

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

ANÁLISE DE DESEMPENHO DA APLICAÇÃO DE BALANCEAMENTO DE CARGA EM SISTEMAS MULTIPROCESSADORES¹
PERFORMANCE ANALYSIS OF LOAD BALANCING APPLICATION IN MULTIPROCESSOR SYSTEMS

Vinicius Ribas Samuel Dos Santos², Edson Luiz Padoin³

¹ Trabalho apoiado por UNIJUI e CNPq. Pesquisa realizada no contexto do Laboratório Internacional Associado LICIA

² Aluno do curso de Ciência da Computação da Unijuí, Bolsista PIBIC/CNPq, vinicius.ribas@unijui.edu.br

³ Professor Orientador do Departamento de Ciências Exatas e Engenharias, padoin@unijui.edu.br

RESUMO

Este artigo apresenta as melhorias desenvolvidas no Balanceador de Carga AverageLB. Ele coleta informações tanto do sistema quanto da aplicação em tempo real e as utiliza nas decisões. Seu objetivo é reduzir o tempo total de execução da aplicação, bem como melhorar o número de migrações de processos. Os resultados apresentaram reduções significativas na taxa de migrações de processos em comparação com outros Balanceadores de Carga.

ABSTRACT

This article introduces the improvements developed in the AverageLB Load Balancer. It collects information from both the system and the real-time application and uses them in decisions. Its goal is to reduce the overall execution time of the application, as well as to improve the number of process migrations. The results showed significant reductions in the rate of process migrations compared to other Load Balancers.

INTRODUÇÃO

A necessidade de alto desempenho, proveniente de um crescimento da produção de software deu abertura para o desenvolvimento dos sistemas computacionais, surgidos para suprir essa demanda.

A modelagem de um problema complexo pode culminar em um desbalanceamento de carga e excessiva comunicação entre tarefas [Padoin et al. 2014]. Por isso, buscando utilizar de forma mais eficiente todos os recursos disponíveis pelos sistemas, diferentes balanceadores de carga tem sido aplicados em modelos computacionais. Balanceadores de Carga(BC), nome dado à estratégia, tem por objetivo detectar e corrigir o desbalanceamento de carga, aprimorando a utilização dos recursos disponíveis a aplicação [Padilha and Padoin 2016].

Este artigo apresenta as melhorias desenvolvidas no BC AverageLB. Ele coleta informações tanto do sistema quanto da aplicação em tempo real e as utiliza nas decisões. Seu objetivo é reduzir o número de migrações de processos uma vez que migrações de processos impactam no tempo total

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

de execução da aplicação. Os resultados apresentaram reduções significativas na taxa de migrações de objetos em comparação com outros Balanceadores de Carga

METODOLOGIA

Alguns ambientes de programação adotam uma metodologia baseada na medição das cargas dos objetos que executam em cada processador. Para isso, o BC coleta automaticamente estatísticas da carga computacional e da comunicação destes objetos e armazena estas informações em um banco de dados. Este banco de dados vai ajudar o BC a decidir quando e onde migrar os objetos [Jyothi et al. 2004].

Para realizar a análise de desempenho, dentre diversos balanceadores de carga com diferentes algoritmos, o C HARM ++ possui estratégias que trabalham tanto com as cargas das unidades de processamento, quanto com a comunicação entre os chares.

A estratégia para balanceamento de carga no algoritmo proposto, chamado de AverageLB, constitui-se de uma melhoria na estratégia utilizada no algoritmo GreedyLB. Essa melhoria busca equilibrar as cargas entre os processadores reduzindo o número de migrações. A implementação utiliza uma abordagem centralizada e busca atingir balanceamento levando em consideração a média aritmética das cargas de cada processador [Arruda et al. 2015].

O ambiente para validação da proposta possui um processador Intel modelo i7-6500U. Cada processador possui 4 núcleos com 2 SMT/núcleo, totalizando 8 núcleos. Para os testes, utilizou-se o sistema operacional Linux Manjaro 16.10 com kernel versão 4.4.33 – 1. A versão do C HARM ++ utilizada foi a 6.5.1 e do compilador g++ a versão 6.2.1. Os balanceadores de carga foram submetidos a simulações utilizando o benchmark LB Test. Esse benchmark é uma ferramenta bastante utilizada na avaliação de balanceadores de carga, sendo disponibilizado pelo próprio ambiente de programação, o C HARM ++. Além do LB test, foi utilizado o benchmark Neighbor para auxiliar na análise do desempenho dos balanceadores.

Em ambos os benchmarks, as simulações foram feitas com 50 e 100 processos, com sincronizações definidas a cada 10 iterações. O benchmark KNeighbor foi configurado para realizar simulações com 50 iterações por tarefa, já o LB Test foi configurados para realizar testes com 150 iterações. O benchmark LB test foi submetido a uma carga computacional variante entre 500ms e 8000ms.

