

# Análise de Desempenho e do Consumo Energético em MPI e OpenMP para Aplicações do NAS Parallel Benchmarks em uma Arquitetura com Memória Compartilhada

Lucas R. de Araujo<sup>1</sup>, Andrea S. Charão<sup>1</sup>, João Vicente F. Lima<sup>1</sup>

<sup>1</sup>Laboratório de Sistemas de Computação  
Universidade Federal de Santa Maria

{lraraujo, andrea, jvlima}@inf.ufsm.br

**Resumo.** *Este trabalho apresenta análises de tempo e de consumo energético de aplicações do NAS Parallel Benchmarks. A análise utiliza duas implementações de cada aplicação, MPI e OpenMP, que são executadas em uma arquitetura de memória compartilhada. Por fim, as aplicações têm o desempenho avaliado e comparado através dos dados obtidos. De maneira geral, os benchmarks desenvolvidos em MPI apresentaram os melhores resultados na comparação.*

## 1. Introdução

Em computação de alto desempenho, a escolha de modelos adequados para comunicação inter/intra processos é determinante para aproveitar o paralelismo da arquitetura disponível. Implementações dos padrões MPI e OpenMP possibilitam diferentes modelos de comunicação: o primeiro através de troca de mensagens e o segundo através de memória compartilhada entre as *threads*. Ao longo do tempo, muitos trabalhos já se dedicaram a investigar comparativamente o desempenho destes modelos, sobre diferentes gerações de arquiteturas [Krawezik e Cappello 2003, Utrera et al. 2015].

Outro aspecto crítico que tem acompanhado investigações de desempenho é o consumo e a eficiência energética, dado seu impacto sobre custos e sobre o meio ambiente. Considerando esses fatores, a disponibilidade de *benchmarks* como o NPB (*NAS Parallel Benchmarks*) possibilita a comparação entre MPI e OpenMP através de aplicações que realizam comunicação de diferentes formas e que disponibilizam métricas de desempenho ao final da execução. Além disso, com relação à eficiência energética, as arquiteturas mais recentes possuem contadores de *hardware* para obtenção de métricas relacionadas ao consumo energético durante a execução das aplicações.

Dessa forma, dada uma arquitetura com memória compartilhada, o presente trabalho tem como objetivo avaliar os tempos de execução de *benchmarks* que necessitam de comunicação durante o processamento, considerando duas implementações diferentes: MPI e OpenMP. Além disso, busca-se avaliar métricas de energia e potência consumidas durante a execução das aplicações. Com isso, procura-se conhecer os casos em que uma implementação apresenta melhores resultados que outra, para a arquitetura em questão.

## 2. Trabalhos Relacionados

Diversos trabalhos na área utilizaram o NPB para análises de desempenho. Um dos trabalhos primordiais na análise comparativa de MPI e OpenMP [Krawezik e Cappello 2003] também utilizou *benchmarks* do NPB em um ambiente de memória compartilhada para

tal comparação. O trabalho apresenta variações em torno da implementação OpenMP, porém em arquiteturas mais antigas.

Ao tratar de eficiência energética em computação de alto desempenho, há um crescente número de trabalhos desenvolvidos. Além das análises, há um trabalho [Schöne et al. 2014] que exhibe conceitos e introduz, de maneira geral, a análise de desempenho no campo de eficiência energética através de ferramentas e métricas existentes.

Em um trabalho recente [Klôh et al. 2018] há a utilização de aplicações do NPB para avaliação de desempenho e eficiência energética de um ambiente heterogêneo. O trabalho utiliza o EDP (*Energy Delay Product*) para avaliação do consumo energético. O EDP considera o produto do tempo de execução com o gasto em energia ( $s * J$ ).

### 3. Metodologia

Os experimentos foram realizados em um servidor NUMA SGI UV2000 com 48 núcleos distribuídos em 8 processadores Intel® Xeon® CPU E5-4617 (2.90GHz de frequência) de 6 núcleos cada e 512GB de memória RAM.

Os *benchmarks* do conjunto escolhidos para a análise foram CG (*Conjugate Gradient*), FT (*Fourier Transform*) e MG (*Multi-Grid*). Eles se caracterizam pela necessidade de comunicação, que ocorre de maneira diferente considerando as duas implementações. As classes de tamanho B, C e D foram escolhidas para execução. Os parâmetros para cada uma encontram-se disponíveis em um repositório no GitHub<sup>1</sup>.

