UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ
UNIOESTE – CAMPUS DE FOZ DO IGUAÇU
CENTRO DE ENGENHARIAS E CIÊNCIAS EXATAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

ANÁLISE COMPARATIVA DE TEMPO DE CPU

Impacto do Perfil de Uso de Recursos (CPU vs. Memória)

Autor: José Lucas Hoppe Macedo

Professor(a): Fabiana Frata Furlan Peres

Foz do Iguaçu – PR
18 de junho de 2025

Sumário

Lista de Tabelas

Lista de Figuras

1 Introdução

Este trabalho tem como objetivo analisar e comparar o tempo de CPU gasto na execução de dois programas com perfis computacionais distintos: um intensivo em processamento (CPU-bound) e outro intensivo em acesso à memória (Memory-bound). Ao medir e confrontar o tempo de CPU em ambos os cenários, busca-se entender como a natureza da carga de trabalho e a arquitetura de hardware subjacente influenciam o desempenho. A análise, realizada em duas configurações de hardware diferentes, visa demonstrar como o tempo do processador é consumido de forma distinta quando a tarefa é limitada pelo poder de cômputo puro ou pela velocidade do subsistema de memória.

2 Descrição dos Programas

2.1 Programa A – Contador de Primos (CPU-bound)

O primeiro programa, 'CPUBound.java', foi projetado para maximizar o uso das unidades de processamento da CPU. Sua função é calcular a quantidade de números primos em um vasto intervalo (de 2 a 50.000.000), uma tarefa que envolve um volume massivo de operações aritméticas. O desempenho deste programa é, portanto, um reflexo direto da capacidade de processamento bruto do processador.

2.2 Programa B – Manipulação de Árvore Binária (Memory-bound)

O segundo programa, 'MemoryBound.java', foi criado para gerar uma carga de trabalho extremamente intensiva em acessos à memória RAM. Ele aloca e manipula uma árvore de busca binária massiva, executando as seguintes operações: inserção de 10 milhões de nós, realização de 20 milhões de buscas, 1000 percursos completos pela estrutura e a clonagem da árvore inteira por 40 vezes. O objetivo é forçar o processador a realizar um grande número de acessos a endereços de memória dispersos, resultando em frequentes *cache misses*. Consequentemente, o tempo de CPU medido para este programa não reflete apenas o processamento, mas também inclui o tempo em que a CPU permanece ociosa (*stalled*), aguardando os dados serem trazidos da memória principal para os caches.

3 Descrição dos Hardwares

Tabela 1: Especificações resumidas dos sistemas de teste

	Hardware 1 (PC)	**Hardware 2 (Notebook)**
CPU	Intel Core i5-9400F 6 c/6 t, 2.9–4.1 GHz, 9 MB L3	Intel Core i7-1165G7 4 c/8 t, 2.8–4.7 GHz, 12 MB L3
RAM	16 GB DDR4-2400 dual-channel	8 GB DDR4-2667 single-channel
Storage	SSD NVMe Kingston A2000 500GB	SSD SATA Kingston SA400S37480G 480GB

4 Sistema Operacional e Ferramentas de Medição

- **SO**: Windows 11 Pro (utilizado em ambas as máquinas).
- **Linguagem**: Java, utilizando o ambiente de execução da OpenJDK versão 23.
- **Ferramenta de Medição**: Para medir o tempo de execução em ambos os programas, foi utilizada a biblioteca nativa do Java 'java.lang.management'. Especificamente, o método 'ManagementFactory.getThreadMXBean().getCurrentThreadCpuTime()' foi invocado no início e no fim do bloco de código principal. Este método retorna o tempo total que a CPU dedicou à thread de execução.

5 Metodologia

Cada um dos dois programas foi executado 10 vezes consecutivas em cada um dos dois hardwares. O tempo de CPU, em segundos, foi registrado para cada execução. Os resultados consolidados, apresentados sob a forma de tabelas e gráficos de média aritmética, são utilizados para a análise comparativa de desempenho.

6 Resultados

As tabelas e gráficos a seguir apresentam o tempo de CPU medido para cada programa.

