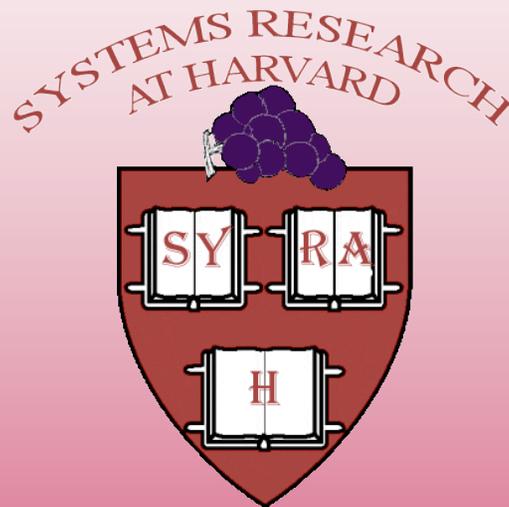


What's Happening in Hardware and how it affects Software



October 15, 2007

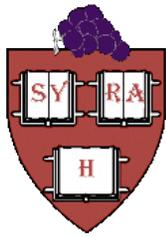
Margo Seltzer

School of Engineering and Applied Sciences

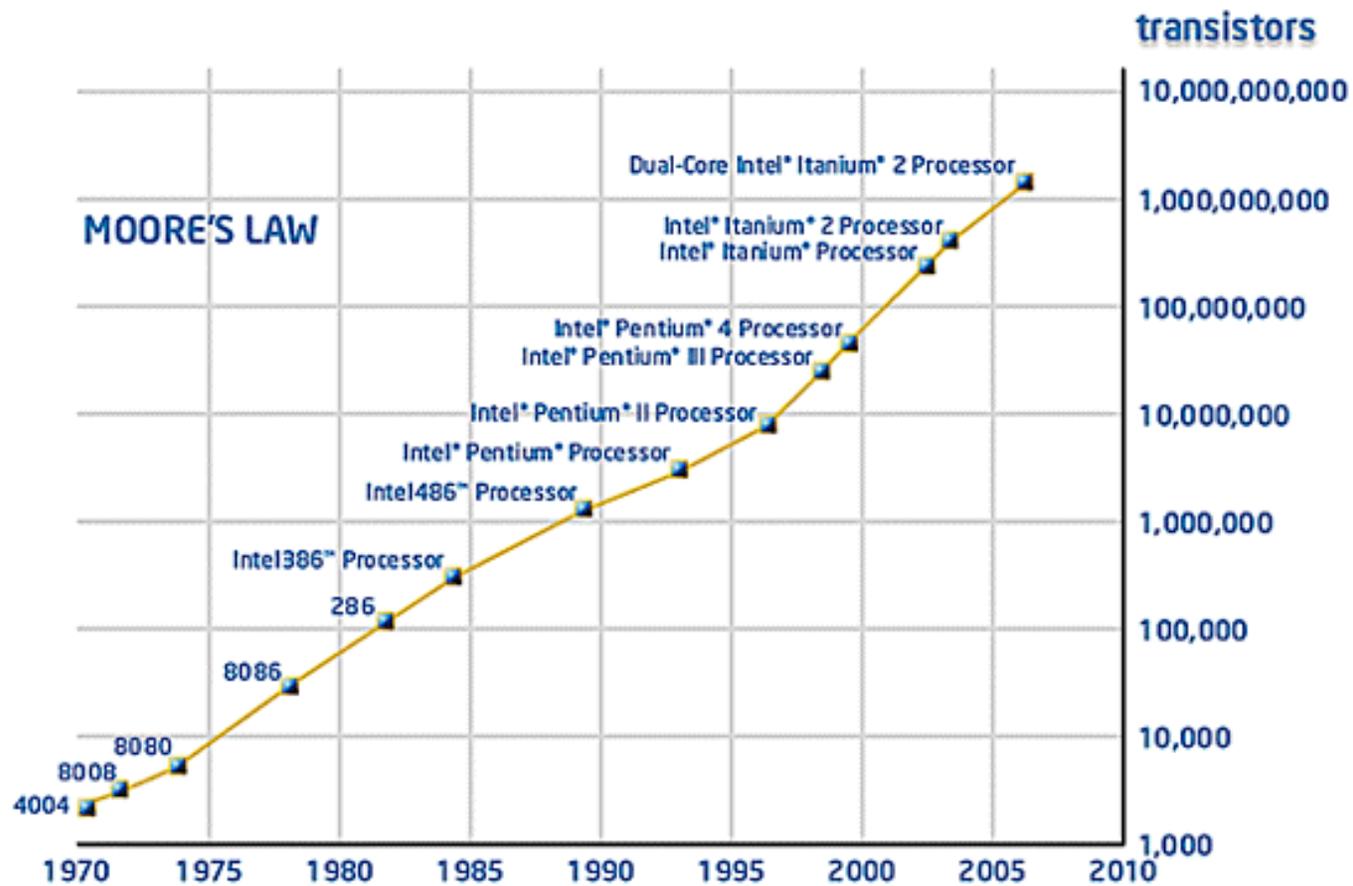


Moore's Law

- 1965: Gordon Moore, CEO Intel observes:
 - The number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every two years.



