

IMT2112 - Algoritmos Paralelos en Computación Científica

Operaciones colectivas

Elwin van 't Wout

5 de septiembre de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl

Clase previa

- Multithreading
- Programar en OpenMP y C++

Agenda

- ¿Como analizar el rendimiento de operaciones colectivas estándares?

Álgebra lineal de alto rendimiento

Capítulo 6 del libro de Eijkhout

Cálculo científico

- Los capítulos 4 y 5 tratan de métodos numéricos para ecuaciones diferenciales y álgebra lineal
- Se espera que el mayor parte de este contenido sea conocido
 - leer los capítulos
 - leer los apuntes del curso Cálculo Científico

Álgebra lineal de alto rendimiento

- Paralelización de métodos comunes en álgebra lineal numérica
- Importante para la paralelización de memoria compartida y memoria distribuida
- Diseño del algoritmo y análisis de desempeño

Operaciones colectivas

Sección 6.1 del libro de Eijkhout

Transferencia de datos

- Latencia: el retraso inicial entre la solicitud y la llegada de los datos
 - tiempo (segundos o ciclos)
- Ancho de banda: la taza de transferencia de datos
 - cantidad por tiempo (p.ej. bit/s)
- El tiempo total para transferir datos es:
$$\text{tiempo} = \text{latencia} + \frac{\text{cantidad}}{\text{ancho de banda}}$$

Parámetros de rendimiento

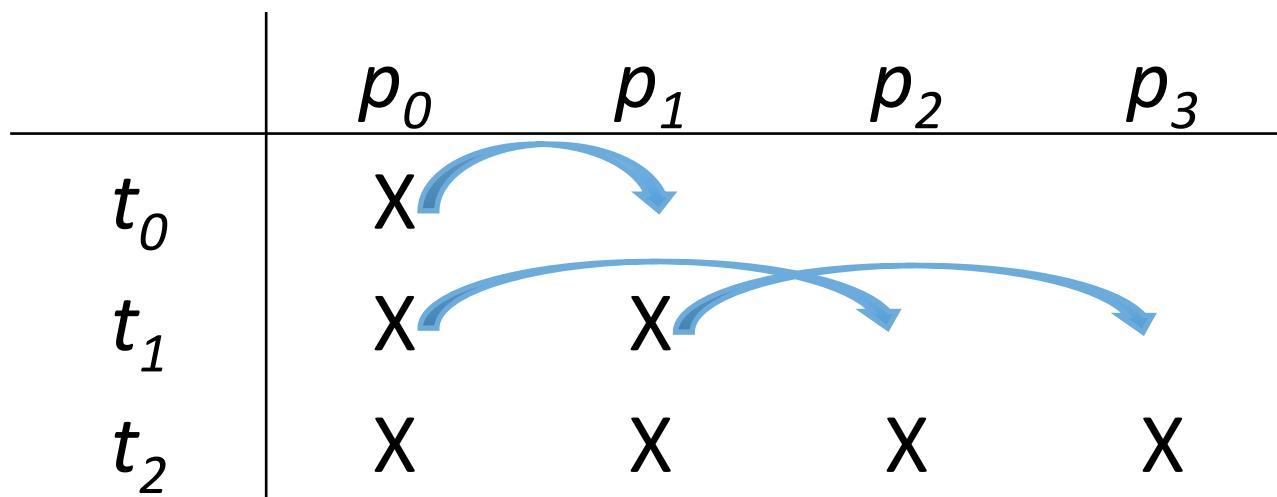
- El análisis de rendimiento depende de:
 - p : número de procesadores
 - n : cantidad de datos (bits o números de punto flotante)
 - α : latencia (segundos o ciclos)
 - β : ancho de banda inverso (p.ej. s/bit)
 - γ : rendimiento de cálculo inverso (p.ej. s/float)

Árbol de expansión

- Suposiciones para el modelo del análisis de rendimiento
 - cada procesador puede enviar y recibir solo 1 mensaje al mismo tiempo
 - todos los procesadores están conectados entre sí
- Se llama un ‘árbol de expansión mínimo’ (*minimum spanning tree*)

