

Modelos de programación paralela

César Pedraza Bonilla

Universidad Nacional

capedrazab@unal.edu.co

9 de agosto de 2017

Overview

- 1 Modelos de arquitecturas paralelas
- 2 Paralelización de programas.
- 3 Niveles de paralelismo.
- 4 Distribución de datos.
- 5 Distribución de datos.
- 6 Intercambio de datos.
- 7 Procesos e hilos.

- 1 Paralelismo a nivel de bits.
- 2 Paralelismo por segmentación.
- 3 Paralelismo por múltiples unidades funcionales.
- 4 Paralelismo a nivel de proceso o hilos.

- ① Single Instruction - Single Data. (SISD)
- ② Multiple Instruction - Single Data. (SIMD)
- ③ Single Instruction - Multiple Data. (SIMD)
- ④ Multiple Instruction - Multiple Data. (MIMD)

Diferentes modelos de abstracción:

- 1 Modelo de máquina. Describe HW, SOs, etc.
- 2 Modelo de arquitectura. Describe redes de interconexión, organización de la memoria, etc.
- 3 Modelo computacional. Define el modelo formal y analítico de una arquitectura para predecir el funcionamiento de un programa en una máquina.
- 4 Modelo de programación. Semántica de elaboración de programas paralelos.

Paralelización de programas.

Es necesario tener en cuenta la dependencia de datos.

- ❶ Descomposición de cálculos. Se pretende generar las suficientes tareas para ocupar los cores. Para que el mapeo y la planificación sean efectivos.
- ❷ Asignación de tareas a procesos o hilos. El número de procesos o hilos no necesariamente coincide con el número de cores. Balanceo de carga.
- ❸ Mapeo a procesadores físicos. Idealmente asignar un proceso a un core. Si es diferente esta tarea la realiza el planificador.

Niveles de paralelismo.

- 1 A nivel de instrucciones. Procesadores superescalares, pentium en adelante.
- 2 A nivel de datos.
- 3 A nivel de anidamiento.
- 4 A nivel funcional.

El paralelismo puede ser implícito o explícito. En el primero no se ocupa el programador → compiladores.

Patrones de programación paralela:

- 1 Creación de procesos o hilos.
- 2 Fork - Join.
- 3 ParBegin - ParEnd.
- 4 SPMD, SIMD.
- 5 Master/Slave

Distribución de datos.

Muchos programas manipulan datos en forma de vectores o matrices. Es necesario hacer una correcta distribución de dichos datos.

Distribución de datos:

- 1 dimensión. Blockwise, cyclic data, block cyclic data.

a) blockwise

1	2	3	4	5	6	7	8
P ₁	P ₂	P ₃	P ₄				

cyclic

1	2	3	4	5	6	7	8
P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	P ₃	P ₄

block-cyclic

1	2	3	4	5	6	7	8	9	10	11	12
P ₁	P ₂	P ₃	P ₄	P ₁	P ₂			P ₁	P ₂		

Figura: 1 dimensión. *TomadodeRauberParallelProgramming*

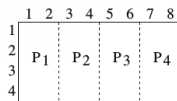
Distribución de datos.

Muchos programas manipulan datos en forma de vectores o matrices. Es necesario hacer una correcta distribución de dichos datos.

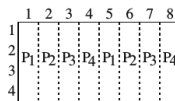
Distribución de datos:

- 2 dimensiones. Blockwise and cyclic tomando filas o columnas.

b) blockwise



cyclic



block-cyclic

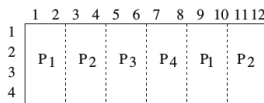


Figura: 2 dimensiones. *TomadodeRauberParallelProgramming*

- Parametrizada. Se distribuye teniendo en cuenta variables.

A tener en cuenta para controlar las partes de un programa. Depende de memoria compartida o distribuida. *Ej*

Operaciones de comunicaciones:

- Modelos de paso de mensajes.
- Tipos de operaciones:
 - Single transfer.
 - Single broadcast.
 - Single accumulation.
 - Gather.
 - Scatter.
 - Multibroadcast.
 - Multi-accumulation.
 - Total exchange.

Proceso completo. Programa en ejecución con todos sus segmentos.

Proceso ligero o hilo. Segmento de código de un programa que se ha replicado.