Laboratorio: Concurrencia y paralelismo en Python

Objetivos

El objetivo de esta actividad es probar los distintos mecanismos que existen en Python para simular procesos de acceso en paralelo o concurrente y para acelerar un código con grandes necesidades de cálculo y poder comparar los rendimientos.

Se aplicarán los distintos métodos tanto a la ejecución de algoritmos como al procesamiento de datos.

Para el desarrollo de esta actividad es necesario haber estudiado previamente los temas 6 y 7 de la asignatura, profundizando especialmente en las funciones de creación de hilos y procesos, sus formas de sincronización y comunicación y las decisiones de paralelización.

Descripción de la actividad.

La actividad consta de tres tareas.

1. Simulación de un problema de sincronización de hilos.
2. Ejecución de una función sobre un conjunto de datos grande de forma secuencial, con hilos y con Pool de procesos. Análisis de rendimientos

Tarea 1. Simulación de sincronización entre hilos

Se va a tratar el problema clásico del barbero durmiente que simula lo que ocurre con algunos procesos gestionados por el sistema operativo. En este problema se simula el funcionamiento de una barbería.

Mientras no hay clientes en la barbería, el barbero duerme. Cuando un cliente entra, si el barbero está durmiendo pasa directamente al sillón del barbero a cortarse el pelo, pero si el barbero está ocupado se sienta en la sala de espera hasta que le puedan atender. En la sala solo hay tres sillas, luego como máximo puede haber tres clientes esperando. El cliente que no consigue entrar espera fuera de la puerta. **Opcional, se puede implementar el protocolo de que el cliente que no pueda entrar porque la sala esta llena o no vuelva o se vaya y lo reintente después de un tiempo**. **No es necesario implementar esto último.**

Una vez que se cortan el pelo, los clientes se van.

Este problemas se va a resolver mediante programación concurrente usando los hilos de threading. Para poder hacer las implementaciones puede usar el código implementado en el notebook **BarberoDurmienteLaboratorio. ipynb** que se adjunta en la actividad.

En esta tarea se plantea:

1. Defina los elementos que faltan en la implementación proporcionada para completar la simulación concurrente, explique con comentarios en el notebook las decisiones que ha tomado y prueba el ejemplo con 1 barbero y 6 clientes. Concretamente se pide lo siguiente:
   * Definición e inicialización de semáforos, bloqueos, condiciones y variables compartidas.
   * Definición e inicialización de hilos.
   * Implementación de los protocolos antes de cortar y después de cortar que realizará el barbero.
   * Implementación de los protocolos al entrar en la barbería, protocolo de espera en la sala si es el caso, protocolo de salida de la sala y entrada al sillón de cortar .

Tarea 2. Procesamiento de un corpus de texto

Una de las aplicaciones del paralelismo en el que se consigue un rendimiento considerables es en el procesamiento de lenguaje natural, en el que se deben realizar distintas tareas que grana intensidad de cómputo. En esta tarea se va a cargar uno de los corpus que proporciona NLTK, concretamente el corpus *Brown,* sobre un DataFrame*.*

En ele corpus se realizará un procesamiento con varias transformaciones en las palabras del texto y además contará el número de palabras de cada frase y se añadirá como columna al DataFrame.

Se proporciona el código base para poder realizar la implementación. Este código se debe de ejecutar sobre el interprete o bien usar algún entorno de desarrollo como *Spyder*, pero no se podrá ejecutar sobre un *notebook* .

En el notebook **procesamientocorpus.ipynb**, se explican las distintas partes del código dejando los huecos en los que se debe añadir la implementación pedida para esta tarea.

El código será ejecutado en secuencial y por un Pool de procesos. Se deben comparar los rendimientos

Por tanto en esta tarea se pide:

* + Implementación de las instrucciones de transformación
  + Implementar la función que cuenta palabras e incorporar la columna al DataFrame
  + Implementación de la ejecución en secuencial
  + Implementación de la ejecución paralela mediante un pool de procesos.
  + Mostrar los resultados de algunas de las frases del corpus en la ejecución secuencial y paralela
  + Mostrar los tiempos de ejecución.
  + Comparar y analizar los rendimientos

\***Importante: Todo el código debe estar comentado y se debe explicar las decisiones tomadas**

Para esta tarea se debe:

1. Adjuntar los archivos o notebooks que se describe a continuación. Aunque no se pueda ejecutar el código paralelo.
2. **El código Python se debe entregar en un archivo .py** para que pueda ser ejecutado por el docente.

\***Es importante consultar la rúbrica**

Forma de entrega.

En este trabajo se deberá entregar:

* 1. **Notebook de la tarea 1,** debidamente comentado y explicado.
  2. **Notebook de la tarea 2 o archivo de texto editable** debidamente comentado y explicado. Se podrá incluir y ejecutar la parte secuencial. Mostrando resultados y análisis de los rendimientos.
  3. **Documento** en el que se explique la parte paralela, mostrando re4sulatdos. Se puede incluir en el notebook anterior aunque no se pueda ver la ejecución o en un archivo de texto editable, no pdf.
  4. **Archivo de código Python (.py) con todo el código de las dos partes para que pueda ser ejecutado.**

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Recorridos iterativos de un árbol binario | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Tarea 1 (definiciones e inicializaciones) | * Se definen e inicializan todos los mecanismos de sincronización * Se definen e inicializan todos los hilos * Se realiza el análisis de la ejecución | 1 | 10% |
| Tarea 1 (implementación del barbero) | * Se implementa el protocolo inicial del barbero, antes de atender a un cliente * Se implementa el protocolo del barbero tras atender a un cliente * Se explican todas las implementaciones y las decisiones tomadas | 2 | 20% |
| Tarea 1  (implementación de los clientes) | * Se implementa el protocolo inicial del cliente, cuando llega a la barbería * Se implementa el mecanismo para despertar al barbero * Se implementa el protocolo del cliente cuando se le da acceso al sillón de corte * Se explican todas las implementaciones y las decisiones tomadas | 2 | 20% |
| Tarea 2  (funciones) | * Se implementan las funciones pedidas y la función principal * Se explican todas las implementaciones y las decisiones tomadas | 1 | 10% |
| Tarea 2  ( ejecución secuencial) | * Se implementa la ejecución secuencial * Se muestran resultados parciales * Se obtiene el tiempo de ejecución * Se explican todas las implementaciones y las decisiones tomadas | 1 | 10% |
| Tarea 2  (ejecución paralela) | * Se implementa la ejecución paralela, con el pool de procesos * Se muestran resultados parciales * Se obtiene el tiempo de ejecución * Se explican todas las implementaciones y las decisiones tomadas | 2 | 20% |
|  | * Se hace el análisis del rendimiento y se extraen conclusiones | 1 | 10% |
|  |  | **10** |  |