Textbook # 1: Introduction to Parallel Programming, 2011, By Peter S. Pacheco
Textbook # 2: Introduction to Parallel Computing, 2nd Ed (2003), By Ananth Grama
Textbook # 3: Programming Massively Parallel Processors A Hands-on Approach,
3rd Ed (2017), By David B. Kirk Wen-mei W. Hwu

// 0. Reading Assignment for Background Review

OS Revision Reading assignment: The von Neumann architecture, Processes, multitasking, Thread Basics, Why Threads?, The POSIX Thread API, Tips for Designing Asynchronous Programs. **QUIZ # 0** (in all sections)

Course mechanics

Midterms 30, Final 50, Quizzes 5, Assignment 5, PDC programming Project 10
Unfair practices in sessionals and exams: First time offenders (providers+copiers)shall be punished with 10 weightage.

- Project splitted into: Proposal, Midterm evaluation, Final (DEMO and Viva)
- Processing of large data e.g. 1000x1000 matrixes, 500K elements etc.

// 1. PDC Theory

Week # 1-2-3-4:

W1: CHAPTER 1 Why Parallel Computing? (Textbook # 1)

1.1 Why We Need Ever-Increasing Performance, 1.2 Why We're Building Parallel Systems,
1.3 Why We Need to Write Parallel Programs, 1.4 How Do We Write Parallel Programs?,
1.5 Course coverage, 1.6 Concurrent, Parallel, Distributed

https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/x2d2f703b37b450a3:parallel-and-distributed-computing/a/parallel-computing

W2/W3/W4 CHAPTER 2 Parallel Hardware and Parallel Software (Textbook # 1)
* Topics covered in previous courses will be covered by self-reading

2.2 Modifications to the von Neumann Model
(Memory Hierarchy and Locality, The basics of caching, Cache mappings, Caches, and programs: an example, Instruction-level parallelism, Hardware multithreading)
2.3 Parallel Hardware
(SIMD systems, MIMD systems, Interconnection networks, Cache coherence, Shared Memory versus distributed memory, data and task parallelism,)
2.4 Parallel Software
(Coordinating the processes/threads, Shared-memory, Distributed-memory, Programming hybrid systems)
2.5 Input and Output
2.6 Performance (Speedup and Efficiency, Amdahl's law, Scalability, Gustafson's law)
2.7 Parallel Program Design
2.8 Writing and Running Parallel Programs

Performance Analysis Techniques (may need handouts)

- Profiling and Benchmarking, Load Balancing, Communication Overhead,

Tuning Techniques (may need handouts)

- Algorithm Optimization and adaptations, Granularity, Synchronization (locks and barriers, avoid delays by minimizing critical sections), Compiler Optimizations.

// 2. Introduction to OpenMP and MPI

Week # 5-6 (midterm #1)-7:

Chapter 7. Programming Shared Address Space Platforms (Textbook # 2)
Section 7.10. **OpenMP**: a Standard for Directive-Based Parallel Programming
`End-of-the-Chapter Problems: 7.1 to 7.16`

/* Week # 4-5: Quiz #1 (2.5), OpenMP Assignment # 1 (2.5) */
/* Week # 6: Midterm # 1 (15) */
/* Week # 7: Start of Semester Project after proposal approval (2) */

Week # 8-9:

Chapter 6. Programming Using the Message-Passing Paradigm (Textbook # 2)
Section 6.1. Principles of Message-Passing Programming
Section 6.2. The Building Blocks: Send and Receive Operations (Only simple buffered Send/Receive)
Section 6.3. **MPI**: the Message Passing Interface
Section 6.5. Overlapping Communication with Computation
Section 6.6-7. Collective Communication and Computation Operations, Groups and Communicators
`End-of-the-Chapter Problems: 6.1 to 6.9`

Parallel Algorithms (Discuss parallel code of two in class and implementation using semester project)

- Parallel Sorting (Bitonic Sort, QuickSort), Parallel Binary Search, Depth-First/Breadth-First Searches, Parallel Shortest Path Algorithms, Parallel Minimum Spanning Tree.

// 3. Parallel Algorithm Design (using two research papers)

Week # 10-11-12 (midterm #2)-13:

[2 lectures] **Chapter 3. Principles of Parallel Algorithm Design (Textbook # 2)**
Section 3.3. Characteristics of Tasks and Interactions
Section 3.6. Parallel Algorithm Models

1. "**The Hadoop Distributed File System**", Konstantin Shvachko et. al. Yahoo!
2. "**MapReduce: Simplified Data Processing on Large Clusters**", Jeffrey Dean/Sanjay Ghemawat, Google.

/* Week 10-11: Quiz #2 (2.5), MPI Assignment # 2 (2.5) */
/* Week # 12: Midterm # 2 (15) */
/* Week # 13-14: Evaluation of Semester Project (6+2) */

// 4. GPGPU Programming + AWS ParallelClusters platform

Week # 14-15:

[Textbook # 3] **CHAPTER 2-Data-Parallel Computing + CHAPTER 3-Scalable Parallel Execution (few topics)**

- Data Parallelism, CUDA C Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Kernel Launch, CUDA Thread Organization, Mapping Threads to Multidimensional Data, Querying Device Properties

[Handouts] AWS ParallelCluster on Amazon EC2

- Introduction, architecture, brief implementation and performance results.

/* Week #16: Final Exam (50) - 40% from Mid1+Mid2 syllabus not given in Mid papers */

—-------(X)----------