

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

BALANCEADOR DE CARGA PARA DIMINUIÇÃO DO CONSUMO DE ENERGIA DE SISTEMAS PARALELOS COM O MODELO DE PROGRAMAÇÃO CHARM++¹
PARALLEL SYSTEMS POWER SAVING ORIENTED LOAD BALANCEMENT WITH THE CHARM++ PROGRAMMING MODEL

Giovane Da Rosa Lizot², Edson Luiz Padoin³, Vinicius Mânica Mastella⁴, Vinícius Ribas Samuel Dos Santos⁵, Cleber Cristiano Sartorio⁶

¹ Trabalho parcialmente apoiado por UNIJUI e CNPq. Pesquisa realizada no contexto do Laboratório Internacional Associado LICIA e tem recebido recursos do edital da VRPGPE de bolsa e PIBIC/UNIJUI

² Aluno do curso de Ciência da Computação da Unijuí e Bolsista PIBIC/UNIJUI

³ Professor Orientador do Departamento de Ciências Exatas e Engenharias

⁴ Aluno do curso de Ciência da Computação da Unijuí e Bolsista PIBIC/UNIJUI,

⁵ Aluno do Curso de Ciência da Computação UNIJUI

⁶ Egresso do curso de Ciência da Computação UNIJUI

INTRODUÇÃO

Com o crescimento do número de processador nos atuais sistemas paralelos a busca por formas eficientes de utilização de todo o seu potencial passa ser eminente. Um dos problemas enfrentados é o desbalanceamento de carga gerado pelas aplicações paralelas. Da mesma forma, muitas pesquisas buscam uma melhor eficiência energética destes sistemas computacionais. Assim, diferentes centros de pesquisa visam o desenvolvimento de técnicas de concessão de desempenho para aplicações científicas aprimorando processamento e eficiência energética, sendo este segundo um desafio para ser estudado e proposto neste trabalho.

A nível de hardware, os processadores atuais permitem a redução da frequência por meio de estratégias de Dynamic Voltage and Frequency Scaling (DVFS). Já a nível de software, muitas estratégias de balanceamento de carga, como as implementadas com o modelo de programação Charm++, focam em reduzir o tempo de execução [ZHENG 2011].

O desbalanceamento de carga, um dos maiores culpados, ocasiona que as máquinas paralelas não consigam alcançar todo o aproveitamento desejado da estrutura que lhes é disponível.

O crescimento exponencial do consumo de energia na execução de soluções HPC [DONG 2010], é proporcional ao resultado da saturação da frequência de clock do processador [LE SUEUR 2010], tornando-se uma preocupação [PINHEIRO 2001]. Assim, um dos maiores limitantes na escalabilidade de futuros sistemas de HPC se dá pelo aumento no consumo energético dos sistemas, principalmente em decorrência dos avanços tecnológicos [PADOIN 2013].

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

METODOLOGIA

Para a implementação da proposta foram realizadas pesquisas bibliográficas sobre balanceadores de carga, aplicações paralelas e estratégias de DVFS. Dentre os balanceadores de carga pesquisados, selecionou-se os balanceadores de carga AverageLB [ARRUDA 2015] e EnergyLB [PADOIN 2014] para servirem de base para o desenvolvimento do novo balanceador. Adotou-se a plataforma de programação Charm++, extensão da linguagem C++ e propicia um ambiente de programação paralela orientada a objetos. Seu desenvolvimento foi feito pelo Laboratório de Programação Paralela da Universidade de Illinois, em 1993. O Charm++ oferece suporte a diversas plataformas, permitindo a execução de programas desenvolvidos neste modelo tanto em ambientes com memória compartilhada como também, distribuída.

O primeiro balanceador selecionado, AverageLB [ARRUDA 2015], tem como estratégia de balanceamento uma abordagem do tipo centralizada, o qual toma decisões em um único processo. A estratégia do algoritmo leva em consideração a média aritmética de cada processador, calculando suas cargas com o intuito de reduzir o número de migrações, buscando um equilíbrio entre as cargas [FREYTAG 2015]. Já o segundo, EnergyLB, utiliza a estrutura de balanceamento já disponível no sistema em tempo de execução [PADOIN 2014]. Ele é subdividido em duas abordagens de balanceamento, conforme a plataforma da aplicação. O algoritmo do Fine-Grained EnergyLB habilita a aplicação de balanceamento de carga, visando uma melhor distribuição de tarefas entre núcleos.

