Accelerated Computing with GPUs and Intel Xeon Phi

Summer 2020

Prof. Karen L. Karavanic

karavan@pdx.edu

http://web.cecs.pdx.edu/~karavan

Portland State UNIVERSITY

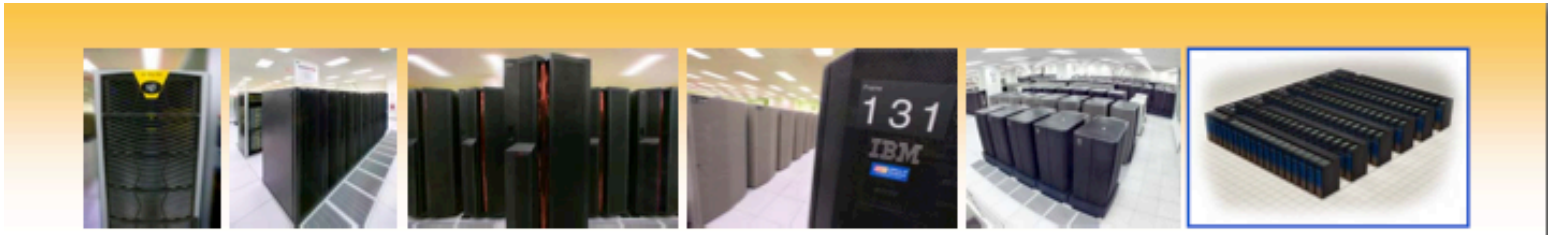
What is Accelerated Computing ?

- Heterogeneous Computing
- Idea: Combine general purpose x86 CPU(s) with one or more *accelerators*
- Idea: Use different hardware for different types of computation
- Graphics Processing Units (GPU) really bad at sequential computing, extremely fast and efficient at parallel computing
- Advantages: speedup, power savings
- Today's state of the art to accomplish large-scale computing
- Different types of accelerators: GPUs, APUs, FPGAs, Intel Xeon Phi, …
- Our focus will be CUDA/GPUs with some Intel Xeon Phi

2. What is a High End Computer Platform Today?

- Built up of thousands of processors/nodes/computers
- Nodes connected to work together (network, interconnect)
- Each Node may contain a number of processors and one or more accelerators
- (HPC/Scientific) Run Parallel Programs:
  - message passing: between nodes
  - Multithreading: within each node
- (High End Servers/Scientific and Commercial) Run Map/Reduce, Analytics, Web Servers, Database Servers
- (Cloud) Warehouse-Scale Computers: 50-100,000 servers, networking, power and cooling

Accelerators Motivation #1: Power and Cooling



Power Efficiency and the Top500

John Shalf and David Bailey

Lawrence Berkeley National Laboratory
Top500 Birds of a Feather
SC2006, Tampa Florida
November 14, 2006

BIPS

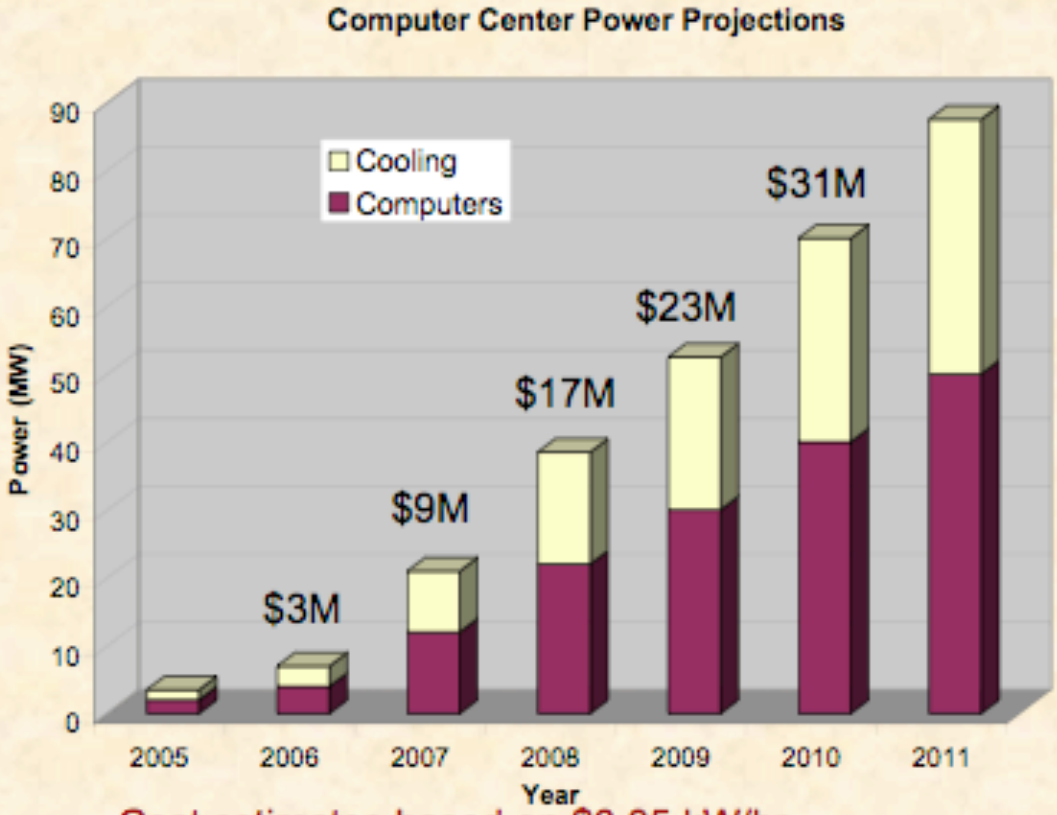
BERKELEY LAB

NERSC

Office of Science
U.S. DEPARTMENT OF ENERGY

ORNL Computing Power and Cooling 2006 - 2011

- Immediate need to add 8 MW to prepare for 2007 installs of new systems
- NLCF petascale system could require an additional 10 MW by 2008
- Need total of 40-50 MW for projected systems by 2011
- Numbers just for computers: add 75% for cooling
- Cooling will require 12,000 – 15,000 tons of chiller capacity



Cost estimates based on $0.05 kW/hr

Annual Average Electrical Power Rates $/MWh

Site	FY 2005	FY 2006	FY 2007	FY 2008	FY 2009	FY 2010
LBNL	43.70	50.23	53.43	57.51	58.20	56.40 *
ANL	44.92	53.01				
ORNL	46.34	51.33				
PNNL	49.82	N/A				

Data taken from Energy Management System-4 (EMS4). EMS4 is the DOE corporate system for collecting energy information from the sites. EMS4 is a web-based system that collects energy consumption and cost information for all energy sources used at each DOE site. Information is entered into EMS4 by the site and reviewed at Headquarters for accuracy.

Accelerators Motivation #2: Compute Capability

- www.top500.org
- Linpack
- Floating Point Operations per Second (FLOP/s)

- June 9, 2008: The Roadrunner breaks the "PetaFLOP barrier" - computer achieves a rate of 1.026 petaFLOP/s

*Whatta*FLOPS ???

- Floating Point Operations Per Second
- MFLOPS 10^{6}
- GFLOPS 10^{9}
- TFLOPS 10^{12}
- PetaFLOPS – 10^{15}
  1,000,000,000,000,000
- ___FLOPS – 10^{18} coming soon…..

