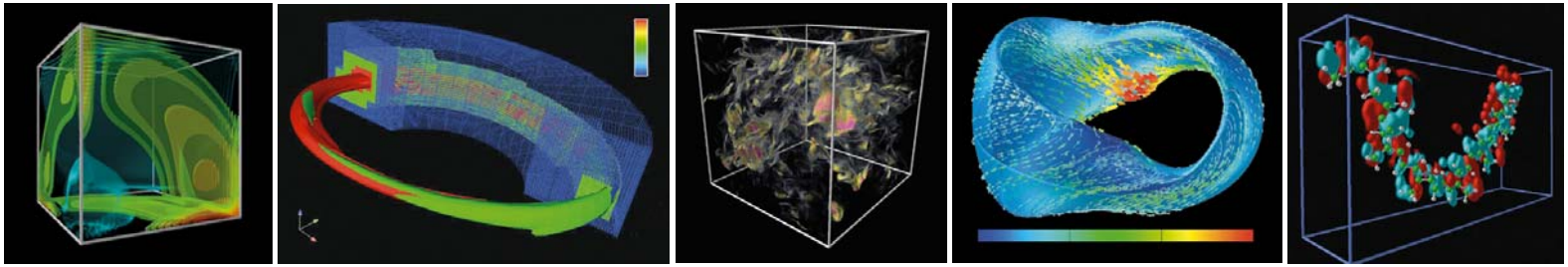


Progress in Supercomputing: The Top Three Breakthroughs of the Last 20 Years and the Top Three Challenges for the Next 20 Years

Horst Simon
Associate Laboratory Director
Lawrence Berkeley National Laboratory

ISC 2005 Heidelberg
June 22, 2005

BERKELEY LAB

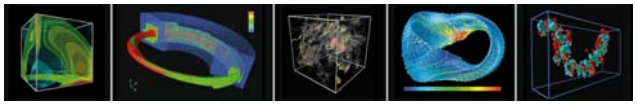
NERSC

Office of Science
U.S. DEPARTMENT OF ENERGY

Signpost System 1985

Cray-2

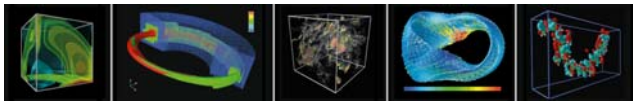
- 244 MHz (4.1 nsec)
- 4 processors
- 1.95 Gflop/s peak
- 2 GB memory (256 MW)
- 1.2 Gflop/s LINPACK R_max
- 1.6 m^2 floor space
- 0.2 MW power



Signpost System in 2005

IBM BG/L @ LLNL

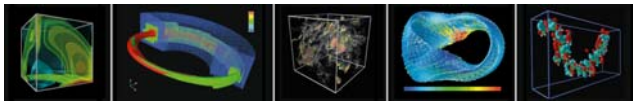
- 700 MHz (x 2.86)
- 65,536 nodes (x 16,384)
- 180 (360) Tflop/s peak (x 92,307)
- 32 TB memory (x 16,000)
- 135 Tflop/s LINPACK (x 110,000)
- 250 m^2 floor space (x 156)
- 1.8 MW power (x 9)



1985 versus 2005

- custom built vector mainframes
- 30 Mflops sustained is good performance
- vector Fortran
- proprietary operating system
- remote batch only
- no visualization
- no tools, hand tuning only
- dumb terminals
- remote access via 9600 baud
- single software developer, develops and codes everything
- serial, vectorized algorithms

- commodity massively parallel platforms
- 1 Tflops sustained is good performance
- Fortan/C with MPI, object orientation
- Unix, Linux
- interactive use
- visualization
- parallel debugger, development tools
- high performance desktop
- remote access via 10 Gb/s; grid tools
- large group developed software, code share and reuse
- parallel algorithms



The Top 10 Major Accomplishments in Supercomputing 1985 – 2005

- My own personal opinion
- Selected by “impact” and “change in perspective”

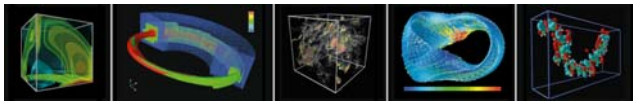
10) The TOP500 list

9) NAS Parallel Benchmark

8) The “grid”

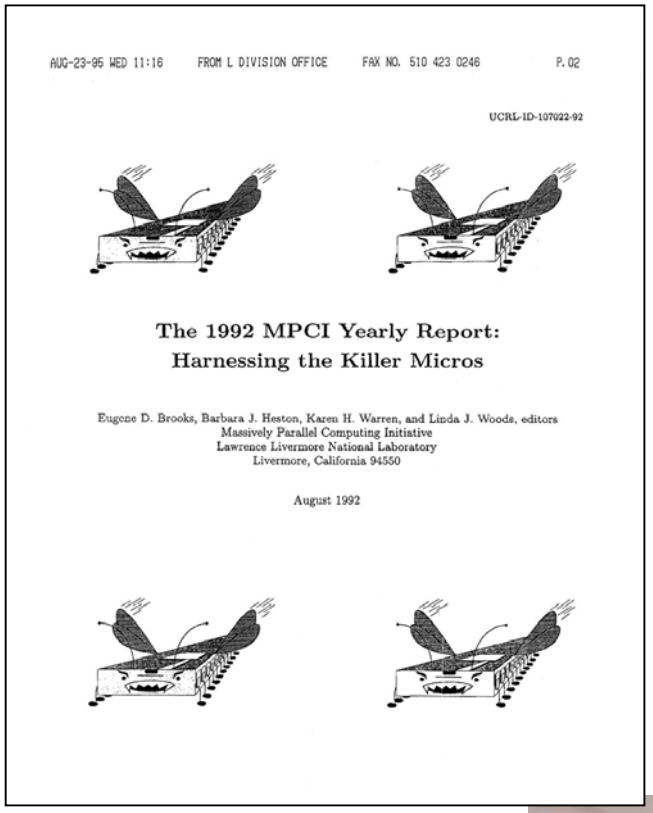
7) Hierarchical algorithms: multigrid and fast multipole methods

6) HPCC initiative and Grand Challenge applications



#5 The "Attack of the Killer Micros"

- First used by Eugene Brooks (LLNL) at Supercomputing 89
- Became a catchy shorthand expression for the technology change from custom ECL to commodity CMOS
- Commodity CMOS micro processors did change the face of supercomputing, but they were neither inevitable, nor the only technology choice



AUG-23-95 WED 11:16 FROM L DIVISION OFFICE FAX NO. 510 423 0246 P.02

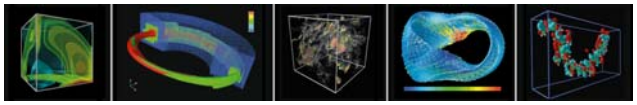
UCRL-ID-107022-92

The 1992 MPCI Yearly Report:
Harnessing the Killer Micros

Eugene D. Brooks, Barbara J. Heston, Karen H. Warren, and Linda J. Woods, editors
Massively Parallel Computing Initiative
Lawrence Livermore National Laboratory
Livermore, California 94550

August 1992

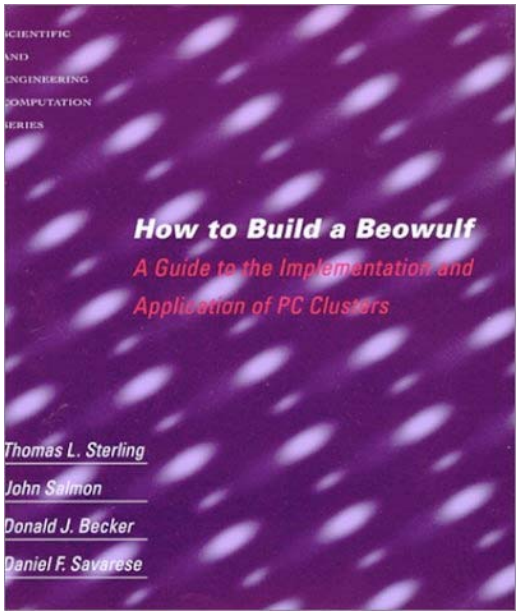
Intel I860XP - a killer micro at its time



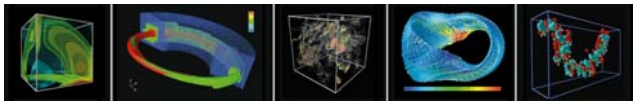
Office of Science
U.S. DEPARTMENT OF ENERGY

#4 Beowulf Clusters

- Thomas Sterling et al. established vision of low cost, high end computing
- Demonstrated effectiveness of PC clusters for some (not all) classes of applications
- Provided software and conveyed findings to broad community (great PR) through tutorials and book (1999)
- Made parallel computing accessible to large community worldwide; broadened and democratized HPC
- However effectively stopped HPC architecture innovation for at least a decade; narrower market for custom systems

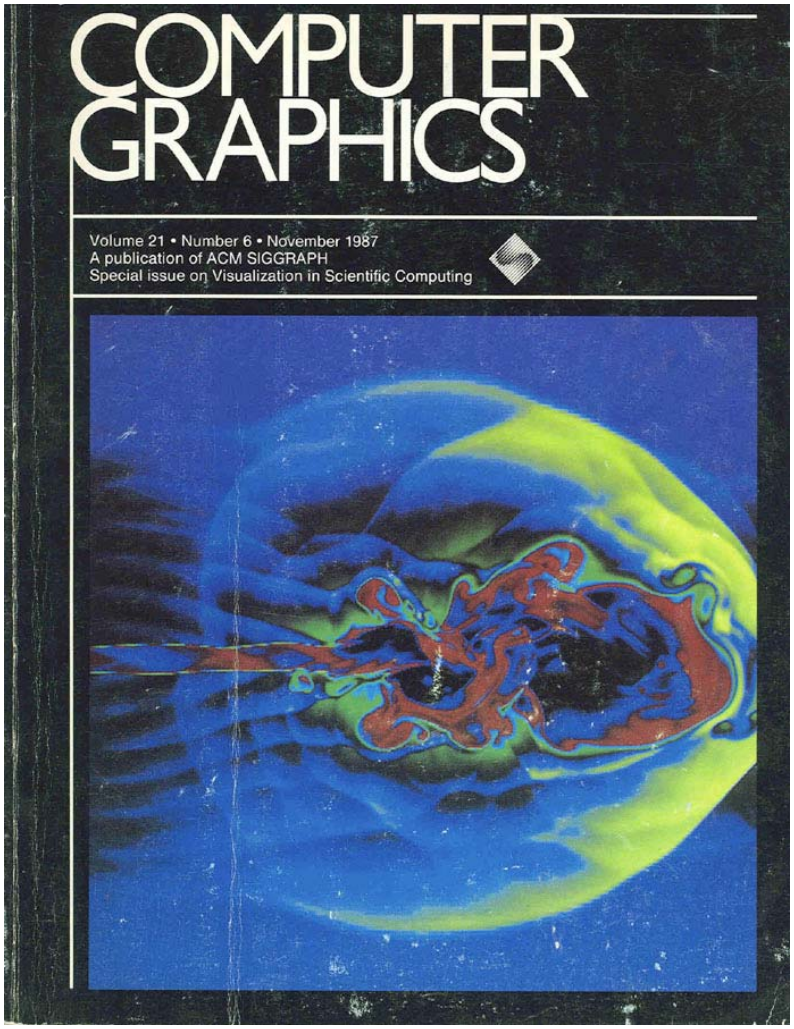


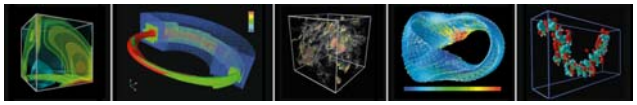
BERKELEY LAB



#3 Scientific Visualization

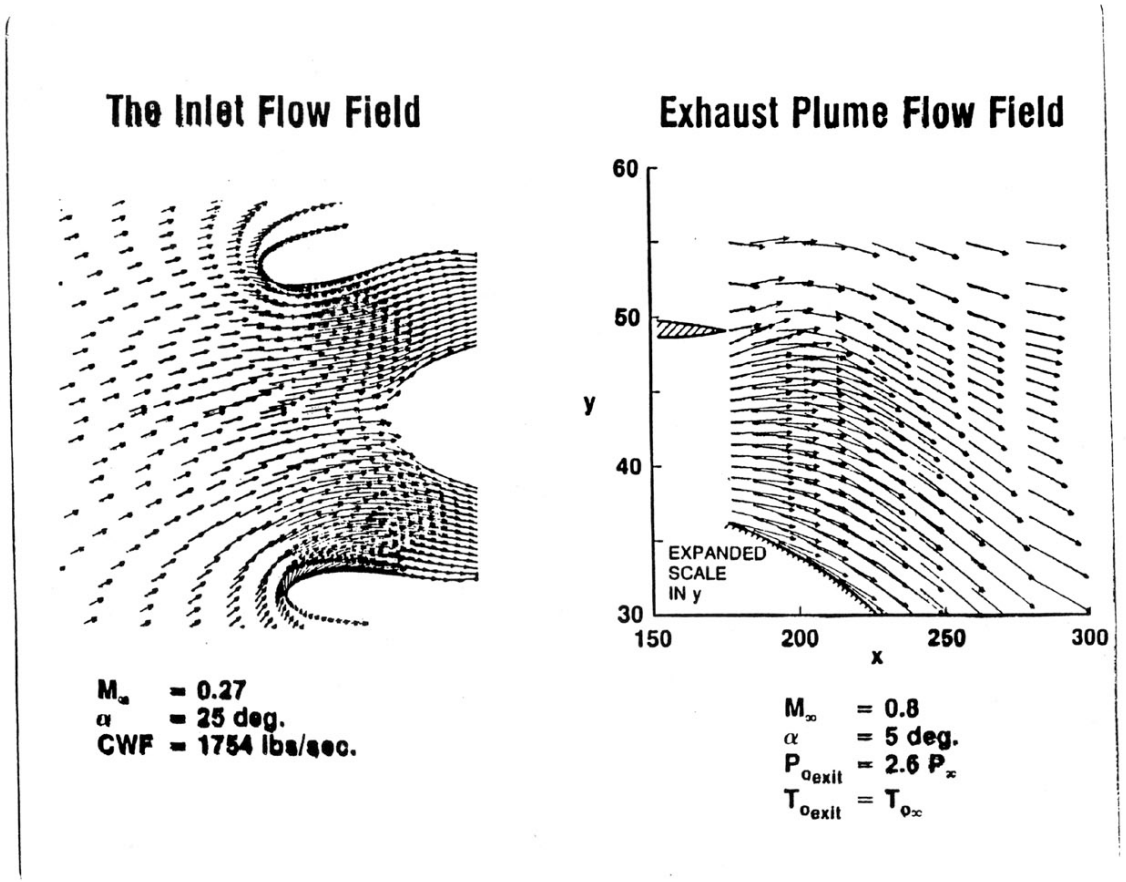
- **NSF Report, "Visualization in Scientific Computing" established the field in 1987 (edited by B.H. McCormick, T.A. DeFanti, and M.D. Brown)**
- **<u>Change in point of view</u>: transformed computer graphics from a technology driven subfield of computer science into a medium for communication**
- **Added artistic element**
- **The role of visualization is "to reveal concepts that are otherwise invisible" (Krystof Lenk)**





Office of Science
U.S. DEPARTMENT OF ENERGY

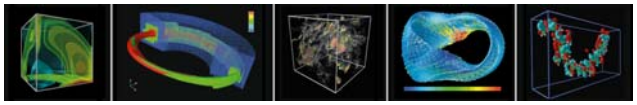
Before Scientific Visualization (1985)



Computer graphics typical of the time:

- **2 dimensional**
- **line drawings**
- **black and white**
- **"vectors" used to display vector field**

Images from a CFD report at Boeing (1985).



After scientific visualization (1992)

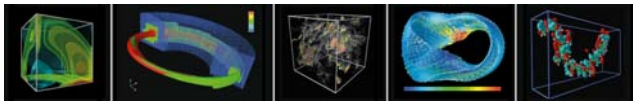
The impact of scientific visualization seven years later:

- **3 dimensional**
- **use of “ribbons” and “tracers” to visualize flow field**
- **color used to characterize updraft and downdraft**

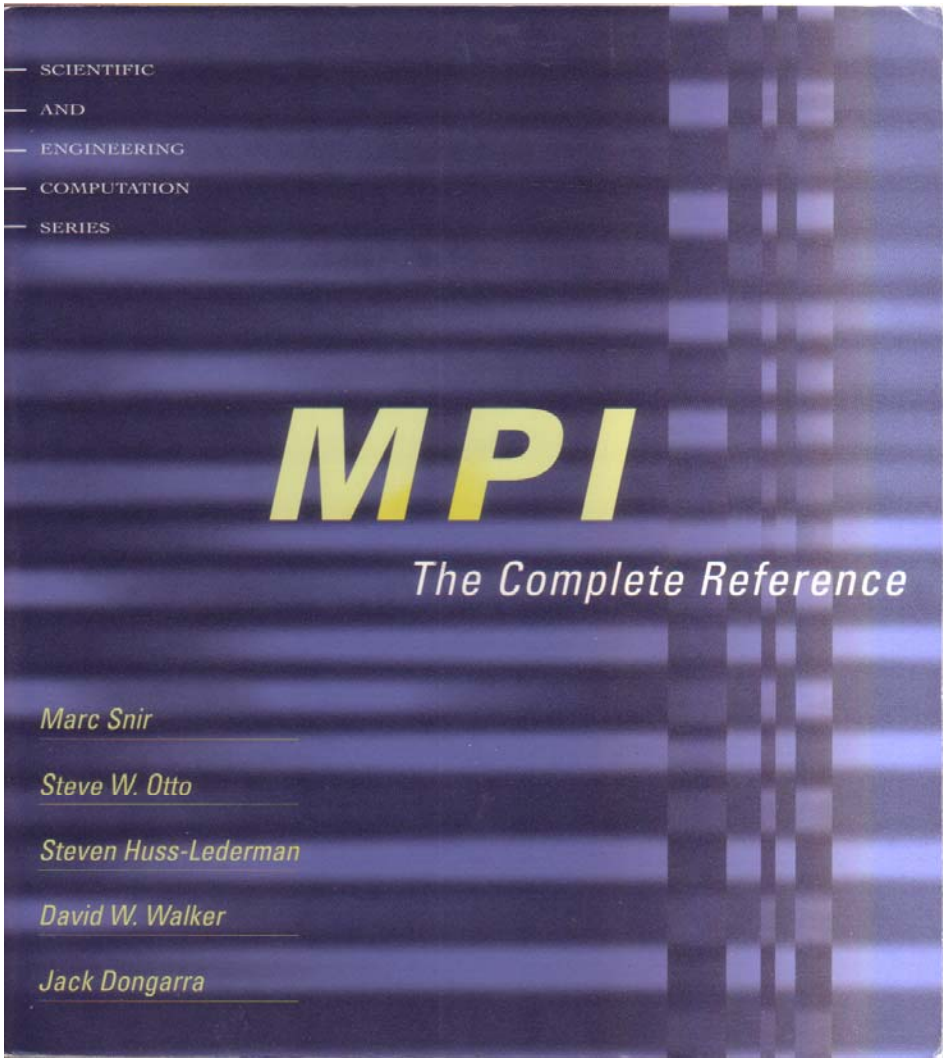
Images from “Supercomputing and the Transformation of Science” by Kauffman and Smarr, 1992; visualization by NCSA; simulation by Bob Wilhelmson, NCSA

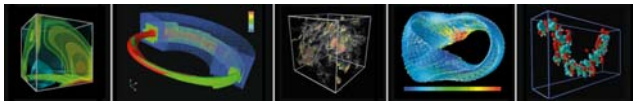
Orange ribbons represent tracers that rise through the depth of the storm in the updraft and blue ribbons represent tracers that eventually fall to the ground in the downdraft in this image tracing particle paths from approximately 75 minutes after the start of the simulation to 2 hours and 18 minutes after.

Tracer ribbons illustrate the development of streamwise vorticity during a 20-minute period. The relative magnitude of the vorticity is largest in the lower part of the storm, where the ribbon is most tightly coiled.



#2 Message Passing Interface (MPI)





Office of Science
U.S. DEPARTMENT OF ENERGY

Parallel Programming 1988

- At the 1988 “Salishan” conference there was a bake-off of parallel programming languages trying to solve five scientific problems
- The “Salishan Problems” (ed. John Feo, published 1992) investigated four programming languages
  - Sisal, Haskel, Unity, LGDF
- Significant research activity at the time
- The early work on parallel languages is all but forgotten today

March 22, 1988
G-6340-HDS-019

To: K. Neves 7L-20

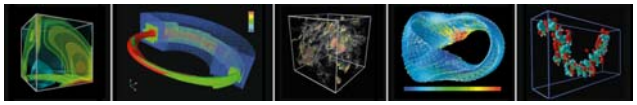
cc: A. Erisman 7L-20 J. Kowalik 7L-24
J. Phillips 7L-21 J. Presti 7L-24
Applied Mathematics Staff

Subject: Salishan Conference on High-Speed Computing, Glenendon Beach, Oregon.

The annual Conference on High-Speed Computing, sponsored by Livermore and Los Alamos, was held as usual at the Salishan resort on the Oregon coast March 21 - 25, 1988. In a first for this series of conferences, it did not rain during the entire conference period. Below follows a detailed trip report, which has been merged from the reports given by myself and David Bailey at NASA Ames.

The conference had been billed as a gathering of computer scientists to discuss issues in high-performance computing. However, this years meeting was dominated by a lengthy discussion of parallel languages, which was not inspiring at all. About half of the conference was spent on this discussion of parallel languages. The setup was very promising. Specialists in five different languages were given four different problems each, and were asked to write a program solving these problems. The plan was to compare the different solutions and to get an appreciation of the potential of each language to express parallelism. The contending groups were Kenneth Dritz of Argonne, who advocated Ada, Paul Hudak of Yale, who preached the merits of Haskell (an in-house "functional" language), Robert Babb of the Oregon Graduate Center, who touted the merits of his LGDF-2 (a large grain data flow language), John Feo of Livermore, who backed the Livermore Sisal system, and Mani Chandra of Texas, who preached for Unity, a graph-based multiprocessing language system.

For several reasons the sessions turned into an often heated discussion of details among the specialists which was not illuminating for the general audience. One of the problems was that each of the proponents of a language was asked to give a short overview over the language. This overview was insufficient to give real understanding of the issues. The situation could be compared to giving somebody who has never heard about Fortran a 20 minute presentation, and then switch the discussion to the relative merits of say multitasking versus microtasking on Cray computers. The Ada advocates suggested that their language was the way to go, simply because it has been standardized. The "functional" language advocates brushed aside those detractors who pointed out that the parallelism inherent in the original problem had been completely obscured in the code, and asked the audience to accept on faith the obvious statement that functional languages are more "elegant". They also asked the audience to accept the fact that concurrent programs written in functional languages are fundamentally easier to compile than those in other languages, without any baacking of this claim. The Sisal advocates asked the audience to admire the elegance of their solution to one of the problems, ignoring the fact that other solutions, which used another algorithm, ran many times faster.



Office of Science
U.S. DEPARTMENT OF ENERGY

Parallel Programming 1990

Parallel Processing

0. Automatic parallelizing compiler - low level language

1. Compiler directives

2. Add-on to existing sequential languages (monitors, send/receive, SCHEDULE, macros, subroutines)

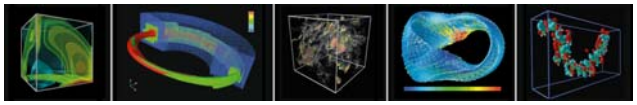
3. Standard parallel features in existing languages - PFC

4. Special parallel languages - SISAL, Linda, Strand

5. High-level languages - Lisp (pure), Prolog (pure)

6. New Programming models - spread sheet, neural networks

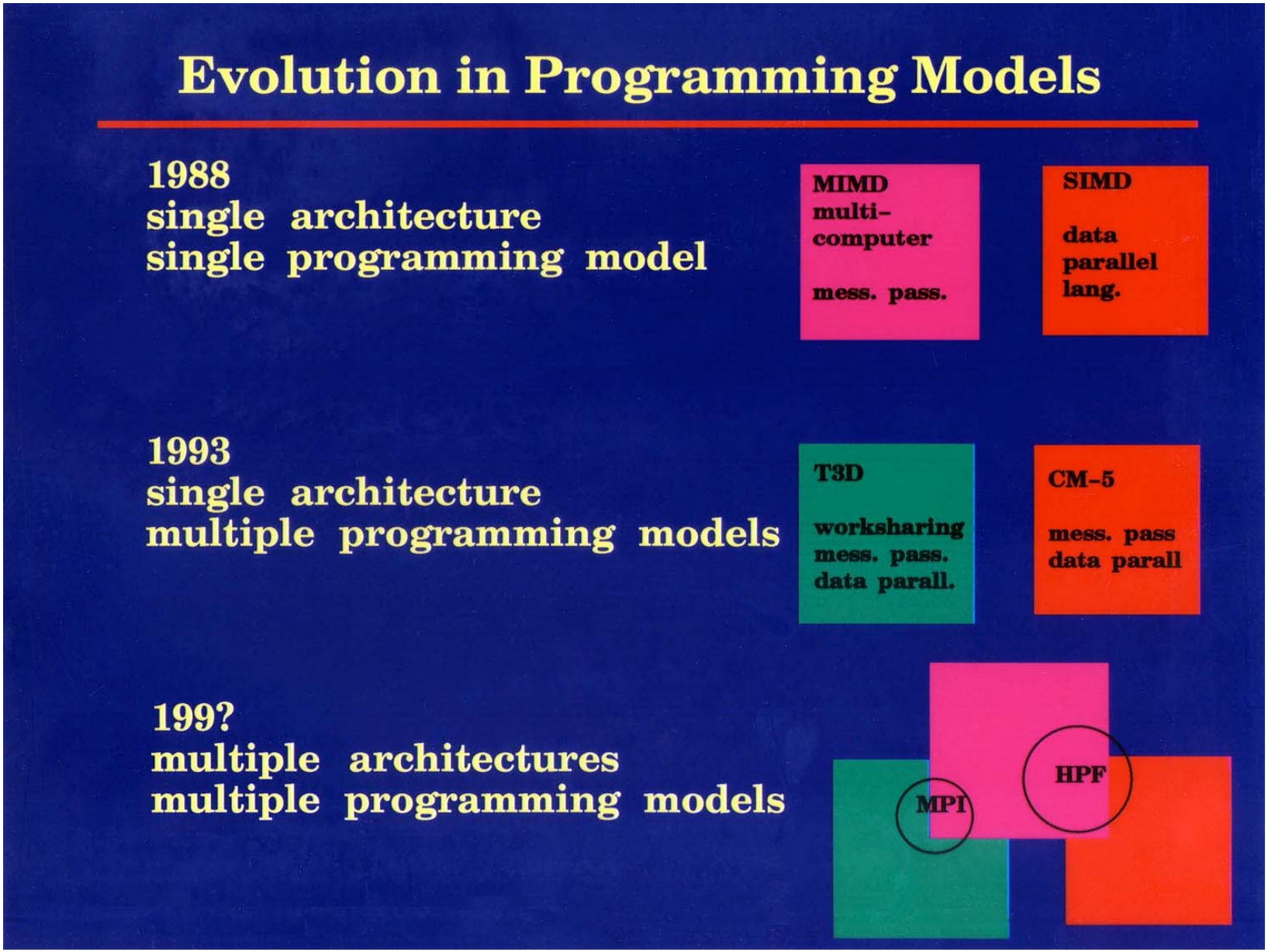
- The availability of real parallel machines moved the discussion from the domain of theoretical CS to the pragmatic application area
- In this presentation (ca. 1990) Jack Dongarra lists seven approaches to parallel processing
- Note that message passing libraries are a sub-item on 2)



From a presentation by

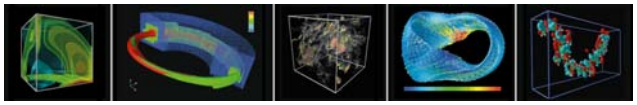
HDS in Mannheim , 1994

Parallel Programming 1994



Parallel Programming 1996

- At NERSC in 1996 among scientific users
  - 30% used PVM
  - 30% used MPI
  - 30% used HPF
  - 10% used SHMEM or CRAFT
- It was only in about 1998 that it was clear that MPI was the choice model for parallel programming



Until 2010: A New Parallel Programming Methodology? - NOT

The software challenge: overcoming the MPI barrier

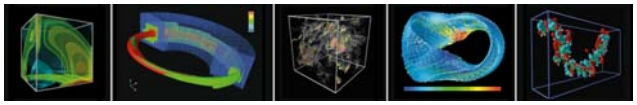
- **MPI created finally a standard for applications development in the HPC community**
- **Standards are always a barrier to further development**
- **The MPI standard is a least common denominator building on mid-80s technology**

Programming Model reflects hardware!

From a presentation by

HDS in Heidelberg, 2001

"I am not sure how I will program a Petaflops computer, but I am sure that I will need MPI somewhere" – HDS 2001



Office of Science
U.S. DEPARTMENT OF ENERGY

#1 Scaled Speed-Up

SIAM J. Sci. Stat. Comput.
Vol. 9, No. 4, July 1988

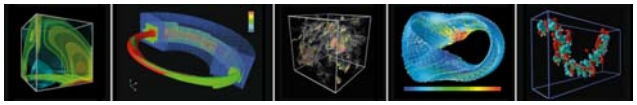
001

DEVELOPMENT OF PARALLEL METHODS FOR A 1024-PROCESSOR HYPERCUBE*

JOHN L. GUSTAFSON†, GARY R. MONTRY†, AND ROBERT E. BENNER†

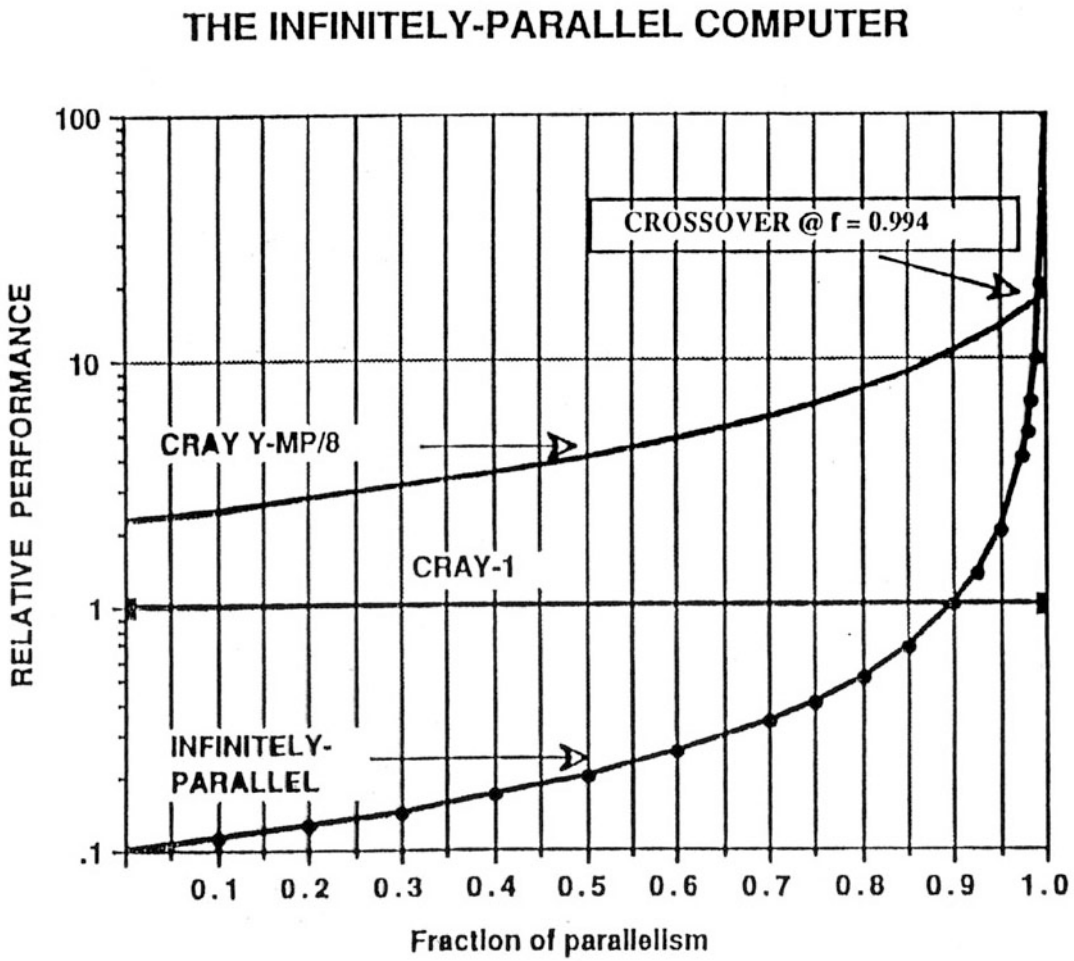
Abstract. We have developed highly efficient parallel solutions for three practical, full-scale scientific problems: wave mechanics, fluid dynamics, and structural analysis. Several algorithmic techniques are used to keep communication and serial overhead small as both problem size and number of processors are varied. A new parameter, *operation efficiency*, is introduced that quantifies the tradeoff between communication and redundant computation. A 1024-processor MIMD ensemble is measured to be 502 to 637 times as fast as a single processor when problem size for the ensemble is fixed, and 1009 to 1020 times as fast as a single processor when problem size *per processor* is fixed. The latter measure, denoted *scaled speedup*, is developed and contrasted with the traditional measure of parallel speedup. The scaled-problem paradigm better reveals the capabilities of large ensembles, and permits detection of subtle hardware-induced load imbalances (such as error correction and data-dependent MFLOPS rates) that may become increasingly important as parallel processors increase in node count. Sustained performance for the applications is 70 to 130 MFLOPS, validating the massively parallel ensemble approach as a practical alternative to more conventional processing methods. The techniques presented appear extensible to even higher levels of parallelism than the 1024-processor level explored here.

