

Algoritmos y Programación Paralela

Dra. Ing. Ana Cori Morón

CLASE 1

Historia y optimización

- A nivel de hardware
 - Antes un CPU, conjunto de instrucciones
 - Luego aumento la Velocidad y cantidad de memoria
 - Aumento de Velocidad y cantidad de almacenamiento
 - Aumento de Velocidad y redes
- A nivel de software
 - Paralelización de código
 - Código distribuido
 - Optimización de código

Introducción

- Porción de código de programación secuencial:
- ¿Qué partes se pueden paralelizar?

- Mas ejemplos

- Otros ejemplos de actividades cotidianas
- Preparar una ensalada, cavar hoyos en un
- jardin

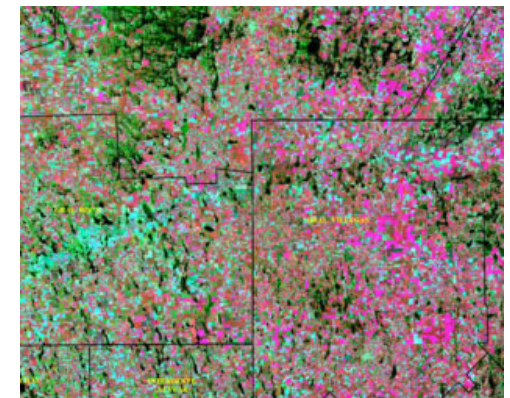
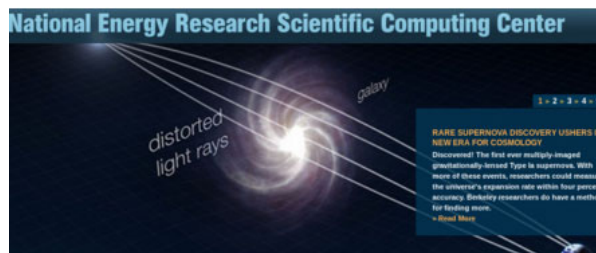
```
a = 0
Para i = 0 hasta n-1
    a = a + x[i]
Fin_para
b=0
Para i = 0 hasta n-1
    b = b + y[i]
Fin_para
Para i = 0 hasta n-1
    z[i] = x[i] / b + y[i] / a
Fin_para
Para i = 0 hasta n-1
    y[i] = (a + b) * y[i]
Fin_para
```

Aplicaciones

- Aproximación y cálculo de constantes y/o funciones numéricas.
- Predicción del clima y cambio climático
- Análisis de imágenes multispectro
- Astronomía
- Simular parques eólicos para saber cuanta energía eléctrica generan.

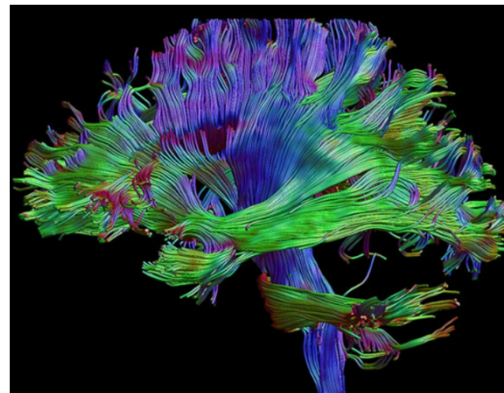


**National Oceanic and
Atmospheric Administration**
U.S. Department of Commerce



Aplicaciones

- Para un matriz hallar El determinante La traspuesta La inversa
- Ciencia de datos
- Aeropuertos
- Renderizado de imágenes y animación
- Diagnostico de enfermedades mentales a través de resonancia magnética
- Diseño industrial y robótica



INTRODUCCIÓN

- En la década de los 60 se presentan ciertos tipos de problemas, los cuales eran solucionados de manera ineficiente (secuencial).
- En 1972 el científico Brinch Hansen crea el lenguaje de alto nivel Concurrent Pascal para solucionar los problemas de manera más eficiente.
- Hitos que marcan el inicio de la programación concurrente:
 - La aparición de los hilos o threads
 - La aparición del lenguaje JAVA
 - La aparición del internet