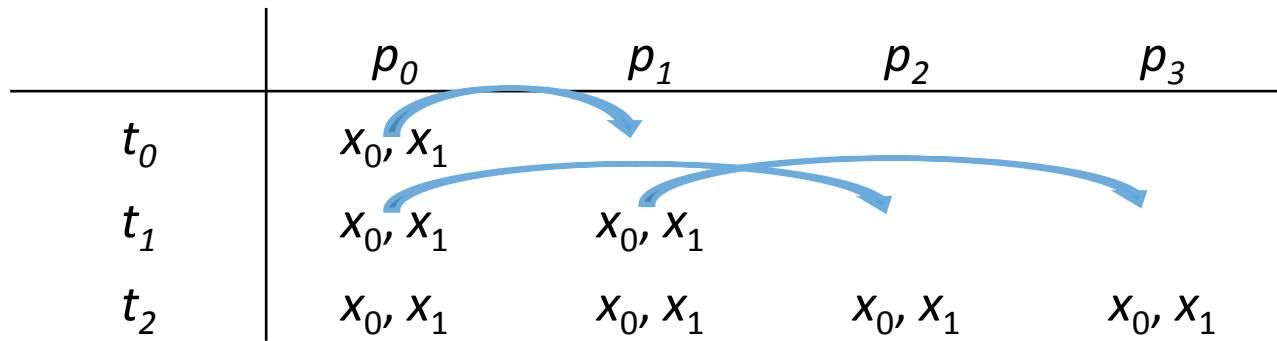
Árbol de expansión

- Enviar mensajes a un grupo de p procesadores
- La cantidad de pasos de tiempo para “activar” el clúster es de $\lceil \log_2(p) \rceil$



Broadcast

- Estado inicial: un procesador tiene n elementos de datos
- Tarea: copiar los datos a todos los demás procesadores
- Límite inferior sobre el costo total: $T_{\text{broadcast}} \geq \log_2(p)\alpha + n\beta$
- Para este algoritmo de ejemplo: $\log_2(p)(\alpha + n\beta)$



Gather

- Estado inicial: cada procesador tiene n/p elementos de datos
- Tarea: recopilar todos los n elementos de datos en un procesador
- Límite inferior sobre el costo total
 - latencia de un árbol de expansión mínimo
 - transferencia de al menos $(p-1)n/p$ elementos de datos
$$T_{gather} \geq \log_2(p)\alpha + (p - 1)\frac{n}{p}\beta$$
 - en la práctica se necesita un mayor ancho de banda

