Exascale:
Applications, Architectures y Codesign

Andy White

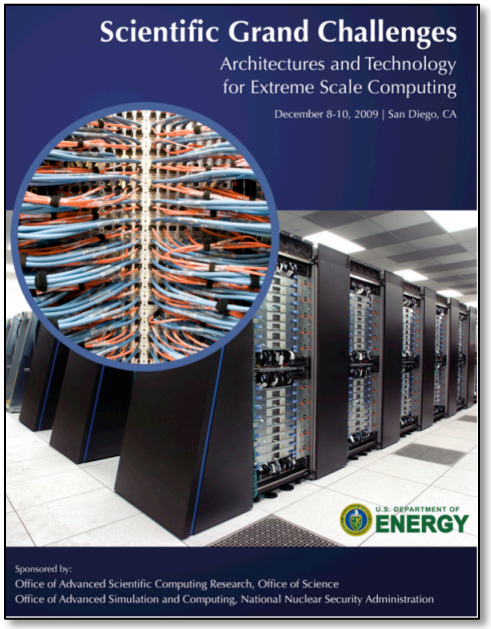
Los Alamos National Laboratory

T = 39,134

Exascale applications and technology have been identified by community.

- Town Hall Meetings April-June 2007
- Scientific Grand Challenges Workshops Nov, 2008 – Oct, 2009
  - Climate Science (11/08),
  - High Energy Physics (12/08),
  - Nuclear Physics (1/09),
  - Fusion Energy (3/09),
  - Nuclear Energy (5/09),
  - Biology (8/09),
  - Material Science and Chemistry (8/09),
  - National Security (10/09)
  - Architecture & Technology (12/2009)
  - Cross-cutting technologies (2/10)
- Exascale Steering Committee
  - "Denver" vendor NDA visits 8/2009
  - SC09 vendor feedback meetings
  - …
- International Exascale Software Project

- Architectures and Technology for Extreme Scale Computing
  - Collaboration and co-design
  - Focus on node software and hardware architecture
  - Managing greater than 1 billion way parallelism
  - Managing errors



Science Partnership for Extreme-scale Computing

Co-design is the key ingredient of the exascale initiative.

Uncertainty Quantification

DUAL

- Application #1: Codes, algorithms, models, theory
- Application #2: Codes, algorithms, models, theory
- Application #3: Codes, algorithms, models, theory
- . . .

May be both classified and unclassified

COMMUNITY

Applied mathematics:
Solvers, grids & meshes, PDEs, multi-scale, multi-physics, …

Computer Science:
Programming models, debuggers, performance, OS, file system, …

PROPRIETARY

Focused technology R&D (e.g. memory, optics, storage, …)

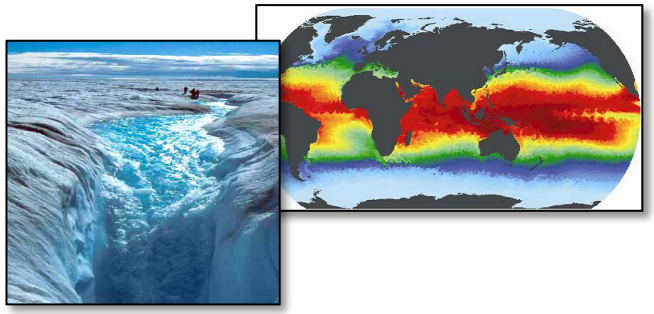
Laboratory-Industry Partnership #1
Integrated technology R&D
System acquisition & deployment

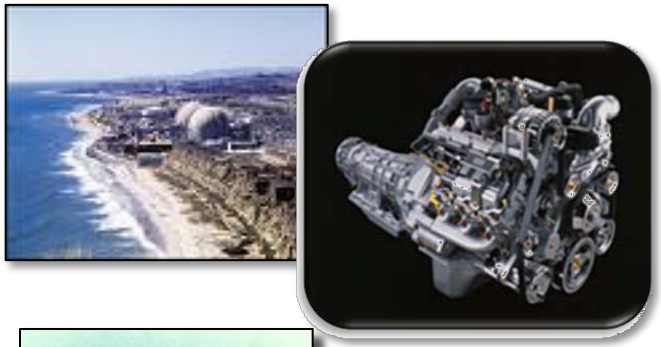
. . .

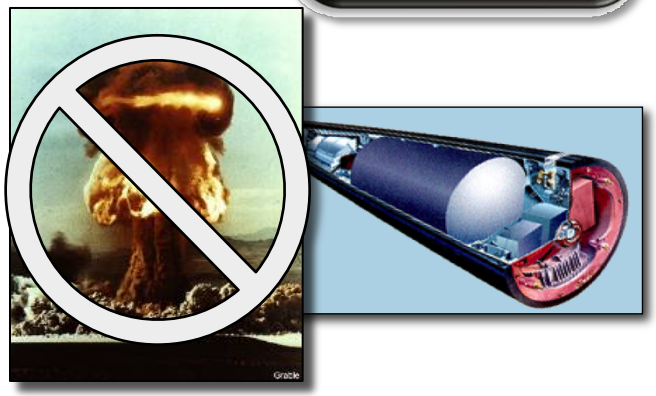
Co-design

Success is clearly defined for the exascale initiative.

- **Success of the initiative is:**
  - **Transformational capabilities in national nuclear security, climate, energy and science enabled by predictive exascale simulations**
  - **U.S. industry leadership in information technology lead by aggressive exascale technology development**
  - **Competitive advantage for U.S. energy-related and other industries**
- **Co-design of applications, computational environment and platforms is critical**
  - **Application teams must have dual responsibility**
  - **Simulation environment will**
    - **Be common across all applications and platforms**
    - **Leverage open source software and product support**
  - **Long term industry partnerships are essential to success of this 10 year initiative**
    - **Must leverage and influence the business plan of vendor partners**
    - **Joint R&D and leveraged community efforts reduce risk**







You can run but you can't hide.

Joseph Louis Barrow

- **System power** is a first class constraint on exascale system performance, effectiveness and TCO.
- **Exascale processor** will have an 100 – 1000x increase in parallelism, design is critical to meet power, performance, price, productivity and predictive goals.
- **System memory** is an important component of meeting exascale power (bandwidth) and applications (storage) goals.
- **Programming model**. Existing programming models will not be effective on nodes developed over the next decade, whether exascale or not. Early investment is critical to provide applications effective access to 2015 system.
- **Reliability and resiliency are very difficult at this** scale and require new check-point restart implementation and better understanding of effects and management of errors.
- **Operating System redesign for exascale** is essentiall for node performance at scale and for efficient support of new programming models and run time systems.
- ***HPC co-design strategy and implementation*** *requires a set of a hierarchical performance models and simulators as well as commitment from apps, software and architecture communities.*

Swim lanes affect the number of threads that the system needs to support.

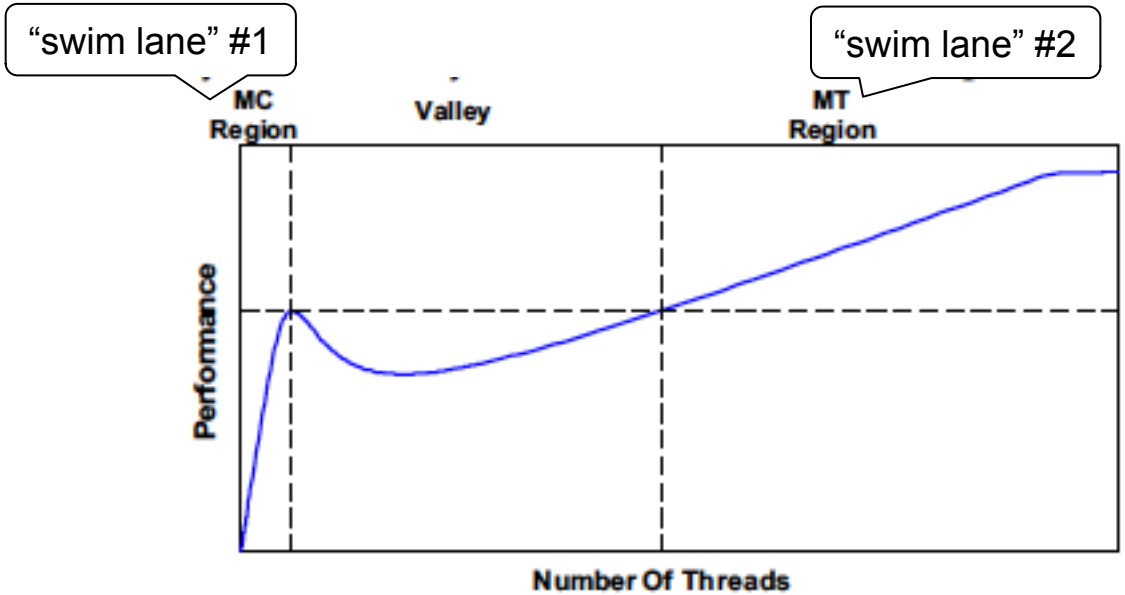
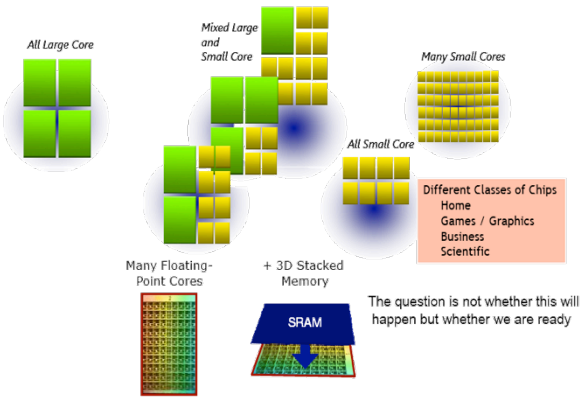


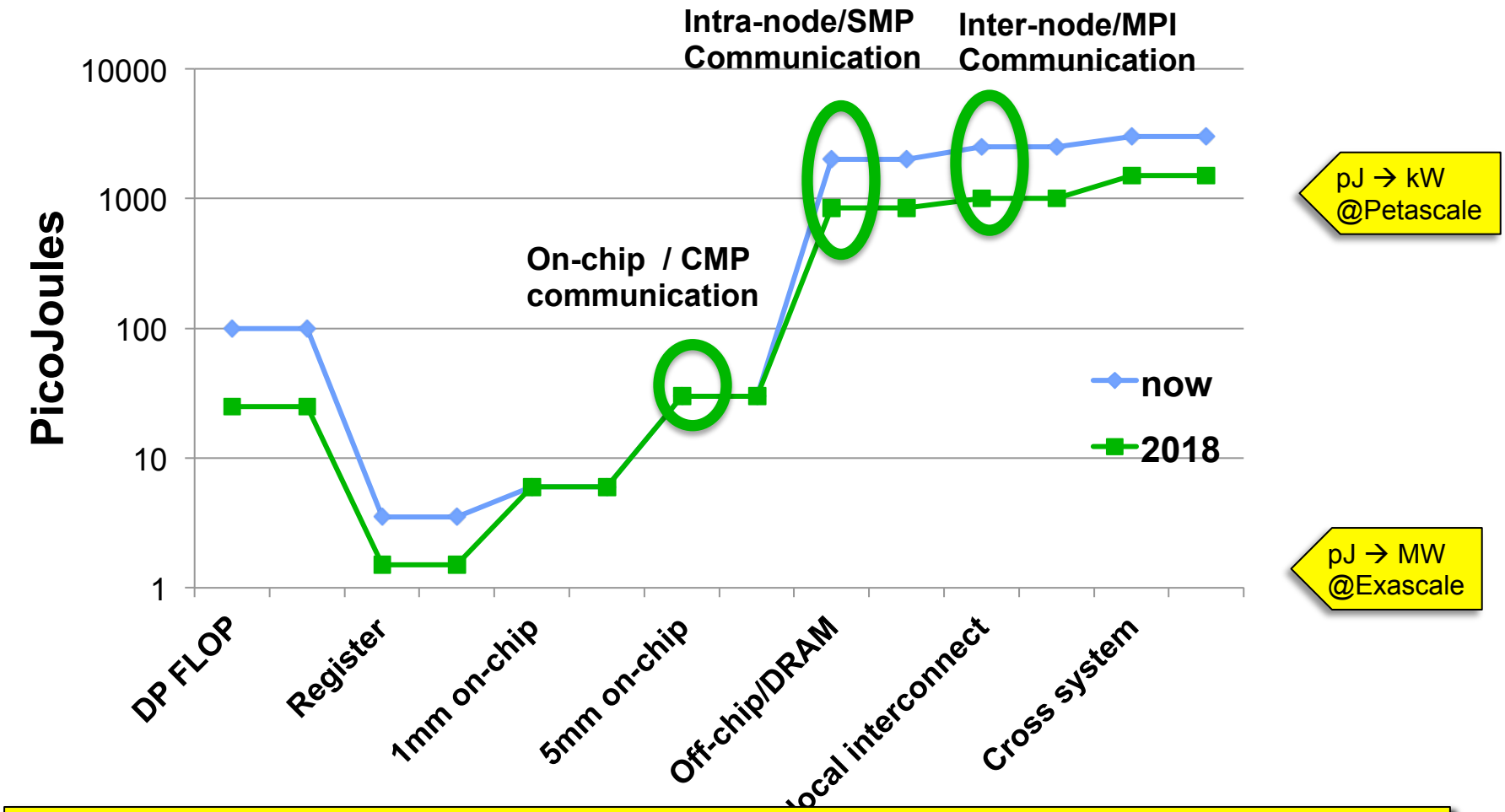
Fig. 1. Performance of a unified many-core (MC) many-thread (MT) machine exhibits three performance regions, depending on the number of threads in the workload.



There are currently two basic design points for achieving high performance in technical applications. In the future it is expected that these design points may (or may not) become more Integrated.

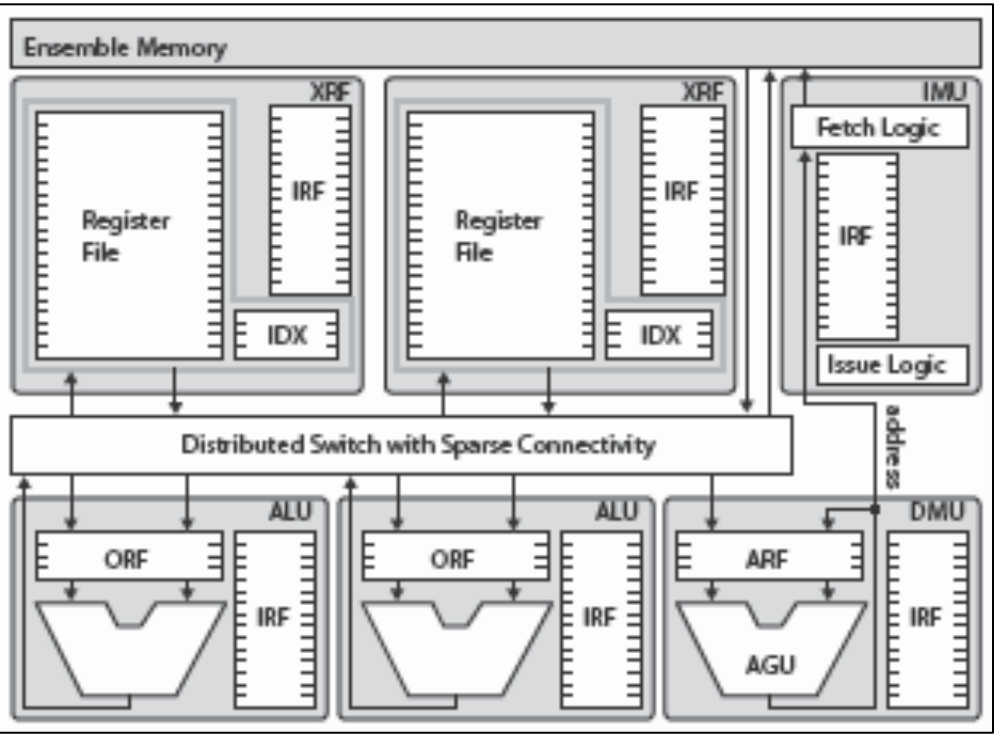
Many-core vs. many-thread machines: stay away from the valley, IEEE 2009

Investments in architecture R&D and application locality are critical.



"The Energy and Power Challenge is the most pervasive … and has its roots in the inability of the [study] group to project any combination of currently mature technologies that will deliver sufficiently powerful systems in any class at the desired levels."
DARPA IPTO exascale technology challenge report

Embedded processor example: it's about architecture and moving data.

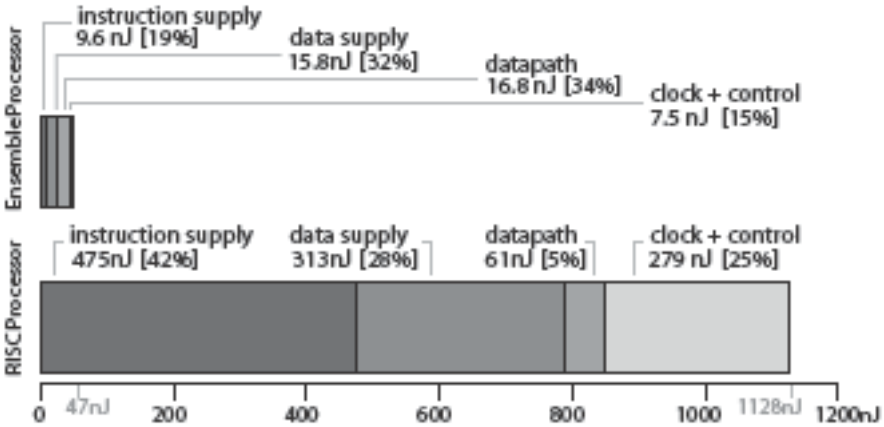


ENSEMBLE PROCESSOR

TABLE I

PROCESSOR CONFIGURATIONS AND DETAILS

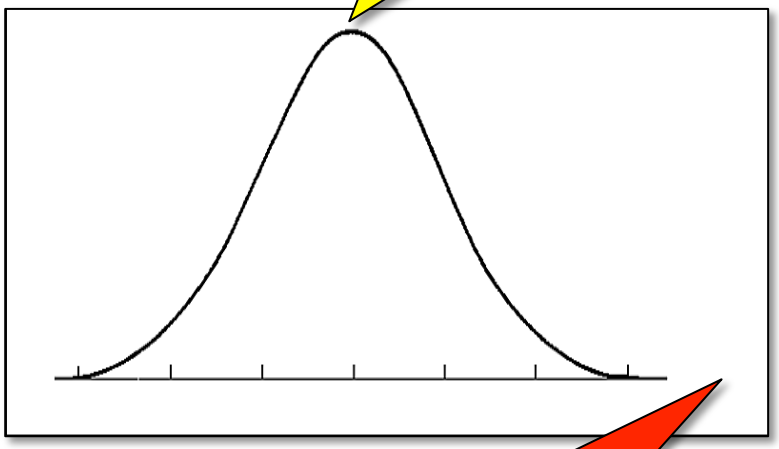
Ensemble Processor			
Technology	TSMC CL013G (V_{DD}=1.2V)		
Clock Frequency	200 MHz		
Average Power	28 mW		
Multipliers	16-bit + 40-bit acc.	16.5 pJ/op	
IRFs	64 128-bit registers	16 pJ/read	18 pJ/write
XRFs	32 32-bit registers	14 pJ/read	8.7 pJ/write
ORFs	8 32-bit registers	1.3 pJ/read	1.8 pJ/write
ARF	8 16-bit registers	1.1 pJ/read	1.6 pJ/write
Ensemble Memory	8KB	33 pJ/read	29 pJ/write
RISC Processor			
Technology	TSMC CL013G (V_{DD}=1.2V)		
Clock Frequency	200 MHz		
Average Power	72 mW		
Multiplier	16-bit + 40-bit acc.	16.5 pJ/op	
Register File	40 32-bit registers	17 pJ/read	22 pJ/write
Instruction Cache	8KB (2-way)	107 pJ/read	121 pJ/write
Data Cache	8KB (2-way)	131 pJ/read	121 pJ/write



An Energy-Efficient Processor Architecture for Embedded Systems, IEEE Computer Arch Letters, 2007

Factors affecting resilience @ exascale

Number of components
both memory and processors will increase mean time to failure, interrupt



Number of operations
ensure that system will sample the tails of the probability distributions

