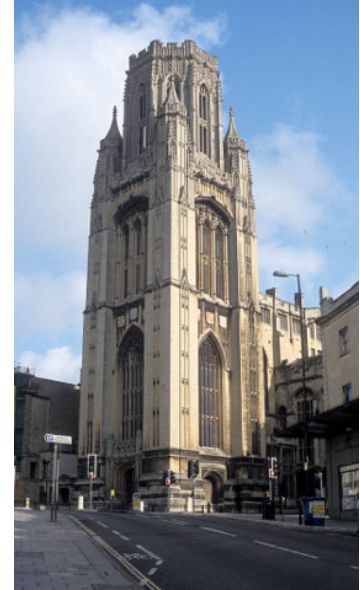


Heterogeneous Many-core Computing Trends: Past, Present and Future



Simon McIntosh-Smith
University of Bristol, UK



Agenda

- Important technology trends
- Heterogeneous Computing
- The Seven Dwarfs
- Important implications
- Conclusions

🌿 The real Moore's Law

Moore's Law graph, 1965

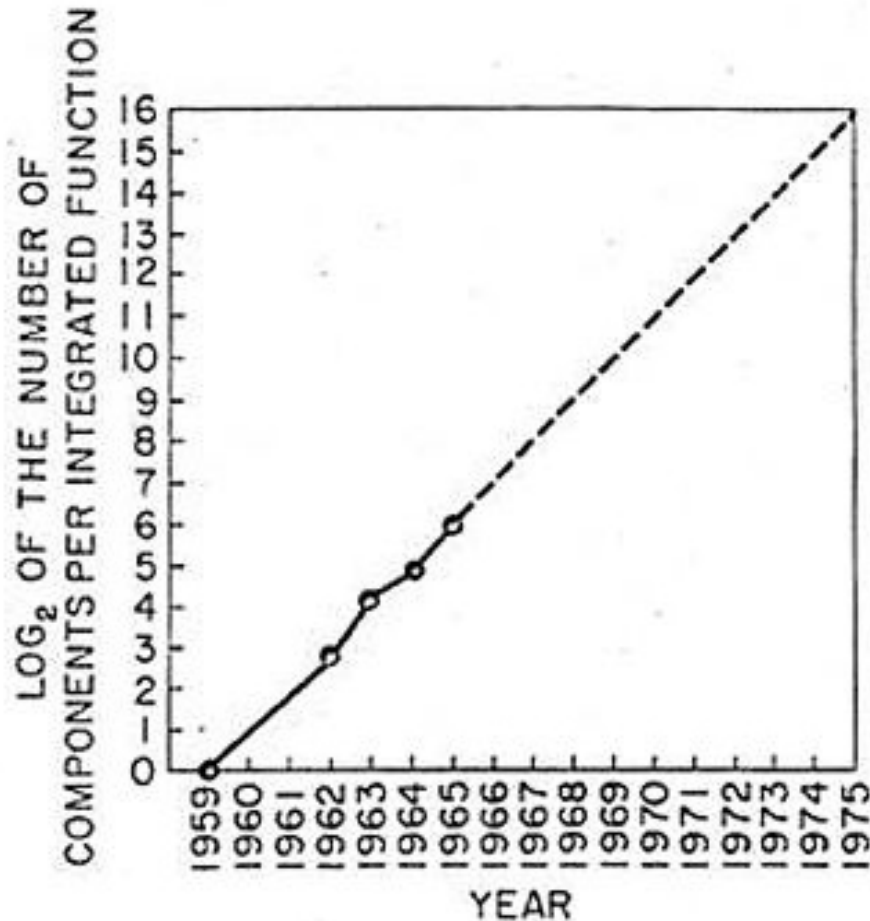
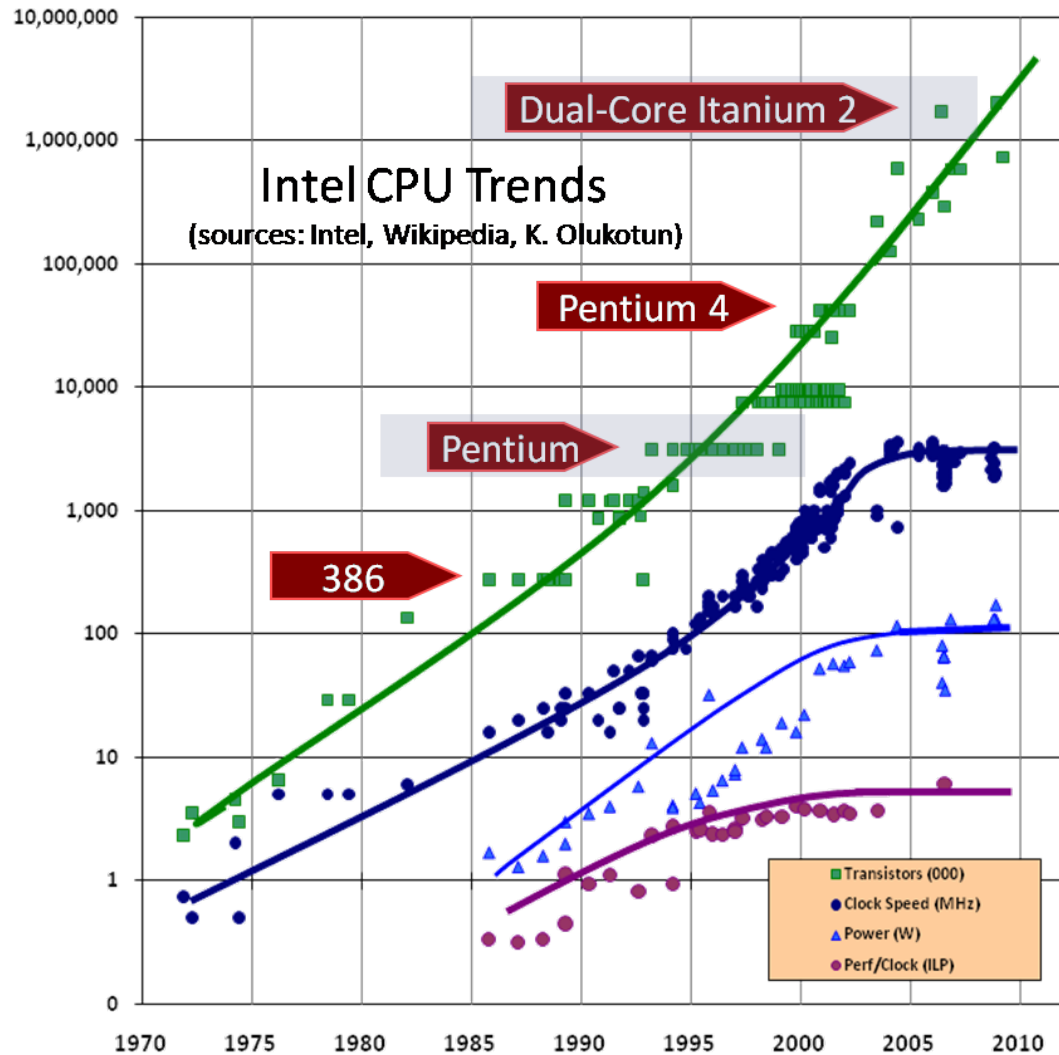


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

45 years ago, Gordon Moore observed that the number of transistors on a single chip was doubling rapidly

🌟 Important technology trends



The real Moore's Law

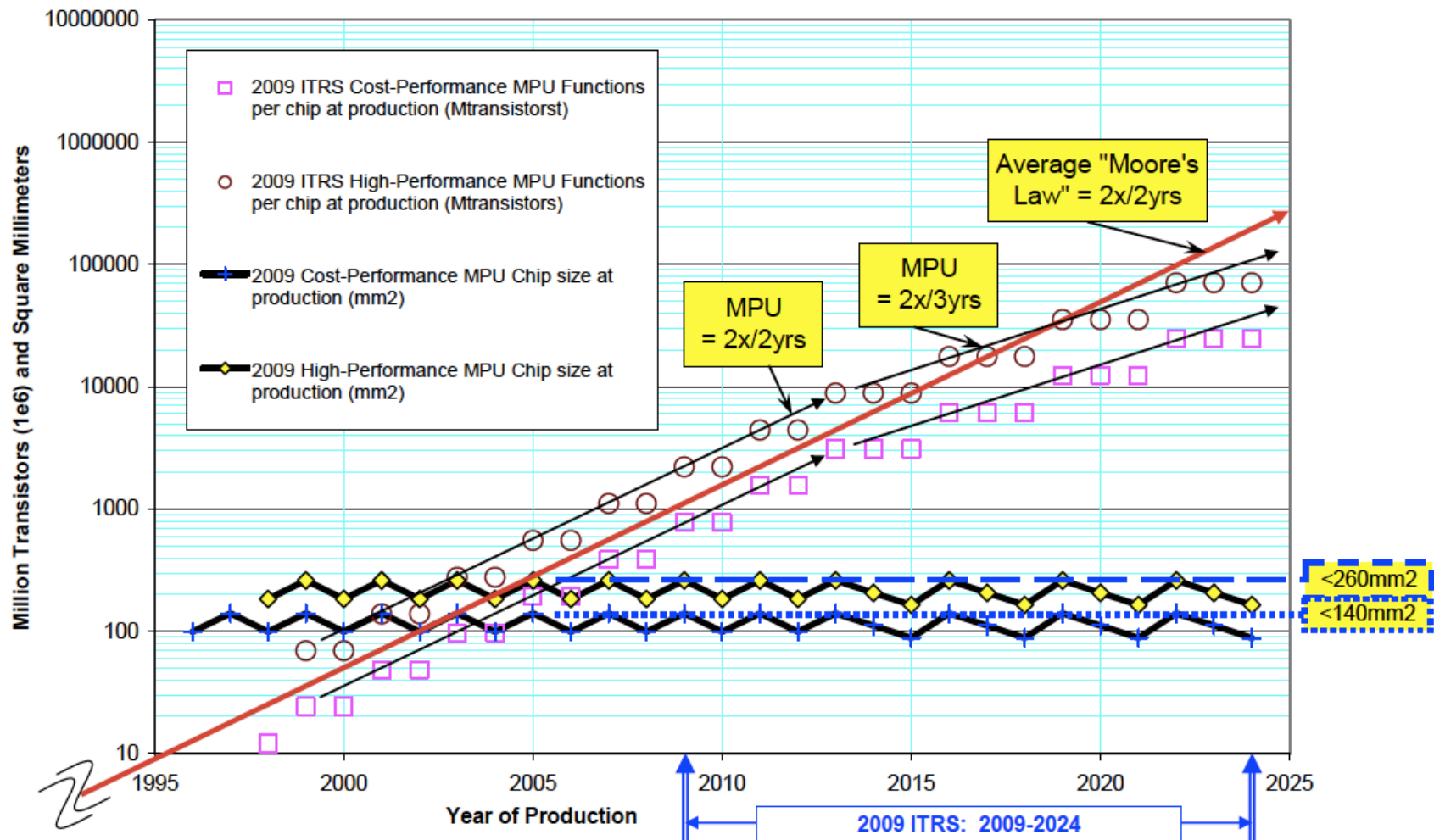
The clock speed plateau

The power ceiling

Instruction level parallelism limit

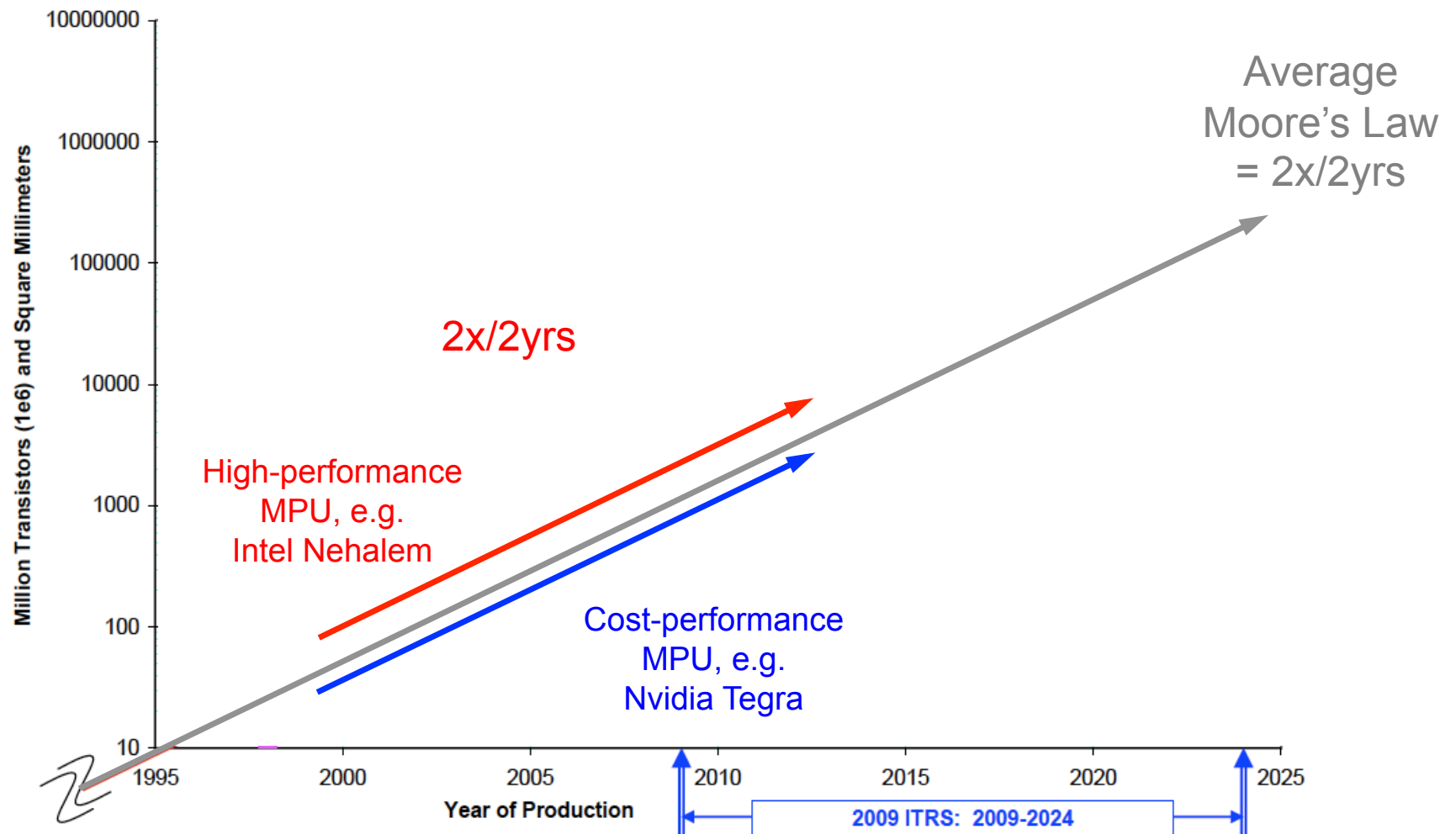
Moore's Law today

2009 ITRS - Functions/chip and Chip Size



