

**UNIVERSIDADE FEDERAL DO PAMPA– UNIPAMPA**  
**BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

HEITOR MAURO CHAVEZ HUARACHI

NICOLAS URIEL RAMOS

**RELATÓRIO DE SISTEMAS OPERACIONAIS**

BAGÉ – RIO GRANDE DO SUL

2022

## 1. INTRODUÇÃO

O trabalho tem como objetivo apresentar alguns conceitos de threads e algumas informações da máquina operada pelo software.

Todo o código desenvolvido ao decorrer deste trabalho pode ser conferido no github<sup>1</sup>.

## 2. HARDWARE EXECUTADO

O trabalho foi executado em um desktop de mesa, onde suas especificações estarão listadas abaixo:

- Processador ryzen 5 3400g(4 cores e 8 threads)
- Placa Mãe b450
- Memória RAM 16gb 3000mhz ddr4
- SSD 500GB

Com isso demos progresso ao código e testando a capacidade e tempo em que executa o programa criado com os desejos do professor.

## 3. GRÁFICO DE DESEMPENHO

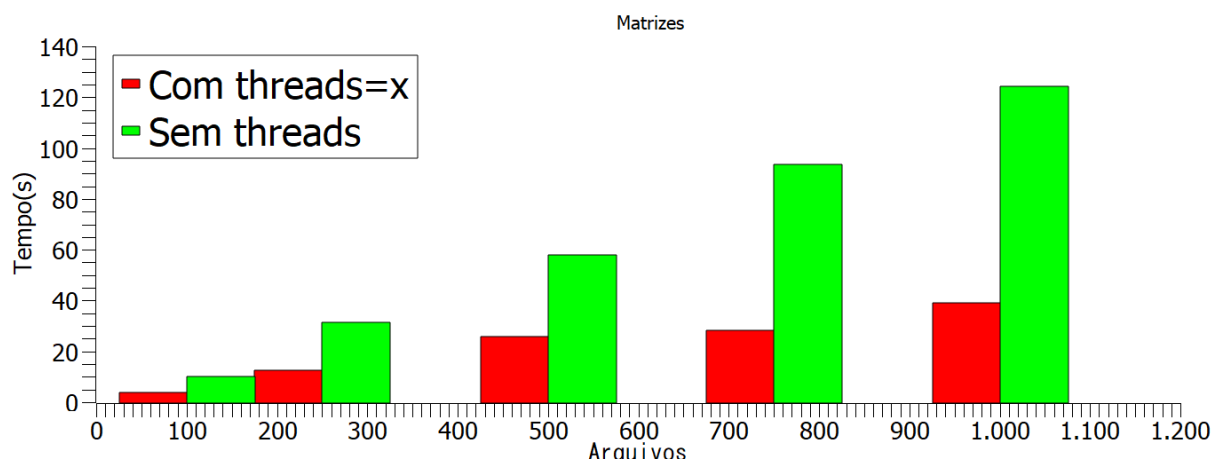
Para fazer o desempenho do código foi utilizado duas ferramentas para fazer esse processo com conhecimentos de laboratório de física 1, utilizamos um software chamado “Scidavis” para criação do gráfico, junto com o comando “htop” do linux, onde é mostrado todas as informações necessárias para definir o trabalho como as threads criadas, a cache e a quantidade de memória RAM utilizada. Foi desenvolvido um trabalho chamado “main.py”, onde dentro dele está contido todo o software, e o “sem\_threads” onde dentro desse software esta o mesmo trabalho, mas sem uso de nenhuma thread para ter um guia de desempenho..

---

<sup>1</sup> <https://github.com/elcabriton/Matrizes>

#### 4. SOFTWARE

O gráfico que será apresentado abaixo mostra o tempo com threads e sem threads, lembrando que o número de threads e o mesmo número de arquivos, e o sem threads esta explicito na legenda do software. A quantidade de arquivos para se fazer o grafico foram, 100,250,500,750,1000 arquivos para os dois programas.



Fonte: Autoria Própria.

De acordo com a figura acima se pode notar que o uso de threads realmente deu desempenho ao programa, caindo pela metade o tempo de execução do programa. De 750 arquivos, e 1000 arquivos, pela quantidade alta de threads acabou aumentando ainda mais o desempenho, tendo um terço do valor do desempenho do “sem\_threads”. Com o uso de threads e arquivos acima de 500 o Sistema Operacional acaba dando um erro “Too many open files” como forma de proteção, por conta desse problema se obteve a solução em linux(Não achamos como arrumar isso no windows), onde no console digitamos “ulimit -a”, e logo após isso o comando “ulimit -Sn 3000” onde colocamos o valor de arquivos abertos em 3000 dando mais espaço para o software trabalhar.

#### 5. Conclusão

Podemos concluir desse trabalho que com o uso de threads realmente aumenta o desempenho do programa, onde houve diminuição do tempo de execução de cada ciclo de arquivos pela metade ou mais, dando mais desempenho ao software.

Uma dificuldade de construir o software foi o uso de número em notação científica, porque ao passar o código em pandas ele convertia automaticamente, tendo que encontrar algum modo para se alocar o número em notação científica(leitura da documentação do

pandas). Outro problema foi o uso de CSV, onde acabou dificultando passar um arquivo “.CSV” para “.txt”.

Esse tipo de técnica com threads pode ser utilizada em outros trabalhos para aumentar o desempenho e melhorar cada vez mais os softwares utilizados dentro da parte acadêmico ou até mesmo para trabalhos em empresas.