

IMT2112 - Algoritmos Paralelos en Computación Científica

Granularidad

Elwin van 't Wout

29 de agosto de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl

Clase previa

- Speedup y eficiencia paralela
- Leyes de Amdahl y Gustafson

Escabilidad

- Escalabilidad fuerte (*strong scaling*)
 - aumentar el número de procesadores para un código dado
 - la referencia es un código secuencial o un código paralelo en una pequeña cantidad de procesadores
- Escalabilidad débil (*weak scaling*)
 - aumentar la cantidad de datos o instrucciones junto al número de procesadores
 - esto permite una mayor eficiencia paralela

Agenda

- ¿Como acceder los datos de manera eficiente en una arquitectura paralela?
- ¿Como clasificar el tipo de paralelismo?

La computación paralela

Sección 2.3 del libro de Eijkhout

La taxonomía de Flynn

- SISD: single instruction single data
 - los procesadores tradicionales
- SIMD: single instruction multiple data
 - *vector machines* y tarjetas gráficas
- MISD: multiple instructions single data
 - no en el mercado de consumo
- MIMD: multiple instructions multiple data
 - clústers y computadores paralelos

Arquitectura de memoria paralela

Sección 2.4 del libro de Eijkhout

Motivación

- Ya hemos visto que para las computadoras secuenciales, la transferencia de datos suele ser el factor limitante para el rendimiento
- Esto es tanto más cierto para las computadoras paralelas que tienen una gama de dispositivos de almacenamiento

Tipos de memoria

- Tenga en cuenta que tenemos ambos
 - memoria físicamente separada y
 - memoria funcionalmente separada con diferentes espacios de direcciones (*dataspace*)
- La principal distinción en los tipos de memoria es
 - memoria compartida
 - memoria distribuida

Acceso uniforme a la memoria

- Acceso uniforme a la memoria (UMA)
 - cualquier ubicación de memoria es accesible para todos los procesadores
- Multiprocesamiento simétrico (SMP)
 - El modelo de programación para arquitecturas UMA
- Computadoras multi-core y multiprocesador
 - los buses de memoria necesitan un gran ancho de banda
 - factible solo para un pequeño número de procesadores

Acceso no uniforme a la memoria

- Acceso a memoria no uniforme (NUMA)
 - memoria distribuida físicamente
 - el mismo espacio de direcciones
- Cualquier ubicación de memoria es accesible para cada procesador pero con un tiempo de acceso diferente
- Arquitectura multiprocesador
 - memoria caché distribuida
 - coherencia de caché
 - gastos generales de comunicación
- NUMA es también factible para un red de procesadores
 - memoria compartida distribuida / virtual

Memoria distribuida

- Memoria distribuida
 - memoria separada física y lógicamente
- Los procesadores solo pueden acceder a los datos en otra ubicación pasando información explícitamente
 - interfaz de paso de mensajes (MPI - Message Passing Interface)
- La mejor arquitectura para escalar redes
- La arquitectura más difícil para el programador

Granularidad

Sección 2.5 del libro de Eijkhout

Tipos de paralelismo

- Paralelismo de datos o paralelismo de grano fino
 - la misma instrucción para muchos elementos de datos
- Paralelismo funcional / instrucción
 - diferentes instrucciones sobre elementos de datos independientes
- Paralelismo de tareas
 - subprogramas independientes

Tipos de paralelismo

- Convenientemente paralelo
 - también llamado “*embarrassing parallel*”
 - una tarea simple para muchos elementos de entrada independientes
- Paralelismo de grano medio
 - tareas simples para porciones independientes de *input*
- Estos términos pueden referirse al algoritmo o la realización real del software

Granularidad de tareas

- La ‘granularidad’ de un algoritmo paralelo es la cantidad de trabajo que puede realizar una unidad de procesamiento antes de tener que comunicarse con otros procesos
 - para el paralelismo a nivel de instrucción esto es del orden de una sola instrucción
 - para el paralelismo de tareas esto está en el orden de un subprograma

