Architecture-aware Algorithms and Software for Peta and Exascale Computing

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

Overview

- Look at High Performance Computing today
- A New Benchmark for HPC
- Top 10 Challenges for Extreme Scale Computing

State of Supercomputing in 2014

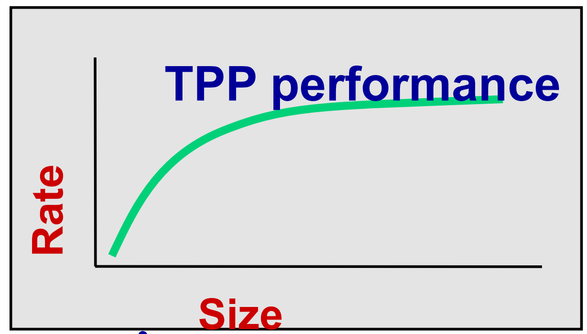
- Interest in supercomputing is now worldwide, and growing in many new markets (over 50% of Top500 computers are in industry).
- Pflops computing fully established with 31 systems.
- Exascale projects exist in many countries and regions.
- Three technology "swim lanes" or architecture possibilities are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs)
  - Special purpose lightweight cores (e.g. IBM BG)

TOP500 supercomputer

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

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

  Ax=b, dense problem

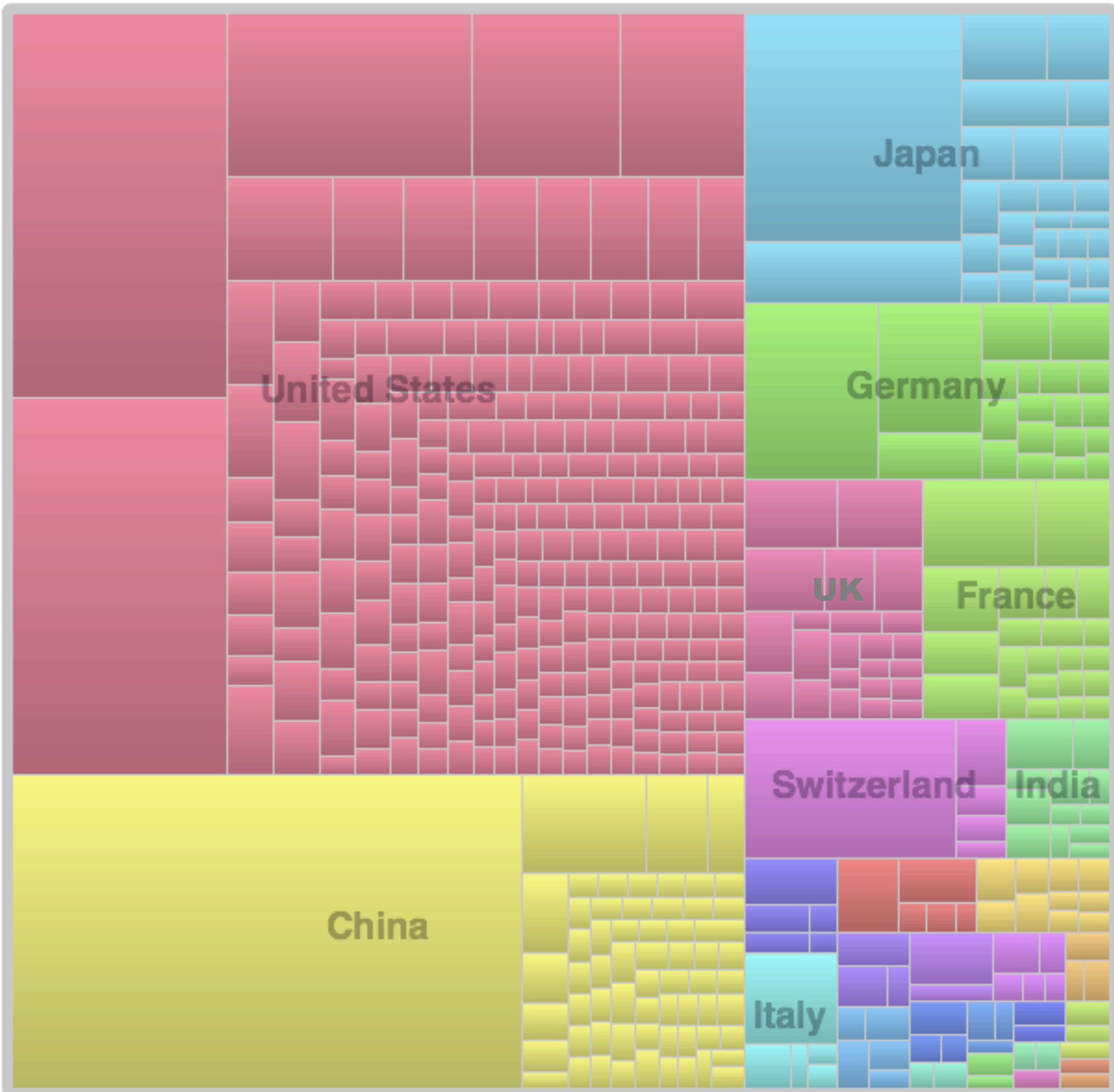


- Updated twice a year
  SC'xy in the States in November
  Meeting in Germany in June
- All data available from **www.top500.org**

Countries Share of Top500

November 2013

US & China Dominate with ~2/3 of the Performance



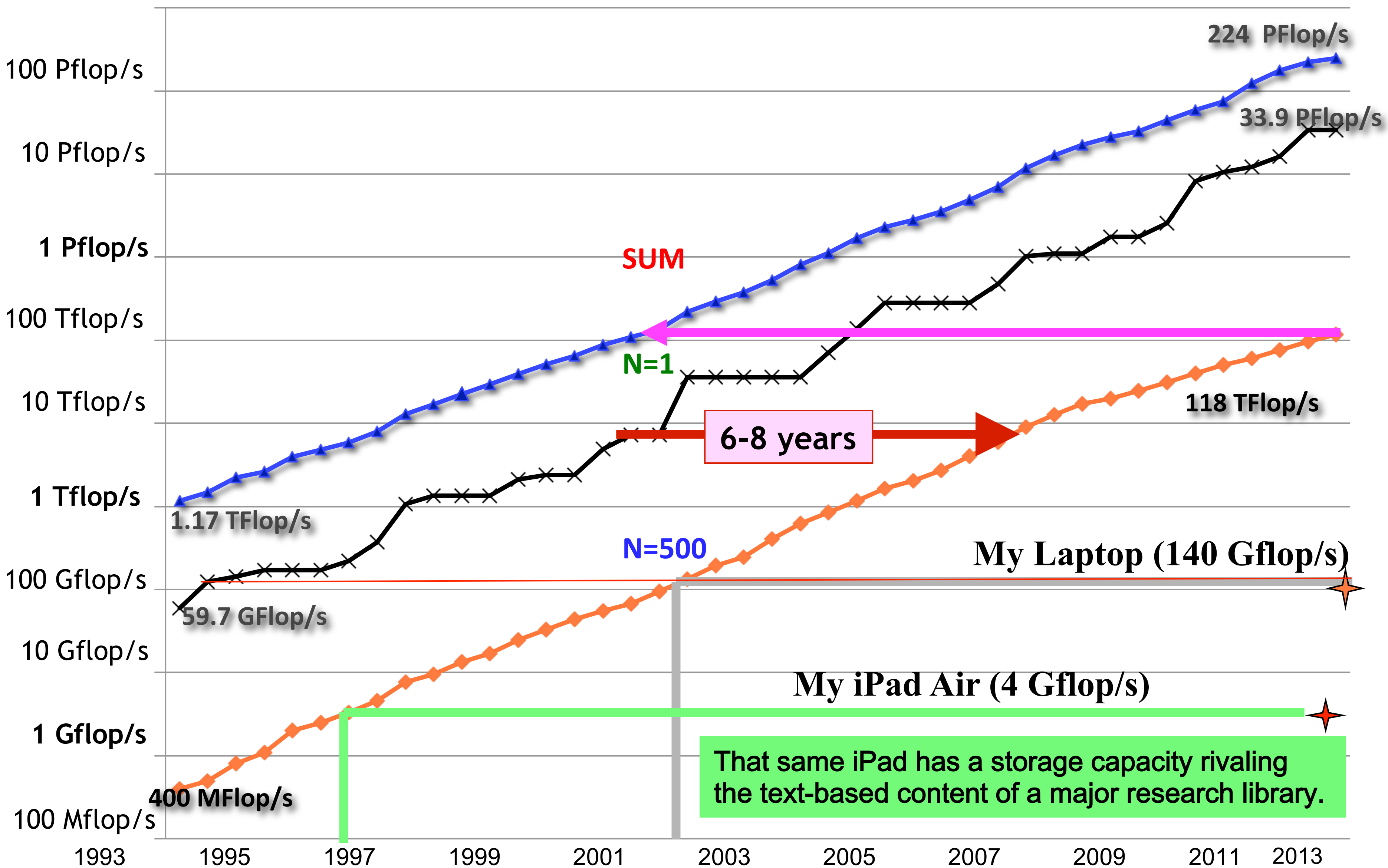
Absolute Counts

US:	267
China:	63
Japan:	28
UK:	23
France:	22
Germany:	20

% of Flop/s

US:	48.5%
China:	19.4%

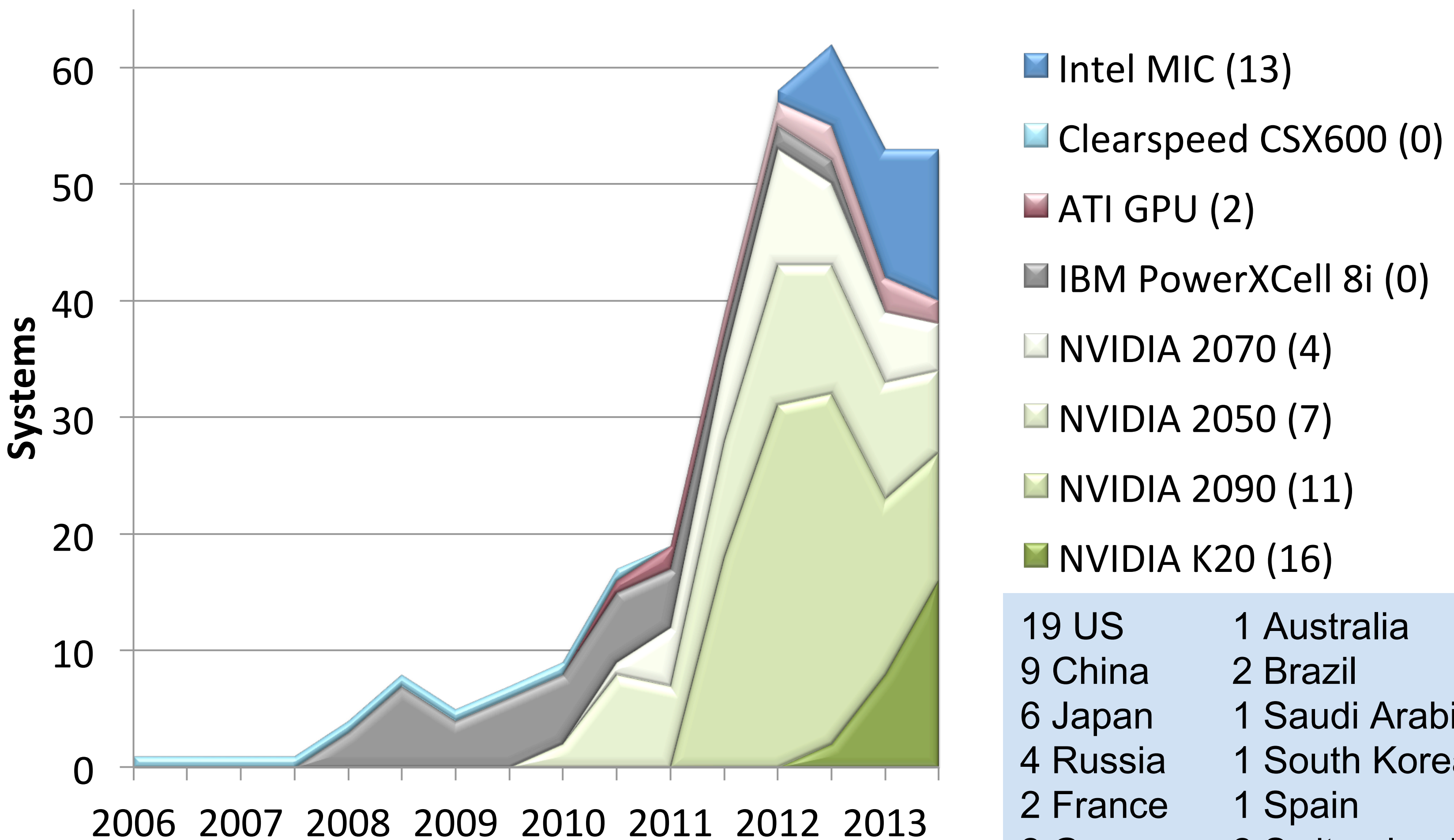
Performance Development of HPC Over the Last 20 Years From Top500



Current The TOP 10 Systems

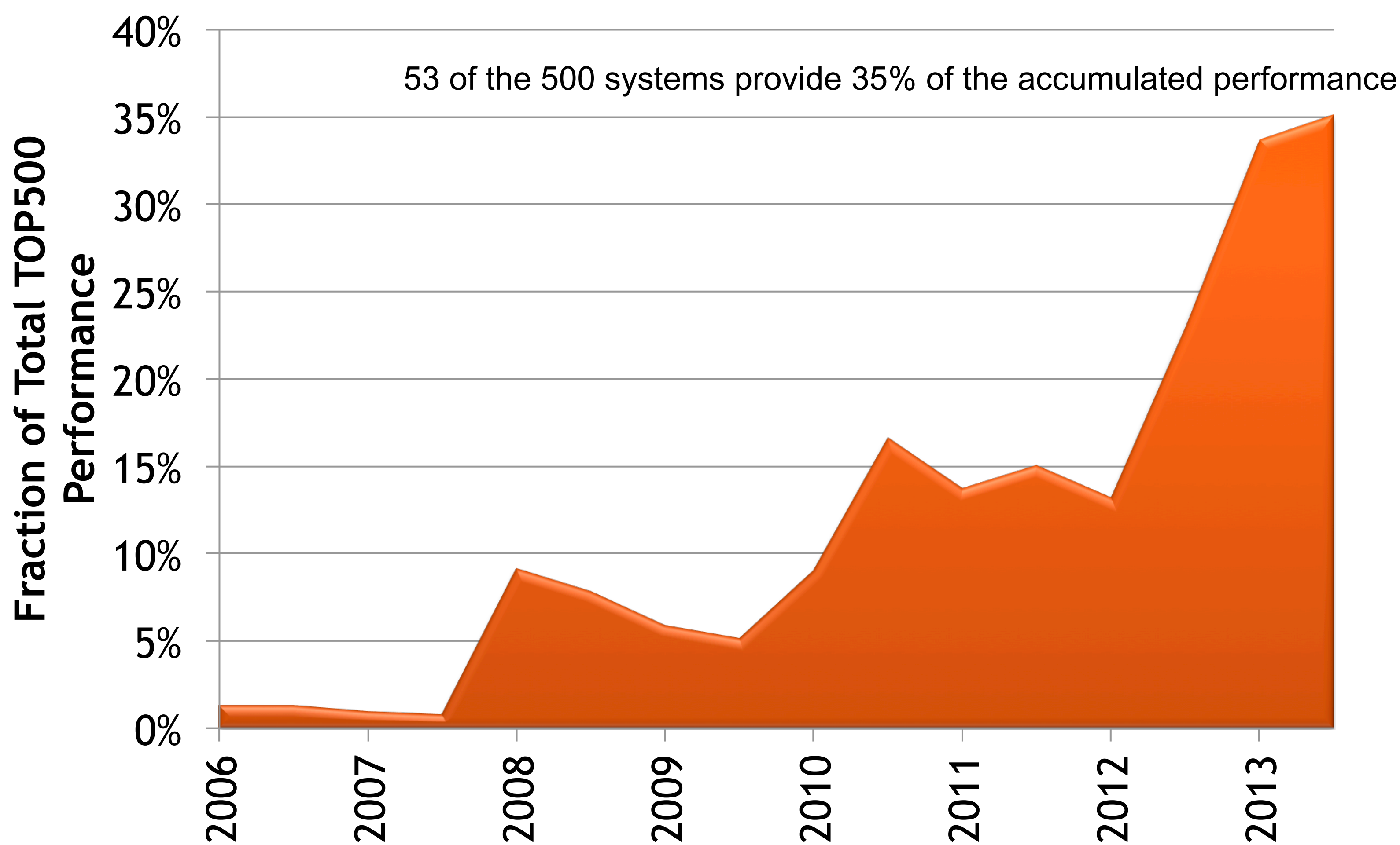
Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	MFlops /Watt
1	National Super Computer Center in Guangzhou	Tianhe-2 NUDT, Xeon 12C 2.2GHz + IntelXeon Phi (57c) + Custom	China	3,120,000	33.9	62	17.8	1905
2	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7 (16C) + Nvidia Kepler GPU (14c) + Custom	USA	560,640	17.6	65	8.3	2120
3	DOE / NNSA L Livermore Nat Lab	Sequoia, BlueGene/Q (16c) + custom	USA	1,572,864	17.2	85	7.9	2063
4	RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx (8c) + Custom	Japan	705,024	10.5	93	12.7	827
5	DOE / OS Argonne Nat Lab	Mira, BlueGene/Q (16c) + Custom	USA	786,432	8.16	85	3.95	2066
6	Swiss CSCS	Piz Daint, Cray XC30, Xeon 8C + Nvidia Kepler (14c) + Custom	Swiss	115,984	6.27	81	2.3	2726
7	Texas Advanced Computing Center	Stampede, Dell Intel (8c) + Intel Xeon Phi (61c) + IB	USA	204,900	5.17	61	4.5	1489
8	Forschungszentrum Juelich (FZJ)	JuQUEEN, BlueGene/Q, Power BQC 16C 1.6GHz+Custom	Germany	458,752	5.01	85	2.30	2178
9	DOE / NNSA L Livermore Nat Lab	Vulcan, BlueGene/Q, Power BQC 16C 1.6GHz+Custom	USA	393,216	4.29	85	1.97	2177
10	Leibniz Rechenzentrum	SuperMUC, Intel (8c) + IB	Germany	147,456	2.90	91*	3.42	848
500	Banking	HP	USA	22,212	.118	50		

Accelerators (53 systems)

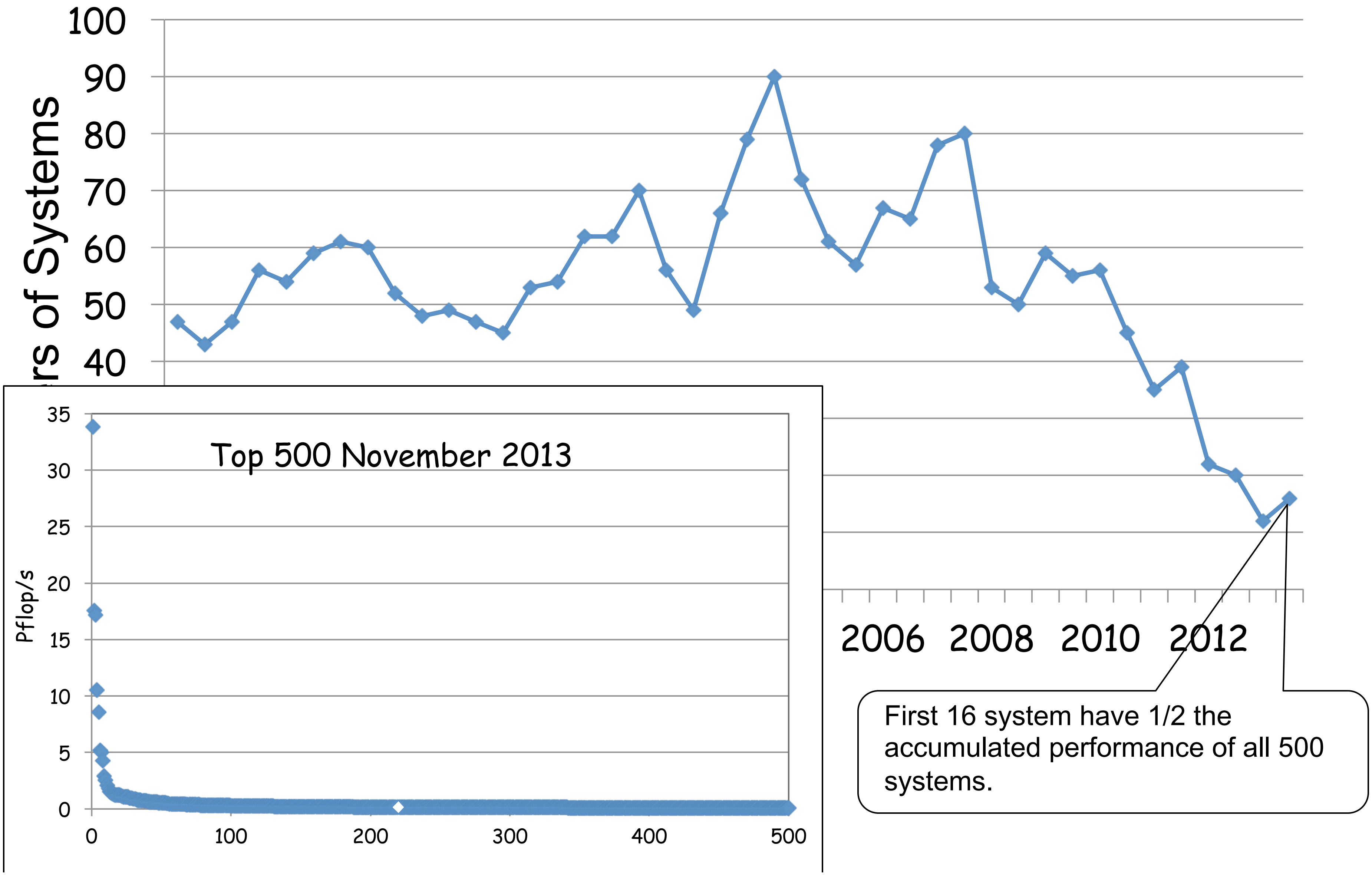


19 US	1 Australia
9 China	2 Brazil
6 Japan	1 Saudi Arabia
4 Russia	1 South Korea
2 France	1 Spain
2 Germany	2 Switzerland
2 India	1 UK
1 Italy	
1 Poland	

Top500 Performance Share of Accelerators



For the Top 500: Rank at which Half of Total Performance is Accumulated



Commodity plus Accelerator Today

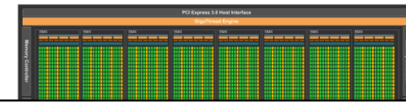
Commodity

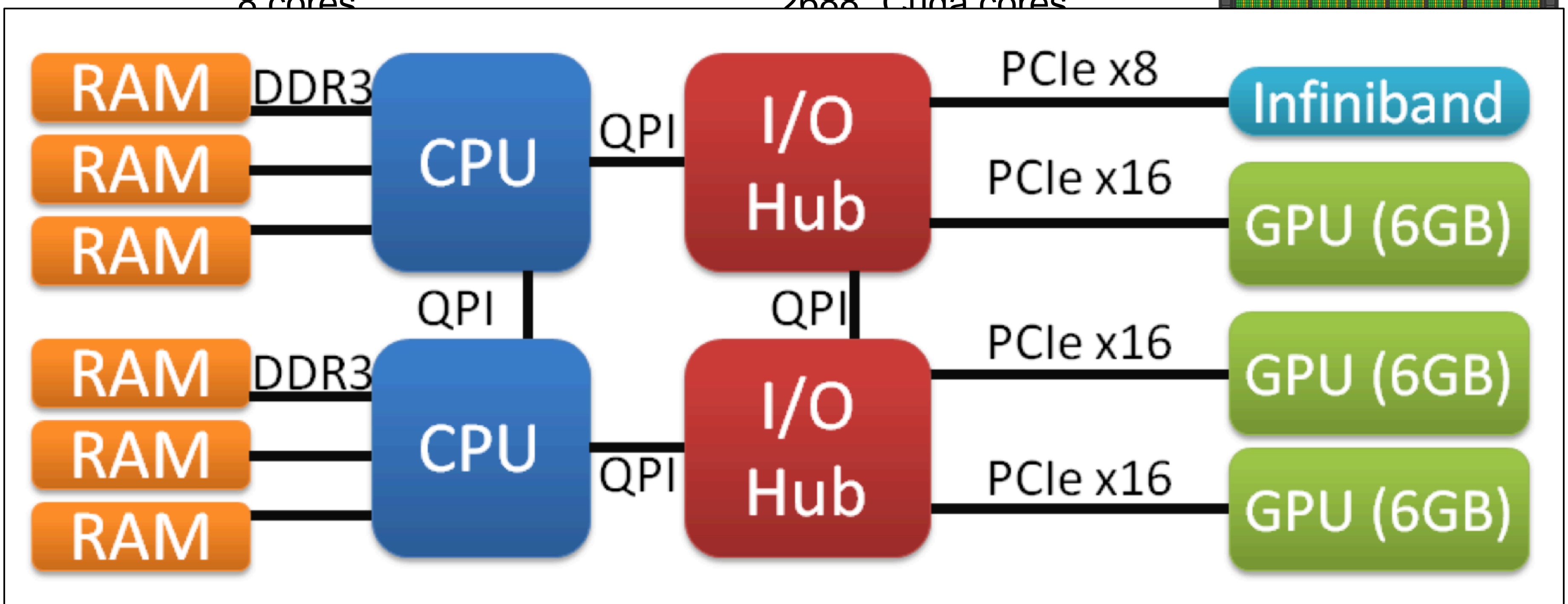
Intel Xeon
8 cores

Accelerator (GPU)

Nvidia K20X "Kepler"
2688 "Cuda cores"

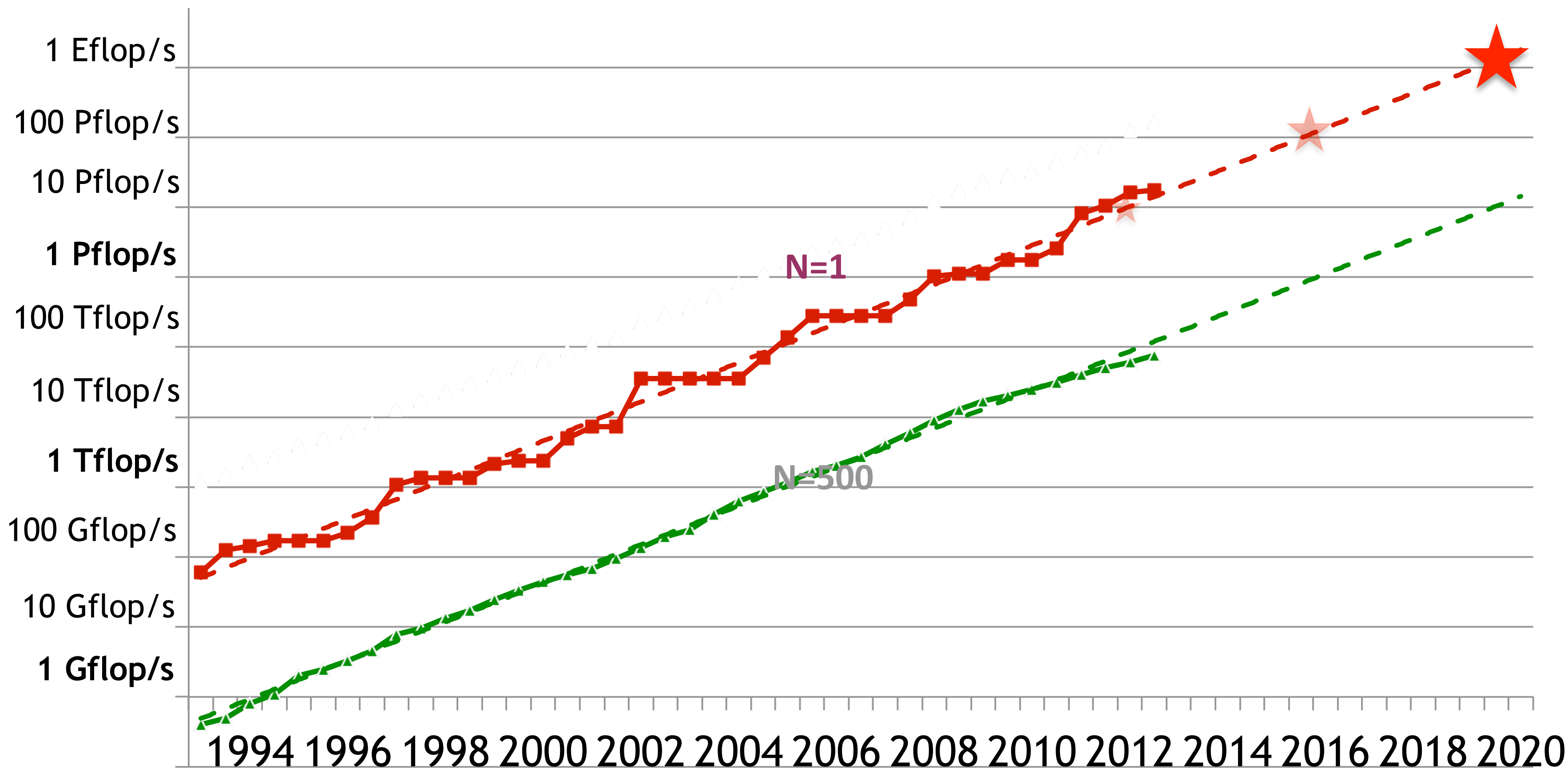
192 Cuda cores/SMX
2688 "Cuda cores"





Interconnect
PCI-e Gen2/3 16 lane
64 Gb/s (8 GB/s)
1 GW/s

Performance Development in Top500



Linpack Benchmark Started 36 Years Ago

- In the late 70's the fastest computer ran LINPACK at 14 Mflop/s
- In the late 70's floating point operations were expensive compared to other operations and data movement
- Matrix size, n = 100
  - That's what would fit in memory

$\frac{\frac{2}{3}n^3 + 2n^2 \text{ ops}}{\text{time}}$

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

Facility		TIME N=100 secs.	UNIT micro-secs.	Computer	Type	Compiler
NCAR	14.0	.049	0.14	CRAY-1	S	CFT, Assembly BLAS
LASL	4.64	.148	0.43	CDC 7600	S	FTN, Assembly BLAS
NCAR	3.58	.192	0.56	CRAY-1	S	CFT
LASL	3.27	.210	0.61	CDC 7600	S	FTN
Argonne	2.31	.297	0.86	IBM 370/195	D	H
NCAR	1.91	.359	1.05	CDC 7600	S	Local
Argonne	1.77	.388	1.33	IBM 3033	D	H
NASA Langley	1.40	.489	1.42	CDC Cyber 175	S	FTN
U. Ill. Urbana	1.36	.506	1.47	CDC Cyber 175	S	Ext. 4.6
LLL	1.24	.554	1.61	CDC 7600	S	CHAT, No optimize
SLAC	1.19	.579	1.69	IBM 370/168	D	H Ext., Fast mult.
Michigan	1.09	.631	1.84	Amdahl 470/V6	D	H
Toronto	.772	.890	2.59	IBM 370/165	D	H Ext., Fast mult.
Northwestern	.477	1.44	4.20	CDC 6600	S	FTN
Texas	.356	1.93*	5.63	CDC 6600	S	RUN
China Lake	.352	1.95*	5.69	Univac 1110	S	V
Yale	.265	2.59	7.53	DEC KL-20	S	F20
Bell Labs	.197	3.46	10.1	Honeywell 6080	S	Y
Wisconsin	.197	3.49	10.1	Univac 1110	S	V
Iowa State	.194	3.54	10.2	Itel AS/5 mod3	D	H
U. Ill. Chicago	[illegible]	4.10	11.9	IBM 370/158	D	G1
Purdue	[illegible]	5.69	16.6	CDC 6500	S	FUN
U. C. San Diego	[illegible]	13.1	38.2	Burroughs 6700	S	H
Yale	[illegible]	17.1*	49.9	DEC KA-10	S	F40

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

- The Benchmark evolved over time and today, the matrix size is arbitrary; looking at the rate of execution, make it as fast as possible.

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 merge 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	

The High Performance Linpack (HPL) Benchmark has a Number of Problems

- 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 an Exaflop 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 22 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!!
- 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

Ugly Things about HPL

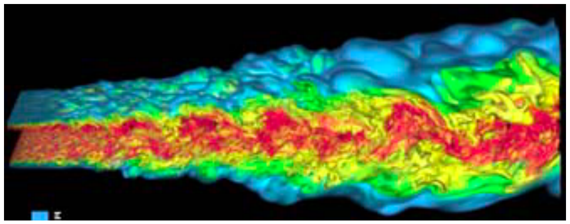
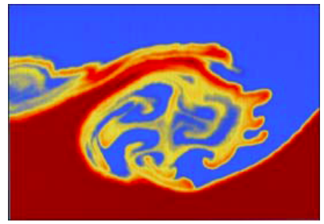
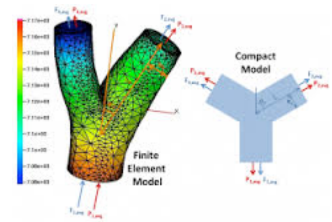
- Doesn't probe the architecture; only one data point
- Constrains the technology and architecture options for HPC system designers.
  - Skews system design.
- Floating point benchmarks are not quite as valuable to some as data-intensive system measurements

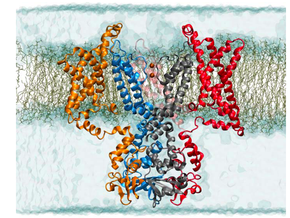
Many Other Benchmarks

- TOP500
- Green 500
- Graph ~~500~~ 160
- Sustained Petascale Performance
- HPC Challenge
- Perfect
- ParkBench
- SPEC-hpc
- Big Data Top100
- Livermore Loops
- EuroBen
- NAS Parallel Benchmarks
- Genesis
- RAPS
- SHOC
- LAMMPS
- Dhrystone
- Whetstone
- I/O Benchmarks

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

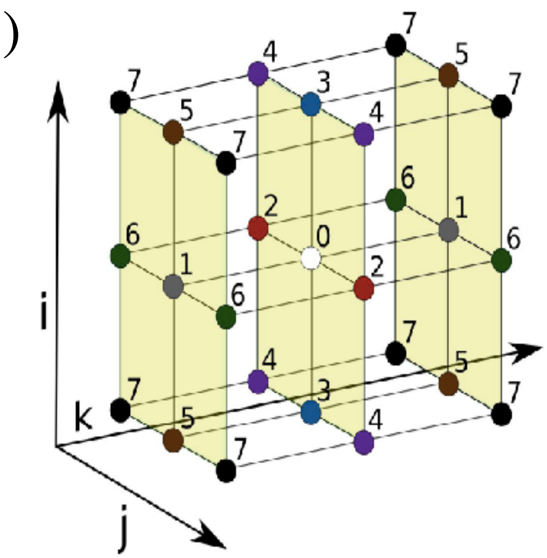
- High Performance Conjugate Gradient (HPCG).
- Solves $Ax=b$, A large, sparse, b known, x computed.
- An optimized implementation of PCG contains essential computational and communication patterns that are prevalent in a variety of methods for discretization and numerical solution of PDEs

- Patterns:
  - Dense and sparse computations.
  - Dense and sparse collective.
  - Data-driven parallelism (unstructured sparse triangular solves).

- Strong verification and validation properties

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.
  - 7 – 18 on boundary.
  - Symmetric positive definite.



