

Avaliando o Impacto do AMD Turbo Core no Consumo de Energia e Desempenho de Aplicações Paralelas

Sandro Matheus V. N. Marques¹, Marcelo C. Luizelli¹ Fábio Diniz Rossi³,
Antonio Carlos S. Beck², Arthur F. Lorenzon¹

¹ Universidade Federal do Pampa – Campus Alegrete (UNIPAMPA)
Alegrete – RS – Brasil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

³Instituto Federal Farroupilha - Campus Alegrete – Alegrete – RS – Brasil

sandro-matheus@alunos.unipampa.edu.br

Resumo. *O Turbo Core é uma técnica que objetiva melhorar o desempenho das aplicações através do aumento da frequência de operação em processadores AMD. Porém, em aplicações paralelas, a área carece de estudos sobre a sua influência. Desta maneira, este trabalho mostra que o Turbo Core pode acelerar a aplicação paralela em até 13%, porém, consumindo até 37% mais energia.*

1. Introdução

Com a popularização das arquiteturas *multicore*, diversas técnicas têm sido desenvolvidas para controlar individualmente os atributos de cada núcleo. Uma delas, o *dynamic voltage and frequency scaling* (DVFS), tem como objetivo principal possibilitar a redução de potência dissipada e conseqüentemente, o consumo de energia sem impactar de maneira agressiva no desempenho de aplicações através da redução de voltagem e frequência de operação do núcleo [Etinski et al. 2012]. Adicionalmente, DVFS também pode ser utilizado por técnicas de *boosting* para aumentar a frequência de operação em prol de um melhor desempenho quando há espaço de potência remanescente, como é o caso do AMD *Turbo Core* [Branover et al. 2012].

O AMD *Turbo Core* é uma técnica que visa melhorar o desempenho dos núcleos do processador através do aumento da frequência de operação enquanto há espaço para aumentar a potência dentro do *Thermal Design Power* (TDP) de sua arquitetura. Contudo, ao atingir um determinado limite de TDP, o processador dinamicamente diminui a frequência e voltagem de forma a controlar o consumo energético e a temperatura do sistema [Branover et al. 2012].

Diferentes trabalhos têm avaliado o uso de técnicas de *boosting*. J. Charles et al. [Charles et al. 2009] avaliam o desempenho e consumo de energia do Intel *Turbo-Boost*. B. Acun et al. [Acun et al. 2016] analisam o comportamento relacionado a temperatura, potência e frequência dos processadores contidos em supercomputadores durante o uso do Intel *Turbo-Boost*. H. Khdr et al. [Khdr et al. 2018], avalia o processo de degradação do processador durante o uso do Intel *Turbo-Boost* entre curto e longo prazo.

Diferentemente dos trabalhos já realizados, este artigo apresenta uma avaliação preliminar do AMD *Turbo Core* em aplicações sequenciais e paralelas considerando o

desempenho, consumo de energia e do *trade-off* entre estes dois através do *energy-delay product* (EDP). O restante do artigo está organizado como segue: A Seção 2 descreve o funcionamento do AMD *Turbo Core* enquanto que a Seção 3 apresenta a metodologia utilizada neste trabalho. Os resultados são discutidos na Seção 4 enquanto que a Seção 5 apresenta as considerações finais deste trabalho.

2. AMD Turbo Core

O AMD Turbo Core é uma técnica de *boosting* que opera diretamente modificando os *P-States* de um processador, com intuito de alterar o seu desempenho ou consumo de energia. Os *P-States* são pontos de operação de frequência e voltagem, divididos em *P0* até *P(n)-states*. Cada *P-State* representa uma frequência com voltagem específica, sendo *P0* a maior frequência que a CPU consegue suportar. Desta maneira, o processador se encontra em turbo máximo quando está em *P0*, permanecendo assim enquanto a potência estiver dentro da margem de TDP.

O Turbo Core age individualmente em cada núcleo do processador. Entretanto, conforme o número de cores em operação no processador aumenta, também aumenta a temperatura, consumo energético e potência dissipada. Neste caso, quanto maior o número de cores ativos, menor será a frequência máxima suportada por cada núcleo [Branover et al. 2012]. Por exemplo, considerando o processador AMD Ryzen 7 1700¹ (utilizado neste estudo), quando até dois núcleos estão sendo utilizados, a frequência máxima de operação de cada um destes dois núcleos é de 3.7Ghz enquanto que os demais núcleos operam na frequência base (3.0 GHz). Por outro lado, quando até quatro núcleos estão ativos, a frequência máxima de cada um se mantém em 3.4 GHz. Por fim, quando mais de quatro núcleos estão ativos, todos os núcleos operam na frequência base.

