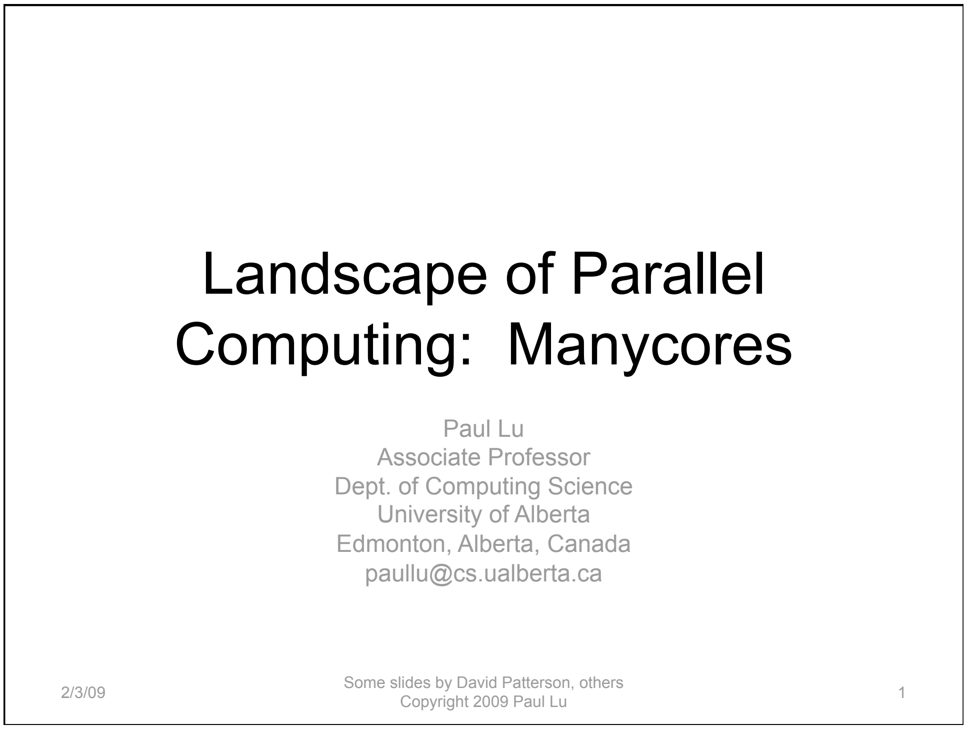
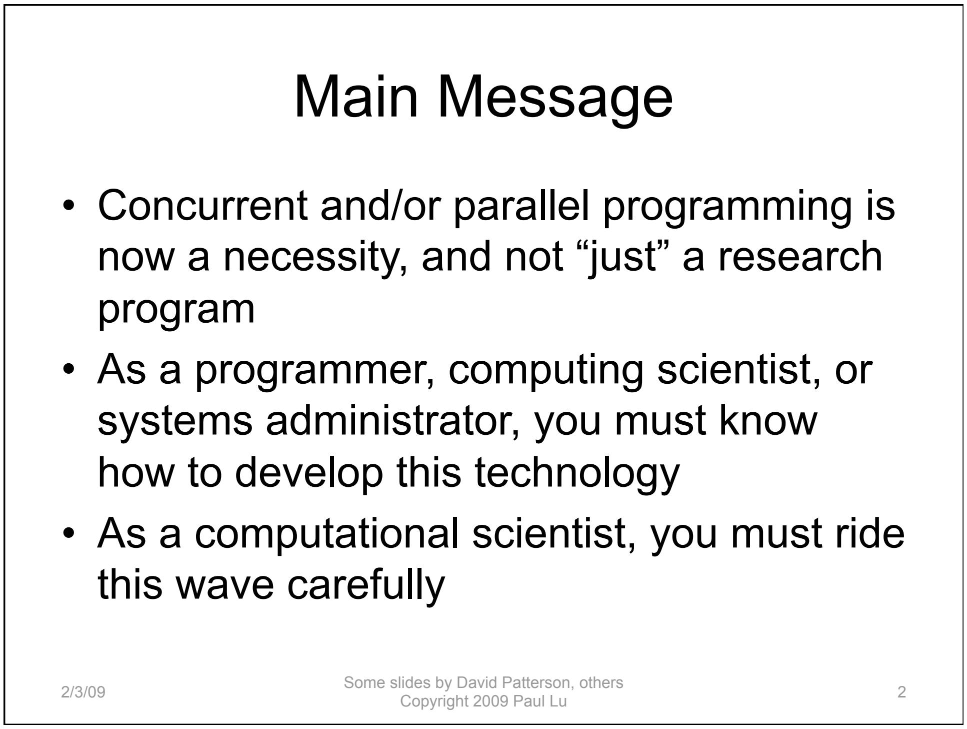


# Landscape of Parallel Computing: Manycores

Paul Lu
Associate Professor
Dept. of Computing Science
University of Alberta
Edmonton, Alberta, Canada
paullu@cs.ualberta.ca

