

# Análise comparativa do desempenho de uma arquitetura de CPU Híbrida

Yuri Theodoro da Silva, Vinícius Garcia Pinto

Centro de Ciências Computacionais

Universidade Federal do Rio Grande (FURG), Rio Grande – RS – Brasil

yuritheodorods@gmail.com, vinicius.pinto@furg.br

**Resumo.** *Este artigo tem como objetivo analisar o desempenho de uma arquitetura de CPU híbrida, analisando o impacto da tecnologia SMT e do aumento dinâmico da frequência do processador. Para isso, avaliamos métricas como tempo médio de execução e aceleração, comparando uma arquitetura homogênea convencional com uma arquitetura híbrida. Na arquitetura homogênea o tempo de execução é relativamente estável, enquanto na híbrida há variações mais significativas, assim como observamos que o SMT teve um impacto maior na híbrida.*

## 1. Introdução

O OpenMP é uma API que oferece suporte à programação paralela em memória compartilhada para as linguagens C, C++ e Fortran. Ele permite o desenvolvimento de aplicações paralelas em diversas plataformas como desktops e supercomputadores. A especificação do OpenMP é mantida pela OMP Architecture Review Board (ARB) [OpenMP ARB 2025].

O NAS Parallel Benchmarks (NPB) é um grupo de programas desenvolvido pela Divisão de Supercomputação Avançada da NASA, com o intuito de avaliar o desempenho de supercomputadores paralelos. Esses benchmarks são amplamente reconhecidos como indicadores padrão de desempenho em computação e estão disponíveis em três versões, sendo elas NPB-MPI (versão paralelizada com MPI, bastante usada para execução em clusters e sistemas distribuídos.), NPB-OMP (versão paralelizada com OpenMP, mais recomendada para execução em sistemas de memória compartilhada (SMP) e NPB-Serial (Implementação sequencial usada como referência para comparação de desempenho) [NAS 2025].

No contexto do hardware, diversas tecnologias foram desenvolvidas a fim de aprimorar o desempenho de processadores modernos. O *Simultaneous Multithreading* (SMT) torna possível a execução de múltiplos threads em cada núcleo físico, permitindo que ele processe duas tarefas ao mesmo tempo, o que pode melhorar o desempenho ao distribuir melhor a carga de trabalho, tornando o processamento mais eficiente. O Hyper-Threading é uma implementação de SMT disponível em processadores Intel. O TurboBoost, por sua vez, é outra tecnologia da Intel, que faz o ajuste dinamicamente na velocidade dos núcleos do processador além da frequência base quando há necessidade de mais desempenho. O que faz com que o processador possa trabalhar mais rápido em momentos de alta demanda para concluir tarefas intensivas de forma mais eficiente. Entretanto, o Sistema Operacional não controla em quais momentos tal ajuste ocorrerá, o único controle possível é desativar o mesmo por completo.

Mais recentemente, a Intel introduziu uma arquitetura híbrida com dois tipos diferentes de núcleos, os P-cores (*Performance-cores*) e E-cores (*Efficient-cores*). Os P-cores foram projetados para auxiliar em tarefas que exijam alto desempenho, como por exemplo jogos e edição de vídeo, enquanto os E-cores tem o foco em eficiência energética, o que o torna ideal para tarefas em segundo plano. Essa combinação permite um gerenciamento melhor e mais equilibrado da carga de trabalho, combinando desempenho e eficiência energética de forma inteligente [Intel 2025].

Neste trabalho, analisamos o impacto do Hyper-Threading (SMT) no desempenho dos clusters Tuco-Tuco e Vagoneta, tanto com 1 thread quanto multithread, avaliando o tempo médio de execução e a aceleração. A Seção 2 apresenta a metodologia usada para a realização dos testes. Na Seção 3 apresentamos e discutimos os resultados dos testes e seu impacto no desempenho dos clusters. Por fim, na Seção 4, concluímos o estudo e apontamos ideias para trabalhos futuros.

## **2. Trabalhos Relacionados**

Diversos estudos exploram a otimização de desempenho em arquiteturas multicore assimétricas, como o Intel Alder Lake. O trabalho [Moori et al. 2024] mostra como a distribuição eficiente de threads entre P-cores e E-cores pode melhorar o desempenho de programas paralelos. O trabalho [Saez and Prieto-Matias 2022] investiga como a tecnologia Intel Thread Director otimiza a alocação de tarefas entre núcleos. Esses trabalhos são fundamentais para aprimorar o uso do OpenMP e a execução em sistemas modernos com múltiplos núcleos. Além disso, o trabalho [Gonçalves et al. 2024] demonstra como a variabilidade de processos na arquitetura Alder Lake pode afetar o desempenho e a eficiência energética, com a diferença de até 42% no energy-delay product (EDP) entre núcleos com as mesmas características. Isso destaca a importância de levar em conta variações de desempenho entre núcleos ao otimizar o equilíbrio entre desempenho e consumo energético, especialmente em arquiteturas heterogêneas.

