UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

Facultad de Ingeniería

Ingeniería en Ciencias de la Computación

COMPUTO PARALELO Y DISTRIBUÍDO

1.46 Actividad 14: Multithreading vs Multiprocessing

Trabajo de: ADRIAN (ADORA) GONZÁLEZ DOMÍNGUEZ *[359834]*

Asesor: JOSE SAUL DE LIRA MIRAMONTES

1 de octubre de 2024

Aspecto	**Multithreading**	**Multiprocessing**
Definición	Implica múltiples hilos que comparten el mismo espacio de memoria dentro de un solo proceso.	Implica múltiples procesos, cada uno con su propio espacio de memoria.
Uso de Memoria	Menor uso de memoria, ya que los hilos comparten la memoria.	Mayor uso de memoria, ya que cada proceso tiene su propio espacio de memoria.
Rendimiento (Tareas dependientes de CPU)	Menos efectivo para tareas dependientes de la CPU debido al GIL (Global Interpreter Lock) en Python.	Más efectivo para tareas dependientes de la CPU, ya que cada proceso puede ejecutarse en un núcleo de CPU separado.
Rendimiento (Tareas dependientes de I/O)	Mejor rendimiento para tareas dependientes de I/O (p. ej., lectura de archivos, redes).	Menos efectivo para tareas dependientes de I/O debido al sobrecoste de crear procesos.
Paralelismo o concurrencia	Implementa concurrencia.	Implementa paralelismo, ya que cada proceso puede ejecutarse de forma independiente.
En computación paralela	Python no soporta multithreading.	Python soporta multiprocessing.
Sobrecarga	Menor sobrecarga en el cambio de contexto y creación de hilos.	Mayor sobrecarga en la creación de procesos y la comunicación entre procesos.
Casos de Uso	Mejor para tareas que implican operaciones de I/O	Mejor para tareas que requieren un uso intensivo de la CPU y cálculo pesado.

	(p. ej., lectura de archivos, redes).	
Ejemplos de Tareas	Web scraping, procesamiento de archivos, manejo de solicitudes de usuarios.	Cálculos científicos, procesamiento de imágenes, análisis de datos.

```
import time, os
from threading import Thread, current_thread
from multiprocessing import Process, current_process

COUNT = 200000000
SLEEP = 10

def io_bound(sec):

    pid = os.getpid()
    threadName = current_thread().name
    processName = current_process().name

    print(f"{pid} * {processName} * {threadName} \
        ---> Start sleeping...")
    time.sleep(sec)
    print(f"{pid} * {processName} * {threadName} \
        ---> Finished sleeping...")

def cpu_bound(n):

    pid = os.getpid()
    threadName = current_thread().name
    processName = current_process().name

    print(f"{pid} * {processName} * {threadName} \
        ---> Start counting...")
```

```
    while n>0:
        n -= 1

    print(f"{pid} * {processName} * {threadName} \
        ---> Finished counting...")

def test_time(execute):
    start = time.time()
    execute()
    end = time.time()
    print('Time taken in seconds -', end - start)

def test_io():
    io_bound(SLEEP)
    io_bound(SLEEP)

def test_io_threads():
    t1 = Thread(target = io_bound, args =(SLEEP, ))
    t2 = Thread(target = io_bound, args =(SLEEP, ))
    t1.start()
    t2.start()
    t1.join()
    t2.join()

def test_io_processes():
    p1 = Process(target = io_bound, args =(SLEEP, ))
    p2 = Process(target = io_bound, args =(SLEEP, ))
    p1.start()
    p2.start()
    p1.join()
    p2.join()

def test_cpu():
    cpu_bound(COUNT)
    cpu_bound(COUNT)

def test_cpu_threads():
    t1 = Thread(target = cpu_bound, args =(COUNT, ))
    t2 = Thread(target = cpu_bound, args =(COUNT, ))
```

```
    t1.start()
    t2.start()
    t1.join()
    t2.join()

def test_cpu_processes():
    p1 = Process(target = cpu_bound, args =(COUNT, ))
    p2 = Process(target = cpu_bound, args =(COUNT, ))
    p1.start()
    p2.start()
    p1.join()
    p2.join()

if __name__=="__main__":
    test_time(test_io)
    test_time(test_io_threads)
    test_time(test_io_processes)
    test_time(test_cpu)
    test_time(test_cpu_threads)
    test_time(test_cpu_processes)
```

Tiempos de ejecución			
	Secuencial	Multithreading	Multiprocessing
Tareas de IO	20.00s	10.01s	10.11s
Tareas de CPU	15.57s	14.75s	9.48s

```
$ python PDC.October_1_2024.Activity_1.46.py
73708 * MainProcess * MainThread         ---> Start sleeping...
73708 * MainProcess * MainThread         ---> Finished sleeping...
73708 * MainProcess * MainThread         ---> Start sleeping...
73708 * MainProcess * MainThread         ---> Finished sleeping...
Time taken in seconds - 20.00798511505127
73708 * MainProcess * Thread-1 (io_bound)         ---> Start sleeping...
73708 * MainProcess * Thread-2 (io_bound)         ---> Start sleeping...
73708 * MainProcess * Thread-1 (io_bound)         ---> Finished sleeping...73708 * MainProcess * Thre
ad-2 (io_bound)         ---> Finished sleeping...

Time taken in seconds - 10.012621641159058
126564 * Process-1 * MainThread         ---> Start sleeping...
26092 * Process-2 * MainThread         ---> Start sleeping...
126564 * Process-1 * MainThread         ---> Finished sleeping...
26092 * Process-2 * MainThread         ---> Finished sleeping...
Time taken in seconds - 10.118456602096558
```

```
73708 * MainProcess * MainThread          ---> Start counting...
73708 * MainProcess * MainThread          ---> Finished counting...
73708 * MainProcess * MainThread          ---> Start counting...
73708 * MainProcess * MainThread          ---> Finished counting...
Time taken in seconds - 15.573946952819824
73708 * MainProcess * Thread-3 (cpu_bound)          ---> Start counting...
73708 * MainProcess * Thread-4 (cpu_bound)          ---> Start counting...
73708 * MainProcess * Thread-4 (cpu_bound)          ---> Finished counting...
73708 * MainProcess * Thread-3 (cpu_bound)          ---> Finished counting...
Time taken in seconds - 14.756812334060669
152952 * Process-3 * MainThread          ---> Start counting...
49880 * Process-4 * MainThread          ---> Start counting...
152952 * Process-3 * MainThread          ---> Finished counting...
49880 * Process-4 * MainThread          ---> Finished counting...
Time taken in seconds - 9.48815631866455
```