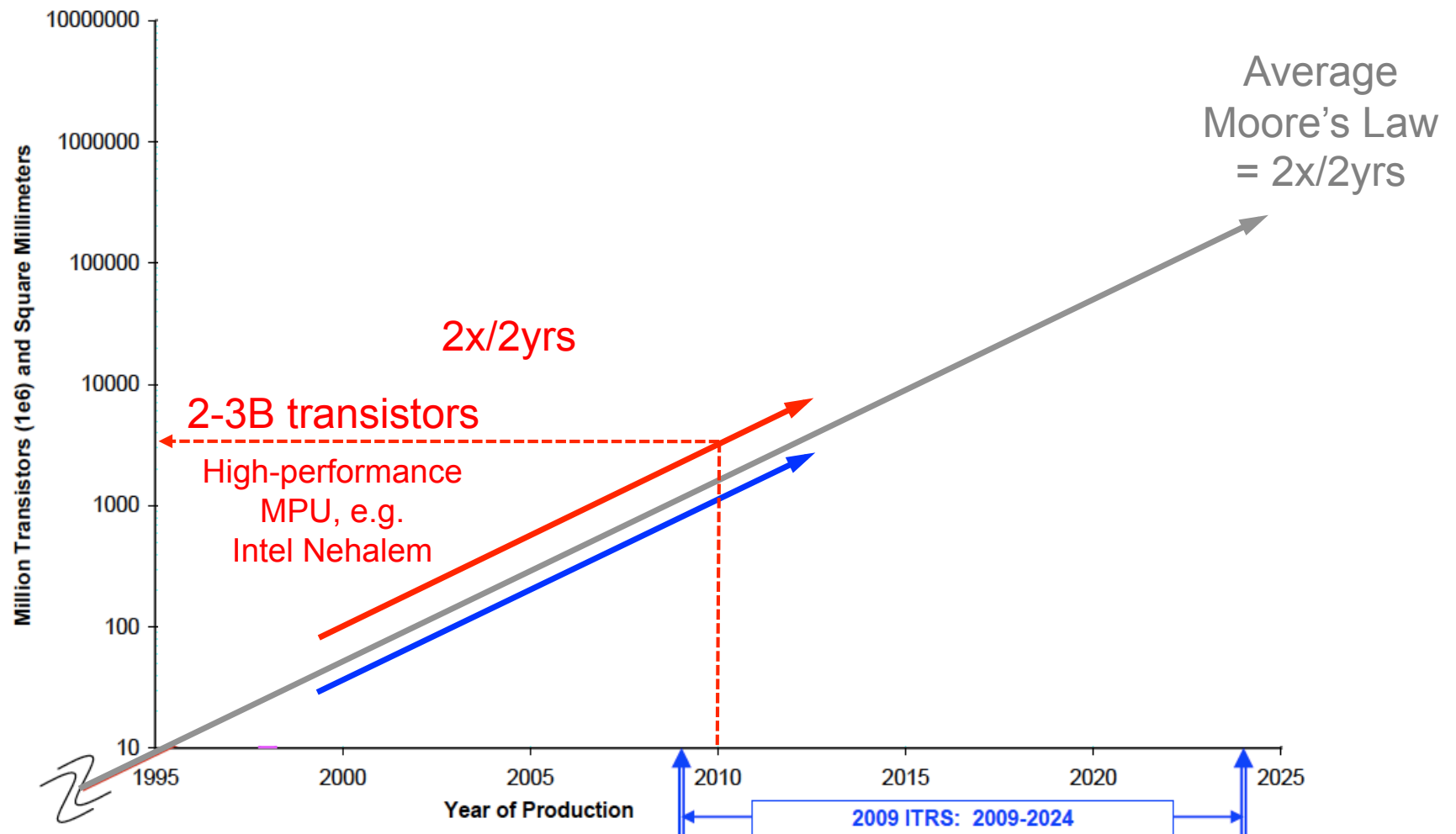
Moore's Law today

2009 ITRS - Functions/chip and Chip Size



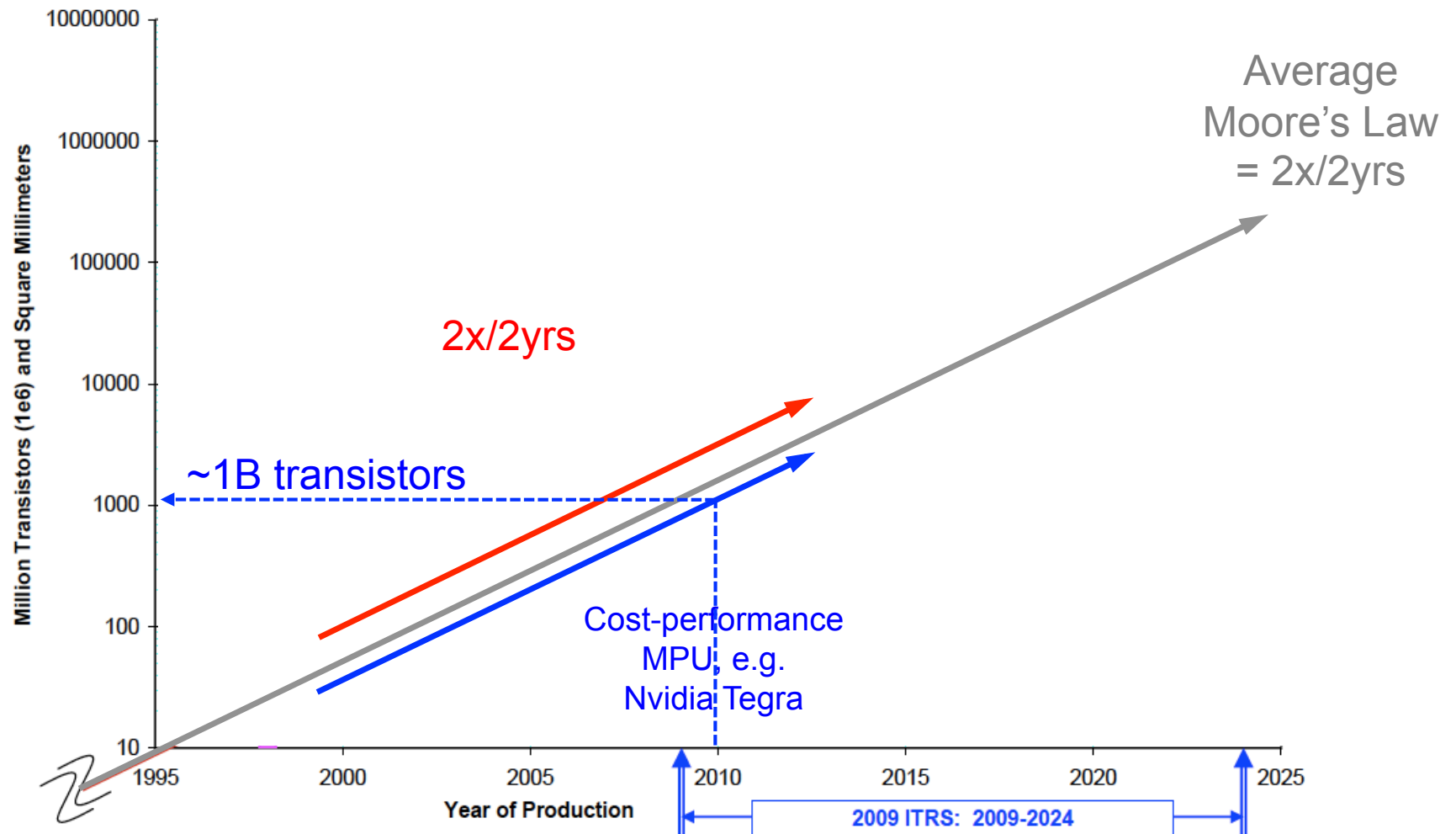
Moore's Law today

2009 ITRS - Functions/chip and Chip Size



