

Resumen Unidad 4 - Introducción a la Computación Paralela

- Técnica que permite ejecutar varias instrucciones al mismo tiempo usando dos o más unidades de procesamiento.
- Se basa en dividir grandes problemas en partes más pequeñas que se resuelven simultáneamente.

Analogía: Recoger canicas entre varias personas en lugar de hacerlo solo.

Ventajas

- Resolución de problemas complejos.
- Resultados más rápidos.
- Mejor relación rendimiento/costo.
- Alta escalabilidad.

Desventajas

- Mayor consumo de energía.
 - Programación más compleja.
 - Dificultades de sincronización y comunicación.
 - Mayor posibilidad de fallos.
-

Taxonomía de Flynn

Clasificación de arquitecturas paralelas según instrucciones y datos:

- **SISD**: Una instrucción, un dato (computadoras tradicionales).
 - **MISD**: Varias instrucciones, un dato (raro, usado en control).
 - **SIMD**: Una instrucción, múltiples datos (útil en gráficos).
 - **MIMD**: Varias instrucciones, múltiples datos (usado actualmente en servidores).
-

Tipos de Paralelismo

1. **Nivel de bit**: Procesamiento simultáneo de varios bits.
 2. **Nivel de instrucción (ILP)**: Múltiples instrucciones a la vez en un procesador.
 3. **De datos**: La misma operación sobre múltiples datos (ej. imágenes).
 4. **De tareas**: Cada procesador ejecuta tareas distintas.
-

Arquitecturas de Memoria Compartida

- **UMA (Uniform Memory Access)**:
 - Memoria central compartida.
 - Acceso uniforme para todos los procesadores.
 - Fácil de programar.
 - **NUMA (Non-Uniform Memory Access)**:
 - Cada procesador tiene memoria local.
 - Acceso rápido a su memoria, más lento a la de otros.
 - Mejora rendimiento en sistemas grandes.
-



Otros Tipos de Computación Paralela

- **SPMD**: Un mismo programa en múltiples datos (ej. CUDA).
 - **MPP**: Muchos procesadores en paralelo para problemas grandes (supercomputadoras).
-



Computadoras Secuenciales

- Un solo procesador ejecuta instrucciones una por una.
 - Basado en el modelo **Von Neumann**:
 - CPU, RAM, unidad de control, ALU.
 - Ciclo F-D-E: Fetch, Decode, Execute.
-



Organización de Memoria

- **Memoria Compartida**: Un solo espacio de direcciones (Java, ForkJoin, etc.).
 - **Memoria Distribuida**: Cada nodo tiene su memoria; se comunican por mensajes (sockets, MPI).
-



Redes de Interconexión en Paralelismo



Redes Dinámicas (Indirectas)

- Conmutadores intermedios.
- Permiten múltiples rutas.
- Adaptables y escalables.



Medio Compartido

- Todos escuchan las transmisiones (ej. Wi-Fi, Ethernet).
- Simples pero con problemas de rendimiento y seguridad.

Redes Conmutadas

- Usan interruptores para direccionar mensajes.
 - Tipos:
 - En **anillo**, **mall**, **hipercubo**.
 - Métodos:
 - **Circuitos**, **paquetes**, **mensajes**.
-

Sistemas de Memoria Distribuida

- Cada nodo tiene:
 - Su CPU.
 - Memoria privada.
 - Comunicación solo por mensajes.
 - Alta escalabilidad.
 - Alta latencia en comunicación.
-

Casos de Estudio y Aplicaciones

- **Medicina**: Análisis de imágenes, modelado de enfermedades.
 - **IA / Machine Learning**: Entrenamiento y predicción en tiempo real.
 - **Industria**: Simulación y optimización de diseños.
 - **Simulaciones científicas**: Clima, dinámica de fluidos.
-

Beneficios Generales

- Mejor aprovechamiento de recursos.
- Procesamiento más rápido.
- Ahorro de tiempo y costos.
- Resolución de problemas imposibles de tratar secuencialmente.