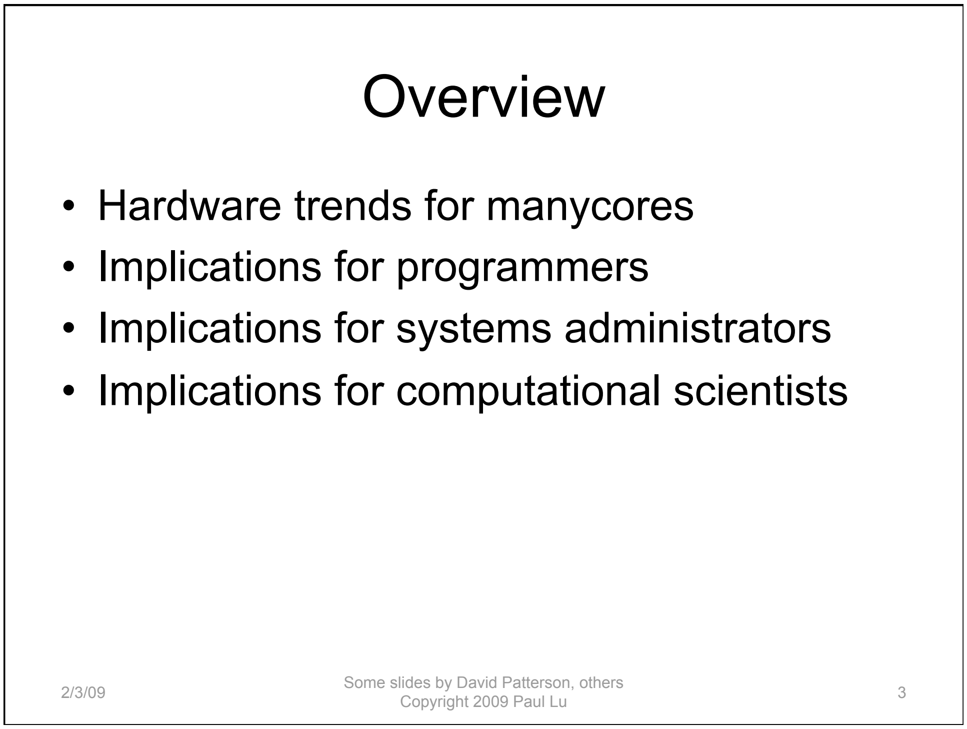
Some slides by David Patterson, others



## Main Message

- Concurrent and/or parallel programming is now a necessity, and not “just” a research program
- As a programmer, computing scientist, or systems administrator, you must know how to develop this technology
- As a computational scientist, you must ride this wave carefully

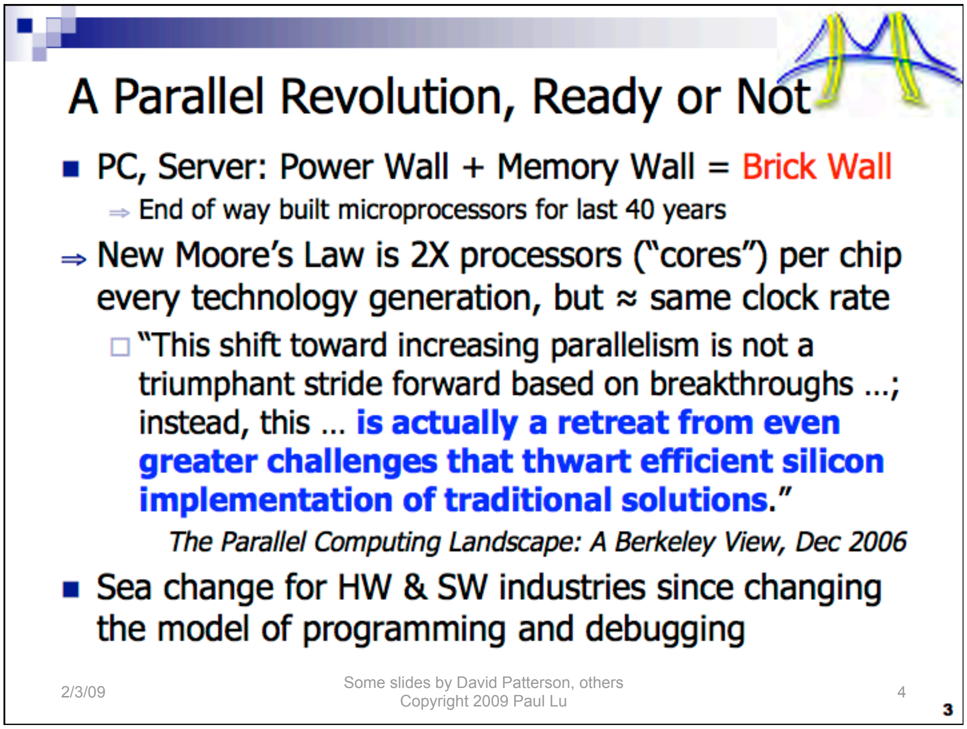
Some slides by David Patterson, others



## Overview

- Hardware trends for manycores
- Implications for programmers
- Implications for systems administrators
- Implications for computational scientists

