

IMT2112 - Algoritmos Paralelos en Computación Científica

Programar el paso de mensajes

Elwin van 't Wout

24 de septiembre de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl

Clase previa

- La multiplicación matriz por matriz en paralelo
- Analizar el rendimiento de algoritmos paralelos

Agenda

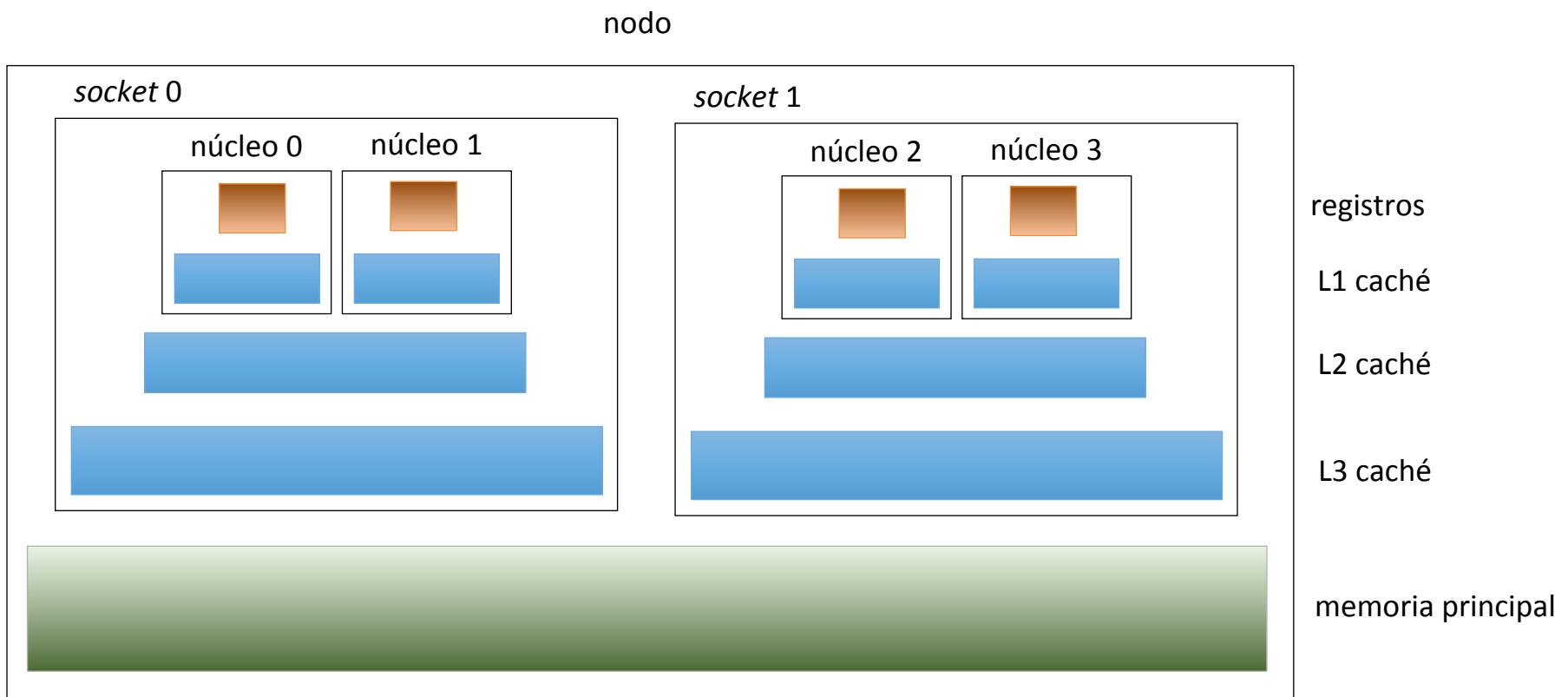
- ¿Como programar en sistemas de memoria distribuida?