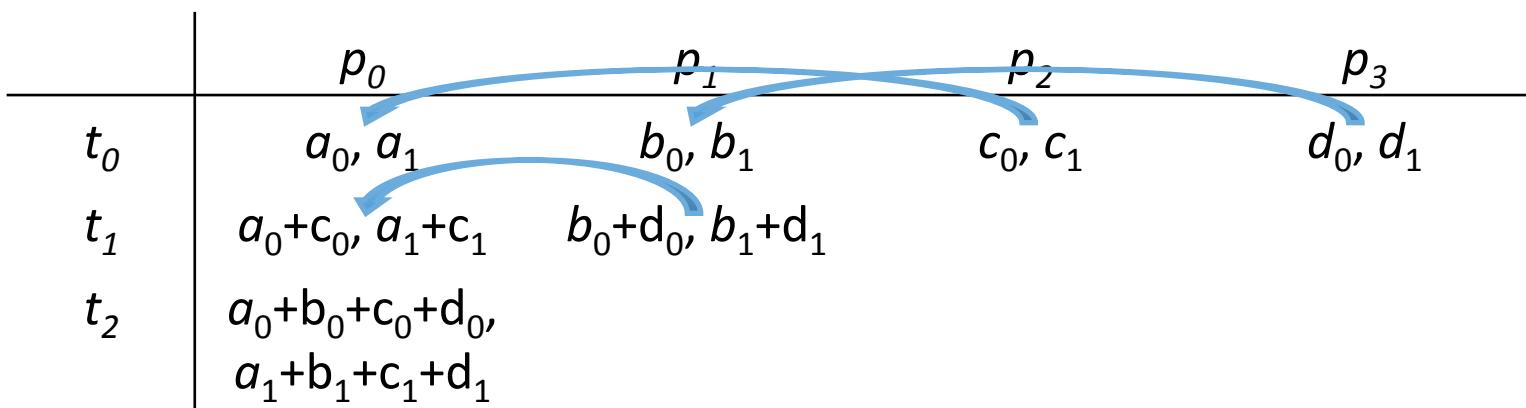
Allgather

- Estado inicial: cada procesador tiene n/p elementos de datos
- Tarea: recopilar todos los n elementos de datos en todos los procesadores
- Límite inferior sobre el costo total
 - cada procesador realiza una operaciones de *gather*, idealmente todas al mismo tiempo
 - cada procesador recibe n/p elementos de datos de $p-1$ procesadores

$$T_{\text{allgather}} \geq \log_2(p)\alpha + (p - 1)\frac{n}{p}\beta$$

Reduction

- Estado inicial: cada procesador tiene n elementos de datos
- Tarea: un procesador necesita una combinación de los elementos de datos en todos los procesadores
- Un ejemplo es calcular n sumas de p números



Reduction

- Estado inicial: cada procesador tiene n elementos de datos
- Tarea: un procesador necesita una combinación de los elementos de datos en todos los procesadores
- Límite inferior sobre el costo total
 - comunicación similar a una *gather*
 - un procesador tiene que recibir a menos n datos
 - el número de operaciones requeridas es $(p-1)n$, que se distribuyen idealmente sobre p procesadores

$$T_{\text{reduction}} \geq \log_2(p)\alpha + n\beta + \frac{(p-1)n}{p}\gamma$$

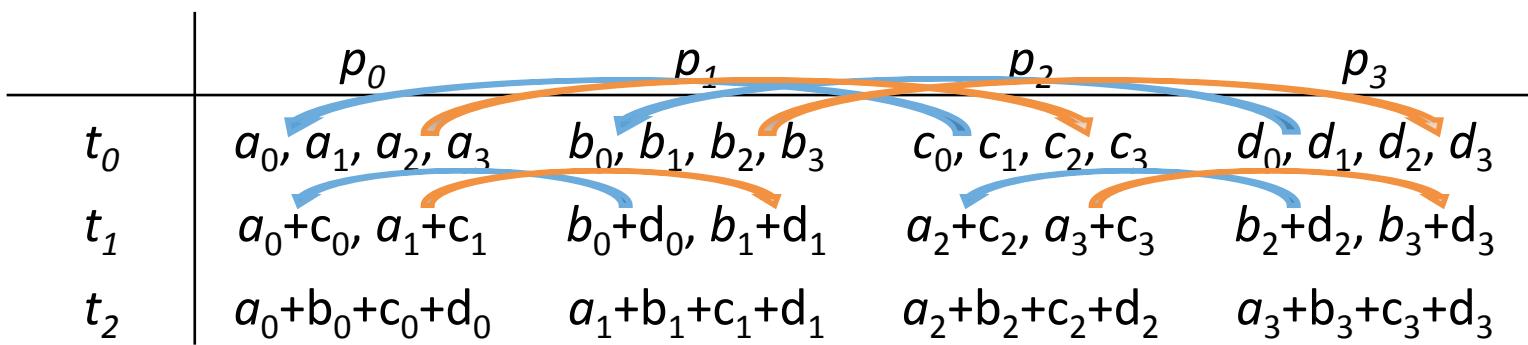
Allreduce

- Estado inicial: cada procesador tiene n elementos de datos
- Tarea: todos los procesadores necesitan una combinación de los elementos de datos en todos los procesadores
- Límite inferior sobre el costo total
 - para una paralelización perfecta, la latencia y el tiempo de cálculo son los mismos que para la reducción
 - todos los procesadores tienen que enviar y recibir $(p-1)n$ elementos de datos, idealmente distribuidos entre los p procesadores

$$T_{\text{allreduce}} \geq \log_2(p)\alpha + \frac{2(p-1)n}{p}\beta + \frac{(p-1)n}{p}\gamma$$