2/3/09 Some slides by David Patterson, others Copyright 2009 Paul Lu 3



## A Parallel Revolution, Ready or Not

- PC, Server: Power Wall + Memory Wall = Brick Wall
  - ⇒ End of way built microprocessors for last 40 years

⇒ New Moore's Law is 2X processors ("cores") per chip every technology generation, but ≈ same clock rate

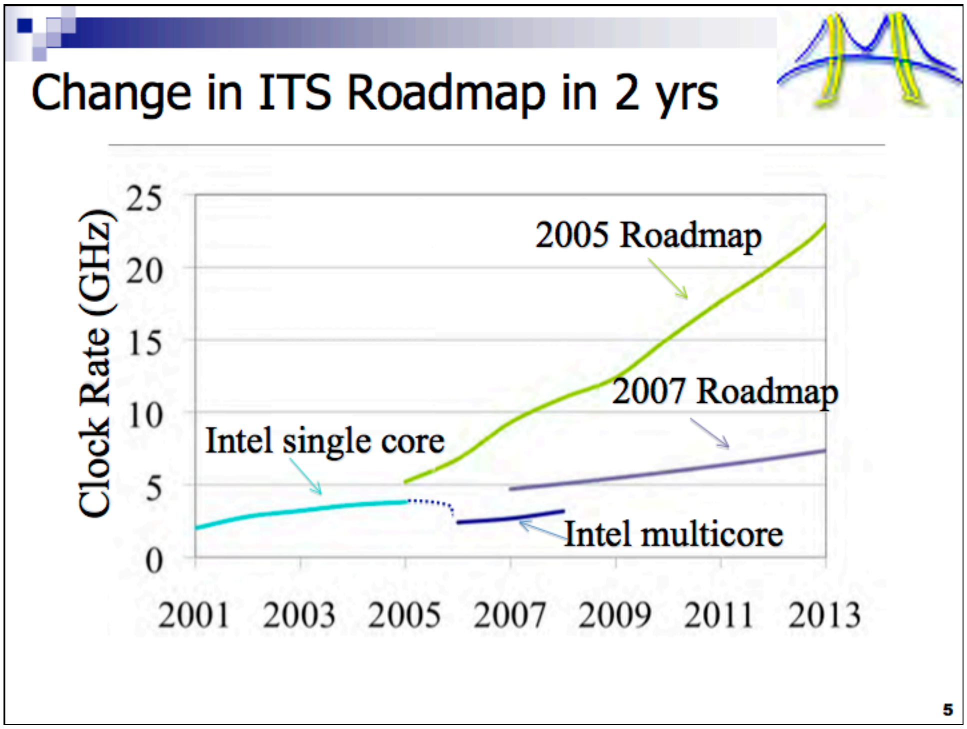
- □ "This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs …; instead, this … **is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional solutions**."

  *The Parallel Computing Landscape: A Berkeley View, Dec 2006*

- Sea change for HW & SW industries since changing the model of programming and debugging

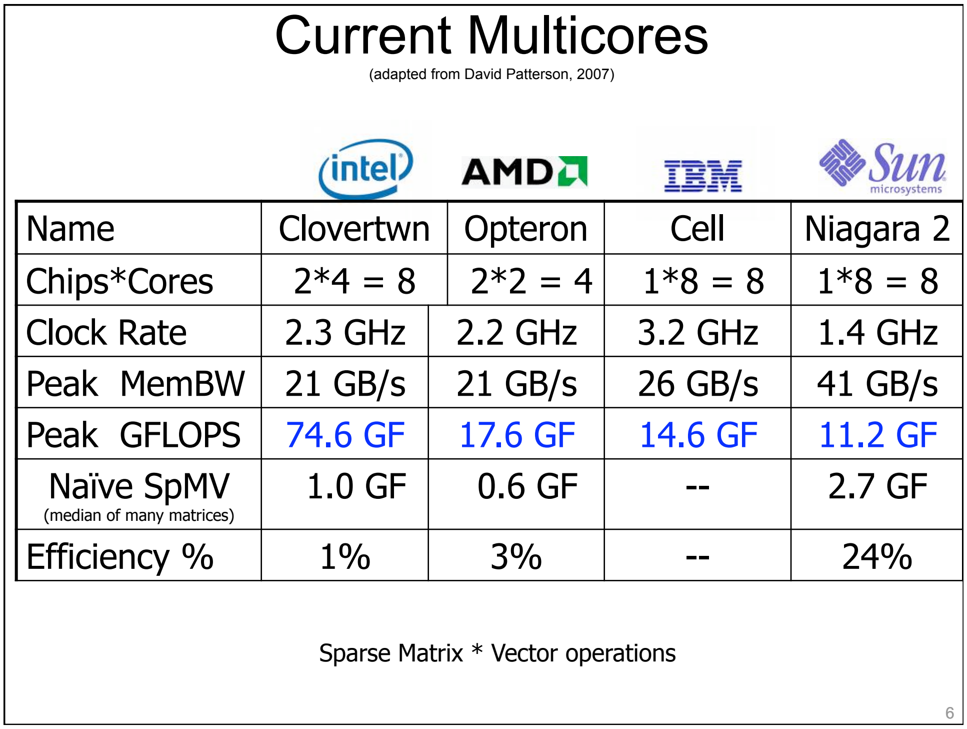
2/3/09 Some slides by David Patterson, others Copyright 2009 Paul Lu 4

