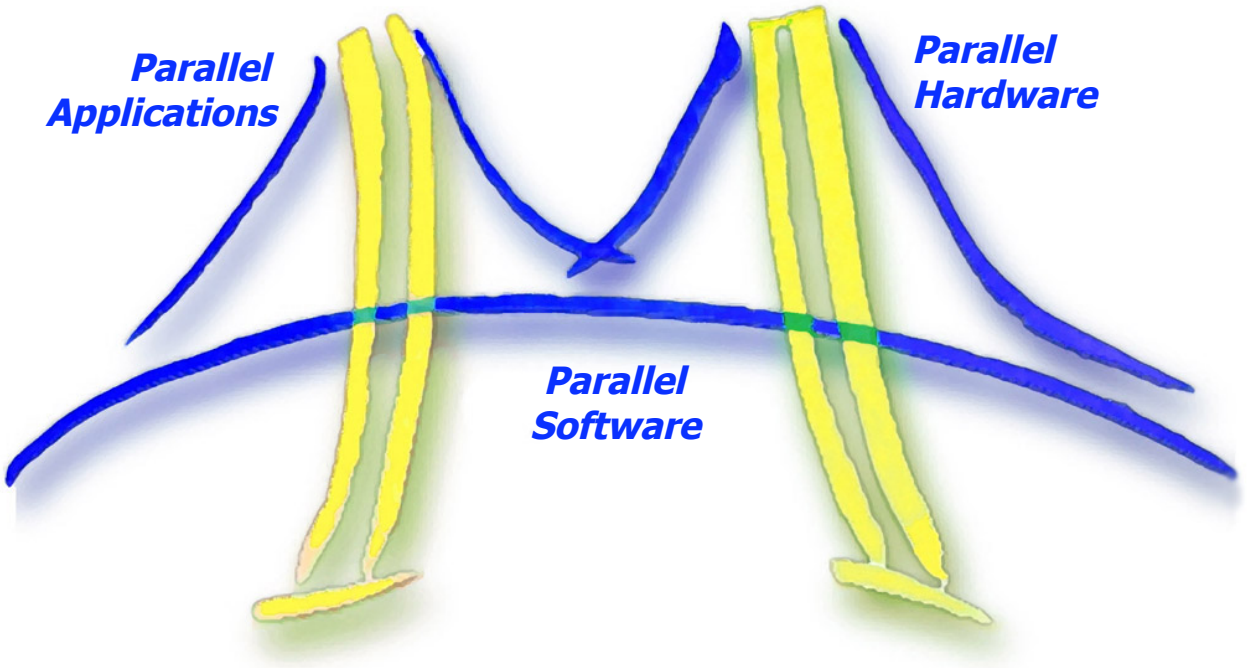



# The Parallel Computing Challenge

Dave Patterson
UC Berkeley

March, 2008

# 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
  - New "Moore's Law" is 2X processors per chip every 2 years
  - Duo core, Quad core, …
- Goal: Productive, Efficient, Correct Programming of 100+ cores & scale as double cores every 2 years (!)

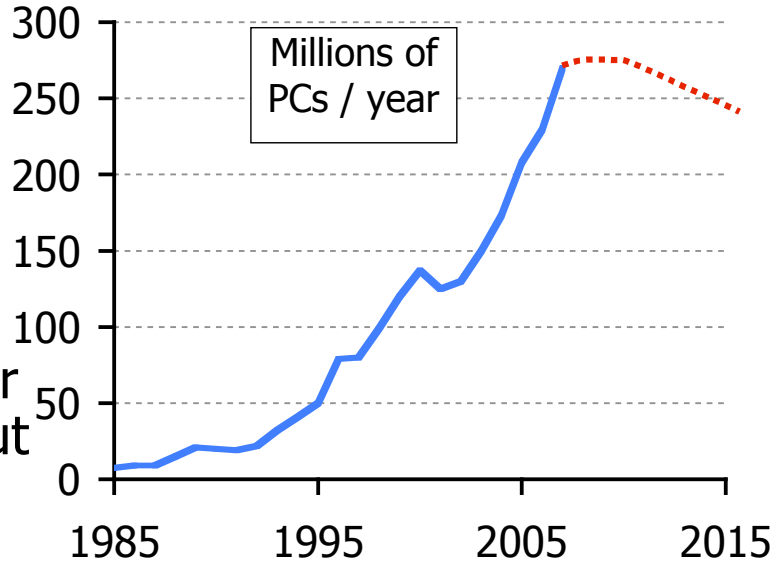
# P.S. Parallel Revolution Likely to Fail

- 100% failure rate of Parallel Computer Companies from 1970s, 1980s, 1990s, …
  - Convex, Encore, Inmos (Transputer), MasPar, NCUBE, Kendall Square Research, Sequent, (Silicon Graphics), Thinking Machines, …
- John Hennessy, President, Stanford University:
  *"…when we start talking about parallelism and ease of use of truly parallel computers, we're talking about a problem that's as hard as any that computer science has faced. … I would be panicked if I were in industry."*

"A Conversation with Hennessy & Patterson," *ACM Queue Magazine*, 4:10, 1/07.

# Suppose software stop getting faster

- What if IT goes from a *growth* industry to a *replacement* industry?
  - If SW can't effectively use 32, 64, ... cores per chip
    ⇒ SW no faster on new computer
    ⇒ Only buy if computer wears out




- Impact on US economy if end of "Moore's Law"?
  - How much productivity tied to IT?
  - How much IT tied to faster computers?
- Opportunity to lose US lead in IT if others solve the problem
  - If someone in China invents a Mandarin-based programming language that solves the parallel computing problem, then I'll need to learn Mandarin

# How to succeed at the hardest problem to face computer science?

- Recruit the best minds to help
- Academic & industrial research
  - Led to 19 multibillion dollar IT industries
- "Pain killers sell; vitamins don't"
  - Try to restart federal funding?
  - Joint with industry?

# Reasons for Optimism towards Parallel Challenge this time

- End of sequential microprocessor/faster clock rates
  - No looming sequential juggernaut to kill parallel revolution
- SW & HW industries fully committed to parallelism
  - End of La-Z-Boy Programming Era
- Open Source Software movement means that SW stack can evolve more quickly than in past
- Field Programmable Gate Arrays as hardware prototype to ramp up parallel research vs. building custom chips (RAMP)
- Moore's Law continues, so soon can put 1000s of simple cores on an economical chip
- Communication between cores within a chip at very low latency and very high bandwidth
  - Processor-to-Processor fast even if Memory slow
- All cores equal distance to shared main memory
  - Fewer data distribution challenges for software to get performance