Challenges of Exascale Computing

Dick Watson

Lawrence Livermore National Laboratory

Talk given at University of San Francisco

May 3, 2011

The slides, some with minor edits for this talk, came from several authors. The source of each slide is keyed to the References

LLNL-PRES-482474

Increasing Machine Capability

- Gigaflop = one billion (1,000,000,000,000) floating point operations (flops) per second

Got here in 1985 – Cray-2

- Teraflop = ~1024 gigaflops, or roughly 1 trillion flops

Got here in 1997 – Cray ASCI Red

- Petaflop = ~1 quadrillion (or 10^{15})flops, or 1024 teraflops

Got here in 2008 – IBM Roadrunner

- Exaflop = 1 quintillion (or 10^{18}) flops, or 1 million teraflops

Hope to get here around 2020

Source [4]

Key Message

- The transition from petascale to exascale will be characterized by significant and dramatic changes in hardware and software architectures.
- This transition will be disruptive, but create unprecedented opportunities for computer and computational science R&D.

Source [6]

Exascale Challenges

Exascale ≠ Petascale X 1000

- Total concurrency in the applications must rise by a factor of ~1 million;
- Memory per processor falls dramatically which makes current weak scaling approaches problematic;
- For both power and performance reasons, locality of data and computation is much more important
- The failure rates for components and manufacturing variability make it unreasonable to assume the computer is deterministic. This is true for performance today and will affect the results of computations by 2018 due to silent errors.
- Synchronization will be very expensive. In addition, work required to manage synchronization is high.
- The I/O system at all levels – chip to memory, memory to I/O node, I/O node to disk— will be much harder to manage due to the relative speeds of the components.

Source [4]

DOE mission imperatives require simulation and analysis for policy and decision making

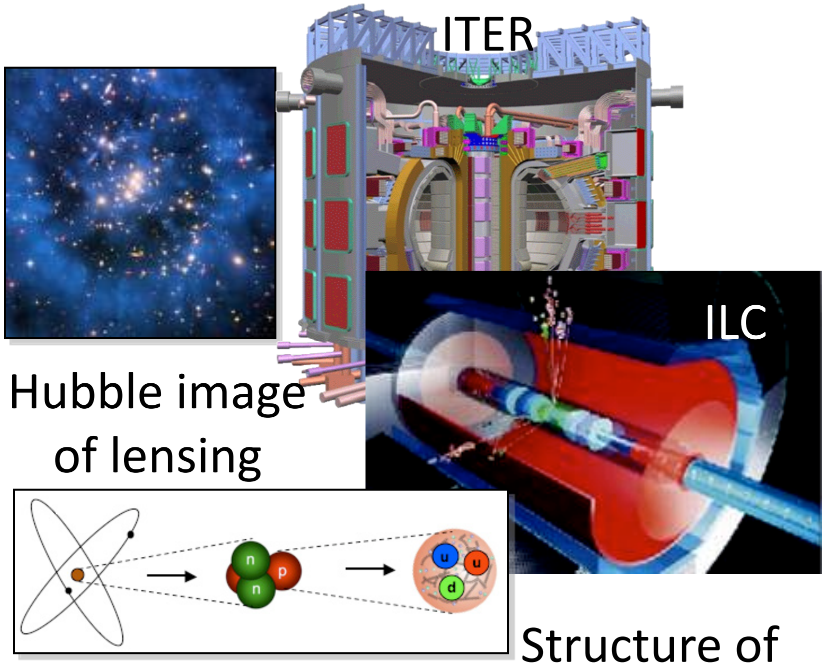
- *Climate Change*: Understanding, mitigating and adapting to the effects of global warming
  - Sea level rise
  - Severe weather
  - Regional climate change
  - Geologic carbon sequestration
- *Energy*: Reducing U.S. reliance on foreign energy sources and reducing the carbon footprint of energy production
  - Reducing time and cost of reactor design and deployment
  - Improving the efficiency of combustion energy systems
- *National Nuclear Security*: Maintaining a safe, secure and reliable nuclear stockpile
  - Stockpile certification
  - Predictive scientific challenges
  - Real-time evaluation of urban nuclear detonation

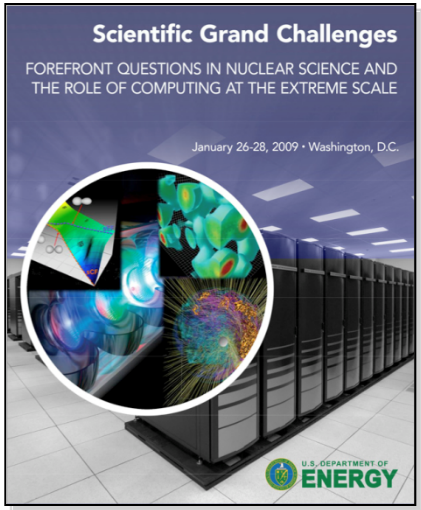
Accomplishing these missions requires exascale resources.

Source [7]

Exascale simulation will enable fundamental advances in basic science.

- High Energy & Nuclear Physics
  - Dark-energy and dark matter
  - Fundamentals of fission fusion reactions
- Facility and experimental design
  - Effective design of accelerators
  - Probes of dark energy and dark matter
  - ITER shot planning and device control
- Materials / Chemistry
  - Predictive multi-scale materials modeling: observation to control
  - Effective, commercial technologies in renewable energy, catalysts, batteries and combustion
- Life Sciences
  - Better biofuels
  - Sequence to structure to function



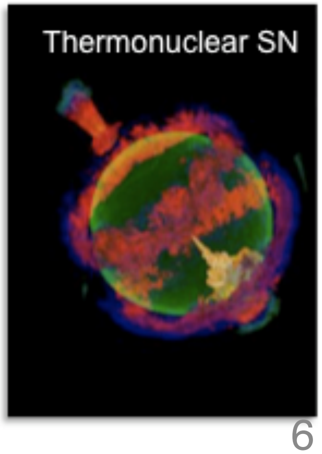


Scientific Grand Challenges

FOREFRONT QUESTIONS IN NUCLEAR SCIENCE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE

January 26-28, 2009 • Washington, D.C.

U.S. DEPARTMENT OF ENERGY



These breakthrough scientific discoveries and facilities require exascale applications and resources.

Source [7]

ASCI Red: World's Most Powerful Computer in 1999

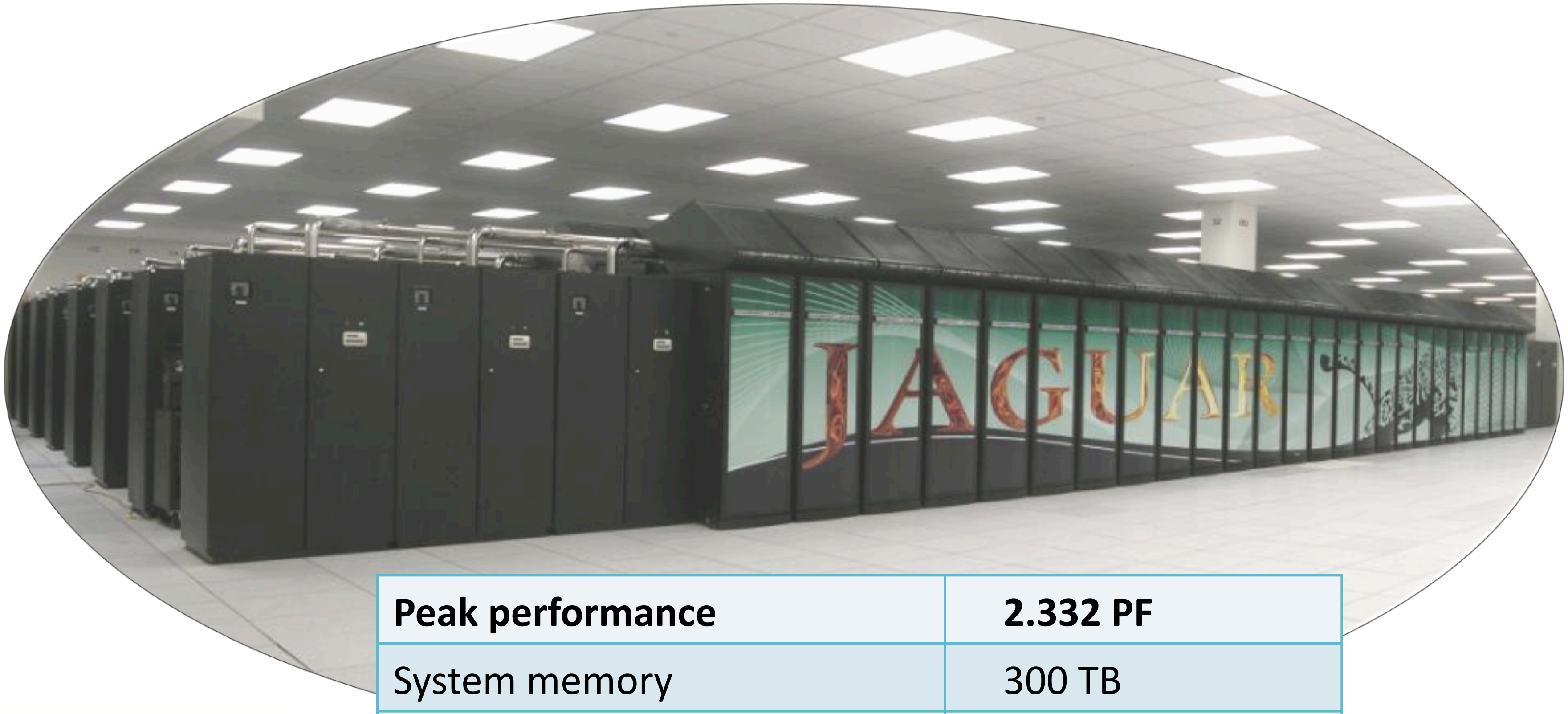
TOP500® SUPERCOMPUTER SITES

#1 Nov. 1999

Peak performance	**3.154 TF**
System memory	1.212 TB
Disk space	12.5 TB
Processors	9298
Power	850 kW

Source [6]

Jaguar: World’s most powerful computer in 2009



Peak performance	**2.332 PF**
System memory	300 TB
Disk space	10 PB
Processors	224K
Power	6.95 MW