3



## Change in ITS Roadmap in 2 yrs

5



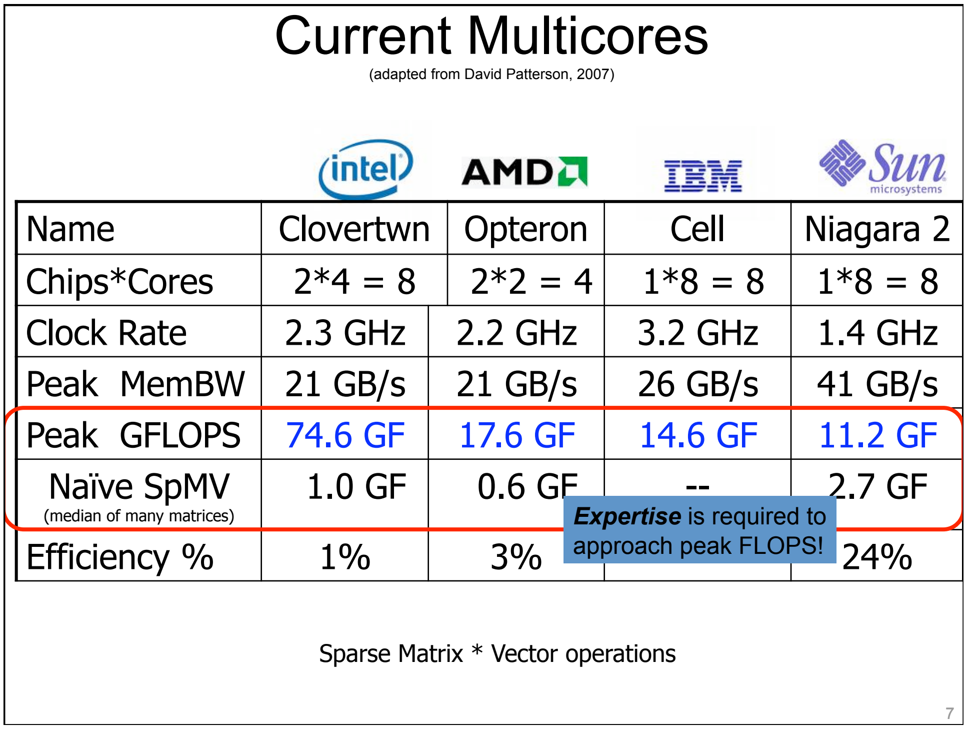
## Current Multicores

(adapted from David Patterson, 2007)

| | intel | AMD | IBM | Sun microsystems |
|---|---|---|---|---|
| Name | Clovertwn | Opteron | Cell | Niagara 2 |
| Chips*Cores | 2*4 = 8 | 2*2 = 4 | 1*8 = 8 | 1*8 = 8 |
| Clock Rate | 2.3 GHz | 2.2 GHz | 3.2 GHz | 1.4 GHz |
| Peak MemBW | 21 GB/s | 21 GB/s | 26 GB/s | 41 GB/s |
| Peak GFLOPS | 74.6 GF | 17.6 GF | 14.6 GF | 11.2 GF |
| Naïve SpMV<br>(median of many matrices) | 1.0 GF | 0.6 GF | -- | 2.7 GF |
| Efficiency % | 1% | 3% | -- | 24% |