La programación distribuida

Sección 2.6.3 del libro de Eijkhout

Parte 1 del libro “Parallel Programming in MPI and OpenMP” de Eijkhout

Los procesadores *multi-core*



La memoria paralela

- Arquitecturas de memoria
 - Compartido: todos los procesadores usan el mismo espacio de memoria
 - Distribuido: diferentes espacios de memoria para diferentes procesadores
- Paradigma de programación
 - Multithreading (OpenMP)
 - Paso de mensajes (MPI)
- La comunicación entre nodos en el clúster es responsabilidad del programador

La biblioteca MPI

- MPI: *Message Passing Interface*
- Comenzó en 1993 como una estandarización de muchas bibliotecas propietarias diferentes para la computación de memoria distribuida
 - Rápidamente se convirtió en el estándar de facto
- MPI es un estándar de biblioteca, no un lenguaje de programación
 - Implementaciones populares: OpenMPI, MPICH
 - Estas son bibliotecas gratuitas de código abierto

La biblioteca MPI

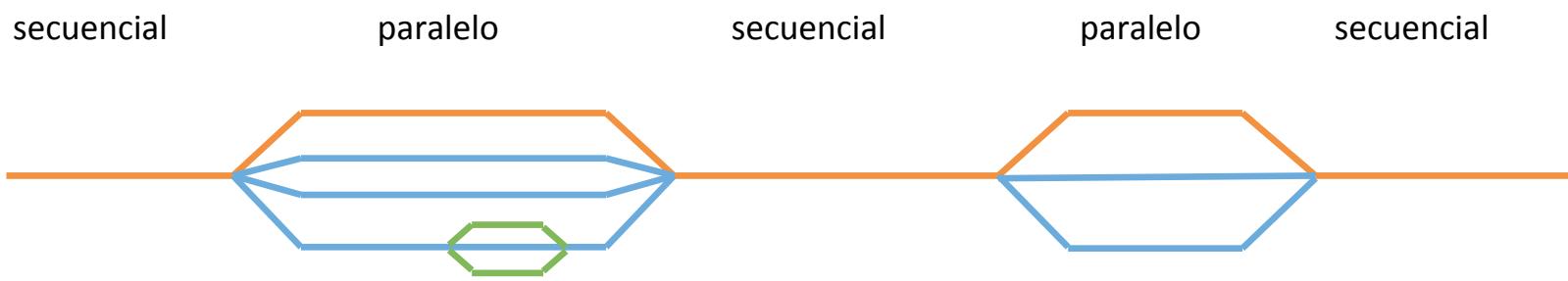
- Interfaces para C, C++ y Fortran (y Python)
 - No necesitas aprender un nuevo idioma
- MPI tiene alrededor de 250 rutinas
 - Por lo general, solo se necesita una docena de ellos
- Las rutinas MPI incluyen
 - gestión de procesos
 - comunicación nodo a nodo
 - operaciones colectivas

La biblioteca MPI

- MPI cumple con el modelo de *single program multiple data* (SPMD)
- Cada proceso ejecuta el mismo ejecutable
 - por lo tanto, también el mismo código
- Cada proceso tiene su propio espacio de datos

Modelo de programación

- OpenMP usa múltiples hilos



- MPI usa múltiples programas/procesos



MPI y OpenMP

- MPI admite paralelización de memoria distribuida
 - Las instrucciones están codificadas explícitamente para cada procesador (núcleo / socket / nodo)
 - MPI también se puede usar para memoria compartida
- MPI es un lenguaje de nivel mas bajo que OpenMP
 - Mayor esfuerzo de implementación
 - Más capacidades

MPI y OpenMP

- Computación híbrida de memoria compartida y distribuida
 - MPI para paralelización de memoria distribuida
 - Multithreading con OpenMP para paralelización de memoria compartida
- No siempre es mejor que solo MPI
 - MPI puede detectar memoria compartida
 - El multithreading dinámico de OpenMP incurre *overhead*

Usar MPI

- Compilar
 - `mpic++ code.cpp`
- Ejecutar
 - `mpirun a.out`
 - `mpirun -np 4 a.out`