Key words. fluid dynamics, hypercubes, MIMD machines, multiprocessor performance, parallel computing, structural analysis, supercomputing, wave mechanics

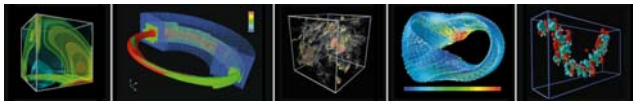


Office of Science
U.S. DEPARTMENT OF ENERGY

The argument against massive parallelism (ca. 1988)



From a presentation by Jack Worlton, LANL, at NCSA 1989



The argument against massive parallelism (ca. 1988)

Massively–Parallel Approaches (continued)

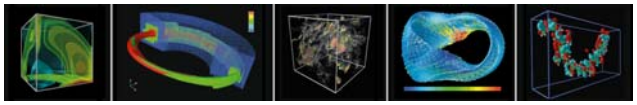
For continued performance improvement —

Case 1	$\sigma(p) = c \log_2 p$	$p \leq 6931$	$S_p \leq 704$
Case 2	$\sigma(p) = \frac{c}{2} p$	$p \leq 141$	$S_p \leq 70$
Case 3	$\sigma(p) = \frac{c}{2} p \log_2 p$	$p \leq 53$	$S_p \leq 29$

where the normalization coefficient $c = 0.0001$.

Theoretical papers such as this "proved" that the speed-up on real parallel systems is limited

From a presentation by Jim Hack at DOE Salishan High Speed Computing, 1988

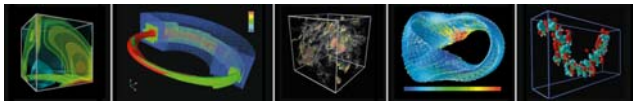


Scaled Speed Up

Helped the community to overcome a conceptual barrier

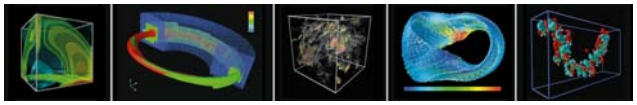
- **Exposed the fallacy of the fixed size speed-up**
- **Focused back on the fact that we want to use larger computers in order to solve larger problems**
- **Opened the way for parallel applications development**

Without this change in our understanding of parallel computing the early successes in adapting parallel computers would not have happened



Three Challenges for 2005 - 2025

- These are my personal opinion
- The challenges are in chronological order
  - 2005 - 2012
  - 2010 - 2018
  - 2015 - 2025

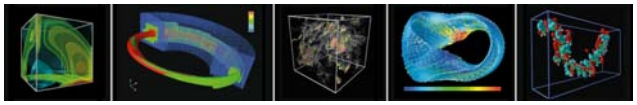


Challenge 2005 - 2010: Scaling Applications to Petascale Systems

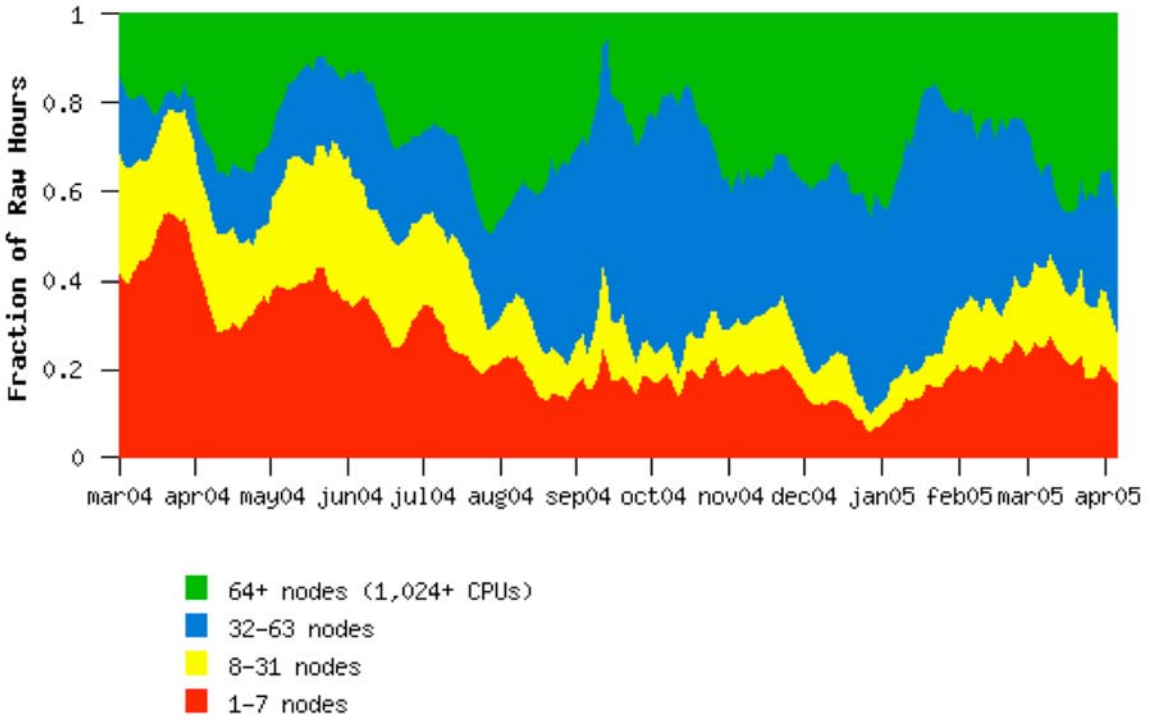
(Assume nominal Petaflop/s system with 100,000 commodity processors of 10 Gflop/s each)

Three major issues:

- **Scaling to 100,000 processors and multi-core processors**
- **Topology sensitive interconnection network**
- **Memory Wall**

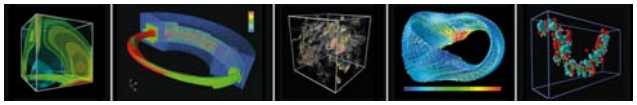


Application Status in 2005



Parallel job size at NERSC

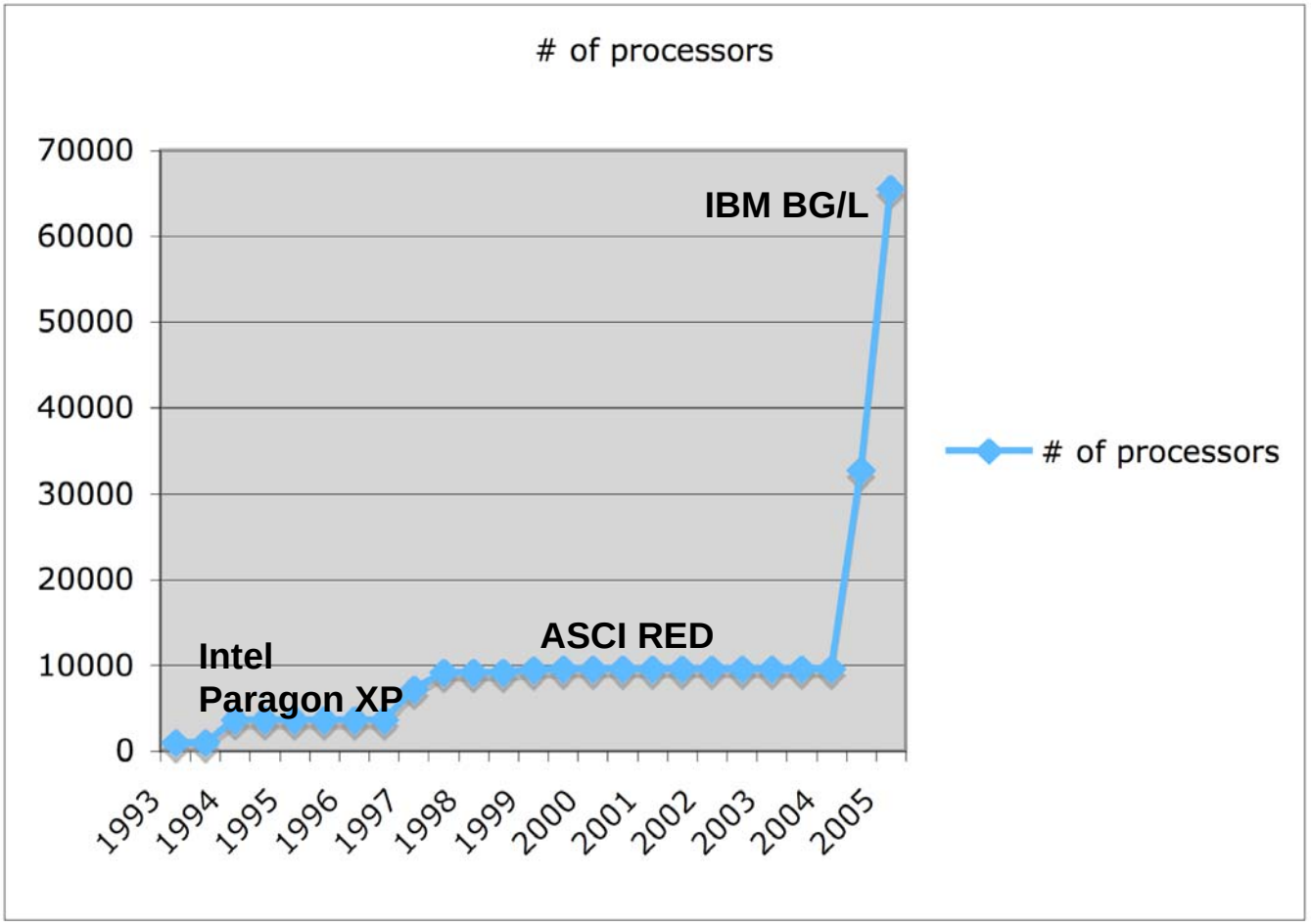
- **A few Teraflop/s sustained performance**
- **Scaled to 512 - 1024 processors**

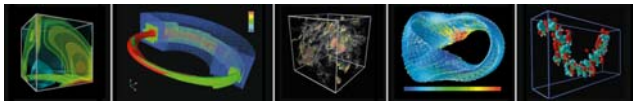


Office of Science
U.S. DEPARTMENT OF ENERGY

Parallelism has Stagnated for a Decade

Number of processors in the most highly parallel system in the TOP500

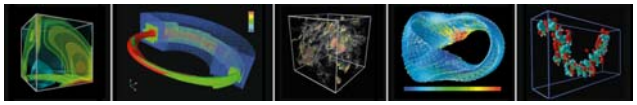




Office of Science
U.S. DEPARTMENT OF ENERGY

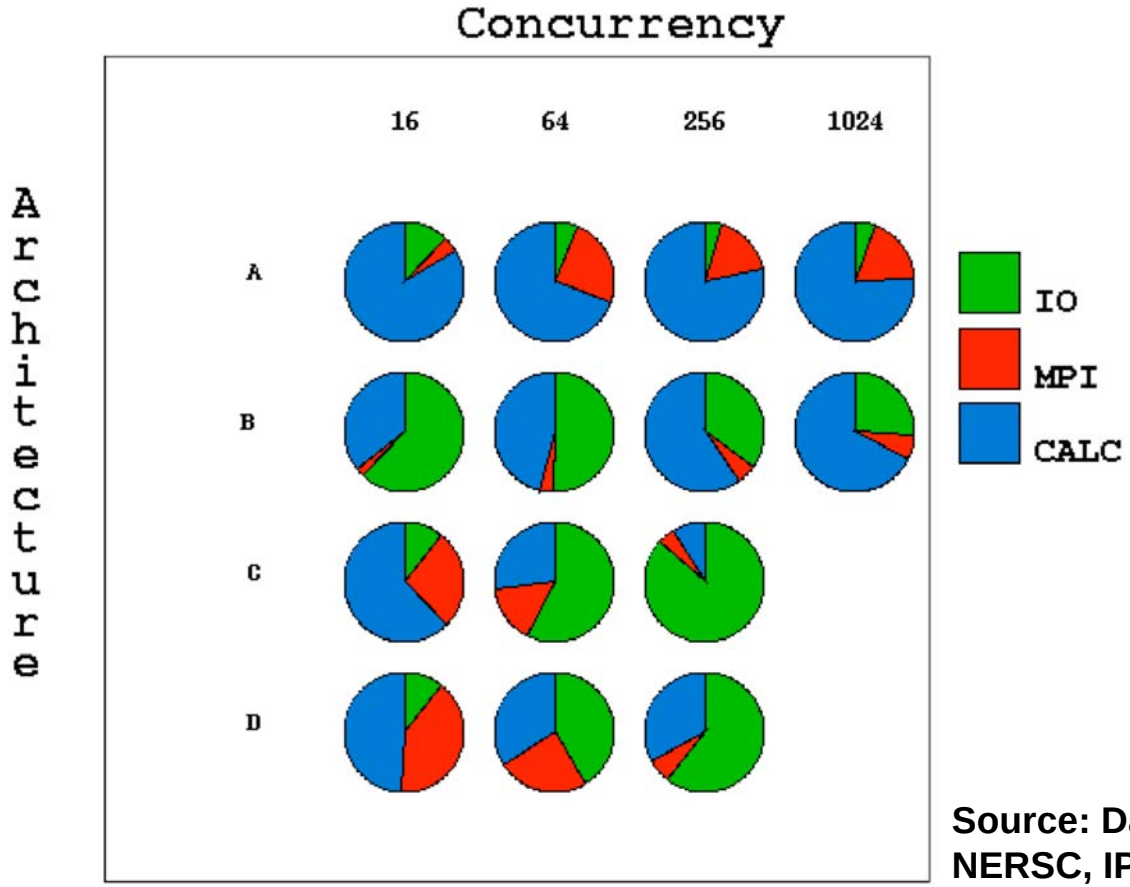
Integrated Performance Monitoring (IPM)

- **brings together multiple sources of performance metrics into a single profile that characterizes the overall performance and resource usage of the application**
- **maintains low overhead by using a unique hashing approach which allows a fixed memory footprint and minimal CPU usage**
- **open source, relies on portable software technologies and is scalable to thousands of tasks**
- **developed by David Skinner at NERSC (see http://www.nersc.gov/projects/ipm/)**

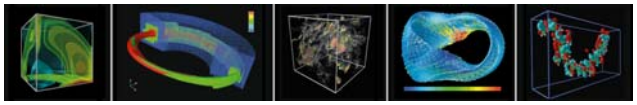


Scaling Portability: Profoundly Interesting

A high level description of the performance of cosmology code MADCAP on four well known architectures.

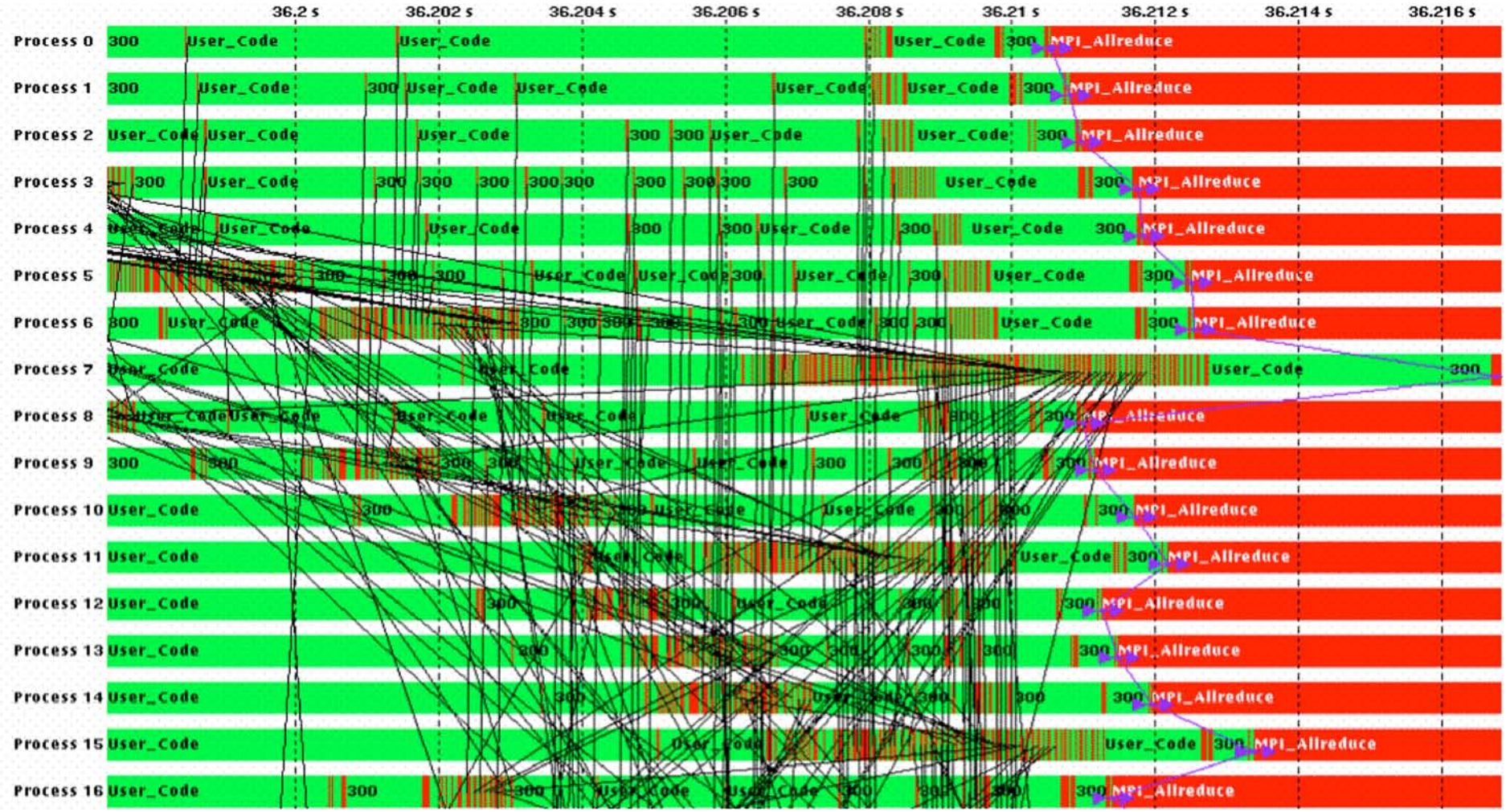


Source: David Skinner, NERSC, IPM project
http://www.nersc.gov/projects/ipm/

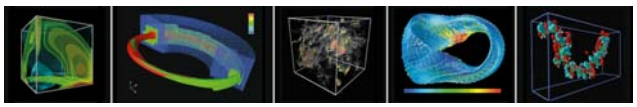


Office of Science
U.S. DEPARTMENT OF ENERGY

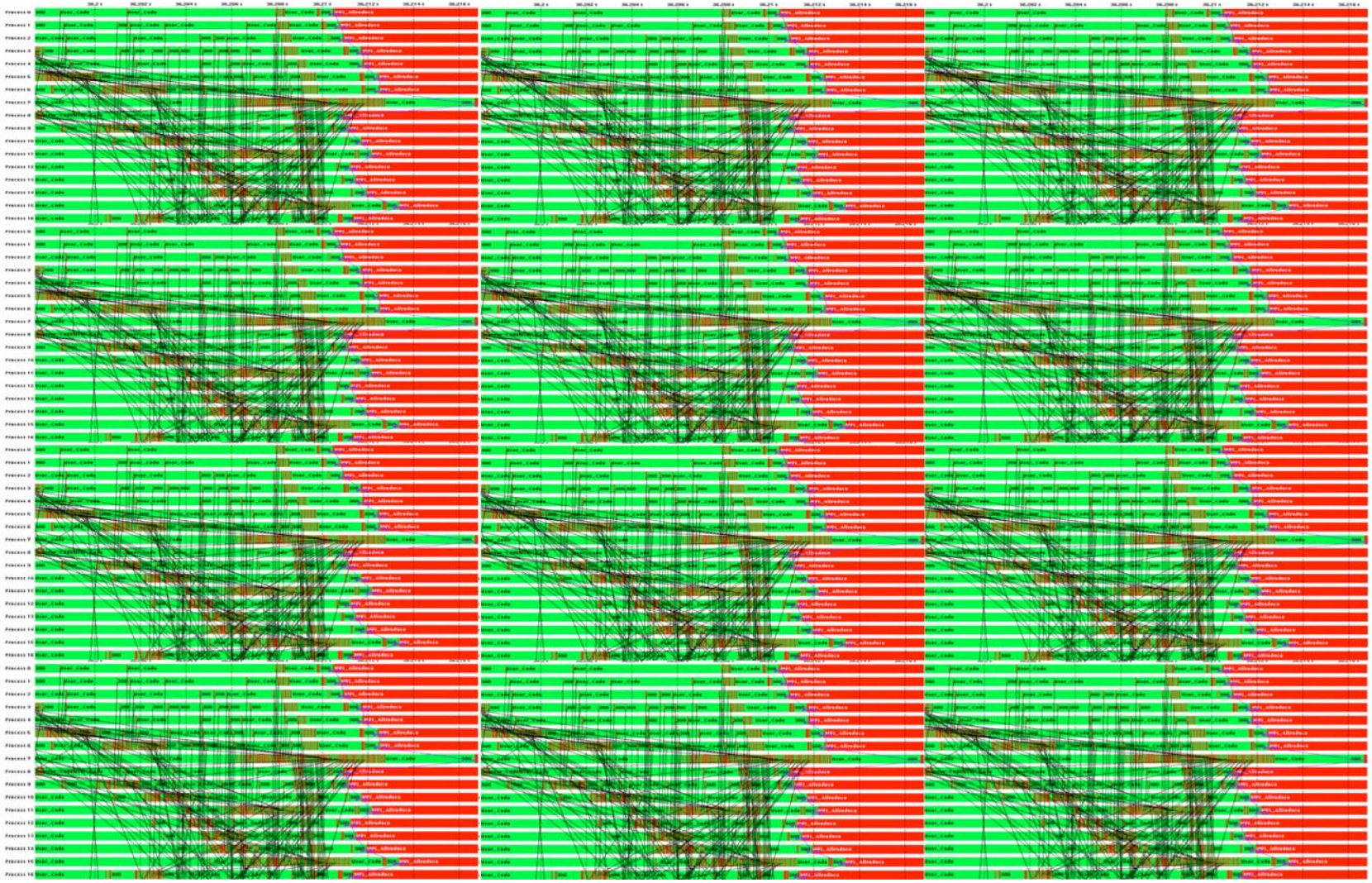
16 Way for 4 seconds

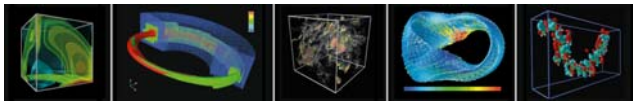


(About 20 timestamps per second per task) *(1…4 contextual variables)



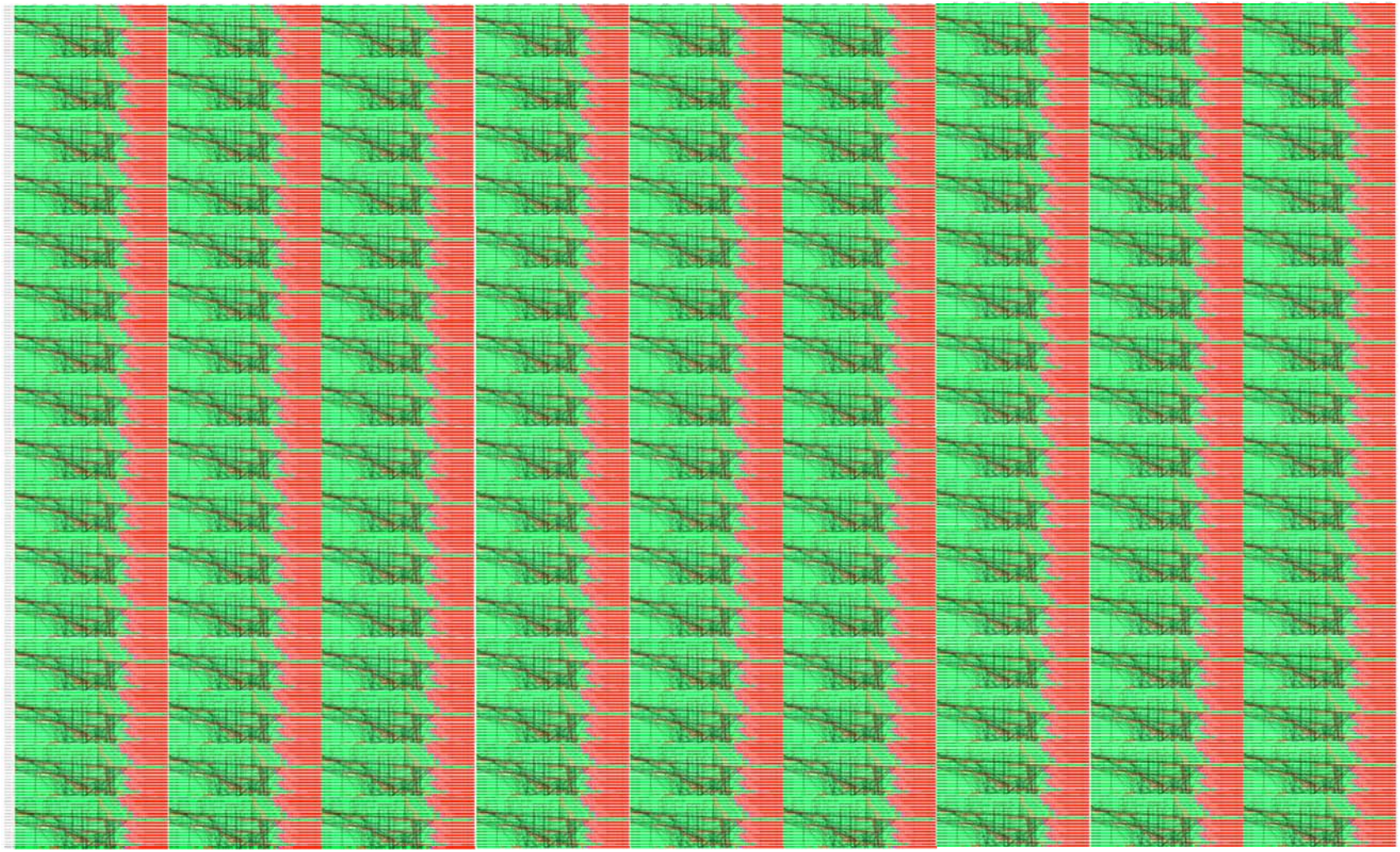
64 way for 12 seconds

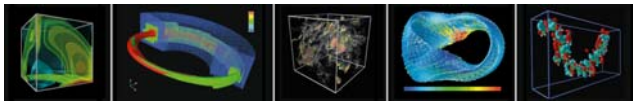




Office of Science
U.S. DEPARTMENT OF ENERGY

256 Way for 36 Seconds





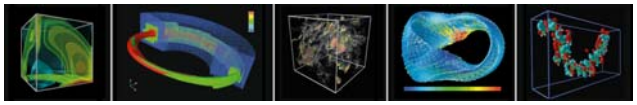
Office of Science
U.S. DEPARTMENT OF ENERGY

Applications on Petascale Systems will need to deal with

(Assume nominal Petaflop/s system with 100,000 commodity processors of 10 Gflop/s each)

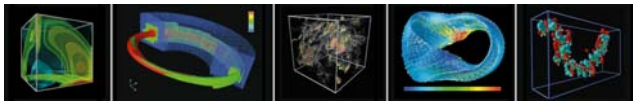
Three major issues:

- **Scaling to 100,000 processors and multi-core processors**
- **Topology sensitive interconnection network**
- **Memory Wall**

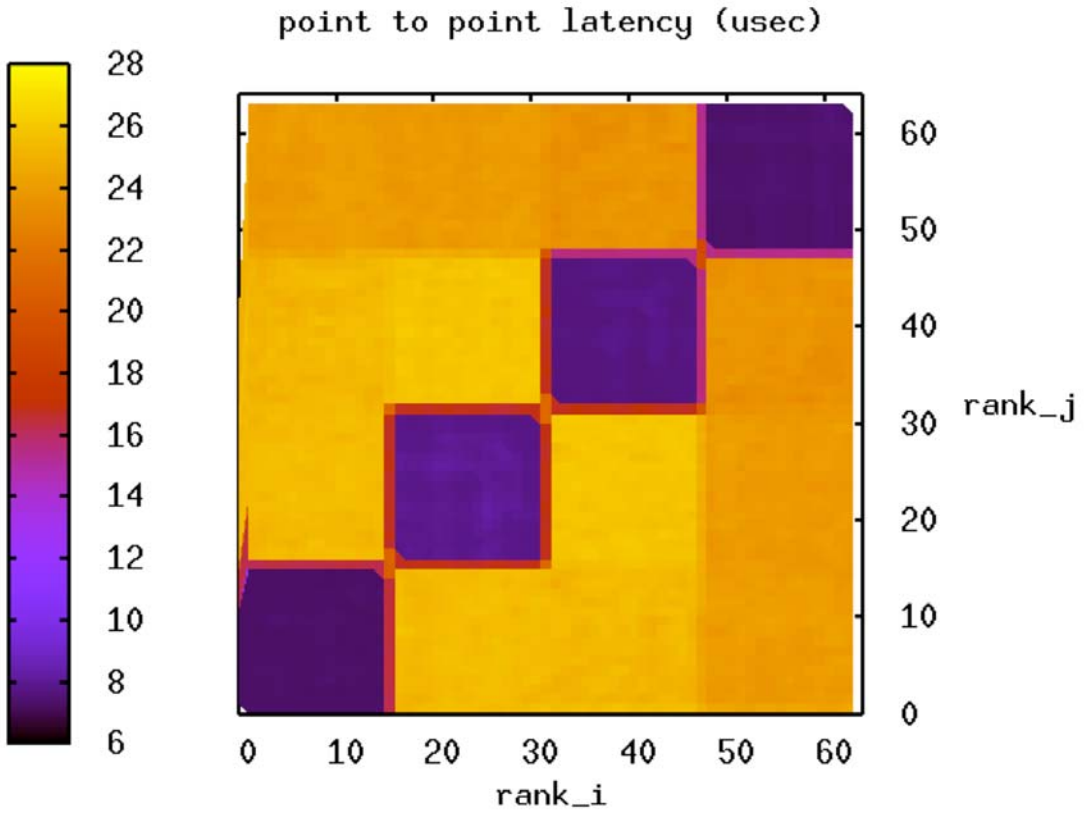


Future Interconnects

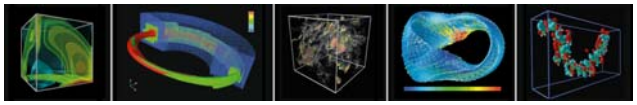
- Currently most applications work with a flat MPI model, this is already a simplification
- More processors means more complex interconnects and topology sensitivity
- Example: BG/L
  - five different interconnection networks
  - latency dependent on distance



Even today's machines are interconnect topology sensitive

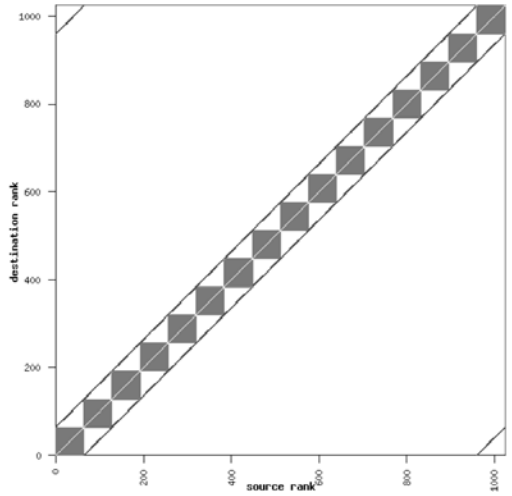


Four (16 processor) IBM Power 3 nodes with Colony switch

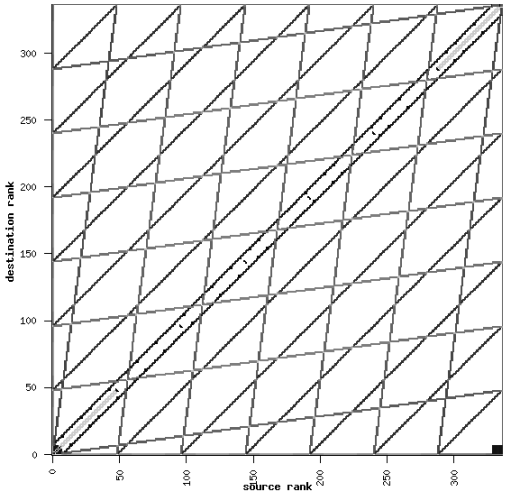


Application Topology

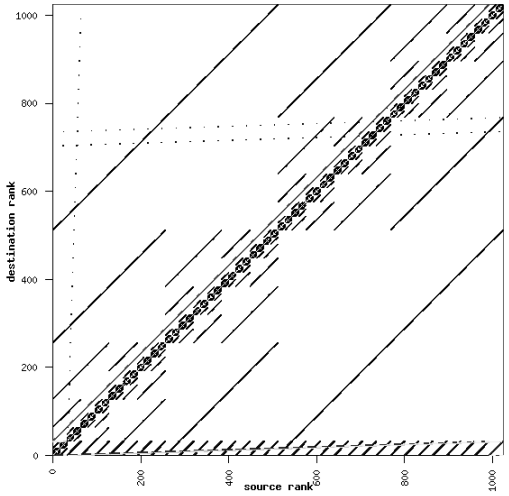
1024 way MILC



336 way FVCAM

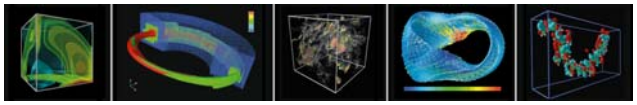


1024 way MADCAP



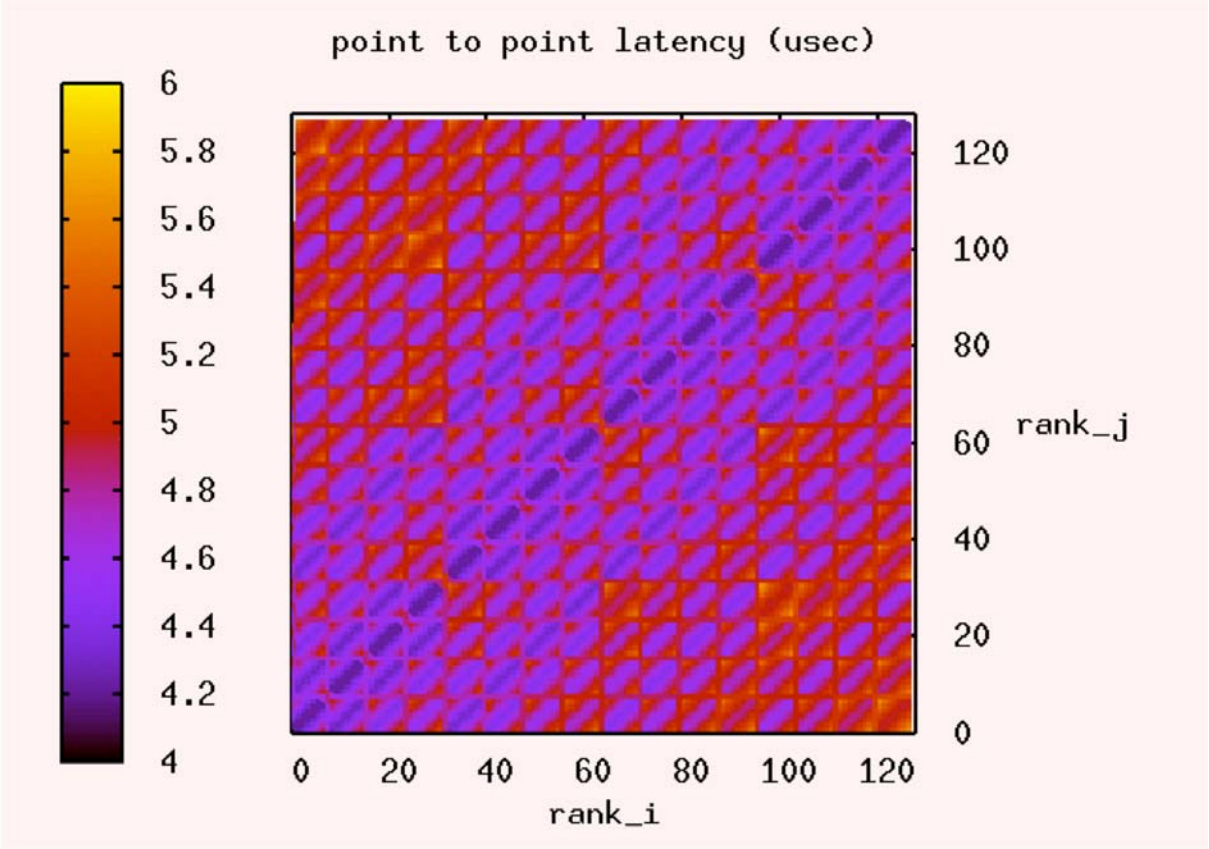
"Characterizing Ultra-Scale Applications Communincations Requirements", by John Shalf et al., submitted to SC05

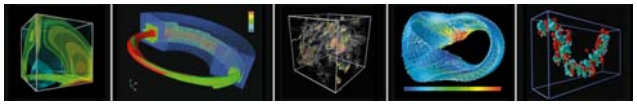
If the interconnect is topology sensitive, mapping will become an issue (again)



Office of Science
U.S. DEPARTMENT OF ENERGY

Interconnect Topology BG/L



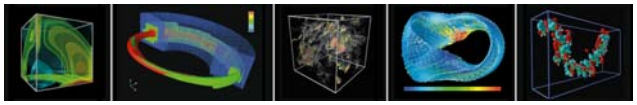


Applications on Petascale Systems will need to deal with

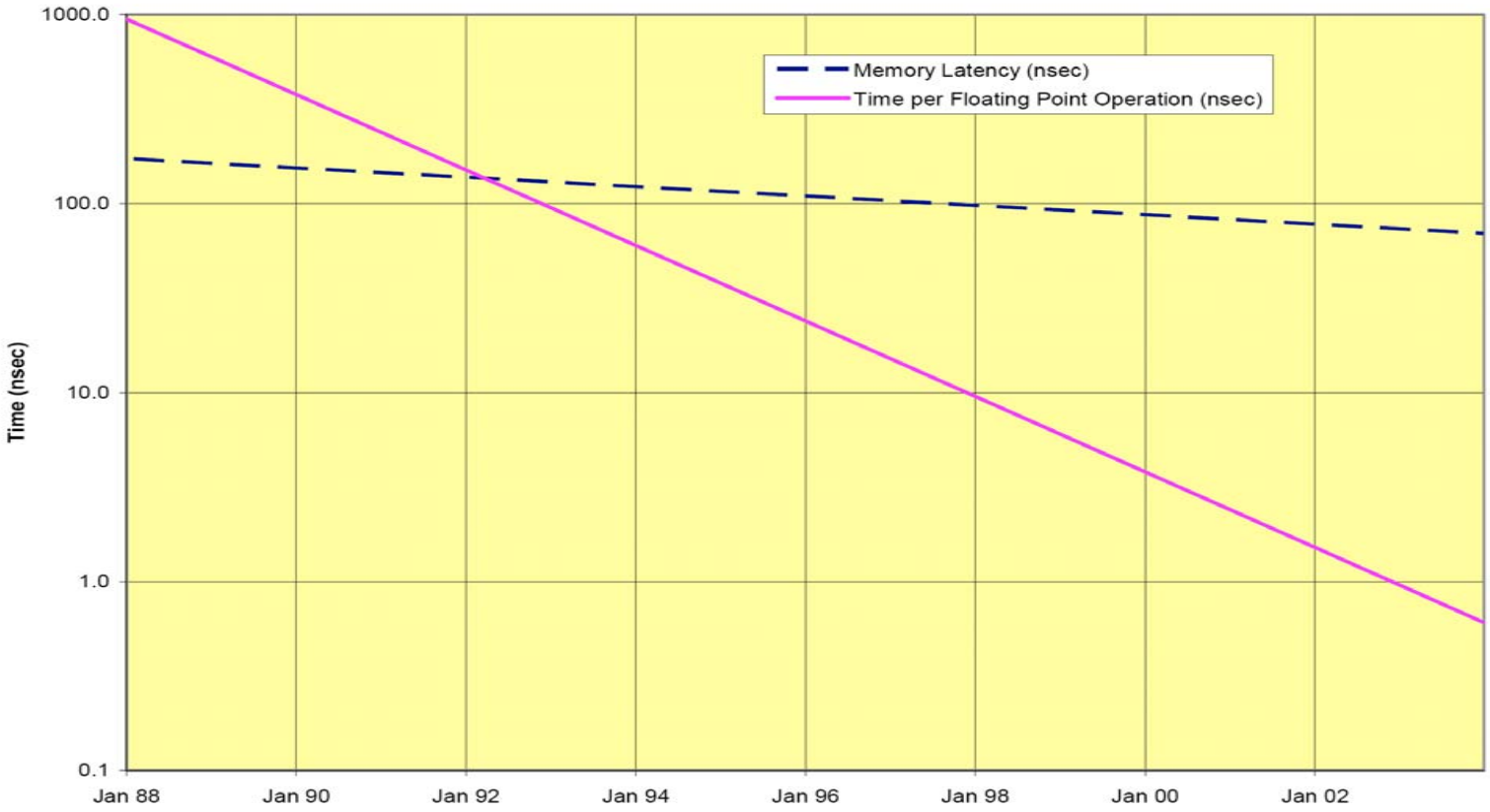
(Assume nominal Petaflop/s system with 100,000 commodity processors of 10 Gflop/s each)

Three major issues:

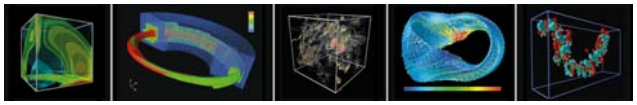
- **Scaling to 100,000 processors and multi-core processors**
- **Topology sensitive interconnection network**
- **Memory Wall**



The Memory Wall



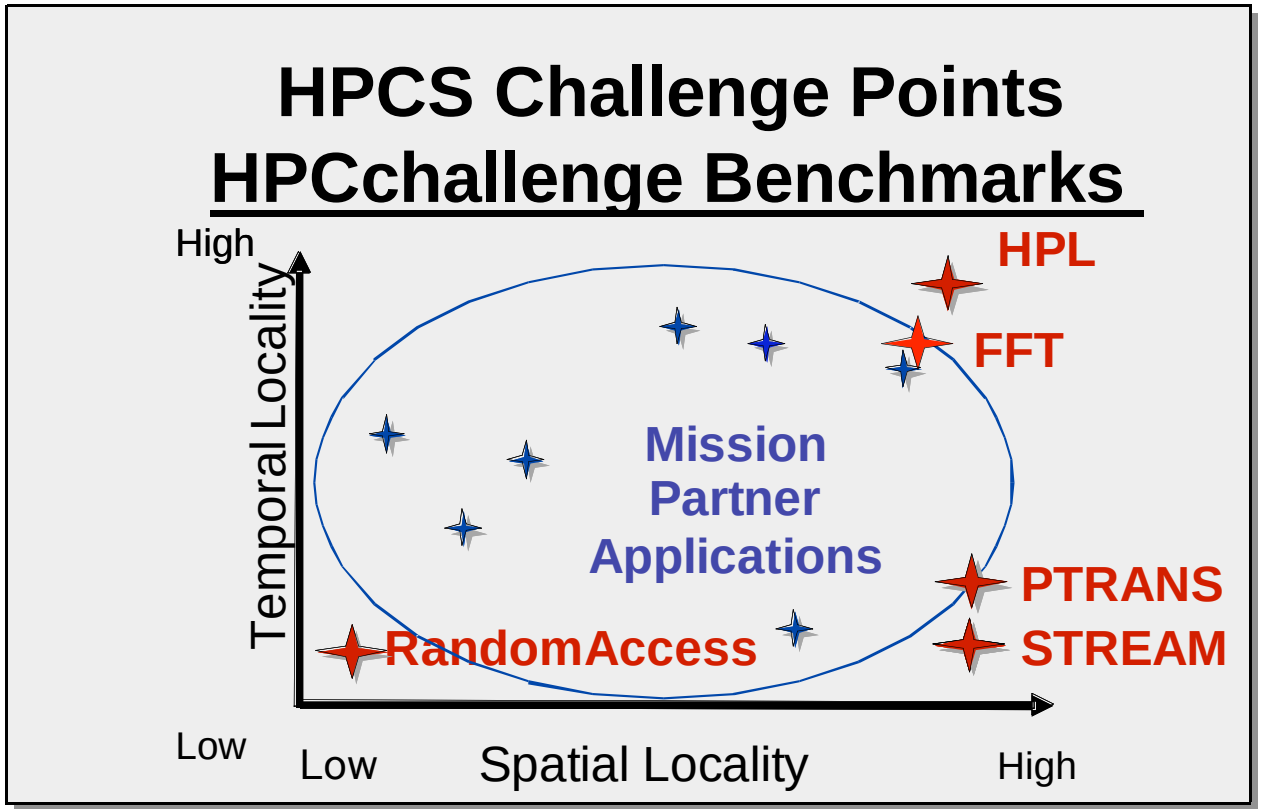
Source: "Getting up to speed: The Future of Supercomputing", NRC, 2004



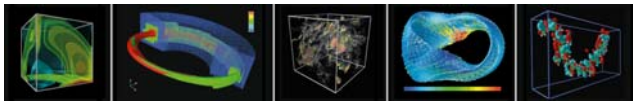
Office of Science
U.S. DEPARTMENT OF ENERGY

Characterizing Memory Access

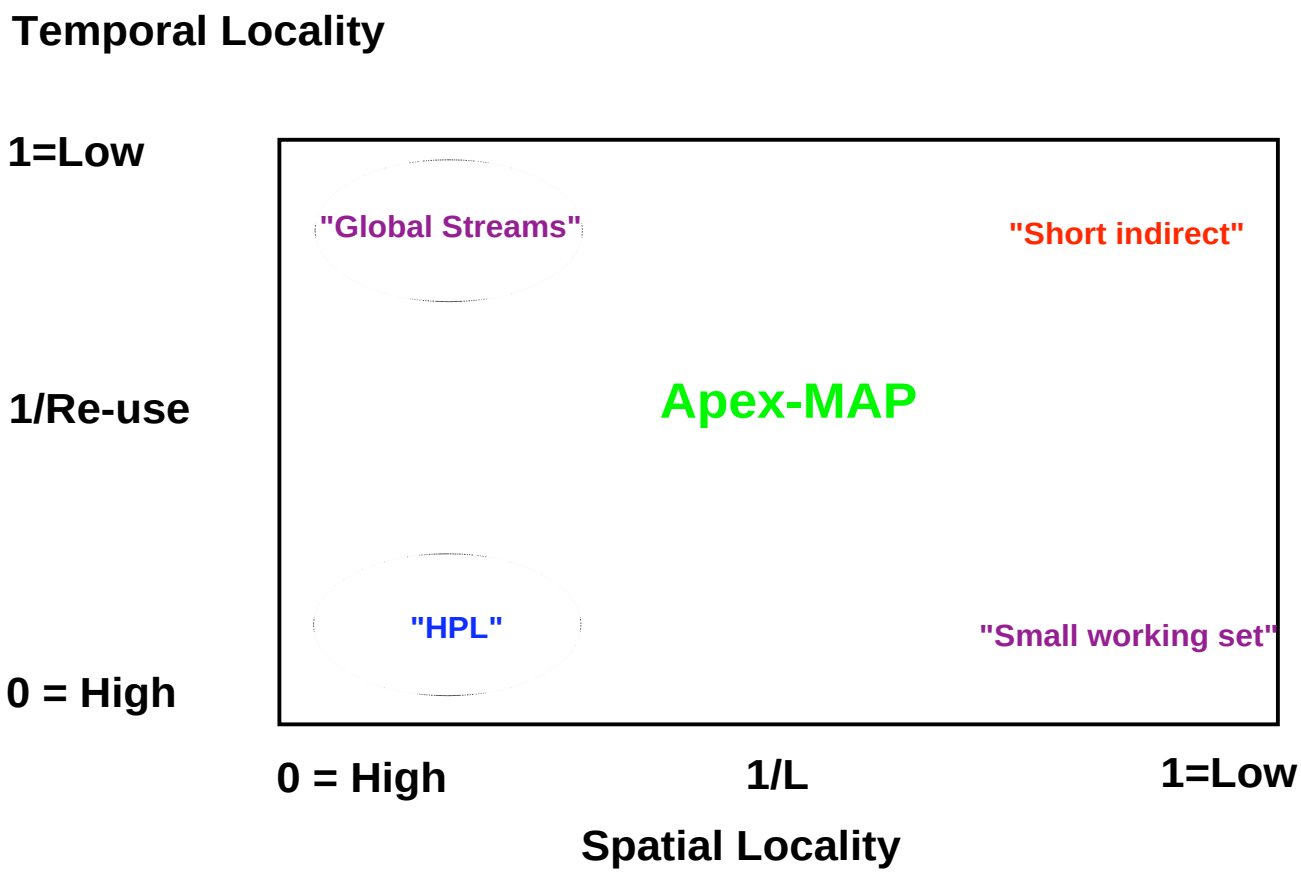
Memory Access Patterns/Locality



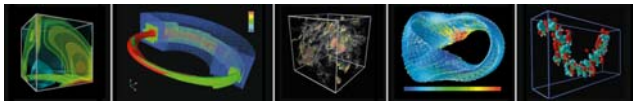
Source: David Koester, MITRE



Apex-MAP characterizes architectures through a synthetic benchmark

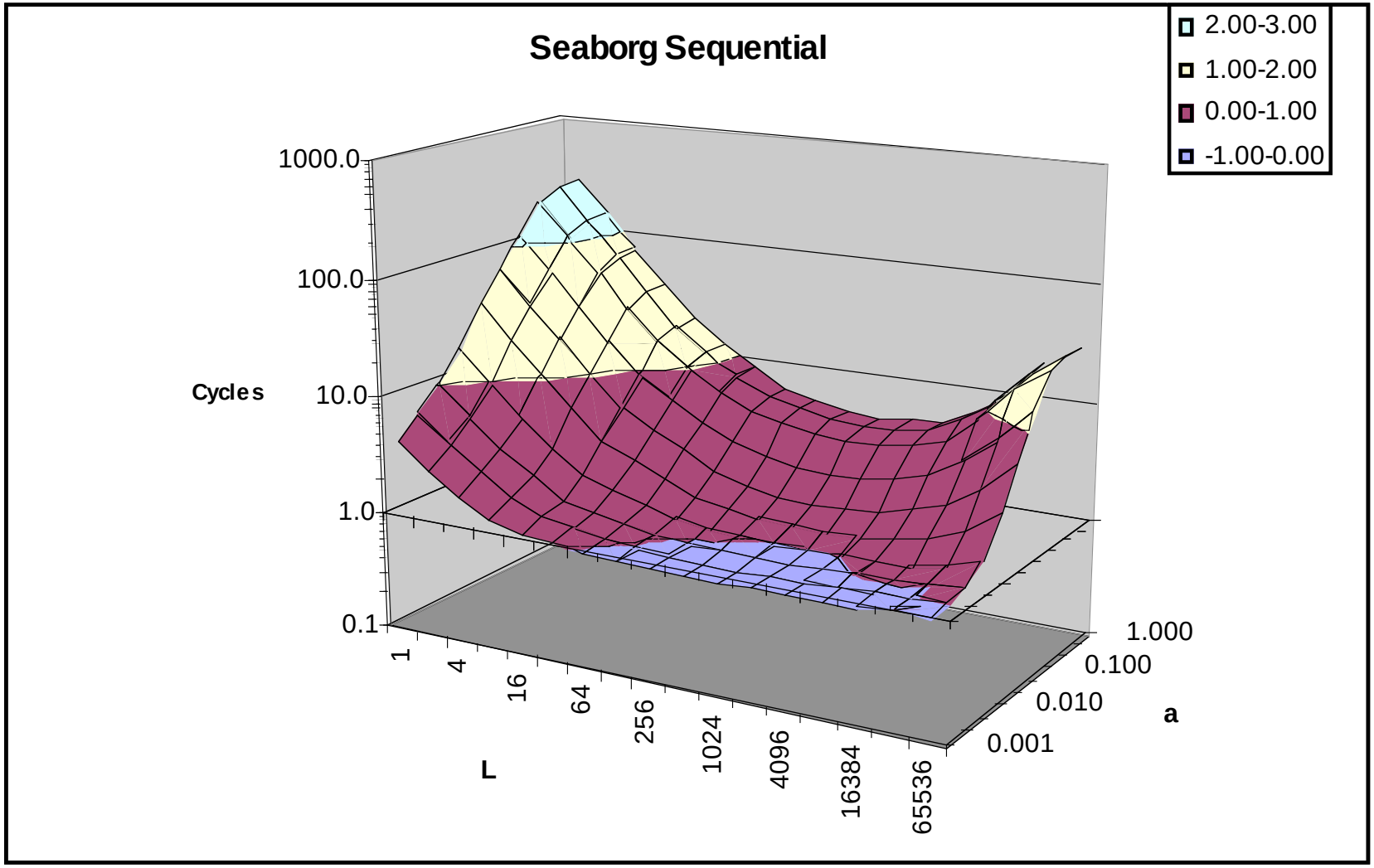


Source: Erich Strohmaier, NERSC, LBNL

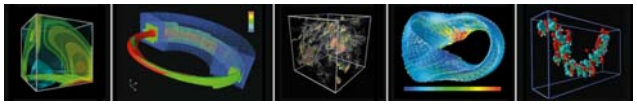


Office of Science
U.S. DEPARTMENT OF ENERGY

Apex-Map Sequential

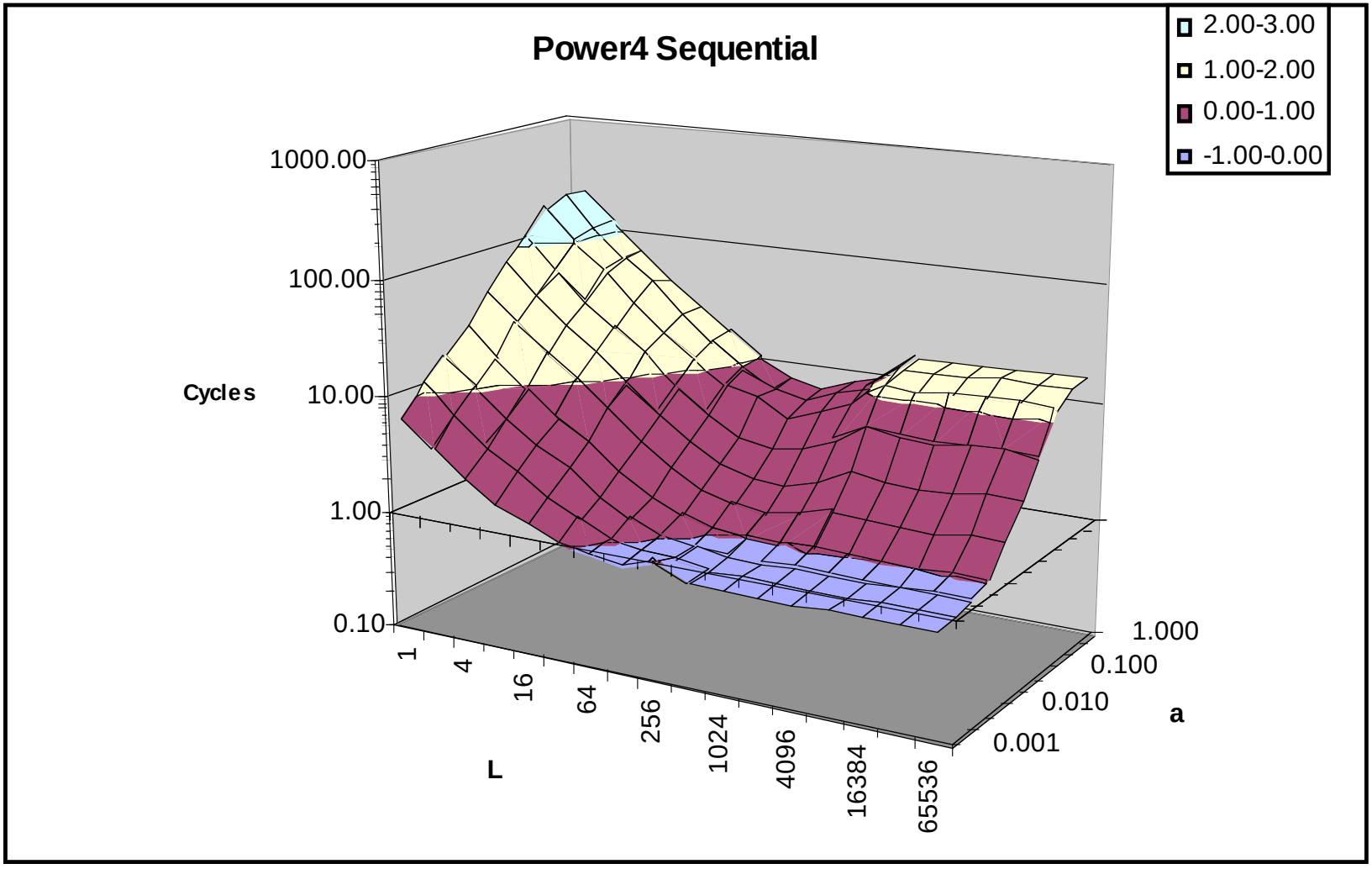


Source: Erich Strohmaier, NERSC, LBNL

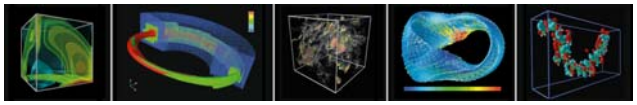


Office of Science
U.S. DEPARTMENT OF ENERGY

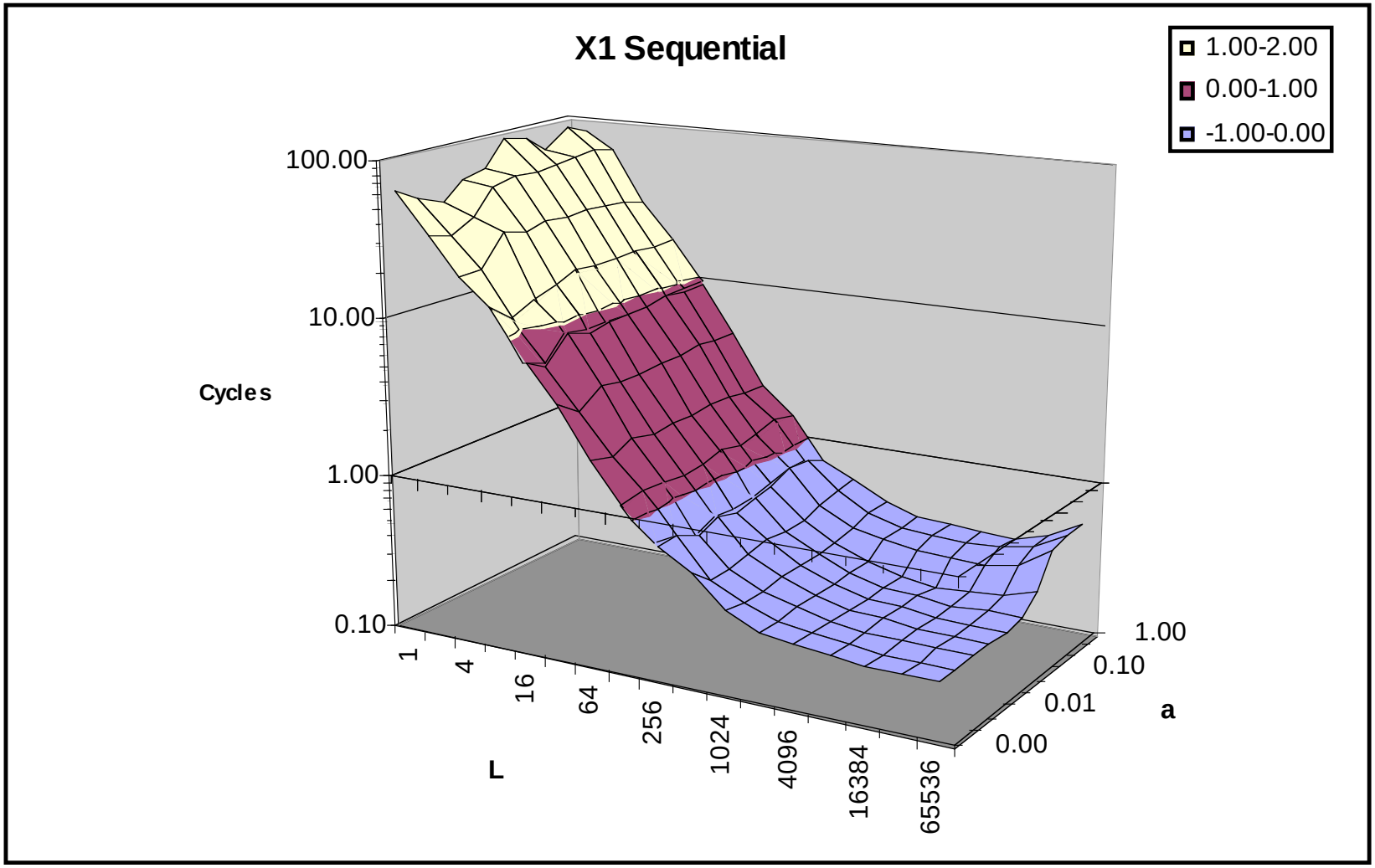
Apex-Map Sequential



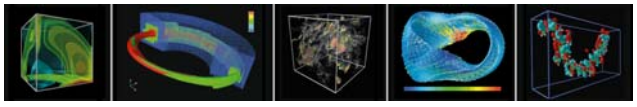
Source: Erich Strohmaier, NERSC, LBNL



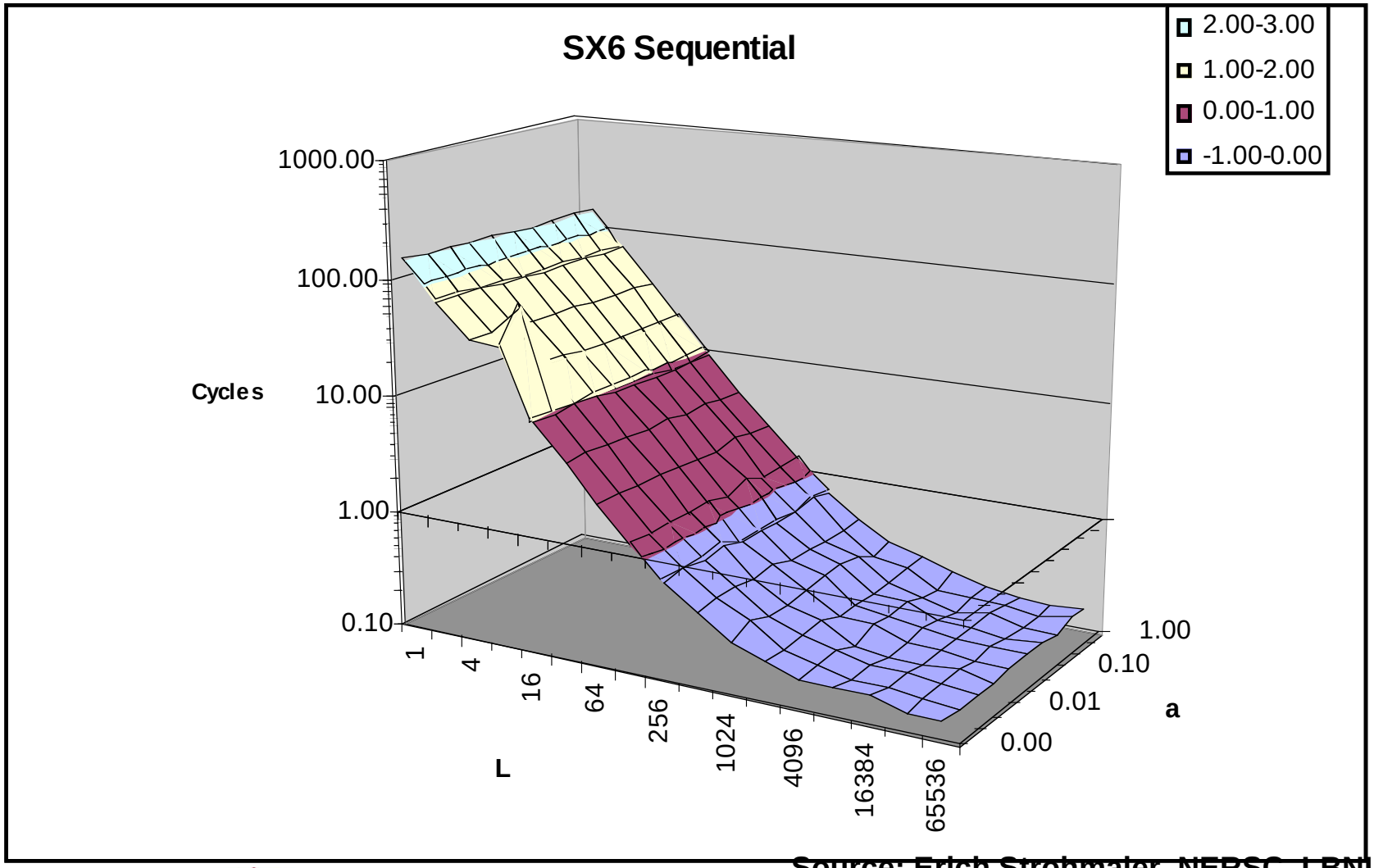
Apex-Map Sequential



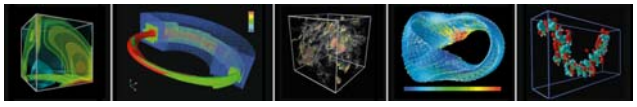
Source: Erich Strohmaier, NERSC, LBNL



Apex-Map Sequential



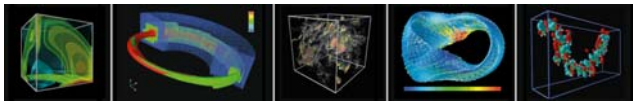
Source: Erich Strohmaier, NERSC, LBNL



Challenge 2010 - scaling to Petaflops level

- Applications will face (at least) three challenges
  - Scaling to 100,000s of processors
  - Interconnect topology
  - Memory access

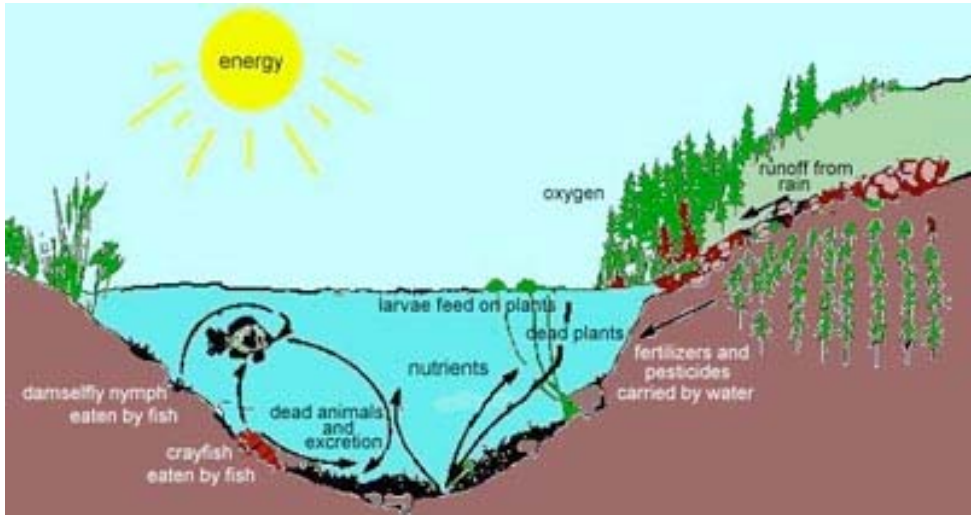
- We have yet to scale to the 100,000 processor level