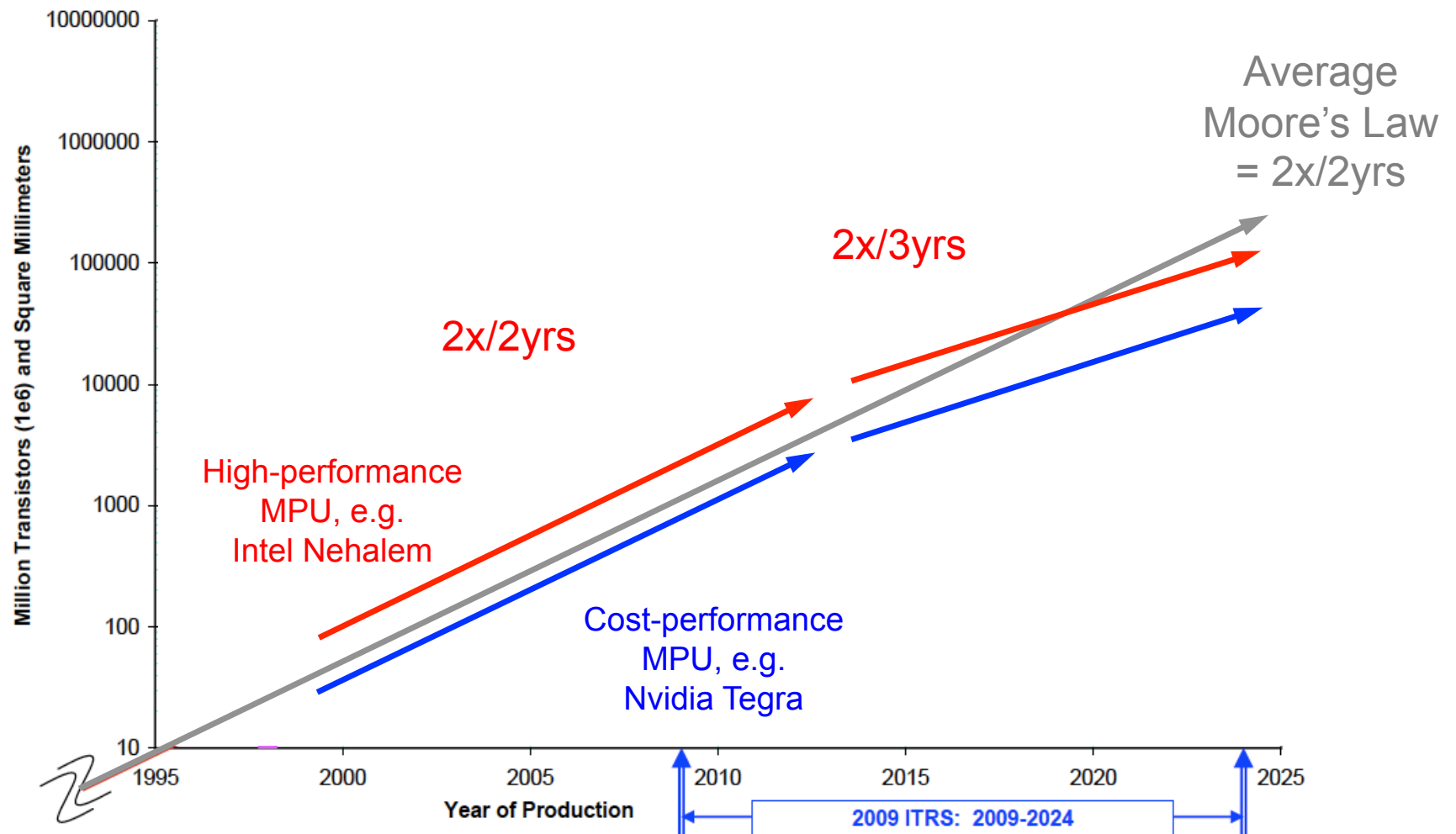
Moore's Law today

2009 ITRS - Functions/chip and Chip Size



Moore's Law today

2009 ITRS - Functions/chip and Chip Size



🔥 What to do with billions of transistors?