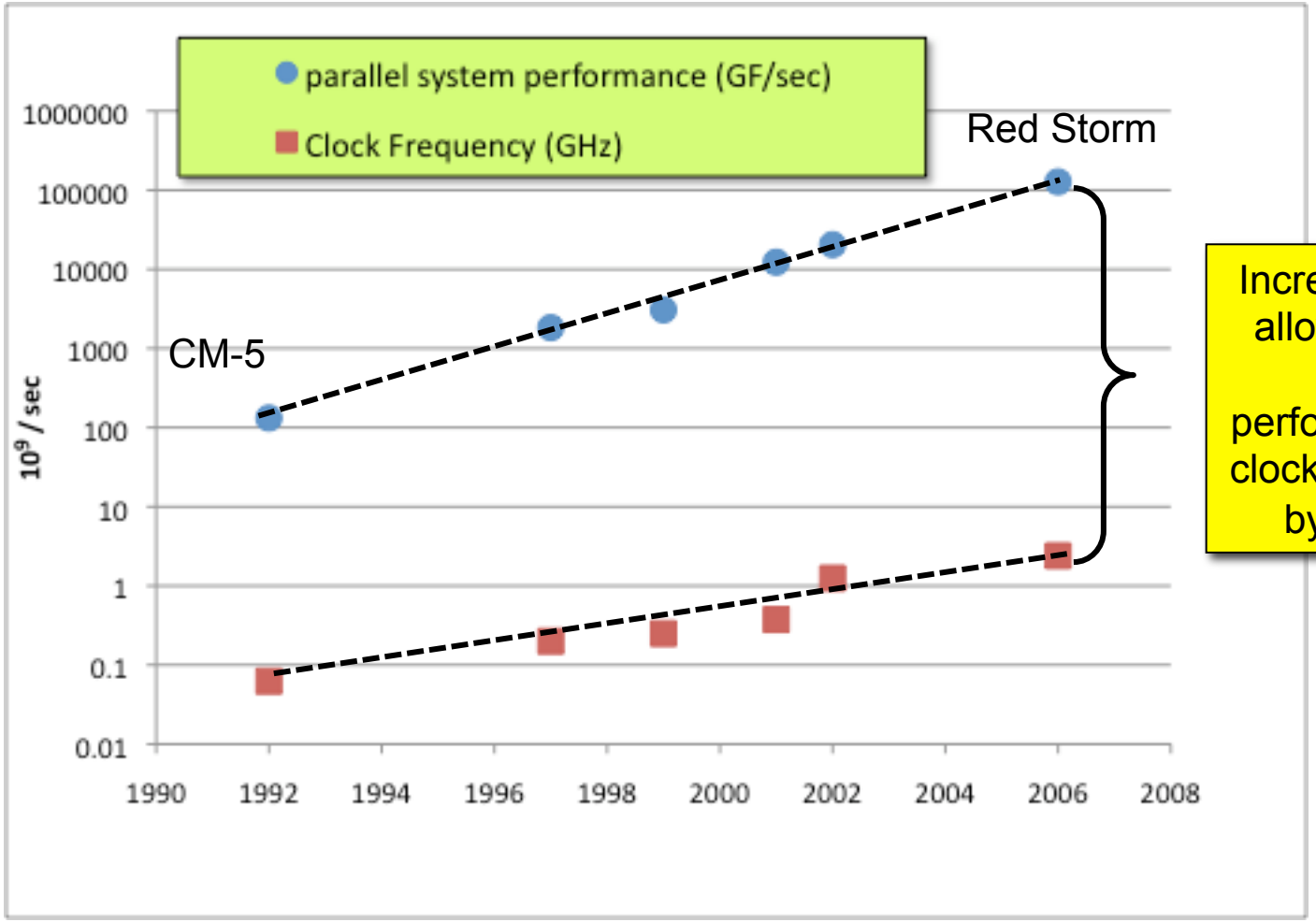
- **Smaller circuit sizes, running at lower voltages to reduce power consumption, increases the probability of errors**
- **Heterogeneous systems make error detection and recovery even harder, for example, error recovery on GPU system will require managing up to 100 threads**
- **Increasing system and algorithm complexity makes improper interaction of separate components more likely.**
- **In will cost power, performance and $ to add additional HW detection and recovery logic right on the chips to detect silent errors.**

	Transient	Persistent
Detected		
Undetected		

Uncertainty comes in a variety of shapes and sizes

	Parametric	Structural	Relational
Theory & models	•Calibrated parameters in models	•Unknown effects omitted from models •Extrapolation	•Multi-scale, multi-physics effects
Algorithms	•Discretization error	•Extrapolation	•Multiple time scales in operator split algorithms
Applications code	•Convergence criteria	•Errors in apps code	•Data mapping among different components
Computation and communication	•Rounding errors	•Silent data corruption	•Race conditions among separate components of system
Operating system & environment	•ECC error rates (chip bit errors)	•System parameters set incorrectly •Chip temperature excursions	•System policy mis-match (e.g. memory management)
Observations & data assimilation	•Statistical variation in experimental data	•Unknown systematic errors in data	•Contextual mismatch of observational and computational data

Concurrency is one key ingredient in getting to exaflop/sec

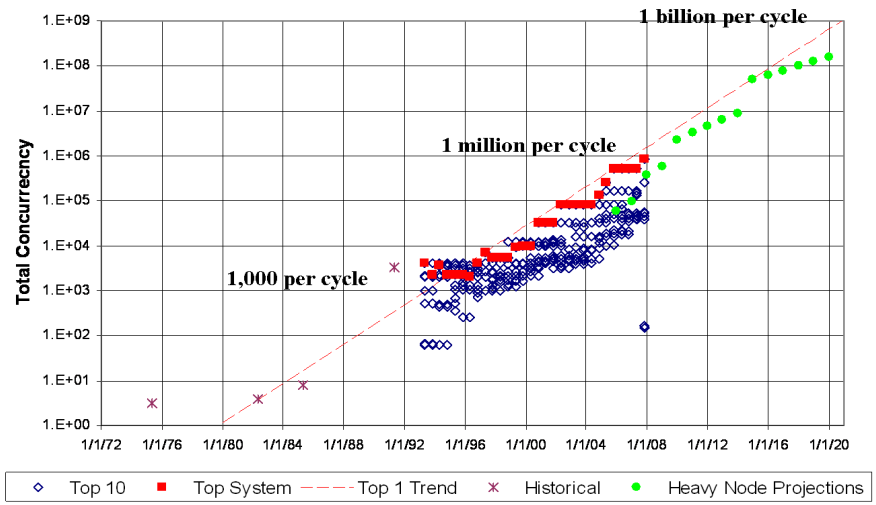


Increased parallelism allowed a 1000-fold increase in performance while the clock speed increased by a factor of 40

and power, resiliency, programming models, memory bandwidth, I/O, …

Programming models and environments require early investment.

- **Barriers:** **Delivering a large-scale scientific instrument that is productive and fast.**
  - **O(1B) way parallelism in Exascale system**
    - **Maybe 100B threads!**
  - **O(1K) way parallelism in a processor chip**
    - **Massive lightweight cores for low power**
    - **Some "full-feature" cores lead to heterogeneity**
  - **Data movement costs power and time**
    - **Software-managed memory (local store)**
  - **Programming for resilience**
  - **Science goals require complex codes**
- **Technology Investments**



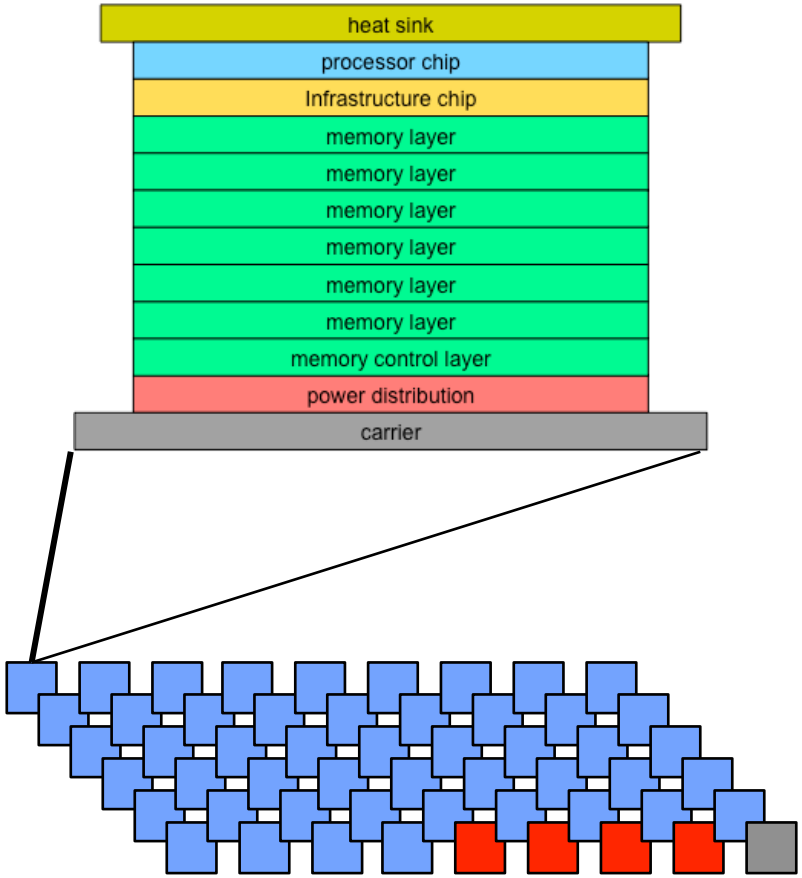
How much parallelism must be handled by the program?

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

  - **Extend inter-node models for scalability and resilience, e.g., MPI, PGAS (includes HPCS)**
  - **Develop intra-node models for concurrency, hierarchy, and heterogeneity by adapting current scientific ones (e.g., OpenMP) or leveraging from other domains (e.g., CUDA, OpenCL)**
  - **Develop common low level runtime for portability and to enable higher level models**
- **Technical Gap:**
  - **No portable model for variety of on-chip parallelism methods or new memory hierarchies**
  - **Goal: Hundreds of applications on the Exascale architecture; Tens running at scale**

Programming Model Approaches

- **Hierarchical approach: intra-node + inter-node**
  - **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**

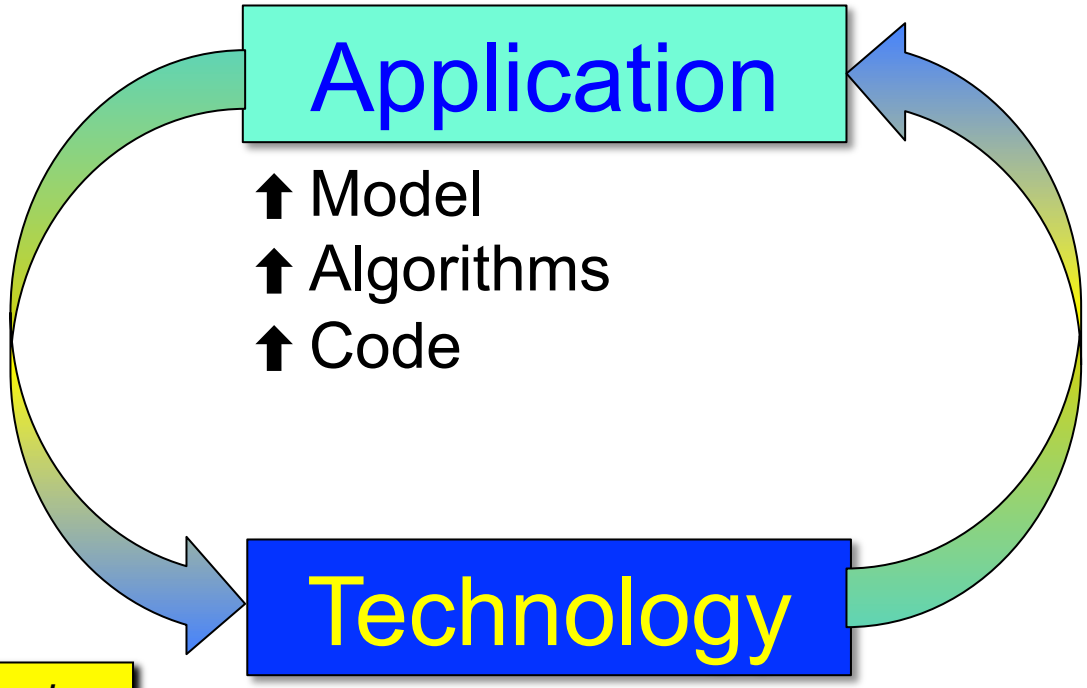


System architecture targets are aggressive in schedule and scope.

System attributes	2010	"2015"		"2018"	
System peak	2 PF/s	200 Petaflop/sec		**≥ 1 Exaflop/sec**	
Power	6 MW	15 MW		**≤ 20 MW**	
System memory	0.3 PB	5 PB		**64 PB**	
Node performance	125 GF/s	500 GF/s	5 TF/s	**1 TF/s**	**10 TF/s**
Node memory BW (consistent with 0.4 B/F)	25 GB/s	200 GB/s	2 TB/s	**400 GB/s**	**4 TB/s**
Node concurrency	12	100	1,000	**1,000**	**10,000**
System size (nodes)	18,700	400,000	40,000	**1,000,000**	**100,000**
Node link BW (consistent with 0.1 B/F)	1.5 GB/s	50 GB/sec	0.5 TB/sec	**100 GB/s**	**1 TB/sec**
Mean time before application failure	days	≥ 24 hours		**≥ 24 hours**	
IO	0.2 TB/s			**60 TB/s**	

Co-design is essential to manage complexity and optimize results

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

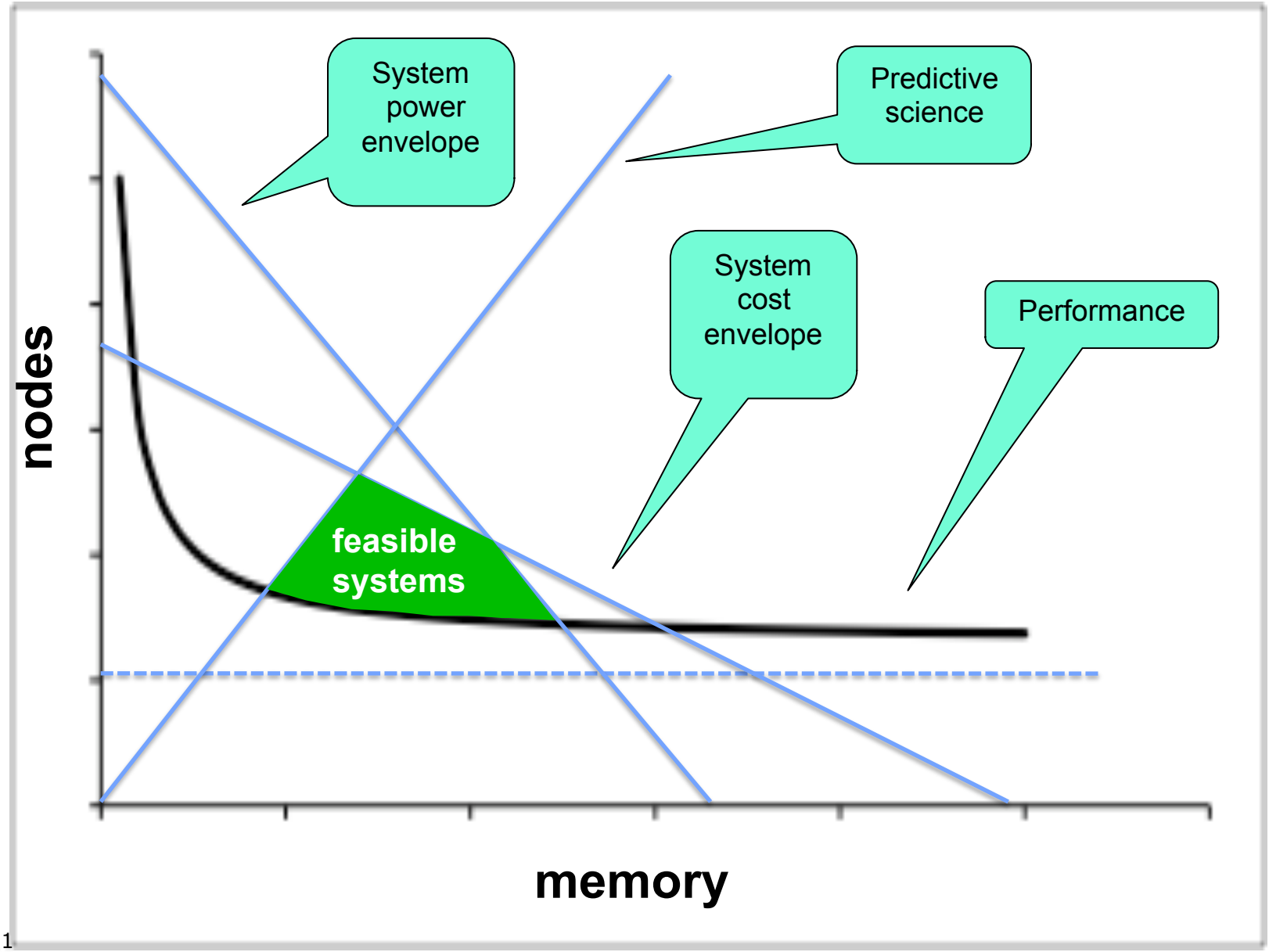
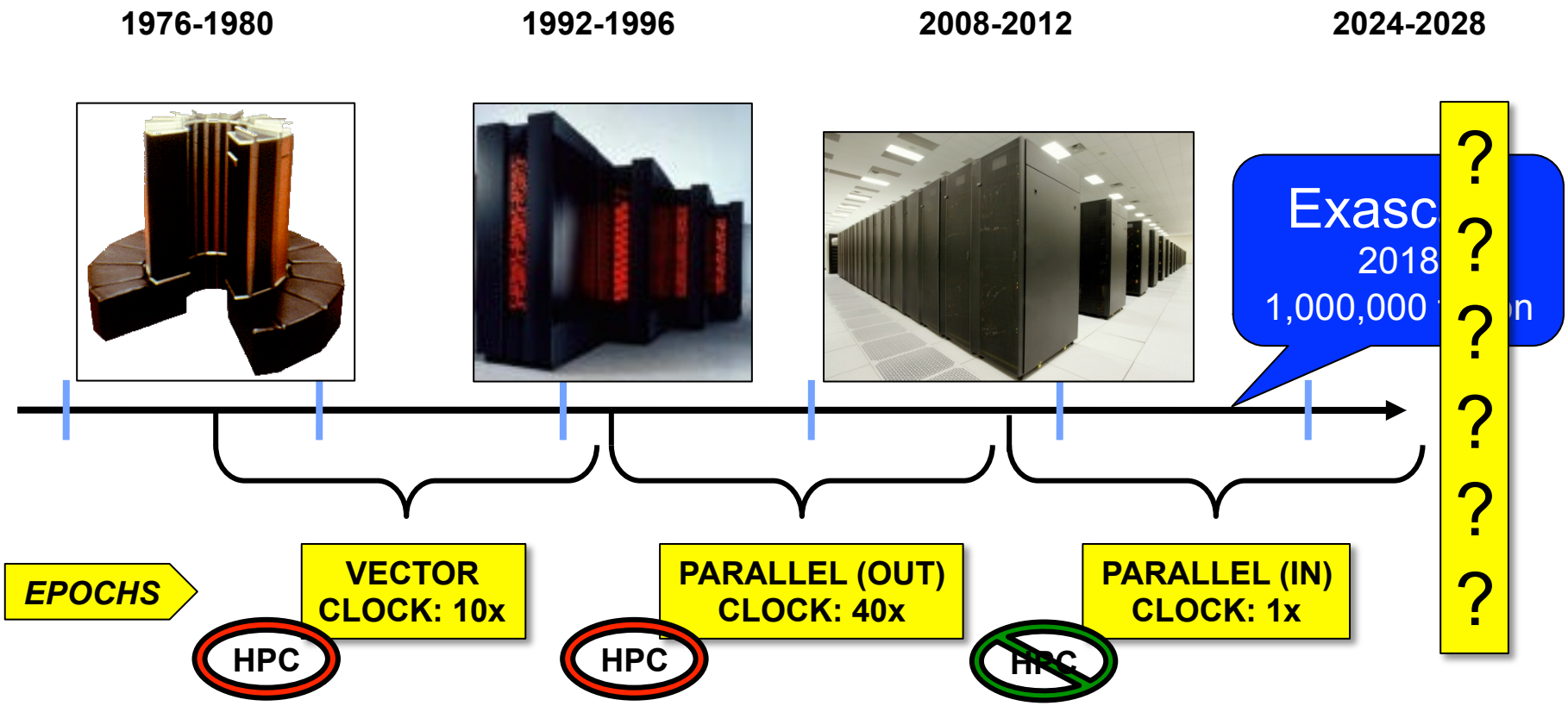


Key issues
Power?
Performance?
Price?
Parallelism?
Productivity?

Now, we must expand the co-design space to find better solutions:
- *new applications & algorithms,*
- *better technology and performance.*

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

The trade space for exascale is very complex.



The history of high performance computing is the history of DOE HPC.