3. Metodologia

Os experimentos foram realizados no processador AMD Ryzen 7 1700 de 8 núcleos com o Sistema Operacional Ubuntu, *kernel* v. 4.4.0. Este processador utiliza-se da técnica *Simultaneous Multi-Threading*(SMT), portanto, podendo executar um total de até 16 threads simultaneamente [Oi 2018]. Além disso, a frequência mais alta sem utilizar-se do modo *Turbo Core* é de 3.0Ghz. Por outro lado, quando o modo Turbo Core está ativo, o chip pode operar em frequências de até 3.7Ghz com dois núcleos. Entretanto, como dito anteriormente, as frequências diminuem conforme a utilização de mais núcleos.

Três *kernels* da suíte do *NAS Parallel Benchmarks* foram escolhidos: *lower-upper gauss-seidel solver* (LU), *multi-grid on a sequence of meshes* (MG), e *unstructured adaptive mesh* (UA). Estes *kernels* foram escolhidos pois são amplamente utilizados para avaliação arquitetural de sistemas computacionais. Eles foram executados com o conjunto de entradas da classe C e compilados com o GCC v8.1.0 usando a flag de otimização -O3. Cada aplicação foi executada com diferentes números de *threads*: 1 até 16, que representa o total de *threads* que o processador pode executar. Em todos os testes, o *governor* do DVFS estava configurado para modo *ondemand*, que é o padrão do sistema operacional em uso. A energia consumida foi medida pela suíte de ferramentas AMD uProf. Os resultados apresentados na próxima seção são a média de dez execuções com um desvio padrão inferior que 0.5%.

¹<https://www.amd.com/pt/products/cpu/amd-ryzen-7-1700>

4. Resultados Experimentais

As Figuras 1, 2 e 3 apresentam os resultados de tempo de execução, consumo de energia e EDP de cada aplicação para cada número de *threads*. Considerando inicialmente o desempenho das aplicações paralelas, podemos notar que a execução sequencial (apenas uma *thread*) com o *Turbo Core* ativo possibilitou melhores resultados. No melhor caso, obtido com o *kernel* UA, a aplicação executou 13% mais rápida. No entanto, conforme o número de *threads* aumenta, a melhoria no desempenho causada pela ativação do *Turbo Core* é reduzida até a obtenção de resultados similares à execução com o *Turbo Core* desativado. Este comportamento ocorre porque quando apenas um núcleo está sendo utilizado, a frequência de operação com o *Turbo Core* ativo é de até 3.7GHz enquanto que a frequência é de 3.0GHz quando está inativo. Porém, quando o número de *threads* aumenta, a frequência máxima de operação dos núcleos é reduzida (conforme discutido na Seção 2), diminuindo os ganhos de desempenho.

Embora o uso do *Turbo Core* permita atingir melhor desempenho na execução de até 4 *threads*, o aumento da frequência de operação (e conseqüentemente na voltagem - *Vdd*), tem papel fundamental no consumo de energia. Podemos acompanhar na Figura 2 que para todos os casos, a execução com o *Turbo Core* ativo consumiu mais energia. Este comportamento ocorre porque a frequência e voltagem possuem impacto quadrático na energia do processador, e portanto, qualquer aumento nelas resultará num maior consumo de energia [Lorenzon et al. 2018, Lorenzon et al. 2015, Lorenzon et al. 2016]. No caso mais significativo, a execução com o *Turbo Core* ativo resultou num consumo de energia 37% maior para a aplicação LU (2 *threads*). Por fim, considerando o *trade-off* entre consumo de energia e desempenho através da métrica EDP (Figura 3, para todos os casos, a execução com o *Turbo Core* desativado possibilitou melhores resultados.

