**Programación Paralela**

Consiste en la explotación de varios procesadores para que trabajen de forma conjunta en la resolución de un problema computacional. Normalmente cada procesador trabaja en una parte del problema y se realiza intercambio de datos entre los procesadores. Según cómo se realice este intercambio podemos tener modelos distintos de programación paralela. Los dos casos básicos son:

* Disponer de una memoria a la que pueden acceder directamente todos los elementos de proceso (procesadores), que se usará para realizar la transferencia de datos.
* En este caso tenemos el modelo de Memoria Compartida (Shared Memory Model), que corresponde a sistemas que tienen una memoria compartida para todos los procesadores, pero también a sistemas de Memoria Virtual Compartida, donde la memoria está distribuida en el sistema (posiblemente porque módulos distintos de memoria están asociados a procesadores distintos) pero se puede ver como una memoria compartida porque cada procesador puede acceder directamente a todos los módulos de memoria, tanto si los tiene directamente asociados como si están asociados a otros procesadores.Existen herramientas específicas de programación en memoria compartida. Las más extendidas son pthreads y OpenMP, que se puede considerar en la actualidad el estándar de este tipo de programación.
* Que cada procesador tenga asociado un bloque de memoria al que puede acceder directamente, y ningún procesador puede acceder directamente a bloques de memoria asociados a otros procesadores. Así, para llevar a cabo el intercambio de datos será necesario que cada procesador realice explícitamente la petición de datos al procesador que dispone de ellos, y este procesador haga un envío de los datos. Este es el modelo de Paso de Mensajes. Hay varios entornos de programación de Paso de Mensajes (PVM , BSP...) y el estándar actual es MPI .

**Necesidad de Programación Paralela**

La necesidad de la computación paralela se origina por las limitaciones de los computadores secuenciales: integrando varios procesadores para llevar a cabo la computación es posible resolver problemas que requieren de más memoria o de mayor velocidad de cómputo. También hay razones económicas, pues el precio de los computadores secuenciales no es proporcional a su capacidad computacional, sino que para adquirir una máquina el doble de potente suele ser habitual tener que invertir bastante más del doble; mientras que la conexión de varios procesadores utilizando una red nos permite obtener un aumento de prestaciones prácticamente proporcional al número de procesadores con un coste adicional mínimo.

La programación paralela es una solución para solventar esos problemas, pero presenta otras dificultades. Algunas dificultades son físicas, como la dificultad de integración de componentes y la disipación de calor que conlleva, o la mayor complejidad en el acceso a los datos, que se puede llegar a convertir en un cuello de botella que dificulta la obtención de buenas prestaciones, aunque se aumente la capacidad computacional teórica del sistema. Hay también dificultades lógicas, como son la mayor dificultad de desarrollar compiladores y entornos de programación eficientes para los sistemas paralelos, que son más complejos que los secuenciales, o la necesidad de utilizar programación paralela en vez de la programación secuencial, que es bastante más sencilla.

**Aspectos en la programación paralela**

* Diseño de computadores paralelos teniendo en cuenta la escalabilidad y comunicaciones.
* Diseño de algoritmos eficientes, no hay ganancia si los algoritmos no se diseñan adecuadamente.
* Métodos para evaluar los algoritmos paralelos: ¿Cuán rápido se puede resolver un problema usando una máquina paralela?,¿Con que eficiencia se usan esos procesadores?
* En lenguajes para computadores paralelos deben ser flexibles para permitir una implementación eficiente y fácil de programar.
* Los programas paralelos deben ser portables y los compiladores paralelizantes.

**Modelos de computadoras**

* Modelo SISD
* Simple instrucción, simple Data.
* Consiste en un procesador que recibe un cierto flujo de instrucciones y de datos.
* Un algoritmo para esta clase dice ser secuencial o serial.
* Modelo MISD
* Múltiple Instrucción, simpledata.
* Existencia de N procesadores con un simple flujo de datos ejecutando instrucciones diferentes en cada procesador.
* Modelo SIMD
* 6. Simple Instrucción, múltiple Data.
* 7. Existen JN procesadores ejecutando la misma instrucción sobre los mismos o diferentes datos.
* 8. Existencia de datos compartidos.
* Modelo MIMD
* 9. Multiple Instruction, multiple data.
* 10. Existen N procesadores ejecutando instrucciones diferentes sobre los mismos o diferentes datos.
* 11. Existencia de datos compartidos.

**Métodos para conectar los procesadores en paralelo**

* Malla.
* Árbol Binario.
* Hiperarbol.
* Piramidal.
* Mariposa.
* HiperCubo.
* Hiper Cubo Cíclico.
* Transpaso-Intercambio.

**Referencias bibliográficas.**

Rohit Chandra, Ramesh Menon, Leo Dagum, David Kohr, Dror Maydan, and Jeff McDonald. Parallel Programming in OpenMP. Morgan Kauffman, 2001

Francisco Almeida, Domingo Giménez, José Miguel Mantas, and Antonio M. Vidal. Introducción a la programación paralela. Paraninfo Cengage Learning, 2008