## **3. Metodologia**

A fim de avaliar o impacto das tecnologias de paralelismo no desempenho dos processadores, foram realizados testes utilizando a aplicação Embarrassingly Parallel (EP) do NPB. Essa aplicação foi escolhida pois realiza apenas cálculos paralelos sem dependência entre tarefas, o que permitiu uma análise mais direta do efeito do número de threads e das características dos núcleos do processador. Os testes foram divididos em dois principais experimentos, onde no primeiro variamos o número de threads para medir o speedup, ou seja, como o desempenho melhora à medida que mais threads são utilizadas. No segundo experimento, rodamos a aplicação de forma sequencial, alterando o code ID, que define em quais núcleos do processador a execução ocorre. Isso nos permitiu comparar o desempenho entre tipos diferentes de núcleos, como os P-cores e E-cores.

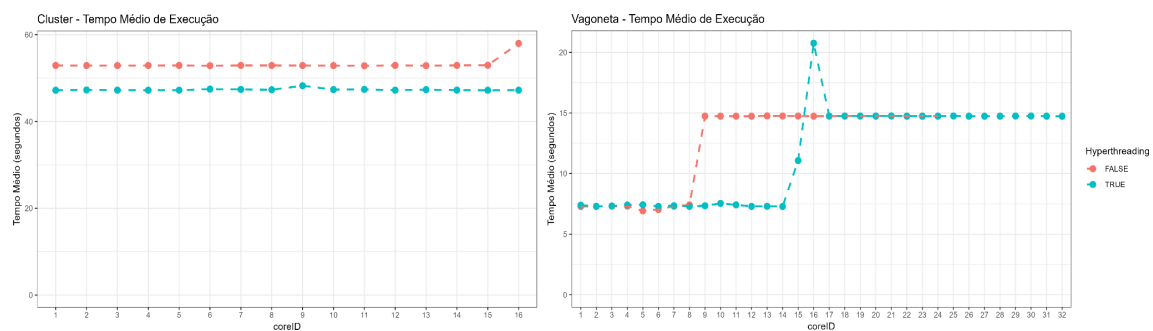
Os testes foram executados em duas máquinas distintas para avaliar o impacto do hardware. A primeira máquina possui um processador Intel Core i9 de 14 geração,

com núcleos híbridos (8 P-cores e 16 E-cores) e suporte a Hyper-Threading nos P-Cores executando Ubuntu Server 24.04 LTS. A segunda máquina conta com dois processadores Intel Xeon E5-2640 v3, com 8 núcleos homogêneos e SMT e roda o sistema operacional Linux CentOS 7.3. As execuções foram feitas utilizando o compilador GCC com suporte ao OpenMP. Para facilitar a identificação nos resultados, a máquina híbrida será referida como *Vagoneta*, enquanto a máquina com processadores homogêneos será chamada de *Tuco-Tuco*.

Para garantir resultados mais consistentes, foram realizadas cinco repetições de cada teste, registrando os tempos médios para análise. Esse número de repetições ajuda a diminuir variações ocasionais e permite obter uma avaliação mais precisa do comportamento do hardware sob diferentes tipos de execuções.

#### 4. Resultados

Na Figura 1 podemos observar o tempo de execução das máquinas Tuco-Tuco e Vagoneta com 1 thread variando o núcleo CPU onde tal thread é executada, com e sem a presença de Hyper-Threading e Turbo Boost.



**Figura 1. Tempo Médio 1 thread comparação**

No Tuco-Tuco, o tempo de execução permanece estável e tem um leve aumento sem Hyper-Threading. Já na Vagoneta, há uma variação maior, com aumento significativo sem Hyper-Threading a partir do coreID 8 e um pico com Hyper-Threading no coreID 16. Isto pode ser explicado pois o Hyper-Threading está disponível apenas nos P-Cores (núcleos 0 a 7). Quando ele é desativado, os IDs 8 a 15 deixam de corresponder a núcleos lógicos dos P-Cores e passam a corresponder a E-Cores físicos. Nota-se que o impacto do Hyper-Threading foi muito maior na Vagoneta.