Transistors per Year



Wikipedia: Moore's Law

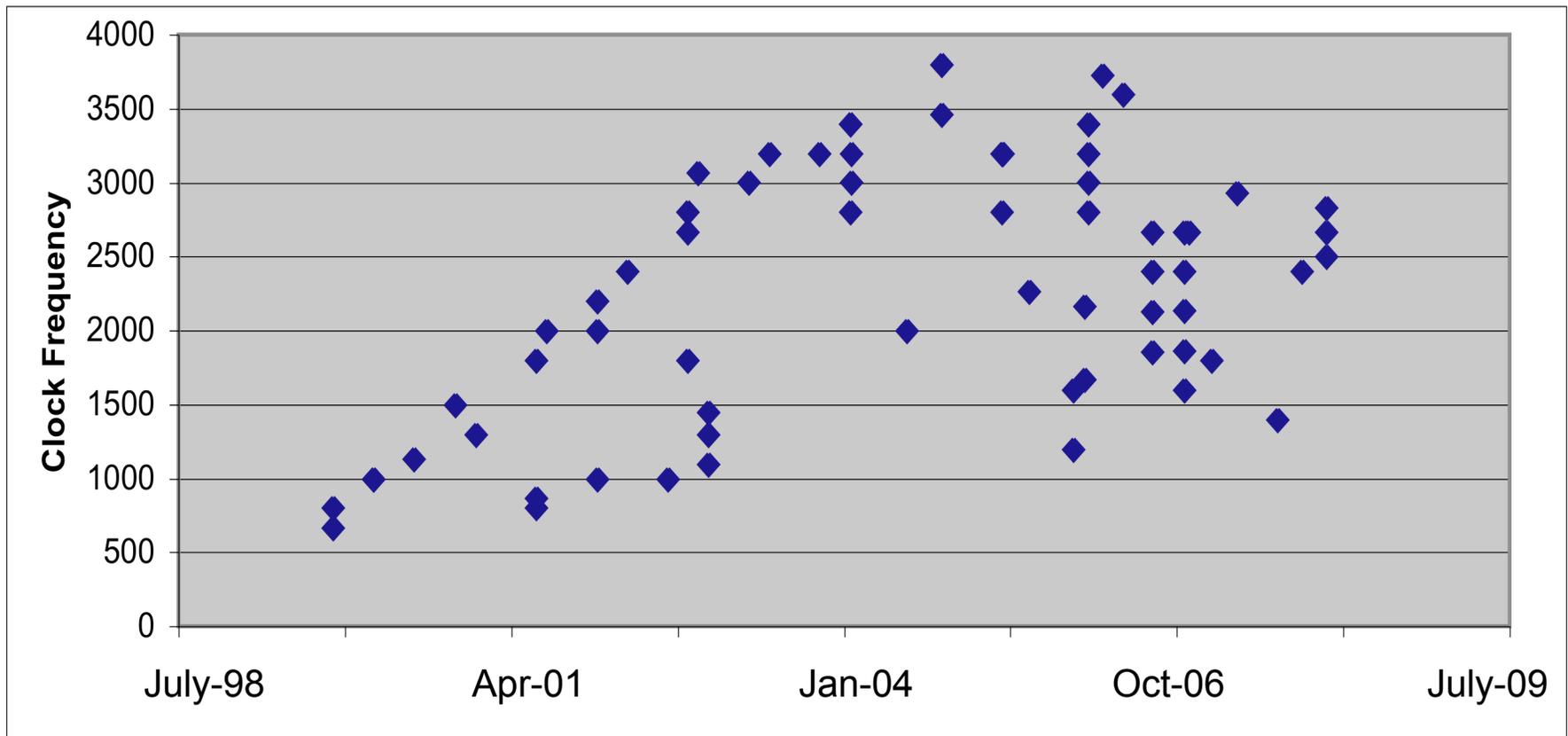


So What?

- More transistors => Smaller transistors
- Smaller transistors => Shorter paths
- Shorter paths => Greater speed
- More than just the speed of light:
 - Pipelining
 - Increased caches
 - Out of order execution
 - Parallelism

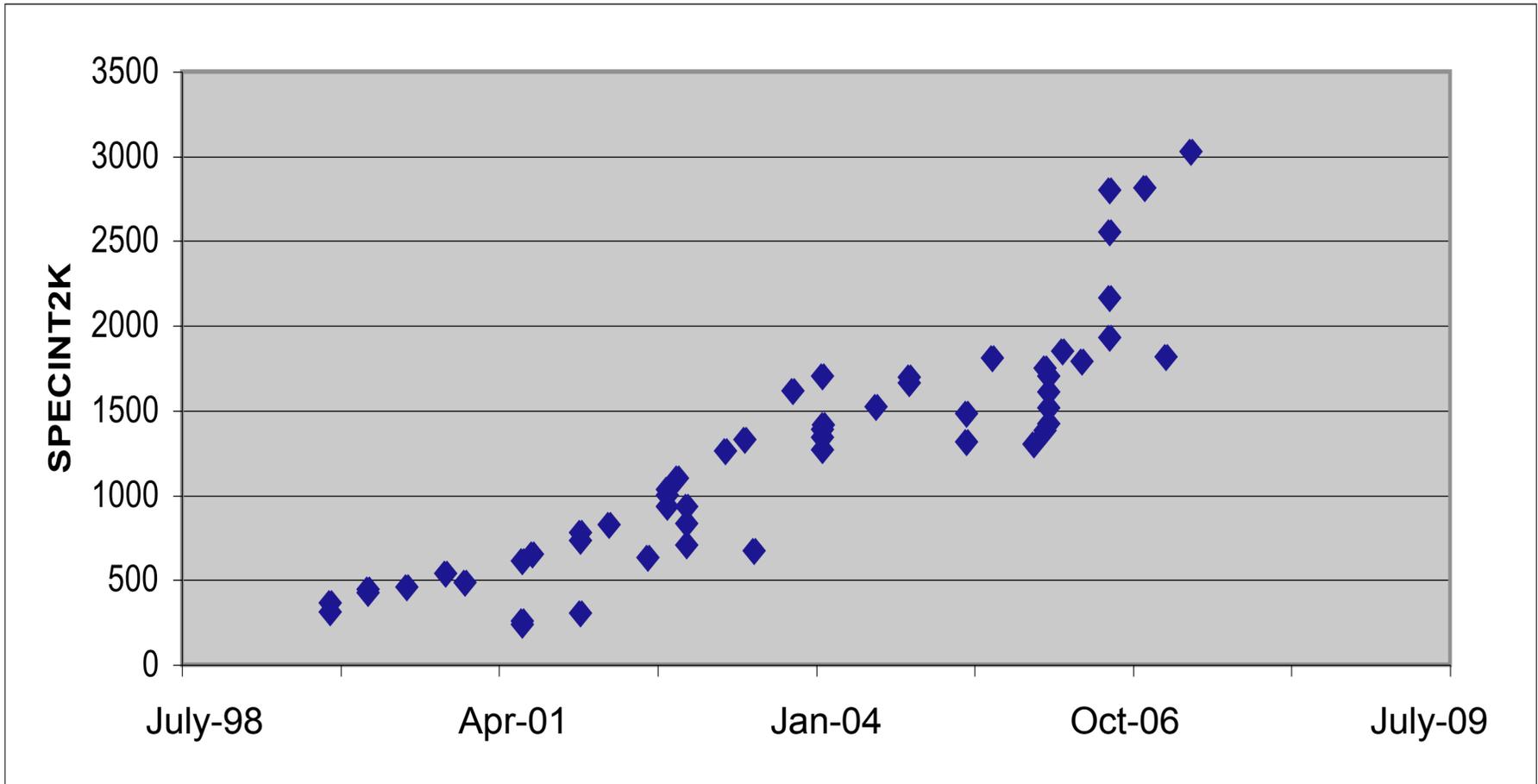


Clock Frequency vs Year





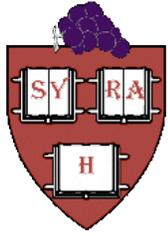
Performance by Year



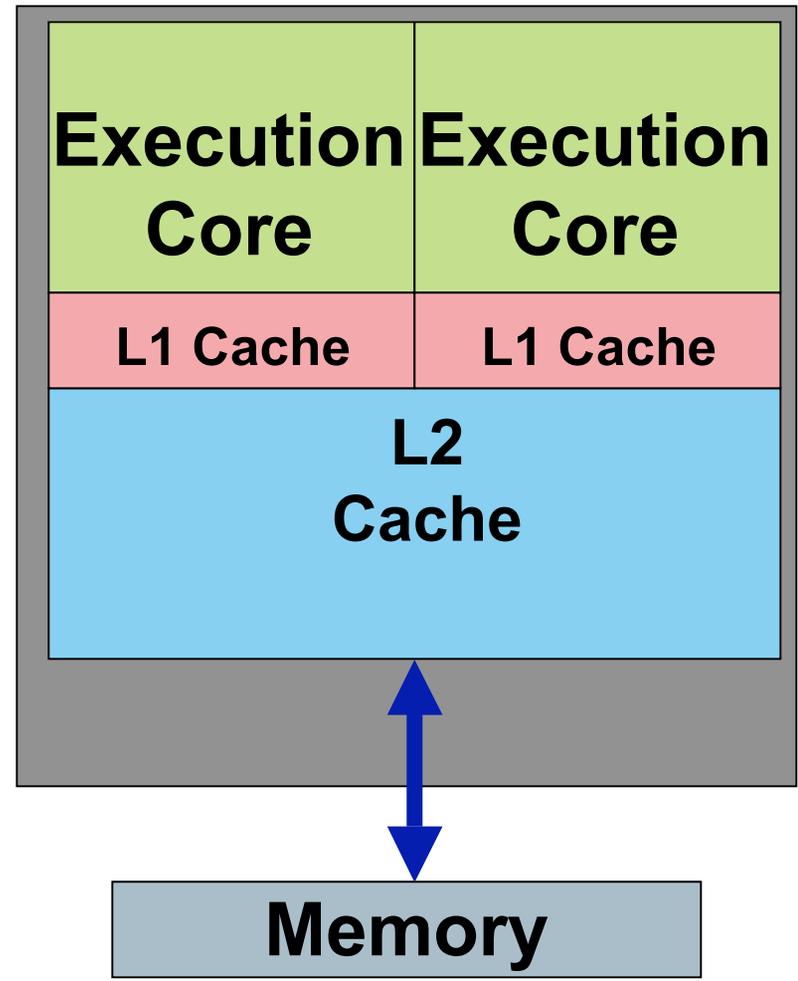
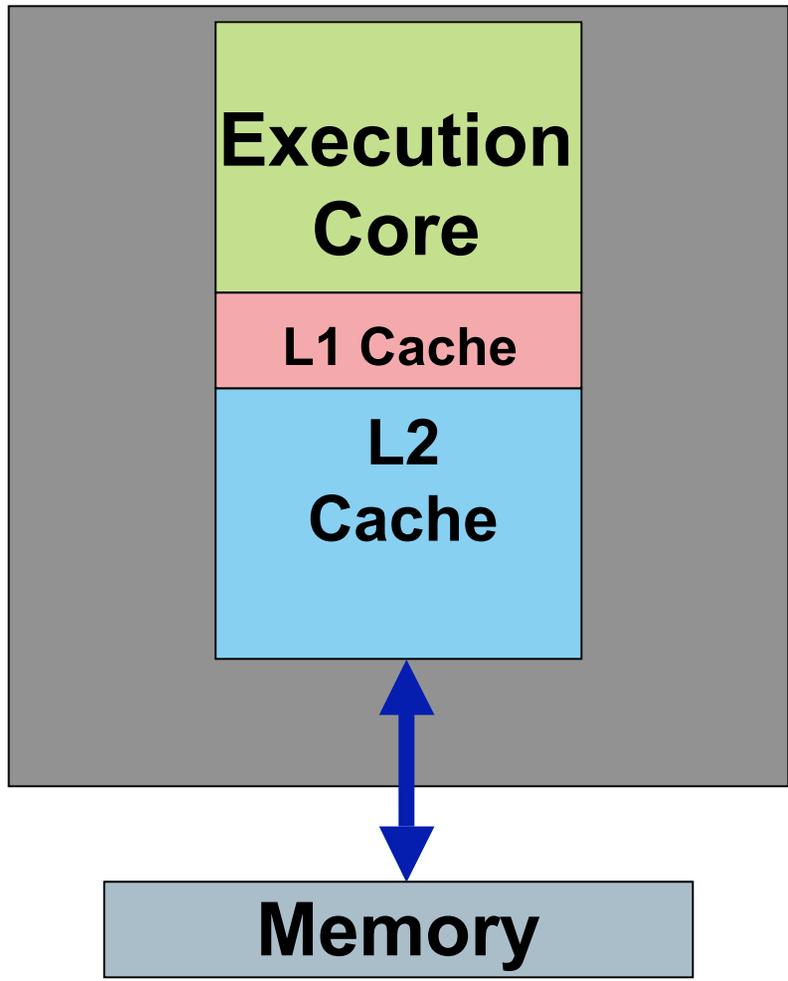


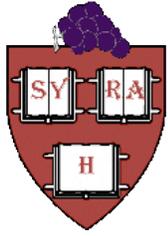
What to do with Transistors

- Performance seems to be leveling off.
- So, what do you do with your extra transistors?

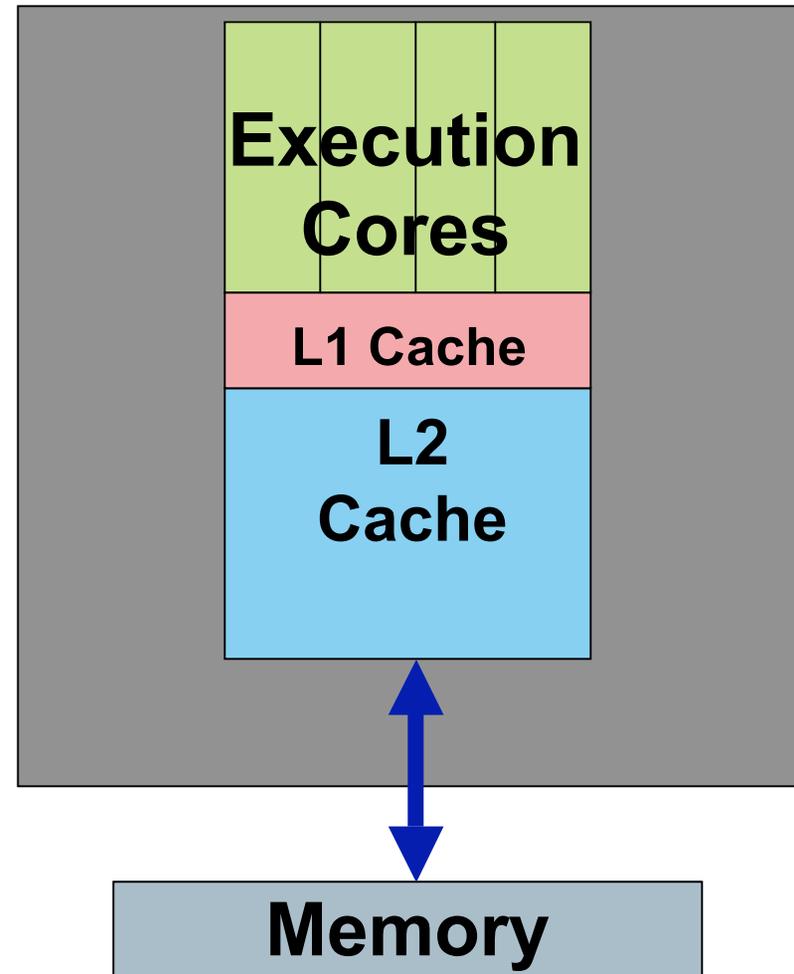
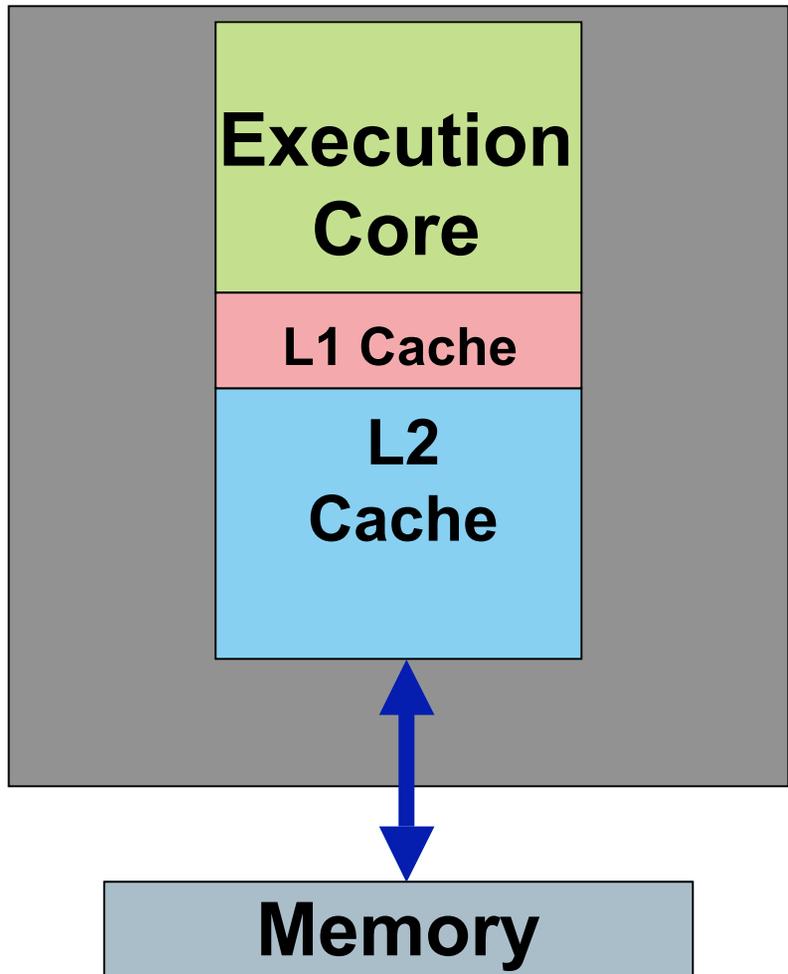


Multicore



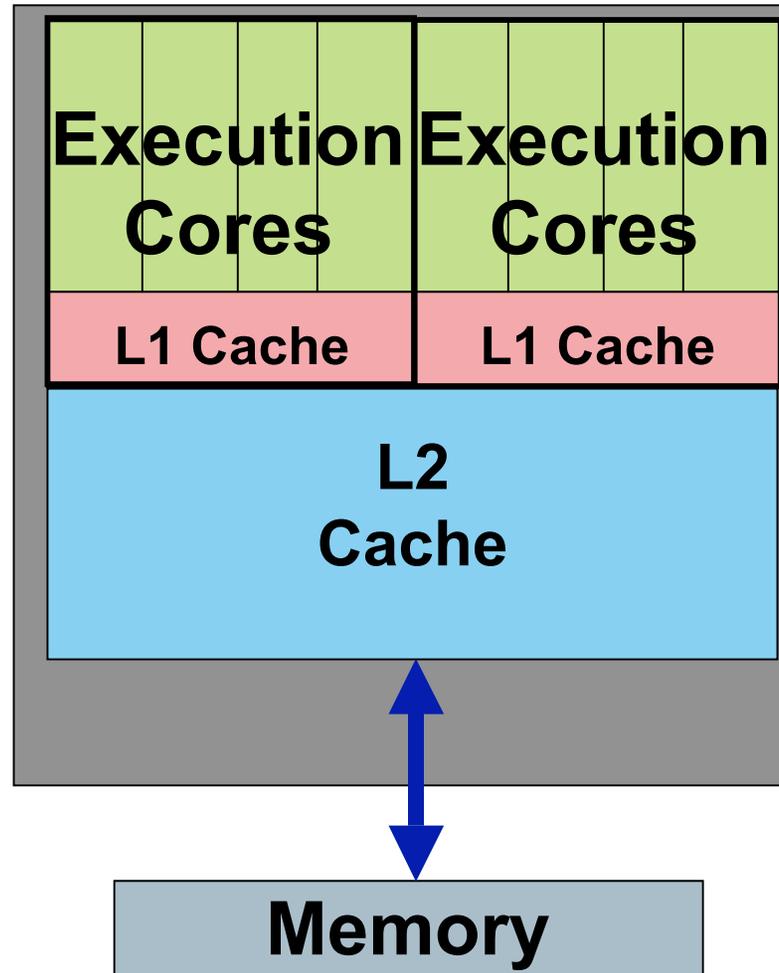


Multithread (Hyperthreaded)