Para execução das aplicações em MPI utilizou-se a implementação Open MPI, versão 2.1.5. Já para as execuções em OpenMP, a versão 4.5 foi utilizada pelo compilador `gfortran`. As métricas de tempo de execução foram fornecidas pelo próprio *benchmark*. Já para a obtenção das medidas de energia e potência consumidas utilizou-se a ferramenta Likwid, versão 4.3.0, através do comando `likwid-powermeter`.

As aplicações foram executadas com 8, 16 e 32 processos (MPI) ou *threads* (OpenMP). Para cada conjunto específico de variáveis foram realizadas 20 execuções e para cada amostra obtida de tempos de execução foi realizada uma avaliação estatística recomendada em um trabalho anterior [Schmidt et al. 2017].

### 4. Resultados

As execuções forneceram os tempos de execução e medidas de energia e potência consumidas. Com esses dados, houve a comparação das implementações dos *benchmarks* e calculou-se o EDP, através das médias de tempo de execução e de energia consumida.

Em relação aos tempos de execução obtidos, é possível visualizar na comparação realizada na Figura 1(a) que o tempo de execução das aplicações implementadas em OpenMP é superior, na comparação, na maioria dos casos. Nessa figura, ficam visíveis os grupos de *benchmarks* e a diferença existente para cada um deles na comparação. O *benchmark* FT apresenta os tempos de execução mais próximos, onde a diferença de tempo não chega a dobrar, enquanto o *benchmark* MG apresenta as maiores diferenças, onde o tempo de execução para o OpenMP é entre 4 e 8 vezes maior. No caso intermediário, o *benchmark* CG apresenta bons resultados para os casos de 8 *threads* nas classes B e C,

---

<sup>1</sup><https://github.com/lucasroges/erad19>

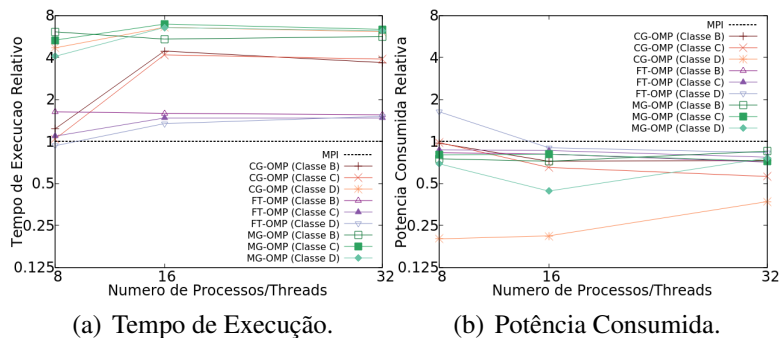


Figura 1. Comparação entre MPI e OpenMP para as métricas identificadas.

enquanto o caso da classe D exibe uma diferença de tempo maior, semelhante ao observado para o *benchmark* MG. O único caso em que uma aplicação em OpenMP apresenta melhor tempo de execução foi para o *benchmark* FT, classe D, executando com 8 *threads*, onde a aplicação executa em 0.96 do tempo da outra implementação.

Na sequência, os dados de potência e de energia consumidas durante a execução das aplicações, obtidos através da ferramenta Likwid, representam a soma do consumo de todos os *sockets* da máquina, para os contadores nos processadores e na memória.

Os dados de potência, na Figura 1(b), mostram um melhor desempenho para os casos do OpenMP. No gráfico observa-se um único caso que consome mais potência, em relação ao caso do MPI, que é o caso onde o OpenMP obteve melhor tempo de execução, executando 1.06 mais rapidamente, enquanto o consumo de potência foi 1.63 maior.

De maneira geral, apesar do menor consumo, as diferenças são pequenas como é possível observar na própria escala do gráfico. Enquanto para a potência, na Figura 1(b), as diferenças variam em um intervalo de aproximadamente 0.5 (2 vezes menor), com um único caso, o do *benchmark* CG (classe D), apresentando o consumo de potência entre 0.20 e 0.37 para as execuções em OpenMP, na comparação com o MPI, na Figura 1(a), as diferenças para o tempo de execução variam em um intervalo maior.

