GERENCIAMENTO DE ENERGIA EM AMBIENTE DE CLUSTER DE ALTO PROCESSAMENTO

Charles Moacir Pauletto Sartori[1], Carlos Oberdan Rolim[2]

Universidade Regional Integrada do Alto Uruguai e das Missões URI – Campus de Santo Ângelo

{charles.sartori[1],oberdan[2]}@gmail.com

1 INTRODUÇÃO

Em um escopo de cluster de alto processamento podemos encontrar bibliotecas paralelas, como o MPI. Essa biblioteca tem como objetivo a comunicação dos nodos do cluster através da troca de mensagem usando operações coletivas. Em um cluster de alto desempenho, os processadores geralmente possuem o mecanismo Dynamic Voltage and Frequency Scaling (DVFS), assim sendo possível alterar a voltagem e freqüência do processador dinamicamente. DVFS é uma das melhores formas de reduzir o consumo de energia, pode ser aplicada com bastante eficiência em três pontos críticos de um sistema paralelo que faz uso das bibliotecas MPI, slack time (acontece quando o programa paralelo precisa esperar por uma sincronização), collective operations (estado onde a troca de mensagem entre os nodos pela rede pode ocupar uma grande quantidade de tempo) ou ociosidade do sistema, ou seja, quando os nodos do cluster estiverem sem dados a ser processados.

Mesmo que o estudo e uso de DVFS em cluster de alto processamento seja uma área relativamente nova, temos encontrado grandes avanços, muitos pesquisadores buscam desenvolver técnicas para melhorar o gasto de energia. Este trabalho tem como objetivo apresentar um método para redução de energia, fazendo uso de DVFS em um ambiente de cluster de alto processamento utilizando a biblioteca MPI, onde o programador será o responsável por fazer as chamadas às funções para dinamicamente alterar a freqüência da CPU, estas chamadas serão feitas dentro do próprio algoritmo paralelo.

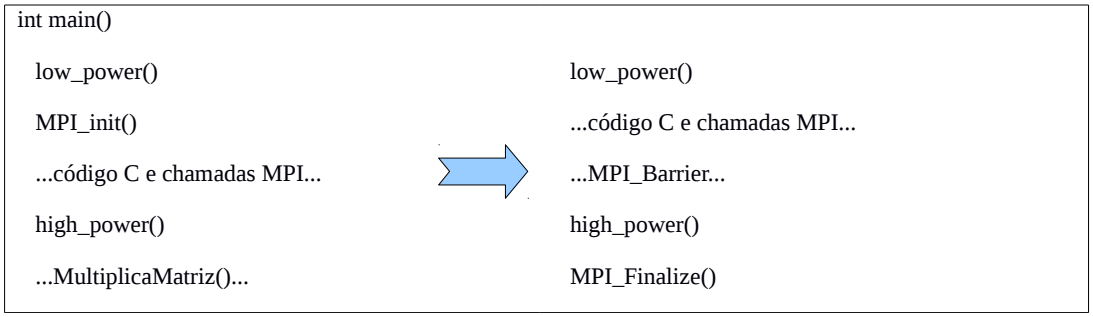
2 SOLUÇÃO PROPOSTA

A maioria dos micro processadores modernos voltados para alto desempenho possuem o mecanismo de dynamic voltage and frequency scaling (DVFS) para o controle de consumo de energia em tempo de

execução[10]. Nesse trabalho foi utilizado cpufreq para o controle da frequência da CPU. Cpufreq é uma referência a infra-estrutura do kernel linux responsável por implementar o controle da frequência da CPU. Outros autores como [3] e [4] usam o cpufreq em seus trabalhos.

2.1 USANDO DVFS PARA REDUZIR ENERGIA

Nesse trabalho foi escolhido um algoritmo de multiplicação de matrizes[11] para os testes. O algoritmo possue algumas funções MPI como: MPI_Bcast, MPI_Send, MPI_Recv e MPI_Barrier. A abordagem usada neste algoritmo foi a seguinte:



2.2 O Método

Como podemos ver acima temos dois estados possíveis, high_power(frequência máxima) e low_power(frequência mínima), não usamos frequências intermediárias em nenhum dos testes realizados. Várias abordagens poderiam ser feitas, mas optamos por usar a frequência máxima somente na parte crítica do algoritmo, que seria a própria multiplicação.

Cada estado (high_power e low_power) são funções C que fazem parte do algoritmo MPI, essas funções são responsáveis a fazer a chamadas a rotinas do cpufreq. No caso da função high_power(), **cpufreq-set -g perfromance** (essa rotina seta a cpu na máxima frequência), na função low_power(), **cpufreq-set -g powersave** (essa rotina seta a cpu na frequência mínima).

3 TESTES

Para realização dos testes utilizamos três computadores reais, sendo dois deles notebooks. A seguir a configuração de cada um:

Notebook: loiva-note	Notebook: gabriel-note	Desktop: pexe-pc
Modelo: AMD Turion 64 x2 TL-58 RAM: 2 GB DDR2 CPU MHz: 1900	Modelo: AMD Turion x2 RM-74 RAM: 2 GB DDR2 CPU MHz: 2200	Modelo: AMD Phenom II X4 955 Processor RAM: 8GB DDR3 CPU MHz: 3200

Tabela 1. Configuração dos computadores

Em cada computador acima foi utilizado o mesmo sistema operacional e versão do Mpich, Ubuntu 10.04 e mpich2-1.4 respectivamente, para os testes e projecão dos gráficos e demais resultados foi desenvolvido uma ferramenta que chamamos de CylSOS.

