

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

## **PROPOSTA DE UMA ESTRATÉGIA DE PROGRAMAÇÃO EXPLÍCITA COM ANOTAÇÃO EM CÓDIGO EM BUSCA DE EFICIÊNCIA ENERGÉTICA<sup>1</sup>**

**Bruno Mokan Muenchen<sup>2</sup>, João Vicente Ferreira Lima<sup>3</sup>.**

<sup>1</sup> Projeto de pesquisa realizado pelo Programa de Pós-graduação em Informática da UFSM

<sup>2</sup> Aluno do Programa de Pós Graduação em Informática da Universidade Federal de Santa Maria

<sup>3</sup> Professor Doutor do Programa de Pós Graduação em Informática da Universidade Federal de Santa Maria

Este artigo apresenta uma proposta de estratégia de programação explícita, fazendo uso de diretivas, com o objetivo de identificar regiões de código onde a frequência de processamento pode ser reduzida, a fim de reduzir o consumo energético da aplicação sem que haja um grande impacto em seu desempenho.

### **Introdução**

O consumo energético de sistemas computacionais aumenta junto com o crescimento do poder computacional destes. Isso tem gerado um aumento no foco por eficiência energética em High Performance Computing (HPC), ao invés de um foco apenas em desempenho. Até alguns anos atrás, a demanda se dava exclusivamente pela construção de sistemas com desempenho cada vez maior, onde as indústrias competiam entre si em busca de criar processadores cada vez mais rápidos. Para vencer essa competição elas aumentavam a frequência de trabalho de seus produtos, até que chegaram aos limites que os processadores suportavam sem superaquecer, mas aumentando também o consumo de energia dos mesmos, desperdiçando esta energia em forma de energia térmica (MUENCHEN, 2013).

Com a crescente demanda por energia em HPC surgiram iniciativas que levam em conta reduzir este consumo, criando sistemas energeticamente eficientes, aumentando o desempenho e reduzindo o consumo (PADOIN, 2016). Por este motivo, processadores incorporam tecnologias como Dynamic Voltage and Frequency Scaling (DVFS), que torna possível reduzir a frequência do processador e, assim, diminuir sua demanda por energia.

Junto da redução de frequência e consequente redução de consumo do processador, vem a redução do desempenho do mesmo. Por este motivo, este trabalho apresenta a proposta de uma técnica de programação explícita, onde o desenvolvedor identifica regiões do código que podem ser executadas com uma frequência de clock menor, reduzindo o consumo energético.

### **Estado da Arte**

Existem diversos trabalhos relacionados a eficiência energética em HPC. Alguns deles apresentam estratégias de balanceamento de carga, outros apresentam técnicas de escalonamento com base em consumo energético.

Padoin (2016) apresenta uma estratégia de balanceamento de carga baseada no consumo de energia, chamada ENERGYLB, que considera características da plataforma, irregularidade e dinamicidade de carga das aplicações para melhorar a eficiência energética.

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

Liu, et al. (2011) propõe um algoritmo de escalonamento energeticamente eficiente que pode fazer uma melhor troca entre desempenho e consumo em clusters heterogêneos.

Yang, et al (2002) propõe uma técnica de mapeamento de tarefas concorrentes de uma forma energeticamente eficiente em plataformas heterogêneas onde o tempo de execução e consumo energético das tarefas são diferentes para cada processador, fazendo com que seja possível encontrar diversas possibilidades de ordenação das tarefas e de atribuição de processadores. Ainda, gera um conjunto Pareto-ótimo em que cada ponto é melhor que os outros em eficiência energética ou tempo de execução, ou em ambos, usando essa informação para economia de energia usando DVFS para encontrar uma solução global.

Em Liu, et al (1999) é discutida uma técnica DFVS per-core de economia de energia em sistemas embarcados. O método apresentado busca explorar operações memory-bound onde o dimensionamento de frequência tem um impacto menor. Além disso, são estudados os efeitos do tempo de transição da voltagem, sobrecarga e perdas do regulador on-chip nos benefícios obtidos no uso de DVFS.