TOP500® SUPERCOMPUTER SITES

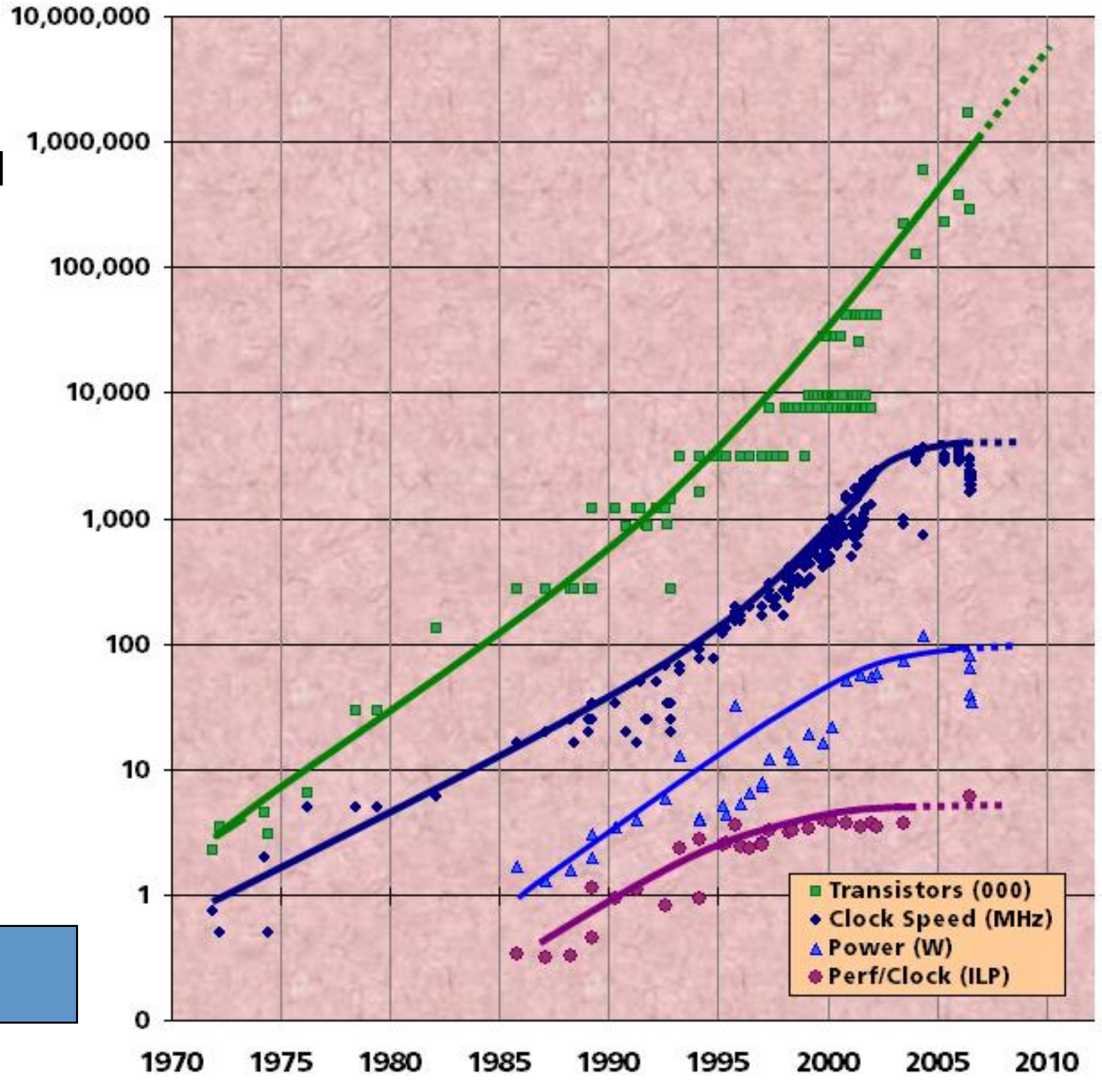
#1 Nov. 2009

Source [6]

Traditional Sources of Performance Improvement are Flat-Lining (2004)

- New Constraints
  - 15 years of *exponential* clock rate growth has ended
- Moore's Law reinterpreted:
  - How do we use all of those transistors to keep performance increasing at historical rates?
  - Industry Response: #cores per chip doubles every 18 months *instead* of clock frequency!

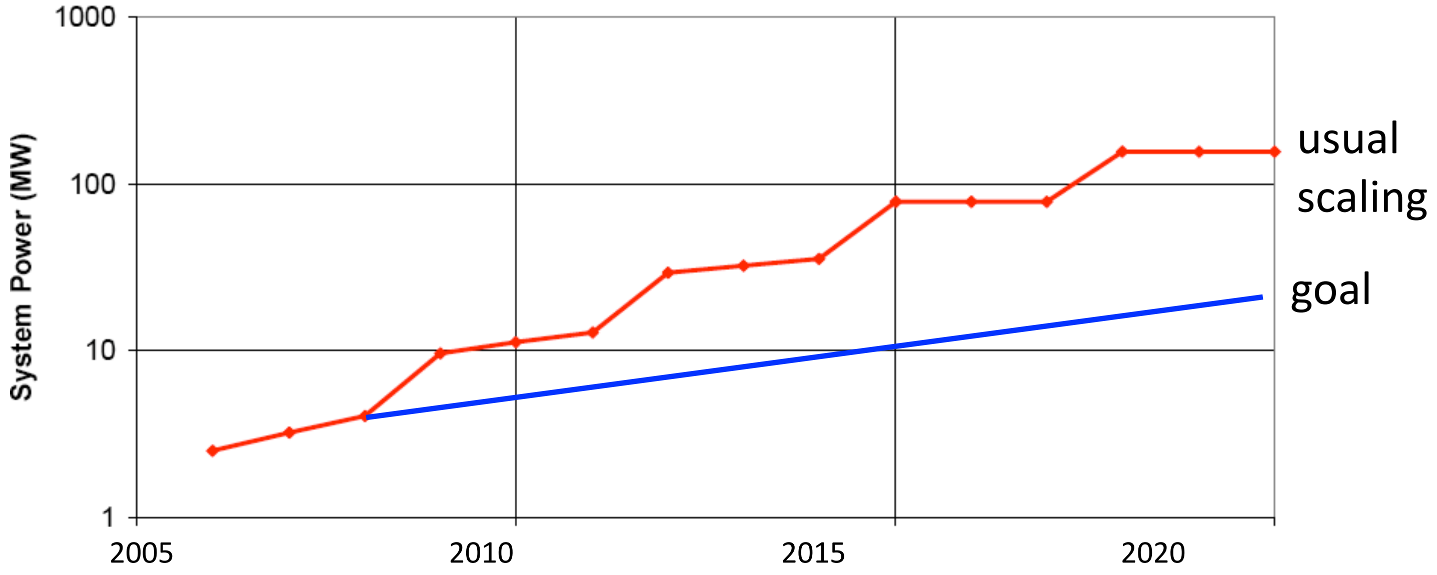
Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith



Source [8]

Exascale Is All About Energy Efficient Computing

- At $1M per MW, energy costs are substantial
- 1 petaflop in 2010 uses 3 MW
- 10 petaflop in 2011 uses 15 MW
- 1 exaflop in 2018 at 200 MW with “usual” scaling
- 1 exaflop in 2018 at 20 MW is target



Source [3]

Reducing power is fundamentally about architectural choices & process technology

- **Processor (10x-20x)**
  - **Reducing data movement (functional reorganization, > 20x)**
  - **Domain/Core power gating and aggressive voltage scaling**
- **Memory (2x-5x)**
  - **New memory interfaces (optimized memory control and xfer)**
  - **Extend DRAM with non-volatile memory**
- **Interconnect (2x-5x)**
  - **More interconnect on package**
  - **Replace long haul copper with integrated optics**
- **Data Center Energy Efficiencies (10%-20%)**
  - **Higher operating temperature tolerance**
  - **Power supply and cooling efficiencies**

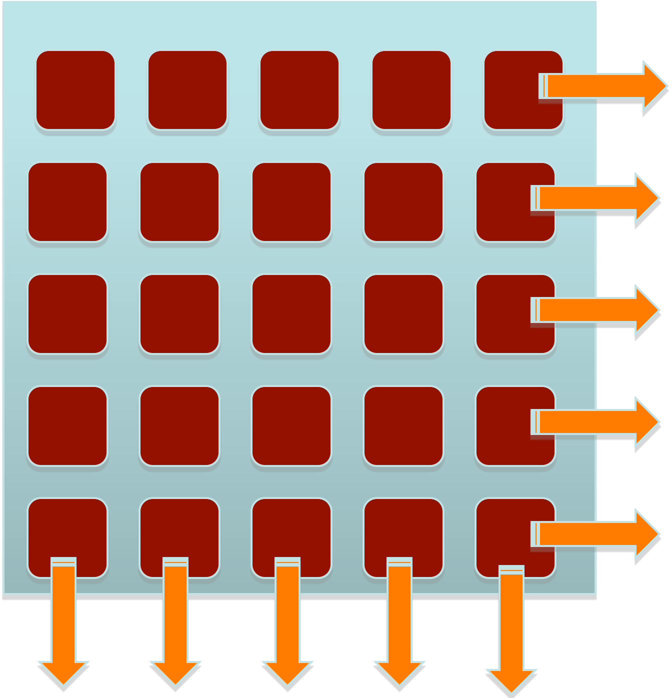
Source [7]

Potential System Architecture Targets

System attributes	2010	“2015”		“2018”	
System peak	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32-64 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/sec	1 TB/sec	0.4 TB/sec	4 TB/sec
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW	1.5 GB/s	20 GB/sec		200 GB/sec	
MTTI	days	O(1day)		O(1 day)	

Source [7]