Sparse Matrix * Vector operations

6



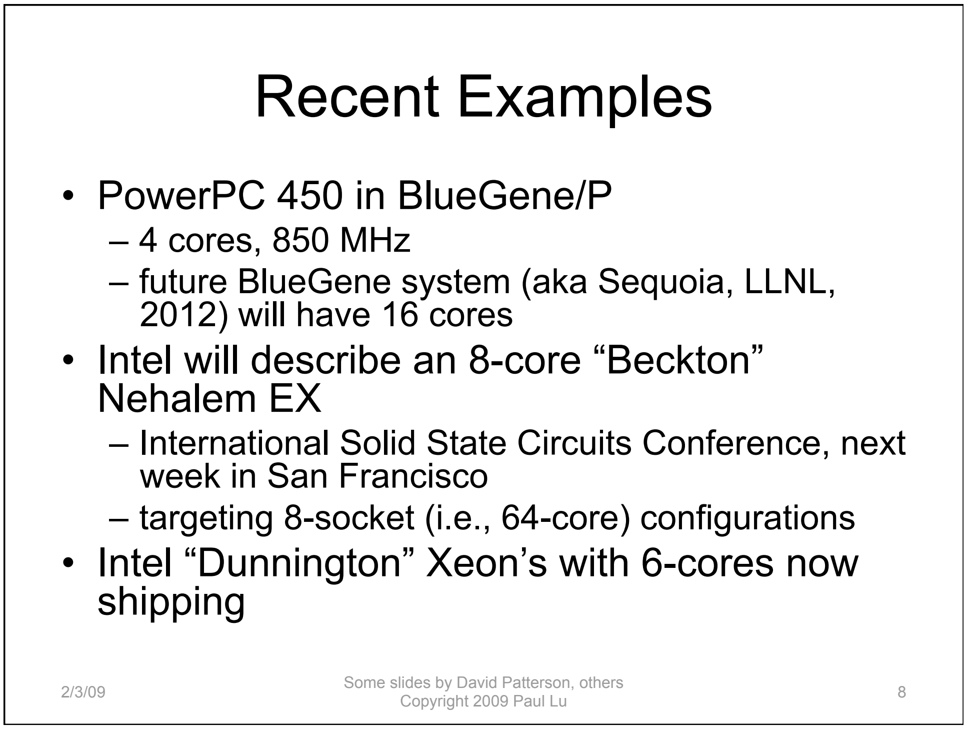
# Current Multicores

(adapted from David Patterson, 2007)

| | intel | AMD | IBM | Sun microsystems |
|---|---|---|---|---|
| Name | Clovertwn | Opteron | Cell | Niagara 2 |
| Chips*Cores | 2*4 = 8 | 2*2 = 4 | 1*8 = 8 | 1*8 = 8 |
| Clock Rate | 2.3 GHz | 2.2 GHz | 3.2 GHz | 1.4 GHz |
| Peak MemBW | 21 GB/s | 21 GB/s | 26 GB/s | 41 GB/s |
| Peak GFLOPS | 74.6 GF | 17.6 GF | 14.6 GF | 11.2 GF |
| Naïve SpMV<br>(median of many matrices) | 1.0 GF | 0.6 GF | -- | 2.7 GF |
| Efficiency % | 1% | 3% | | 24% |

***Expertise*** is required to approach peak FLOPS!

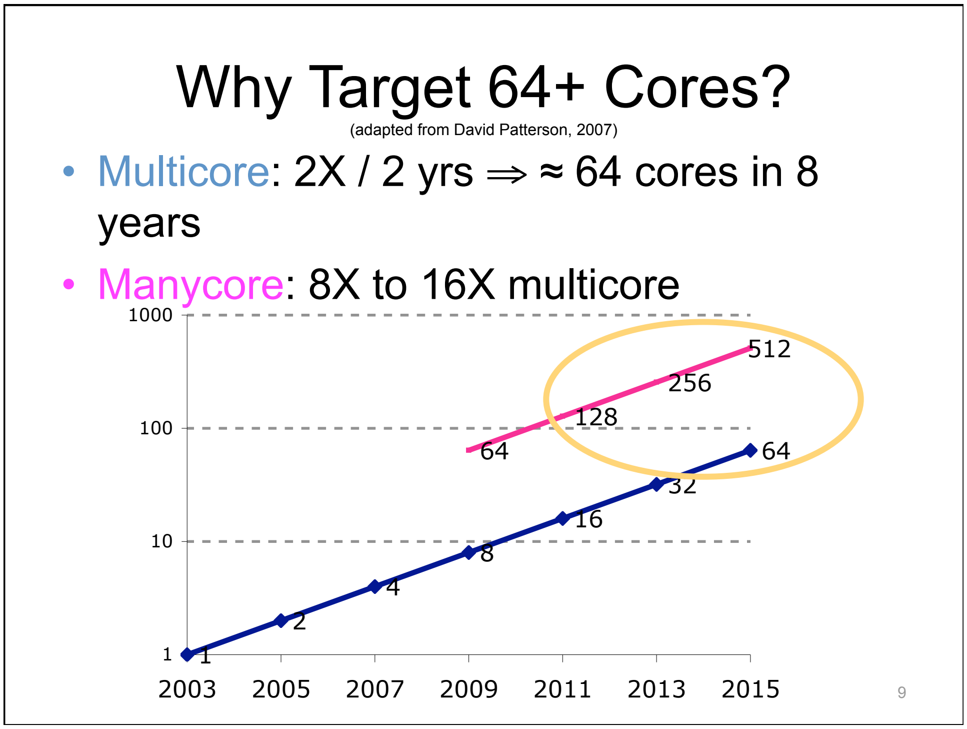
Sparse Matrix * Vector operations



# Recent Examples

- PowerPC 450 in BlueGene/P
  - 4 cores, 850 MHz
  - future BlueGene system (aka Sequoia, LLNL, 2012) will have 16 cores
- Intel will describe an 8-core “Beckton” Nehalem EX
  - International Solid State Circuits Conference, next week in San Francisco
  - targeting 8-socket (i.e., 64-core) configurations
- Intel “Dunnington” Xeon’s with 6-cores now shipping