Tabela 2: Tempo de CPU em segundos para o programa *CPU-Bound*

Teste N.º	**H1 (PC)**	**H2 (Notebook)**
1	154	39
2	155	38
3	155	39
4	155	39
5	155	38
6	155	38
7	154	42
8	155	45
9	155	44
10	155	42
Média	**154.8**	**40.4**

Tabela 3: Tempo de CPU em segundos para o programa *Memory-Bound*

Teste N.º	**H1 (PC)**	**H2 (Notebook)**
1	62	67
2	60	65
3	63	61
4	62	61
5	61	62
6	61	64
7	60	61
8	60	63
9	66	63
10	60	64
Média	**61.5**	**63.1**

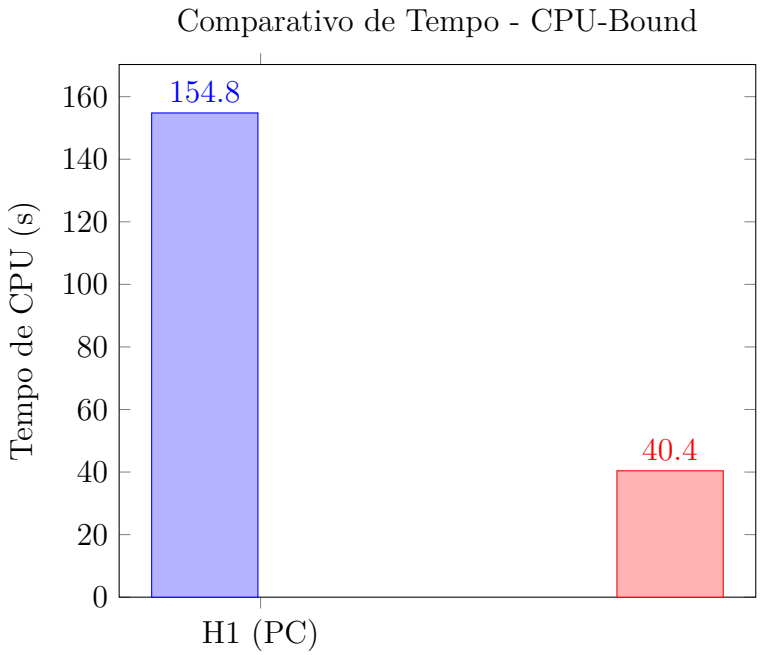

Figura 1: Comparação do tempo médio de CPU para o programa CPU-Bound (Menor é melhor)

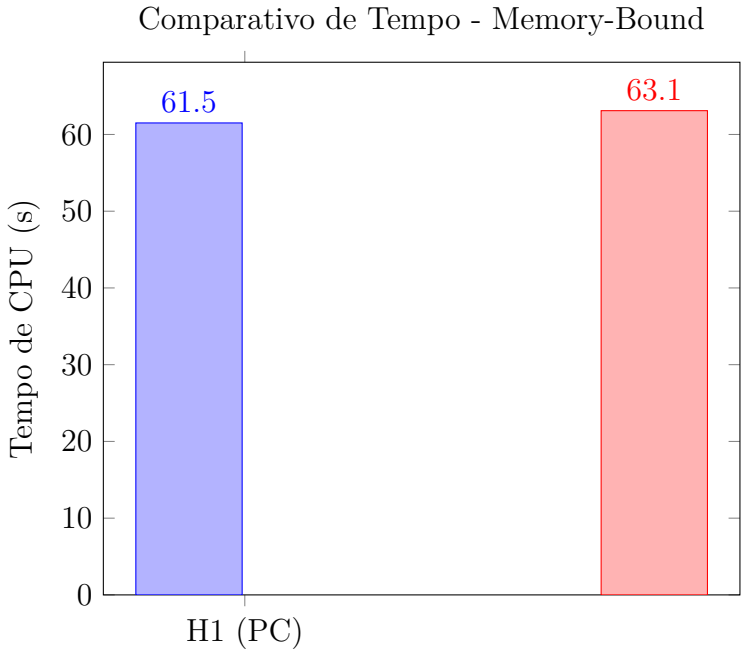

Figura 2: Comparação do tempo médio de CPU para o programa Memory-Bound (Menor é melhor)

7 Análise Comparativa

A análise dos tempos de CPU revela uma dinâmica de desempenho complexa e dependente da carga de trabalho.

- **Qual programa consumiu mais tempo de CPU?** O resultado variou significamente entre os dois.

  - No **H1 (PC)**, o programa **CPU-Bound foi o mais lento** (154,8s vs. 61,5s).
  - No **H2 (Notebook)**, o programa **Memory-Bound foi o mais lento** (63,1s vs. 61,5s).