Accelerators Motivation #2: Compute Capability

- www.top500.org
- Linpack
- Floating Point Operations per Second (FLOP/s)

- June 9, 2008: The Roadrunner breaks the "PetaFLOP barrier" - computer achieves a rate of 1.026 petaFLOP/s

Accelerators Motivation #2: Compute Capability

- www.top500.org
- Linpack
- Floating Point Operatings per Second (FLOP/s)

- June 9, 2008: The Roadrunner computer achieves a rate of 1.026 petaFLOP/s

- Yes, that's 1.026×10^{15} floating point operations per second

Accelerators Motivation #2: Compute Capability

- www.top500.org
- Linpack
- Floating Point Operatings per Second (FLOP/s)

- June 9, 2008: The Roadrunner computer achieves a rate of 1.026 petaFLOP/s

- Yes, that's 1.026 *quadrillion* floating point operations per second
- How did they do it?
  - 6,948 dual-core AMD Opteron chips
  - 12,960 8-core Cell BE *(Playstation)* chips *as accelerators*

Example: NVIDIA Kepler (K20)

- Peak double precision: 1.17 TeraFLOPs
- Peak single precision: 3.52 TeraFLOPs

Example: NVIDIA Kepler (K20)

- Peak double precision: 1.17 TeraFLOPs
- Peak single precision: 3.52 TeraFLOPs

- *Yes, that's just for one Graphics card !!*

Example: NVIDIA Kepler (K20)

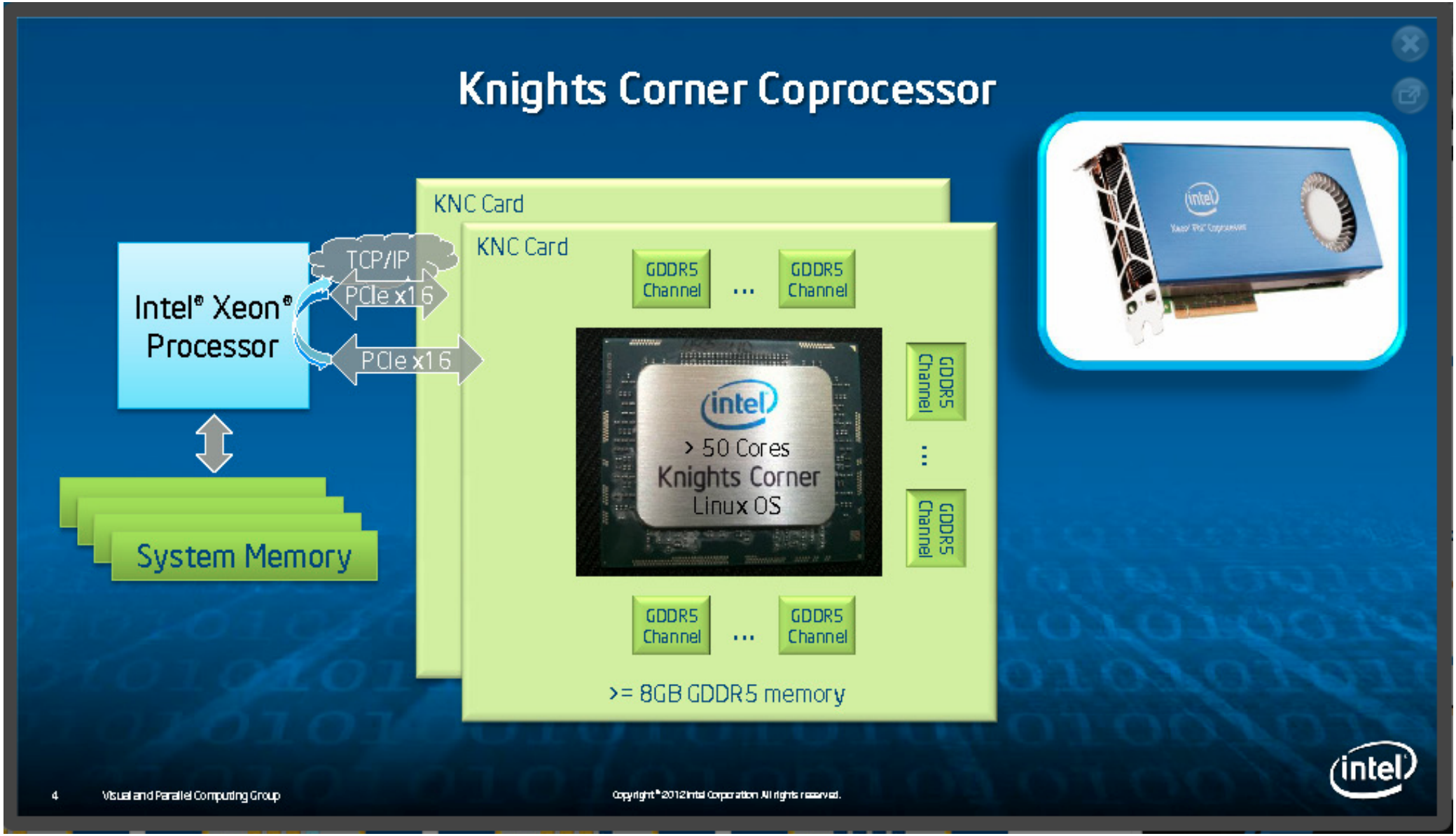
- Peak double precision: 1.17 TeraFLOPs
- Peak single precision: 3.52 TeraFLOPs