Some slides by David Patterson, others

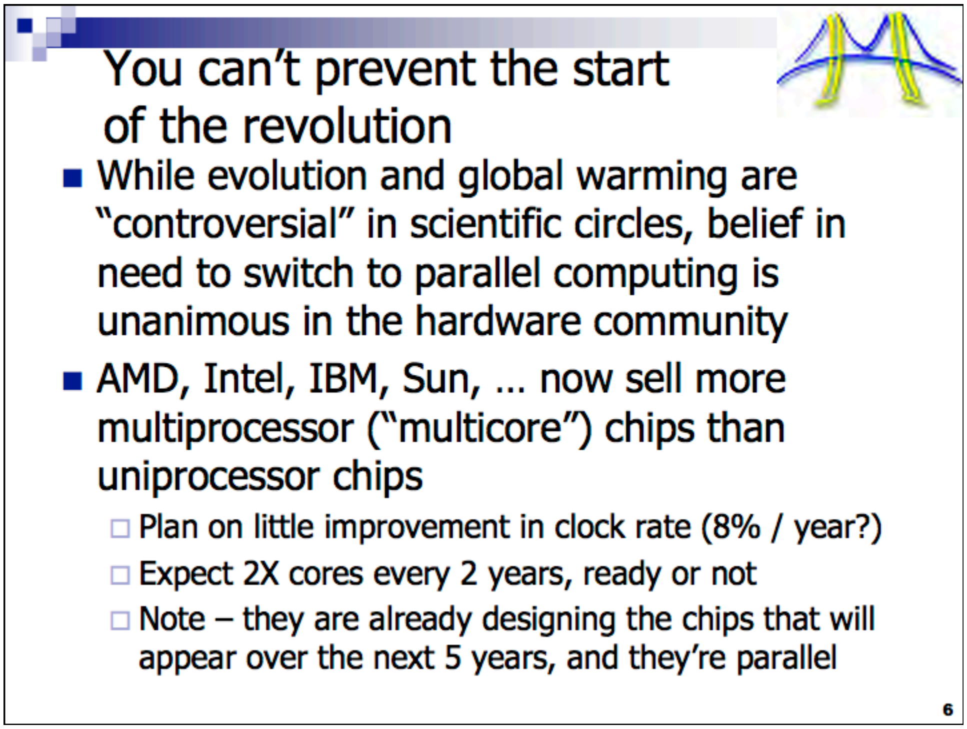


# Why Target 64+ Cores?

(adapted from David Patterson, 2007)

- Multicore: 2X / 2 yrs ⇒ ≈ 64 cores in 8 years
- Manycore: 8X to 16X multicore

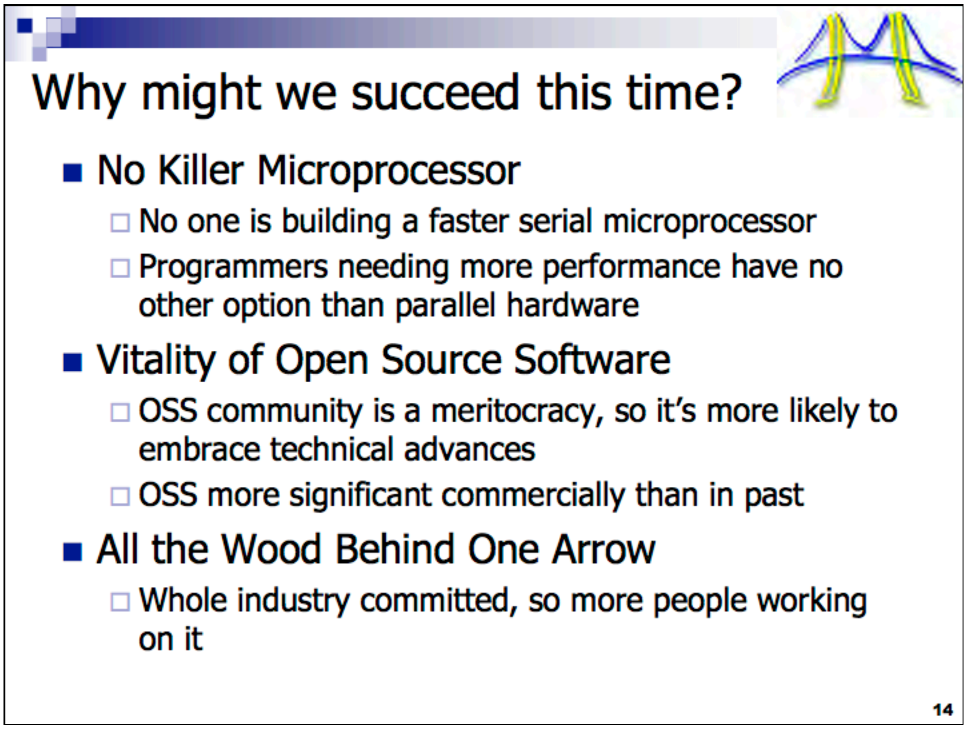
9



# You can't prevent the start of the revolution

- While evolution and global warming are "controversial" in scientific circles, belief in need to switch to parallel computing is unanimous in the hardware community
- AMD, Intel, IBM, Sun, … now sell more multiprocessor ("multicore") chips than uniprocessor chips
  - □ Plan on little improvement in clock rate (8% / year?)
  - □ Expect 2X cores every 2 years, ready or not
  - □ Note – they are already designing the chips that will appear over the next 5 years, and they're parallel

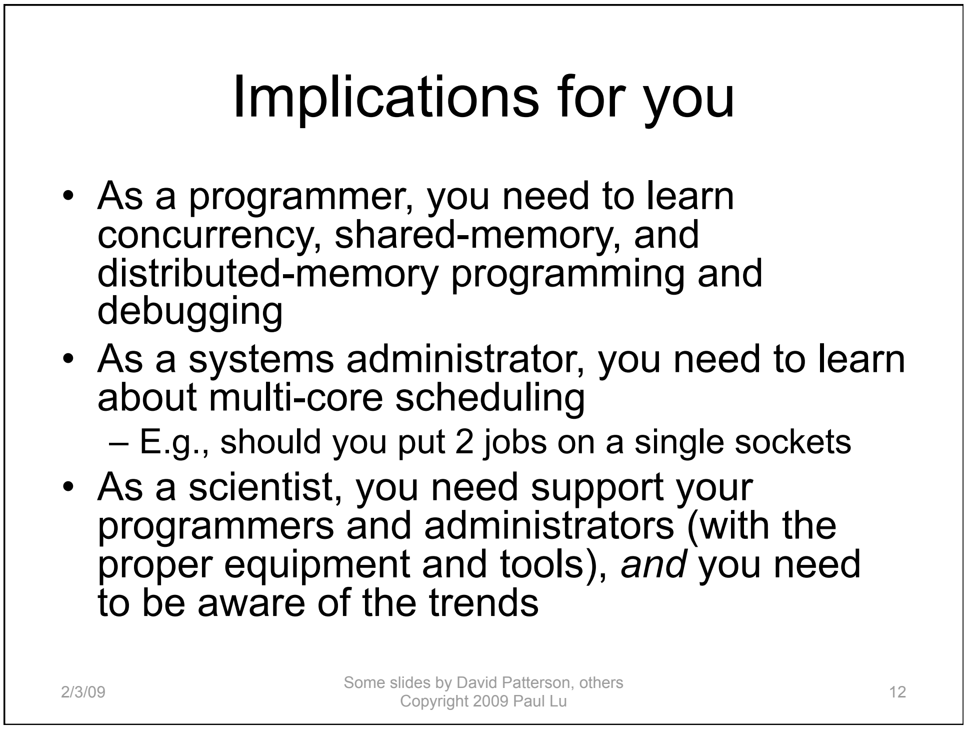
6



## Why might we succeed this time?

- No Killer Microprocessor
  - No one is building a faster serial microprocessor
  - Programmers needing more performance have no other option than parallel hardware
- Vitality of Open Source Software
  - OSS community is a meritocracy, so it's more likely to embrace technical advances
  - OSS more significant commercially than in past
- All the Wood Behind One Arrow
  - Whole industry committed, so more people working on it