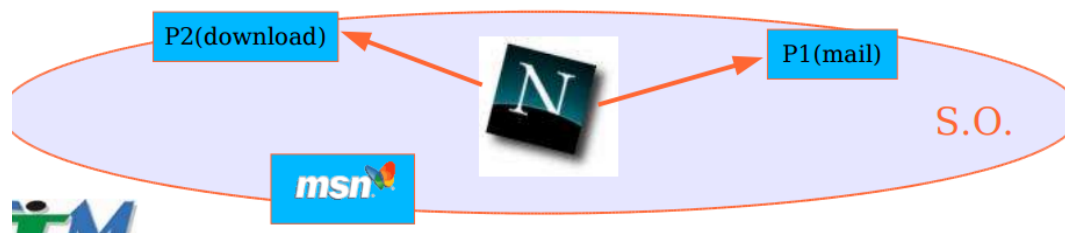
Conceptos fundamentales



- Concurrencia
 - Según la RAE; acaecimiento o concurso de varios sucesos en un mismo tiempo.
- Coincidencia, concurso simultáneo de varias circunstancias.
- Juntarse en un mismo lugar o tiempo.
- Coincidir en el espacio o en el tiempo dos o mas personas o cosas.
- Si reemplazamos la palabra sucesos por la palabra procesos.
- En informática, se da cuando dos o mas procesos existen simultáneamente.

Conceptos fundamentales

- Programa; es un conjunto de instrucciones, es algo estático, puede compararse con la palabra CLASE en el ámbito de la POO.
- Proceso; es la ejecución de un programa o parte de un programa, es algo dinámico, puede compararse con la palabra OBJETO en el ámbito de la POO.
- Por ejemplo consideremos un servidor de aplicaciones donde reside una aplicación de navegador de internet y existen varios usuarios ejecutando ese navegador cada uno navegando por sitios web diferentes.



Concurrencia

- Según Palma Méndez dos procesos son concurrentes cuando la primera instrucción de uno de ellos se ejecuta después de la primera instrucción del otro proceso y antes de la última.
- Es decir existe un solapamiento en la ejecución de las instrucciones.
- **La concurrencia aparece cuando dos o más procesos son contemporáneos. Un caso particular es el paralelismo (programación paralela).**
- Si ambos procesos se ejecutan al mismo tiempo entonces hablamos de **programación paralela**.
- Un programa al ponerse en ejecución puede dar lugar a mas de un proceso, cada uno de ellos ejecutando una parte del programa.
- Continuando con el ejemplo del programa de navegador de internet puede dar lugar a varios procesos, uno que controla las acciones del usuario, otro que controla las peticiones al servidor, etc.

Programación Concurrente

- Un concepto mas acertado de PROCESO, es, la actividad asíncrona susceptible de ser asignada a un procesador.
- Un SSOO, es un programa con varios procesos en ejecución al mismo tiempo, planificador, controlador de HD, de impresora, etc.
- Los procesos pueden competir o colaborar entre sí por los recursos del sistema. Para llevar a cabo las tareas de colaboración y competencia entre los procesos existen **mecanismos de comunicación y sincronización.**

Programación Concurrente

- Es la disciplina que se encarga del **estudio de las notaciones y técnicas que describen la ejecución concurrente de las instrucciones de un programa y resuelve los problemas de sincronización y comunicación** que son inherentes a la ejecución concurrente.
- Se ocupa del análisis, diseño, implementación y depuración de programas concurrentes.
- No se ocupa del hardware sobre el cual dichos programas concurrentes se ejecutan.

Programación Concurrente

- Según Ricardo Galli, es la composición de módulos que se ejecutan independientemente, de forma asíncrona y no determinista.
- Cada módulo se ejecuta como un proceso independiente y asíncrono.