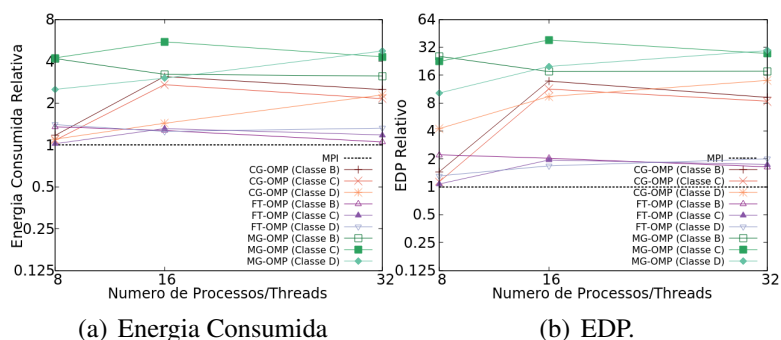


Figura 2. Comparação entre MPI e OpenMP para as métricas identificadas.

Para a obtenção da energia consumida, há a utilização das duas métricas abordadas até o momento: tempo de execução e potência consumida. Como tratado anteriormente, a pequena diferença de potência existente não compensa as diferenças, maiores, encontradas na comparação da métrica de tempo de execução. Dessa forma, o comportamento da

energia consumida na comparação, apresentado na Figura 2(a), segue, majoritariamente, o comportamento visto para a métrica de tempo de execução (Figura 1(a)).

## 5. Considerações Finais

Com relação a comparação de desempenho entre MPI e OpenMP considerando os tempos de execução, o comportamento é semelhante a um já observado anteriormente [Krawezik e Cappello 2003]. Mesmo sendo um trabalho mais antigo, o único caso em que os *benchmarks* em OpenMP obtiveram melhor desempenho geral foi em uma implementação SPMD (*Single Program Multiple Data*) própria dos autores. Além dos resultados semelhantes, foi possível observar o consumo energético para os *benchmarks*.

De maneira geral, observou-se melhor desempenho em tempo de execução para a implementação em MPI e o mesmo para o consumo energético. A potência consumida pelas aplicações implementadas em MPI é maior, porém em pequena proporção, o que, em conjunto com os tempos de execução elevados das aplicações implementadas em OpenMP, gera um maior consumo de energia por parte dessas aplicações na comparação.

Dessa forma, as métricas de energia e de tempo de execução, elevadas nas aplicações em OpenMP, ao serem unidas no cálculo de EDP (citado na Seção 2), favorecem as aplicações em MPI, como é possível observar na Figura 2(b), mostrando que essas apresentam melhor desempenho considerando, ou não, a questão energética.

Para trabalhos futuros, deseja-se investigar os motivos para os bons resultados obtidos com a execução do *benchmark* CG com 8 *threads* em relação a casos com mais *threads*. Os motivos para diferenças maiores ou menores de desempenho entre as implementações, para cada um dos *benchmarks*, também poderão ser investigados.

## 6. Agradecimentos

Este trabalho foi parcialmente financiado pelo projeto “GREEN-CLOUD: Computação em Cloud com Computação Sustentável” (#162551-0000 488-9), no programa FAPERGS-CNPq PRONEX 12/2014.

## Referências

- Klôh, V., Yokoyama, D., Yokoyama, A., Silva, G., Ferro, M., e Schulze, B. (2018). Performance and energy efficiency evaluation for HPC applications in heterogeneous architectures.
- Krawezik, G. e Cappello, F. (2003). Performance comparison of MPI and three OpenMP programming styles on shared memory multiprocessors. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pages 118–127.
- Schmidt, A., Lima Pilla, L., e Zanon Boito, F. (2017). *Fundamentos de Estatística para Análise de Desempenho*, page 22.
- Schöne, R., Treibig, J., Dolz, M. F., Guillen, C., Navarrete, C., Knobloch, M., e Rountree, B. (2014). Tools and methods for measuring and tuning the energy efficiency of HPC systems. *Scientific programming*, 22(4):273–283.
- Utrera, G., Gil, M., e Martorell, X. (2015). In search of the best MPI-OpenMP distribution for optimum Intel-MIC cluster performance. In *2015 International Conference on High Performance Computing Simulation (HPCS)*, pages 429–435.