- Key Idea: Single Instruction Multiple Data
- Example: Vector sum C = A + B

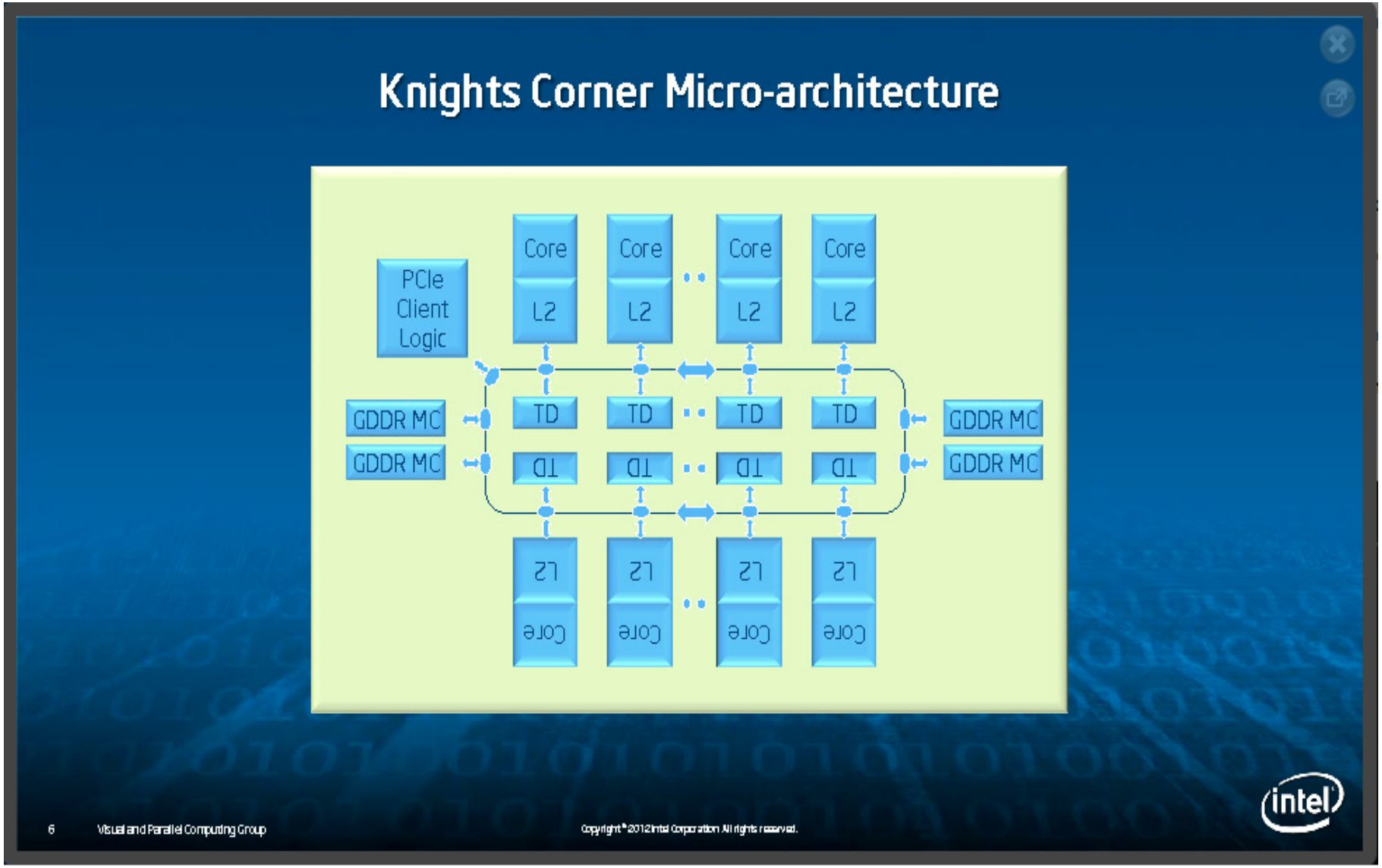
```
for (int i = 0; i < n; i++)
     C[i] = A[i] + B[i];
```

- We are executing the same instruction *n* times, with different data each time
- Can we do this in parallel?

Example: Intel Phi Coprocessor



Example: Intel Phi Coprocessor



Today: Trinity @LANL

- Architecture Cray XC30
- Memory capacity >2 PB of DDR4 DRAM
- Peak performance >40 PF
- Number of compute nodes >19,000
- Processor architecture:
  Intel Haswell & Knights Landing ("Xeon Phi")
- Parallel file system capacity (usable) >80 PB
- Parallel file system bandwidth (sustained) 1.45 TB/s
- Burst buffer storage capacity (usable) 3.7 PB
- Burst buffer bandwidth (sustained) 3.3 TB/s
- Footprint <5,200 sq ft
- Power requirement <10 MW

Continuing Challenges: Scaling, Power, Cooling

3. Accelerated Computing Course Structure

- Week 1: See course web page for tentative calendar, slides:
  - http://web.cecs.pdx.edu/~karavan/gpu
- Week 2-8: All materials will be on the google drive in a shared folder: CS 435 535 Accelerated
  - Accessible only by registered students
  - If your class registration isn't complete at the end of week 1 let me know or you will lose access
- The lectures will not fully reproduce the readings – you must do both
- We will spend zoom time with lecture, hands on demo and group work
  - Please let me know if something is not working for you. I have been teaching for 20 years and zoom teaching for 2.5 months!

3. Accelerated Computing Course Structure

- Readings include material not in lecture but needed for homework
- Homeworks are designed for independent learning
  - Questions and discussions - yes BUT do your own work
  - Include programming in C/C++, CUDA, OpenACC
- Small Group Projects allow you to explore a topic in more detail
  - I will post a list of projects and you will select your 1st, 2nd, 3rd choice
  - I will form the groups
  - Presentations last day of class
- Grading Breakdown
  - Homeworks 60%
  - Project 40%

4. Introductions: Professor Karavanic

- Stuyvesant High School (NYC):
  - Public, Math and Science, admission by exam
- New York University: B.A. Computer Science
  - I am a “First Generation College Student”
  - I completed my degree while working full-time
- University of Wisconsin – Madison:
  - M.S. Computer Science
  - Ph.D. Computer Science
    - WARF Fellow, IBM intern, NASA GSRP Fellow
- Portland State University 2000 ->
  - LLNL, SDSC, New Mexico Consortium
- Current ResearchProjects:
  - Drought Prediction, Holistic HPC Workflow Performance, SMM-based Runtime Integrity Checking

4. Introductions: Professor Karavanic

- How to reach me:
  - Email: please take care with subject lines
  - Zoom:
    - Weekly Office Hours: TBD
    - OR
    - Email for an appointment (zoom – video optional)
    - OR
    - Ask questions in email
  - I do enjoy your questions, feedback and interest
  - I will do my best to reply to email quickly
  - karavan@pdx.edu
- Special Pandemic rule in effect: Please feel free to call me Karen

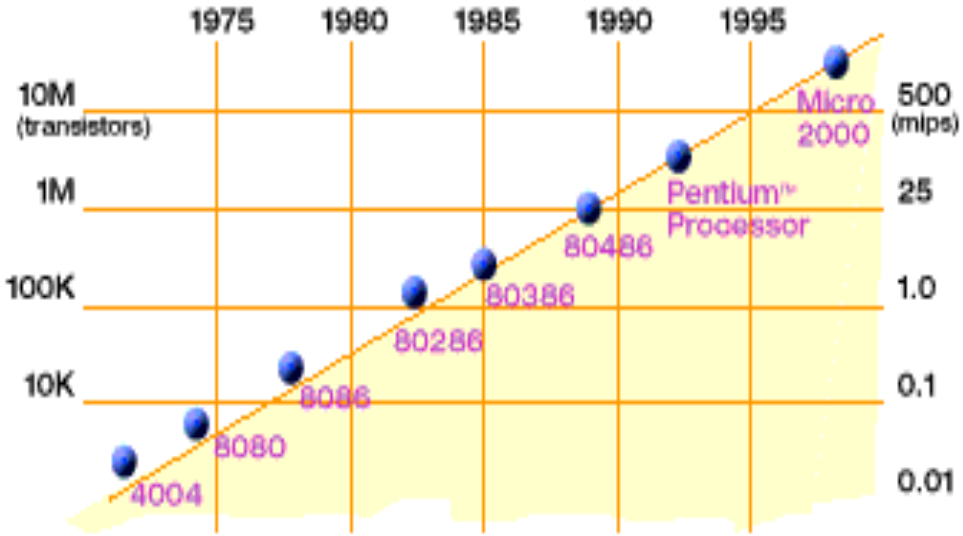
Introductions: You

Please email to me at karavan@pdx.edu

- 1. Your name
- 2. Your goals for this course (including what grade you want)
- 3. Your experience with C and C++
- 4. Your experience with parallel programming, CUDA, Xeon Phi (if any)
- 5. Any topics of particular interest?

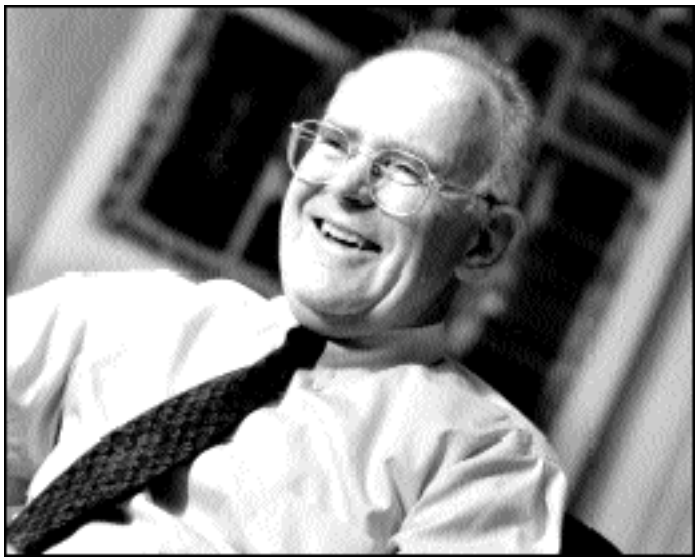
5. Multicore and Manycore Computing
How did we get here?

Technology Trends: Microprocessor Capacity



2X transistors/Chip Every 1.5 years
Called "Moore's Law"

Microprocessors have become smaller, denser, and more powerful.