14



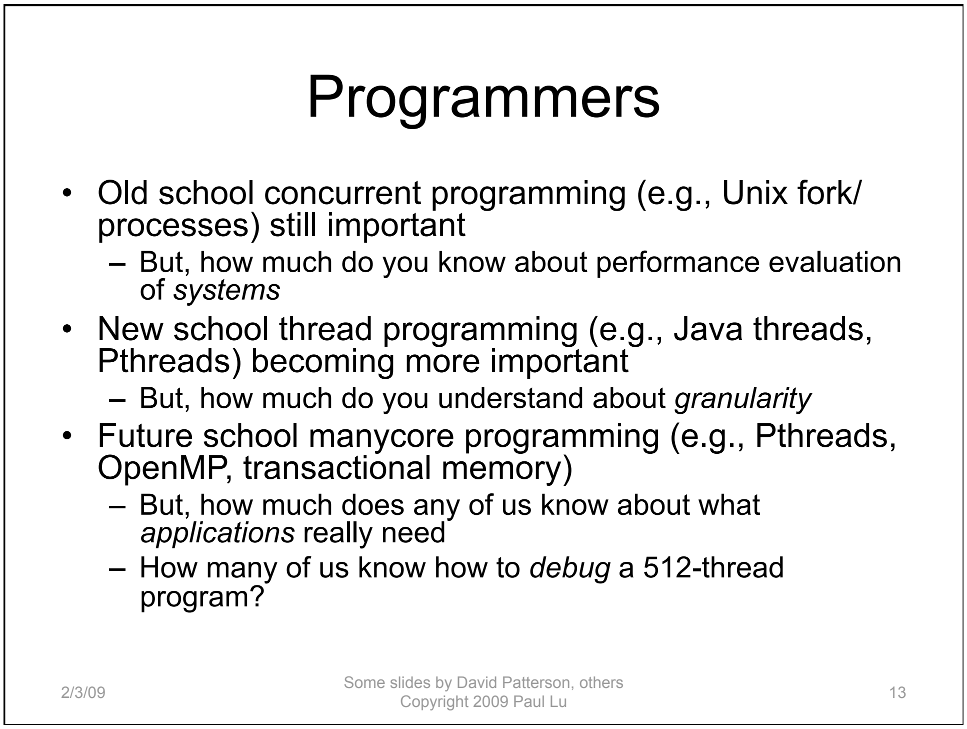
## Implications for you

- As a programmer, you need to learn concurrency, shared-memory, and distributed-memory programming and debugging
- As a systems administrator, you need to learn about multi-core scheduling
  - E.g., should you put 2 jobs on a single sockets
- As a scientist, you need support your programmers and administrators (with the proper equipment and tools), *and* you need to be aware of the trends

2/3/09

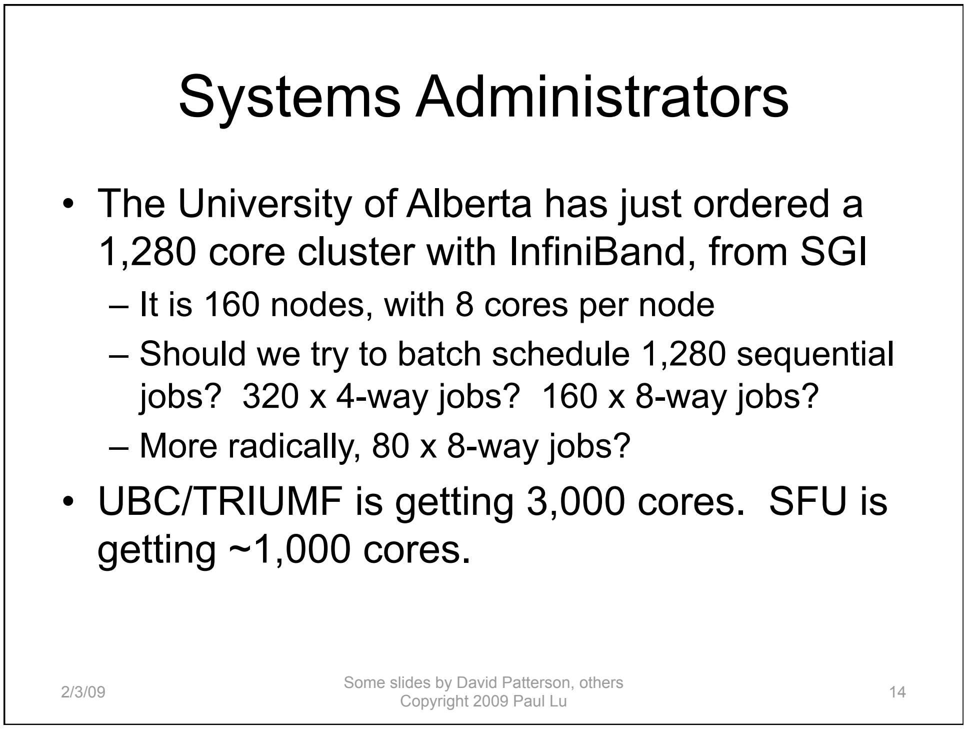
Some slides by David Patterson, others
Copyright 2009 Paul Lu

12



# Programmers

- Old school concurrent programming (e.g., Unix fork/ processes) still important
  - But, how much do you know about performance evaluation of *systems*
- New school thread programming (e.g., Java threads, Pthreads) becoming more important
  - But, how much do you understand about *granularity*
- Future school manycore programming (e.g., Pthreads, OpenMP, transactional memory)
  - But, how much does any of us know about what *applications* really need
  - How many of us know how to *debug* a 512-thread program?



# Systems Administrators

- The University of Alberta has just ordered a 1,280 core cluster with InfiniBand, from SGI
  - It is 160 nodes, with 8 cores per node
  - Should we try to batch schedule 1,280 sequential jobs? 320 x 4-way jobs? 160 x 8-way jobs?
  - More radically, 80 x 8-way jobs?
- UBC/TRIUMF is getting 3,000 cores. SFU is getting ~1,000 cores.

# Computational Scientists

- Good news: Many applications already parallelized for threads or MPI
- Bad news: Who’s going to (re-)write the next generation of parallel application?
  - Who’s going to help the computing scientists write the next generation of software?
  - Is a delay of a few years for optimized applications going to affect your research?

# Concluding Remarks

- Hardware trends imply that concurrent and/or parallel programming is now a necessity
- There are many computing science problems to be solved in the multicore/ manycore era.
- As a computational scientist, you must ride this wave carefully: medium-term and long-term planning will help *your* research