Mediante esses estudos, e com base nas informações adquiridas através da leitura sobre balanceadores de carga, uma proposta de um novo balanceador foi moldada, tendo o intuito de aprimorar a tomada de decisões do BC AverageLB. A ideia principal é aplicar técnicas de DVFS juntamente com migrações de tarefas e reduzir o tempo de execução e o consumo de energia.

Utilizou-se um equipamento com processador modelo Intel Core i5-3230M com 4 cores físicos para a realização dos testes e validar a proposta. O equipamento tem instalado sistema operacional Linux Ubuntu 16.04 com kernel versão 4.4.33-1, Charm++ versão 6.5.1 e compilador g++ versão 6.2.1.

Para a análise dos resultados alcançados com o balanceador proposto este foi comparado a outros dois balanceadores, estes disponibilizados pelo ambiente de programação do Charm++. Os quais são:

- RefineLB: Balanceador que possui abordagem centralizada e é baseado em refinamento. Esse BC move objetos dos core mais sobrecarregados para os menos carregados até atingir uma média, que é definida através de um método específico, limitando o número do objetos migrados [ARRUDA 2015].
- GreedyLB: É um algoritmo de balanceamento de carga guloso. Esse BC remove todas as tarefas de seus núcleos e as mapeia em ordem decrescente de carga entre os núcleos com as menores cargas [FREITAS 2016]. Essa estratégia migra objetos mais pesados para o processador com a

01 a 04 de outubro de 2018

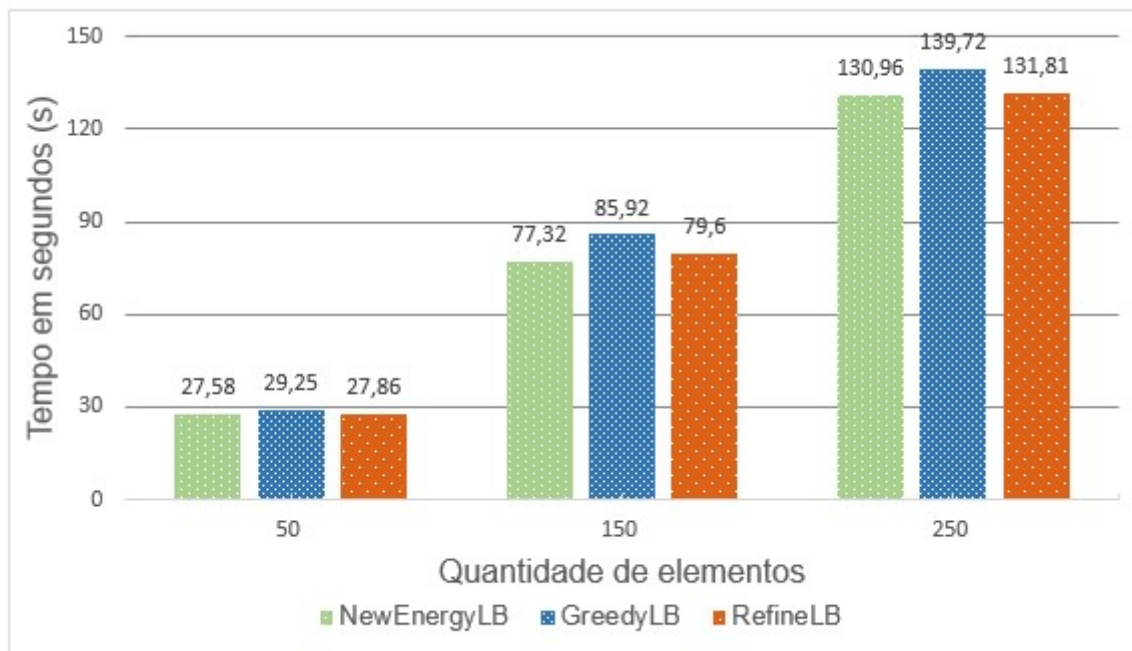
Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijui

menor carga, até que a carga de todos os processadores esteja próxima à carga média. Para analisar o desempenho do balanceador de carga proposto, testes foram realizados com o benchmark KNeighbor disponibilizado pelo ambiente do Charm++. O benchmark KNeighbor foi configurado com 150000 mensagens, 150 iterações por tarefa e sincronização a cada 10 iterações. Testes foram realizados com 50, 150 e 250 tarefas. Cada um dos testes realizados neste trabalho foi repetido 10 vezes, para se atingir um erro relativo menor que 5% e 95% de confiança estatística para uma Student's t-distribution. Entre cada um dos testes foi deixado o sistema em idle por, no mínimo, 20 segundos, de modo que a demanda de potência do sistema se estabilizasse.

