



Visão Geral do Curso

Computação Paralela

Faculdade de Computação e Informática

Outline

❑ Visão do Curso

- O que é Computação Paralela?
- O que esperamos dos alunos?
- O que você aprenderá neste curso?

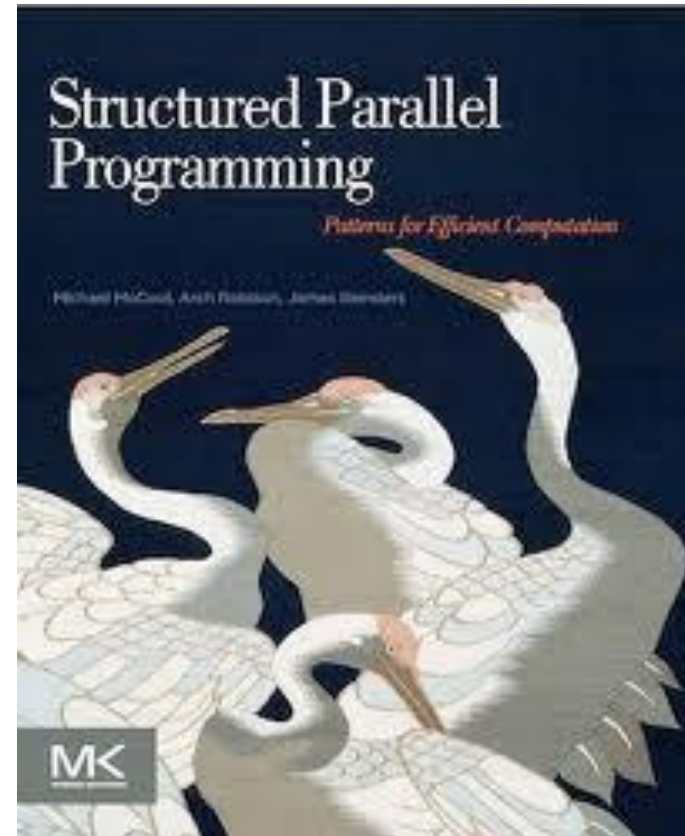
❑ Computação Paralela

- O que é?
- O que motiva a computação paralela?
- Tendência que moldam a área
- Problemas de grande escala e alto desempenho
- Tipos de arquiteturas paralelas
- Computação paralela escalável e desempenho



Livros do curso

- ❑ “Structured Parallel Programming: Patterns for Efficient Computation,” Michael McCool, Arch Robinson, James Reinders, 1st edition, Morgan Kaufmann, ISBN: 978-0-12-415993-8, 2012
<http://parallelbook.com/>
- ❑ Apresenta programação paralela de um ponto de vista de padrões relevantes para a computação paralela
 - **Utilizaremos até Aula 2 - Arquitetura de Computadores Paralelos**
- ❑ Foco no uso de programação paralela em memória compartilhada



Livros do Curso

- *An Introduction to Parallel Programming, Pacheco, P. , 1st Ed, Elsevier, 2011*
 - **Utilizaremos para OpenMP e MPI**
- *Parallel Programming for Multicore and Cluster Systems, Rauber, T., Runger, G., Springer Verlag, 2010.*
- *Introduction to Parallel Computing, A. Grama, A. Gupta, G. Karypis, V. Kumar, Addison Wesley, 2nd Ed., 2003*
 - **Utilizaremos para os algoritmos**

Plano do Curso

- ❑ Organizado de modo a cobrir as principais áreas da computação paralela nas aulas
 - Arquitetura (2w)
 - Tecnologias Paralelismo (OpenMP, Pthreads, MPI) (3w)
 - Algoritmos (4 w)
 - Ferramentas (1 w)
 - Aplicações (2 w)
 - Tópicos especiais (1 w)
- ❑ Laboratório de Programação paralela para desenvolvimentos de atividades.
- ❑ Projeto de Curso

Aulas - funcionamento

- ❑ O livro e outros materiais online são a sua principal fonte para fundamentação e ampliação do conhecimento em computação paralela
- ❑ As aulas são mais interativas
 - Suplementam outras fontes de informação
 - Cobrem alguns tópicos prioritários
 - Apresentam uma perspectiva do professor
 - Os slides estarão disponíveis no Moodle
- ❑ Nas aulas de laboratório teremos, eventualmente, slides para comentarmos alguns tópicos específicos.

Laboratório de Programação

- ❑ Utilizaremos a infra do Mackenzie
 - Com uma máquina virtual em Linux
 - O ideal seria cada um ter (mesmo que seja só em casa) uma máquina com Linux para desenvolver os laboratórios e projeto. Pode ser uma VM.
- ❑ Utilizaremos OpenMP, Pthreads e MPI
- ❑ Tentaremos abordar GPU

Exercícios do curso

- ❑ Lições para casa
 - Voltados para preparação para provas
- ❑ Laboratórios de Programação Paralelas
 - Exercícios com padrões de programação paralela
 - Programas utilizarão OpenMP, Pthreads e MPI
 - Eventualmente, utilizaremos GPU também.
- ❑ Projeto do curso
 - Programação, apresentação e relatório

Visão Geral do curso

- ❑ Área ampla/velha da ciência da computação que lida com:
 - Arquitetura, sistemas de HW/SW, linguagens, paradigmas de programação, algoritmos e modelos teóricos.
 - Computação em Paralelo.
- ❑ Desempenho é a *razão da existência* do paralelismo.
 - Computação de alto desempenho
 - Dirige (lidera) a revolução na ciência computacional
- ❑ Tópicos de estudo
 - Arquiteturas Paralelas
 - Programação Paralela
 - Algoritmos Paralelo
 - Modelos de desempenho e ferramentas paralelas
 - Aplicações Paralelas



O que você aprenderá neste curso?

- ❑ Entendimento profundo do projeto de computadores paralelos
- ❑ Conhecimento de como programar sistemas de computação paralela
- ❑ Entendimento da programação paralela baseada em padrões
- ❑ Exposição a diferentes formas de algoritmos paralelos
- ❑ *Experiência prática utilizando um cluster paralelo*
- ❑ Experiência em modelagem de desempenho paralelo
- ❑ Técnicas para análise empírica de desempenho

Processamento Paralelo - O que é isto?

- ❑ Um *computador paralelo* é um sistema computacional que utiliza múltiplos elementos de processamento simultaneamente de modo cooperativo para resolver um problema computacional.
- ❑ *Processamento paralelo* inclui técnicas e tecnologias que tornam possível fazer computação em paralelo:
 - Hardware, redes, sistemas operacionais, bibliotecas paralelas, linguagens, compiladores, algoritmos, ferramentas, . . .
- ❑ Computação paralela é uma evolução da computação serial
 - Paralelismo é natural
 - Problemas de computação diferem em nível/tipo de paralelismo
- ❑ Paralelismo é sempre sobre desempenho! Verdade?

Concorrência

- ❑ Considere múltiplas tarefas a serem executadas em um computador
- ❑ Tarefas são concorrentes com relação uma às outras se:
 - Elas *podem* executar ao mesmo tempo (*execução concorrente*)
 - Isso implica que não existem dependências entre as tarefas
- ❑ Dependências
 - Se uma tarefa requer resultados produzidos por outra tarefa para poder executar corretamente, a execução desta tarefa é *dependente*.
 - Se duas tarefas são dependentes, elas não são concorrentes
 - Alguma forma de sincronização deve ser utilizada para forçar (satisfazer) as dependências.
- ❑ Concorrência é fundamental à ciência da computação
 - Sistemas operacionais, bancos de dados, redes, . . .

Concorrência e Paralelismo

- ❑ Concorrente não é a mesma que paralelo! Por quê?
- ❑ Execução paralela
 - Tarefas concorrentes *podem de verdade* executar ao mesmo tempo
 - Múltiplos recursos (processamento) **devem** ser disponíveis
- ❑ **Paralelismo = concorrência + hardware “paralelo”**
 - Ambos são necessários
 - Encontrar oportunidades de execução concorrente
 - Desenvolver a aplicação para executar em paralelo
 - Rodar a aplicação em um hardware paralelo
- ❑ Uma aplicação paralela é uma aplicação concorrente?
- ❑ Uma aplicação paralela roda com um processador paralelo?
Por que sim? Por que não?

Paralelismo

- ❑ Existem granularidades de paralelismo (execução paralela) nos programas
 - Processos, threads, rotinas, declarações, instruções, ...
 - Pense sobre quais são os elementos de software que executam concorrentemente
- ❑ Estes elementos precisam ser suportados por recursos de hardware
 - Processadores, cores, ... (execução de instruções)
 - Memória, DMA, redes, ... (outras operações associadas)
 - Todos os aspectos da arquitetura de um computador oferece oportunidades para a execução em hardware paralelo
- ❑ Concorrência é uma condição necessária para paralelismo
 - Onde podemos encontrar concorrência?
 - Como a concorrência é expressa para explorar sistemas paralelos?

Por que utilizar processamento paralelo?

- ❑ Duas razões básicas (ambas relacionadas a desempenho)
 - Tempo mais rápido para encontrar uma solução (tempo de resposta)
 - Resolver problemas maiores (no mesmo tempo)
- ❑ Outros fatores que motivam o processamento paralelo
 - Uso efetivo de recursos de máquina
 - Eficiência de Custo
 - Contornar limitações de memória
- ❑ Máquinas seriais têm limitações inerentes
 - Velocidade do processador, gargalos de memória
- ❑ Paralelismo se tornou o futuro da computação
- ❑ Desempenho é ainda a principal preocupação
- ❑ **Paralelismo = concorrência + HW paralelo + desempenho**



Perspectivas sobre Processamento Paralelo

- ❑ Arquitetura de computadores paralelos
 - Qual o hardware necessário para execução em paralelo?
 - Projeto dos sistemas computacionais
- ❑ Sistema Operacional (Paralelo)
 - Como administrar os aspectos do sistema em um computador paralelo
- ❑ Programação Paralela
 - Bibliotecas (baixo nível, alto nível)
 - Linguagens
 - Ambientes de desenvolvimento de software
- ❑ Algoritmos paralelos
- ❑ Avaliação do desempenho paralelo
- ❑ Ferramentas paralelas
 - Desempenho, analytics, visualização ...



Por que estudar computação paralela hoje?

- ❑ Arquitetura de computação
 - Inovações frequentemente levam a novos modelos de programação
- ❑ Convergência tecnológica
 - O super PC está se tornando comum
 - Notebooks e supercomputadores são fundamentalmente similares!
 - Tendências fazem com que diversas abordagens convirjam
- ❑ Tendências tecnológicas tornam a computação paralela inevitável
 - Processadores multi-core chegaram para ficar!
 - Praticamente todo sistema computacional está operando em paralelo
- ❑ Entender os princípios fundamentais e as compensações de projeto
 - Programação, suporte de sistemas, comunicação, memória,
 - Desempenho
- ❑ Paralelismo é o futuro da computação



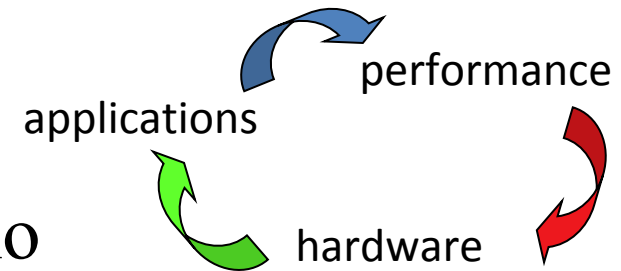
Inevitabilidade da Computação Paralela

- ❑ Aplicações demandam
 - Necessidade insaciável por ciclos de computação
- ❑ Tendências tecnológicas
 - Processadores e memória
- ❑ Tendências de arquitetura
- ❑ Econômicas
- ❑ Tendências atuais
 - Microprocessadores de hoje tem suporte a multiprocessadores
 - Servidores e *workstations* disponíveis como multiprocessadores
 - Microprocessadores de amanhã serão multiprocessadores
 - Multi-core é definitivo e a razão (num cores)/processador está crescendo
 - Aceleradores (GPUs, sistemas de jogos)



Características das Aplicações

- ❑ Desempenho do SW demandam avanços no HW
- ❑ Novas aplicações tem maiores demandas de desempenho
 - Aumento exponencial no desempenho do microprocessador
 - Inovações na integração das arquiteturas paralelas



- ❑ Faixa de requerimento de desempenho
 - Desempenho do sistema como um todo também tem que melhorar
 - Requerimentos de desempenho requerem engenharia de computação
 - Custos são endereçados através de avanços tecnológicos

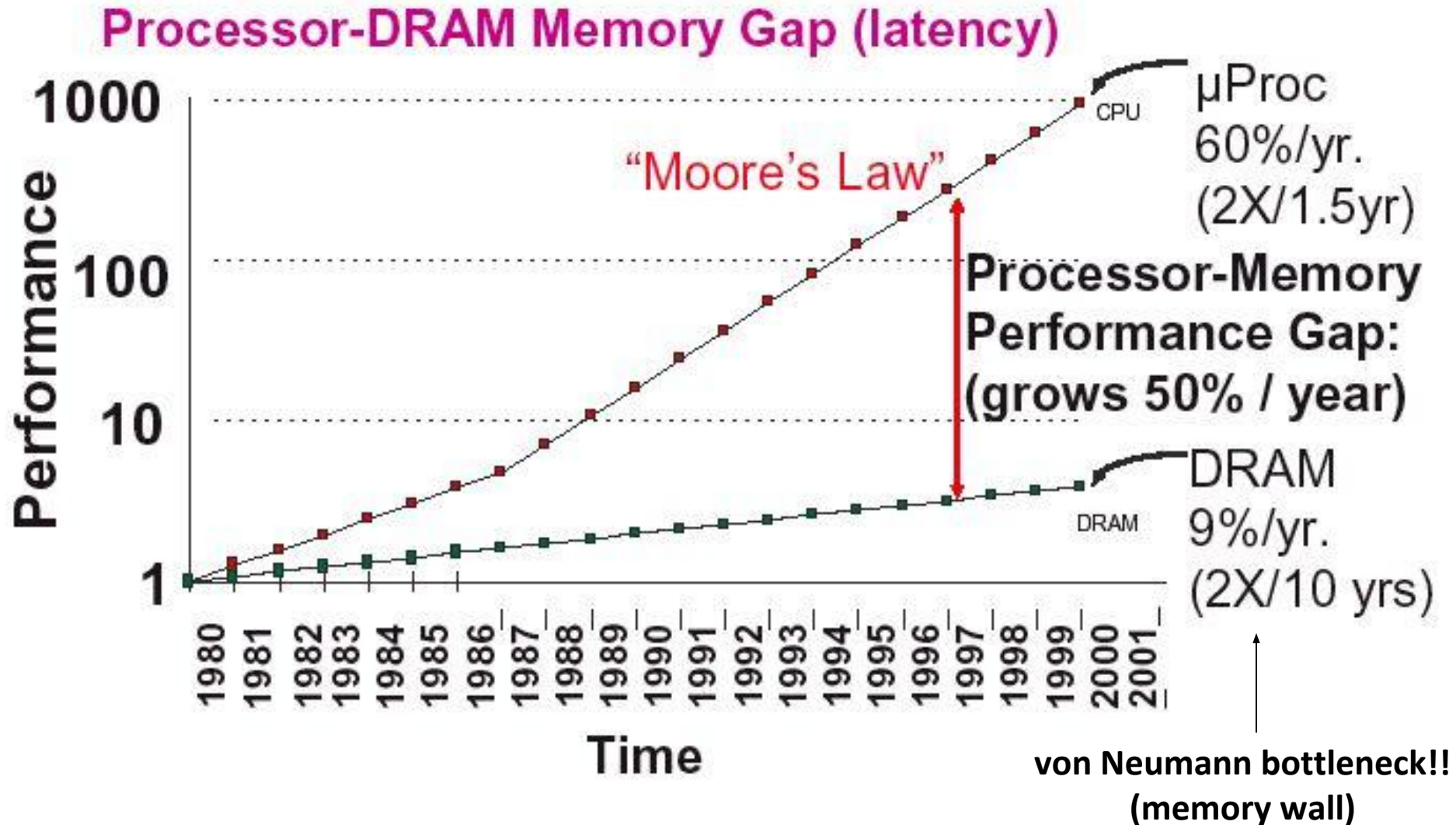
Questões (amplas) de Arquiteturas Paralelas

- ❑ Alocação de Recursos
 - Quantos elementos de processamento?
 - Quão poderosos são os elementos?
 - Quanto de memória?
- ❑ Sincronização, comunicação e acesso a dados
 - Como os elementos cooperam e comunicam?
 - Como os dados são transmitidos entre os processadores?
 - Quais as abstrações e primitivas para cooperação?
- ❑ Desempenho e estabilidade
 - Como tudo isso se traduz em desempenho?
 - Como escala?

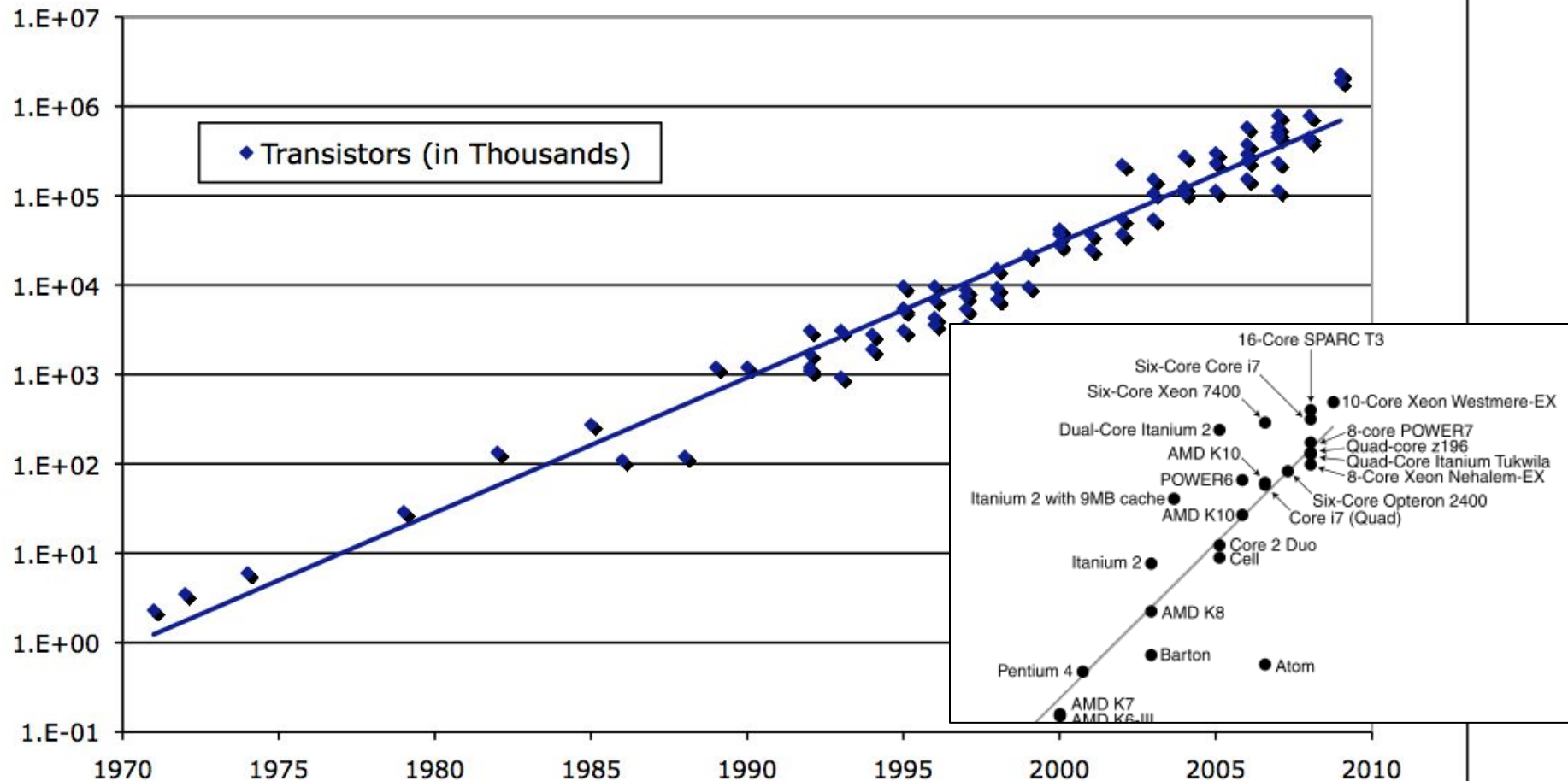
Alavancando a Lei de Moore

- ❑ Mais transistores = mais oportunidades de paralelismo
- ❑ Microprocessadores
 - Paralelismo implícito
 - ◆ pipelining
 - ◆ múltiplas unidades funcionais
 - ◆ superscalar
 - Paralelismo explícito
 - ◆ Instruções SIMD
 - ◆ Instruções longas funcionam

O que impulsiona arquitetura de computação paralela?



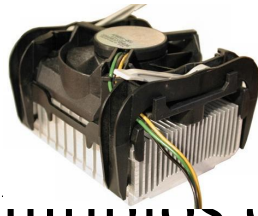
Núm Transistores por Microprocessador (1971-2011)



Data from Kunle Olukotun, Lance Hammond, Herb Sutter,
Burton Smith, Chris Batten, and Krste Asanović
Slide from Kathy Yelick

O que tem acontecido nos últimos anos?

- ❑ Fabricantes de chips de processamento aumentaram o desempenho aumentando o clock da CPU
 - Cavalgando na Lei de Moore
- ❑ Até que os chips ficaram muito quentes!
 - Maior frequência de clock \Rightarrow maior potência elétrica
 - Dissipador de calor do Pentium 4 • Fritando um ovo no Pentium 4



- ❑ Acrescenta múltiplos cores para aumentar desempenho
 - Mantém ou reduz a frequência de clock
 - Mantém nos mesmos níveis de requisitos de energia

Crescimento da Densidade de Potência

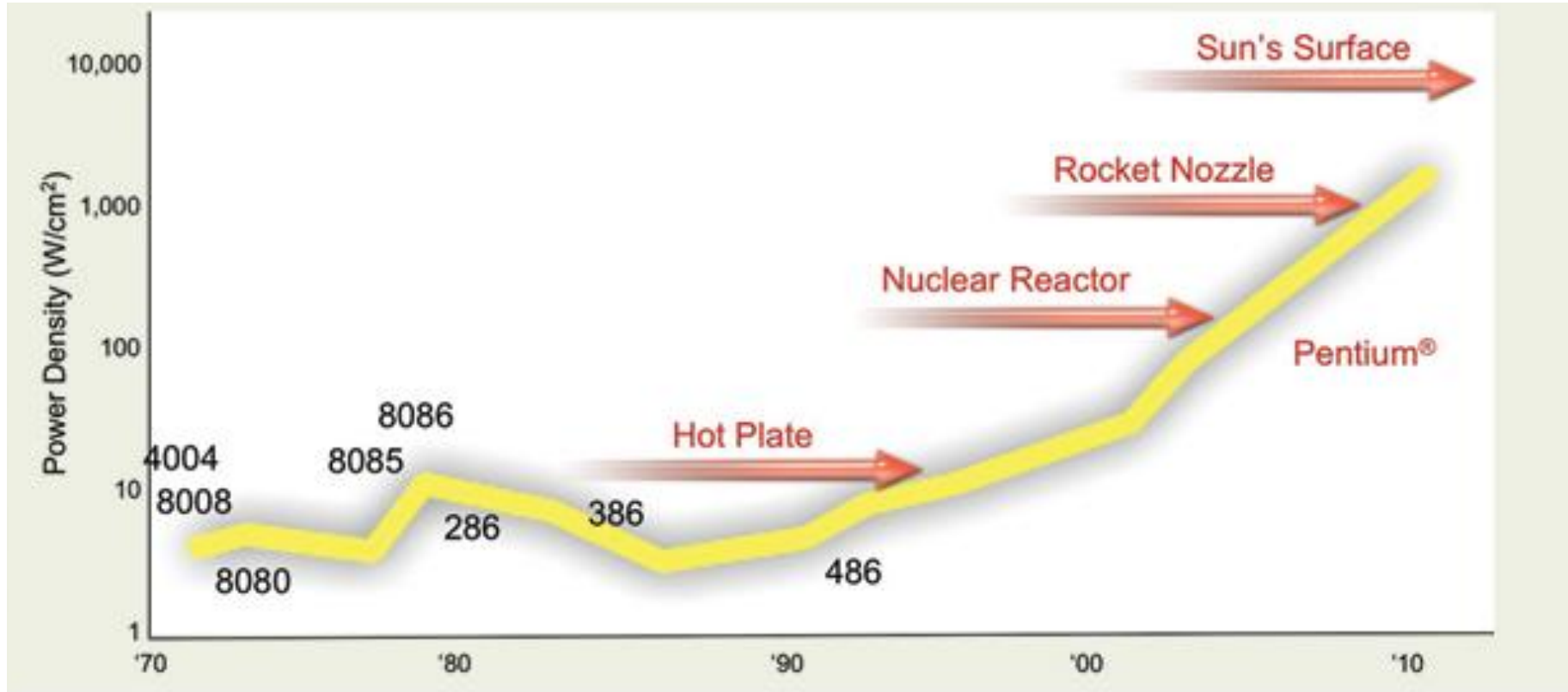
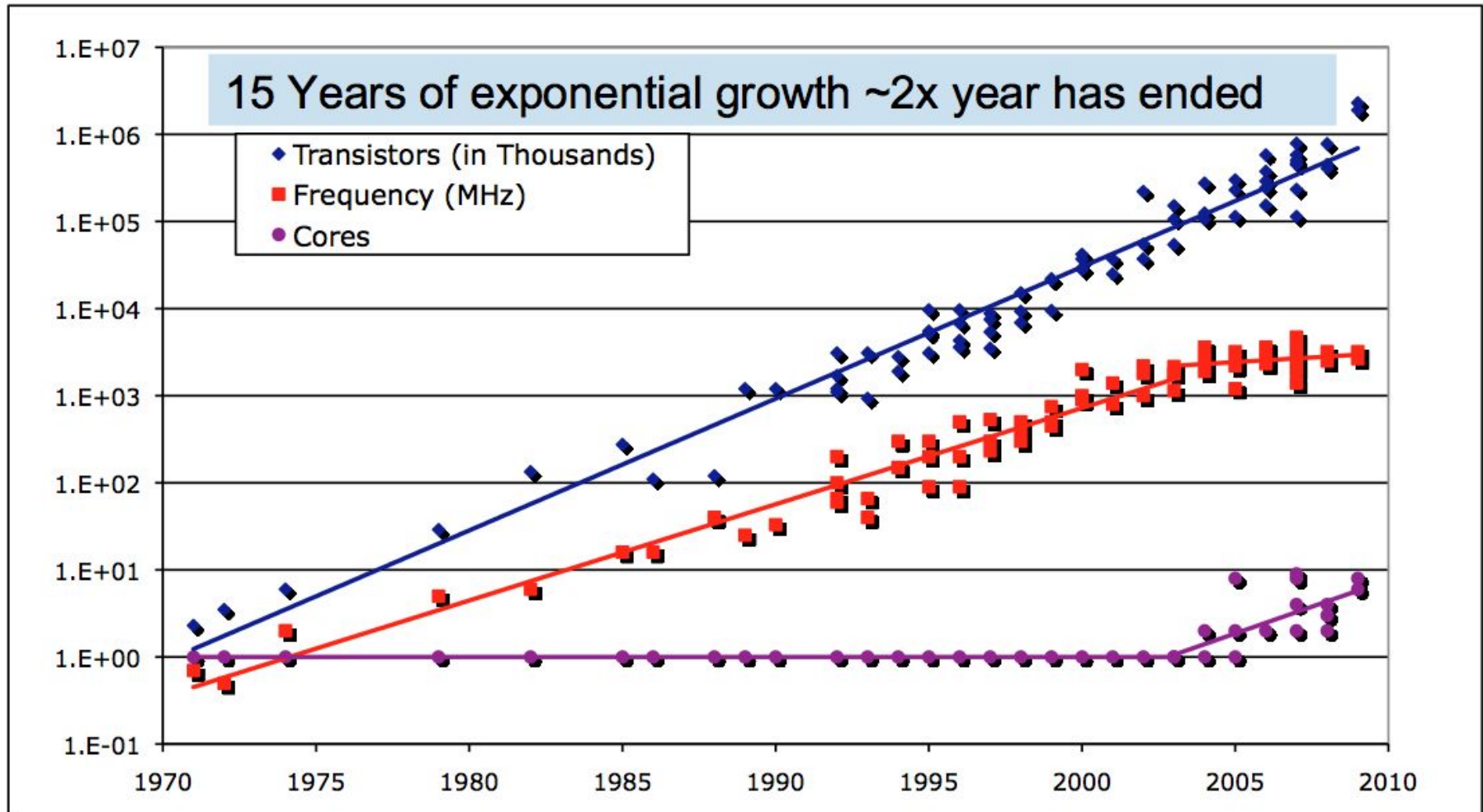