Ventajas de la Programación Concurrente

- Mejor aprovechamiento de la CPU
- Velocidad de ejecución, al subdividir un programa en procesos, éstos se pueden “repartir” entre procesadores o gestionar en un único procesador según importancia.
- Solución de problemas inherentemente concurrentes
 - Sistemas de control, Captura de datos, análisis, sistemas de tiempo real, control de vuelo, monitorización de pacientes, reservas online
 - Tecnologías web, servidores de chat, correo.
 - Aplicaciones basadas en interfaces de usuarios, el usuario hace varias peticiones a la aplicación gráfica, navegador web
 - Simulación
 - SGBD

Sistemas informáticos concurrentes

- Sistemas Operativos
- Sistemas de Gestión de BD.
- Sistemas de tiempo real.
- Sistemas distribuidos.

Lenguajes

Pascal, C, Fortran , Cobol



Secuenciales



único hilo de control

Java, Ada, Posix



Concurrentes



varios hilos de control

Desventajas

- El trabajar con procesos concurrentes añade complejidad a la tarea de programar.
- Los programas paralelos son más difíciles de escribir que los programas secuenciales, ya que se requiere que haya una comunicación y sincronización entre las tareas que se han paralelizado.
- Mayor consumo de energía
- Altos costos por producción y mantenimiento

Programación Paralela

- Es el uso de varios procesadores trabajando juntos para resolver una tarea común.
- El modo de uso de los procesadores puede ser diseñado por el programador:
 - Cada procesador trabaja en una porción del problema.
 - Los procesos pueden intercambiar datos, a través de la memoria o por una red de interconexión
- El paralelismo es una forma de ejecutar programas concurrentes. La programación concurrente es una forma de estructurar los programas, no el número de procesadores que se usa para su ejecución.

Ventajas Programación Paralela

- La programación paralela permite:
 - Resolver problemas que no caben en una CPU.
 - Resolver problemas que no se resuelven en un tiempo razonable
- Se pueden ejecutar
 - Problemas mayores
 - Más rápidamente (aceleración)
 - Más problemas

Programa concurrente, paralelo y distribuido

- Programa concurrente, es el conjunto de acciones o instrucciones que se pueden ejecutar simultáneamente.
- Programa paralelo, es un programa concurrente diseñado para ejecutarse en un sistema multiprocesador.
- Programa distribuido, es un programa paralelo diseñado para ejecutarse en un sistema distribuido, es decir en una red de procesadores que no comparten memoria.

Ejemplos

- Cálculo secuencial, concurrente (1 CPU) y paralelo (2 CPU) del número PI.
- Cálculo de la suma de los elementos de un vector de forma secuencial, concurrente y paralela.
- Multiplicación de matrices

Tiempos de ejecución (PI)

3.1416026534897203	Secuencial	Concurrente	Paralelo
Real	<u>0m0.032s</u>	0m0.689s	0m1.497s
User	<u>0m0.027s</u>	0m0.333s	0m0.493s
Sys	<u>0m0.002s</u>	0m0.347s	0m1.210s

Tiempos de ejecución (Vector)

887459712	Secuencial	Concurrente	Paralelo
Real	0m3.694s	0m3.725s	<u>0m3.179s</u>
User	0m3.031s	0m3.068s	<u>0m3.025s</u>
Sys	<u>0m0.626s</u>	0m0.632s	0m0.638s

Trabajo en clase

- Porción de código de programación secuencial:
- ¿Qué partes se pueden paralelizar?

- Mas ejemplos

- Otros ejemplos de actividades cotidianas
- Preparar una ensalada, cavar hoyos en un
- jardin

```
a = 0
Para i = 0 hasta n-1
    a = a + x[i]
Fin_para
b=0
Para i = 0 hasta n-1
    b = b + y[i]
Fin_para
Para i = 0 hasta n-1
    z[i] = x[i] / b + y[i] / a
Fin_para
Para i = 0 hasta n-1
    y[i] = (a + b) * y[i]
Fin_para
```

TAREA

- LEER CAPITULO I TOMAS PALMA
- TRAER DOS EJEMPLOS DE UN PROGRAMA QUE SE PUEDE PARALELIZAR Y 2 EJEMPLOS DE SOFTWARE PARA DISEÑAR PROGRAMAS PARALELOS