Future Multi-core Computing
COMP60011

Ian Watson & Mikel Lujan
Advanced Processor Technologies Group

http://www.cs.manchester.ac.uk/apt/COMP60011/

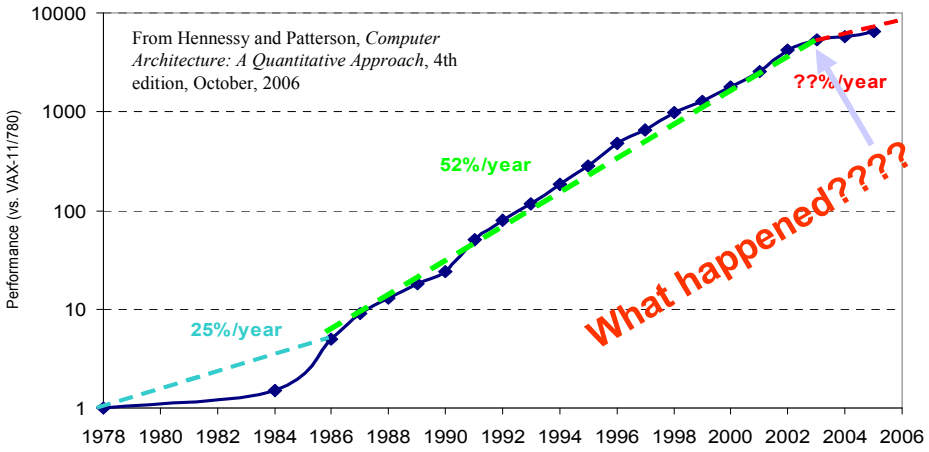
Multi-Cores are Coming (here?)

- Many processors in normal desktops/laptops are 'dual core' some are 'quad core'
  - What does this mean?
  - Why is it happening?
  - How are they different?
  - Where are they going?
  - Do they change anything?

The need for Multi-Core

- For over 30 years the performance of processors has doubled every 2 years (Moore's Law)
- Driven mainly by shrinkage of circuits
- Smaller circuits go faster
- In early 2000s the rate started to decrease
- But, smaller circuits mean more transistors per chip

Uniprocessor Performance



- VAX : 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2002
- RISC + x86: ??%/year 2002 to present

The End of "Good Times"

- Slowdown for several reasons
  - Power density increasing (more watts per unit area) - cooling a serious problem
  - Small transistors have less predictable characteristics
  - Architectural innovation hitting design complexity problems
  - Memory does not get faster at the same rate as processing

So the solution is replication

- Put multiple CPUs (cores) on a single integrated circuit (chip)
- Use them in parallel to achieve higher performance
- Simpler to design than a more complex single processor
- Need more computing power – just add more cores?

But How to Connect Them?

- Could have independent processor/store pairs with interconnection network
- But majority opinion (mainly due to software issues) is that shared memory is the right answer for general purpose
- But when we consider more than a few cores (>8?) shared memory becomes more difficult

Can We Use Multiple Cores?

- Small numbers of cores can be used for separate tasks – e.g. run a virus checker on one core and application on another
- But if we want increased performance on a single application we need to move to parallel programming
- General purpose parallel programming is known to be hard – consensus is that new approaches are needed

So We Have Problems

- We don't know how to engineer extensible memory systems
- We don't know how to write general purpose parallel programs
- If we develop new approaches to parallel programming does it fit with existing serial processor hardware designs?
- All these are current research issues

Aims of the Course

- To understand current multi-core systems and their limitations
- To study research directions both in hardware and software for (more highly parallel) future multi-core systems
- To gain practical experience of using both real and simulated multi-core systems

Teaching Style

- 50% Watson & 50% Lujan
- First day
  - standard teaching for foundations
  - Introductory laboratory
- Days 2 to 5 a mixture of
  - Directed reading and
  - Lab sessions
- Directed reading
  - Selection of technical and research papers
  - Work in groups
  - Each paper will have a "leader" group
  - Presentation of the paper by the "leader" group
  - Followed by technical discussion by all the groups

Technical Discussion

- What is the problem?
- What is the solution?
- How are solutions evaluated?
- What are the limitations?
- What are the assumptions?
- How can solutions be improved?
- What other approaches could be taken?

Lab Sessions

- Hands on experience with
  - programming multi-core processors
    - Threads plus locks
    - Non-blocking algorithms
    - Transactional memory
    - Sun Microsystems has donated equipment (Niagara 2)
  - using computer architecture simulators

Logistics

- Period 4
  - 9th Nov - 14th Dec 2009
  - Mondays
- Pre-requisites
  - Java programming
  - Basic knowledge of computer architecture
- Assessment
  - 50% Exam
  - 20% Presentations & Discussions
  - 30% Labs

Fundamentals - The Major Issues (1)

- We don't know how to build multi-core processors with shared memory which are extensible – i.e. with lots of cores
- To understand why we need to examine some aspects of processor structure in more detail
- In particular
  - Current Processor Complexity
  - Caches – the crux of a modern memory system
  - Interconnection networks

Fundamentals - The Major Issues (2)

- Even if we had an idealised multi-core processor, programming it for general purpose applications is complex and error prone.
- To understand why we need to examine several aspects of approaches to parallel software
  - The nature of parallelism in programs
  - Parallel sub-tasks (Threads)
  - The need to synchronise threads
  - Problems with synchronisation
  - Processor support for synchronisation