Final

CSE 30321 Computer Architecture I							
Niemier, Michael Thaddeus	01	52	12/13/2010	10:30 AM	12:30 PM	127	Hayes-Healy Center

If you haven't done so already, do your CIF

Final is open book, open notes

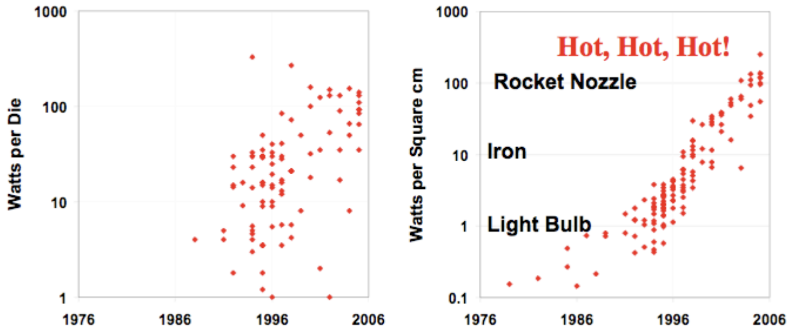
What to take away from last 4-5 lectures

Historical sources of performance gain

- Pipelining
- Technology Scaling
- Memory Hierarchies
- Hardware support for threads
- …
- All the easy stuff done
  - (And to some extent being taken away!)

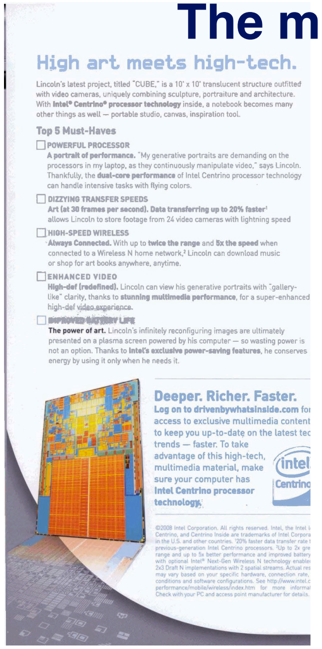
Technology drive to multi-core computing

Why the clock flattening? POWER!!!!



The multi-core solution:

Fundamentally, built on datapath, memory hierarchy discussed in class.



High art meets high-tech.

Lincoln's latest project, titled "CUBE," is a 10' x 10' translucent structure outfitted with video cameras, uniquely combining sculpture, portraiture and architecture. With **Intel® Centrino® processor technology** inside, a notebook becomes many other things as well — portable studio, canvas, inspiration tool.

Top 5 Must-Haves

□ **POWERFUL PROCESSOR**
A portrait of performance. "My generative portraits are demanding on the processors in my laptop, as they continuously manipulate video," says Lincoln. Thankfully, the **dual-core performance** of Intel Centrino processor technology can handle intensive tasks with flying colors.

□ **DIZZYING TRANSFER SPEEDS**
Art (at 30 frames per second). Data transferring up to 20% faster[1] allows Lincoln to store footage from 24 video cameras with lightning speed

□ **HIGH-SPEED WIRELESS**
Always Connected. With up to **twice the range** and **5x the speed** when connected to a Wireless N home network,[2] Lincoln can download music or shop for art books anywhere, anytime.

□ **ENHANCED VIDEO**
High-def (redefined). Lincoln can view his generative portraits with "gallery-like" clarity, thanks to **stunning multimedia performance**, for a super-enhanced high-def video experience.

□ **IMPROVED BATTERY LIFE**
The power of art. Lincoln's infinitely reconfiguring images are ultimately presented on a plasma screen powered by his computer — so wasting power is not an option. Thanks to **Intel's exclusive power-saving features**, he conserves energy by using it only when he needs it.

Deeper. Richer. Faster.
Log on to drivenbywhatsinside.com for access to exclusive multimedia content to keep you up-to-date on the latest tec trends — faster. To take advantage of this high-tech, multimedia material, make sure your computer has **Intel Centrino processor technology.**

©2008 Intel Corporation. All rights reserved. Intel, the Intel l Centrino, and Centrino Inside are trademarks of Intel Corpora in the U.S. and other countries. [1]20% faster data transfer rate t previous-generation Intel Centrino processors. [2]Up to 2x gre range and up to 5x better performance and improved battery with optional Intel® Next-Gen Wireless N technology enable 2x3 Draft N implementations with 2 spatial streams. Actual res may vary based on your specific hardware, connection rate, conditions and software configurations. See http://www.intel.c performance/mobile/wireless/index.htm for more informat Check with your PC and access point manufacturer for details.

- Processor complexity is good enough
- Transistor sizes can still scale
- Slow processors down to manage power
- Get performance from...

Parallelism

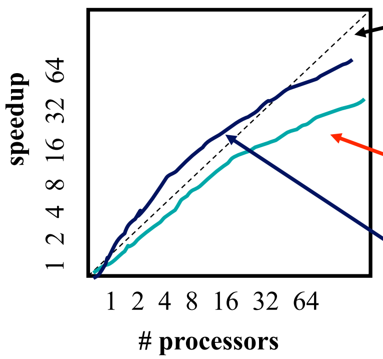
Top 5 Must-Haves

□ **POWERFUL PROCESSOR**
A portrait of performance. "My generative portraits are demanding on the processors in my laptop, as they continuously manipulate video," says Lincoln. Thankfully, the **dual-core performance** of Intel Centrino processor technology can handle intensive tasks with flying colors.

(i.e. 1 processor, 1 ns clock cycle
vs.
2 processors, 2 ns clock cycle)

Would like to do deliver speedups equal to the number of cores

- Time(1) / Time(P) for P processors
  - note: must use the best ***sequential*** algorithm for Time(1); the parallel algorithm may be different.

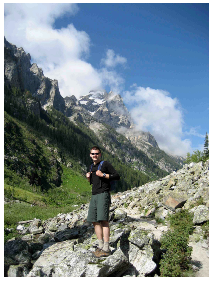


"linear" speedup (ideal)

typical: rolls off w/some # of processors

occasionally see "superlinear"... why?

Could program P processors as if your off for a nice pleasant hike…



But you should always be "bear aware"…

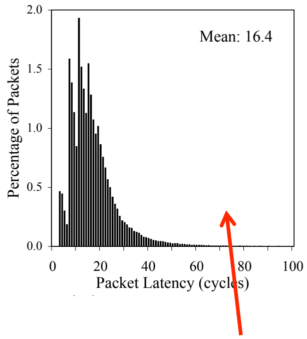
BE BEAR AWARE
Don't attract BEARS with food.
Keep sites clean and odorless.
Store all food in secure place.
A FED BEAR IS A DEAD BEAR

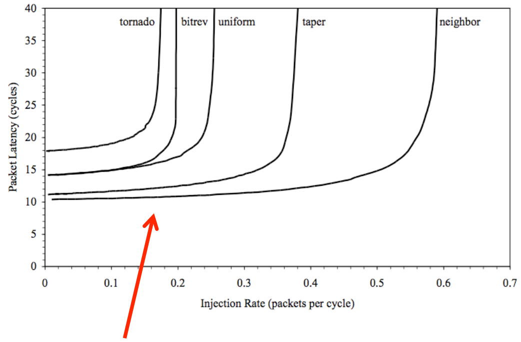
What are "the bears" ???

Impediments to Parallel Performance

(*independent of problem/algorithm itself*)

- Contention for access to shared resources
  - i.e. multiple accesses to limited # of memory banks or shared cache may dominate system scalability
  - The interconnection network itself



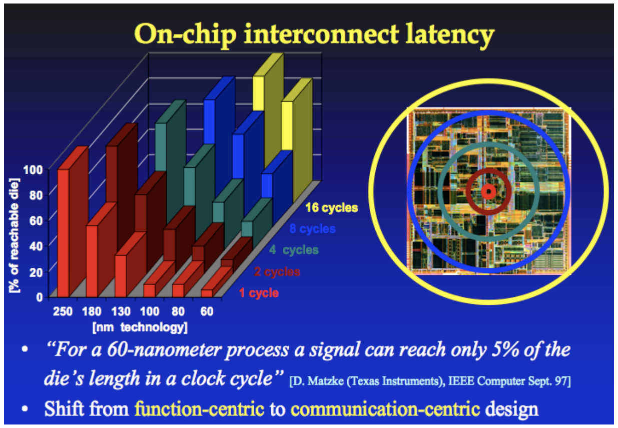


Cannot necessarily presume a uniform communication time form core-to-core

Impediments to Parallel Performance

(*independent of problem/algorithm itself*)

- Latency
  - Multiple clock cycles to cross chip or to communicate from core-to-core even in absence of contention



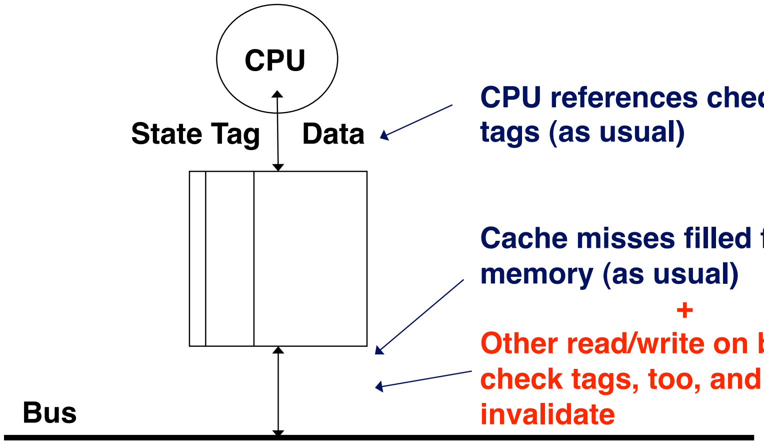
On-chip interconnect latency

- *"For a 60-nanometer process a signal can reach only 5% of the die's length in a clock cycle"* [D. Matzke (Texas Instruments), IEEE Computer Sept. 97]
- Shift from function-centric to communication-centric design

Impediments to Parallel Performance

(*independent of problem/algorithm itself*)

- Cache coherency
  - P1 writes, P2 can read → need consistent data
    - Protocols can enable $ coherency but not free



CPU references check cache tags (as usual)

Cache misses filled from memory (as usual)

+

Other read/write on bus must check tags, too, and possibly invalidate

Impediments to Parallel Performance

(*independent of problem/algorithm itself*)

- Programming languages, environments, & methods:
  - Need simple semantics that can expose computational properties to be exploited by large-scale architectures

What if you write good code for a 4-core chip and then get an 8-core chip?

- i.e. tune code to deal with communication latencies, etc.

Impediments to Parallel Performance

(*independent of problem/algorithm itself*)

$$\text{Speedup} = \frac{1}{\left[1 - \text{Fraction}_{\text{parallelizable}}\right] + \dfrac{\text{Fraction}_{\text{parallelizable}}}{N}}$$

- All "bears" adversely affect Fraction$_{\text{parallelizable}}$ and hence speedup
- Should have quantitative sense of overhead when writing software
  - Tricky because it can change from chip-to-chip!

Computer Architecture…

Parallel software…

Can keep bears at bay, but need to think hard about how

We ran into a bear along the way…