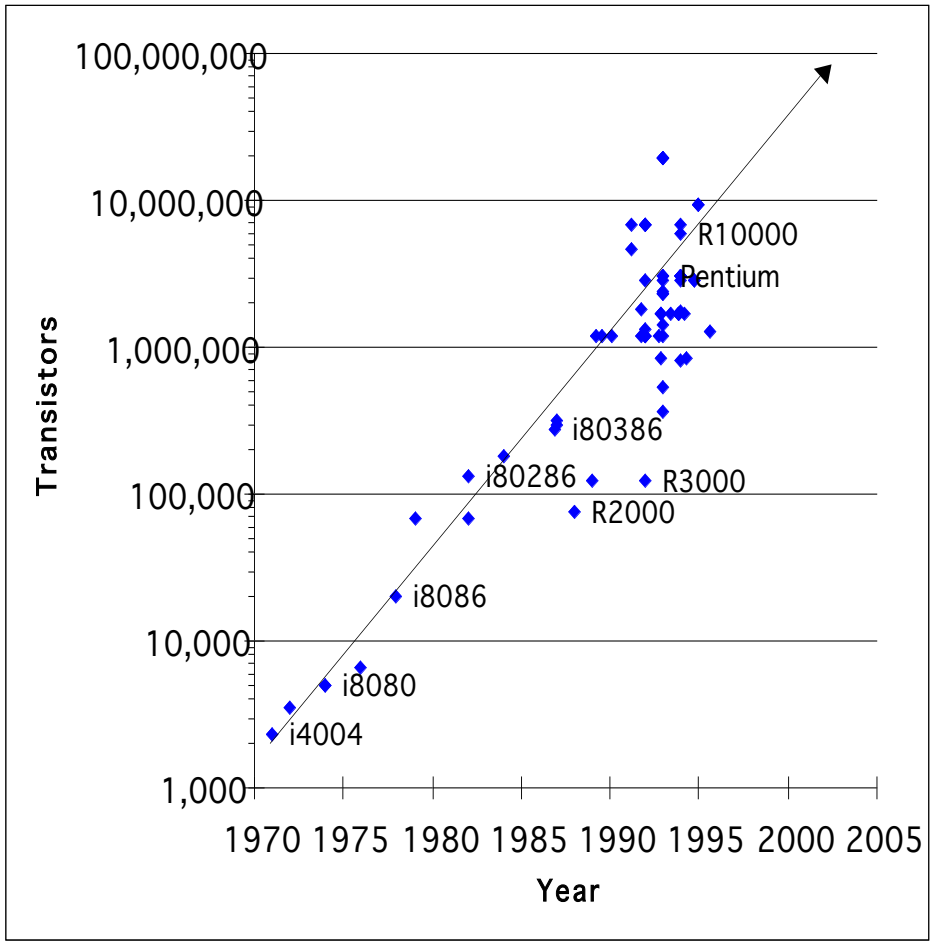


Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

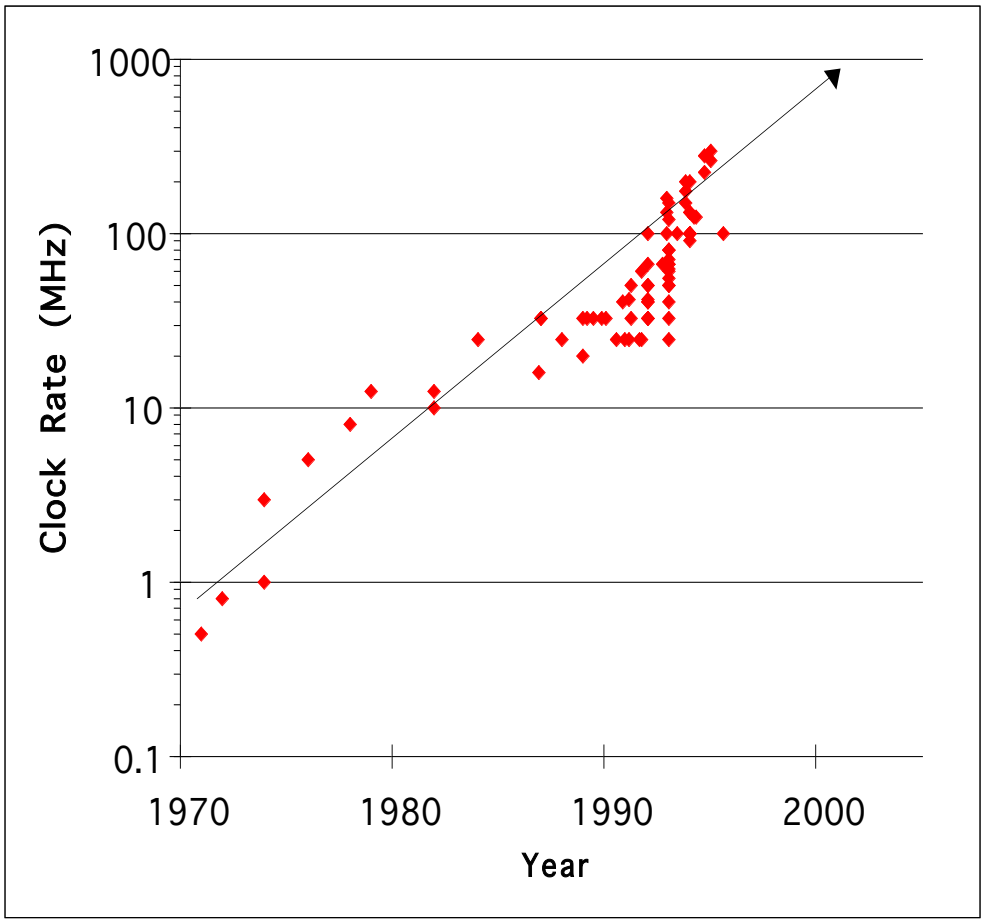
Slide source: Jack Dongarra

Microprocessor Transistors and Clock Rate

Growth in transistors per chip



Increase in clock rate

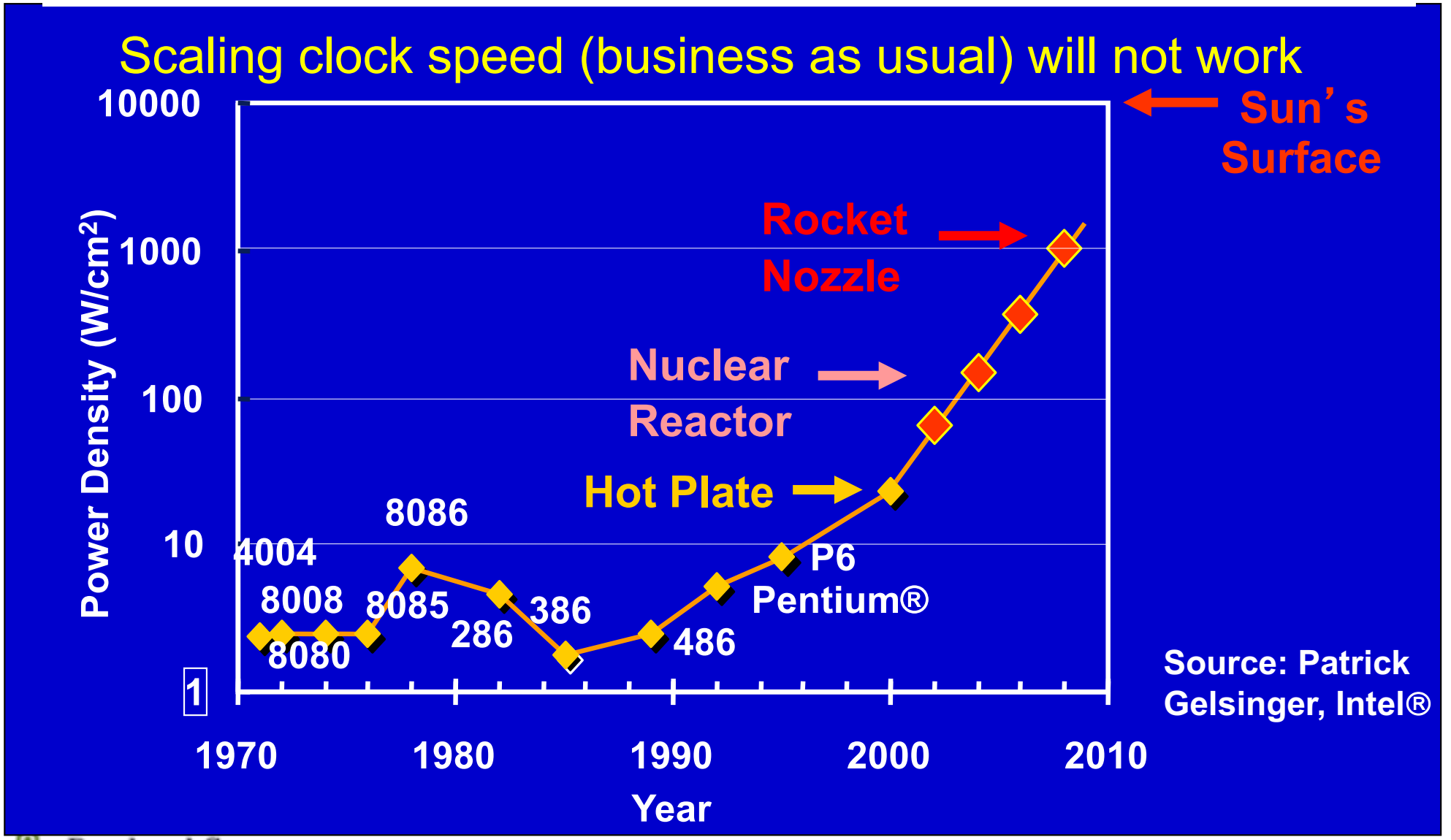


Why bother with parallel programming? Just wait a year or two…

Limit #1: Power density

Can soon put more transistors on a chip than can afford to turn on.

-- Patterson '07



Parallelism Saves Power

- Exploit explicit parallelism for reducing power

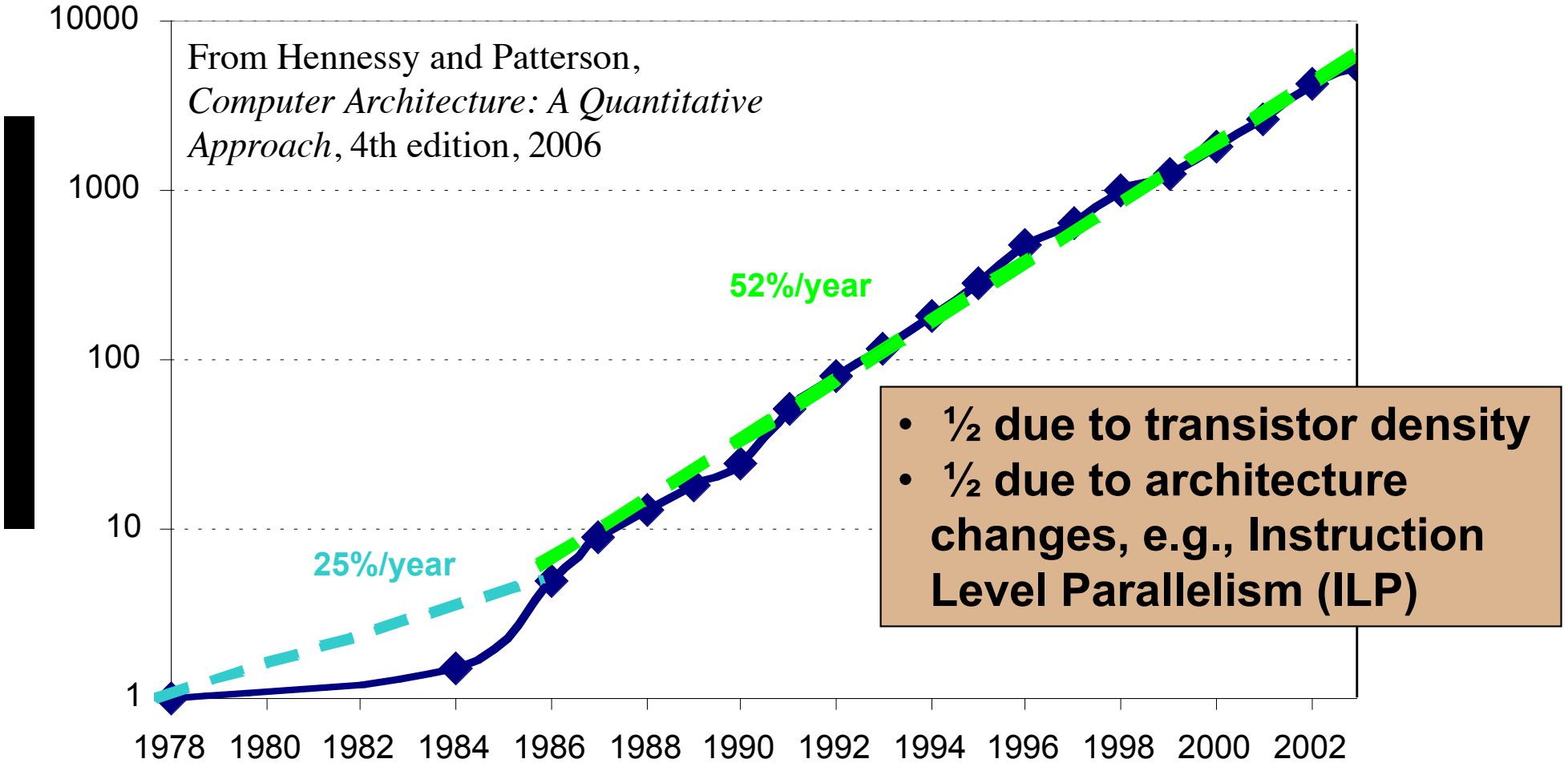
$$\text{Power} = (C * V^2 * F)/4 \qquad \text{Performance} = (\text{Cores} * F)*1$$