- **A diferença está relacionada ao tipo de recurso utilizado?** Sim. Essa inversão demonstra como a arquitetura do hardware muda a natureza do gargalo. O H1, com sua CPU mais antiga, sofreu mais com a computação intensiva do que com os acessos à memória. Já o H2, com sua CPU moderna e eficiente, executou a tarefa computacional rapidamente, fazendo com que o tempo de espera por dados da memória se tornasse seu principal gargalo.

- **Análise de Desempenho entre Hardwares:**

  - No teste **CPU-Bound**, o H2 (Notebook) foi quase 4 vezes mais rápido que o H1 (PC). Este resultado, apesar de o H2 ter menos núcleos, evidencia superioridade arquitetura de CPU (Tiger Lake vs. Coffee Lake). O maior IPC (instruções por ciclo), a maior contagem de threads (8 vs. 6) e o cache L3 maior permitiram ao H2 uma performance computacional muito superior.
  - No teste **Memory-Bound**, o desempenho foi praticamente idêntico, com o H1 (PC) sendo marginalmente mais rápido (61,5s vs 63,1s). Este "empate técnico"é o resultado de um equilíbrio de forças: o H1 se beneficia da maior largura de banda de sua RAM em *dual-channel*, enquanto o H2 se beneficia da eficiência de seu controlador de memória mais moderno e de seu cache L3 maior, que reduz a latência. Para esta carga específica, as duas abordagens de arquitetura de memória resultaram em performance muito similar.

Estratégias de Otimização:

- Para o programa **CPU-bound**, a otimização mais eficaz seria a paralelização explícita (dividir o trabalho entre múltiplas threads) para utilizar todos os núcleos/threads disponíveis, além do uso de algoritmos matematicamente mais eficientes (ex: Crivo de Eratóstenes).

- Para o programa **Memory-bound**, o objetivo seria melhorar a localidade dos dados para reduzir as falhas de cache. Estratégias como o uso de estruturas de dados mais compactas (que caibam melhor no cache) ou algoritmos que acessem a memória de forma mais sequencial poderiam diminuir o tempo de espera da CPU e, por consequência, o tempo total de execução.

8 Conclusão

A avaliação do tempo de CPU em duas cargas de trabalho distintas demonstrou que a performance de um sistema é uma relação complexa entre a arquitetura do hardware e a natureza do software. Os resultados desmistificam a ideia de que um único componente define o desempenho.

Para tarefas **CPU-bound**, ficou evidente que os avanços na microarquitetura (maior IPC) são um fator dominante. A CPU moderna do Notebook (H2) superou a CPU de Desktop mais antiga (H1), mesmo com menos núcleos, provando eficiência computacional.

Em contrapartida, em tarefas **memory-bound**, a disputa se torna mais acirrada. O resultado quase idêntico entre os dois sistemas expôs um equilíbrio entre a largura de banda da memória (vantagem do H1) e a eficiência do cache e do controlador de memória (vantagem do H2).

A descoberta mais significativa foi a inversão do gargalo: a CPU antiga do H1 sofreu mais com a tarefa computacional, enquanto a CPU moderna do H2 foi tão eficiente que tornou a espera pela memória seu maior obstáculo.

Referências

HENNESSY, J.; PATTERSON, D. *Computer Architecture: A Quantitative Approach*. 6. ed. Morgan Kaufmann, 2019. Nenhuma citação.

TANENBAUM, A. S.; AUSTIN, T. *Structured Computer Organization*. 6. ed. Pearson, 2013. Nenhuma citação.

INTEL CORPORATION. **Processador Intel® Core™ i5-9400F (cache de 9M, até 4,10 GHz): Especificações**. 2024. Disponível em: <https://www.intel.com.br/content/www/br/pt/products/sku/190883/intel-core-i59400f-processor-9m-cache-up-to-4-10-ghz/specifications.html>. Acesso em: 17 jun. 2025. Nenhuma citação.

INTEL CORPORATION. **Processador Intel® Core™ i7-1165G7 (cache de 12M, até 4,70 GHz): Especificações**. 2024. Disponível em: <https://www.intel.com.br/content/www/br/pt/products/sku/208658/intel-core-i71165g7-processor-12m-cache-up-to-4-70-ghz/specifications.html>. Acesso em: 17 jun. 2025. Nenhuma citação.