centro universitário FEI

CENTRO UNIVERSITÁRIO FEI

Avenida Humberto de Alencar Castelo Branco, 3972, CEP: 09850-901 São Bernardo do Campo
Telefone: (011) 4353-2900 Fax (011) 4109-5994

Curso de Ciências da Computação

Atividade:
SpeedUp

Autores:	Data de emissão:
Matheus Elias Cruz - R.A.: 22.118.167-0	30/08/2021
Revisor:	
Profº Ricardo de Carvalho Destro	

Índice

Código **2**

Questões **3**

Pergunta 1 - Quanto mais threads, o SpeedUp melhora? 3

Pergunta 2 - Por que se aumentamos MUITO a quantidade de threads,perdemos o SpeedUp? 3

Melhorias **4**

Loop de código 4

Gráfico de tempos 4

Gráfico de Speed Ups 5

Algoritmo Multiprocessamento 5

Para pensar **5**

1 Código

Para construção do algoritmo, utilizei como base o modelo disponibilizado pelo professor, utilizando o website Replit para implementar as melhorias e o aplicativo Visual Studio Code para teste do mesmo.

O código fonte da atividade Speed Up se encontra na linguagem Python e está disponível no Github e no Replit, ambos links disponíveis abaixo:

https://github.com/MatheusEliasC/SpeedUp

https://replit.com/@MatheusElias/SpeedUp?v=1

2 Questões

Pergunta 1 - Quanto mais threads, o SpeedUp melhora?

No caso do algoritmo que calcula se os números são primos (cpu intense), quanto mais threads pior, pois o overhead causado pelas thread consome mais tempo de processamento do que o algoritmo simples sem threads. Um multiprocessamento seria mais eficiente pois utilizaria outros cores do processador para agilizar o processo.

Pergunta 2 - Por que se aumentamos MUITO a quantidade de threads,perdemos o SpeedUp?

Porque gastamos mais tempo com o overhead das threads do que com o processamento.

3 Melhorias

Loop de código

Como melhorias do código, incluímos um loop para que possamos tirar uma média de speedups antes de calcular o resultado:

```
for i in range(ciclos):
```

Com a aplicação num ciclo, podemos gerar alguns gráficos que mostram o andamento da execução:

Gráfico de tempos

O primeiro gráfico, mostra a relação entre o tempo de execução do Método Simples com o Método Thread

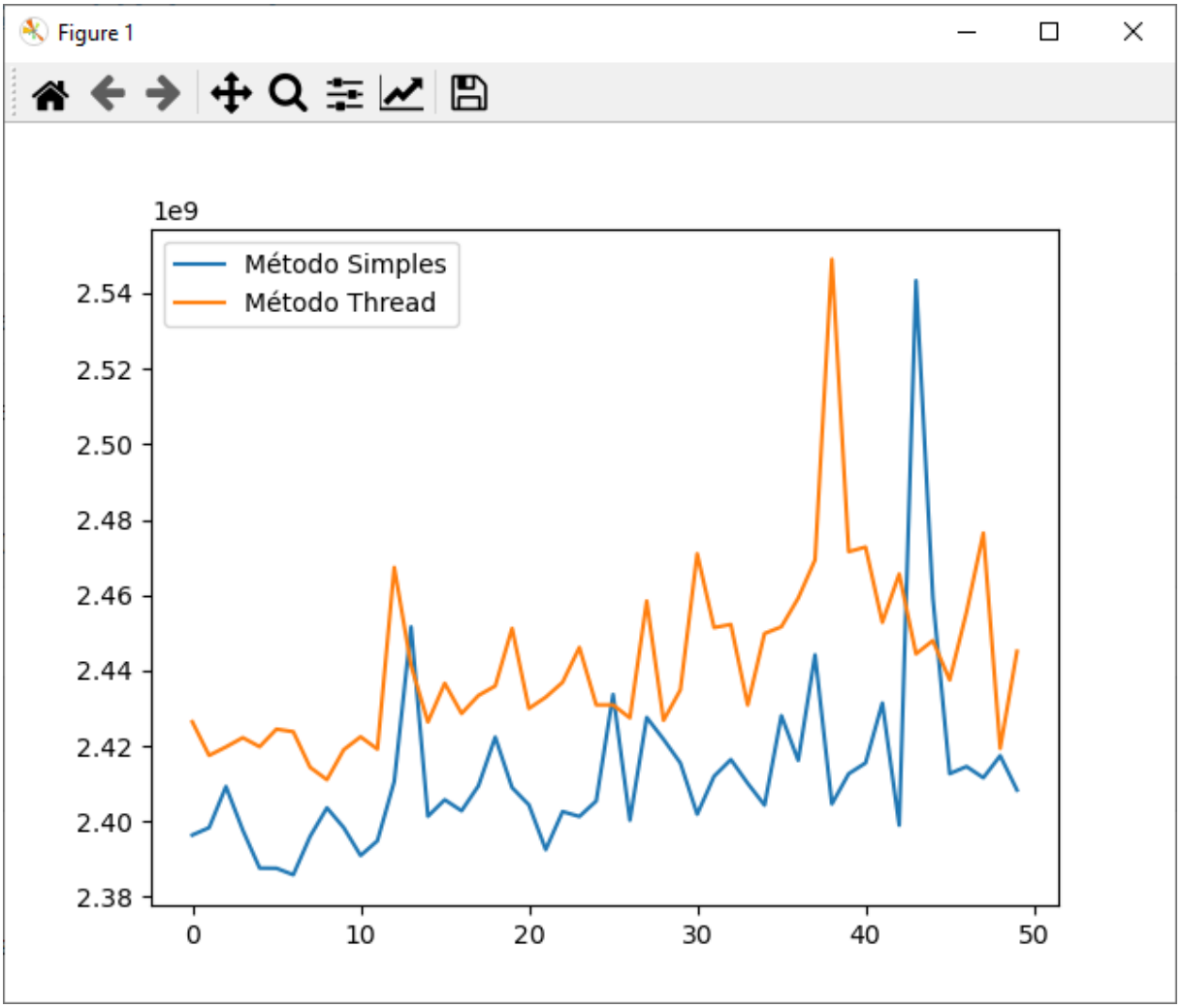
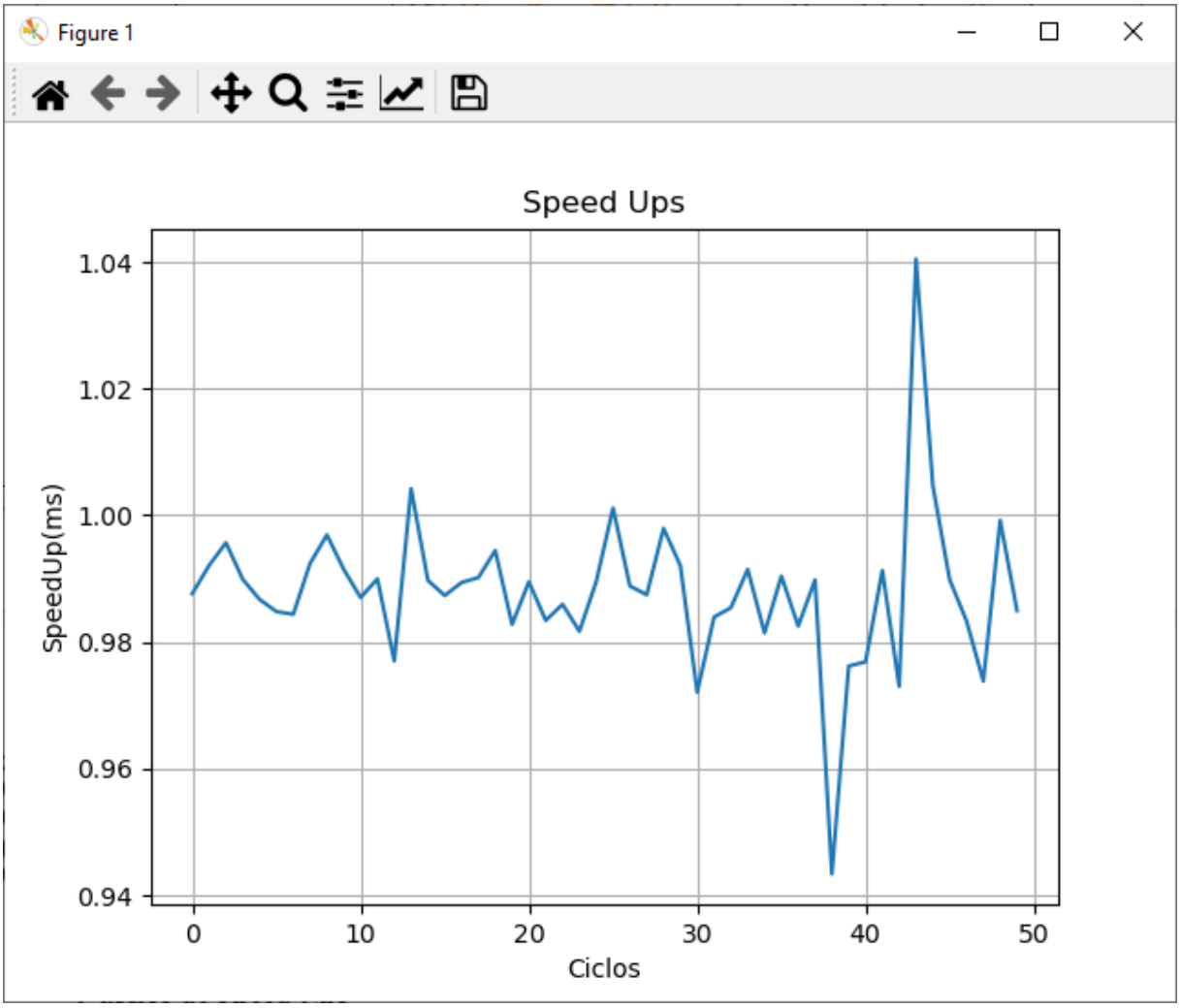


Gráfico de Speed Ups

O segundo gráfico mostra a relação dos speedups calculados com o número de ciclos



Algoritmo Multiprocessamento

Além dos gráficos, o main2.py atua com o algoritmo em multiprocessamento, utilizando de forma correta o código e o máximo da cpu.

Esta função só está disponível em Linux, para testar, utilize o Repl.it, substituindo o main.py pelo main2.py

4 Para pensar

Pergunta: É possível melhorar a "solução" simples apresentada?

Se os números forem armazenados em um banco de dados e o código tenha que esperar a consulta desses números, passamos de "cpu intense" para "io intense" onde o cpu pode trabalhar durante essa espera.