Capacitance Voltage Frequency

- **Using additional cores**
  - Increase density (= more transistors = more capacitance)
  - Can increase cores (2x) and performance (2x)
  - Or increase cores (2x), but decrease frequency (1/2): same performance at ¼ the power
- **Additional benefits**
  - Small/simple cores → more predictable performance

Limit #2: Hidden Parallelism Tapped Out

Application performance was increasing by 52% per year as measured by the SpecInt benchmarks here



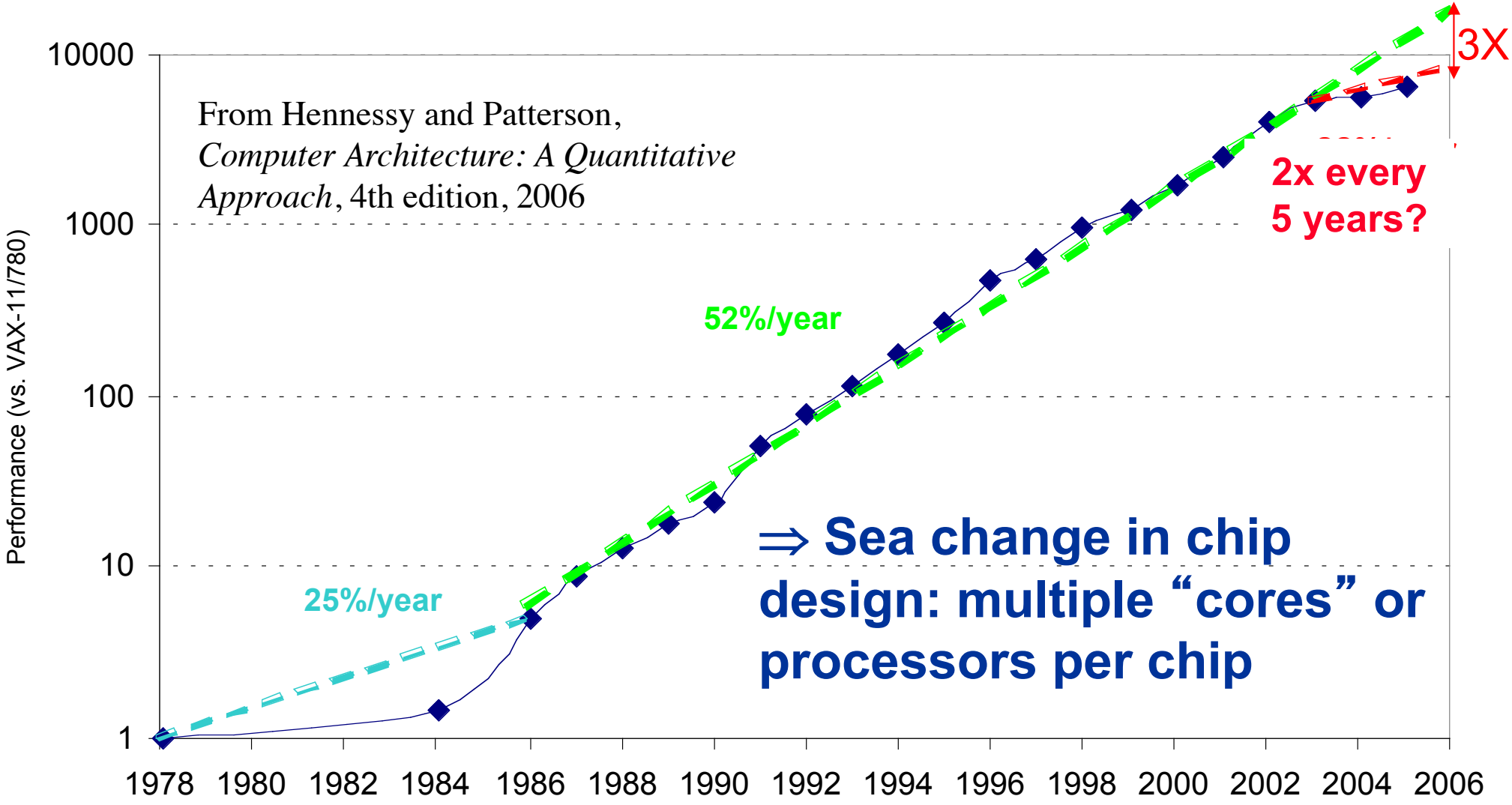
- ½ due to transistor density
- ½ due to architecture changes, e.g., Instruction Level Parallelism (ILP)

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

Limit #2: Hidden Parallelism Tapped Out

- **Superscalar (SS) designs were the state of the art; many forms of parallelism not visible to programmer**
  - **multiple instruction issue**
  - **dynamic scheduling: hardware discovers parallelism between instructions**
  - **speculative execution: look past predicted branches**
  - **non-blocking caches: multiple outstanding memory ops**
- **Unfortunately, these sources had been used up**