- Lots more cores on-chip
 - Core designs will stay roughly the same
- But power consumption must be held in check
 - Chip voltages can't be dialled down any more
 - Clock speeds may *decrease!*
 - Memory bandwidth per core may *decrease!*
 - Memory per core may *decrease!*
- Different types of cores
 - ***Heterogeneous computing!***
 - E.g. a few heavyweight (x86) cores together with many more lightweight (GPU) cores

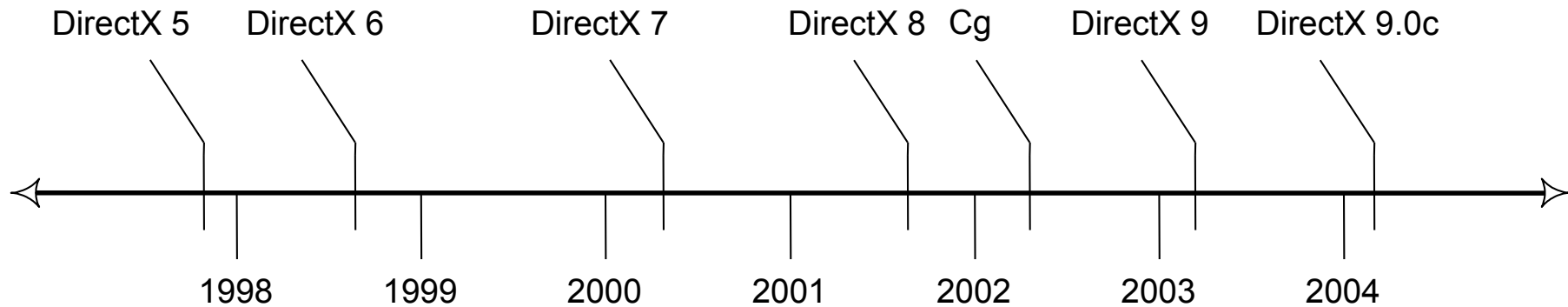
🔥 Heterogeneous computing is not new

- Most systems are *already* heterogeneous
 - PCs have CPU, GPU, network processor, I/O processor, ...
 - Has been a common approach in embedded systems since the early '90s

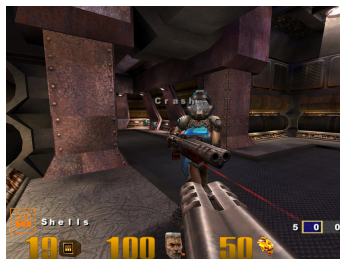


- But now heterogeneous systems are starting to include several different types of *general-purpose, programmable* processors
 - Users have to programme more than one type of processor to get the most out of a system

🔥 GPUs driven by advances in graphics APIs



Half-Life



Quake 3



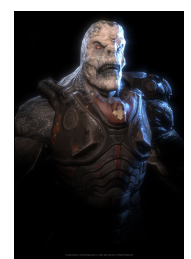
Giants



Halo



Far Cry



UE3

David Kirk and Wen-mei W. Hwu, 2007

🔥 Graphics API timeline

In 1999 DirectX 7 added simple programmable pixel shading


In 2003 DirectX 9 made this much more flexible

- Could write a general program
- Executed for every pixel
- Nearly unlimited number of interpolated inputs, texture lookups and math operations
- Enabled sophisticated calculations at every pixel
- Critically added ability to ***branch*** and ***execute floating point operations***



GPGPU computing

GPGPU (General-Purpose computation on Graphics Processing Units)

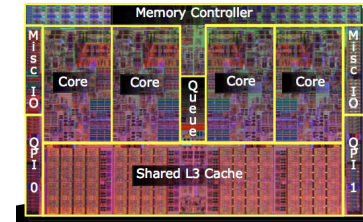
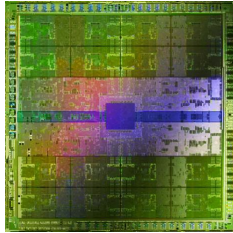
- Term first coined by Mark Harris in 2002
- <http://gpgpu.org/> 
- The first GPGPU applications were still graphics-oriented (ray tracing, video, ...)
- Also found early use in Seismic Processing
 - FFT intensive, a something GPUs are good at
- Early work also covered BLAS, PDEs, RNGs

🔥 From GPGPU to ...

Truly general purpose parallel processors

- Fully-fledged parallel languages such as Nvidia's Cuda started to appear in 2006
- GPUs started to add 64-bit floating point
- Remaining graphics-oriented limitations rapidly disappearing
- True High Performance Computing features about to appear in some GPUs, e.g. Nvidia's *Fermi*

🔥 Comparing Fermi and Nehalem



- 512 simple cores
 - ~3 billion transistors
 - ~1.5GHz
 - ~1,500 GFLOPS S.P.
 - ~750 GFLOPS D.P.
 - ~190 GBytes/s
 - IEEE 754-2008 support
 - ECC on all memories
- 4 complex cores
 - 731 million transistors
 - ~3GHz
 - 96 GFLOPS S.P.
 - 48 GFLOPS D.P.
 - ~30 GBytes/s
 - IEEE 754-1985 support
 - ECC on all memories

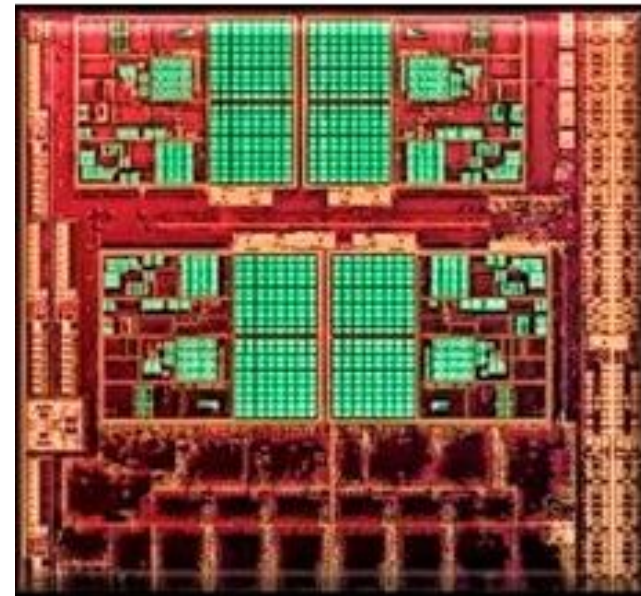
Future GPU architectures

- Tens of thousands of cores per chip
- Highly integrated (mainstream)
- Shared memory models
- Easier to use programming models

🌟 The future is now...

AMD's first "Fusion" chip, disclosed at ISSCC in San Francisco earlier this month

- 'Llano' Accelerated Processing Unit (APU)
- 32nm
- Integrates a quad core x86 CPU with a DirectX 11 capable GPU in the same chip



🔥 Emerging standards

- OpenCL, DirectCompute, ...



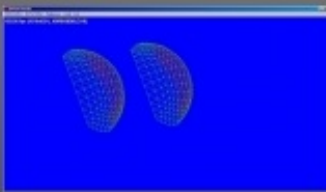
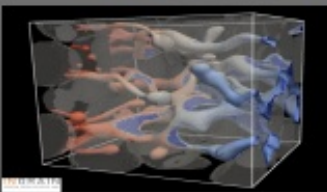


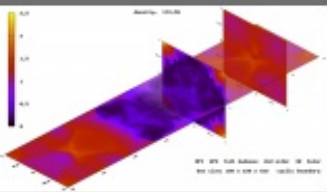
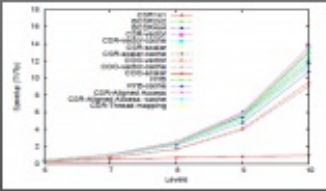

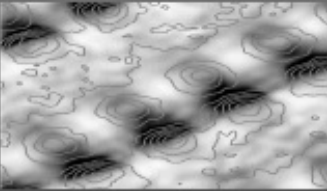


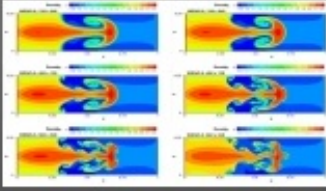


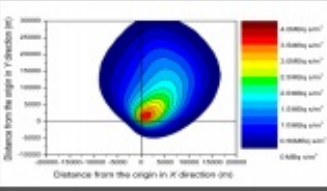
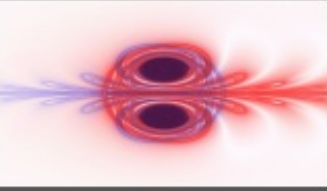
🔥 Heterogeneous systems in the Top500

- Tokyo Tech's TSUBAME was first in 2006
 - Started with ClearSpeed, now using GPUs
- Now several systems in existence, more on their way:
 - #2 is RoadRunner, the first PetaFLOP system
 - #5 is the Tianhe-1 System in China which delivers 563 TFLOPS from Intel x86 + AMD GPUs



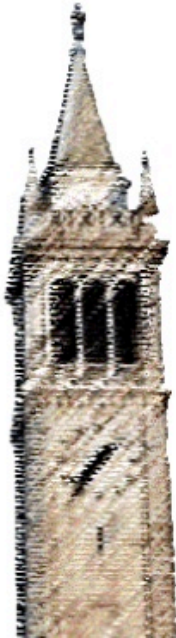
LATEST CUDA NEWS New NVIDIA GPU Technology Conference Content Posted Online