27-point stencil operator

HPCG Design Philosophy

- Relevance to broad collection of important apps.
- Simple, single number.
- Few user-tunable parameters and algorithms:
  - The system, not benchmarker skill, should be primary factor in result.
  - Algorithmic tricks don’t give us relevant information.
- Algorithm (PCG) is vehicle for organizing:
  - Known set of kernels.
  - Core compute and data patterns.
  - Tunable over time (as was HPL).
- Easy-to-modify:
  - _ref kernels called by benchmark kernels.
  - User can easily replace with custom versions.
  - Clear policy: Only kernels with _ref versions can be modified.

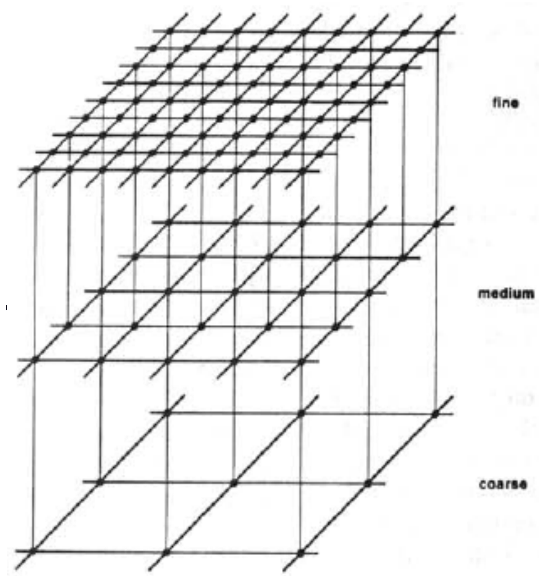
PCG ALGORITHM

- $p_0 := x_0, r_0 := b\text{-}Ap_0$
- Loop i = 1, 2, …
  - $z_i := M^{-1}r_{i-1}$
  - if i = 1
    - $p_i := z_i$
    - $a_i :=$ `dot_product`(r_{i-1}, z)
  - else
    - $a_i :=$ `dot_product`(r_{i-1}, z)
    - $b_i := a_i/a_{i\text{-}1}$
    - $p_i := b_i{*}p_{i-1}+z_i$
  - end if
  - $a_i :=$ `dot_product`(r_{i-1}, z_i) /`dot_product`$(p_i, A{*}p_i)$
  - $x_{i+1} := x_i + a_i{*}p_i$
  - $r_i := r_{i-1} - a_i{*}A{*}p_i$
  - 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().

(In 2D, something like this)



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

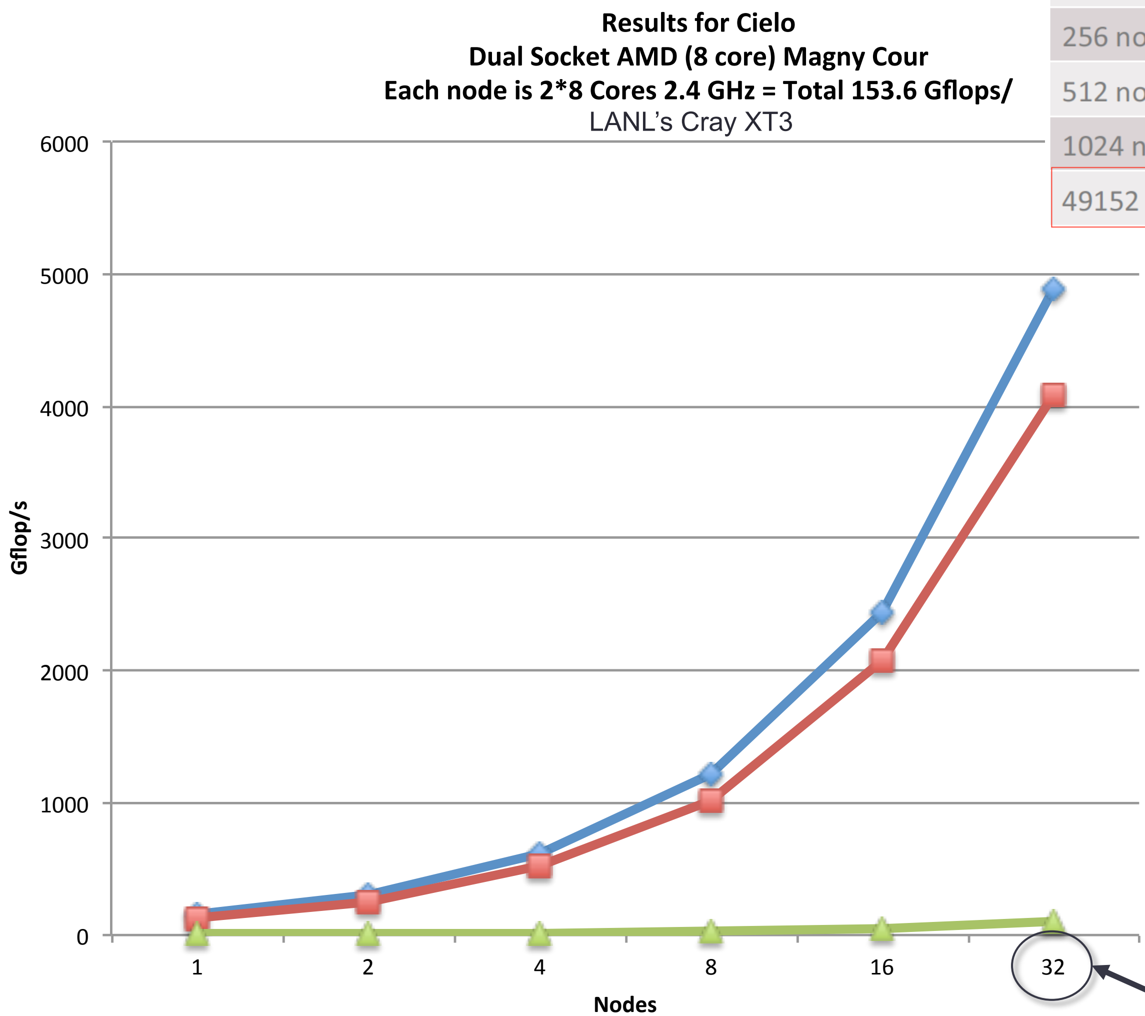
```
LA = tril(A); UA = triu(A); DA = diag(diag(A));

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

Performance "Shock" (Unoptimized Version)

Mira Partition Size ANL's IBM BG/Q	Peak Gflops	Sustained Gflops	% of peak
64 nodes	13107.2	73.4	0.56%
128 nodes	26214.4	147.43	0.56%
256 nodes	52428.8	293.8	0.56%
512 nodes	104857.6	587.97	0.56%
1024 nodes	209715.2	1176.69	0.56%
49152 nodes	10066329.6	55177.6	0.55%

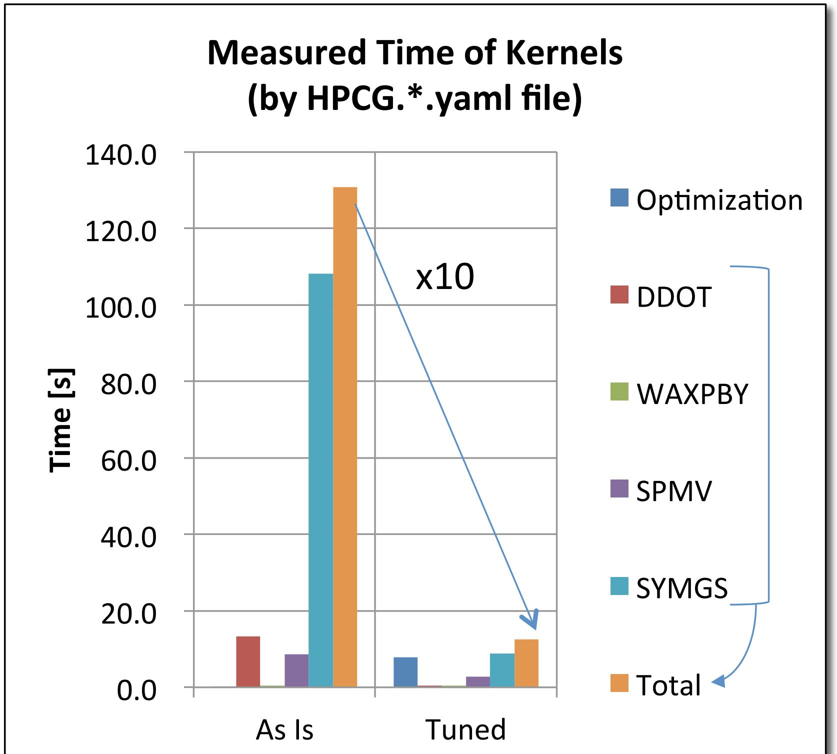
Courtesy Kalyan Kumaran, Argonne



Courtesy Mahesh Rajan, Sandia

512 MPI Processes

Tuning result on the K computer



8 Processes, 8 Threads/Process (Peak 128x8 GFLOPS)

Summary of "as is" code on the K

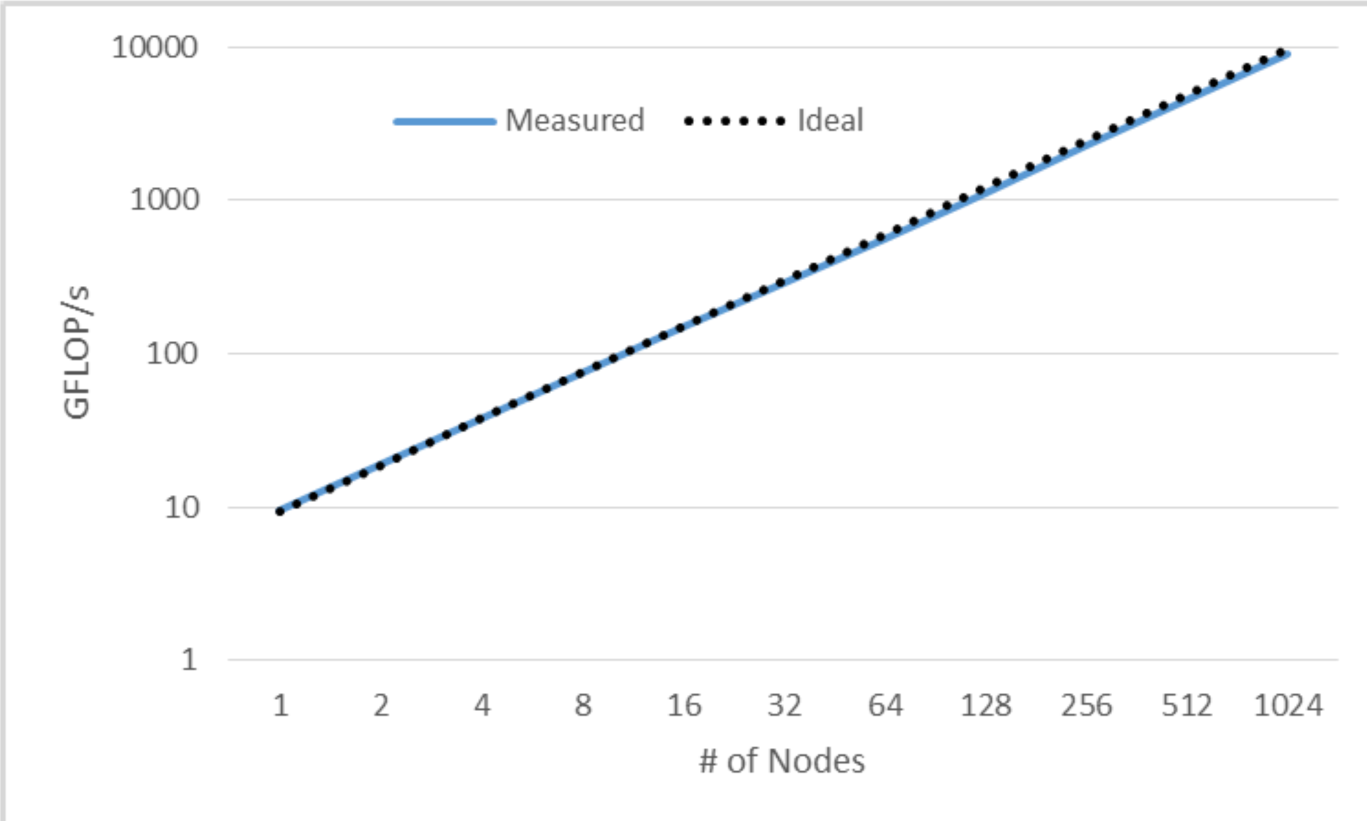
- Parallel scalability shouldn't be obstacle for large scale problem
- We are focusing on single CPU performance improvement

Improvement

- Total x10 speed up now
  - Continuous memory for matrix
  - Multi-coloring for SYMGS multi-threading
- Under Studying
  - Node re-ordering for SPMV
  - Advanced matrix storage way
  - And so on

Slide courtesy Naoya Maruyama, RIKEN AICS, and Fujitsu

Multi-node Scaling



Stampede cluster, dual socket of 8-core SNB, 2.7 GHz

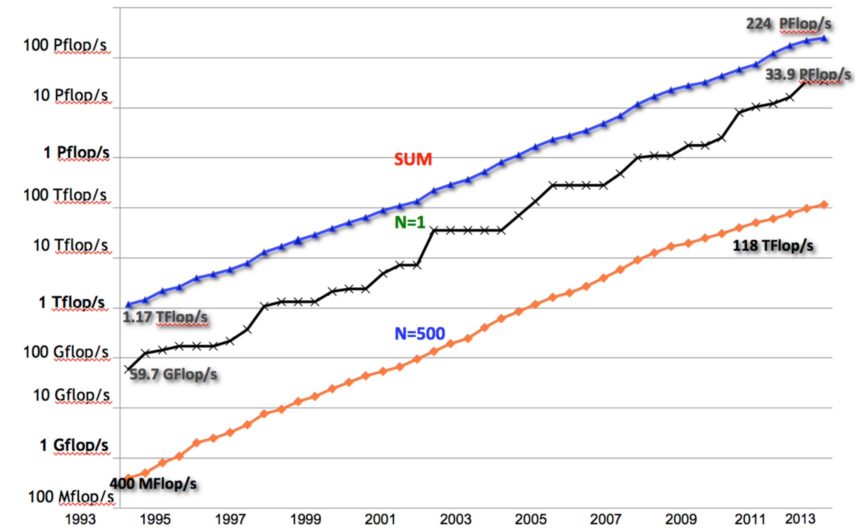
2 MPI processes per node (1 MPI process per skt. for NUMA)

160^3 input per MPI process

93% parallelization efficiency with 1024 nodes

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.
  - Similar perhaps to the Green500 listing.



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

Today's #1 System

Systems	2013 Tianhe-2
System peak	**55 Pflop/s**
Power	**18 MW** **(3 Gflops/W)**
System memory	1.4 PB (1.024 PB CPU + .384 PB CoP)
Node performance	3.43 TF/s (2 CPU +3 CoP)
Node concurrency	24 cores CPU + 171 cores CoP
Node Interconnect BW	6.36 GB/s
System size (nodes)	16,000
Total concurrency	3.12 M 12.48M threads (4/core)
MTTF	Few / day

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

Systems	2013 Tianhe-2
System peak	**55 Pflop/s**
Power	**18 MW** **(3 Gflops/W)**
System memory	1.4 PB (1.024 PB CPU + .384 PB CoP)
Node performance	3.43 TF/s (2 CPU +3 CoP)
Node concurrency	24 cores CPU + 171 cores CoP
Node Interconnect BW	6.36 GB/s
System size (nodes)	16,000
Total concurrency	3.12 M 12.48M threads (4/core)
MTTF	Few / day

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

Systems	2013 Tianhe-2	2020-2022	Difference Today & Exa
System peak	**55 Pflop/s**	**1 Eflop/s**	**~20x**
Power	**18 MW (3 Gflops/W)**	**~20 MW (50 Gflops/W)**	**O(1) ~15x**
System memory	1.4 PB (1.024 PB CPU + .384 PB CoP)	32 - 64 PB	~50x
Node performance	3.43 TF/s (2 CPU +3 CoP)	1.2 or 15TF/s	O(1)
Node concurrency	24 cores CPU + 171 cores CoP	O(1k) or 10k	~5x - ~50x
Node Interconnect BW	6.36 GB/s	200-400GB/s	~40x
System size (nodes)	16,000	O(100,000) or O(1M)	~6x - ~60x
Total concurrency	3.12 M 12.48M threads (4/core)	O(billion)	~100x
MTTF	Few / day	Many / day	O(?)

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

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
   - With current semiconductor technologies, all proposed exascale designs would consume ~200 MW of power.
   - 20 – 40 MW, comparable to that used by commercial cloud data centers

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
   - Cost to move a datum exceeds the cost of a floating point operation,
   - Necessitating very energy efficient low latency, high bandwidth interconnects for fine-grained data exchanges among hundreds of thousands of processors.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
   - New memory technologies, including processor-in-memory, stacked memory, non-volatile memory approaches.
   - Memory per node will necessarily be smaller than in current designs.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
   - Today failures infrequent.
   - At very large scale, systemic resilience in the face of regular component failures will be essential.
   - Dynamic, adaptive energy management must become an integral part of system software, for both economic and technical reasons.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming systems that express massive parallelism, data locality, and resilience
   - The widely used CSP model (i.e. MPI) places the burden of locality and parallelization on applications.
   - More expressive programming models are needed that can deal with this behavior and simplify the developer’s efforts.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming environments that express massive parallelism, data locality, and resilience
6. **Data management:**
   - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
   - Efficient *in situ* data analysis will require restructuring of scientific workflows and applications.
   - Techniques for data coordinating and mining

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming environments that express massive parallelism, data locality, and resilience
6. **Data management:**
   - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
7. **Exascale Algorithms:**
   - Reformulating science problems and refactoring their solution algorithms for exascale systems.
   - Adapting them to billion-way parallelism will require redesigning, or even reinventing, the algorithms, and potentially reformulating the science problems.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming environments that express massive parallelism, data locality, and resilience
6. **Data management:**
   - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
7. **Exascale Algorithms:**
   - Reformulating science problems and refactoring their solution algorithms for exascale systems.
8. **Algorithms for discovery, design, and decision:**
   - Facilitating mathematical optimization and uncertainty quantification for exascale discovery, design, and decision making.
   - Large-scale computations are themselves experiments that probe the sample space of numerical models.
   - Understanding the sensitivity of computational predictions to model inputs and assumptions, particularly when involving complex, multidisciplinary applications is dependent on new tools and techniques for application validation and assessment.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming environments that express massive parallelism, data locality, and resilience
6. **Data management:**
   - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
7. **Exascale Algorithms:**
   - Reformulating science problems and refactoring their solution algorithms for exascale systems.
8. **Algorithms for discovery, design, and decision:**
   - Facilitating mathematical optimization and uncertainty quantification for exascale discovery, design, and decision making.
9. **Resilience and correctness:**
   - Ensuring correct scientific computation in face of faults, reproducibility, and algorithm verification challenges.
   - With frequent transient and permanent faults, lack of reproducibility in collective communication, and new mathematical algorithms with limited verification, computation validation and correctness assurance rise dramatically in importance for the next generation of massively parallel systems.

Top 10 Challenges to Exascale

1. **Energy efficiency:**
   - Creating more energy efficient circuit, power, and cooling technologies.
2. **Interconnect technology:**
   - Increasing the performance and energy efficiency of data movement.
3. **Memory Technology:**
   - Integrating advanced memory technologies to improve both capacity and bandwidth.
4. **Scalable System Software:**
   - Developing scalable system software that is power and resilience aware.
5. **Programming systems:**
   - Inventing new programming environments that express massive parallelism, data locality, and resilience
6. **Data management:**
   - Creating data management software that can handle the volume, velocity and diversity of data that is anticipated.
7. **Exascale Algorithms:**
   - Reformulating science problems and refactoring their solution algorithms for exascale systems.
8. **Algorithms for discovery, design, and decision:**
   - Facilitating mathematical optimization and uncertainty quantification for exascale discovery, design, and decision making.
9. **Resilience and correctness:**
   - Ensuring correct scientific computation in face of faults, reproducibility, and algorithm verification challenges.
10. **Scientific productivity:**
    - Increasing the productivity of computational scientists with new software engineering tools and environments.
    - Unless researcher productivity increases, the time to solution may be dominated by application development, not computation.

Algorithmic and Mathematics Challenges

Advances in mathematical models, algorithms, and analysis for exascale simulations to enable extreme-scale science

- **Exascale computing driven by grand-challenge science**
  - More resources for more complete and sophisticated models
  - Answering new scientific questions will require rethinking, reformulating and developing new mathematical techniques
  - New predictive simulation and analysis capabilities
- **Advances in algorithms synergistic with hardware improvements**

Machine improvements tend to improve base or coefficient

$$O(N^p)$$

Model and algorithm improvements can improve exponent

- **Today's algorithms will not (are really hard to) run efficiently on future exascale machines**

Major Changes to Software & Algorithms

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

  - Data movement is expense
  - Flop/s are cheap, so are provisioned in excess

Summary

- Major Challenges are ahead for extreme computing
  - Parallelism $O(10^9)$
    - Issues with Math & Algorithm formulation and Programming
  - Hybrid
    - Peak and HPL may be very misleading
    - No where near close to peak for most apps, (5 - 10% of peak)
  - Fault Tolerance
    - Today Sequoia BG/Q node failure rate is 1.25 failures/day
  - Power
    - 50 Gflops/w (today at 2 Gflops/w)
- We will need completely new approaches and technologies to reach the Exascale level
- International collaboration is more important than ever.