Future of On-Chip Architecture

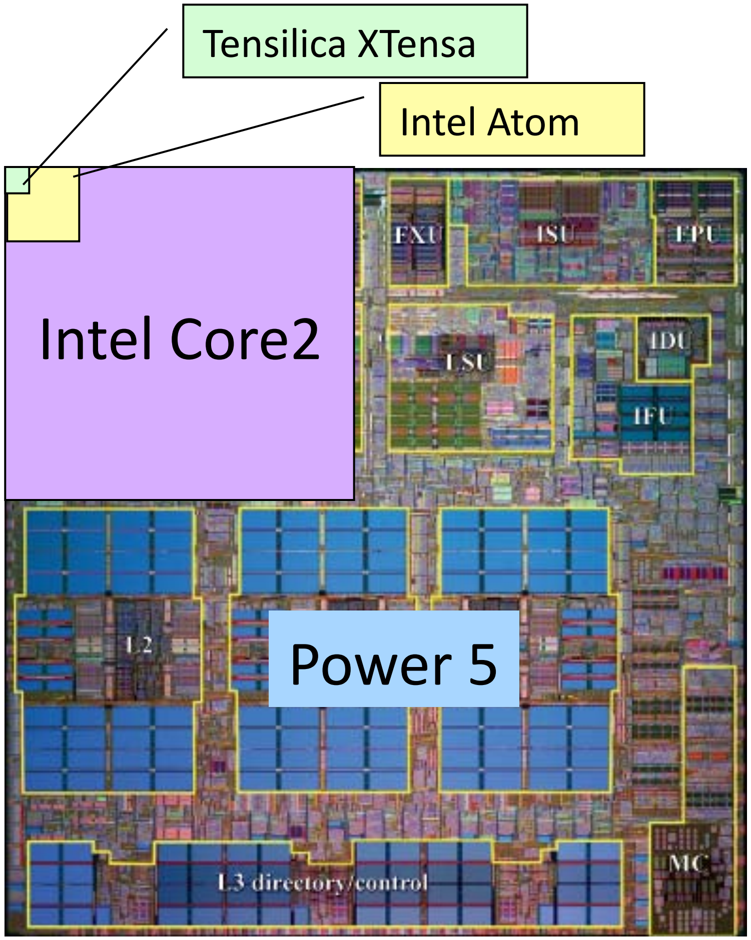


Scale-out for Planar geometry

- ~1000-10k simple cores /Chip
  - 4-8 wide SIMD or VLIW bundles
  - Either 4 or 50+ HW threads
- On-chip communication Fabric
  - Low-degree topology for on-chip communication (torus or mesh)
  - *Scale cache coherence?*
  - Global (nonCC memory)
  - Shared register file (clusters)
- Off-chip communication fabric
  - Integrated directly on an SoC
  - Reduced component counts
  - Coherent with TLB (no pinning)

Source [3]

Low-Power Design Principles

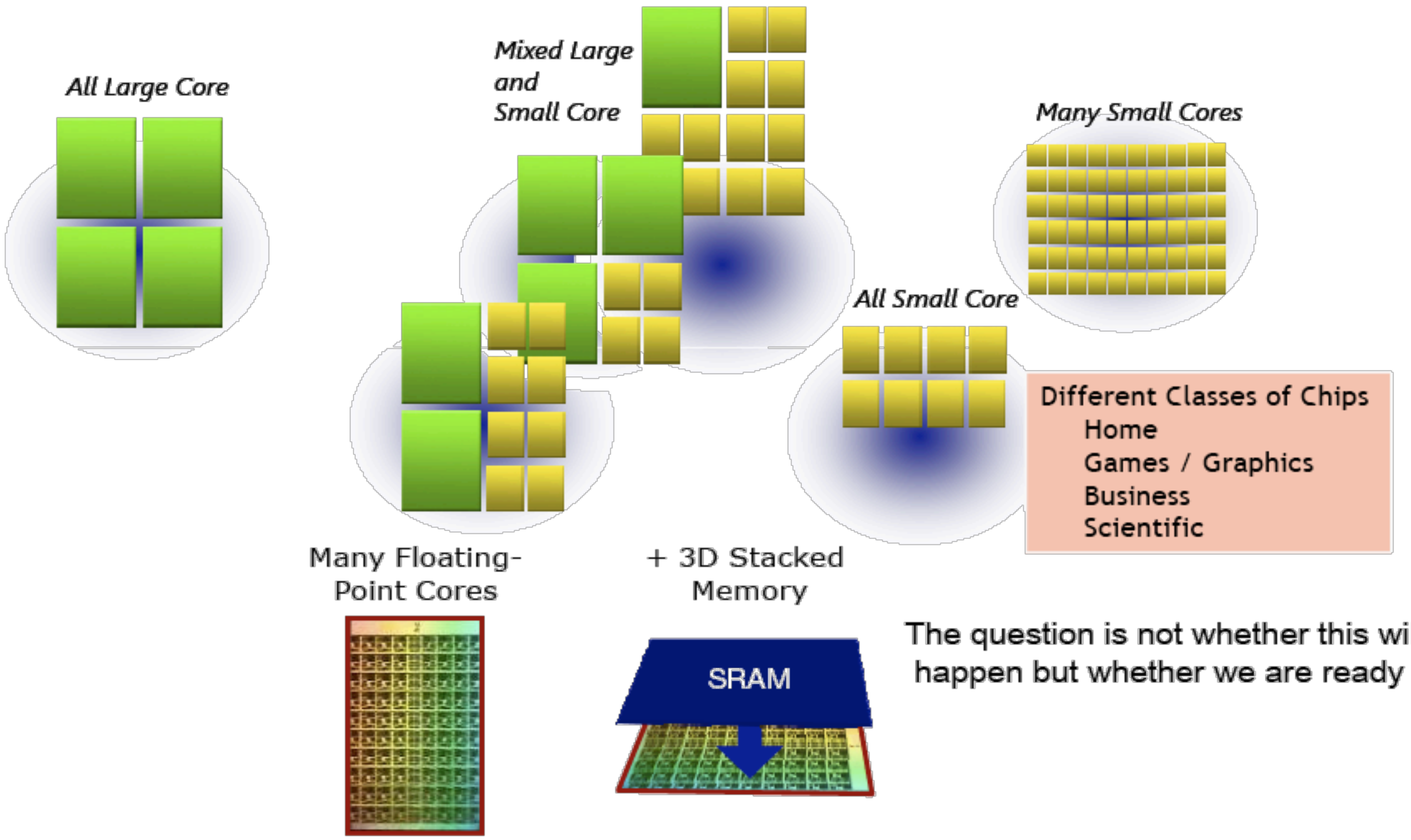


- Cubic power improvement with lower clock rate due to V^2F
- Slower clock rates enable use of simpler cores
- Simpler cores use less area (lower leakage) and reduce cost
- Tailor design to application to REDUCE WASTE

This is how iPhones and MP3 players are designed to maximize battery life and minimize cost

Source [6]

What’s Next?

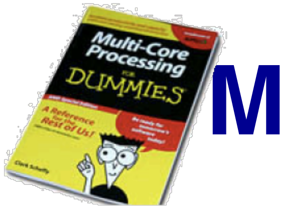


Different Classes of Chips
- Home
- Games / Graphics
- Business
- Scientific

The question is not whether this will happen but whether we are ready

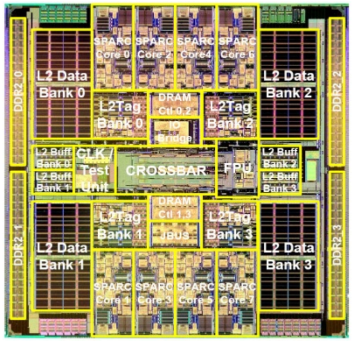
Source: Jack Dongarra, ISC 2008

Source [6]


Multi-Core Processing for DUMMIES
A Reference for the Rest of Us!

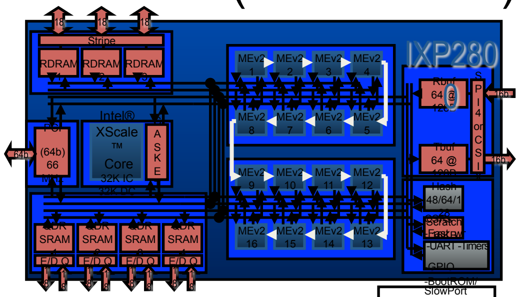
Multicore comes in a wide variety

- Multiple parallel general-purpose processors (GPPs)
- Multiple application-specific processors (ASPs)

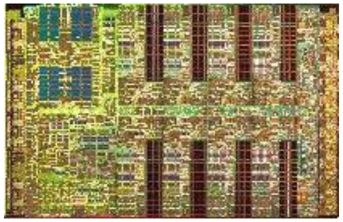


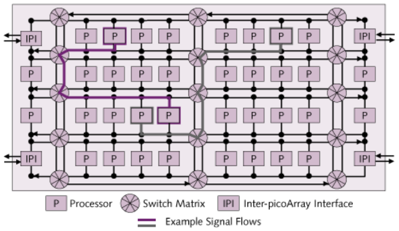
Sun Niagara
8 GPP cores (32 threads)

Intel Network Processor
1 GPP Core
16 ASPs (128 threads)

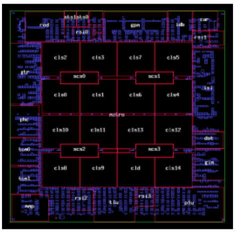


IBM Cell
1 GPP (2 threads)
8 ASPs

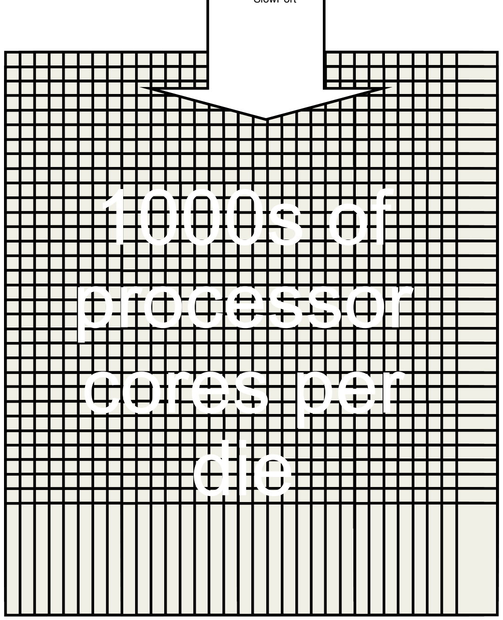




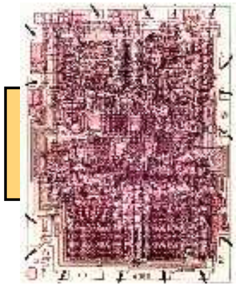
Picochip DSP
1 GPP core
248 ASPs



Cisco CRS-1
188 Tensilica GPPs



1000s of processor cores per die



Intel 4004 (1971):
4-bit processor,
2312 transistors,
~100 KIPS,
10 micron PMOS,
11 mm^2 chip

"The Processor is the new Transistor" [Rowen]

Source [8]

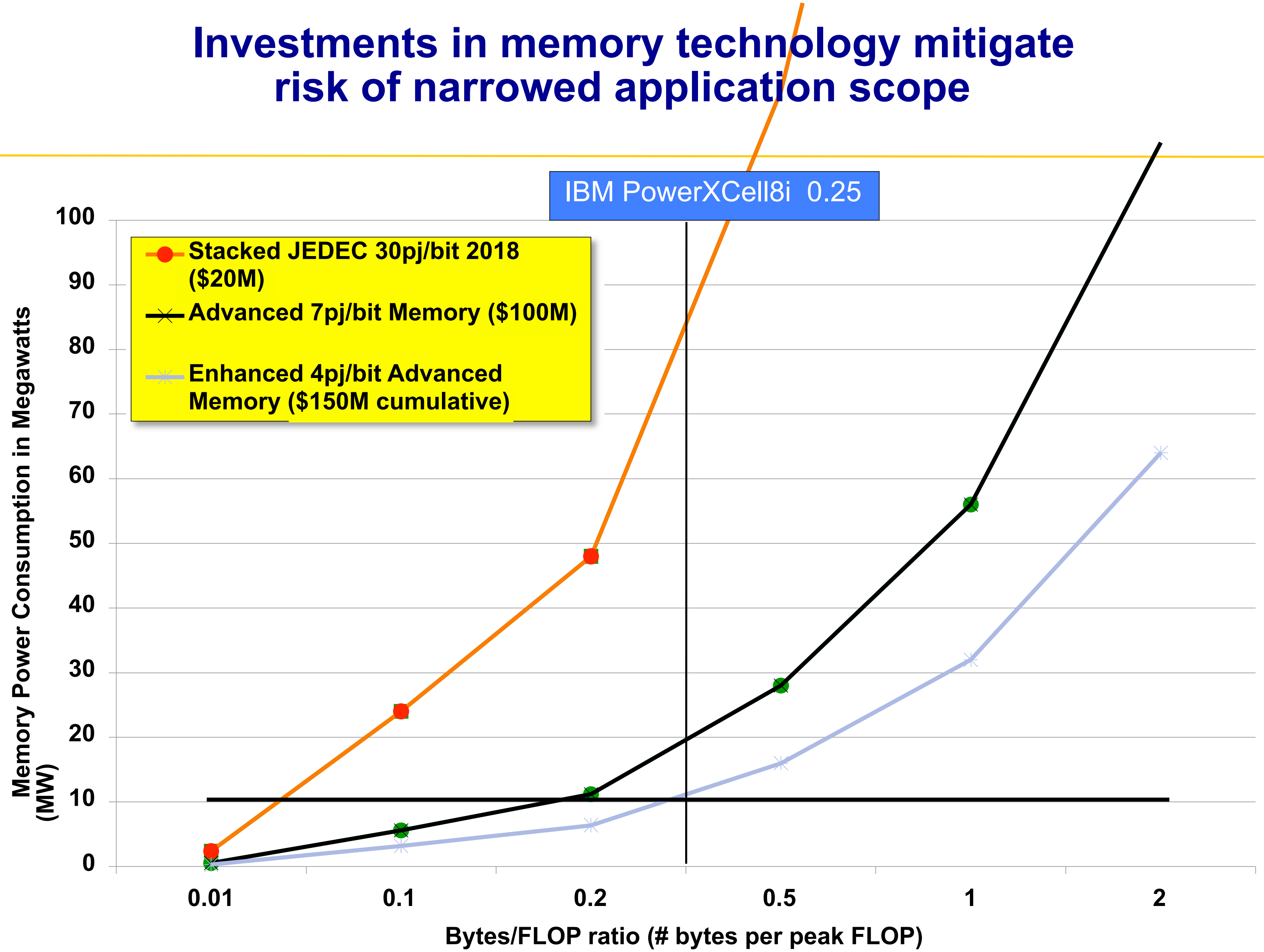
Science at Scale

- “From a scientist’s perspective, the ratio of memory to processor is critical in determining the size of the problem that can be solved. Remember that the processor dictates how much computing can be done; the memory dictates the size of the problem that can be handled. In the Exascale design…there is 500 times more compute power, however only 30 times the memory, so applications cannot just scale to the speed of the machine. Scientists and computer scientists will have to rethink how they are going to use these systems. This factor of >10 loss in memory/compute power means potentially totally redesigning the current application codes.”

P.49 ASCAC Exascale report, October 2010

Source [4]

Investments in memory technology mitigate risk of narrowed application scope

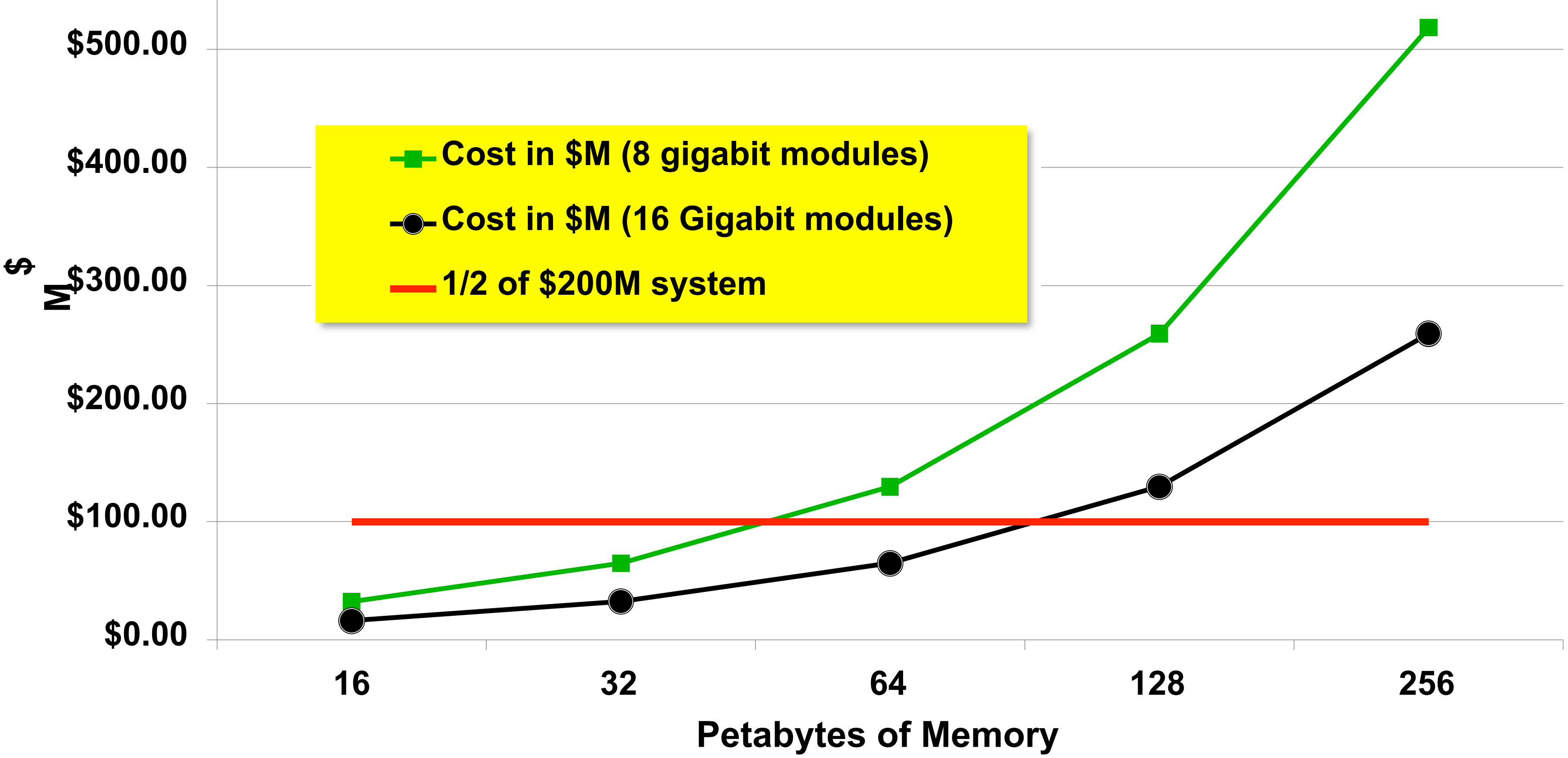


Source [7]

Cost of Memory Capacity
for two different potential memory Densities

- Memory density is doubling every three years; processor logic, every two
  - Project 8 Gigabit DIMMs in 2018
  - 16 Gigabit if technology acceleration

- Storage costs are dropping gradually compared to logic costs
  - Industry assumption is $1.80/memory chip is median commodity cost

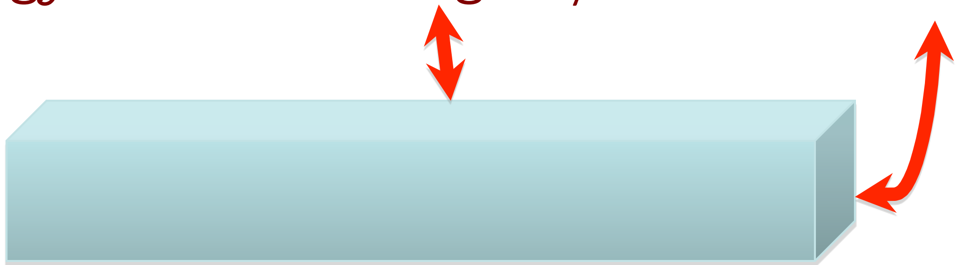


Source [7]

The problem with Wires:

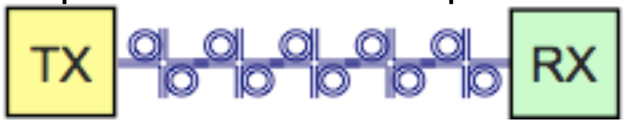
Energy to move data proportional to distance

- Cost to move a bit on copper wire:
  - energy = bitrate * Length^2 / cross-section area

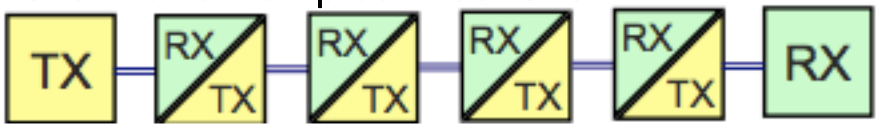


- Wire data capacity constant as feature size shrinks
- *Power cost to move bit proportional to distance*
- *~1TByte/sec max feasible off-chip BW (10GHz/pin)*
- *Photonics reduces distance-dependence of bandwidth*

Photonics requires no redrive and passive switch little power



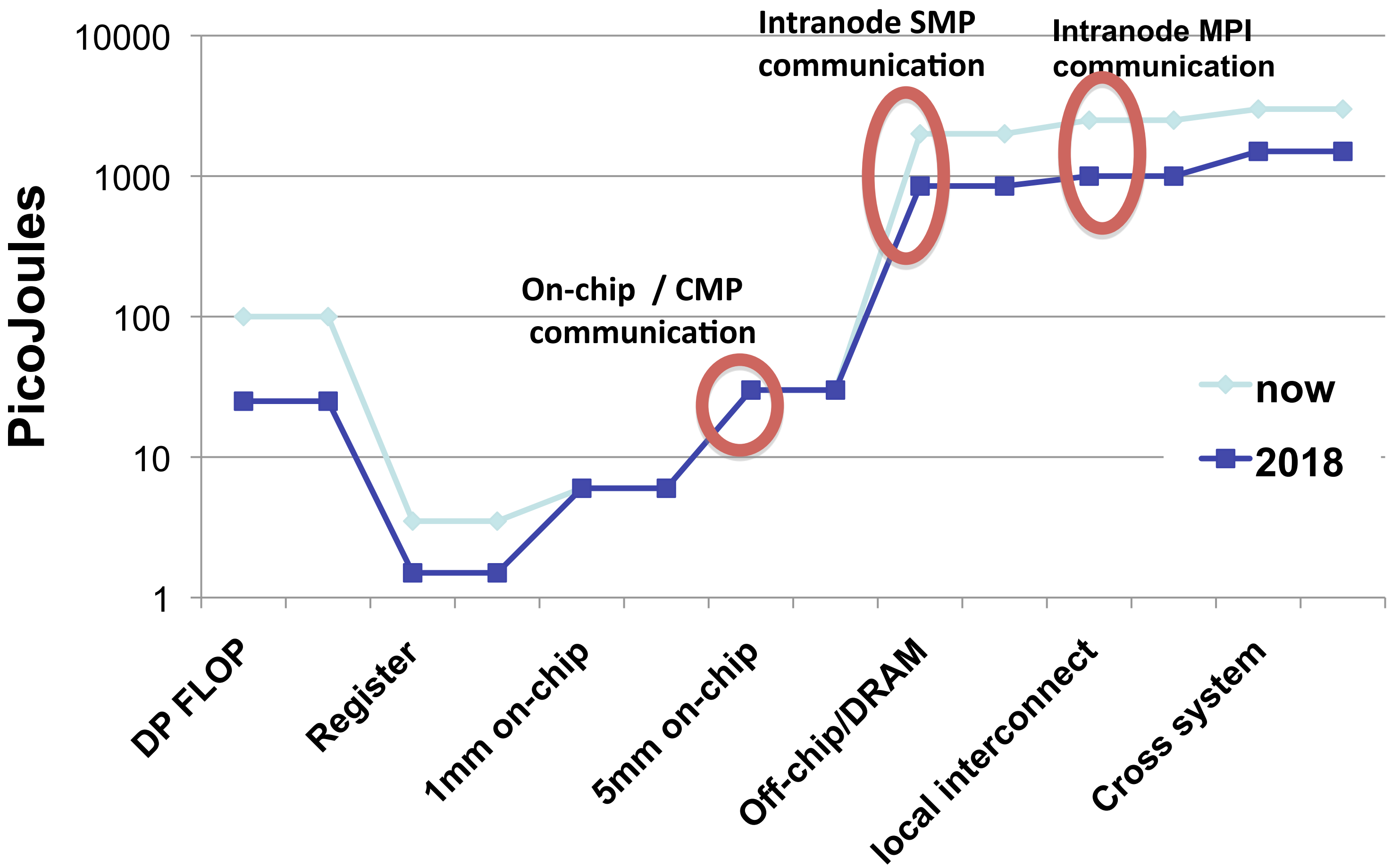
Copper requires signal amplification even for on-chip connections



Source [5]

Data movement costs will not significantly improve in 2018

Energy Efficiency will require careful management of data locality



Important to know when data is on-chip and when data is off-chip!

Source [3]

The Problem with Caches

- **Automatic cache virtualizes the notion of on-chip vs. off-chip memory**
  - Makes on-chip memory indistinguishable from off-chip
  - But energy cost is ~100x if data is off-chip
  - But if you have explicit on-chip memory, then what does that mean for cache-coherence?

- **If you want performance and reduced power, you really need to know the difference between on & off chip**
  - *You can ignore it and be correct, but penalty is ~100x power*

This is why flat programming models for parallelism are <u>NOT</u> in the solution space

If local store is in solution space, then what does it mean to have cache-coherence between local stores?

Source [6]

The Need for Resiliency: Factors Driving up the Fault Rate

It is more than just the increase in the number of components

Number of components both memory and processors will increase by an order of magnitude which will increase hard and soft errors.

Smaller circuit sizes, running at lower voltages to reduce power consumption, increases the probability of switches flipping spontaneously due to thermal and voltage variations as well as radiation, increasing soft errors.

Power management cycling significantly decreases the components lifetimes due to thermal and mechanical stresses.

Resistance to add additional HW detection and recovery logic right on the chips to detect silent errors. Because it will increase power consumption by 15% and increase the chip costs.

Heterogeneous systems make error detection and recovery even harder, for example, detecting and recovering from an error in a GPU can involve hundreds of threads simultaneously on the GPU and hundreds of cycles to drain pipelines to begin recovery.

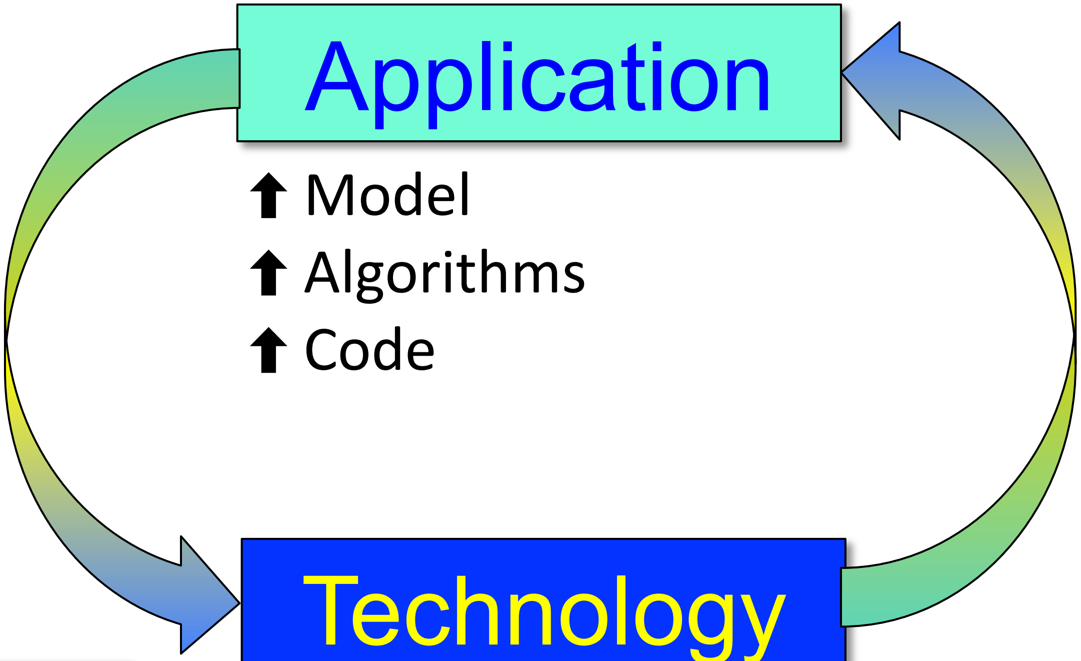
Increasing system and algorithm complexity makes improper interaction of separately designed and implemented components more likely.

Number of operations (10^{23}in a week) ensure that system will traverse the tails of the operational probability distributions.

Source [7]

Co-design expands the feasible solution space to allow better solutions

Application driven:
Find the best technology to run this code.
Sub-optimal



Application

⬆ Model

⬆ Algorithms

⬆ Code

Technology

⊕ architecture

⊕ programming model

⊕ resilience

⊕ power

Technology driven:
Fit your application to this technology.
Sub-optimal.