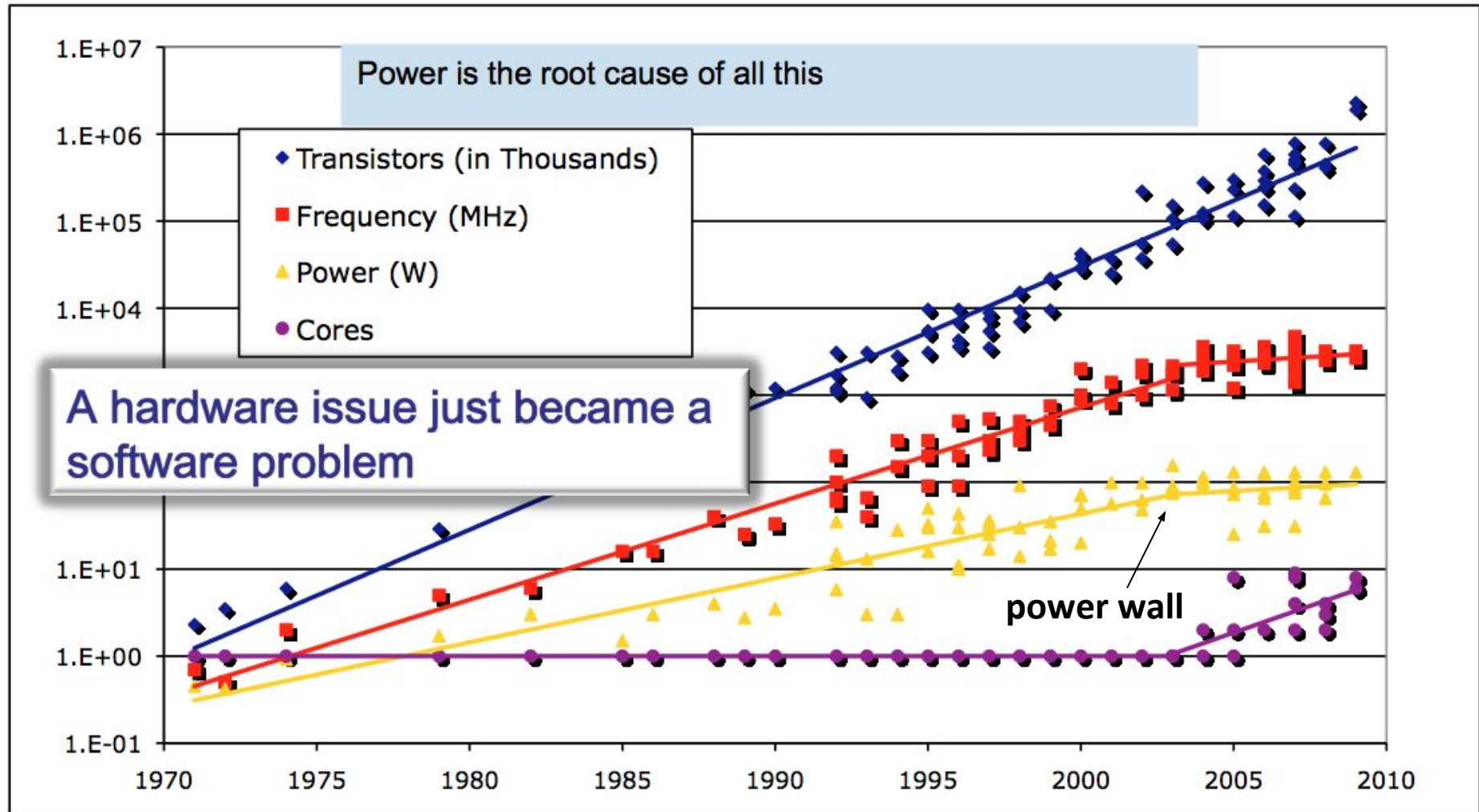
Figure courtesy of Pat Gelsinger, Intel Developer Forum, Spring 2004

O que impulsiona arquitetura de computação paralela?



Data from Kunle Olukotun, Lance Hammond, Herb Sutter,
Burton Smith, Chris Batten, and Krste Asanović
Slide from Kathy Yelick

O que impulsiona arquitetura de computação paralela?



Classificando Sistemas Paralelos - Taxonomia de Flynn

- ❑ Distingue arquiteturas multiprocessador em duas dimensões independentes
 - *Instrução e Dados*
 - Cada dimensão pode ter somente um estado: *Single* ou *Multiple*
- ❑ SISD: Single Instruction, Single Data
 - Máquina Serial (não paralela)
- ❑ SIMD: Single Instruction, Multiple Data
 - Arrays de processadores e máquinas vetoriais
- ❑ MISD: Multiple Instruction, Single Data (estranho)
- ❑ MIMD: Multiple Instruction, Multiple Data
 - Sistemas de computação paralela mais comuns

Tipos de arquiteturas paralelas

- ❑ Paralelismo no nível de instrução
 - Paralelismo capturado no processamento de instruções
- ❑ Processadores vetoriais
 - Operações em múltiplos dados armazenados em registradores vetoriais
- ❑ Multiprocessador de memória compartilhada (SMP)
 - Múltiplos processadores compartilhando a memória
 - Multiprocessador simétrico (SMP)
- ❑ Multicomputador
 - Múltiplos computadores conectados por rede
 - Cluster de memória distribuída
- ❑ Processadores Massivamente Paralelos (MPP)



Fases de arquiteturas de supercomputadores (paralelos)

- ❑ Fase 1 (1950s): execução sequencial de instruções
- ❑ Fase 2 (1960s): problema na execução sequencial
 - Execução em Pipeline, estações de reserva
 - Paralelismo no Nível de Instrução (ILP)
- ❑ Fase 3 (1970s): processadores vetoriais
 - Unidades aritméticas com pipeline
 - Registradores, sistemas de memória com múltiplos bancos em paralelo
- ❑ Fase 4 (1980s): SIMD e SMPs
- ❑ Fase 5 (1990s): MPPs e clusters
 - Processadores de comunicação sequencial
- ❑ Fase 6 (>2000): muitos cores, aceleradores, escala, ...

Expectativas de Desempenho

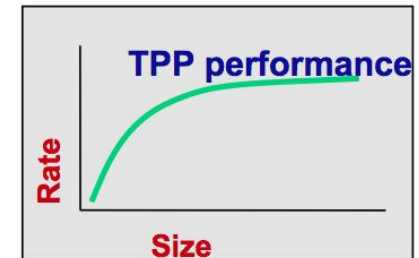
- ❑ Se cada processador tem k MFLOPS e existem p processadores, nós esperaríamos ver $k \cdot p$ MFLOPS de desempenho? Correto?
- ❑ Se leva 100 segundos em um processador, deveria levar 10 segundos em 10 processadores? Correto?
- ❑ Várias causas afetam o desempenho
 - Cada uma precisa ser entendida separadamente
 - Mas elas interagem umas com as outras de modo complexo
 - ◆ uma solução para um problema pode criar outro
 - ◆ um problema pode mascarar outro
- ❑ Escalonamento (sistema, tamanho do problema) pode mudar condições
- ❑ É preciso entender o espaço de desempenho

Escalabilidade

- ❑ Um problema pode escalonar para utilizar muitos processadores
 - O que isto significa?
- ❑ Como avaliar a escalabilidade?
- ❑ Como avaliamos a efetividade da escalabilidade?
- ❑ Avaliação comparativamente
 - Se dobramos o número de processadores, o que esperar?
 - A escalabilidade é linear?
- ❑ Utilizar uma medida de eficiência de paralelização
 - A eficiência é mantida conforme o tamanho do problema aumenta?
- ❑ Aplicamos métricas de desempenho

Top 500 Benchmarking Methodology

- ❑ Listing of the world's 500 most powerful computers
- ❑ Yardstick for high-performance computing (HPC)
 - Rmax : maximal performance Linpack benchmark
 - ◆ dense linear system of equations ($Ax = b$)
- ❑ Data listed
 - Rpeak : theoretical peak performance
 - Nmax : problem size needed to achieve Rmax
 - N1/2 : problem size needed to achieve 1/2 of Rmax
 - Manufacturer and computer type
 - Installation site, location, and year
- ❑ Updated twice a year at SC and ISC conferences



