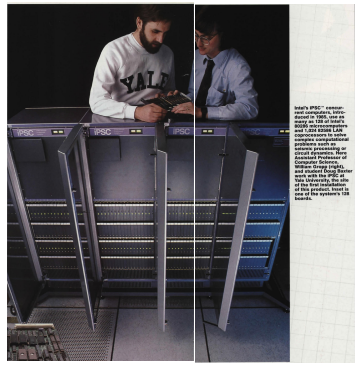
Introduction to High Performance Computing Trends and Opportunities

William Gropp
NCSA and Department of Computer Science
wgropp.cs.Illinois.edu

Assumptions and Caveats

- I'm not an economist
- I do have expertise in developing algorithms, tools, and applications for HPC
  - Algorithms for solving large systems of linear and nonlinear equations using parallel computers, especially equations from PDEs
  - Programming models and systems, including the parallel algorithms for their implementation on millions of processes (starting with a few in the 1980s)
  - Techniques for creating and managing code for performance
- I run a center whose mission is to solve the problems and challenges facing us today by using advanced computing and data
- I am assuming a wide range of HPC experience, and so I will focus on putting HPC in context and talking about the trends in computing
- Last year there were excellent presentations on XSEDE, which provides extensive support for the use of HPC resources supported by NSF
  - I won't duplicate much of that – look at those presentations or https://www.xsede.org/



What is High Performance Computing?

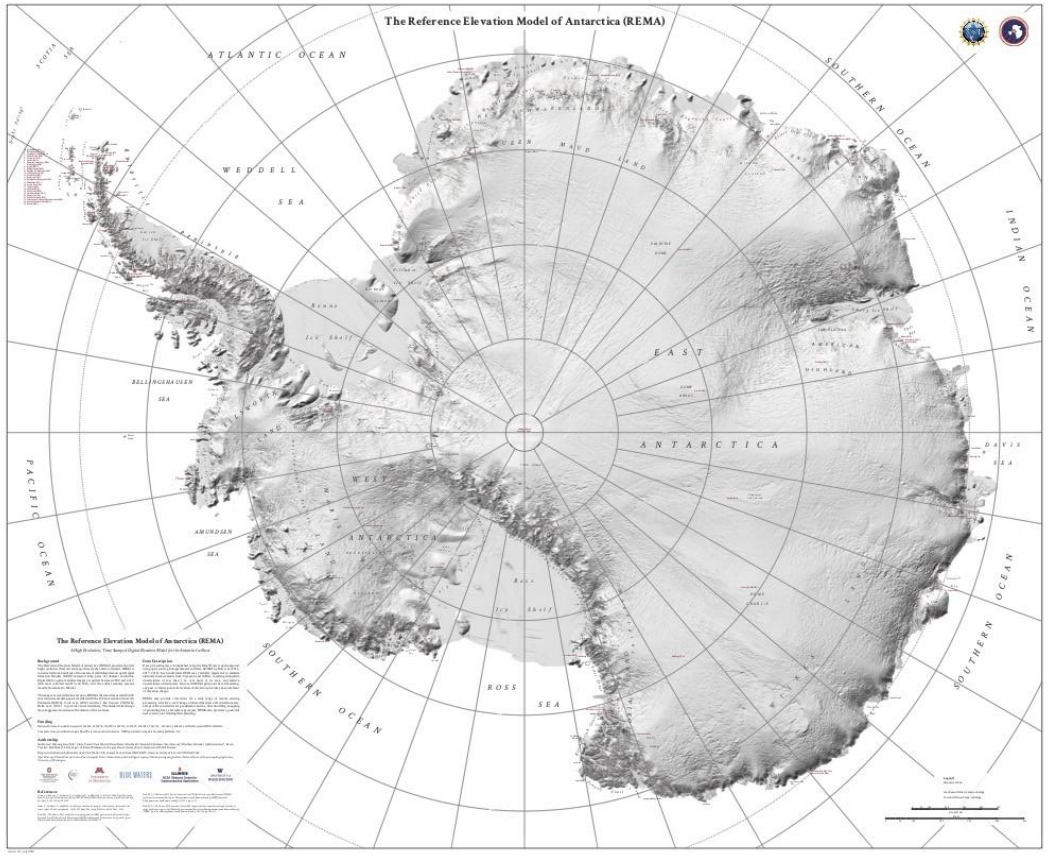
- My definition:
  Computing where performance is important
- Many different cuts, including
- By capability
  - Capability and Capacity
  - “Leadership” systems
- By use
  - “Tightly coupled”
  - High Throughput
  - Big data; Machine/Deep Learning
- By configuration
  - Homogenous and heterogeneous
  - With accelerators (e.g., GPU)
  - Cloud
- Note not a single metric for “high”

A *very* simplified view of computing.
Leadership systems now have accelerators of some sort, possibly integrated on chip

NAP: https://doi.org/10.17226/21886

Some Examples of the Use of HPC

- Simulations using Partial Differential Equations
  - A 10^4 x 10^4 x 10^4 grid is $10^{(12+1)}$ bytes – 10TB. Problems needing 10^{15} bytes (1 PetaByte) are solved today
- N-body simulations
  - Range from molecular dynamics of biomolecules to evolution of the universe
- Analysis of large data sets
  - Images, genome sequences, research publications
- Large collections of separate computations
  - Uncertainty quantification



The Reference Elevation Model of Antarctica (REMA)
https://www.pgc.umn.edu/data/rema/

HPC Hardware Architecture (at the extreme)

Next Generation System?
All Heterogeneous
Increasing diversity in accelerator choices

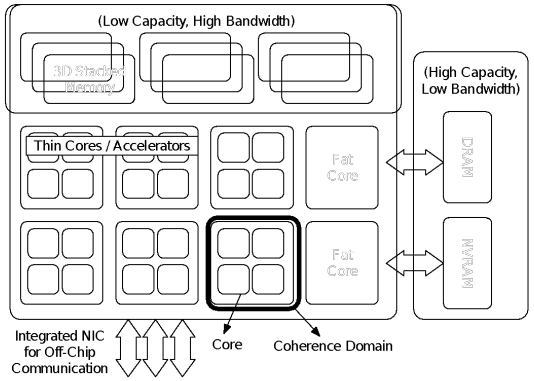
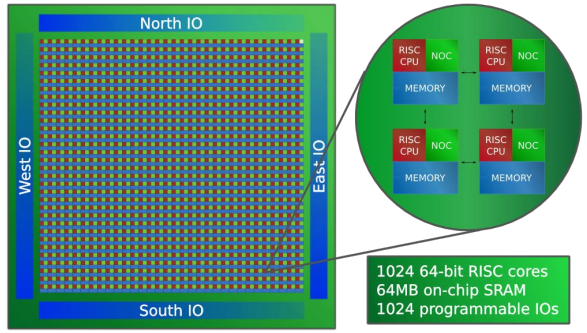
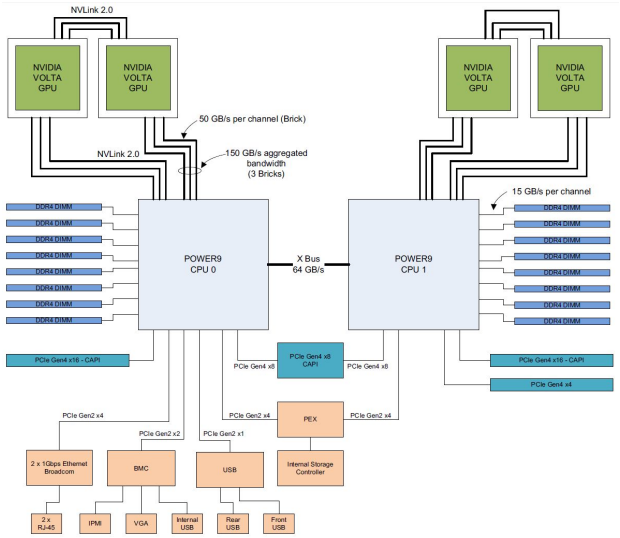


Figure 2.1: Abstract Machine Model of an exascale Node Architecture





From “Abstract Machine Models and Proxy Architectures for Exascale Computing Rev 1.1,” J Ang et al

Adapteva Epiphany-V

- 1024 RISC processors
- 32x32 mesh
- Very high power efficiency (70GF/W)

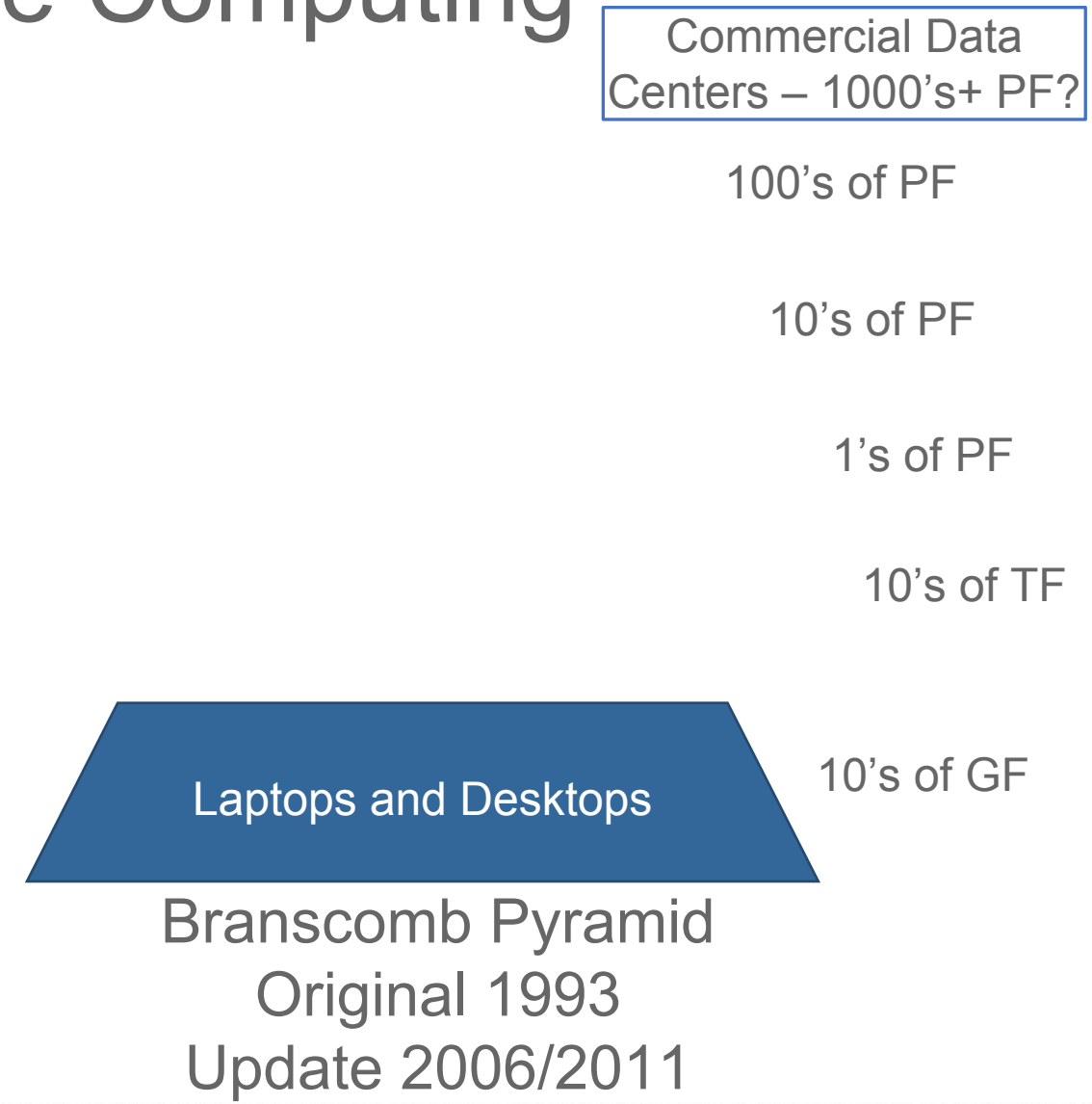
DOE Sierra

- Power 9 with 4 NVIDA Volta GPU
- 4320 nodes

NCSA Deep Learning System
16 nodes of Power 9 with 4 NVIDIA Volta GPU + FPGA

Trends in High Performance Computing

- Common to say computing performance grows exponentially
  - Consequence – just wait a while for more speed
- Moore's Law
  - Really an observation about semiconductor feature size
  - Relation to performance better described by *Dennard scaling*
- Reality more complex
  - The performance of different parts of computing systems have improved at vastly different rates
    - Floating point and integer oomputaitons
    - Memory access
    - Disk/SSD access
  - Moore's "Law" isn't – really an imperative that has become an expectation, driving progress – but limited by physics

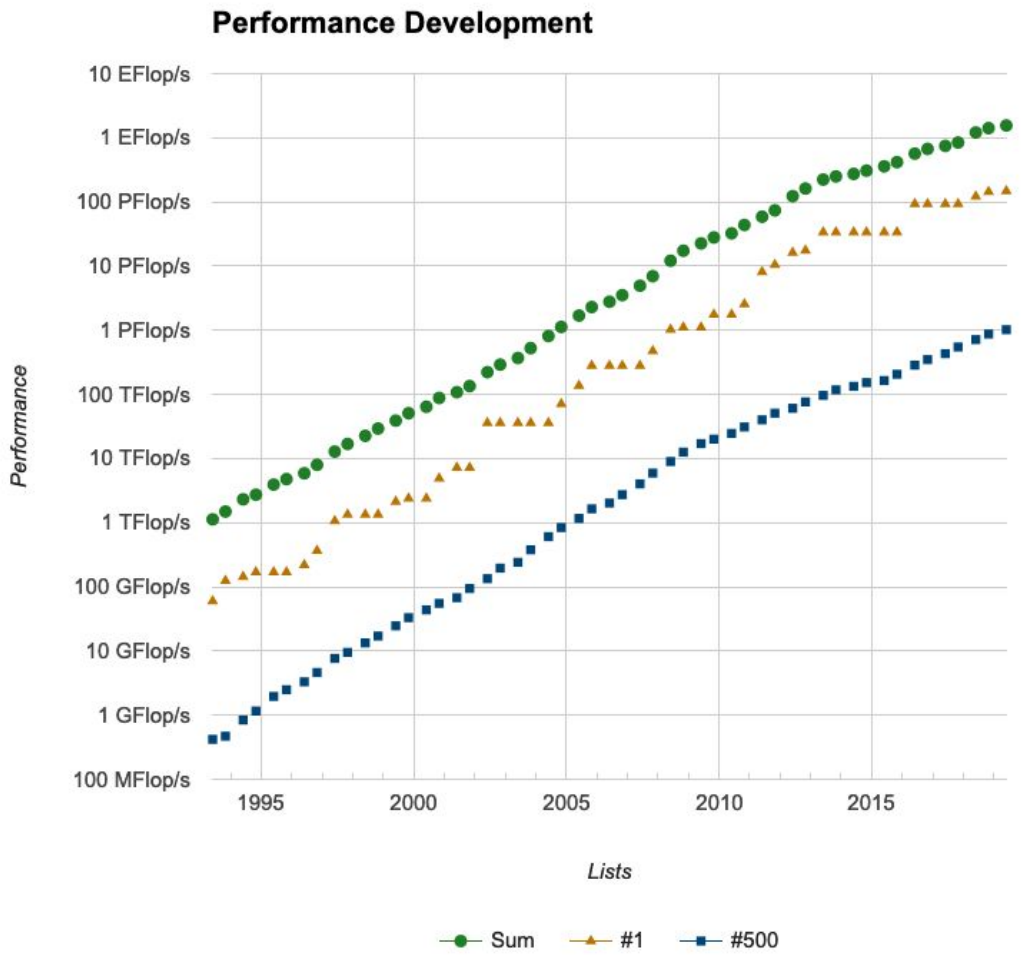


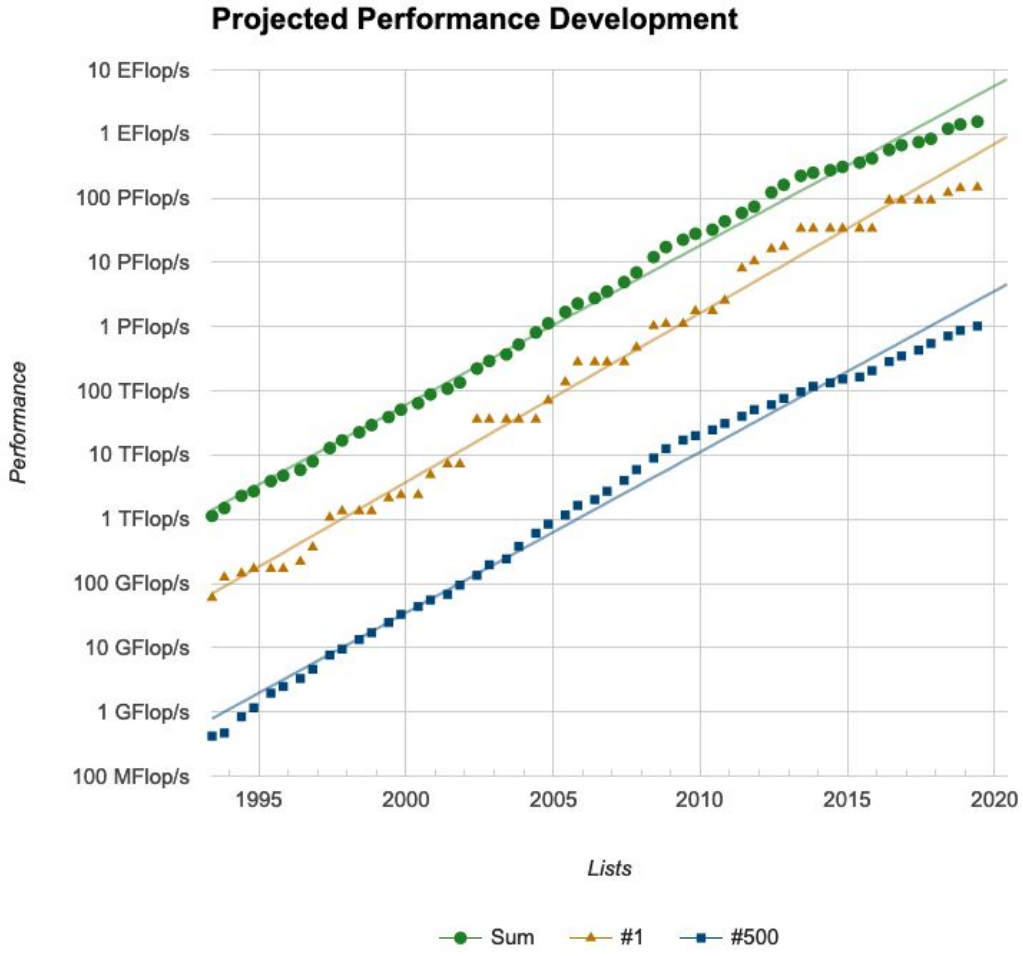
Branscomb Pyramid
Original 1993
Update 2006/2011

Measuring Performance: Benchmarks in HPC

- Model computations used in applications
- For HPC, best known is High Performance Linpack (HPL), and the list of top systems according to this is the top500 (top500.org)
- Solves a linear system of equations using Gaussian Elimination
  - System is *dense* – most (all in practice) matrix elements are non-zero
  - Representative of many numerical calculations when originally proposed
  - Not as representative today, but dense matrix operations on single cores/nodes common and important, e.g., spectral elements, deep learning
- Other benchmarks include
  - High Performance Conjugate Gradient (HPCG) – A sparse version of HPL; more like current PDE simulations
  - Graph 500 set, based on several graph kernels
  - Application-specific benchmarks, e.g., used for procurements, evaluations, …
- HPL data collected for over 26 years
  - What does it tell us about trends?

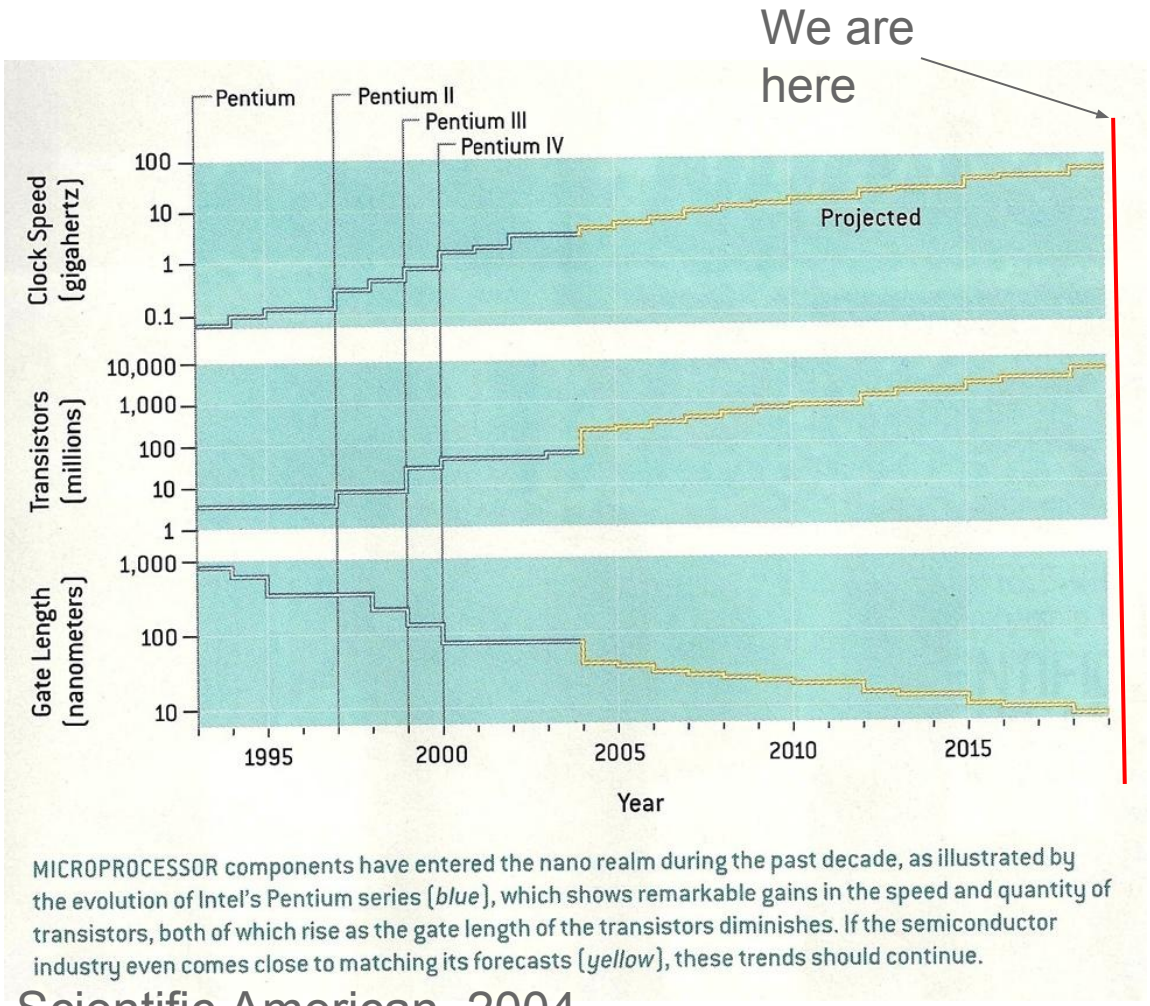
Top500 Performance Trends





Images from https://www.top500.org/statistics/perfdevel/

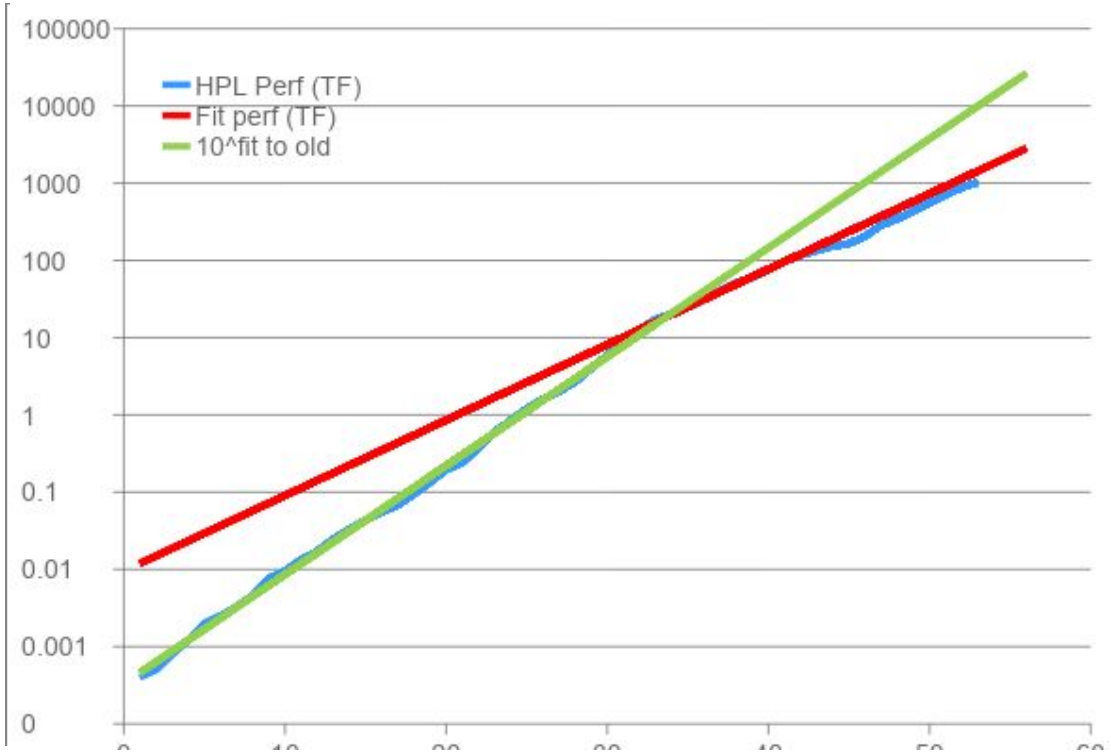
Two Views of Moore's "Law"



MICROPROCESSOR components have entered the nano realm during the past decade, as illustrated by the evolution of Intel's Pentium series (*blue*), which shows remarkable gains in the speed and quantity of transistors, both of which rise as the gate length of the transistors diminishes. If the semiconductor industry even comes close to matching its forecasts (*yellow*), these trends should continue.

Scientific American, 2004

The Bottom of the Top500



Two Lists of Top Systems – June 2017

Top500 – Dense Matrix

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel 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
3	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
4	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
5	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
6	DOE/SC/LBNL/NERSC United States	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	622,336	14,014.7	27,880.7	3,939

HPCG – Sparse Matrix

June 2017 HPCG Results

Rank	Site	Computer	Cores	HPL Rmax (Pflop/s)	TOP500 Rank	HPCG (Pflop/s)	Fraction of Peak
1	RIKEN Advanced Institute for Computational Science Japan	**K computer** – , SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10.510	8	0.6027	5.3%
2	NSCC / Guangzhou China	**Tianhe-2 (MilkyWay-2)** – TH-IVB-FEP Cluster, Intel Xeon 12C 2.2GHz, TH Express 2, Intel Xeon Phi 31S1P 57-core NUDT	3,120,000	33.863	2	0.5801	1.1%
3	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** – Sunway MPP, SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93.015	1	0.4808	0.4%
4	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** – Cray XC50, Intel Xeon E5-2690v3 12C 2.6GHz, Aries interconnect, NVIDIA Tesla P100 Cray	361,760	19.590	3	0.4767	1.9%
5	Joint Center for Advanced High Performance Computing Japan	**Oakforest-PACS** – PRIMERGY CX600 M1, Intel Xeon Phi Processor 7250 68C 1.4GHz, Intel Omni-Path Architecture Fujitsu	557,056	13.555	7	0.3855	1.5%
6	DOE/SC/LBNL/NERSC USA	**Cori** – XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries Cray	632,400	13.832	6	0.3554	1.3%

Figures from top500.org and hpcg-benchmark.org

Two Lists of Top Systems – June 2018

Top500 – Dense Matrix

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,282,544	122,300.0	187,659.3	8,806
2	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
3	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States	1,572,480	71,610.0	119,193.6	
4	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	32,576.6	1,649
6	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272

HPCG – Sparse Matrix

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	HPCG (TFlop/s)
1	1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,282,544	122,300.0	2925.75
2	3	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States	1,572,480	71,610.0	1795.67
3	16	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	602.74
4	9	**Trinity** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States	979,968	14,137.3	546.12
5	6	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	486.40
6	2	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	480.85

Figures from top500.org

Two Lists of Top Systems – June 2019

Leadership Systems

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	DOE/SC/Oak Ridge National Laboratory United States	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM	2,414,592	148,600.0	200,794.9	10,096
2	DOE/NNSA/LLNL United States	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband IBM / NVIDIA / Mellanox	1,572,480	94,640.0	125,712.0	7,438
3	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
4	National Super Computer Center in Guangzhou China	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 NUDT	4,981,760	61,444.5	100,678.7	18,482
5	Texas Advanced Computing Center/Univ. of Texas United States	**Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR Dell EMC	448,448	23,516.4	38,745.9	
6	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	387,872	21,230.0	27,154.3	2,384

Rank	TOP500 Rank	System	Cores	Rmax (TFlop/s)	HPCG (TFlop/s)
1	1	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	2925.75
2	2	**Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	1795.67
3	20	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect , Fujitsu RIKEN Advanced Institute for Computational Science (AICS) Japan	705,024	10,510.0	602.74
4	7	**Trinity** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect , Cray Inc. DOE/NNSA/LANL/SNL United States	979,072	20,158.7	546.12
5	8	**AI Bridging Cloud Infrastructure (ABCI)** - PRIMERGY CX2570 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	508.85
6	6	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	387,872	21,230.0	496.98
7	3	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	480.85

Figures from top500.org

NCSA

Two Graph Benchmarks (June 2018)

Breadth First Search

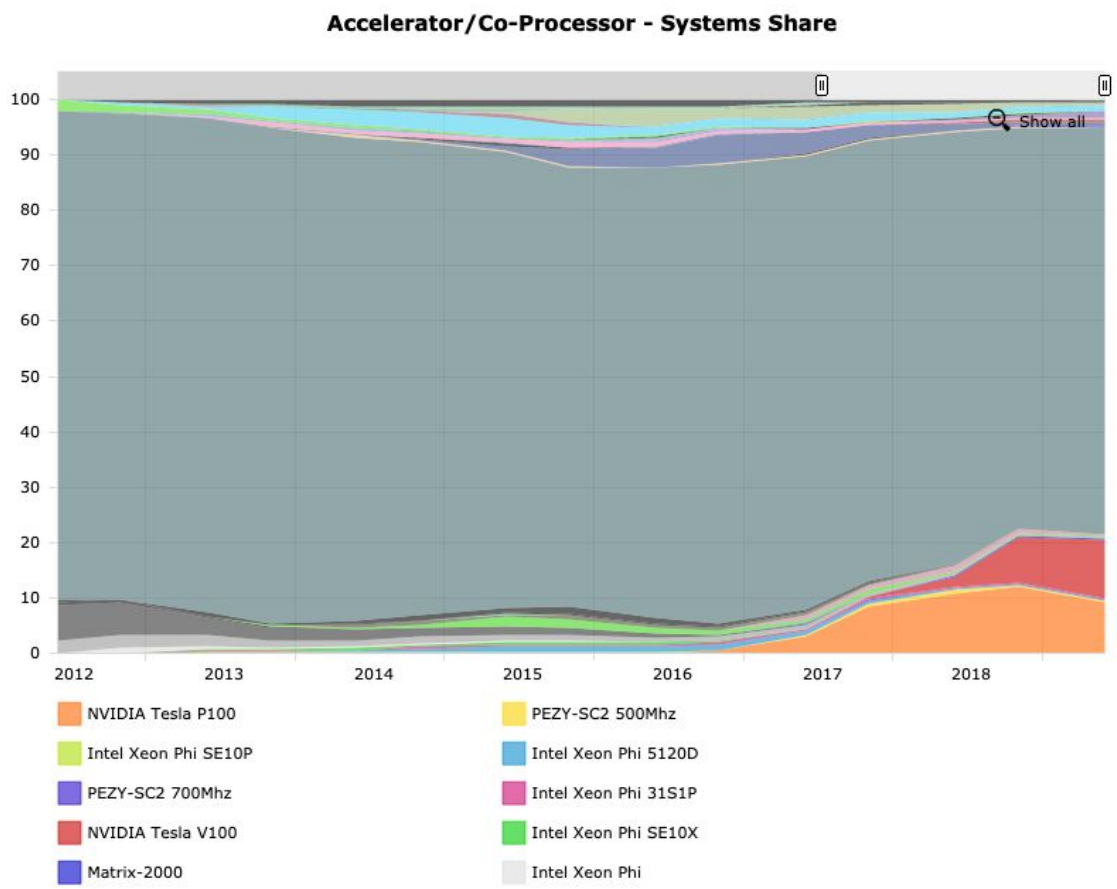
RANK	PREVIOUS RANK	MACHINE	VENDOR	TYPE	NETWORK	INSTALLATION SITE	LOCATION	COUNTRY
1	1	K computer	Fujitsu	Custom	Tofu	RIKEN Advanced Institute for Computational Science (AICS)	Kobe Hyogo	Japan
2	2	Sunway TaihuLight	NRCPC	Sunway MPP	Sunway	National Supercomputing Center in Wuxi	Wuxi	China
3	3	DOE/NNSA/LLNL Sequoia	IBM	BlueGene/Q Power BQC 16C 1.60 GHz	Custom	Lawrence Livermore National Laboratory	Livermore CA	USA
4	4	DOE/SC /Argonne National Laboratory Mira	IBM	BlueGene/Q Power BQC 16C 1.60 GHz	Custom	Argonne National Laboratory	Chicago IL	USA
5	5	JUQUEEN	IBM	BlueGene/Q Power BQC 16C 1.60 GHz	Custom	Forschungszentrum Juelich (FZJ)	Juelich	Germany

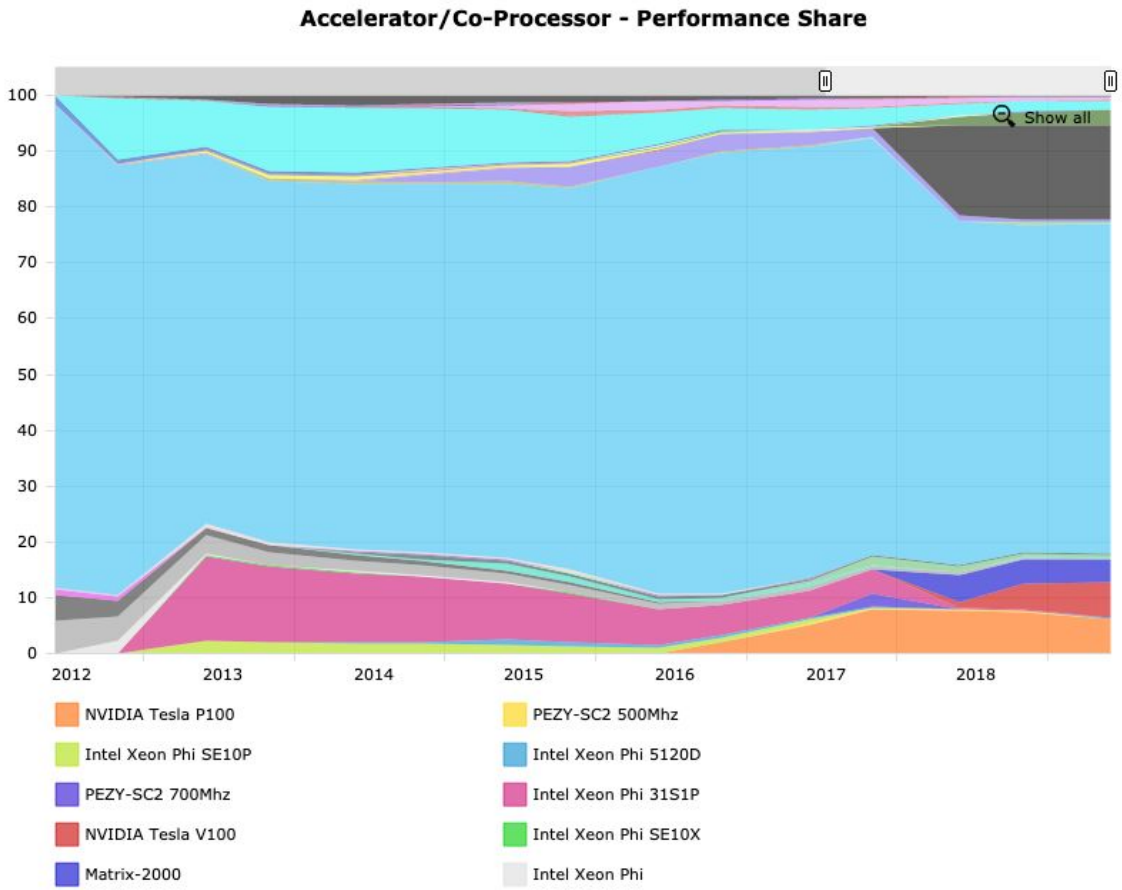
Single Source Shortest Path

RANK	PREVIOUS RANK	MACHINE	VENDOR	TYPE	NETWORK	INSTALLATION SITE	LOCATION	COUNTRY
1	1	Undisclosed Cray XE6	Cray	MPP	Gemini	National Computing Facility	University	United States
2	new	Alkindi-CPU	Dell	Single node commodity server		The University of British Columbia	Vancouver	Canada
3	new	Xeon Server	Dell	4 x Intel(R) Xeon(R) Gold 5115 CPU @ 2.40GHz		Industry	BoiseID	United States
4	2	University of Notre Dame cluster	Dell	cluster	Ethernet	University of Notre Dame	University of Notre Dame	United States
5	3	Alkindi27	Dell	Commodity Machine / Intel Xeon E5-2695 v3 (2 Sockets 28 cores 56 hardware		The University of British Columbia	Vancouver BC	Canada

Figures from graph500.org

Growth of Accelerators in the Top500





Some Observations

- Leading systems exploit specialized processors
  - NVIDIA GPUs for many; DSP-based engines for others
- Same approaches used in everything from cellphones to large data systems (e.g., Google TPU; Microsoft FPGA search accelerators)
- I/O, Memory often the critical resource
  - Compare HPCG and Graph500 to HPL performance
- Systems (and software and algorithms) optimized for algorithm/data structure combinations
  - Not just for science domains
  - Many opportunities to build communities around shared tools and expertise

Changing Landscape of Computing

- Dennard scaling ended more than a decade ago
  - The popular interpretation of “Moore’s Law” in terms of performance is really due to Dennard scaling
- Moore’s law (never really a “law”, more an imperative) is ending
  - Because of redefinition in terms of progress, will not have a definitive end date
- Ability to gather, share, combine, and explore data is creating new demands and opportunities
- Specialization is driving computer architecture, and hence software and algorithms
  - Not new, but the extent to which systems are specializing is making “business as usual” unworkable

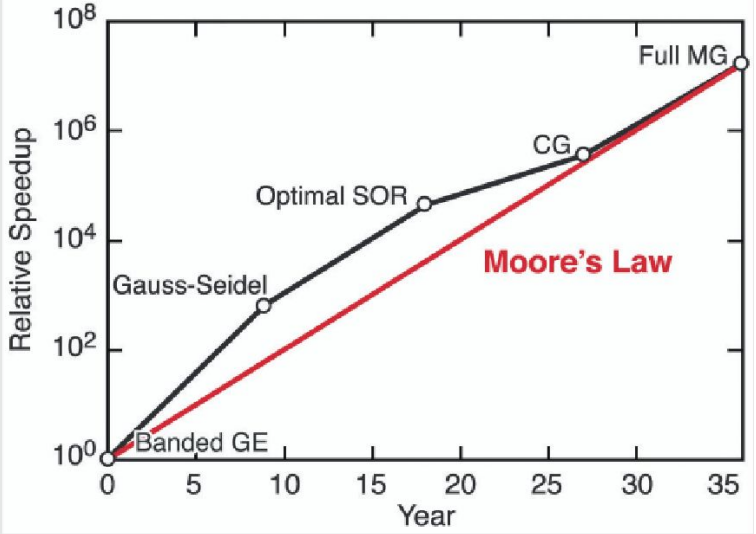
Do I really need HPC?

- How fast *should* my code run?
  - Performance models can help here
  - Should be based on the algorithms and data used
  - Typically needs to consider separately
    - Computations performance
    - Memory moved
      - Within a node
      - Between nodes
    - Data accessed
    - For parallel computations, also effective concurrency
  - Relatively simple models with a startup cost and an asymptotic rate often surprising effective
    - Some adjustments needed for multicore nodes, e.g.,
    - *Modeling MPI Communication Performance on SMP Nodes: Is it Time to Retire the Ping Pong Test*, W Gropp, L Olson, P Samfass, Proceedings of EuroMPI 16, https://doi.org/10.1145/2966884.2966919
- You can also rely on established applications and libraries that have been tuned for HPC systems

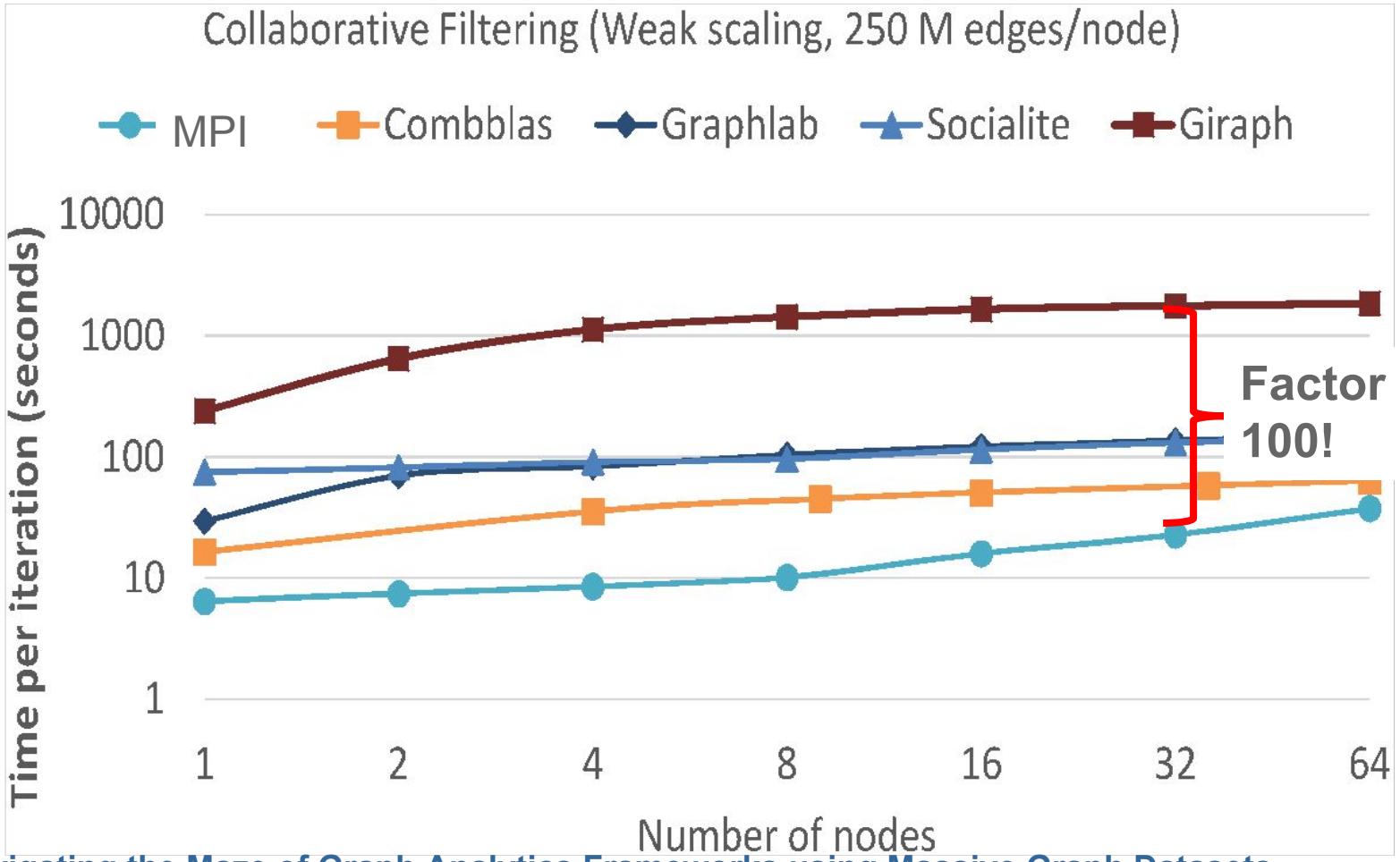
Algorithms Vs. Machines

- Is it better to improve the algorithm or the machine?
- Both of course!
- Algorithm improvements have been substantial – E.g. solve a sparse linear system
- Algorithm and Hardware improvement provided similar speedup*
- *Note that Full MG is O(1) per mesh point – no more room to improve asymptotically
  - Without changing the problem – different model, different approximation, etc.

Method	Storage	Flops
GE (banded)	n^5	n^7
Gauss-Seidel	n^3	$n^5 \log n$
Optimal SOR	n^3	$n^4 \log n$
CG	n^3	$n^{3.5} \log n$
Full MG	n^3	n^3



One Example of Tradeoff in Performance and Productivity



Navigating the Maze of Graph Analytics Frameworks using Massive Graph Datasets
Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Jiwon Seo, Jongsoo Park, M. Amber Hassaan, Shubho Sengupta, Zhaoming Yin, and Pradeep Dubey; Proceedings of SIGMOD'14

Diversion: Where are the real problems in using HPC Systems?

- HPC Focus is typically on scale
  - “How will we program a million (or a billion) cores?
  - “What can use use to program these machines?”
- The real issues are often overlooked
  - Performance models still (mostly) process to process and single core
    - Node bottlenecks missed; impacts design from hardware to algorithms
  - Dream of “Performance Portability” stands in the way of practical solutions to “transportable” performance
  - Increasingly complex processor cores and nodes
  - HPC I/O requirements impede performance, hurt reliability

Programming Models and Systems

- In past, often a tight connection between the execution model and the programming approach
  - Fortran: FORmula TRANslation to von Neumann machine
  - C: e.g., “register”, ++ operator match PDP-11 capabilities, needs
- Over time, execution models and reality changed but programming models rarely reflected those changes
  - Rely on compiler to “hide” those changes from the user – e.g., auto-vectorization for SSE(n)
- Consequence: Mismatch between users’ expectation and system abilities.
  - Can’t fully exploit system because user’s mental model of execution does not match real hardware
  - Decades of compiler research have shown this problem is extremely hard – can’t expect system to do everything for you.

The Easy Part – Internode communication

- Often focus on the “scale” in extreme scale as the hard part
  - How to deal with a million or a billion processes?
  - But really not too hard
    - Many applications have large regions of regular parallelism
  - Or nearly impossible
    - If there isn’t enough independent parallelism
  - Challenge is in handling definition and operation on distributed data structures
  - Many solutions for the internode programming piece
  - The dominant one in technical computing is the Message Passing Interface (MPI)

Modern MPI

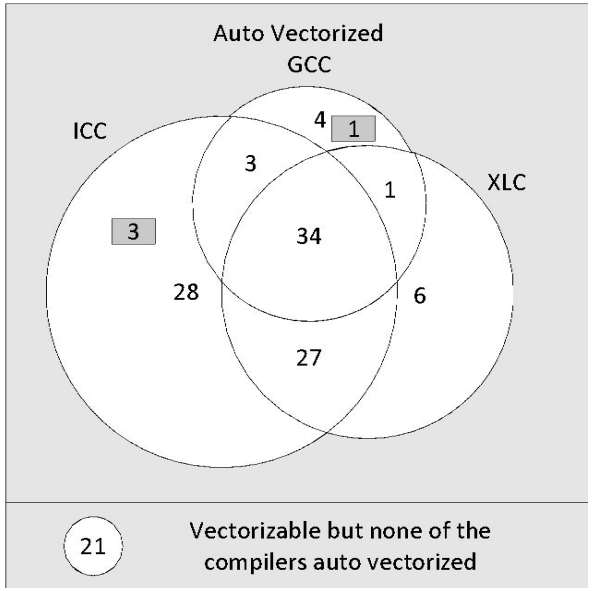
- MPI is much more than message passing
  - I prefer to call MPI a programming *system* rather than a programming *model*
    - Because it implements several programming *models*
- Major features of MPI include
  - Rich message passing, with nonblocking, thread safe, and persistent versions
  - Rich collective communication methods
  - Full-featured one-sided operations
    - Many new capabilities over MPI-2
    - Include remote atomic update
  - Portable access to shared memory on nodes
    - Process-based alternative to sharing via threads
    - (Relatively) precise semantics
  - Effective parallel I/O that is not restricted by POSIX semantics
    - But see implementation issues …
  - Perhaps most important
    - Designed to support “programming in the large” – creation of libraries and tools
- MPI continues to evolve – MPI “next” Draft released at SC in Dallas last November

Applications Still Mostly MPI-Everywhere

- "the larger jobs (> 4096 nodes) mostly use message passing with no threading." – Blue Waters Workload study, https://arxiv.org/ftp/arxiv/papers/1703/1703.00924.pdf
- Benefit of programmer-managed memory locality
  - Memory performance nearly stagnant (will High Bandwidth Memory save us?)
  - Parallelism for performance implies locality must be managed effectively
- Benefit of a single programming system
  - Often stated as desirable but with little evidence
  - Common to mix Fortran, C, Python, etc.
  - But…Interface between systems must work well, and often don't
    - E.g., for MPI+OpenMP, who manages the cores and how is that negotiated?

The Hard Part: Intranode Performance Example: Generating Fast Code for Loops

- Long history of tools and techniques to produce fast code for loops
  - Vectorization, streams, etc., dating back nearly 40 years (Cray-1) or more
- Many tools for optimizing loops for both CPUs and GPUs
  - Compiler (auto) vectorization, explicit programmer use of directives (e.g., OpenMP or OpenACC), lower level expressions (e.g., CUDA, vector intrinsics)
- Is there a clear choice?
  - Not for vectorizing compilers (e.g., see S. Maleki, Y. Gao, T. Wong, M. Garzarán, and D. Padua, *An Evaluation of Vectorizing Compilers*. PACT 2011)
  - Probably not for the others
    - Similar results for GPU programming
  - Vector tests part of baseenv; OpenACC and OpenMP vectorization tests under development (and some OpenACC examples follow)
- Need to separate description of semantics and operations from particular programming system choices



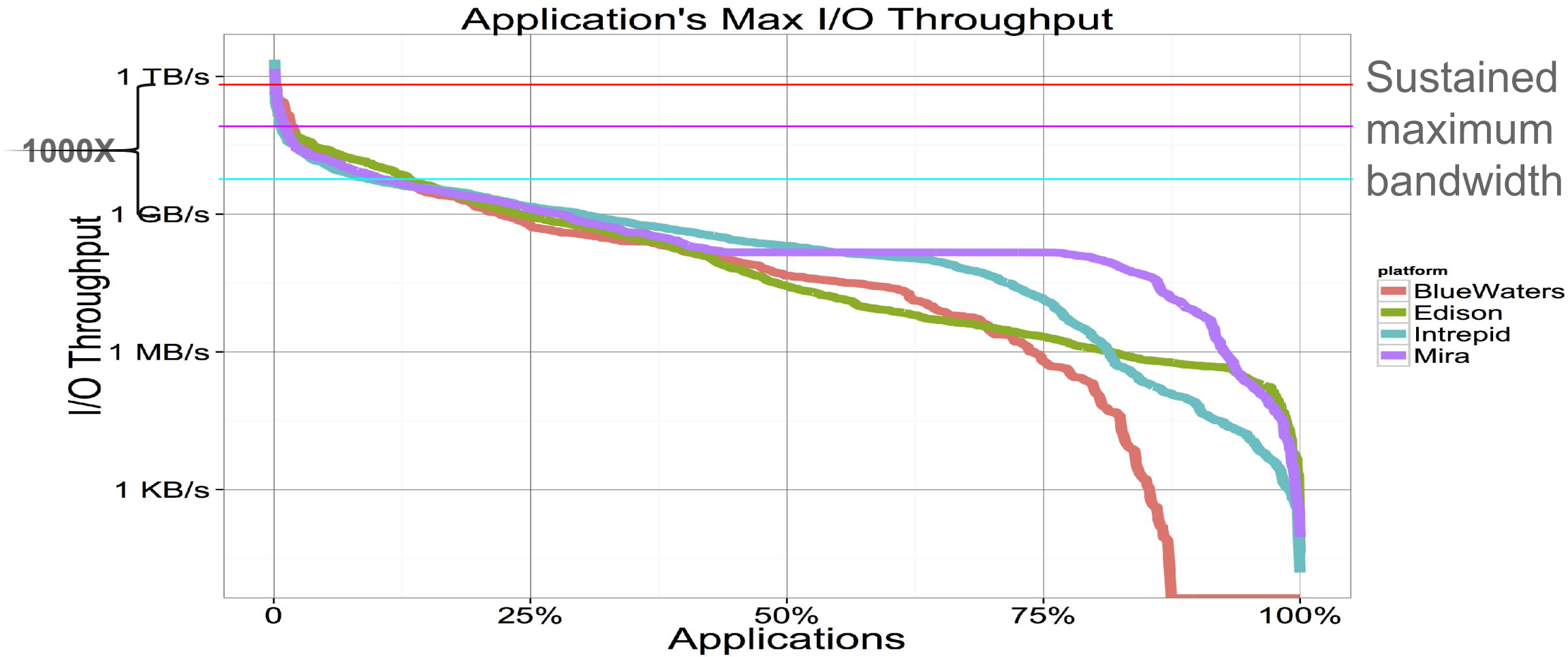
Often Overlooked – IO Performance Often Terrible

- Applications just assume I/O is awful and can't be fixed
- Even simple patterns not handled well
- Example: read or write a submesh of an N-dim mesh at an arbitrary offset in file
- Needed to read input mesh in PlasComCM. Total I/O time less than 10% for long science runs (that is < 15 hours)
  - But long init phase makes debugging, development hard

	Original	Meshio	Speedup
PlasComCM	4500	1	4500
MILC	750	15.6	48

- Meshio library built to match application needs
- Replaces many lines in app with a single *collective* I/O call
- Meshio https://github.com/oshkosher/meshio
- Work of Ed Karrels

Just how bad Is current I/O performance?



Sustained maximum I/O bandwidth

"*A Multiplatform Study of I/O Behavior on Petascale Supercomputers*," Huong Luu, Marianne Winslett, William Gropp, Robert Ross, Philip Carns, Kevin Harms, Prabhat, Suren Byna, and Yushu Yao, proceedings of HPDC'15.

Summary: Challenges in Building HPC Applications

- Popular focus in on the computation
- Much of the limitations in performance are due to memory
  - Is the data needed available?
  - How easy and fast is it to access?
  - Data moves in aggregates (e.g., cache lines, memory rows, network packets, disk blocks)
- All is not lost!
  - A hierarchy of tools exist
    - Low-level programming systems (C/Fortran, OpenMP, MPI, CUDA, OpenACC, …)
    - Software libraries on top of these (PETSc, Trilinos, SCALAPACK, …)
    - Higher level systems on top of those (Matlab, R, Python, …)
    - Applications and workflows on these (NAMD, LS-DYNA, …)
- (End of Diversion)

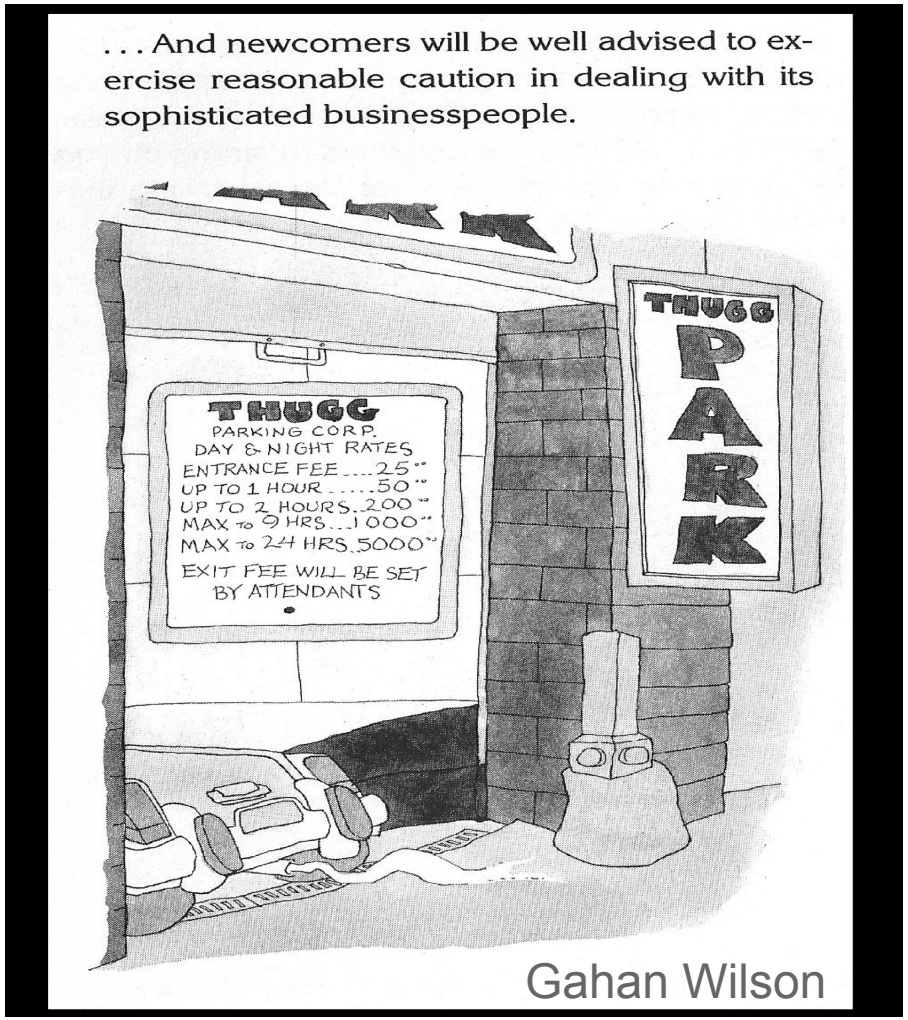
Sources of HPC for research

- Federal agencies
  - NSF through XSEDE and PRAC
  - DOE through INCITE
  - Other agencies through specific systems – e.g., DoD HPCMP
- Institutions
  - Many provide some shared resource
    - Illinois has 324 nodes in a campus cluster + a share of Blue Waters + others
    - Indiana just announced a substantial Cray supercomputer
    - …
  - It appears NSF is increasing expecting institutions to provide some HPC for researchers
- Cloud (commercial and otherwise)
  - Cycles cannot be stored, so if you are *very* flexible, you may be able to get a great deal – e.g., https://aws.amazon.com/ec2/spot/ https://cloud.google.com/preemptible-vms/ https://docs.microsoft.com/en-us/azure/batch/batch-low-pri-vms https://www.ibm.com/cloud/blog/transient-virtual-servers

Requesting Time on National Resources

- Ask for time (national requests)
  - https://portal.xsede.org/allocations/research
  - https://www.olcf.ornl.gov/2019/04/15/incite2020/ (closed in June, but annual call)
  - https://science.osti.gov/ascr/Facilities/Accessing-ASCR-Facilities (ASCR general info)
  - https://www.hpc.mil/ (for work on DOD Grants – see https://www.hpc.mil/images/hpcdocs/users/New_Users_Brief_2_Who_May_Run_on_HPCMP_Resources_rev2_distro_a.pdf)
  - Check your favorite agency
- Take advantage of “exploratory” or “startup” allocations to benchmark, show readiness
  - E.g, https://portal.xsede.org/allocations/startup
- Focus on science, but also show that you are using the resource wisely
  - XSEDE 5x oversubscribed; DOE INCITE similar or worse
- Look for partners with complementary expertise
  - Or try to start as part of an established project

Clouds and All That



- Cloud computing is a term for many things, including
  - Load sharing among users
  - Flexible allocation on demand
  - Framework for data and software sharing
- Multiple cost studies show beneficial for users (compared to a dedicated system) with
  - Less than 20-30% use of a dedicated system
  - Highly variable use that is uncorrelated with other users
- Typical supercomputing systems run at 80+% utilization
  - Clouds would be *more* expensive
- You don't need to believe anyone – do the numbers yourself
  - But do them carefully!

More on Cloud Cost studies

- DOE Magellan report
  - https://www.osti.gov/scitech/servlets/purl/1076794
  - Measured performance (hence cost) on cloud system, compared to supercomputer centers
  - Clouds 3-5X *more* expensive
  - Not surprising – margins thin on hardware, availability on demand requires excess capacity – cost is higher
- National Academy Report
  - http://tinyurl.com/advcomp17-20
  - Update on cloud pricing
    - Adds File (Data) I/O, networking
    - Compute power vague (achieved performance more dependent on memory bandwidth, latency, cache capabilities)
    - Magellan conclusion still hold
- NASA Report "Evaluating the Suitability of Commercial Clouds for NASA's High Performance Computing Applications: A Trade Study"
  - https://www.nas.nasa.gov/assets/pdf/papers/NAS_Technical_Report_NAS-2018-01.pdf
  - Another update on cloud pricing; similar results
- "Do the numbers"

Where do Clouds Fit?

- Clouds provide a complimentary service model
  - Access to systems (different configurations, sometimes at scale)
  - On-demand access
  - Access to different (and often newer) software frameworks
  - Easy ways to share data with services (“Data Lakes”)
- Complement center resources
  - Lower cost but not on demand
  - Expert support (not uncommon to get 2-10x performance improvement)
  - Increasing real-time needs for instruments
- Not either/or – can and do use both to solve problems

https://aws.amazon.com/blogs/aws/natural-language-processing-at-clemson-university-1-1-million-vcpus-ec2-spot-instances/

Summary

- High performance computing is any computing where performance is important
- The technology (both hardware and software) are mostly familiar
  - Programming languages, operating systems and runtimes, nodes are often “server” versions of commodity products
  - HPC ecosystem typically batch-oriented (for reasons of cost), but alternative interfaces such as science gateways are available
- Parallelism (needed for performance and scaling of resources such as memory) does introduce challenges
  - In software and in algorithms
- The technology is going though a disruptive period
  - End of Dennard scaling leading to architectural innovation, ending over 30 years of hardware and software stability
- There are many sources of HPC help
  - Access to systems
  - Advice and help on applications and workflows
  - Communities of users and developers

Questions For the Workshop

- What is the greatest challenge (e.g., access, software, performance, productivity?)
- What are the three things that would have the most impact?
- What are the most important categories of HPC needed? At what scale?
- How should different communities organize to accelerate their use of HPC?