

Parallel Programming

MSBD5009

Course Introduction

Course Background

- An MSBD ELECTIVE
 - Assume CSE C/C++ programming, OS, Algorithms
 - Structured lectures based on reference books
 - Teach parallel programming knowledge
 - Practice parallel programming in three APIs
 - Workload
 - Three programming assignments + final exam
 - Exclusion COMP5112

Course Topics

- Introduction to parallel computer architectures
- Principles of parallel algorithm design
- Shared-memory programming models
- Message passing programming models
- Data-parallel programming models for GPUs
- Case studies of parallel algorithms, systems, and applications
- Hands-on experience with writing parallel programs for tasks of interest

Parallel Computer Architectures

- Review on OS and Computer Architecture
 - The von Neumann architecture
 - Processes, multitasking, and threads
 - Modifications to the von Neumann Model
 - Caches
 - Virtual memory
 - Instruction-level parallelism
 - Hardware multithreading
- Parallel Hardware
 - SIMD systems
 - MIMD systems
 - Interconnection networks
 - Cache coherence
 - Shared-memory versus distributed-memory

Principles of parallel algorithm design

- Preliminaries
 - Decomposition, Tasks, and Dependency Graphs
 - Granularity, Concurrency, and Task-Interaction
 - Processes and Mapping
- Decomposition Techniques
- Mapping Techniques for Load Balancing
- Methods for Containing Interaction Overheads
- Parallel Algorithm Models

Message passing programming models

- Principles of Message-Passing Programming
- Building Blocks: Send and Receive Operations
- MPI: the Message Passing Interface
- Collective Communication and Computation Operations
 - Gather, Scatter, Prefix, Reduction, Broadcast, Barrier, and so on

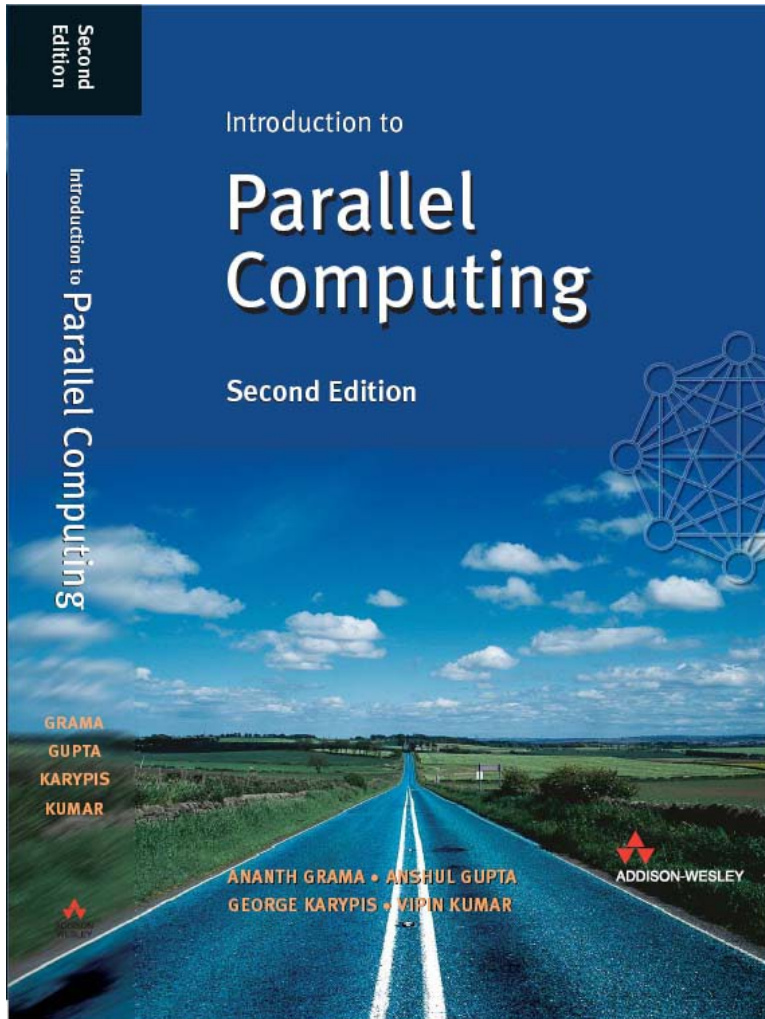
Shared-memory programming models

- Pthreads
 - Critical sections, busy-waiting, mutexes
 - Producer-Consumer Synchronization and Semaphores
 - Barriers and Condition Variables
 - Read-Write Locks
 - Caches, Cache Coherence, and False Sharing
 - Thread safety
- OpenMP

Data-parallel programming for GPUs

- CUDA C Language APIs
- CUDA Execution Model
- CUDA Memories
- Performance Considerations
- Parallel Patterns
 - Gather, Scatter, Reduction, Prefix Scan, and so on
- Case Studies

Reference Book 1



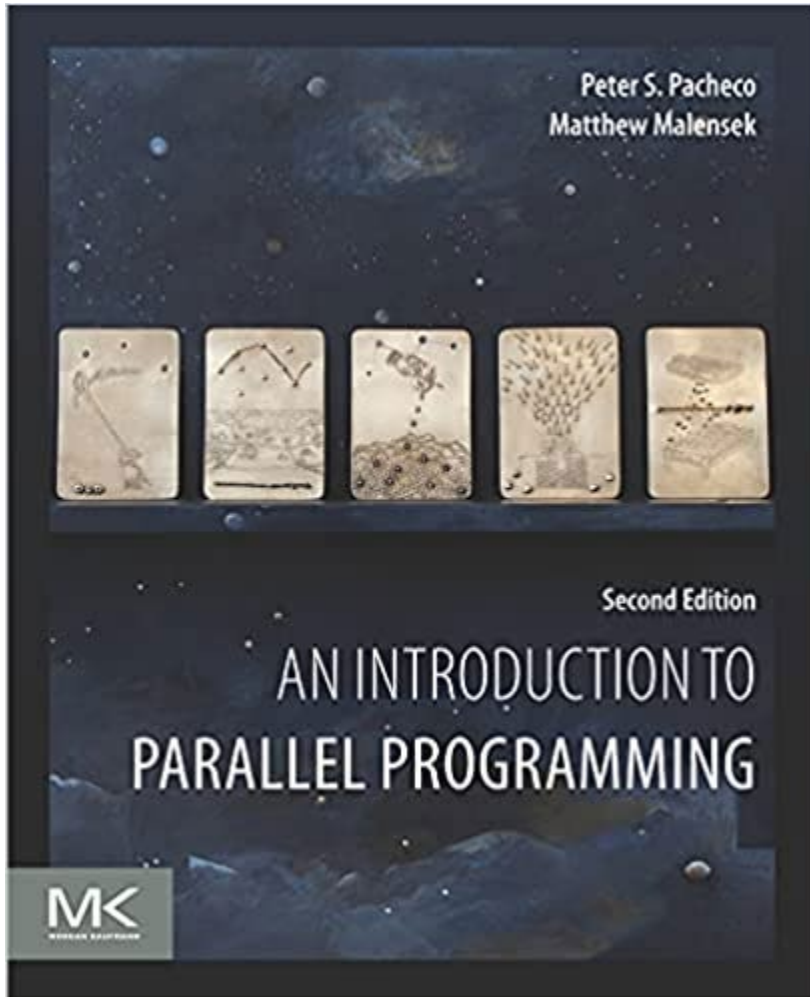
**Introduction to Parallel Computing
2nd edition**

By Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar.

Addison Wesley, 2003.

<https://www.cs.purdue.edu/homes/ayg/book/index.html>

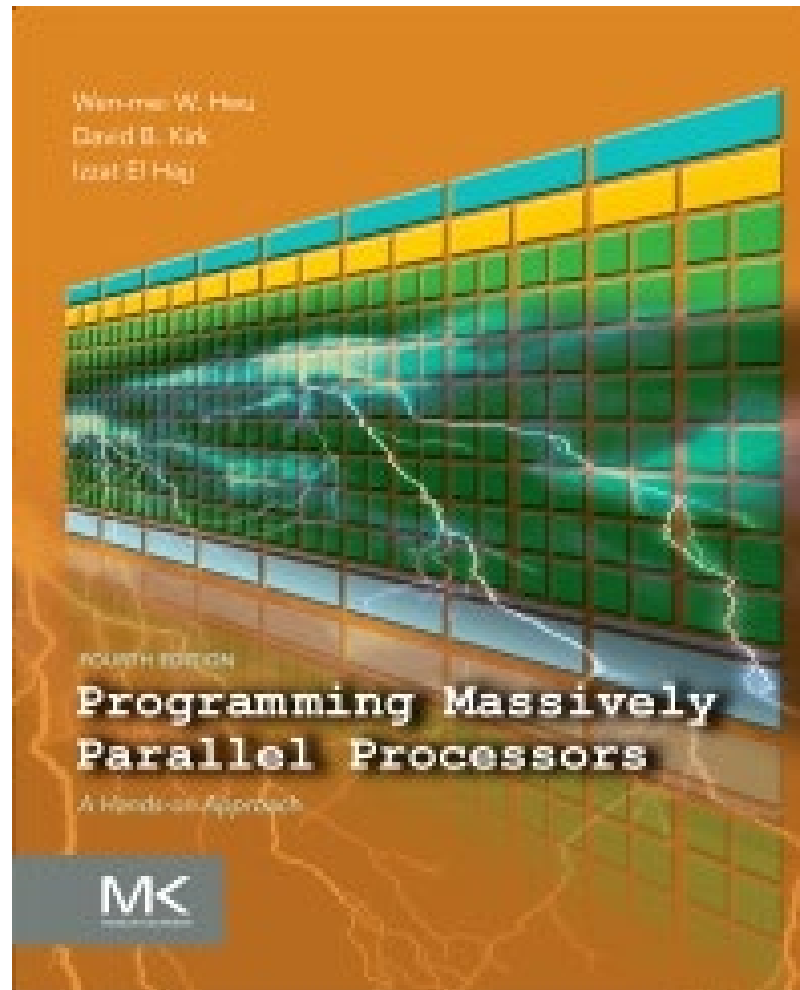
Reference Book 2



An Introduction to Parallel Programming. 2nd edition
By Peter Pacheco and
Matthew Malensek, 2022

<https://www.cs.usfca.edu/~peter/ipp2/index.html>

Reference Book 3



**Programming Massively
Parallel Processors:
A Hands-on Approach.
4th Edition
Wen-mei Hwu, David Kirk,
Izzat El Hajj
2022**

**[https://www.elsevier.com/
books/programming-
massively-parallel-
processors/hwu/978-0-
323-91231-0](https://www.elsevier.com/books/programming-massively-parallel-processors/hwu/978-0-323-91231-0)**

Lecture Time and Venue

- 13 lectures on Saturdays 3:00-5:50pm
 - Feb 8-May 10 (No class on Apr 19 public holiday)
- Room 2407 Lifts 17,18

Workload & Assessment

- Tentative plan
 - Three programming assignments 40%
 - Week 4, 7, 10 on MPI, Pthreads, CUDA
 - All assignments on a single topic (e.g., shortest path)
 - Sequential version program given (a few hundred lines of code)
 - Parallel program skeleton given
 - Your task is to fill in a few parallel components (tens of lines of code)
 - One final exam 60%
 - Programming: fill in code, similar to assignments
 - Short answer questions on concepts from course material

Lab Facilities

HKUST SuperPOD

- Each MSBD student has an account.
- TA will guide you to set up.
- Each account has 2000HKD credit in total for your entire study in the program.
- Get Started:

<https://itso.hkust.edu.hk/services/academic-teaching-support/high-performance-computing/superpod>

Academic Integrity

- You can discuss with others on ideas and bugs.
- Do not look up (e.g., using AI) solution code.
- Do not share or post your code.
- All code that you submit (other than skeleton code) should be written by you alone.
- Code plagiarism detection will be performed.
- Misconduct will be reported and penalized.