3.1 TESTE DE CONTROLE DA FREQUÊNCIA E VOLTAGEM

Executamos o algoritmo de multiplicação de matrizes que vimos acima várias vezes com tamanhos de matrizes diferentes, no primeiro teste setamos uma matriz de 1000x1000, segundo teste 1500x1500, terceiro teste 2000x2000 e por ultimo 3000x3000.

Para os testes de voltagem assumimos que a voltagem mínima e máxima de cada processador, de acordo com o site do fabricante, seria equivalente a frequẽncia mínima e máxima, como podemos ver na tabela abaixo:

	loiva-note	gabriel-note	pexe-pc
Frequência máxima	1900 Mhz	2200 MHz	3200 MHz
Frequência mínima	800 MHz	550 MHz	800 Mhz
Voltagem máxima	1.125 V	1.200 V	1.425 V
Voltagem mínima	1.075 V	0.750 V	0.825 V

Tabela 2. Frequência/voltagem mínima e máxima.

1) Primeiro Teste: matriz 1000x1000

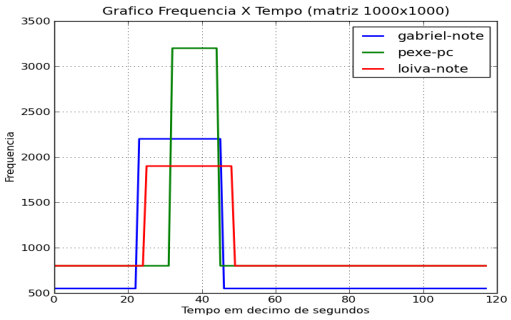


Figura 10. Gráfico de frequência

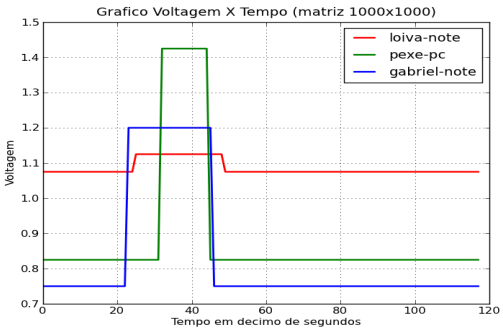


Figura 11. Gráfico de voltagem

Nos gráficos acima, podemos observar a variação da frequência e da voltagem. Vemos também que a variação de voltagem do computador loiva-note é muito inferior aos demais, com essa baixa variação de voltagem a enconomia de energia do computador loiva-note será muito menor comparado aos outros computadores, consequentemente a economia de energia total também sera prejudicada.

4 RESULTADOS

Essa sessão descreve os resultados dos nossos experimentos usando DVFS em um cluster real a partir de um algoritmo MPI de multiplicação de matrizes. Com todos os testes feitos, temos os seguintes resultados para cada multiplicação de matriz:

MATRIZ	TEMPO ANTES (s)	TEMPO DEPOIS (s)	DESEMPENHO (%)	ENERGIA SALVA (%)
1000*1000	8.08	9.14	11.5973741794311	24.96
1500*1500	28.07	32.31	13.122872175797	24.18
2000*2000	57.55	66.85	13.9117427075542	22.78
3000*3000	192.67	228.66	15.7395259337007	18.68

Tabela 3. Tabela de desempenho de energia salva.

Na tabela acima podemos observar que a perda no desempenho aumenta conforme o tamanho das matrizes, isso acontece principalmente por que consideramos como área crítica(área onde a frequênncia está no máximo) do algoritmo somente a multiplicação das matrizes, e não por exemplo a inicialização delas. Observamos também que a energia salva diminue confirme aumenta o tamanho das matrizes, quanto maior for as matrizes, a fatia de tempo necessária para a multiplicação será maior.

5 CONCLUSÃO

Esse trabalho propos um método para controle de energia utilizando DVFS considerando um cluster real, chegamos a ter entre 18% e 24% de economia de energia em nossos testes, essa economia poderia ter sido ainda maior caso a variação de voltagem do computador loiva-note fosse maior, mas como penalidade tivemos entre 11% e 15% de aumento no tempo total de execução do algoritmo MPI. Considerando um ambiente de cluster de alto processamento essa queda no desempenho pode não ser admissível, já em sistemas onde o tempo para execução não é o principal fator determinante, o método proposto pode ser viável.

6 REFERÊNCIAS

[1] DONG, Young; CHEN, Juan; YANG, Xuejun; YANG, Canqun; PENG, Lin, **Low Power Optimization for MPI Collective Operations,** The 9th Proceedings Conference for Young Computer Science, 2008.

[2] LIM, Min; FREEH, Vincent; LOWENTHAL, David, **Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs,** Proceedings of the 2006 ACM/IEEE SC|06 Conference, 2006.

[3] WANG, Shuen-Tai; LI, Chin-Hung; SHEN, Cherng-Yeu, **An Efficient Approach for Reducing Power Consumption in a Production-Run Cluster,** Third International Joint Conference on Computational Science and Optimization, 2010.