

IMT2112 - Algoritmos Paralelos en Computación Científica

Multiprocessing
Elwin van 't Wout
8 de agosto de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl

Agenda

- ¿Como corre una computador un cálculo?
 - ¿Como ejecutar las instrucciones?
 - ¿Como mover los datos?
- ¿Como correr múltiples procesos?

Motivación

Para crear algoritmos eficientes es importante entender la arquitectura de un procesador

Recuerde que un clúster o supercomputador es un conjunto de procesadores ‘normales’

La arquitectura de Von Neumann

Sección 1.1 del libro de Eijkhout

La arquitectura de Von Neumann

- Antiguamente, las computadoras fueran programados a mano
- El principio de la arquitectura de Von Neumann es almacenar el programa
 - *stored-program architecture*
- Las instrucciones se manejan como datos
 - Esto permite el diseño de compiladores y editores

La arquitectura de Von Neumann

- El segundo principio de la arquitectura de Von Neumann es manejar instrucciones como la secuencia
 - fetch
 - execute
 - store
- La parte del procesador que ejecuta instrucciones se llama la CPU (*Central Processing Unit*)
- Los datos estan guardados en la memoria del procesador (RAM - *Random Access Memory*)
 - El almacenamiento es distinto y es parte del disco duro

Procesos y *threads*

- Procesos
 - Un programa que ejecuta tareas a través de la creación de *threads*
- *Threads* (hilo)
 - Un flujo secuencial de instrucciones de bajo nivel

Procesos y *threads*

- *Multiprocessing*
 - Un programa que crea otros programas
 - Cada programa tiene su propio *dataspace*
- *Multithreading*
 - Un programa que crea *threads* que corren simultáneamente
 - Todos los *threads* comparten el mismo *dataspace*
- *Hyperthreading*
 - Usar mas *threads* que el número de núcleos disponible, y cambiar entre ellos rápidamente

Lenguajes de programación

- Lenguajes interpretados
 - Código se ejecuta línea por línea
 - Instrucciones de alto nivel
 - Python, Julia, R, etc.
- Lenguajes compilados
 - Código es convertido en un ejecutable
 - Dos pasos: compilar y ejecutar
 - Fortran, C, C++, Java, etc.

Lenguajes interpretados

- Python tiene un bloqueo (GIL - *global interpreter lock*)
 - Hay un único *native thread* en el programa
 - Prohíbe el paralelismo entre diferentes líneas de código
 - Permite el paralelismo en rutinas externas
 - Por ejemplo *NumPy* que utiliza LAPACK y BLAS
- Python tiene una variedad a bibliotecas para la programación paralela
 - Cada una tiene alcances y limitaciones

Multiprocessing en Python

- La biblioteca “multiprocessing” implementa paralelismo en Python según el paradigma de *multiprocessing*
 - Efectivamente esta creando distintos programas de Python
 - Hay un *overhead* en crear y terminar programas
 - No hay datos compartidos o comunicación entre los programas
 - Solo se puede aplicar a un submuestra de la funcionalidad de Python
 - Sirve para problemas relativamente sencillo de parallelizar

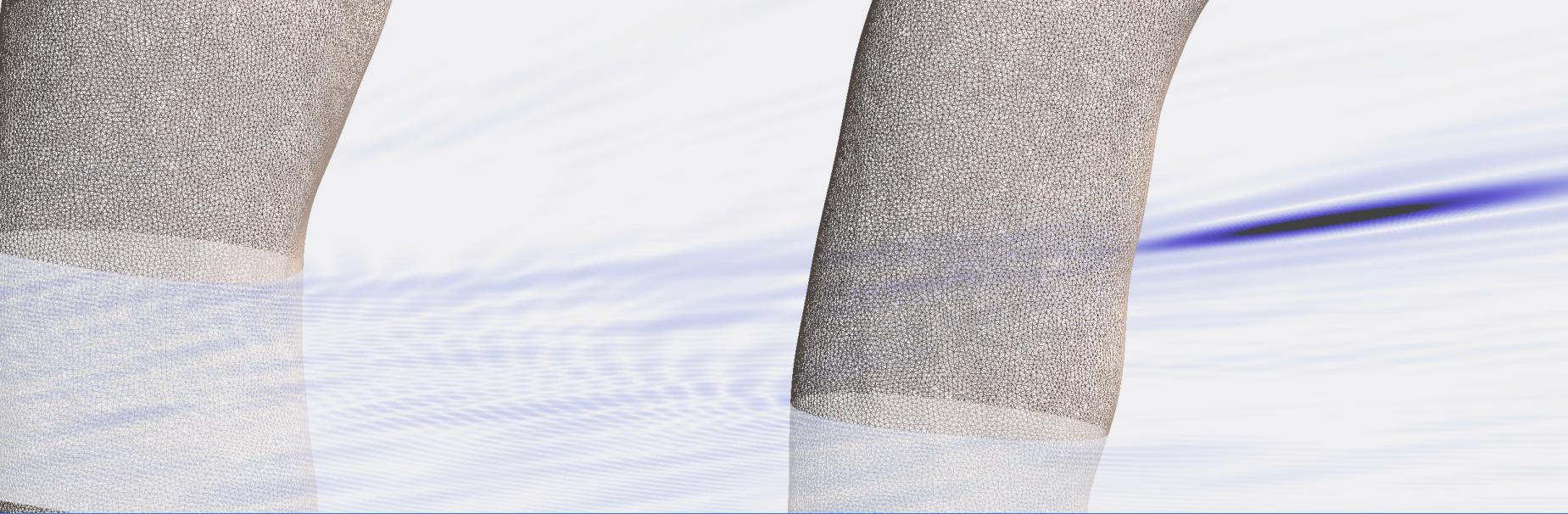
Resumen

¿Como se ejecuta instrucciones en una computador?

- La arquitectura de Von Neumann
 - Las instrucciones se manejan como datos
 - Secuencias de *fetch, execute, store*

¿Cómo paralelizar un programa secuencial?

- Multiprocessing



IMT2112 - Algoritmos Paralelos en Computación Científica

Multiprocessing
Elwin van 't Wout
8 de agosto de 2019



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Facultad de Matemáticas • Escuela de Ingeniería

imc.uc.cl