



Universidade Federal do Maranhão - UFMA
Centro de Ciências Exatas e Tecnologia
Bacharelado Interdisciplinar em Ciência e Tecnologia
Disciplina: Sistemas Operacionais
Profa. Me. Alana Oliveira Meireles Teixeira
Alunos: Samuel Silva e Silva, Guilherme Roberto

Laboratório 01 - Multiprocessamento vs Threading in Python

Para que possamos observar de forma clara as diferenças entre a utilização de multiprocessos e *threads* no *python 3*, criamos um programa simples utilizando a função *math.sqrt(x)*, função essa pertencente ao pacote *math*, que retorna a raiz quadrada do número *x*. Para gerar uma alta carga de processamento, utilizamos a estrutura de repetição *for in* para rodar *math.sqrt(x)* com números de 0 a 4000000. Para aferição do tempo entre o início e fim do programa, utilizamos a biblioteca *time*.

```
#multiprocessing
from multiprocessing import Process
import os
import math
import time

start = time.time()

def calc():
    for i in range(0, 4000000):
        math.sqrt(i)

processes = []

for i in range(os.cpu_count()):
    print("registering process %d" %i)
    processes.append(Process(target=calc))

for process in processes:
    process.start()

for process in processes:
    process.join()

end = time.time()
print("response time: ", end - start)

#threading
from threading import Thread
import os
import math
import time

start = time.time()

def calc():
    for i in range(0, 4000000):
        math.sqrt(i)

threads = []

for i in range(os.cpu_count()):
    print("registering thread %d" %i)
    threads.append(Thread(target=calc))

for thread in threads:
    thread.start()

for thread in threads:
    thread.join()

end = time.time()
print("response time: ", end - start)
```

Figura 1: Multiprocessing e Threading

Antes de executarmos o programa, o *IDLE* permaneceu utilizando 0% da *CPU*, com dois processos registrados *pythonw.exe* no gerenciador de tarefas detalhado do *Windows 10*, como é possível ver no registro abaixo. O uso da *CPU* se mantinha entre 10% e 15%.

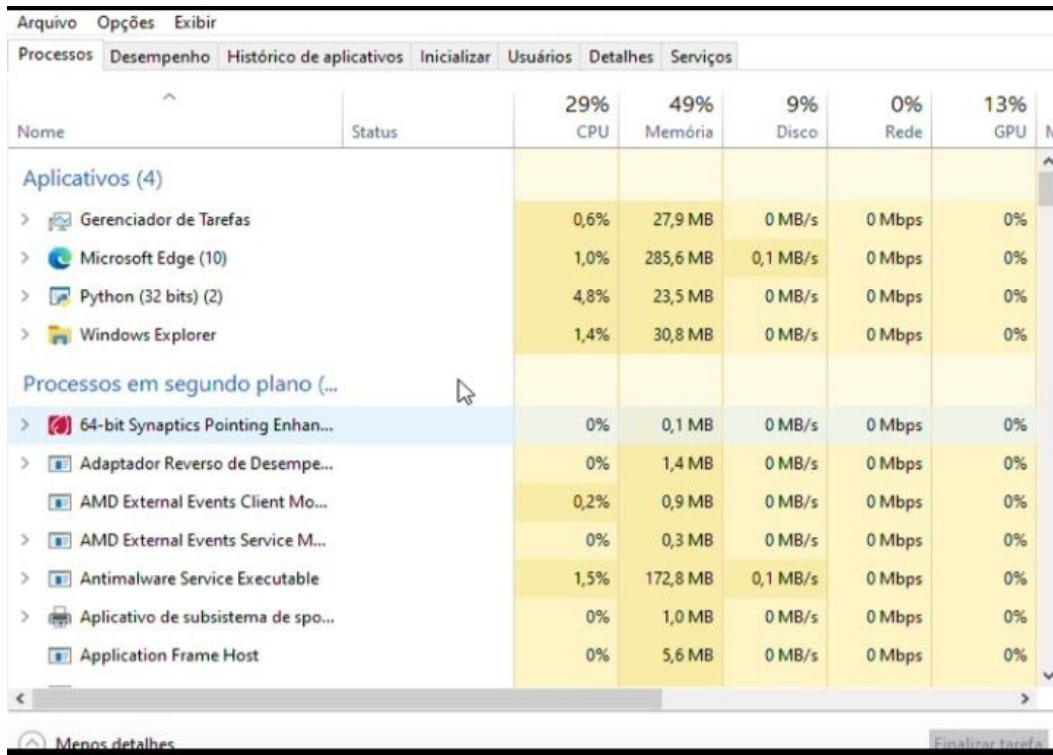
Arquivo Opções Exibir						
Processos Desempenho Histórico de aplicativos Inicializar Usuários Detalhes Serviços						
Nome	Status	11% CPU	48% Memória	7% Disco	0% Rede	12% GPU
Aplicativos (4)						
> Gerenciador de tarefas		1,5%	24,8 MB	0 MB/s	0 Mbps	0%
> Microsoft Edge (12)		1,2%	429,0 MB	0,1 MB/s	0 Mbps	0%
> Python (32 bits) (2)		0%	27,7 MB	0 MB/s	0 Mbps	0%
> Windows Explorer		0,1%	32,8 MB	0 MB/s	0 Mbps	0%
Processos em segundo plano (...)						
> 64-bit Synaptics Pointing Enhanc...		0%	0,1 MB	0 MB/s	0 Mbps	0%
> Adaptador Reverso de Desempe...		0%	1,0 MB	0 MB/s	0 Mbps	0%
AMD External Events Client Mo...		0%	0,8 MB	0 MB/s	0 Mbps	0%
AMD External Events Service M...		0%	0,3 MB	0 MB/s	0 Mbps	0%
Antimalware Service Executable		0%	111,3 MB	0 MB/s	0 Mbps	0%
Aplicativo de subsistema de spo...		0%	1,0 MB	0 MB/s	0 Mbps	0%
Application Frame Host		0%	6,1 MB	0 MB/s	0 Mbps	0%

