READINGS & SCHEDULE -- 2015

L00: Class 1 (1) – Intro, motivation, overview

- "How to Write Fast Numerical Code: A Small Introduction," S. Chellappa, et al. ; CMU Tech Report
- "Programmability: Design Costs and Payoffs Using AMD GPU Streaming Languages and Traditional Multi-Core Libraries," Weber, SAAHPC, Extended Abstract and Lecture Slides
- "How to Write Fast Code," Markus Pueschel, Lecture Slides, Class 1
- "Debunking the 100x GPU versus CPU Myth: An evaluation of throughput computing on CPU and GPU," V.W. Lee, et al. ISCA10.

L01a, L01b: Class 2 (1) – Review memory hierarchy, Memory-aware optimizations

- "How to Write Fast Numerical Code: A Small Introduction," S. Chellappa, et al. ; CMU Tech Report
- Basic memory hierarchy from any standard computer architecture textbook
- "Computer Systems: A Programmer's Perspective," Bryant & O'Hallaron, Chapter 6 (parts).

L02a, L02b: Classes 3-4 (2) -- CPU-aware optimizations, compiler interactions

- "Computer Systems: A Programmer's Perspective," Bryant & O'Hallaron, Chapter 5.
- "How to Write Fast Numerical Code: A Small Introduction," S. Chellappa, et al. ; CMU Tech Report

L03a, L03b: Class 5-6 (2) – Vector Processing, SIMD, Data Parallel Programming, Vector extensions

- Slides from David Patterson
- "How to Write Fast Numerical Code: A Small Introduction," S. Chellappa, et al. ; CMU Tech Report
- H&P 5e Appendix G, Vector Processors
- "How to Write Fast Code," Markus Pueschel, Lecture Slides, Classes 13-14
- "Introduction to SSE Programming," Alex Fr, The Code Project, http://www.codeproject.com/KB/recipes/sseintro.aspx

L04: Class 7 (1) – Advanced single-core optimizations

- Slides from Michelle Hugue and David Patterson
- "How to Write Fast Numerical Code: A Small Introduction," S. Chellappa, et al. ; CMU Tech Report
- "Exploiting Instruction-Level Parallelism with Software Approaches," H&P 3e Chapter 4, Sections 1-4
- "Hardware and Software for VLIW and EPIC," H&P 4e Appendix G, pp. G1-G15
- "Static Multiple Issue," P&H 4e Chapter 4.10, pp. 393-398
- "Loop Unrolling Tutorial," Michelle Hugue, University of Maryland

---

Part II – Classes 8-13

L05: Classes 8, 10 (2) – Parallel Programming

- "Parallel Computer Architecture -- Draft," Culler, et al.: Chapter 2, "Parallel Programming"
- "Parallel Computer Architecture -- Draft," Culler, et al., Chapter 3, Sections 3.1, 3.2.2, 3.4.1, "Programming for Performance"
- "Type Architectures," L. Snyder

L06: Class 9 – Programming with Threads, Review Concurrency and Synchronization

- Slides from H. Casanova, U Delaware, and U Hawaii
- "POSIX Threads Programming," B. Barney, LLNL

L07, L08: Class 11 – Review Concurrency and Synchronization, OpenMP
- "Operating Systems Concepts" by Peterson and Silberschatz, Chapter 9
- Slides from BU SCV

L09: Classes 12 (1) – Cache for Shared Memory
- "Parallel Computer Architecture", by Culler, et al. Chapter 5, Sections 5.1-5.4

L10: Classes 13 (1) – Synchronization implementation for shared memory processors
- "Parallel Computer Architecture", by Culler, et al. Chapter, Section 5.5
- "Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors," by J.M. Mellor-Crummy & M.L. Scott.
- Lecture Notes -- M. Herlihy

Part III – Classes 14-19, 21

L11-L16: Classes 14-19, 21 (7) – NVIDIA GPUs, CUDA, Performance optimization, case studies
- Slides from Kirk and Hwu
- "Programming Massively Parallel Processors" by Kirk and Hwu
- Various case studies, esp. from Molecular Dynamics

Class 20-- Mid-Term
Classes 22, 23 – Replace with group meetings with instructor
Classes 24-26 -- Project presentations