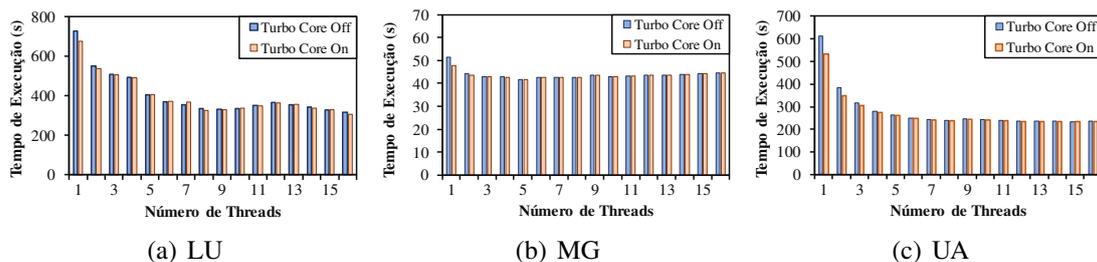


Figura 1. Tempo de Execução

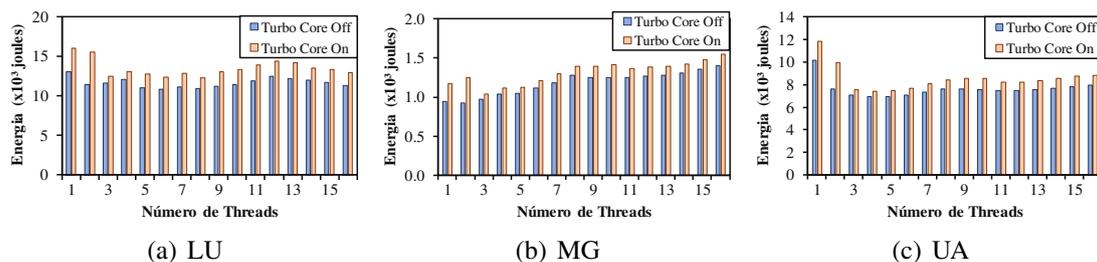


Figura 2. Consumo de Energia

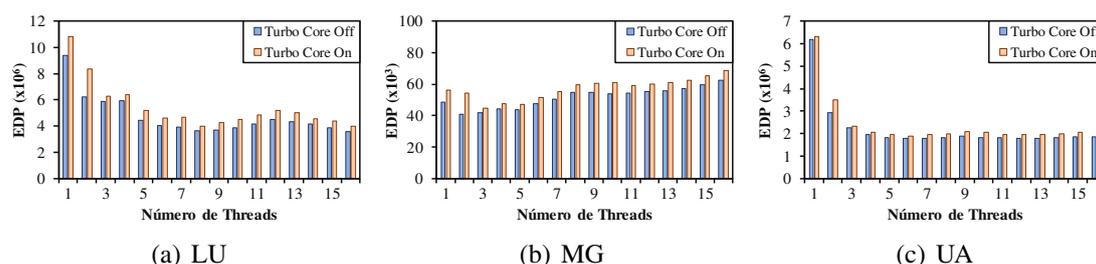


Figura 3. Energy-Delay Product

5. Conclusão

Este trabalho realizou uma análise preliminar do uso do AMD *Turbo Core* em aplicações paralelas. A partir da execução de um conjunto de *benchmarks* paralelos, mostrou-se que a ativação do AMD *Turbo Core* apresenta melhor desempenho na execução de até quatro *threads*, e que a partir deste momento, os resultados são similares à execução com o *Turbo Core* desativado. Por outro lado, quando as aplicações paralelas são executadas com o *Turbo Core* ativo, existe um elevado consumo de energia devido ao aumento da voltagem e frequência de operação do processador. Como trabalhos futuros, pretende-se expandir o ambiente de execução para compreender outras aplicações, *DVFS governors* e métricas.

Referências

- Acun, B., Miller, P., and Kale, L. V. (2016). Variation among processors under turbo boost in hpc systems. In *International Conference on Supercomputing*, page 6. ACM.
- Branover, A., Foley, D., and Steinman, M. (2012). Amd fusion apu: Llano. *Ieee Micro*, 32(2):28–37.
- Charles, J., Jassi, P., Ananth, N. S., Sadat, A., and Fedorova, A. (2009). Evaluation of the intel® core™ i7 turbo boost feature. In *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pages 188–197. IEEE.
- Etinski, M., Corbalán, J., Labarta, J., and Valero, M. (2012). Understanding the future of energy-performance trade-off via dvfs in hpc environments. *Journal of Parallel and Distributed Computing*, 72(4):579–590.
- Khdr, H., Amrouch, H., and Henkel, J. (2018). Aging-aware boosting. *IEEE Transactions on Computers*.
- Lorenzon, A. F., Cera, M. C., and Beck, A. C. S. (2015). On the influence of static power consumption in multicore embedded systems. In *ISCAS*, pages 1374–1377. IEEE.
- Lorenzon, A. F., Cera, M. C., and Beck, A. C. S. (2016). Investigating different general-purpose and embedded multicores to achieve optimal trade-offs between performance and energy. *Journal of Parallel and Distributed Computing*, 95:107–123.
- Lorenzon, A. F., Oliveira, C. C. D., Souza, J. D., and Filho, A. C. S. B. (2018). Aurora: Seamless optimization of openmp applications. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–15.
- Oi, H. (2018). Energy efficiency study of ryzen microprocessor. In *SoutheastCon 2018*, pages 1–5. IEEE.