



Universidad de Guadalajara.
Centro Universitario de Ciencias Exactas e Ingenierías.



Ejercicio 4.1

An Introduction to Scaling Distributed Python Applications

Computación tolerante a fallas NRC 179961

Nombre del alumno: Loredó Padilla Orlando Javier

Código: 217560328

Departamento: Departamento de ciencias computacionales

Nombre del maestro: López Franco Michel Emanuel

Carrera: Ingeniería en computación

Sección: D06

Ciclo: 23B

Fecha: 11/9/2023.

Introducción

Utilizaremos los hilos en Python para ver el funcionamiento de estos y como pueden aplicarse a programas muchos más complejos para hacer sistemas tolerantes a fallas.

Programa

Es programa es demasiado sencillo, intenté simular múltiples procesos usando hilos, para ello cargué 3 for con diferentes sleep para que los hilos actúen en lo que alguna función espera su turno para actuar. En el programa el proceso principal actuará cada 10 segundos entonces en esos lapsos de tiempo los demás estarán trabajando a su propio ritmo.

```
"""
import threading
import time

def primera_descarga(something):
    for i in range(0, 51):
        print("+Descarga en progreso:", i*2, "%")
        if i == 50:
            print("+++++Descarga finalizada+++++ \n")
            time.sleep(1)

def segunda_descarga(something):
    for i in range(0, 26):
        print("-Actualización de datos: ", i*4, "%")
        if i == 25:
            print("-----Actualización finalizada-----\n")
            time.sleep(3)

t = threading.Thread(target=primera_descarga, args=("hello",))
t2 = threading.Thread(target=segunda_descarga, args=("hello",))
print("Iniciando procesos \n")
t.start()
t2.start()
for j in range(0, 11):
    print("/Subida de archivos: ", j*10, "%")
    if j == 10:
        print("////////Archivo subido////////")
        time.sleep(10)

t.join()
t2.join()
print("Procesos finalizados")
```

Aquí se puede ver que por ejemplo por cada proceso de subida la descarga se ejecutará 10 veces.

Procesos finalizados

```
In [4]: runfile('E:/Escuela/codigos/hilos.py', wdir='E:/Escuela/codigos')
```

Iniciando procesos

```
+Descarga en progreso: 0 %  
-Actualización de datos: 0 %  
/Subida de archivos: 0 %  
+Descarga en progreso: 2 %  
+Descarga en progreso: 4 %  
-Actualización de datos: 4 %  
+Descarga en progreso: 6 %  
+Descarga en progreso: 8 %  
+Descarga en progreso: 10 %  
-Actualización de datos: 8 %  
+Descarga en progreso: 12 %  
+Descarga en progreso: 14 %  
+Descarga en progreso: 16 %  
-Actualización de datos: 12 %  
+Descarga en progreso: 18 %  
/Subida de archivos: 10 %  
+Descarga en progreso: 20 %  
+Descarga en progreso: 22 %  
-Actualización de datos: 16 %  
+Descarga en progreso: 24 %  
+Descarga en progreso: 26 %  
+Descarga en progreso: 28 %  
+Descarga en progreso: 30 %
```

De tal modo que terminaran de manera casi simultánea.

```
+Descarga en progreso: 32 %  
+Descarga en progreso: 98 %  
/Subida de archivos: 50 %  
+Descarga en progreso: 100 %  
+++++Descarga finalizada++++++  
  
-Actualización de datos: 68 %  
-Actualización de datos: 72 %  
-Actualización de datos: 76 %  
/Subida de archivos: 60 %  
-Actualización de datos: 80 %  
-Actualización de datos: 84 %  
-Actualización de datos: 88 %  
-Actualización de datos: 92 %  
/Subida de archivos: 70 %  
-Actualización de datos: 96 %  
-Actualización de datos: 100 %  
-----Actualización finalizada-----  
  
/Subida de archivos: 80 %  
/Subida de archivos: 90 %  
/Subida de archivos: 100 %  
////////Archivo subido////////  
Procesos finalizados
```

Conclusión

Los hilos son una buena técnica para ejecutar tareas de manera mucho más eficiente, si existen tareas en las cuales existen espacios en lo que no se realiza nada colocar hilos para que otros procesos se puedan ejecutar en esos descansos llega a ser útil, estaríamos aprovechando todo el tiempo muerto que existe al momento de hacer una sola tarea. La aplicación que tiene a la tolerancia a fallas es que si un proceso llega a fallar no estará interrumpiendo la ejecución de otros, si bien comparten el espacio de trabajo no están vinculados por lo cual si perdemos el progreso de uno no perderemos por completo el tiempo, ya que estuvimos trabajando en otras cosas al mismo tiempo.

Referencias

- Fawcett A. (2021). An Introduction to Scaling Distributed Python Applications en educative. Recuperado el 9/9/2023 en: <https://www.educative.io/blog/scaling-in-python>