

IMT2112 - Algoritmos Paralelos en Computación Científica

# La multiplicación matriz-vector en paralelo

Elwin van 't Wout

10 de septiembre de 2019



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl

# Clase previa

- Las operaciones colectivas

# Agenda

- ¿Como implementar y analizar la multiplicación matriz por vector en paralelo?

# La multiplicación matriz-vector en paralelo

Sección 6.2 del libro de Eijkhout

# Transferencia de datos

- Calcular  $y = Ax$  para una matriz de tamaño  $N \times N$
- Partitionar la matriz para paralelización
- El tamaño de los trozos se puede elegir de acuerdo con:
  - cantidad de procesadores
  - tamaño de la memoria
  - equilibrio de carga

# Paralelización por bloques de filas

- Partitionar la matriz  $A$  en bloques de filas
  - asigne diferentes trozos de filas completas a los procesadores
  - partitionar el vector  $y$  en consecuencia
- Ventajas
  - fácil de implementar
  - transferencia de datos y localidad
    - especialmente para almacenamiento por filas
- Desventajas
  - particionamiento no deseado del vector  $x$
  - la eficiencia paralela no escala bien

# Paralelización por bloques de filas

- Copias del vector  $x$  en todos los procesadores
  - uso innecesario de la memoria
  - peligroso para la integridad de los datos
    - especialmente para potencias matriciales ( $A^n x$ )
- Particionamiento del vector  $x$  sobre los procesadores
  - una operación de *allgather*
  - es necesario usar *buffers* para la reutilización de datos

# Paralelización por bloques de filas

- Transferencia de datos: *allgather* del vector  $x$
- Computación: *matvec* sobre  $p$  procesadores
- Escalabilidad débil
  - escala el número de procesadores con el tamaño de la matriz
  - la eficiencia paralela escala con la raíz cuadrada
$$E_p = \mathcal{O} \left( \frac{1}{\sqrt{p}} \right)$$
  - esto no es muy bueno, pero tampoco malo

# Paralelización por bloques de columnas

- Partitionar la matriz en bloques de columnas
- Partitionar los vectores también
- Transferencia de datos: *reduce-scatter*
- Computación: *matvec* sobre  $p$  procesadores
- Escalabilidad débil igual que la versión de bloques de filas

$$E_p = \mathcal{O} \left( \frac{1}{\sqrt{p}} \right)$$

# Paralelización por bloques

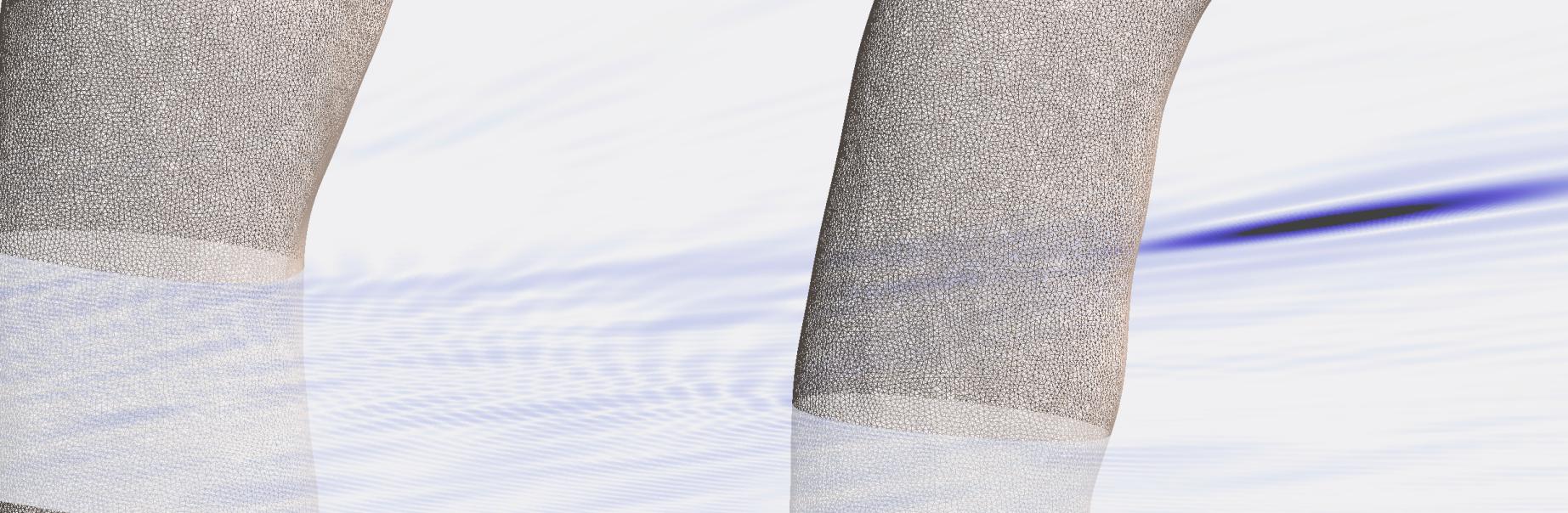
- Partitionar la matriz en  $r$  bloques de filas y  $c$  bloques de columnas
- Vectores de partición en  $p = rc$  trozos
- El algoritmo
  - *allgather* sobre las filas
  - multiplicaciones locales de matriz por vector
  - *reduce-scatter* sobre las columnas
- Escalabilidad débil:  $E_p = \mathcal{O}\left(\frac{1}{\log_2(p)}\right)$

# Resumen

- La multiplicación matriz por vector en paralelo
- Partitionar la matriz en bloques
- Escalabilidad débil del algoritmo

# Clase siguiente

- La descomposición  $LU$  en paralelo



IMT2112 - Algoritmos Paralelos en Computación Científica

# La multiplicación matriz-vector en paralelo

Elwin van 't Wout

10 de septiembre de 2019



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl