



## **HPC** **Introduction**

2020/21

# Key components of Advanced Computing



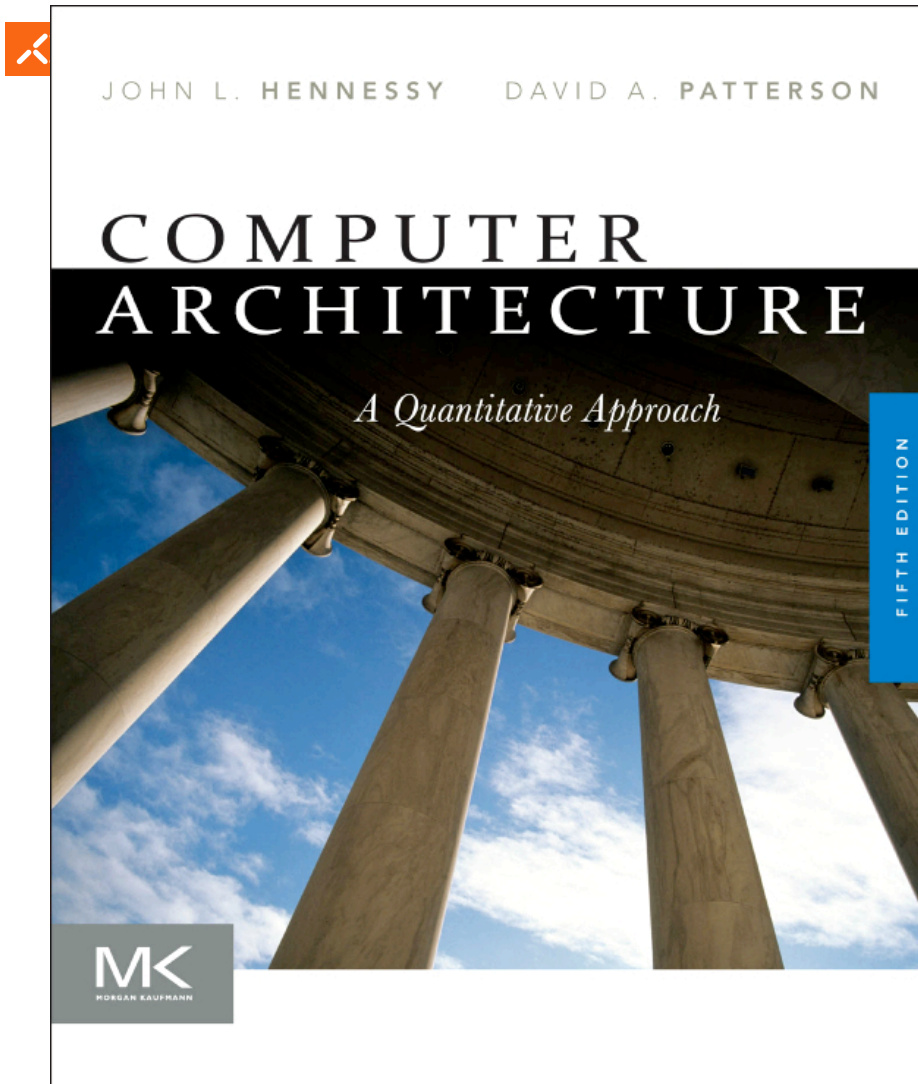
- Physical resources (*computing systems*)
  - Infrastructures
- Development of efficient parallel app's
  - Multi-threads in a shared memory environment  
(a server)
  - Multi-processes in a distributed memory environment  
(a cluster)

# Physical Resources: computing systems



- Sets of interconnected parallel servers
  - HPC systems: TOP500 rankings
  - Each server
    - with several PUs that share memory address space
    - may have computing accelerators (*GPU, ...*)
  - Each PU
    - supporting several levels of parallel execution: ILP, multi-scalar single-threaded, data parallel (vector), multi-threaded, multi-process in shared/distributed memory
- A memory system hierarchy
  - RAM distributed among servers (*communicate with MPI*)
  - RAM may shared among servers (*requires special h/w*)
  - Storage on dedicated servers of the “computer” file system

# Key textbook in Advanced Architecture



## Computer Architecture, 5th Edition

**Hennessy & Patterson**

### Table of Contents

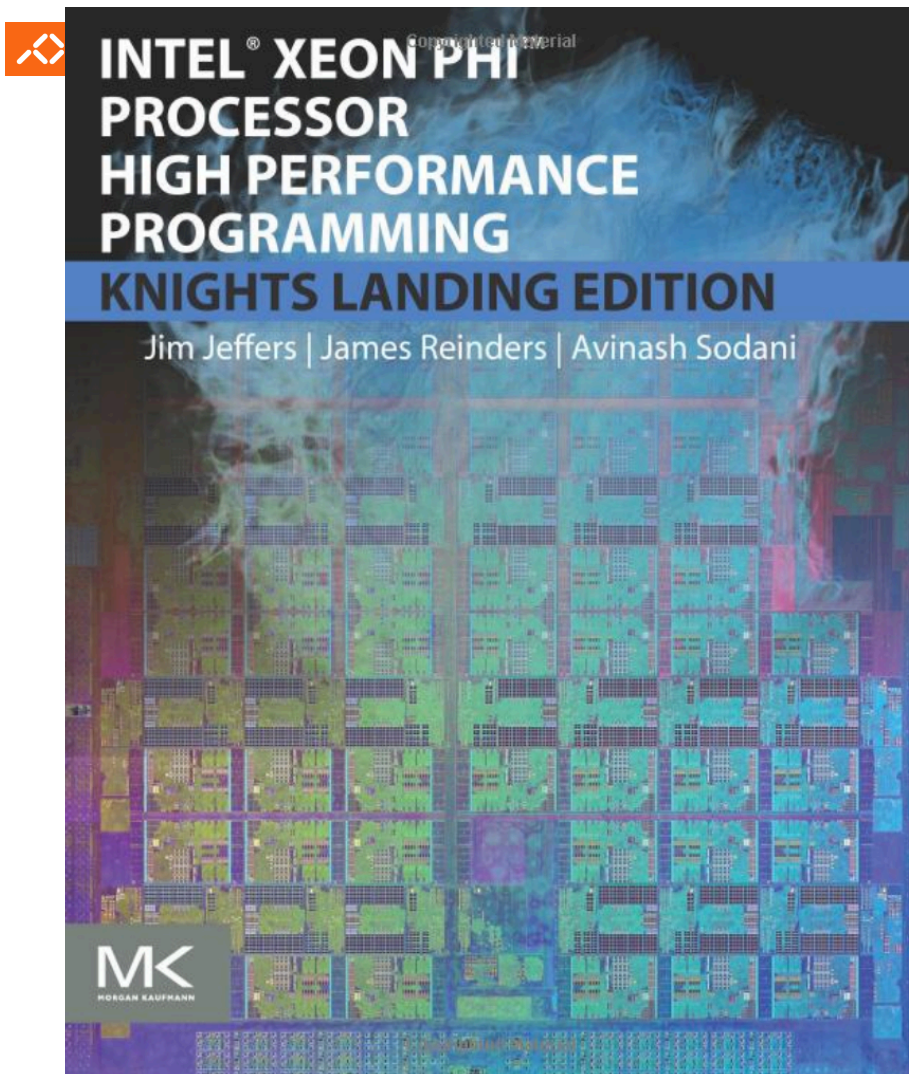
#### Printed Text

- Chap 1: Fundamentals of Quantitative Design and Analysis
- Chap 2: Memory Hierarchy Design
- Chap 3: Instruction-Level Parallelism and Its Exploitation
- Chap 4: Data-Level Parallelism in Vector, SIMD, and GPU Architectures
- Chap 5: Multiprocessors and Thread-Level Parallelism
- Chap 6: The Warehouse-Scale Computer
- App A: Instruction Set Principles
- App B: Review of Memory Hierarchy
- App C: Pipelining: Basic and Intermediate Concepts

#### Online

- App D: Storage Systems
- App E: Embedded Systems
- App F: Interconnection Networks
- App G: Vector Processors
- App H: Hardware and Software for VLIW and EPIC
- App I: Large-Scale Multiprocessors and Scientific Applications
- App J: Computer Arithmetic
- App K: Survey of Instruction Set Architectures
- App L: Historical Perspectives

## Recommended textbook (1)



### Table of Contents

#### Section I: Knights Landing.

**Chapter 1:** Introduction

**Chapter 2:** Knights Landing Overview

**Chapter 3:** Programming MCDRAM and Cluster Modes

**Chapter 4:** Knights Landing Architecture

**Chapter 5:** Intel Omni-Path Fabric

**Chapter 6:** Arch Optimization Advice

#### Section II: Parallel Programming

**Chapter 7:** Programming Overview for Knights Landing

**Chapter 8:** Tasks and Threads

**Chapter 9:** Vectorization

**Chapter 10:** Vectorization Advisor

**Chapter 11:** Vectorization with SDLT

**Chapter 12:** Vectorization with AVX-512 Intrinsics

**Chapter 13:** Performance Libraries

**Chapter 14:** Profiling and Timing

**Chapter 15:** MPI

**Chapter 16:** PGAS Programming Models

**Chapter 17:** Software Defined Visualization

**Chapter 18:** Offload to Knights Landing

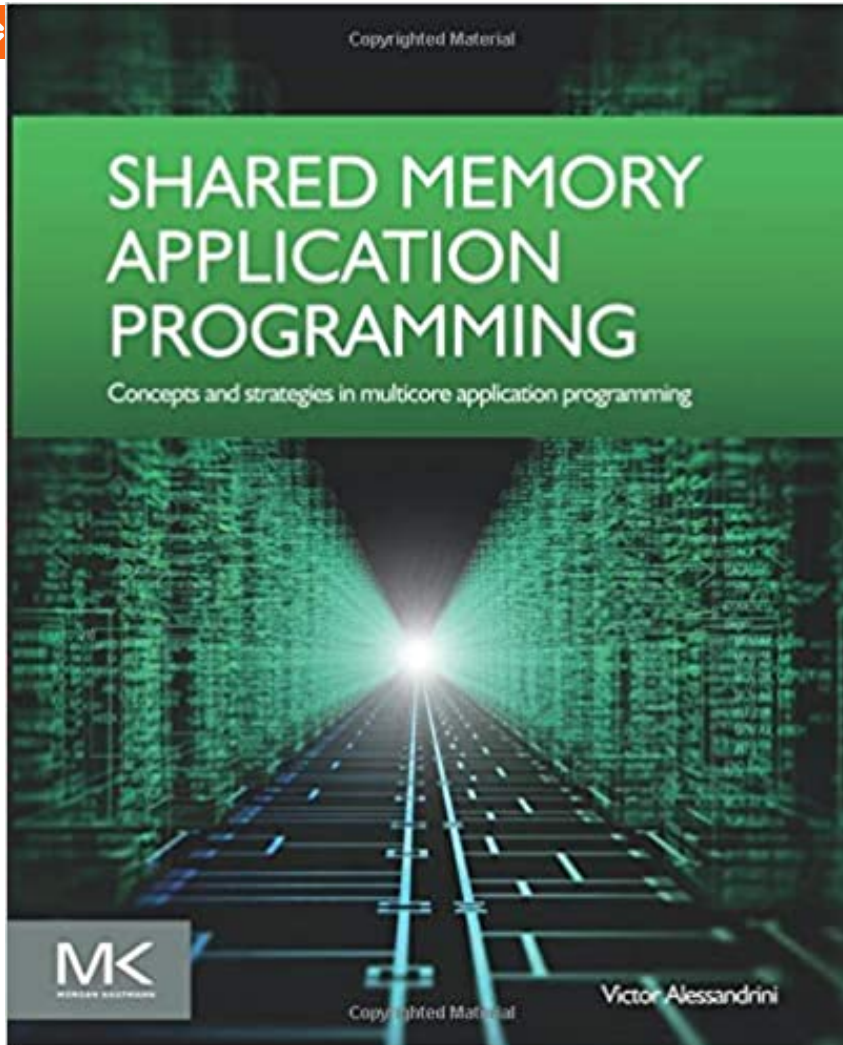
**Chapter 19:** Power Analysis

#### Section III: Pearls

**Chapters 20-26:** Results on LAMMPS, SeisSol, WRF, N-Body Simulations, Machine Learning, Trinity mini-applications and QCD are discussed.



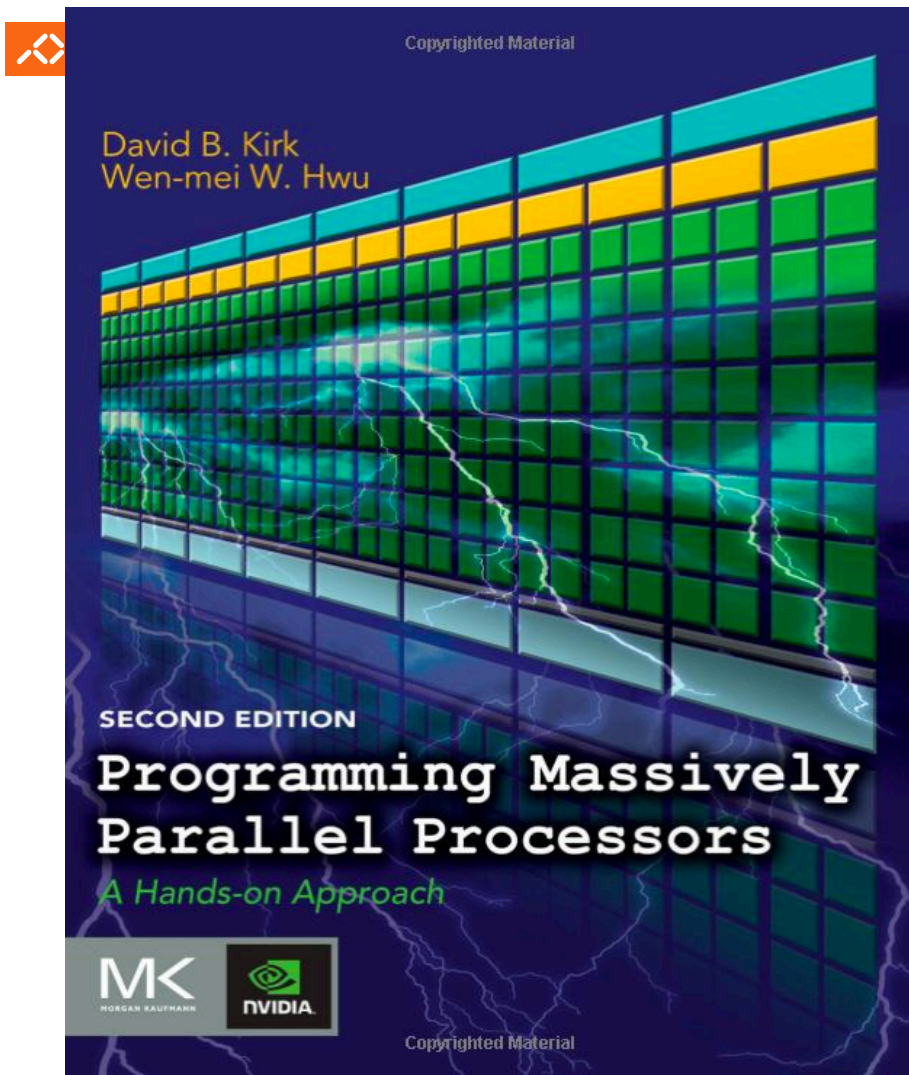
## Recommended textbook (1)



### Table of Contents

- 1: Introduction and Overview
- 2: Introducing Threads
- 3: Creating and Running Threads
- 4: Thread-Safe Programming
- 5: Concurrent Access to Shared Data
- 6: Event Synchronization
- 7: Cache Coherency and Memory Consistency
- 8: Atomic Types and Operations
- 9: High-Level Synchronization Tools
- 10: OpenMP
- 11: Intel Threading Building Blocks
- 12: Further Thread Pools
- 13: Molecular Dynamics Example
- 15: Pipelining Threads
- 16: Using the TBB Task Scheduler
- Annex A: Using the Software
- Annex B: C++ Function Objects and Lambda Expressions

## Recommended textbook (2)



### Contents

- 1 Introduction
- 2 History of GPU Computing
- 3 Introduction to Data Parallelism and CUDA C
- 4 Data-Parallel Execution Model
- 5 CUDA Memories
- 6 Performance Considerations
- 7 Floating-Point Considerations
- 8 Parallel Patterns: Convolution
- 9 Parallel Patterns: Prefix Sum
- 10 Parallel Patterns: Sparse Matrix-Vector Multiplication
- 11 Application Case Study: Advanced MRI Reconstruction
- 12 Application Case Study: Molecular Visualization and Analysis
- 13 Parallel Programming and Computational Thinking
- 14 An Introduction to OpenCL
- 15 Parallel Programming with OpenACC
- 16 Thrust: A Productivity-Oriented Library for CUDA
- 17 CUDA FORTRAN
- 18 An Introduction to C11 AMP
- 19 Programming a Heterogeneous Computing Cluster
- 20 CUDA Dynamic Parallelism
- 21 Conclusion and Future Outlook