  - Algorithms
  - Tools
  - System Software



Challenge 2010 - 2018: Developing a New Ecosystem for HPC

From the NRC Report on "The Future of Supercomputing":

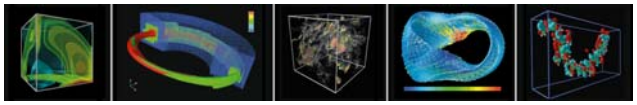
- **Platforms, software, institutions, applications, and people who solve supercomputing applications can be thought of collectively as an ecosystem**
- **Research investment in HPC should be informed by the ecosystem point of view - progress must come on a broad front of interrelated technologies, rather than in the form of individual breakthroughs.**



Pond ecosystem image from

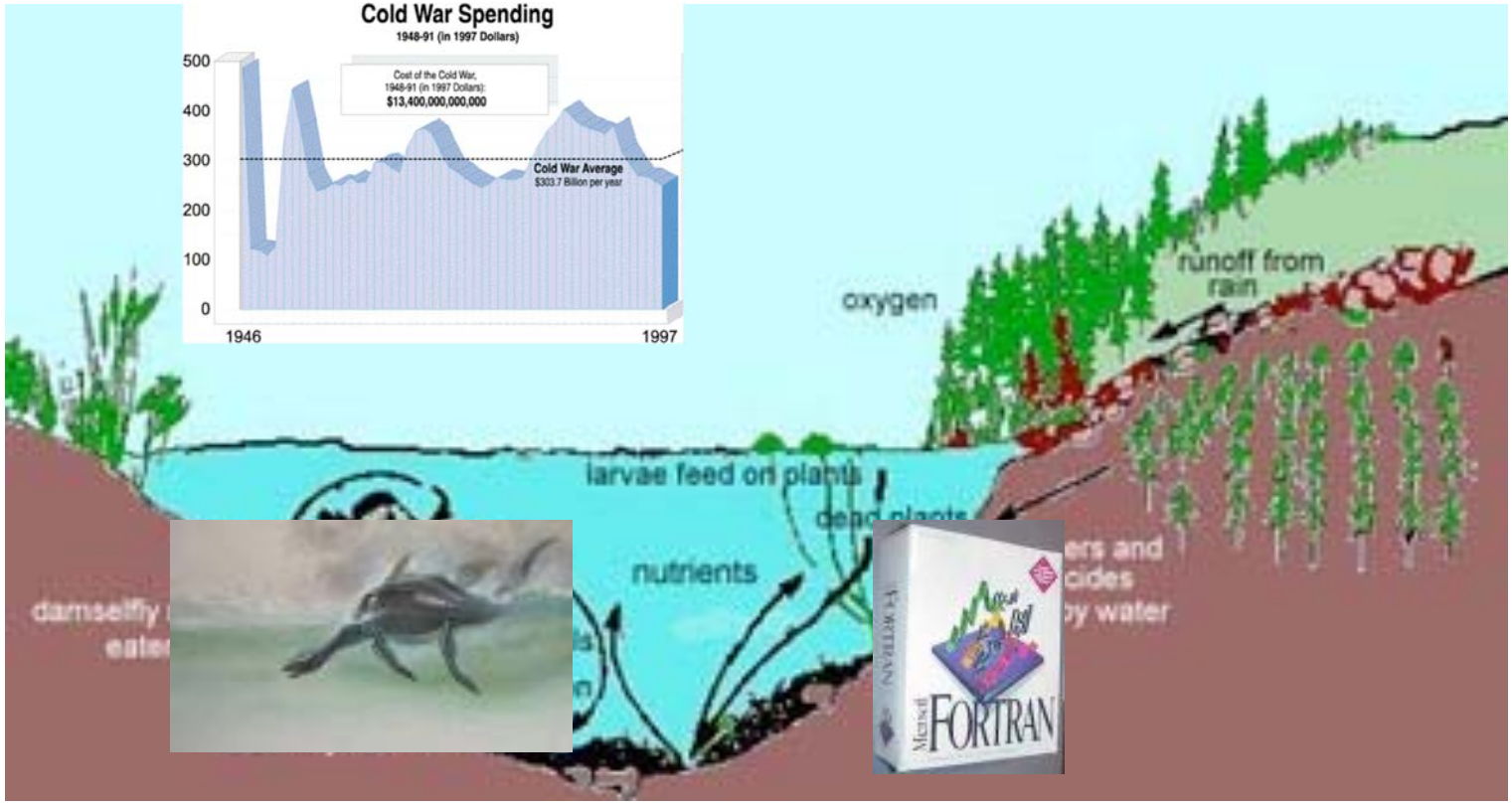
http://www.tpwd.state.tx.us/expltx

/eft/txwild/pond.htm



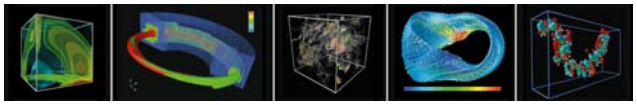
Supercomputing Ecosystem (1988)

Cold War and Big Oil spending in the 1980s



Powerful Vector Supercomputers

20 years of Fortran applications base in physics codes and third party apps

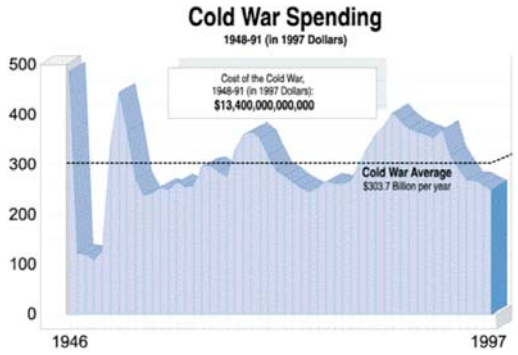


Berkeley Lab

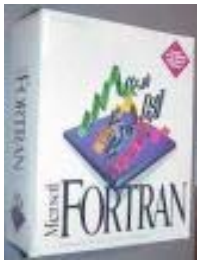
Office of Science
U.S. Department of Energy

Supercomputing Ecosystem (until about 1988)

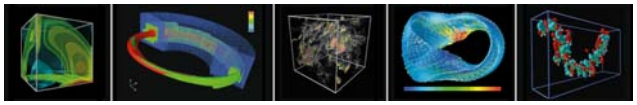
Cold War and Big Oil spending in the 1980s



Powerful Vector Supercomputers



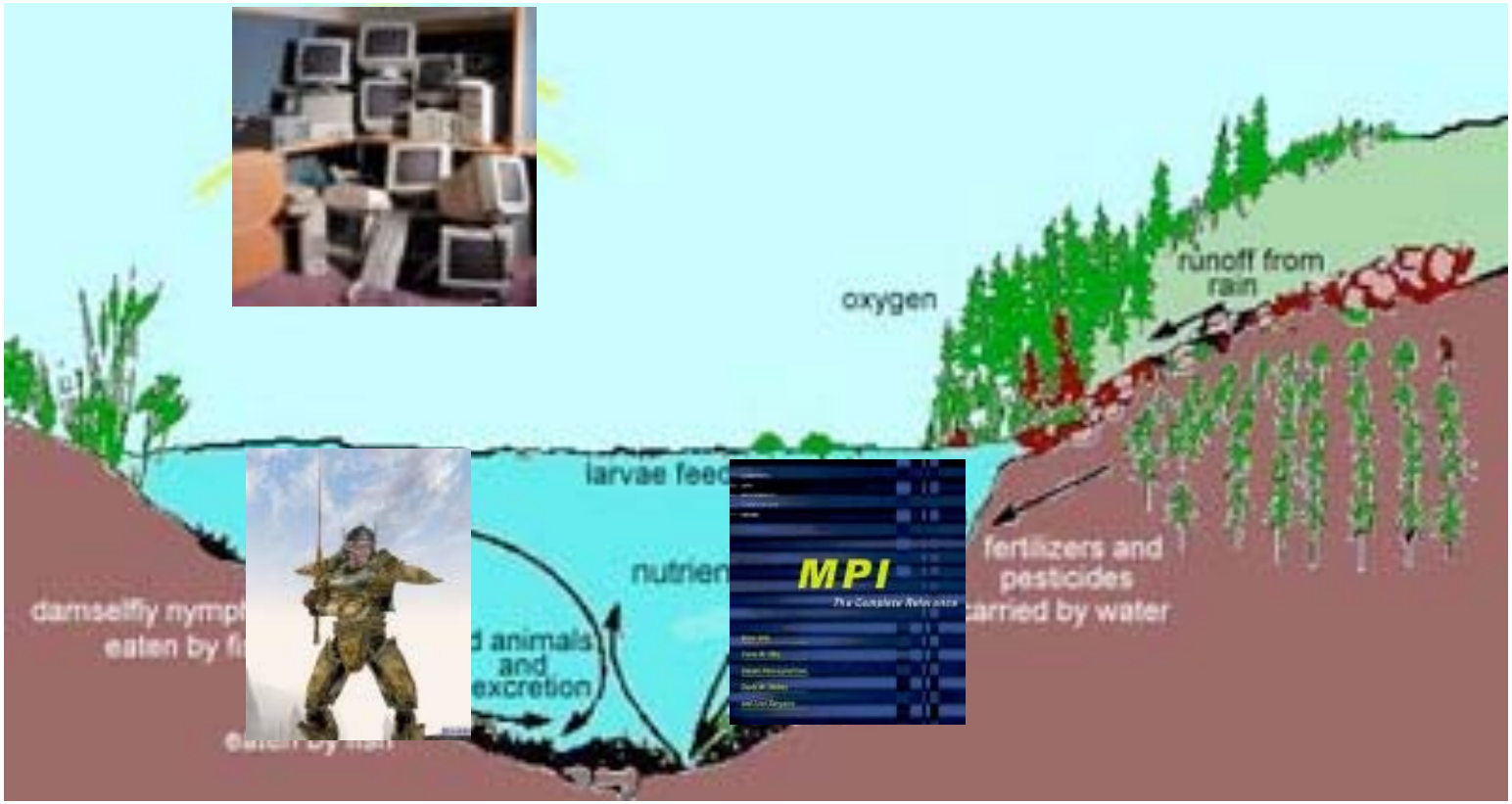
20 years of Fortran applications base in physics codes and third party apps



Office of Science
U.S. DEPARTMENT OF ENERGY

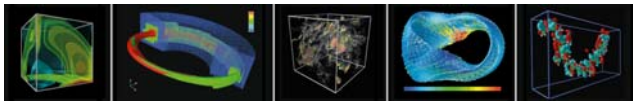
Supercomputing Ecosystem (2005)

Commercial Off The Shelf technology (COTS)



“Clusters”

12 years of legacy MPI applications base



Office of Science
U.S. DEPARTMENT OF ENERGY

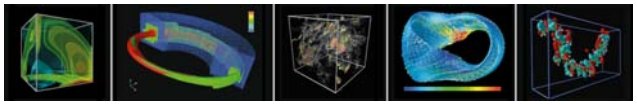
Supercomputing Ecosystem (2005)

Commercial Off The Shelf technology (COTS)

“Clusters”

12 years of legacy MPI applications base

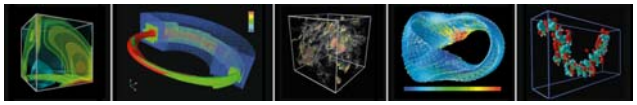
BERKELEY LAB



Office of Science
U.S. DEPARTMENT OF ENERGY

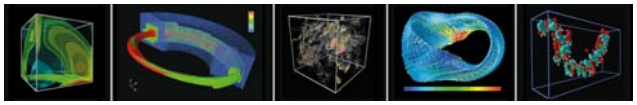
How Did We Make the Change?

- **Massive R&D Investment**
  - **HPCC in the US**
  - **Vigorous computer science experimentation in languages, tools, system software**
  - **Development of Grand Challenge applications**
- **External Driver**
  - **Industry transition to CMOS micros**
- **All changes happened virtually at once**
  - **Ecosystem change**



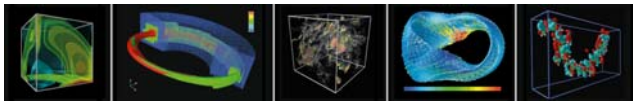
Observations on the 2005 Ecosystem

- It is very stable
  - attempts of re-introducing old species failed (X1)
  - attempts of introducing new species failed (mutation of Blue Gene 1999 to BG/L 2005)
- It works well
  - just look around the room
- So why isn't everybody happy and content?



Challenge 2010 - 2018: Developing a New Ecosystem for HPC

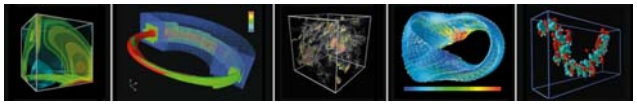
- The supercomputing community is aware that the current situation is suboptimal for HPC
  - "divergence problem" and Blue Planet at NERSC
  - concern about the "right" benchmarks
- The current ecosystem will become untenable after about 2010 in the face of the architectural and software challenges
- How are we going to change the ecosystem?
- What are we going to change it into?
- DARPA HPCS is on the right track combining requirements for new architecture, new languages, and insistence on commercialization by vendors
- But, will a $150M program be enough to change a $6 - 8B industry?



Challenge 2015 - 2025: The Coming "Flattening" of Moore's Law

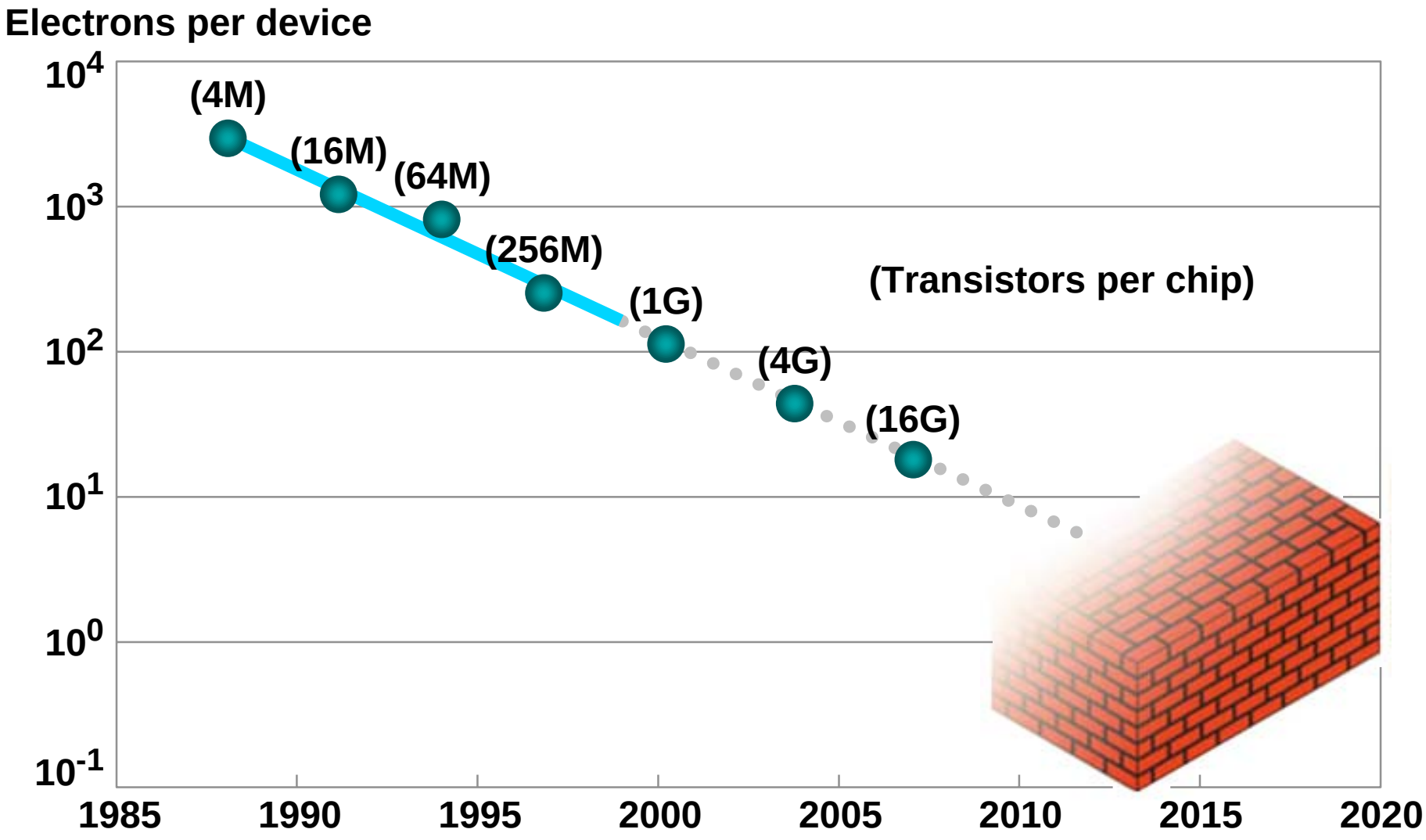
Some time between 2015 and 2025 the continued performance growth of semiconductor based microprocessors will end.

for more details see presentation by Erik DeBenedictis, Sandia, at www.zettaflops.org

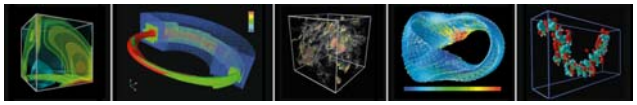


Office of Science
U.S. DEPARTMENT OF ENERGY

Vanishing Electrons (2016)



Source: Joel Birnbaum, HP, Lecture at APS Centennial, Atlanta, 1999

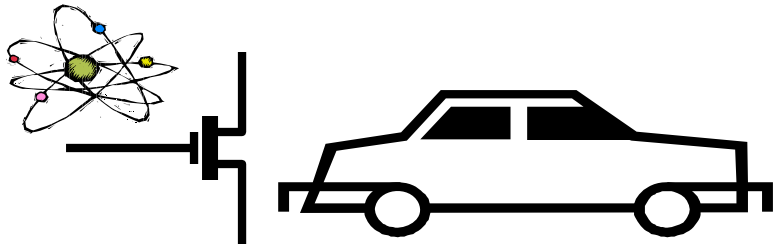


Office of Science
U.S. DEPARTMENT OF ENERGY

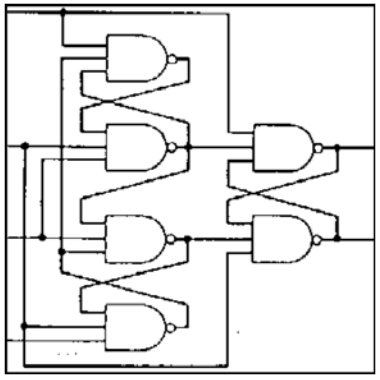
FM Radio and End of Moore's Law

(Adapted from Erik DeBenedictis, Sandia, at www.zettaflops.org)

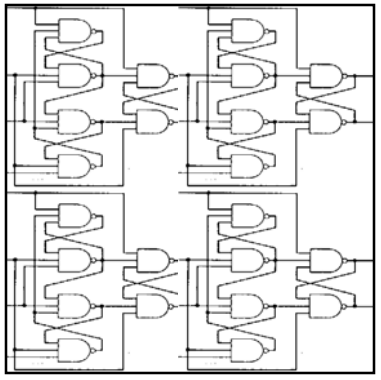
Distance



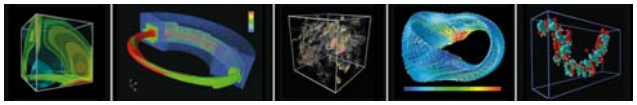
Driving away from FM transmitter→less signal
Noise from electrons → no change



Shrink



Increasing numbers of gates→less signal power
Noise from electrons → no change



Office of Science
U.S. DEPARTMENT OF ENERGY

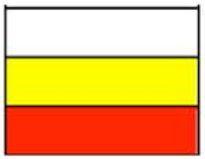
Semiconductor Roadmap

YEAR OF PRODUCTION	2010	2013	2016
DRAM ½ PITCH (nm)	45	32	22
MPU / ASIC ½ PITCH (nm)	50	35	25
MPU PRINTED GATE LENGTH (nm)	25	18	13
MPU PHYSICAL GATE LENGTH (nm)	18	13	9
Physical gate length high-performance (HP) (nm) [1]	18	13	9
Equivalent physical oxide thickness for high-performance T_{ox} (EOT)(nm) [2]	0.5-0.8	0.4-0.6	0.4-0.5
Gate depletion and quantum effects electrical thickness adjustment factor (nm) [3]	0.5	0.5	0.5
T_{ox} electrical equivalent (nm) [4]	1.2	1.0	0.9
Nominal power supply voltage (V_{dd}) (V) [5]	0.6	0.5	0.4
Nominal high-performance NMOS sub threshold leakage current, $I_{sd,leak}$ (at 25°C) (μA/μm) [6]	3	7	10
Nominal high-performance NMOS saturation drive current , I_{dd} (at V_{dd}, at 25°C) (μA/μm) [7]	1200	1500	1500
Required percent current-drive "mobility/transconductance improvement" [8]	30%	70%	100%
Parasitic source/drain resistance (Rsd) (ohm-μm) [9]	110	90	80
Parasitic source/drain resistance (Rsd) percent of ideal channel resistance (V_{dd}/I_{dd}) [10]	25%	30%	35%
Parasitic capacitance percent of ideal gate capacitance [11]	31%	36%	42%
High-performance NMOS device τ ($C_{gate} * V_{dd} / I_{dd}$-NMOS)(ps) [12]	0.39	0.22	0.15
Relative device performance [13]	4.3	7.2	10.7
Energy per (W/L_{gate}=3) device switching transition ($C_{gate}*(3*L_{gate})*V^2$) (fJ/Device) [14]	0.015	0.007	0.002
Static power dissipation per (W/Lgate=3) device (Watts/Device) [15]	9.7E-08	1.4E-07	1.1E-07

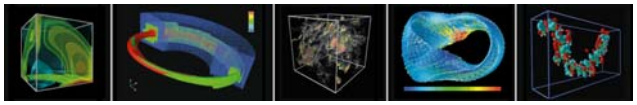
White—Manufacturable Solutions Exist, and Are Being Optimized

Yellow—Manufacturable Solutions are Known

Red—Manufacturable Solutions are NOT Known



BERKELEY LAB

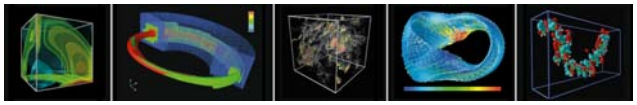


Office of Science
U.S. DEPARTMENT OF ENERGY

ITRS Device Review 2016

Technology	Speed (min-max)	Dimension (min-max)	Energy per gate-op	Comparison
CMOS	30 ps-1 μs	8 nm-5 μm	4 aJ	
RSFQ	1 ps-50 ps	300 nm- 1μm	2 aJ	Larger
Molecular	10 ns-1 ms	1 nm- 5 nm	10 zJ	Slower
Plastic	100 μs-1 ms	100 μm-1 mm	4 aJ	Larger+Slower
Optical	100 as-1 ps	200 nm-2 μm	1 pJ	Larger+Hotter
NEMS	100 ns-1 ms	10-100 nm	1 zJ	Slower+Larger
Biological	100 fs-100 μs	6-50 μm	.3 yJ	Slower+Larger
Quantum	100 as-1 fs	10-100 nm	1 zJ	Larger

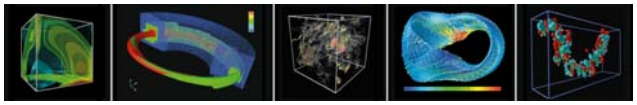
Data from ITRS ERD Section, quoted from Erik DeBenedictis, Sandia Lab.



Office of Science
U.S. DEPARTMENT OF ENERGY

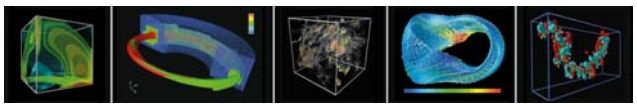
Challenge 2015 - 2025: The Coming "Flattening" of Moore's Law

- There is no active, strong research program anywhere that addresses this challenge
- Alternative technology solutions are feasible, but won't come by themselves
- After 50 years of exponential growth, how will the industry adjust to a no-growth scenario?



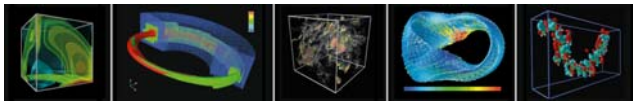
The Three Future Challenges

- **Learn how to scale**
- **Change the ecosystem**
- **Deal with the “flattening”of Moore’s law**



Hans, Congratulations on 20 years of ISC!

Thank you from all of us for providing such an excellent venue for learning and discussions.



Office of Science
U.S. Department of Energy