Additional Reading for COMP35112

John Gurd

March 15th 2012

Lecture 1 – Introduction

Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 4th Edition, Morgan Kaufman, 2007. Chapter 1.

Lecture 2 – The World of Parallelism

Parallel Computer Architecture: A Hardware/Software Approach, D.E. Culler and J.P. Singh, Morgan Kaufman, 1999. Chapter 1.

Lecture 3 – Parallel Programming using Data Sharing

Consult the Java library class documentation on the web for information about the Java features. Google "pthreads" leads to lots of documentation about pthreads.

Lecture 4 – Shared Memory Multiprocessors

Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 4th Edition, Morgan Kaufman, 2007. Chapter 4, Sections 1-3.

Lecture 5 – Other Cache Coherence Protocols

Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 4th Edition, Morgan Kaufman, 2007. Chapter 4, Section 4.

Lecture 6 – Programming with Locks and Barriers

The Art of Multiprocessor Programming, M. Herlihy and N. Shavit, Morgan Kaufman, 2008. This book is generally useful on this topic, especially Chapter 9. Note however that some of the material presented is rather more advanced than that covered in the course unit.

Lecture 7 – Hardware Support for Synchronisation

Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 4th Edition, Morgan Kaufman, 2007. Chapter 4, Section 5.

Lecture 8 – Lock-Free Data Structures

The Art of Multiprocessor Programming, M. Herlihy and N. Shavit, Morgan Kaufman, 2008. This book is generally useful on this topic, especially Chapter 10. As mentioned above, some of the material presented is rather more advanced than that covered in the course unit.

Lecture 9 – OpenMP and MPI

Patterns for Parallel Programming, T. Mattson, B. Sanders B.Massingill, Addison-Wesley, 2005. The lecture was largely based on Appendices A and B of this book. The specification for OpenMP is readily available at http://www.openmp.org.

Lecture 10 – Speculation

Design Space Exploration of a Software Speculative Parallelization Scheme, M. Cintra and D.R. Llanos, *IEEE Transactions on Parallel and Distributed Systems*, vol. 16 no. 6, pp. 562-576, June 2005.

Lecture 11 – Transactional Memory (1)

Transactional Memory, T. Harris, J. Larus and R. Rajwar, 2nd Edition, Morgan & Claypool, 2010. This is the definitive book on this topic, but it covers far more material than was presented in this lecture and the next lecture. Chapter 18 of **The Art of Multiprocessor Programming**, M. Herlihy and N. Shavit, Morgan Kaufman, 2008 is also useful (and again goes well beyond the lectures).

Lecture 12 – Hardware Support for Transactional Memory

Transactional Memory, T. Harris, J. Larus and R. Rajwar, 2nd Edition, Morgan & Claypool, 2010. See also: **Transactional Coherence and Consistency: Simplifying Parallel Hardware and Software**, L. Hammond, B.D. Carlstrom, V. Wong, M. Chen, C. Kozyrakis, K. Olukotun, *IEEE Micro*, vol. 24 no. 6, pp. 92-103, November-December 2004.

Lecture 13 – Memory Consistency

Computer Architecture: A Quantitative Approach, J.L. Hennessy and D.A. Patterson, 4th Edition, Morgan Kaufman, 2007. Chapter 4, Section 6.

Lecture 14 – GPUs, CUDA and OpenCL

Programming Massively Parallel Processors, D.B. Kirk and W.W. Hwu, Morgan Kaufman, 2010 is an entire book on this topic. See also: **NVIDIA Tesla: A Unified Graphics and Computing Architecture**, E. Lindholm, J. Nickolls, S. Oberman and J. Montrym, *IEEE Micro*, vol. 28 no. 2, pp. 39-55, March-April 2008.

Lecture 15 – Functional Programming and Dataflow Principles

Introduction to Functional Programming, R. Bird and P. Wadler, Prentice-Hall International, 1988. **Functional Programming**, A. Field and P. Harrison, Addison Wesley, 1988. **Functional Programming**, P. Henderson, Prentice-Hall International, 1980. These cover functional programming in general and go well beyond what is needed for this course unit. An interesting paper about parallel implementation of such languages is: **Parallel Implementations of Functional Programming Languages**, S. Peyton Jones, *Computer Journal*, vol. 32 no. 2, pp. 175-186, April 1989.
The Manchester Prototype Dataflow Computer, J.R. Gurd, C.C. Kirkham and I. Watson, *Communications of the ACM*, vol. 28 no. 1, pp. 34-52, January 1985. **A Report on the SISAL Language Project**, J. Feo, D. Cann and R. Oldehoeft, *Journal of Parallel and Distributed Computing*, vol. 10 no. 4, pp. 349-366, December 1990.