Código de MPI

- Header
 - `#include <mpi.h>`
- Inicializar
 - `MPI::Init(NULL, NULL);`
 - `MPI::Is_initialized();`
- Terminar
 - `MPI::Finalize();`
 - `MPI::Is_finalized();`

Información estática de MPI

- Versión de MPI
 - MPI::VERSION
 - MPI::SUBVERSION
 - MPI::Get_version(int *version, int *subversion);
- Nombre de procesador
 - MPI::Get_processor_name(char[n] procname, int *name_len)

Procesos en MPI

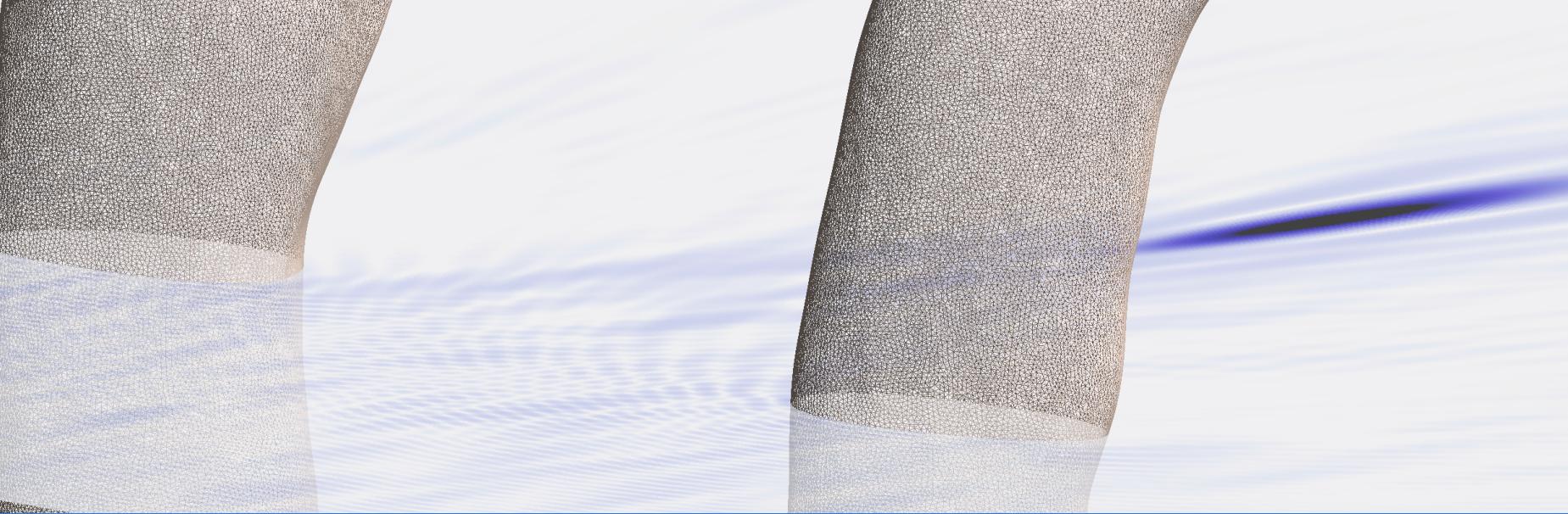
- Cada proceso en MPI tiene un ‘rango’
 - Un número entero que indica la etiqueta del proceso
 - No cambia durante la ejecución
- El proceso con rango 0 a menudo se usa como el proceso maestro
- El comunicador de todos los procesadores se almacena en el objeto `MPI_COMM_WORLD`
- Hay funciones de MPI para el tamaño y rango
 - `MPI::Comm_size (MPI_COMM_WORLD, int * size);`
 - `MPI::Comm_rank (MPI_COMM_WORLD, int * rank);`

Resumen

- Programar en memoria distribuida
- La biblioteca MPI

Clase siguiente

- La programación con MPI



IMT2112 - Algoritmos Paralelos en Computación Científica

Programar el paso de mensajes

Elwin van 't Wout

24 de septiembre de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl