Introduction to GPU Accelerated Computing: 1. History of Computer Architecture Many-Core, GPU, and other ideas...

University

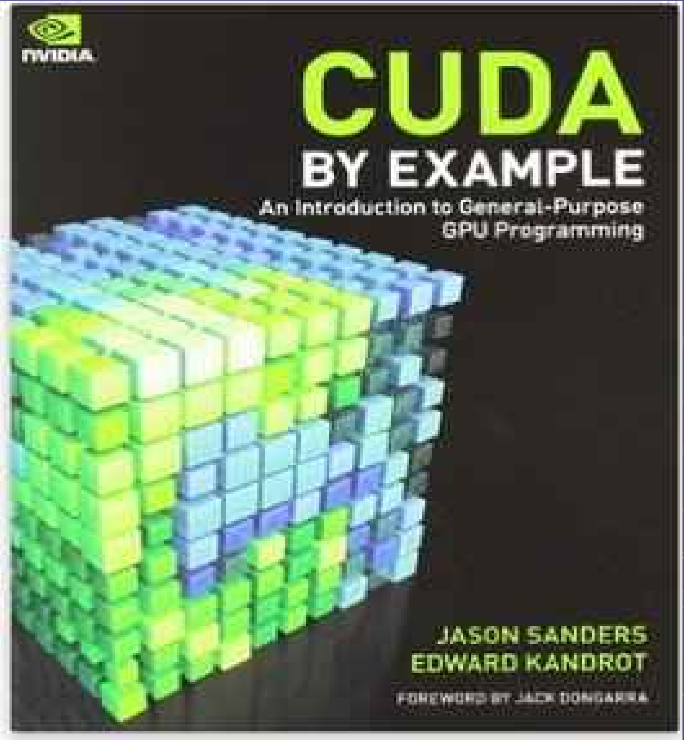
Rainer Spurzem
Astronomisches Rechen-Inst., ZAH, Univ. of Heidelberg, Germany
National Astronomical Observatories (NAOC), Chinese Academy of Sciences
Kavli Institute for Astronomy and Astrophysics (KIAA), Peking University

spurzem@nao.cas.cn
http://silkroad.bao.ac.cn

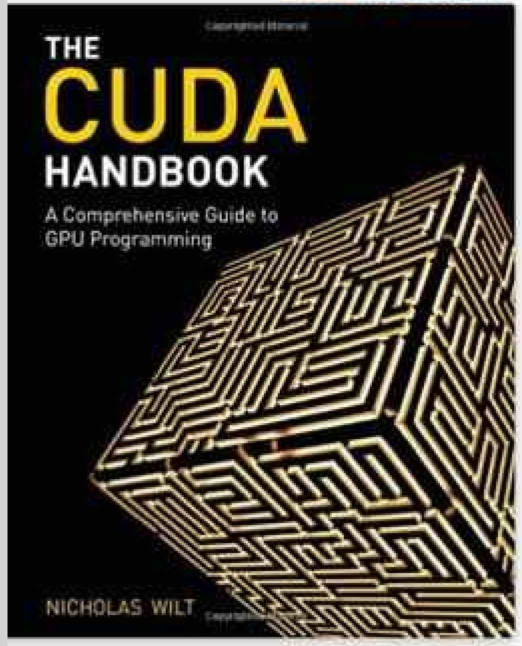
Introduction to GPU Accelerated Computing
July 25-28, 2016

Table of Contents (subject to change):

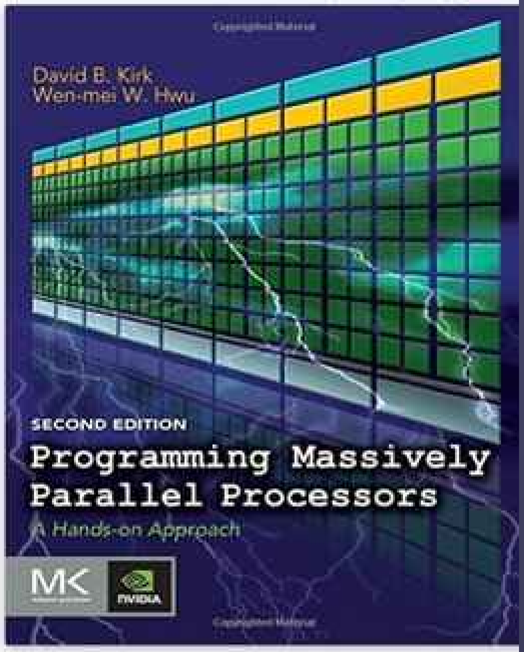
Literature



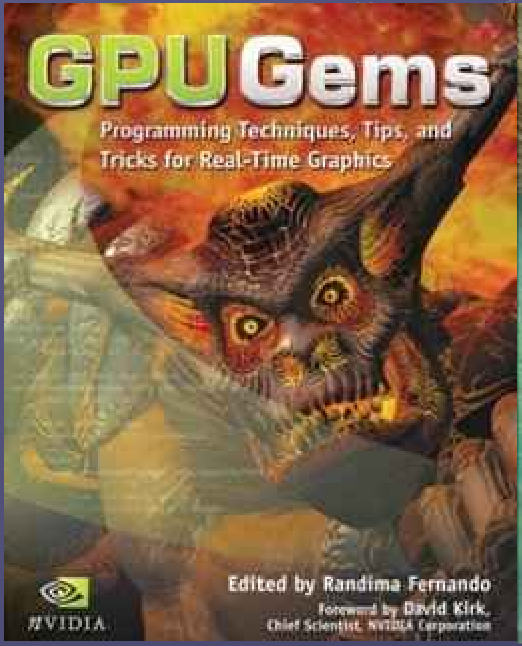
NVIDIA
CUDA
BY EXAMPLE
An Introduction to General-Purpose GPU Programming
JASON SANDERS
EDWARD KANDROT
FOREWORD BY JACK DONGARRA



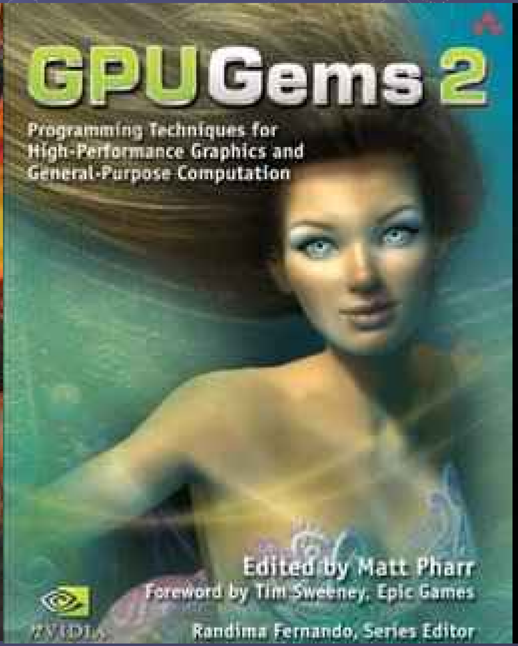
THE
CUDA
HANDBOOK
A Comprehensive Guide to GPU Programming
NICHOLAS WILT



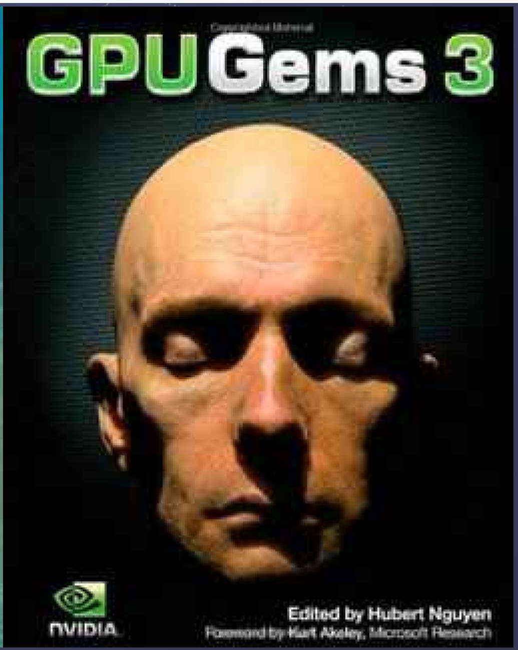
David B. Kirk
Wen-mei W. Hwu
SECOND EDITION
Programming Massively Parallel Processors
A Hands-on Approach
MK
NVIDIA



GPU Gems
Programming Techniques, Tips, and Tricks for Real-Time Graphics
Edited by Randima Fernando
Foreword by David Kirk, Chief Scientist, NVIDIA Corporation
NVIDIA



GPU Gems 2
Programming Techniques for High-Performance Graphics and General-Purpose Computation
Edited by Matt Pharr
Foreword by Tim Sweeney, Epic Games
Randima Fernando, Series Editor
NVIDIA



GPU Gems 3
Edited by Hubert Nguyen
Foreword by Kurt Akeley, Microsoft Research
NVIDIA

Observations (Experiment)

Theory

Computational Physics

History

Erik Holmberg (1908-2000)

Dissertation Univ. Lund (Schweden) (1937):

``A study of double and multiple galaxies´´

Galaxies often in Groups and Pairs

Irregular Distribution of Satellite Galaxies

(Holmberg-Effect)

Father of numerical astrophysics?

» **...with 200 light bulbs**

Geschichte

The Astrophysical Journal, Nov. 1941

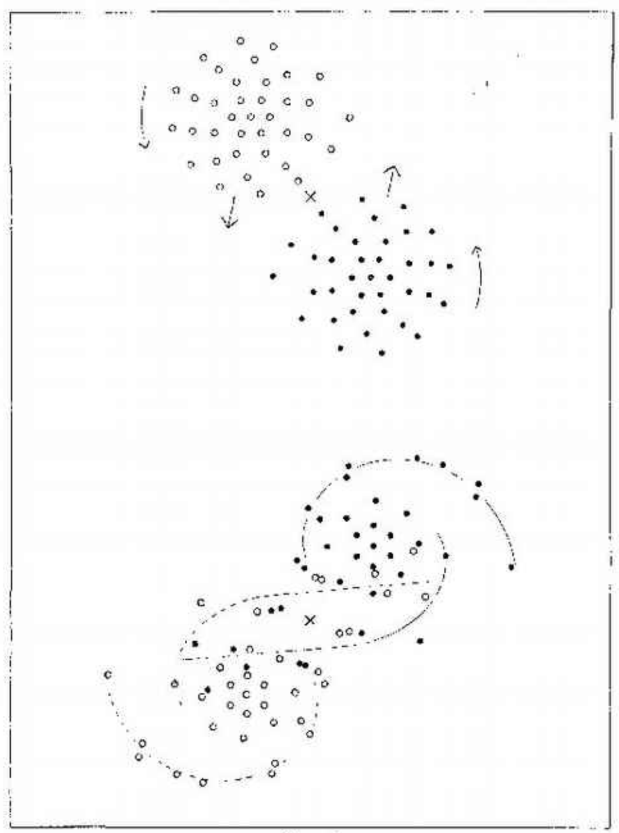


Fig. 4b

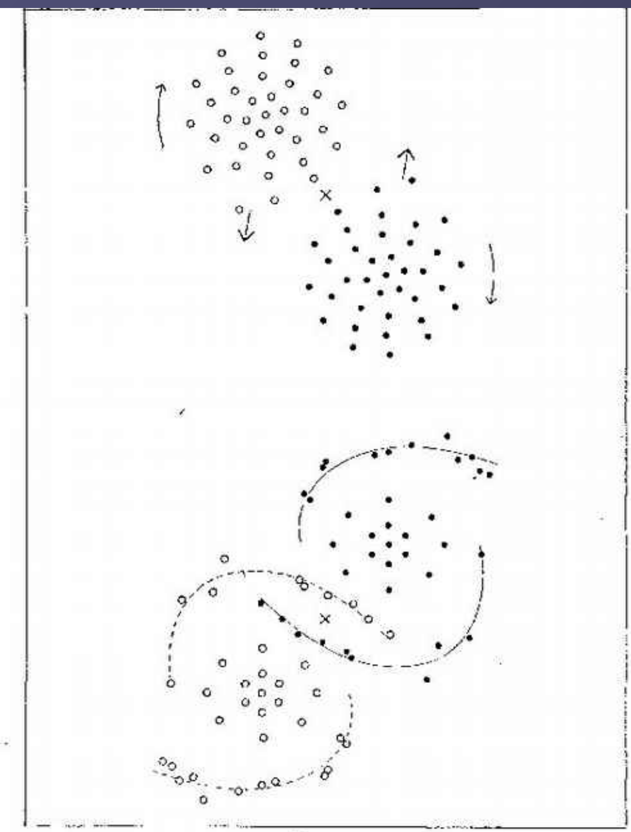


Fig. 4a

HARDWARE

...before von Neumann...

- Konrad Zuse (1910-1995) Berlin

Invented freely programmable Computer

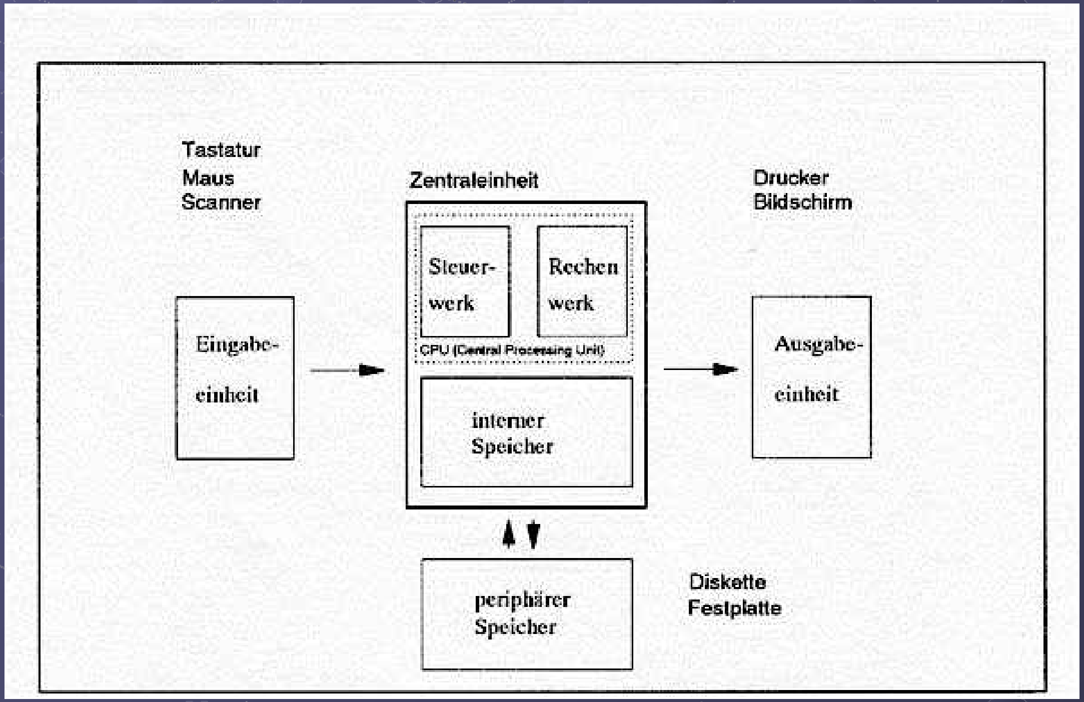
Z1 in parental flat 1936

HARDWARE

- John von Neumann (1903-1957)

Born Budapest, Lecturer Berlin, since 1930 Princeton Univ.

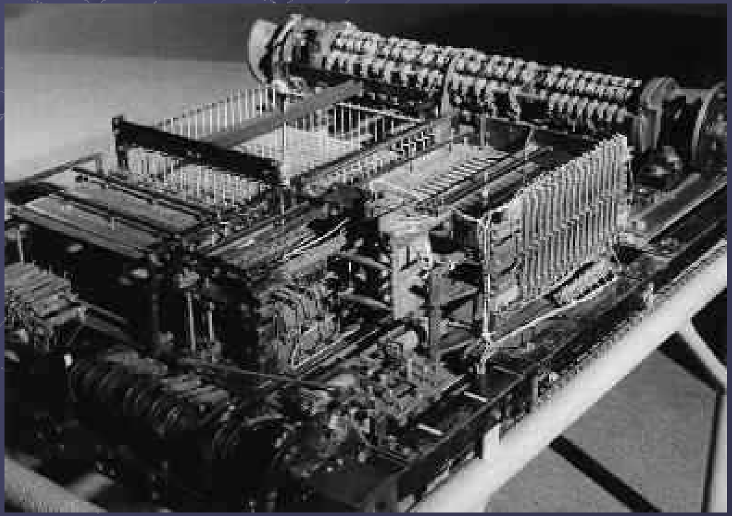
Requirements for the Construction of an electronic computing device(1946)



Computer

Zuse Z4: 1944 Berlin, 1950 Zürich, 1954 Frankreich

1959 Deutsches Museum München



Computing Speed 0.03 MHz

Memory 256 byte

Astronomisches Rechen-Institut (ARI) at Univ. of Heidelberg, Germany

Siemens 2002 Computer in 1964 At ARI

History

Astronomisches Rechen-Institut in Heidelberg
Mitteilungen Serie A Nr. 14

Die numerische Integration des *n*-Körper-Problemes für Sternhaufen I

Von

SEBASTIAN VON HOERNER

Mit 3 Textabbildungen

(Eingegangen am 10. Mai 1960)

Tabelle 5. *Zahl der gegenseitigen Umläufe, Häufigkeit des Auftretens und kleinster gegenseitiger Abstand D_m der engsten Paare.* (Alle engsten Paare mit mehr als zwei vollen Umläufen wurden notiert)

Umläufe	Häufigkeit	D_m
2—3	11	0.0102
3—5	9	0.0177
5—10	5	0.0070
10—20	2	0,0141
20—50	1	0.0007
50—100	1	0.0035
100—200	1	0.0039

Astronomisches Rechen-Institut in Heidelberg
Mitteilungen Serie A Nr. 19

Die numerische Integration des *n*-Körper-Problems für Sternhaufen, II.

Von

SEBASTIAN VON HOERNER

Mit 10 Textabbildungen

(Eingegangen am 19. November 1962)

S.v. Hoerner, Z.f.Astroph. 1960, 63

Siemens 2002
N=4,8,12,16 (4 Trx)

N=16,25 (40 Trx)

Computer

- ``The father of supercomputing

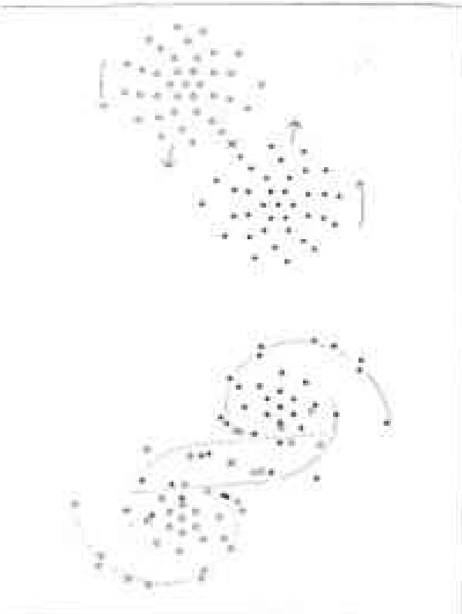
- Seymour Cray (1925-1996)

CRAY1: Vectorregisters (1976)
160 Mflop, 80 MHz, 8 MByte RAM
CRAY2: (1984)
1Gflop, 120MHz, 2GByte RAM

Computer

Supercomputer
JUGENE
IBM Blue Gene
At FZ Jülich,
Germany

Opening Ceremony June 2008



Holmberg, 1937/1941

NGC 4038/NGC 4039

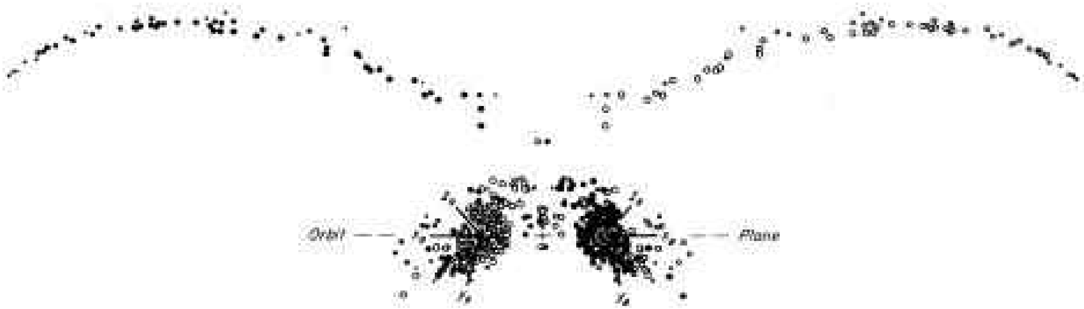
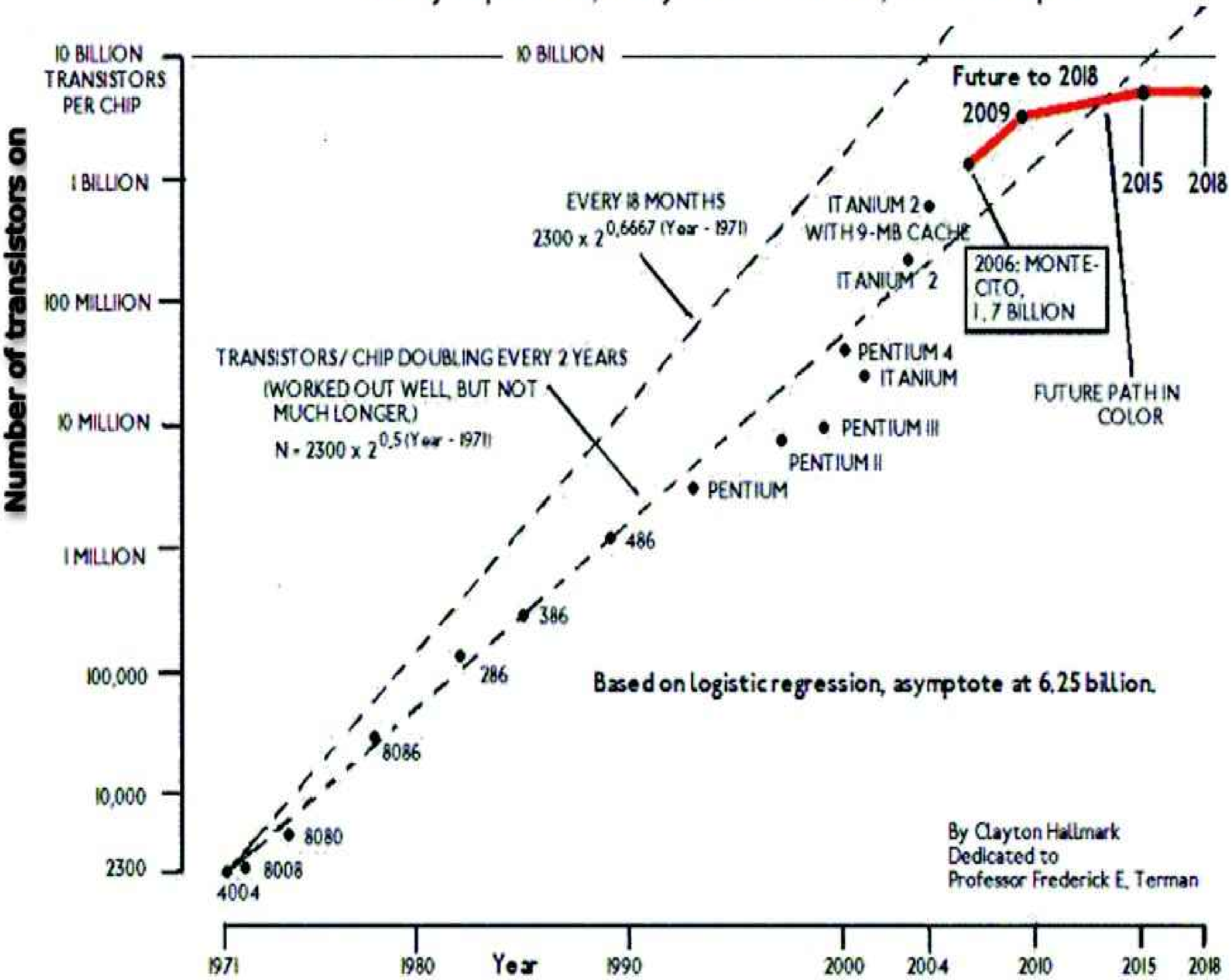


FIG. 23.—Symmetric model of NGC 4038/9. Here two identical disks of radius $0.75R_{min}$ suffered an $e \approx 0.5$ encounter with orbit angles $i_n = i_s = 60°$ and $\omega_n = \omega_s = -30°$ that appeared the same to both. The above all-inclusive views of the debris and remnants of these disks have been drawn exactly normal and edge-on to the orbit plane; the latter viewing direction is itself 30° from the line connecting the two pericenters. The viewing time is $t = 15$, or slightly past apocenter. The filled and open symbols again disclose the original loyalties of the various test particles.

Toomre & Toomre,1972, ApJ, 178, 623

Moore's Law Ending (Red Line):
Delayed products, Delayed 45nm / 32 nm, Reduced Capex



Computational Science...

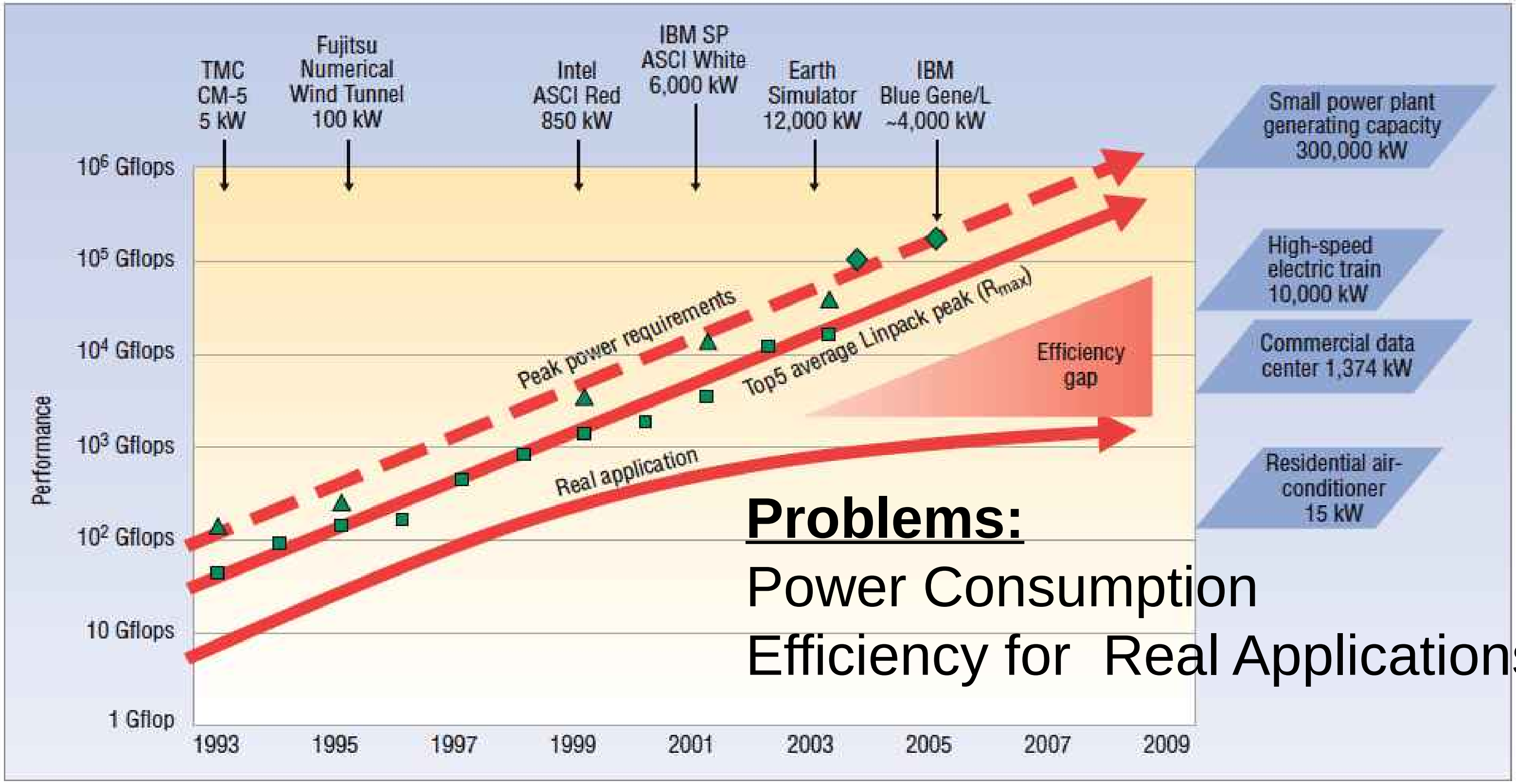
...after von Neumann...

Exaflop/s?

Petaflop/s

Teraflop/s

Gigaflop/s



Problems:

Power Consumption

Efficiency for Real Applications

Figure 1. Rising power requirements. Peak power consumption of the top supercomputers has steadily increased over the past 15 years.

Thanks to Horst Simon, LBNL/NERSC for this diagram.

SPECIAL HARDWARE

CPUs
Central Processing Units

General Purpose oriented

1-12 Cores

Up to 4 pipes per core using Vector Units

Fully Programmable, many languages available

Very well studied

Max. 125W per processor

GPUs
Graphic Processing Units

Graphics oriented

16-512 Cores

Massively Parallel Architecture, specialized instructions for parallel processing

Fully programmable, but limited languages

Algorithms not fully explored

Max. 400W per card

FPGAs
Field Programmable Gate Arrays

Custom designs, best for processing streaming data

Programmable Logic, Architecture is custom-built for the required application

Requires extensive knowledge to program, development time is longer than CPUs and GPUs

Application interface is custom built on each case

Max. 60W per FPGA

ASICs
Application Specific Integrated Circuits

Fully custom designs, built for a specific application

Not flexible, cannot be changed once it is built

Development is even more specialized than FPGAs

Power consumption varies with the application, usually best performance per Watt

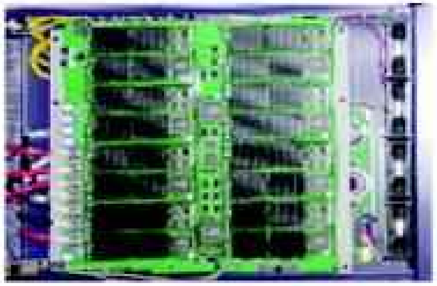
Slide: Guillermo Marcus

HARDWARE

GRAPE-6 Gravity/Coulomb Part

- G6 Chip: 0.25μ 2MGate ASIC, 6 Pipelines
- at 90MHz, 31Gflops/chip
- 48Tflops full system (March 2002)
- Plan up to 72Tflops full system (in 2002)
- Installed in Cambridge, Marseille, Drexel, Amsterdam, New York (AMNH), Mitaka (NAO), Tokyo, etc.. **New Jersey, Indiana, Heidelberg**

GRAPE-6



1998, 120 Gflops

Developers: Junichiro Makino, Toshiyuki Fukushige, Hiroshi Daisaka, Eiichiro Kokubo, Masaki Koga, Makoto Taiji, Ken Namura

GRAPE-6: Massively-Parallel Special-Purpose Computer for Astrophysical Particle Simulations

Sales Information

The Green500 List - November 2010

Listed below are the November 2010 The Green500's energy-efficient supercomputers ranked from 1 to 100.

http://www.green500.org

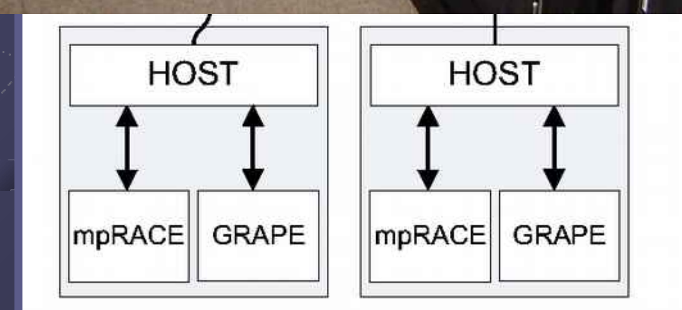
Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	1684.20	IBM Thomas J. Watson Research Center	NNSA/SC Blue Gene/Q Prototype	38.80
2+	1448.03	National Astronomical Observatory of Japan	GRAPE-DR accelerator Cluster, Infiniband	24.59
2	958.35	GSIC Center, Tokyo Institute of Technology	HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows	1243.80
3	933.06	NCSA	Hybrid Cluster Core i3 2.93Ghz Dual Core, NVIDIA C2050, Infiniband	36.00

VolkswagenStiftung

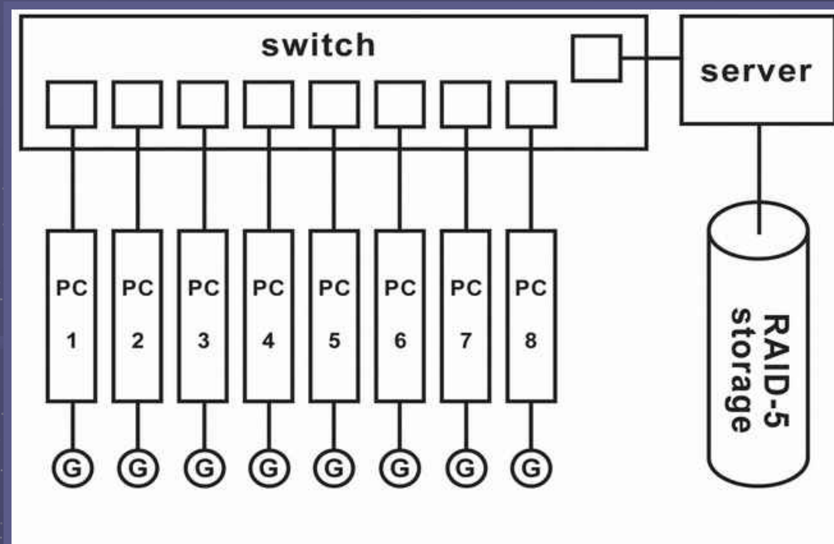
GRACE Cluster

4 Tflops (32 micro-GRAPE6)
Dual Port Infiniband
4 MPRACE-1 reconfigurable
(soon: 32 MPRACE-2)

GRAPE + MPRACE
= GRACE



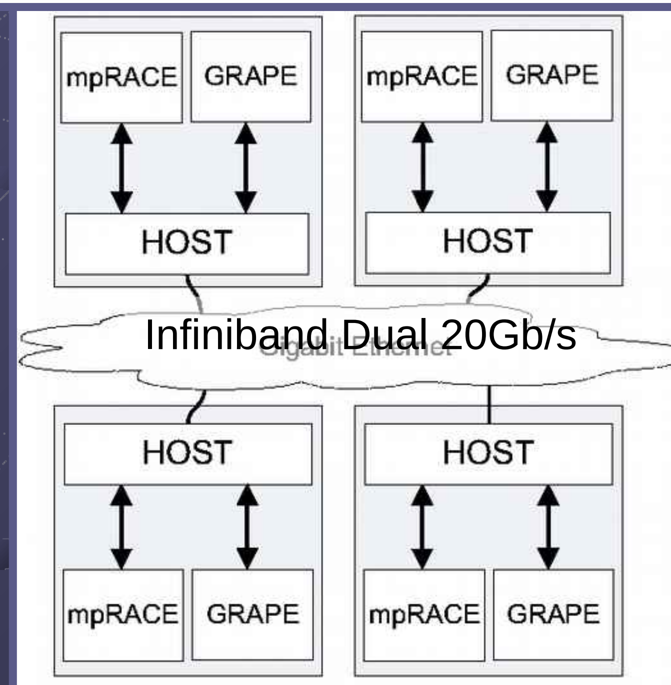
FPGA...



Heidelberg titan 32 node cluster

GRACE = GRAPE + MPRACE

- 32 dual-Xeon 3.2 GHz nodes
- 32 GRAPE6a
- 32 FPGA
- 7 TB RAID
- Dual port Infiniband link (20 Gb/s)
- Speed: ~4 Tflops
- N up to 4M
- Cost: ~380K EUR
- Funding: Volkswagen/Baden-Württemberg

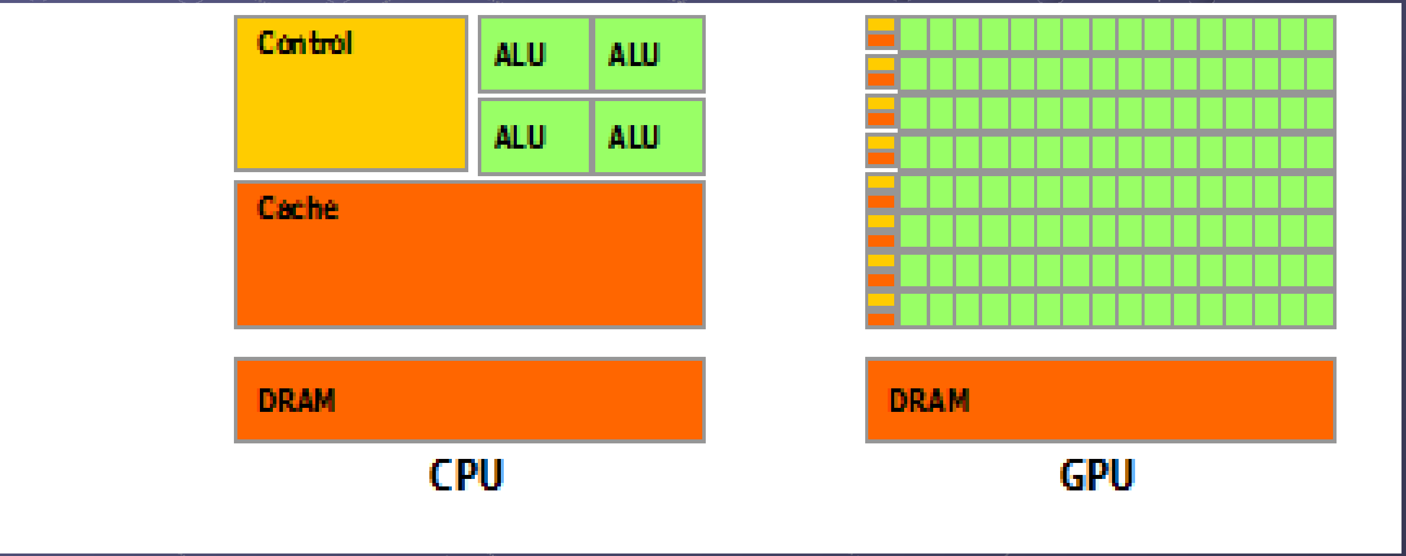


Graphics Processors (GPU) as General Purpose Supercomputers (GPGPU)

2008…
GeForce 9800 GTX, 128 Stream Proc., 512 MB
GeForce 9800 GX2, 256 Stream Proc., 1 GB
GeForce 9800 GT, 64 Stream Proc., 512 MB
[…]
2009: Tesla ~200 Proc., 4GB
2010: Fermi ~400 Proc., 4GB
2013: Kepler K20, ~2500 Procs., 6GB

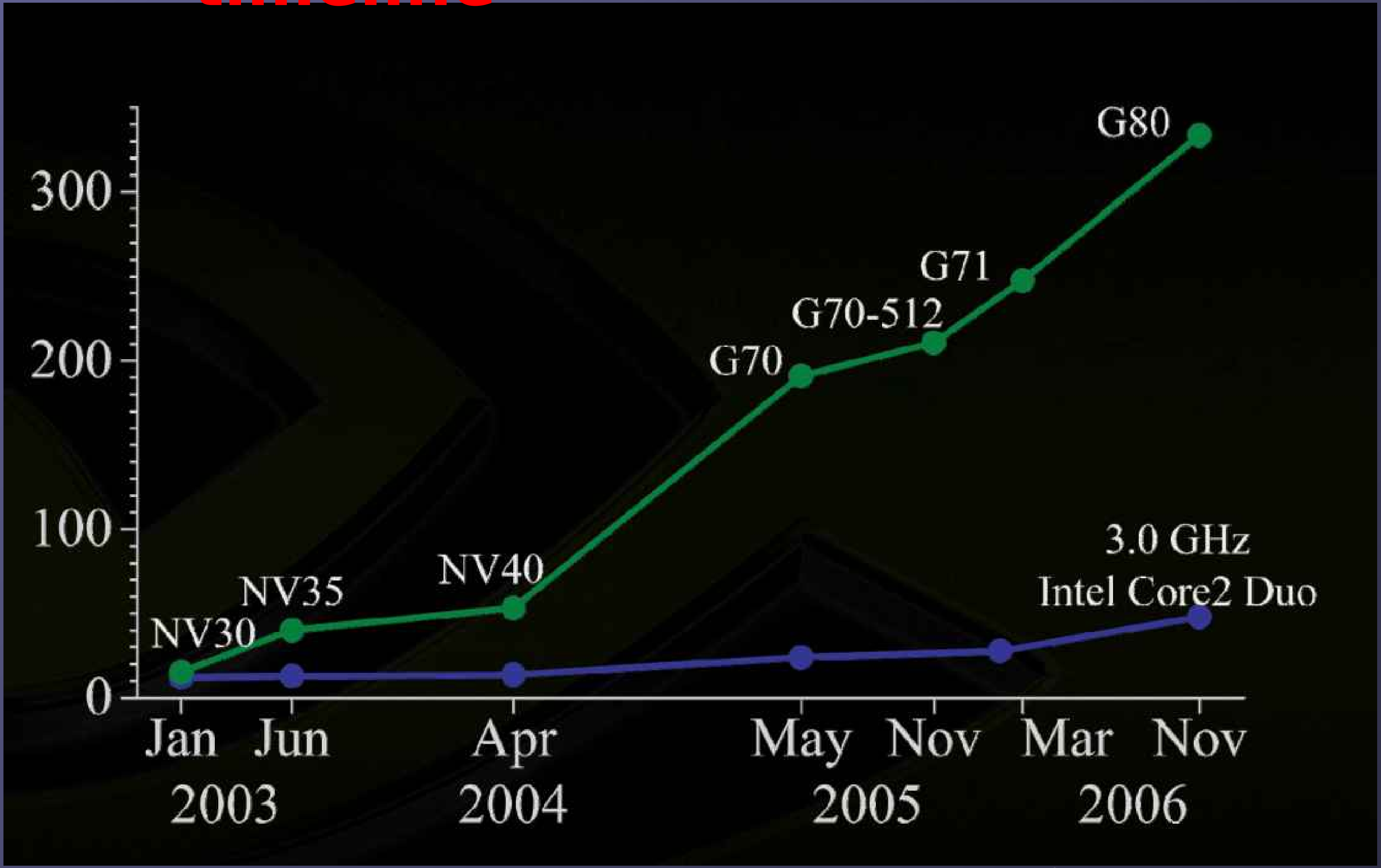
CPU and GPU; from CUDA NVIDIA Developer Zone at

http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

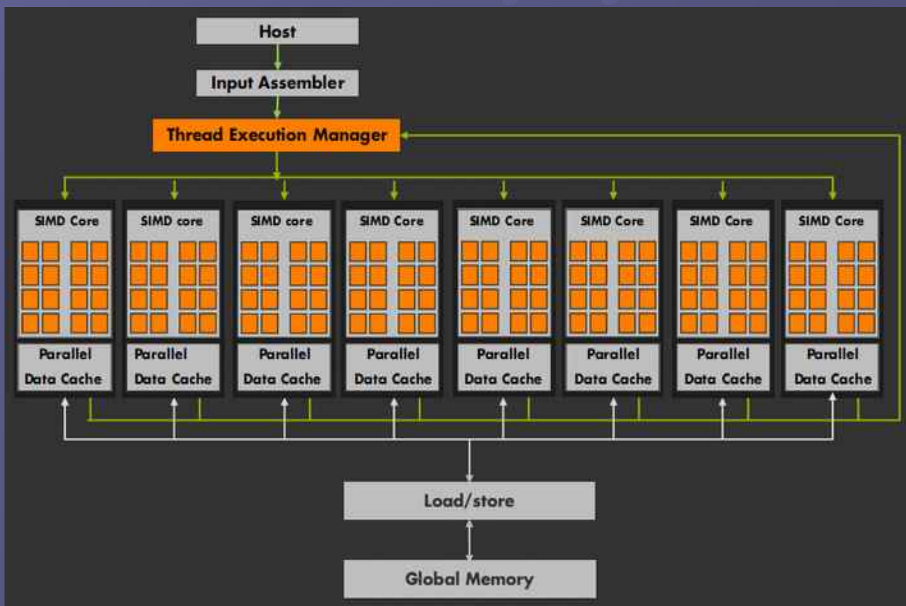


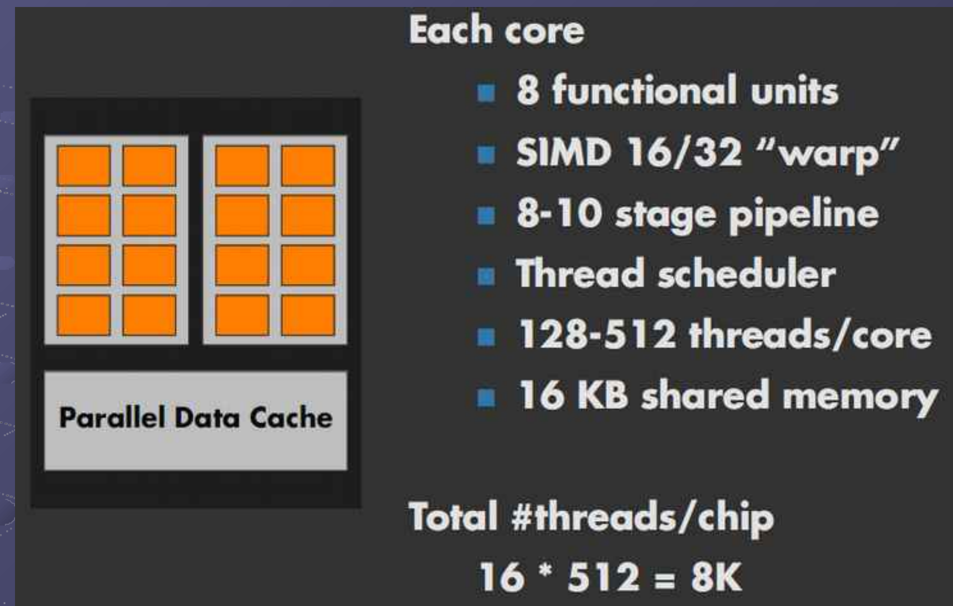
“The GPU devotes more transistors to computing”
“favours data parallel operations”

CPU vs. GPU speedup timeline



Hardware around 2006

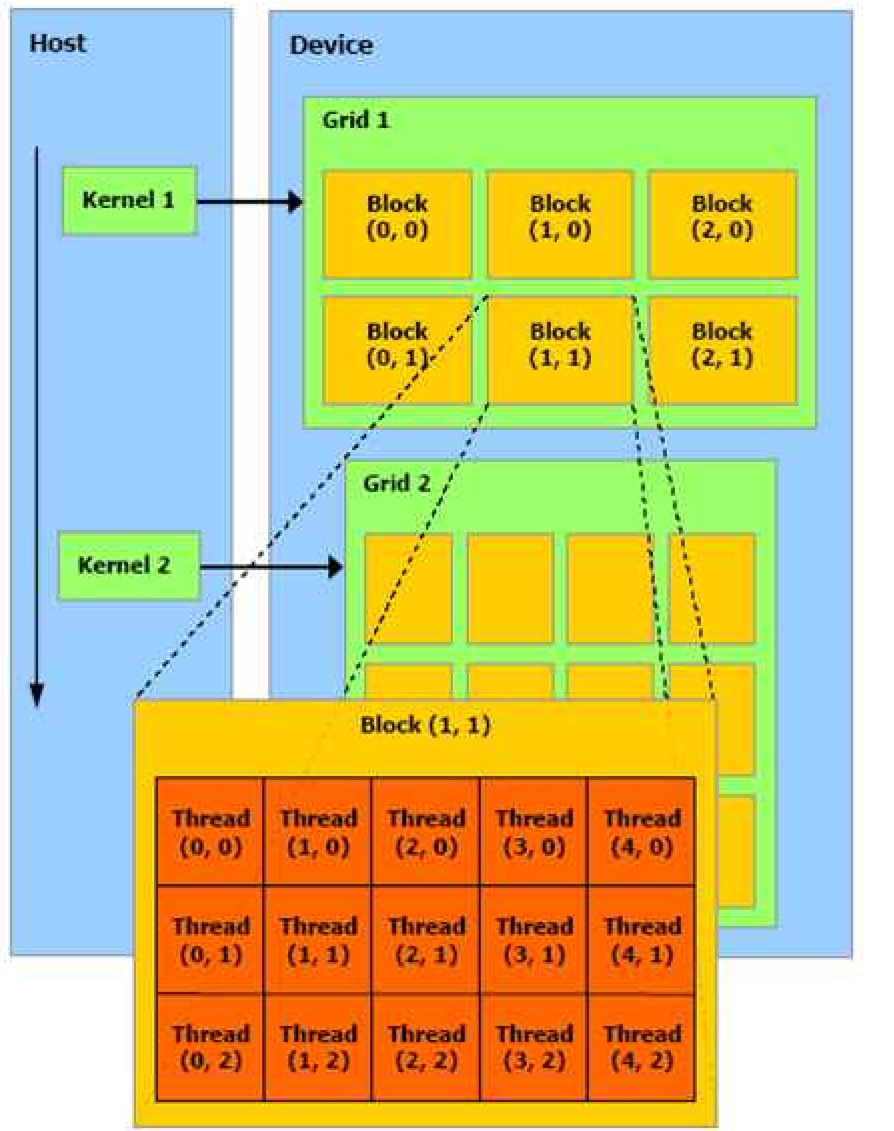




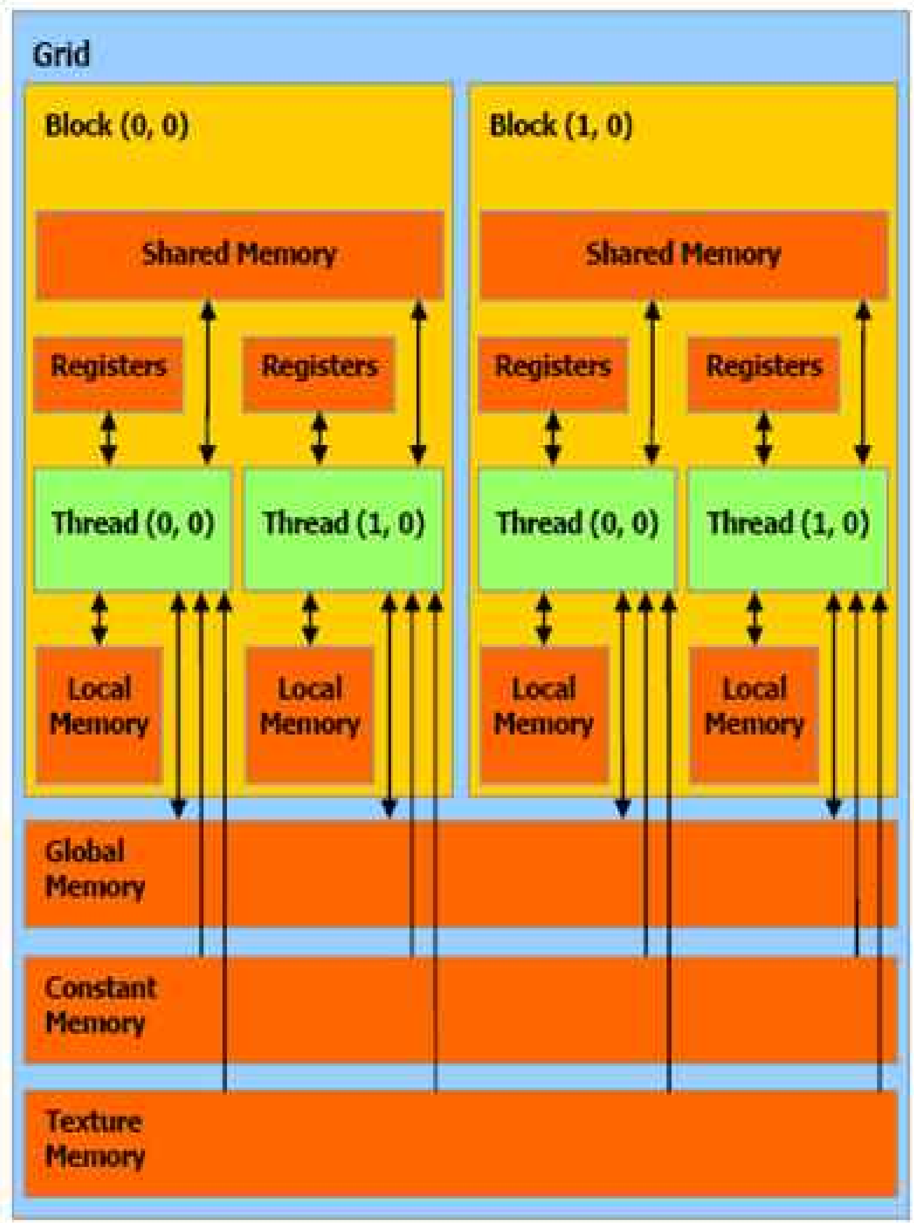
GeForce 8800 GTX:

575 MHz * 128 processors * 2 flop/inst * 2 inst/clock = 333 Gflops

GPU Structure From: http://geco.mines.edu/tesla/cuda_tutorial_mio/



The host issues a succession of kernel invocations to the device. Each kernel is executed as a batch of threads organized as a grid of thread blocks

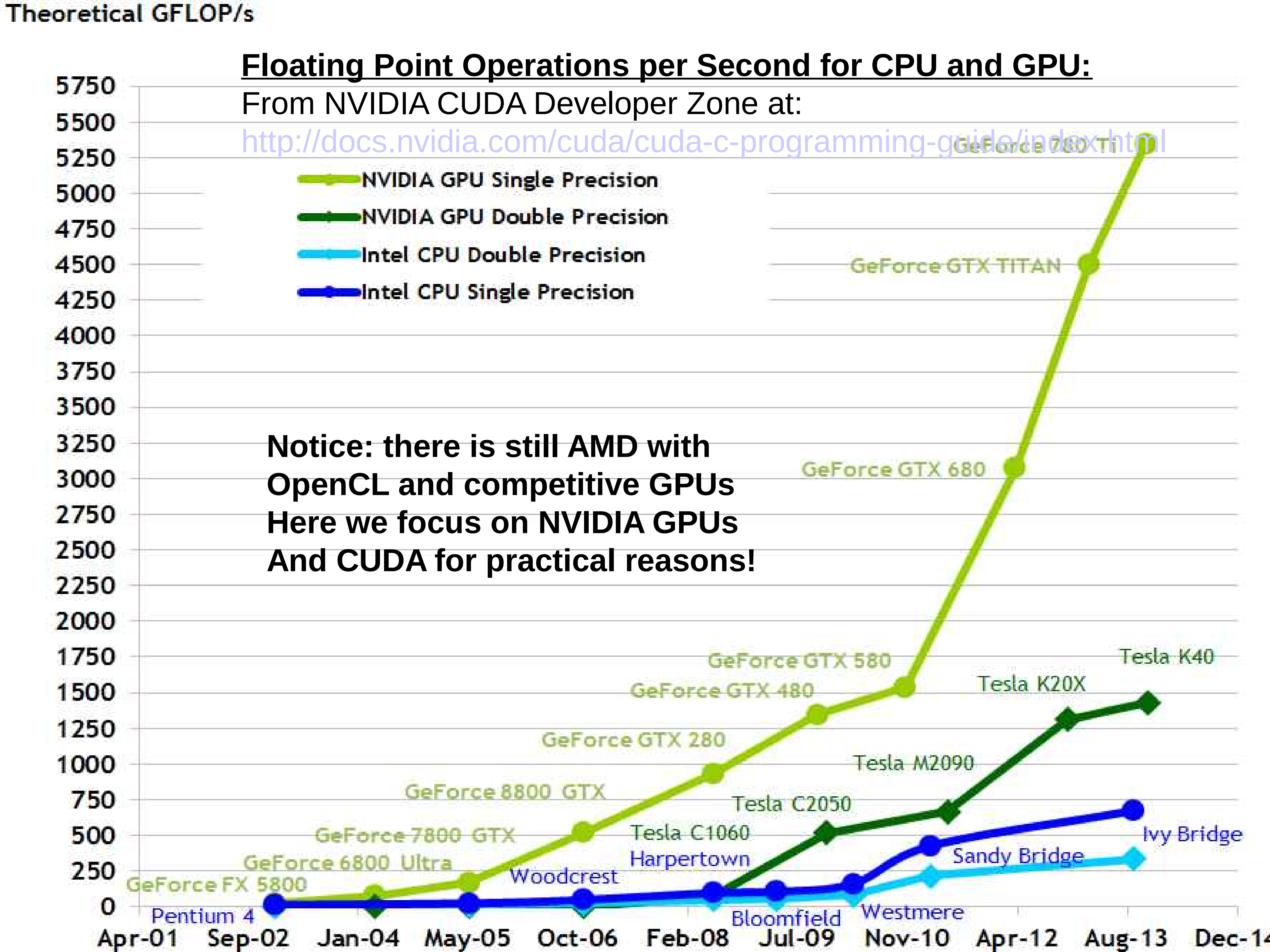


Floating Point Operations per Second for CPU and GPU:

From NVIDIA CUDA Developer Zone at:

http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

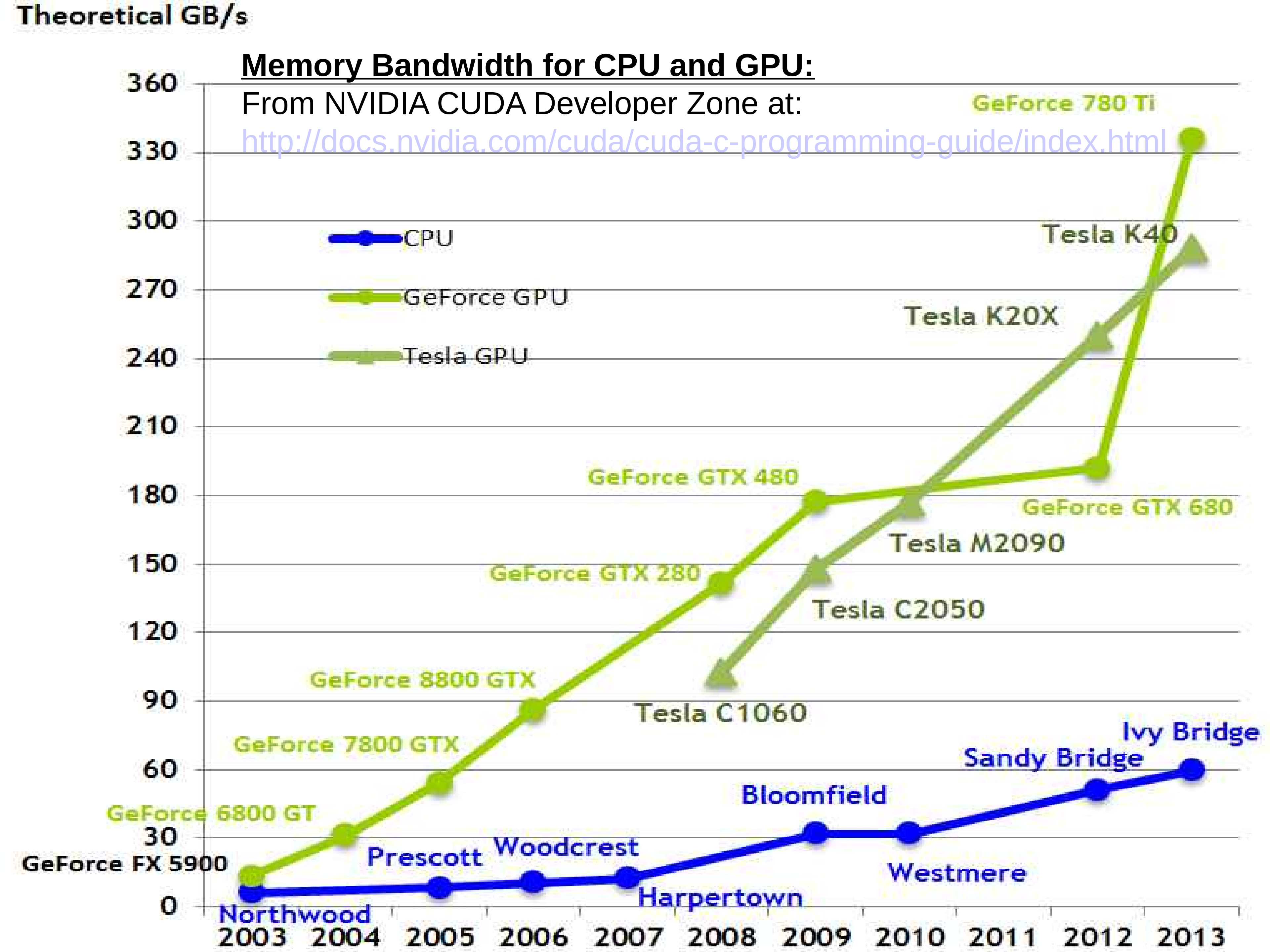
Notice: there is still AMD with OpenCL and competitive GPUs Here we focus on NVIDIA GPUs And CUDA for practical reasons!



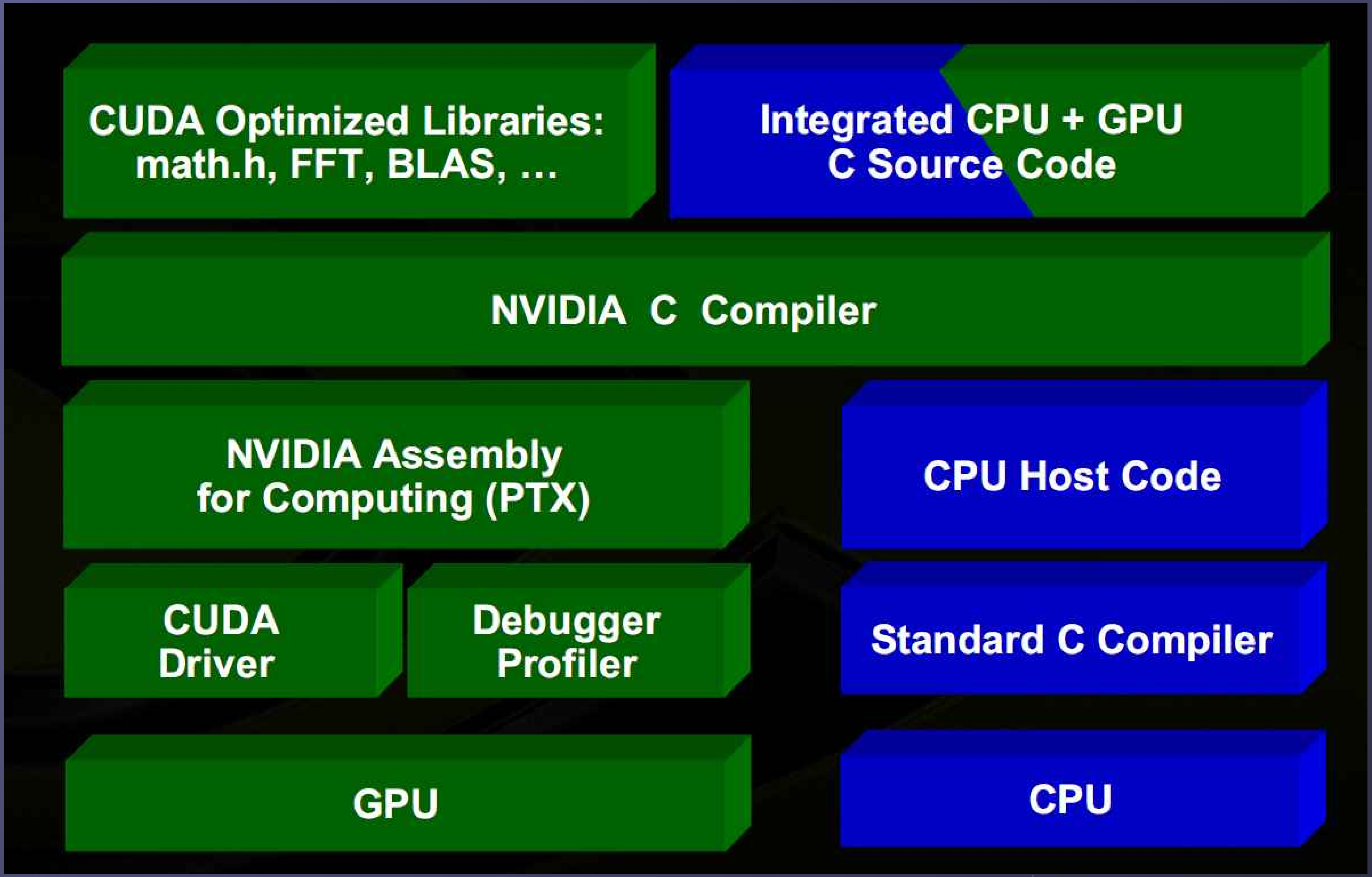
Memory Bandwidth for CPU and GPU:

From NVIDIA CUDA Developer Zone at:

http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html



CUDA


CUDA Optimized Libraries:
math.h, FFT, BLAS, …
Integrated CPU + GPU
C Source Code
NVIDIA C Compiler
NVIDIA Assembly
for Computing (PTX)
CPU Host Code
CUDA
Driver
Debugger
Profiler
Standard C Compiler
GPU
CPU

Simple CUDA example

CPU C program

```
void addMatrix(float *a, float *b,
               float *c, int N)
{
  int i, j, index;
  for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
      index = i + j * N;
      c[index]=a[index] + b[index];
    }
  }
}

void main()
{
  .....
  addMatrix(a, b, c, N);
}
```

CUDA C program

```
__global__ void addMatrix(float *a,float *b,
                          float *c, int N)
{
  int i=blockIdx.x*blockDim.x+threadIdx.x;
  int j=blockIdx.y*blockDim.y+threadIdx.y;
  int index = i + j * N;
  if ( i < N && j < N)
    c[index]= a[index] + b[index];
}

void main()
{
  ..... // allocate & transfer data to GPU
  dim3 dimBlk (blocksize, blocksize);
  dim3 dimGrd (N/dimBlk.x, N/dimBlk.y);
  addMatrix<<<dimGrd,dimBlk>>>(a, b, c,N);
}
```

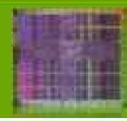
GPU Computing Applications

Libraries and Middleware

CUFFT CUBLAS CURAND CUSPARSE	CULA MAGMA	Thrust NPP	VSIPL SVM OpenCurrent	PhysX OptiX	iray	MATLAB Mathematica

Programming Languages

C	C++	Fortran	Java Python Wrappers	DirectCompute	Directives (e.g. OpenACC)



CUDA-Enabled NVIDIA GPUs

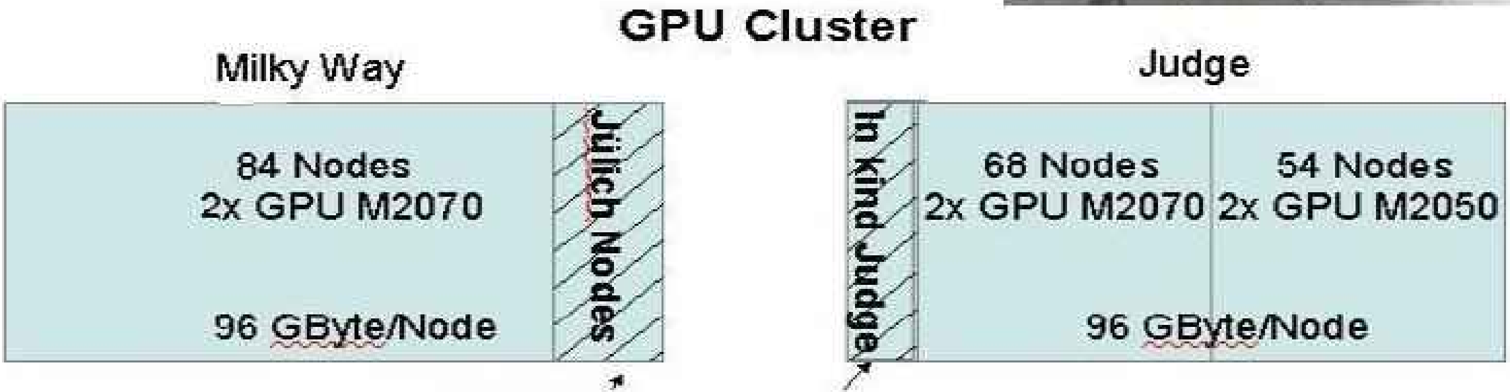
Kepler Architecture (compute capabilities 3.x)	GeForce 600 Series	Quadro Kepler Series	Tesla K20 Tesla K10
Fermi Architecture (compute capabilities 2.x)	GeForce 500 Series GeForce 400 Series	Quadro Fermi Series	Tesla 20 Series
Tesla Architecture (compute capabilities 1.x)	GeForce 200 Series GeForce 9 Series GeForce 8 Series	Quadro FX Series Quadro Plex Series Quadro NVS Series	Tesla 10 Series

NAOC laohu cluster

Beijing, China

Milky Way GPU **cluster.**

SFB 881 – The Milky Way System
Collaboration with FZ Jülich, Germany
206 nodes x 24 = 4944 CPU cores (@ 2.8 GHz)
206 x 96 GB ~ 20 TB RAM CPU memory
408 GPUs M2070/M2050 ~ 200k GPU threads
~ 2 TB GPU device memory
since mid. 2012 jointly operated.
nodes "judge123 - judge206" – MW part.



Heidelberg

Kepler GPU cluster

VolkswagenStiftung

Kepler GPU cluster

12 nodes = 12 x 16 = 192 CPU cores (@ 2 GHz)

12 x 64 GB = 768 GB RAM CPU memory

12 GPUs K20m = 12 x 2496 ~ 30k GPU threads

12 x 4.8 GB ~ 57 GB GPU device memory

4 x Xilinx Virtex-6 FPGA (ML 605)

since beg. 2013 operated.

Top 10 List November 2010

From www.top500.org - list of fastest supercomputers in the world... ... last year Nov. 2010:

Rank	Site	System
1	National Supercomputing Center in Tianjin China	**Tianhe-1A** - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C NUDT ***GPU***
2	DOE/SC/Oak Ridge National Laboratory United States	**Jaguar** - Cray XT5-HE Opteron 6-core 2.6 GHz Cray Inc.
3	National Supercomputing Centre in Shenzhen (NSCS) China	**Nebulae** - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU Dawning ***GPU***
4	GSIC Center, Tokyo Institute of Technology Japan	**TSUBAME 2.0** - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows NEC/HP ***GPU***
5	DOE/SC/LBNL/NERSC United States	**Hopper** - Cray XE6 12-core 2.1 GHz Cray Inc.
6	Commissariat a l'Energie Atomique (CEA) France FR	**Tera-100** - Bull bullx super-node S6010/S6030 Bull SA
7	DOE/NNSA/LANL United States	**Roadrunner** - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband IBM
8	National Institute for Computational Sciences/University of Tennessee United States	**Kraken XT5** - Cray XT5-HE Opteron 6-core 2.6 GHz Cray Inc.
9	Forschungszentrum Juelich (FZJ) Germany	**JUGENE** - Blue Gene/P Solution IBM
10	DOE/NNSA/LANL/SNL United States	**Cielo** - Cray XE6 8-core 2.4 GHz Cray Inc.

China Grabs Supercomputing Leadership Spot in Latest Ranking of World's Top 500 Supercomputers

Thu, 2010-11-11 22:42

MANNHEIM, Germany; BERKELEY, Calif.; and KNOXVILLE, Tenn.—The 36th edition of the closely watched TOP500 list of the world's most powerful supercomputers confirms the rumored takeover of the top spot by the Chinese Tianhe-1A system at the National Supercomputer Center in Tianjin, achieving a performance level of 2.57 petaflop/s (quadrillions of calculations per second).

天河

"天河一号"超级计算机系统
TH-1 supercomputer

Landmark result of the important project "High Efficient Supercomputer and Grid Service Environment" supported by National 863 Program.

- Built by National University of Defense Technology, with the cooperation of National Supercomputer Center in Tianjin (NSCC -TJ) and Inspur (Beijing) Electronic Information Industry Co., Ltd.

Host system of NSCC-TJ, installed in Tianjin Binhai New Area.

A backbone node of the national grid of China.

NCSA director: GPU is future of supercomputing

by Brooke Crothers

The director of the National Center for Supercomputing Applications has seen the future of supercomputing and it can be summed up in three letters: GPU.

Thom Dunning, who directs the NCSA and the Institute for Advanced Computing Applications and Technologies at the famed supercomputing facilities on the campus of University of Illinois at Urbana-Champaign, says high-performance computing will begin to move toward graphics processing units or GPUs. Not coincidentally, **this is exactly what China has done to achieve the world's fastest speeds with its "Tianhe-1A"** supercomputer. That computer combines about 7,000 Nvidia GPUs with 14,000 Intel CPUs: the only hybrid CPU-GPU system in the world of that scale.

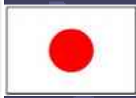
Thom Dunning directs the Institute for Advanced Computing Applications and Technologies and the NCSA.

"What we're really seeing in the efforts in China as well as the ones we have in the U.S. is that GPUs are what the future will look like," said Dunning in a phone interview Thursday. "What we're seeing is the beginning of something that's going to be happening all over the world."

NCSA already has a small CPU-GPU hybrid system. "It's something we have been working on for a number of years. We have a CPU-GPU cluster for the NCSA academic community. Made up of Intel CPUs and Nvidia GPUs. A 50 teraflop machine," he said. (Note that **Oak Ridge National Laboratories is also installing a hybrid system now**.)

Top 10 List 2011 –----- 2012

1	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom
2	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect
3	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom
4	SuperMUC - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR
5	Tianhe-1A - NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050
6	Jaguar - Cray XK6, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA 2090
7	Fermi - BlueGene/Q, Power BQC 16C 1.60GHz, Custom
8	JuQUEEN - BlueGene/Q, Power BQC 16C 1.60GHz, Custom
9	Curie thin nodes - Bullx B510, Xeon E5-2680 8C 2.700GHz, Infiniband QDR
10	Nebulae - Dawning TC3600 Blade System, Xeon X5650 6C 2.66GHz, Infiniband QDR, NVIDIA 2050



Rank	Site	System
1	National University of Defense Technology China ***Xeonϕ***	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluste Intel Xeon E5-2692 12C 2.200GHz, TH Expre Intel Xeon Phi 31S1P NUDT
2	DOE/SC/Oak Ridge National Laboratory United States ***GPU***	**Titan** - Cray XK7 , Opteron 6274 16C 2.200G Cray Gemini interconnect, NVIDIA K20x Cray Inc.
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.6 GHz, Custom IBM
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GH Custom IBM
6	Texas Advanced Computing Center/Univ. of Texas United States ***Xeonϕ***	**Stampede** - PowerEdge C8220, Xeon E5-26 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell
7	Forschungszentrum Juelich (FZJ) Germany	**JUQUEEN** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM
8	DOE/NNSA/LLNL United States	**Vulcan** - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM
9	Leibniz Rechenzentrum Germany	**SuperMUC** - iDataPlex DX360M4, Xeon E5-2 8C 2.70GHz, Infiniband FDR

Nr. 1,2 Supercomputer from China: 96/33 Pflop/s Linpack
Wuxi/Guangzhou/Tianjin National Supercomputing Center
Taihu 10 mill. cores

Tianhe-2 (MilkyWay-2) - TH-IV
E5-2692 12C 2.200GHz, TH Ex
31S1P

32000 Intel Xeon 12 core,
48000 Intel Phi Accelerators 57 Core

Test of Taihu planned;
But:
Local cluster with new GPUs at NAOC gives much more resources.

Top 10 List June 2016

	Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
	1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
	2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeo E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
USA	3	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interco NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
USA	4	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
	5	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	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
USA	6	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
USA	7	DOE/NNSA/LANL/SNL United States	**Trinity** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	301,056	8,100.9	11,078.9	
Swiss	8	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
	9	HLRS - Hochstleistungsrechenzentrum Stuttgart Germany	**Hazel Hen** - Cray XC40, Xeon E5-2680v3 12C 2.5GHz, Aries interconnect Cray Inc.	185,088	5,640.2	7,403.5	
Saudi-A.	10	King Abdullah University of Science and Technology Saudi Arabia	**Shaheen II** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	196,608	5,537.0	7,235.2	2,834

Chinese Processor

Xeonϕ

GPU

Computational Science...

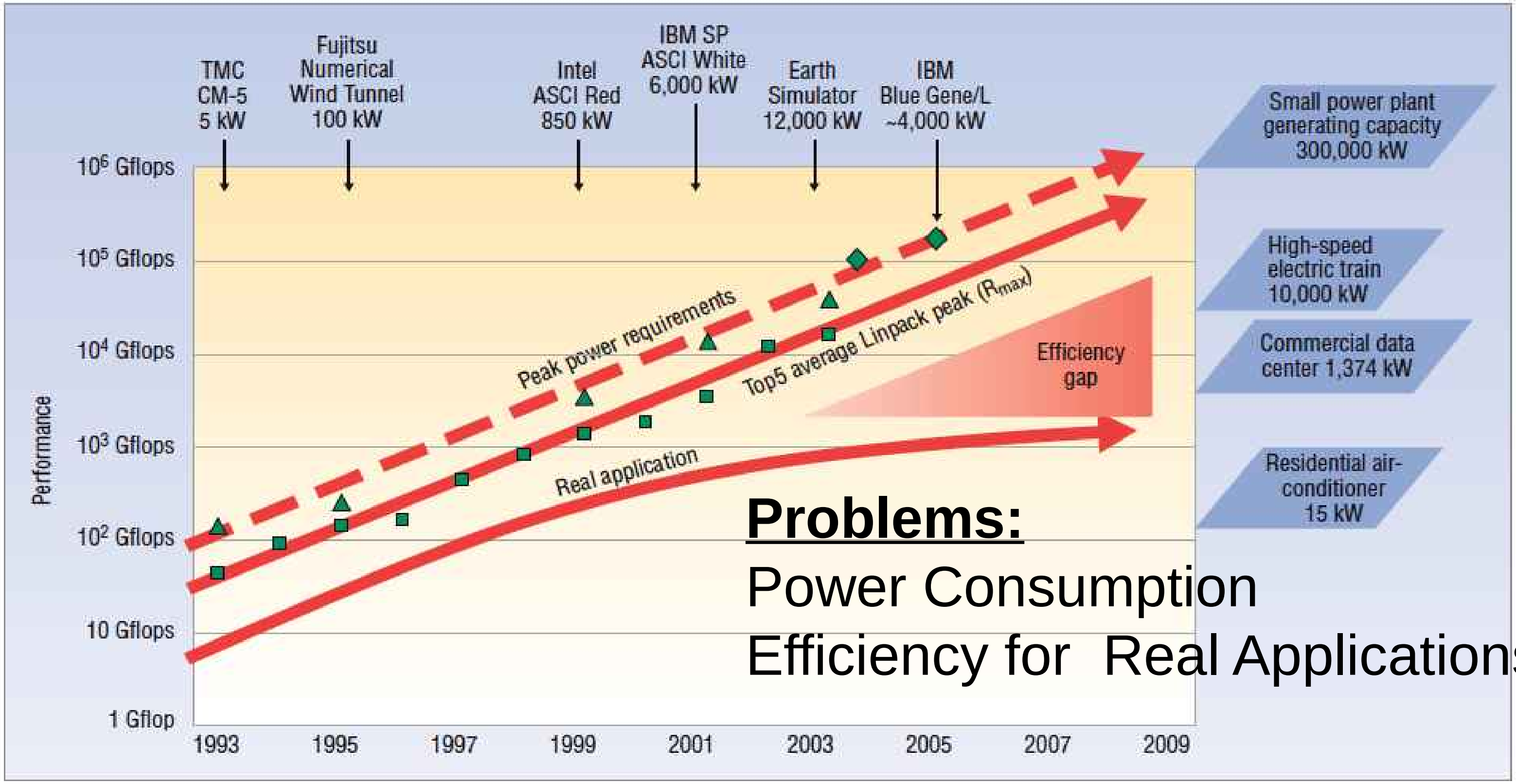
...after von Neumann...

Exaflop/s?

Petaflop/s

Teraflop/s

Gigaflop/s



Problems:

Power Consumption

Efficiency for Real Applications

Figure 1. Rising power requirements. Peak power consumption of the top supercomputers has steadily increased over the past 15 years.

Thanks to Horst Simon, LBNL/NERSC for this diagram.

Green500 Rank	MFLOPS/W	Site	System	Total Power(kW)
1	7031.4	Institute of Physical and Chemical Research (RIKEN)	ExaScaler-1.4 80Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SC	50.3
2	5331.5	GSIC Center, Tokyo Institute of Technology	LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.1GHz, Infiniband FDR, NVIDIA Tesla K80	51.1
3	5272.1	GSI Helmholtz Center	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.2
4	4778.5	Institute of Modern Physics (IMP), Chinese Academy of Sciences	Sugon Cluster W780I, Xeon E5-2640v3 8C 2.6GHz, Infiniband QDR, NVIDIA Tesla K80	65
5	4112.1	Stanford Research Computing Center	Cray CS-Storm, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, Nvidia K80	190
6	3856.9	IT Company	Inspur TS10000 HPC Server, Xeon E5-2620v3 6C 2.4GHz, 10G Ethernet, NVIDIA Tesla K40	58
7	3775.5	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110
8	3775.5	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110
9	3775.5	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110
10	3775.5	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110

Intel MIC Hardware

INSPUR, NAOC - 2013.XI.26

icpc … "-mmic" … 61 x 4 = 244 x 1.1 GHz omp cores !!!

Full fp64 !!!

Intel MIC Hardware

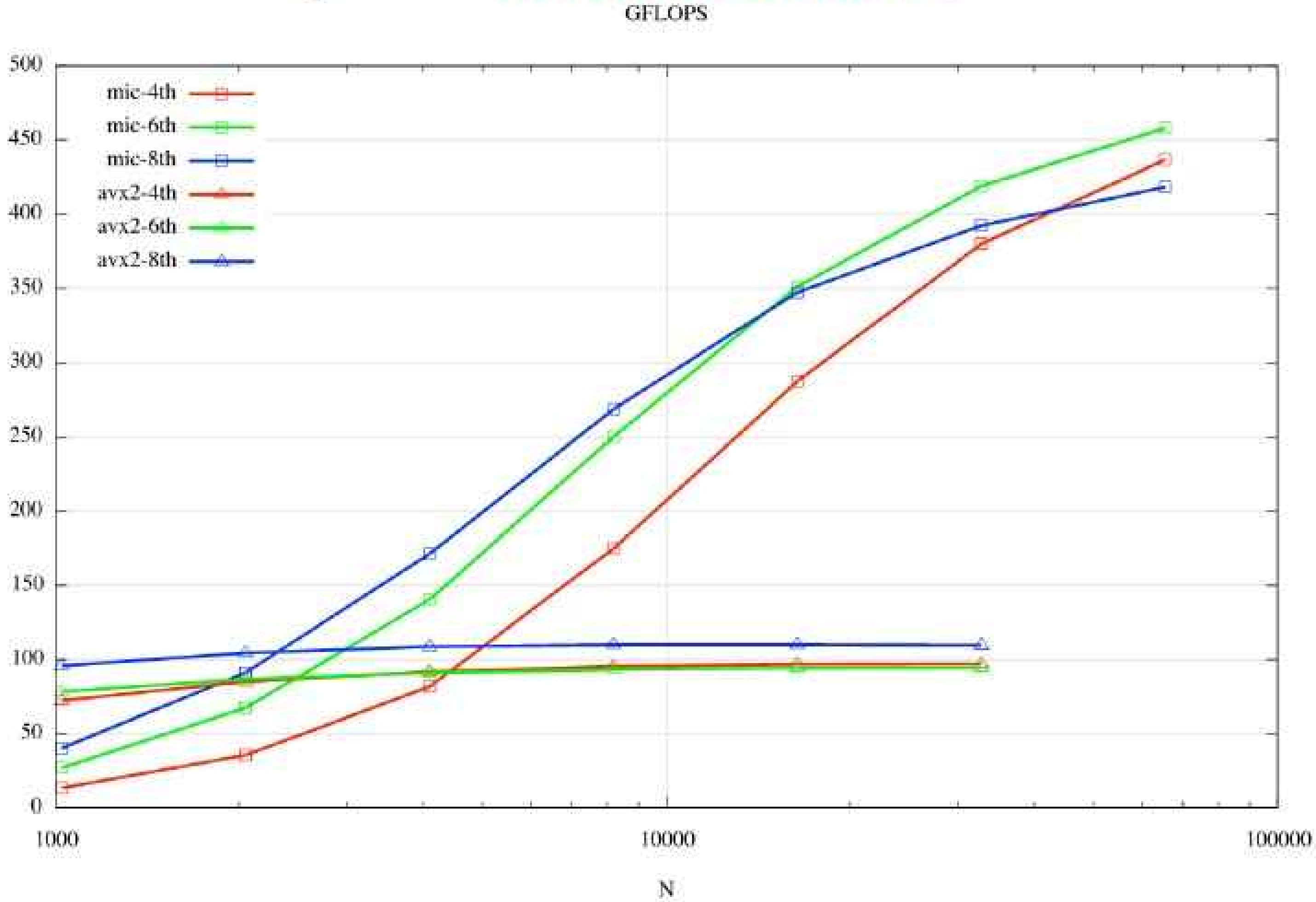
Intel® Xeon Phi™ Coprocessor Family Reference Table

SKU #	Form Factor, Thermal	Peak Double Precision	Max # of Cores	Clock Speed (GHz)	GDDR5 Memory Speeds (GT/s)	Peak Memory BW	Memory Capacity (GB)	Total Cache (MB)	Board TDP (Watts)	Process
SE10P (special edition)	PCIe Card, Passively Cooled	1073 GF	61	1.1	5.5	352	8	30.5	300	22nm
SE10X (special edition)	PCIe Card, No Thermal Solution	1073 GF	61	1.1	5.5	352	8	30.5	300	
5110P	PCIe Card, Passively Cooled	1011 GF	60	1.053	5.0	320	8	30	225	
3100 Series	PCIe Card, Actively Cooled	>1 TF	Disclosed at 3100 series launch (H1'13)		5.0	240	6	28.5	300	
	PCIe Card, Passively Cooled	> 1 TF			5.0	240	6	28.5	300	

PCIe Card, Actively Cooled

PCIe Card, Passively Cooled

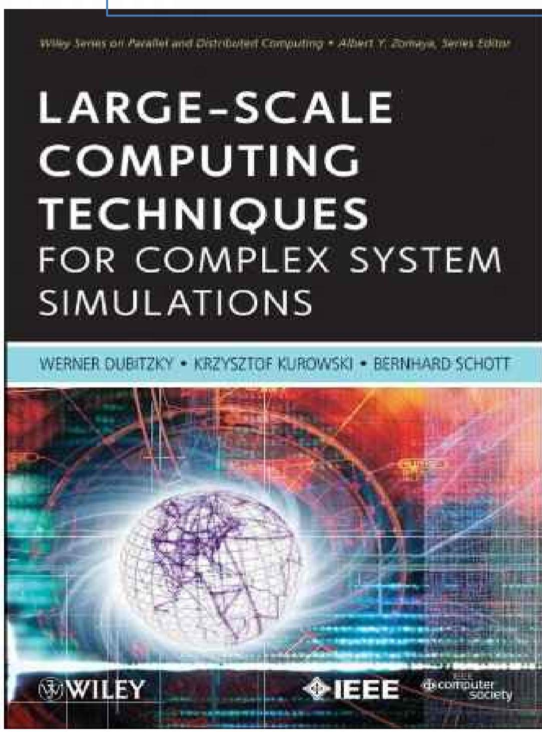
φGPU Hermite results

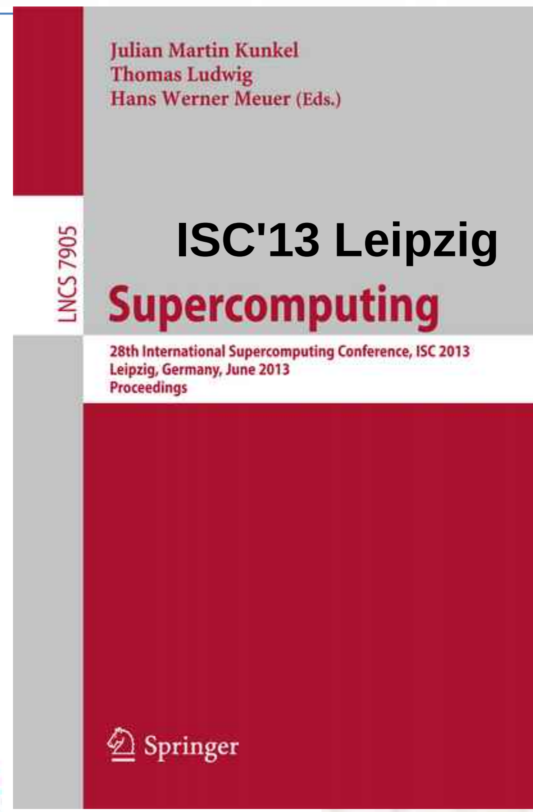


PRACE Award - 2011

Astrophysical Particle Simulations with Large Custom GPU Clusters on Three Continents

Rainer Spurzem, *et al*, Chinese Academy of Sciences & University of Heidelberg





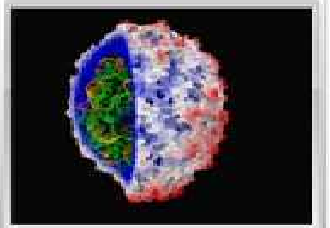
中国科学院国家天文台
NATIONAL ASTRONOMICAL OBSERVATORIES ,CHINESE ACADEMY OF SCIENCES

北京大學
PEKING UNIVERSITY

Speedups using GPU vs. CPU

146X

Interactive visualization of volumetric white matter connectivity[1]

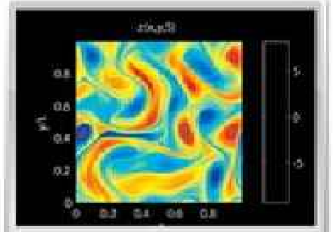


36X

Ionic placement for molecular dynamics simulation on GPU[2]

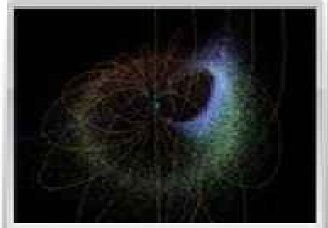
18X

Transcoding HD video stream to H.264 for portable video[3]



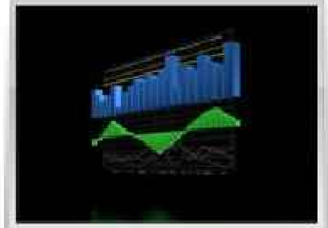
17X

Simulation in Matlab using .mex file CUDA function[4]



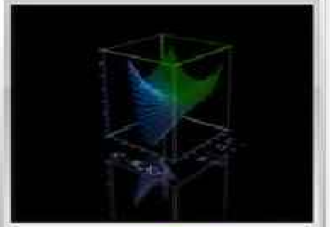
100X

Astrophysics N-body simulation[5]



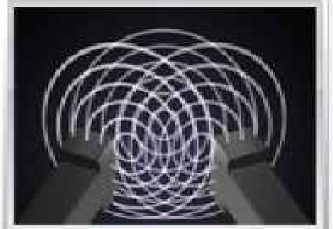
149X

Financial simulation of LIBOR model with swaptions[6]



47X

GLAME@lab: M-script API for linear Algebra operations on GPU[7]

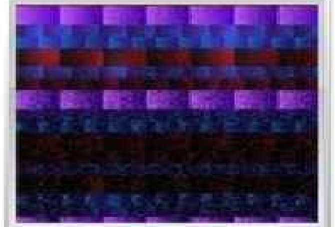


20X

Ultrasound medical imaging for cancer diagnostics[8]

24X

Highly optimized object oriented molecular dynamics[9]



30X

Cmatch exact string matching - find similar proteins & gene sequences[10]

Towards Peta-Scale Green Computation

— applications of the GPU supercomputers in CAS

http://www.nvidia.com/gtc2010-content

GPU TECHNOLOGY CONFERENCE

GTC 2010 | Sept 20-23, 2010
San Jose Convention Center, San Jose, California
Watch the Keynote Recordings

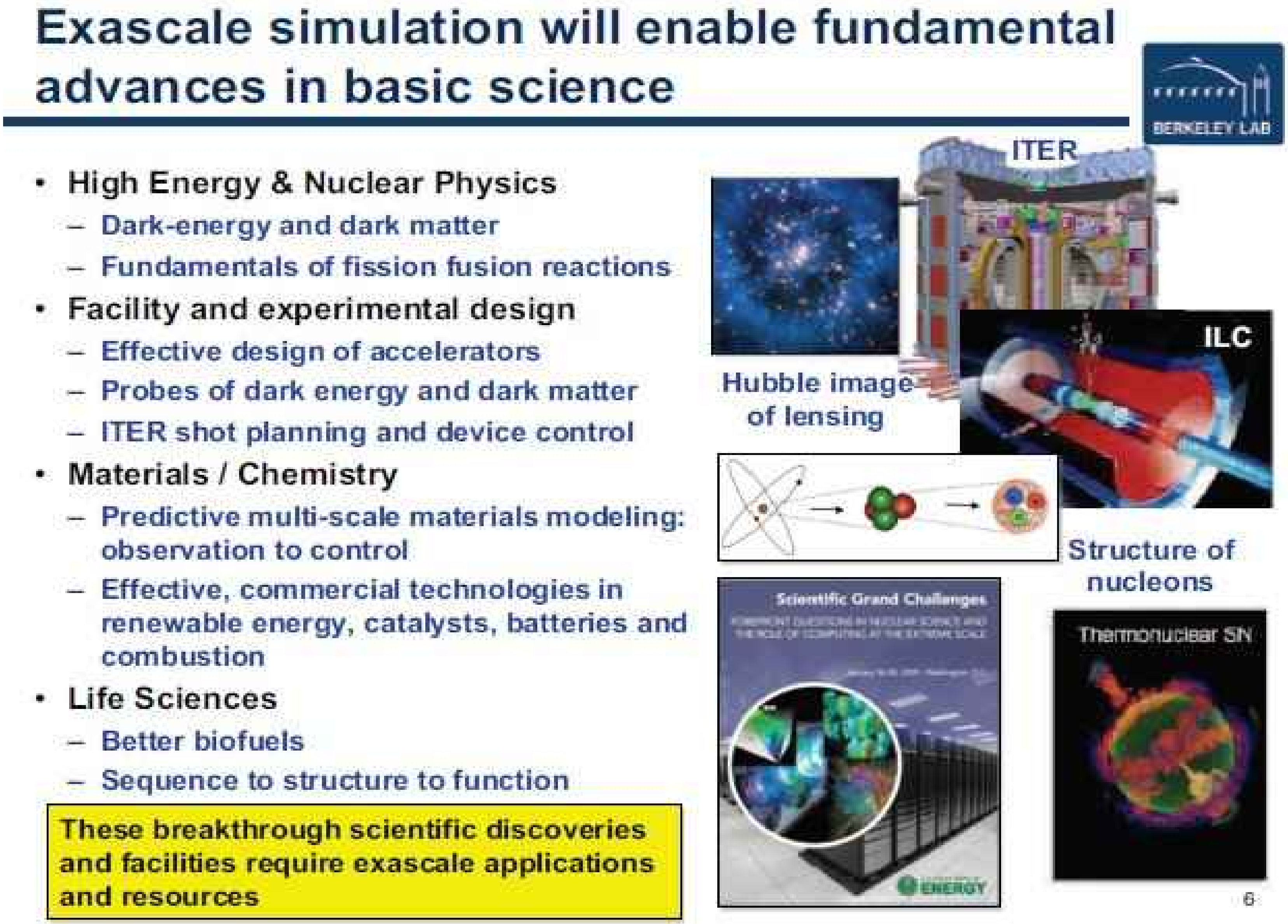
- Algorithms & Numerical Techniques
- Astronomy & Astrophysics
- Audio Processing
- Cloud Computing
- Computational Fluid Dynamics
- Computer Graphics
- Computer Vision
- Databases & Data Mining
- Digital Content Creation
- Embedded & Automotive
- Energy Exploration
- Film
- Finance
- General Interest
- GPU Accelerated Internet
- High Performance Computing
- Imaging
- Life Sciences
- Machine Learning & Artificial Intelligence
- Medical Imaging & Visualization
- Mobile & Tablet & Phone
- Molecular Dynamics
- Neuroscience
- Physics Simulation
- Programming Languages & Techniques
- Quantum Chemistry
- Ray Tracing
- Signal Processing
- Stereoscopic 3D
- Tools & Libraries
- Video Processing

Wei Ge
Xiaowei Wang
Inst. of Proc. Eng.

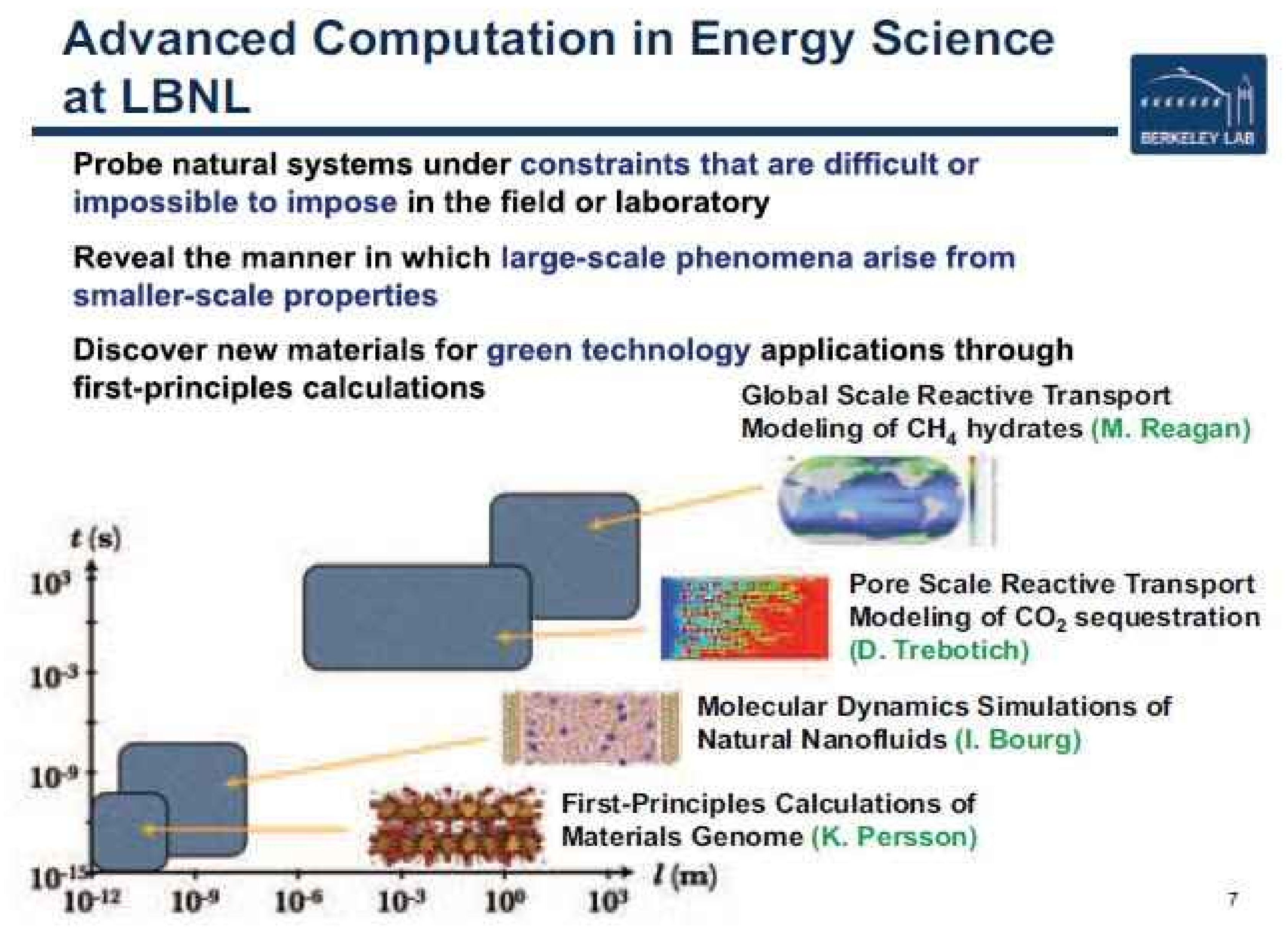
Yunquan Zhang
Inst. of Software

Rainer Spurzem
Nat. Astro. Obs. Chn.

Long Wang
SC Center



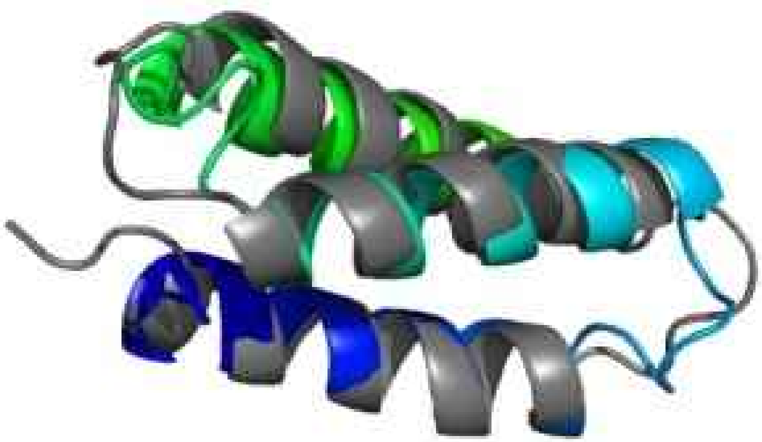
Slide: Horst Simon, Exascale Computing



Slide: Horst Simon, Exascale Computing

Research http://www.fz-juelich.de/ias/jsc/EN/Research/research_node.html

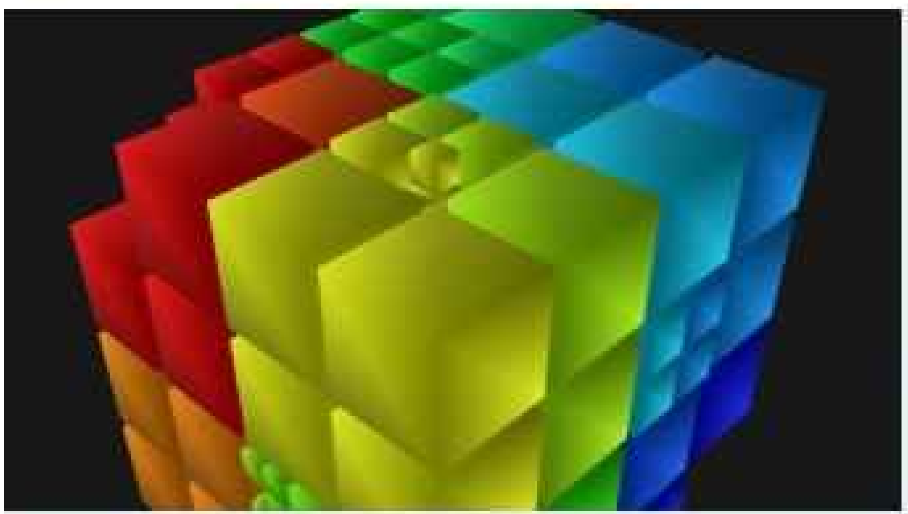
JSC's research and development concentrates on mathematical modelling and numerical, especially parallel algorithms for quantum chemistry, molecular dynamics and Monte-Carlo simulations. The focus in the computer sciences is on cluster computing, performance analysis of parallel programs, visualization, computational steering and grid computing.



Modelling and Simulation

The simulation of complex systems in natural science or engineering depends on the development of adequate mathematical models. Thus the development of realistic and yet efficient models is a core activity at JSC. Examples of simulations are:

- Computational Plasma Physics
- Protein Folding
- Quantum Information Processing
- Civil Security and Traffic



Algorithms and Methods

Efficient simulations need powerful algorithms and methods. JSC focusses on the development of the following methods:

- Fast Coulomb Solvers
- Parallel-In-Time Integration
- Fast Multipole Method
- Parallel I/O

Molecular Dynamics

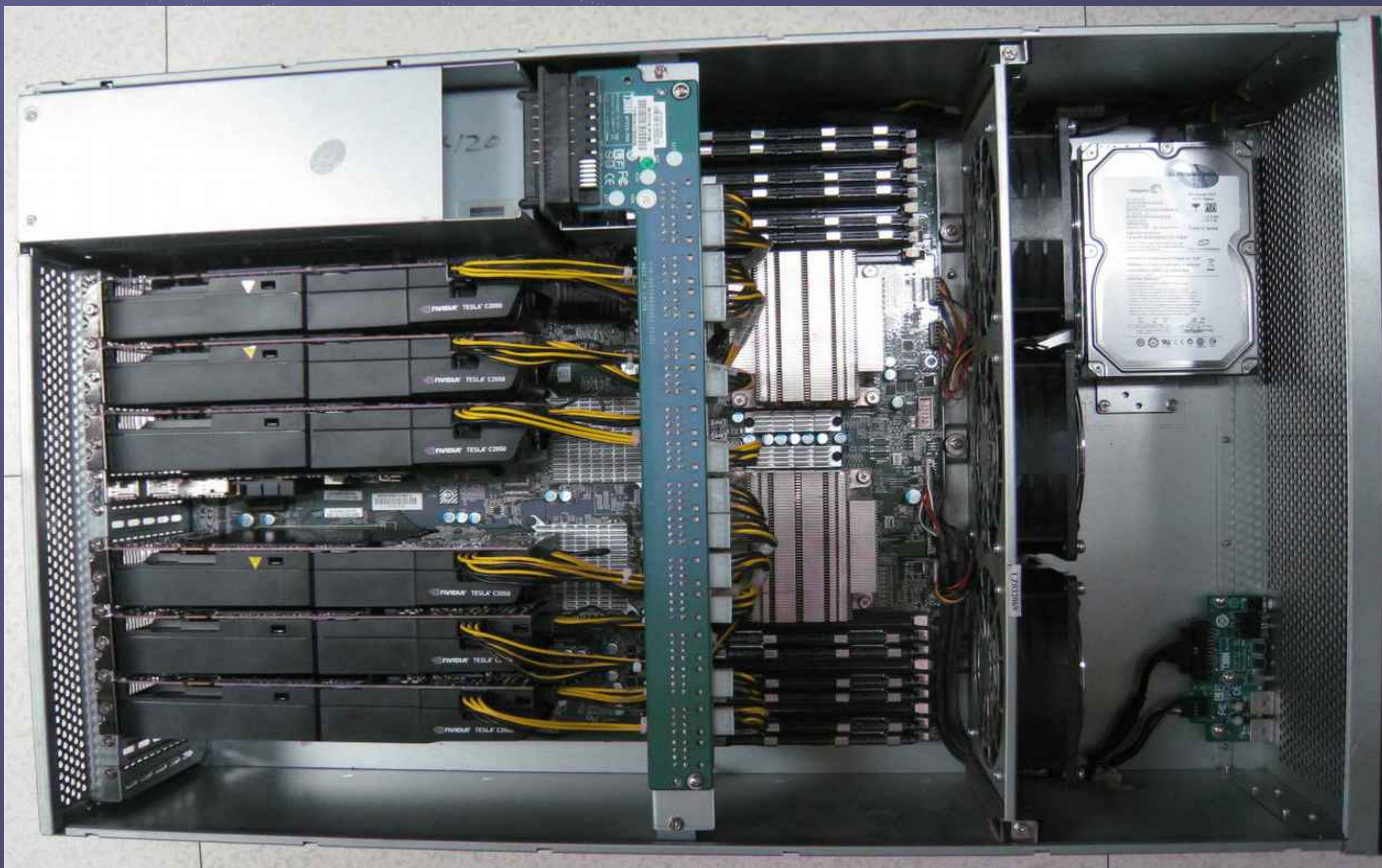
Fermi-based GPU supercomputer IPE (2010.04.24)

Rpeak SP：	2Pflops
Rpeak DP：	1Pflops
Linpack:	207.3T (Top500 19th)
Mflops/Watt:	431 (Green500 8th)
Total RAM：	17.2TB
Total VRAM：	6.6TB
Total HD：	360TB
Inst. Comm.：	H3C GE
Data Comm.：	Mellanox QDR IB
Occupied area：	150 sq.m.
Weight：	12.6 tons
Max Power：	600kW(computing) 200kW(cooling)
System：	CentOS 5.4, PBS
Monitor：	Ganglia, GPU monitor
Languages：	C, C++, CUDA 3.1 , OpenCL

中国科学院过程工程研究所
Institute Of Process Engineering, Chinese Academy Of Sciences

IPE CAS 372 node 6xC2050 cluster

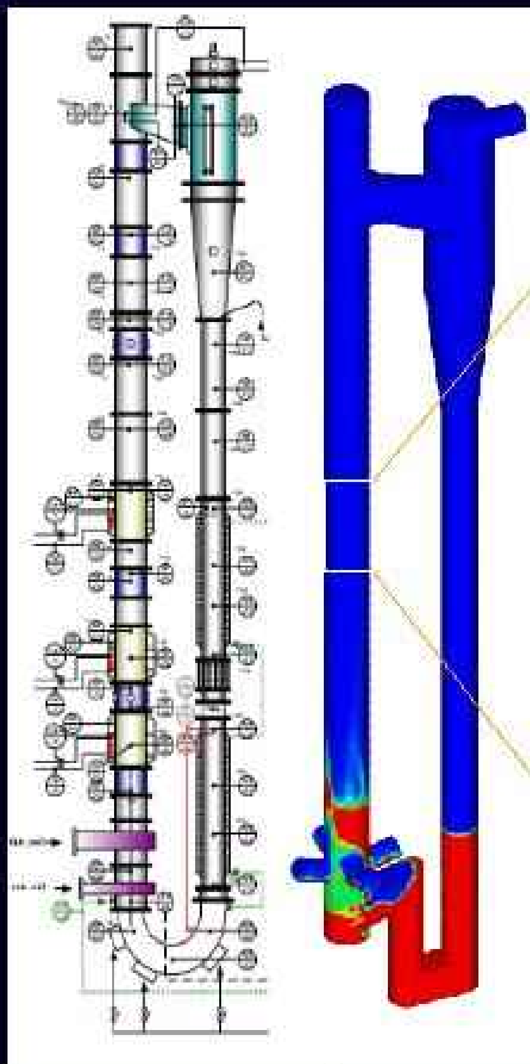
2232 GPU = 2.2 Pflops SP / 1.1 Pflops DP



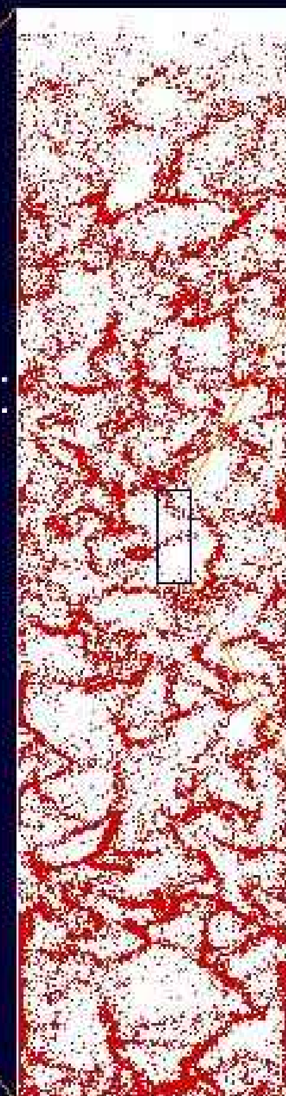
DNS of gas-solid flow： >20x speedup (1C1060/1E5430 core)

120K Particles + 400M pseudo-particles

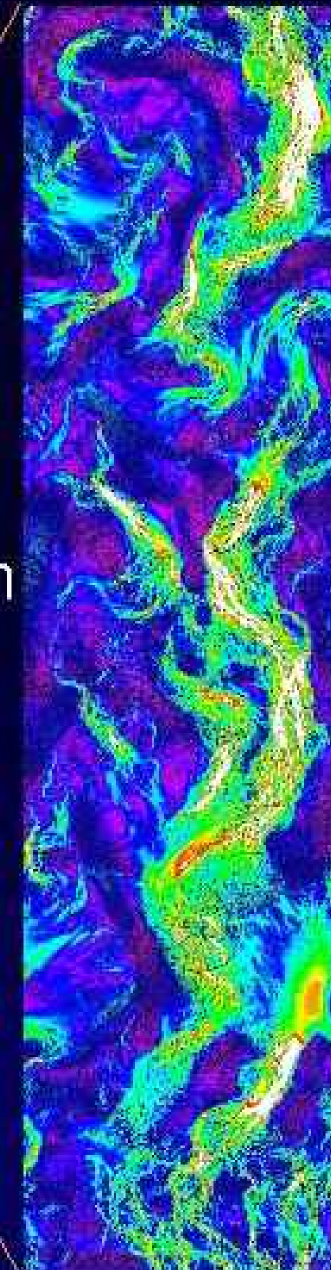
Reactor:
0.4*20m
3D



Section:
0.4*1m
2D



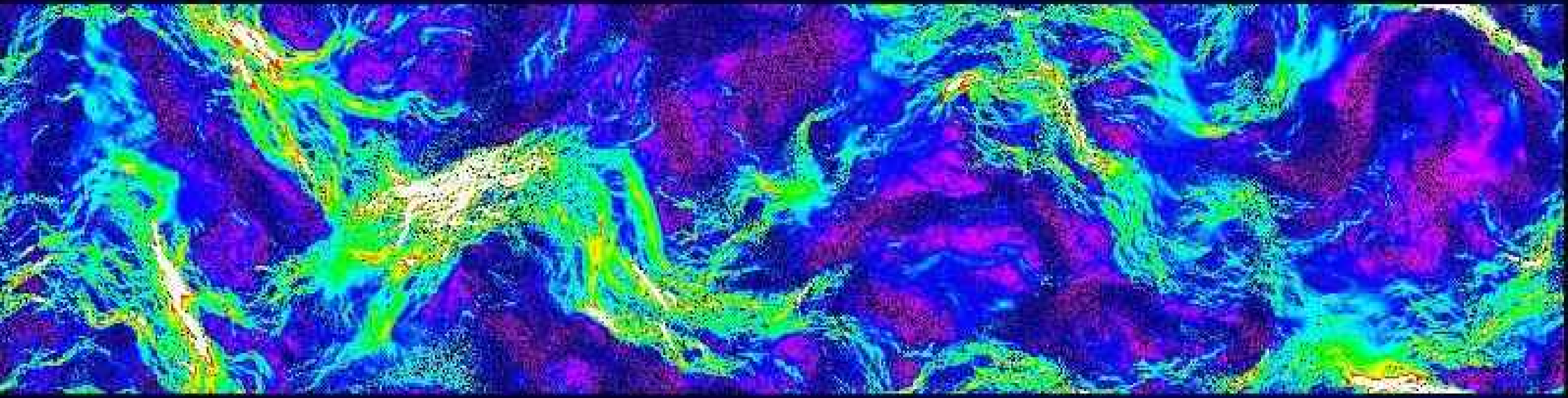
Cell:
2*10cm
2D



Animation Challenge:

9600x2400 → 1200x300 pixels

1000 → 17 frames



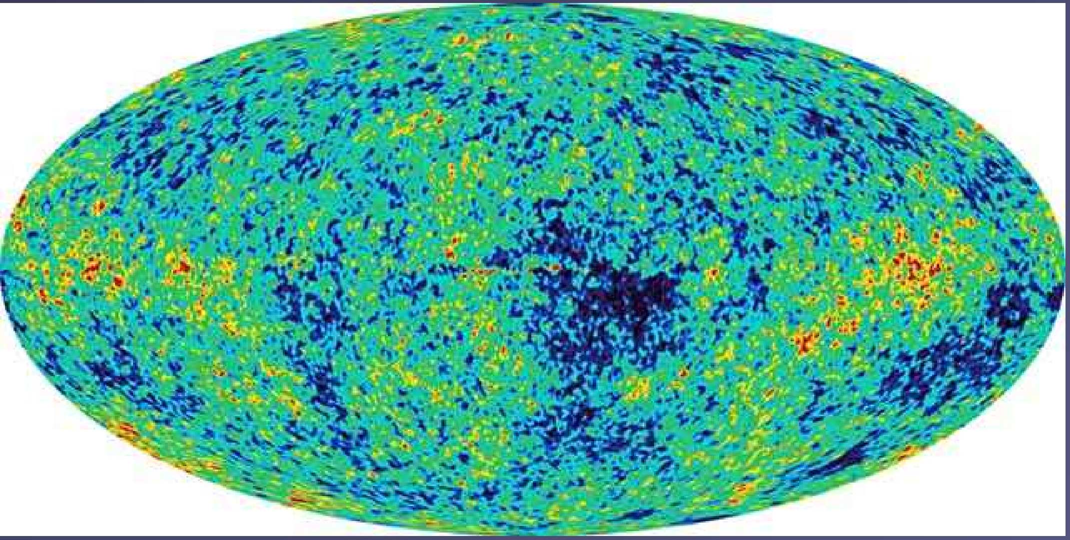
Computer Physics - Astrophysics

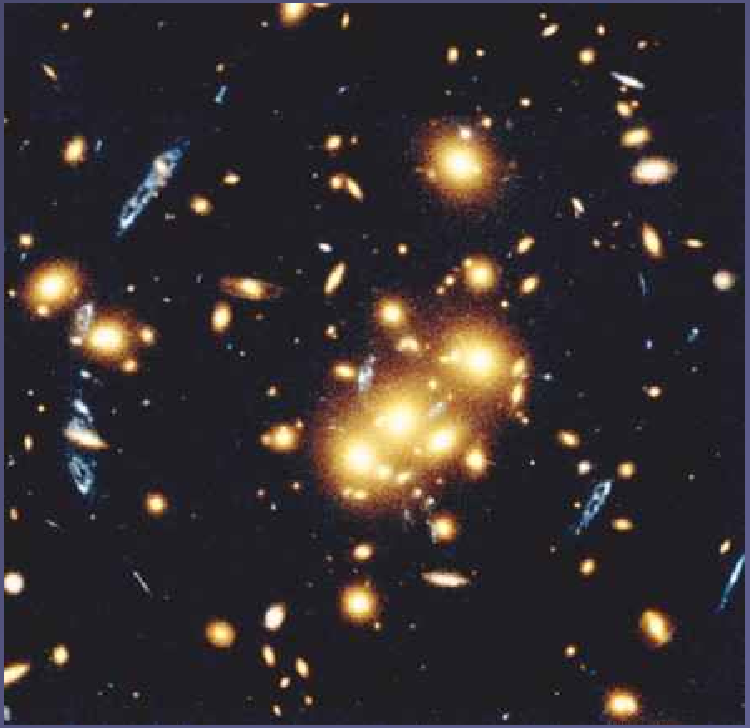
Cosmology

Computer Physics – Astrophysics

- Structure Formation in the Universe

In the year 100.000....

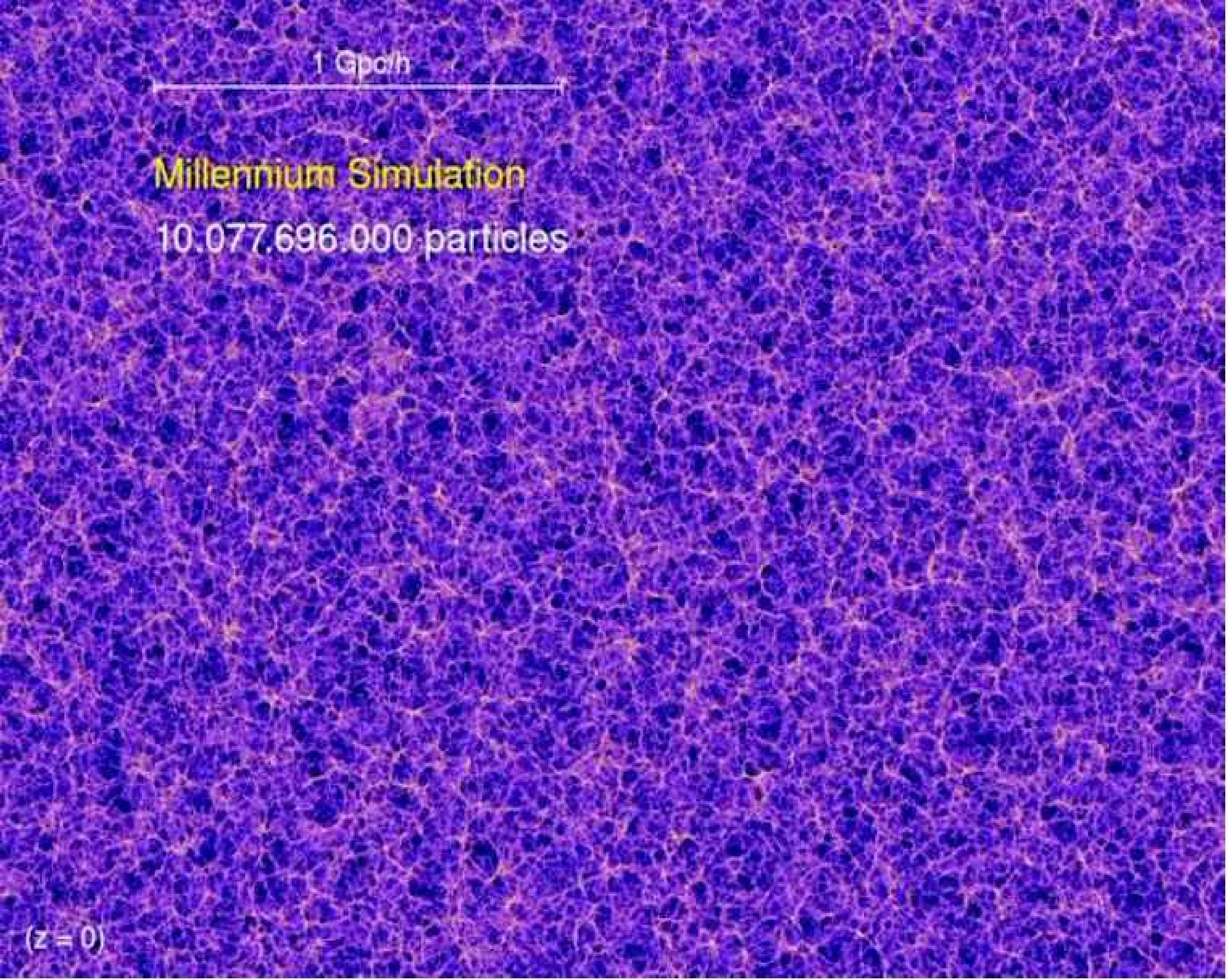




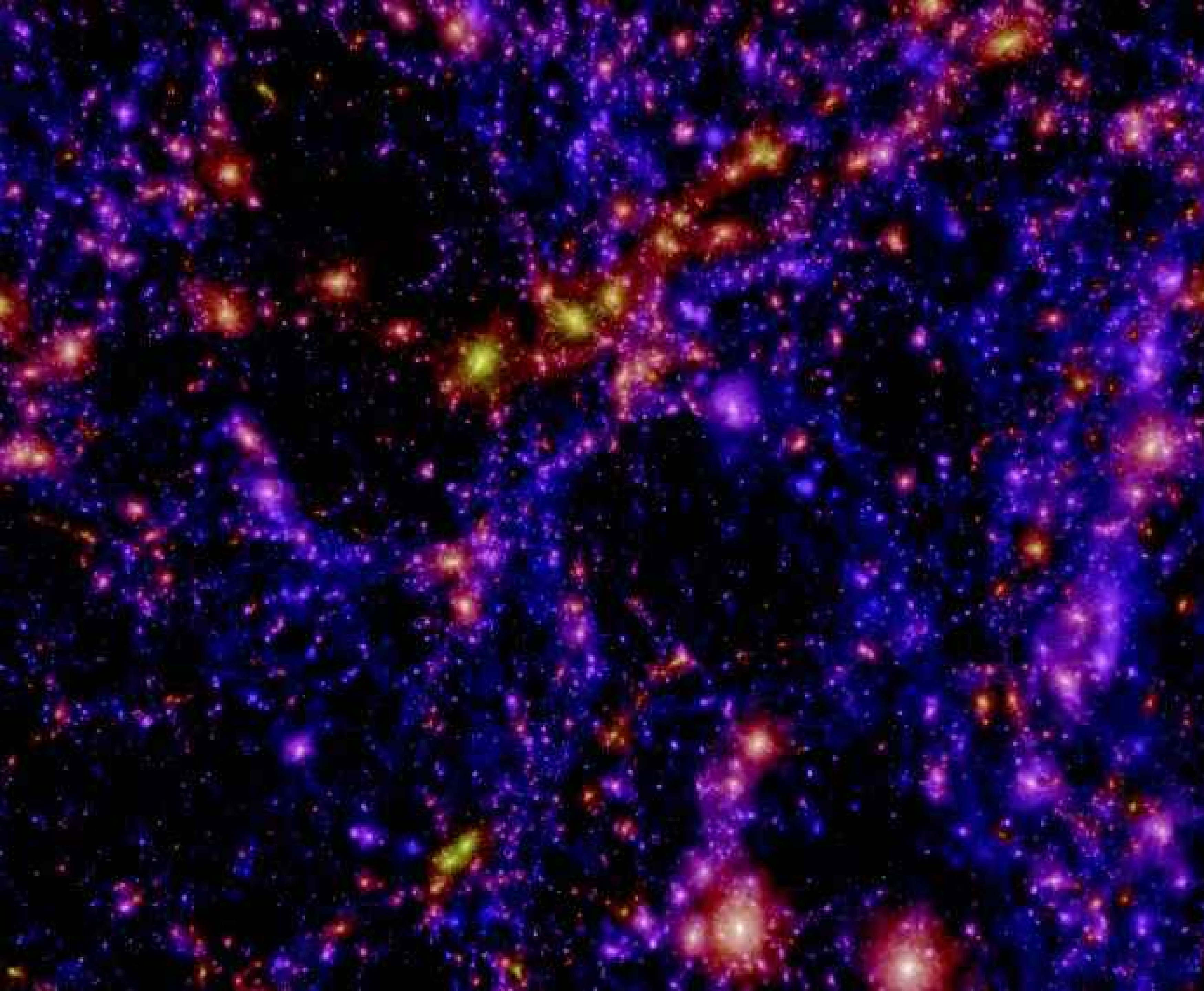
- Wilkinson Microwave Anisotropy Probe (WMAP)

(Cosmic Microwave Background)

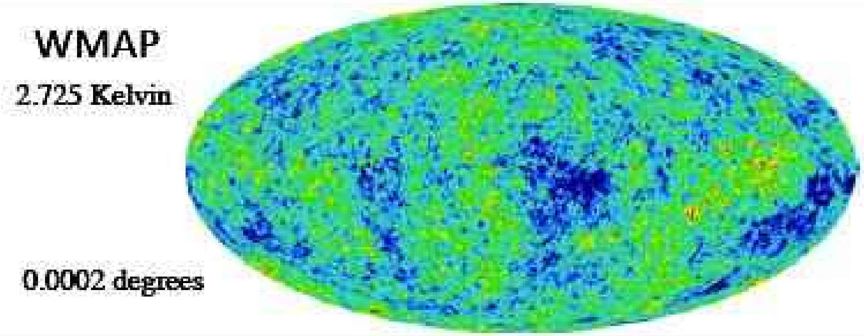
...and ``today´´

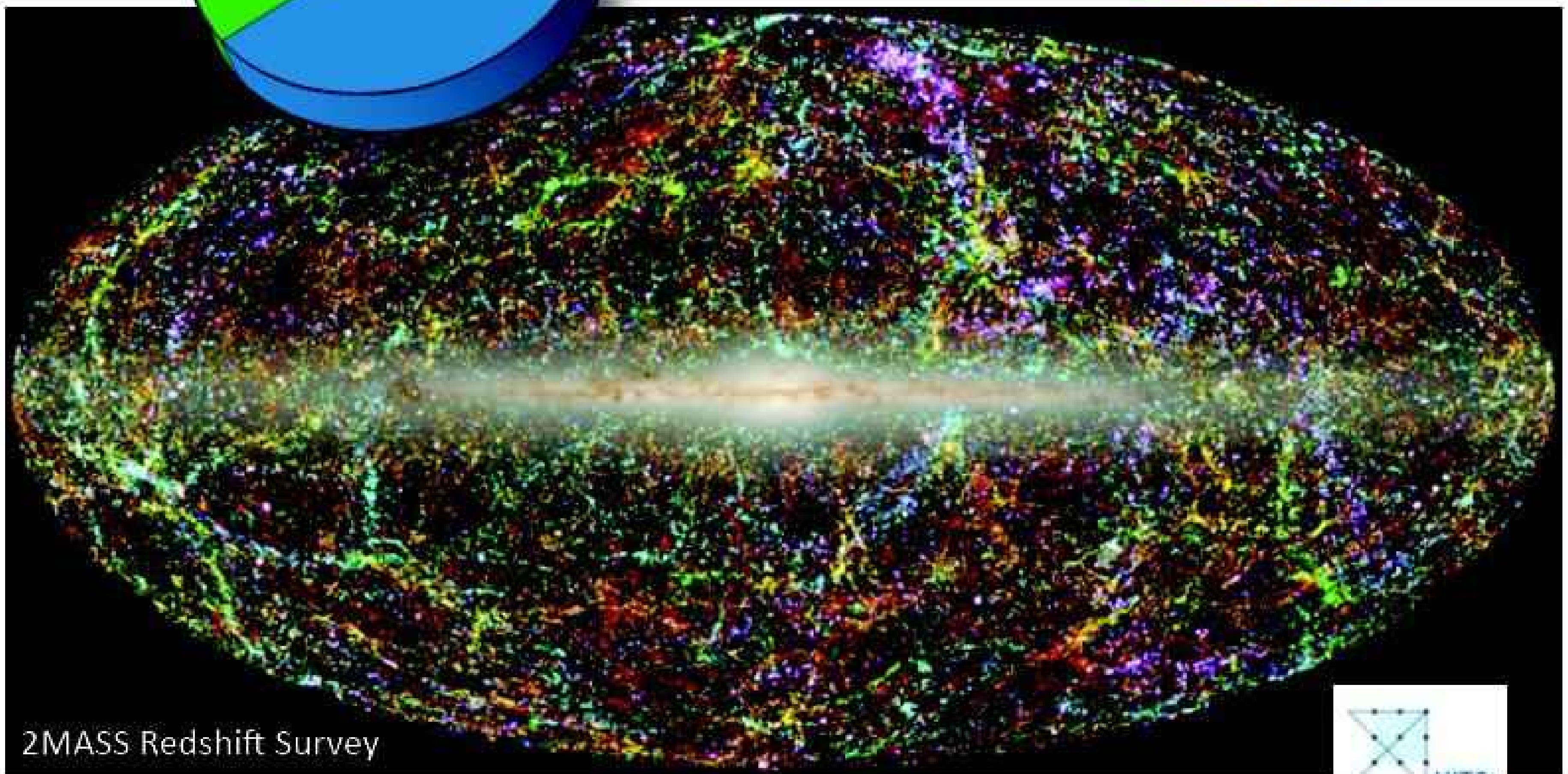


Millenium Simulaiton (Springel et al.)



Millenium Simulaiton (Springel et al.)

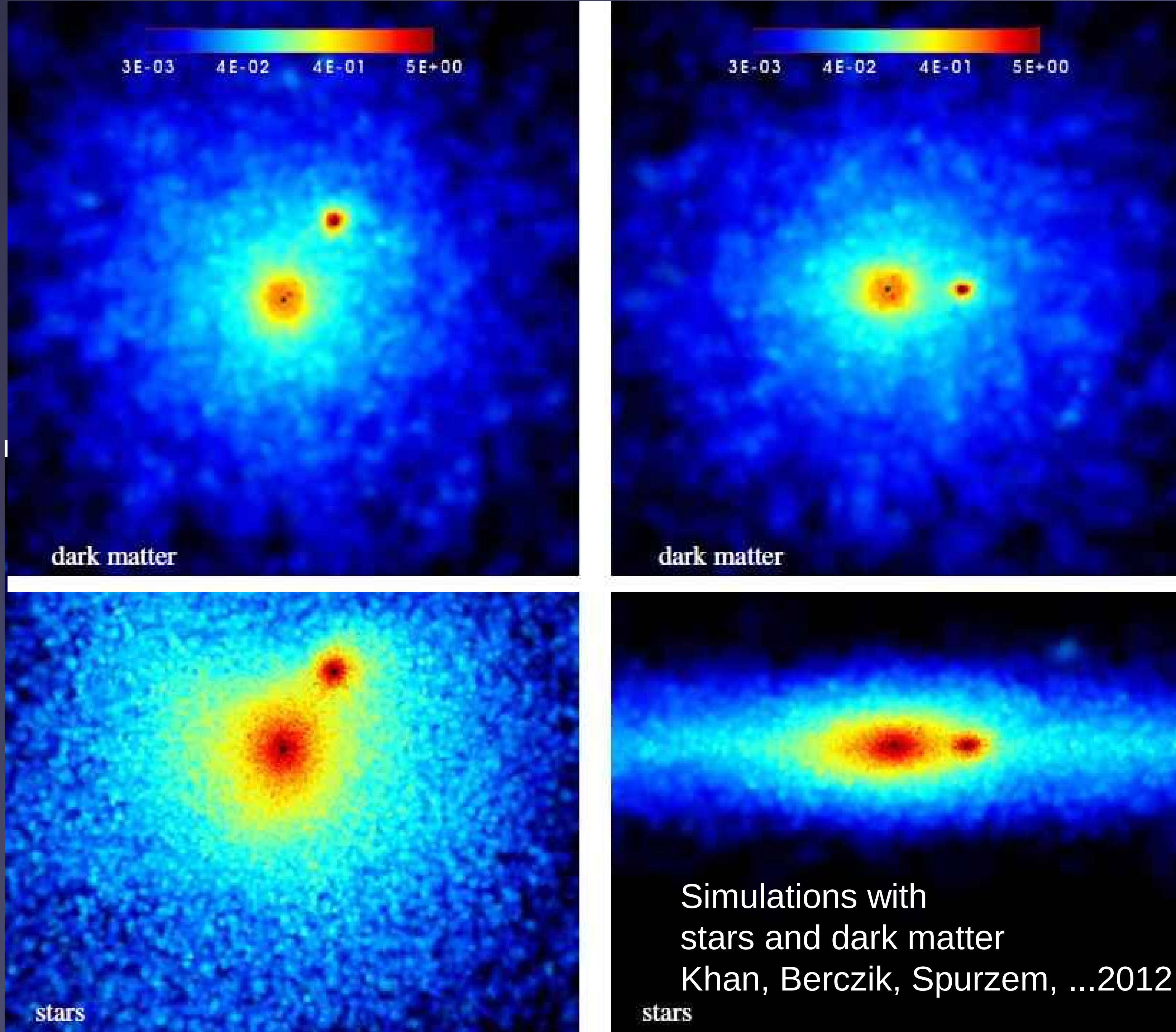




(Image: T.H. Jarrett (IPAC/SSC))

Computer Physics - Astrophysics

Black Holes in Galaxies and Star Clusters



Simulations with
stars and dark matter
Khan, Berczik, Spurzem, ...2012

Post-Newtonian Dynamics Gravitational Wave Templates

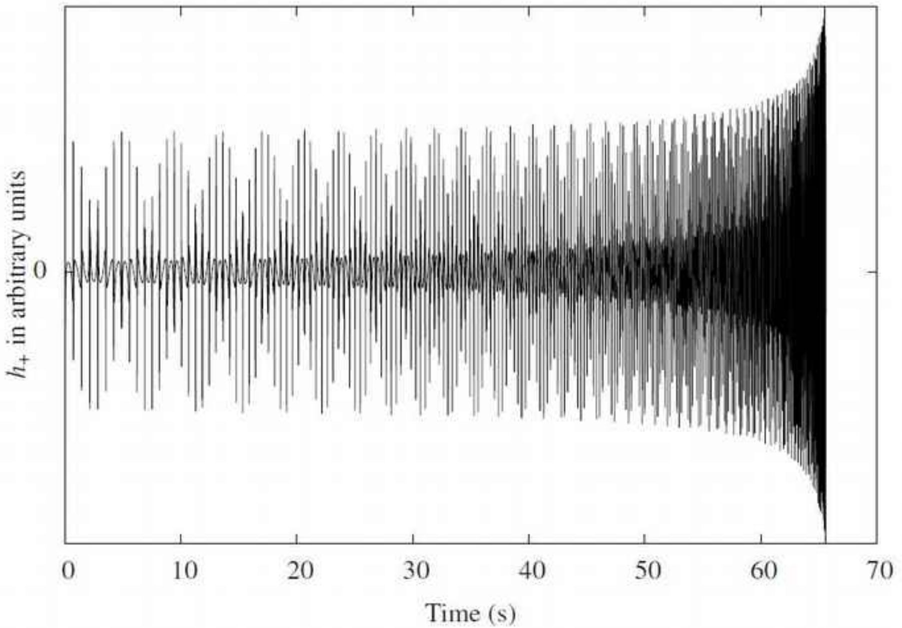


Figure 3.11: Waveform for two equal mass objects on a an orbit with $e = 0.5$.

Handle spin-orbit and spin-spin coupling (P.Brem, R. Spurzem, Univ. Heidelberg)

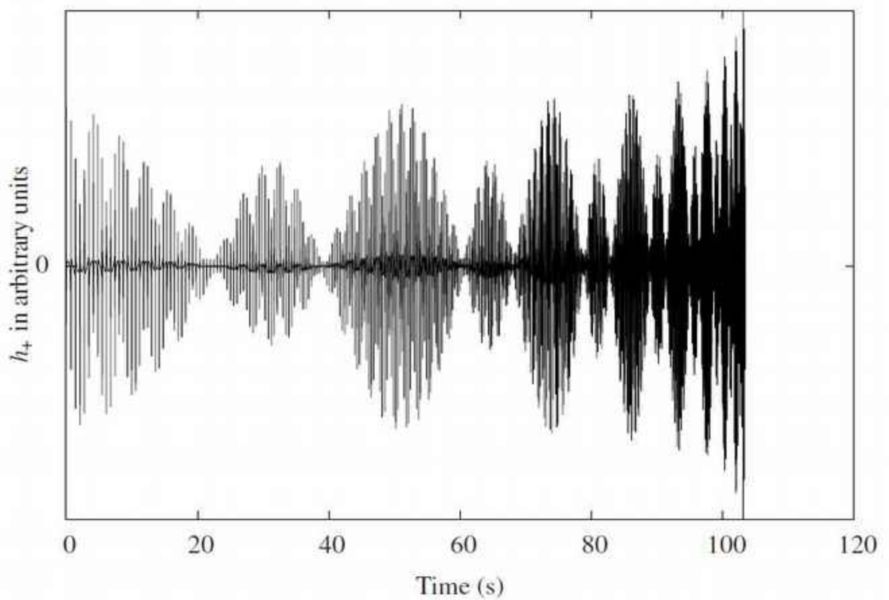


Figure 3.12: Waveform for two objects with a mass ratio of $q = 1/10$ on an orbit with $e = 0.5$ and spins $a_{1,x} = 1.0$, $a_{2,y} = 1.0$.

Example: VIRGO Detector in Cascina near Pisa, Italy

VIRGO – Pisa 3km
LIGO – Livingston, LA
Hanford, WA
1km
GEO600 – Hannover
600m
AIGO – Australien
(planned, 5 km)

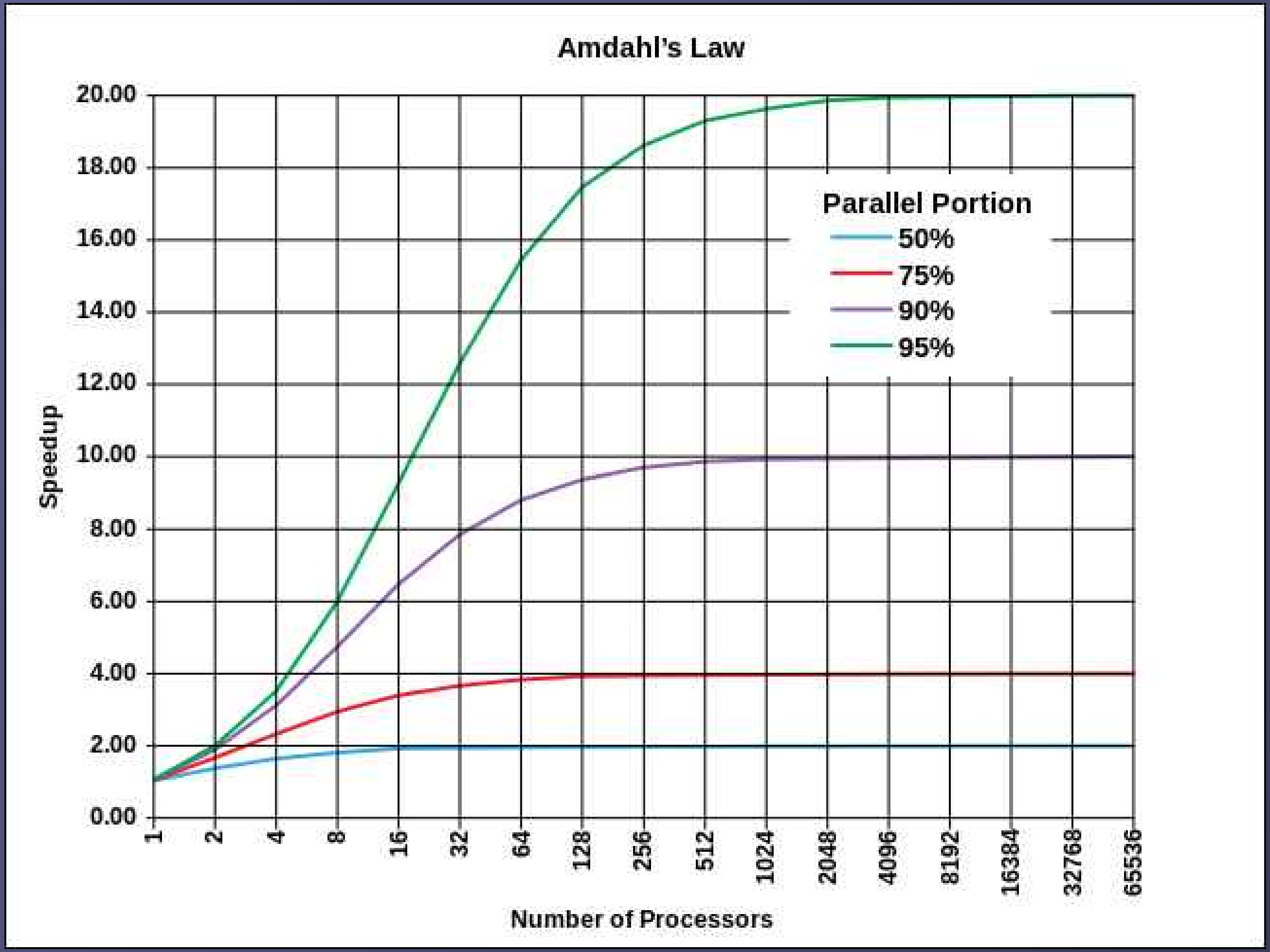
http://www.ligo-la.caltech.edu/
http://www.ego-gw.it
http://www.geo600.uni-hannover.de

Outreach to 50 Millionen light years (Neutron Stars)

Parallel Computing

Some basic ideas

Amdahl's Law (Gene Amdahl 1967)



Evolution according to Amdahl's law of the theoretical speedup of the execution of a program in function of the number of processors executing it, for different values of p. The speedup is limited by the serial part of the program. For example, if 95% of the program can be parallelized, the theoretical maximum speedup using parallel computing would be 20 times.

Calculate Amdahl's Law:

Let X be the part of my program (in terms of computing time) which can be parallelised. The sequential computing time Tseq is normalized to unity (1), and can be expressed as:

Tseq = 1 = X + (1-X)

The parallel computing time Tpar under ideal conditions (ideal load balancing, ultrafast communication):

Tpar = X/p + (1-X) with processor number (core number) p

Then the speed-up of the program S = Tseq / Tpar :

S = 1 / (1-X+X/p)

Note the limit if p is very large: S = 1/(1-X). And if X ~ 1: S ~ p

With communication overhead:

Tpar = X/p + (1-X) + Tcomm → **S = 1 / (1-X+X/p+Tcomm)**

If Tcomm independent of p we have for large p: S = 1 / (1-X + Tcomm) = const.

ΦGPU – NBODY Code

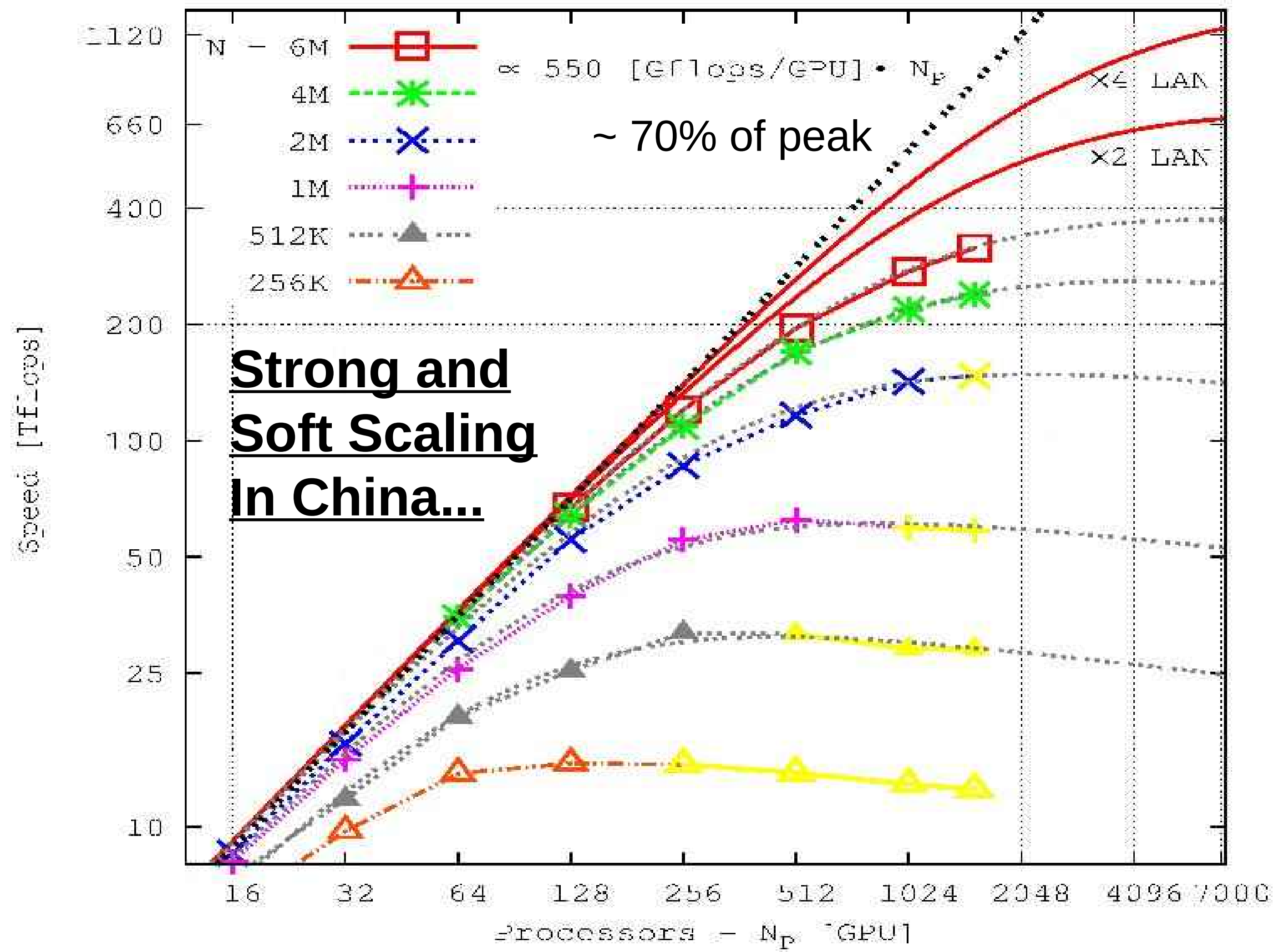
中国科学院国家天文台

National Astronomical Observatories, CAS

350 Teraflop/s
1600 GPUs .
440 cores
= 704.000
GPU-Cores

Using
Mole-8.5
of
IPE/CAS
Beijing

Berczik et al. 2013



中国科学院过程工程研究所

Institute Of Process Engineering, Chinese Academy Of Sciences

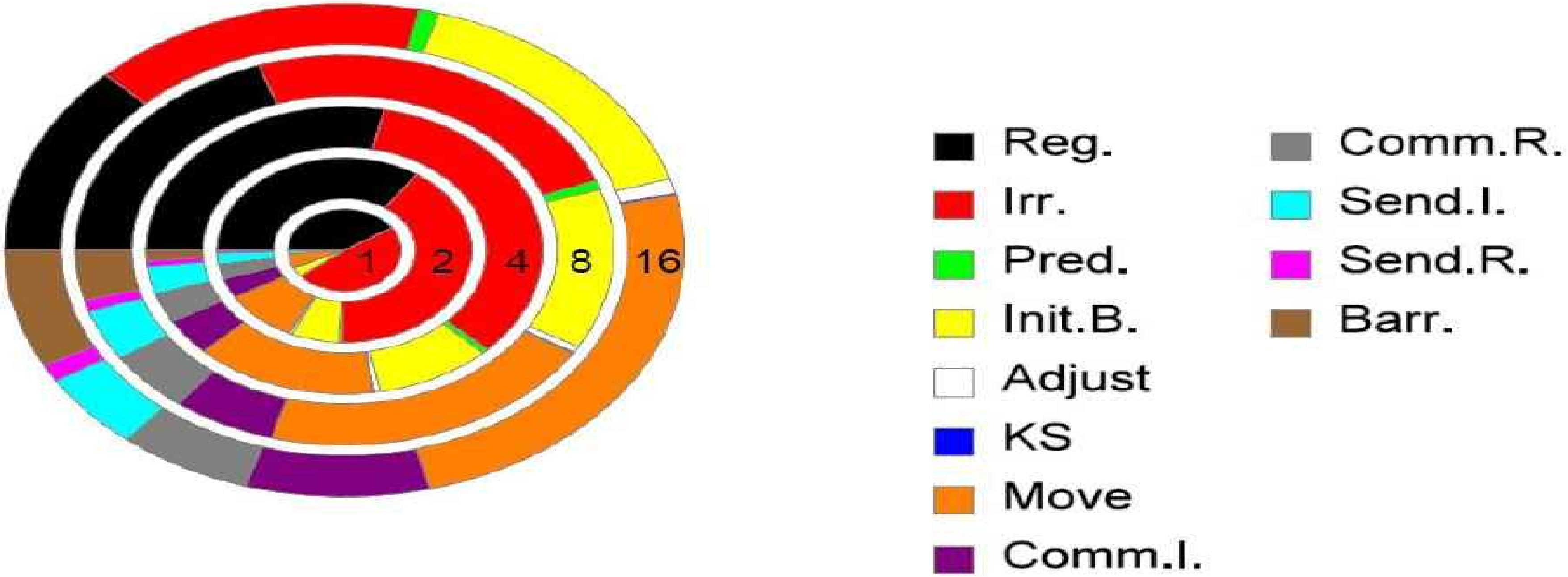
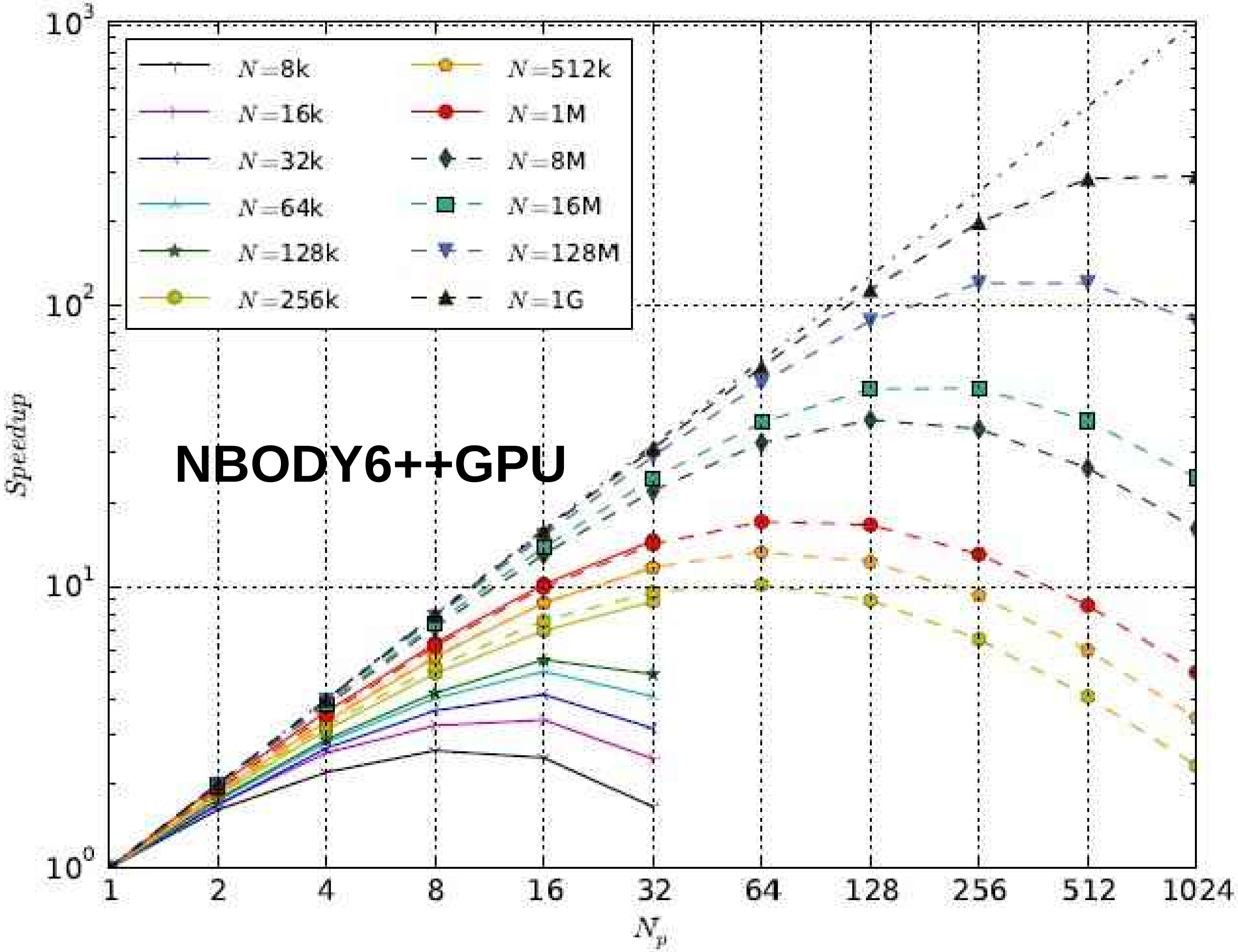


Table 1 Main components of NBODY6++

Description	Timing variable	Expected scaling N	Expected scaling N_p	Fitting value [sec]
Regular force computation	$T_{\rm reg}$	$\mathcal{O}(N_{\rm reg} \cdot N)$	$\mathcal{O}(N_p^{-1})$	$(2.2 \cdot 10^{-9} \cdot N^{2.11} + 10.43) \cdot N_p^{-1}$
Irregular force computation	$T_{\rm irr}$	$\mathcal{O}(N_{\rm irr} \cdot \langle N_{nb} \rangle)$	$\mathcal{O}(N_p^{-1})$	$(3.9 \cdot 10^{-7} \cdot N^{1.76} - 16.47) \cdot N_p^{-1}$
Prediction	$T_{\rm pre}$	$\mathcal{O}(N^{kn_p})$	$\mathcal{O}(N_p^{-kp_p})$	$(1.2 \cdot 10^{-6} \cdot N^{1.51} - 3.58) \cdot N_p^{-0.5}$
Data moving	$T_{\rm mov}$	$\mathcal{O}(N^{kn_{m1}})$	$\mathcal{O}(1)$	$2.5 \cdot 10^{-6} \cdot N^{1.29} - 0.28$
MPI communication (regular)	$T_{\rm mcr}$	$\mathcal{O}(N^{kn_{cr}})$	$\mathcal{O}(kp_{cr} \cdot \frac{N_p-1}{N_p})$	$(3.3 \cdot 10^{-6} \cdot N^{1.18} + 0.12)(1.5 \cdot \frac{N_p-1}{N_p})$
MPI communication (irregular)	$T_{\rm mci}$	$\mathcal{O}(N^{kn_{ci}})$	$\mathcal{O}(kp_{ci} \cdot \frac{N_p-1}{N_p})$	$(3.6 \cdot 10^{-7} \cdot N^{1.40} + 0.56)(1.5 \cdot \frac{N_p-1}{N_p})$
Synchronization	$T_{\rm syn}$	$\mathcal{O}(N^{kn_s})$	$\mathcal{O}(N_p^{kp_s})$	$(4.1 \cdot 10^{-8} \cdot N^{1.34} + 0.07) \cdot N_p$
Sequential parts on host	$T_{\rm host}$	$\mathcal{O}(N^{kn_h})$	$\mathcal{O}(1)$	$4.4 \cdot 10^{-7} \cdot N^{1.49} + 1.23$



Huang, Berczik, Spurzem, Res. Astron. Astroph. 2016, 16, 11.

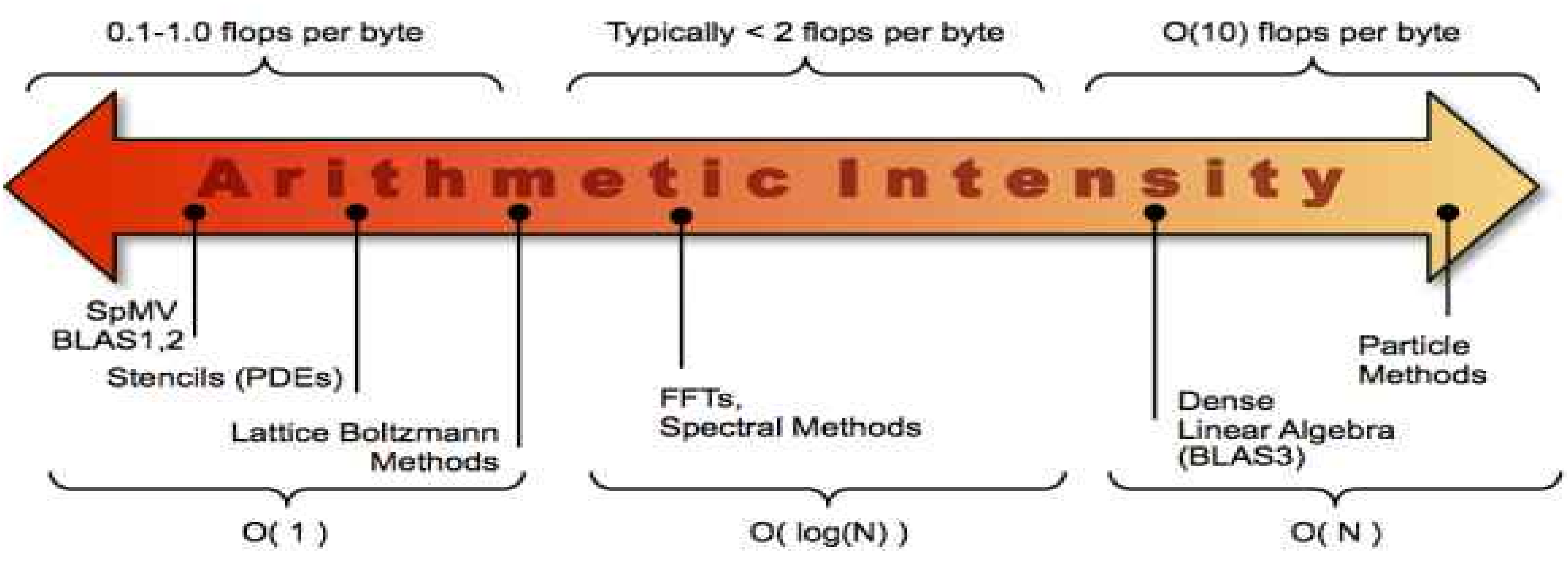
Fig. 2 The speed-up (S) of NBODY6++ as a function of particle number (N) and processor number (N_p). Solid points are the measured speed-up ratio between sequential and parallel wall-clock time, dash lines predict the performance of larger scale simulations further. The symbols used in figure have the magnitudes: $1k = 1,024$, $1M = 1k^2$ and $1G = 1k^3$.

Roofline Performance Model (LBL)

http://crd.lbl.gov/departments/computer-science/PAR/research/roofline

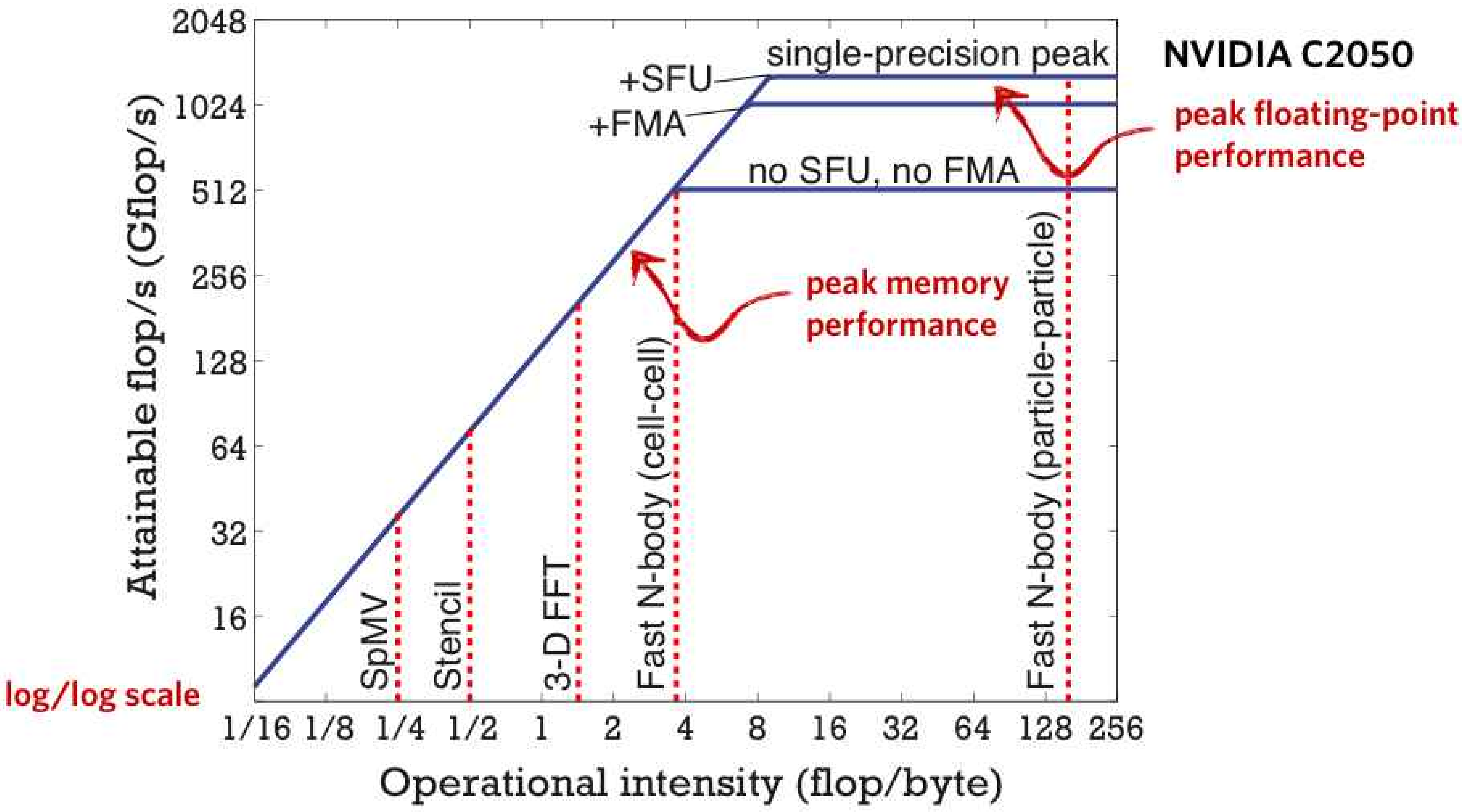
Arithmetic Intensity

The core parameter behind the Roofline model is Arithmetic Intensity. Arithmetic Intensity is the ratio of total floating-point operations to total data movement (bytes).



Roofline Performance Model (LBL)

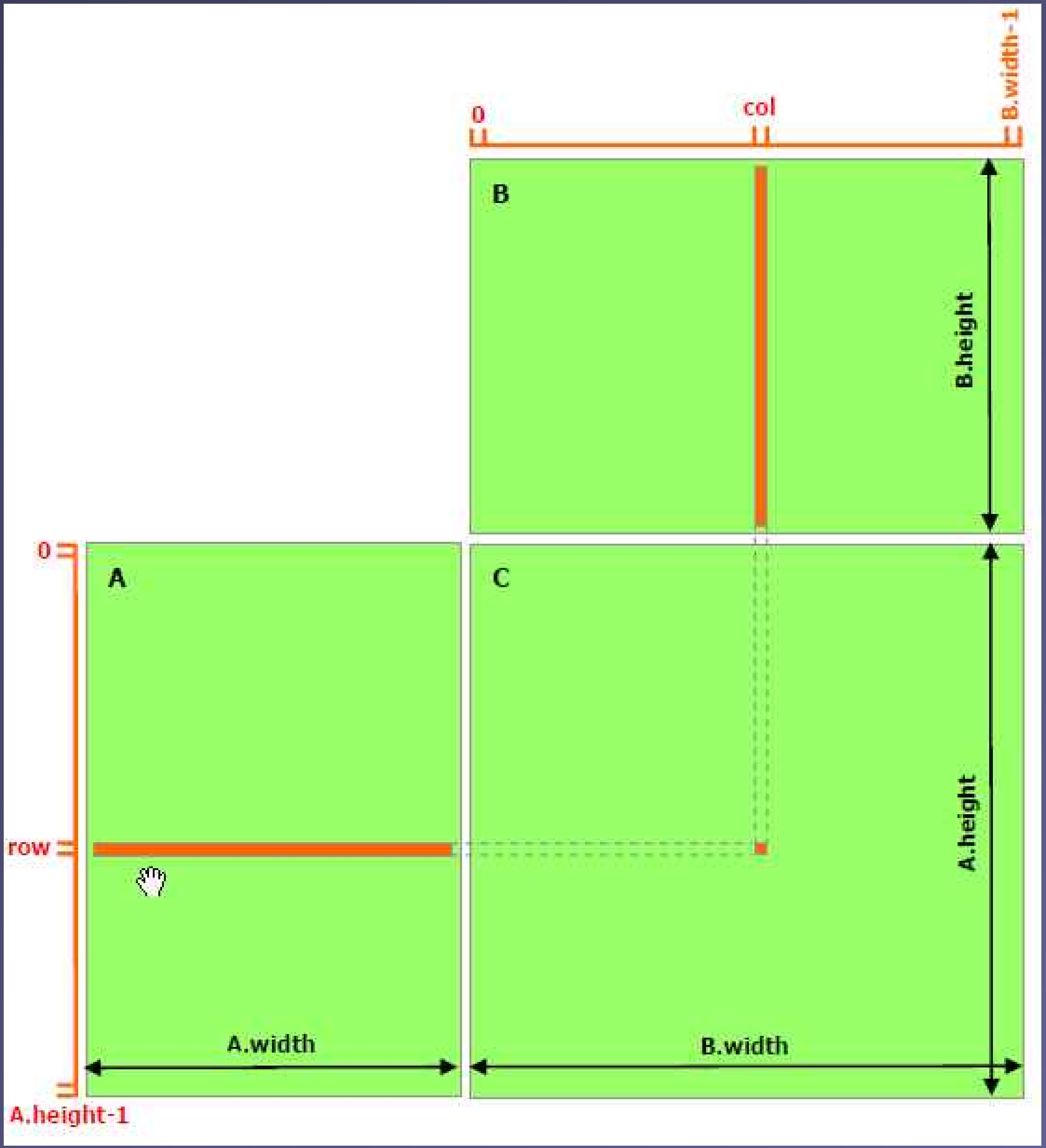
http://lorenabarba.com/wp-content/uploads/2012/01/roofline_slide.png



Parallel Computing

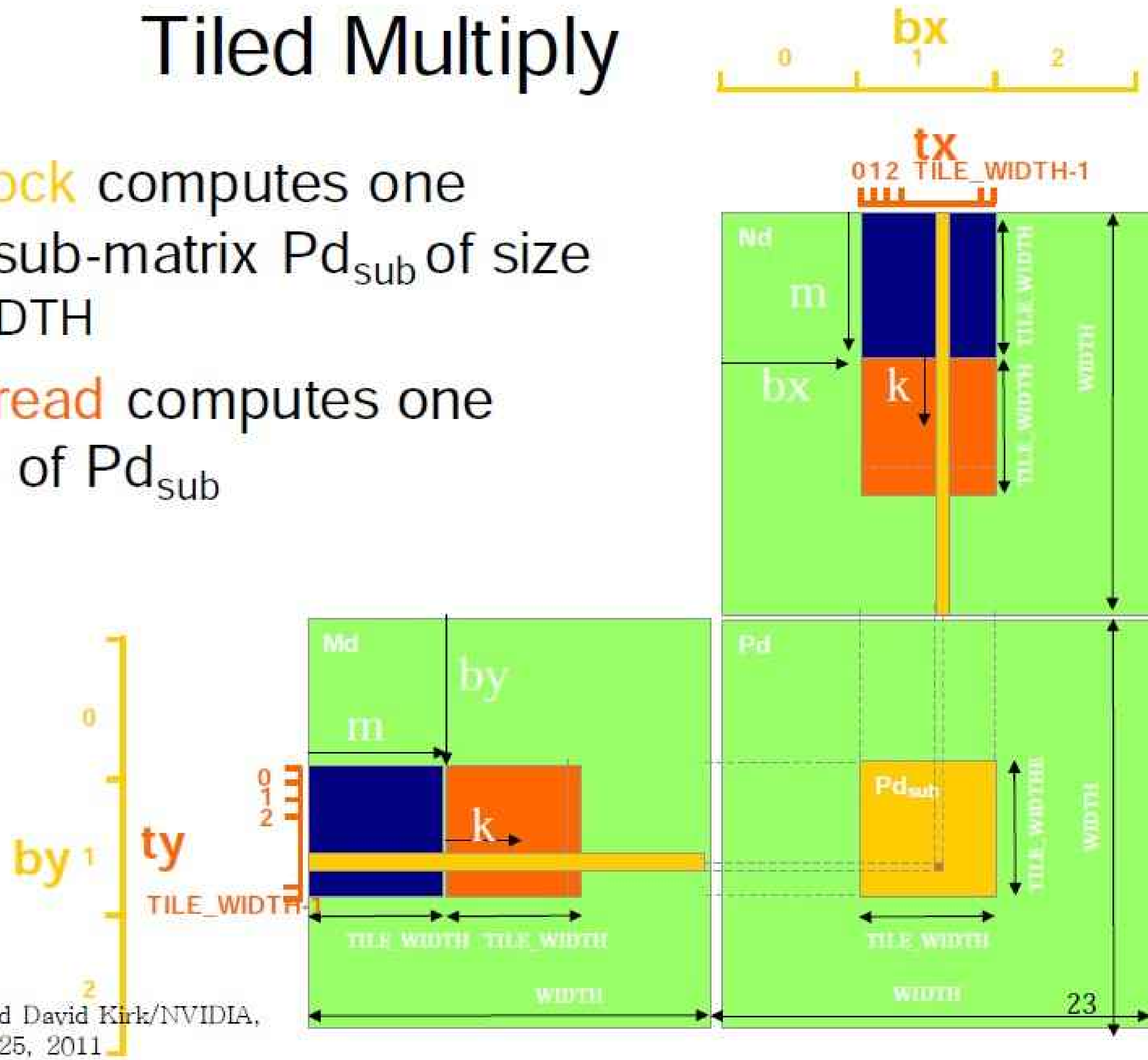
Matrix Multiply and Debugging

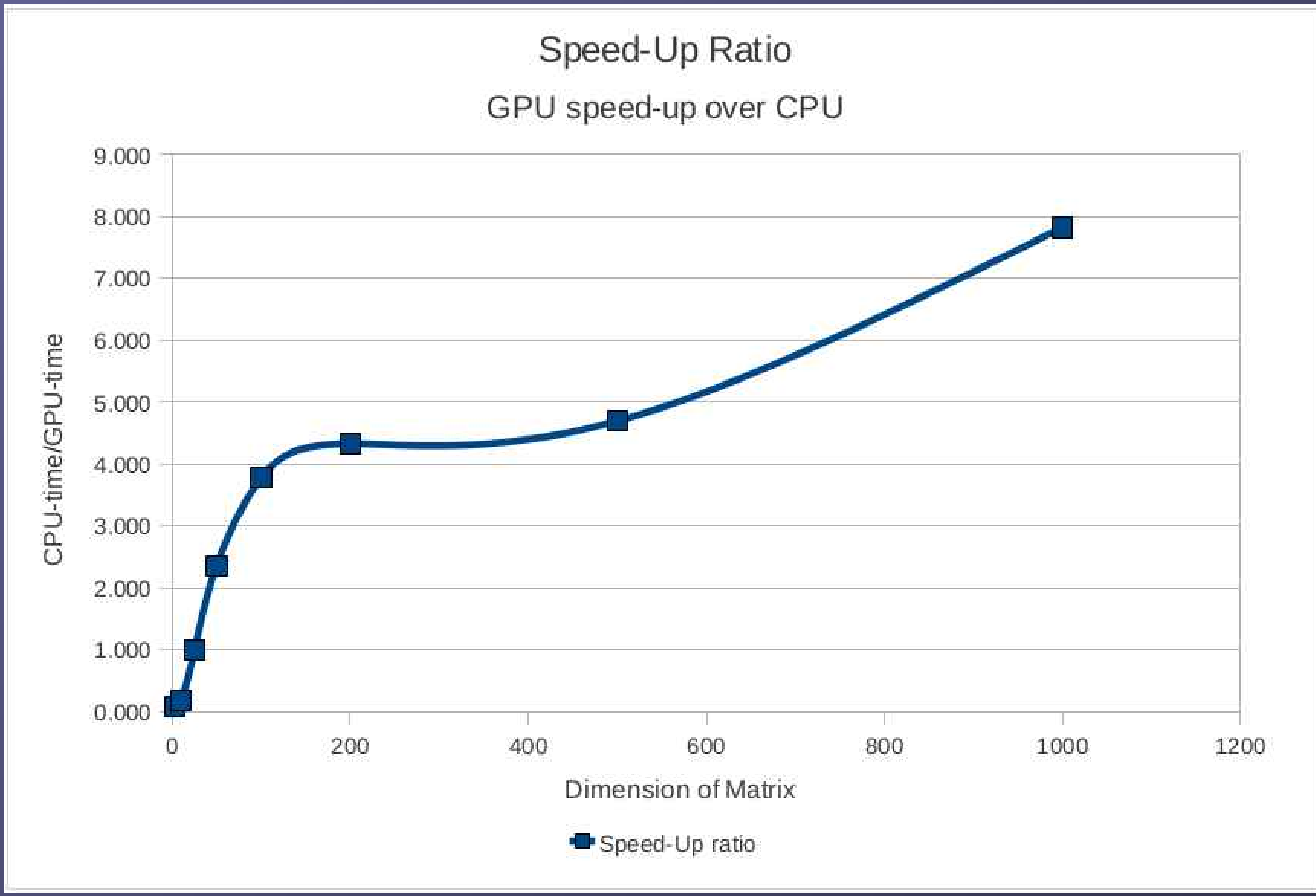
Intuitive multiply



Tiled Multiply

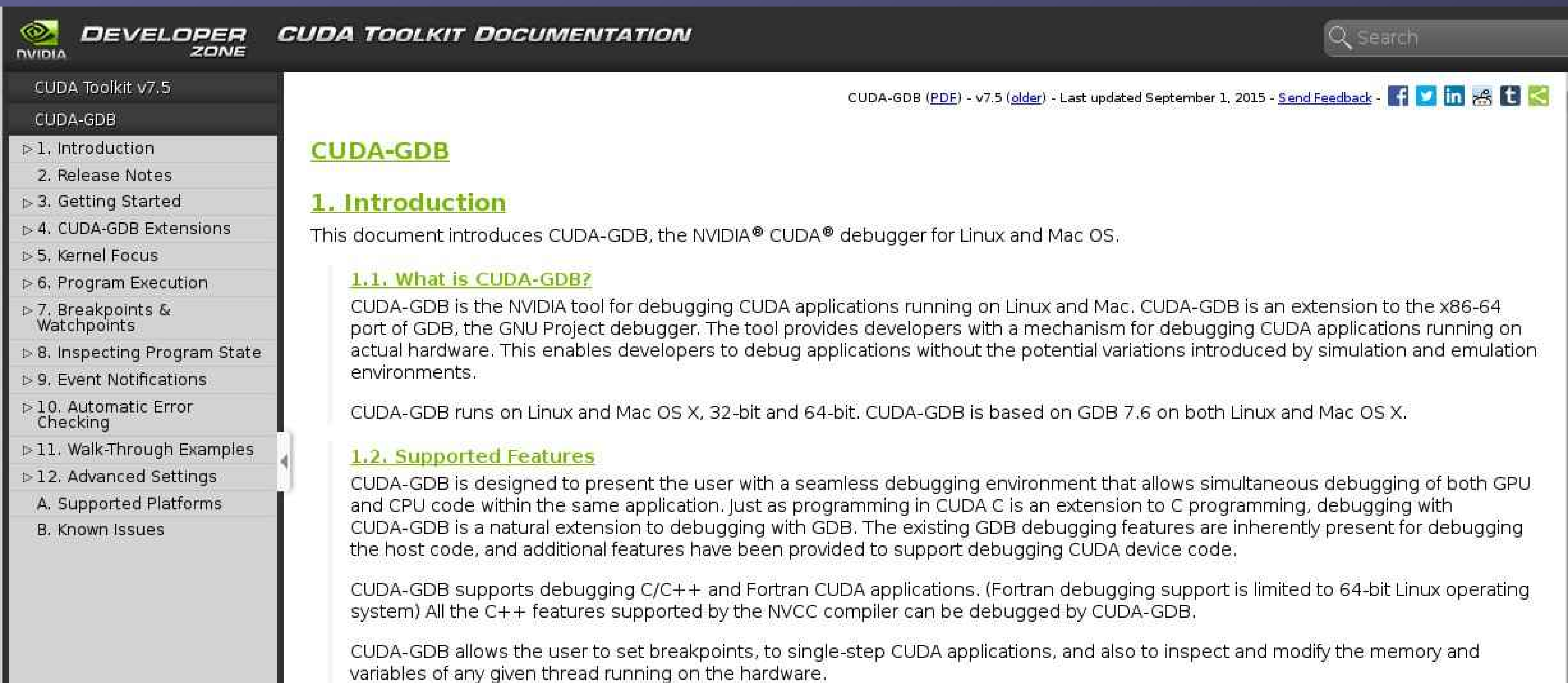
- Each block computes one square sub-matrix Pd_{sub} of size TILE_WIDTH
- Each thread computes one element of Pd_{sub}




Speed-Up Ratio
GPU speed-up over CPU
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000
0.000
CPU-time/GPU-time
0
200
400
600
800
1000
1200
Dimension of Matrix
Speed-Up ratio

CUDA – GNU Debugger – CUDA-gdb

http://docs.nvidia.com/cuda/cuda-gdb/index.html


NVIDIA DEVELOPER ZONE CUDA TOOLKIT DOCUMENTATION

Search

CUDA-GDB (PDF) - v7.5 (older) - Last updated September 1, 2015 - Send Feedback -

CUDA-GDB

1. Introduction

This document introduces CUDA-GDB, the NVIDIA® CUDA® debugger for Linux and Mac OS.

1.1. What is CUDA-GDB?

CUDA-GDB is the NVIDIA tool for debugging CUDA applications running on Linux and Mac. CUDA-GDB is an extension to the x86-64 port of GDB, the GNU Project debugger. The tool provides developers with a mechanism for debugging CUDA applications running on actual hardware. This enables developers to debug applications without the potential variations introduced by simulation and emulation environments.

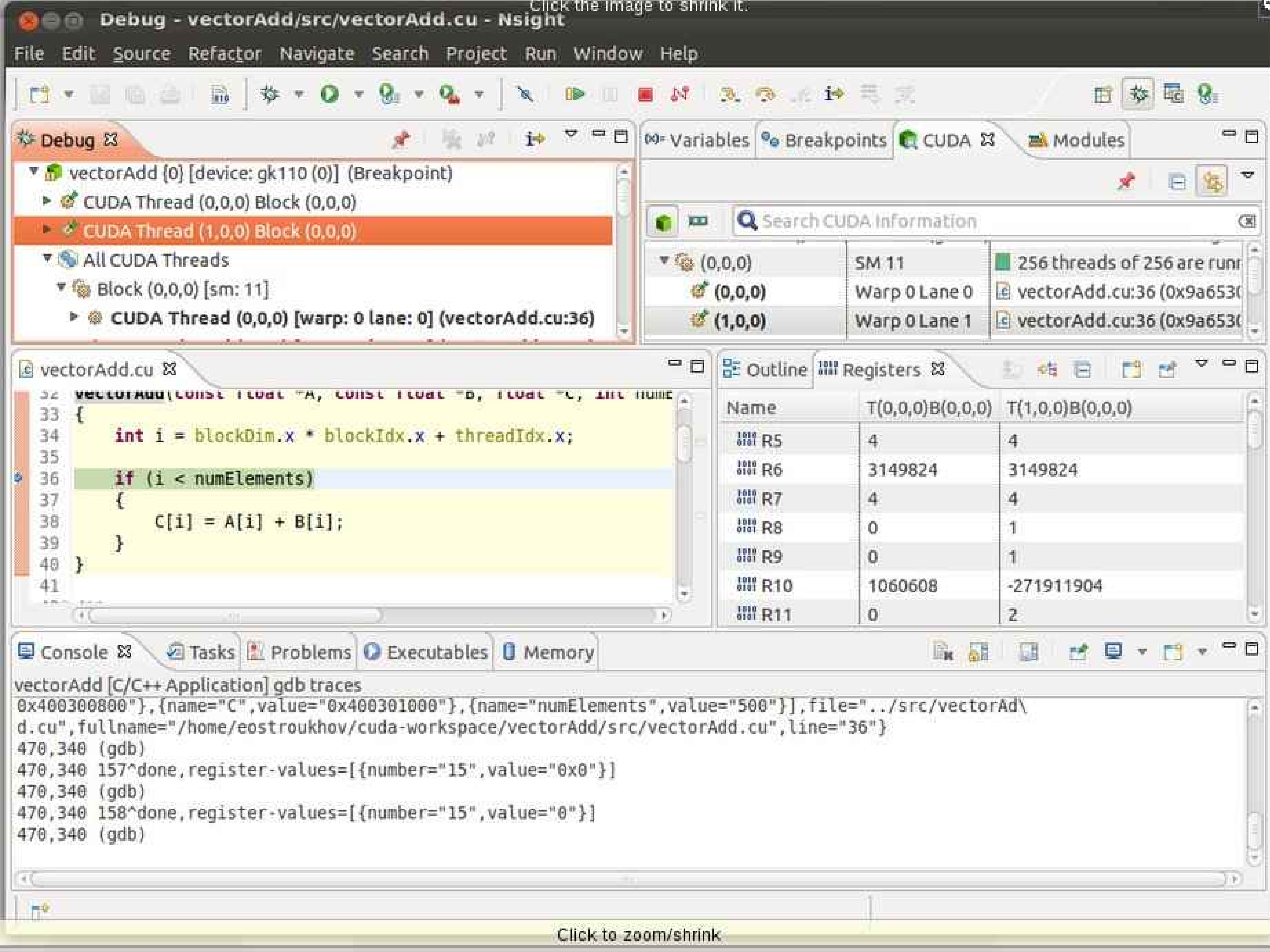
CUDA-GDB runs on Linux and Mac OS X, 32-bit and 64-bit. CUDA-GDB is based on GDB 7.6 on both Linux and Mac OS X.

1.2. Supported Features

CUDA-GDB is designed to present the user with a seamless debugging environment that allows simultaneous debugging of both GPU and CPU code within the same application. Just as programming in CUDA C is an extension to C programming, debugging with CUDA-GDB is a natural extension to debugging with GDB. The existing GDB debugging features are inherently present for debugging the host code, and additional features have been provided to support debugging CUDA device code.

CUDA-GDB supports debugging C/C++ and Fortran CUDA applications. (Fortran debugging support is limited to 64-bit Linux operating system) All the C++ features supported by the NVCC compiler can be debugged by CUDA-GDB.

CUDA-GDB allows the user to set breakpoints, to single-step CUDA applications, and also to inspect and modify the memory and variables of any given thread running on the hardware.


Click the image to shrink it.
Debug - vectorAdd/src/vectorAdd.cu - Nsight
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
vectorAdd (0) [device: gk110 (0)] (Breakpoint)
CUDA Thread (0,0,0) Block (0,0,0)
CUDA Thread (1,0,0) Block (0,0,0)
All CUDA Threads
Block (0,0,0) [sm: 11]
CUDA Thread (0,0,0) [warp: 0 lane: 0] (vectorAdd.cu:36)
Variables Breakpoints CUDA Modules
Search CUDA Information
(0,0,0) SM 11 256 threads of 256 are runr
(0,0,0) Warp 0 Lane 0 vectorAdd.cu:36 (0x9a653
(1,0,0) Warp 0 Lane 1 vectorAdd.cu:36 (0x9a653
vectorAdd.cu
33 {
34 int i = blockDim.x * blockIdx.x + threadIdx.x;
35
36 if (i < numElements)
37 {
38 C[i] = A[i] + B[i];
39 }
40 }
41
Outline Registers
Name T(0,0,0)B(0,0,0) T(1,0,0)B(0,0,0)
R5 4 4
R6 3149824 3149824
R7 4 4
R8 0 1
R9 0 1
R10 1060608 -271911904
R11 0 2
Console Tasks Problems Executables Memory
vectorAdd [C/C++ Application] gdb traces
0x400300800"},{name="C",value="0x400301000"},{name="numElements",value="500"}],file="../src/vectorAd\
d.cu",fullname="/home/eostroukhov/cuda-workspace/vectorAdd/src/vectorAdd.cu",line="36"}
470,340 (gdb)
470,340 157^done,register-values=[{number="15",value="0x0"}]
470,340 (gdb)
470,340 158^done,register-values=[{number="15",value="0"}]
470,340 (gdb)
Click to zoom/shrink

Additional deeper material:

Lectures by Prof. Wen-Mei Hwu Chicago in Berkeley 2012 and Beijing 2013, see **http://iccs.lbl.gov/workshops/tutorials.html** (down on page links to all lecture files, also available on request from spurzem@nao.cas.cn)

Lecture1: Computational thinking
Lecture2: Parallelism Scalability
Lecture3: Blocking Tiling
Lecture4: Coarsening Tiling
Lecture5: Data Optimization
Lecture6: Input Binning
Lecture7: Input Compaction
Lecture8: Privatization
See also:
http://freevideolectures.com/Course/2880/Advanced-algorithmic-techniques-for-GPUs/1

北京大學 PEKING UNIVERSITY