PROPOSTA PARA BALANCEAMENTO DE CARGA DINÂMICO EM SISTEMAS DE TRANSAÇÕES ELETRÔNICAS FINANCEIRAS

Alexandre Luis de Andrade, Cristiano André da Costa, Rodrigo da Rosa Righi

Programa de Pós-graduação em Computação Aplicada (PIPCA)
Universidade do Vale do Rio dos Sinos (UNISINOS)
São Leopoldo – RS – Brasil

alexandreluisandrade@gmail.com, rrrighi@unisinos.br, cac@unisinos.br

Resumo. *Este trabalho apresenta uma proposta de balanceamento de carga para um sistema de transações eletrônicas financeiras. Neste sistema, as transações são recebidas e distribuídas entre máquinas processadoras segundo o algoritmo de balanceamento Round Robin. Devido às limitações do sistema atual, com relação à heterogeneidade de carga e recursos, será proposta GetLB, uma infraestrutura que utiliza a heurística LL (Load Level). Testes preliminares em um protótipo de GetLB, apresentaram resultados com ganhos em disponibilidade e uso de recursos.*

1 Introdução

Transações eletrônicas financeiras, em inglês EFT (*Eletronic Funds Transfer*), representam uma realidade em expansão que impulsiona a aproximação entre consumidores e fornecedores. Nestes sistemas, as transações são recebidas de terminais de venda e transmitidas para um centro de processamento, que decodifica, executa e retorna os resultados no menor tempo possível (SOUSA et al., 2009). Em especial, o presente trabalho está enquadrado no dia-a-dia da empresa GetNet e no balanceamento de carga em um de seus sistemas de processamento, composto por uma série de subsistemas interdependentes que processam as transações.

Nesta arquitetura, as transações recebidas em um nó centralizador são distribuídas para Máquinas Processadoras (MP) segundo o algoritmo de balanceamento Round Robin (RR). Esse algoritmo é adequado para situações nas quais as MP são homogêneas e o tempo de rede entre elas e o nó centralizador também é uniforme. Contudo, essa configuração pode ser restritiva para um sistema transacional que precisa estar dimensionado para atender uma empresa em expansão geográfica com MPs em diferentes localizações. Em adição, a atuação do algoritmo RR possibilita que transações sejam transmitidas para MPs com que possuem alta carga de trabalho.

2 Proposta de arquitetura GetLB e escalonador LL

Dado o contexto de restrições do algoritmo RR e do crescimento do mercado de cartões, este trabalho apresenta uma nova arquitetura capaz de lidar com heterogeneidade de máquinas, de carga e também de rede. A arquitetura proposta, chamada GetLB, proporciona balanceamento de carga a partir de um chaveador, que usa dados das MP para efetuar o escalonamento. Os dados obtidos da monitoração são utilizados para cálculo do nível de carga (ou LL, *Load Level*) das máquinas, que corresponde à heurística para balanceamento de carga. Diferente do RR, para cada transação *i*, LL calcula *n* funções *LL(i, j)*, onde *j* é a MP alvo e *n* a quantidade das mesmas. Para tal, serão verificadas as condições de cada MP para atender à demanda conforme índices de ocupação de CPU, memória, disco e estado das filas, entre outros. O cálculo de LL é dado pelas Equações (1) a (4) apresentadas abaixo. A eficiente combinação desses fatores para o despacho de transações configura a contribuição científica do trabalho. Em adição, a arquitetura GetLB também se destaca por suportar notificações originadas das MP para o chaveador, que contribui para evitar *overhead* de comunicação através de eventos assíncronos, tais como parada ou perda de capacidade de processamento de alguma MP. Por fim, GetLB suporta que as MP estejam espalhadas pela Internet e utiliza o cálculo de latência para atingir cada uma na fórmula da heurística de escalonamento LL.

