

Programación Paralela: Práctica de laboratorio 2

Lenin Pavón Alvarez ¹, Lucía Martínez Rivas², Miguel Ángel Perez de León³

¹Nikpalewia in notemachtiane Laboratorio, Facultad de Ciencias,
UNAM

²Nikpalewia in notemachtiane, Facultad de Ciencias, UNAM

³Temachtiane, Facultad de Ciencias, UNAM

Febrero 2024

Índice

1	Introducción	1
1.1	Requerimientos	2
1.2	Formato de entrega	2
2	Conceptos básicos de Arquitectura de Computadoras	2
3	Paradigmas de memoria	3
4	Ejercicios de programación	4
4.1	Python	4
A	Agradecimientos	4

1. Introducción

Esta es la primera práctica de laboratorio del Seminario de Ciencias de la Computación A: *Introducción a la programación en paralelo con MPI, OpenMP y CUDA*. Está dividida en dos partes: el marco teórico y la parte práctica.

1.1. Requerimientos

Para realizar esta práctica se debe contar con acceso a internet y a un equipo de cómputo, ya sea local o remoto.

1.2. Formato de entrega

Se deben de subir los siguientes archivos al repositorio de GitHub en un directorio un directorio con tu nombre y primer apellido (e.g. LeninPavon):

1. Un archivo de markdown (o en su defecto un pdf) respondiendo a las preguntas teóricas.
2. El código fuente correspondiente a los ejercicios de programación con terminación adecuada

Otro formato de entrega aceptable es

1. Un notebook (archivo con extensión .ipynb) donde se respondan las preguntas teóricas, y se encuentre el código fuente correspondiente a la sección de Python

2. Conceptos básicos de Arquitectura de Computadoras

En este curso nos interesa entender a un dispositivo de cómputo (computadora) como un equipo que puede realizar las siguientes funciones:

1. Almacenamiento temporal (memoria, caché, registros)
2. Control (contadores, tablas)
3. Operaciones (ALU)
4. Comunicaciones (Transductores)

Estas funciones requieren de diferentes elementos físicos para cumplir estas funciones. A nivel **computadoras** la podemos distribuir físicamente a lo largo de varios *nodos* con sus respectivas unidades de *almacenamiento* y formas de *conectividad*. Cada **nodo** cuenta con *procesadores*, *memoria* y *métodos de entrada y salida*¹

A nivel **procesador** deben de existir **registros**, **unidades de lógica y aritmética** y una **unidad de control** que corren a diferentes **ciclos de reloj**.

¹En inglés, entrada se dice “input” y a la salida de un comando se le llama “output”, cuando se vean las iniciales I/O se debe de pensar en métodos de entrada de argumentos de un comando y de salida del mismo.

Cuando se quiere preservar un resultado de un comando, la representación de la salida se debe almacenar en un espacio físico dentro de un **dispositivo de memoria**, que dependiendo del presupuesto y necesidades puede variar en tipo, velocidad de escritura y eficiencia. Además diferentes **protocolos** pueden usarse para almacenar dichos resultados.

Por último, si queremos que la computadora se pueda comunicar entre sí debemos de físicamente decidir dónde colocar las conexiones, es decir debemos de tomar decisiones con respecto de la **topología** de la red y qué **protocolos de red** se van a seguir.

La arquitectura de una computadora es

“La forma de organizar todos los elementos de una computadora”

Así la arquitectura va a ser siempre una consecuencia del uso que se le quiera dar al equipo. Los bancos suelen tener equipos con muy pocos procesadores, son equipos diseñados para correr de manera ininterrumpida y sin falla alguna por varios años: son maratónicos. En cambio, las supercomputadoras suelen tener fallos en el equipo de manera constante, por lo que usan dispositivos y protocolos resilientes a la pérdida parcial de la información.

Investiga la respuesta a las siguientes preguntas

1. ¿Qué significa Atócatl?
2. ¿Cuántos años tiene el equipo?
3. ¿Cuántos nodos tiene Atócatl? ¿Qué funciones tiene cada nodo?
4. ¿Cuáles son las especificaciones de Atócatl?
5. ¿Qué dispositivos de almacenamiento tiene Atócatl?
6. ¿Se respalda toda la información? ¿Por qué?
7. ¿Qué tipo de conexiones tiene Atócatl? ¿Por qué?

3. Paradigmas de memoria

8. ¿Todas las unidades de procesamiento que tiene Atócatl son CPU?
9. ¿Cómo se diferencia el uso de memoria en una CPU a comparación de una GPU?
10. ¿Cuál es la diferencia entre usar una GPU y usar varios procesadores tipo CPU con MPI?

11. ¿Qué tipos de problemas puedes resolver con OpenMP?
12. ¿Qué tipos de problemas puedes resolver con MPI?
13. ¿Qué tipos de problemas puedes resolver con CUDA?

4. Ejercicios de programación

4.1. Python

Think Python ejercicio 7.3 Escribe un programa para calcular la raíz cuadrada de un número y compara con respecto al resultado generado por la función `math.sqrt` y calcula el error entre ambas funciones. También responde donde queda este problema en la taxonomía de Flynn, si añadimos más datos dónde quedaría y si es paralelizable este programa

Escribe un script en Shell que reciba como primer argumento tu nombre y del segundo al término de los argumentos ejecute el script de python que escribiste en el inciso anterior.

Advent of Code 2023 día 1 Escribe un código en C que indique el primer y último dígito en una cadena de texto.

A. Agradecimientos

Esta práctica es posible solamente gracias a la generosa formación brindada en el marco del Programa de Formación de Becarios en Tecnologías de la Información y la Comunicación de la Dirección General de Tecnologías de la Comunicación de la UNAM. Particularmente, se basan en las notas del curso de “Arquitectura de Computadoras” impartido por José Luis Gordillo a inicios de 2024.