

Introducción a la Computación Evolutiva

Dr. Carlos A. Coello Coello

Departamento de Computación

CINVESTAV-IPN

Av. IPN No. 2508

Col. San Pedro Zacatenco

México, D.F. 07300

email: ccoello@cs.cinvestav.mx

[http: //delta.cs.cinvestav.mx/~ccoello](http://delta.cs.cinvestav.mx/~ccoello)

Nociones de Paralelismo

Podemos definir el **procesamiento en paralelo** como la ejecución concurrente (o simultánea) de instrucciones en una computadora.

Dicho procesamiento puede ser en la forma de eventos que ocurran:

- (1) Durante el mismo intervalo de tiempo
- (2) En el mismo instante
- (3) En intervalos de tiempo traslapados

Nociones de Paralelismo

La motivación más obvia del paralelismo es el incrementar la eficiencia de procesamiento.

Existen muchas aplicaciones que demandan grandes cantidades de tiempo de procesamiento y que, por ende, resultan beneficiadas de contar con arquitecturas en paralelo.

Nociones de Paralelismo

Una de las frecuentes confusiones respecto al paralelismo es que se cree que al contar con una computadora que tenga n procesadores trabajando en el mismo problema, éste podrá resolverse n veces más rápido. Esto es falso.

Nociones de Paralelismo

Al usar varios procesadores para una misma tarea, debemos tomar en cuenta que existirán:

- Problemas de comunicación entre ellos.
- Conflictos al intentar acceder la memoria.
- Algoritmos ineficientes para implementar el paralelismo del problema.

Por tanto, si tenemos n procesadores, el incremento de velocidad normalmente no será de n veces.

Nociones de Paralelismo

Existe un límite inferior respecto del incremento de velocidad real al tener n procesadores. A este límite inferior se le conoce como la **Conjetura de Minsky**, y es de $\log_2 n$.

Nociones de Paralelismo

Aunque el límite superior depende realmente de si se considera a todo el programa (incluyendo la parte de entrada y salida, la cual suele ser secuencial), suele aceptarse que éste está definido por $\frac{n}{\ln n}$ (Hwang & Briggs, 1984).

De estos 2 límites puede inferirse fácilmente que no resulta útil agregar más y más procesadores, si lo que queremos es hacer más rápida a una computadora.

Nociones de Paralelismo

La eficiencia de un sistema de cómputo se mide en términos de sus capacidades tanto en hardware como en software. A dicha medida de eficiencia se le conoce como **rendimiento total** (*throughput*) y se define como la cantidad de procesamiento que puede realizarse en un cierto intervalo de tiempo.

Nociones de Paralelismo

Una técnica que ha conducido a incrementos notables del rendimiento total de un sistema de cómputo es el **proceso de encauzamiento** (*pipelining*).

Nociones de Paralelismo

Este proceso de encauzamiento es análogo a una línea de ensamblaje en una planta industrial. Una función a ejecutarse por una computadora es dividida en sub-funciones más pequeñas, y se diseña hardware separado (llamado **etapa**) para cada una de estas subfunciones. Estas etapas están conectadas entre sí, de manera que forman un solo cauce (o *pipeline*) que realiza la función original.

Nociones de Paralelismo

Michael J. Flynn (1966) introdujo un esquema para clasificar la arquitectura de una computadora basado en la forma en la que la máquina relaciona sus instrucciones con los datos que procesa. Flynn definió el término **stream** (*flujo*) como una secuencia de elementos, ya sea datos o instrucciones, ejecutados u operados por un procesador.

Nociones de Paralelismo

La clasificación de Flynn es la siguiente:

- **SISD**: *Single Instruction Stream, Single Data Stream*
- **SIMD**: *Single Instruction Stream, Multiple Data Stream*
- **MISD**: *Multiple Instruction Stream, Single Data Stream*
- **MIMD**: *Multiple Instruction Stream, Multiple Data Stream*

Nociones de Paralelismo

Una computadora **SISD** es la computadora serial convencional que todos conocemos, en la cual las instrucciones se ejecutan una por una, y se usa una sola instrucción para lidiar con, cuando mucho, una operación sobre los datos. Aunque es posible introducir cierto nivel de paralelismo en estas computadoras (usando *pipelining*), la naturaleza secuencial de la ejecución de sus instrucciones la coloca en esta categoría.

Nociones de Paralelismo

En una computadora **SIMD**, una sola instrucción puede iniciar un gran número de operaciones. Estas instrucciones (llamadas **vectoriales**) se ejecutan de manera secuencial (una a la vez), pero son capaces de trabajar sobre varios flujos de datos a la vez. También en este caso es posible usar *pipelining* para acelerar la velocidad de procesamiento.

Nociones de Paralelismo

La clase **MISD** implica la ejecución de varias instrucciones operando simultáneamente sobre un solo dato. Este modelo es únicamente teórico, porque no existen computadoras que caigan dentro de esta categoría.

Nociones de Paralelismo

Una computadora **MIMD** se caracteriza por la ejecución simultánea de más de una instrucción, donde cada instrucción opera sobre varios flujos de datos. Ejemplos de esta arquitectura son los sistemas multiprocesadores.

Para lo relacionado con AGs paralelos, sólo hablaremos de las computadoras SIMD y MIMD.