$$LL(i,j) = Receive(i,j) + Processing(i,j) \quad (1)$$

$$Receive(i,j) = n\_bytes(i) \,.\, time\_to\_transfer(j) \quad (2)$$

$$Processing(i,j) = trans\_time(i,j) + \sum_{z=0}^{m} trans\_time(z,j) \quad (3)$$

$$transac\_time(i,j) = \frac{n_{inst}(i)}{f_{clk}(j)} + \sum_{z=0}^{m}\left(t_{aces(z,j)} + t_{srv(z,j)}\right) + \left(HDio(i).t_{operHD(j)}\right) + \left(RAMio(i).t_{operRAM(j)}\right) \quad (4)$$

Como prova de conceito, desenvolveu-se um protótipo funcional de GetLB em Java RMI. Testes preliminares mostraram a viabilidade técnica da proposta, conforme se pode verificar na Figura 1, na qual está representada uma situação de enfileiramento de transações em uma MP, e quais os resultados obtidos ao aplicar o algoritmo RR e a heurística LL. Resultados demonstram que RR permite o envio de transações para uma MP com pouca capacidade de processamento até o ponto de ocorrer indisponibilidade. A mesma situação é contornada com uso de LL.

A versão final de GetLB será implementada em outro *middleware*, uma vez que RMI trabalha sobre TCP e faz uso de serialização e reflexão de Java, tidos como os processos mais custosos dessa linguagem. Assim, o próximo passo consiste em implementar GetLB com sistemas de comunicação mais eficientes, como SNMP (*Simple Network Management Protocol*) ou ainda, diretamente sobre mensagens UDP (SCHONWALDER, 2007).

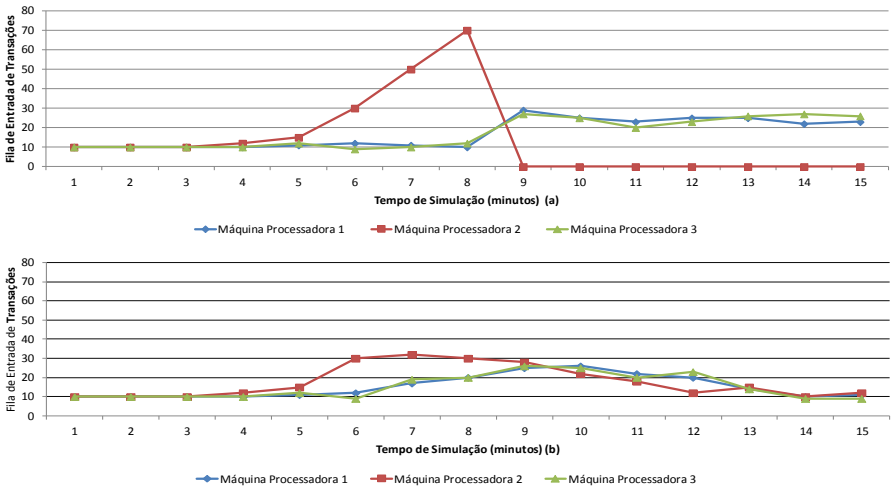

Figura 1. Ao alto (a) está demonstrado enfileiramento e a perda de transações em MP2 com balanceamento de carga RR. Abaixo (b) a mesma situação é contornada com uso de LL.

3 Conclusão

Com a infraestrutura GetLB, além das questões pertinentes ao balanceamento de carga, se obtém uma maneira eficaz de evitar perda de transações, visto que a configuração é tida como pró-ativa e não reativa quanto ao tratamento de congestionamento de transações. Nos testes preliminares realizados com o protótipo se verificou que a arquitetura resultante dessa abordagem de escalonamento melhora o compartilhamento de recursos, otimizando o tempo médio de resposta com eficiente balanceamento de cargas em um sistema escalável e de fácil manutenção.

4 Referências bibliográficas

SCHONWALDER, A. Pras J., HARVAN, J. Schippers M., e R. van de Meent, "SNMP Traffic Analysis: Approaches, Tools, and First Results", in 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007. IM 07, 2007, p. 323-332.

SOUSA, E.; MACIEL, P.; ARAUJO, C.; CHICOUT, F. Performability evaluation of EFT systems for SLA assurance. In: PARALLEL DISTRIBUTED PROCESSING, 2009. IPDPS 2009.