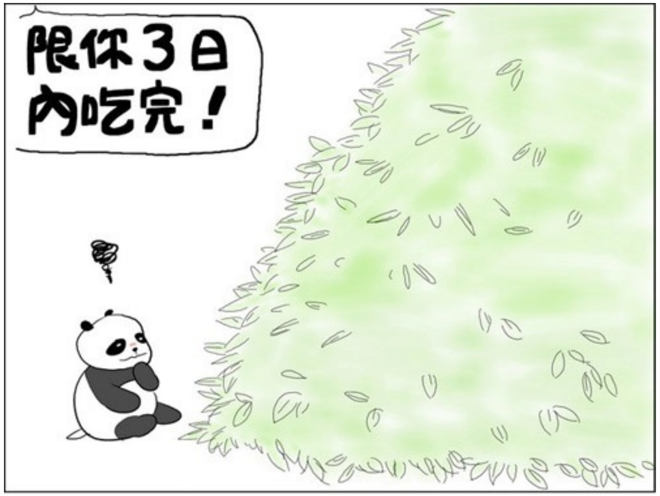
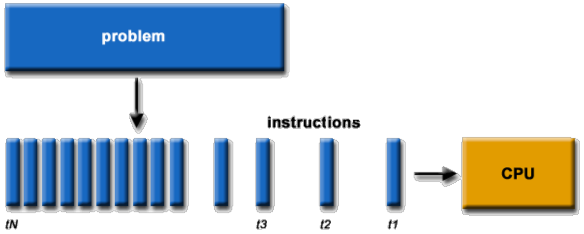
Parallel Programming

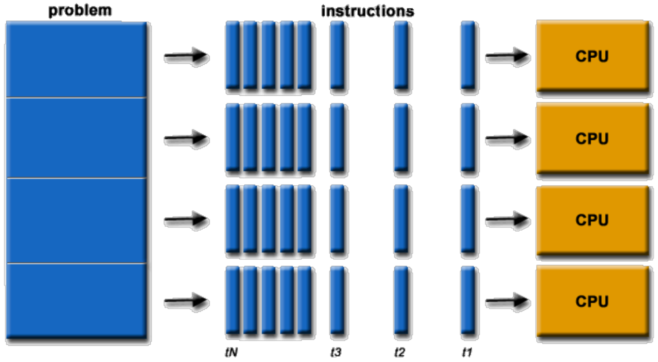
Course Introduction

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

Parallel Processing/Computing





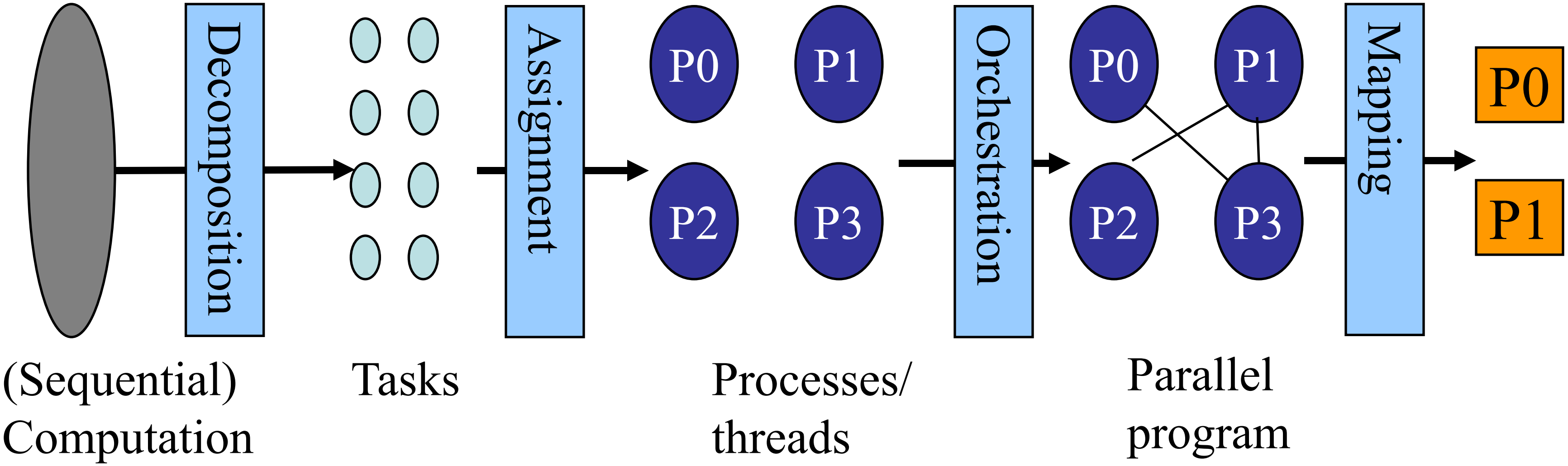


Creating a Parallel Program

- 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)

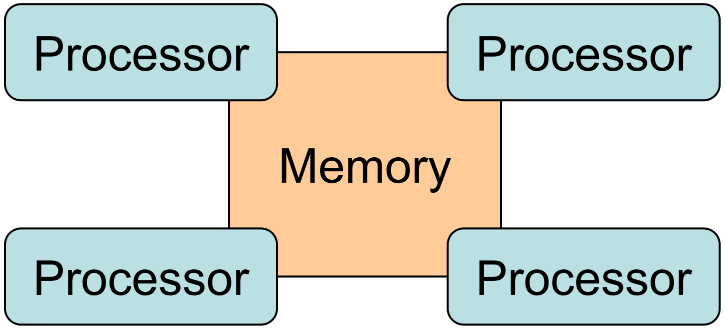
Steps for Creating a Parallel Program

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



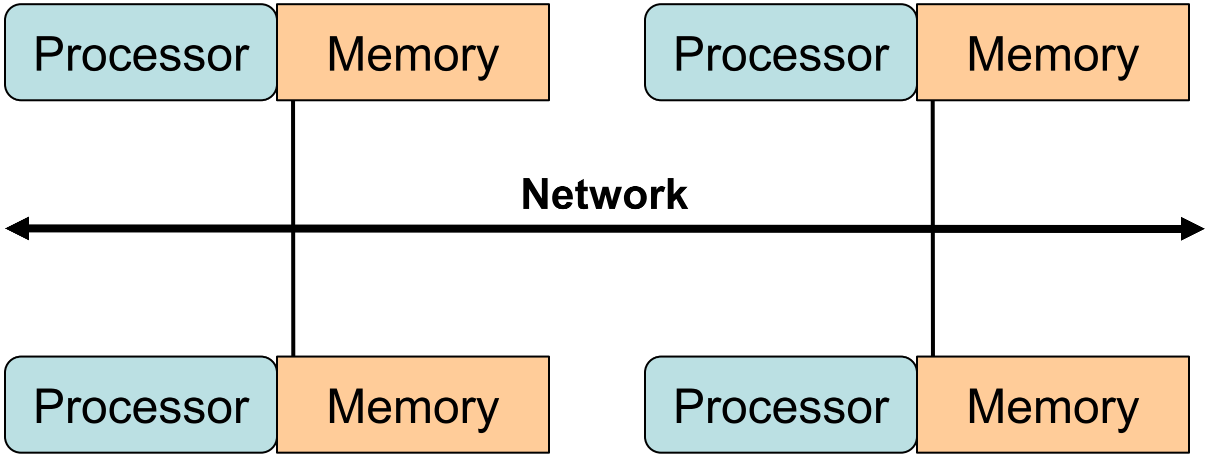
Shared-Memory Systems

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



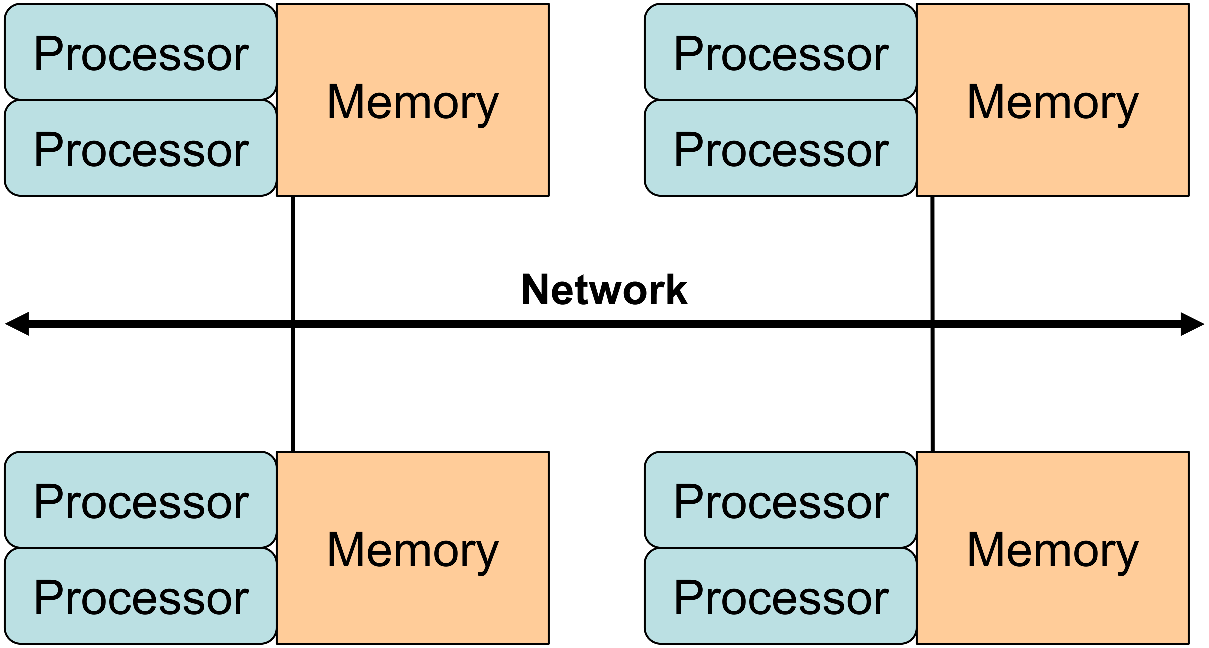
Distributed-Memory Systems

- 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: Tuesdays 15:30-16:20 and Fridays 10:10-12:00
  - Place: EDB27

- Course website
  - http://www.cs.nctu.edu.tw/~ypyou/courses/PP-f18/
    - The URL is also provided on my Web page
    - Authorization required to access course materials

Prerequisites

- You are assumed to have knowledge of the fundamentals of computer science
- You are assumed to have experience with programming, especially in C/C++ language

Aims of This Course

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

Lecture Topics

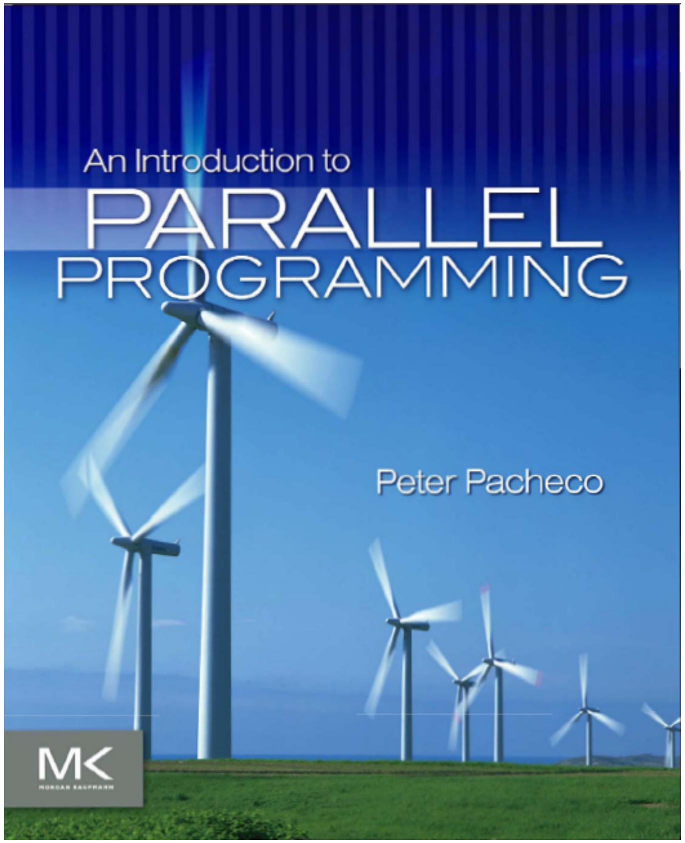
- Background
  - Parallel and distributed programming
  - Introduction to parallel hardware and software
- Shared-memory programming
  - Pthreads and OpenMP
- Distributed-memory programming
  - MPI
- GPGPU programming
  - CUDA and OpenCL
- Cloud programming
  - Hadoop (MapReduce)

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 (60%)
    - 5 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 (40%)
    - At most 3 students form a group to work on development of parallel applications
    - Proposal (5%)
    - Final oral presentation (15%)
    - Final report (20%)
- These weights are subject to minor variation

Project Schedule

- Group registration due on October 5, 2018
  - Registration link will be announced later
- Project proposal due on October 30, 2018
- Presentation slides due by 23:59 the day prior to your presentation
- Final report and source codes due on January 11, 2019