Parallelism & Concurrence Part 1

Introducción

La programación concurrente y paralela surge de la necesidad de optimizar el rendimiento de los sistemas computacionales. Tradicionalmente, la programación ha sido secuencial, basada en un solo procesador que ejecuta instrucciones en un orden específico y determinista. Sin embargo, con el aumento de la demanda de aplicaciones más rápidas y eficientes, se ha recurrido a la concurrencia y el paralelismo para mejorar el rendimiento.

Conceptos y Definiciones

1. `Programación Secuencial`:
   - Ejecuta instrucciones una tras otra en un solo procesador.
   - Es determinista: Con los mismos datos de entrada, siempre produce el mismo resultado.
   - Sigue el modelo de Von Neumann con una solo línea de control de flujo.
2. `Programación Concurrente`:
   - Se originó en los años 60 junto con los SO y la programación en sistemas.
   - Permite que múltiples procesos se ejecuten aparentemente al mismo tiempo.

- ▼ No es determinista: el resultado puede variar debido a la ejecución simultanea de procesos.

3. `Programación Paralela`:
   - ▼ Implica la ejecución de múltiples procesos en varios procesadores.
   - ▼ Es físicamente concurrente, a diferencia de la concurrencia que puede ser lógica.

Acontecimientos que Impulsaron la Programación Concurrente

- **Concepto de hilos**: Permite ejecutar programas más rápidos comparado con los procesos.
- **Lenguajes Orientados a Objetos**: Como Java, que soportan primitivas de concurrencia.
- **Internet**: Aplicaciones como navegadores o chats utilizan técnicas concurrentes.

Concurrencia vs. Paralelismo

- **Concurrencia**: Es la capacidad de gestionar varias tareas que pueden ejecutarse de manera intercalada en el tiempo.
- **Paralelismo**: Es la capacidad de ejecutar varias tareas al mismo tiempo en múltiples procesadores.

Área de aplicación

- Visualización
- Base de Datos Distribuidas
- Simulaciones
- Predicciones científicas

Clasificación de Arquitecturas

- **Sistemas MIMD**: Con memoria distribuida y red de interconexión dedicada (Massive Parallel Processing - MPP).
- **Topologías**: Anillo, árbol, malla, hipercubo, etc.

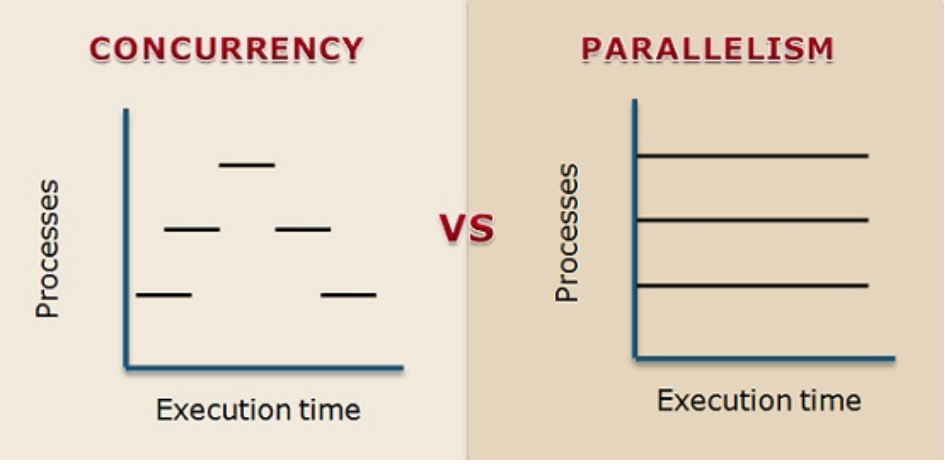
Características Deseadas de una Aplicación Paralela

- Alto rendimiento
- Eficiencia
- Escalabilidad
- Transparencia
- Portabilidad

Ejemplo Gráfico

Imagina una fábrica con varias líneas de producción:

- **Concurrencia**: La línea de producción tiene diferentes estaciones (procesos) que trabajan en partes diferentes de un producto, pero solo hay una herramienta (CPU) que se pasa entre estaciones. Aun que solo una estación usa la herramienta a la vez, el producto se mueve entre estaciones (procesos) rápidamente, creando la apariencia de trabajo simultaneo.
- **Paralelismo**: Varias líneas de producción, cada una con sus propias herramientas (CPUs), trabajando en diferentes productos al mismo tiempo. Aquí, múltiples productos se realizan al mismo tiempo.



Conclusión

La programación concurrente y paralela es esencial para desarrollar aplicaciones eficientes y rápidas en el mundo moderno. Saber la diferencia entre concurrencia y paralelismo, y cómo aplicarlos correctamente, permite diseñar sistemas que maximicen el uso de los recursos disponibles, mejorando el rendimiento y la capacidad de respuesta de las aplicaciones.