| | | | | |
|---|---|--|--|--|
|  <p>QView</p> |  <p>Multiphase flow in porous media</p> <p>100 x</p> |  <p>Graphic processing unit-accelerated mutual information-based 3D image registration</p> |  <p>Computing the Longest Common Transposition-Invariant Subsequence with GPU</p> |  <p>nHD</p> <p>173 x</p> |
|  <p>GPU based Sparse Grid Technique for Solving Multidimensional Options Pricing PDE</p> <p>1000 x</p> |  <p>Mersenne Twister for Graphic Processors (MTGP)</p> |  <p>Accelerating Geo-Science and Engineering System Simulations on Graphics Hardware</p> <p>30 x</p> |  <p>Towards a multi-GPU solver for the three-dimensional two-phase incompressible Navier-Stokes equations</p> <p>16 x</p> |  <p>GPU accelerated analysis of financial markets</p> <p>80 x</p> |
|  <p>Acceleration of a Finite-Difference WENO Scheme for Large-Scale Simulations on Many-Core Architectures</p> <p>50 x</p> |  <p>ClusterTech Financial Library in GPU</p> <p>30 x</p> |  <p>GPU-Assisted Surface Reconstruction on Locally-Uniform Samples</p> |  <p>Stochastic Lagrangian Particle Model for Air Pollution</p> <p>120 x</p> |  <p>Optimization of FTLE Calculation</p> <p>1000 x</p> |

The Seven Dwarfs

The Landscape of Parallel Computing Research: A View from Berkeley



*Krste Asanovic
Ras Bodik
Bryan Christopher Catanzaro
Joseph James Gebis
Parry Husbands
Kurt Keutzer
David A. Patterson
William Lester Plishker
John Shalf
Samuel Webb Williams
Katherine A. Yelick*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-183
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html>

December 18, 2006

- First described by Phil Colella at LBNL in 2004
- Expanded to 13 dwarfs by a group of researchers at Berkeley in 2006

🔥 What are the Seven Dwarfs?

Describe key algorithmic kernels in many scientific applications

1. Dense linear algebra – *BLAS, ScaLAPACK*
2. Sparse linear algebra – *SpMV, SuperLU*
3. Spectral methods – *FFT*
4. N-body methods – *Fast Multipole*
5. Structured grids – *Lattice Boltzmann*
6. Unstructured grids – *ABAQUS, Fluent*
7. Monte Carlo

Seven Heterogeneous Dwarfs

1. Dense linear – *excellent progress*

- MAGMA – see later talk
- FLAME – earlier talk at SIAM PP10
- Vendor libraries – CUBLAS, ACML, NAG, ...

2. Sparse linear algebra

- Iterative solvers – *good progress*
 - Nathan Bell and Michael Garland (NVIDIA Research) have general-purpose iterative solvers using efficient sparse matrix-vector multiplication
 - Andreas Klöckner (Brown University) has “Iterative CUDA” package based on same SpMV products
 - Manfred Liebmann & colleagues (University of Graz) have implemented algebraic multigrid

Seven Heterogeneous Dwarfs

3. Spectral methods – *good progress*

- FFT libraries from vendors
- “Auto-Tuning 3-D FFT Library for CUDA GPUs”
Akira Nukada, Satoshi Matsuoka, Tokyo Institute of Technology, SC09
 - Very fast, 160 GFLOPS for 256^3 32-bit 3D FFT

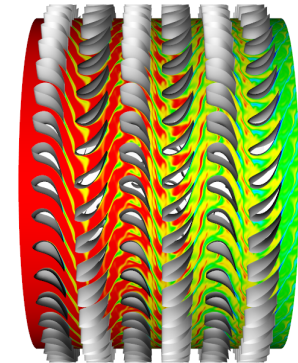
4. N-body methods – *excellent progress*

- NAMD/VMD – UIUC
- OpenMM, Folding@Home – Stanford
- Fast multipole methods - “42 TFlops Hierarchical N-body Simulations on GPUs with Applications in both Astrophysics and Turbulence”, Hamada et al, SC09

🔥 Seven Heterogeneous Dwarfs

5. Structured grids – *excellent progress*

- “Turbostream” turbulent fluid flow application framework, Pullan and Brandvik, Cambridge
 - 20X speedup
- Datta et al SC08
- Jonathan Cohen at NVIDIA Research developing a library called OpenCurrent



Seven Heterogeneous Dwarfs

6. Unstructured grids – *good progress*

- Several projects underway in the CFD community
- Rainald Löhner (GMU – Washington DC)
- Jamil Appa (BAE Systems)
- Graham Markell / Paul Kelly (Imperial)
- Mike Giles (Oxford) working with Markell, Kelly and others on a general-purpose, open-source framework called OP2
- Others underway

🔥 Seven Heterogeneous Dwarfs

7. Monte Carlo – *excellent progress*

- Massively parallel, an excellent fit
- Vendors providing examples
- Mike Giles (Oxford) working with NAG to develop a GPU library of RNG routines
 - E.g. mrg32k3a and Sobol generators
 - <http://www.nag.co.uk/numeric/GPUs/>
- Lots of work in this space

🔥 Summing up GPU experiences

- There's been a lot of hype
- Speed-ups greater than 10X should be viewed with suspicion
 - The hardware is “only” ~10X faster after all...
- But real progress now being made on the Seven Dwarfs
- ***Higher level application templates, libraries and auto-tuners will be essential!***

🌟 Important takeaways

- Heterogeneous computing is here to stay
- Even single chips will contain tens of thousands of cores
- It is ***crucial*** that anyone developing software is aware of this!
- Design your software to scale on future systems, not limited to the parallelism of the past (Petascale servers only 13 years away...)