Top 10 (November 2013)

Different architectures

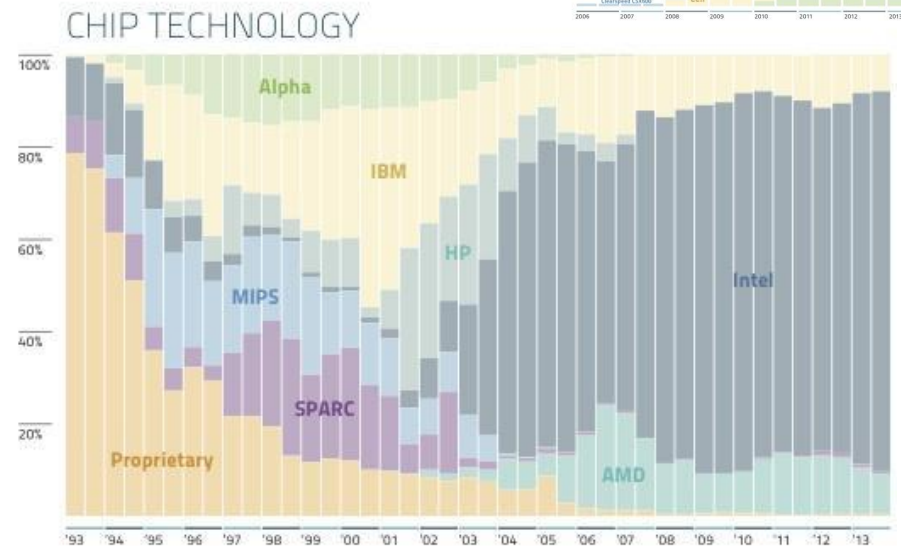
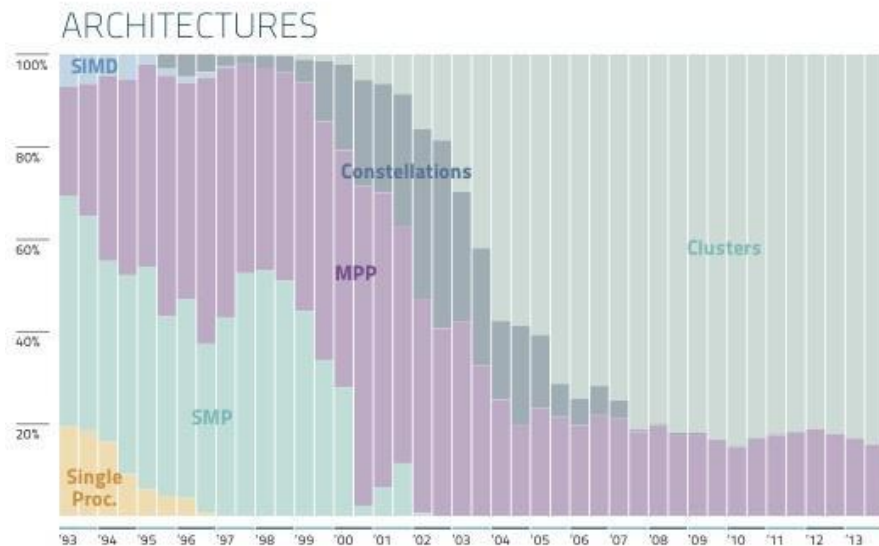
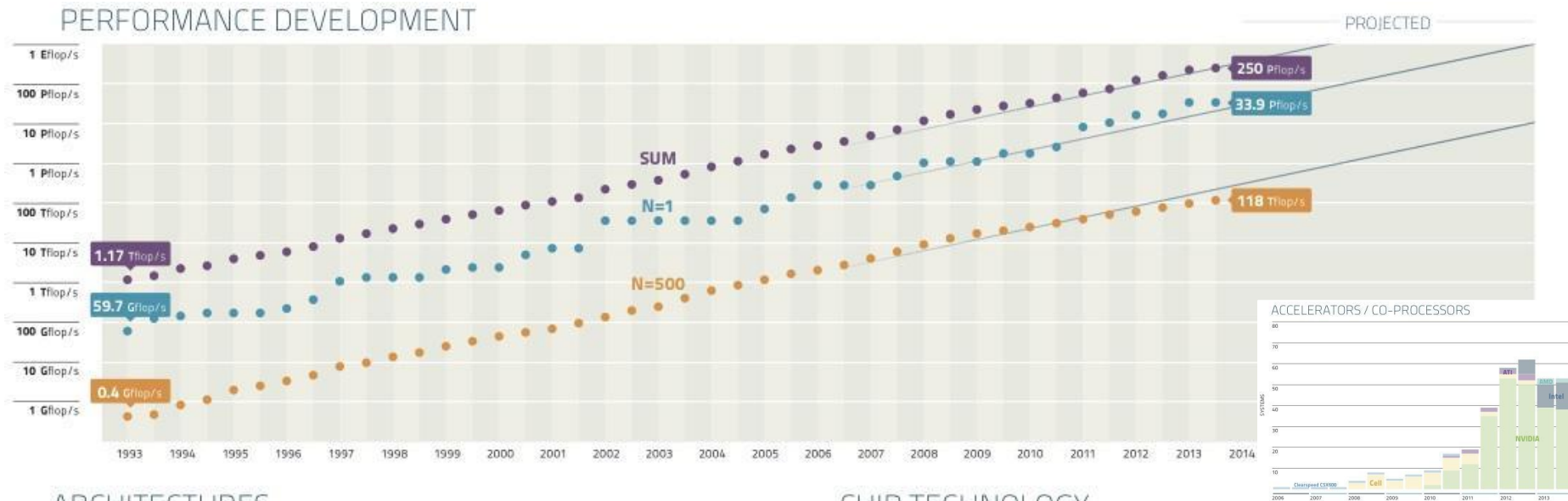
Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer , SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre (CSCS) Switzerland	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	Texas Advanced Computing Center/Univ. of Texas United States	Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
8	Forschungszentrum Juelich (FZJ) Germany	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301
9	DOE/NNSA/LLNL United States	Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393,216	4,293.3	5,033.2	1,972
10	Leibniz Rechenzentrum Germany	SuperMUC - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR IBM	147,456	2,897.0	3,185.1	3,423



Top 500

- ❑ Vá ao site do top500 e estude as 3 arquiteturas que estão no topo da listagem
 - tipo de arquitetura
 - performance em k FLOPS
 - desde quando está no topo
 - curiosidades

Top 500 – Performance (November 2013)



#1: *NUDT Tiahne-2 (Milkyway-2)*

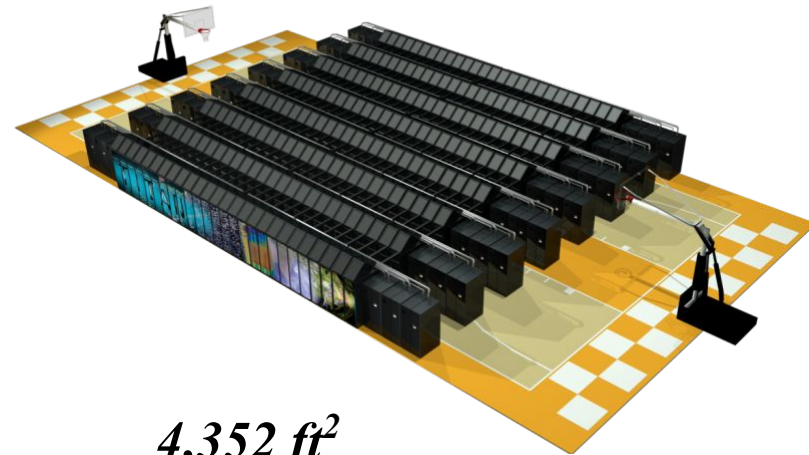
- ❑ Compute Nodes have 3.432 Tflop/s per node
 - 16,000 nodes
 - 32000 Intel Xeon CPU
 - 48000 Intel Xeon Phi
- ❑ Operations Nodes
 - 4096 FT CPUs
- ❑ Proprietary interconnect
 - TH2 express
- ❑ 1PB memory
 - Host memory only
- ❑ Global shared parallel storage is
- ❑ Cabinets: $125+13+24=162$
 - Compute, communication, storage
 - ~750 m²