Figura 2: Uso da CPU em condições normais

Arquivo Opções Exibir						
Processos Desempenho Histórico de aplicativos Inicializar Usuários Detalhes Serviços						
Nome	PID	Status	Nome de u...	CPU	Memória (...)	Virtualização d...
nierserver.exe	7964	Em execução	Comp	00	544 K	Desabilitado
nimdnsResponder.exe	4592	Em execução	SISTEMA	00	1.028 K	Não permitido
NisSrv.exe	1852	Em execução	SERVIÇO L...	00	1.508 K	Não permitido
nisvcloc.exe	3896	Em execução	SISTEMA	00	356 K	Não permitido
NIWebServiceContai...	5908	Em execução	SISTEMA	00	412 K	Não permitido
NIWebServiceContai...	3556	Em execução	SISTEMA	00	416 K	Não permitido
OfficeClickToRun.exe	4352	Em execução	SISTEMA	00	18.028 K	Não permitido
PresentationFontCac...	6352	Em execução	SERVIÇO L...	00	596 K	Não permitido
pythonw.exe	3780	Em execução	Comp	00	7.500 K	Desabilitado
pythonw.exe	12116	Em execução	Comp	00	16.632 K	Desabilitado
RadeonSettings.exe	7312	Em execução	Comp	00	4.984 K	Desabilitado
RAVBg64.exe	2120	Em execução	SISTEMA	00	712 K	Não permitido
RAVBg64.exe	2556	Em execução	SISTEMA	00	628 K	Não permitido
RAVBg64.exe	7976	Em execução	Comp	00	244 K	Desabilitado
RAVBg64.exe	9032	Em execução	Comp	00	820 K	Desabilitado
Registry	96	Em execução	SISTEMA	00	9.292 K	Não permitido
RegSvc.exe	3904	Em execução	SISTEMA	00	392 K	Não permitido
RemindersServer.exe	6752	Suspensão	Comp	00	0 K	Desabilitado
RtkAudioService64.exe	2904	Em execução	SISTEMA	00	372 K	Não permitido
RtkNGUI64.exe	8968	Em execução	Comp	00	1.196 K	Desabilitado
rundll32.exe	12640	Em execução	Comp	00	740 K	Desabilitado
RuntimeBroker.exe	13140	Em execução	Comp	00	6.680 K	Desabilitado
RuntimeBroker.exe	7460	Em execução	Comp	00	1.420 K	Desabilitado

Figura 3: Processo pythonw.exe

Primeiro executamos o programa contendo *Multiprocess*, constatamos alterações no uso de *CPU*, que aumentou para 30% (4,8% sendo declarado como *Python (32 bits)*) e também na quantidade de processos *pythonw.exe* no gerenciador de tarefas detalhado, totalizaram seis processos do *python 3*. O tempo de execução foi de 0.3 ms



The screenshot shows the Windows Task Manager Performance tab. The top bar indicates overall system usage: CPU at 29%, Memory at 49%, Disk at 9%, Network at 0%, and GPU at 13%. Below this, a table lists running applications and background processes with their individual resource usage.

Nome	Status	29% CPU	49% Memória	9% Disco	0% Rede	13% GPU
Aplicativos (4)						
> Gerenciador de Tarefas		0,6%	27,9 MB	0 MB/s	0 Mbps	0%
> Microsoft Edge (10)		1,0%	285,6 MB	0,1 MB/s	0 Mbps	0%
> Python (32 bits) (2)		4,8%	23,5 MB	0 MB/s	0 Mbps	0%
> Windows Explorer		1,4%	30,8 MB	0 MB/s	0 Mbps	0%
Processos em segundo plano (...)						
> 64-bit Synaptics Pointing Enhanc...		0%	0,1 MB	0 MB/s	0 Mbps	0%
> Adaptador Reverso de Desempe...		0%	1,4 MB	0 MB/s	0 Mbps	0%
> AMD External Events Client Mo...		0,2%	0,9 MB	0 MB/s	0 Mbps	0%
> AMD External Events Service M...		0%	0,3 MB	0 MB/s	0 Mbps	0%
> Antimalware Service Executable		1,5%	172,8 MB	0,1 MB/s	0 Mbps	0%
> Aplicativo de subsistema de spo...		0%	1,0 MB	0 MB/s	0 Mbps	0%
> Application Frame Host		0%	5,6 MB	0 MB/s	0 Mbps	0%

Figura 4: *Multiprocessing*

Arquivo Opções Exibir						
Processos Desempenho Histórico de aplicativos Inicializar Usuários Detalhes Serviços						
Nome	PID	Status	Nome de u...	CPU	Memória (...)	Virtualização d...
nierserver.exe	7964	Em execução	Comp	00	544 K	Desabilitado
nimdnsResponder.exe	4592	Em execução	SISTEMA	00	1.028 K	Não permitido
NisSrv.exe	1852	Em execução	SERVIÇO L...	00	1.492 K	Não permitido
nisvcloc.exe	3896	Em execução	SISTEMA	00	356 K	Não permitido
NIWebServiceContai...	5908	Em execução	SISTEMA	00	412 K	Não permitido
NIWebServiceContai...	3556	Em execução	SISTEMA	00	416 K	Não permitido
OfficeClickToRun.exe	4352	Em execução	SISTEMA	00	18.028 K	Não permitido
PresentationFontCac...	6352	Em execução	SERVIÇO L...	00	596 K	Não permitido
pythonw.exe	6816	Em execução	Comp	00	7.528 K	Desabilitado
pythonw.exe	11896	Em execução	Comp	00	3.408 K	Desabilitado
pythonw.exe	1304	Em execução	Comp	00	3.932 K	Desabilitado
pythonw.exe	1192	Em execução	Comp	00	3.272 K	Desabilitado
pythonw.exe	9128	Em execução	Comp	00	3.416 K	Desabilitado
pythonw.exe	12116	Em execução	Comp	02	16.676 K	Desabilitado
RadeonSettings.exe	7312	Em execução	Comp	00	4.984 K	Desabilitado
RAVBg64.exe	2120	Em execução	SISTEMA	00	712 K	Não permitido
RAVBg64.exe	2556	Em execução	SISTEMA	00	628 K	Não permitido
RAVBg64.exe	7976	Em execução	Comp	00	244 K	Desabilitado
RAVBg64.exe	9032	Em execução	Comp	00	820 K	Desabilitado
Registry	96	Em execução	SISTEMA	00	9.292 K	Não permitido
RegSvc.exe	3904	Em execução	SISTEMA	00	392 K	Não permitido
RemindersServer.exe	6752	Suspenso	Comp	00	0 K	Desabilitado
RtkAudioService64.exe	2904	Em execução	SISTEMA	00	372 K	Não permitido

Figura 5: Seis tarefas *pythonw.exe*

Já ao executarmos o programa contendo *Threading*, notamos um grande aumento no uso do *CPU* feito pelo *Python (32 bits)*, que ficou entre 18% e 20% durante toda a execução. O número de processos *Pythonw.exe* permaneceu no estado normal, sem surgimento de mais tarefas, permanecendo somente duas. O tempo de execução foi de 3,5 segundos.

Nome	Status	29% CPU	49% Memória	10% Disco	0% Rede	12% GPU
Aplicativos (4)						
Gerenciador de Tarefas		1,6%	24,8 MB	0 MB/s	0 Mbps	0%
Microsoft Edge (12)		0,5%	432,6 MB	0,1 MB/s	0 Mbps	0%
Python (32 bits) (2)		18,1%	28,2 MB	0 MB/s	0 Mbps	0%
Windows Explorer		0,8%	32,9 MB	0 MB/s	0 Mbps	0%
Processos em segundo plano (...)						
64-bit Synaptics Pointing Enhanc...		0%	0,1 MB	0 MB/s	0 Mbps	0%
Adaptador Reverso de Desempe...		0%	1,0 MB	0 MB/s	0 Mbps	0%
AMD External Events Client Mo...		0,1%	0,8 MB	0 MB/s	0 Mbps	0%
AMD External Events Service M...		0%	0,3 MB	0 MB/s	0 Mbps	0%
Antimalware Service Executable		0%	111,5 MB	0,1 MB/s	0 Mbps	0%
Aplicativo de subsistema de spo...		0%	1,0 MB	0 MB/s	0 Mbps	0%
Application Frame Host		0%	6,1 MB	0 MB/s	0 Mbps	0%

Figura 6: *Threading*

Constatamos assim, que o *Multiprocessing* faz uso dos diferentes núcleos do processador, dividindo a carga de processamento entre eles, otimizando o tempo de processamento, o que é constatado no surgimento de mais tarefas no gerenciador de tarefas detalhado. Já o *Threading* faz um maior uso de um único núcleo do processador, o que é observado com as altas taxas de utilização do processador feitas pelo processo *Python (32 bits)* e pelo não surgimento de novas tarefas no gerenciador de tarefas detalhado.

O uso de *Multiprocess* e *Threading*

Podemos diferenciar o conceito de paralelismo e concorrência da seguinte forma: paralelismo é fazer as coisas ao mesmo tempo, enquanto concorrência é lidar com elas ao mesmo tempo. Dessa forma é facilitado o entendimento da diferença de *Multiprocess* e *Threading*, enquanto o primeiro divide o processo em partes gerando o paralelismo em processamento(mais de uma tarefa executada por vez), o segundo cria fluxos em um mesmo processo, como múltiplas *pipelines*, diminuindo o tempo ocioso do processador, otimizando a experiência do usuário.

Assim, o *Threading* é indicado para aplicações com diversas operações de E/S, pois, enquanto uma fatia de determinado fluxo aguarda resposta por E/S, o processamento atende um outro fluxo, agilizando a execução.

Porém, em *Python 3*, o uso de múltiplas CPUs por threads é bloqueado pelo GIL (*Python global interpreter lock*), ferramenta utilizada para resolver *Bugs* de gerenciamento de memória, mas que acaba limitando o uso a uma única CPU.

Já o *Multiprocess* é mais indicado para aplicações com alta carga de processamento contínuo, pois, ainda que abrir um novo processo para cada núcleo de processamento não proporcione a economia de recursos do *Threading*, o poder de processamento é aumentado pela divisão de tarefas entre núcleos, proporcionando um tempo de execução mais otimizado.

Multiprocess burla o GIL, criando processos de multiprocessamento que vão para cada núcleo de processamento separadamente. Esses processos derivam do seccionamento do código, através das duas diferentes classes *Process* e *Pool*. Vale lembrar que as seções do código que serão multiprocessadas não podem ser dependentes de resultados anteriores, não podem necessitar de execução em ordem específica e não podem retornar nada que seja necessário ser acessado posteriormente no código.

A principal diferença entre essas classes é: A classe *Process* envia cada tarefa para um processador diferente. Enquanto a classe *Pool* envia conjuntos de tarefas para processadores diferentes.

Entendemos que *Multiprocess* e *Threading* são duas ferramentas poderosas, e quando usadas juntas, otimizam e muito a execução de um processo, aumentando o paralelismo e melhorando como um todo a experiência do usuário e o tempo de resposta.