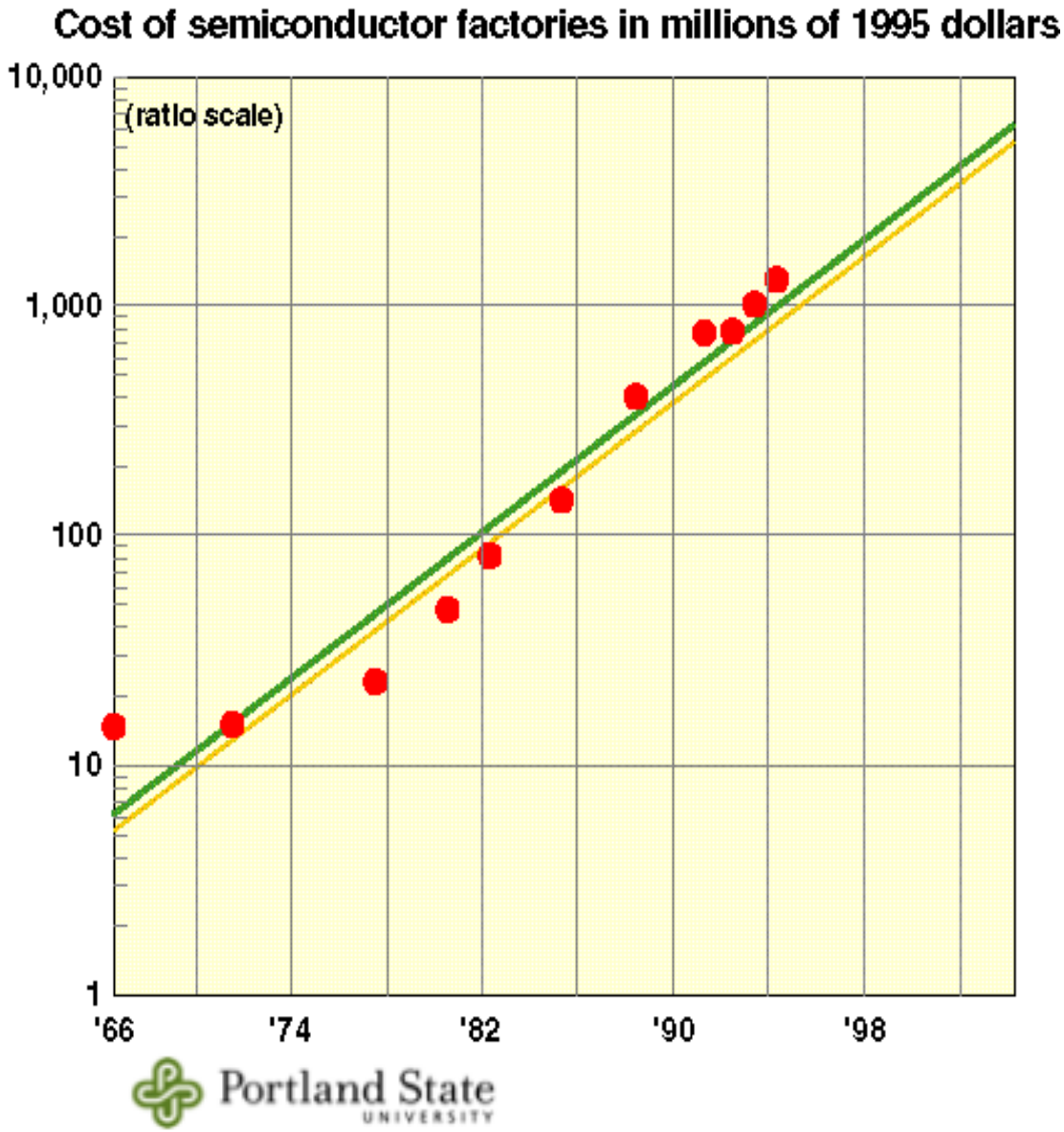
Uniprocessor Performance (SPECint) Today



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

Limit #3: Chip Yield

Manufacturing costs and yield problems limit use of density

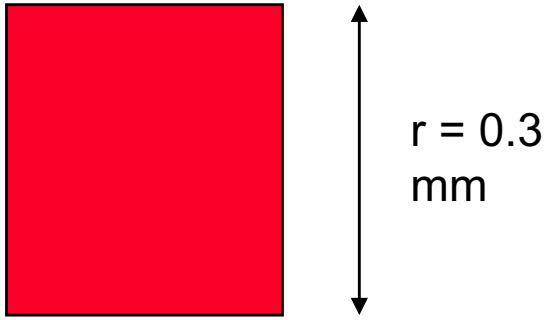


- **Moore's (Rock's) 2nd law: fabrication costs go up**
- **Yield (% usable chips) drops**

- **Parallelism can help**
  - **More smaller, simpler processors are easier to design and validate**
  - **Can use partially working chips:**
  - **E.g., Cell processor (PS3) is sold with 7 out of 8 "on" to improve yield**

Limit #4: Speed of Light (Fundamental)

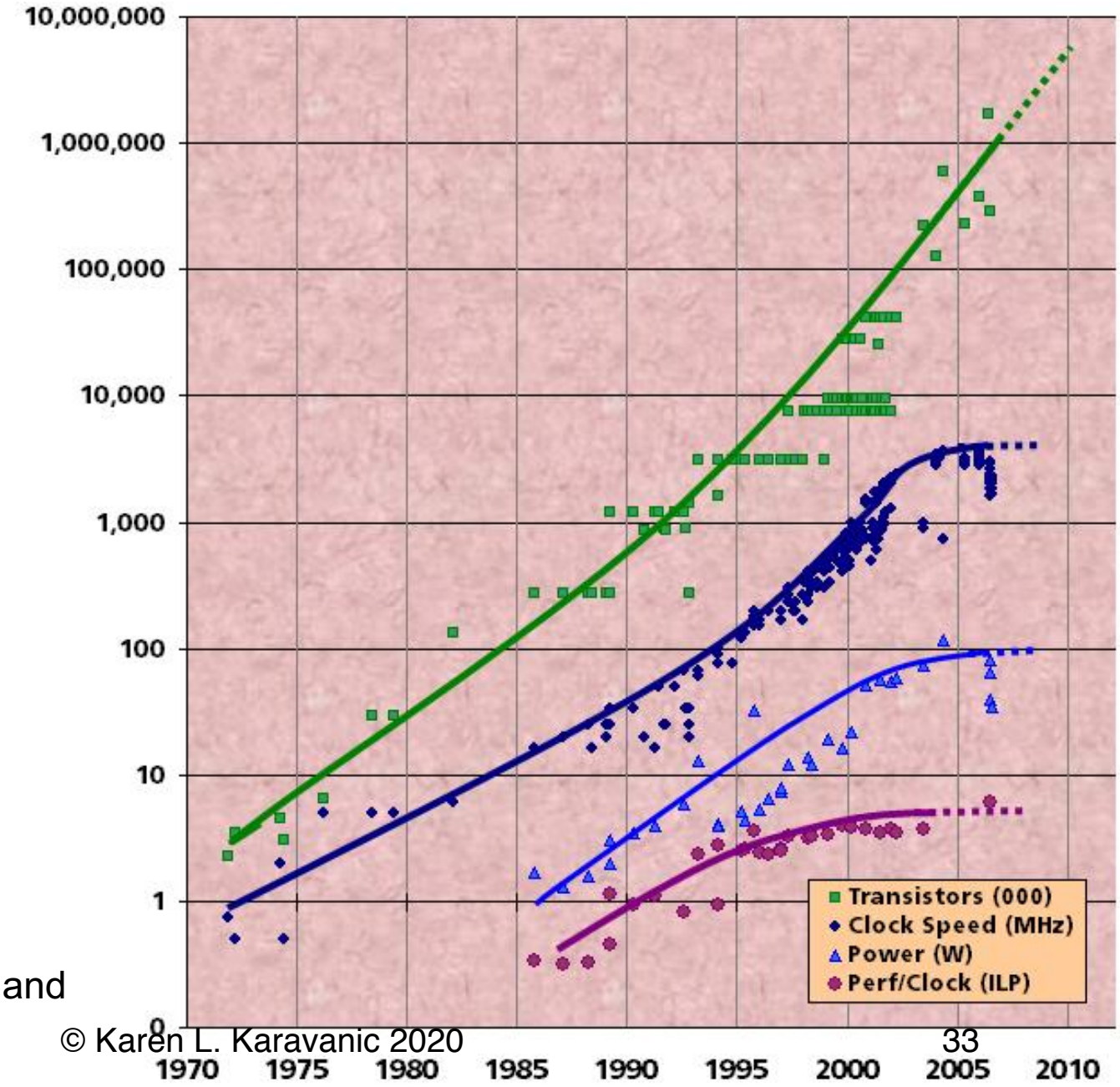
1 Tflop/s, 1 Tbyte sequential machine



- Consider the 1 Tflop/s sequential machine:
  - Data must travel some distance, r, to get from memory to CPU.
  - To get 1 data element per cycle, this means 10^{12} times per second at the speed of light, c = $3x10^8$ m/s. Thus r < $c/10^{12}$ = 0.3 mm.
- Now put 1 Tbyte of storage in a 0.3 mm x 0.3 mm area:
  - Each bit occupies about 1 square Angstrom, or the size of a small atom.
- No choice but parallelism