#2: *ORNL Titan Hybrid System (Cray XK7)*



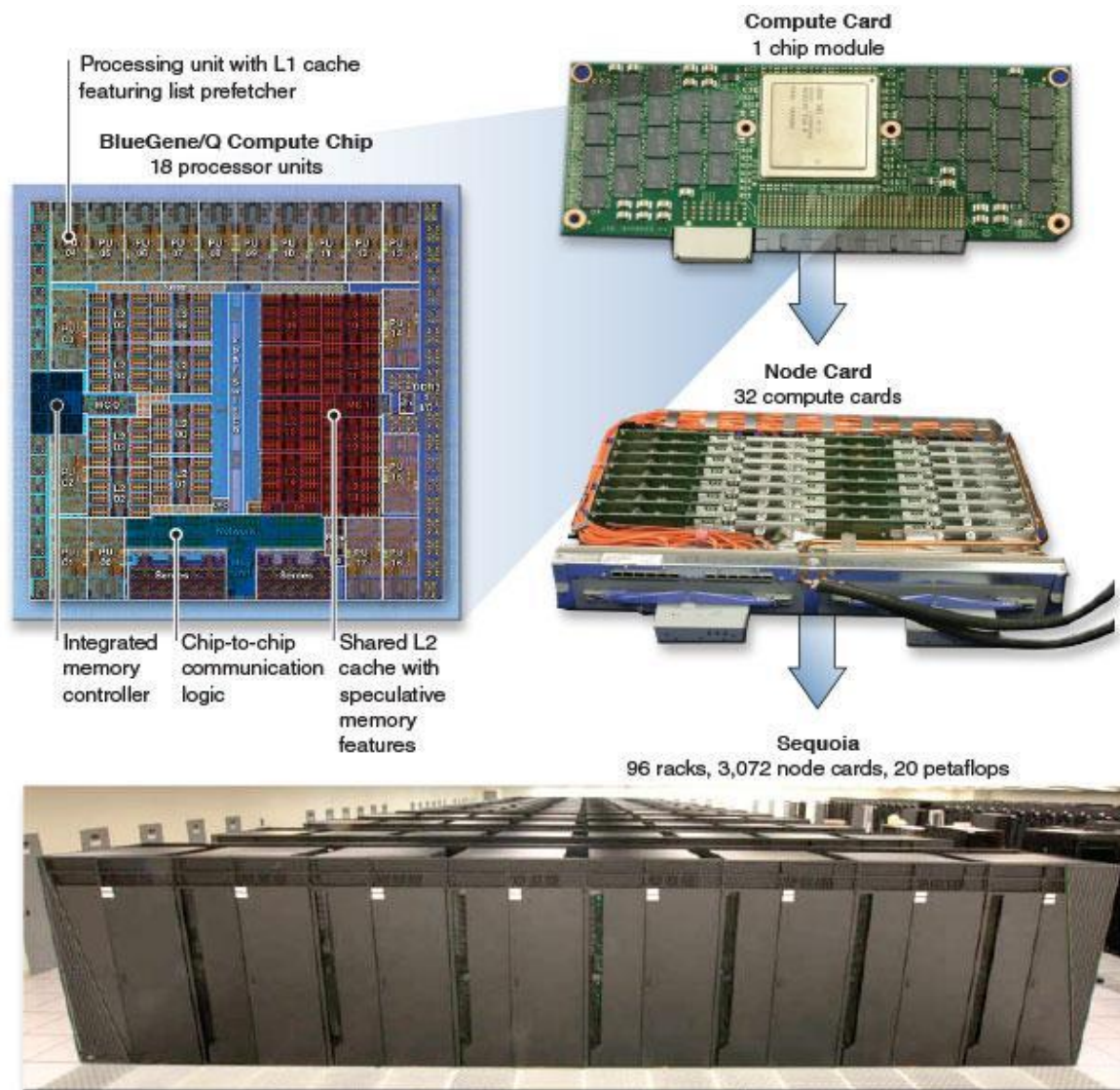
- ❑ Peak performance of 27.1 PF
 - 24.5 GPU + 2.6 CPU
- ❑ 18,688 Compute Nodes each with:
 - 16-Core AMD Opteron CPU
 - NVIDIA Tesla “K20x” GPU
 - 32 + 6 GB memory
- ❑ 512 Service and I/O nodes
- ❑ 200 Cabinets
- ❑ 710 TB total system memory
- ❑ Cray Gemini 3D Torus Interconnect
- ❑ 8.9 MW peak power



4,352 ft²

#3: *LLNL Sequoia (IBM BG/Q)*

- ❑ Compute card
 - 16-core PowerPC A2 processor
 - 16 GB DDR3
- ❑ Compute node has 98,304 cards
- ❑ Total system size:
 - 1,572,864 processing cores
 - 1.5 PB memory
- ❑ 5-dimensional torus interconnection network
- ❑ Area of 3,000 ft²



Contemporary HPC Architectures

Date	System	Location	Comp	Comm	Peak (PF)	Power (MW)
2009	Jaguar; Cray XT5	ORNL	AMD 6c	Seastar2	2.3	7.0
2010	Tianhe-1A	NSC Tianjin	Intel + NVIDIA	Proprietary	4.7	4.0
2010	Nebulae	NSCS Shenzhen	Intel + NVIDIA	IB	2.9	2.6
2010	Tsubame 2	TiTech	Intel + NVIDIA	IB	2.4	1.4
2011	K Computer	RIKEN/Kobe	SPARC64 VIIIfx	Tofu	10.5	12.7
2012	Titan; Cray XK6	ORNL	AMD + NVIDIA	Gemini	27	9
2012	Mira; BlueGeneQ	ANL	SoC	Proprietary	10	3.9
2012	Sequoia; BlueGeneQ	LLNL	SoC	Proprietary	20	7.9
2012	Blue Waters; Cray	NCSA/UIUC	AMD + (partial) NVIDIA	Gemini	11.6	
2013	Stampede	TACC	Intel + MIC	IB	9.5	5
2013	Tianhe-2	NSCC-GZ (Guangzhou)	Intel + MIC	Proprietary	54	~20

Top 10 (Top500 List, June 2011)

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak
1	RIKEN Advanced Inst for Comp Sci	K Computer Fujitsu SPARC64 VIIIfx + custom	Japan	548,352	8.16	93
2	Nat. SuperComputer Center in Tianjin	Tianhe-1A, NUDT Intel + Nvidia GPU + custom	China	186,368	2.57	55
3	DOE / OS Oak Ridge Nat Lab	Jaguar, Cray AMD + custom	USA	224,162	1.76	75
4	Nat. Supercomputer Center in Shenzhen	Nebulea, Dawning Intel + Nvidia GPU + IB	China	120,640	1.27	43
5	GSIC Center, Tokyo Institute of Technology	Tusbame 2.0, HP Intel + Nvidia GPU + IB	Japan	73,278	1.19	52
6	DOE / NNSA LANL & SNL	Cielo, Cray AMD + custom	USA	142,272	1.11	81
7	NASA Ames Research Center/NAS	Plelades SGI Altix ICE 8200EX/8400EX + IB	USA	111,104	1.09	83
8	DOE / OS Lawrence Berkeley Nat Lab	Hopper, Cray AMD + custom	USA	153,408	1.054	82
9	Commissariat a l'Energie Atomique (CEA)	Tera-10, Bull Intel + IB	France	138,368	1.050	84
10	DOE / NNSA Los Alamos Nat Lab	Roadrunner, IBM AMD + Cell GPU + IB	USA	122,400	1.04	76