Já nas Figuras 2 e 3, temos os tempos médios e aceleração das duas máquinas, onde ambos se mantêm estáveis, com o tempo médio reduzindo conforme o número de threads aumenta e com o Tuco-Tuco tendo um desempenho um pouco melhor com o Hyper-Threading, porém com a Vagoneta sendo muito mais rápida, com um tempo de execução extremamente baixo comparado ao Tuco-Tuco.

## 5. Conclusão<sup>1</sup>

O trabalho analisou o impacto do Hyper-Threading (SMT) no desempenho dos cluster Tuco-Tuco e Vagoneta, observando o tempo médio e aceleração tanto com 1 thread como multithread e comparou os resultados entre eles. Como resultado, podemos observar que na vagoneta ocorreram instabilidades com uma frequência maior, enquanto o Tuco-Tuco se mostrou mais estável.

Como trabalho futuro, propõe-se realizar outros testes com diferentes aplicações, avaliando o impacto do Hyper-Threading em situações diferentes, além de incluir métricas de consumo de energia.

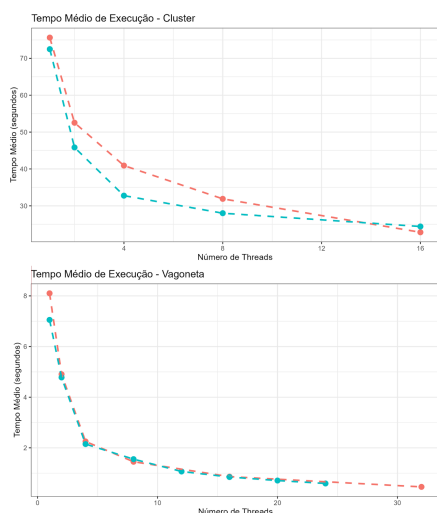


Figura 2. Tempo Médio Multithread

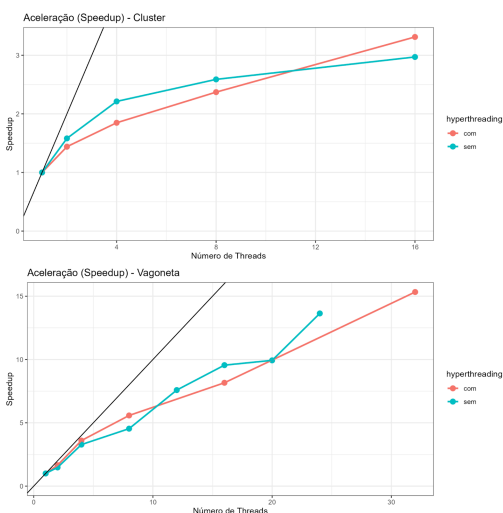


Figura 3. Aceleração Multithread

## Referências

- OpenMP Architecture Review Board. (2025) “The OpenMP API specification for parallel programming”, <https://www.openmp.org/>, Acesso em Janeiro de 2025.
- NASA Advanced Supercomputing (NAS) Division. (2025) “NAS Parallel Benchmarks (NPB)” <https://www.nas.nasa.gov/software/npb.htm>, Acesso em Janeiro de 2025.
- Intel Corporation. (2025) “How Hybrid Design Works”. Whitepaper, 2025. <https://www.intel.com.br/content/www/br/pt/gaming/resources/how-hybrid-design-works.html>, Acesso em Janeiro de 2025.
- Moori, M.K., Rocha, H.M.G.D.A., Lorenzon, A. F., & Beck, A. C. S (2024). Efficient Thread Tuning for Asymmetric Multicores. *Proceedings of the 37th SBC/SBMicro/IEEE Symposium on Integrated Circuits and Systems Design (SBCCI)*, 1-5.
- Saez, J. C., & Prieto-Matias, M.(2022). Evaluation of the Intel Thread Director Technology on an Alder Lake Processor. *Proceedings of the 13th ACM SIGOPS Asia-Pacific Workshop on Systems (APSys '22)*, 61-67.
- Gonçalves, T. dos S., Beck, A. C. S., & Lorenzon, A. F. (2024). Análise da Variabilidade de Processos na Execução de Tarefas em Arquiteturas Heterogêneas. *Anais da Escola Regional de Alto Desempenho da Região Sul (ERAD-RS)*, 24, 41-44.

<sup>1</sup> Trabalho parcialmente financiado pela FAPERGS (PROBIC e ARD/ARC 23/2551-0000861-0).