Reduce-scatter

- Estado inicial: cada procesador tiene n elementos de datos
- Tarea: ejecutar n reducciones en n procesadores
- Ejemplo: suma de columnas en una matriz almacenada por filas



Reduce-scatter

- Estado inicial: cada procesador tiene n elementos de datos
- Tarea: ejecutar n reducciones en n procesadores
- Límite inferior sobre el costo total
 - comunicación como una operación *allgather*
 - cálculo como una operación *allreduce*

$$T_{\text{reduce-scatter}} \geq \log_2(p)\alpha + \frac{(p-1)n}{p}\beta + \frac{(p-1)n}{p}\gamma$$

Reduce-scatter

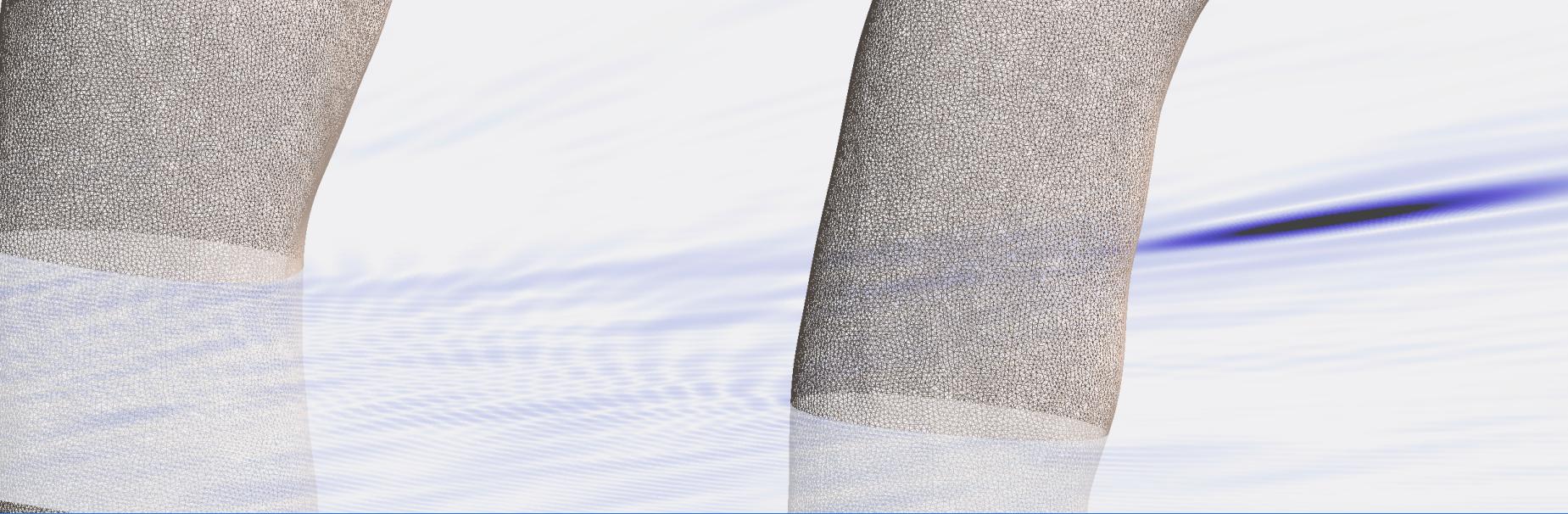
- *Broadcast*
 - un procesador envía elementos de datos a todos los procesadores
- *Gather*
 - un procesador recopila elementos de datos de todos los procesadores
- *Reduction*
 - un procesador realiza una combinación de datos por elementos de todos los procesadores
- *Allgather y allreduce*
 - todos los procesadores realizan una operación de recolección o reducción
- *Reduce-scatter*
 - todos los procesadores realizan una operación de reducción diferente

Resumen

- Operaciones colectivas
- Árbol de expansión
- Análisis de rendimiento

Clase siguiente

- La multiplicación matriz-vector en paralela



IMT2112 - Algoritmos Paralelos en Computación Científica

Operaciones colectivas

Elwin van 't Wout

5 de septiembre de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl