CEMRACS 2016

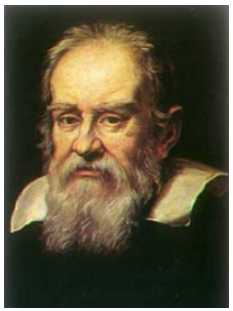
Numerical challenges in parallel scientific computing

July 18th - August 26th

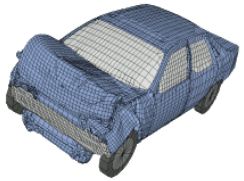
Algorithms for future emerging technologies

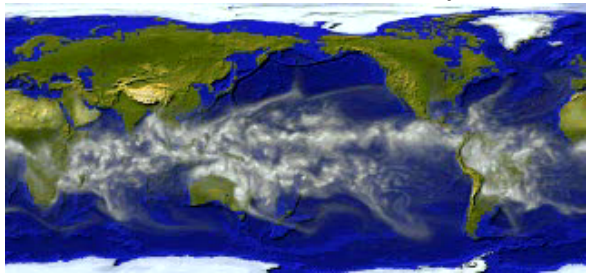
Jack Dongarra
University of Tennessee
Oak Ridge National Lab
University of Manchester

Simulation: The Third Pillar of Science



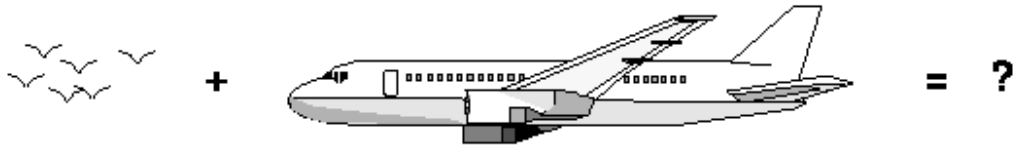
- Traditional scientific and engineering paradigm:
  1) Do theory or paper design.
  2) Perform experiments or build system.
- Limitations:
  - Too difficult -- build large wind tunnels.
  - Too expensive -- build a throw-away passenger jet.
  - Too slow -- wait for climate or galactic evolution.
  - Too dangerous -- weapons, drug design, climate experimentation.
- Computational science paradigm:
  3) Use high performance computer systems to simulate the phenomenon
     - Base on known physical laws and efficient numerical methods.

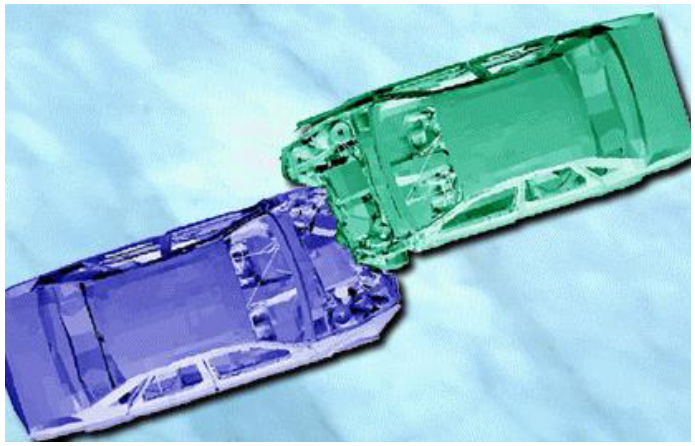


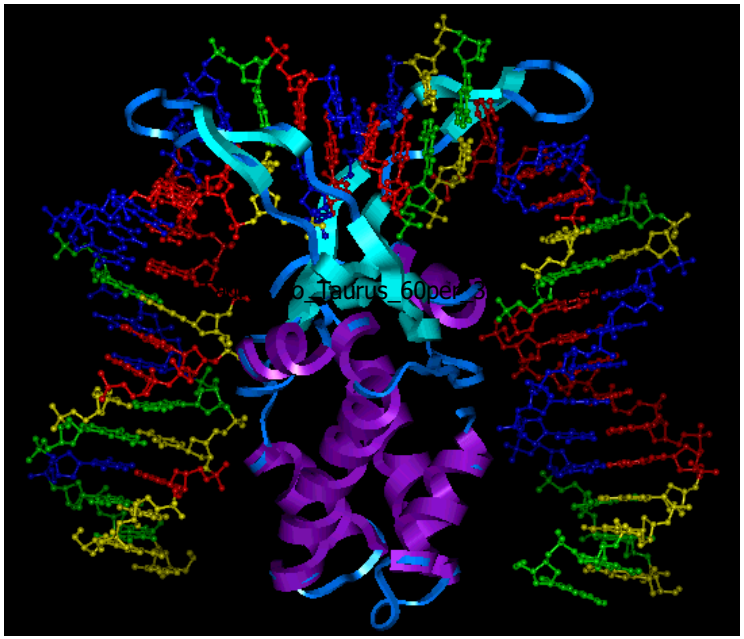


Why Turn to Simulation?

- When the problem is too . . .
  - Complex
  - Large / small
  - Expensive
  - Dangerous
- to do any other way.

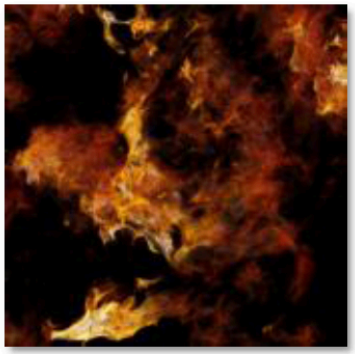






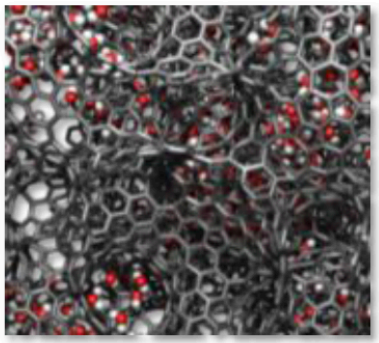
Computational Science

Applications to Energy



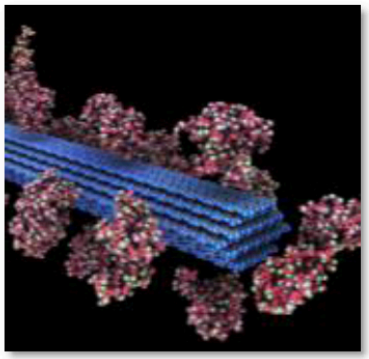
Turbulence

Understanding the statistical geometry of turbulent dispersion of pollutants in the environment.



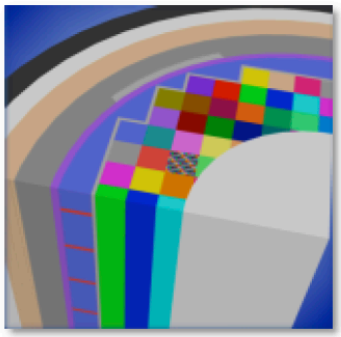
Energy Storage

Understanding the storage and flow of energy in next-generation nanostructured carbon tube supercapacitors



Biofuels

A comprehensive simulation model of lignocellulosic biomass to understand the bottleneck to sustainable and economical ethanol production.

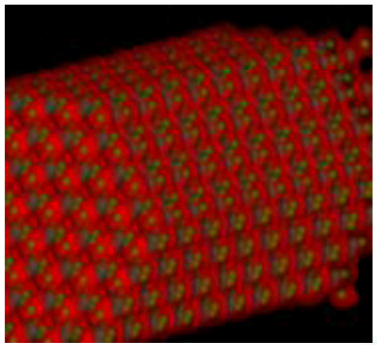


Nuclear Energy

High-fidelity predictive simulation tools for the design of next-generation nuclear reactors to safely increase operating margins.

Smart Truck

Aerodynamic forces account for ~53% of long haul truck fuel use. ORNL's Jaguar predicted 12% drag reduction and yielded EPA-certified 6.9% increase in fuel efficiency.

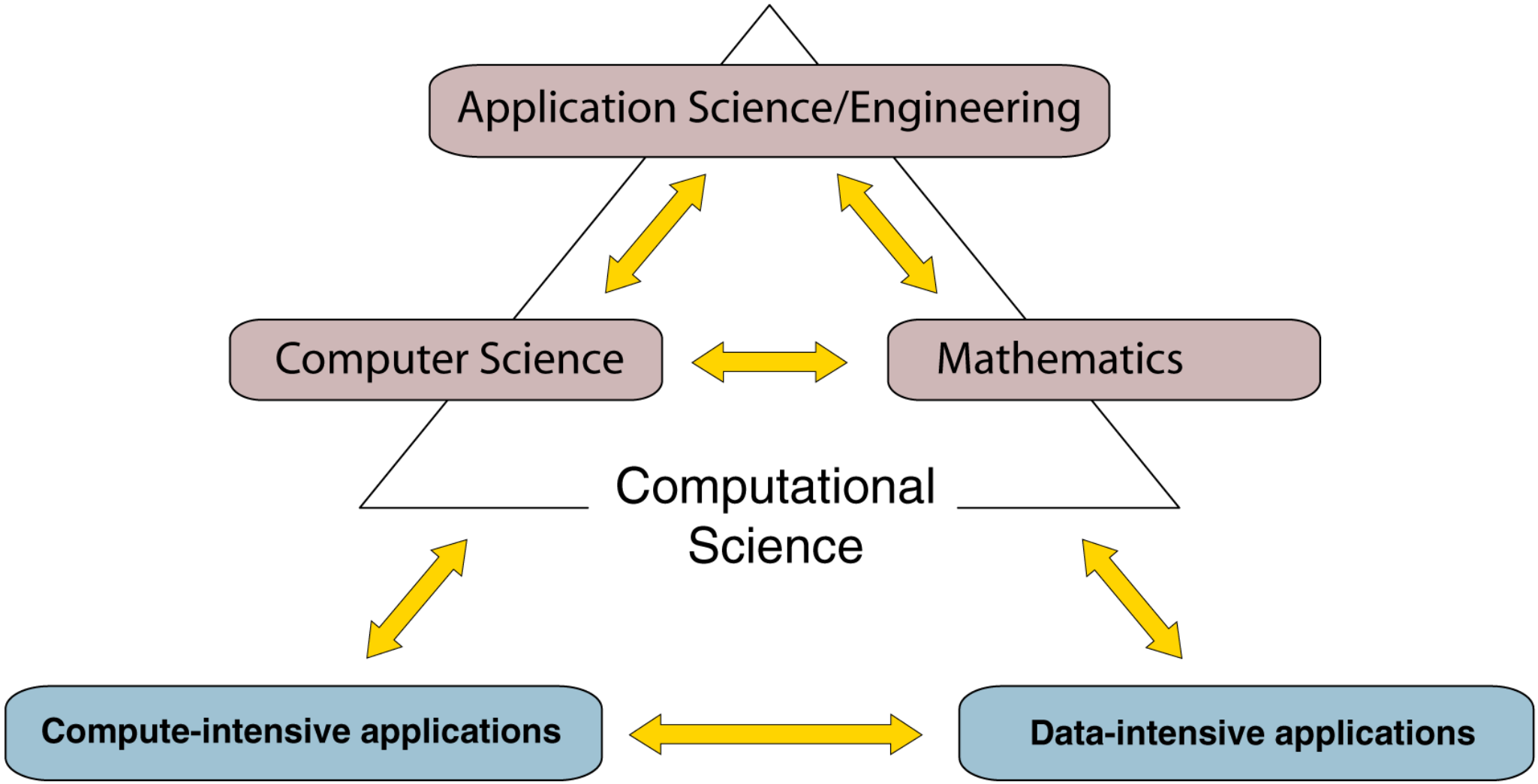


Nano Science

Understanding the atomic and electronic properties of nanostructures in next-generation photovoltaic solar cell materials.

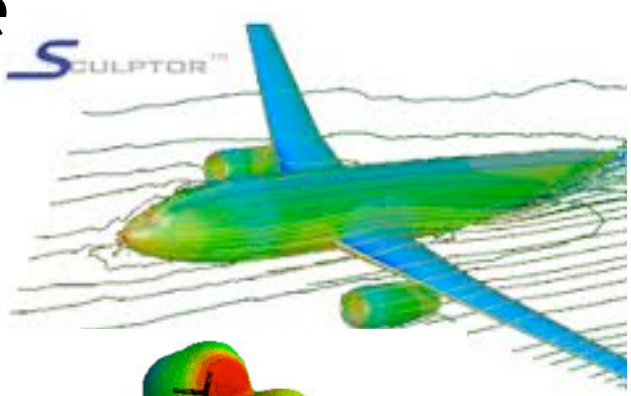
Source: Steven E. Koonin, DOE

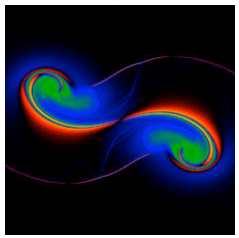
Computational Science Fuses Three Distinct Elements:

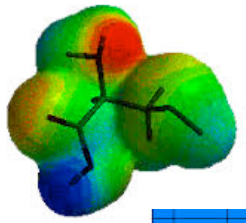


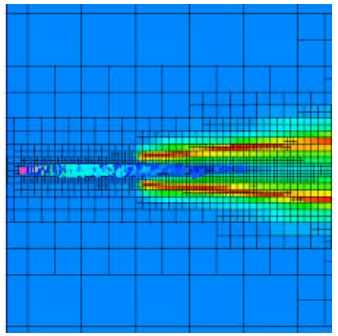
Wide Range of Applications that Depend on HPC is Incredibly Broad and Diverse

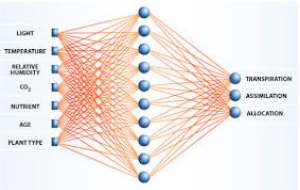
- Airplane wing design,
- Quantum chemistry,
- Geophysical flows,
- Noise reduction,
- Diffusion of solid bodies in a liquid,
- Adaptive mesh refinement,
- Computational materials research,
- Deep learning in neural networks,
- Stochastic simulation,
- Massively parallel data mining,
- …





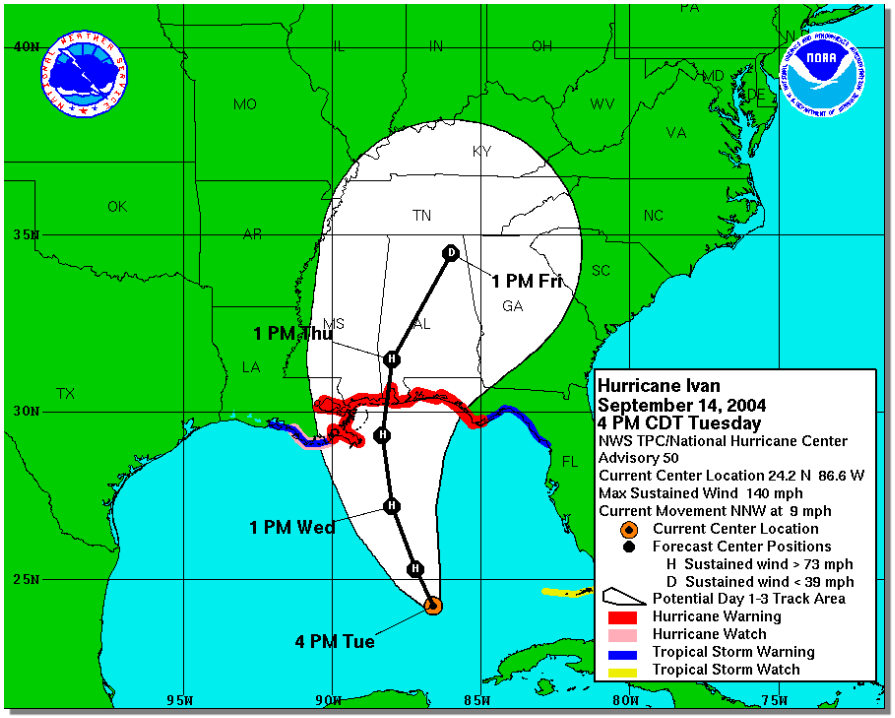






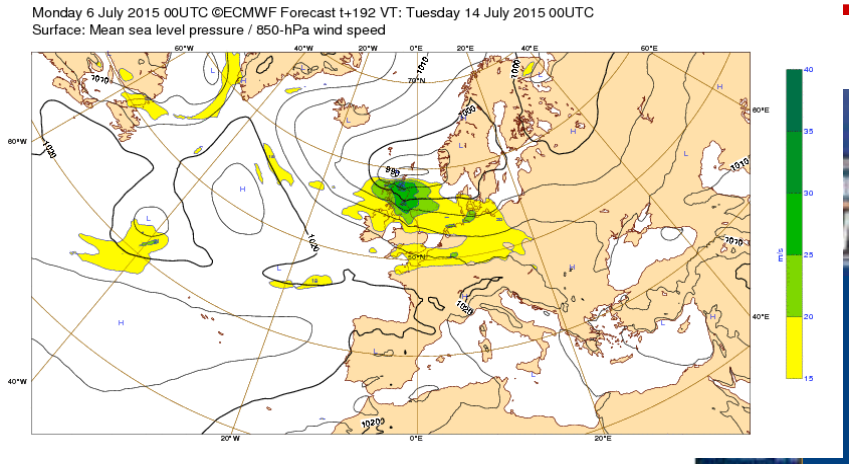
Weather and Economic Loss

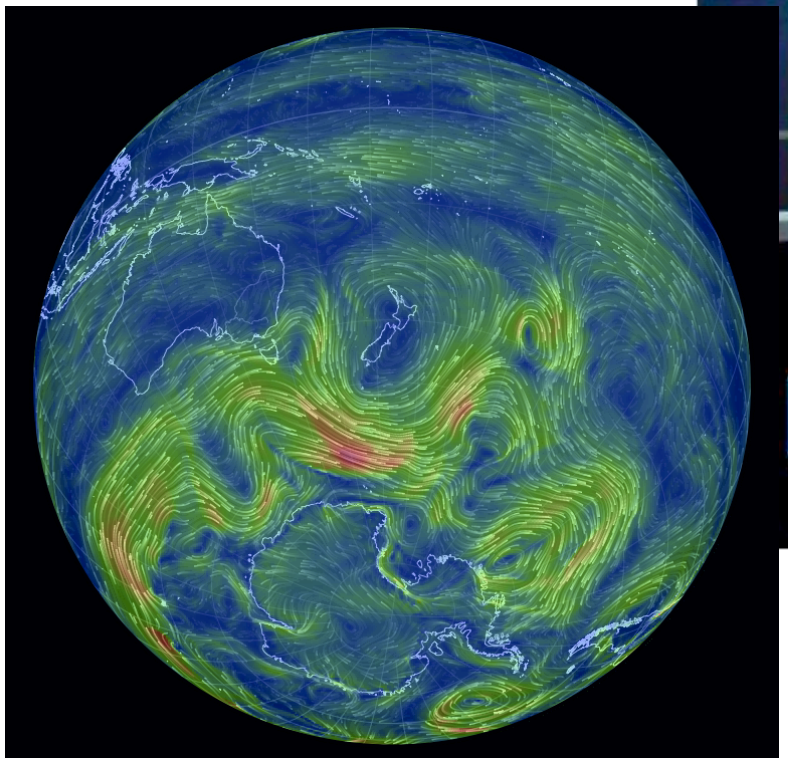
- $10T U.S. economy
  - 40% is adversely affected by weather and climate
- $1M in loss to evacuate each mile of coastline
  - we now over warn by 3X!
  - average over warning is 200 miles, or $200M per event
- Improved forecasts
  - lives saved and reduced cost
- LEAD
  - Linked Environments for Atmospheric Discovery
    - » Oklahoma, Indiana, UCAR, Colorado State, Howard, Alabama, Millersville, NCSA, North Carolina



Source: Kelvin Droegemeier, Oklahoma

Supercomputers Touch Everyone with Weather Forecasting





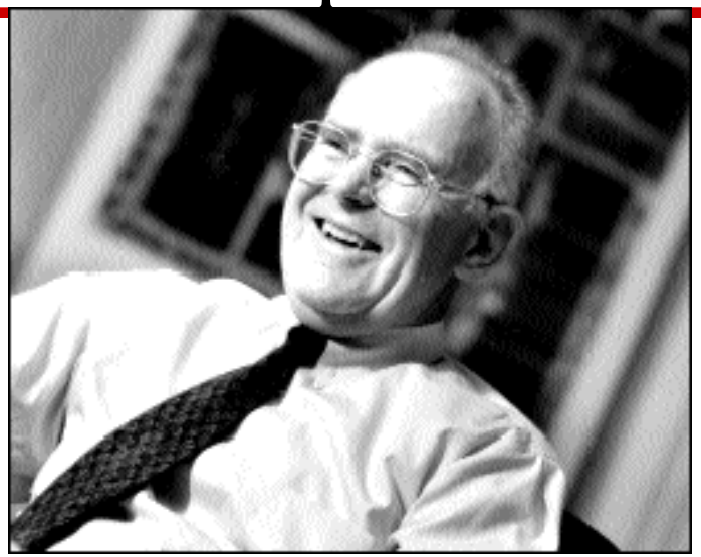
Look at the Fastest Computers

- **Strategic importance of supercomputing**
  - **Essential for scientific discovery**
  - **Critical for national security**
  - **Fundamental contributor to the economy and competitiveness through use in engineering and manufacturing**
- **Supercomputers are *the tool for solving the most challenging problems through simulations***

High-Performance Computing Today

- In the past decade, the world has experienced one of the most exciting periods in computer development.
- Microprocessors have become smaller, denser, and more powerful.
- The result is that microprocessor-based supercomputing is rapidly becoming the technology of preference in attacking some of the most important problems of science and engineering.

Technology Trends: Microprocessor Ca

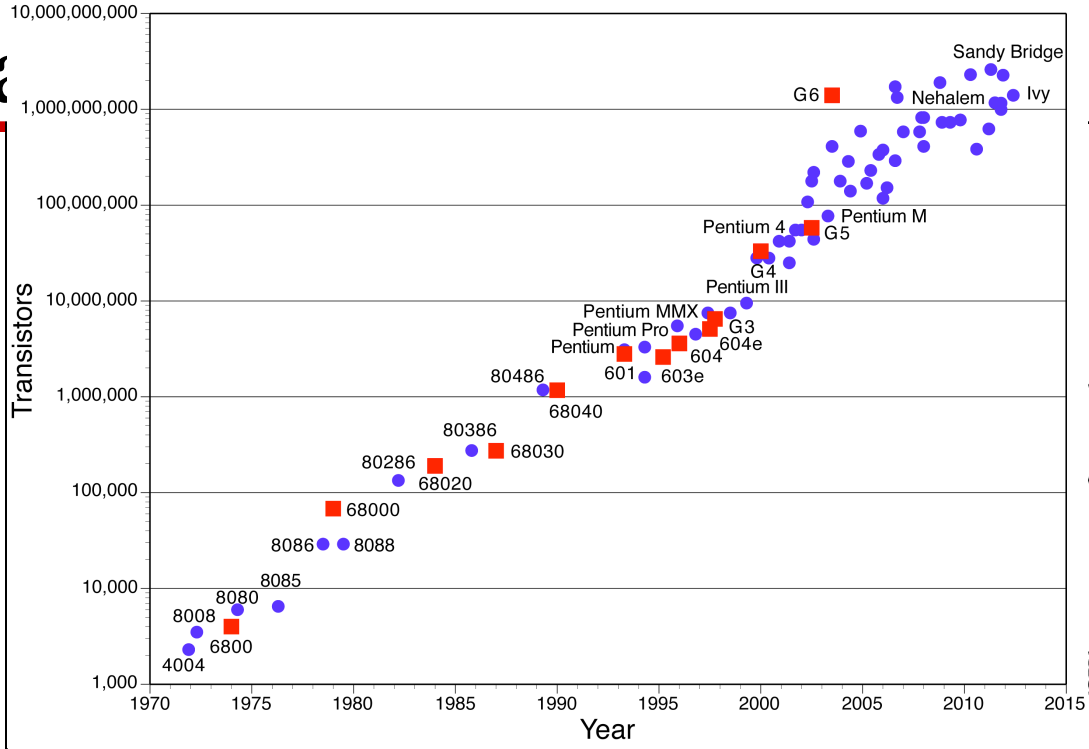


Gordon Moore (co-founder of Intel) Electronics Magazine, 1965

Number of devices/chip doubles every 18 months

2X transistors/Chip Every 1.5 years

Called "Moore's Law"



The future of integrated electronics is the future of electronics itself. The advantages of integration will bring about a proliferation of electronics, pushing this science into many new areas.

Integrated circuits will lead to such wonders as home computers—or at least terminals connected to a central computer—automatic controls for automobiles, and personal portable communications equipment. The electronic wristwatch needs only a display to be feasible today.

But the biggest potential lies in the production of large systems. In telephone communications, integrated circuits in digital filters will separate channels on multiplex equipment. Integrated circuits will also switch telephone circuits and perform data processing.

Computers will be more powerful, and will be organized in completely different ways. For example, memories built of integrated electronics may be distributed throughout the machine instead of being concentrated in a central unit. In addition, the improved reliability made possible by integrated circuits will allow the construction of larger processing units. Machines similar to those in existence today will be built at lower costs and with faster turn-around.

Present and future

By integrated electronics, I mean all the various technologies which are referred to as microelectronics today as well as any additional ones that result in electronics functions supplied to the user as irreducible units. These technologies were first investigated in the late 1950's. The object was to miniaturize electronics equipment to include increasingly complex electronic functions in limited space with minimum weight. Several approaches evolved, including microassembly techniques for individual components, thin-film structures and semiconductor integrated circuits.

Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the various approaches.

The advocates of semiconductor integrated circuitry are already using the improved characteristics of thin-film resistors by applying such films directly to an active semiconductor substrate. Those advocating a technology based upon

The author

Dr. Gordon E. Moore is one of the new breed of electronic engineers, schooled in the physical sciences rather than in electronics. He earned a B.S. degree in chemistry from the

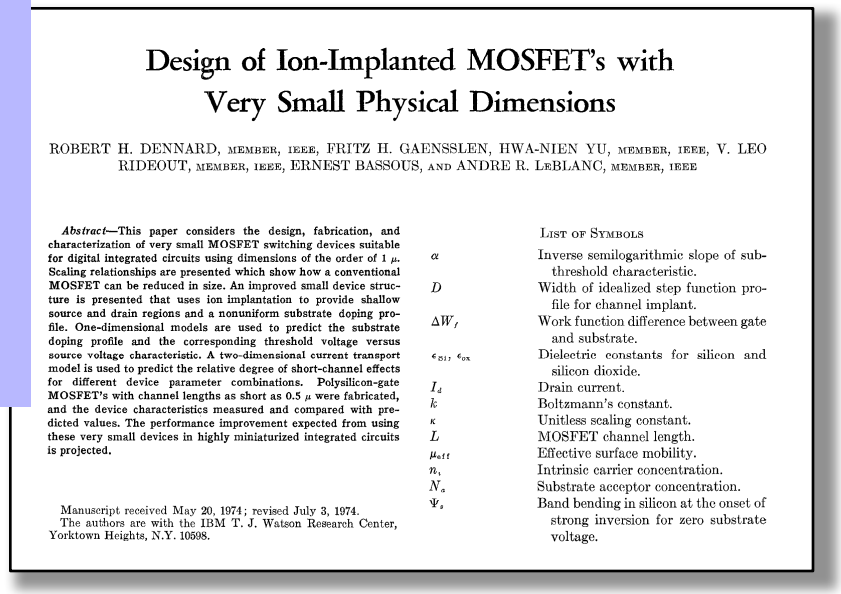
Moore's *Secret Sauce*: Dennard Scaling

Moore's Law put lots more transistors on a chip…but it's Dennard's Law that made them useful

Dennard observed that voltage and current should be proportional to the linear dimensions of a transistor

Dennard Scaling :

- Decrease feature size by a factor of λ and decrease voltage by a factor of λ ; then
- # transistors increase by λ^2
- Clock speed increases by λ
- **Energy consumption does not change**



Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions

ROBERT H. DENNARD, MEMBER, IEEE, FRITZ H. GAENSSLEN, HWA-NIEN YU, MEMBER, IEEE, V. LEO RIDEOUT, MEMBER, IEEE, ERNEST BASSOUS, AND ANDRE R. LEBLANC, MEMBER, IEEE

Abstract—This paper considers the design, fabrication, and characterization of very small MOSFET switching devices suitable for digital integrated circuits using dimensions of the order of 1 μ. Scaling relationships are presented which show how a conventional MOSFET can be reduced in size. An improved small device structure is presented that uses ion implantation to provide shallow source and drain regions and a nonuniform substrate doping profile. One-dimensional models are used to predict the substrate doping profile and the corresponding threshold voltage versus source voltage characteristic. A two-dimensional current transport model is used to predict the relative degree of short-channel effects for different device parameter combinations. Polysilicon-gate MOSFET's with channel lengths as short as 0.5 μ were fabricated, and the device characteristics measured and compared with predicted values. The performance improvement expected from using these very small devices in highly miniaturized integrated circuits is projected.

Manuscript received May 20, 1974; revised July 3, 1974.
The authors are with the IBM T. J. Watson Research Center, Yorktown Heights, N.Y. 10598.

List of Symbols

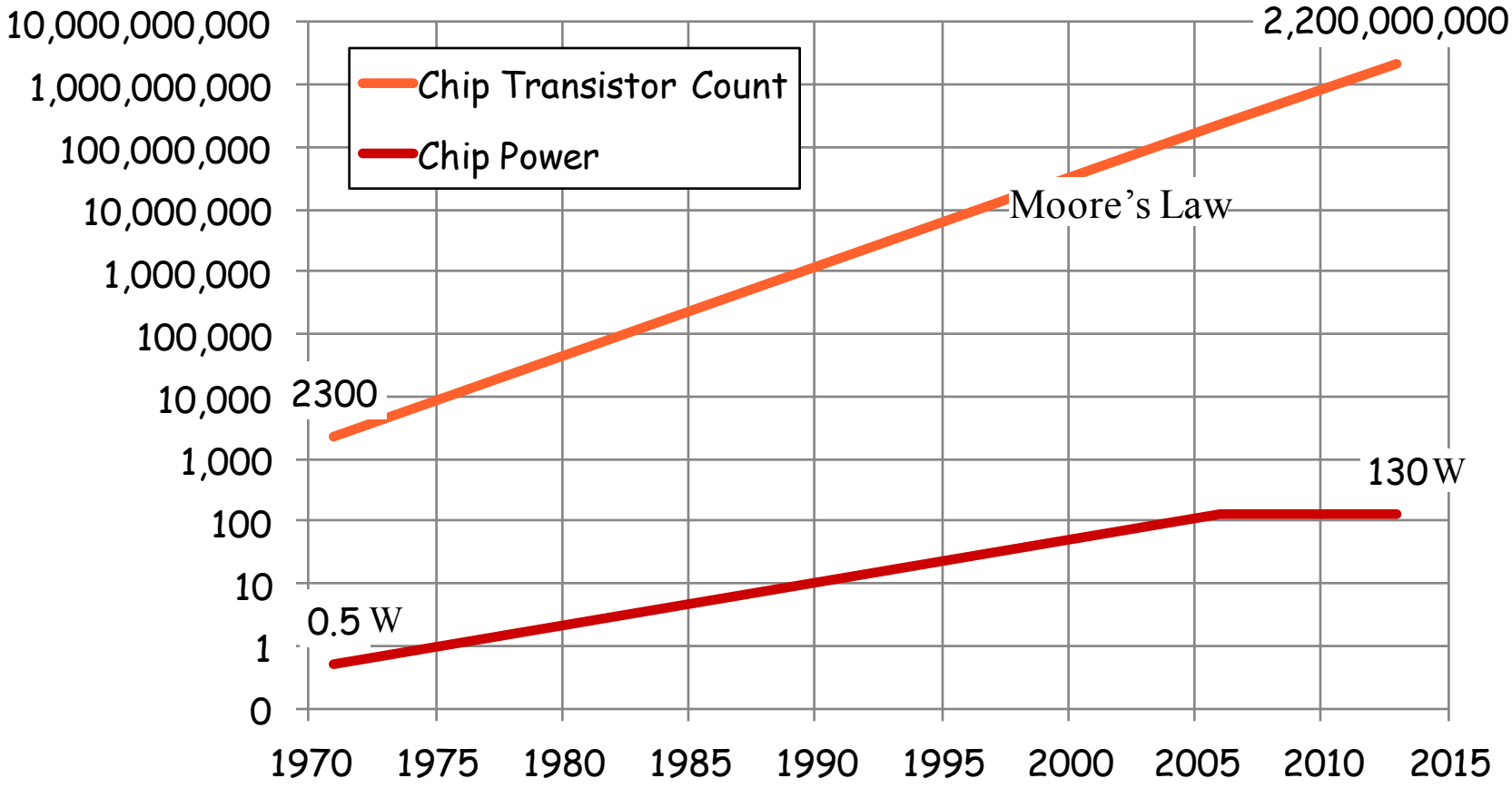
α	Inverse semilogarithmic slope of subthreshold characteristic.
D	Width of idealized step function profile for channel implant.
ΔW_f	Work function difference between gate and substrate.
$\epsilon_{si}, \epsilon_{ox}$	Dielectric constants for silicon and silicon dioxide.
I_d	Drain current.
k	Boltzmann's constant.
κ	Unitless scaling constant.
L	MOSFET channel length.
μ_{eff}	Effective surface mobility.
n_i	Intrinsic carrier concentration.
N_a	Substrate acceptor concentration.
Ψ_s	Band bending in silicon at the onset of strong inversion for zero substrate voltage.

2x transistor count
40% faster
50% more efficient

[Dennard, Gaensslen, Yu, Rideout, Bassous, Leblanc, **IEEE JSSC**, 1974]

Unfortunately Dennard Scaling is Over: What is the Catch?

Breakdown is the result of small feature sizes, current leakage poses greater challenges, and also causes the chip to heat up

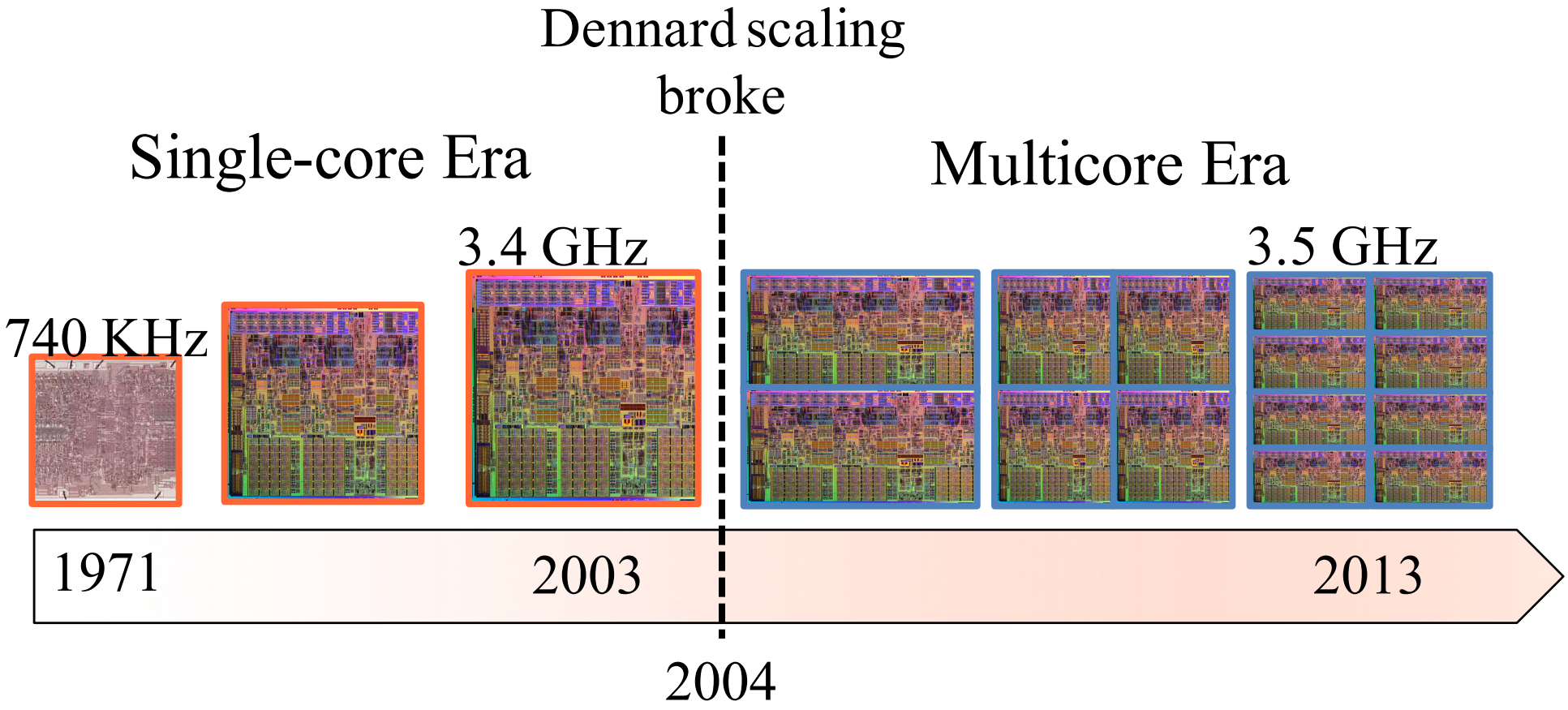


Powering the transistors without melting the chip

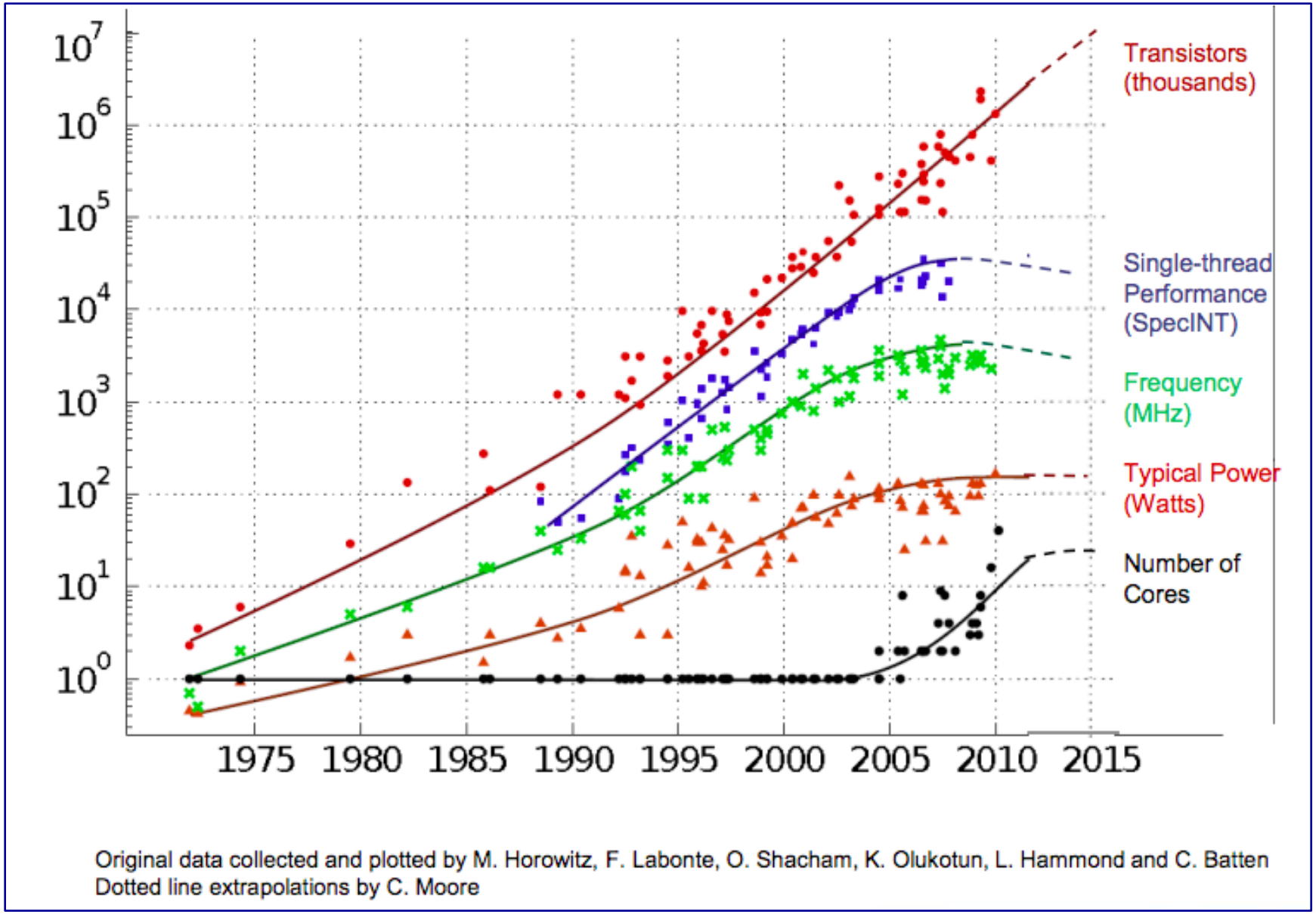
Dennard Scaling Over

Evolution of processors

The primary reason cited for the breakdown is that at small sizes, current leakage poses greater challenges, and also causes the chip to heat up, which creates a threat of thermal runaway and therefore further increases energy costs.

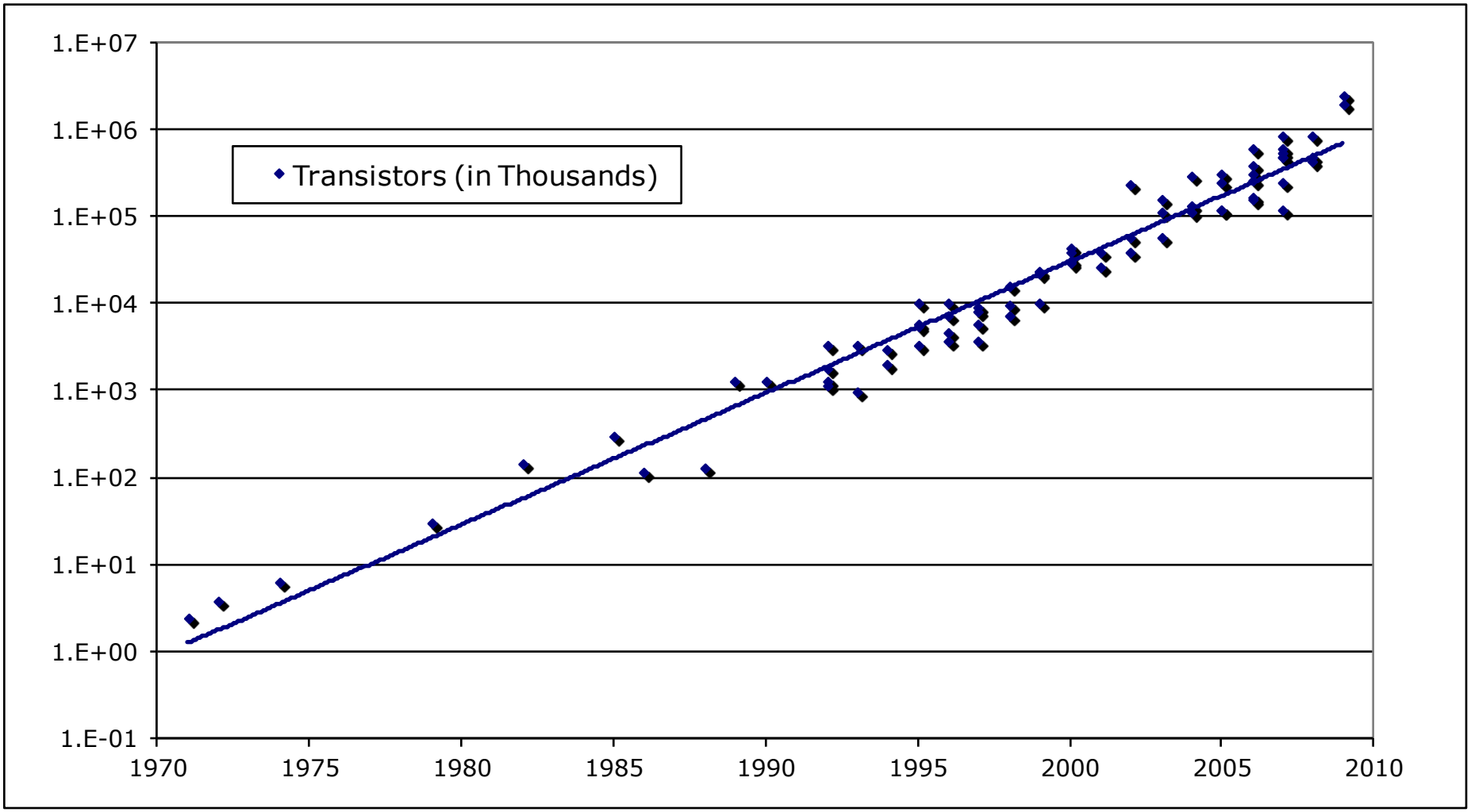


Dennard scaling is dead



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

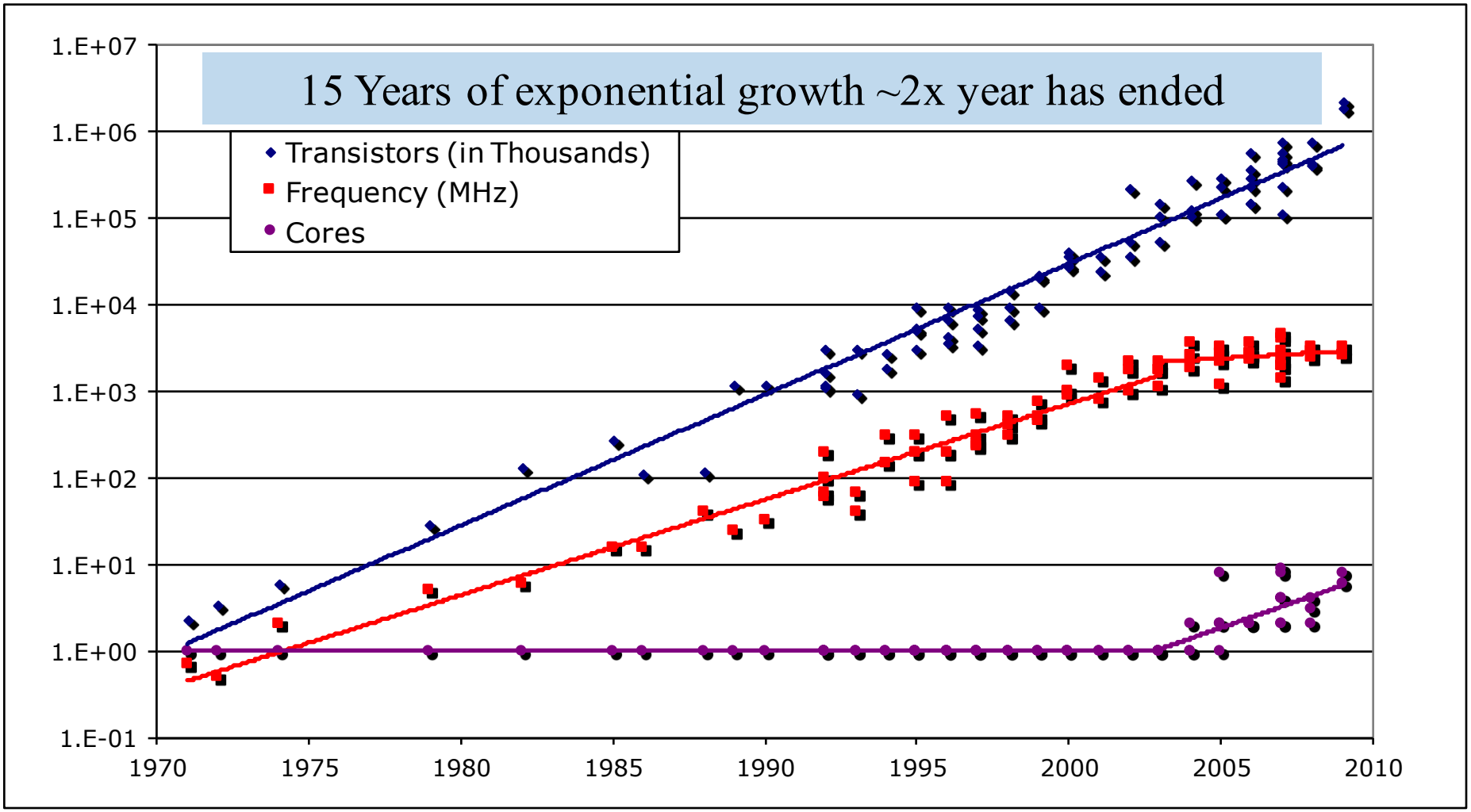
Moore's Law is Alive and Well



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

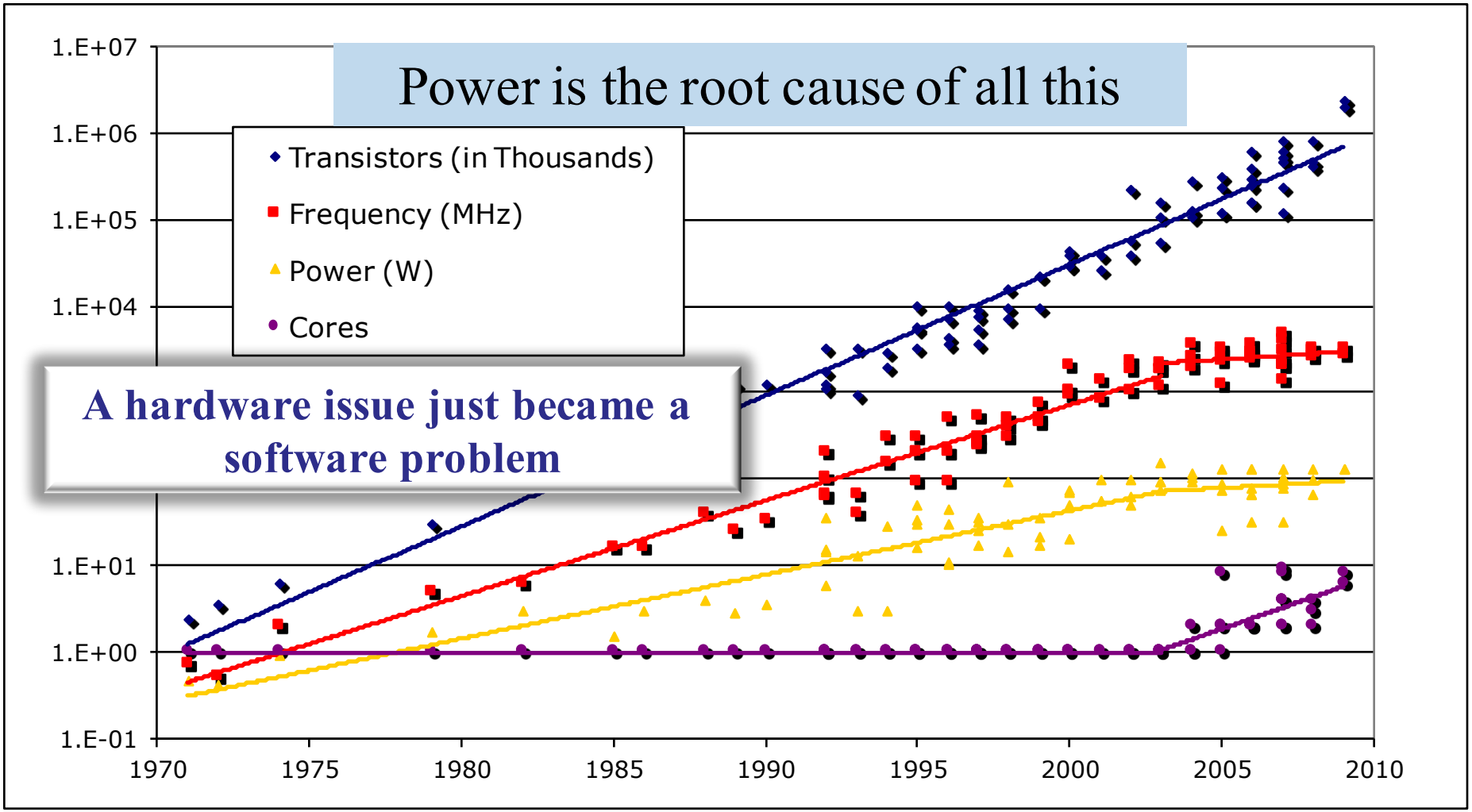
But Clock Frequency Scaling Replaced by Scaling Cores / Chip



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

Performance Has Also Slowed, Along with Power



Data from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç

Slide from Kathy Yelick

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- Frequency ∝ Voltage
- Power ∝ Frequency^3

	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X

Power Cost of Frequency

- Power ∝ Voltage^2 x Frequency (V^{2}F)
- Frequency ∝ Voltage
- Power ∝ Frequency^3

	Cores	V	Freq	Perf	Power	PE (Bops/watt)
Superscalar	1	1	1	1	1	1
"New" Superscalar	1X	1.5X	1.5X	1.5X	3.3X	0.45X
Multicore	2X	0.75X	0.75X	1.5X	0.8X	1.88X

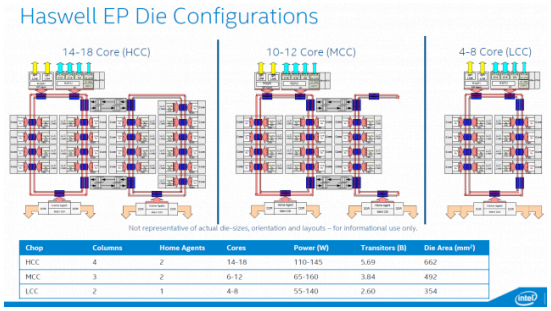
(Bigger # is better)

50% more performance with 20% less power

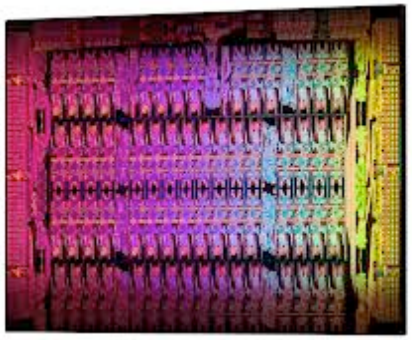
Preferable to use multiple slower devices, than one superfast device

Today's Multicores

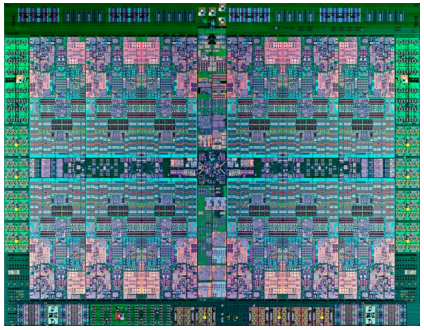
All of Top500 Systems Are Based on Multicore



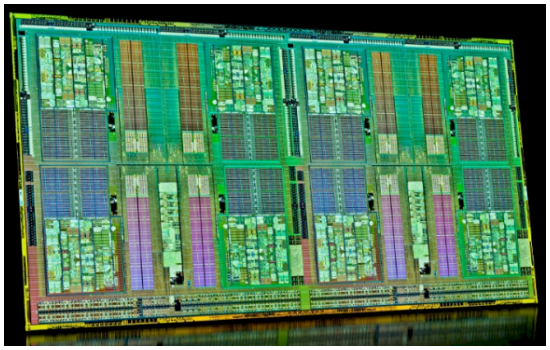
Intel Haswell (18 cores)



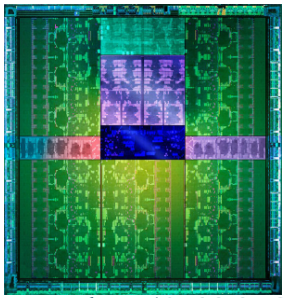
Intel Xeon Phi
(72 cores)



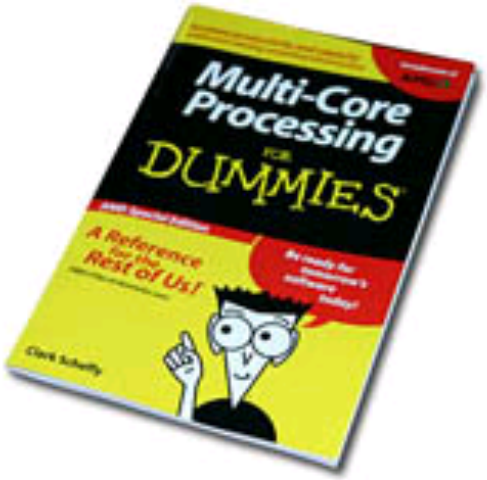
IBM Power 8 (12 cores)

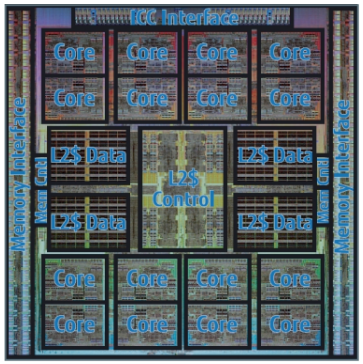


AMD Interlagos (16 cores)



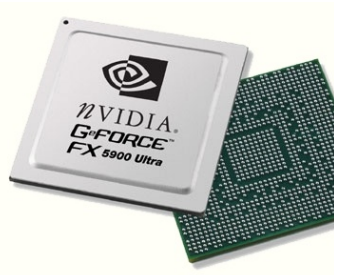
Nvidia Kepler (2688 Cuda cores
14 "regular cores")



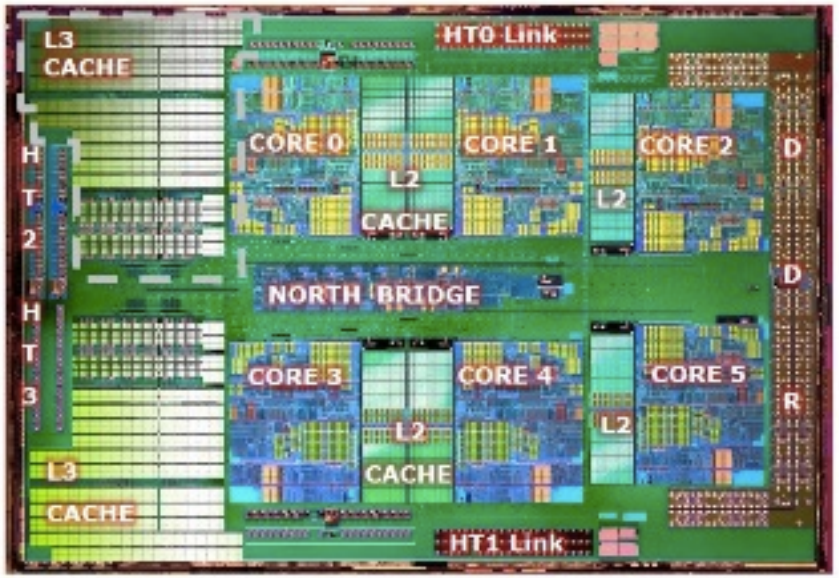


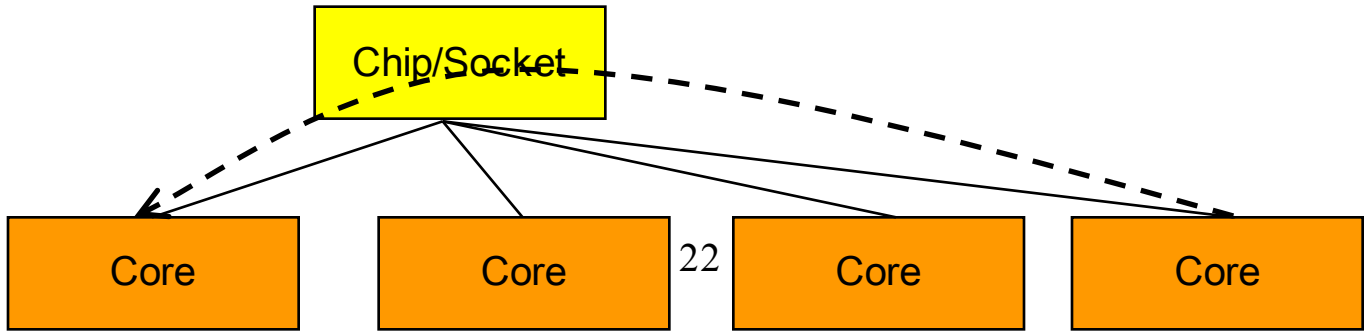
Fujitsu Venus (16 cores)

ShenWei (260 core)

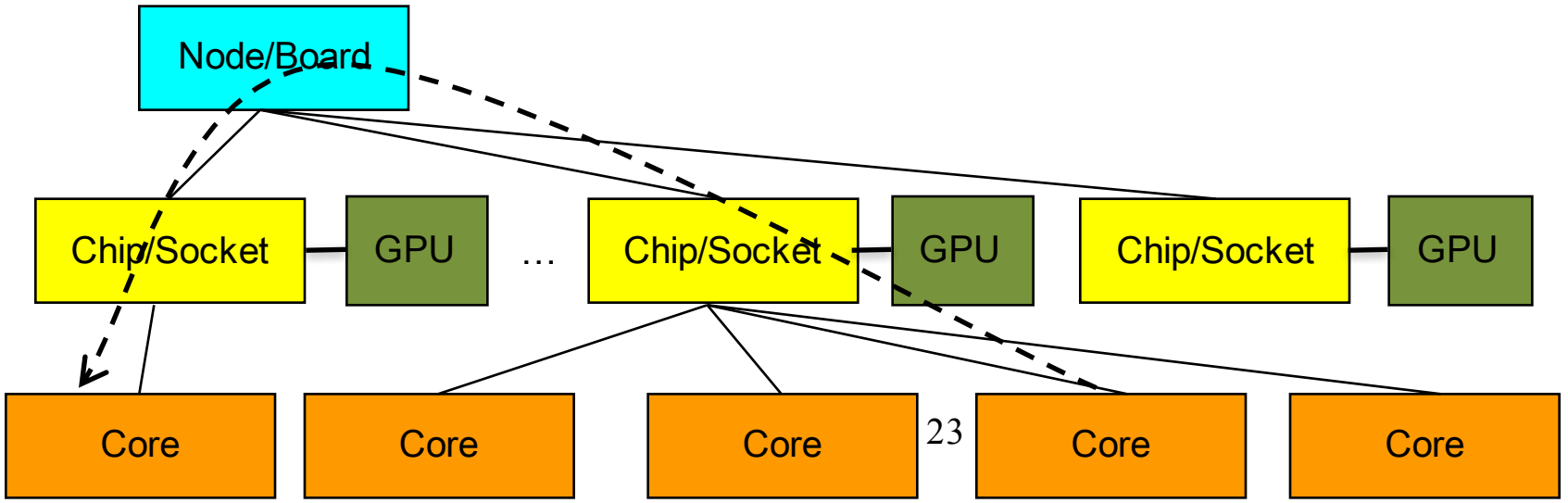


Example of typical parallel machine



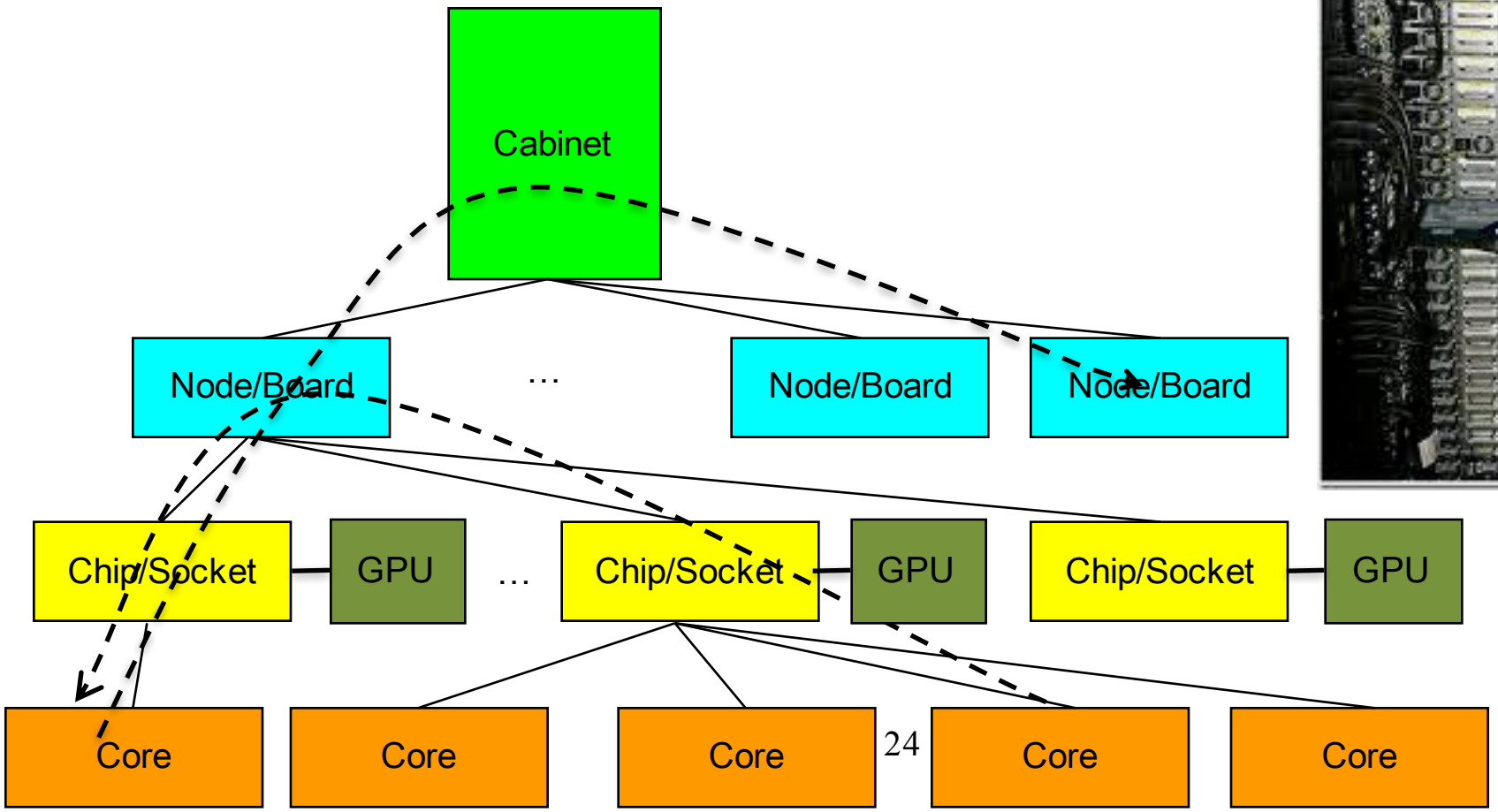


Example of typical parallel machine



Example of typical parallel machine

Shared memory programming between processes on a board and a combination of shared memory and distributed memory programming between nodes and cabinets



Example of typical parallel machine

Combination of shared memory and distributed memory programming

What do you mean by performance?

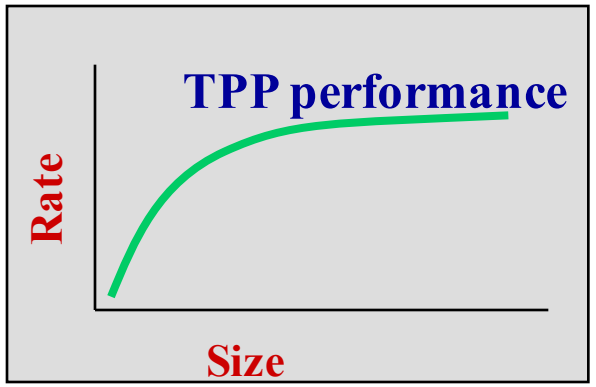
- **What is a xflop/s?**
  - xflop/s is a rate of execution, some number of floating point operations per second.
    - » Whenever this term is used it will refer to 64 bit floating point operations and the operations will be either addition or multiplication.

- **What is the theoretical peak performance?**
  - The theoretical peak is based not on an actual performance from a benchmark run, but on a paper computation to determine the theoretical peak rate of execution of floating point operations for the machine.
  - The theoretical peak performance is determined by counting the number of floating-point additions and multiplications (in full precision) that can be completed during a period of time, usually the cycle time of the machine.
  - For example, an Intel Xeon 5570 quad core at 2.93 GHz can complete 4 floating point operations per cycle or a theoretical peak performance of 11.72 GFlop/s per core or 46.88 Gflop/s for the socket.

H. Meuer, H. Simon, E. Strohmaier, & JD

- Listing of the 500 most powerful Computers in the World
- Yardstick: Rmax from LINPACK MPP

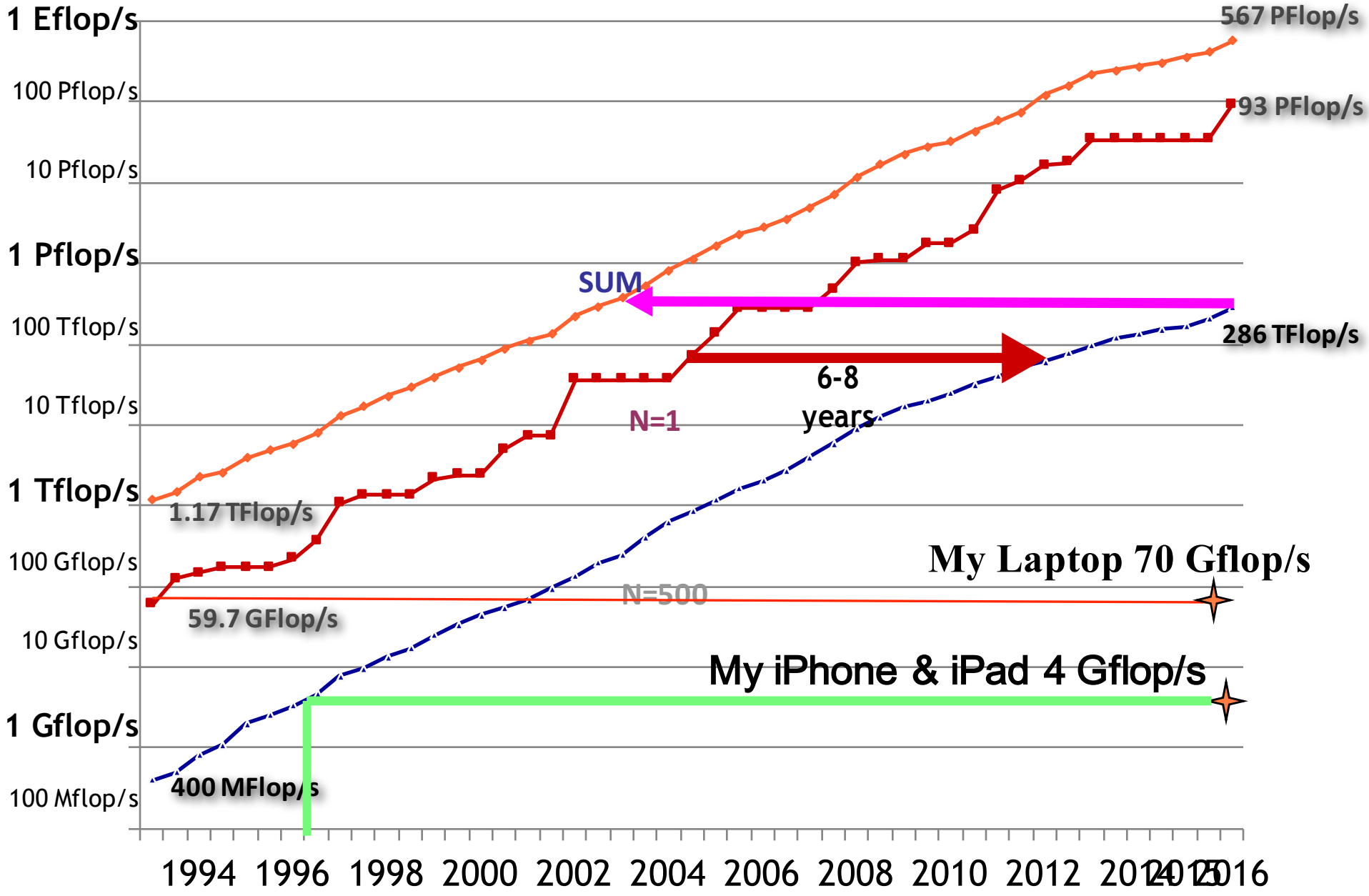
$Ax=b$, *dense problem*



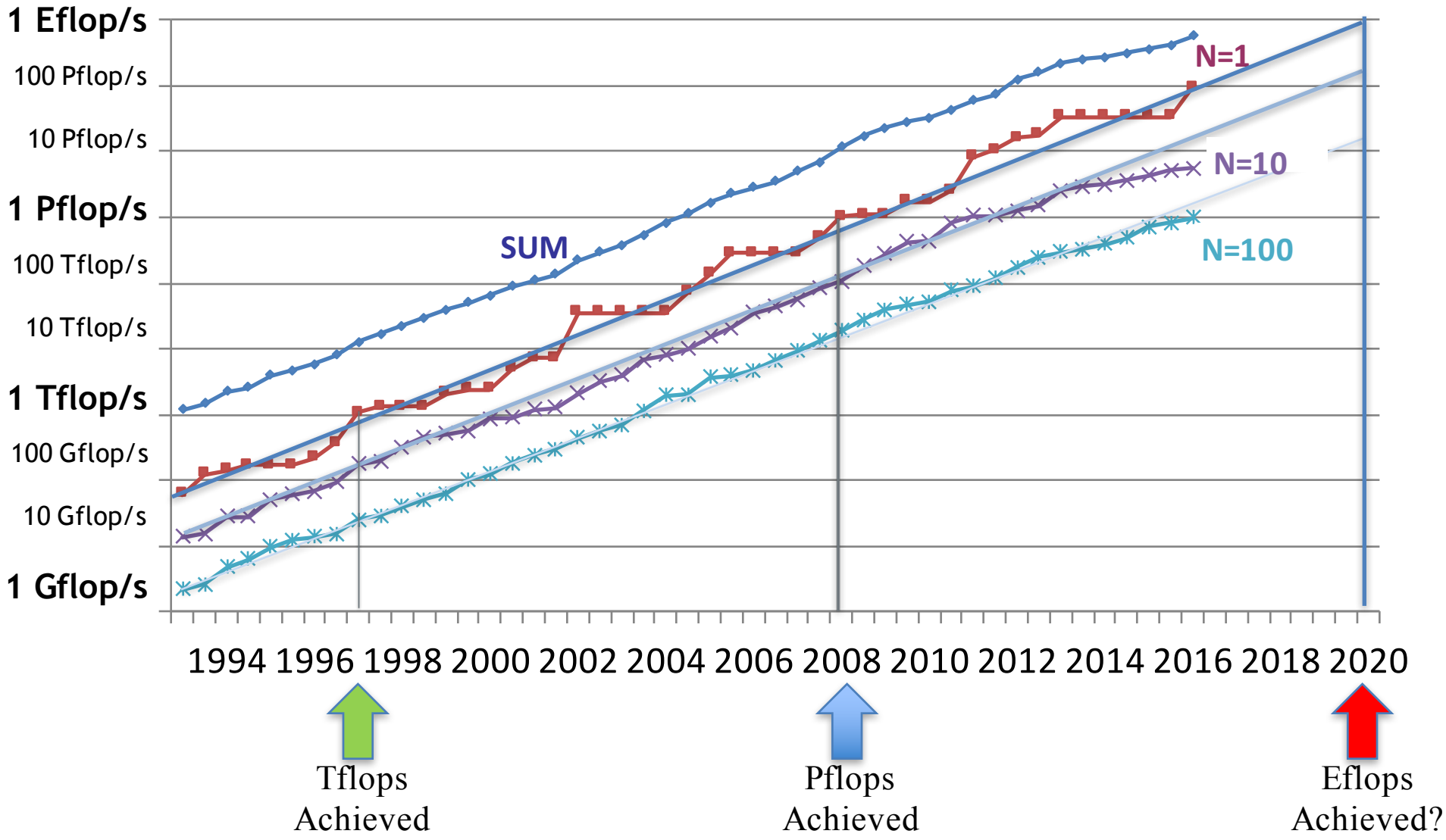
- Updated twice a year
  SC‘xy in the States in November
  Meeting in Germany in June

- All data available from **www.top500.org**

Performance Development of HPC over the Last 24 Years from the Top500



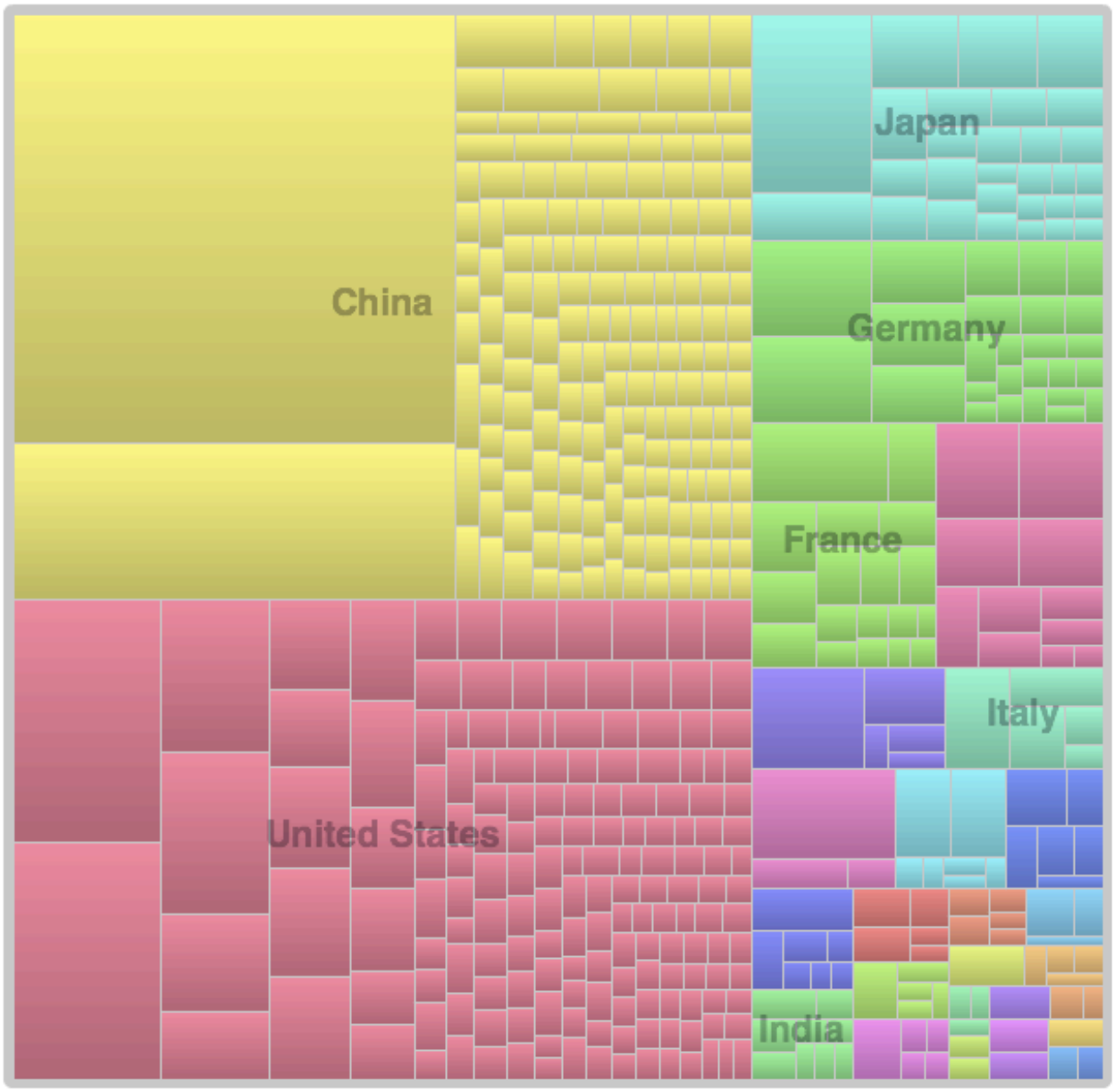
PERFORMANCE DEVELOPMENT



State of Supercomputing Today

- Pflops (> 10^{15} Flop/s) computing fully established with 95 systems.
- Three technology architecture possibilities or "swim lanes" are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs) (93 systems)
  - Lightweight cores (e.g. ShenWei, ARM, Intel's Knights Landing)
- Interest in supercomputing is now worldwide, and growing in many new markets (around 50% of Top500 computers are used in industry).
- Exascale (10^{18} Flop/s) projects exist in many countries and regions.
- Intel processors have largest share, 91% followed by AMD, 3%.

Countries Share



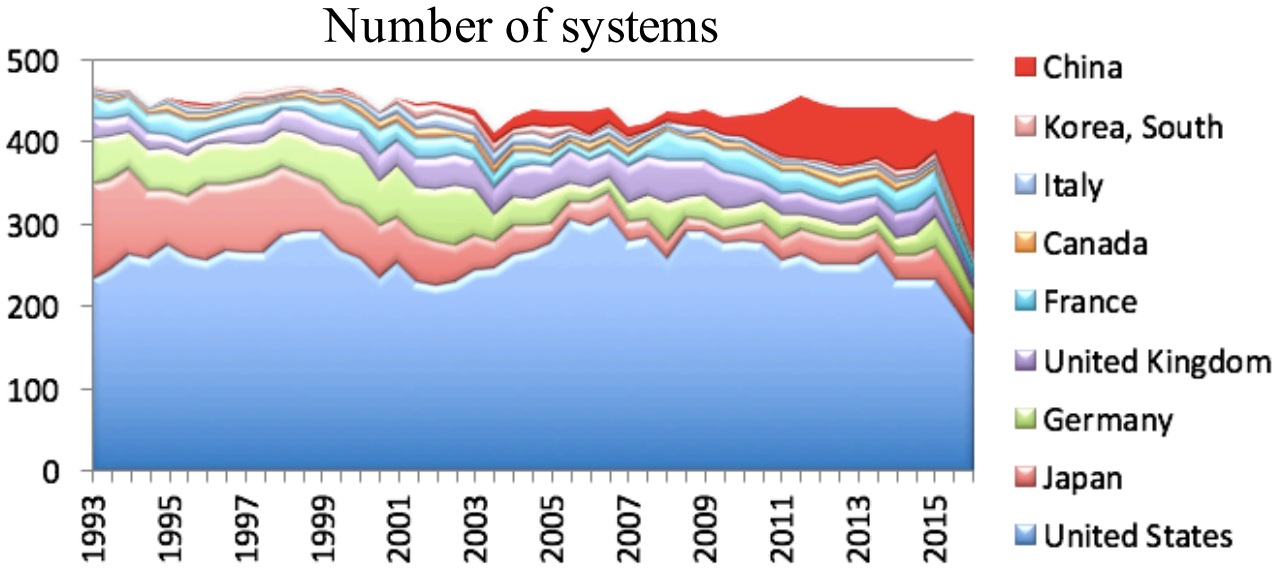
COUNTRY	NUMBER OF SUPERCOMPUTERS
China	**167**
United States	165
Japan	29
Germany	26
France	18
Britain	12
India	9
Russia	7
South Korea	7
Poland	6
other	54

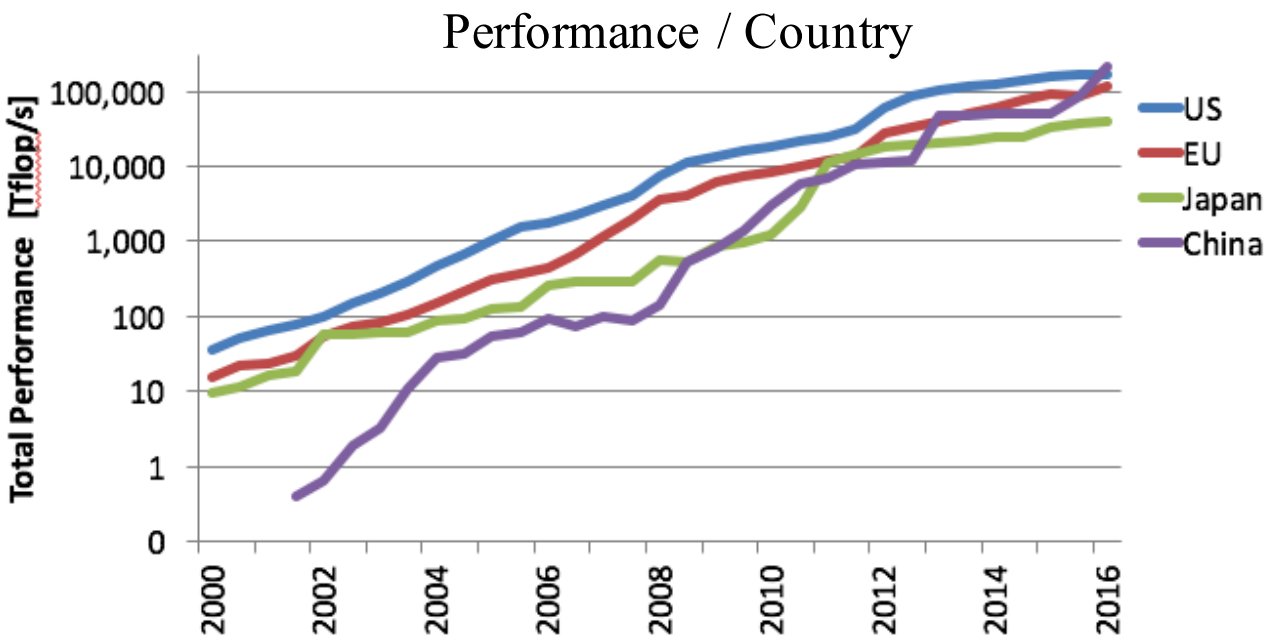
China has 1/3 of the systems, while the number of systems in the US has fallen to the lowest point since the TOP500 list was created.

June 2016: The TOP 10 Systems

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/ Watt
1	National Super Computer Center in Wuxi	Sunway TaihuLight, SW26010 (260C) + Custom	China	10,649,000	93.0	74	15.4	6.04
2	National Super Computer Center in Guangzhou	Tianhe-2 NUDT, Xeon (12C) + IntelXeon Phi (57c) + Custom	China	3,120,000	33.9	62	17.8	1.91
3	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7, AMD (16C) + Nvidia Kepler GPU (14c) + Custom	USA	560,640	17.6	65	8.21	2.14
4	DOE / NNSA L Livermore Nat Lab	Sequoia, BlueGene/Q (16C) + custom	USA	1,572,864	17.2	85	7.89	2.18
5	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx (8C) + Custom	Japan	705,024	10.5	93	12.7	.827
6	DOE / OS Argonne Nat Lab	Mira, BlueGene/Q (16C) + Custom	USA	786,432	8.16	85	3.95	2.07
7	DOE / NNSA / Los Alamos & Sandia	Trinity, Cray XC40,Xeon (16C) + Custom	USA	301,056	8.10	80	4.23	1.92
8	Swiss CSCS	Piz Daint, Cray XC30, Xeon (8C) + Nvidia Kepler (14c) + Custom	Swiss	115,984	6.27	81	2.33	2.69
9	HLRS Stuttgart	Hazel Hen, Cray XC40, Xeon (12C) + Custom	Germany	185,088	5.64	76	3.62	1.56
10	KAUST	Shaheen II, Cray XC40, Xeon (16C) + Custom	Saudi Arabia	196,608	5.54	77	2.83	1.96
500	Internet company	Inspur Intel (8C) + Nnvidia	China	5440	.286	71		

Countries Share

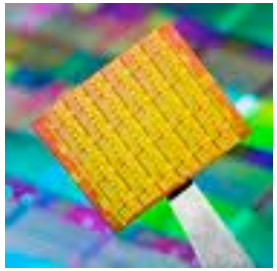




Future Computer Systems

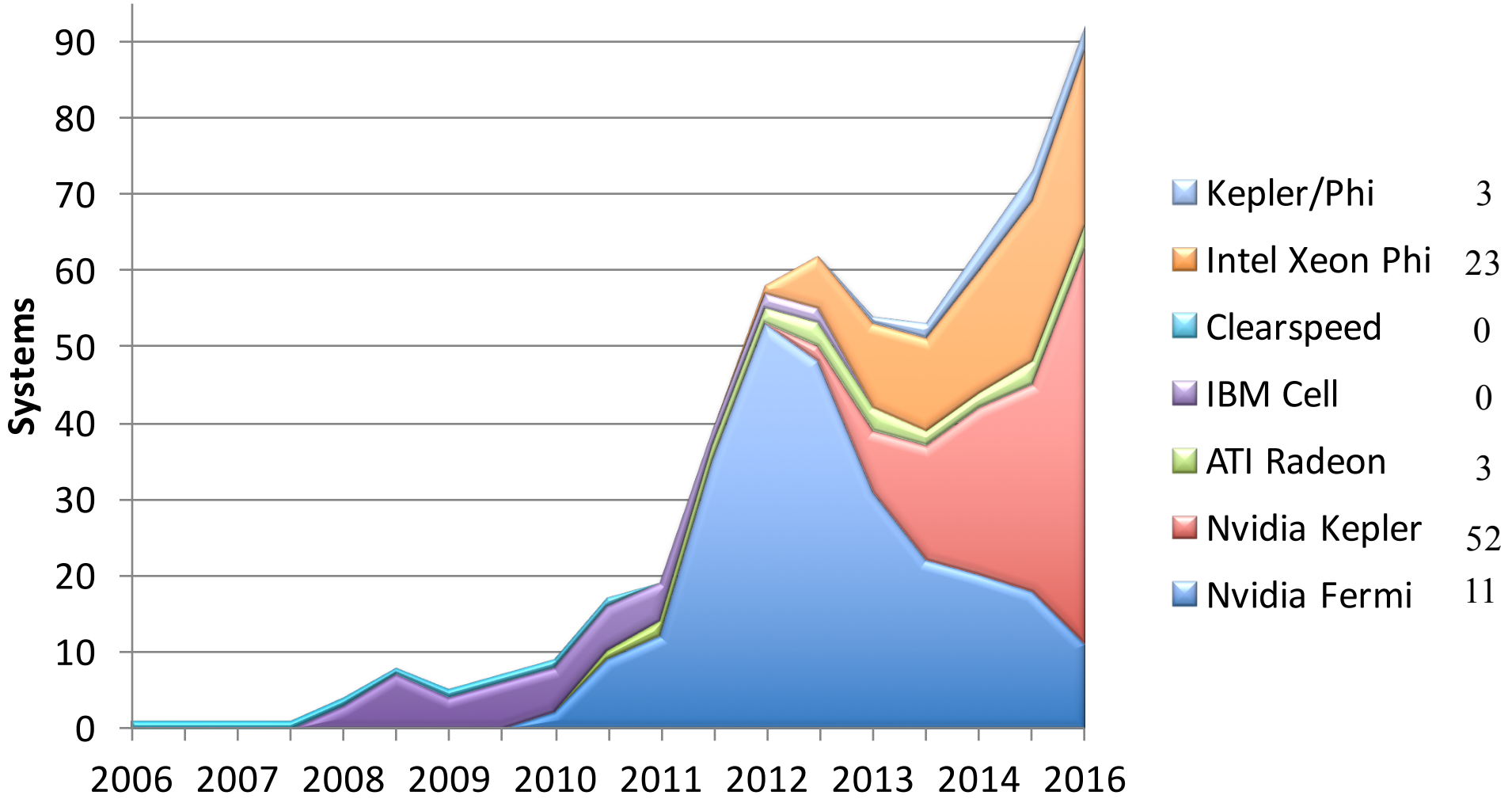
- Most likely be a hybrid design
- Think standard multicore chips and accelerator (GPUs)

- Today accelerators are attached over slow links
- Next generation more integrated
- Intel's Xeon Phi
  - 288 "threads" 72 cores

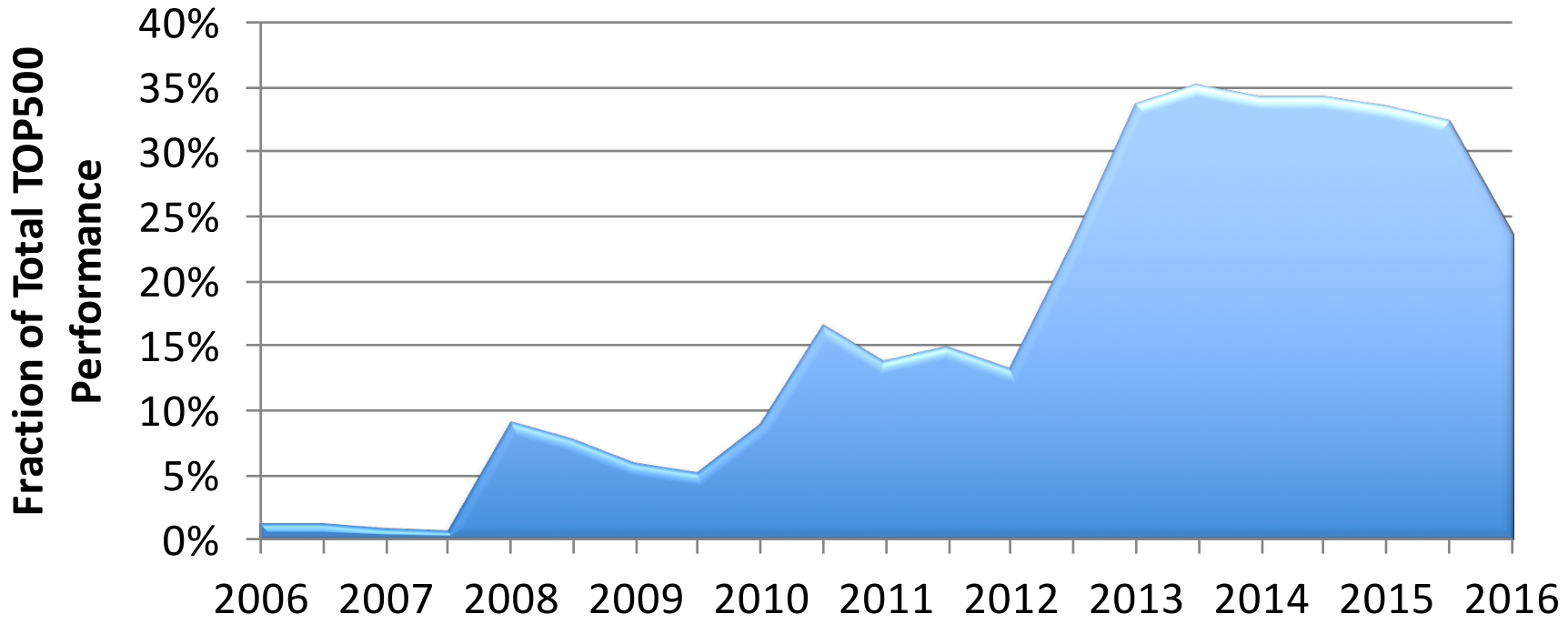


- AMD's Fusion
  - Multicore with embedded graphics ATI
- Nvidia's Kepler with 2688 "Cuda cores", 14 cores

ACCELERATORS



PERFORMANCE SHARE OF ACCELERATORS



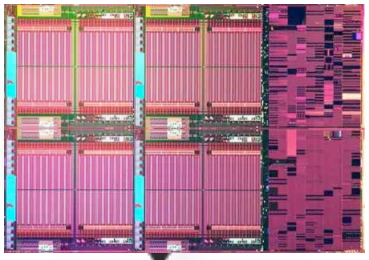
Commodity plus Accelerator Today

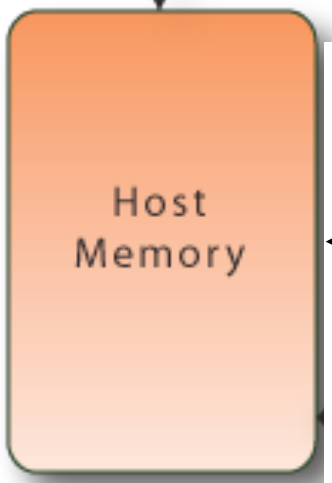
Commodity

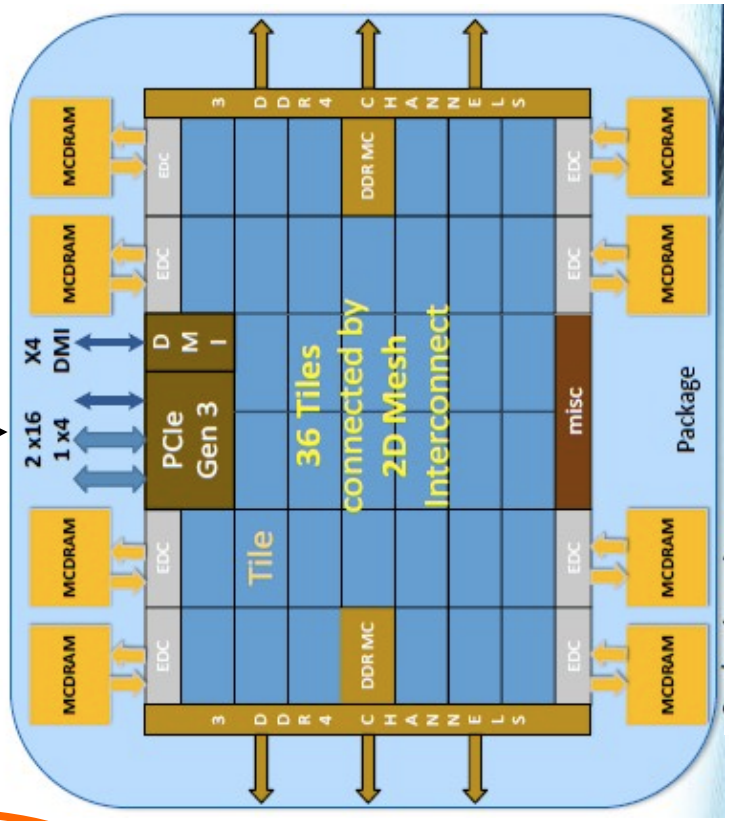
Intel Xeon
8 cores
3 GHz
8*4 ops/cycle
96 Gflop/s (DP)

Accelerator/Co-Processor

Intel Xeon Phi (KNL)
72 "cores"
32 flops/cycle/core
1.4 GHz
72*1.4*32 ops/cycle
3.22 Tflop/s (DP) or 6.45 Tflop/s (SP)



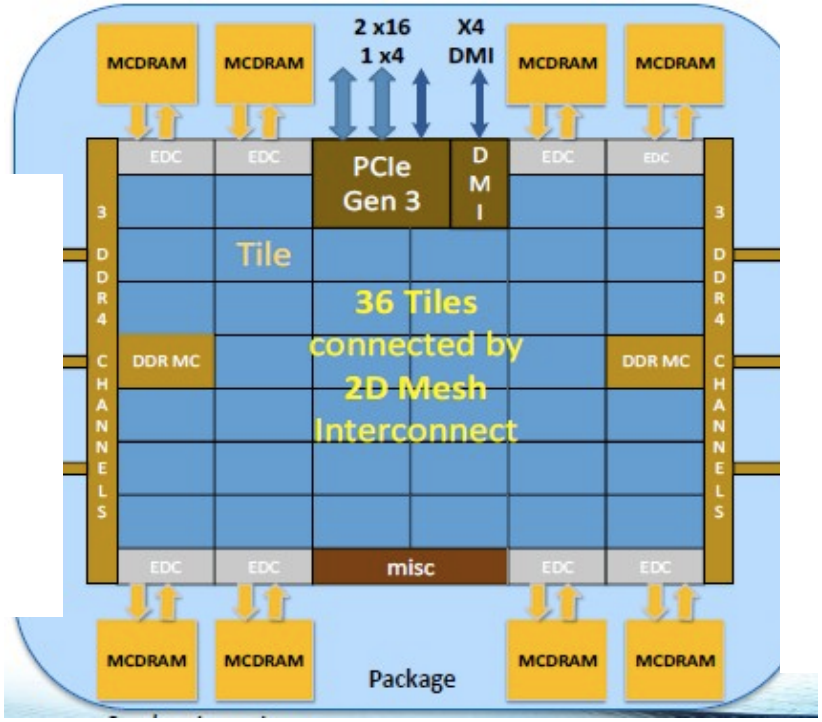




Interconnect
PCI-X 16 lane
64 Gb/s (8 GB/s)
1 GW/s

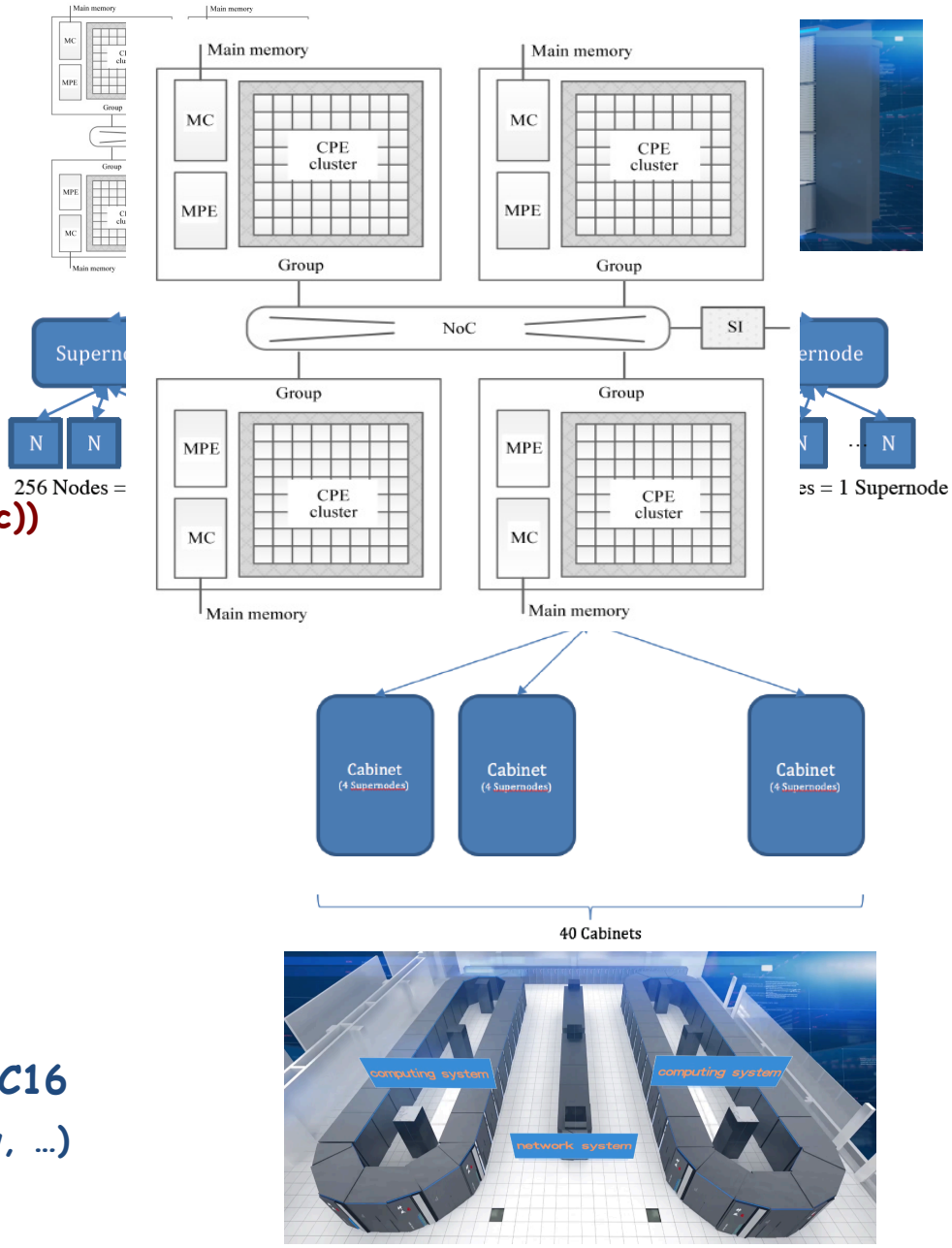
Accelerator Today

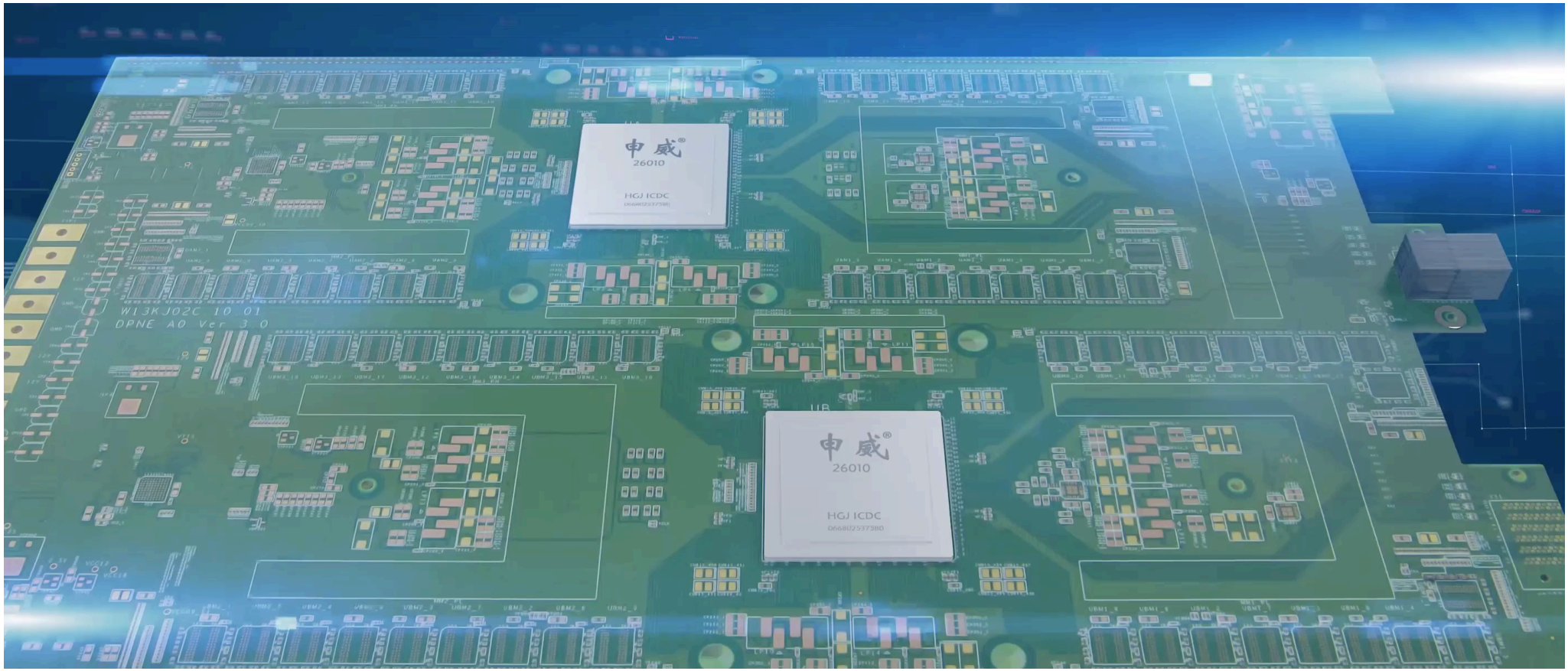
Intel Xeon Phi (KNL)
72 “cores”
32 flops/cycle/core
1.4 GHz
72*1.4*32 ops/cycle
3.22 Tflop/s (DP) or 6.45 Tflop/s (SP)

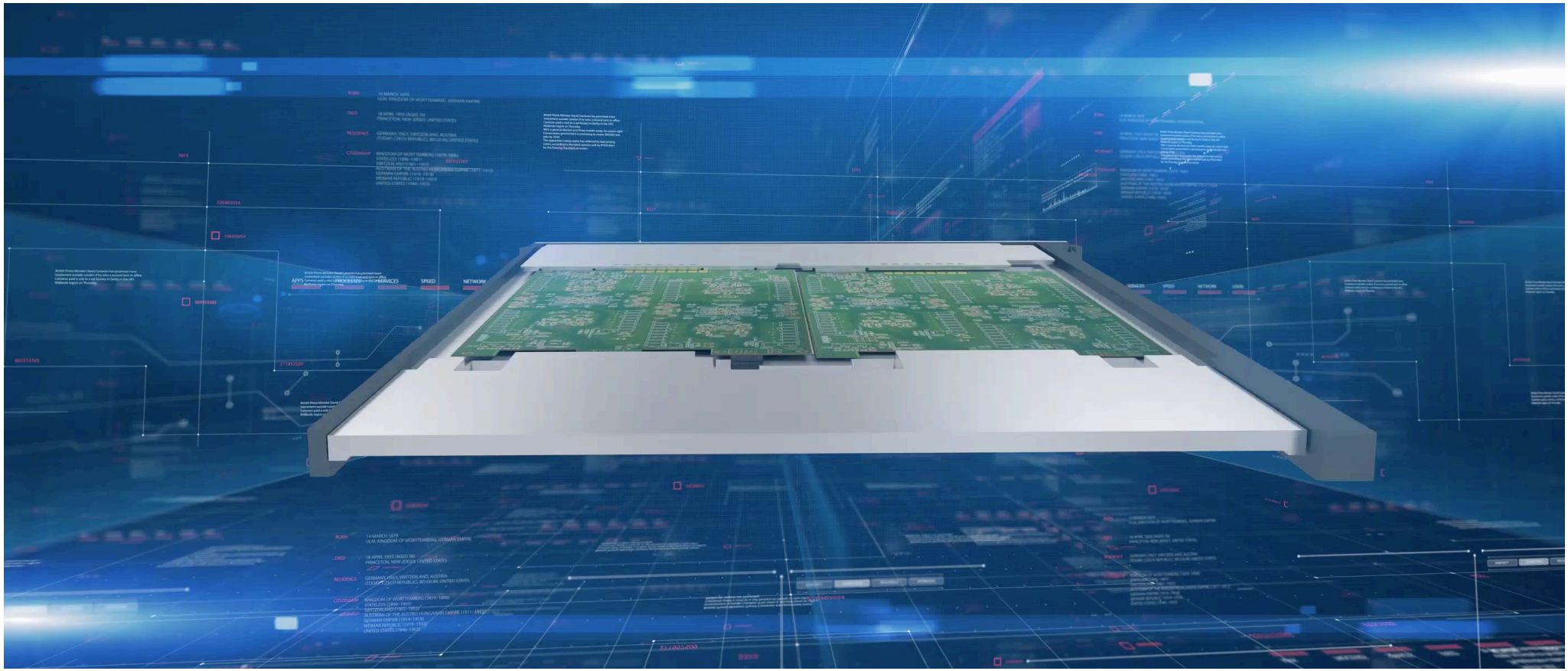


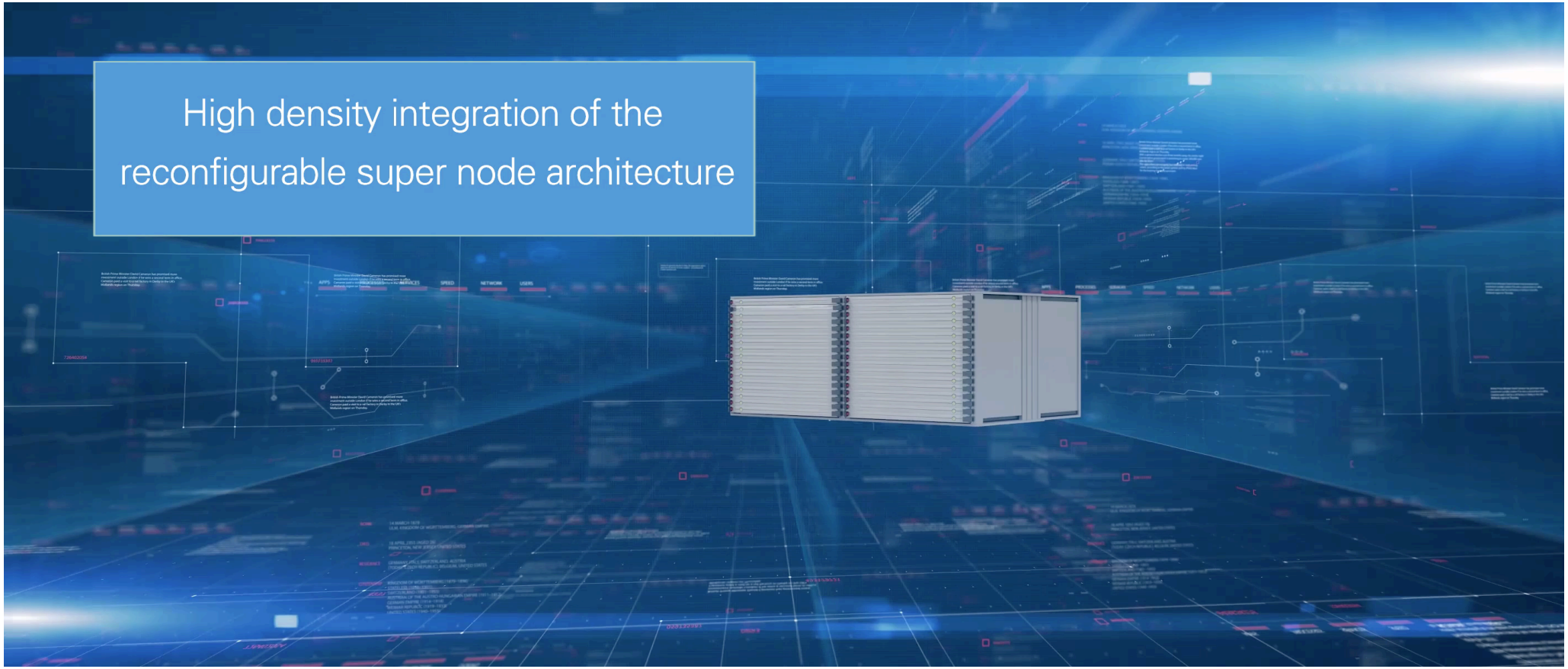
Sunway TaihuLight http://bit.ly/sunway-2016

- SW26010 processor
- Chinese design, fab, and ISA
- 1.45 GHz
- Node = 260 Cores (1 socket)
  - 4 – core groups
    - 64 CPE, No cache, 64 KB scratchpad/CG
    - 1 MPE w/32 KB L1 dcache & 256KB L2 cache
  - 32 GB memory total, 136.5 GB/s
  - ~3 Tflop/s, (22 flops/byte)
- Cabinet = 1024 nodes
  - 4 supernodes=32 boards(4 cards/b(2 node/c))
  - ~3.14 Pflop/s
- 40 Cabinets in system
  - 40,960 nodes total
  - 125 Pflop/s total peak
- 10,649,600 cores total
- 1.31 PB of primary memory (DDR3)
- 93 Pflop/s HPL, 74% peak
- 0.32 Pflop/s HPCG, 0.3% peak
- 15.3 MW, water cooled
  - 6.07 Gflop/s per Watt
- 3 of the 6 finalists Gordon Bell Award@SC16
- 1.8B RMBs ~ $280M, (building, hw, apps, sw, …)




申威®
26010
HGJ ICDC
申威®
26010
HGJ ICDC

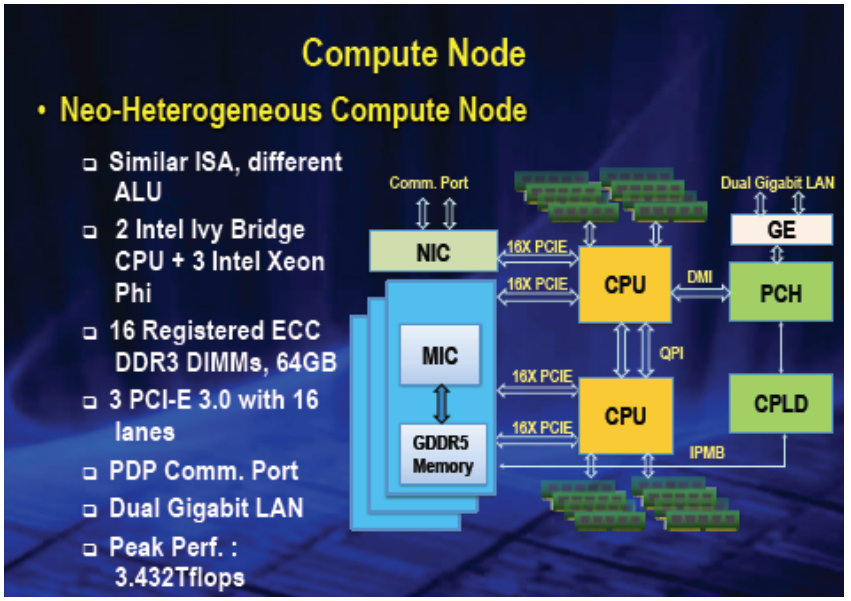

APPS
PROCESSES
SERVICES
SPEED
NETWORK
USERS

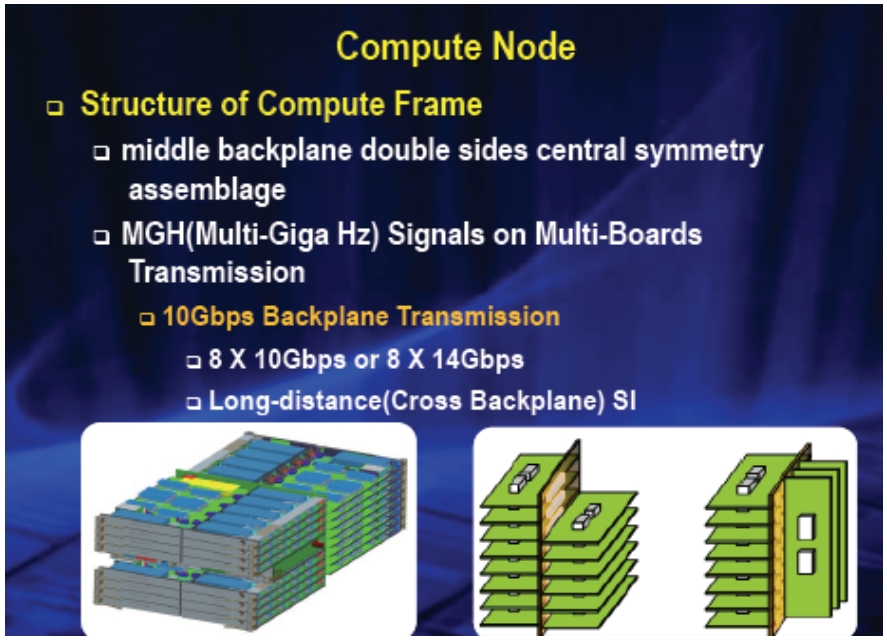

High density integration of the
reconfigurable super node architecture


SHENWEI TAIHU LIGHT

Tianhe-2 (Milkyway-2) 3+ years old

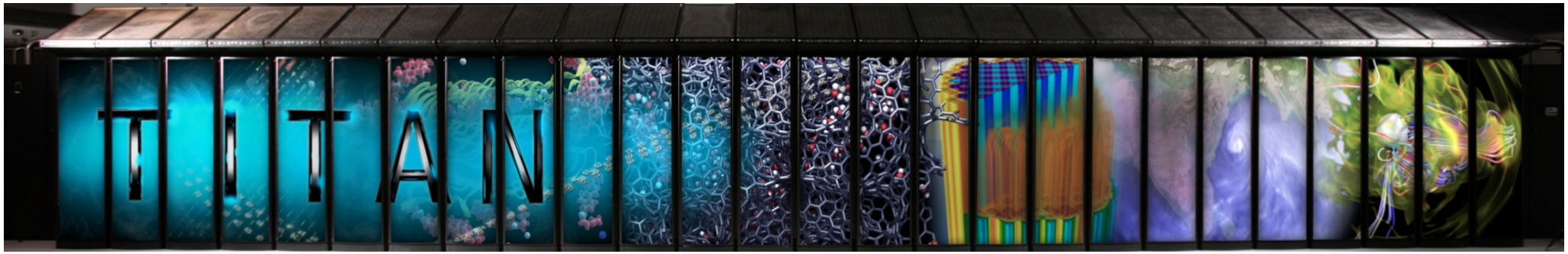
- China, 2013: *the 34 PetaFLOPS*
- Developed in cooperation between NUDT and Inspur for National Supercomputer Center in Guangzhou
- Peak performance of 54.9 PFLOPS
  - 16,000 nodes contain 32,000 Xeon Ivy Bridge processors and 48,000 Xeon Phi accelerators totaling 3,120,000 cores
  - 162 cabinets in 720m^2 footprint
  - Total 1.404 PB memory (88GB per node)
  - Each Xeon Phi board utilizes 57 cores for aggregate 1.003 TFLOPS at 1.1GHz clock
  - Proprietary TH Express-2 interconnect (fat tree with thirteen 576-port switches)
  - 12.4 PB parallel storage system
  - 17.6MW power consumption under load; 24MW including (water) cooling
  - 4096 SPARC V9 based Galaxy FT-1500 processors in front-end system

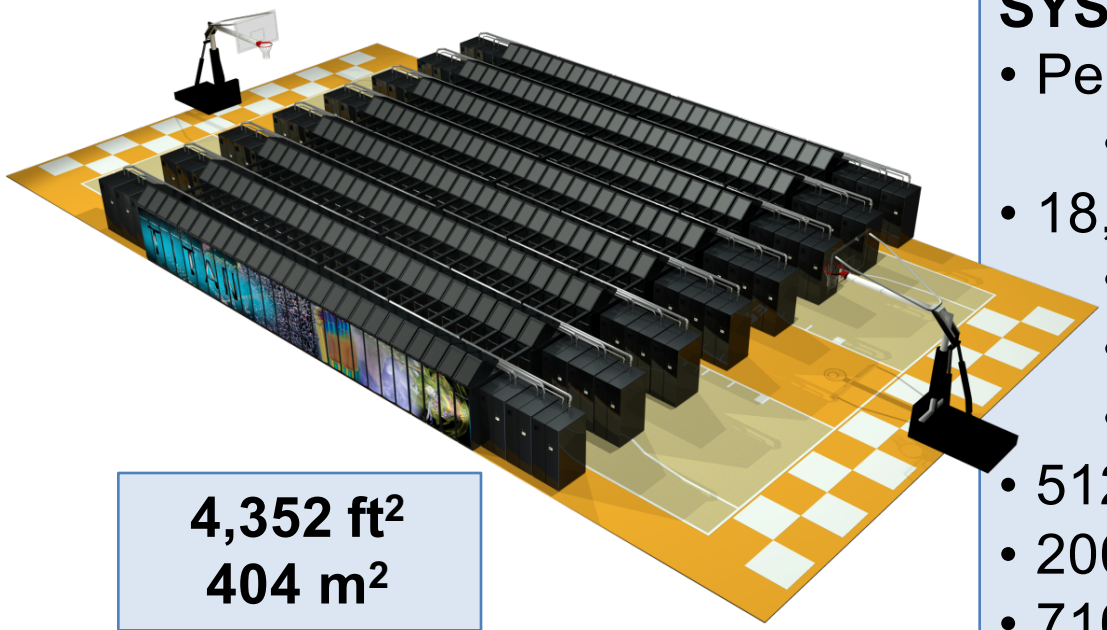




ORNL's "Titan" Hybrid System: Cray XK7 with AMD Opteron and NVIDIA Tesla processors

4 years old





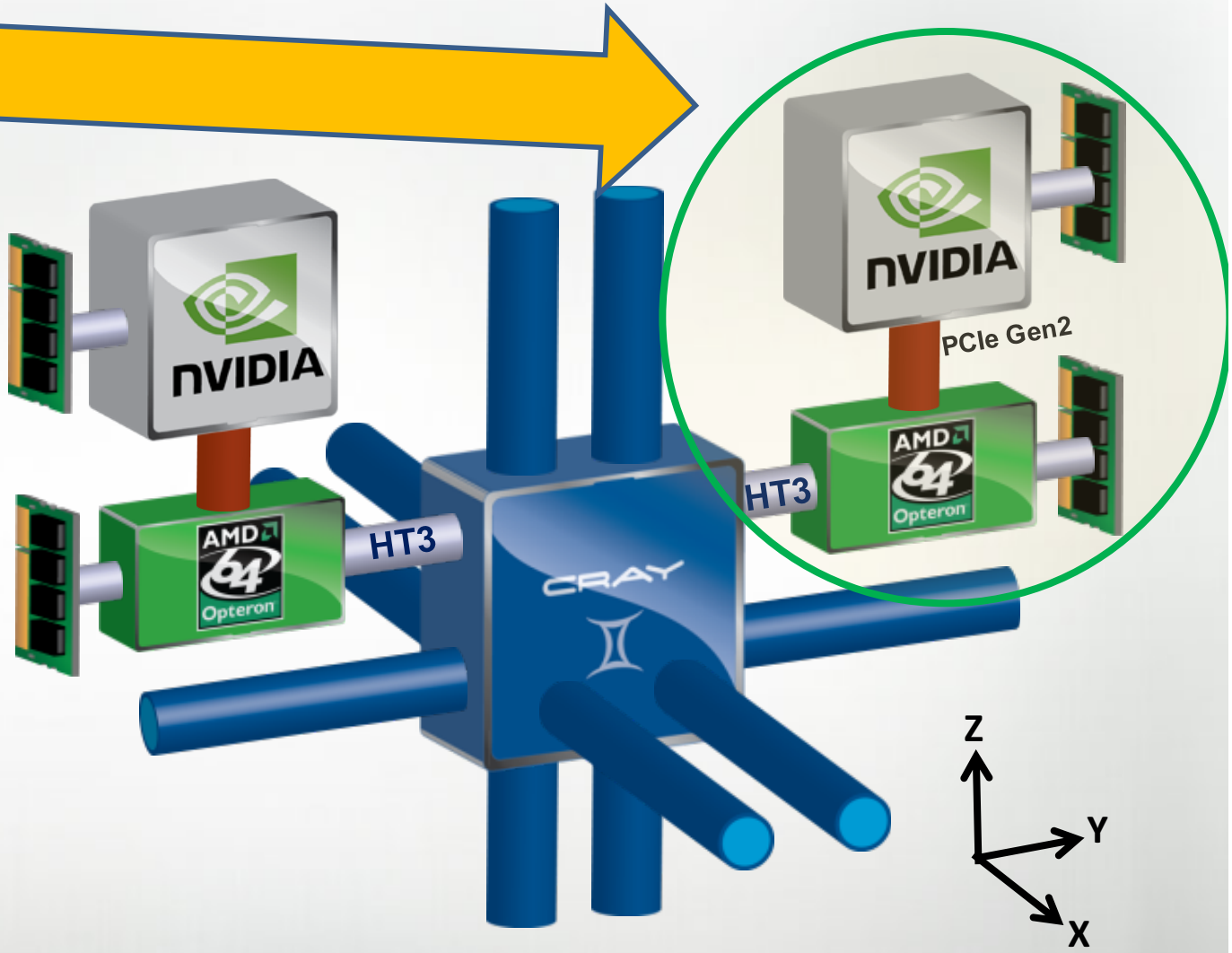
4,352 ft²
404 m²

SYSTEM SPECIFICATIONS:

- Peak performance of 27 PF
  - 24.5 Pflop/s GPU + 2.6 Pflop/s AMD
- 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
- 9 MW peak power

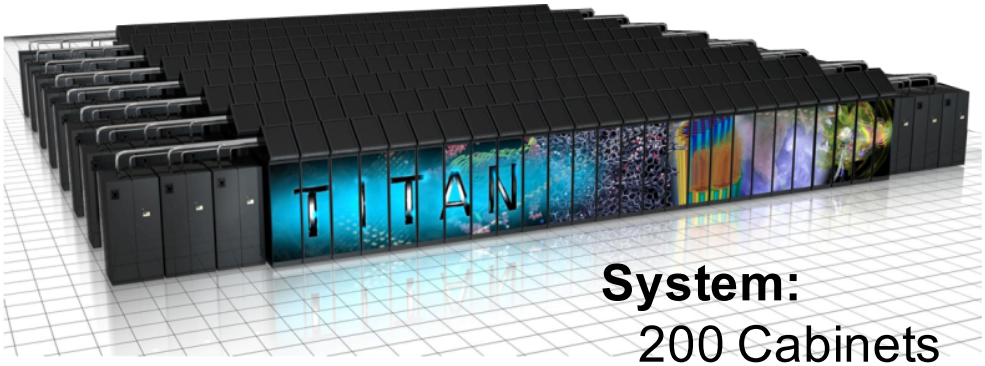
Cray XK7 Compute Node

XK7 Compute Node Characteristics
AMD Opteron 6274 Interlagos 16 core processor
Tesla K20x @ 1311 GF
Host Memory 32GB 1600 MHz DDR3
Tesla K20x Memory 6GB GDDR5
Gemini High Speed Interconnect

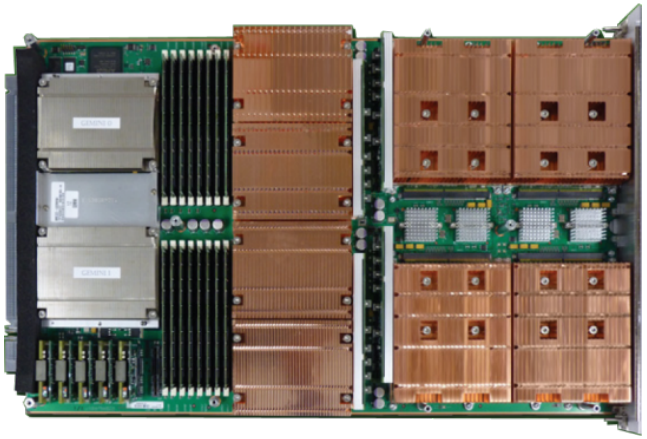


Slide courtesy of Cray, Inc.

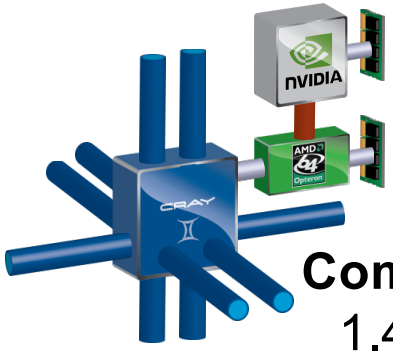
Titan:
Cray XK7 System



System:
200 Cabinets
18,688 Nodes
27 PF
710 TB



Cabinet:
24 Boards
96 Nodes
139 TF
3.6 TB



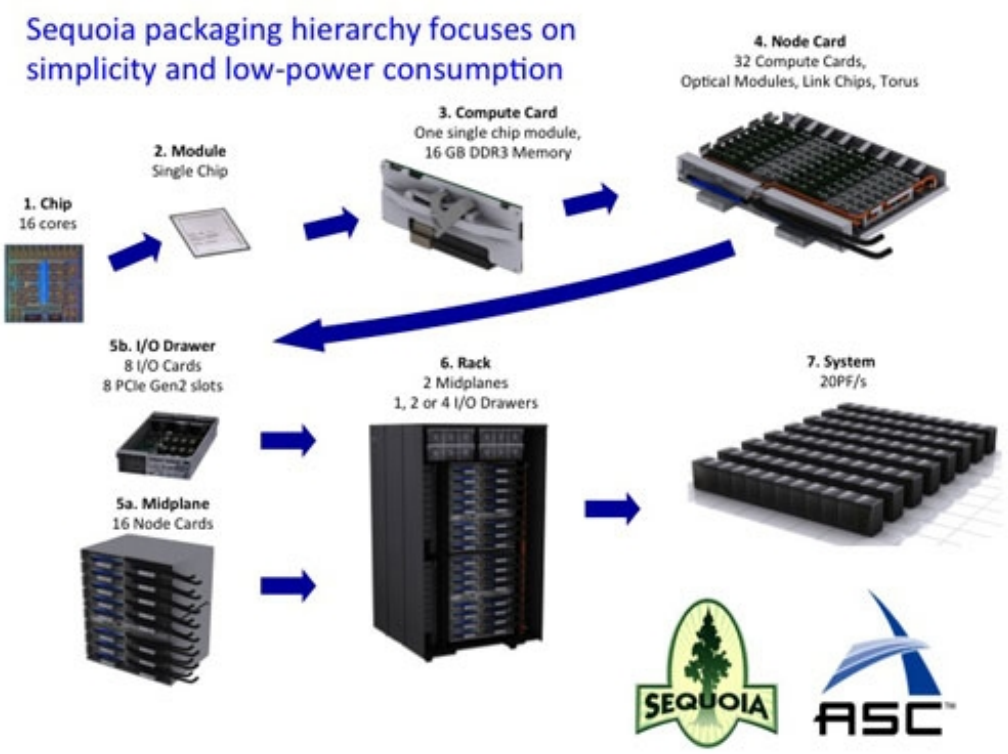
Board:
4 Compute Nodes
5.8 TF
152 GB

Compute Node:
1.45 TF
38 GB

Sequoia

4+ years old

- USA, 2012: *BlueGene strikes back*
- Built by IBM for NNSA and installed at LLNL
- 20,123.7 TFLOPS peak performance
  - Blue Gene/Q architecture
  - 1,572,864 total PowerPC A2 cores
  - 98,304 nodes in 96 racks occupy 280m²
  - 1,572,864 GB DDR3 memory
  - 5-D torus interconnect
  - 768 I/O nodes
  - 7890kW power, or 2.07 GFLOPS/W
  - Achieves 16,324.8 TFLOPS in HPL (#1 in June 2012), about 14 PFLOPS in HACC (cosmology simulation), and 12 PFLOPS in Cardioid code (electrophysiology)



Lawrence Livermore National Laboratory

IBM

NNSA National Nuclear Security Administration

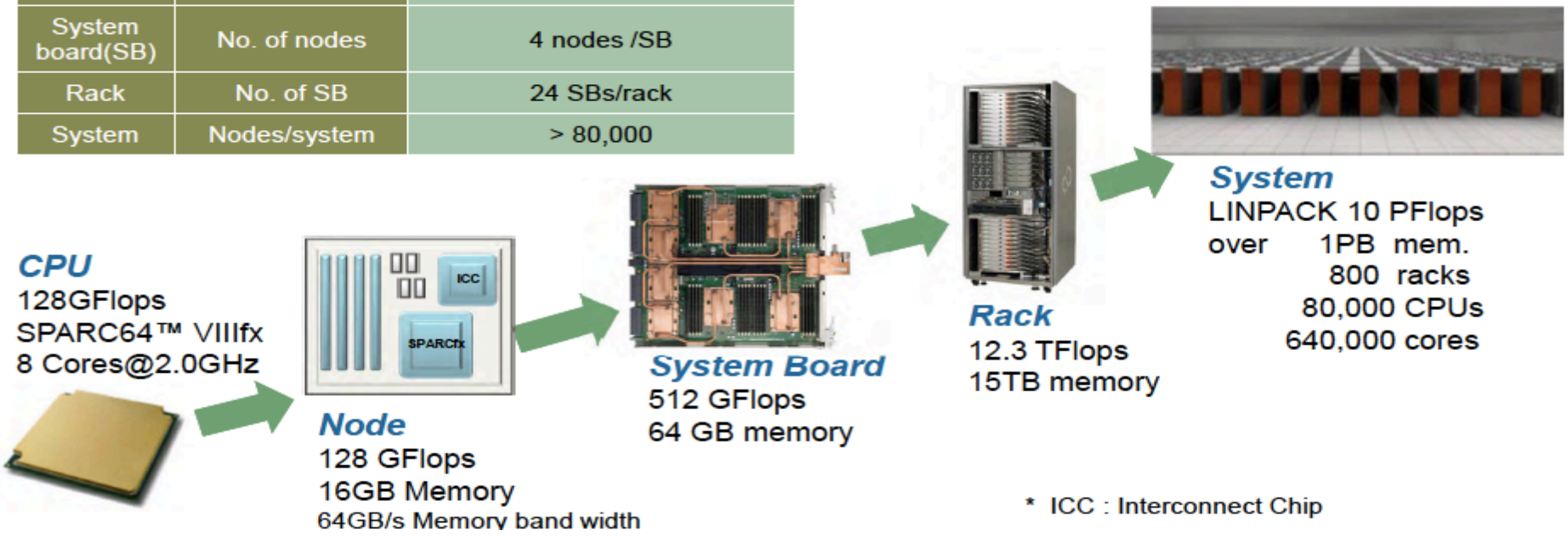
Japanese K Computer

5.5 years old

K computer Specifications

CPU (SPARC64 VIIIfx)	Cores/Node	8 cores (@2GHz)
	Performance	128GFlops
	Architecture	SPARC V9 + HPC extension
	Cache	L1(I/D) Cache : 32KB/32KB L2 Cache : 6MB
	Power	58W (typ. 30 C)
	Mem. bandwidth	64GB/s.
Node	Configuration	1 CPU / Node
	Memory capacity	16GB (2GB/core)
System board(SB)	No. of nodes	4 nodes /SB
Rack	No. of SB	24 SBs/rack
System	Nodes/system	> 80,000

Inter-connect	Topology	6D Mesh/Torus
	Performance	5GB/s. for each link
	No. of link	10 links/ node
	Additional feature	H/W barrier, reduction
	Architecture	Routing chip structure (no outside switch box)
Cooling	CPU, ICC*	Direct water cooling
	Other parts	Air cooling

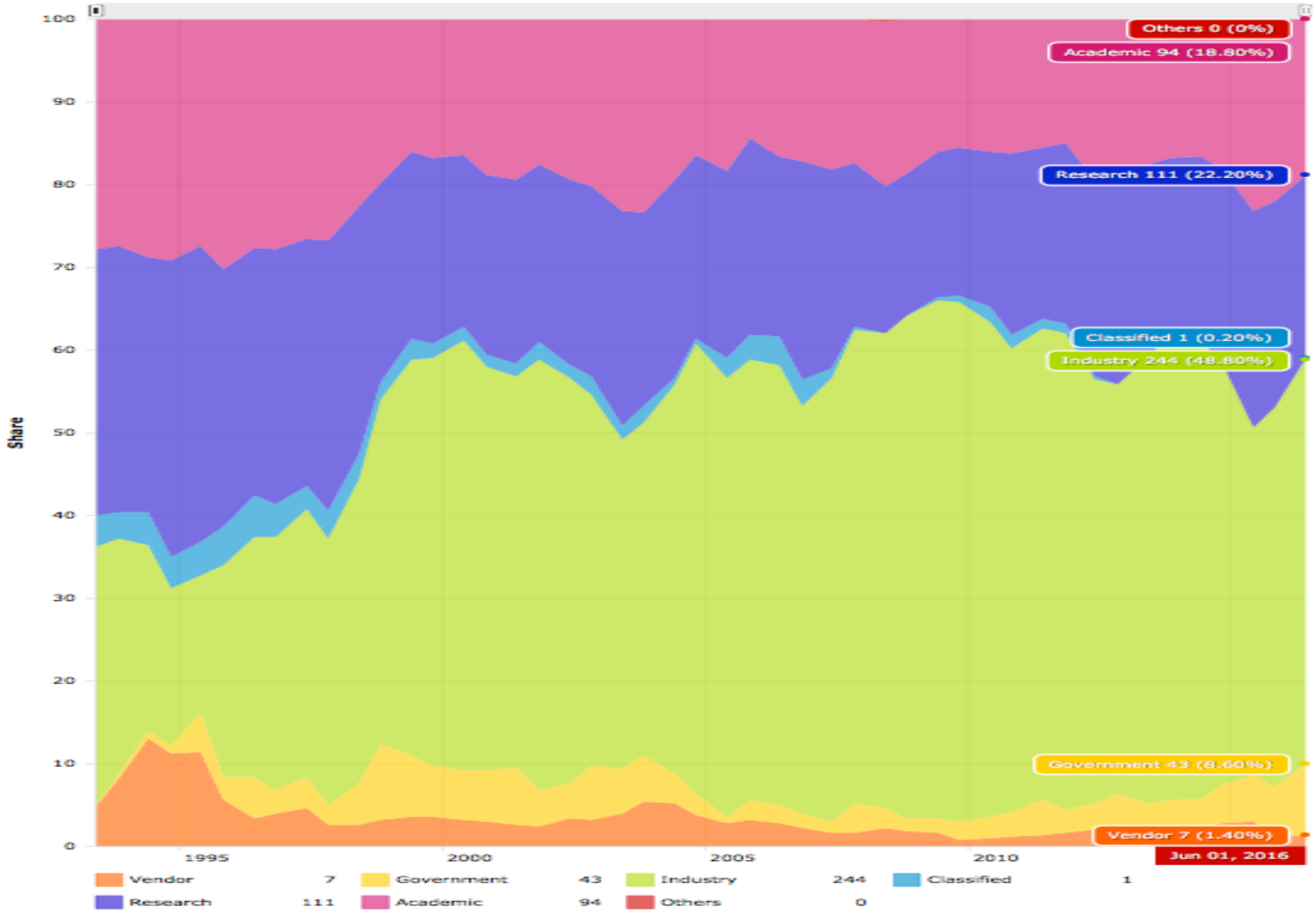


Linpack run with 705,024 cores at 10.51 Pflop/s (88,128 CPUs), 12.7 MW; 29.5 hours
Fujitsu to have a 100 Pflop/s system in 2014

12 - Top500 Systems in UK

Rank	Name	Computer	Site	# Cores	Rmax	Efficiency
17		Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries	ECMWF	126468	3944680	93%
18		Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries	ECMWF	126468	3944680	93%
29		Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries	UK Meteorological Office	89856	2801782	93%
30		Cray XC40, Xeon E5-2695v4 18C 2.1GHz, Aries	UK Meteorological Office	89856	2801782	93%
50	ARCHER	Cray XC30, Intel Xeon E5 v2 12C 2.700GHz, Aries	EPSRC/University of Edinburgh	118080	1642536	64%
56	Blue Joule	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	STFC Daresbury Lab	131072	1431102	85%
82	DiRAC	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	University of Edinburgh	98304	1073327	85%
100	Spruce A	SGI ICE X, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR	AWE	44520	958734	96%
126	Spruce B	SGI ICE X, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR	AWE	35640	767504	96%
399	Grace	Lenovo NeXtScale nx360M5, Xeon E5-2630v3 8C 2.4GHz, Infiniband QDR	University College London (UCL)	10944	341300	81%
435	Blackthorn	Bullx B510, Xeon E5-2670 8C 2.600GHz, Infiniband QDR	AWE	17856	318000	86%
500	Helen	SGI ICE X, Xeon E5-2670 8C/ E5-2680v3 12C 2.5GHz, Infiniband FDR	Imperial College London	9792	285908	77%

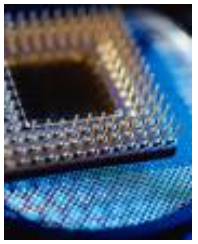
Customer Segments



Industrial Use of Supercomputers

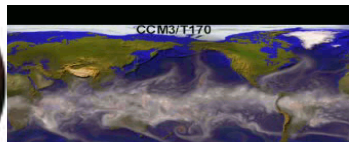
- **Of the 500 Fastest Supercomputer**
  - **Worldwide, Industrial Use is ~48%**







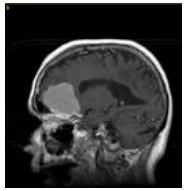




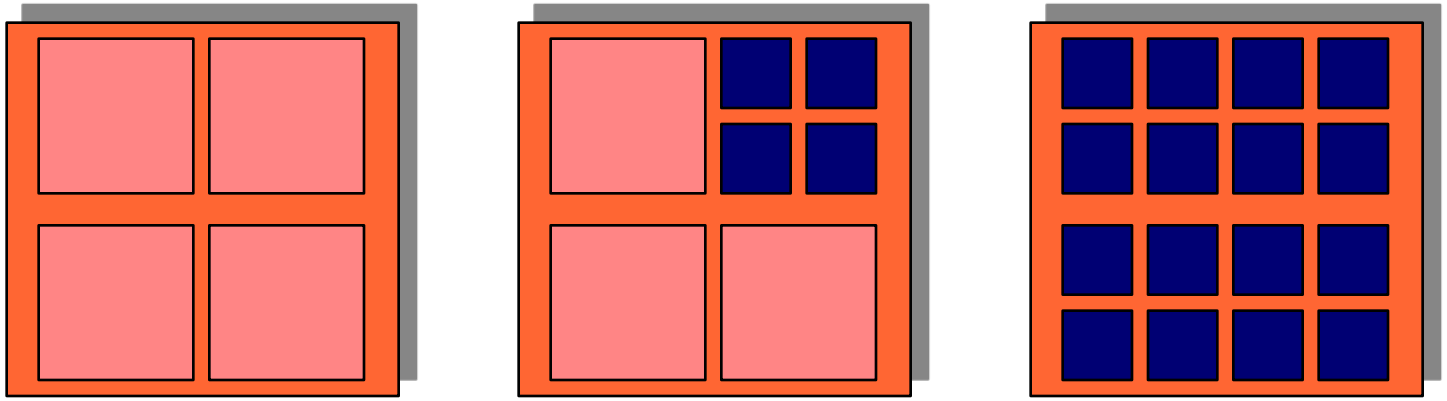

- Aerospace
- Automotive
- Biology
- CFD
- Database
- Defense
- Digital Content Creation
- Digital Media
- Electronics
- Energy
- Environment
- Finance
- Gaming
- Geophysics
- Image Proc./Rendering
- Information Processing Service
- Information Service
- Life Science
- Media
- Medicine
- Pharmaceutics
- Research
- Retail
- Semiconductor
- Telecomm
- Weather and Climate Research
- Weather Forecasting







Multi- to Many-Core



All Complex Cores
e.g. Intel Xeon

Mixed Big & Small Cores

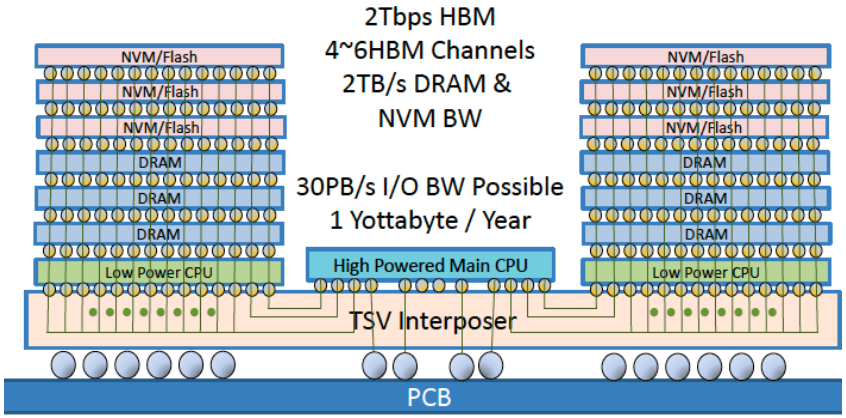
All Small Cores
e.g. Intel Xeon Phi

- **Complex cores: huge, complex, lots of internal concurrency latency hiding**
- **Simple cores: small, simpler core little internal concurrency latency-sensitive**

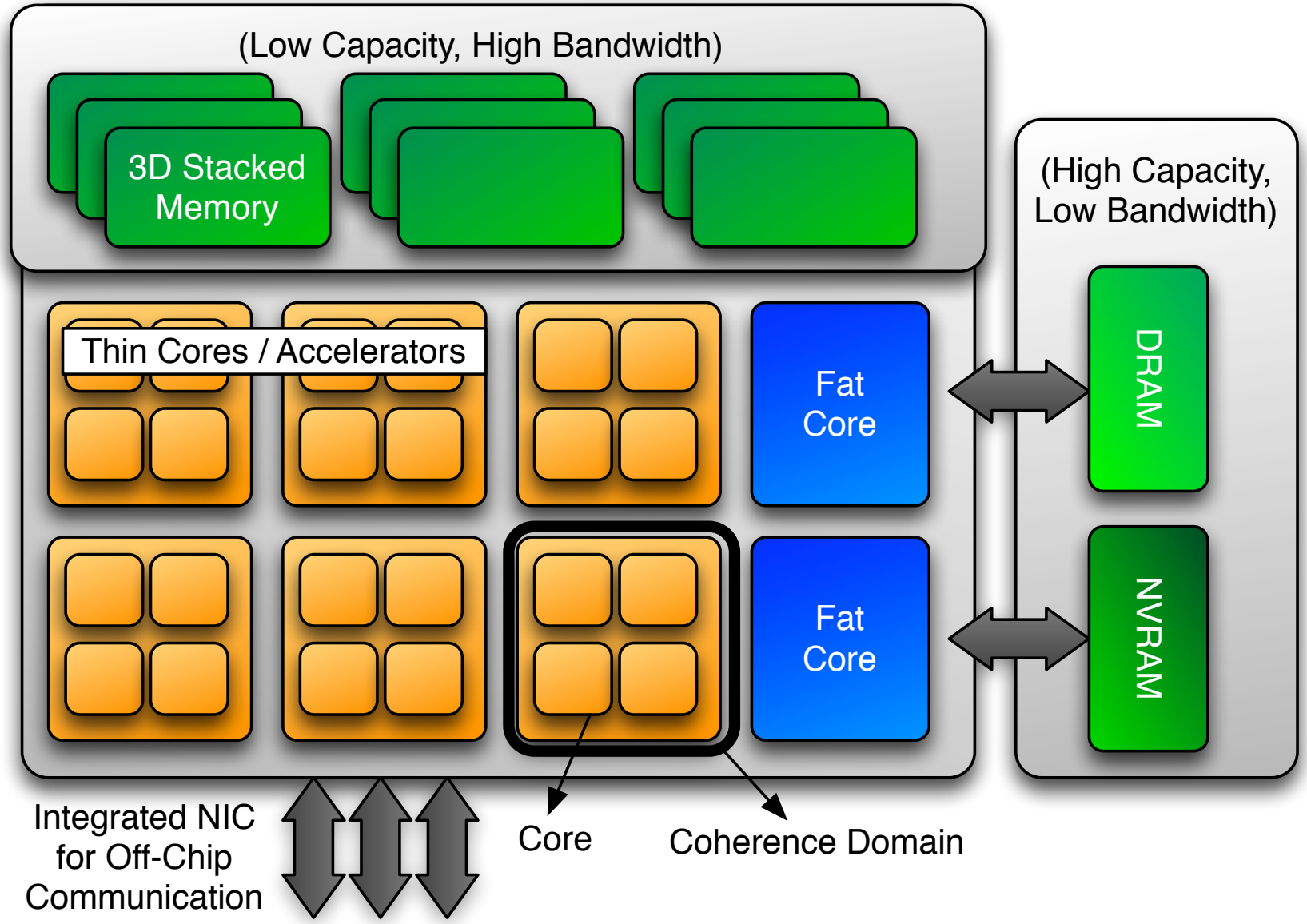
Problem with Multicore

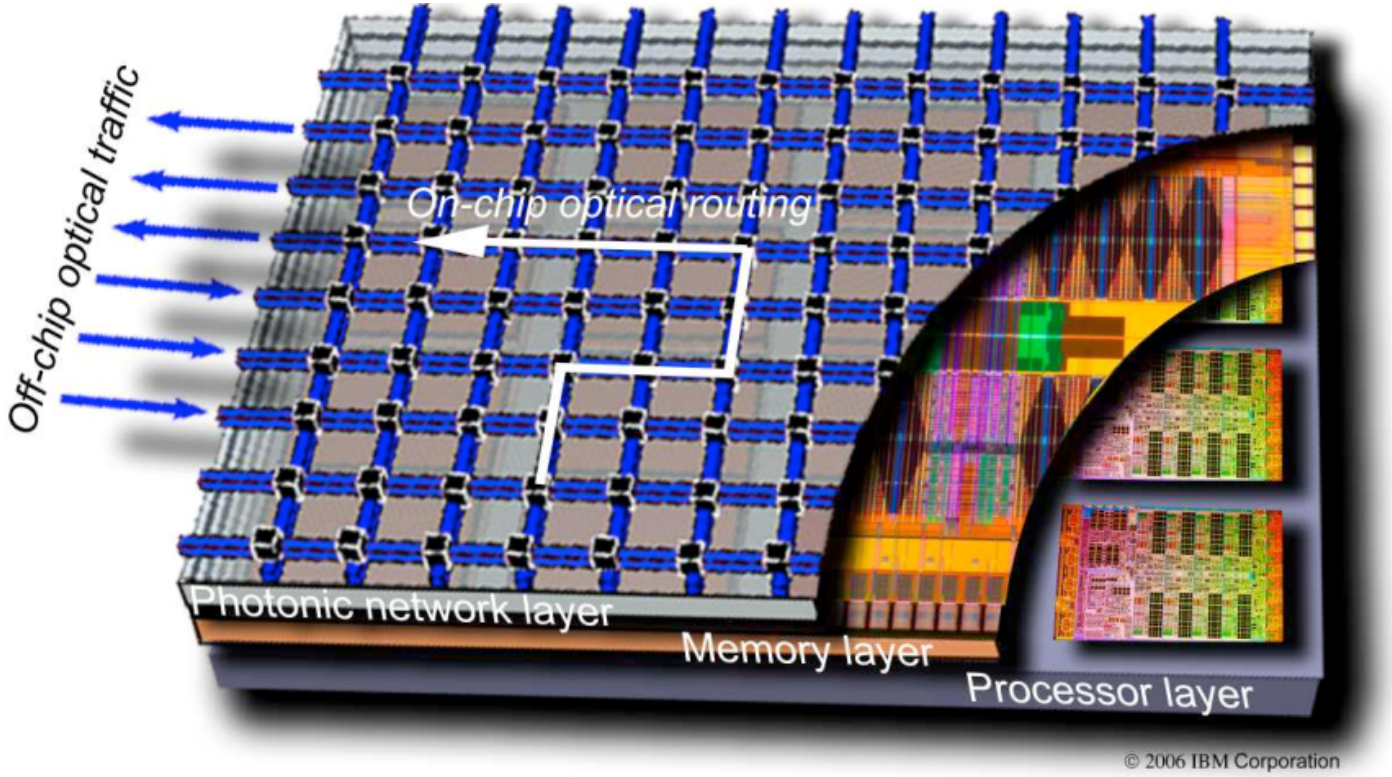
- As we put more processing power on the multicore chip, one of the problems is getting the data to the cores

- Next generation will be more integrated, 3D



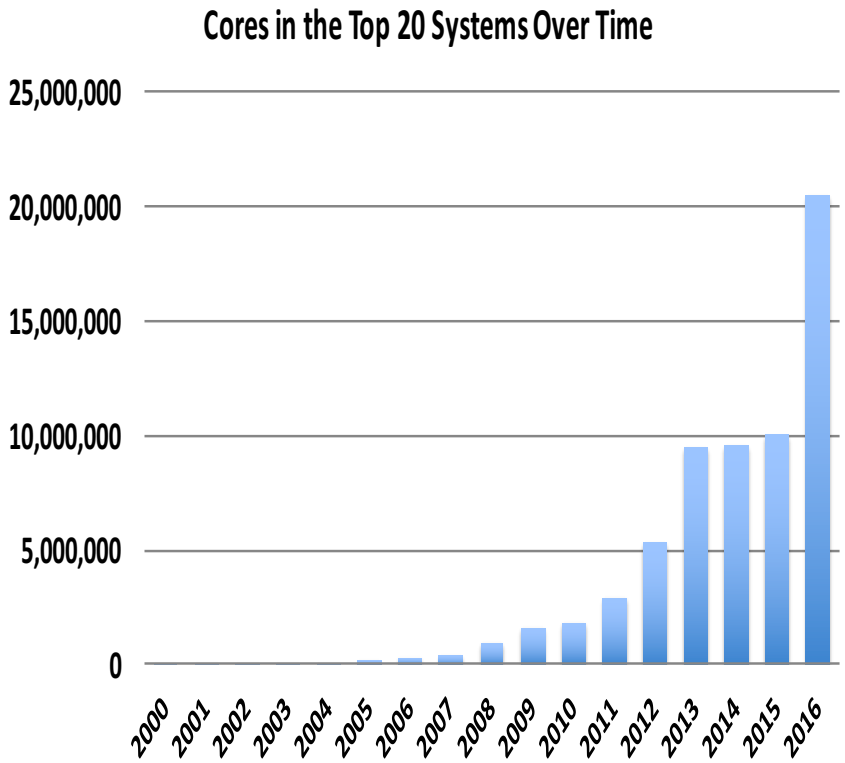
Abstract Machine Model for Exascale




Off-chip optical traffic
On-chip optical routing
Photonic network layer
Memory layer
Processor layer
© 2006 IBM Corporation

Moore's Law Reinterpreted

- Number of cores per chip doubles every 2 year, while clock speed decreases (not increases).
  - Need to deal with systems with millions of concurrent threads
    - Future generation will have billions of threads!
  - Need to be able to easily replace inter-chip parallelism with intro-chip parallelism
- Number of threads of execution doubles every 2 year



Dense Linear Algebra

- Common Operations

$$Ax = b; \quad \min_x \| Ax - b \|; \quad Ax = \lambda x$$

- A major source of large dense linear systems is problems involving the solution of boundary integral equations.
  - The price one pays for replacing three dimensions with two is that what started as a sparse problem in $O(n^3)$ variables is replaced by a dense problem in $O(n^2)$.
- Dense systems of linear equations are found in numerous other applications, including:
  - airplane wing design;
  - radar cross-section studies;
  - flow around ships and other off-shore constructions;
  - diffusion of solid bodies in a liquid;
  - noise reduction; and
  - diffusion of light through small particles.

Existing Math Software - Dense LA

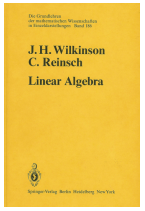
DIRECT SOLVERS	License	Support	Type		Language			Mode		
			Real	Complex	F77/ F95	C	C++	Shared	Accel.	Dist
Chameleon	CeCILL-C	See authors	X	X		X		X	C	M
DPLASMA	BSD	yes	X	X		X		X	C	M
Eigen	Mozilla	yes	X	X			X	X		
Elemental	New BSD	yes	X	X			X			M
ELPA	LGPL	yes	X	X	F90	X		X		M
FLENS	BSD	yes	X	X			X	X		
hmat-oss	GPL	yes	X	X	X	X	X	X		
LAPACK	BSD	yes	X	X	X	X		X		
LAPACK95	BSD	yes	X	X	X			X		
libflame	New BSD	yes	X	X	X	X		X		
MAGMA	BSD	yes	X	X	X	X		X	C/O/X	
NAPACK	BSD	yes	X		X			X		
PLAPACK	LGPL	yes	X	X	X	X				M
PLASMA	BSD	yes	X	X	X	X		X		
rejtrix	by-nc-sa	yes	X				X	X		
ScaLAPACK	BSD	yes	X	X	X	X				M/P
Trilinos/Pliris	BSD	yes	X	X		X	X			M
ViennaCL	MIT	yes	X				X	X	C/O/X	

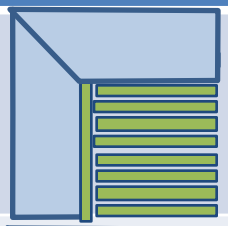
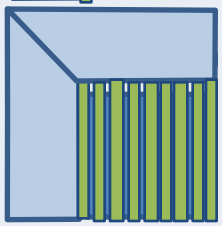
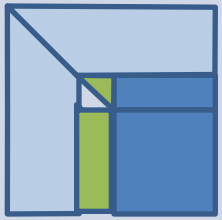
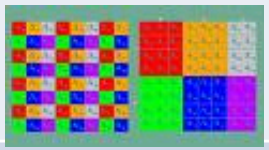
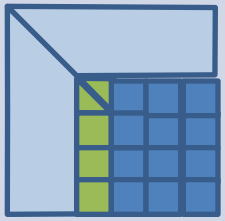
- http://www.netlib.org/utk/people/JackDongarra/la-sw.html
- LINPACK, EISPACK, LAPACK, ScaLAPACK
- PLASMA, MAGMA

DLA Solvers

- We are interested in developing Dense Linear Algebra Solvers
- Retool LAPACK and ScaLAPACK for multicore and hybrid architectures

40 Years Evolving SW and Alg Tracking Hardware Developments



Software/Algorithms follow hardware evolution in time		
EISPACK (70's) (Translation of Algol)	 	Rely on - Fortran, but row oriented
LINPACK (80's) (Vector operations)	 	Rely on - Level-1 BLAS operations - Column oriented
LAPACK (90's) (Blocking, cache friendly)	 	Rely on - Level-3 BLAS operations
ScaLAPACK (00's) (Distributed Memory)	 	Rely on - PBLAS Mess Passing
PLASMA (10's) New Algorithms (many-core friendly)		Rely on - DAG/scheduler - block data layout - some extra kernels

Peak Performance - Per Core

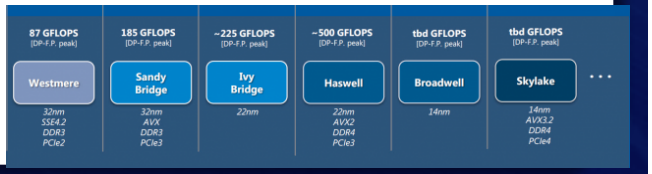
$$\text{FLOPS} = \text{cores} \times \text{clock} \times \frac{\text{FLOPs}}{\text{cycle}}$$

Floating point operations per cycle per core

- Most of the recent computers have FMA (Fused multiple add): (i.e. x ←x + y*z in one cycle)
- Intel Xeon earlier models and AMD Opteron have SSE2
  - 2 flops/cycle DP & 4 flops/cycle SP
- Intel Xeon Nehalem ('09) & Westmere ('10) have SSE4
  - 4 flops/cycle DP & 8 flops/cycle SP
- Intel Xeon Sandy Bridge('11) & Ivy Bridge ('12) have AVX & AVX2
  - 8 flops/cycle DP & 16 flops/cycle SP

We are here

- Intel Xeon Haswell ('13) & (Broadwell ('14)) AVX2
  - 16 flops/cycle DP & 32 flops/cycle SP
  - Xeon Phi (per core) is at 16 flops/cycle DP & 32 flops/cycle SP
- Intel Xeon Skylake ('15)
  - 32 flops/cycle DL & 64 flops/cycle SP



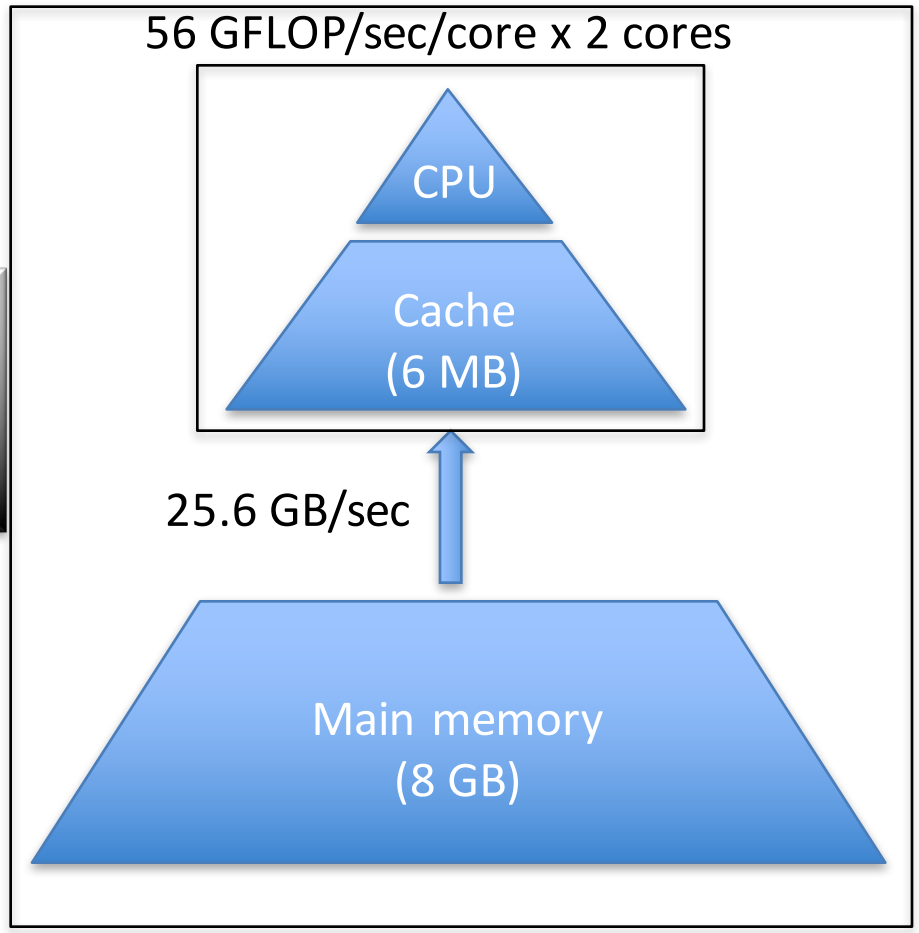
Memory transfer

(Its All About Data Movement)

Example on my laptop: One level of memory

Intel Core i7 4850HQ
Haswell, 2.3 GHz
Turbo Boost 3.5 GHz

16 flops/cycle * 3.5 GHz = 56 Gflop/s per core

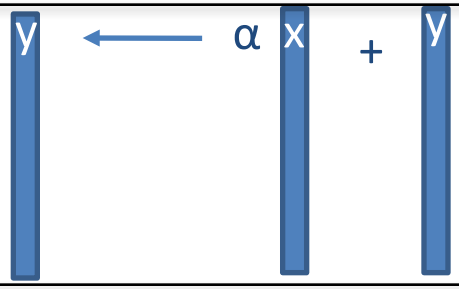


(Omitting latency here.)

The model IS simplified (see next slide) but it provides an upper bound on performance as well. I.e., we will never go faster than what the model predicts. (And, of course, we can go slower …)

FMA: fused multiply-add

AXPY:

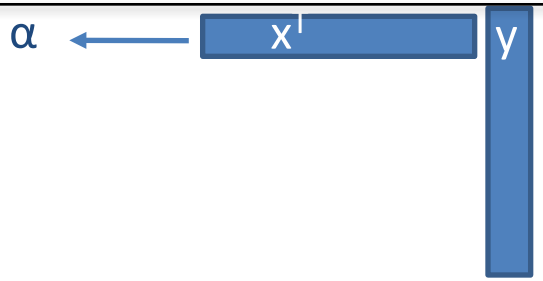



```
for ( j = 0; j < n; j++)
      y[i] += a * x[i];
```

(without increment)

n MUL
n ADD
2n FLOP
n FMA

DOT:




```
alpha = 0e+00;
for ( j = 0; j < n; j++)
      alpha += x[i] * y[i];
```

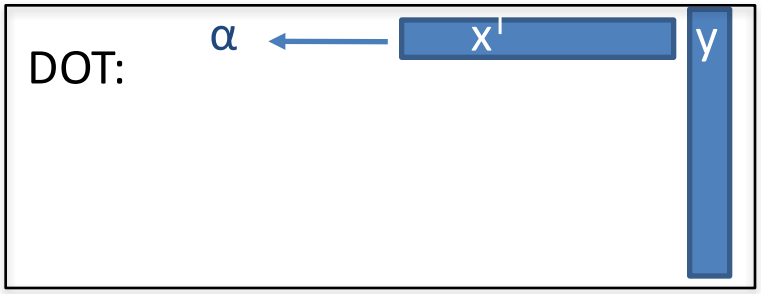
(without increment)

n MUL
n ADD
2n FLOP
n FMA

Note: It is reasonable to expect the one loop codes shown here to perform as well as their Level 1 BLAS counterpart (on multicore with an OpenMP pragma for example).

The true gain these days with using the BLAS is (1) Level 3 BLAS, and (2) portability.

- Take two double precision vectors x and y of size n=375,000.



- Data size:
  - (375,000 double) * (8 Bytes / double) = 3 MBytes per vector

(Two vectors fit in cache (6 Mbytes))

- Time to move the vectors from memory to cache:
  - (6 MBytes) / (25.6 GBytes/sec) = **0.23 ms**
- Time to perform computation of DOT:
  - (2n flop) / (56 Gflop/sec) = **0.01 ms**

Vector Operations

total_time ≥ max (time_comm , time_comp)

= max (0.23ms , 0.01ms) = 0.23ms

Performance = (2 x 375,000 flops)/.23ms = 3.2 Gflop/s

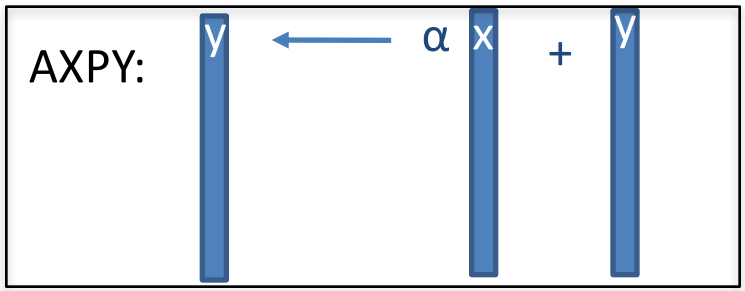
Performance for DOT ≤ 3.2 Gflop/s

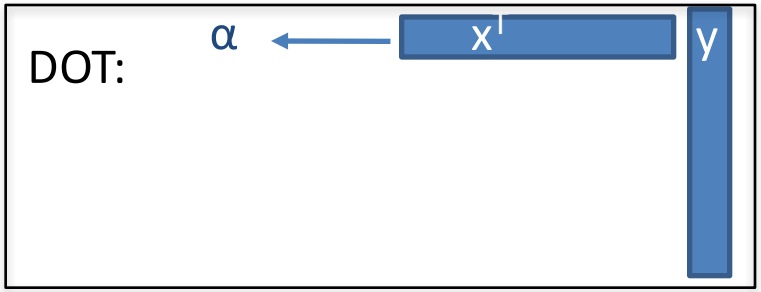
Peak is 56 Gflop/s

We say that the operation is communication bounded. No reuse of data.

Level 1, 2 and 3 BLAS

Level 1 BLAS Matrix-Vector operations

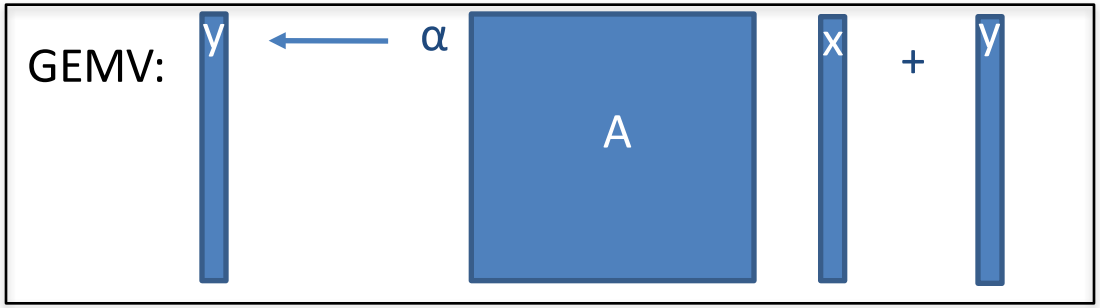




2n FLOP
2n memory reference
AXPY: 2n READ, n WRITE
DOT: 2n READ

RATIO: 1

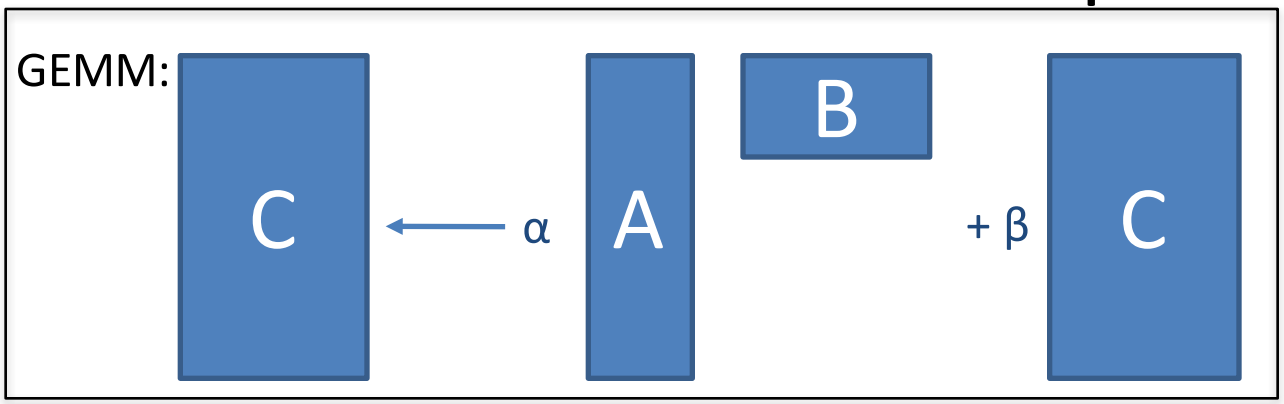
Level 2 BLAS Matrix-Vector operations



$2n^2$ FLOP
n^2 memory references

RATIO: 2

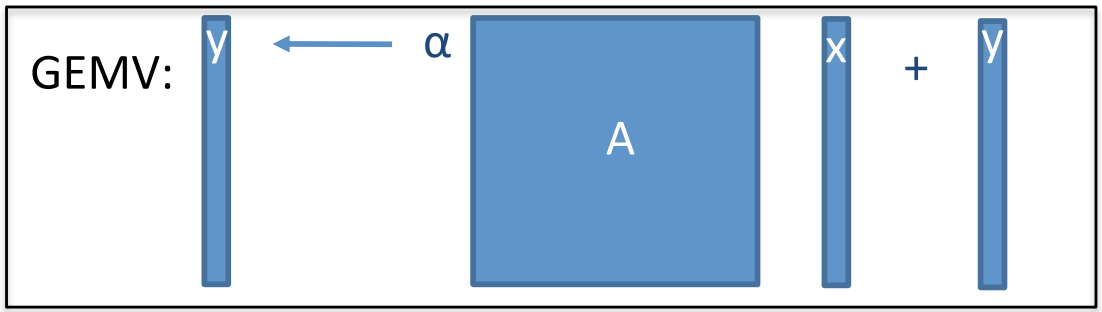
Level 3 BLAS Matrix-Matrix operations



$2n^3$ FLOP
$3n^2$ memory references
$3n^2$ READ, n^2 WRITE

RATIO: 2/3 n

- Double precision matrix A and vectors x and y of size n=860.



- Data size:
  - (860^2 + 2*860 double) * (8 Bytes / double) ~ 6 MBytes

  Matrix and two vectors fit in cache (6 MBytes).

- Time to move the data from memory to cache:
  - (6 MBytes) / (25.6 GBytes/sec) = **0.23 ms**
- Time to perform computation of DOT:
  - ($2n^2$ flop) / (56 Gflop/sec) = **0.26 ms**

Matrix - Vector Operations

total_time ≥ max (time_comm , time_comp)

= max (0.23ms , 0.26ms) = 0.26ms

Performance = (2 x 860^2 flops)/.26ms = 5.7 Gflop/s

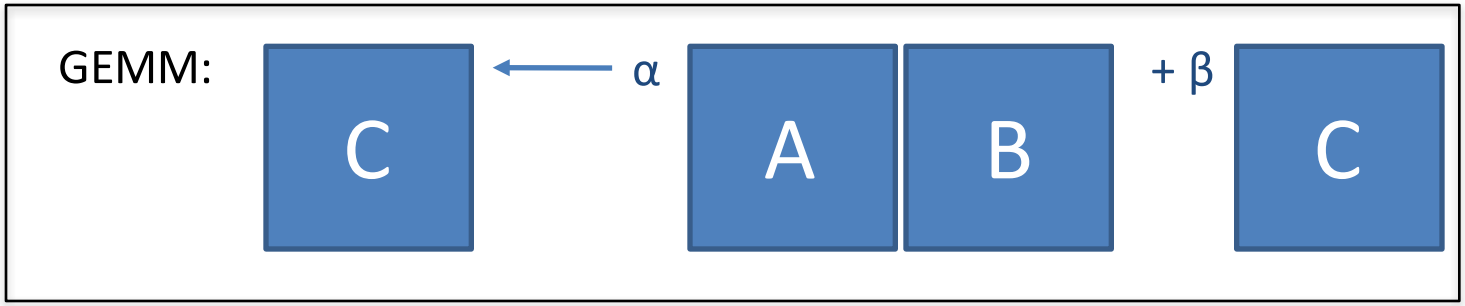
Performance for GEMV ≤ 5.7 Gflop/s

Performance for DOT ≤ 3.2 Gflop/s

Peak is 56 Gflop/s

We say that the operation is communication bounded. Very little reuse of data.

- Take two double precision vectors x and y of size n=500.



- Data size:
  - (500^2 double) * (8 Bytes / double) = 2 MBytes per matrix

  (Three matrices fit in cache (6 MBytes). OK.)

- Time to move the matrices in cache:
  - (6 MBytes) / (25.6 GBytes/sec) = **0.23 ms**

- Time to perform computation in GEMM:
  - ($2n^3$ flop) / (56 Gflop/sec) = **4.46 ms**

Matrix Matrix Operations

total_time ≥ max (time_comm , time_comp)

= max(0.23ms , 4.46ms) = 4.46ms

For this example, communication time is less than 6% of the computation time.

Performance = (2 x 500^3 flops)/4.69ms = 53.3 Gflop/s

There is a lots of data reuse in a GEMM; 2/3n per data element. Has good temporal locality.

If we assume total_time ≈ time_comm +time_comp, we get

Performance for GEMM ≈ 53.3 Gflop/sec

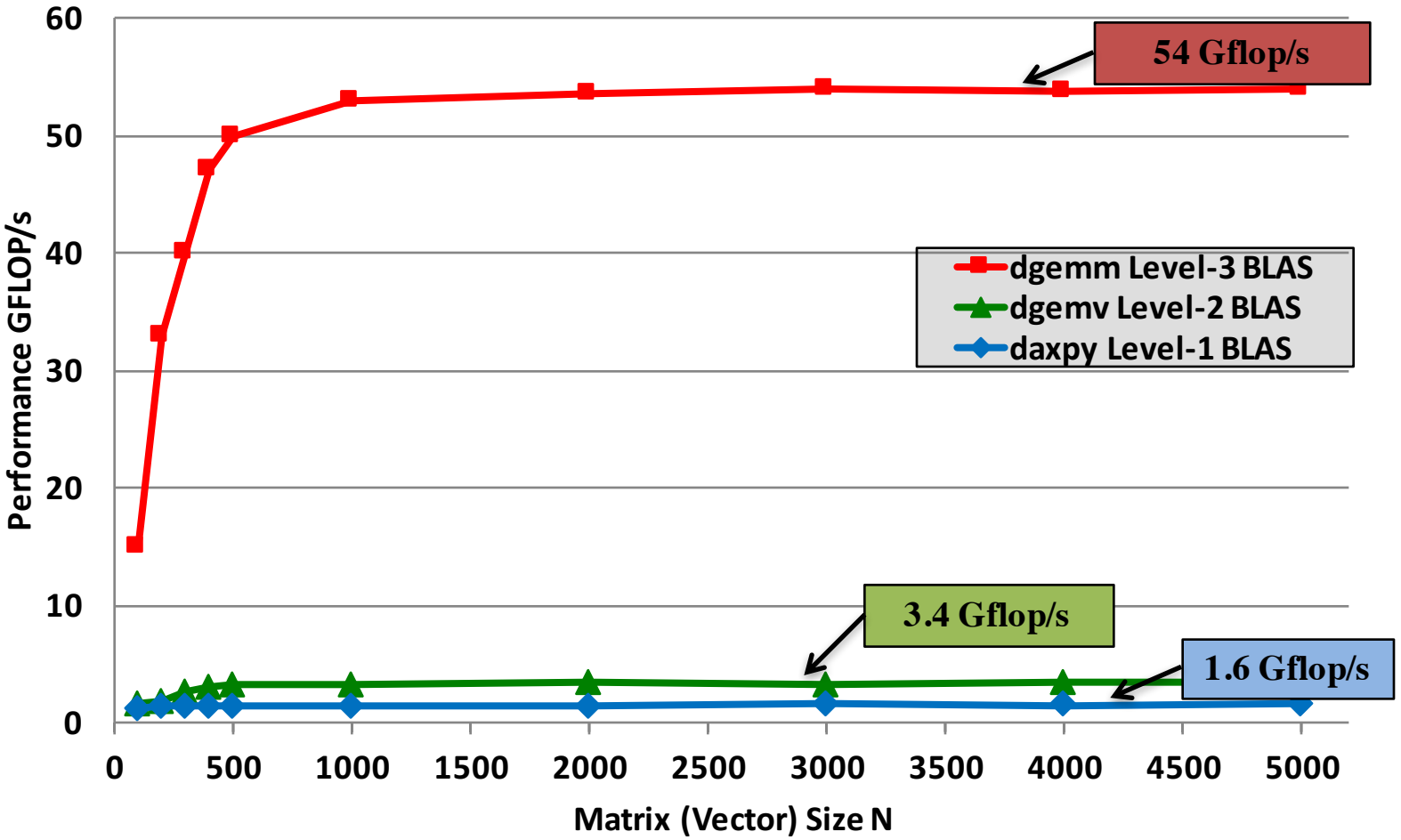
Performance for DOT ≤ 3.2 Gflop/s

Performance for GEMV ≤ 5.7 Gflop/s

(Out of 56 Gflop/sec possible, so that would be 95% peak performance efficiency.)

Level 1, 2 and 3 BLAS

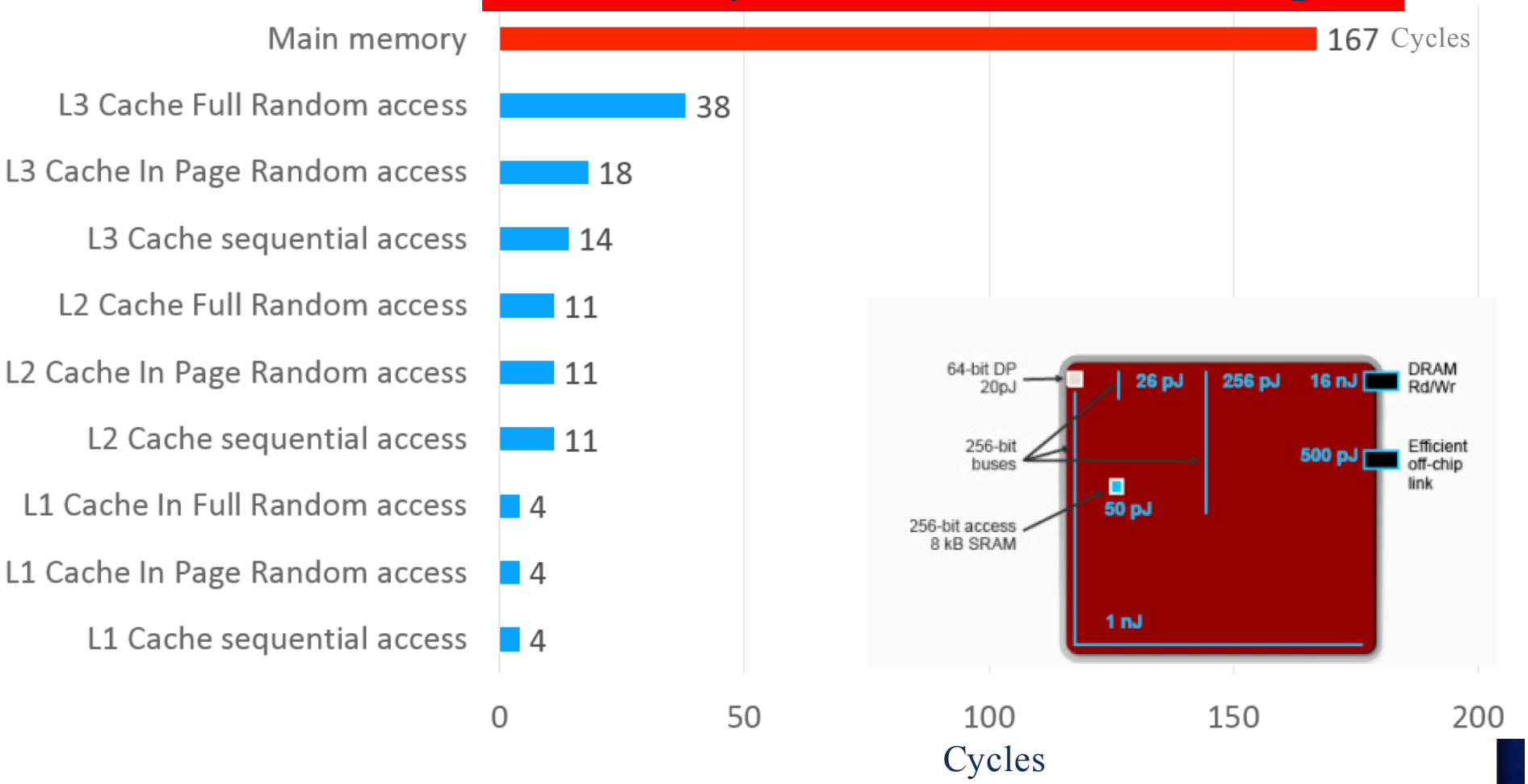
1 core Intel Haswell i7-4850HQ, 2.3 GHz (Turbo Boost at 3.5 GHz);
Peak = 56 Gflop/s



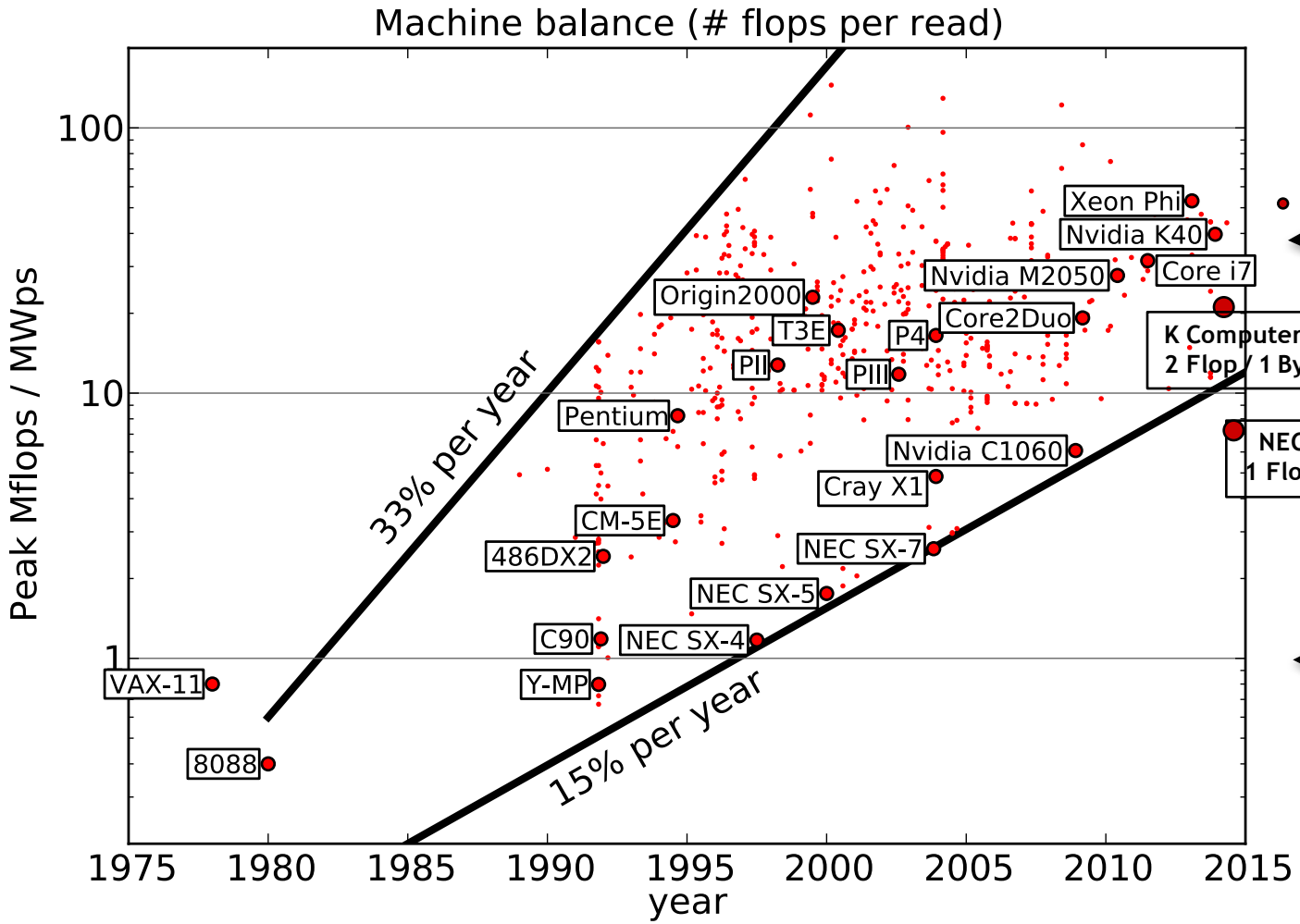
1 core Intel Haswell i7-4850HQ, 2.3 GHz, Memory: DDR3L-1600MHz
6 MB shared L3 cache, and each core has a private 256 KB L2 and 64 KB L1.
The theoretical peak per core double precision is 56 Gflop/s per core.
Compiled with gcc and using Veclib

CPU Access Latencies in Clock Cycles

In 167 cycles can do 2672 DP Flops



Ratio of CPU speed to memory bandwidth increases 15-33% yearly

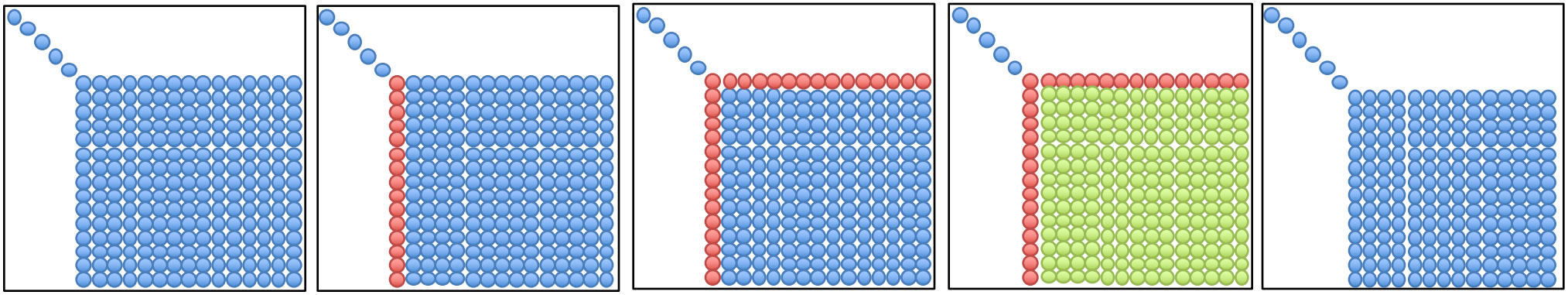


- Flops "free," memory expensive
- Good for dense, BLAS-3 operations (matrix multiply)

- Flops & memory access balanced
- Good for sparse & vector operations

Data from Stream benchmark (McCalpin) and vendor information sheets

The Standard LU Factorization LINPACK 1970's HPC of the Day: Vector Architecture



Factor column with Level 1 BLAS

Divide by Pivot row

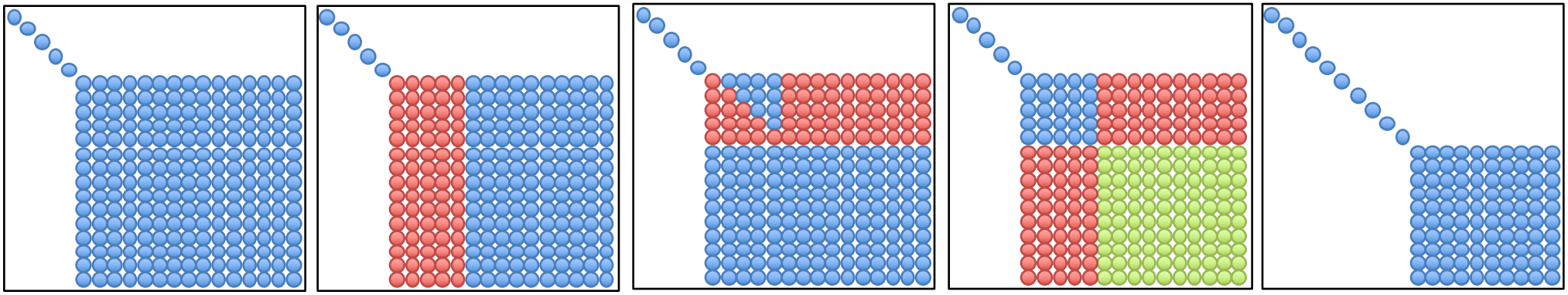
Schur complement update (Rank 1 update)

Next Step

Main points

- Factorization column (zero) mostly sequential due to memory bottleneck
- Level 1 BLAS
- Divide pivot row has little parallelism
- Rank -1 Schur complement update is the only easy parallelize task
- Partial pivoting complicates things even further
- Bulk synchronous parallelism (fork-join)
  - Load imbalance
  - Non-trivial Amdahl fraction in the panel
  - Potential workaround (look-ahead) has complicated implementation

The Standard LU Factorization LAPACK 1980's HPC of the Day: Cache Based SMP



Factor panel with Level 1,2 BLAS

Triangular update

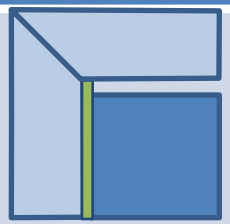
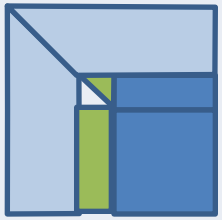
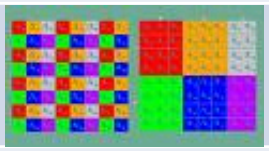
Schur complement update

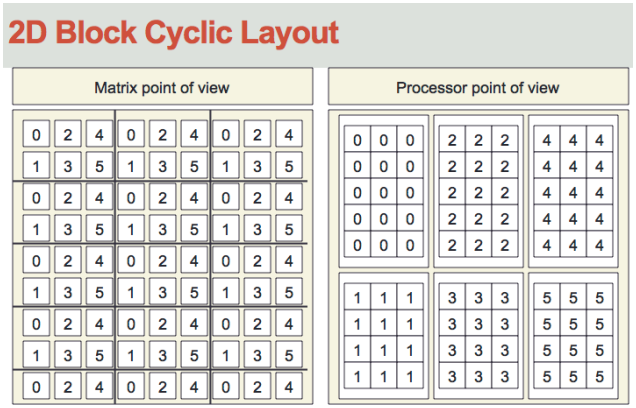
Next Step

Main points

- Panel factorization mostly sequential due to memory bottleneck
- Triangular solve has little parallelism
- Schur complement update is the only easy parallelize task
- Partial pivoting complicates things even further
- Bulk synchronous parallelism (fork-join)
  - Load imbalance
  - Non-trivial Amdahl fraction in the panel
  - Potential workaround (look-ahead) has complicated implementation

Last Generations of DLA Software

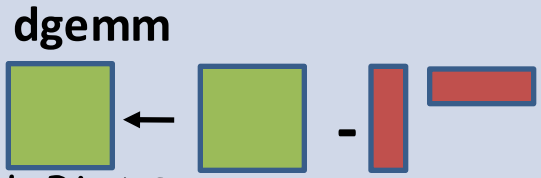
Software/Algorithms follow hardware evolution in time		
LINPACK (70's) (Vector operations)		Rely on - Level-1 BLAS operations
LAPACK (80's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
ScaLAPACK (90's) (Distributed Memory)		Rely on - PBLAS Mess Passing

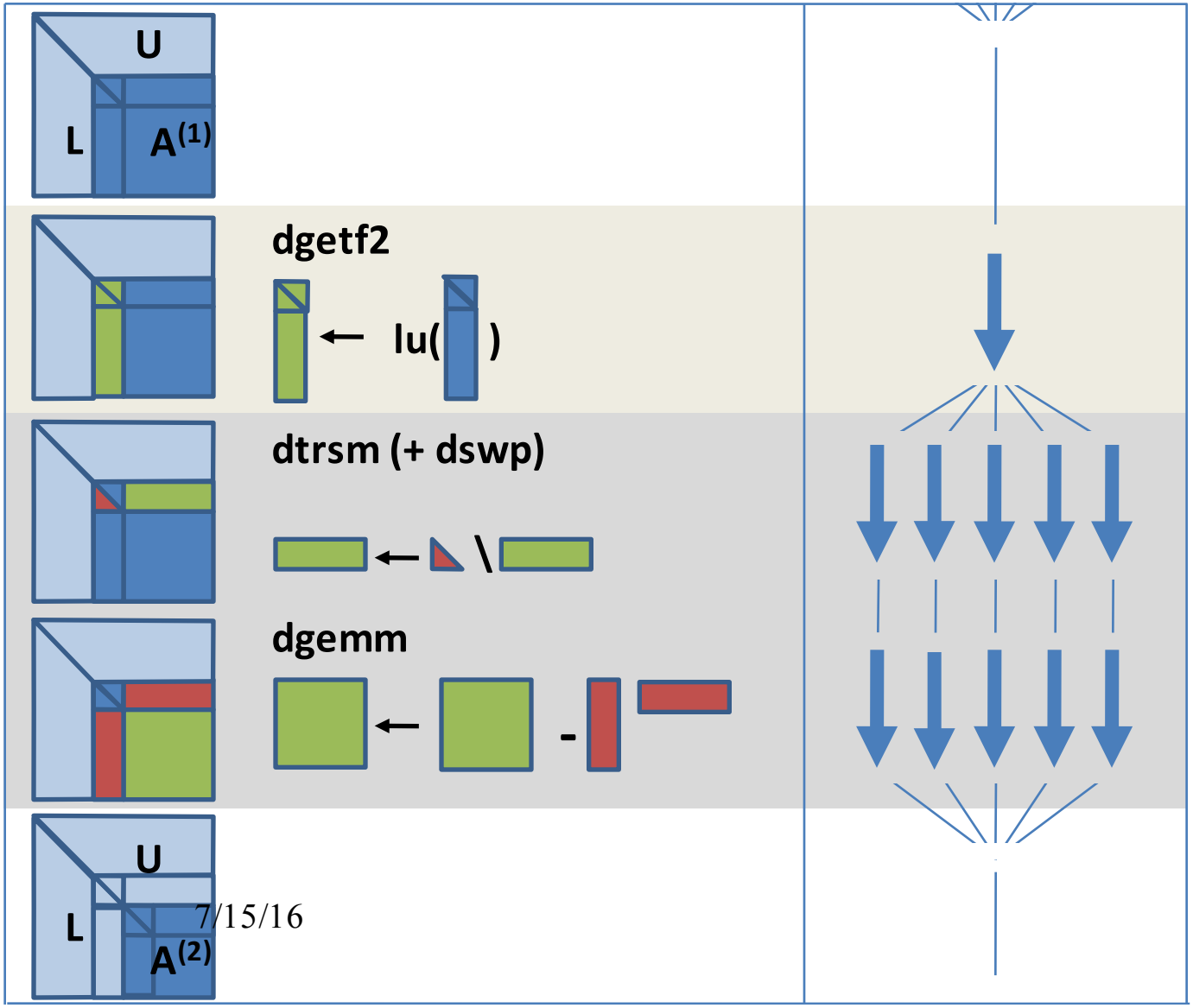


Parallelization of LU and QR.

Parallelize the update:

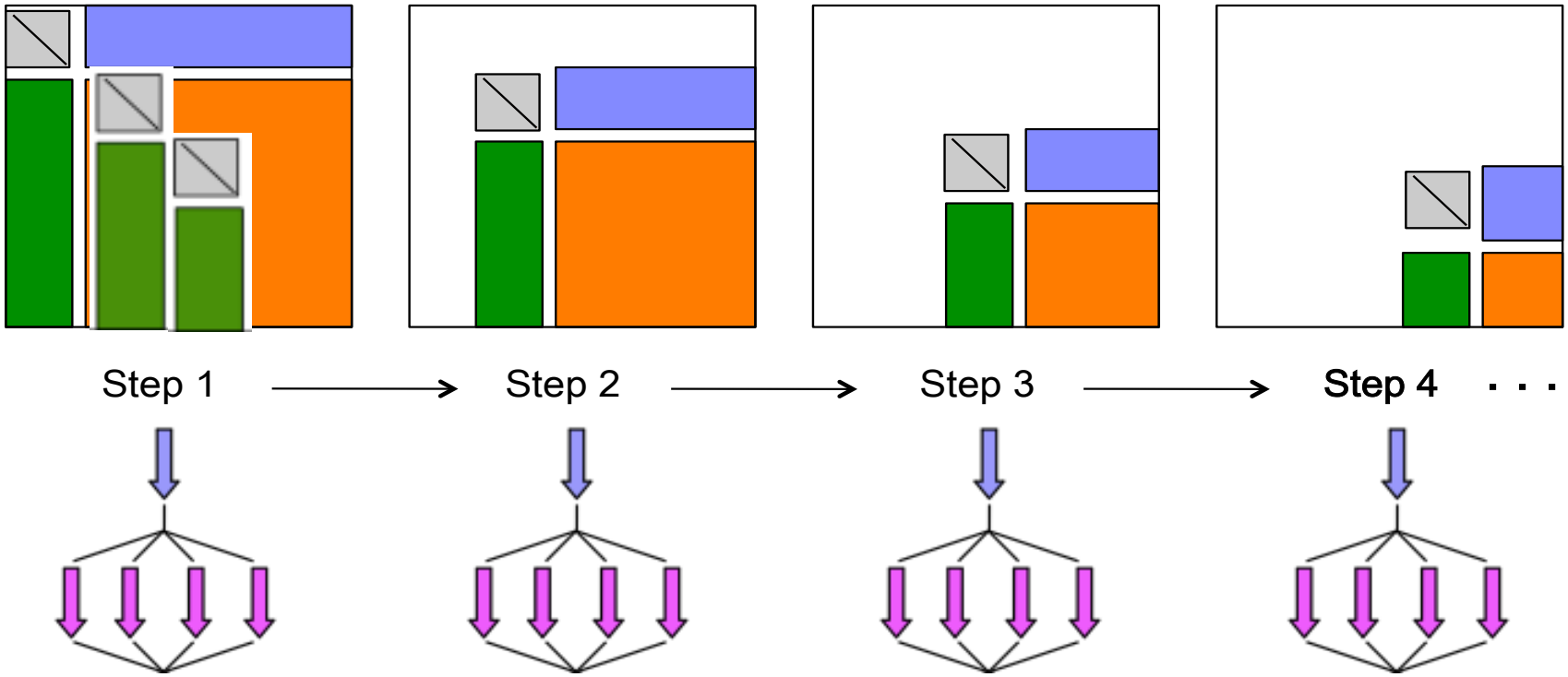
- Easy and done in any reasonable software.
- This is the $2/3n^3$ term in the FLOPs count.
- Can be done efficiently with LAPACK+multithreaded BLAS

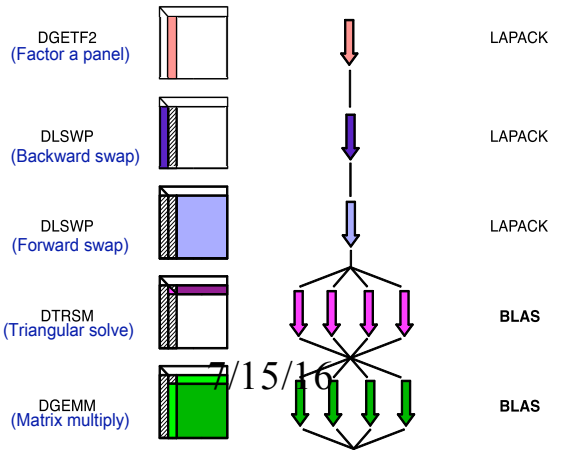




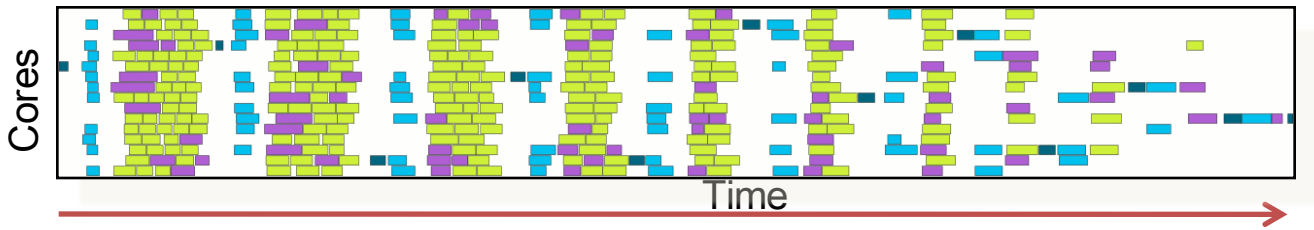
Fork - Join parallelism
Bulk Sync Processing

Synchronization (in LAPACK LU)



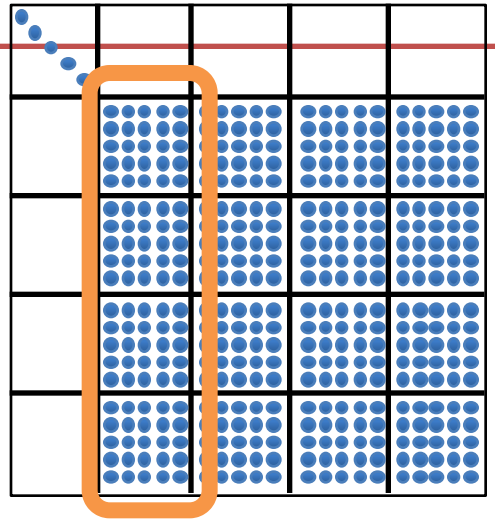


- fork join
- bulk synchronous processing

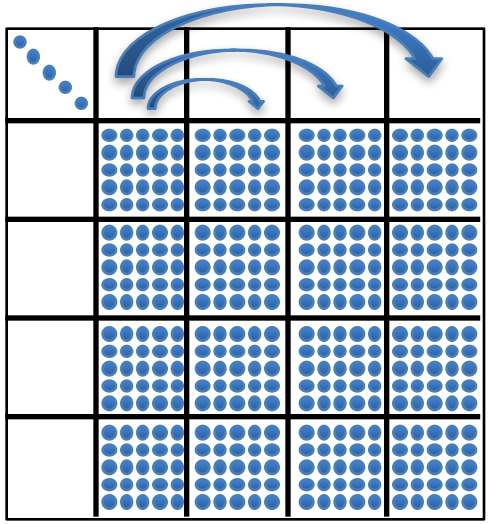


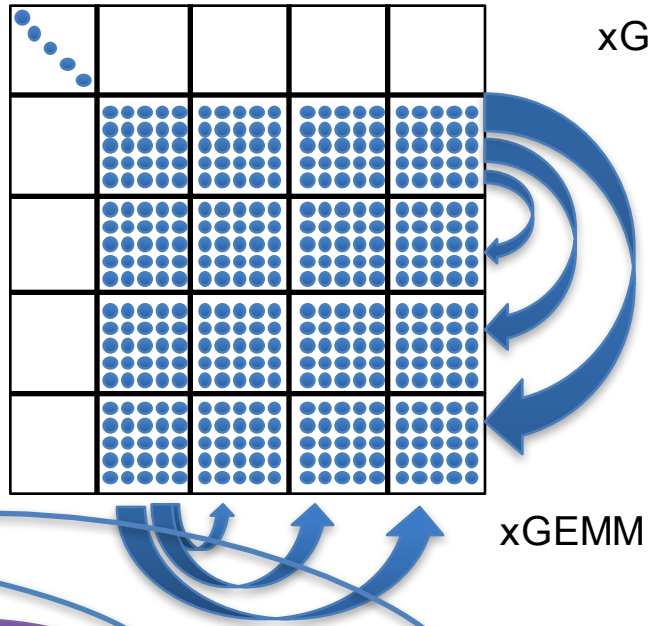
PLASMA LU Factorization

Dataflow Driven



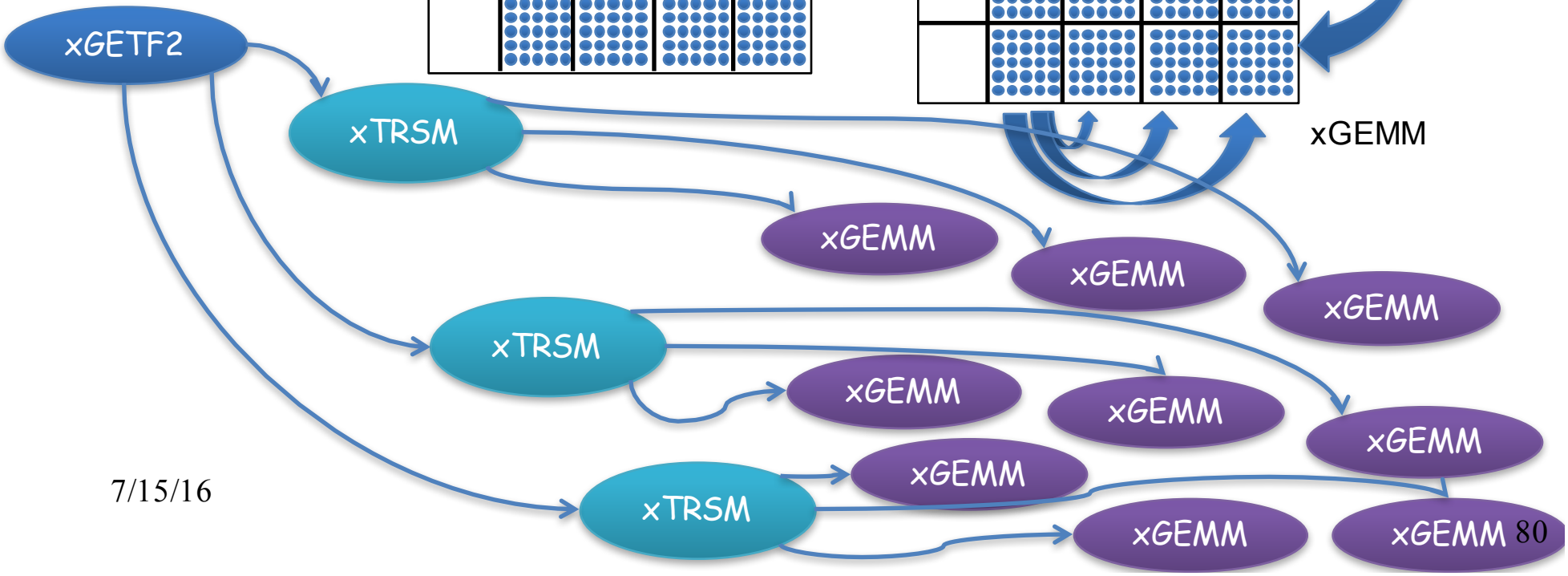
xTRSM





xGEMM

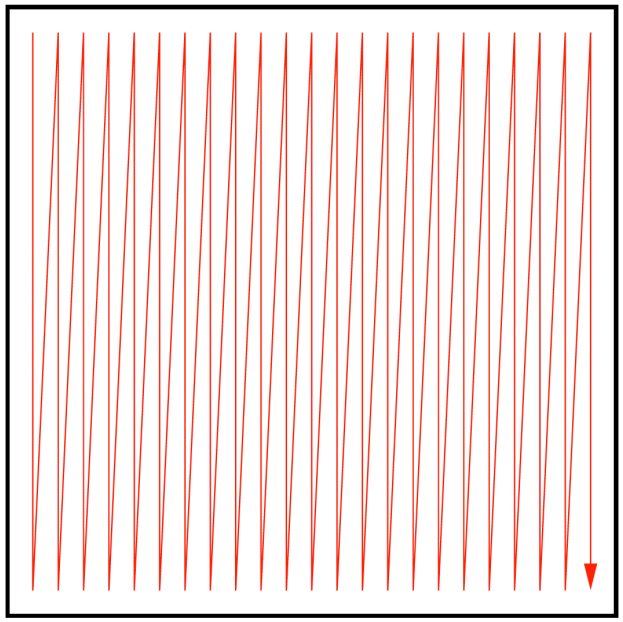
Numerical program generates tasks and run time system executes tasks respecting data dependences.

