

Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

## **BALANCEAMENTO DE CARGA VISANDO A REDUÇÃO DO CONSUMO DE ENERGIA NO MODELO DE PROGRAMAÇÃO CHARM++<sup>1</sup> LOAD BALANCING TO REDUCE ENERGY CONSUMPTION IN CHARM++ PROGRAMMING MODEL**

**Ana Karina Morales Machado<sup>2</sup>, Edson Luiz Padoin<sup>3</sup>**

<sup>1</sup> 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.

<sup>2</sup> Bolsista PIBIC Unijuí

<sup>3</sup> Orientador do projeto de pesquisa.

**Resumo.** *O presente artigo expõe as considerações da proposta de desenvolvimento de um novo balanceador de cargas (BC) para o modelo de programação CHARM++. A política do BC consiste em ajustar a frequência de clock dos cores visando reduzir o consumo de energia dos sistemas.*

### **Introdução**

Sistemas High Performance Computing (HPC) buscam obter uma solução em tempos que a computação sequencial não consegue obter, tendo surgido da necessidade de computar simulações numéricas complexas [SINPAD]. Entretanto, como resultado da modelagem de problemas complexos envolvendo simulações com comportamentos dinâmicos e cálculos baseados em fórmulas complexas no modelo de programação paralela, temos o desbalanceamento de cargas [Arruda 2015].

O desbalanceamento de carga é um dos principais fatores que colocam em risco o desempenho de forma integral do processador [Arruda 2015], fazendo com que as máquinas paralelas não consigam obter o aproveitamento desejado da estrutura que lhes é disponível. Nesse ponto tem-se o balanceamento de carga como tópico importante, especialmente para aplicações que aumentam significativamente a quantidade de computações conforme a simulação evolui [Kale and Zheng 2009].

O crescimento exponencial do consumo de energia na execução de soluções HPC [Dong et al. 2010], resultado da saturação da frequência de clock do processador [Le Sueur and Heiser 2010], tornou o consumo de energia uma preocupação crítica [Pinheiro et al. 2001]. Soluções HPC tornaram-se um desafio para as pesquisas que visam o desenvolvimento de técnicas de concessão de desempenho para aplicações científicas, aprimorando a eficiência energética.

A nível de hardware, os processadores atuais permitem a redução da frequência por meio de estratégias de Dynamic Voltage and Frequency Scaling (DVFS) [Lara 2013]. Já a nível de software, muitas estratégias de BC, como as implementadas com o modelo de programação CHARM++, focam em reduzir o tempo de execução [Zheng et al. 2011].

### **Trabalhos Relacionados**

O Balanceador de Cargas (BC) AverageLB [Arruda 2015] se baseia em uma abordagem do tipo centralizada, em que um único processo, chamado mestre, mantém a árvore de tarefas a serem resolvidas e envia subárvores para os demais processos [SILVA]. O algoritmo foi projetado levando

## Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

em consideração a média aritmética das cargas de cada processador, na busca de equilíbrio entre elas, reduzindo o número de migrações [Freytag et al. 2015].

O Balanceador EnergyLB utiliza a estrutura de balanceamento já disponível no sistema em tempo de execução [Padoin et al. 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. Quando migrar uma tarefa torna-se uma tarefa onerosa, a técnica de ajuste do clock é adotada para alterar a frequência dos núcleos sobrecarregados [Machado et al.].

Já o algoritmo do Coarse-Grained EnergyLB se adequa a plataformas HPC distribuídas, aplicando uma visão hierárquica de árvore de dois níveis distribuindo a carga entre os núcleos dos processadores no nível das folhas. No nível raiz, o BC ajusta a frequência do clock de cada processador de acordo com a sua carga ponderada [Padoin et al. 2014].

### **Balanceador de carga proposto**

Uma vez visualizada a necessidade de remodelar os procedimentos atuais adotados para o processo de balanceamento de carga de aplicações HPC, buscando utilizar os métodos de consumo de energia, essa proposta pretende abster-se da metodologia de migração de tarefas, apenas utilizando técnicas DVFS para o balanceamento de carga e economia de energia.

Fazendo uso da política de média aritmética do BC AverageLB [Arruda 2015], a proposta consiste em utilizar informações de carga dos processadores disponibilizadas pelo ambiente de programação CHARM++. De posse destas informações, o BC pode atualizar a frequência das unidades de processamento de acordo com as suas respectivas cargas e a frequência disponível aos cores, reduzindo assim a demanda de potência média.

A utilização do ambiente de programação CHARM++ se justifica devido à disponibilidade de um framework de balanceamento de carga e o conjunto de funcionalidades para criação de novos balanceadores que a plataforma habilita [Padoin et al. 2014].

### **Metodologia**

Utilizando as informações que são disponibilizadas pelo ambiente de programação, além da média aritmética das cargas, que serve como ponto de equilíbrio de carga entre os processadores, o BC proposto, primeiramente, pondera a carga de cada unidade de processamento em função da sua frequência atual de relógio. Uma vez obtidas as cargas ponderadas, o BC maximiza a frequência do clock das unidades de processamento com maiores cargas, como exibido.