Em Jin, Frumkin e Yan (1999) é mostrado um estudo da oportunidade de economia de energia em sistemas embarcados utilizando DFVS. O trabalho explora um número ótimo de níveis de voltagem para alcançar eficiência energética, levando em conta que um número grande de níveis causa sobrecarga, é mostrado que sistemas com 3 ou 4 níveis são tão eficientes quanto sistemas que podem variar a voltagem de forma arbitrária.

Millani (2015) apresenta uma metodologia de análise de desempenho para encontrar, em regiões paralelas, os melhores candidatos a redução do consumo energético, através da detecção em tempo de compilação das regiões que podem receber dimensionamento de frequência.

Embora existam diversos trabalhos focados na redução do consumo energético em HPC, este é o primeiro trabalho a propor técnicas explícitas de programação em busca da diminuição do consumo, transferindo ao programador a responsabilidade de identificar as regiões que poderão receber dimensionamento de frequência, assim diferenciando-se do apresentado por Millani (2015).

#### Referencial Teórico

Nesta seção abordaremos aspectos das tecnologias existentes que serão utilizadas no futuro. Podemos destacar a utilização de DVFS para dimensionamento de voltagem/frequência de execução do núcleo/processador, que será o alicerce deste projeto. Também falaremos sobre OpenMP, um modelo de programação paralela.

#### DVFS

DVFS é um framework que permite que voltagem e frequência de um processador ou de um core de um processador possam ser ajustados de acordo com sua carga (ROUNTREE et al, 2012). A redução da frequência proporciona economia de energia e menor dissipação térmica. Em outras palavras, ao ajustarmos o clock de um processador para um valor inferior, o mesmo necessitará de uma voltagem menor para trabalhar, reduzindo seu consumo.

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

O ajuste de frequência do DVFS é feito com a utilização do framework CPUFreq, que monitora a necessidade de desempenho do sistema para decidir se há necessidade de aumentar ou diminuir a frequência para reduzir o consumo ou o desperdício de energia. Em sua arquitetura o CPUFreq consiste de dois elementos: o governador, que é quem toma as decisões, e o driver, que é quem age de acordo com as decisões que foram tomadas (TEXAS INSTRUMENTS WIKI, 2016).

No Kernel do Linux, por exemplo, o DVFS possui 5 modos diferentes de operação, sendo que cada modo possui um governador que decide qual a frequência adequada de operação:

Performance: o processador opera na frequência máxima

PowerSave: o processador opera na menor frequência

UserSpace: o usuário escolhe uma das frequência disponíveis

Conservative: a frequência é definida de acordo com um limite superior e um limite inferior na carga do processador. Quando a carga excede o limite superior, a frequência é aumentada. Quando a carga não atinge o limite inferior, a frequência é reduzida.

OnDemand: a frequência é definida para um determinado limite, mas é aumentada ao máximo quando a carga do processador exceder esse limite.

OpenMP

OpenMP é um modelo de programação paralela desenhado para explorar certas características de arquiteturas de computação de memória compartilhada. Em outras palavras, OpenMP é um conjunto de diretivas passadas ao compilador e rotinas de tempo de execução que servem para expressar paralelismo em memória compartilhada (DAGUM, 1998). OpenMP proporciona um caminho incremental para paralelização de software, além de escalabilidade e performance para criação de novos softwares (DAGUM, 1998).

```
int main(int argc, char *argv[]){
    inicializa_matriz();
    #pragma omp parallel sections num_threads(2)
    {
        #pragma omp section
        {
            mergesort(0, MAX/2);
        }
        #pragma omp section
        {
            mergesort((MAX/2)+1, MAX-1);
        }
    }
    merge(A, 0, MAX/2, (MAX/2)+1, MAX-1);
    return 0;
}
```

Figura 1: Exemplo de código com OpenMP

Programas escritos com OpenMP iniciam sua execução como um único processo ou thread, chamada de master thread. A execução segue sequencial até que uma construção paralela seja

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

encontrada (como o par PARALLELL e END PARALLELL). Quando isto acontece, a thread master cria um novo conjunto de threads, que inclui ela mesma (JIN, FRUMKIN e YAN, 1999).