Now, we must expand the co-design space to find better solutions:

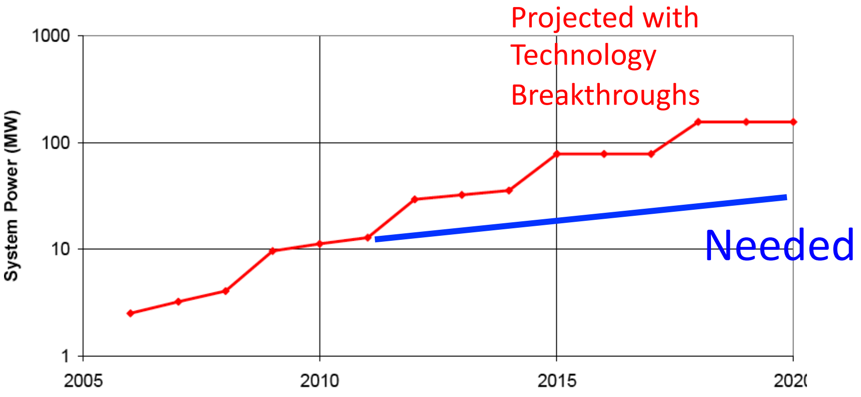
- *new applications & algorithms,*
- *better technology and performance.*

Source [7]

Reviewing

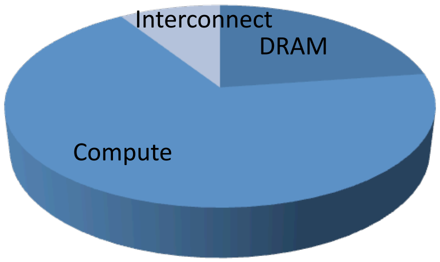
Power

- **Barriers**
  - Power is leading design constraint for computing technology
  - Target ~20MW, estimated > 100MW required for Exascale systems (DARPA, DOE)
  - Efficiency is industry-wide problem (IT technology >2% of US energy consumption and growing)
- **Technical Focus Areas**
  - Energy efficient hardware building blocks (CPU, memory, interconnect)
  - Novel cooling and packaging
  - Si-Photonic Communication
  - **Power Aware Runtime Software and Algorithms**
  - **Programming model support for application power management**
- **Technical Gap**
  - Need 5X improvement in power efficiency over projections that include technological advancements

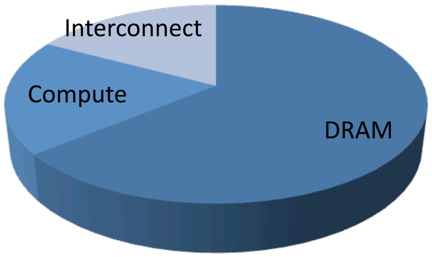


Possible Leadership class power requirements
From Peter Kogge (on behalf of Exascale Working Group), "Architectural *Challenges* at the Exascale Frontier", June 20, 2008

2008 Power Usage



2018 Power Usage



System memory will dominate energy budget if we try to maintain today's ratios

Source [2]

Reliability and Resilience

- **Barriers**
  - Number of system components increasing faster than overall reliability
  - Silent error rates increasing
  - Reduced job progress due to fault recovery if we use existing checkpoint/restart
- **Technical Focus Areas**
  - **Local recovery and migration**
  - **Development of a standard fault model and better understanding of types/rates of faults**
  - Improved hardware and **software** reliability
    - Greater integration across entire stack
  - **Fault resilient algorithms and applications**
  - **New approaches to checkpoint-restart using new non-volatile node-local storage**
- **Technical Gap**
  - Maintaining today's MTTI given 10x - 100X increase in sockets will require:

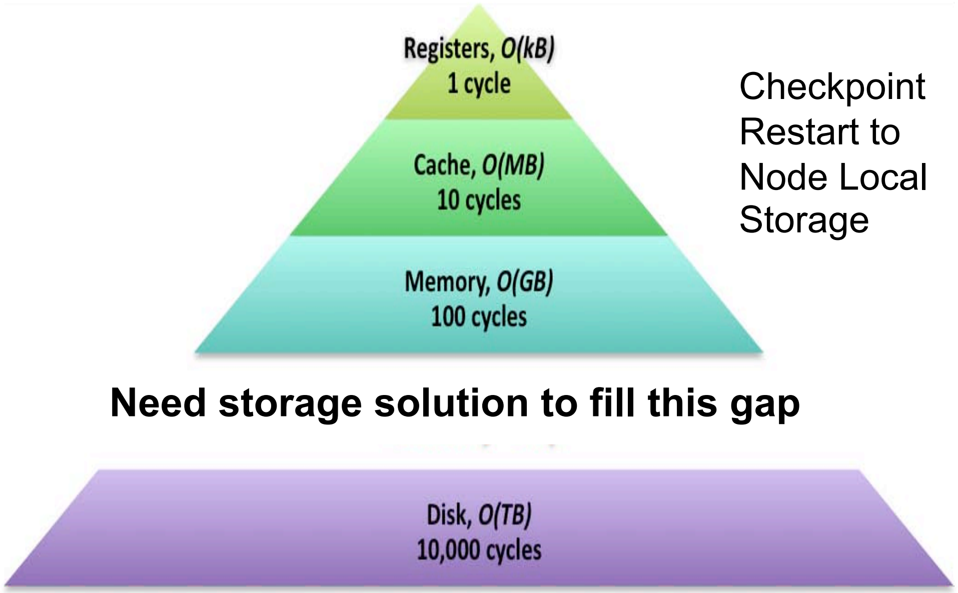
    10X improvement in hardware reliability

    10X in system software reliability, and

    10X improvement due to local recovery and migration as well as research in fault resilient applications

Taxonomy of errors (h/w or s/w)

- **Hard errors**: permanent errors which cause system to hang or crash
- **Soft errors**: transient errors, either correctable or short term failure
- **Silent errors**: undetected errors either permanent or transient. *Concern is that simulation data or calculation have been corrupted and no error reported.*



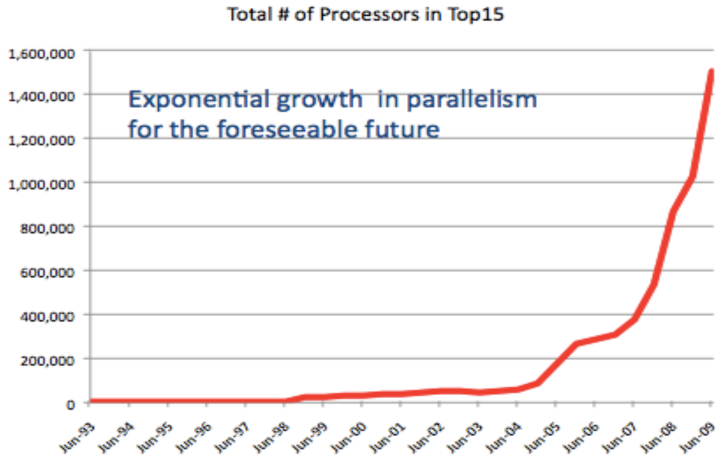
Checkpoint Restart to Node Local Storage

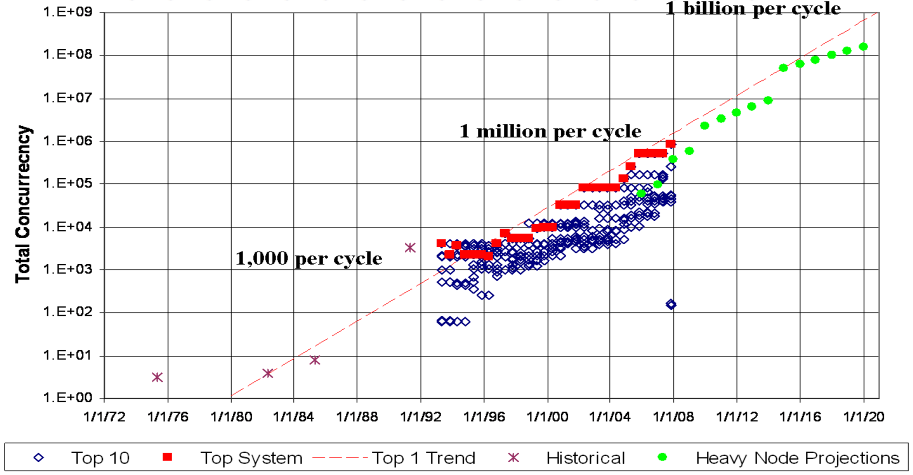
Need storage solution to fill this gap

Source [7]

Parallelism & Locality

- **Barriers**
  - Multiple levels of parallelism
  - Fundamentally breaks scaling assumptions of current software
  - Energy cost for moving data and memory wall
- **Technical Focus Areas**
  - Managing Parallelism
    - **Scalable algorithms**
    - Develop innovative micro-architecture and macro-architectures
  - Managing Locality
    - **Software-managed memory (local store)**
    - **Effective abstractions for explicitly managed memory hierarchies**
    - **Communication avoiding algorithms**
    - **Communication optimized for architecture**
    - **Fine-grained concurrency**
- **Technical Gap**
  - Need 1,000X further scaling of applications.





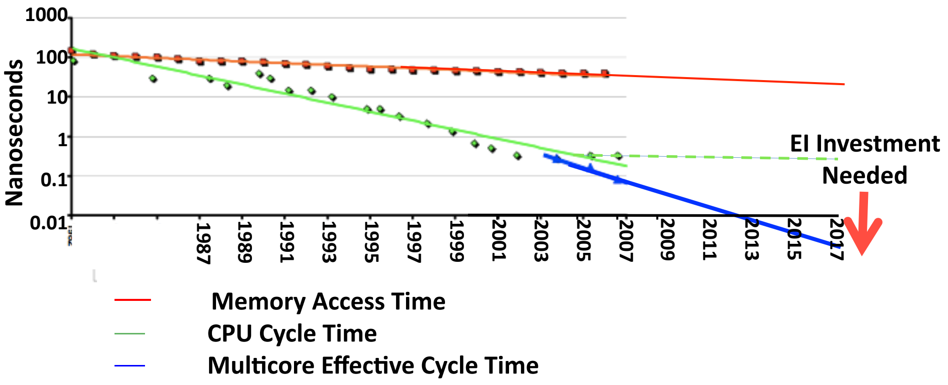
How much parallelism must be handled by the program?

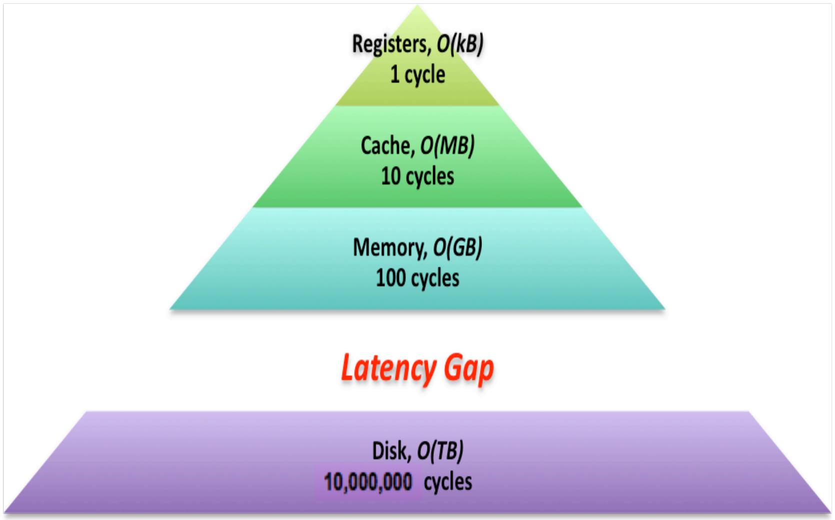
From Peter Kogge (on behalf of Exascale Working Group), "Architectural *Challenges* at the Exascale Frontier", June 20, 2008

Source [2]

Memory and Storage

- **Barriers**
  - *Per-disk performance, failure rates, and energy efficiency no longer improving*
  - *Linear extrapolation of DRAM vs. Multi-core performance means the height of the memory wall is accelerating*
  - *Off-chip bandwidth, latency, combined with poor concurrency are throttling delivered performance*
- **Technical Focus Areas**
  - *Efficient Data Movement*
    - Photonic DRAM interfaces
    - Optical interconnects / routers
    - **Communications optimal algorithms**
  - *New Storage Approaches*
    - Non-volatile memory gap fillers
    - Advanced packaging (chip stacking)
    - **Storage efficient programming models, algorithms and run-time systems**
- **Technical Gap**
  - *Need 10X improvement in memory access speeds to keep current balance with computation.*





Registers, *O(kB)*
1 cycle

Cache, *O(MB)*
10 cycles

Memory, *O(GB)*
100 cycles

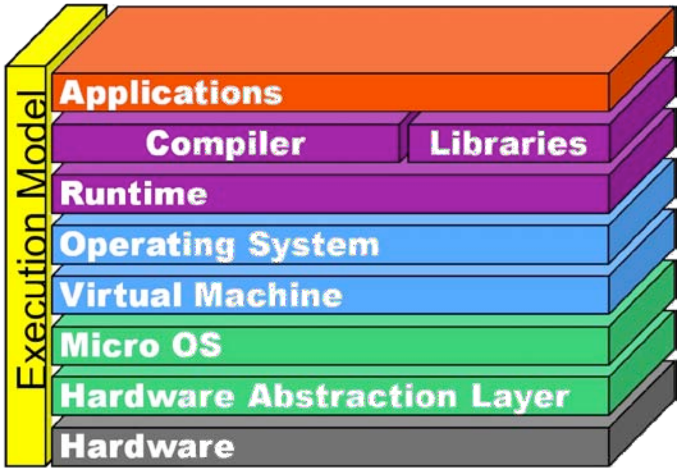
Latency Gap

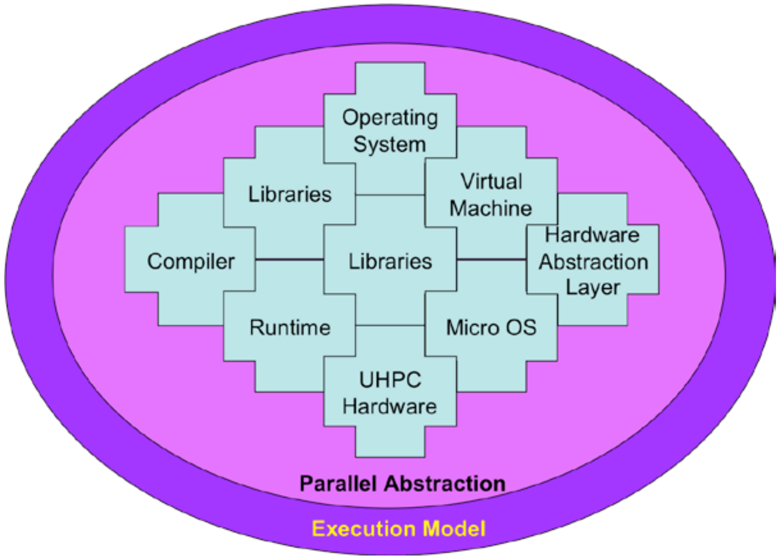
Disk, *O(TB)*
10,000,000 cycles

Source [2]

System software as currently implemented is not suitable for Exascale systems

- **Barriers**
  - System management SW not parallel
  - Current OS stack designed to manage only O(10) cores on node
  - Unprepared for industry shift to NVRAM
  - OS management of I/O has hit a wall
  - Not prepared for massive concurrency
- **Technical Focus Areas**
  - Design HPC OS to partition and manage node resources to support massively concurrency
  - I/O system to support on-chip NVRAM
  - Co-design messaging system with new hardware to achieve required message rates
- **Technical gaps**
  - 10X: in affordable I/O rates
  - 10X: in on-node message injection rates
  - 100X: in concurrency of on-chip messaging hardware/software
  - 10X: in OS resource management



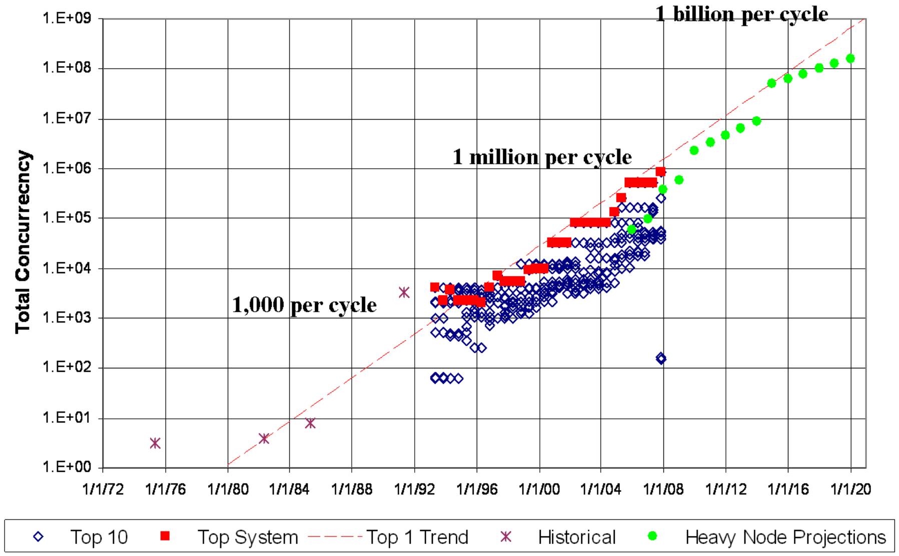


Software challenges in extreme scale systems, *Sarkar, 2010*

Source [7]

Programming Models and Environments

- **Barriers:** Delivering a large-scale scientific instrument that is productive and fast.
  - O(1B) way parallelism in Exascale system
    - Massive lightweight cores for low power
    - Some "full-feature" cores lead to heterogeneity
  - O(1K) way parallelism in a processor
    - Data and independent thread parallelism
  - Data movement costs power and time
    - Software-managed memory (local store)
  - Programming for resilience
  - Science goals require complex codes



How much parallelism must be handled by the program?

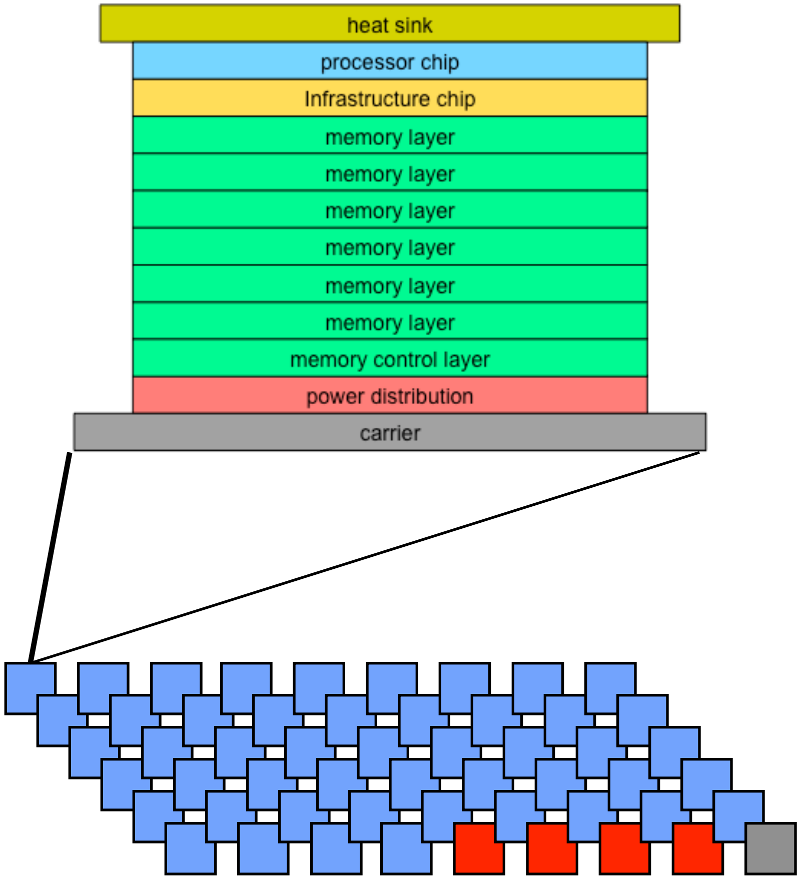
From Peter Kogge (on behalf of Exascale Working Group), "Architectural *Challenges* at the Exascale Frontier", June 20, 2008

- **Technical Focus Areas**
  - Extend existing between-chip models for scalability and resilience, e.g., MPI with support to hide hardware failures and low memory footprint
  - Develop on-chip models for 1K-way concurrency and heterogeneity by adapting current ones (e.g., OpenMP) or leverage models from other domains (e.g., CUDA or OpenCL)
  - Revolutionary: enable new software model for high concurrency across system scales
- **Technical Gap:** Productivity, performance and correctness for 1000x more parallelism on chip while increasing programming productivity of scientists by 10x

Source [7]