Balanceo de carga

Sección 2.10.1 del libro de Eijkhout

Balanceo de carga

- Desequilibrio de carga (*load unbalance*): un procesador se ha estancado debido a la falta de instrucciones (*stall*)
- No hay razón intrínseca (por ejemplo, latencia) para que el procesador esté inactivo

Balanceo de carga

- Paralelismo de tareas
 - ejecución paralela de subprogramas
 - a menudo la cantidad de subprogramas puede ser mayor que la cantidad de procesadores
 - cada subprograma puede tener un tiempo de ejecución diferente
- Balanceo de carga
 - optimización del orden de las tareas
 - optimización de los tamaños de las tareas

Balanceo de carga

- Note la distinción entre
 - distribución de datos y
 - distribución del trabajo
- Equilibre el uso de memoria y el tiempo de ejecución
 - a menudo esto va de la mano con el equilibrio de carga

Balanceo de carga

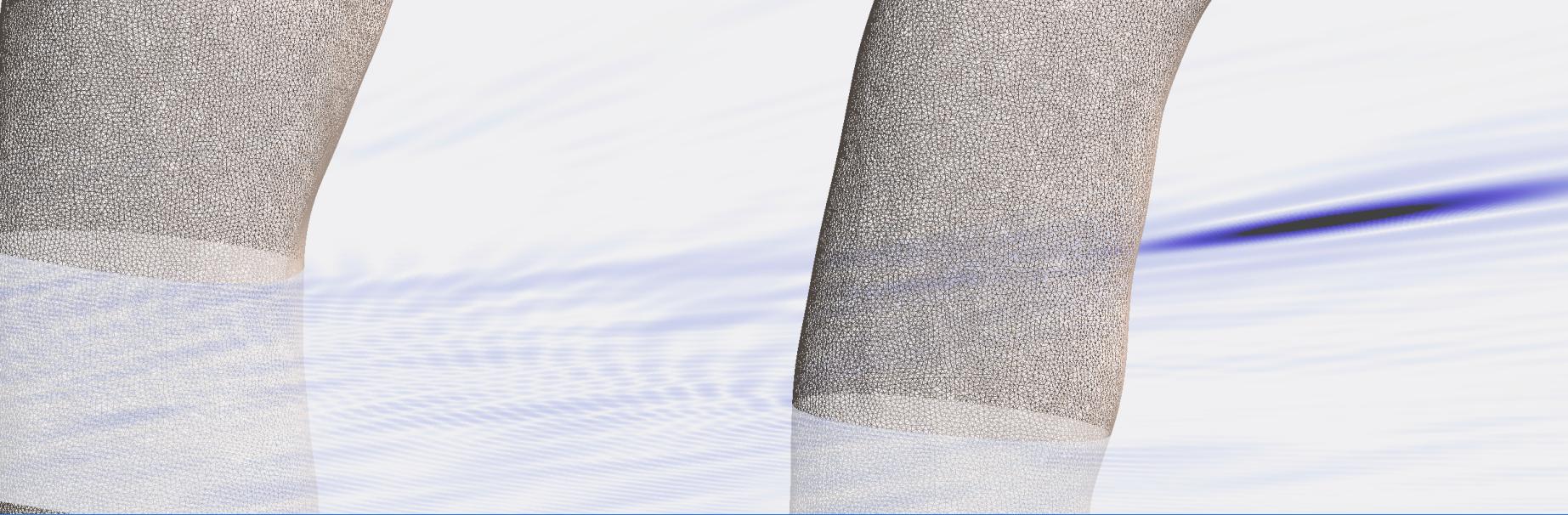
- Equilibrio de carga estática (*static load balancing*)
 - particione la carga de trabajo en la misma cantidad de subprogramas que los procesadores
- Balanceo dinámico de carga (*dynamic load balancing*)
 - sobrecomposición de la obra
 - más subprogramas que procesadores
 - requiere programación de tareas, generalmente realizada en el nodo principal

Resumen

- La taxonomía de Flynn
- Acceso paralelo de memoria
- Granularidad
- Balanceo de carga

Clase siguiente

- Programación paralela
- *Threading*



IMT2112 - Algoritmos Paralelos en Computación Científica

Granularidad

Elwin van 't Wout

29 de agosto de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl