

TOP500 BOF Report

The TOP500 BOF followed its traditional structure.

Certificates for the top ranked system were handed out at the beginning of the BOF and short statements given by the system owners.

An in-depth analysis of the TOP500 dataset was presented. This year for the first time an analysis of the size distributions based on Gini coefficients was discussed.

As in previous years space was allocated for discussion of TOP500 related subjects. This year the new HPCG benchmark was presented and its potential merits were discussed.

The BOF concluded with a Q&A session and was very well attended.

Contact: Erich Strohmaier, erich@top500.org, 510-495-2517

China's Tianhe-2 Supercomputer Maintains

Top Spot on 42nd TOP500 List

MANNHEIM, Germany; BERKELEY, Calif.; and KNOXVILLE, Tenn.—Tianhe-2, a supercomputer developed by China's National University of Defense Technology, retained its position as the world's No. 1 system with a performance of 33.86 petaflop/s (quadrillions of calculations per second) on the Linpack benchmark, according to the 42nd edition of the twice-yearly TOP500 list of the world's most powerful supercomputers. The list was announced Nov. 18 at the SC13 conference in Denver, Colo.

Titan, a Cray XK7 system installed at the Department of Energy's (DOE) Oak Ridge National Laboratory, remains the No. 2 system. It achieved 17.59 Pflop/s on the Linpack benchmark. Titan is one of the most energy efficient systems on the list consuming a total of 8.21 MW and delivering 2.143 gigaflops/W.

Sequoia, an IBM BlueGene/Q system installed at DOE's Lawrence Livermore National Laboratory, is again the No. 3 system. It was first delivered in 2011 and achieved 17.17 Plop/s on the Linpack benchmark.

Fujitsu's K computer installed at the RIKEN Advanced Institute for Computational Science (AICS) in Kobe, Japan, is the No. 4 system with 10.51 Pflop/s on the Linpack benchmark.

Mira, a BlueGene/Q system installed at DOE's Argonne National Laboratory, is No. 5 with 8.59 Plop/s on the Linpack benchmark.

The new entry in the TOP10 is at No. 6 –Piz Daint, a Cray XC30 system installed at the Swiss National Supercomputing Centre (CSCS) in Lugano, Switzerland and now the most powerful system in Europe. Piz Daint achieved 6.27 Pflop/s on the Linpack benchmark. Piz Daint is also the most energy efficient system in the TOP10 consuming a total of 2.33 MW and delivering 2.7 Gflops/W.

Rounding out the TOP10 are Stampede at the Texas Advanced Computing Center of the University of Texas, Austin, which slipped to No. 7; a BlueGene/Q system called JUQUEEN installed at the Forschungszentrum Juelich in Germany is No. 8; No. 9 is taken by Vulcan, another IBM BlueGene/Q system at Lawrence Livermore National Laboratory; and No. 10 is the third system in Europe, the SuperMUC, installed at Leibniz Rechenzentrum in Germany.

The total combined performance of all 500 systems on the list is 250 Pflop/s. Half of the total performance is achieved by the top 17 systems on the list, with the other half of total performance spread among the remaining 483 systems.

Other highlights from the November 2013 TOP500 List, which can be found at www.top500.org, include:

- In all, there are 31 systems with performance greater than a petaflop/s on the list, an increase of five compared to the June 2013 list.
- The No. 1 system, Tianhe-2, and the No. 7 system, Stampede, are using Intel Xeon Phi processors to speed up their computational rate. The No. 2 system Titan and the No. 6 system Piz Daint are using NVIDIA GPUs to accelerate computation.
- A total of 53 systems on the list are using accelerator/co-processor technology, unchanged from June 2013. Thirty-eight (38) of these use NVIDIA chips, two use ATI Radeon, and there are now 13 systems with Intel MIC technology (Xeon Phi).
- Intel continues to provide the processors for the largest share (82.4 percent) of TOP500 systems.
- Ninety-four percent of the systems use processors with six or more cores and 75 percent have processors with eight or more cores.
- The number of systems installed in China has now stabilized at 63, compared with 65 on the last list. China occupies the No. 2 position as a user of HPC, behind the U.S. but ahead of Japan, UK, France, and Germany. Due to Tianhe-2, China this year also took the No. 2 position in the performance share, topping Japan.
- The last system on the newest list was listed at position 363 in the previous TOP500.

Geographical observations

- The U.S. is clearly the leading consumer of HPC systems with 265 of the 500 systems (253 last time). The European share (102 systems compared to 112 last time) is still lower than the Asian share (115 systems, down from 118 last time).
- Dominant countries in Asia are China with 63 systems (down from 65) and Japan with 28 systems (down from 30).
- In Europe, UK, France, and Germany, are almost equal with 23, 22, and 20 respectively.

About the TOP500 List

The first version of what became today's TOP500 list started as an exercise for a small conference in Germany in June 1993. Out of curiosity, the authors decided to revisit the list in November 1993 to see how things had changed. About that time they realized they might be on to something and decided to continue compiling the list, which is now a much-anticipated, much-watched and much-debated twice-yearly event.

The TOP500 list is compiled by Hans Meuer of the University of Mannheim, Germany; Erich Strohmaier and Horst Simon of Lawrence Berkeley National Laboratory; and Jack Dongarra of the University of Tennessee, Knoxville.



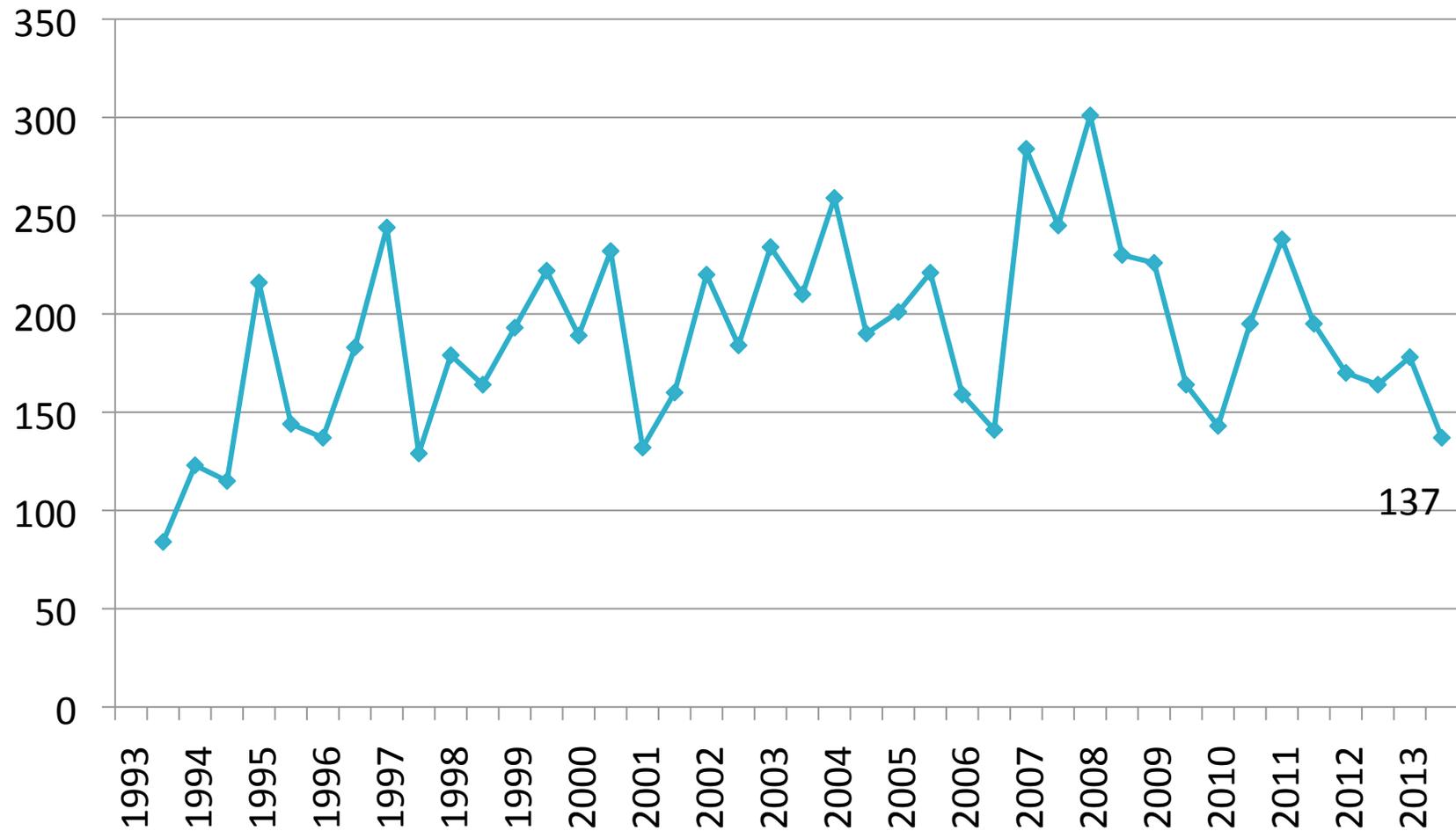
Highlights of the 42nd TOP500 List

Erich Strohmaier, LBNL

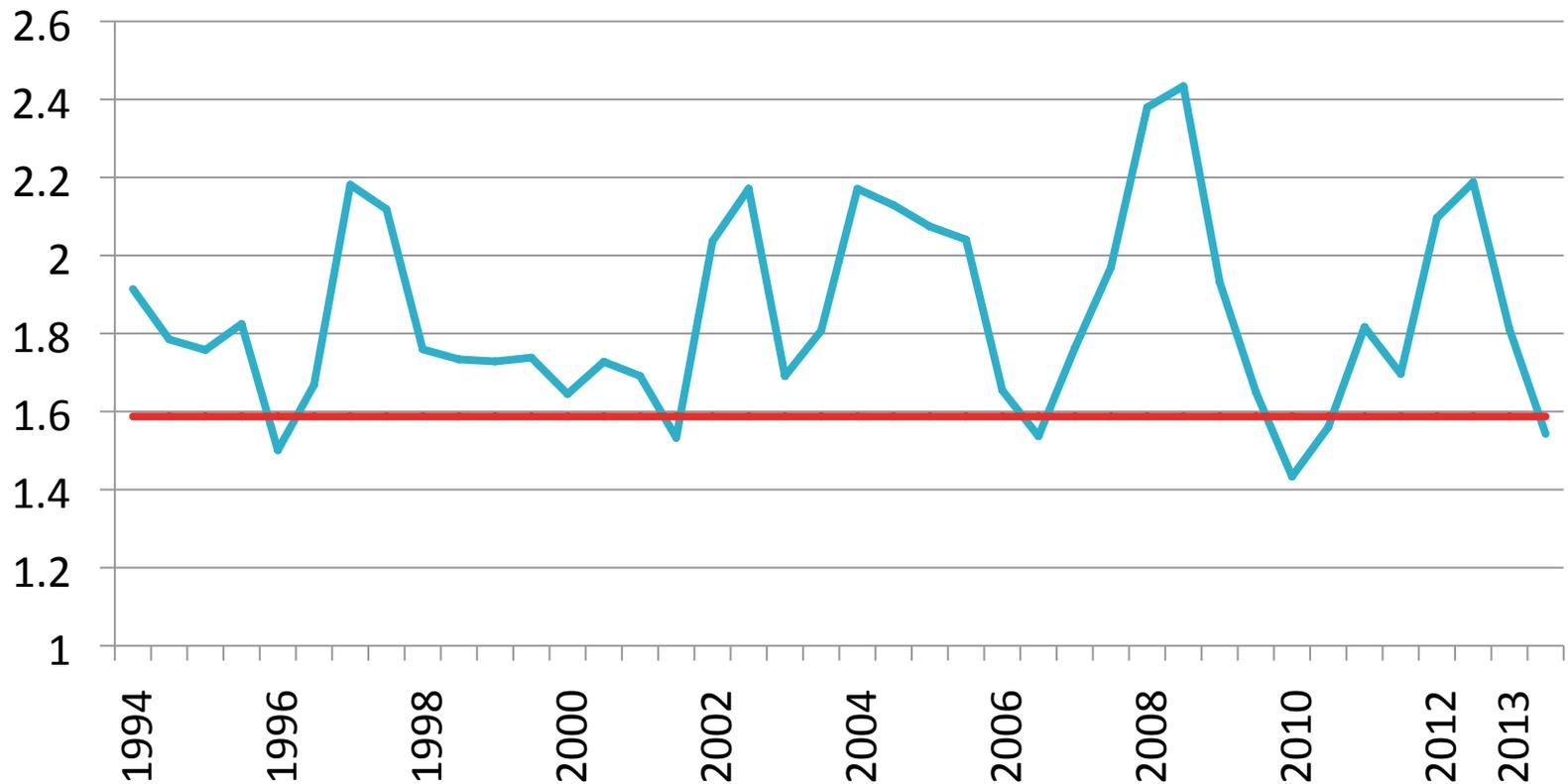
SC13, Denver, CO

#	Site	Manufacturer	Computer	Country	Cores	Rmax [Pflops]	Power [MW]
1	National University of Defense Technology	NUDT	Tianhe-2 NUDT TH-IVB-FEP, Xeon 12C 2.2GHz, IntelXeon Phi	China	3,120,000	33.9	17.8
2	Oak Ridge National Laboratory	Cray	Titan Cray XK7, Opteron 16C 2.2GHz, Gemini, NVIDIA K20x	USA	560,640	17.6	8.21
3	Lawrence Livermore National Laboratory	IBM	Sequoia BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	1,572,864	17.2	7.89
4	RIKEN Advanced Institute for Computational Science	Fujitsu	K Computer SPARC64 VIIIfx 2.0GHz, Tofu Interconnect	Japan	795,024	10.5	12.7
5	Argonne National Laboratory	IBM	Mira BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	786,432	8.59	3.95
6	Swiss National Supercomputing Centre (CSCS)	Cray	Piz Daint Cray XC30, Xeon E5 8C 2.6GHz, Aries, NVIDIA K20x	Switzerland	115,984	6.27	2.33
7	Texas Advanced Computing Center/UT	Dell	Stampede PowerEdge C8220, Xeon E5 8C 2.7GHz, Intel Xeon Phi	USA	462,462	5.17	4.51
8	Forschungszentrum Juelich (FZJ)	IBM	JuQUEEN BlueGene/Q, Power BQC 16C 1.6GHz, Custom	Germany	458,752	5.01	2.30
9	Lawrence Livermore National Laboratory	IBM	Vulcan BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	393,216	4.29	1.97
10	Leibniz Rechenzentrum	IBM	SuperMUC iDataPlex DX360M4, Xeon E5 8C 2.7GHz, Infiniband FDR	Germany	147,456	2.90	3.52

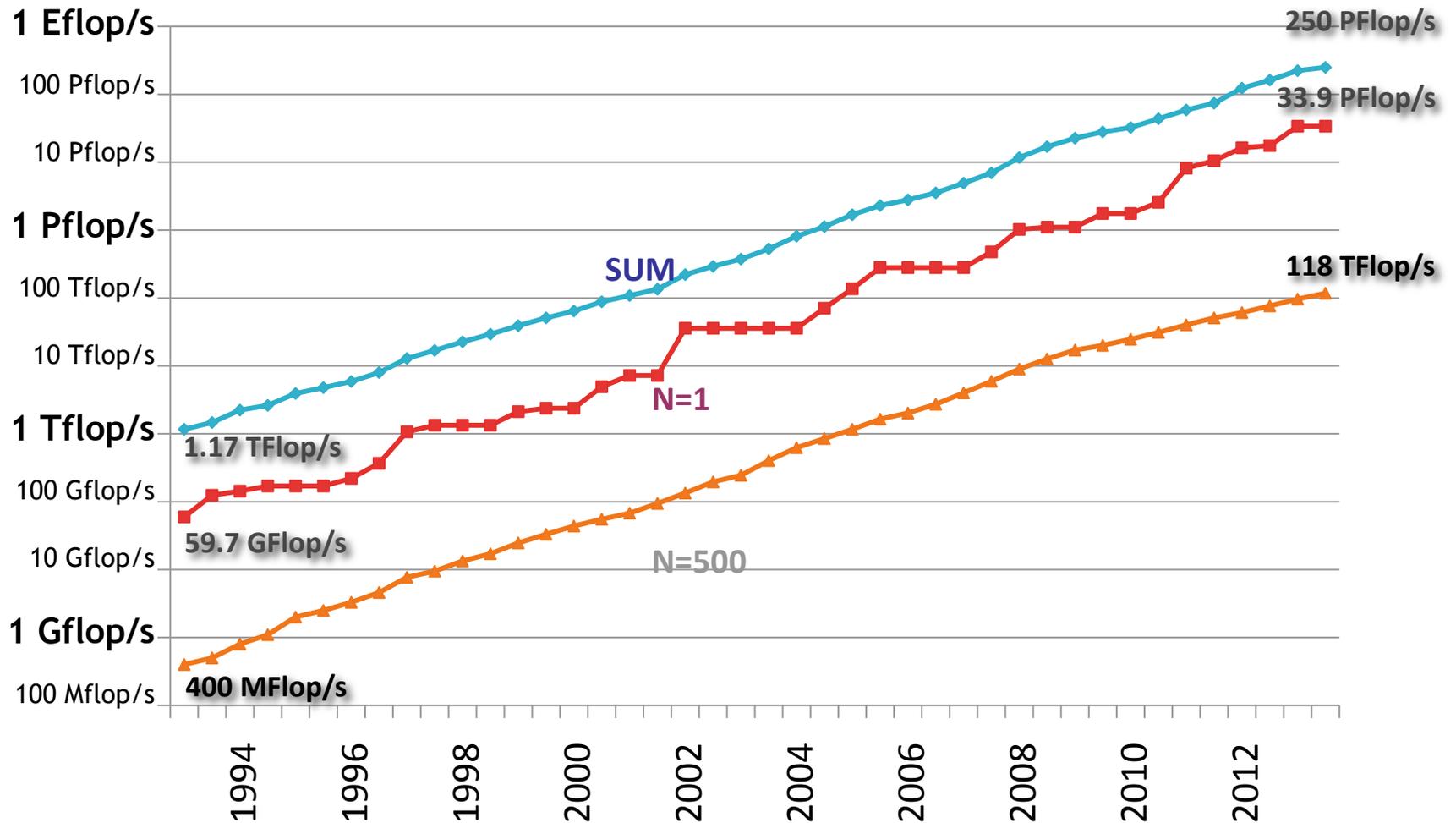
Replacement Rate



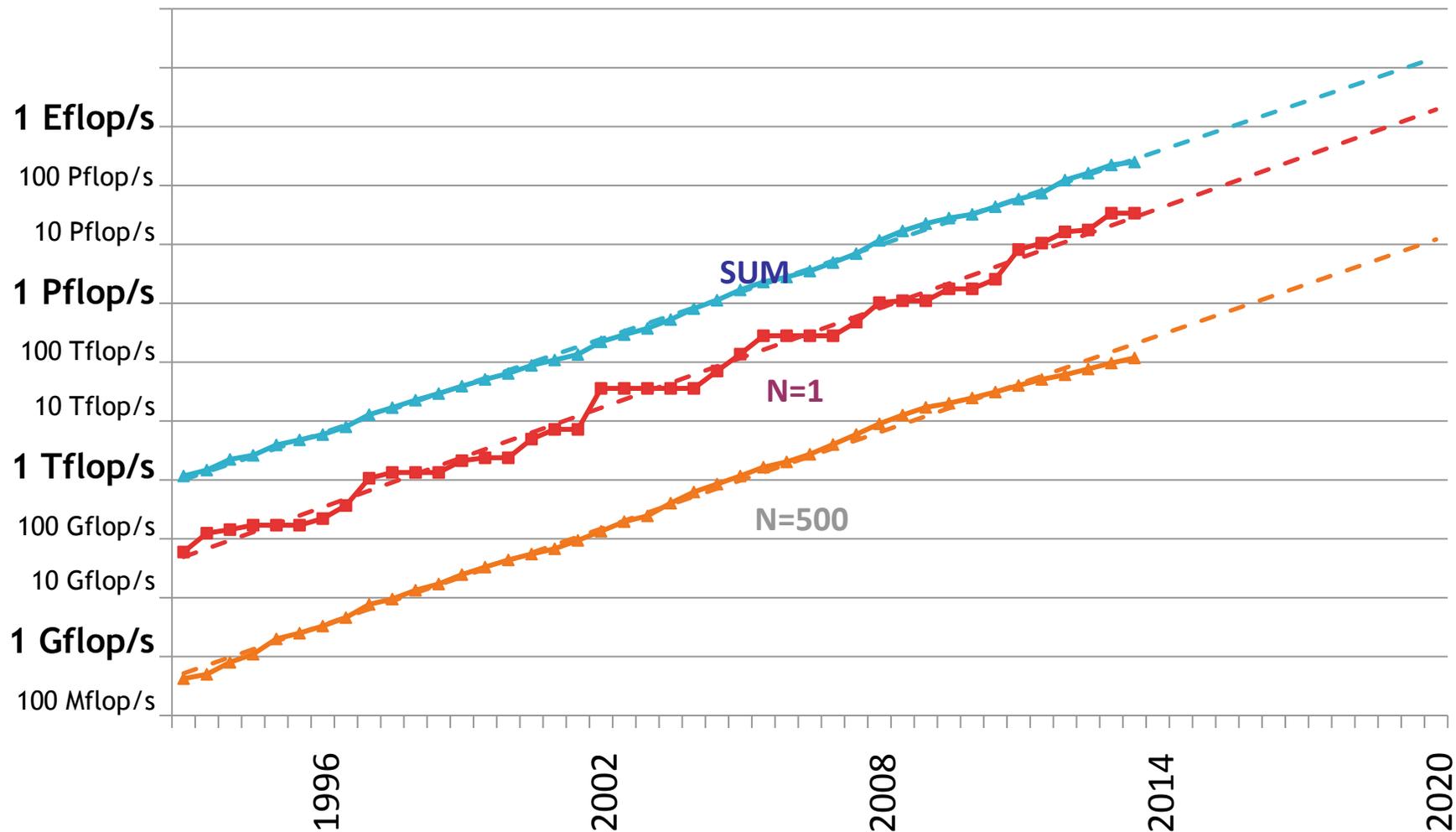
Annual Performance Increase of the TOP500



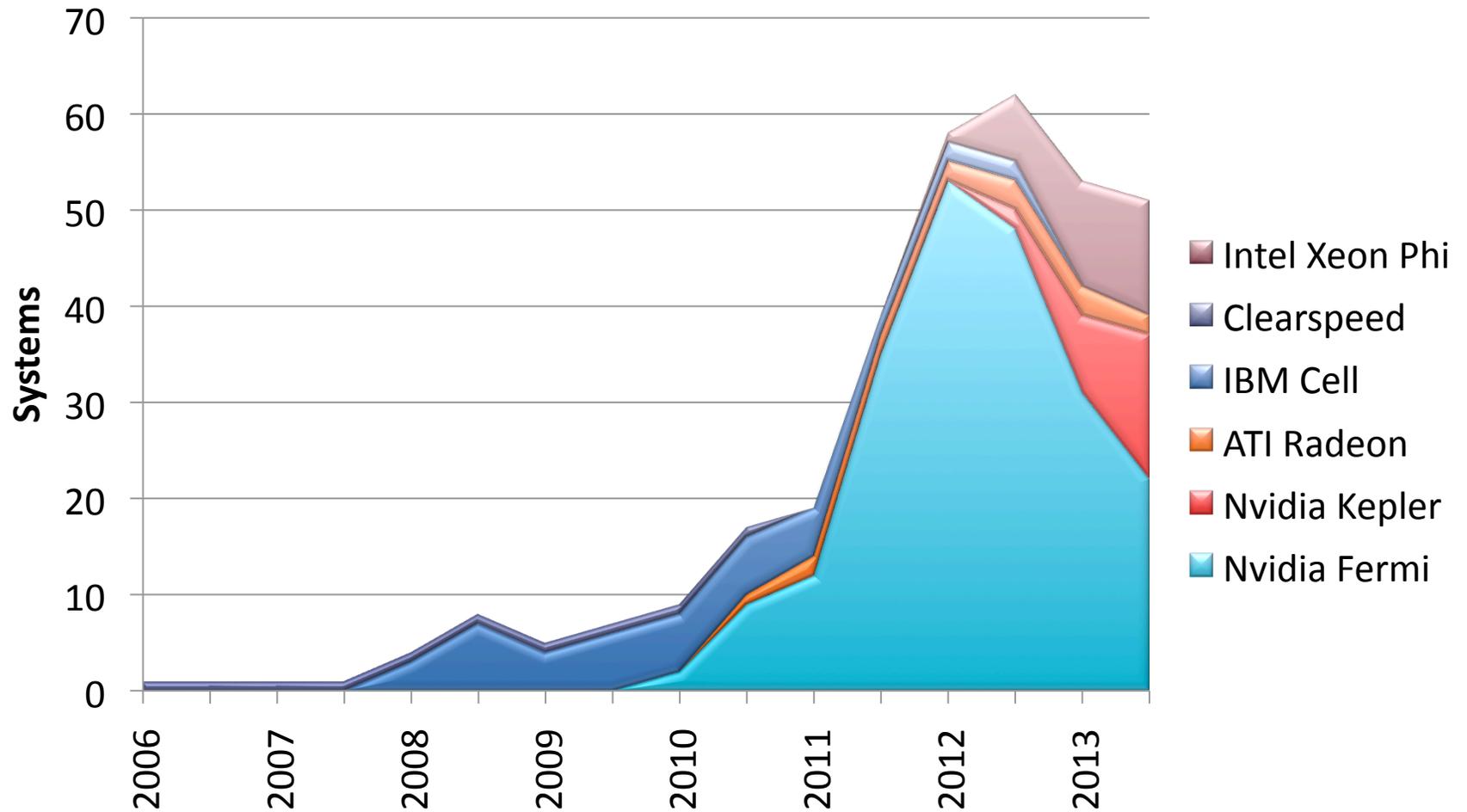
Performance Development



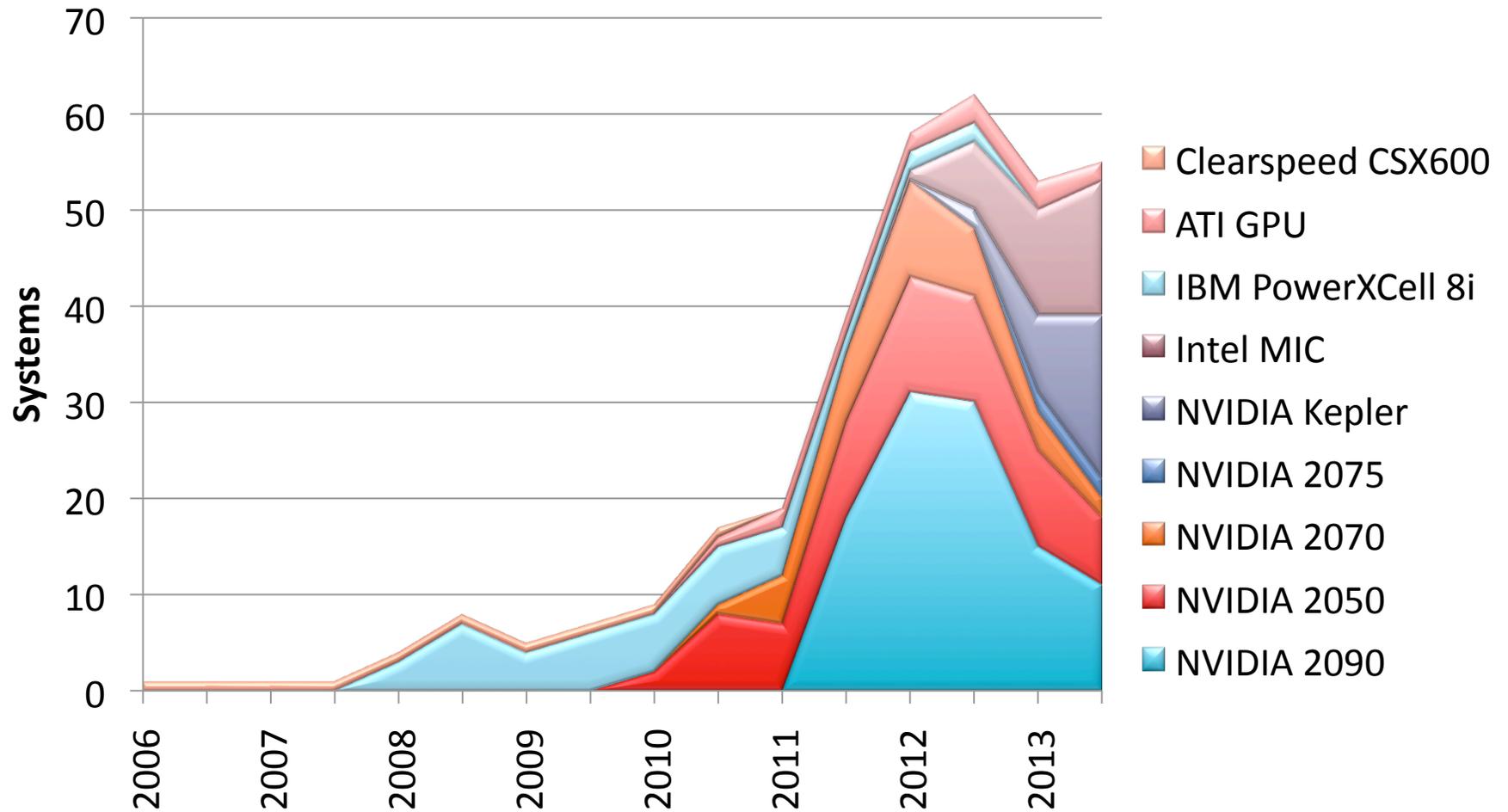
Projected Performance Development



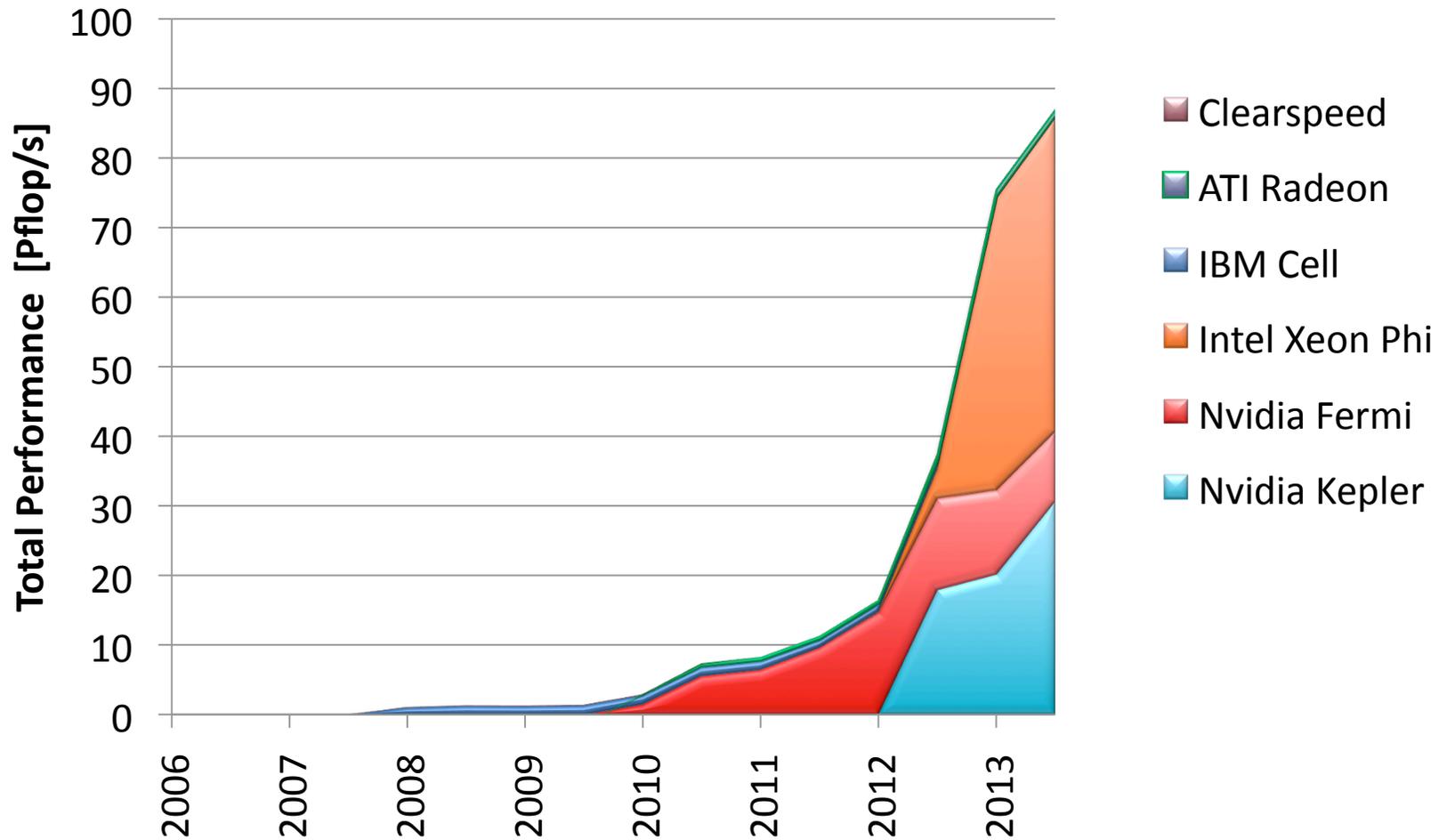
Accelerators



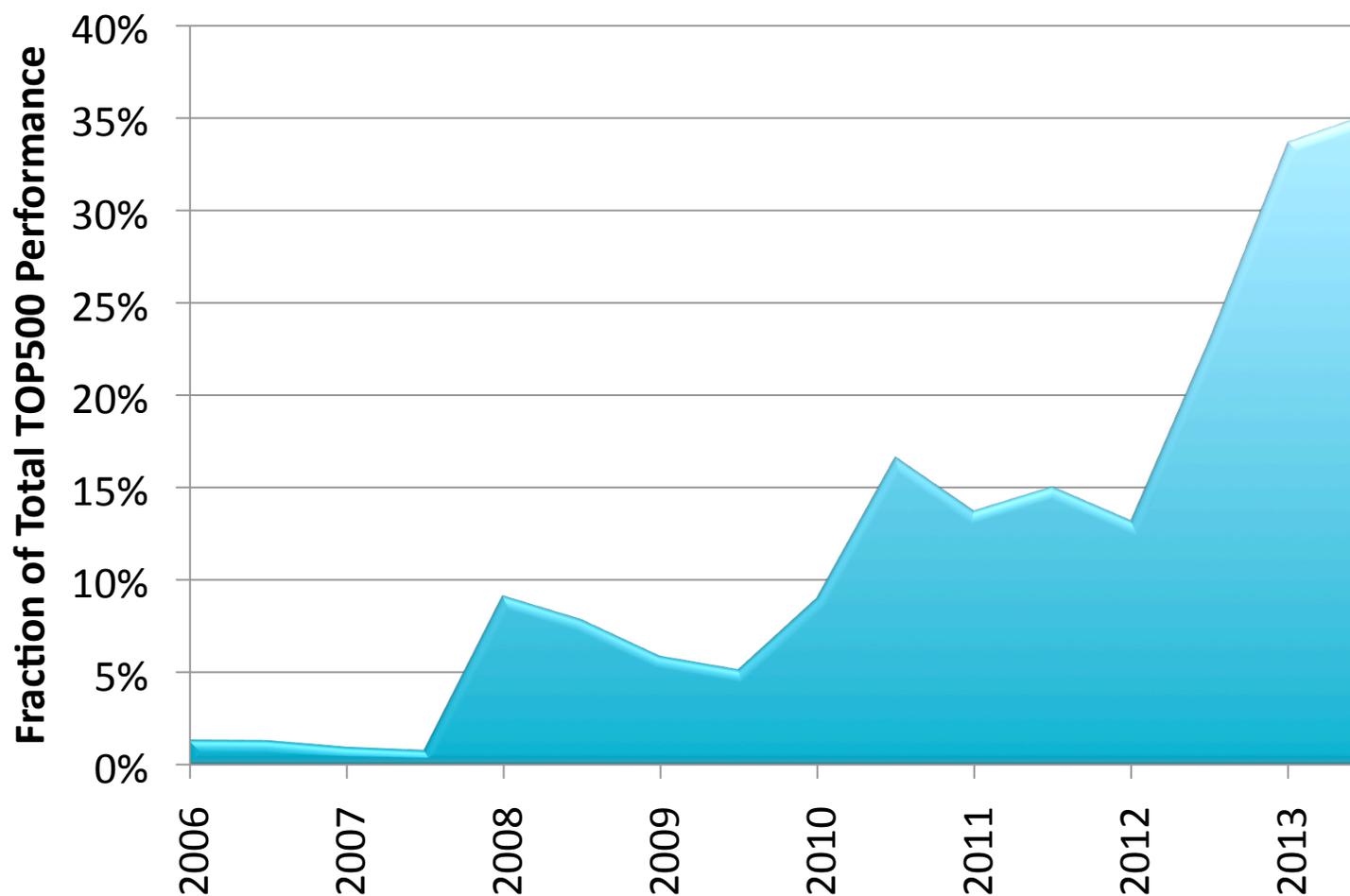
Accelerators



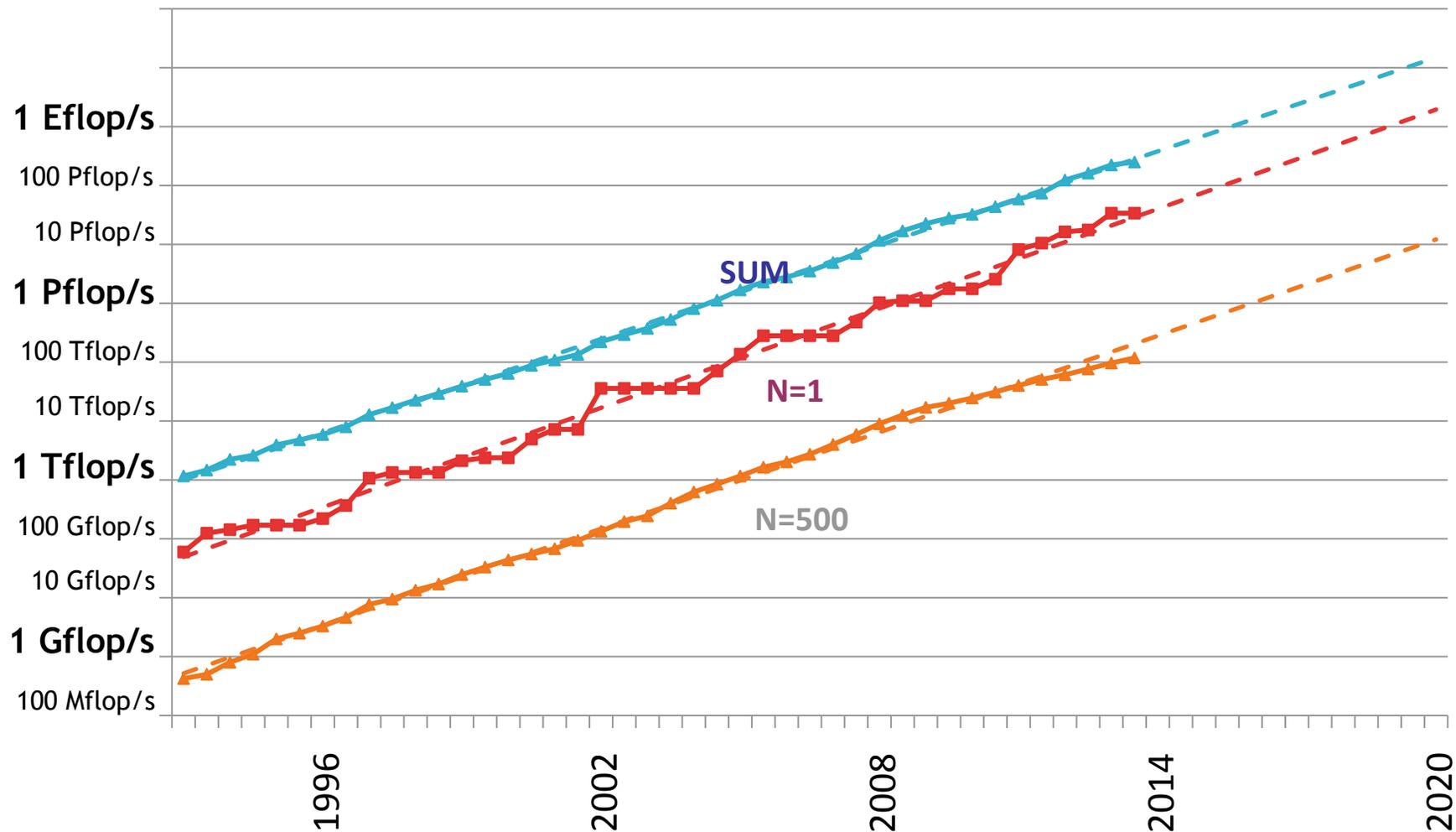
Performance of Accelerators



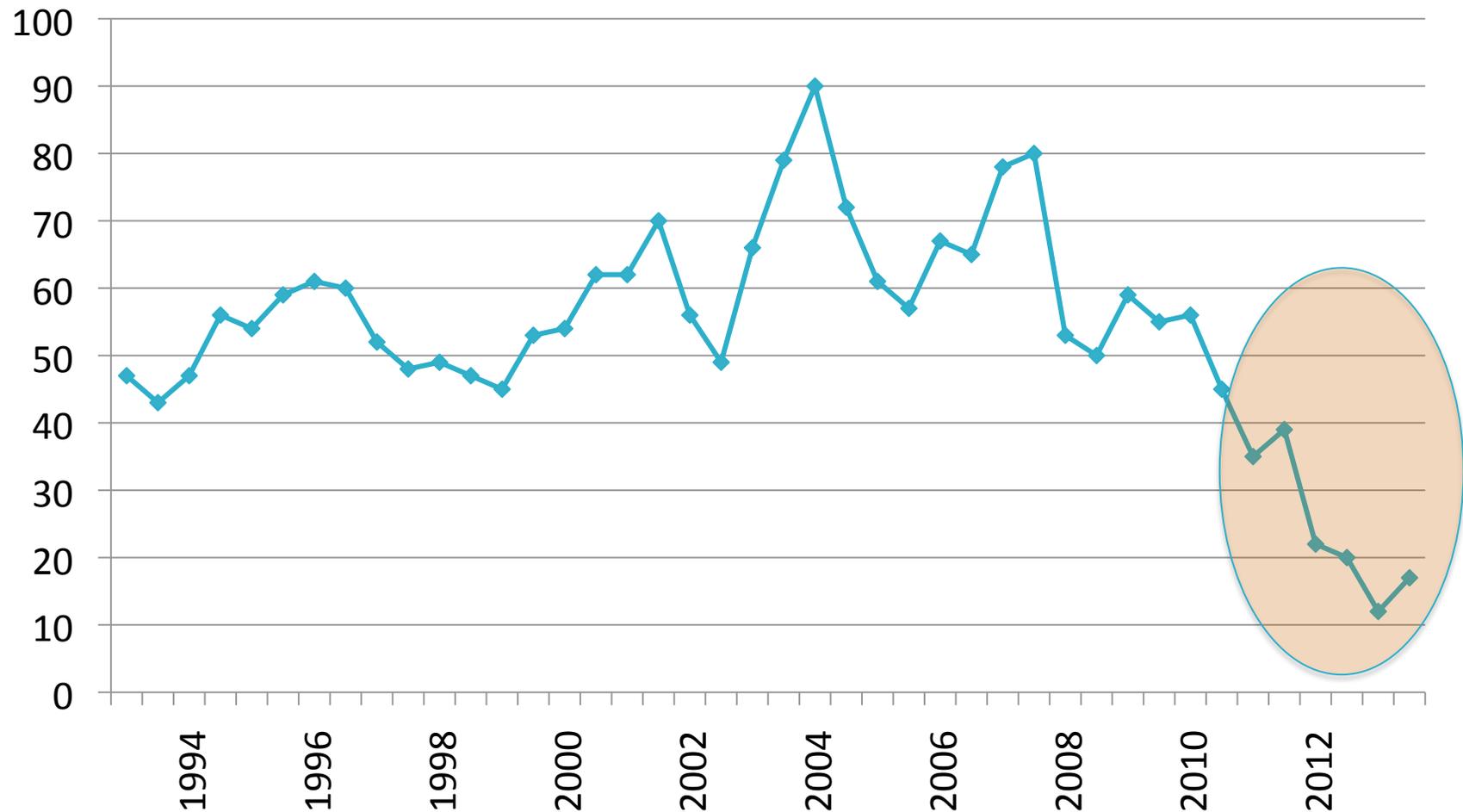
Performance Share of Accelerators



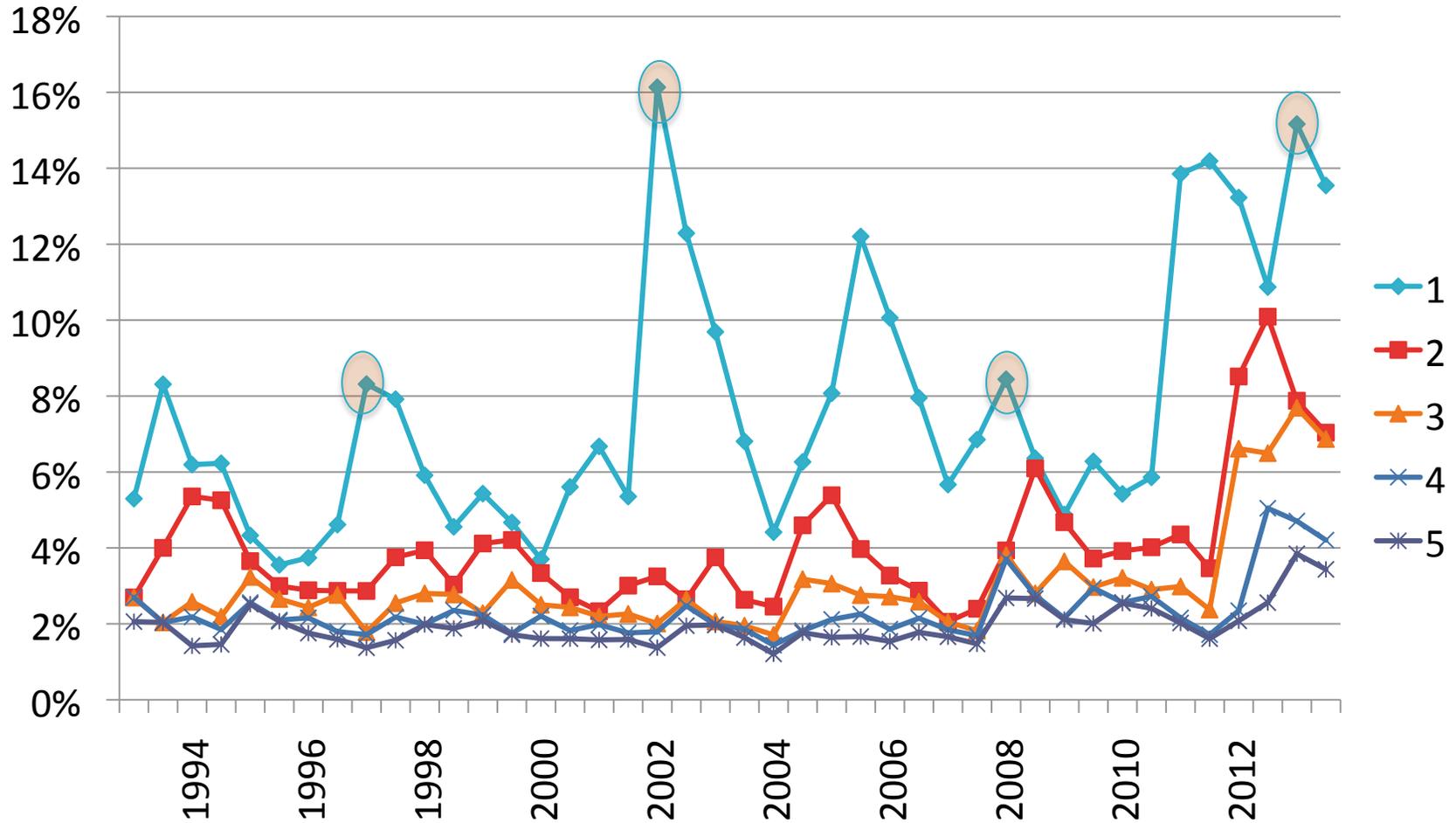
Projected Performance Development



Rank at which Half of total Performance is accumulated

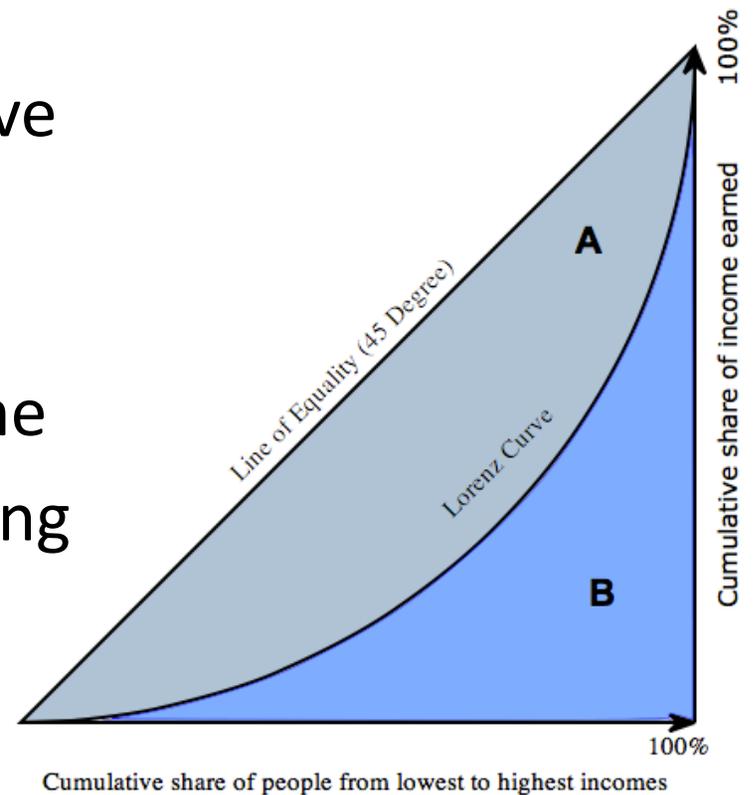


Performance Fraction of TOP5 Systems



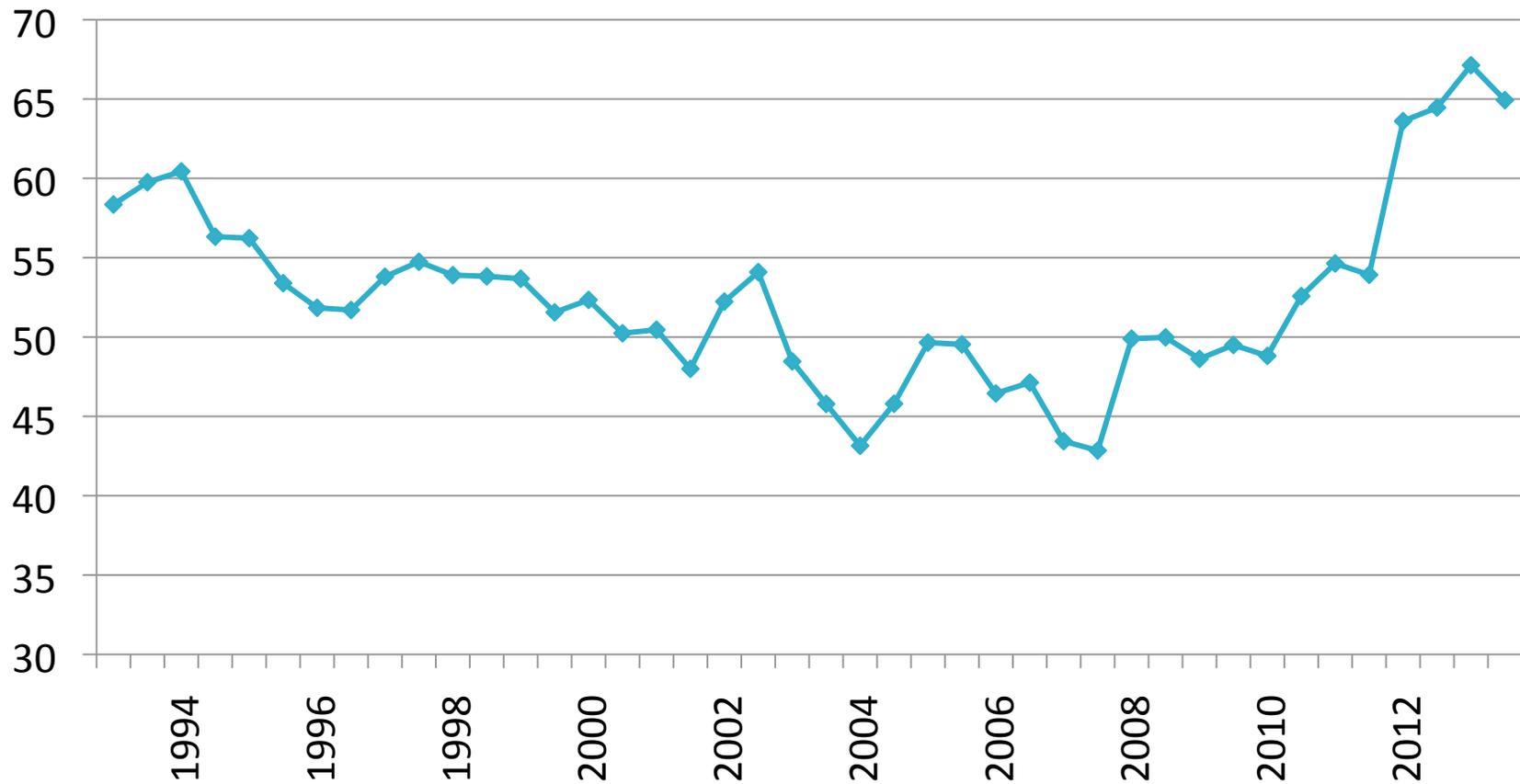
Gini Coefficient

- A measure of statistical dispersion intended to represent inequality
 - Area A above the Lorenz curve (cumulative distribution)
 - $Gini = A/(A+B)$
 - 0: All members have the same
 - 1: One member has everything



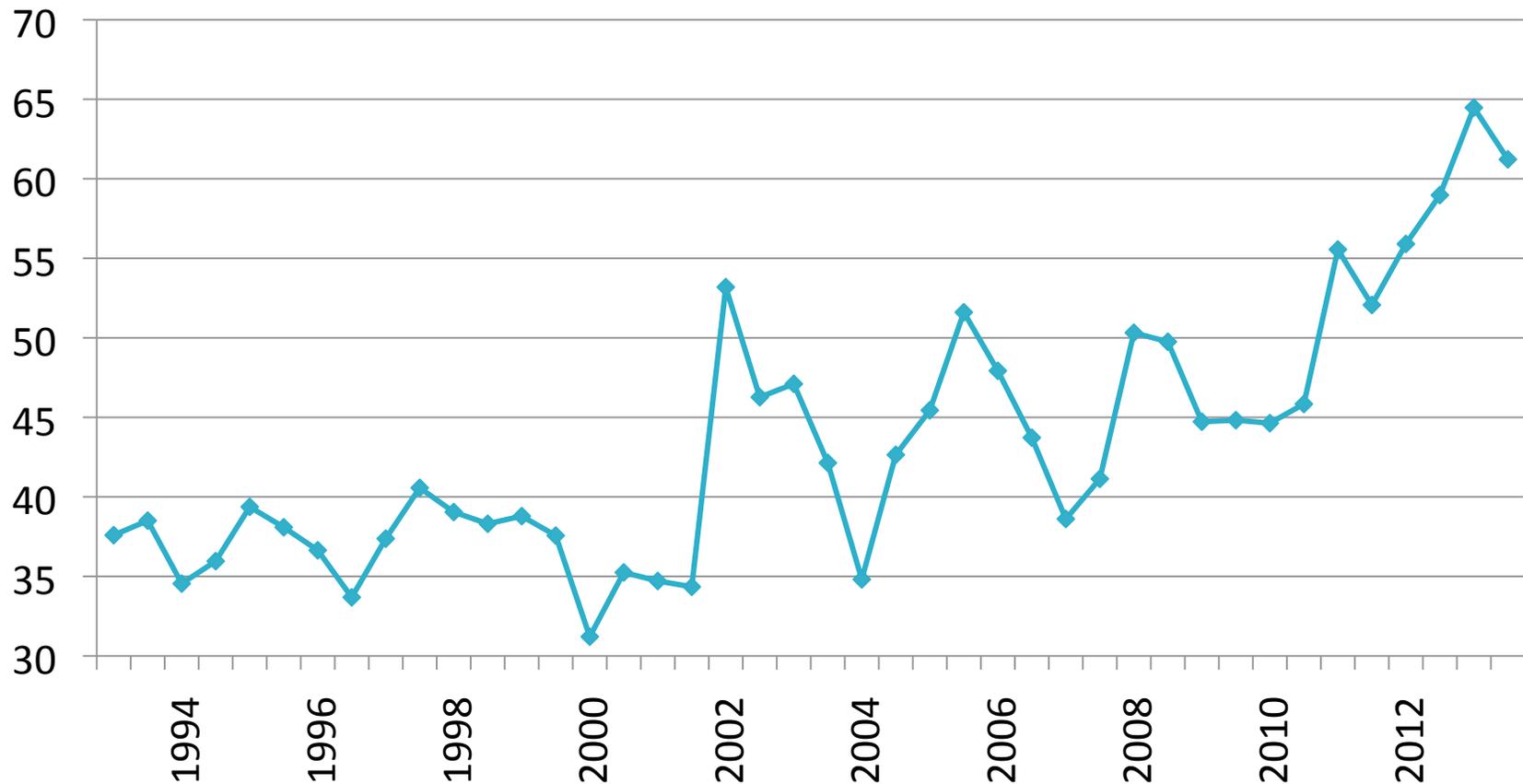
Gini Coefficient of the TOP500

Gini

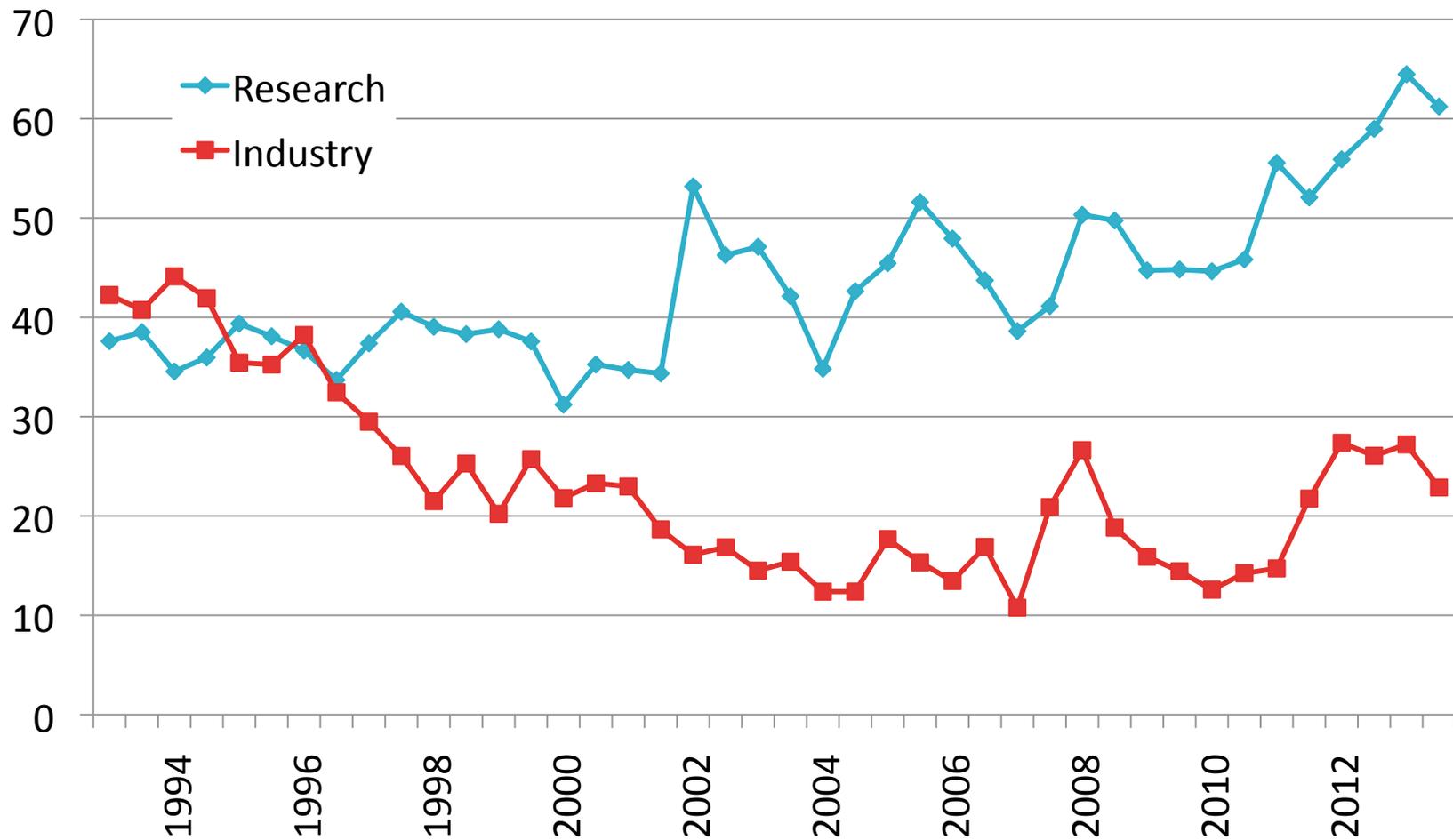


Gini Coefficient of the TOP50 Research/Academic/Classified Systems

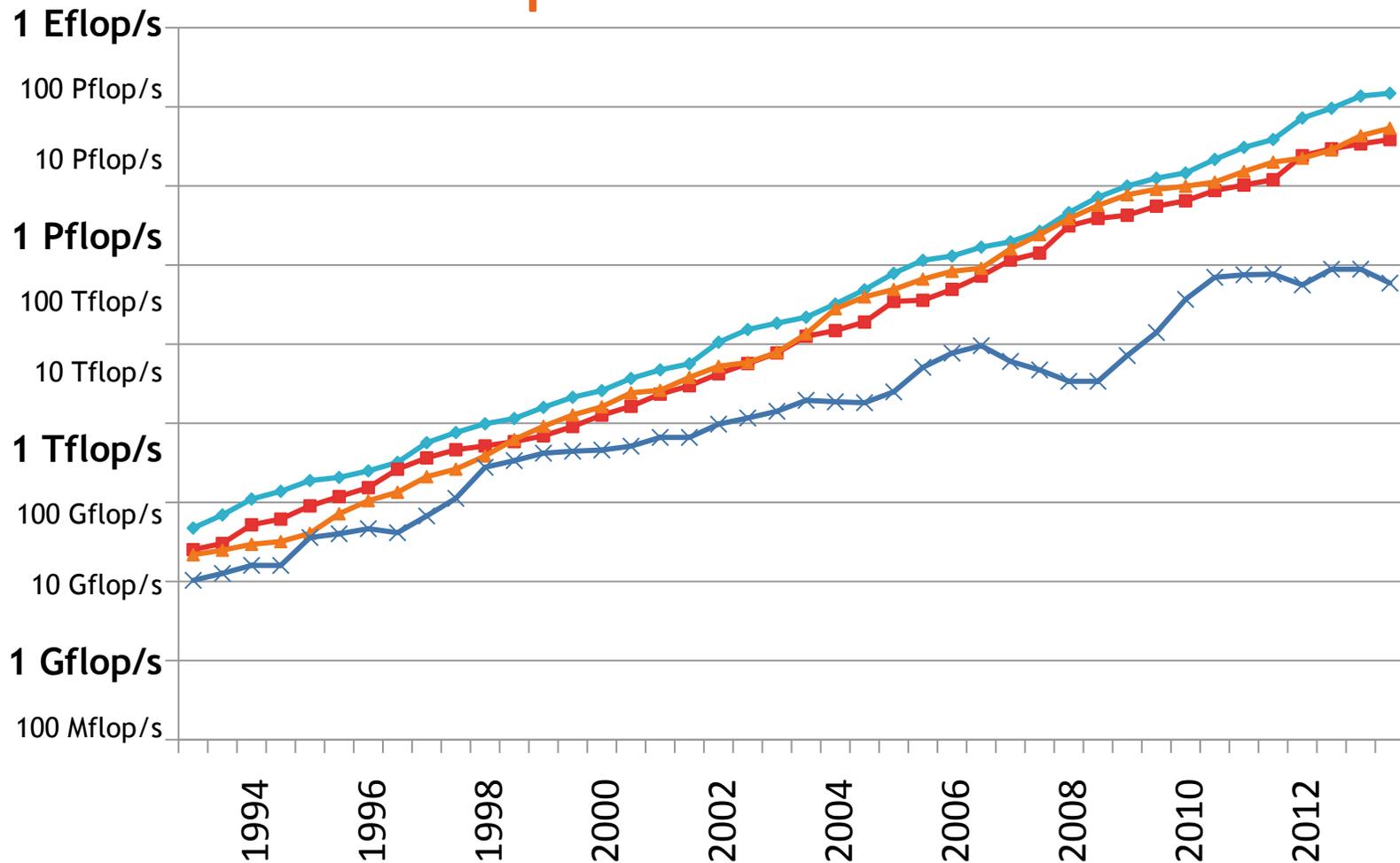
Gini



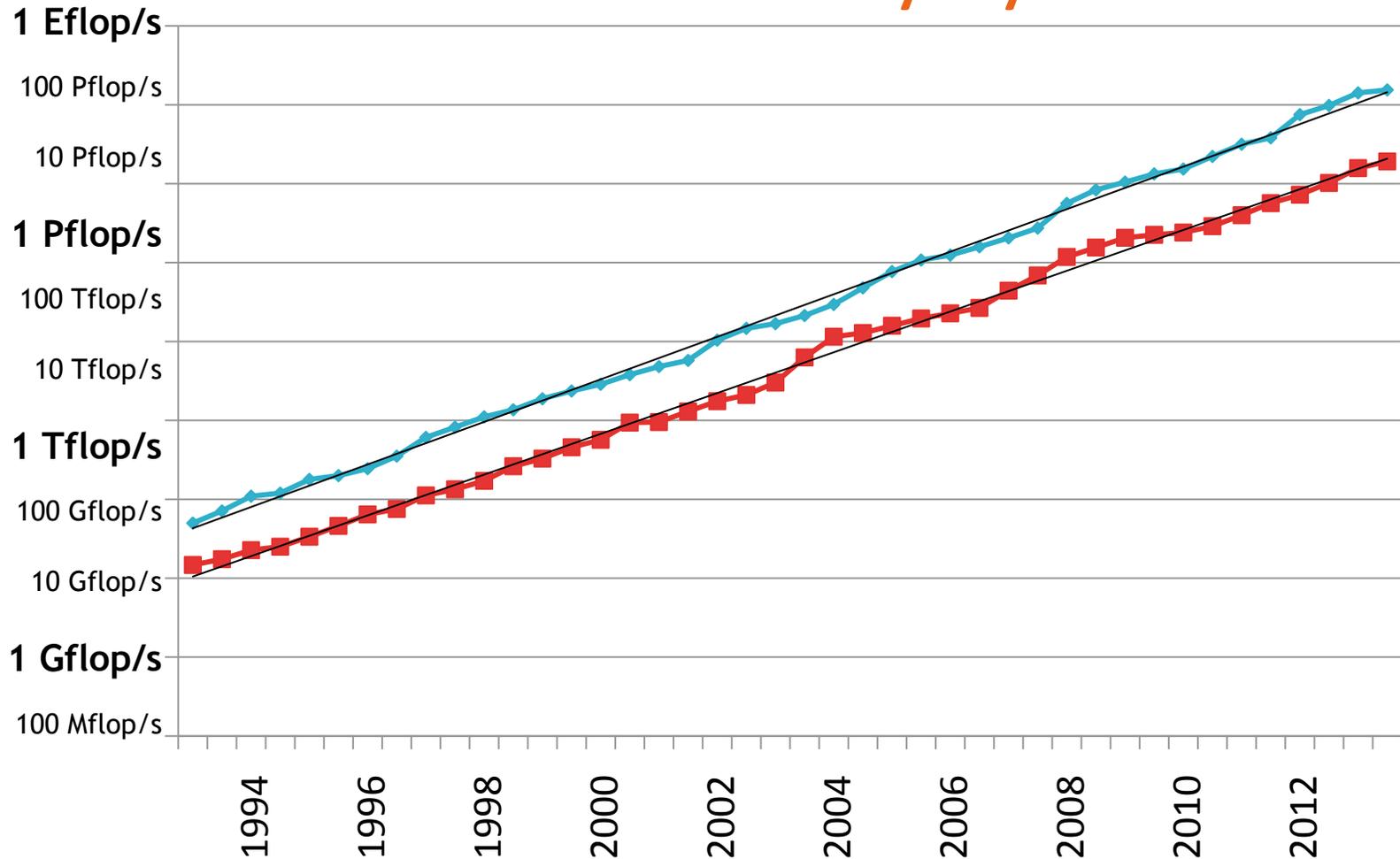
Gini Coefficient of the TOP50 Research and Industry Systems



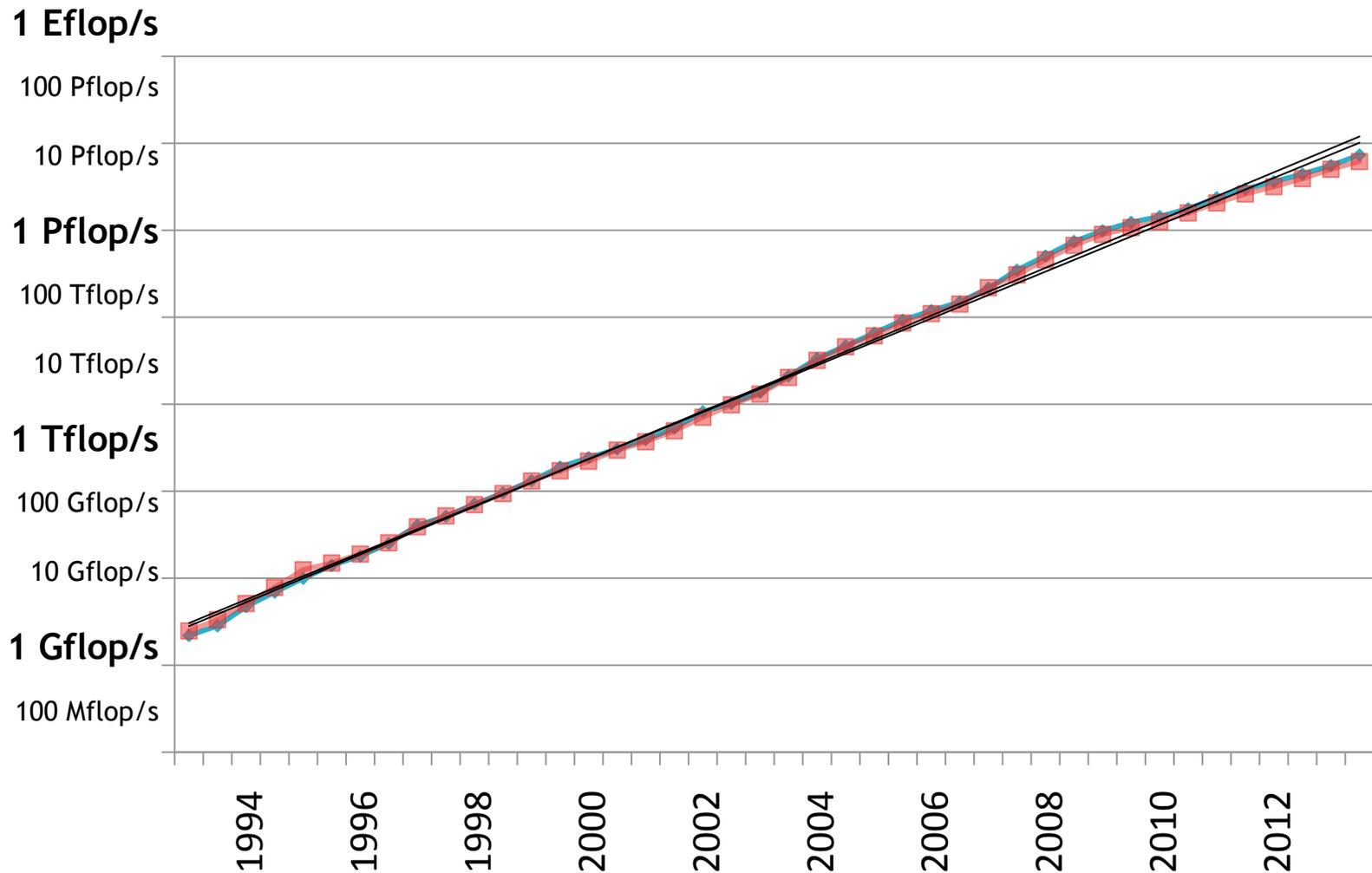
Performance Development of SubGroups in the TOP500



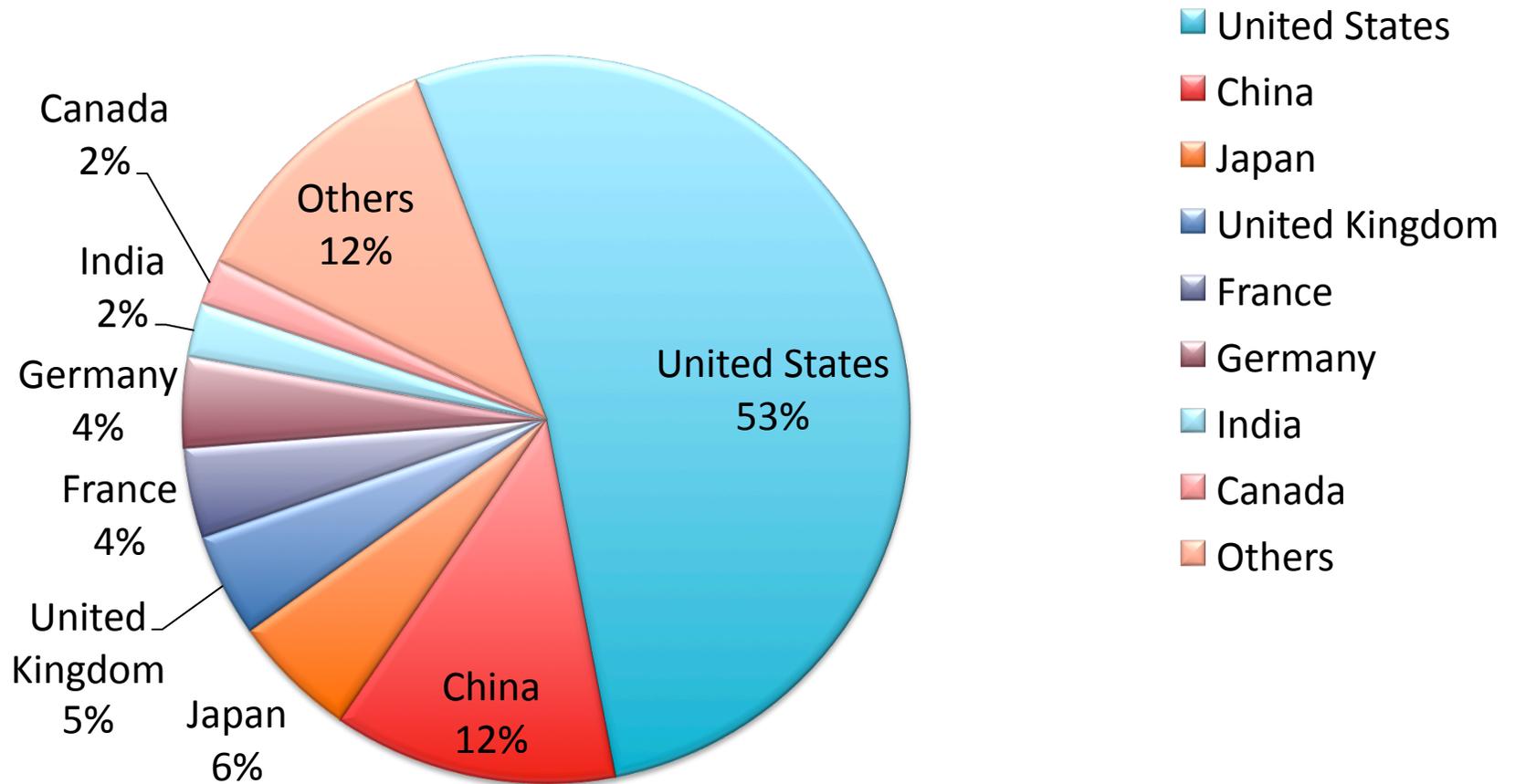
Performance Development of TOP50 Research and Industry Systems



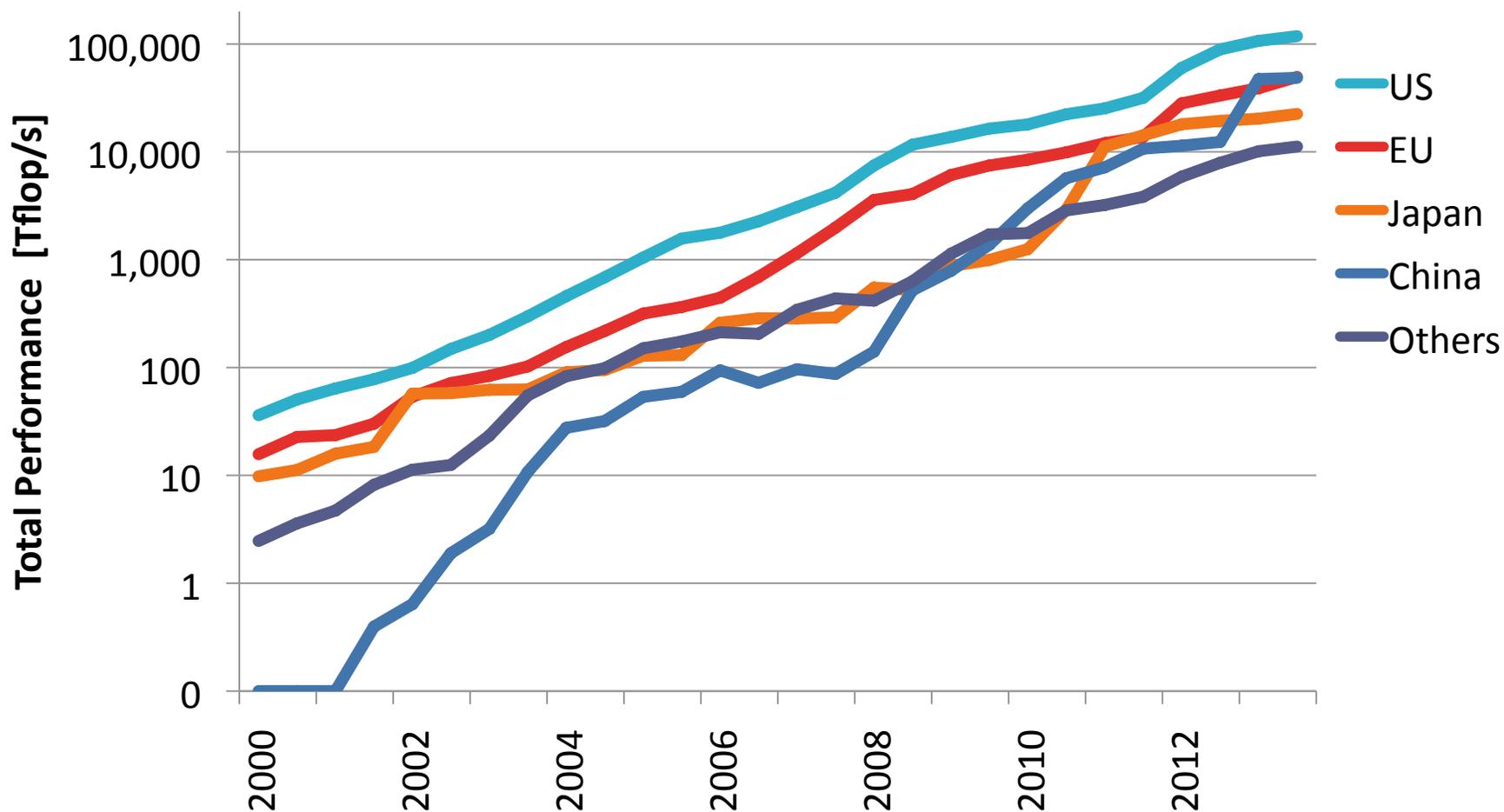
Performance Development of “Bottom-50” Research and Industry Systems



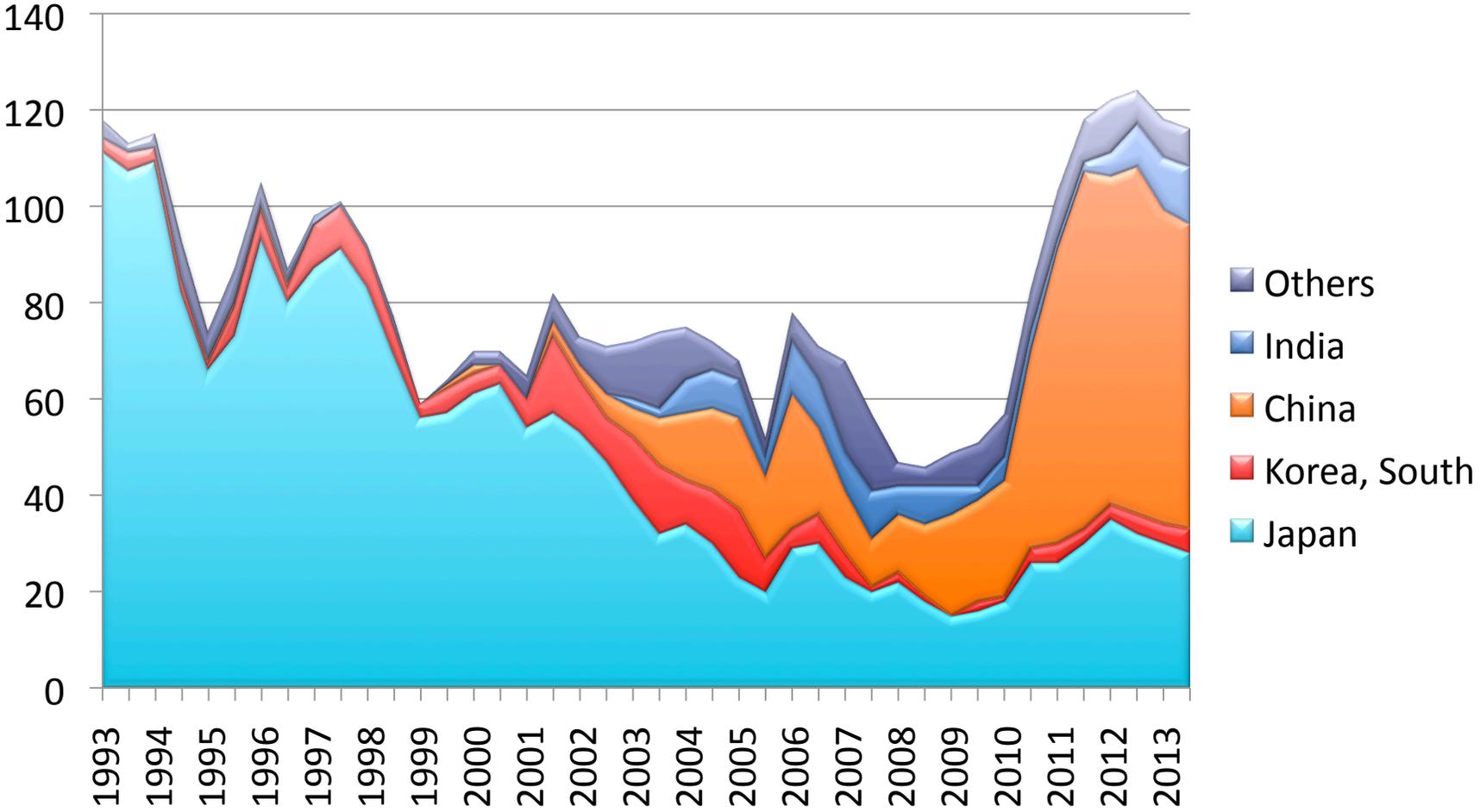
Countries / System Share



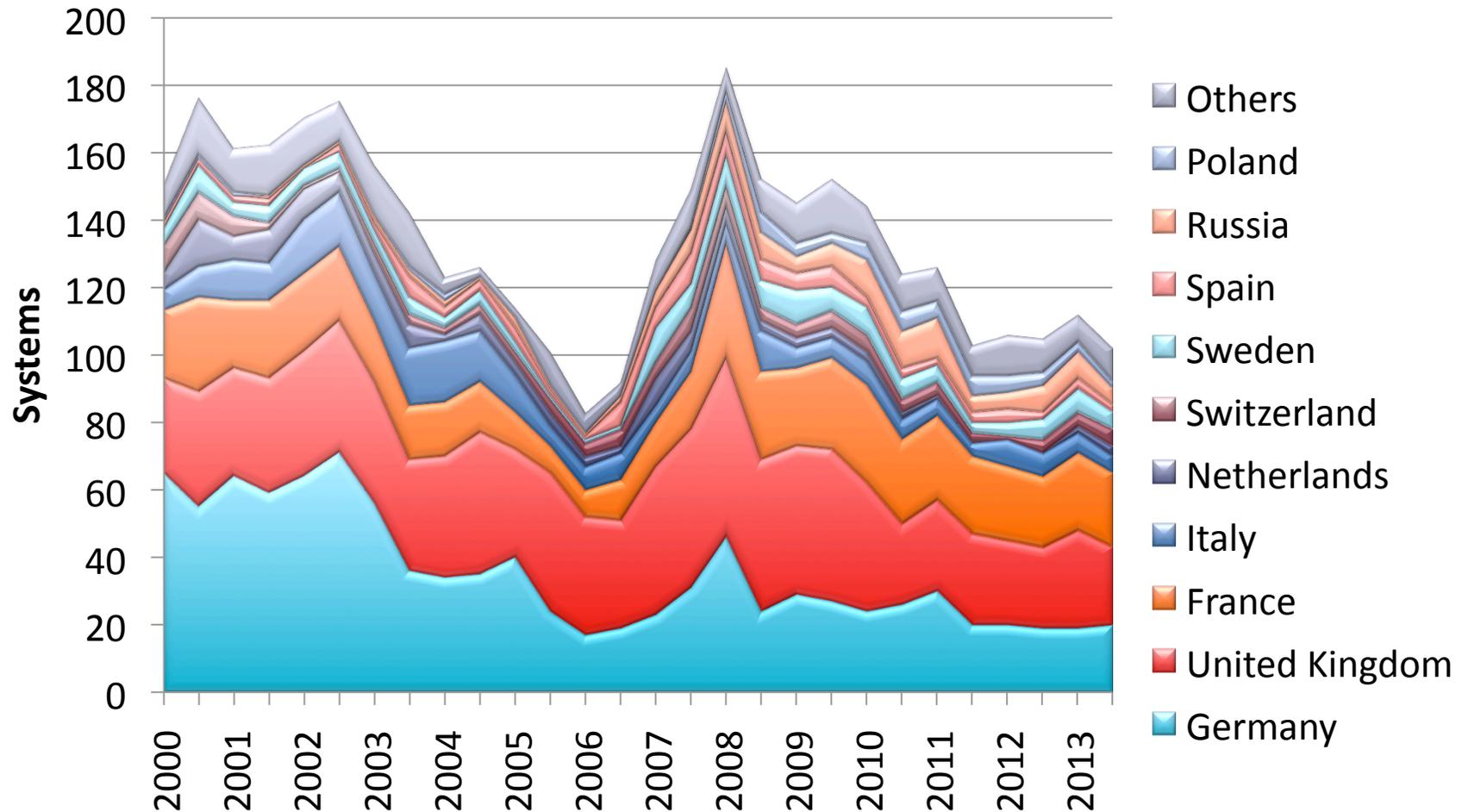
Performance of Countries



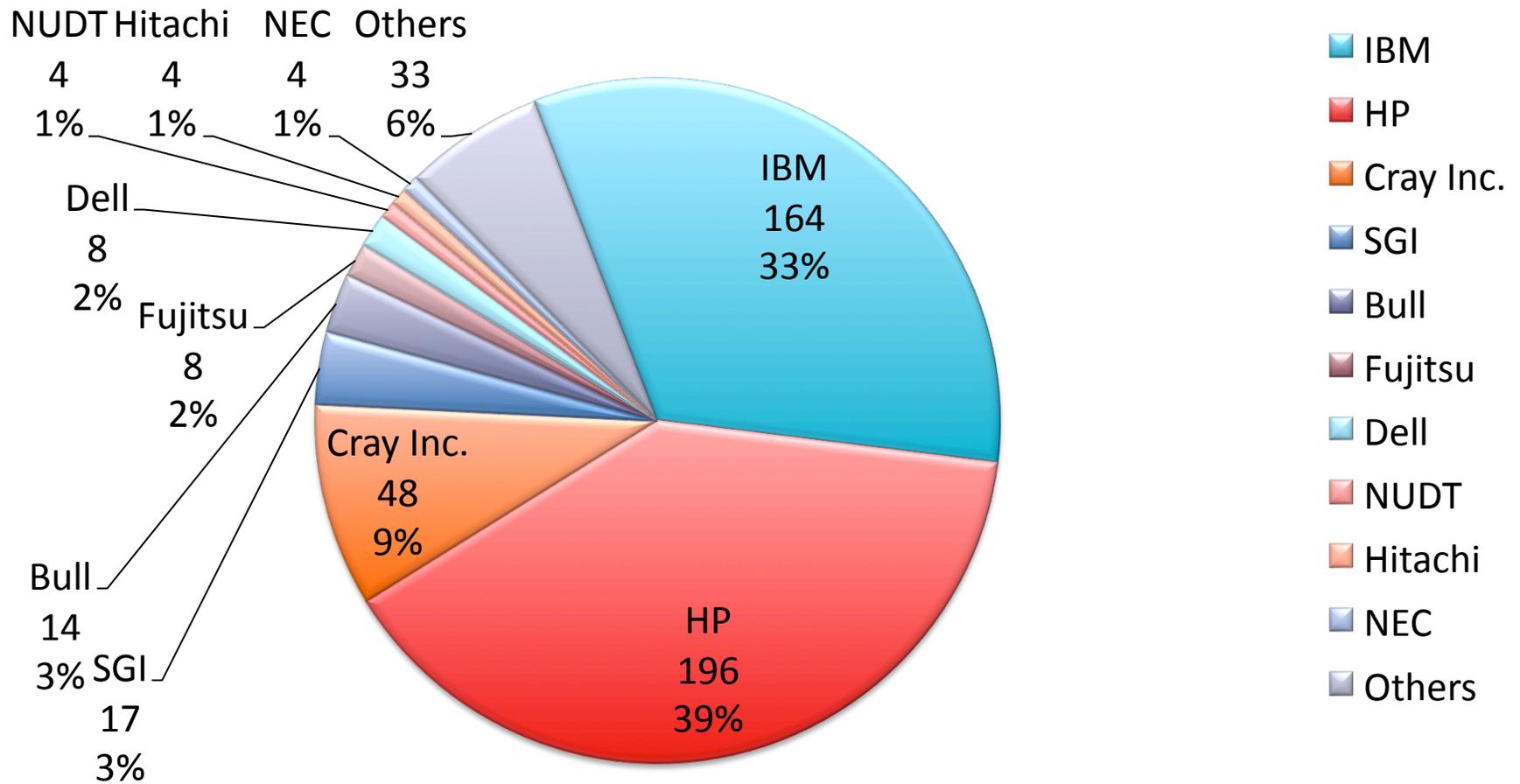
Asian Countries



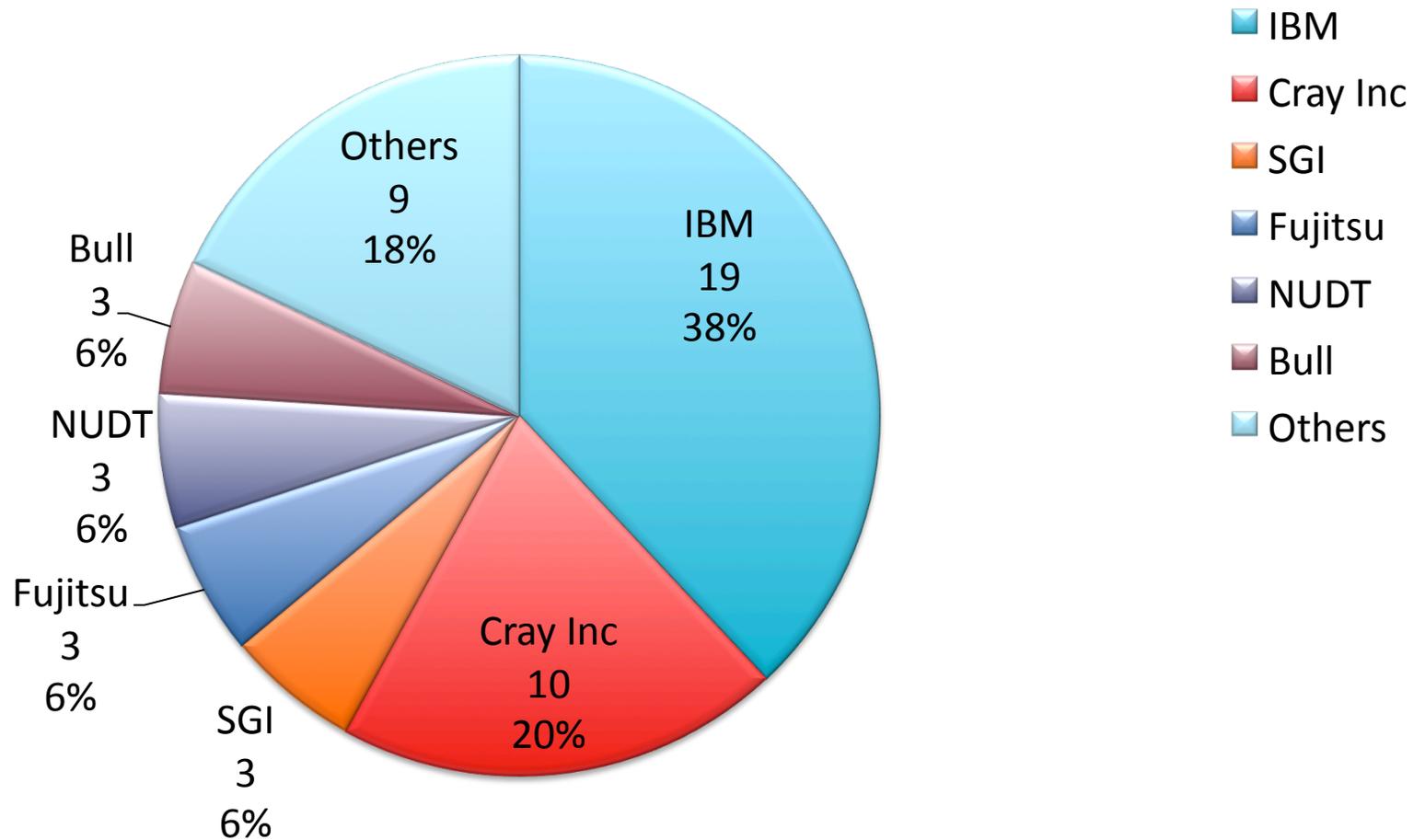
European Countries



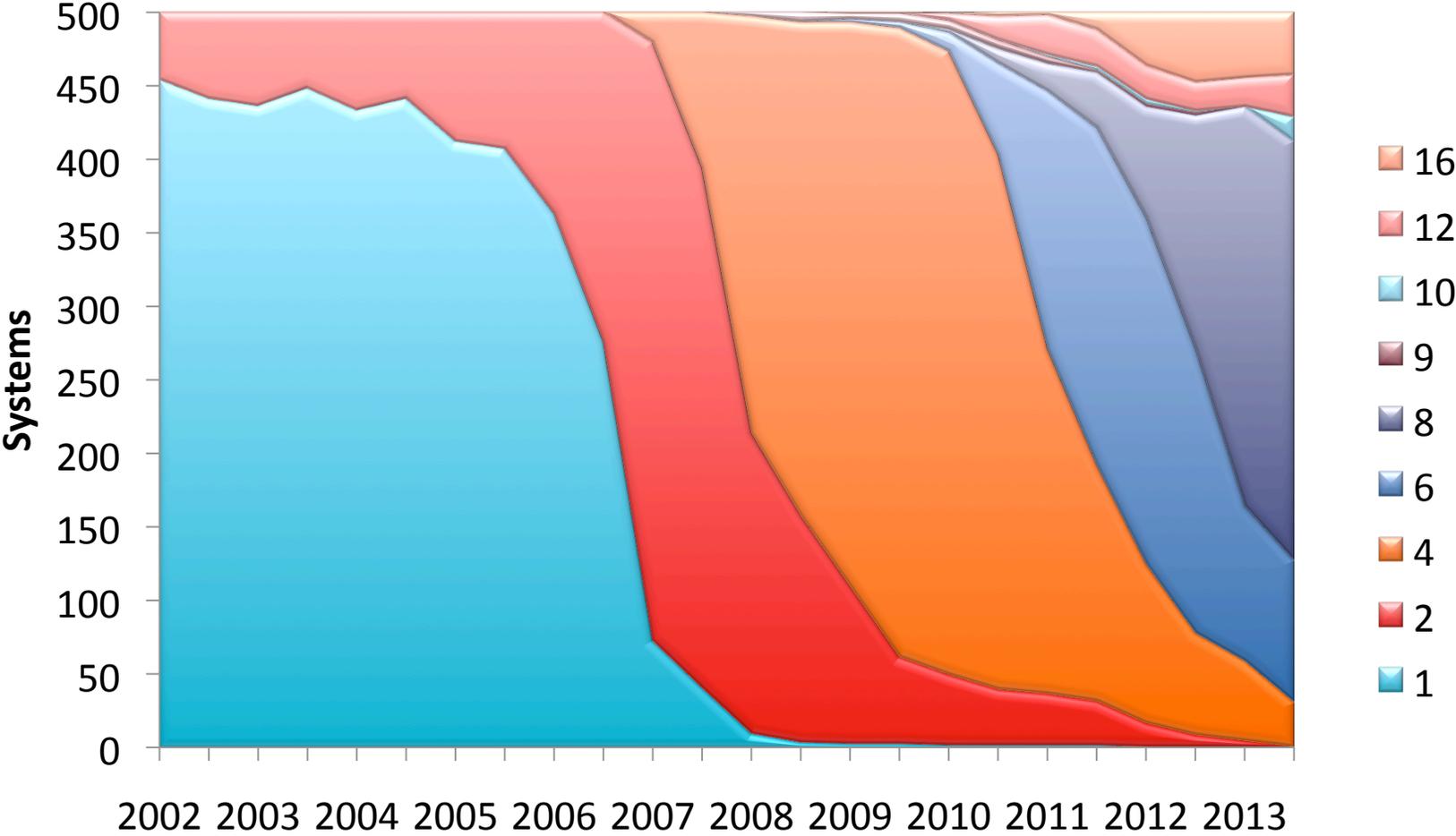
Vendors / System Share



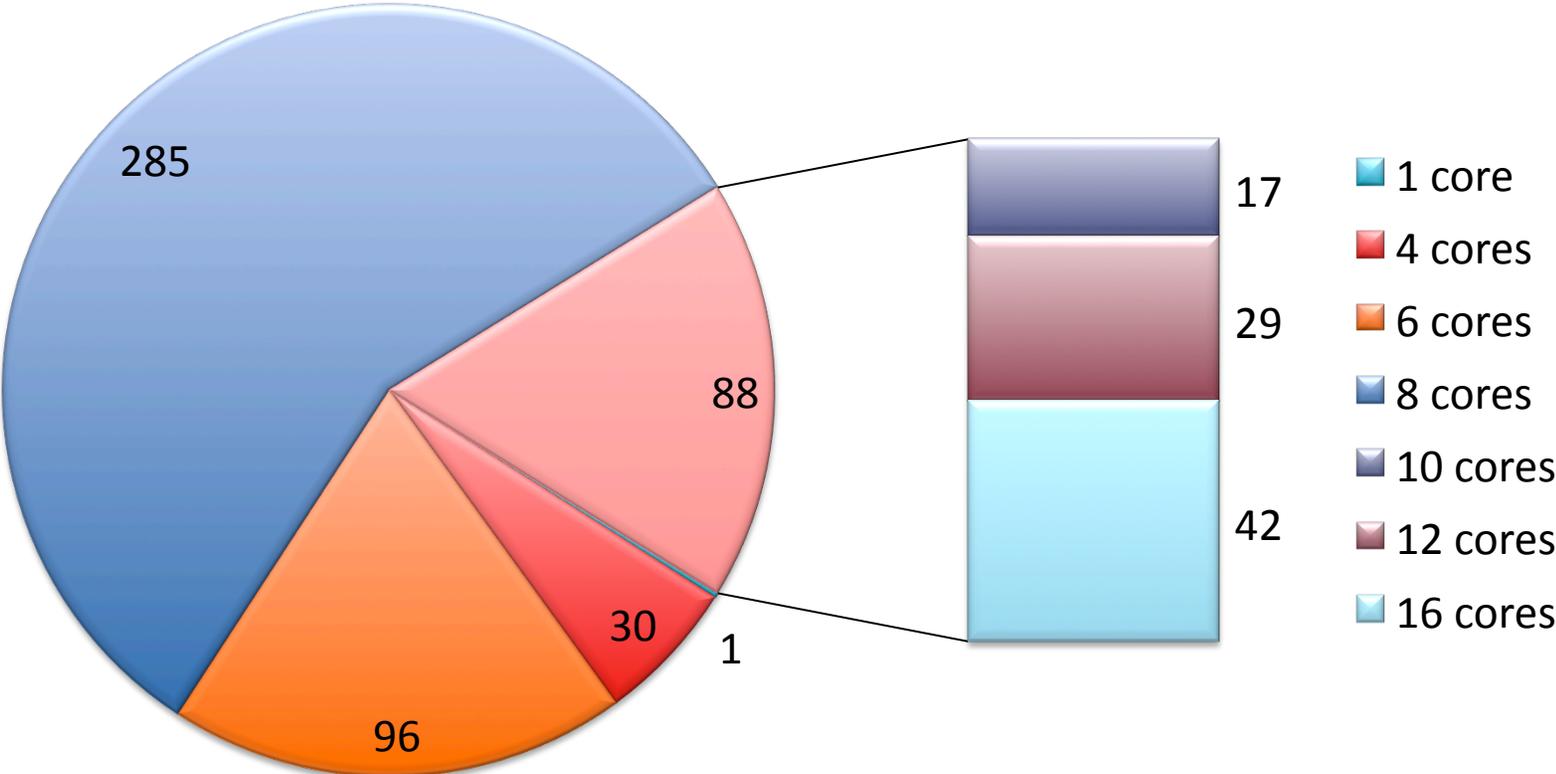
Vendors (TOP50) / System Share



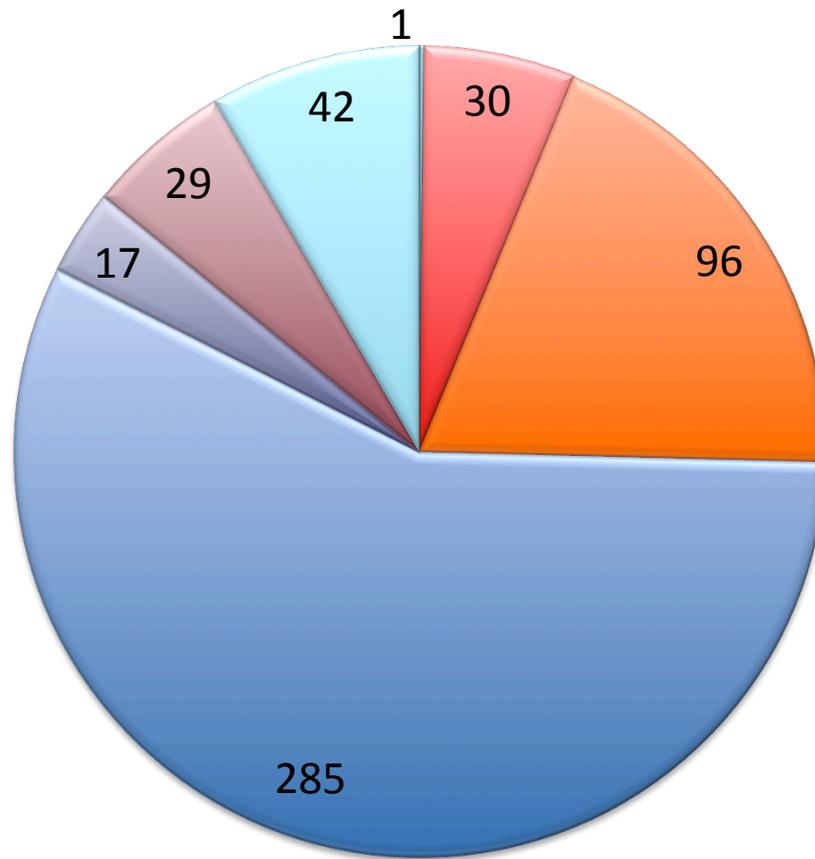
Cores per Socket



Cores per Socket

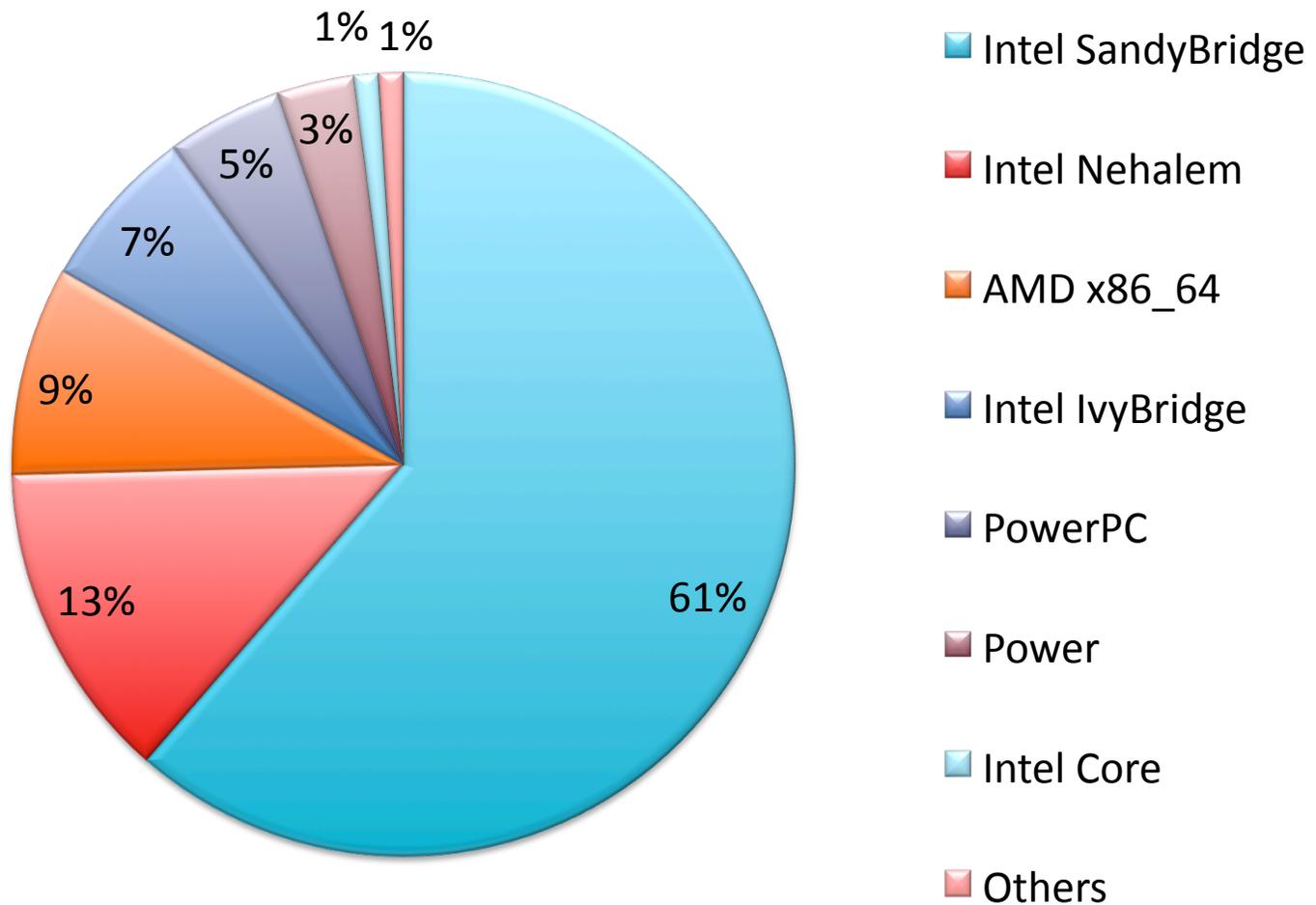


Cores per Socket

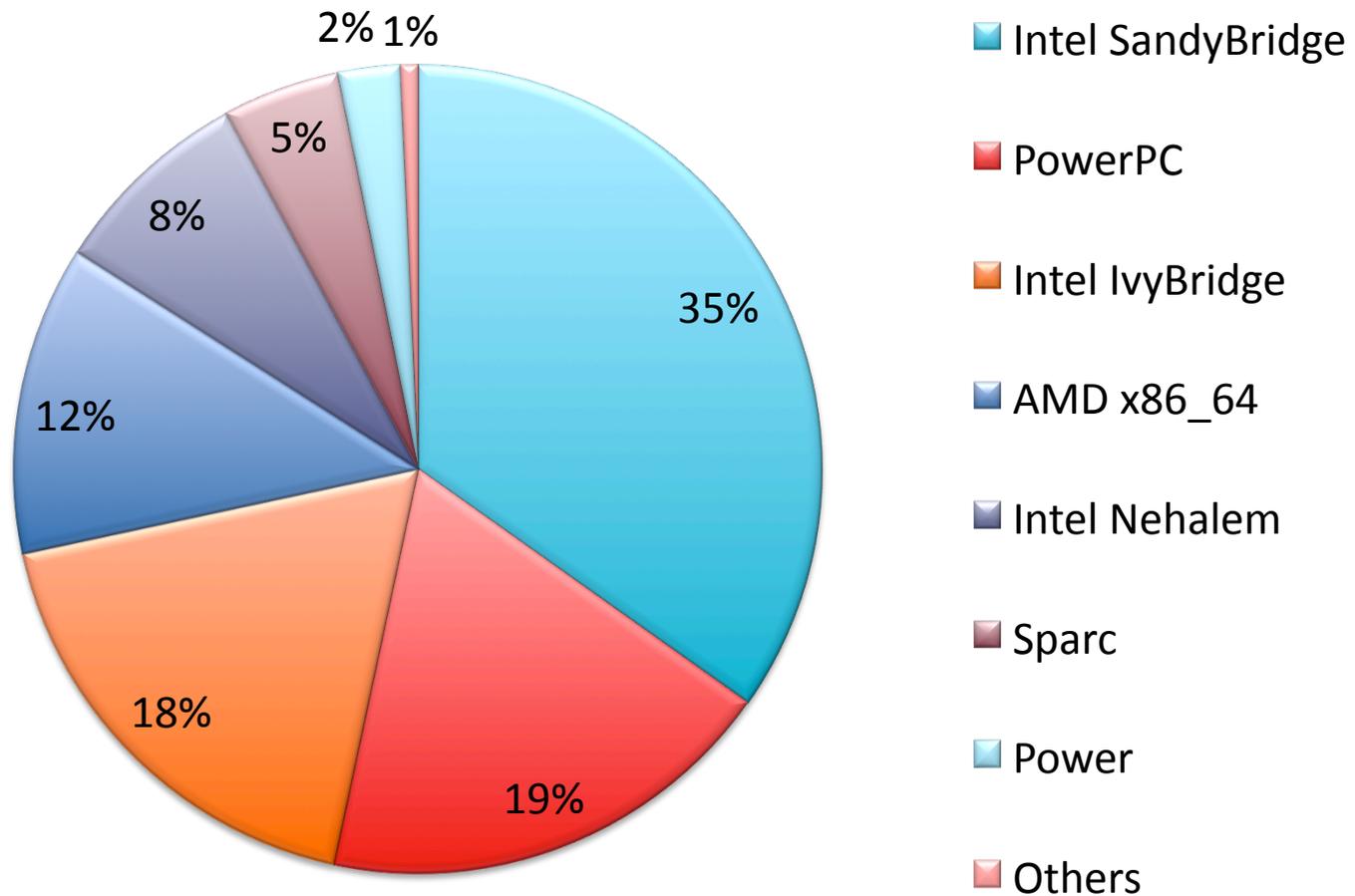


- 1 core
- 4 cores
- 6 cores
- 8 cores
- 10 cores
- 12 cores
- 16 cores

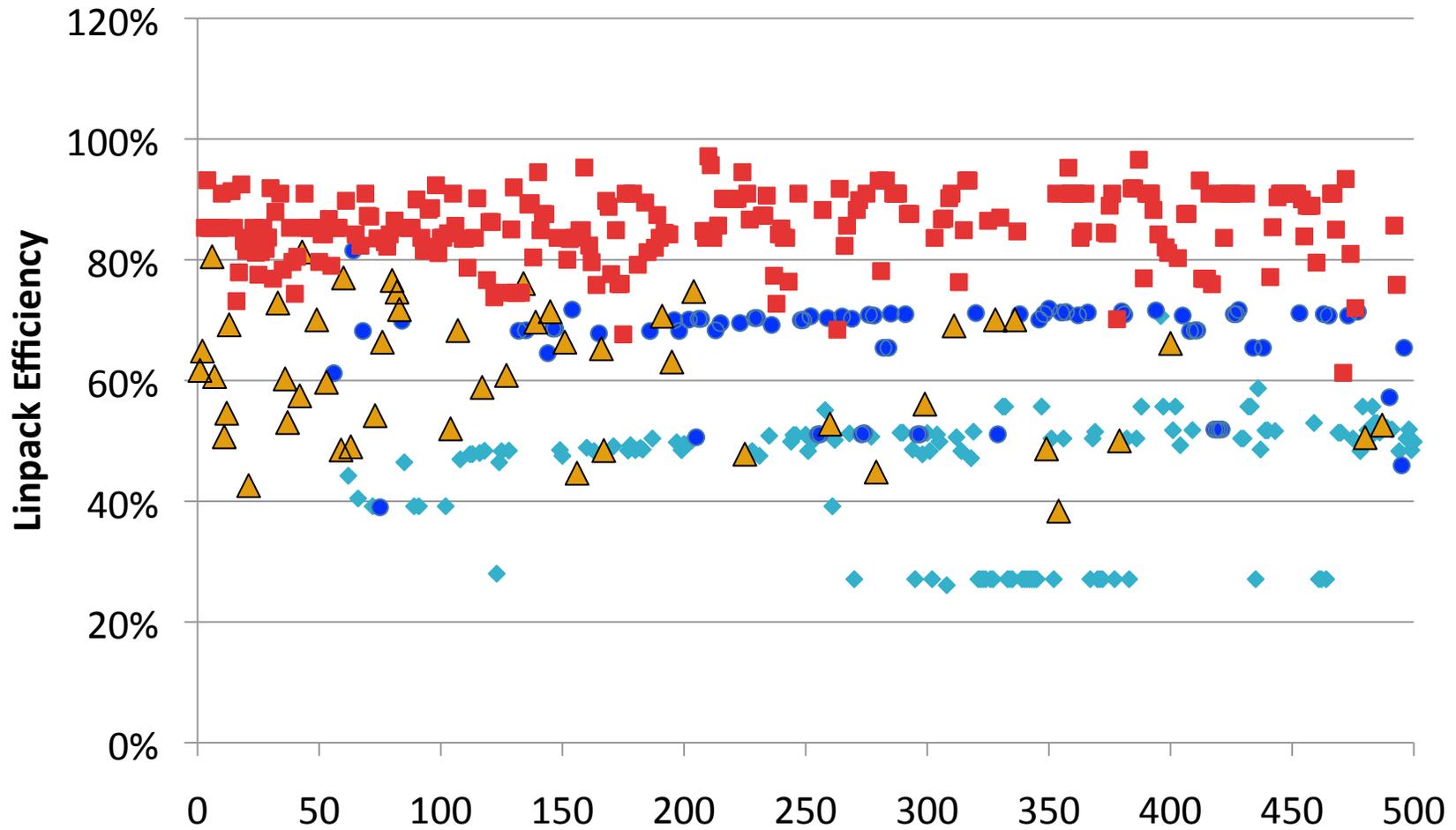
Processors / Systems



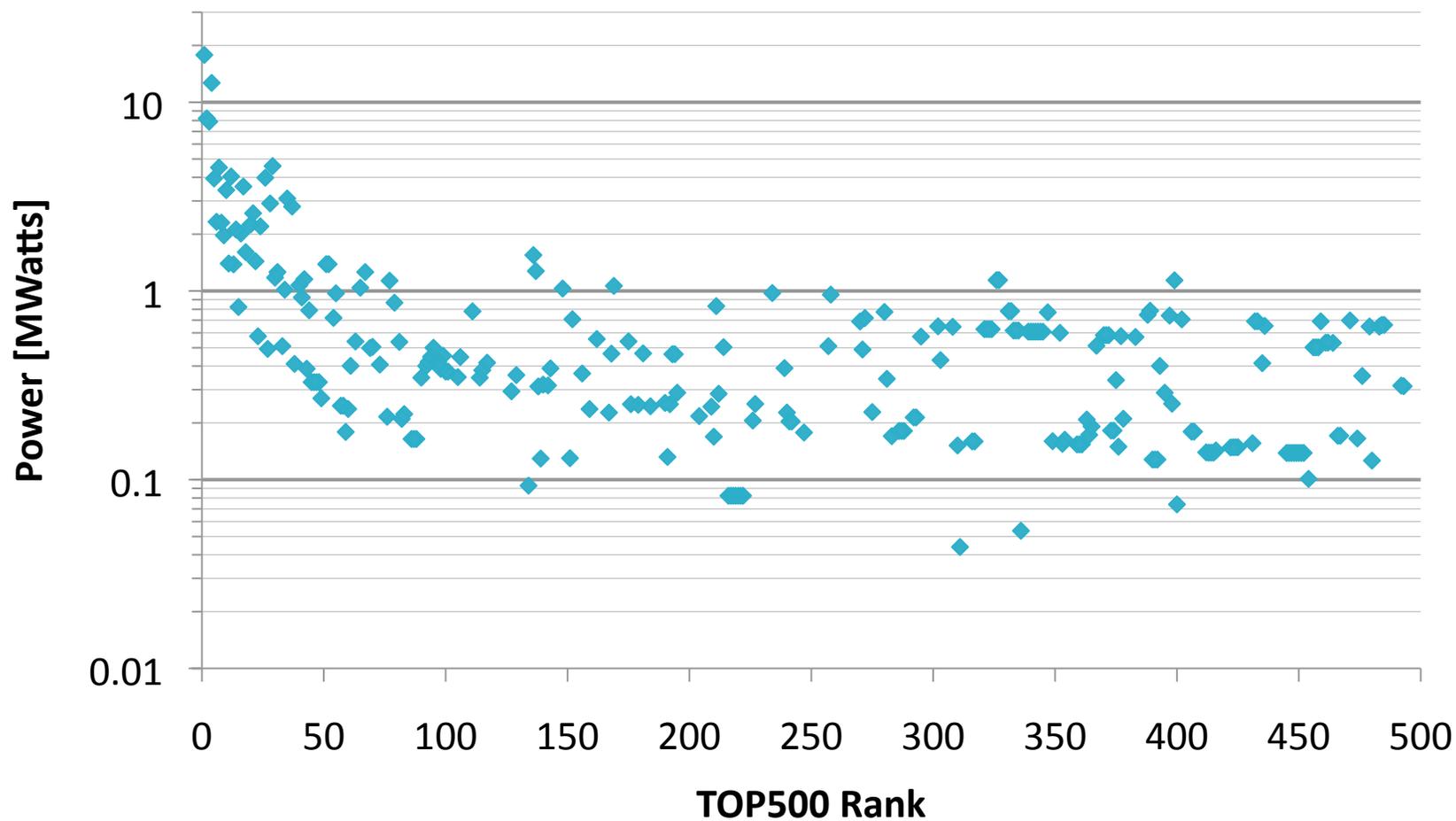
Processors / Performance



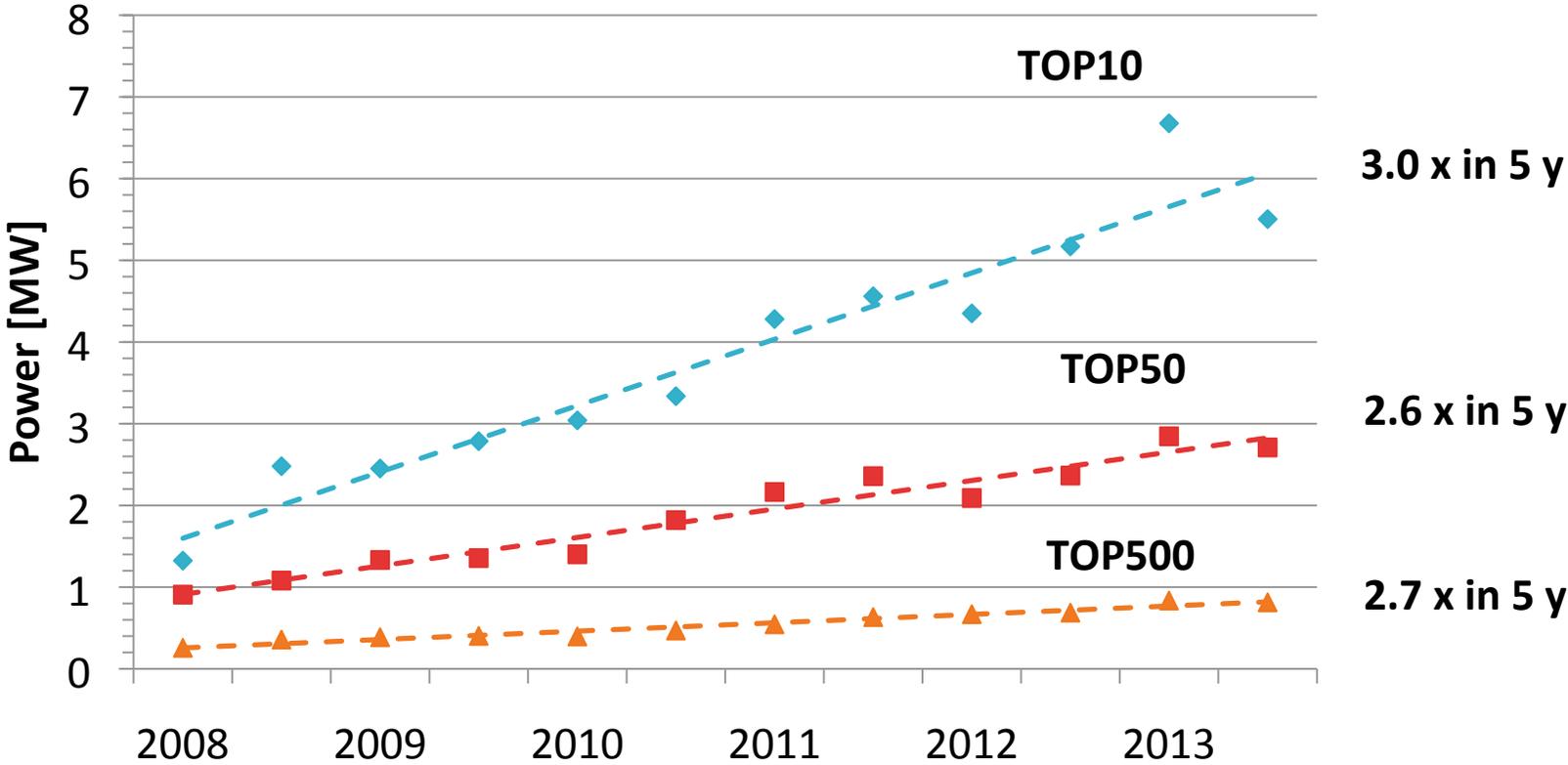
Linpack Efficiency



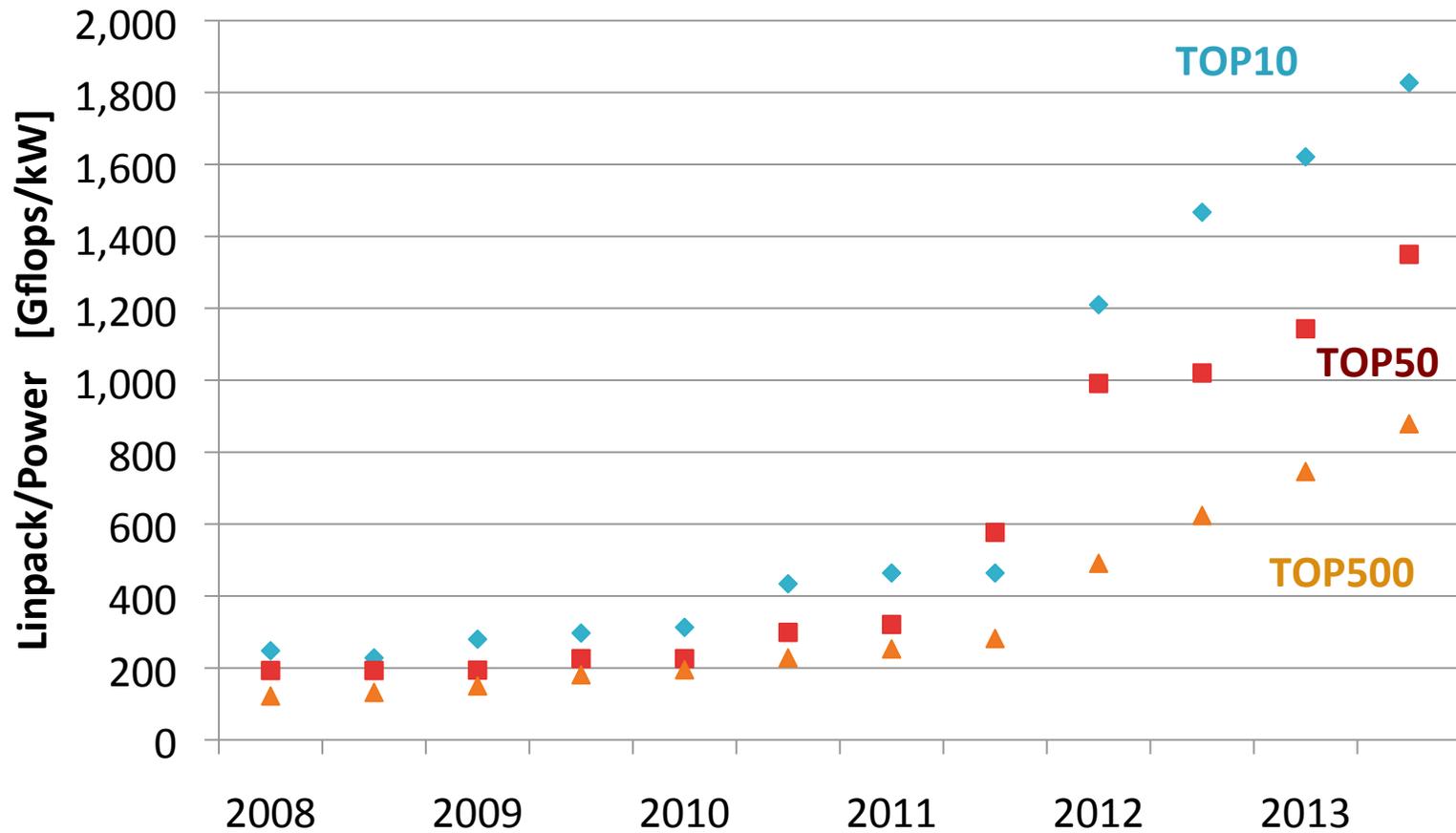
Absolute Power Levels



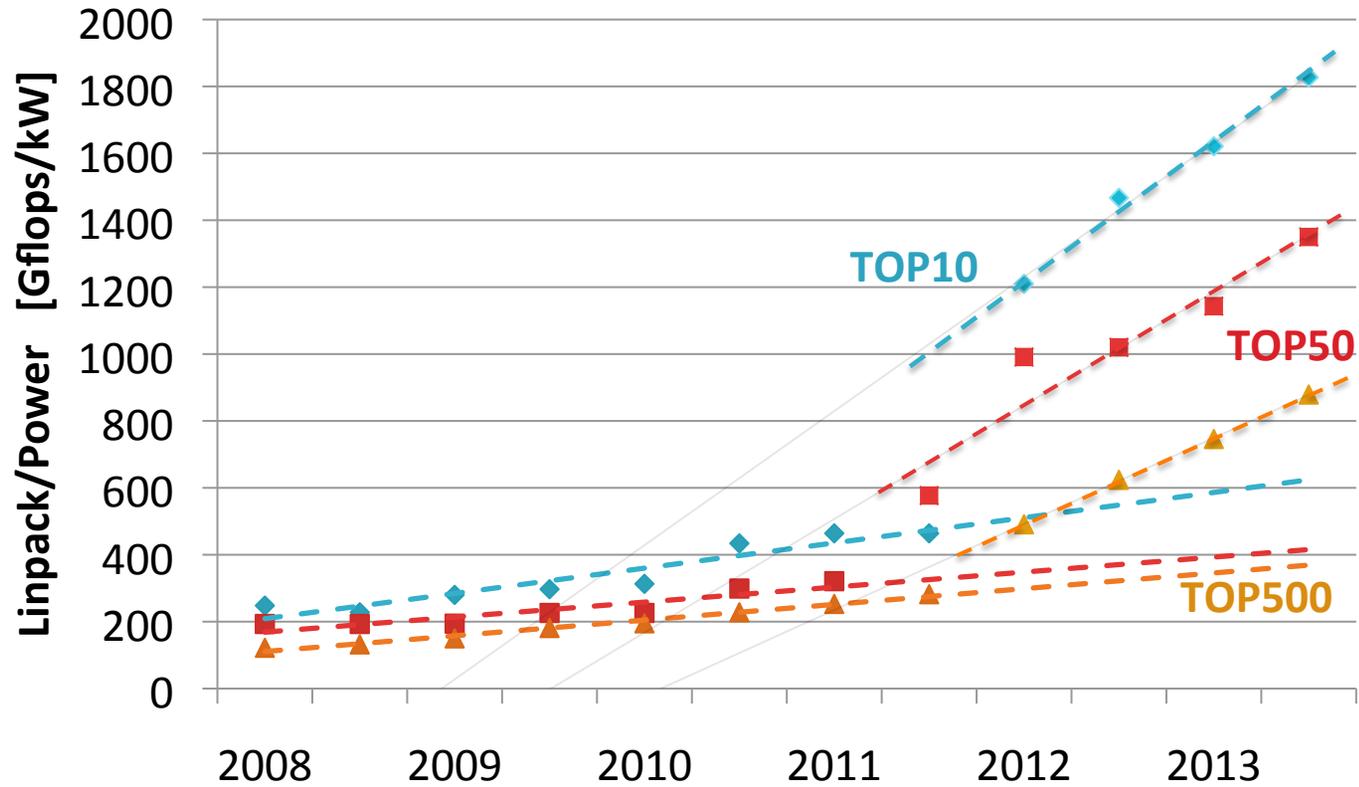
Power Consumption



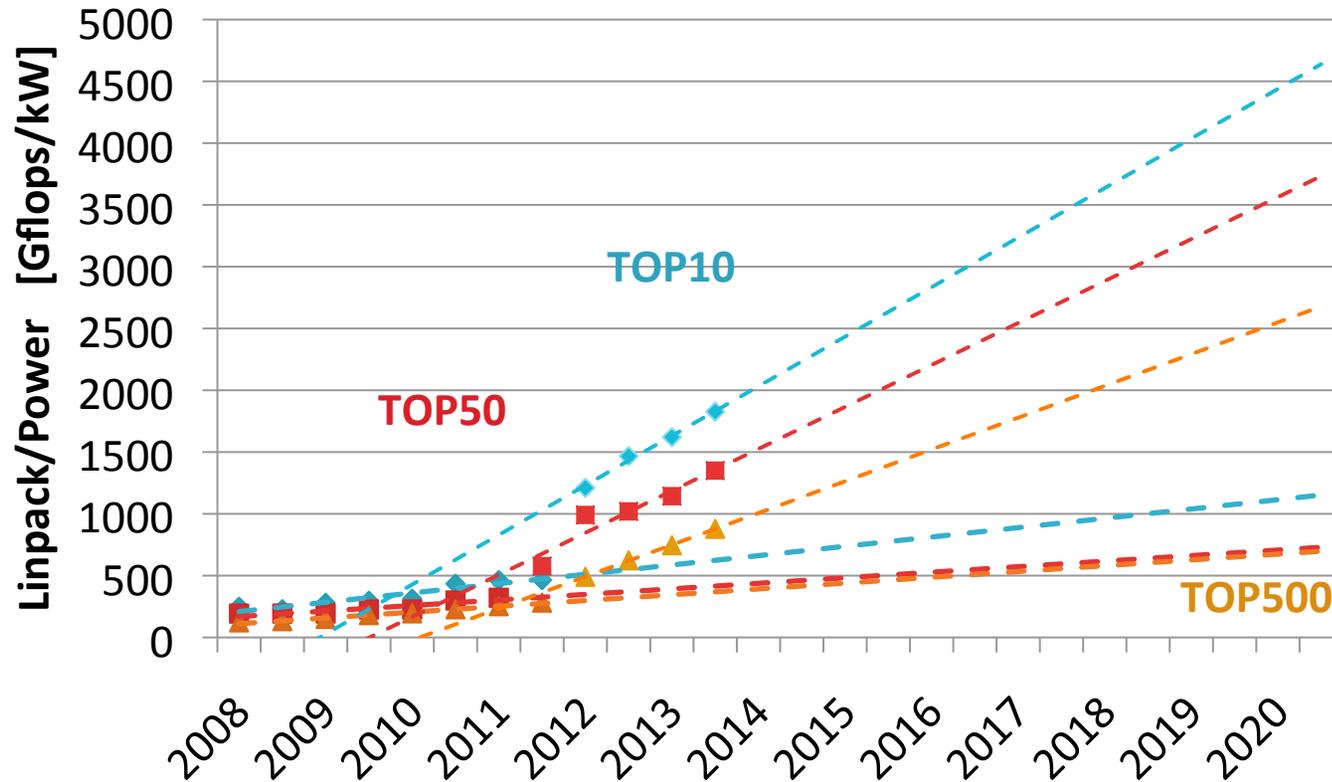
Power Efficiency



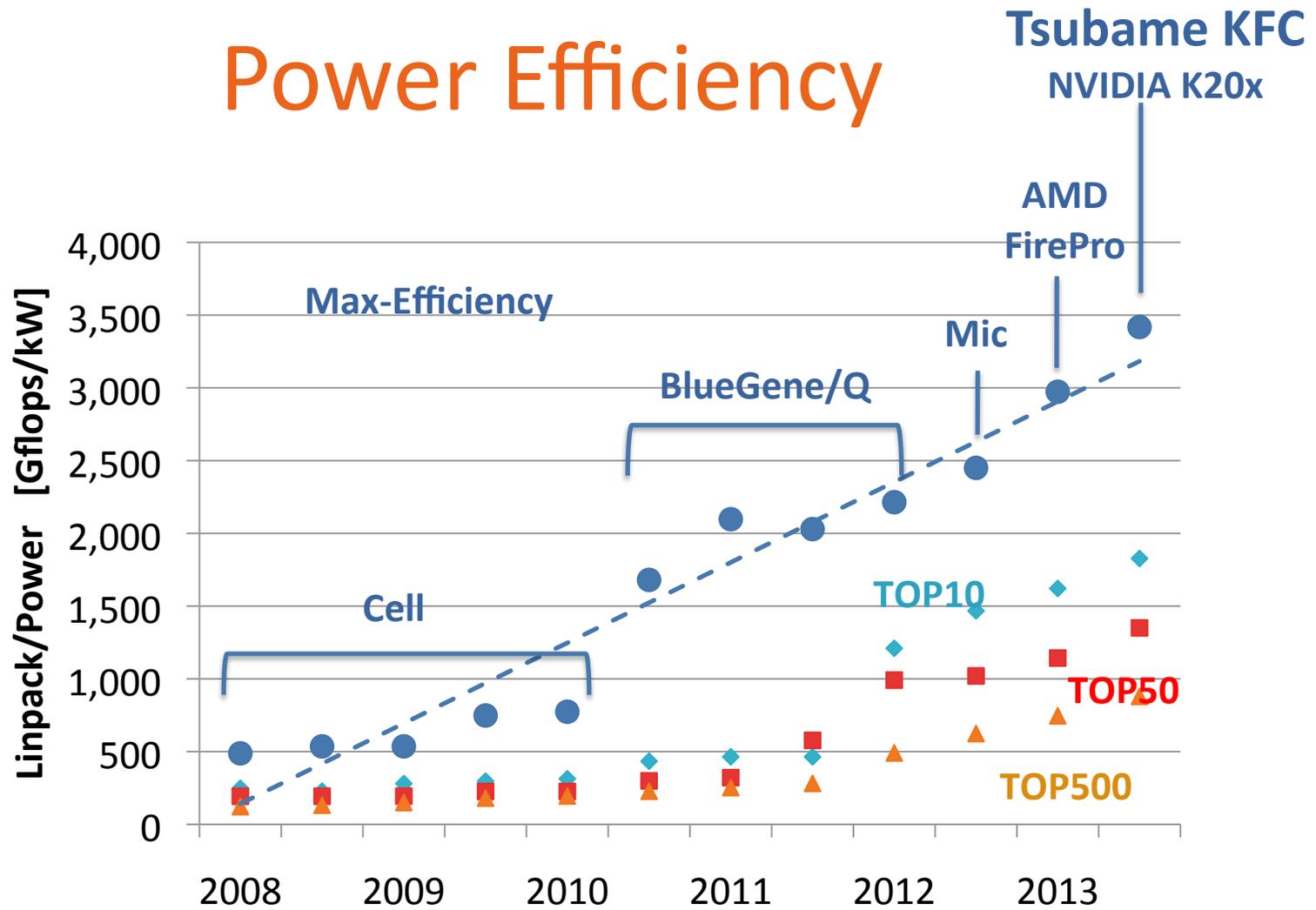
Power Efficiency



Power Efficiency



Power Efficiency



Most Power Efficient Architectures

Computer	Rmax/ Power
Tsubame KFC , NEC, Xeon 6C 2.1GHz, Infiniband FDR, NVIDIA K20x	3,418
HA-PACS TCA , Cray Cluster, Xeon 10C 2.8GHz, QDX, NVIDIA K20x	2,980
SANAM , Adtech, ASUS, Xeon 8C 2.0GHz, Infiniband FDR, AMD FirePro	2,973
iDataPlex DX360M4, Xeon 8C 2.6GHz, Infiniband FDR14, NVIDIA K20x	2,702
Piz Daint , Cray XC30, Xeon 8C 2.6GHz, Aries, NVIDIA K20x	2,697
BlueGene/Q , Power BQC 16C 1.60 GHz, Custom	2,300
HPCC , Cluster Platform SL250s, Xeon 8C 2.4GHz, FDR, NVIDIA K20m	2,243
Titan , Cray XK7 , Opteron 16C 2.2GHz, Gemini, NVIDIA K20x	2,143

TOWARD A NEW (ANOTHER) METRIC FOR RANKING HIGH PERFORMANCE COMPUTING SYSTEMS

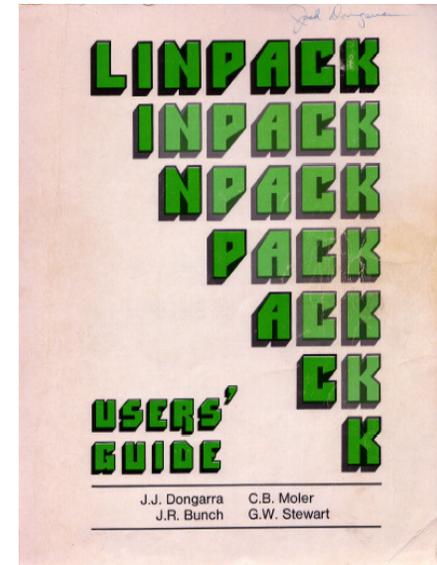
Jack Dongarra & Piotr Luszczek
University of Tennessee/ORNL

Michael Heroux
Sandia National Labs

See: <http://tiny.cc/hpcg>

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 36 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

Began in late 70's

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

$\frac{2}{3} N^3$ $\frac{2N^2}{3}$ ops time

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

Facility	TIME	UNIT	Computer	Type	Compiler
-----	N=100	micro-	-----	----	-----
	secs.	secs.			
NCAR	14.0	0.049	0.14	CRAY-1	S CFT, Assembly BLAS
LASL	4.64	0.148	0.43	CDC 7600	S FTN, Assembly BLAS
NCAR	3.57	0.192	0.56	CRAY-1	S CFT
LASL	3.27	0.210	0.61	CDC 7600	S FTN
Argonne	2.31	0.297	0.86	IBM 370/195	D H
NCAR	1.91	0.359	1.05	CDC 7600	S Local
Argonne	1.77	0.388	1.33	IBM 3033	D H
NASA Langley	1.40	0.489	1.42	CDC Cyber 175	S FTN
U. Ill. Urbana	1.36	0.506	1.47	CDC Cyber 175	S Ext. 4.6
LLL	1.24	0.554	1.61	CDC 7600	S CHAT, No optimize
SLAC	1.19	0.579	1.69	IBM 370/168	D H Ext., Fast mult.
Michigan	1.09	0.631	1.84	Amdahl 470/V6	D H
Toronto	0.77	0.890	2.59	IBM 370/165	D H Ext., Fast mult.
Northwestern	0.47	1.44	4.20	CDC 6600	S FTN
Texas	0.35	1.93*	5.63	CDC 6600	S RUN
China Lake	0.32	1.95*	5.69	Univac 1110	S V
Yale	0.26	2.59	7.53	DEC KL-20	S F20
Bell Labs	0.19	3.46	10.1	Honeywell 6080	S Y
Wisconsin	0.19	3.49	10.1	Univac 1110	S V
Iowa State	0.19	3.54	10.2	Itel AS/5 mod3	D H
U. Ill. Chicago	0.14	4.10	11.9	IBM 370/158	D G1
Purdue	0.14	5.69	16.6	CDC 6500	S FUN
U. C. San Diego	0.06	13.1	38.2	Burroughs 6700	S H
Yale	0.04	17.1*	49.9	DEC KA-10	S F40

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

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 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 $n=12.7 \times 10^6$ (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 $n = 3.3 \times 10^6$
 - .11 Pflop/s #160 today

#1 System on the Top500 Over the Past 20 Years

(16 machines in that club)

9 

6 

2 

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 - 11/13(?)	NUDT Tianhe-2 Intel IvyBridge & Xeon Phi	33,862.	9,960,000	5.4	17.8

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

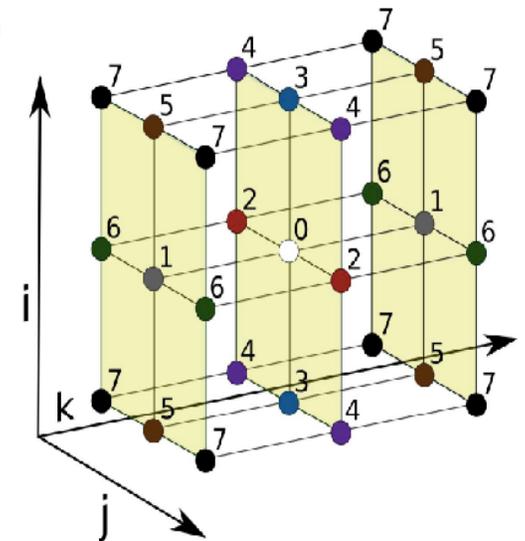
- Top 500
- Green 500
- Graph ~~500~~-161
- Sustained Petascale Performance
- HPC Challenge
- Perfect
- ParkBench
- SPEC-hpc
- Livermore Loops
- EuroBen
- NAS Parallel Benchmarks
- Genesis
- RAPS
- SHOC
- LAMMPS
- Dhrystone
- Whetstone

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 (via spectral properties of CG).

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

Example

- Build HPCG with default MPI and OpenMP modes enabled.

```
export OMP_NUM_THREADS=1
```

```
mpiexec -n 96 ./xhpcg 70 80 90
```

- Results in:

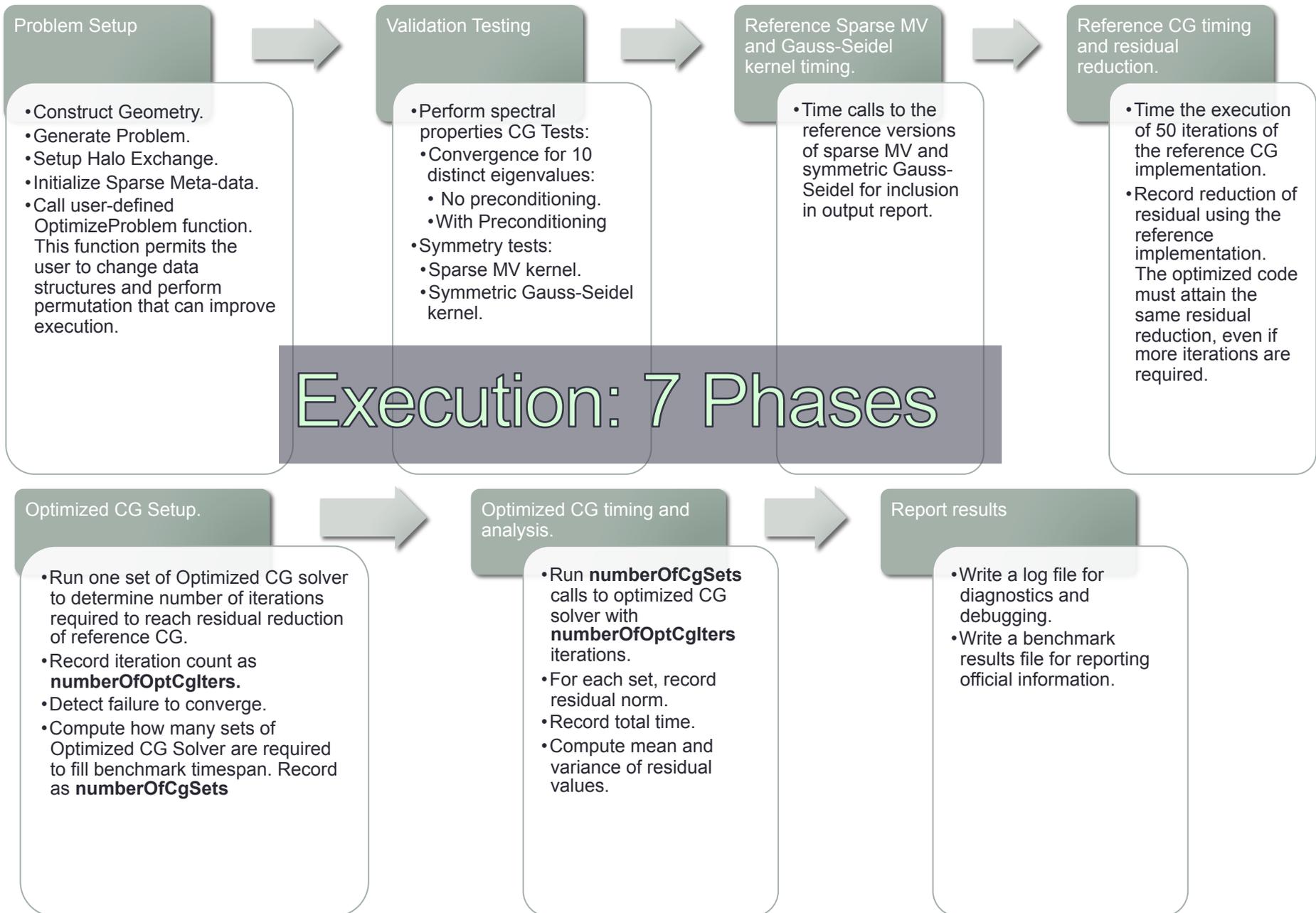
$$n_x = 70, n_y = 80, n_z = 90$$

$$np_x = 4, np_y = 4, np_z = 6$$

- Global domain dimensions: 280-by-320-by-540
- Number of equations per MPI process: 504,000
- Global number of equations: 48,384,000
- Global number of nonzeros: 1,298,936,872
- Note: Changing OMP_NUM_THREADS does not change any of these values.

CG ALGORITHM

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



Problem Setup

- Construct Geometry.
- Generate Problem.
- Setup Halo Exchange.
 - Use symmetry to eliminate communication in this phase.
 - C++ STL containers/algorithms: Simple code, force use of C++.
- Initialize Sparse Meta-data.
- Call user-defined OptimizeProblem function.
 - Permits the user to change data structures and perform permutation that can improve execution.

- Temporarily modify matrix diagonals:
 - (2.0e6, 3.0e6, ... 9.0e6, 1.0e6, ...1.0e6).
 - Offdiagonal still -1.0.
 - Matrix looks diagonal with 10 distinct eigenvalues.
- Perform spectral properties CG Tests:
 - Convergence for 10 distinct eigenvalues:
 - No preconditioning: About 10 iters.
 - With Preconditioning: About 1 iter.
- Symmetry tests:
 - Matrix, preconditioner are symmetric.
 - Sparse MV kernel. $x^T Ay = y^T Ax$
 - Symmetric Gauss-Seidel kernel. $x^T M^{-1}y = y^T M^{-1}x$

Reference Sparse MV and Gauss-Seidel kernel timing.

- Time calls to the reference versions of sparse MV and symmetric Gauss-Seidel for inclusion in output report.

Reference CG timing and residual reduction.

- Time the execution of 50 iterations of the reference CG implementation.
- Record reduction of residual using the reference implementation.
- The optimized code must attain the same residual reduction, *even if more iterations are required*.
- Most graph coloring algorithms improve parallel execution at the expense of increasing iteration counts.

Optimized CG Setup.

- Run one set of Optimized CG solver to determine number of iterations required to reach residual reduction of reference CG.
- Record iteration count as **numberOfOptCgIters**.
- Detect failure to converge.
- Compute how many sets of Optimized CG Solver are required to fill benchmark timespan. Record as **numberOfCgSets**

Optimized CG timing and analysis.

- Run **numberOfCgSets** calls to optimized CG solver with **numberOfOptCgIters** iterations.
- For each set, record residual norm.
- Record total time.
- Compute mean and variance of residual values.

Report results

- Write a log file for diagnostics and debugging.
- Write a benchmark results file for reporting official information.

Example

- Reference CG: 50 iterations, residual drop of $1e-6$.
- Optimized CG: Run one *set* of iterations
 - Multicolor ordering for Symmetric Gauss-Seidel:
 - Better vectorization, threading.
 - But: Takes 65 iterations to reach residual drop of $1e-6$.
 - Overhead:
 - Extra 15 iterations.
 - Computing of multicolor ordering.
 - Compute number of sets we must run to fill entire execution time:
 - $5h/\text{time-to-compute-1-set}$.
 - Results in thousands of CG set runs.
- Run and record residual for each set.
 - Report mean and variance (accounts for non-associativity of FP addition).

Preconditioner

- Symmetric Gauss-Seidel preconditioner
 - (Non-overlapping additive Schwarz)
 - Differentiate latency vs. throughput optimize core sets.

- From Matlab reference code:

Setup:

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

Solve:

```
x = LA\y;
```

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

```
x = UA\x1;
```

Key Computation Data Patterns

- Domain decomposition:
 - SPMD (MPI): Across domains.
 - Thread/vector (OpenMP, compiler): Within domains.
- Vector ops:
 - AXPY: Simple streaming memory ops.
 - DOT/NRM2 : Blocking Collectives.
- Matrix ops:
 - SpMV: Classic sparse kernel (option to reformat).
 - Symmetric Gauss-Seidel: sparse triangular sweep.
 - Exposes real application tradeoffs:
 - threading & convergence vs. SPMD and scaling.

Merits of HPCG

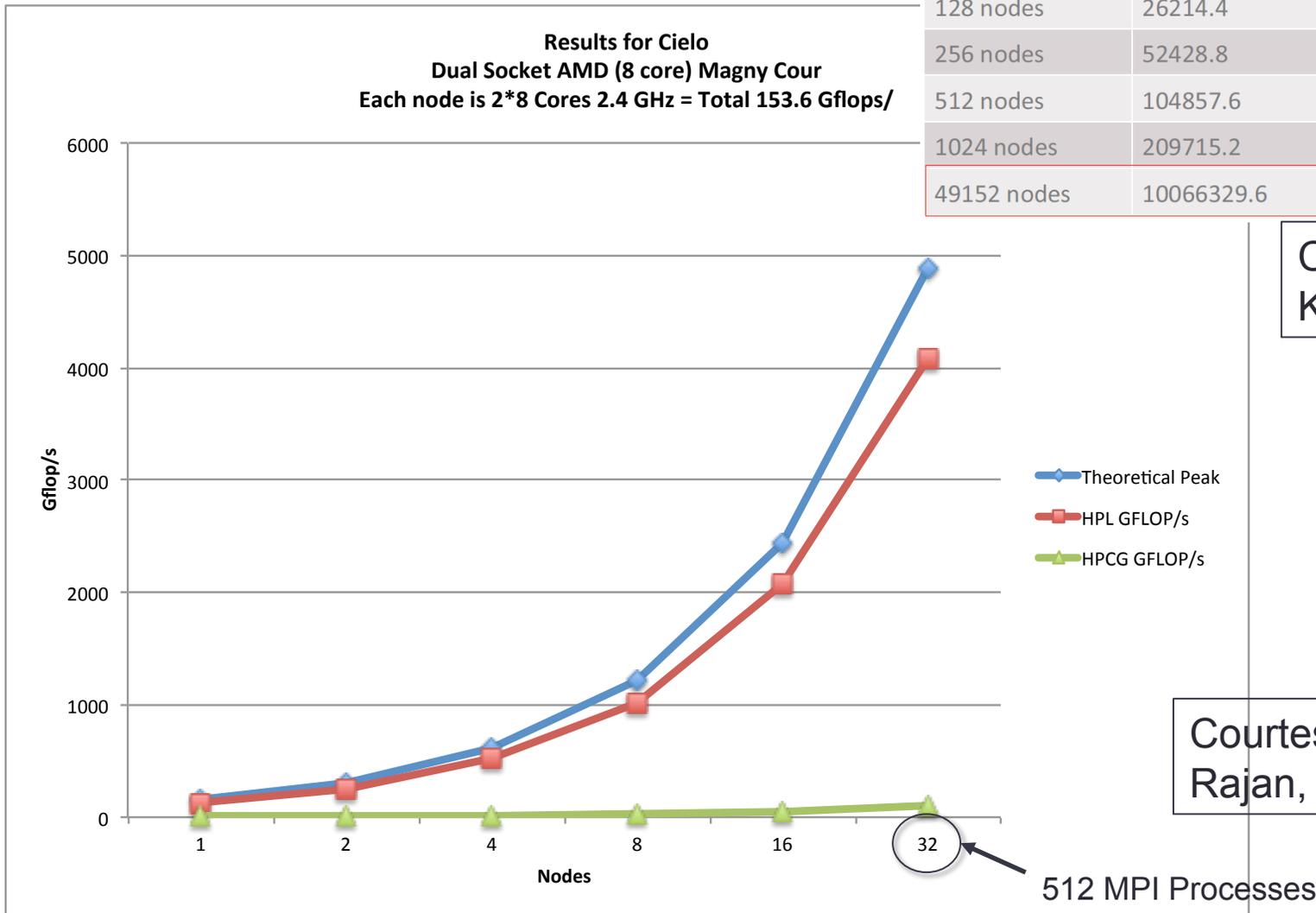
- Includes major communication/computational patterns.
 - Represents a minimal collection of the major patterns.
- Rewards investment in:
 - High-performance collective ops.
 - Local memory system performance.
 - Low latency cooperative threading.
- Detects and measures variances from bitwise identical computations.

COMPUTATIONAL RESULTS

GFLOPS/s “Shock”

Mira Partition Size	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%

Results for Cielo
Dual Socket AMD (8 core) Magny Cour
Each node is 2*8 Cores 2.4 GHz = Total 153.6 Gflops/



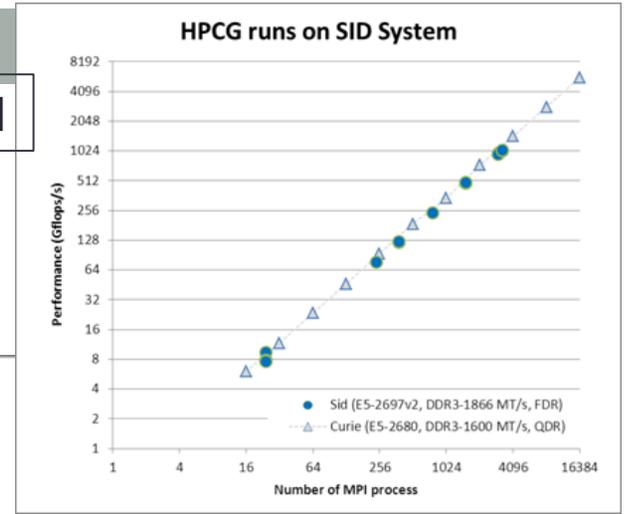
Courtesy Kalyan Kumaran, Argonne

Courtesy Mahesh Rajan, Sandia

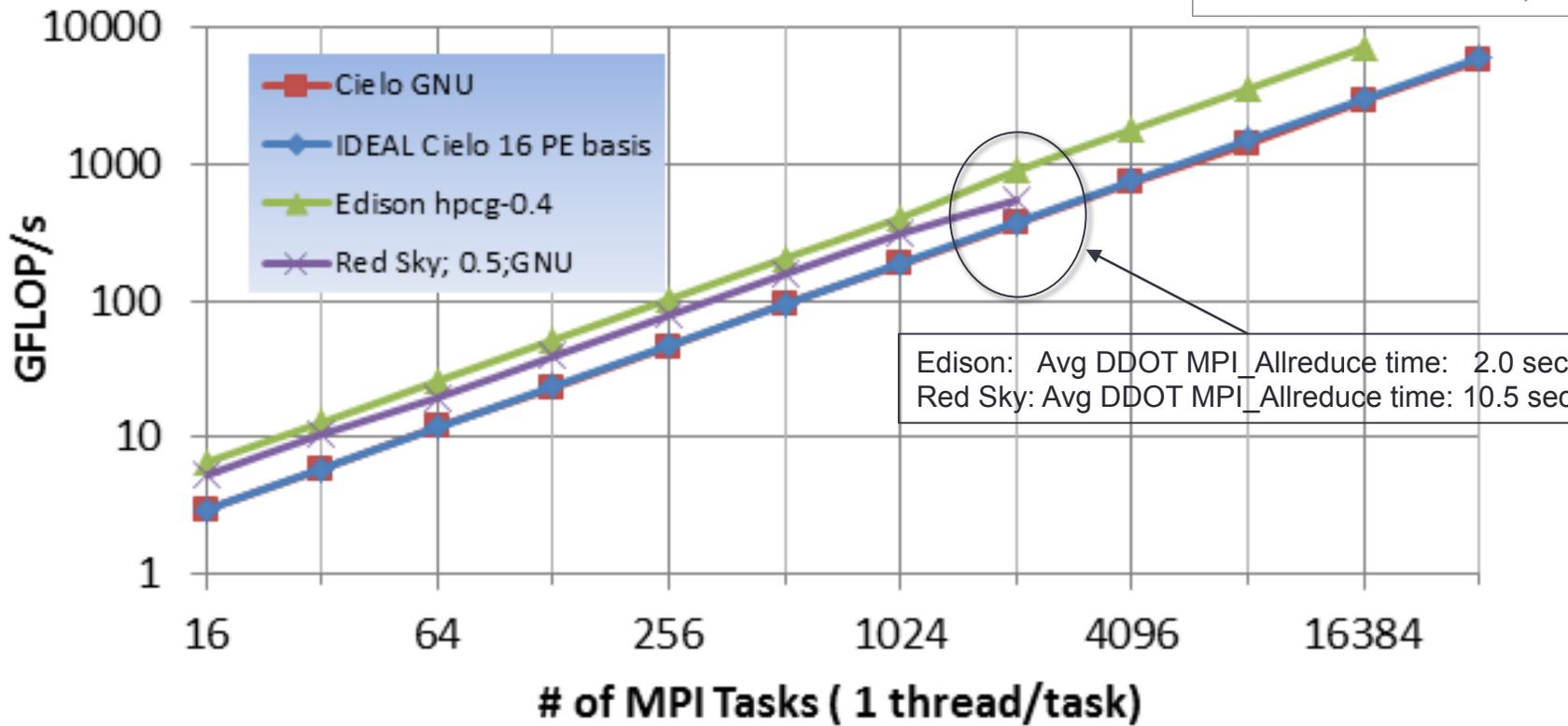
<http://tiny.cc/hpcg>

Results courtesy of Ludovic Saugé, Bull

Cielo, Red Sky, Edison, SID



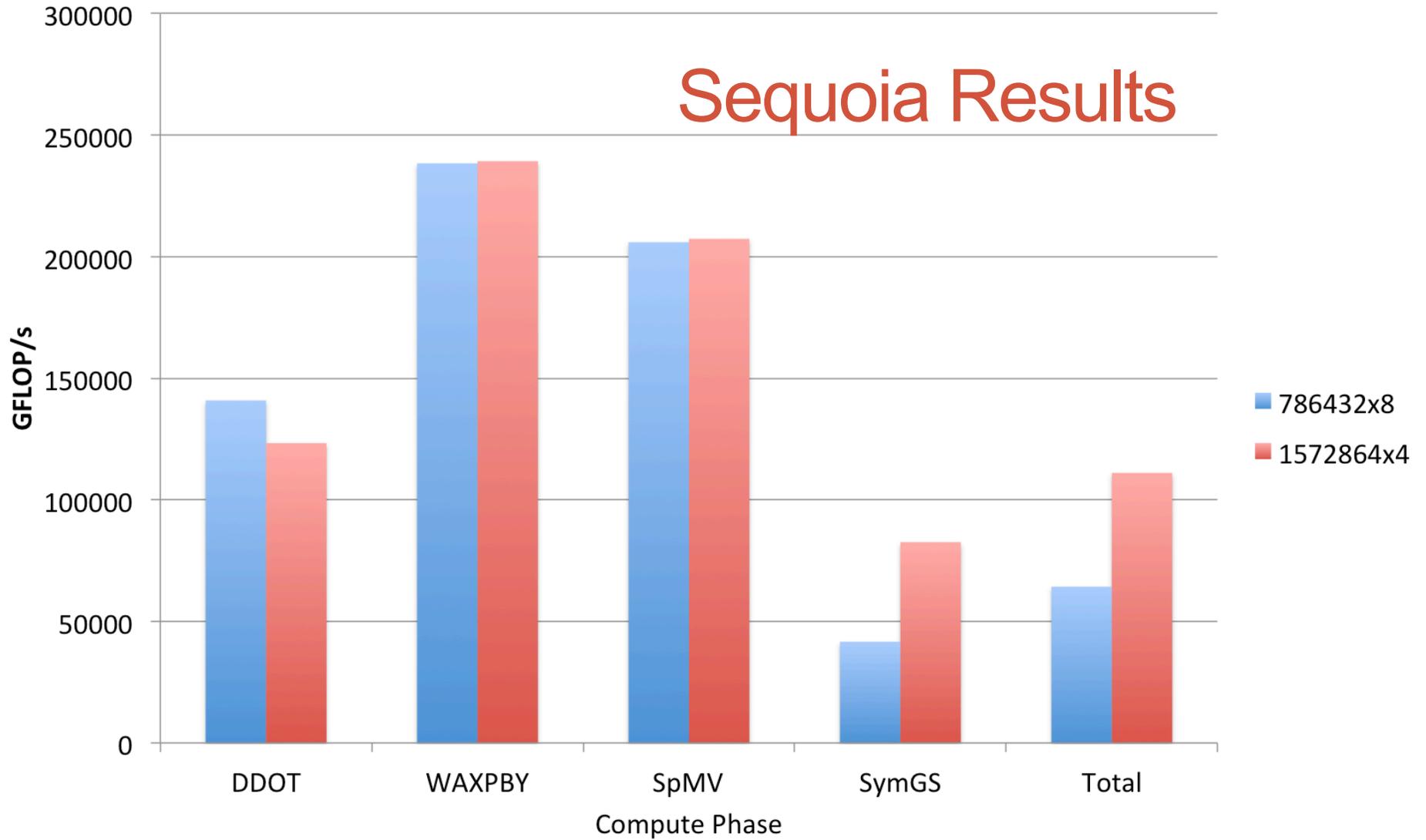
hpcg-0.4 or 0.5; GFLOP/s rating



Results courtesy of M. Rajan, D. Doerfler, Sandia

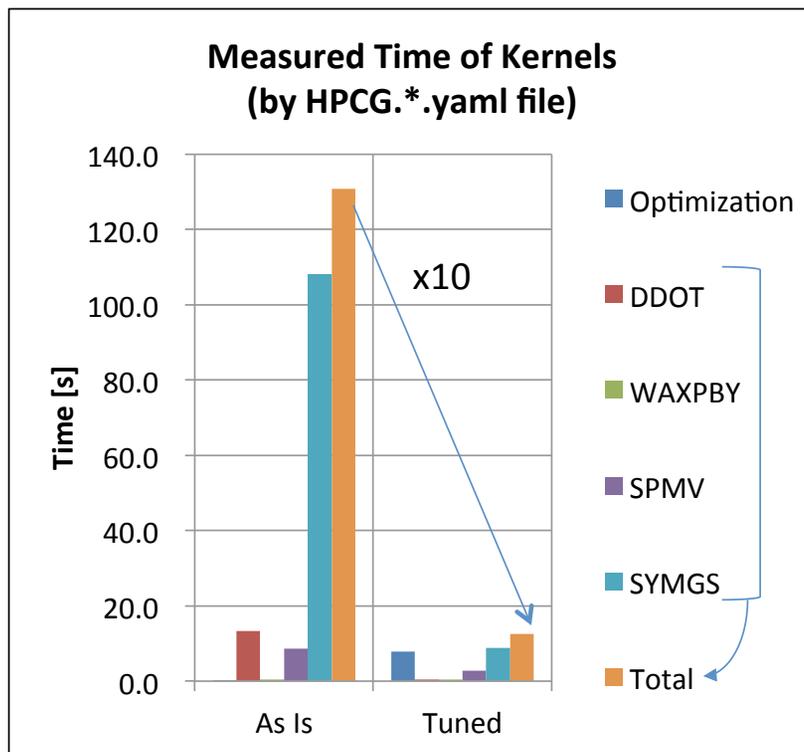
HPCG GFLOP/s on Sequoia: MPI x OpenMP
6.29M total threads, 1.57T equations

Sequoia Results



Results courtesy of Ian Karlin, Scott Futral, LLNL

Tuning result on the K computer



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

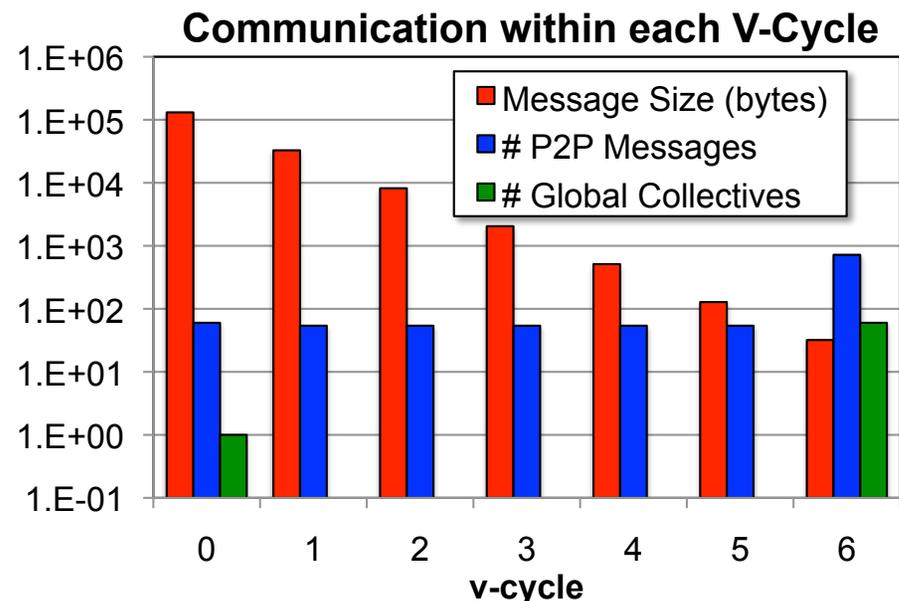
- Total x10 speed up now
 - Continuous memory for matrix
 - Multi-coloring for SYMGS multi-threading
- Under Studying
 - Node re-ordering for SPMV
 - Advanced matrix storage way
 - And so on

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

Slide courtesy Naoya Maruyama, RIKEN AICS, and Fujitsu

Next Steps

- Validate against real apps on real machines.
 - Validate ranking and driver potential.
 - Modify code as needed.
 - Considering multi-level preconditioner.
 - Discussions with LBL show potential to enrich design tradeoff space
 - Repeat as necessary.
- Introduce to broader community.
 - HPCG 1.0 released today.
- Notes:
 - Simple is best.
 - First version need not be last version (HPL evolved).



Graph courtesy Future Technology Group, LBL

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.

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

- <http://tiny.cc/hpcg>

SANDIA REPORT

SAND2013-8752
Unlimited Release
Printed October 2013

HPCG Technical Specification

Michael A. Heroux, Sandia National Laboratories¹
Jack Dongarra and Piotr Luszczek, University of Tennessee

Prepared by
Sandia National Laboratories

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 Laboratories¹

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