Para análise, os resultados dos testes foram comparados com outros dois balanceadores de carga disponibilizados pela plataforma do C HARM ++, que são eles:

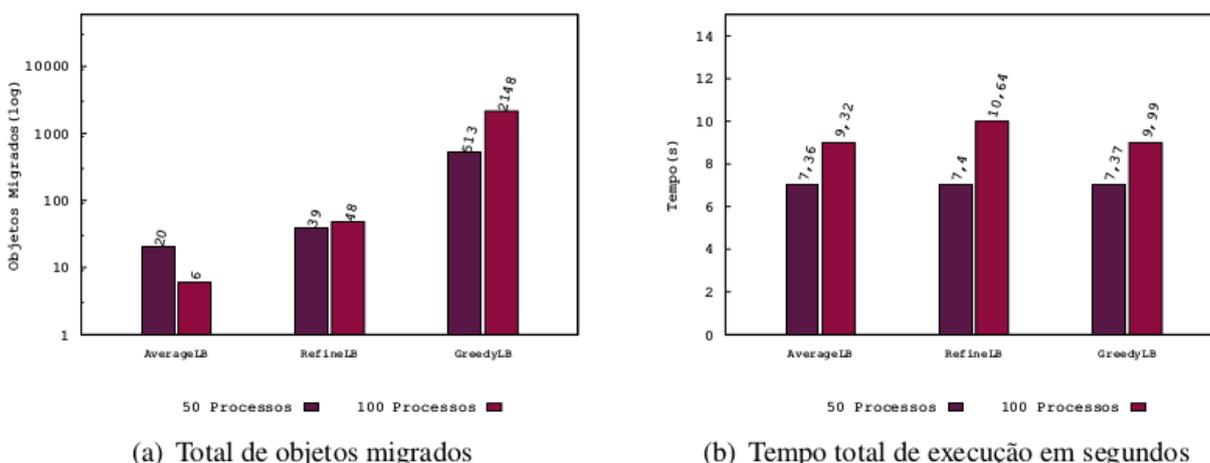
- RefineLB: Seu algoritmo move objetos dos processadores mais sobrecarregados para os menos carregados almejando atingir uma média, sendo limitado o número do objetos migrados [Freytag et al. 2015];
- GreedyLB: Possui um algoritmo de abordagem gulosa. Esse paradigma é frequentemente utilizado na teoria e na prática de otimização combinatória. Seu algoritmo objetiva migrar objetos pesados para o processador com menor carga. Isso se repete até que a carga de todos os processadores alcance uma proximidade com a carga média.

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

RESULTADOS E DISCUSSÃO

Durante os testes, foram levados em consideração o tempo total de execução e a quantidade total de objetos migrados. A Figura 1 mostra os gráficos resultante da simulação usando o benchmark LB Test.

Figura 1. Resultados dos testes utilizando benchmark LB.test



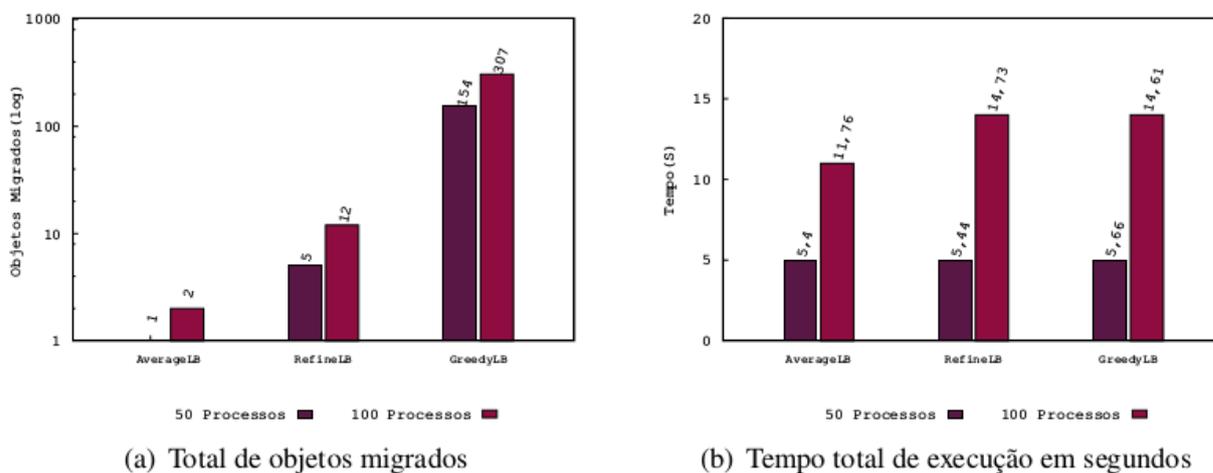
Com base nos tempos totais de execução mensurados, observa-se que tanto com 50 quanto com 100 processos, o menor tempo de execução é alcançado com BC AverageLB. Quando submetido aos testes com 50 processos, o AverageLB apresentou um tempo 0,54% menor que o algoritmo do BC RefineLB. Já com uma quantidade de 100 processos, a diferença aumentou para 14,16% em relação ao algoritmo do RefineLB, que em ambos os testes obteve o maior tempo.