Figure credit: <http://www.netlib.org/utk/people/JackDongarra/SLIDES/korea-2011.pdf>



Top 500 Top 10 (2006)

	Manufacturer	Computer	Rmax [TF/s]	Installation Site	Country	Year	#Proc
1	IBM	BlueGene/L eServer Blue Gene	280.6	DOE/NNSA/LLNL	USA	2005	131,072
2	Sandia/Cray	Red Storm Cray XT3	101.4	NNSA/Sandia	USA	2006	26,544
3	IBM	BGW eServer Blue Gene	91.29	IBM Thomas Watson	USA	2005	40,960
4	IBM	ASC Purple eServer pSeries p575	75.76	DOE/NNSA/LLNL	USA	2005	12,208
5	IBM	MareNostrum JS21 Cluster, Myrinet	62.63	Barcelona Supercomputing Center	Spain	2006	12,240
6	Dell	Thunderbird PowerEdge 1850, IB	53.00	NNSA/Sandia	USA	2005	9,024
7	Bull	Tera-10 NovaScale 5160, Quadrics	52.84	CEA	France	2006	9,968
8	SGI	Columbia Altix, Infiniband	51.87	NASA Ames	USA	2004	10,160
9	NEC/Sun	Tsubame Fire x4600, ClearSpeed, IB	47.38	GSIC / Tokyo Institute of Technology	Japan	2006	11,088
10	Cray	Jaguar Cray XT3	43.48	ORNL	USA	2006	10,424

Performance Development in Top 500

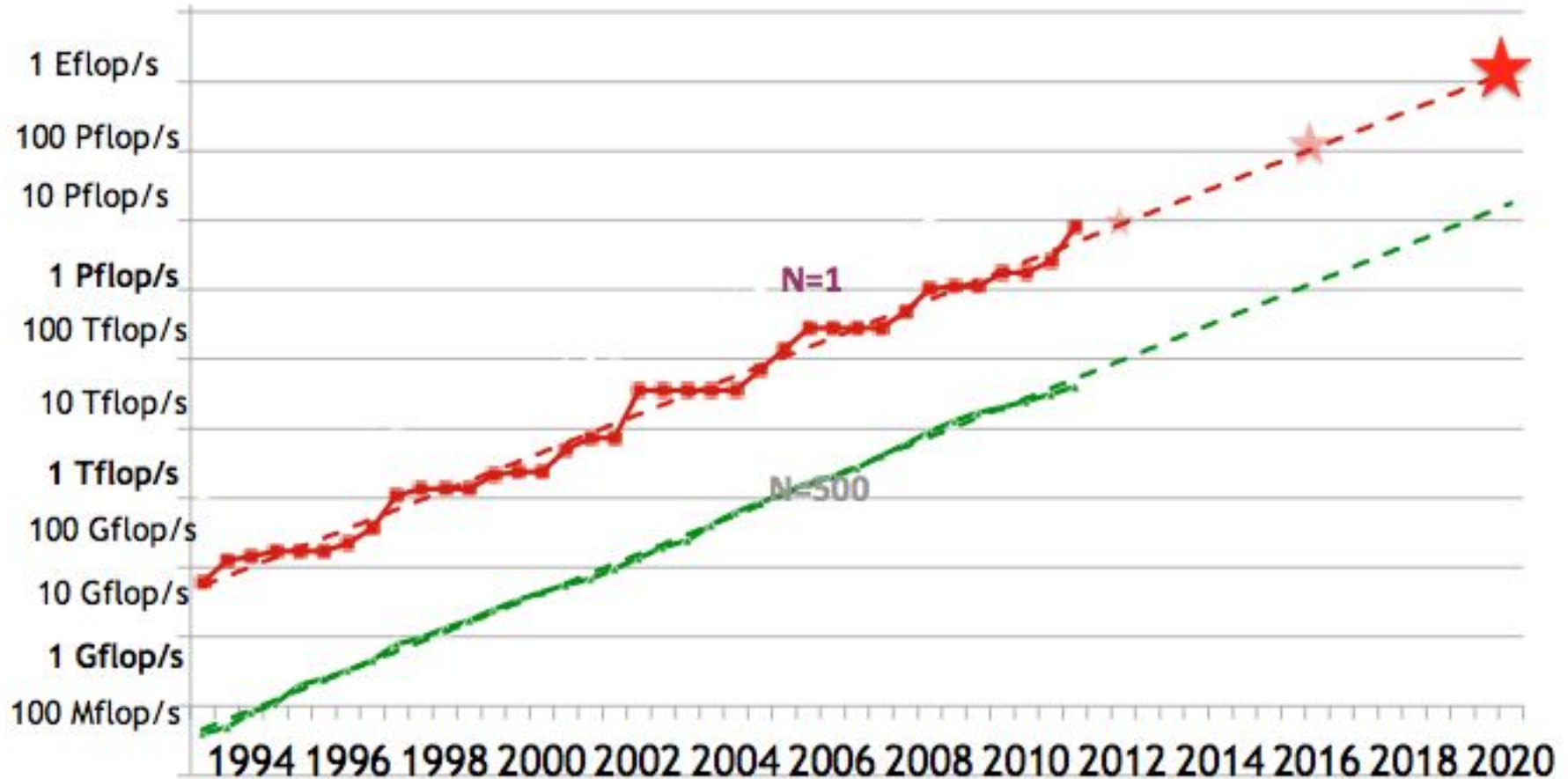


Figure credit: <http://www.netlib.org/utk/people/JackDongarra/SLIDES/korea-2011.pdf>

Exascale Initiative

- ❑ Exascale machines are targeted for 2019
- ❑ What are the potential differences and problems?

Systems	2011 K Computer	2019	Difference Today & 2019
System peak	8.7 Pflop/s	1 Eflop/s	O(100)
Power	10 MW	~20 MW	???
System memory	1.6 PB	32 - 64 PB	O(10)
Node performance	128 GF	1,2 or 15TF	O(10) - O(100)
Node memory BW	64 GB/s	2 - 4TB/s	O(100)
Node concurrency	8	O(1k) or 10k	O(100) - O(1000)
Total Node Interconnect BW	20 GB/s	200-400GB/s	O(10)
System size (nodes)	68,544	O(100,000) or O(1M)	O(10) - O(100)
Total concurrency	548,352	O(billion)	O(1,000)
MTTI	days	O(1 day)	- O(10)

Principais mudanças em software e algoritmos

- ❑ Com o quê nos preocupávamos antes e agora?
- ❑ Precisamos repensar o projeto para exascale
 - Movimento de dados é muito custoso (Por quê?)
 - Flops por segundo são baratos (Por quê?)
- ❑ Precisamos reduzir comunicação e sincronização
- ❑ Precisamos desenvolver algoritmos resilientes em caso de falhas
- ❑ Como lidar com paralelismo massivo?
- ❑ O software deve se adaptar ao hardware (autotuning)

Próximas aulas

- ❑ Arquiteturas de computadores paralelos
- ❑ Modelos de desempenho paralelo