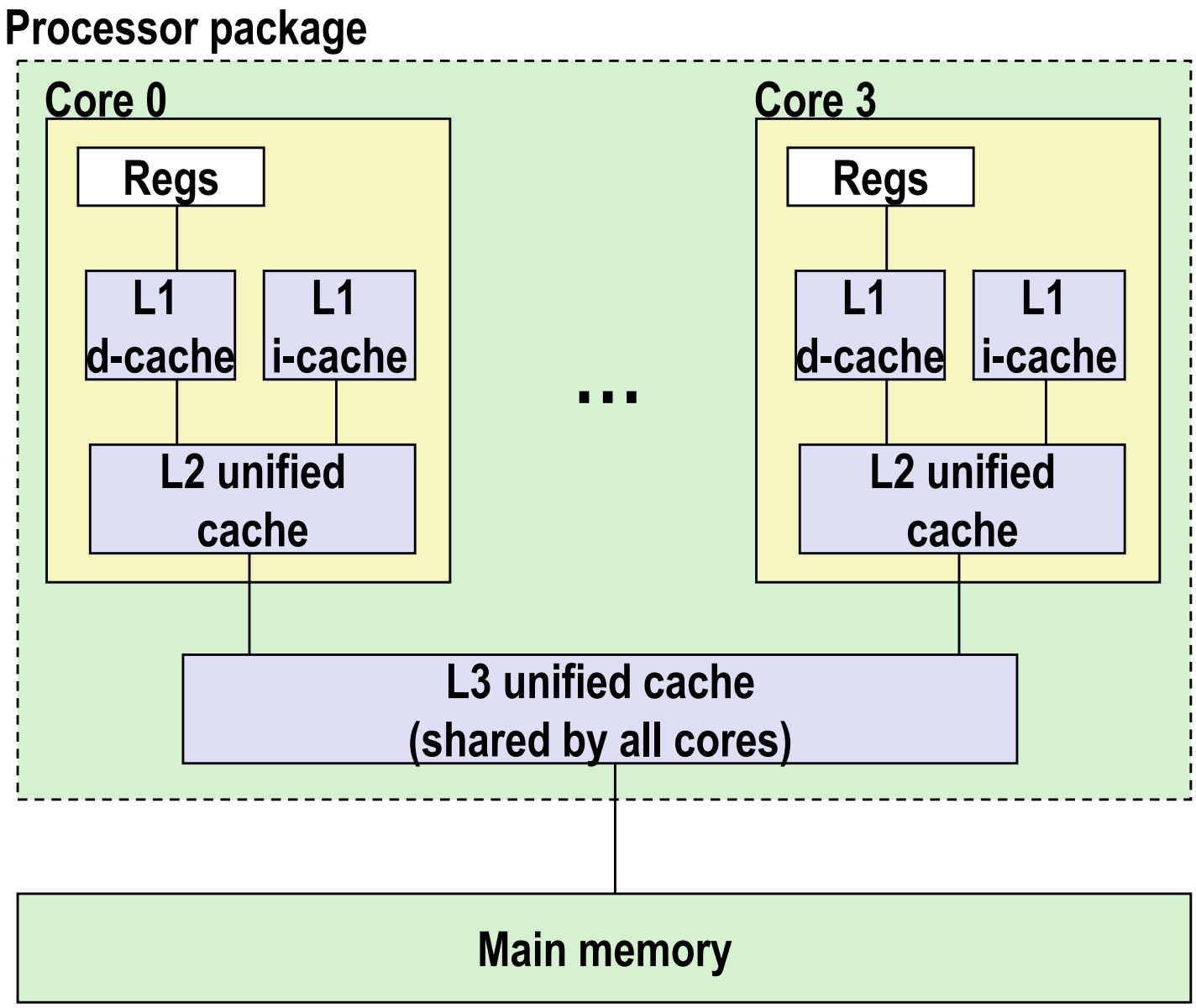
"Sea Change"

- Chip density continuing increase ~2x every 2 years
  - Clock speed is not
  - Number of processor cores may double instead
- Little or no hidden parallelism (ILP) to be found
- Parallelism must be exposed to and managed by software



Source: Intel, Microsoft (Sutter) and Stanford (Olukotun, Hammond)

Intel Core i7 Cache Hierarchy



L1 i-cache and d-cache:
32 KB, 8-way,
Access: 4 cycles

L2 unified cache:
256 KB, 8-way,
Access: 11 cycles

L3 unified cache:
8 MB, 16-way,
Access: 30-40 cycles

Block size: 64 bytes for all caches.

What about Today?

- It is 13 years after the switch to multicore
  - IBM Power8: 4-12 cores, Intel Xeon E7: 4-24 cores
- Post Moore's Law Era
  - We are no longer on the Moore's Law curve !
  - # devices per chip increasing at a slower rate
  - Focus has switched to compute efficiency
    - Large systems now have a *power budget*
- Dark Silicon
  - not all parts of the chip can be powered at once
  - due to thermal constraints
- Domain-specific architectures and Acceleration

What is Manycore ?

- What if we use all of the transisters on a chip for as many cores as we can fit??
- Beyond the edge of number of cores in common “multicore” architectures
- Dividing line is not clearly defined, changes with advances in technology
- Active research, now in embedded & clusters
- Current trend at the high end is to combine CPUs with manycore “accelerators”

What is Manycore ?

- Examples:
  - NVIDIA Fermi Graphics Processing Unit (GPU)
    - First model: 32 “CUDA cores” per SM, 16 SMs
      - (SM = “streaming multiprocessor”)
    - Kepler K20 model: 2496 CUDA cores, peak 3.52 TFlops
    - Tesla V100 model: 5120 CUDA cores, 7.5 TFlops
  - Matrix-2000 accelerators
    - Chinese name “迈创”, meaning “taking a creative step”
    - 128 cores, each can perform 16 Double Precision FLOPs per cycle
  - Intel Xeon Phi 7290
    - Up to 72 cores each
    - Intel AVX-512 vector instructions

Ex: NVIDIA Fermi

Accelerated Computing Today

- Example: TianHe-2A Supercomputer (China)
  - 17,792 nodes with Intel XeonCPUs and Matrix-2000 accel.
  - 4,981,760 cores total
  - peak 100,679 TeraFLOPS
- Example: Summit Supercomputer (U.S.)
  - IBM Power9 processors
  - accelerated with NVIDIA Volta GPUs (“accelerators”)
  - Total # cores: 2,414,592
  - peak 200,794.9
  - achieved on TOP 500 benchmark Nov 2019:
    - 148,600.0 TFlop/s
- See top500.org

Thanks…

- This talk includes slides, ideas, and examples from: Kathy Yelick (UC Berkeley), Wen-mei Hwu (UIUC/NVIDIA)

- This course includes hardware generously provided by NVIDIA. We will use some materials from NVIDIA

- This course includes hardware, technical staff time, software licenses and other resources generously provided by Intel Corp.