O próximo passo é reduzir a frequência das demais unidades de processamento de acordo com as suas cargas, não apresentando assim impacto no tempo de execução global da aplicação. O processo pode ser visto no código apresentado na Figura 1 e na Figura 2.

**Evento: XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA**

```
//verifica se o processador mais carregado está acima da média
if(pe_loads[highestProcessor] > mag) {

    //maximiza o clock do núcleo com maior carga
    stats->procs[highestProcessor].pe_speed = maxFreq;

    for (int j=0; j<stats->n_objs; j++) {
        int cpe = stats->from_proc[j]; // processo atual
        LObjData &odata = stats->objData[j]; //carrega dados do processo atual
        walltime = odata.wallTime;//carga do objeto

        if((pe_loads[lessProcessor] + walltime) < mag) {
            if (stats->to_proc[j] != lessProcessor) {

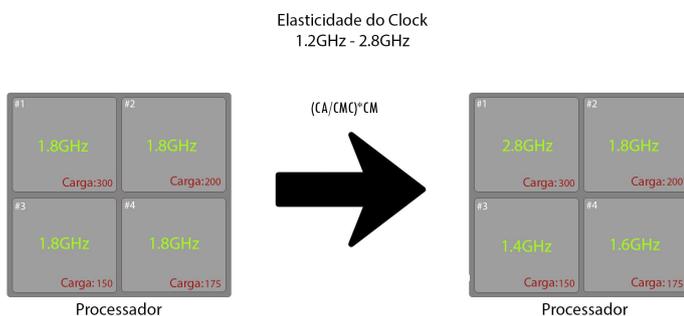
                stats->procs[j].pe_speed = pe_loads[j]/pe_loads[highestProcessor]*maxFreq;

                pe_loads[lessProcessor] += walltime;
                pe_loads[highestProcessor] -= walltime;
            }

            //se o processador com a maior carga passar a ser outro troca
            if(getProcessorWithHighestLoad(stats->nprocs(), pe_loads) != highestProcessor) {
                highestProcessor = getProcessorWithHighestLoad(stats->nprocs(), pe_loads);
            }

            //se o processador com a menor carga passar a ser outro troca
            if(getProcessorWithLessLoad(stats->nprocs(), pe_loads) != lessProcessor) {
                lessProcessor = getProcessorWithLessLoad(stats->nprocs(), pe_loads);
            }
        }
    }
}
```

**Figura 1. Metodologia de alteração do clock**



**Figura 2. Esquema de funcionamento do balanceamento de carga proposto**

Como mostra a Equação **Core\_Clock**, o valor obtido para a nova frequência de clock do núcleo é dada pela divisão do valor da carga atual do núcleo (CA), pela carga máxima encontrada entre os núcleos (CMC). O valor obtido é então multiplicado pela frequência máxima suportada pelo modelo do processador (CM).

$$\text{Core\_Clock} = \left( \frac{CA}{CMC} \right) * CM$$

**Conclusões**

Este artigo apresenta a proposta de um balanceador de cargas com intuito de reduzir a demanda do consumo de energia na execução de aplicações paralelas por meio do ajuste da frequência do clock em função da carga de cada unidade de processamento. Como futuros trabalhos, pretende-se realizar testes em sistemas paralelos, utilizando benchmarks e problemas

**Evento:** XXV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA

reais de computação científica, buscando comprovar o equilíbrio dos elementos de trade-off.

**Palavras-Chave**

CHARM++, Balanceamento de Carga, HPC

**Keywords**

CHARM++, Load Balancing, HPC

**Referências**

- Arruda, G. H. S. (2015). **Balanceamento de carga em sistemas multiprocessadores utilizando o modelo de programação charm++**.
- 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.
- 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.
- Kale, L. V. and Zheng, G. (2009). **Charm++ and ampi: Adaptive runtime strategies via migratable objects**. Advanced Computational Infrastructures for Parallel and Distributed Applications, pages 265-282.
- Lara, V. L. (2013). **Resolução paralela verificada de sistemas de equações lineares: uma abordagem para eficiência energética utilizando dvfs**.
- 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, pages 1-8.
- Machado, A. K. M., dos Santos, V. R., and Padoin, E. L. **Utilização de balanceamento de carga para redução do tempo de execução e do consumo de energia**.
- 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, pages 1-10.
- 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.
- SILVA, J. M. N. **Estratégias de balanceamento de carga para um algoritmo branch-and-bound paralelo para executar em grids computacionais**.
- SINPAD. **Sistema Nacional de Processamento de Alto Desempenho**. <https://www.lncc.br/sinapad/sinapad.php>. [Online; acessado em 26-Maio-2017].
- Zheng, G., Bhatelá, A., Meneses, E., and Kalé, L. V. (2011). **Periodic hierarchical load balancing for large supercomputers**. International Journal of High Performance Computing Applications, 25(4):371-385.