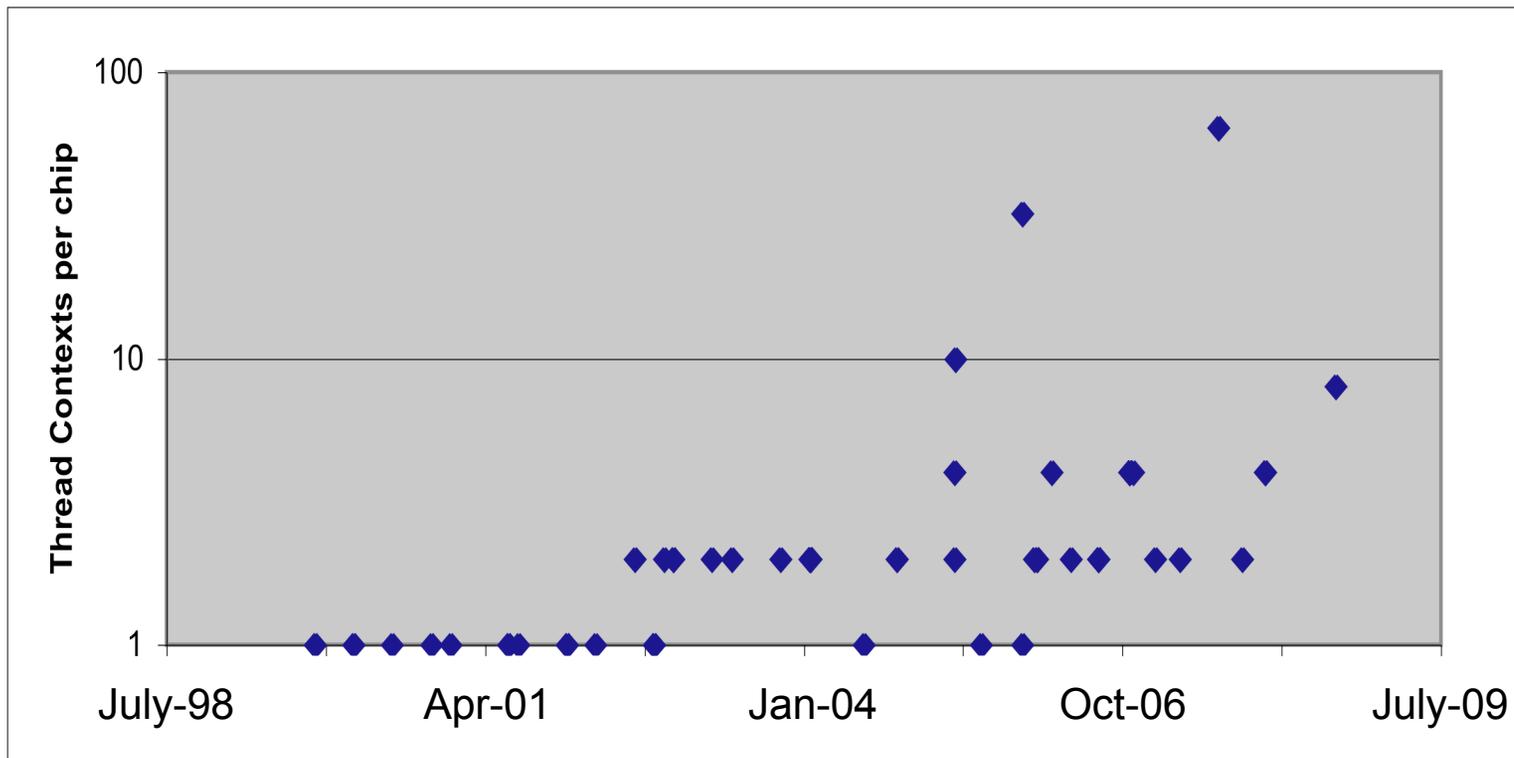


Multithread/Multicore





How many Cores?





Machines of the Future

- 10 cores today
 - 10's of cores tomorrow
 - 1000's of cores before we know it
-
- How do you program 1000's of cores?



Software

- Historically, software you ran got faster with each new generation of processor.
- But, single-thread performance isn't getting any faster.
- Why buy new hardware?
 - Do more things at once?
 - Use better software that uses multiple cores?



More things at once?

- How many things can you do?
 - Email
 - Browsing
 - iTunes
 - Movies
 - Compiling
 - Writing a paper
 - IM-ing
 - Modifying your Facebook page
 - Shopping
 - ...
 - And 991 more things ...



Exploit Parallelism?

- We must make software more capable of taking advantage of parallelism.
- Historically, this has been very, very hard.
 - Limited success in scientific computing
 - Some parallelism in today's applications (e.g., spell checking, etc).



What Does this Mean for you?

- Nothing: you won't be expected to write concurrent programs in CS50.
- Everything: the world is changing
 - You can lead the charge or be run over by it.
 - Think creatively about how you solve problems.
 - Break big tasks into tiny tasks that can be in parallel.
 - Learn new languages (and figure out for what they are useful).
 - Pay attention to technology.
 - Take more CS courses!