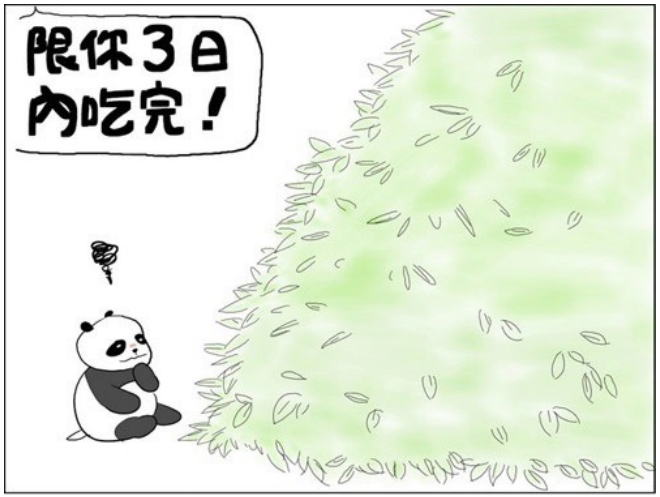
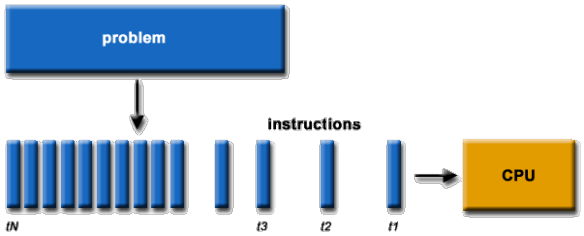
Parallel Programming

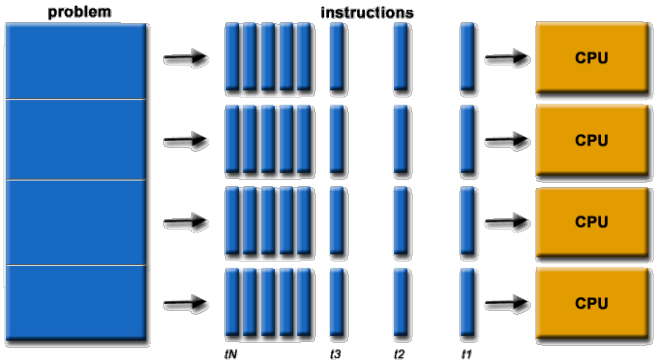
Course Introduction

Professor Yi-Ping You (游逸平)
Department of Computer Science
http://www.cs.nctu.edu.tw/~ypyou/

Parallel Processing/Computing







Finding the integer solutions for $x^3+y^3+z^3=3$

- The first two solutions might be obvious

$$1^3 + 1^3 + 1^3 = 3$$

$$4^3 + 4^3 + (-5)^3 = 3$$

- Is it even possible to know whether other solutions for 3 exist? (Louis Mordell, 1953)

- Roughly 4 million tasks (three hours/task)
- Using Charity Engine's grid comprised over 400,000 computers around the world

https://phys.org/news/2021-03-sum-cubes-puzzle-solution.html
https://www.pnas.org/doi/10.1073/pnas.2022377118
https://github.com/AndrewVSutherland/SumsOfThreeCubes

Motivation

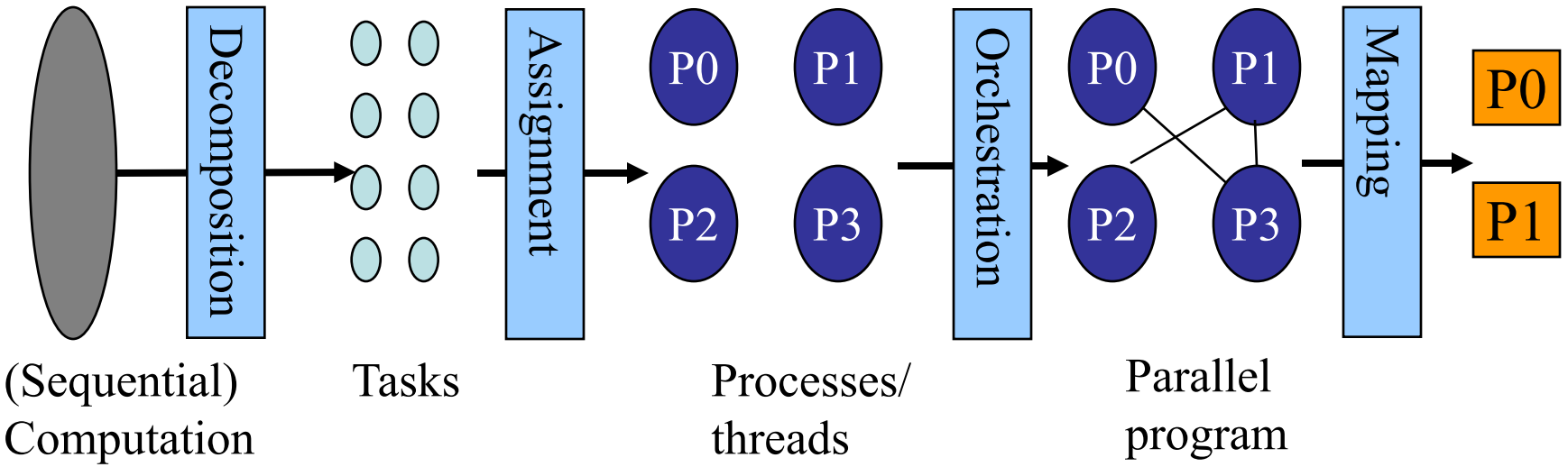
Creating a Parallel Program

- In theory, can be done by programmer, compiler, run-time system, or OS

- In practice, parallel programs are created with
  - Explicitly parallel language (e.g., High Performance Fortran)
  - Library for implementing a programming model
    - Shared-memory library (Pthreads, OpenMP)
    - Distributed-memory library (Message Passing Interface)
    - Heterogeneous-programming library (CUDA, OpenCL)
    - Cluster-based library (MapReduce)

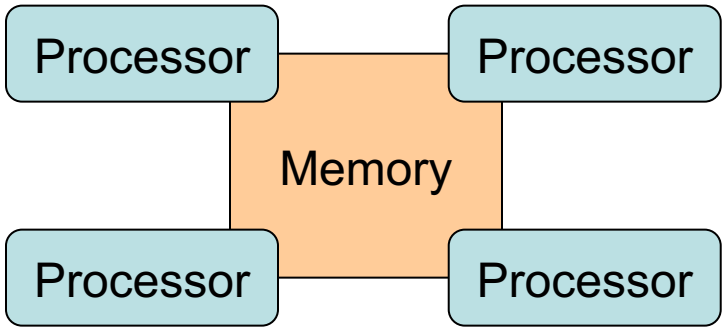
Steps for Creating a Parallel Program

- Decomposition into tasks
- Assignment of tasks to processes/threads
- Orchestration of data access, communication, etc.
- Mapping processes to processors



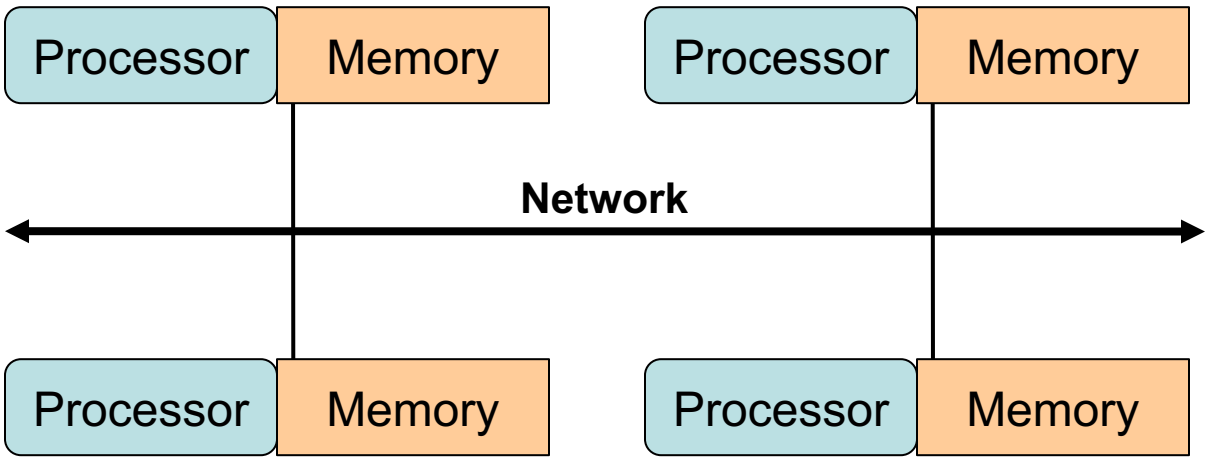
Shared-Memory Systems

- Multiple processors can operate independently but share the same memory resources



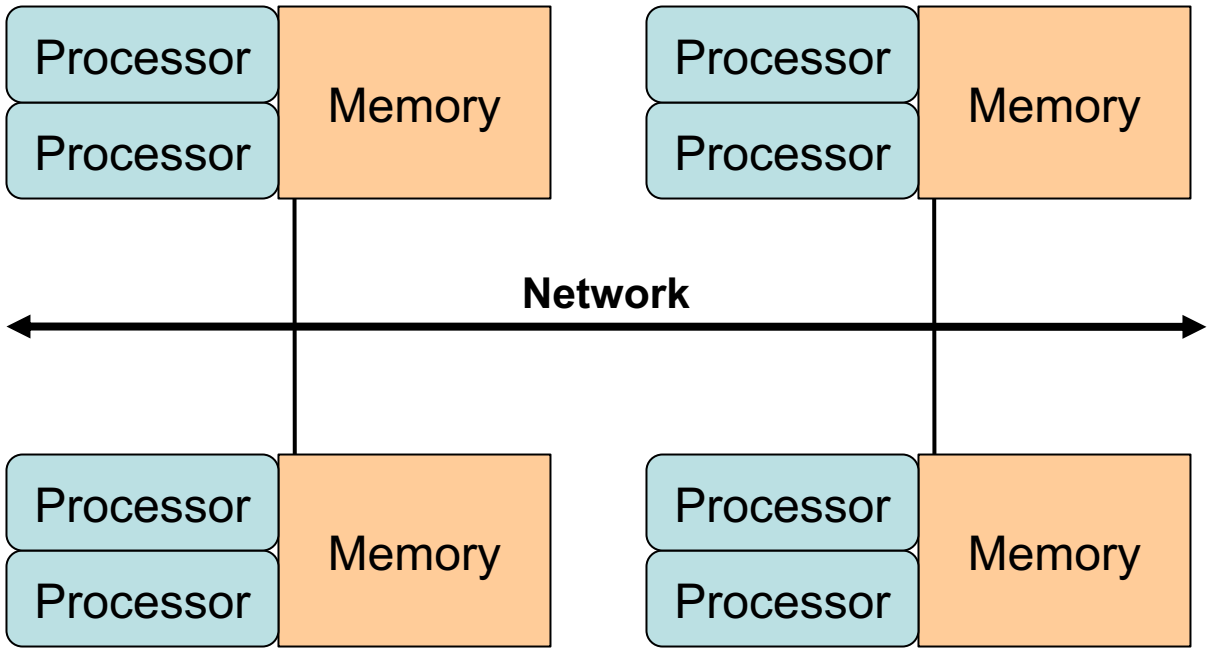
Distributed-Memory Systems

- Processors have their own local memory
- Memory addresses in one processor do not map to another processor
  - So there is no concept of global address space across all processors



Hybrid Distributed-Shared Systems

- The distributed memory component is the networking of multiple shared memory machines, which know only about their own memory - not the memory on another machine



Administrative Stuff

- Course information
  - Parallel Programing
  - Credit: 3
  - Time: Thursdays 9:00-12:00
  - Place: EDB27

- Course website
  - https://nycu-sslab.github.io/PP-f22/
  - The URL is also provided on my Web page
    - Authorization required to access lecture slides

