas MP e subsistemas.

A **contribuição científica** de GetLB é a sua heurística de escalonamento chamada Load Level, ou LL. LL pode ser vista como uma função que possui o tipo da transação e uma máquina alvo como parâmetros. Para cada transação, são calculadas funções LL de acordo com o número de máquinas e o menor índice indica o alvo. O cálculo de LL considera tanto dados estáticos a respeito do tipo de cada transação, quanto dinâmicos a respeito da rede, CPU, memória e disco. Os testes foram executados sobre um fluxo de transações proveniente de rastros da empresa GetNet. Eles mostraram que, mesmo com um algoritmo mais custoso que RR, o mapeamento eficaz de LL é responsável por um melhor tempo final para a conclusão do conjunto de transações testado em ambos clusters homogêneo e heterogêneo. As contribuições de GetLB não estão limitadas ao contexto de sistemas transacionais, mas também podem ser empregadas para balanceamento de carga em sistema de comércio eletrônico, computação em nuvem e programação paralela.

Dado que o presente artigo apresentou uma avaliação da parte de escalonamento de GetLB quanto a heterogeneidade de recursos e transações, trabalhos futuros compreendem testes com dinamicidade de recursos e uso de notificações. Ainda, a avaliação foi feita de modo que as máquinas MP estivessem todas dentro do mesmo cluster. Nessa linha, outro trabalho futuro também consiste no uso de ambientes multi-cluster para a disposição das máquinas que processam as transações. Por fim, também almeja-se uma avaliação qualitativa quanto a dificuldade de implantação de GetLB na realidade da empresa GetNet, uma vez que essa tem interesse em usar a solução proposta para oferecer um melhor tempo de resposta para seus usuários.

Referências

[1] C. Araujo, E. Sousa, P. Maciel, F. Chicout, and E. Andrade, “Performance modeling for evaluation and planning of electronic funds transfer systems with bursty arrival traffic,” in *Intensive Applications and Services, INTENSIVE*, april 2009, pp. 65–70.

³ Dados da World Tourism Organization. Detalhes em <http://www2.unwto.org/>

- [2] H. Matbouli and Q. Gao, "An overview on web security threats and impact to e-commerce success," in Information Technology and e-Services (ICITeS), 2012 International Conference on, 2012, pp. 1–6.
- [3] E. Sousa, P. Maciel, C. Araujo, and F. Chicout, "Performability evaluation of eft systems for sla assurance," in Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on, 2009, pp. 1–8.
- [4] T. Tchakian, B. Basu, and M. O'Mahony, "Real-time traffic flow forecasting using spectral analysis," Intelligent Transportation Systems, IEEE Transactions on, vol. 13, no. 2, pp. 519–526, june 2012.
- [5] M. Tang, "Efficient and scalable monitoring and summarization of large probabilistic data," in Proceedings of the 2013 Sigmod/PODS, New York, NY, USA: ACM, 2013, pp. 61–66.
- [6] M. Maurer, I. Brandic, and R. Sakellariou, "Adaptive resource configuration for cloud infrastructure management," Future Generation Computer Systems, vol. 29, no. 2, pp. 472–487, 2013
- [7] J. Vines, M. Blythe, P. Dunphy, and A. Monk, "Eighty something: banking for the older old," in Proc. of the 25th BCS Conference on Human-Computer Interaction, British Computer Society, 2011, pp. 64–73.
- [8] L. Xiaojing, W. Weiqing, and Z. Liwei, "The mechanism analysis of the impact of ecommerce to the changing of economic growth mode," in Robotics and Applications (ISRA), 2012 IEEE Symposium on, 2012, pp. 698–700.
- [9] R. Sastre, S. Bascon, and F. Herrero, "New electronic funds transfer services over ip," in Electrotechnical Conference, 2006. MELECON 2006, 2006, pp. 733–736.
- [10] P. Seltsikas, G. Marsh, M. Frazier-McElveen, and T. J. Smedinghoff, "Secure government in cyberspace?" in Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, New York, NY, USA: ACM, 2011, pp. 359–361.
- [11] C. Vishik, A. Rajan, C. Ramming, D. Grawrock, and J. Walker, "Defining trust evidence: research directions," in Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, New York, NY, USA: ACM, 2011, pp. 66:1–66:1.
- [12] P. Desnoyers, T. Wood, P. Shenoy, R. Singh, S. Patil, and H. Vin, "Modellus: Automated modeling of complex internet data center applications," ACM Trans. Web, vol. 6, no. 2, pp. 8:1–8:29, 2012.
- [13] L. Berardinelli, V. Cortellessa, and S. Pace, "Modeling and analyzing performance of software for wireless sensor networks," in Proc. of the 2nd Workshop on Software Engineering for Sensor Network Applications, New York, NY, USA: ACM, 2011, pp. 13–18.

- [14] S. N. Mehmood Shah, A. K. B. Mahmood, and A. Oxley, “Analysis and evaluation of grid scheduling algorithms using real workload traces,” in Proc. of the International Conference on Management of Emergent Digital EcoSystems, New York, NY, USA: ACM, 2010, pp. 234–239.
- [15] F. Luan, M. Nygård, and T. Mestl, “A mathematical framework for modeling and analyzing migration time,” in Proceedings of the 10th annual joint conference on Digital libraries, JCDL '10. New York, NY, USA: ACM, 2010, pp.323–332.
- [16] L. Liu, M. Song, X. Luo, H. Bai, S. Wang, and J. Song, “An implementation of the online-payment platform based on saas,” in Web Society (SWS), 2010 IEEE 2nd Symposium on, aug. 2010, pp. 658 –662.
- [17] A. Abounaga and S. Babu, “Workload management for big data analytics,” in Proceedings of the 2013 international conference on Management of data, ser. SIGMOD '13. New York, NY, USA: ACM, 2013, pp. 929–932.