Programming Model Approaches

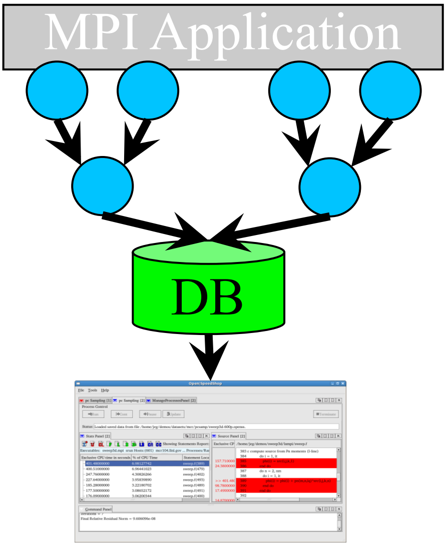
- Hierarchical approach (intra-node + inter-node)
  - [redacted] Part I: Inter-node model for communicating between nodes
    - MPI scaling to millions of nodes: Importance high; risk low
    - One-sided communication scaling: Importance medium; risk low
  - Part II: Intra-node model for on-chip concurrency
    - Overriding Risk: No single path for node architecture
    - OpenMP, Pthreads: High risk (may not be feasible with node architectures); high payoff (already in some applications)
    - New API, extended PGAS, or CUDA/OpenCL to handle hierarchies of memories and cores: Medium risk (reflects architecture directions); Medium payoff (reprogramming of node code)
- Unified approach: single high level model for entire system
  - High risk; high payoff for new codes, new application domains

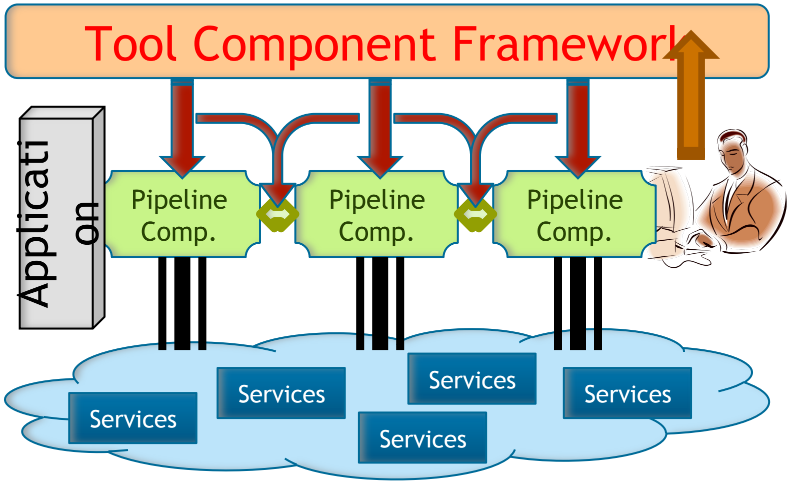


Source [7]

Tools

- **Barriers**
  - Increase in system sizes breaks current collection and analysis approaches
  - New primitives in new programming models not covered by existing tools
  - Current tools unable to correlate system and application data
  - Monolithic tools lack modularization needed for rapid adaptation
- **Technical Focus Areas**
  - **Evaluation and comprehension of node-level resources**
  - **Support for new/evolving programming models**
  - **Correlation between hardware, software, application events and data (including power, resiliency, memory usage, and performance)**
  - **Creation of tool infrastructures that allow quick tool prototyping for specific applications and systems**
  - **Techniques for root cause analyses to enhance performance and validate correctness**
- **Technical Gap**
  - Tool paradigms require a 1000x scalability increase to match applications and production systems and must evolve to reduce information overload



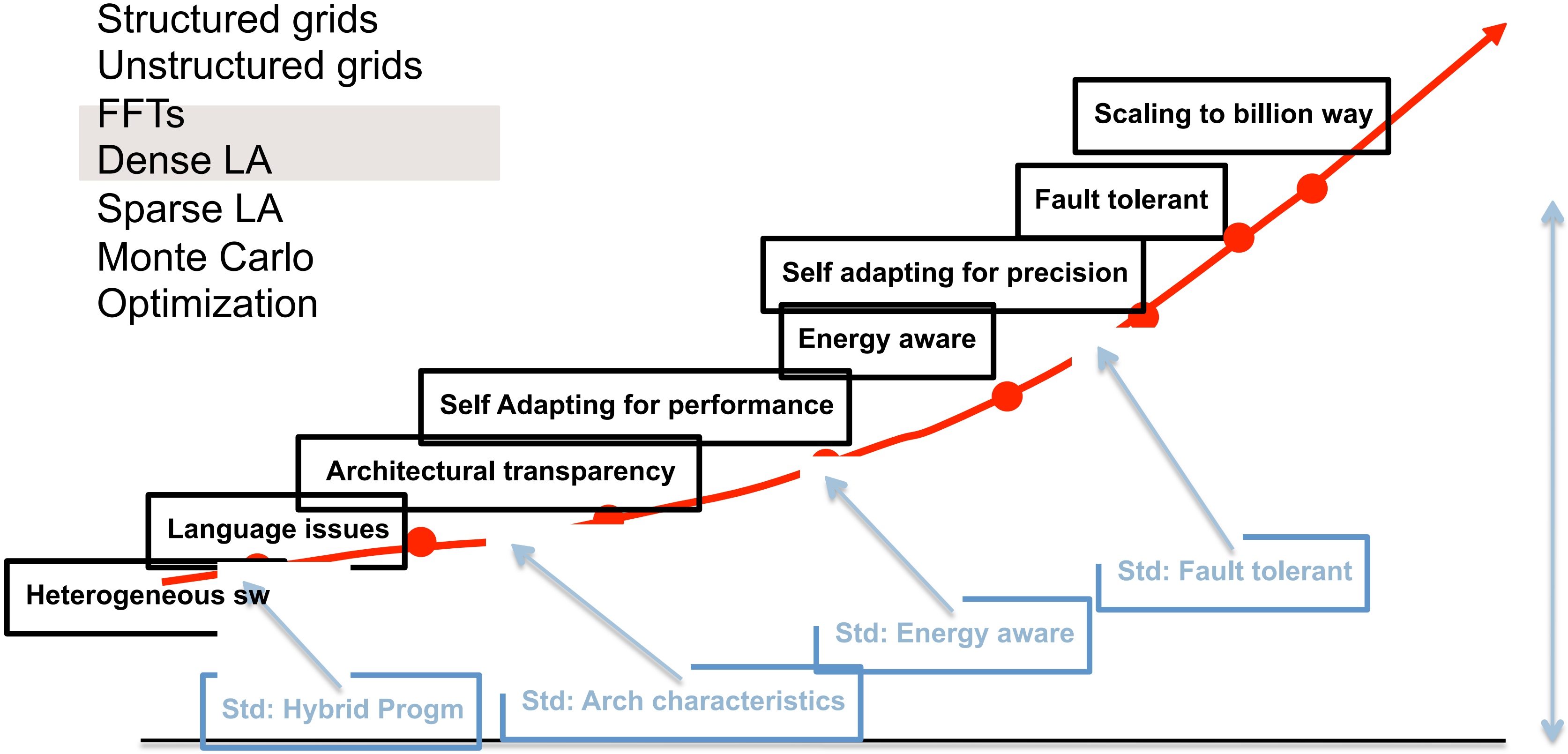


Source [2]

Numerical Libraries

Numerical Libraries

Structured grids
Unstructured grids
FFTs
Dense LA
Sparse LA
Monte Carlo
Optimization



Scaling to billion way

Fault tolerant

Self adapting for precision

Energy aware

Self Adapting for performance

Architectural transparency

Language issues

Heterogeneous sw

Std: Fault tolerant

Std: Energy aware

Std: Arch characteristics

Std: Hybrid Progm

Source [1]

Everything is Connected

Cross-cutting Issues

CS Software Layers And Issues

	Memory & Storage	Energy Effiency	Parallelism & Locality	Resilience	Scalability
Assumed HW Architecture(s)	X	X	X	X	X
System Software	X	X	X	X	X
I/O and Storage	X	X	X	X	X
Tools and Programming Models	X	X	X	X	X
Data analysis and visualization	X	X	X	X	X
Numerical Algorithms	X	X	X	X	X
Frameworks	X	X	X	X	X
Simulators and Models	X	X	X	X	X
Mini-apps	X	X	X	X	X

New Application Characteristics

- Locality, Locality, Locality!
- Billion Way Concurrency;
- Uncertainty Quantification (UQ) must also include hardware variability;
- Flops free - data movement expensive so:
  - Remap multiphysics to put as much work per location on same die;
  - Include embedded UQ to increase concurrency;
  - Include data analysis if you can for more concurrency
  - Trigger output to only move important data off machine;
  - Reformulate to trade flops for memory use.
- Wise use of silicon area

Source [3]

Key Message

- The transition from petascale to exascale will be characterized by significant and dramatic changes in hardware and software architectures.
- This transition will be disruptive, but create unprecedented opportunities for computer and computational science R&D.

Source [6]

References

All the authors below are from or supported by the DOE Office of Science [1,3-8] or National Nuclear Security Administration [2,7]. There is a wide sharing of slides in the exascale community and slides I credit to a particular talk often appear in others. Other references can be found at www.exascale.org.

1. Jack Dongarra, *The Road To Exascale: Hardware And Software Challenges* , SC-09 Exascale Panel, http://www.netlib.org/utk/people/JackDongarra/talks.htm

2. Sudip Dosanjh, *Exascale Computing and the Role of Codesign*, SOS 15, Engelberg, Switzerland March 14, 2011, *www.cscs.ch/fileadmin/SOS15_presentations/Sudip_Dosanjh.pdf*

3. Daniel Hitchcock, *The Challenge of Exascale and Exabyte and Where we are Now*, March 7, 2011, exascaleresearch.labworks.org/uploads/dataforms/PRES_ASCR_Challenge_110228-2.pdf

4. Lucy Nowell , *Data Analysis and Visualization at Exascale*, SC 10 – November 2010, *vis.cs.ucdavis.edu/Ultravis10/Slides/ScienceAtScale_Nowell.pdf*

5. John Shalf, *Exascale Computing Technology Challenges*, ScicomP / SP-XXL 16 San Francisco, May 12, 2010, http://www.spscicomp.org/ScicomP16/presentations/ExascaleChallenges.pdf

6. Horst Simon, *Exascale Challenges for the Computational Science Community*, Oklahoma Supercomputing Symposium 2010, October 6, 2010, symposium2010.oscer.ou.edu/oksupercompsymp2010_talk_simon_20101006.pdf

7. Rick Stevens and Andy White et al, *A decadal DOE plan for providing exascale applications and technologies for DOE mission needs, www.er.doe.gov/ascr/ascac/Meetings/Mar10/AWhite.pdf*

8. Rick Stevens, *Technology and Architecture for Future Large-Scale Computing Systems,* International Exascale Software Project, Santa Fe NM, April 7-8, 2009, http://www.exascale.org/iesp/IESP:Documents