Prerequisites

- This course assumes that you
  - write good code in C/C++ and
  - are familiar with the Linux environment
- Requirements
  - Have taken a C/C++ course or have a fair amount of practical experience with C/C++ programming
  - Know how to connect to remote machines with `ssh`, `scp`, etc.
  - Know how to work on Linux systems (i.e., basic commands such as `cd`, `rm`, and running executables and GNU tools such as `make` and `gcc`)

Aims of This Course

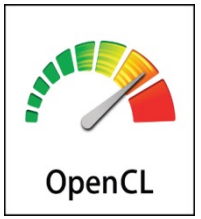
- The skills and knowledge needed to develop applications using parallel programing models

Lecture Topics

- Background
  - Parallel and distributed programming
  - Introduction to parallel hardware and software
- Shared-memory programming
  - Pthreads and OpenMP
- Distributed-memory programming
  - MPI and MapReduce
- GPGPU programming
  - CUDA and OpenCL

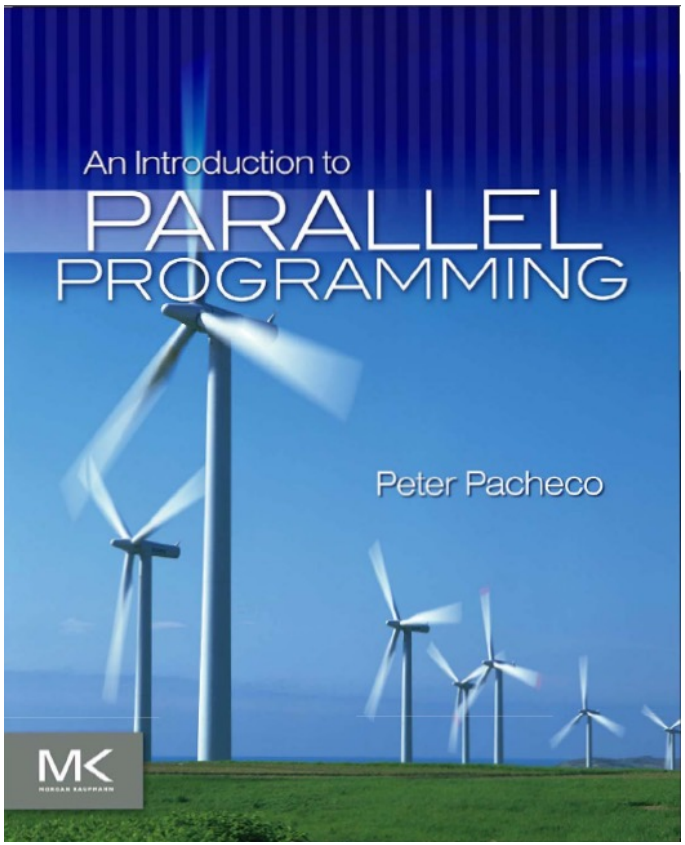
MPICH

Open MPI



Textbook

- Peter Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann; 1 edition (January 21, 2011)



Grading

- Grades will be assigned based on
  - Homework assignments (70%)
    - 6 assignments related to parallel programming
    - Slackers beware!
      - The penalty for late homework is **15% per day** (weekends count as 1 day).
    - NO PLAGIARISM!
      - Homework assignments must be individual work
  - Course project (30%)
    - 3 students form a group to work on development of parallel applications
    - Proposal (4%)
    - Final oral presentation (13%)
    - Final report (13%)
- These weights are subject to minor variation

Project Schedule

- Group registration due on September 29, 2022 (Week 3)
  - https://ppt.cc/fT5Eqx

- Project proposal due on October 27, 2022 (Week 7)
- Presentation slides due by 23:59 day prior to your presentation
- Final report and source codes due on January 3, 2023 (Week 17)

HW0: A Warming-Up Assignment

- To assess whether you are familiar with Makefile and C/C++ programming
- See the URL below for details
  - https://nycu-sslab.github.io/PP-f22/HW0/

Discord

- https://discord.gg/qnR9kw8Z

- For general discussion about the course, especially the assignments