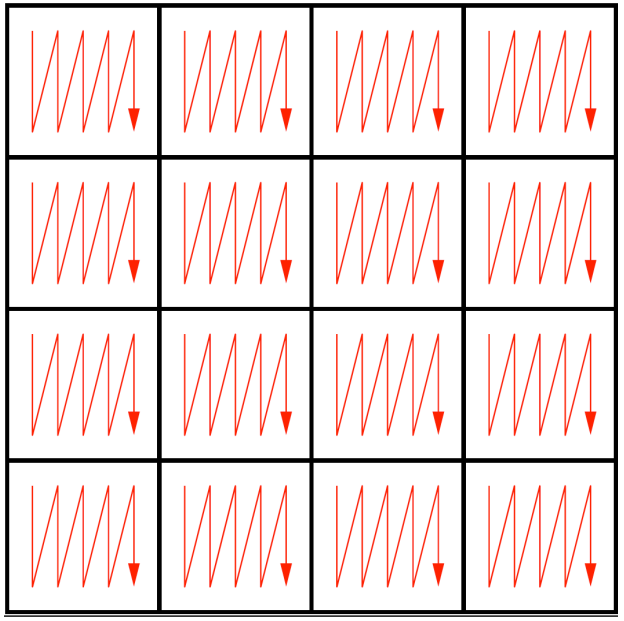


xGEMM

xGEMM

Data Layout is Critical

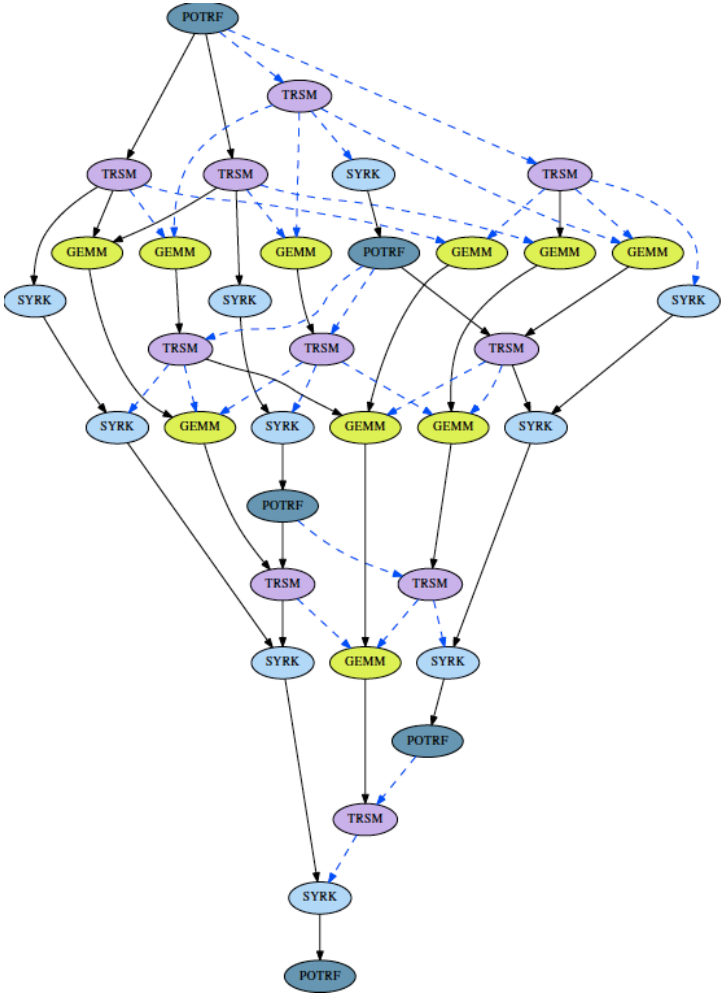




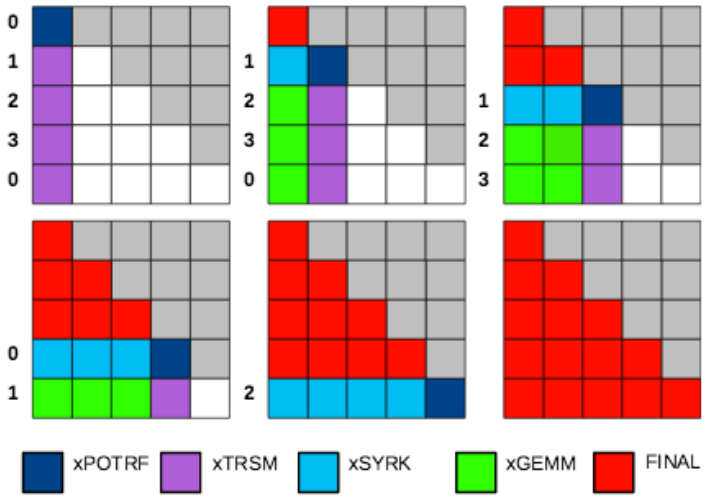
- Tile data layout where each data tile is contiguous in memory
- Decomposed into several fine-grained tasks, which better fit the memory of the small core caches

OpenMP Tasking

- Added with OpenMP 3.0 (2009)
- Allows parallelization of irregular problems
- OpenMP 4.0 (2013) - Tasks can have dependencies
  - DAGs



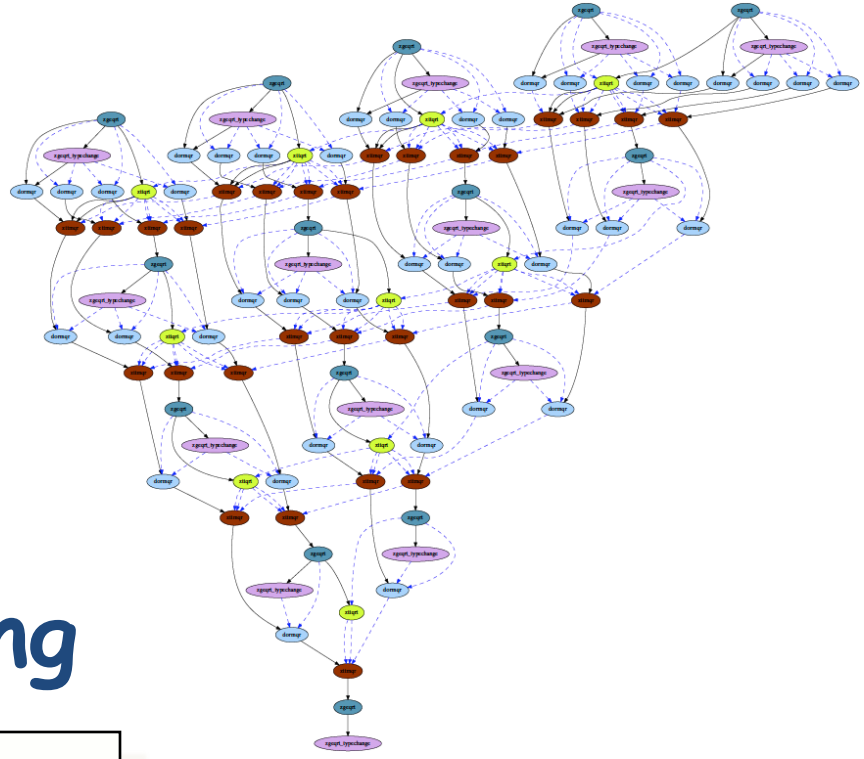
Tiled Cholesky Decomposition



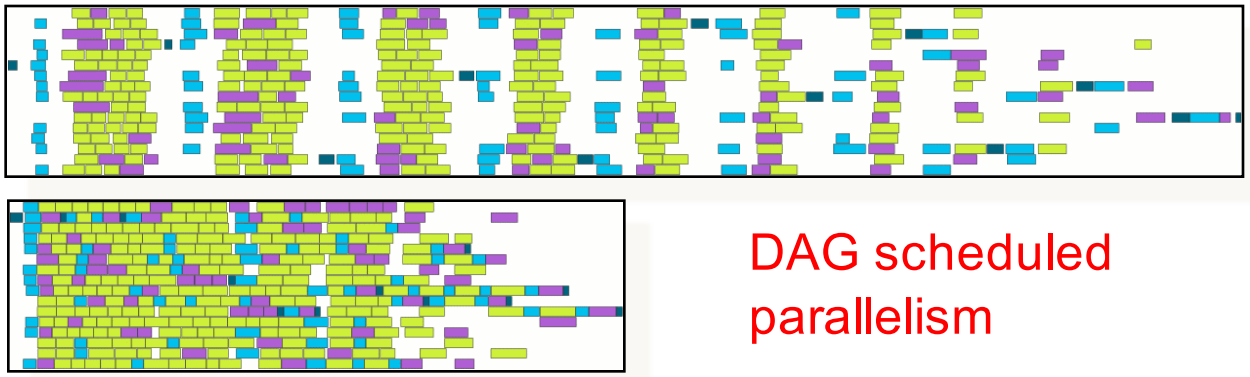

```
#pragma omp parallel
#pragma omp master
{  CHOLESKY( A );  }
CHOLESKY( A ) {
    for (k = 0; k < M; k++) {
        #pragma omp task depend(inout:A(k,k)[0:tilesize]
        {  POTRF( A(k,k) );  }
        for (m = k+1; m < M; m++) {
            #pragma omp task \
                depend(in:A(k,k)[0:tilesize]) \
                depend(inout:A(m,k)[0:tilesize])
            { TRSM( A(k,k), A(m,k) ); }
        }
        for (m = k+1; m < M; m++) {
            #pragma omp task \
                depend(in:A(m,k)[0:tilesize]) \
                depend(inout:A(m,m)[0:tilesize])
            { SYRK( A(m,k), A(m,m) ); }
            for (n = k+1; n < m; n++) {
                #pragma omp task \
                    depend(in:A(m,k)[0:tilesize], \
                           A(n,k)[0:tilesize]) \
                    depend(inout:A(m,n)[0:tilesize])
                {  GEMM( A(m,k), A(n,k), A(m,n) ); }
            }
        }
    }
}
```

The Purpose of a QUARK Runtime

- Objectives
  - High utilization of each core
  - Scaling to large number of cores
  - Synchronization reducing algorithms
- Methodology
  - Dynamic DAG scheduling
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout



- Arbitrary DAG with dynamic **scheduling**

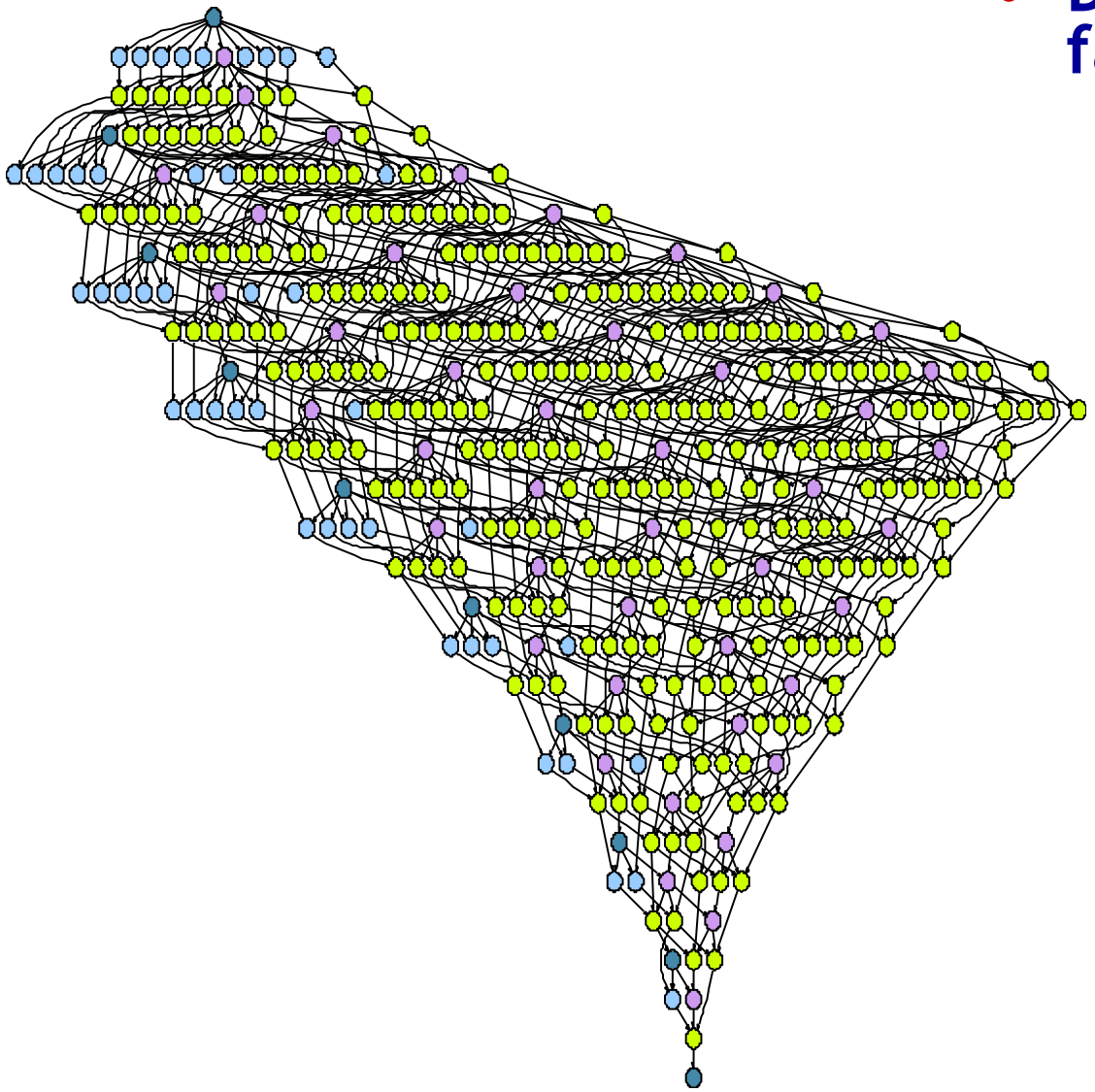


DAG scheduled parallelism

Fork-join parallelism
Notice the synchronization penalty in the presence of heterogeneity.

PLASMA Local Scheduling

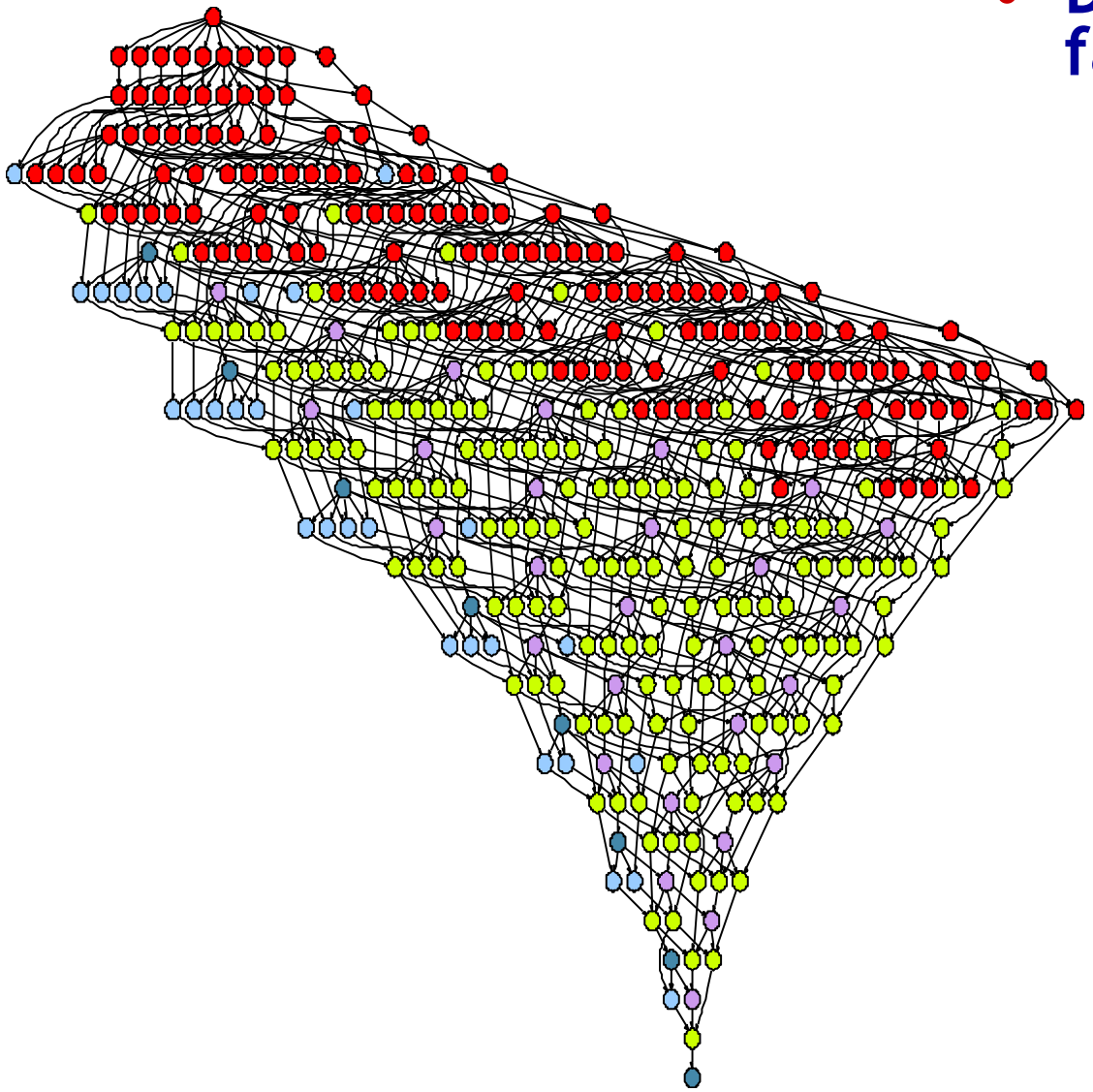
Dynamic Scheduling: Sliding Window



- DAGs get very big, very fast
  - So windows of active tasks are used; this means no global critical path
  - Matrix of NBxNB tiles; NB^3 operation
    - NB=100 gives 1 million tasks

PLASMA Local Scheduling

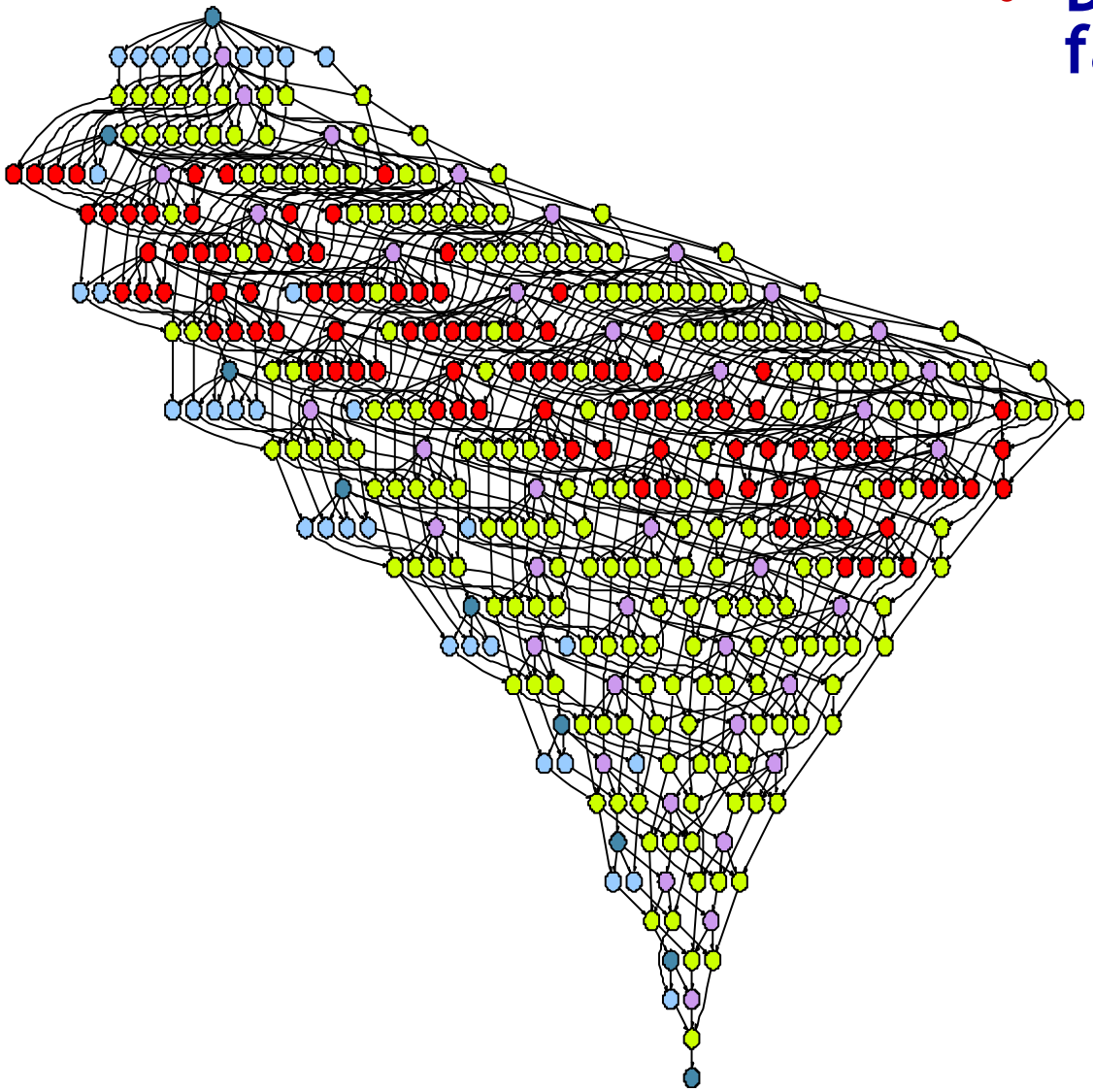
Dynamic Scheduling: Sliding Window



- DAGs get very big, very fast
  - So windows of active tasks are used; this means no global critical path
  - Matrix of NBxNB tiles; NB^3 operation
    - NB=100 gives 1 million tasks

PLASMA Local Scheduling

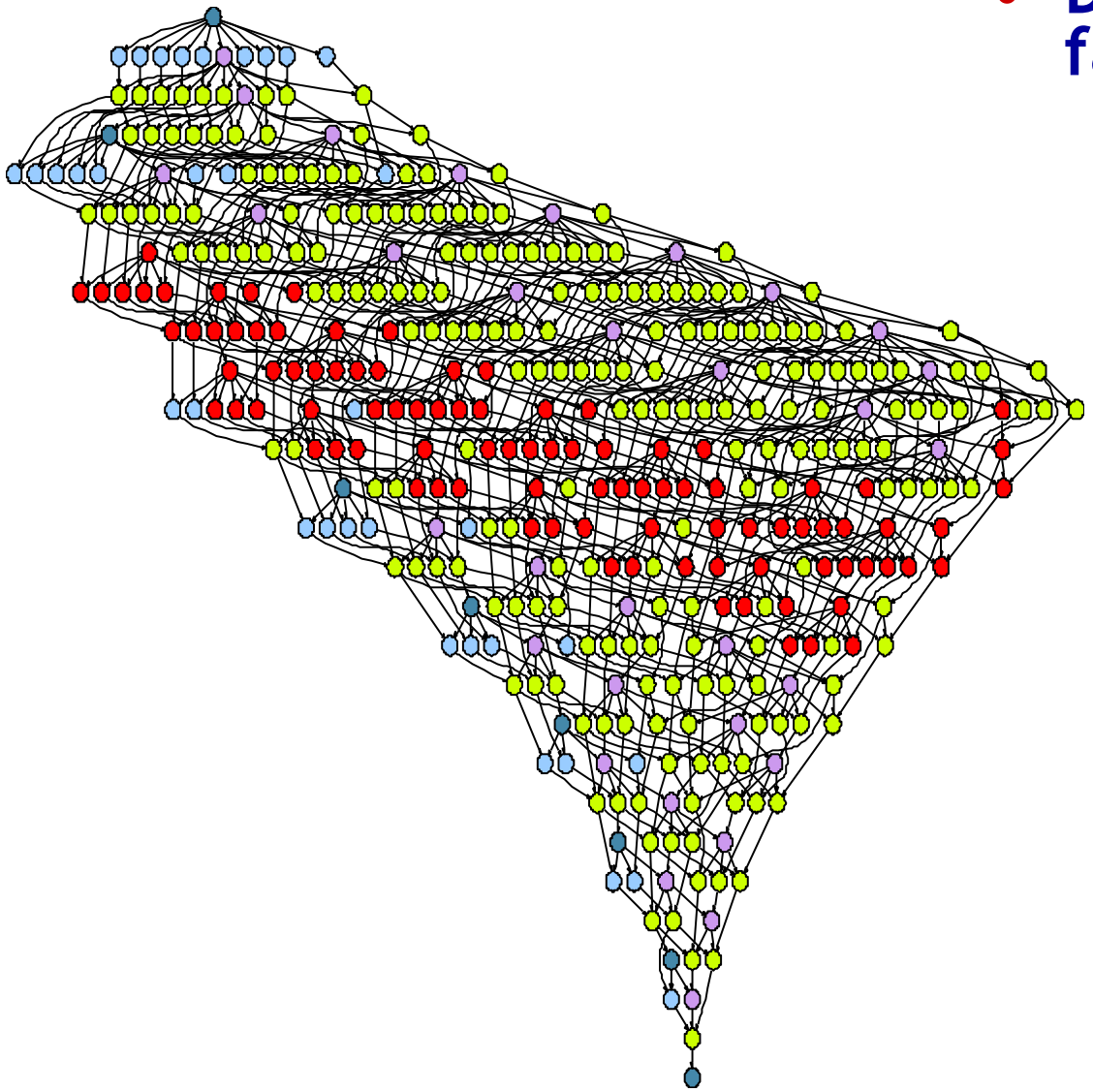
Dynamic Scheduling: Sliding Window



- DAGs get very big, very fast
  - So windows of active tasks are used; this means no global critical path
  - Matrix of NBxNB tiles; NB^3 operation
    - NB=100 gives 1 million tasks

PLASMA Local Scheduling

Dynamic Scheduling: Sliding Window

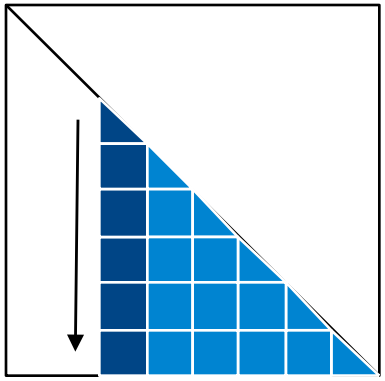


- DAGs get very big, very fast
  - So windows of active tasks are used; this means no global critical path
  - Matrix of NBxNB tiles; NB^3 operation
    - NB=100 gives 1 million tasks

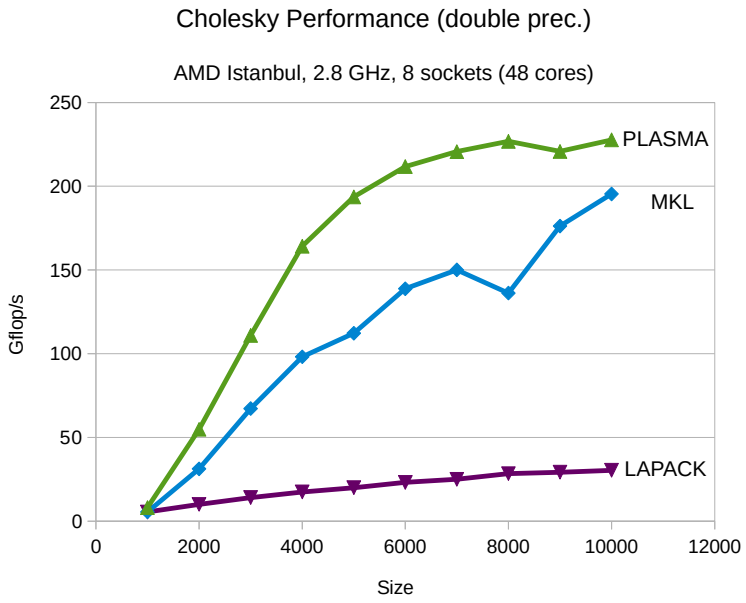
Algorithms

Cholesky

PLASMA_[scdz]**potrf**[_Tile][_Async]()



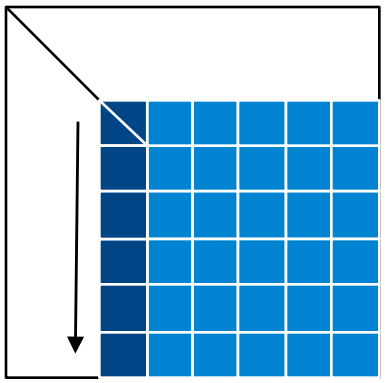
- **Algorithm**
  - equivalent to LAPACK
- **Numerics**
  - same as LAPACK
- **Performance**
  - comparable to vendor on few cores
  - much better than vendor on many cores



Algorithms

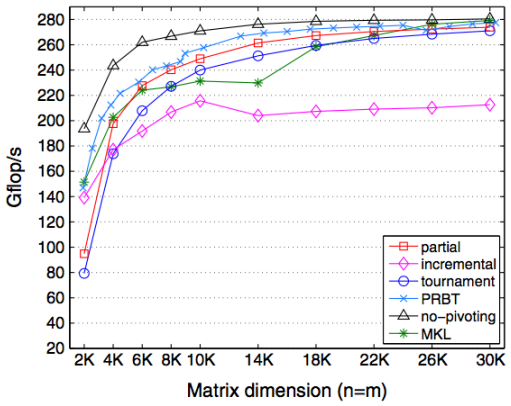
LU

PLASMA_[scdz]**getrf**[_Tile][_Async]()

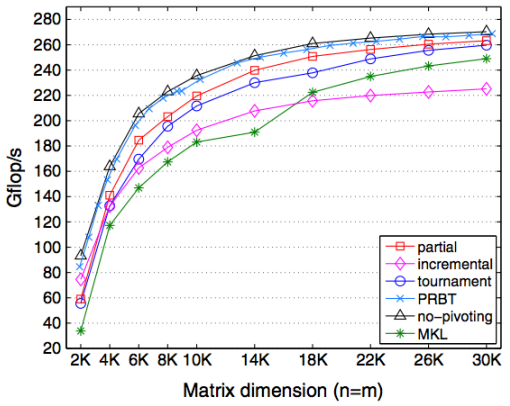


- **Algorithm**
  - equivalent to LAPACK
  - same pivot vector
  - same L and U factors
  - same forward substitution procedure
- **Numerics**
  - same as LAPACK
- **Performance**
  - comparable to vendor on few cores
  - much better than vendor on many cores

16 Sandy Bridge cores



Factorization alone, using16 cores

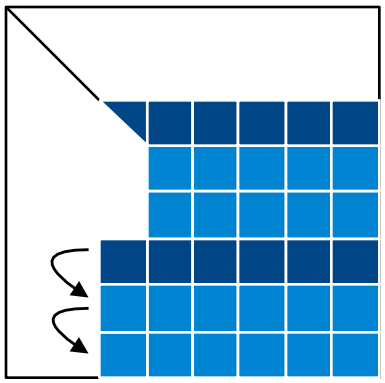


Factorization and solve with iterative refinement, using 16 cores

Algorithms

incremental QR Factorization

PLASMA_[scdz]**geqrt**[_Tile][_Async]()



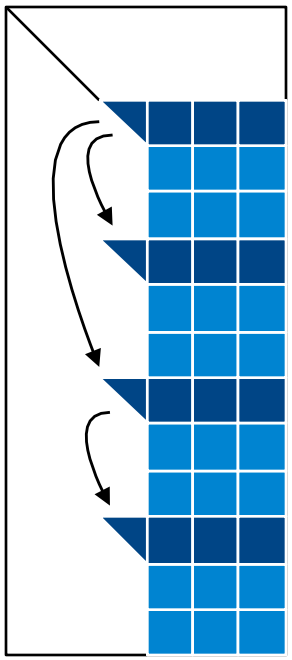
- **Algorithm**
  - the same R factor as LAPACK (absolute values)
  - different set of Householder reflectors
  - different Q matrix
  - different Q generation / application procedure
- **Numerics**
  - same as LAPACK
- **Performance**
  - comparable to vendor on few cores
  - much better than vendor on many cores

Algorithms

incremental QR Factorization (Communication Avoiding)

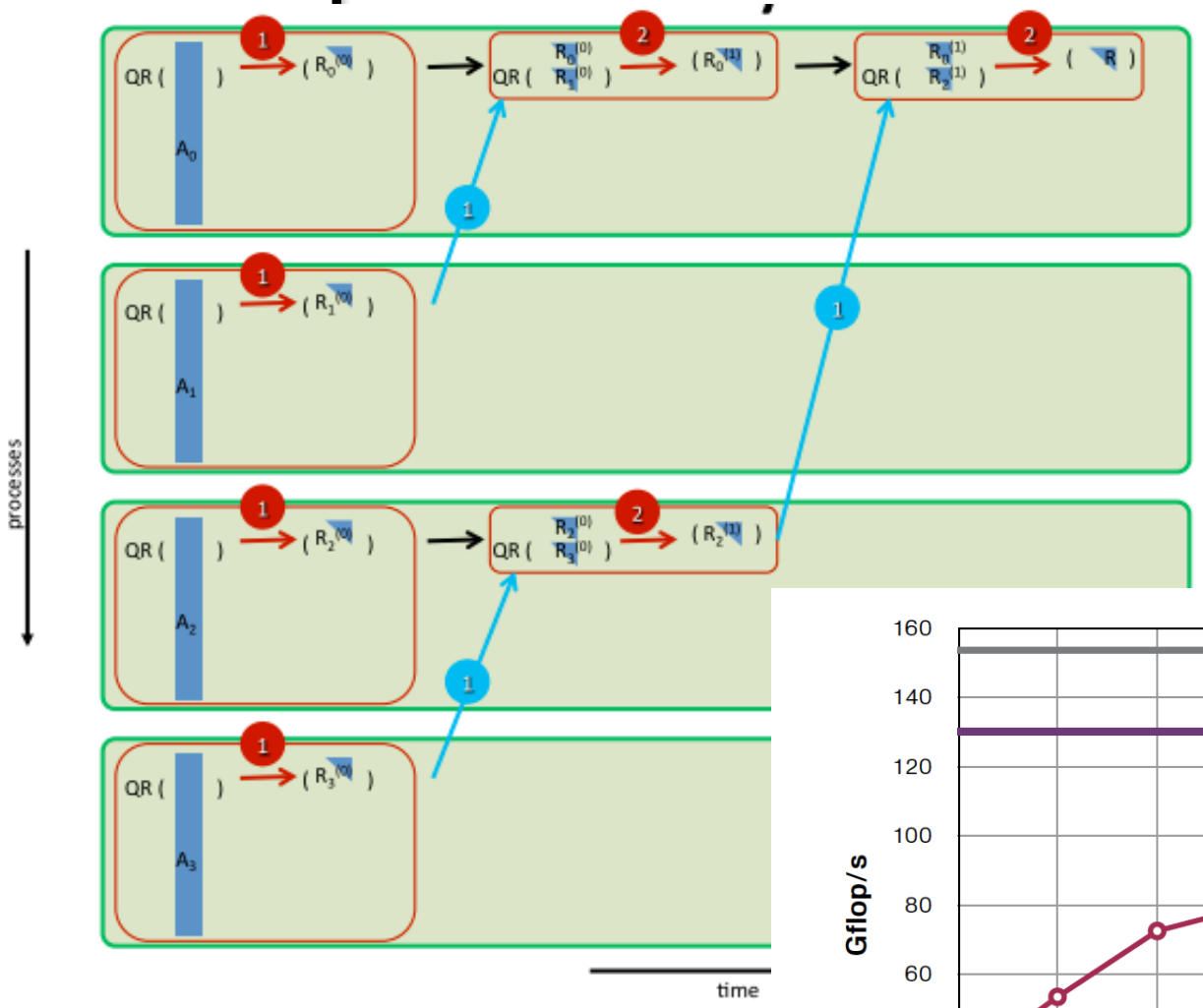
PLASMA_[scdz]**geqrt**[_Tile][_Async]()

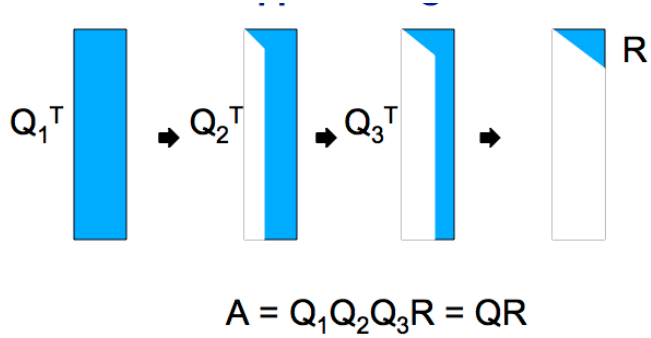
```
PLASMA_Set(
    PLASMA_HOUSEHOLDER_MODE,
    PLASMA_TREE_HOUSEHOLDER);
```

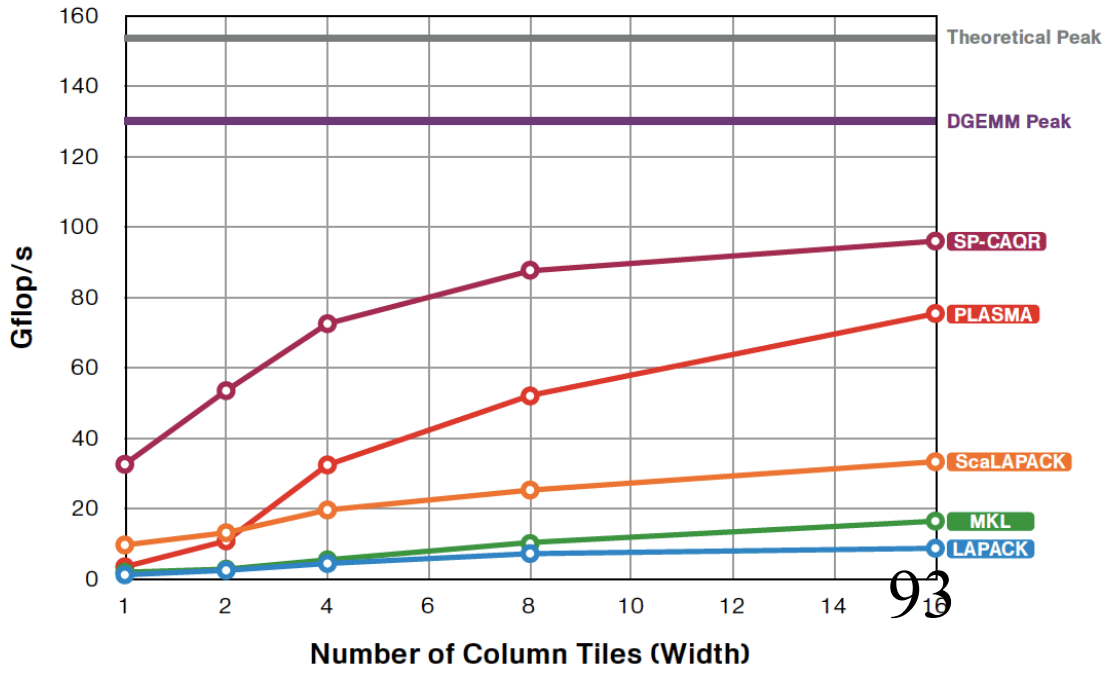


- **Algorithm**
  - the same R factor as LAPACK (absolute values)
  - different set of Householder reflectors
  - different Q matrix
  - different Q generation / application procedure
- **Numerics**
  - same as LAPACK
- **Performance**
  - absolutely superior for tall matrices

Communication Avoiding QR Example





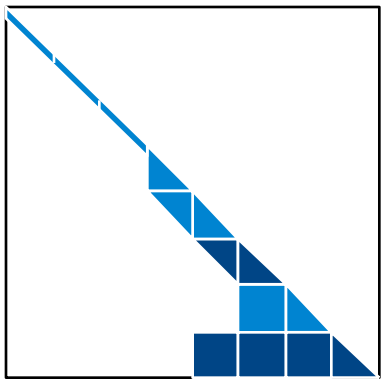


Quad-socket, quad-core machine Intel Xeon EMT64 E7340 at 2.39 GHz.
Theoretical peak is 153.2 Gflop/s with 16 cores.
Matrix size 51200 by 3200

Algorithms

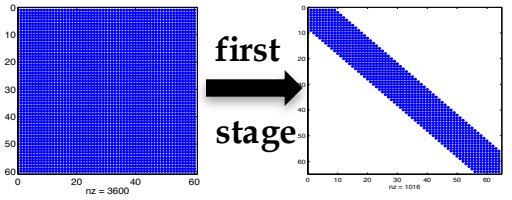
three-stage symmetric EVP

PLASMA_[scdz]**syev**[_Tile][_Async]()

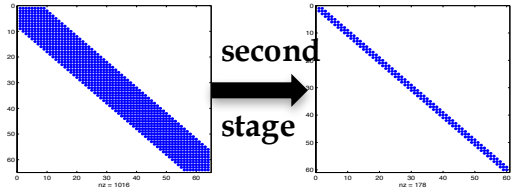


- **Algorithm**
  - two-stage tridiagonal reduction + QR Algorithm
  - fast eigenvalues, slower eigenvectors

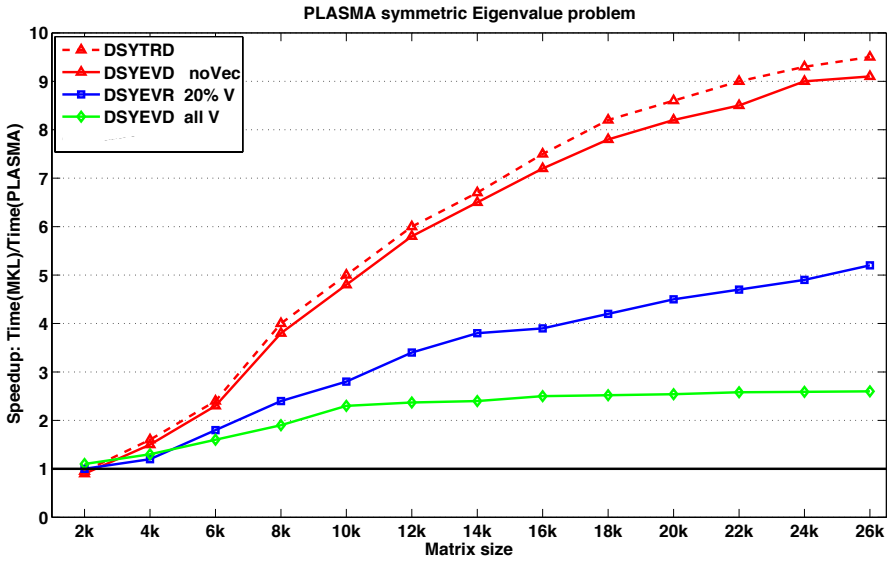
  (possibility to calculate a subset)



- **Numerics**
  - same as LAPACK



- **Performance**
  - comparable to MKL for very small problems
  - absolutely superior for larger problems

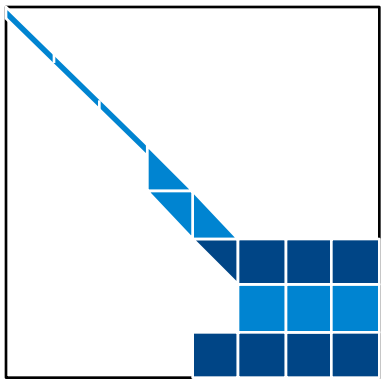


16 cores of Intel Sandy Bridge

Algorithms

three-stage SVD

PLASMA_[scdz]**gesvd**[_Tile][_Async]()

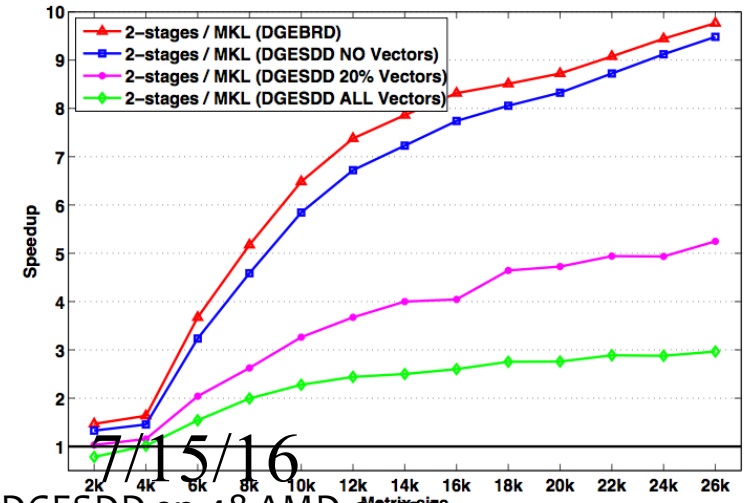


- **Algorithm**
  - two-stage bidiagonal reduction + QR iteration
  - fast singular values, slower singular vectors

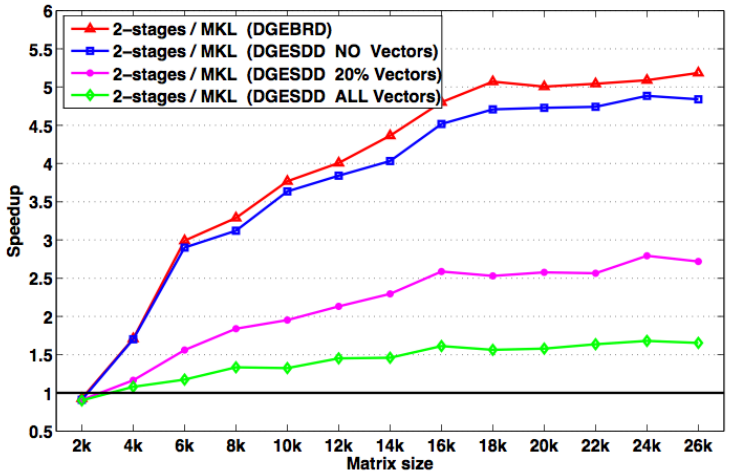
  (possibility of calculating a subset)

- **Numerics**
  - same as LAPACK

- **Performance**
  - comparable with MKL for very small problems
  - absolutely superior for larger problems

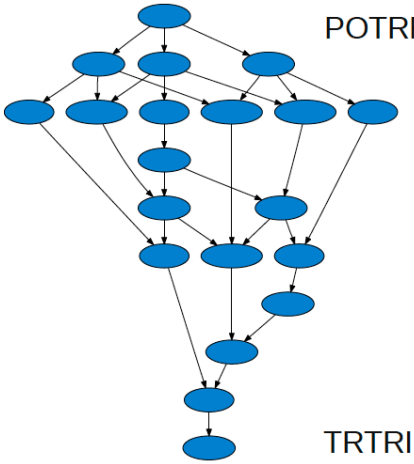


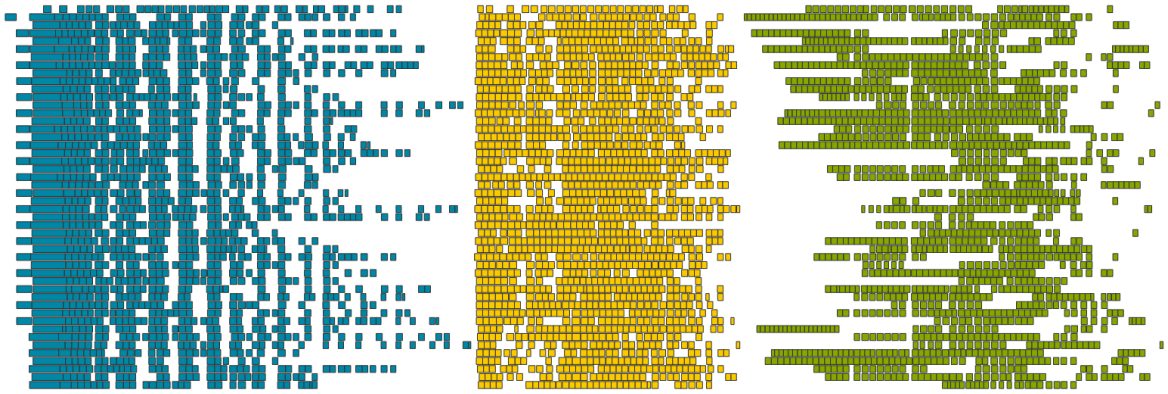
DGESDD on 48 AMD cores

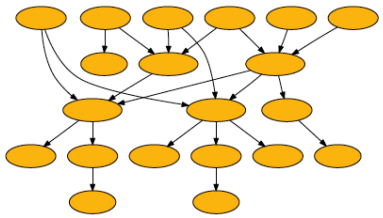


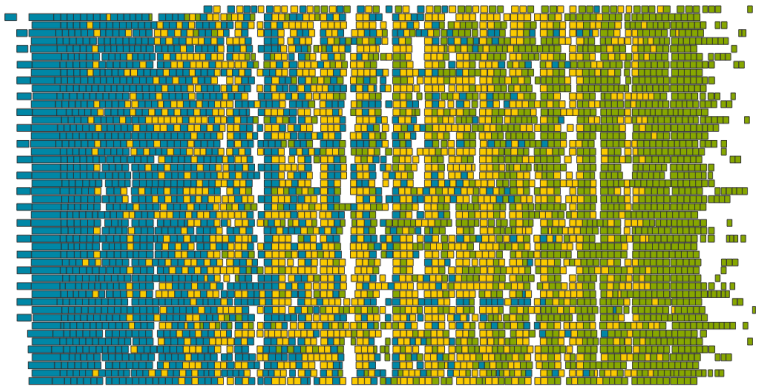
DGESDD on 16 Sandy Bridge cores

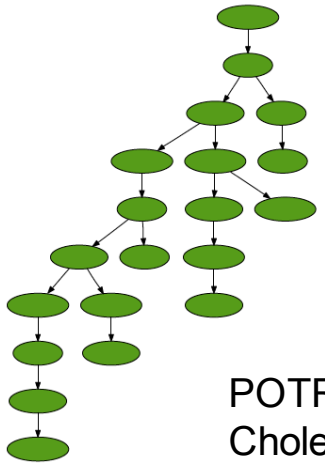
Pipelining: Cholesky Inversion
3 Steps: Factor, Invert L, Multiply L's





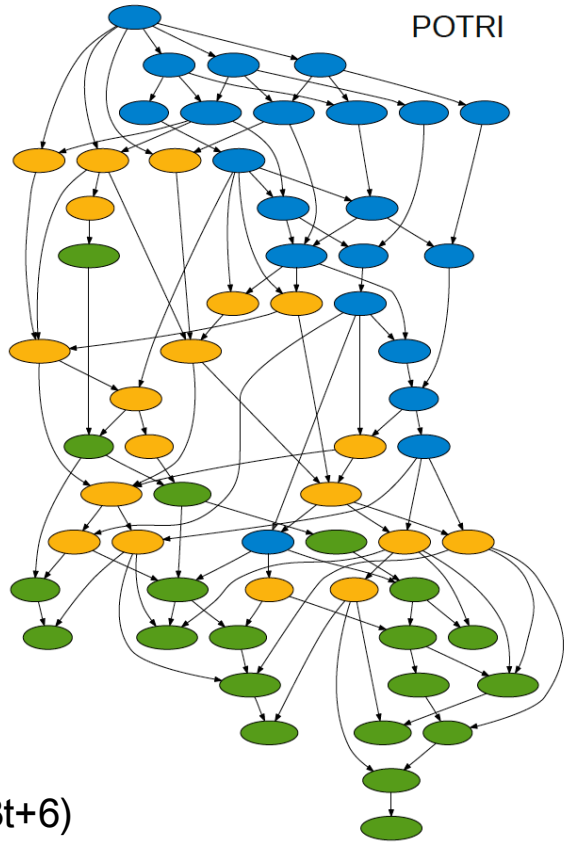






48 cores
POTRF, TRTRI and LAUUM.
The matrix is 4000 x 4000,tile size is 200 x 200,

POTRF+TRTRI+LAUUM: 25 (7t-3)
Cholesky Factorization alone: 3t-2



Pipelined: 18 (3t+6)

Mixed Precision Methods

- Mixed precision, use the lowest precision required to achieve a given accuracy outcome
  - Improves runtime, reduce power consumption, lower data movement
  - Reformulate to find correction to solution, rather than solution; Δx rather than x.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$x_{i+1} - x_i = -\frac{f(x_i)}{f'(x_i)}$$

97

Idea Goes Something Like This…

- Exploit 32 bit floating point as much as possible.
  - Especially for the bulk of the computation
- Correct or update the solution with selective use of 64 bit floating point to provide a refined results
- Intuitively:
  - Compute a 32 bit result,
  - Calculate a correction to 32 bit result using selected higher precision and,
  - Perform the update of the 32 bit results with the correction using high precision.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

```
L U = lu(A)                                   O(n^3)
x = L\(U\b)                                   O(n^2)
r = b – Ax                                    O(n^2)
WHILE || r || not small enough
        z = L\(U\r)                           O(n^2)
        x = x + z                             O(n^1)
        r = b – Ax                            O(n^2)
END
```

  - Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

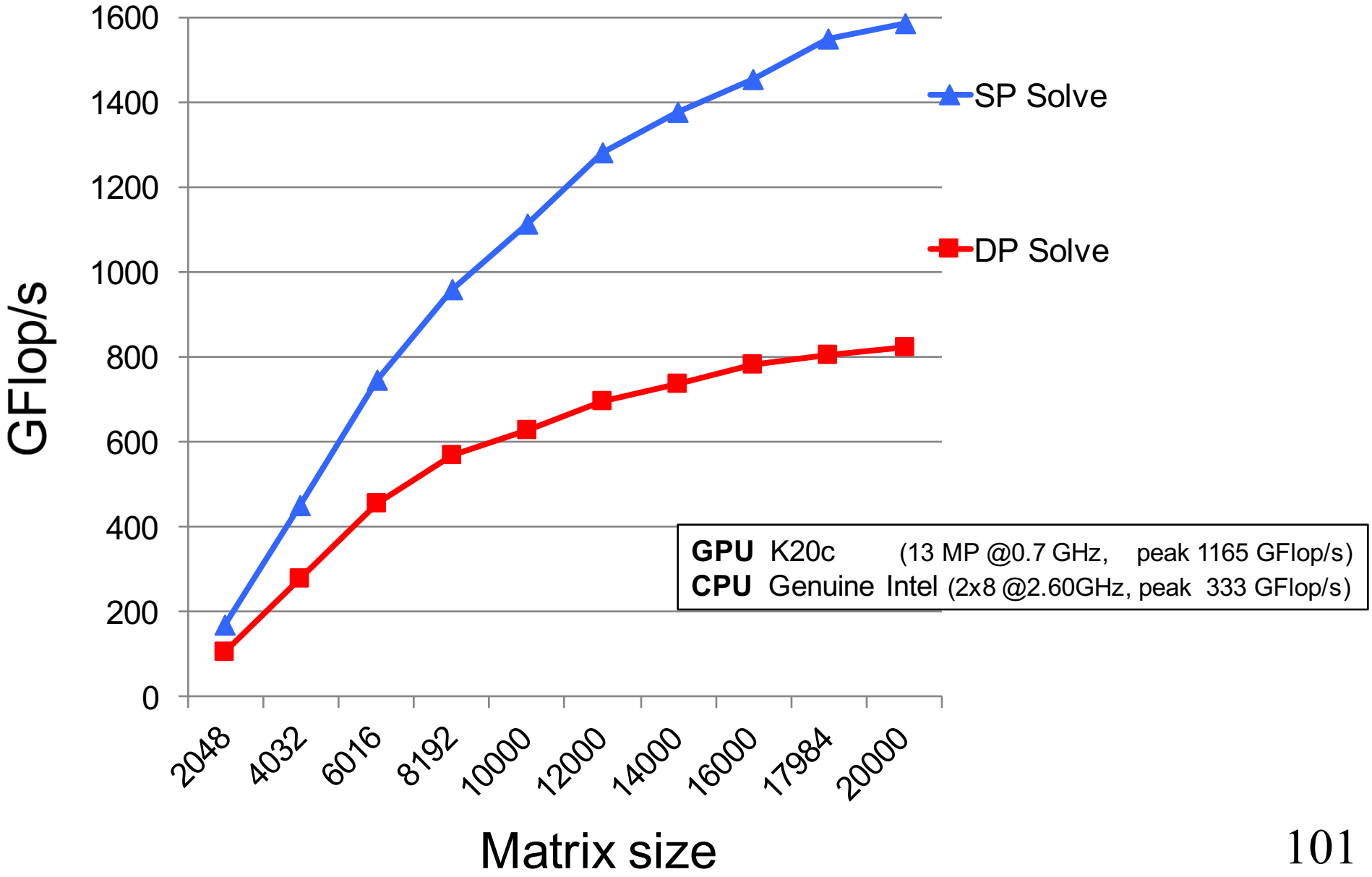
L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.

- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$

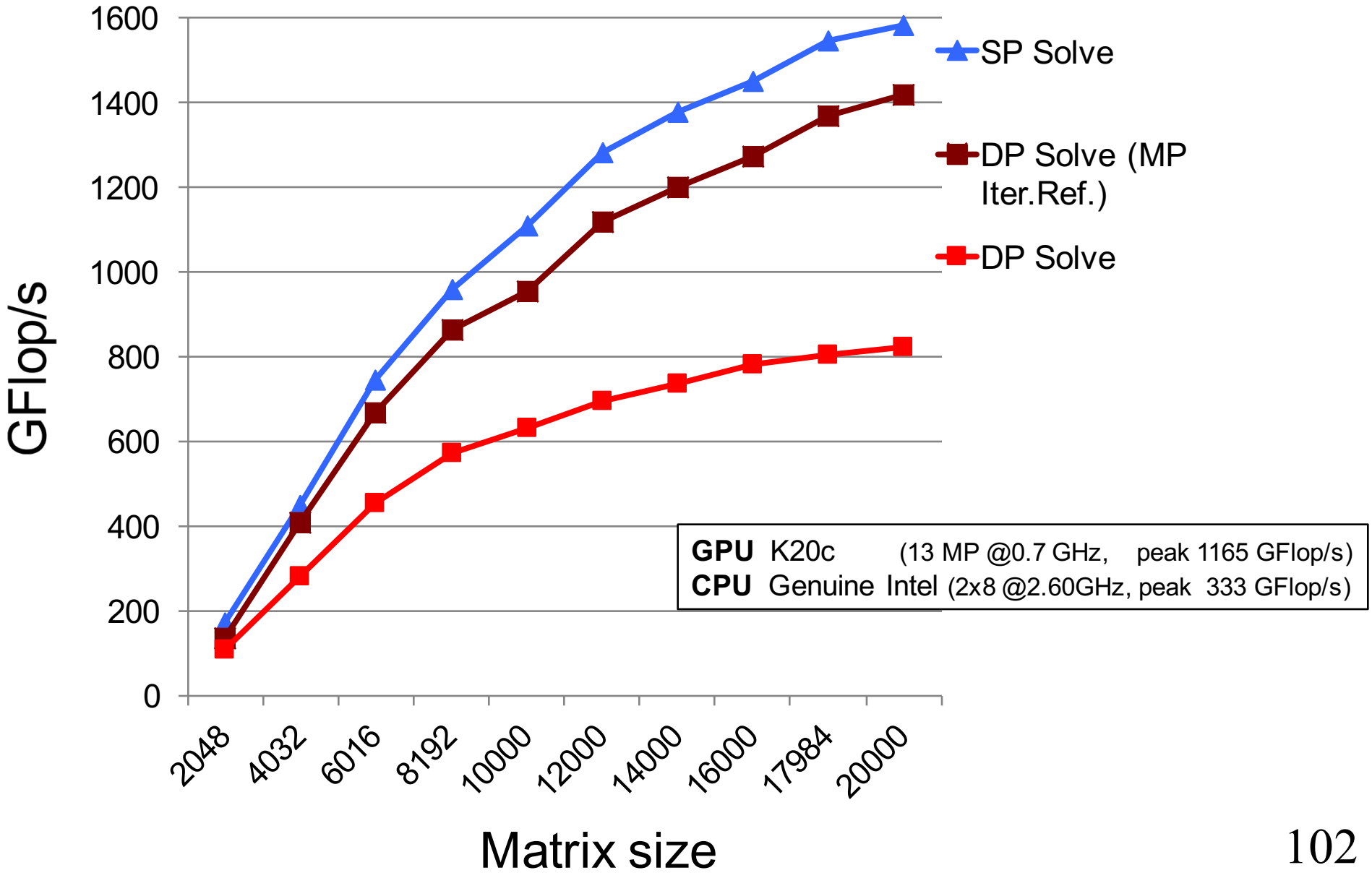
Mixed precision iterative refinement

Solving general dense linear systems using mixed precision iterative refinement



Mixed precision iterative refinement

Solving general dense linear systems using mixed precision iterative refinement



Conventional Wisdom is Changing

Old Conventional Wisdom

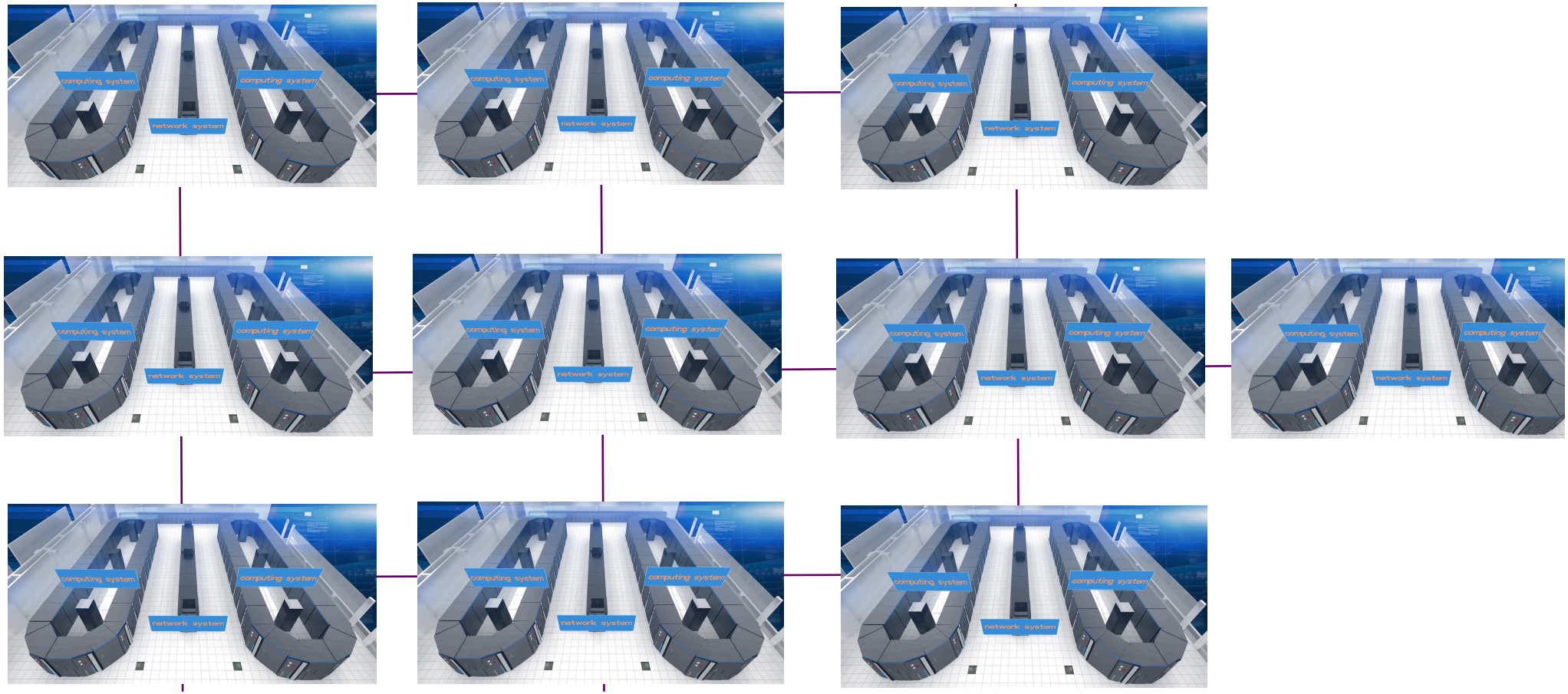
- **Peak clock frequency** as primary limiter for performance improvement
- **Cost:** FLOPs are biggest cost for system: optimize for compute
- **Concurrency:** Modest growth of parallelism by adding nodes
- **Memory scaling:** maintain byte per flop capacity and bandwidth
- **Uniformity:** Assume uniform system performance
- **Reliability:** It's the hardware's problem

New Conventional Wisdom

- **Power** is primary design constraint for future HPC system design
- **Cost:** Data movement dominates optimize to minimize data movement
- **Concurrency:** Exponential growth of parallelism within chips
- **Memory Scaling:** Compute growing 2x faster than capacity or bandwidth
- **Heterogeneity:** Architectural and performance non-uniformity increase
- **Reliability:** Cannot count on hardware protection alone

We Can Build an Exascale System Today?

Connect together 10 Sunway TaihuLight systems



Require 150 MW of power, programming for 100 M threads, and $2.7B price tag

Today's #1 System

Systems	2016 Sunway TaihuLight
System peak	**125.4 Pflop/s**
Power	**15 MW** **(8 Gflops/W)**
System memory	1.31 PB
Node performance	3.06 TF/s
Node concurrency	260 cores
Node Interconnect BW	16 GB/s
System size (nodes)	40,960
Total concurrency	10.6 M
MTTF	Few / day

Exascale System Architecture with a cap of $200M and 20MW

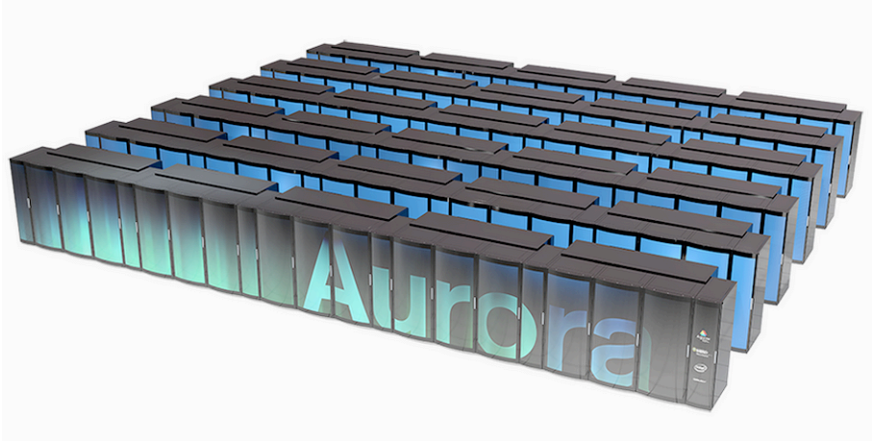
Systems	2016 Sunway TaihuLight
System peak	**125.4 Pflop/s**
Power	**15 MW** **(8 Gflops/W)**
System memory	1.31 PB
Node performance	3.06 TF/s
Node concurrency	260 cores
Node Interconnect BW	16 GB/s
System size (nodes)	40,960
Total concurrency	10.6 M
MTTF	Few / day

Exascale System Architecture with a cap of $200M and 20MW

Systems	2016 Sunway TaihuLight	2020 (may be 2023)	Difference Today & Exa
System peak	**125.4 Pflop/s**	**1 Eflop/s**	**~10x**
Power	**15 MW** **(8 Gflops/W)**	**~20 MW** **(50 Gflops/W)**	**O(1)** **~6x**
System memory	1.31 PB	32 - 64 PB	~50x
Node performance	3.06 TF/s	1.2 or 15TF/s	O(1)
Node concurrency	260 cores	O(1k) or 10k	~5x - ~50x
Node Interconnect BW	16 GB/s	200-400GB/s	~25x
System size (nodes)	40,960	O(100,000) or O(1M)	~6x - ~60x
Total concurrency	10.6 M	O(billion)	~100x
MTTF	Few / day	Many / day	O(?)

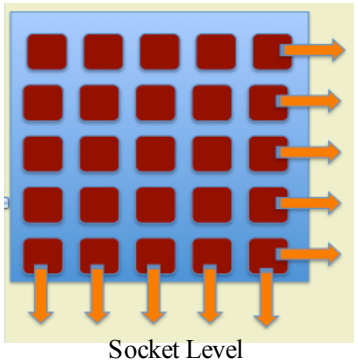
Recent Developments

- US DOE planning to deploy O(100) Pflop/s systems for 2017-2018 - $525M hardware
  - ORNL and LLNL to receive IBM and Nvidia based systems
  - ANL to receive Intel based system
  - After this Exaflops



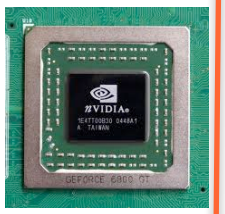
Exascale (10^{18} Flop/s) Systems: Two Possible Swim Lanes

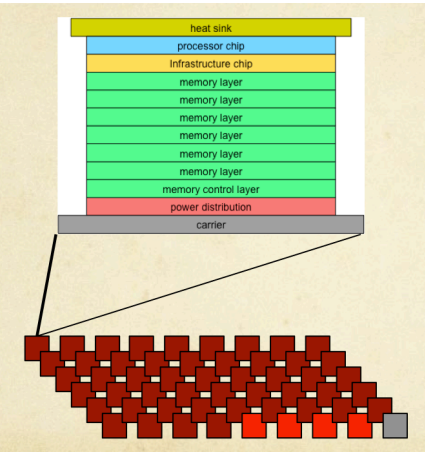
- **Light weight processors (eg ShenWei, ARM, Phi)**
  - ~1 GHz processor (10^9)
  - ~1 Kilo cores/socket (10^3)
  - ~1 Mega sockets/system (10^6)



Socket Level
Cores scale-out for planar geometry

- **Hybrid system (think Acc based)**
  - ~1 GHz processor (10^9)
  - ~10 Kilo FPUs/socket (10^4)
  - ~100 Kilo sockets/system (10^5)





Node Level
3D packaging

Software and Algorithm Must Keep Pace with the Changes in Hardware

- Classical analysis of algorithms is not valid,
  - # of floating point ops ≠ computation time.
- Algorithms and software must take advantage by reducing data movement.
  - Need latency tolerance in our algorithms
- Communication and synchronization reducing algorithms and software are critical.
  - As parallelism grows
- Hardware presents a dynamically changing environment
  - Turbo Boost and OS jitter
- Many existing algorithms can't fully exploit the features of modern architecture

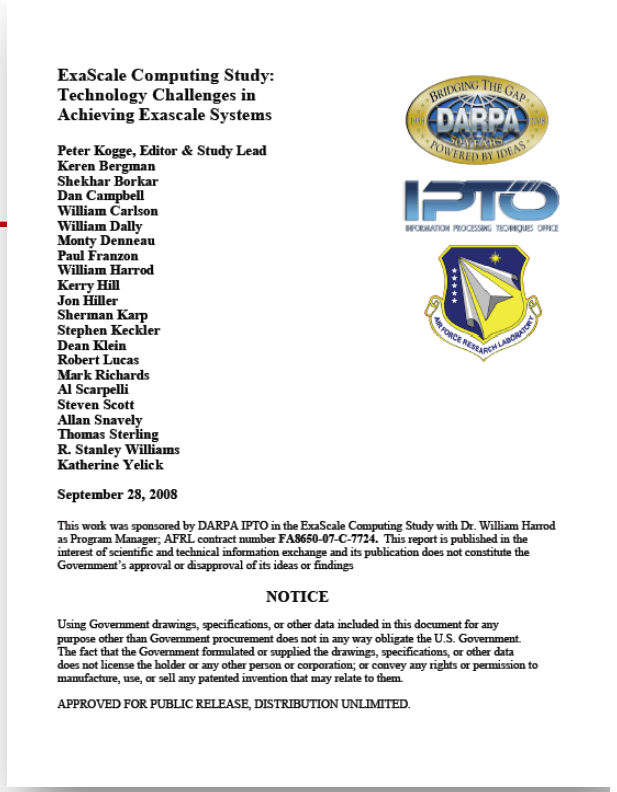
Major Changes to Software

- Must rethink the design of our software
  - Another disruptive technology
    - Similar to what happened with cluster computing and message passing
  - Rethink and rewrite the applications, algorithms, and software

Critical Issues at Peta & Exascale for Algorithm and Software Design

- Synchronization-reducing algorithms
  - Break Fork-Join model
- Communication-reducing algorithms
  - Use methods which have lower bound on communication
- Mixed precision methods
  - 2x speed of ops and 2x speed for data movement
- Autotuning
  - Today's machines are too complicated, build "smarts" into software to adapt to the hardware
- Fault resilient algorithms
  - Implement algorithms that can recover from failures
- Reproducibility of results
  - Today we can't guarantee this. We understand the issues, but some of our "colleagues" have a hard time with this.

Exascale Computing reported in 2008



ExaScale Computing Study:
Technology Challenges in
Achieving Exascale Systems

Peter Kogge, Editor & Study Lead
Keren Bergman
Shekhar Borkar
Dan Campbell
William Carlson
William Dally
Monty Denneau
Paul Franzon
William Harrod
Kerry Hill
Jon Hiller
Sherman Karp
Stephen Keckler
Dean Klein
Robert Lucas
Mark Richards
Al Scarpelli
Steven Scott
Allan Snavely
Thomas Sterling
R. Stanley Williams
Katherine Yelick

September 28, 2008

This work was sponsored by DARPA IPTO in the ExaScale Computing Study with Dr. William Harrod as Program Manager; AFRL contract number **FA8650-07-C-7724.** This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

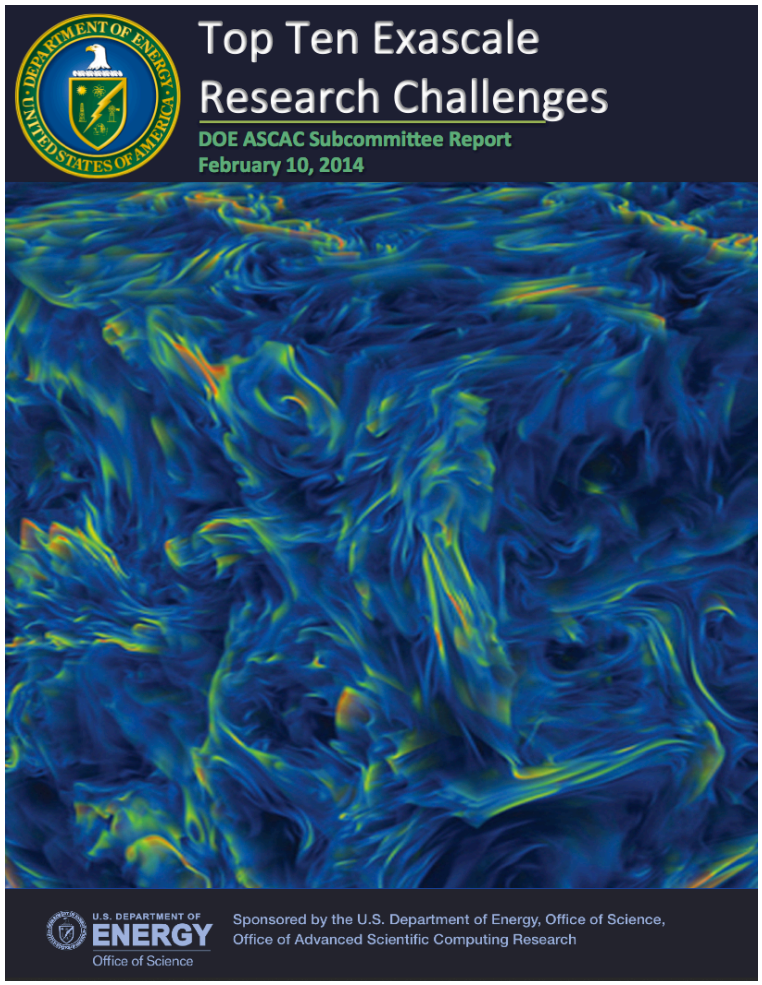
APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

- Exascale systems are likely feasible by 2017±2
- 10-100 Million processing elements (cores or mini-cores) with chips perhaps as dense as 1,000 cores per socket, clock rates will grow more slowly
- 3D packaging likely
- Large-scale optics based interconnects
- 10-100 PB of aggregate memory
- Hardware and software based fault management
- Heterogeneous cores
- Performance per watt — stretch goal 100 GF/watt of sustained performance $\Longrightarrow$ >> 10 - 100 MW Exascale system
- Power, area and capital costs will be significantly higher than for today's fastest systems

Google: exascale computing study

Top 10 Challenges to Exascale

In a recent report U.S. Department of Energy identified ten research challenges (Google "Top 10 Challenges to Exascale")



ASCAC Subcommittee for the Top Ten Exascale Research Challenges

Subcommittee Chair
Robert Lucas (University of Southern California, Information Sciences Institute)

Subcommittee Members
James Ang (Sandia National Laboratories)
Keren Bergman (Columbia University)
Shekhar Borkar (Intel)
William Carlson (Institute for Defense Analyses)
Laura Carrington (UC, San Diego)
George Chiu (IBM)
Robert Colwell (DARPA)
William Dally (NVIDIA)
Jack Dongarra (U. Tennessee)
Al Geist (ORNL)
Gary Grider (LANL)
Rud Haring (IBM)
Jeffrey Hittinger (LLNL)
Adolfy Hoisie (PNLL)
Dean Klein (Micron)
Peter Kogge (U. Notre Dame)
Richard Lethin (Reservoir Labs)
Vivek Sarkar (Rice U.)
Robert Schreiber (Hewlett Packard)
John Shalf (LBNL)
Thomas Sterling (Indiana U.)
Rick Stevens (ANL)

Top 10 Challenges to Exascale

3 Hardware, 4 Software, 3 Algorithms/Math Related

- **Energy efficiency:**
  - Creating more energy efficient circuit, power, and cooling technologies.
- **Interconnect technology:**
  - Increasing the performance and energy efficiency of data movement.
- **Memory Technology:**
  - Integrating advanced memory technologies to improve both capacity and bandwidth.
- **Scalable System Software:**
  - Developing scalable system software that is power and resilience aware.
- **Programming systems:**
  - Inventing new programming environments that express massive parallelism, data locality, and resilience
- **Data management:**
  - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
- **Scientific productivity:**
  - Increasing the productivity of computational scientists with new software engineering tools and environments.
- **Exascale Algorithms:**
  - Reformulating science problems and refactoring their solution algorithms for exascale systems.
- **Algorithms for discovery, design, and decision:**
  - Facilitating mathematical optimization and uncertainty quantification for exascale discovery, design, and decision making.
- **Resilience and correctness:**
  - Ensuring correct scientific computation in face of faults, reproducibility, and algorithm verification challenges.

Conclusions

- For the last decade or more, the research investment strategy has been overwhelmingly biased in favor of hardware.
- This strategy needs to be rebalanced - barriers to progress are increasingly on the software side.
- Moreover, the return on investment is more favorable to software.
  - Hardware has a half-life measured in years, while software has a half-life measured in decades.
- High Performance Ecosystem out of balance
  - Hardware, OS, Compilers, Software, Algorithms, Applications
    - No Moore's Law for software, algorithms and applications

By the way
Performance for your system

- If you are interested in running the Linpack Benchmark on your system see:
  https://software.intel.com/en-us/node/157667?wapkw=mkl+linpack

- http://bit.ly/linpack-bm

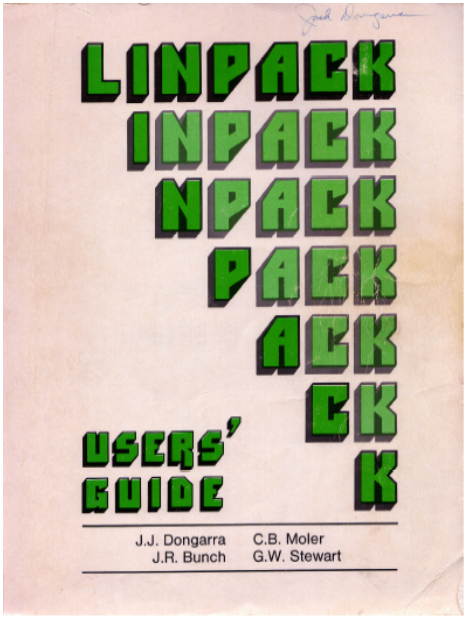
  ./linpack_cd64 < lininput

- Also Intel has a power meter, see:
  https://software.intel.com/en-us/articles/intel-power-gadget-20

  http://bit.ly/intel-power

Confessions of an Accidental Benchmarker

- Appendix B of the Linpack Users' Guide
  - Designed to help users extrapolate execution Linpack software package
- First benchmark report from 1977;
  - Cray 1 to DEC PDP-10



Started 37 Years Ago

Have seen a Factor of 10^9 - From 14 Mflop/s to 34 Pflop/s

- In the late 70's the fastest computer ran LINPACK at 14 Mflop/s
- Today with HPL we are at 34 Pflop/s
  - Nine orders of magnitude
  - doubling every 14 months
  - About 6 orders of magnitude increase in the number of processors
  - Plus algorithmic improvements

UNIT = 10**6 TIME/(1/3 100**3 + 100**2)

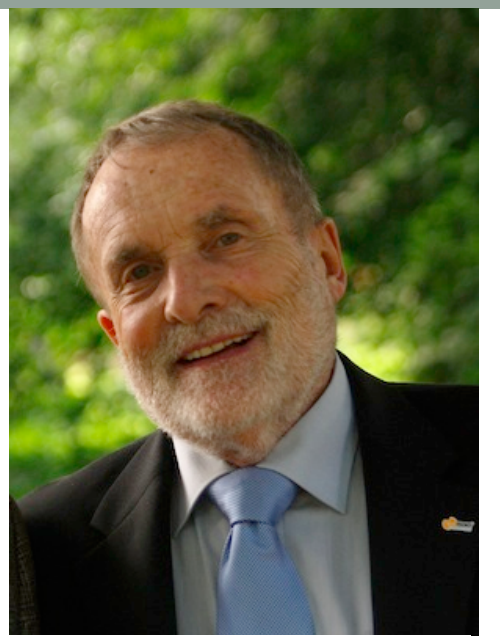
Facility	TIME N=100 secs.	UNIT micro-secs.	Computer	Type	Compiler
NCAR	.049	0.14	CRAY-1	S	CFT, Assembly BLAS
LASL	.148	0.43	CDC 7600	S	FTN, Assembly BLAS
NCAR	.192	0.56	CRAY-1	S	CFT
LASL	.210	0.61	CDC 7600	S	FTN
Argonne	.297	0.86	IBM 370/195	D	H
NCAR	.359	1.05	CDC 7600	S	Local
Argonne	.388	1.33	IBM 3033	D	H
NASA Langley	.489	1.42	CDC Cyber 175	S	FTN
U. Ill. Urbana	.506	1.47	CDC Cyber 175	S	Ext. 4.6
LLL	.554	1.61	CDC 7600	S	CHAT, No optimize
SLAC	.579	1.69	IBM 370/168	D	H Ext., Fast mult.
Michigan	.631	1.84	Amdahl 470/V6	D	H
Toronto	.890	2.59	IBM 370/165	D	H Ext., Fast mult.
Northwestern	1.44	4.20	CDC 6600	S	FTN
Texas	1.93*	5.63	CDC 6600	S	RUN
China Lake	1.95*	5.69	Univac 1110	S	V
Yale	2.59	7.53	DEC KL-20	S	F20
Bell Labs	3.46	10.1	Honeywell 6080	S	Y
Wisconsin	3.49	10.1	Univac 1110	S	V
Iowa State	3.54	10.2	Itel AS/5 mod3	D	H
U. Ill. Chicago	4.10	11.9	IBM 370/158	D	G1
Purdue	5.69	16.6	CDC 6500	S	FUN
U. C. San Diego	13.1	38.2	Burroughs 6700	S	H
Yale	17.1*	49.9	DEC KA-10	S	F40

* TIME(100) = (100/75)**3 SGEFA(75) + (100/75)**2 SGESL(75)

Began in late 70's

time when floating point operations were expensive compared to other operations and data movement

TOP500



- In 1986 Hans Meuer started a list of supercomputer around the world, they were ranked by peak performance.
- Hans approached me in 1992 to put together our lists into the “TOP500”.
- The first TOP500 list was in June 1993.

TOP500® SUPERCOMPUTER SITES

Rank	Site	System	Cores	Rmax (GFlop/s)	Rpeak (GFlop/s)	Power (kW)
1	Los Alamos National Laboratory United States	CM-5/1024 Thinking Machines Corporation	1,024	59.7	131.0	
2	Minnesota Supercomputer Center United States	CM-5/544 Thinking Machines Corporation	544	30.4	69.6	
3	National Security Agency United States	CM-5/512 Thinking Machines Corporation	512	30.4	65.5	
4	NCSA United States	CM-5/512 Thinking Machines Corporation	512	30.4	65.5	
5	NEC Japan	SX-3/44R NEC	4	23.2	25.6	
6	Atmospheric Environment Service (AES)	SX-3/44	4	20.0	22.0	

High Performance Linpack (HPL)

- Is a **widely recognized** and discussed metric for ranking high performance computing systems
- When HPL gained prominence as a performance metric in the early 1990s there **was a strong correlation between its predictions of system rankings and the ranking that full-scale applications would realize**.
- **Computer system vendors pursued designs that would increase their HPL performance**, which would in turn improve overall application performance.
- Today HPL remains **valuable as a measure of historical trends**, and as a stress test, especially for leadership class systems that are pushing the boundaries of current technology.

The Problem

- HPL performance of computer systems are **no longer so strongly correlated to real application performance**, especially for the broad set of HPC applications governed by partial differential equations.

- **Designing a system for good HPL performance can actually lead to design choices that are wrong** for the real application mix, or add unnecessary components or complexity to the system.

Concerns

- The **gap between HPL predictions and real application performance will increase** in the future.
- A computer system with the potential to run **HPL at 1 Exaflops is a design that may be very unattractive for real applications.**
- Future **architectures targeted toward good HPL performance will not be a good match for most applications**.
- This leads us to a think about a different metric

HPL - Good Things

- Easy to run
- Easy to understand
- Easy to check results
- Stresses certain parts of the system
- Historical database of performance information
- Good community outreach tool
- “Understandable” to the outside world

- If your computer doesn’t perform well on the LINPACK Benchmark, you will probably be disappointed with the performance of your application on the computer.

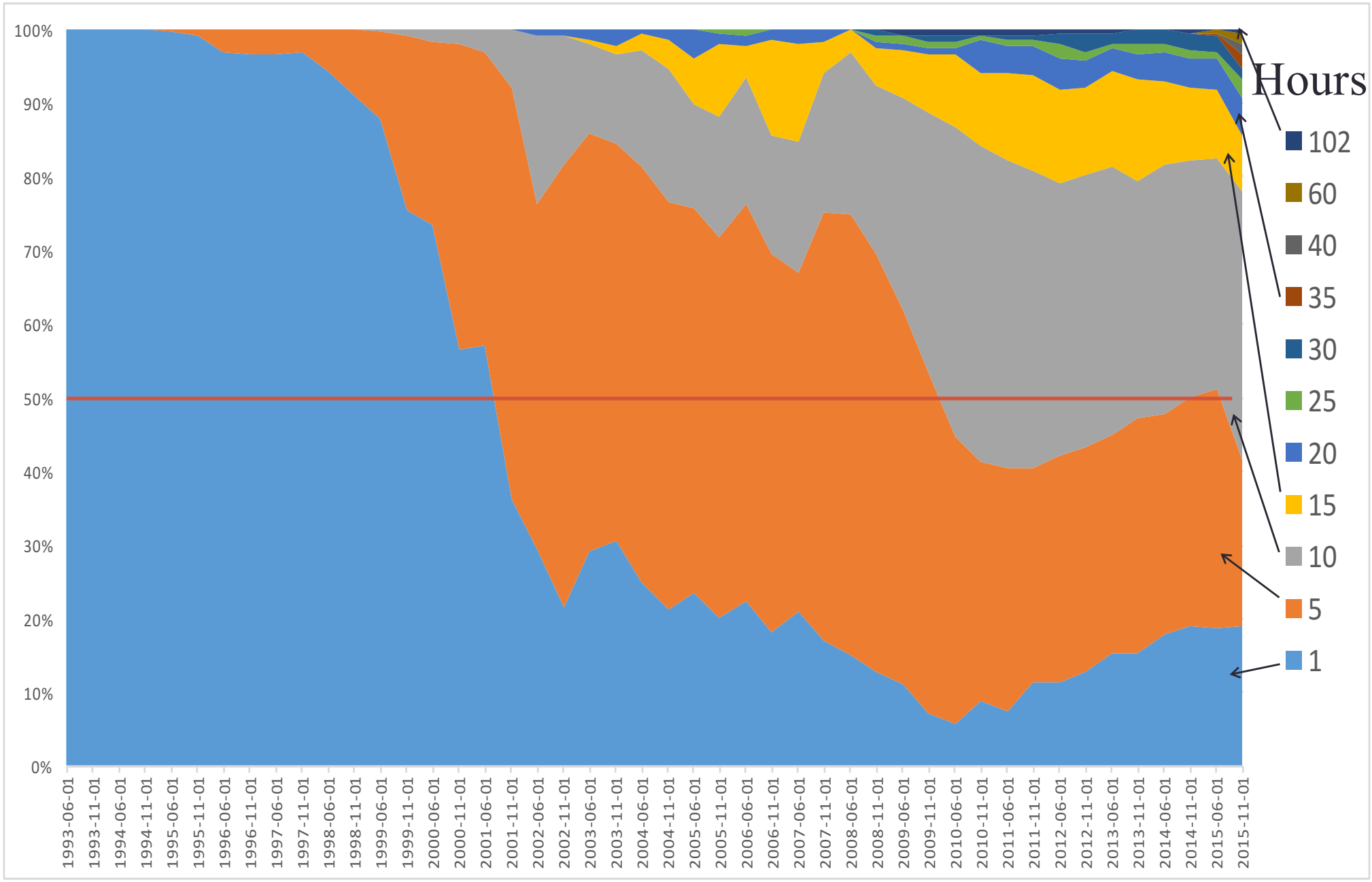
HPL - Bad Things

- LINPACK Benchmark is 36 years old
  - Top500 (HPL) is 20.5 years old
- Floating point-intensive performs $O(n^3)$ floating point operations and moves $O(n^2)$ data.
- No longer so strongly correlated to real apps.
- Reports Peak Flops (although hybrid systems see only 1/2 to 2/3 of Peak)
- Encourages poor choices in architectural features
- Overall usability of a system is not measured
- Used as a marketing tool
- Decisions on acquisition made on one number
- Benchmarking for days wastes a valuable resource

Running HPL

- In the beginning to run HPL on the number 1 system was under an hour.
- On Livermore's Sequoia IBM BG/Q the HPL run took about a day to run.
  - They ran a size of n=12.7 x 10^6 (1.28 PB)
  - 16.3 PFlop/s requires about 23 hours to run!!
  - 23 hours at 7.8 MW that the equivalent of 100 barrels of oil or about $8600 for that one run.
- The longest run was 60.5 hours
  - JAXA machine
    - Fujitsu FX1, Quadcore SPARC64 VII 2.52 GHz
  - A matrix of size n = 3.3 x 10^6
  - .11 Pflop/s #160 today

Run Times for HPL on Top500 Systems



#1 System on the Top500 Over the Past 24 Years (18 machines in that club)

9  6  3 

Top500 List	Computer	r_max (Tflop/s)	n_max	Hours	MW
6/93 (1)	TMC CM-5/1024	.060	52224	0.4	
11/93 (1)	Fujitsu Numerical Wind Tunnel	.124	31920	0.1	1.
6/94 (1)	Intel XP/S140	.143	55700	0.2	
11/94 - 11/95 (3)	Fujitsu Numerical Wind Tunnel	.170	42000	0.1	1.
6/96 (1)	Hitachi SR2201/1024	.220	138,240	2.2	
11/96 (1)	Hitachi CP-PACS/2048	.368	103,680	0.6	
6/97 - 6/00 (7)	Intel ASCI Red	2.38	362,880	3.7	.85
11/00 - 11/01 (3)	IBM ASCI White, SP Power3 375 MHz	7.23	518,096	3.6	
6/02 - 6/04 (5)	NEC Earth-Simulator	35.9	1,000,000	5.2	6.4
11/04 - 11/07 (7)	IBM BlueGene/L	478.	1,000,000	0.4	1.4
6/08 - 6/09 (3)	IBM Roadrunner -PowerXCell 8i 3.2 Ghz	1,105.	2,329,599	2.1	2.3
11/09 - 6/10 (2)	Cray Jaguar - XT5-HE 2.6 GHz	1,759.	5,474,272	17.3	6.9
11/10 (1)	NUDT Tianhe-1A, X5670 2.93Ghz NVIDIA	2,566.	3,600,000	3.4	4.0
6/11 - 11/11 (2)	Fujitsu K computer, SPARC64 VIIIfx	10,510.	11,870,208	29.5	9.9
6/12 (1)	IBM Sequoia BlueGene/Q	16,324.	12,681,215	23.1	7.9
11/12 (1)	Cray XK7 Titan AMD + NVIDIA Kepler	17,590.	4,423,680	0.9	8.2
6/13 - 11/15(6)	NUDT Tianhe-2 Intel IvyBridge & Xeon Phi	33,862.	9,960,000	5.4	17.8
6/16 -	**Sunway TaihuLight System**	93,014.	12,288,000	3.7	15.4

Assumptions

Sandia National Laboratories

- Leadership class system:
  - Cost: $200M
  - Lifetime: 4 years
  - Power consumption: 10MW
- Cost of one MW-year is $1M
- Linpack measurement requires system for a week
  - To achieve a high fraction of peak requires a large problem size so a typical MP Linpack run takes a day
    - Multiple runs are made as initial tests are run with "small" problems
    - Successive tests use larger and larger problem sizes, some of these tests will "fail" – requiring re-runs

From: Jim Ang, SNL; What's the True Cost of LINPACK, Salishan 2013

Ugly Things about HPL

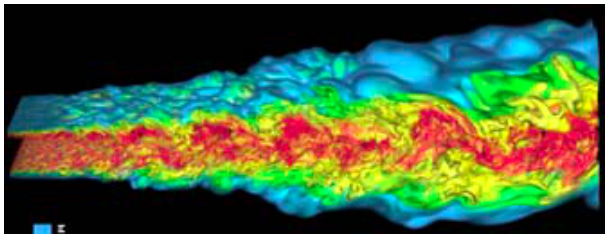
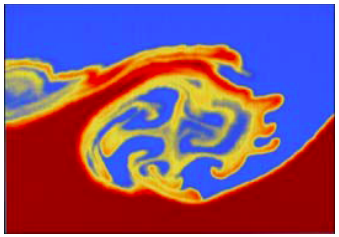
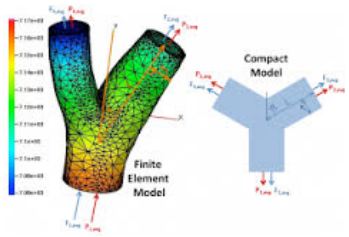
- Doesn’t probe the architecture; only one data point
- Constrains the technology and architecture options for HPC system designers.
  - Skews system design.
- Floating point benchmarks are not quite as valuable to some as data-intensive system measurements

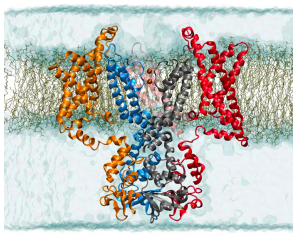
Many Other Benchmarks

- Top 500
- Green 500
- Graph ~~500~~ 174
- Sustained Petascale Performance
- HPC Challenge
- Perfect
- ParkBench
- SPEC-hpc
- Livermore Loops
- EuroBen
- NAS Parallel Benchmarks
- Genesis
- RAPS
- SHOC
- LAMMPS
- Dhrystone
- Whetstone

Goals for New Benchmark

- Augment the TOP500 listing with a benchmark that correlates with important scientific and technical apps not well represented by HPL



- Encourage vendors to focus on architecture features needed for high performance on those important scientific and technical apps.
  - Stress a balance of floating point and communication bandwidth and latency
  - Reward investment in high performance collective ops
  - Reward investment in high performance point-to-point messages of various sizes
  - Reward investment in local memory system performance
  - Reward investment in parallel runtimes that facilitate intra-node parallelism
- Provide an outreach/communication tool
  - Easy to understand
  - Easy to optimize
  - Easy to implement, run, and check results
- Provide a historical database of performance information
  - The new benchmark should have longevity

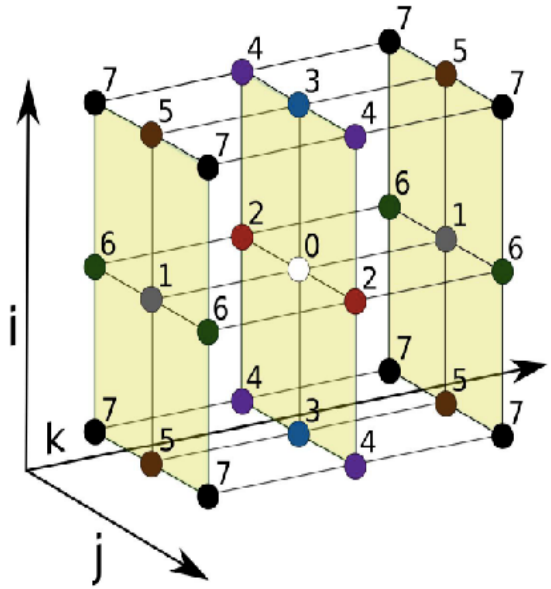
Proposal: HPCG

- High Performance Conjugate Gradient (HPCG).
- Solves $Ax=b$, A large, sparse, b known, x computed.
- An optimized implementation of PCG contains essential computational and communication patterns that are prevalent in a variety of methods for discretization and numerical solution of PDEs

- Patterns:
  - Dense and sparse computations.
  - Dense and sparse collective.
  - Multi-scale execution of kernels via MG (truncated) V cycle.
  - Data-driven parallelism (unstructured sparse triangular solves).
- Strong verification and validation properties (via spectral properties of PCG).

Model Problem Description

- Synthetic discretized 3D PDE (FEM, FVM, FDM).
- Single DOF heat diffusion model.
- Zero Dirichlet BCs, Synthetic RHS s.t. solution = 1.
- Local domain: $(n_x \times n_y \times n_z)$
- Process layout: $(np_x \times np_y \times np_z)$
- Global domain: $(n_x * np_x) \times (n_y * np_y) \times (n_z * np_z)$
- Sparse matrix:
  - 27 nonzeros/row interior.
  - 8 – 18 on boundary.
  - Symmetric positive definite.



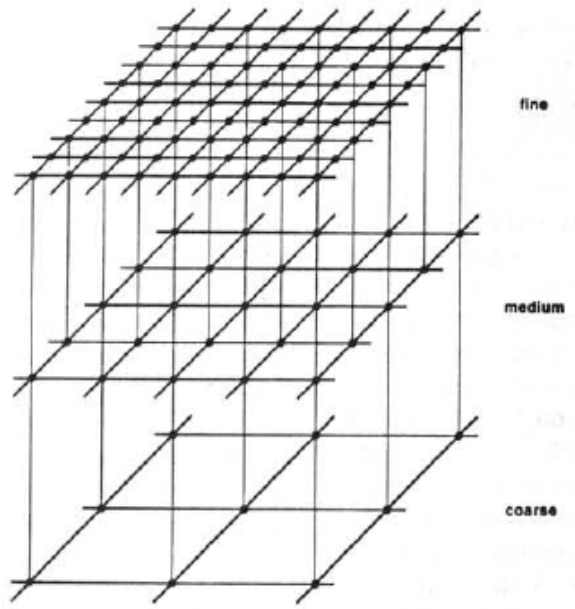
27-point stencil operator

PCG ALGORITHM

- $p_0 := x_0, r_0 := b\text{-}Ap_0$
- Loop $i = 1, 2, \ldots$
  - $z_i := M^{-1} r_{i-1}$
  - if $i = 1$
    - $p_i := z_i$
    - $\alpha_i :=$ `dot_product`(r_{i-1}, z)
  - else
    - $\alpha_i :=$ `dot_product`(r_{i-1}, z)
    - $\beta_i := \alpha_i / \alpha_{i-1}$
    - $p_i := \beta_i * p_{i-1} + z_i$
  - end if
  - $\alpha_i :=$ `dot_product`(r_{i-1}, z_i) /`dot_product`$(p_i, A*p_i)$
  - $x_{i+1} := x_i + \alpha_i * p_i$
  - $r_i := r_{i-1} - \alpha_i * A * p_i$
  - if $\|r_i\|_2 <$ tolerance then Stop
- end Loop

Preconditioner

- Hybrid geometric/algebraic multigrid:
  - Grid operators generated synthetically:
    - Coarsen by 2 in each x, y, z dimension (total of 8 reduction each level).
    - Use same GenerateProblem() function for all levels.
  - Grid transfer operators:
    - Simple injection. Crude but…
    - Requires no new functions, no repeat use of other functions.
    - Cheap.
  - Smoother:
    - Symmetric Gauss-Seidel [ComputeSymGS()].
    - Except, perform halo exchange prior to sweeps.
    - Number of pre/post sweeps is tuning parameter.
  - Bottom solve:
    - Right now just a single call to ComputeSymGS().
    - If no coarse grids, has identical behavior as HPCG 1.X.



- Symmetric Gauss-Seidel preconditioner
  - In Matlab that might look like:

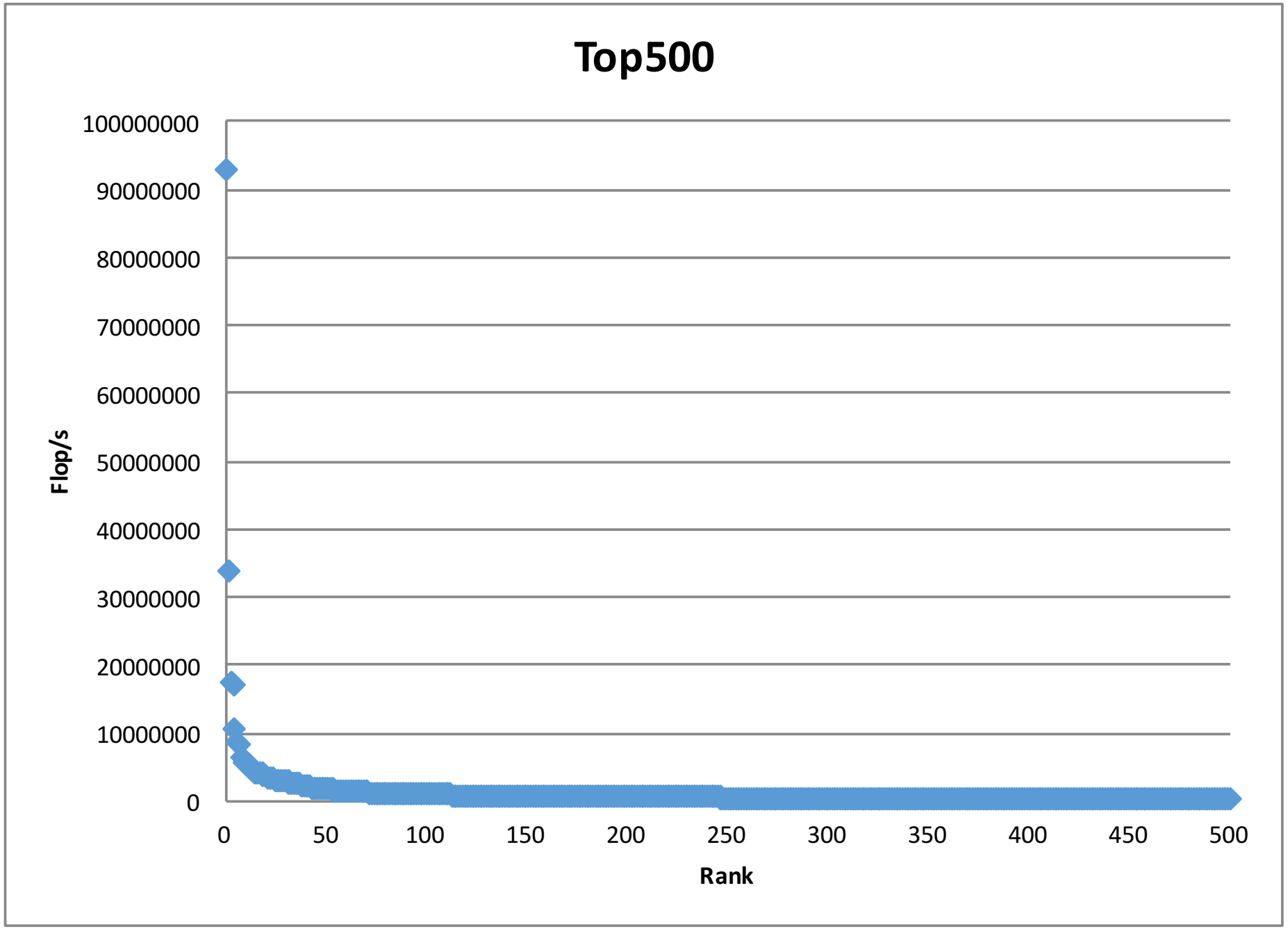
```
LA = tril(A); UA = triu(A); DA = diag(diag(A));

x = LA\y;
x1 = y - LA*x + DA*x; % Subtract off extra
     diagonal contribution
x = UA\x1;
```

HPCG and HPL

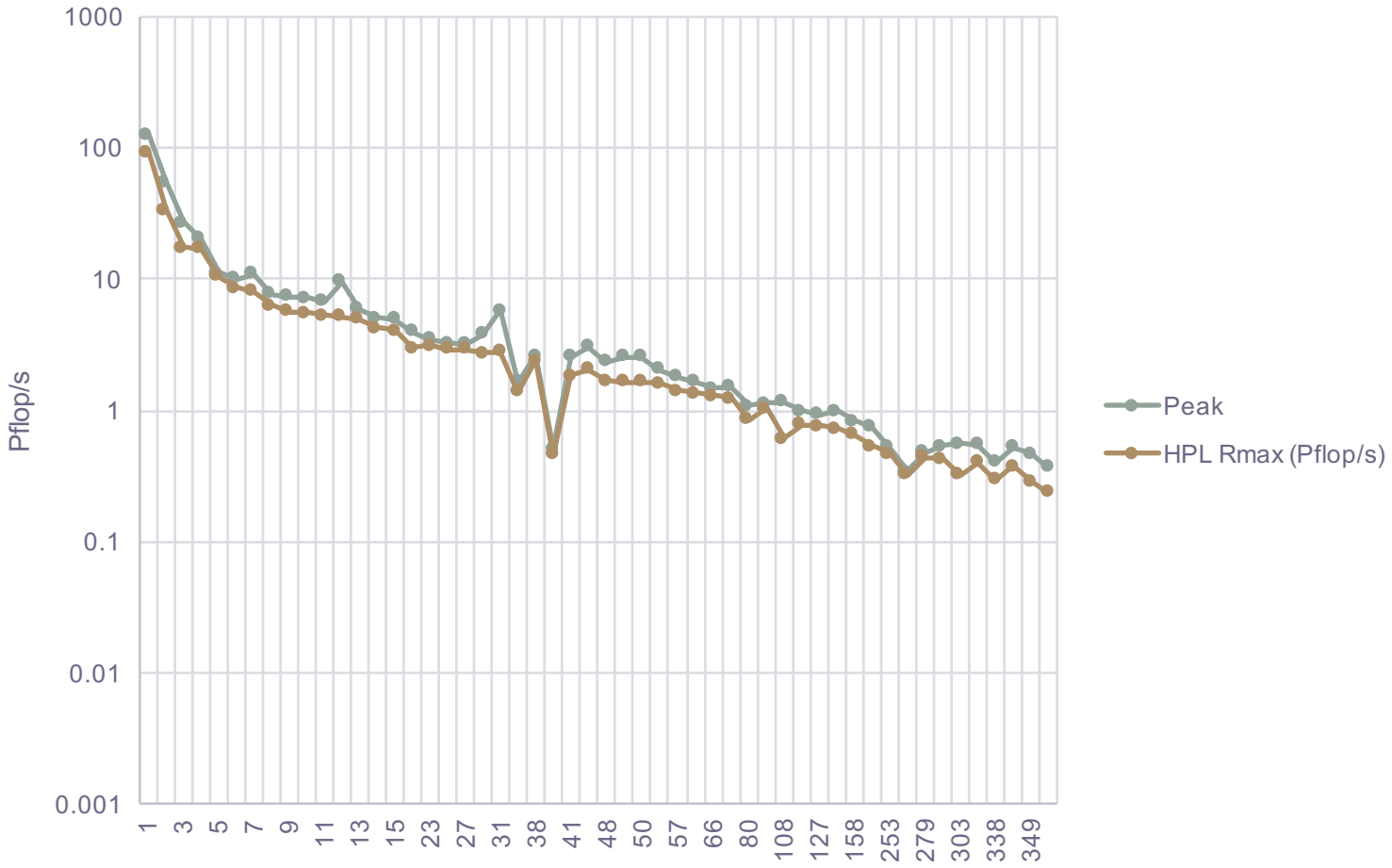
- We are NOT proposing to eliminate HPL as a metric.
- The historical importance and community outreach value is too important to abandon.
- HPCG will serve as an alternate ranking of the Top500.
  - Or maybe top 50 (have 15 systems at the moment).

HPL vs. HPCG: Bookends

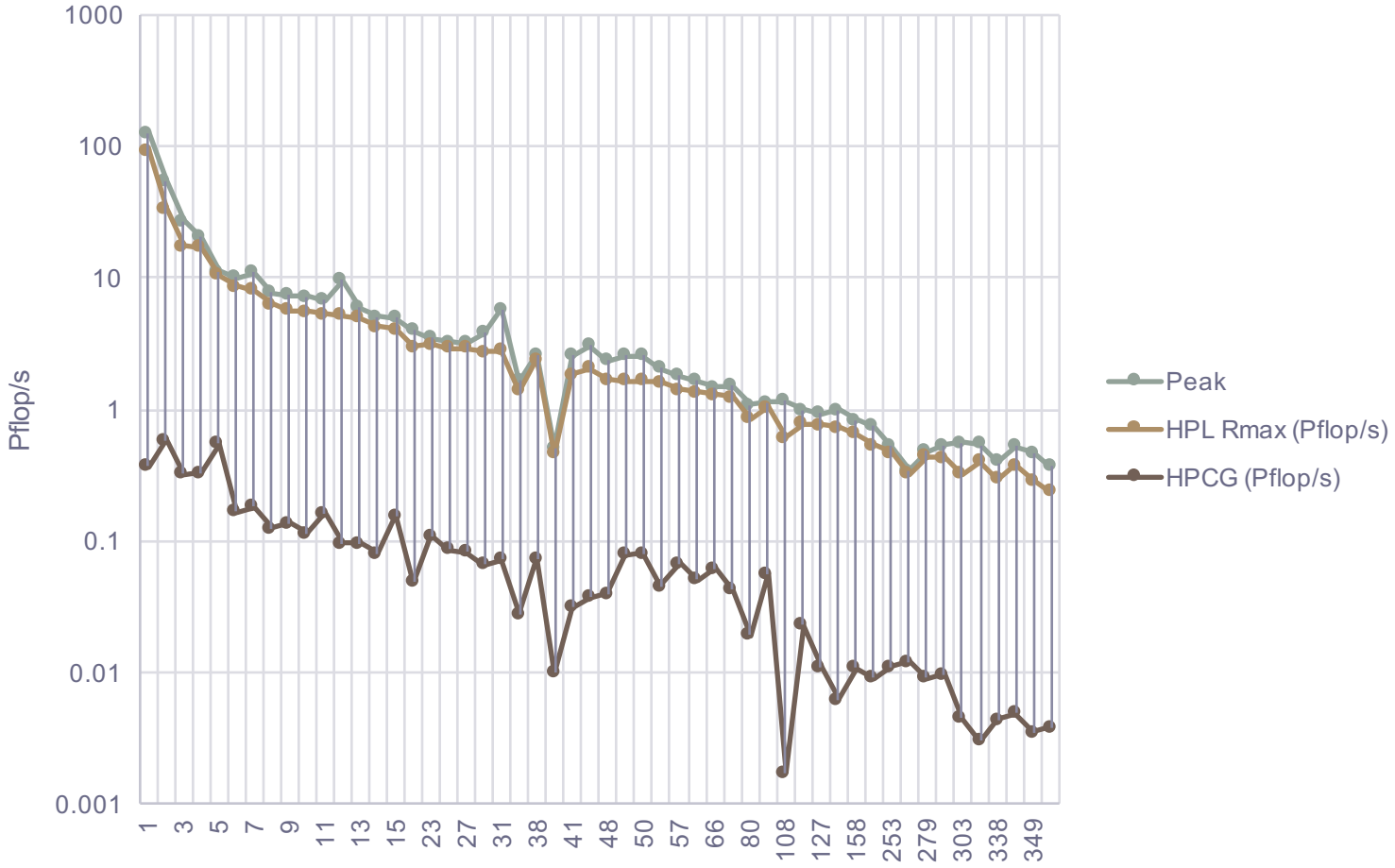
- Some see HPL and HPCG as “bookends” of a spectrum.
  - Applications teams know where their codes lie on the spectrum.
  - Can gauge performance on a system using both HPL and HPCG numbers.
- Problem of HPL execution time still an issue:
  - Need a lower cost option. End-to-end HPL runs are too expensive.
  - Work in progress.


Top500
100000000
90000000
80000000
70000000
60000000
50000000
40000000
30000000
20000000
10000000
0
Flop/s
0
50
100
150
200
250
300
350
400
450
500
Rank

Bookends: Peak, HPL, and HPCG



Bookends: Peak, HPL, and HPCG



1-10

Rank (HPL)	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
1 (2)	NSCC / Guangzhou	Tianhe-2 NUDT, Xeon 12C 2.2GHz + Intel Xeon Phi 57C + Custom	3,120,000	33.863	0.5800	1.7%	1.1%
2 (5)	RIKEN Advanced Institute for Computational Science	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect	705,024	10.510	0.5544	5.3%	4.9%
3 (1)	National Supercomputing Center in Wuxi	Sunway TaihuLight -- SW26010, Sunway	10,649,600	93.015	0.3712	0.4%	0.3%
4 (4)	DOE/NNSA/LLNL	Sequoia - IBM BlueGene/Q	1,572,864	17.173	0.3304	1.9%	1.6%
5 (3)	DOE/SC/Oak Ridge Nat Lab	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x	560,640	17.590	0.3223	1.8%	1.2%
6 (7)	DOE/NNSA/LANL/SNL	Trinity - Cray XC40, Intel E5-2698v3, Aries custom	301,056	8.101	0.1826	2.3%	1.6%
7 (6)	DOE/SC/Argonne National Laboratory	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom	786,432	8.587	0.1670	1.9%	1.7%
8 (11)	TOTAL	Pangea -- Intel Xeon E5-2670, Infiniband FDR	218592	5.283	0.1627	3.1%	2.4%
9 (15)	NASA / Mountain View	Pleiades - SGI ICE X, Intel E5-2680, E5-2680V2, E5-2680V3, Infiniband FDR	185,344	4.089	0.1555	3.8%	3.1%
10 (9)	HLRS/University of Stuttgart	Hazel Hen - Cray XC40, Intel E5-2680v3, Cray Aries	185,088	5.640	0.1380	2.4%	1.9%

11-20

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
11	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x	115,984	6.271	0.1246	2.0%	1.6%
12	KAUST / Jeddah	Shaheen II - Cray XC40, Intel Haswell 2.3 GHz 16C, Cray Aries	196,608	5.537	0.1139	2.1%	1.6%
13	Japan Aerospace eXploration Agency	SORA-MA -- SPARC64 XIfx	103,680	3.157	0.1102	3.5%	3.2%
14	Texas Advanced Computing Center/Univ. of Texas	Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P	522,080	5.168	0.0968	1.9%	1.0%
15	Forschungszentrum Jülich	JUQUEEN - BlueGene/Q	458,752	5.009	0.0955	1.9%	1.6%
16	Information Technology Center, Nagoya University	ITC, Nagoya - Fujitsu PRIMEHPC FX100, SPARC64 Xifx, Tofu interconnect 2	92,160	2.910	0.0865	3.0%	2.7%
17	Leibniz Rechenzentrum	SuperMUC - iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband FDR	147,456	2.897	0.0833	2.9%	2.6%
18	DOE/NNSA/LLNL	Vulcan - IBM BlueGene/Q	393,216	4.293	0.0809	1.9%	1.6%
19	EPSRC/University of Edinburgh	ARCHER - Cray XC30, Intel Xeon E5 v2 12C 2.700GHz, Aries interconnect	118,080	1.643	0.0808	4.9%	3.2%
20	DOE/SC/LBNL/NERSC	Edison - Cray XC30, Intel Xeon E5-2695v2 12C 2.4GHz, Aries interconnect	133,824	1.655	0.0786	4.8%	3.1%

21-30

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
21	National Institute for Fusion Science	Plasma Simulator - Fujitsu PRIMEHPC FX100, SPARC64 Xifx, Tofu Interconnect 2	82,944	2.376	0.0732	3.1%	2.8%
22	GSIC Center, Tokyo Institute of Technology	TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.93GHz, Infiniband QDR, NVIDIA K20x	76,032	2.785	0.0725	2.6%	1.3%
23	Forschungszentrum Jülich	JURECA - T-Platform V-Class Cluster, Xeon E5-2680v3 12C 2.5GHz, Infiniband EDR, NVIDIA Tesla K80/K40	49,476	1.425	0.0683	4.8%	3.8%
24	HLRS/Universitaet Stuttgart	Hornet - Cray XC40, Xeon E5-2680 v3 2.5 GHz, Cray Aries	94,656	2.763	0.0661	2.4%	1.7%
25	Max-Planck-Gesellschaft MPI/IPP	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband FDR	65,320	1.283	0.0615	4.8%	4.2%
26	CEIST / JAMSTEC	Earth Simulator - NEC SX-ACE	8,192	0.487	0.0578	11.9%	11.0%
27	Information Technology Center, The University of Tokyo	Oakleaf-FX -- SPARC64 Ixfx	76,800	1.043	0.0565	5.4%	5.0%
28	CEIST / JAMSTEC	Earth Simulator -- NEC SX-ACE	8,192	0.487	0.0547	11.2%	10.4%
29	CEA/TGCC-GENCI	Curie thin nodes - Bullx B510, Xeon E5-2680 8C 2.700GHz, Infiniband QDR	77,184	1.359	0.0510	3.8%	3.1%
30	Exploration & Production - Eni S.p.A.	HPC2 - iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, NVIDIA K20x	62,640	3.003	0.0489	1.6%	1.2%

31-40

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
31	Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Superieur (GENCI-CINES)	Occigen Bullx B720, Xeon E5-2690v3 12C 2.600GHz, InfiniBand FDR	50,544	1.629	0.0455	2.8%	2.2%
32	International Fusion Energy Research Centre (IFERC), EU(F4E) - Japan Broader Approach collaboration	Helios Bullx B510, Xeon E5-2680 8C 2.700GHz, Infiniband QDR	70,560	1.237	0.0426	3.4%	2.8%
33	Cyfronet	Prometheus - HP ProLiant Intel E5-2680v3, Infiniband FDR	55,728	1.670	0.0399	2.4%	1.7%
34	Lvliang/National University of Defense Technology	Tianhe-2 Lvliang - Intel Xeon E5-2692v2 12C, TH Express-2, Intel Xeon Phi 31S1P	174,720	2.071	0.0376	1.8%	1.2%
35	Moscow State University / Research Computing Center	Lomonosov 2 - Intel Xeon E5-2680V2, Infiniband FDR, NVIDIA K40	37,120	1.849	0.0315	1.7%	1.2%
36	DKRZ - Deutsches Klimarechenzentrum	Mistral -- Intel Xeon E5-2695v4, Infiniband FDR	19,200	1.371	0.0283	2.1%	1.7%
37	Cyberscience Center, Tohoku University	Cyberscience Center, Tohoku University -- NEC SX-ACE	4,096	0.246	0.0279	11.3%	10.7%
38	Stanford University / Palo Alto	Xstream - Dual Intel E5-2680V2, 8-way NVIDIA K80, Infiniband FDR	237,120	0.781	0.0230	2.9%	2.3%
39	CINECA	Fermi - IBM BlueGene/Q	163,840	1.789	0.0216	1.2%	1.0%
40	SURFsara, Amsterdam	Cartesius2 bullx B720, dual socket Intel Xeon E5-2690 v3, Infiniband FDR	25,920	0.848	0.0195	2.3%	1.8%

41-50

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
41	Cyberscience Center / Tohoku University	NEC SX-ACE 4C+IXS	2,048	0.123	0.0150	12.2%	11.4%
42	Cybermedia Center, Osaka University	Osaka U ACE -- NEC SX-ACE	2,048	0.123	0.0142	11.5%	10.8%
43	SGI	SGI ICE X -- Intel Xeon E5-2690v4, Infiniband EDR	16,128	0.602	0.0122	2.0%	1.8%
44	LNCC	Santos Dumont, Bullx Intel E5-2695v2, Infiniband FDR	17,616	0.321	0.0121	3.8%	3.5%
45	Intel	Endeavor - Intel Cluster, Dual Intel Xeon E5-2697v3 14C 2.700GHz, Infiniband FDR, Intel Xeon Phi 7120P	51,392	0.759	0.0112	1.5%	1.2%
46	Meteo France	Beaufix - Bullx DLC B710 Blades, Intel Xeon E5-2697v2 12C 2.7GHz, Infiniband FDR	24,192	0.469	0.0110	2.3%	2.1%
47	Saint Petersburg Polytechnic University	Polytechnic - RSC Tornado Intel E52697v3, Infiniband FDR	17,444	0.658	0.0108	1.6%	1.3%
48	Meteo France	Prolix - Bullx DLC B710 Blades, Intel Xeon E5-2697v2 12C 2.7GHz, Infiniband FDR	23,760	0.465	0.0100	2.1%	1.9%
49	Bull Angers	Manny Bullx B720, Xeon E5-2690v3 12C 2.600GHz, InfiniBand FDR	12,960	0.430	0.0097	2.3%	1.8%
50	University Heidelberg and University Mannheim	bwForCluster - Intel E5-2630v3, Infiniband QDR	7,552	0.241	0.0093	3.9%	3.2%

51-60

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
51	Michigan State University	Laconia -- Intel Xeon E5-2680v4, Infiniband EDR FDR	1,008,760	0.536	0.0091	1.7%	1.2%
52	University of Duisburg-Essen	magnitUDE -- Intel Xeon E5-2650v4, Intel OmniPath	12	0.437	0.0090	2.1%	1.9%
53	CALMIP / University of Toulouse	EOS - Bullx DLC B710 Blades, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR	12,240	0.255	0.0073	2.8%	2.6%
54	Christian-Albrechts-Universitaet zu Kiel	NEC SX-ACE -- NEC SX-ACE	1,024	0.062	0.0068	11.1%	10.5%
55	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC/DL -- Intel Xeon E5-2620-V2, Infiniband FDR	2,720	0.273	0.0068	2.5%	1.6%
56	University of Tuebingen	BinAC -- Intel Xeon E5-2680v4, Infiniband FDR	4,800	0.209	0.0063	3.0%	2.2%
57	The Institute of Atmospheric Physics, Chinese Academy of Sciences	Earth System Numerical Simulator-1 - Intel E5-2680-V3, Infiniband FDR	24,912	0.738	0.0063	0.8%	0.6%
58	Joint Supercomputer Center RAS	MVS-10P - Intel E5-2690, Infiniband FDR, Xeon Phi SE10X	2,992	0.376	0.0049	1.3%	0.9%
59	University of Rijeka	Bura - Bullx Intel E5-2690v3, Infiniband FDR	5,952	0.234	0.0047	2.0%	1.6%
60	CINECA	Galileo - Dual Intel E5-2630 v3 2.4 GHz, Infiniband QDR, Dual NVIDIA K80	2,720		0.0046		1.9%

61-70

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
61	NSC / Linkoping	Bifrost - ASUS, Intel Xeon E5-2640v3 8C 2.6GHz, Intel Truescale Infiniband QDR	10,256	0.326	0.0045	1.4%	0.8%
62	Shanghai Supercomputer Center	Magic Cube II - Intel E5-2680-V3, Infiniband EDR	9,960	0.296	0.0044	1.5%	1.1%
63	Max-Planck-Institut für Mikrostrukturphysik	Cruncher - Intel E5-2680-V3, Intel Truescale Infiniband QDR	12	0.112	0.0040	3.6%	2.8%
64	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	5,120	0.240	0.0039	1.6%	1.0%
65	Chelyabinsk	RSC Tornado SUSU, Intel X5680, Infiniband QDR, Xeon Phi SE10X	4,032	0.288	0.0036	1.2%	0.8%
66	CINECA	Galileo - Dual Intel E5-2630 v3 2.4 GHz, Infiniband QDR, Dual Intel Xeon Phi 7120P	13,600		0.0034		1.5%
67	Atos Angers	Sid - Bullx Intel E5-2680v3, InfiniBand FDR	4,224	0.129	0.0032	2.5%	2.0%
68	St. Petersburg Polytechnic University	Polytechnic RSC PetaStream - Intel E5-2650 v2, Infiniband FDR, Xeon Phi 5120D	232	0.170	0.0031	1.8%	1.2%
69	Supercomputing Center of Chinese Academy of Sciences	Era-2 - Intel E5-2680-V3, Infiniband FDR, Xeon Phi + NVIDIA K20	13560	0.407	0.0030	0.7%	0.6%
70	SURFsara	Cartesius - Bullx B515 cluster, Intel Xeon E5-2450v2 8C 2.5GHz, InfiniBand 4x FDR, Nvidia K40m	3,036	0.154	0.0025	1.7%	1.2%

71-80

Rank	Site	Computer	Cores	Rmax	HPCG	HPCG/HPL	% of Peak
71	CINECA	Galileo - Dual Intel E5-2630 v3 2.4 GHz, Infiniband QDR	6,400		0.0020		1.6%
72	Moscow State University / Research Computing Center	Lomonosov - Intel Xeon X5570/X5670/E5630 2.93/2.53 GHz, PowerXCell 8i Infiniband QDR, Dual NVIDIA Fermi 2070	78,660	0.617	0.0017	0.3%	0.2%
73	IT Services Provider	Aquarius - Intel Xeon E5-2640-V3, Infiniband QDR	8	0.034	0.0014	4.0%	3.2%
74	Joint Supercomputer Center RAS	RSC PetaStream - Intel E5-2667 v2, Infiniband FDR, Intel Xeon Phi 7120D	3,904	0.054	0.0012	2.2%	1.5%
75	Yaqingjie Street 30	hbemc_2016A -- Intel E5-2680v3, Infiniband FDR	2,304		0.0009		
76	Hefei City,Anhui Province	YUJING -- Intel Xeon E5-2680v3, custom	1,440	0.001	0.0008		
77	No.180 Wusidong Road. Baoding City, Hebei Province,P.R.C	KunYu -- Intel Xeon E5-2680v3, Infiniband FDR	960	0.001	0.0006		
78	hongguancun Software Park II, No. 10 West Dongbeiwang Road, Haidian District, Beijing 100193, China	CSRC -- Intel Xeon E5-2680v3, Infiniband FDR	528	0.000	0.0004		
79	18, Xueyuan Road, Haidian District, Beijing,China	geo -- Intel Xeon E5-2680v3, Infiniband FDR	12	0.000	0.0003		
80	CINECA	Pico - Dual Intel Xeon E5-2670v2 2.5 GHz, Gigabit Ethernet	1,200		0.0003		1.1%

Optimized Versions of HPCG

- Intel
  - MKL has packaged CPU version of HPCG
    - See: http://bit.ly/hpcg-intel
  - In the process of packaging Xeon Phi version to be released soon.
- Nvidia
  - Massimiliano Fatica and Evertt Phillips
  - Binary available
    - Contact Massimiliano mfatica@nvidia.com
- Bull
  - Developed by CEA requesting the release

HPCG Tech Reports

Toward a New Metric for Ranking High Performance Computing Systems

- Jack Dongarra and Michael Heroux

HPCG Technical Specification

- Jack Dongarra, Michael Heroux, Piotr Luszczek

SANDIA REPORT
SAND2013- 8752
Unlimited Release
Printed October 2013

HPCG Technical Specification

Michael A. Heroux, Sandia National Laboratories[1]
Jack Dongarra and Piotr Luszczek, University of Tennessee

Prepared by

SANDIA REPORT
SAND2013-4744
Unlimited Release
Printed June 2013

Toward a New Metric for Ranking High Performance Computing Systems

Jack Dongarra, University of Tennessee
Michael A. Heroux, Sandia National Laboratories[1]

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

[1] Corresponding Author, maherou@sandia.gov