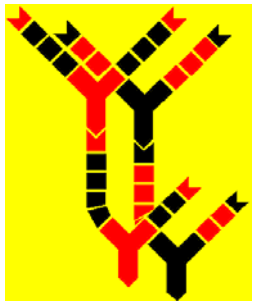


HIGH PERFORMANCE COMPUTING

From Clouds and Big Data to Exascale and Beyond

An International Advanced Workshop
Cetraro – Italy, July 7 – 11, 2014

High Performance Computing Today and Benchmarking the Future

Jack Dongarra

University of Tennessee

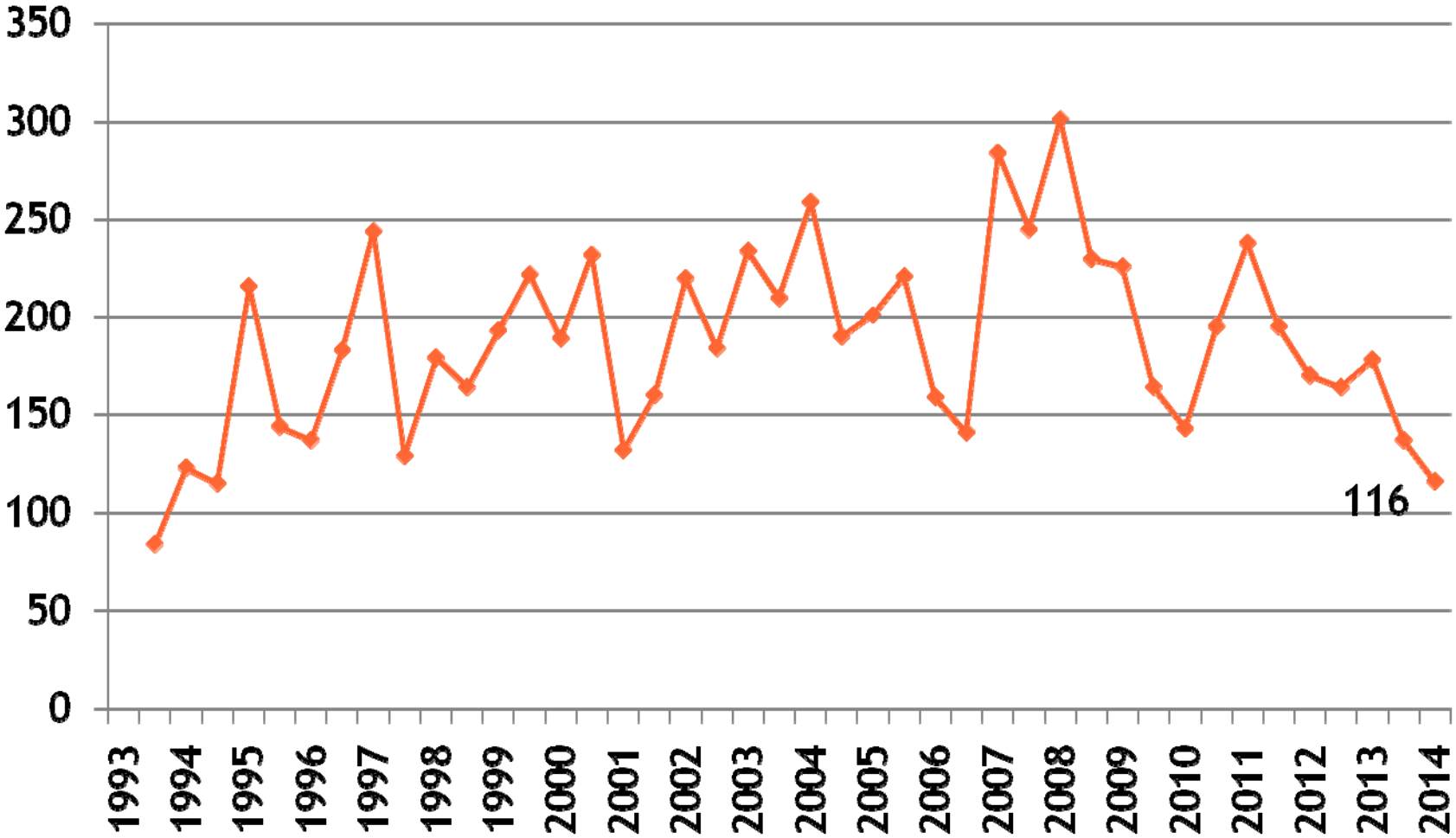
Oak Ridge National Laboratory

Current The TOP 10 Systems

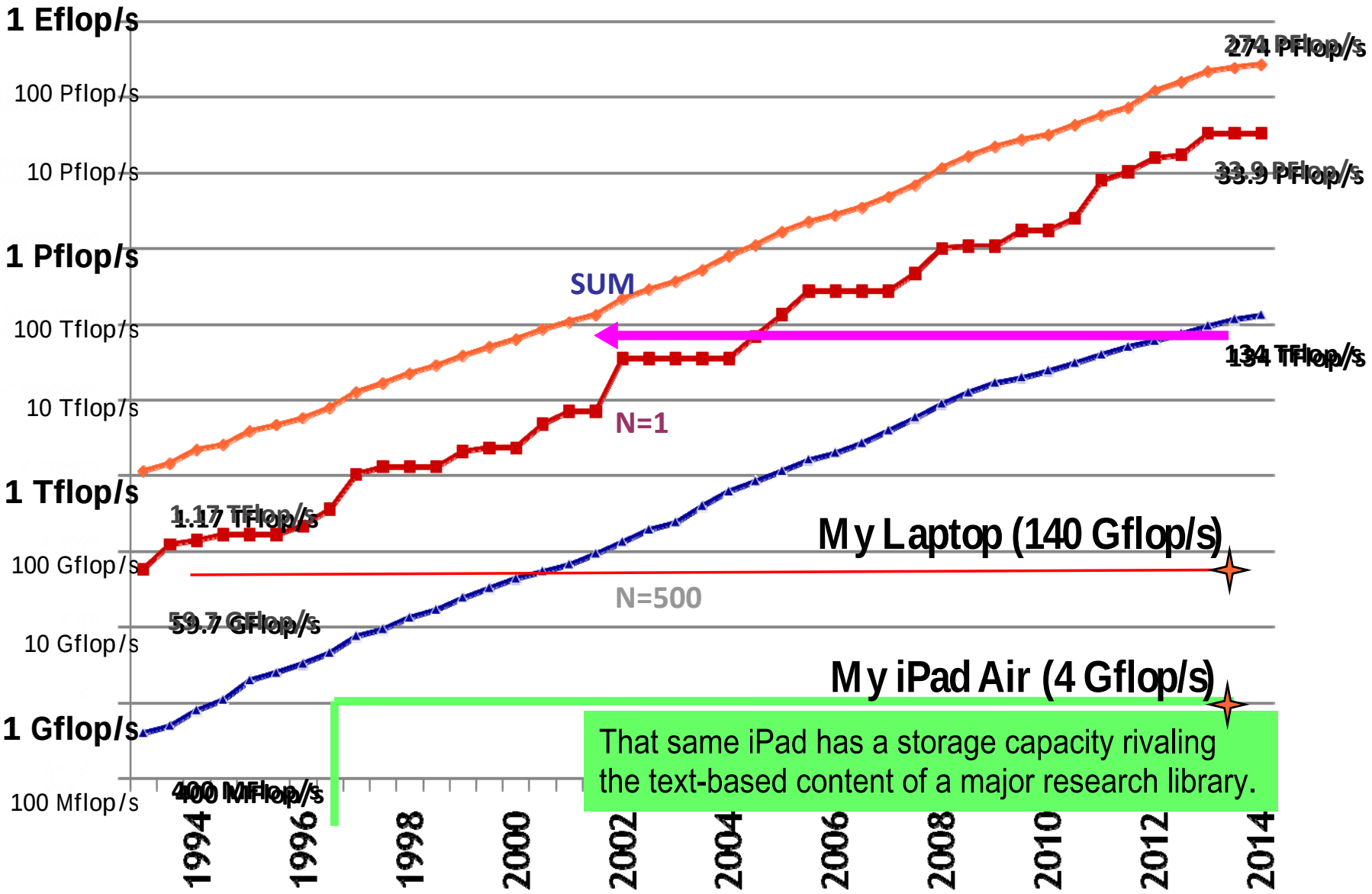
Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	MFlops /Watt
1	National Super Computer Center in Guangzhou	Tianhe-2 NUDT, Xeon 12C 2.2GHz + IntelXeon Phi (57c) + Custom	China	3,120,000	33.9	62	17.8	1905
2	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7 (16C) + Nvidia Kepler GPU (14c) + 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	Government	Cray XC30, Xeon E5 12C 2.7GHz, Custom	USA	225,984	3.14	64		

500	**Meteorological**	Cray XC30	Germany	**7280**	**.134**	**91**		

Replacement Rate



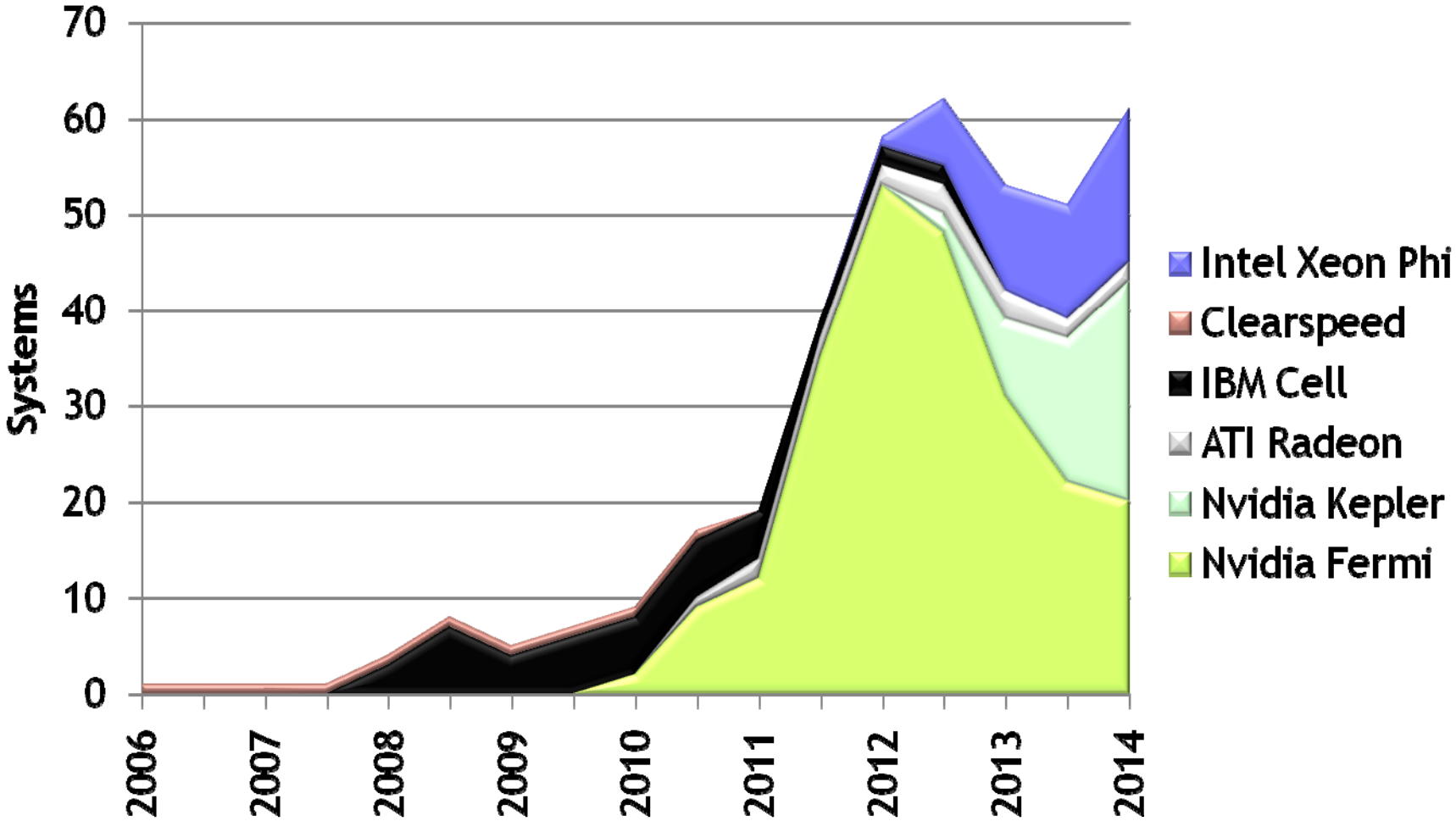
Performance Development of HPC over the Last 22 Years from the Top500



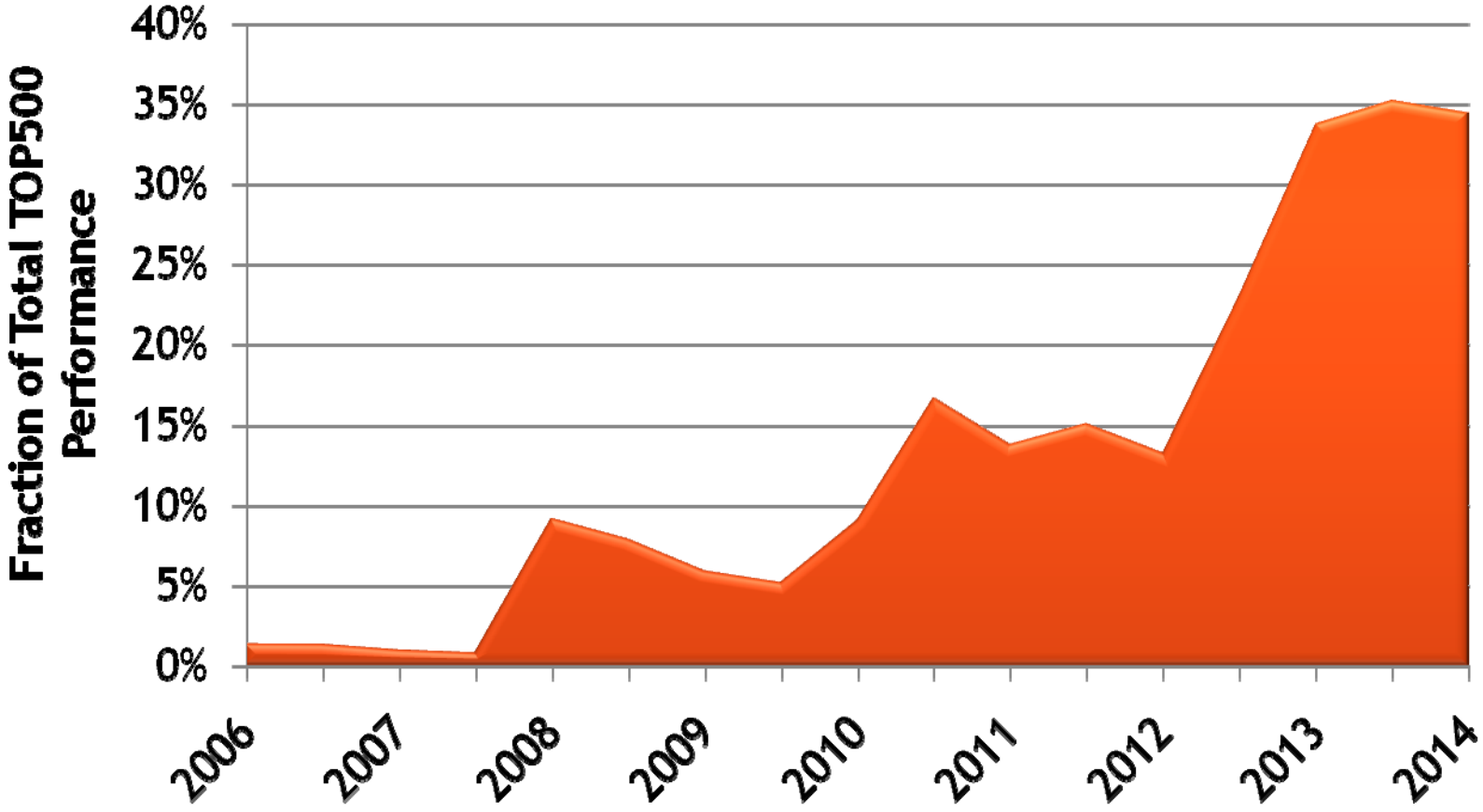
Top500 Factoids

- There are 37 systems > Pflop/s (up 6 from November).
- About 90% of all the systems on the Top500 list are integrated by U.S. vendors, including 65 of the 76 Chinese supercomputers.
- HP has 182 systems on this list, or more than 36%, followed by IBM with 176, or 35%. Cray has 50 or 10%, SGI at 19 systems, and Dell at 8 systems.
- Intel processors largest share, 87% followed by AMD, 6%.
- For the first time, < 50% of Top500 are in the U.S. -- just 233 of the systems are U.S.-based, China #2 w/76.
- IBM's BlueGene/Q is still the most popular system in the TOP10 with four entries.
- Infiniband found in 221 systems, GigE in 202, 10-GigE in 75.

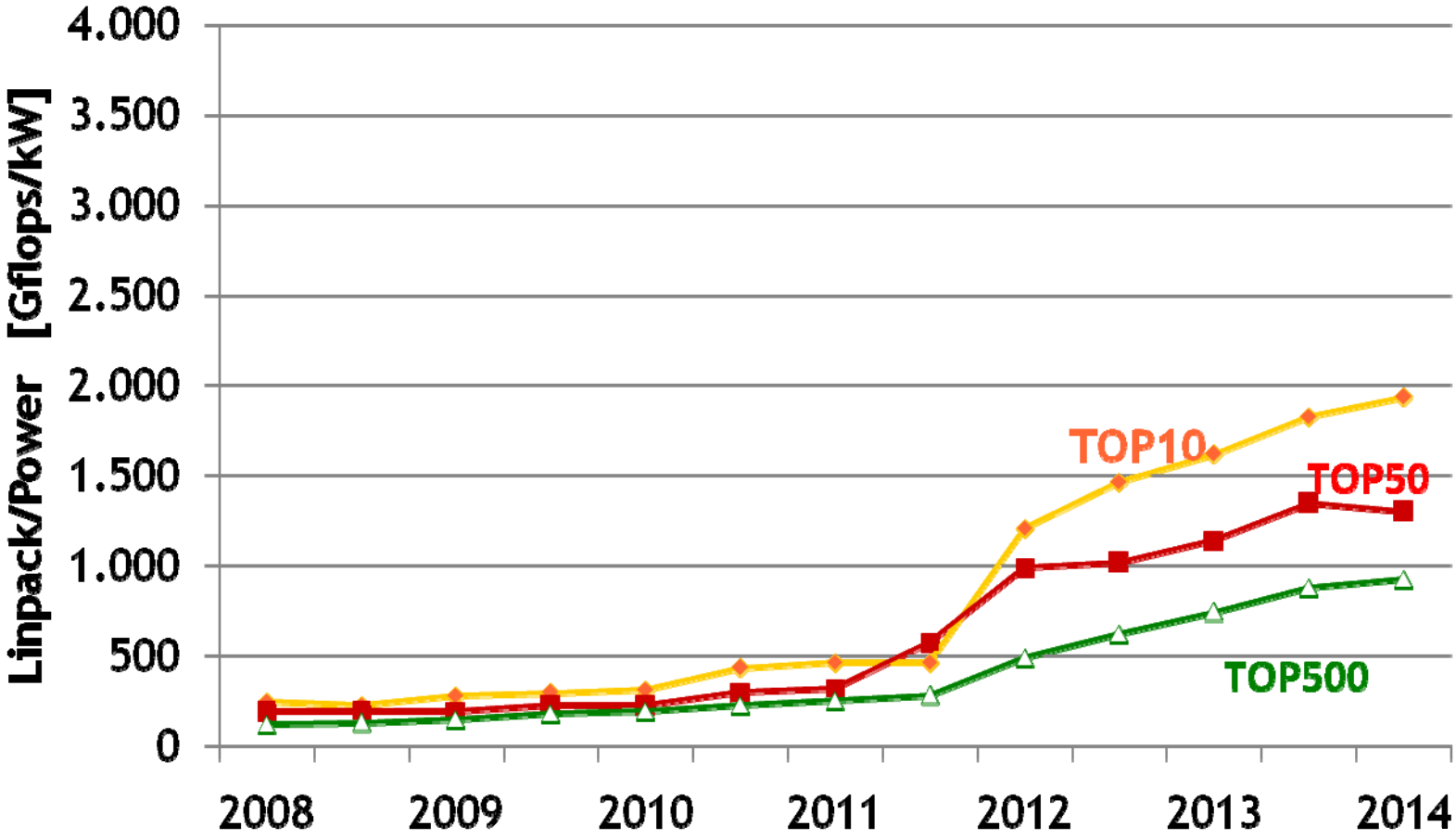
Accelerators



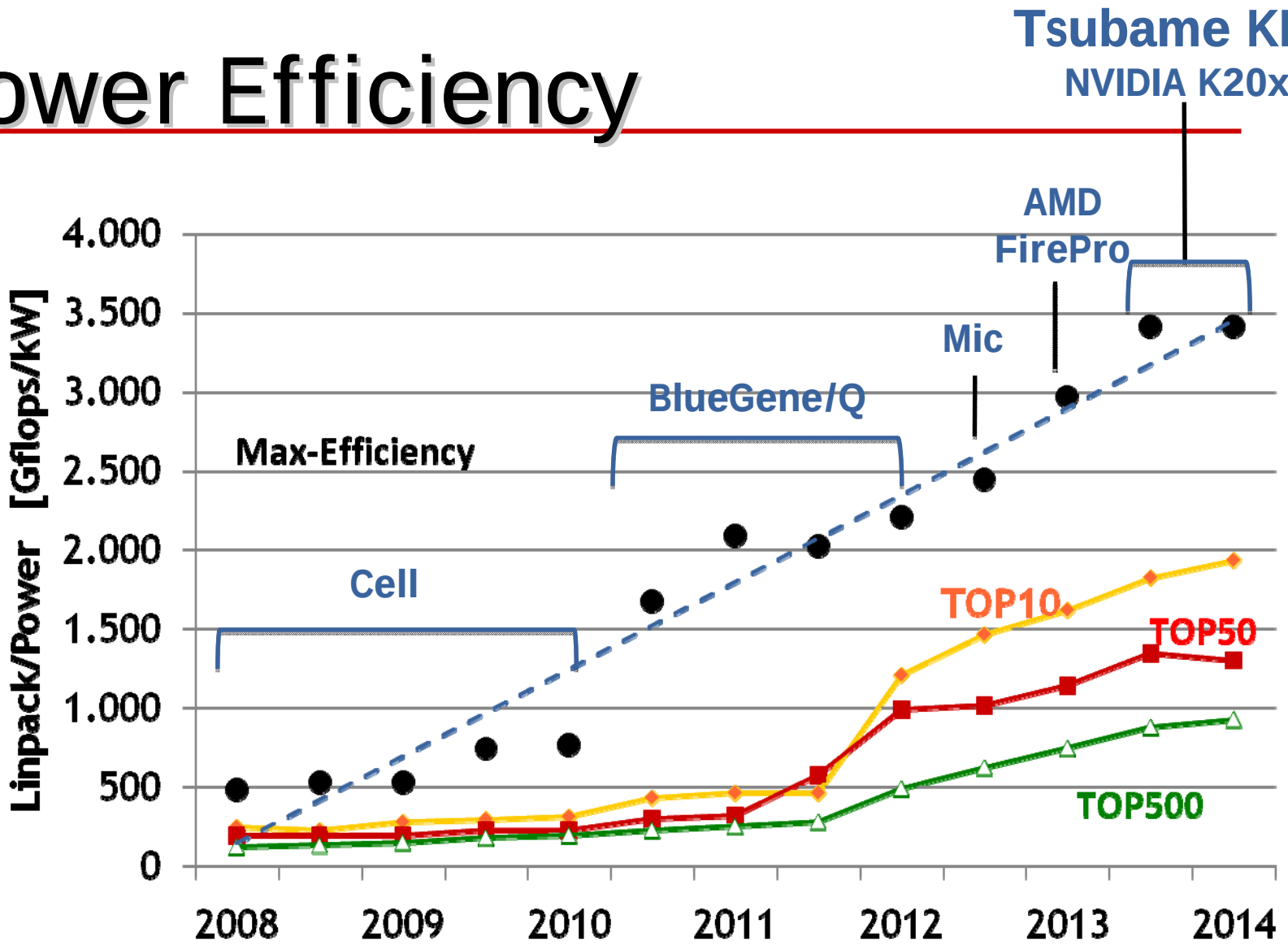
Performance Share of Accelerators



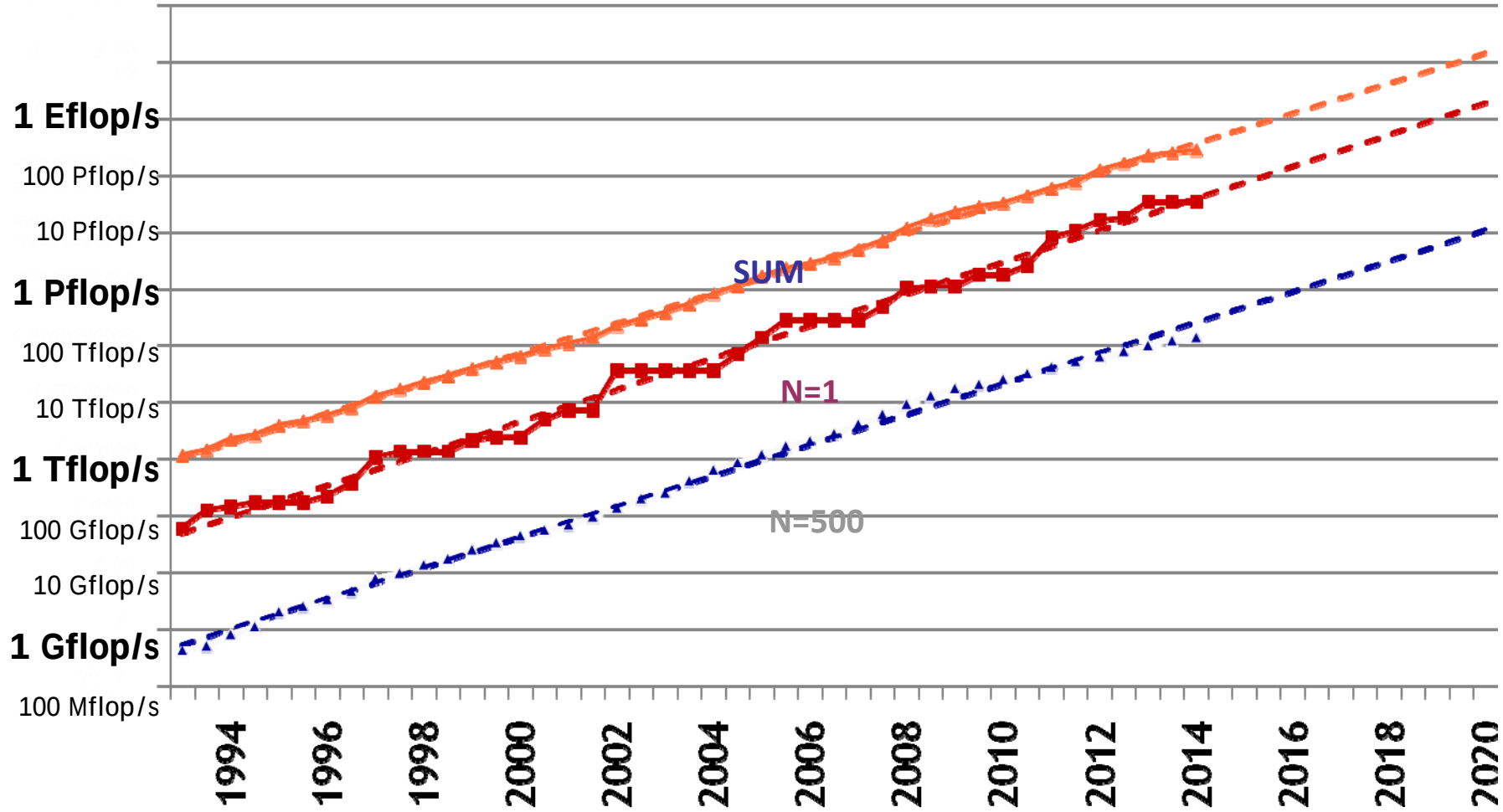
Power Efficiency



Power Efficiency



Projected Performance Development



LINPACK Benchmark (HPL) 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.

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

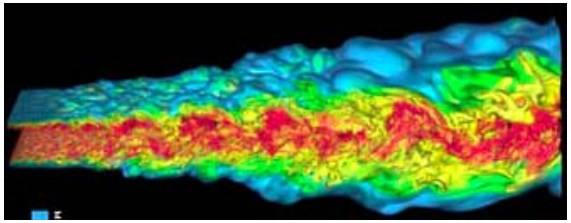
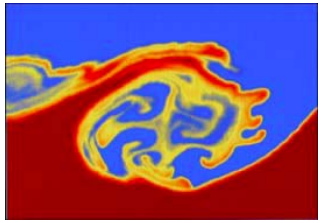
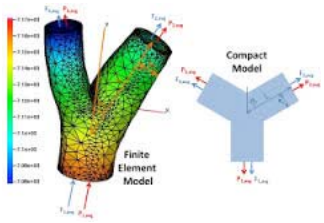
- LINPACK Benchmark is 37 years old
  - TOP500 (HPL) is 21.5 years old
- Floating point-intensive performs $O(n^3)$ floating point operations and moves $O(n^2)$ data.
- No longer so strongly correlated to real apps.
- Reports Peak Flops (although hybrid systems see only 1/2 to 2/3 of Peak)
- Encourages 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

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

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

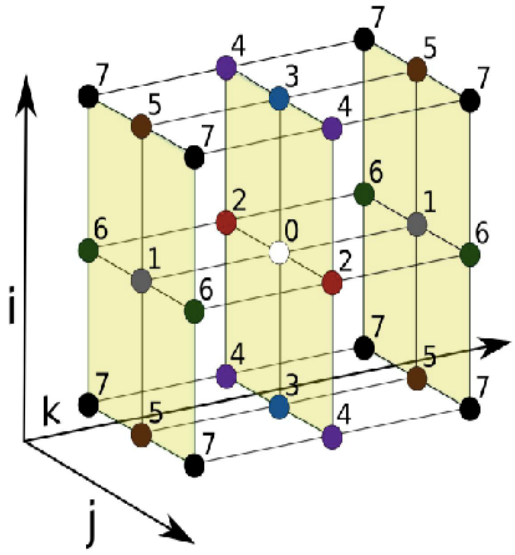
Proposal: HPCG

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



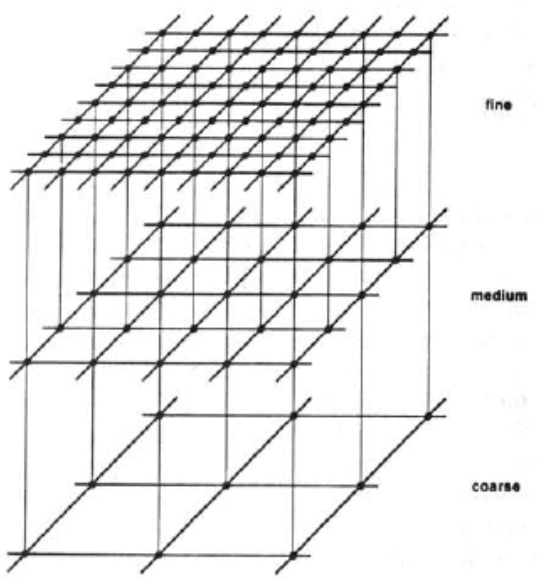
27-point stencil operator

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.

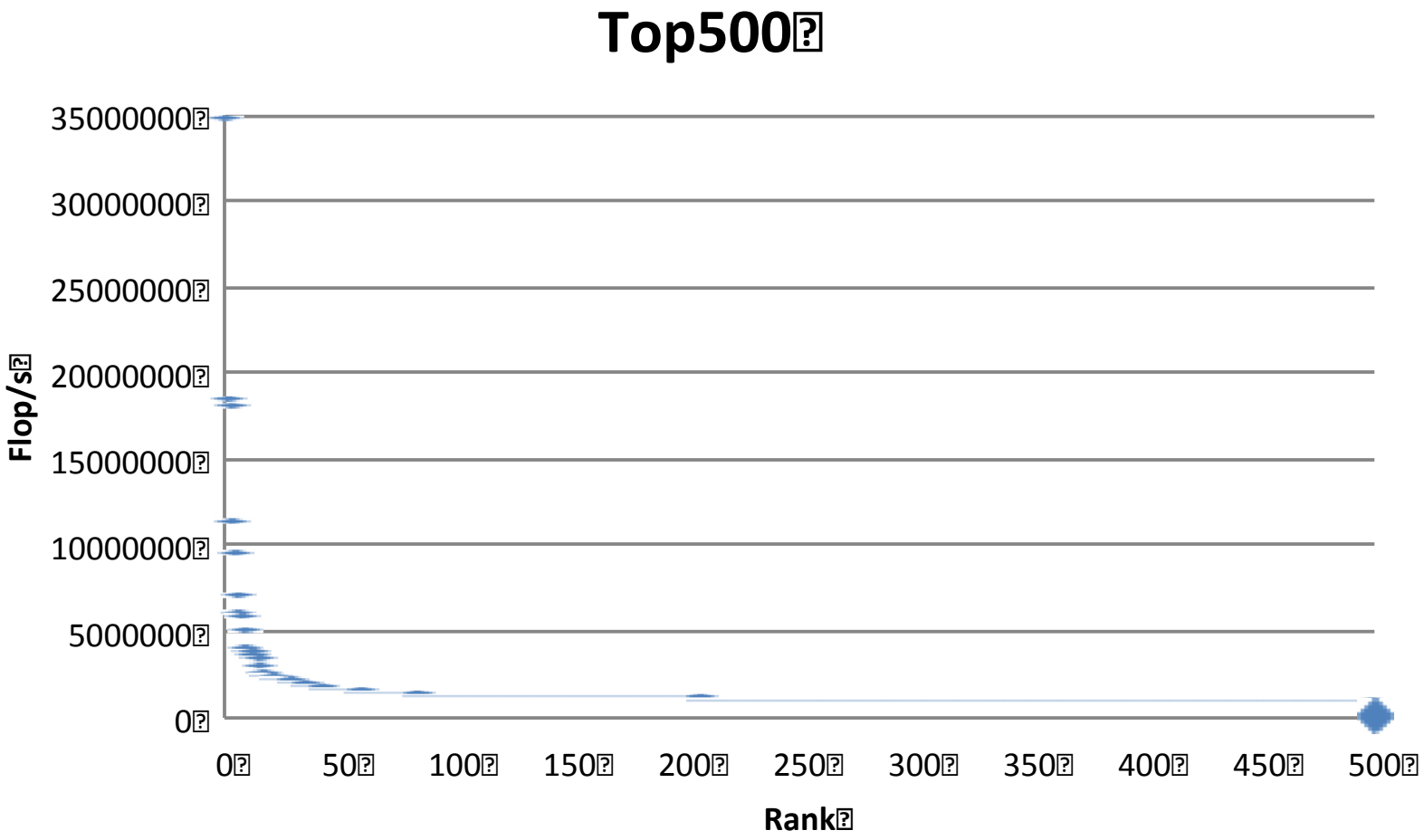
HPL

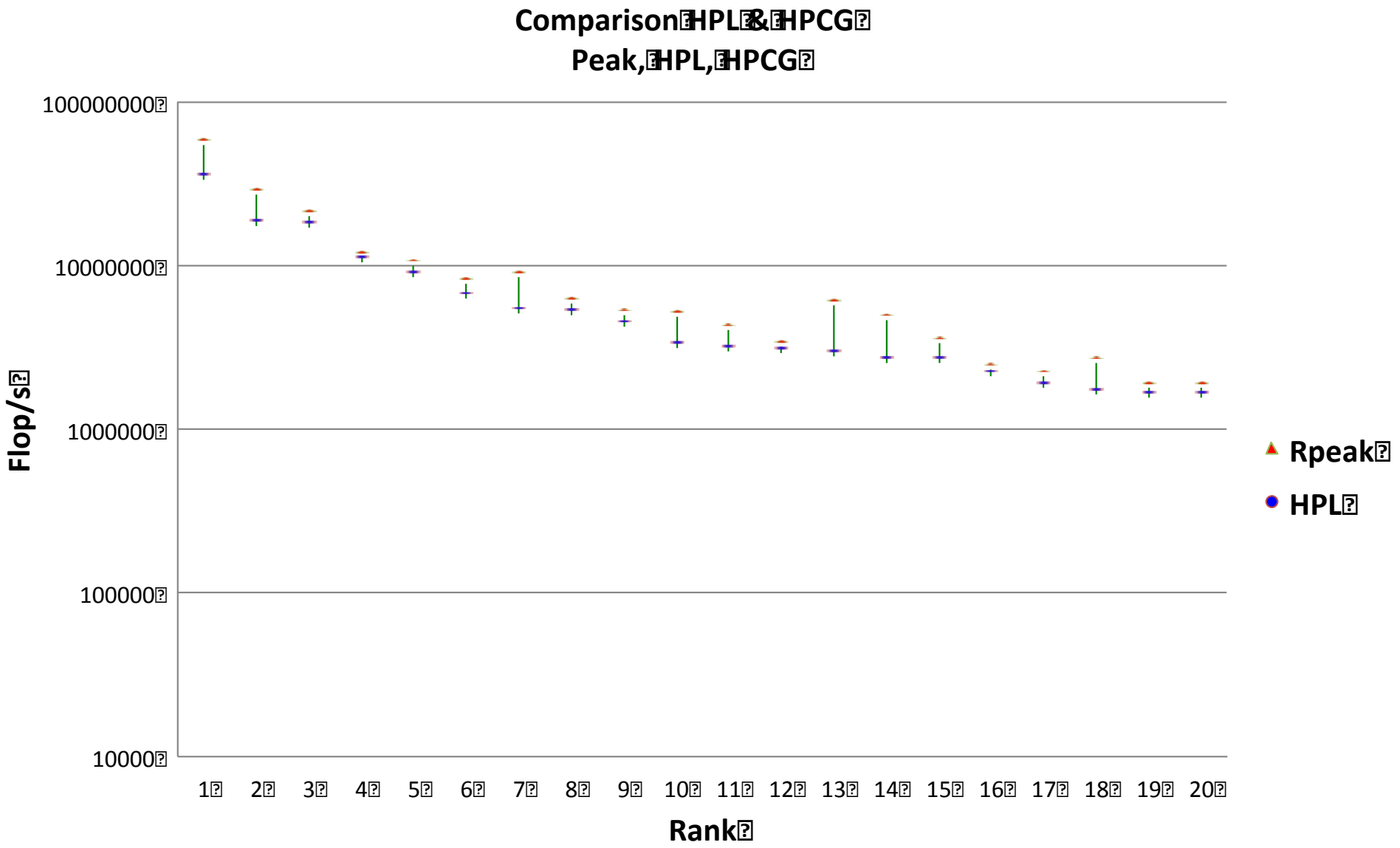
HPCG

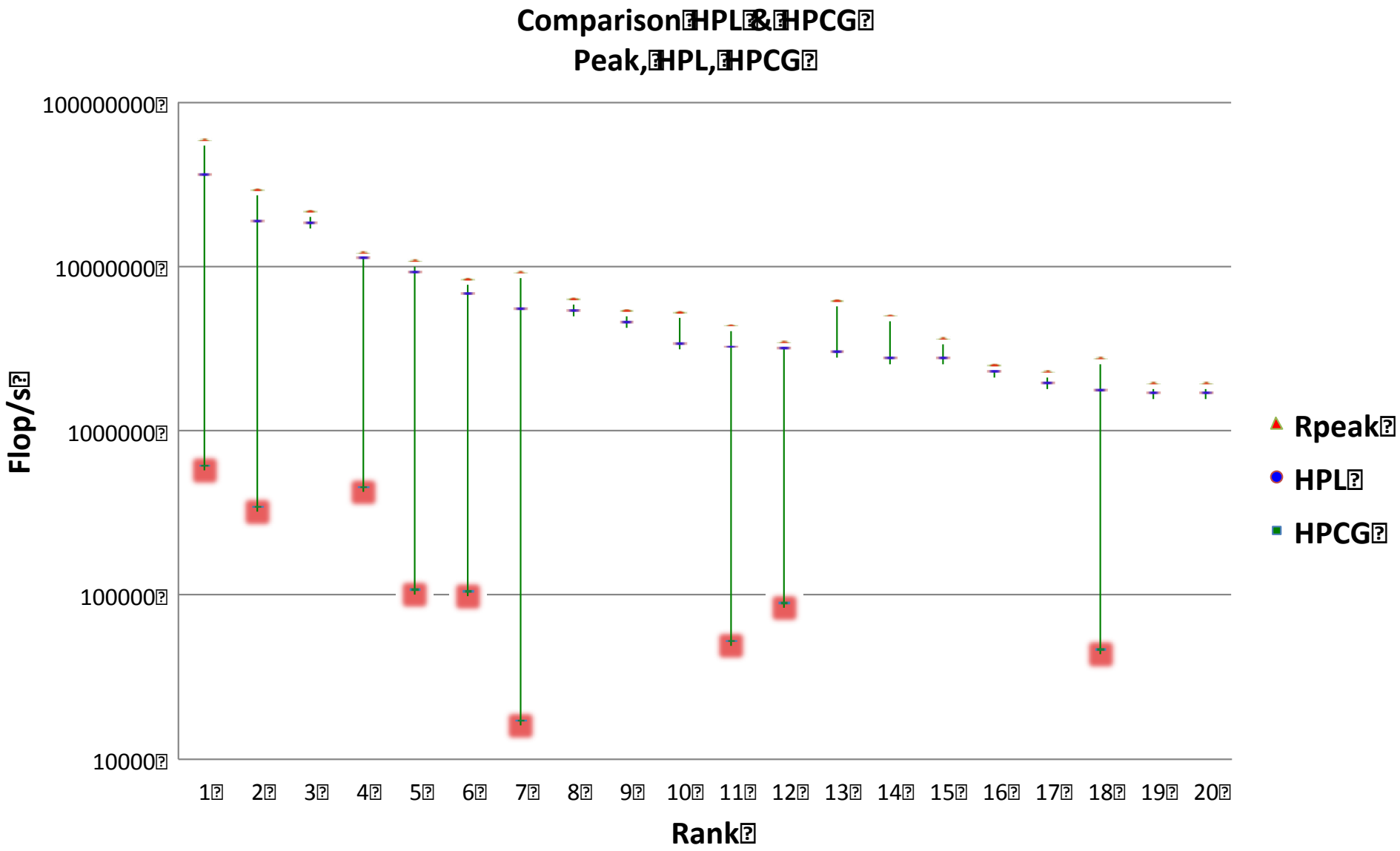
Top15

Site	Computer	Cores	HPL Rmax (Pflops)	HPL Rank	HPCG (Pflops)	HPCG/HPL
NSCC / Guangzhou	Tianhe-2 NUDT, Xeon 12C 2.2GHz + Intel Xeon Phi 57C + Custom	3,120,000	33.9	1	.580	1.7%
RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx 8C + Custom	705,024	10.5	4	.427	4.1%
DOE/OS Oak Ridge Nat Lab	Titan, Cray XK7 AMD 16C + Nvidia Kepler GPU 14C + Custom	560,640	17.6	2	.322	1.8%
DOE/OS Argonne Nat Lab	Mira BlueGene/Q, Power BQC 16C 1.60GHz + Custom	786,432	8.59	5	.101#	1.2%
Swiss CSCS	Piz Daint, Cray XC30, Xeon 8C + Nvidia Kepler 14C + Custom	115,984	6.27	6	.099	1.6%
Leibniz Rechenzentrum	SuperMUC, Intel 8C + IB	147,456	2.90	12	.0833	2.9%
CEA/TGCC-GENCI	Curie tine nodes Bullx B510 Intel Xeon 8C 2.7 GHz + IB	79,504	1.36	26	.0491	3.6%
Exploration and Production Eni S.p.A.	HPC2, Intel Xeon 10C 2.8 GHz + Nvidia Kepler 14C + IB	62,640	3.00	11	.0489	1.6%
DOE/OS L Berkeley Nat Lab	Edison Cray XC30, Intel Xeon 12C 2.4GHz + Custom	132,840	1.65	18	.0439 #	2.7%
Texas Advanced Computing Center	Stampede, Dell Intel (8c) + Intel Xeon Phi (61c) + IB	78,848	.881*	7	.0161	1.8%
Meteo France	Beaufix Bullx B710 Intel Xeon 12C 2.7 GHz + IB	24,192	.469 (.467*)	79	.0110	2.4%
Meteo France	Prolix Bullx B710 Intel Xeon 2.7 GHz 12C + IB	23,760	.464 (.415*)	80	.00998	2.4%
U of Toulouse	CALMIP Bullx DLC Intel Xeon 10C 2.8 GHz + IB	12,240	.255	184	.00725	2.8%
Cambridge U	Wilkes, Intel Xeon 6C 2.6 GHz + Nvidia Kepler 14C + IB	3584	.240	201	.00385	1.6%
TiTech	TUSBAME-KFC Intel Xeon 6C 2.1 GHz + IB	2720	.150	436	.00370	2.5%

* scaled to reflect the same number of cores
# unoptimized implementation

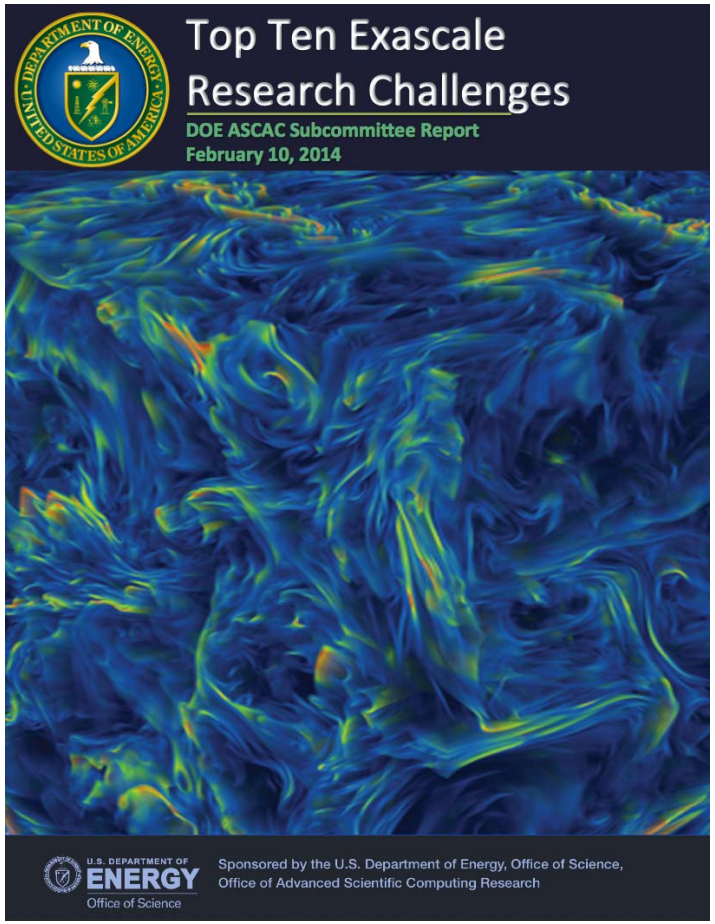

Top500
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
Flop/s
0
50
100
150
200
250
300
350
400
450
500
Rank


Comparison HPL & HPCG
Peak, HPL, HPCG
100000000
10000000
1000000
100000
10000
Flop/s
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Rank
Rpeak
HPL


Comparison HPL & HPCG
Peak, HPL, HPCG
100000000
10000000
1000000
100000
10000
Flop/s
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Rank
Rpeak
HPL
HPCG

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

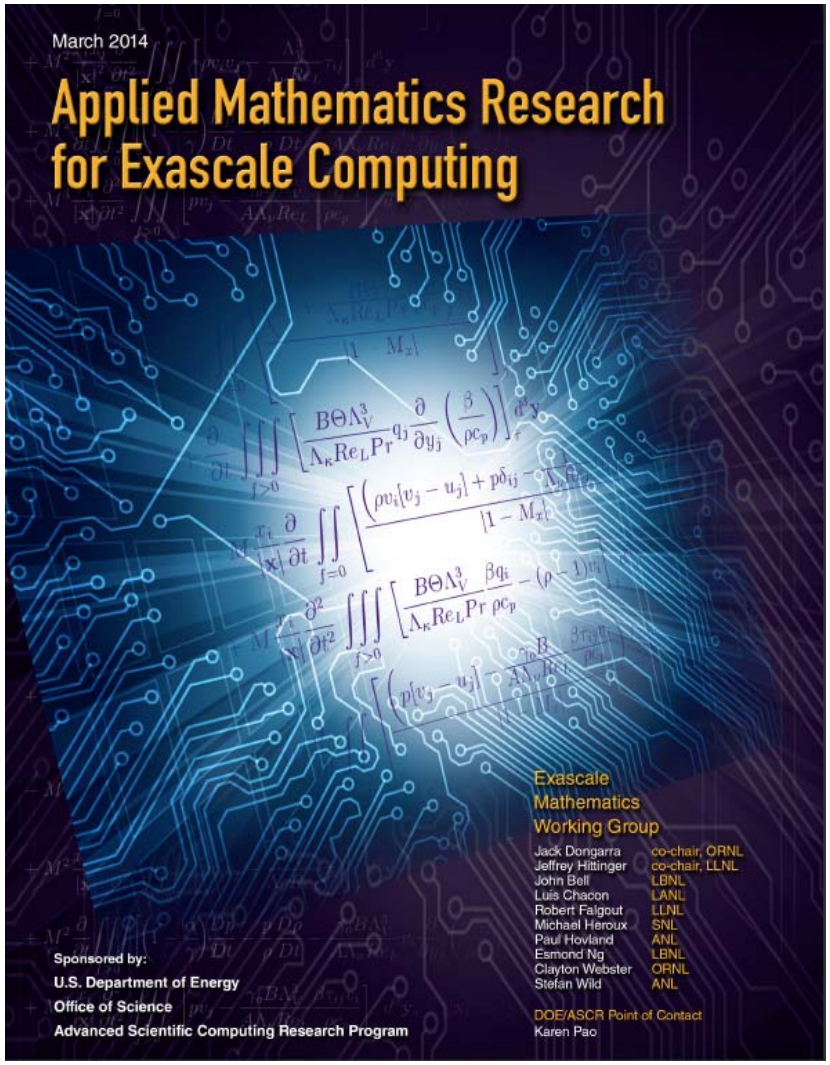
Subcommittee Chair
Robert Lucas (University of Southern California, Information Sciences Institute)

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

- **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.
- **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.
- **Scientific productivity:**
  - Increasing the productivity of computational scientists with new software engineering tools and environments.

Applied Math @ Exascale

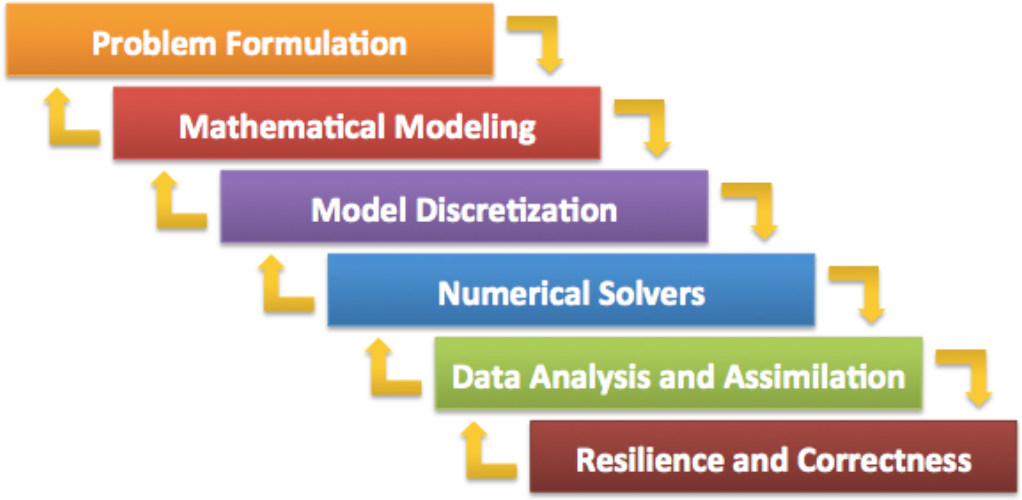


Google "doe applied math exascale"

Typical Software Stack

- Low-level Software
- Operating System
- Runtime System
- Compilers
- Performance Analysis Tools
- Scalable Libraries
- Visualization Tools
- Data and Data Analytics
- File System
- Networks

Applied Mathematics Stack



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**