Observando a quantidade total de objetos migrados, nota-se que, tanto com 50 como com 100 processos, o algoritmo do BC AverageLB apresenta uma menor quantidade de migrações. Com o BC GreedyLB foram migrados 513 e 2148 objetos nos testes com 50 e 100 processos. Já o algoritmo do AverageLB migrou apenas 20 e 6 objetos quando executado com 50 e 100 processos, respectivamente.

A disparidade na quantidade de objetos migrados pelos balanceadores AverageLB e RefineLB com o GreedyLB é consequência da sua estratégia gulosa que migra objetos do processador mais carregado para o menos carregado. Os algoritmos do RefineLB e do AverageLB fazem o cálculo da média de carga para determinar quais objetos devem ser migrados, evitando desperdícios e, conseqüentemente, diminuindo a quantidade de objetos migrados.

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

Figura 2. Resultados dos testes utilizando benchmark KNeighbor



Na Figura 2, que demonstra os resultados do benchmark KNeighbor, observa-se que, mais uma vez, o AVERAGE LB alcançou o menor tempo de execução, bem como também foi o BC que apresentou a menor quantidade de objetos migrados.

Baseado no tempo total de execução aferido nos gráficos da Figura 2, nota-se que, quando realizados os testes com 50 processos, o algoritmo do AverageLB obtém um tempo de execução de 4,81% menor que algoritmo do GreedyLB. Esse, por sua vez, foi o algoritmo com maior tempo de execução.

Com o aumento da quantidade de processos para 100, o algoritmo do AverageLB obteve uma atenuação no tempo de execução, sendo este tempo de 25,26% menor que tempo obtido pelo algoritmo do RefineLB, que marcou o maior tempo de execução.

Com base na quantidade de objetos migrados, o BC AverageLB mostrou um melhor desempenho em comparação com os outros dois balanceadores de carga disponibilizados pelo ambiente de programação. Quando realizados com 50 processos, os testes com o AverageLB migraram apenas 1 objeto. Já o BC GreedyLB migrou 154 objetos, diferença que se manteve quando realizados os testes com 100 processos, onde o AverageLB teve um total 2 objetos migrados, quando o GreedyLB por sua vez migrou 307.

CONSIDERAÇÕES FINAIS

Este artigo apresenta uma análise das melhorias aplicadas no algoritmo do BC AVERAGE LB, comparando seus resultados com os obtidos por outros dois balanceadores fornecidos pelo ambiente de programação C HARM ++. Foram-se utilizados três benchmarks para apresentar a análise de resultados.

Analisando os resultados dos testes em comparação com o RefineLB e o GreedyLB, concluímos

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

que AverageLB mostrou um melhor desempenho nos testes realizados com os dois benchmarks, tanto nas migrações de objetos quanto no tempo total de execução, mostrando que o BC está realizando um bom controle de objetos migrados, evitando desperdícios e colaborando para um balanceamento de carga ágil e preciso.

Como futuros trabalhos, pretende-se implementar o balanceador em sua forma completa, realizando testes em sistemas paralelos, utilizando benchmarks e problemas reais de computação científica, buscando comprovar o equilíbrio dos elementos de trade-off.

AGRADECIMENTOS

Pesquisa tem recebido recursos dos editais PIBIC/UNIJUI, PIBIC/CNPq, programa EU H2020 e do MCTI/RNP-Brasil sob o projeto HPC4E de número 689772.

PALAVRAS-CHAVE

Charm, HPC, Processamento paralelo, Benchmark.

KEYWORDS

Charm, HPC, Parallel processing, Benchmark.

REFERÊNCIAS

Arruda, G., Padoin, E. L., Pilla, L. L., Navaux, P. O. A., and Mehaut, J.-F. (2015). Proposta de balanceamento de carga para redução de migração de processos em ambientes multiprogramados. In XVI Simpósio de Sistemas Computacionais (WSCAD-WIC), pages 1-8, Florianópolis, RJ.

Freytag, G., Arruda, G., Martins, R. S., and Padoin, E. L. (2015). Análise de desempenho da paralelização do problema de caixeiro viajante. Jyothi, R., Lawlor, O. S., and Kalé, L. V. (2004). Debugging support for charm++. In Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, page 264. IEEE.

Padilha, B. S. and Padoin, E. L. (2016). Análise de desempenho da aplicação de balanceamento de carga em benchmark sintéticos. Salão do Conhecimento, 2(2).

Padoin, E. L., Castro, M. B., Pilla, L. L., Bozzetti, T. C., Navaux, P. O. A., and Méhaut, J.-F. (2014). Balanceamento de carga visando redução do consumo de energia para o modelo de programação charm++. XIV Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, Alegrete, RS, Brasil.