#### Apresentação da Proposta

A proposta deste trabalho consiste de uma extensão ao OpenMP similar ao OmpSS, que estende o mecanismo de tarefas do OpenMP para permitir a especificação de dependência entre tarefas e mapear a execução de certas tarefas para um determinado acelerador de hardware (AYGUADÉ et al, 2010).

A extensão proposta permite que o programador crie diferentes regiões paralelas dentro do código, variando seu nível de consumo de energia. Na Figura 2, observa-se o caminho crítico do algoritmo Cholesky usado por Gautier et al, 2013, em vermelho. Fora desta região, ou seja, fora do caminho crítico de execução da aplicação, as tarefas podem ser executadas com clock menor, semelhante a redução da prioridade da tarefa. Regiões de código que ficam dentro do caminho crítico executam com clock maior.

**Modalidade do trabalho:** Ensaio teórico  
**Evento:** XXI Jornada de Pesquisa

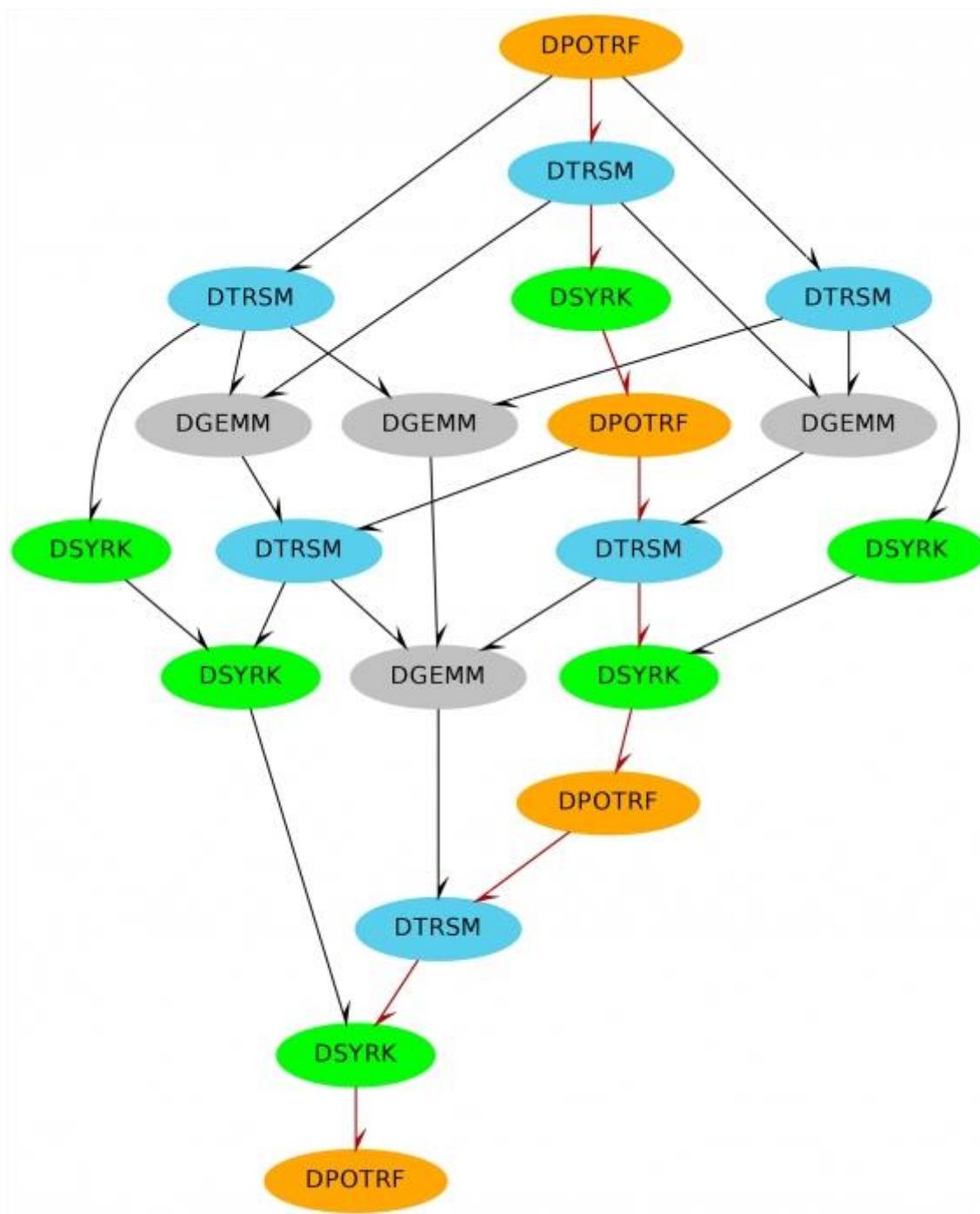


Figura 2: Caminho crítico do algoritmo Cholesky

Esta variação do nível de consumo energético é indicada de forma explícita pelo programador, sendo uma extensão dos construtores do OpenMP, fazendo com que seja possível indicar qual a

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

frequência de cada região paralela através de anotações no código fonte que, posteriormente, serão identificadas em tempo de compilação.

**Resultados Esperados e Trabalhos Futuros**

Esperamos obter, como resultado deste trabalho, uma extensão ao OpenMP que possa ser utilizada para a redução do consumo energético em HPC, através da redução ou aumento do clock do processador de acordo com as anotações de código feitas pelo programador.

Entretanto, na medida em que o clock é reduzido, também há uma redução no desempenho, o que faz com que o tempo de execução de aplicações aumente. Desta forma, a redução de energia na execução de uma aplicação dependerá muito da própria aplicação (ROUNTREE et al, 2012)

Em trabalhos futuros pretendemos definir um cronograma para execução das atividades relacionadas ao desenvolvimento desta extensão, além de definição de algoritmos de benchmark para realização de testes para verificar se a solução apresentada de fato apresenta melhorias no consumo energético.

**Referências Bibliográficas**

AYGUADÉ, E. et al. Extending OpenMP to survive the heterogeneous multi-core era. *International Journal of Parallel Programming*, v. 38, n. 5-6, p. 440–459, 2010.

DAGUM, L.; MENON, R. OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, v. 5, n. 1, p. 46–55, 1998.

GAUTIER, T. et al. XKaapi: A runtime system for data-flow task programming on heterogeneous architectures. *Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium, IPDPS 2013*, p. 1299–1308, 2013.

HUA, S.; QU, G. Approaching the Maximum Energy Saving on Embedded Systems with Multiple Voltages. *International Conference on Computer-Aided Design (ICCAD)*, p. 26–29, 2003.

JIN, H.; FRUMKIN, M.; YAN, J. The OpenMP Implementation of NAS Parallel Benchmarks and Its Performance. n. October, p. 1–26, 1999.

KIM, W. et al. System level analysis of fast, per-core DVFS using on-chip switching regulators. *Proceedings - International Symposium on High-Performance Computer Architecture*, p. 123–134, 2008.

LIU, W. et al. Energy-Aware Task Clustering Scheduling Algorithm for Heterogeneous Clusters. v. 1, p. 34–37, 2011.

MILLANI, L. F. G. A Performance Evaluation Methodology to Find the Best Parallel Regions to Reduce Energy Consumption. n. November, 2015.

**Modalidade do trabalho:** Ensaio teórico

**Evento:** XXI Jornada de Pesquisa

MUENCHEN, B. M. GPGPU : Comparação de aceleradores AMD , NVIDIA e INTEL utilizando a biblioteca OpenCL. 2013.

PADOIN, E. L. Energy-aware load balancing approaches to improve energy efficiency on HPC systems. n. February, 2016. .

ROUNTREE, B. et al. Beyond DVFS: A first look at performance under a hardware-enforced power bound. Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2012, p. 947–953, 2012.

TEXAS INSTRUMENTS WIKI. DVFS User Guide. Disponível em: <[http://processors.wiki.ti.com/index.php/DVFS\\_User\\_Guide](http://processors.wiki.ti.com/index.php/DVFS_User_Guide)>. Acesso em: 29 jun. 2016.

YANG, P. et al. Managing Dynamic Concurrent Tasks in Embedded Real-Time Multimedia Systems. p. 112–119, 2002.