RESULTADOS

Na Figura 1 são apresentados os resultados mensurados com o benchmark kNeighbor. Observa-se que o algoritmo do NewEnergyLB conseguiu menores tempos de execução em relação aos outros dois BCs. Percebe-se que, quando realizados os testes com 50 processos, o NewEnergyLB apresentou um tempo de 5,7% menor que o algoritmo dos BC GreedyLB e 1% menor que o RefineLB.

Figura 1. Resultado dos testes utilizando o benchmark Kneighbor.



Fonte: O Autor.

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijui

Com base nos tempos totais de execução mensurados no benchmark Kneighbor, exibidos na Figura 1, observa-se que tanto com 50, 150 e 250 processos, o menor tempo de execução é alcançado com o novo BC.

Com o aumento da quantidade de processos para 150, o algoritmo do NewEnergyLB obteve uma diferença de 10% em relação ao BC GreedyLB e 2,9% ao RefineLB. Com uma quantidade de 250 processos, a diferença foi de 6,3% e 0,6% em relação aos algoritmos do GreedyLB e RefineLB respectivamente, o que, conseqüentemente, otimiza o tempo de processamento equilibrando as cargas, ajustando a frequência e corrigindo a potência, o que reduz o consumo total de energia.

CONSIDERAÇÕES FINAIS

Neste artigo foi apresentado uma análise das melhorias aplicadas no algoritmo do BC NewEnergyLB que almeja reduzir o consumo de energia na execução de aplicações paralelas por meio do ajuste da frequência do clock de cada unidade de processamento em função de sua carga de trabalho. Analisando os resultados dos testes em comparação com o RefineLB e o GreedyLB, chegamos a conclusão que o BC proposto mostrou um melhor desempenho nos testes realizados com o benchmark Kneighbor. Como futuros trabalhos, pretende-se realizar testes em sistemas paralelos com um maior número de processadores, utilizando benchmarks e problemas reais de computação científica.

Palavras-chave

Charm++, Balanceamento de Carga, Consumo de Energia, Computação de Alto Desempenho.

Keywords

Charm++, Load Balance, Energy consumption, High Performance Computing.

Agradecimentos

Trabalho apoiado com recursos do programa do edital de bolsa PIBIC da UNIJUI.

REFERÊNCIAS

ARRUDA, G. H. S. (2015). **Balanceamento de carga em sistemas multiprocessadores utilizando o modelo de programação Charm++**. In Salão do Conhecimento.

DONG, Y., Chen, J., and Tang, T. (2010). **Power measurements and analyses of massive object storage system**. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, pages 1317-1322. IEEE.

FREITAS, V. M. C. T. de; PILLA, L. L (2016). **Um protótipo de algoritmo de balanceamento de carga guiado pelos próprios núcleos**. In ERAD-RS.

01 a 04 de outubro de 2018

Evento: Bolsistas de Iniciação Científica e Iniciação Tecnológica da Unijuí

FREYTAG, G., Arruda, G., Martins, R. S. M., and Padoin, E. L. (2015). **Análise de desempenho da paralelização do problema de caixeiro viajante**. In XV Escola Regional de Alto Desempenho (ERAD), pages 1-4, Gramado, RS. SBC.

LE SUEUR, E. and Heiser, G. (2010). **Dynamic voltage and frequency scaling: The laws of diminishing returns**. In Proceedings of the 2010 international conference on Power aware computing and systems.

PADOIN, E., Castro, M., Pilla, L., Navaux, P., and Mehaut, J.-F. (2014). **Saving energy by exploiting residual imbalances on iterative applications**. In High Performance Computing (HiPC), 2014 21st International Conference on.

PINHEIRO, E., Bianchini, R., Carrera, E. V., and Heath, T. (2001). **Load balancing and unbalancing for power and performance in cluster-based systems**. In Workshop on compilers and operating systems for low power, volume 180, pages 182-195. Barcelona, Spain.

ZHENG, G. et al. (2011). **Periodic hierarchical load balancing for large supercomputers**. In International Journal of High Performance Computing Applications, SAGE Publications, v. 25, n. 4, p. 371-385.