Exascale Challenges: Applications, Technologies, and Co-Design

Andy White

Los Alamos National Laboratory

LA-UR 11-02200

Success is clearly defined for the exascale initiative.

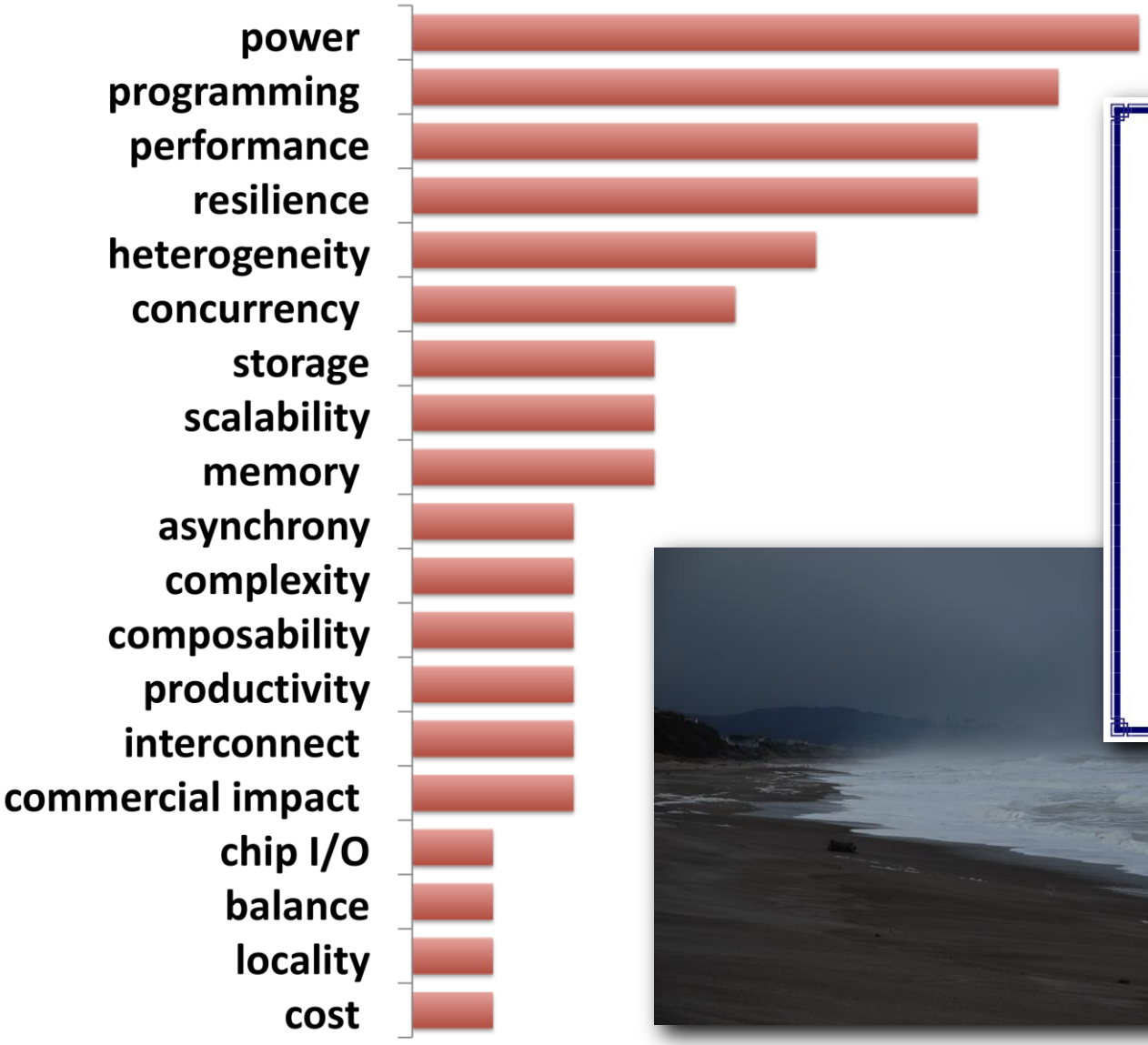
- **Success of the initiative is:**
  - **Transformational capabilities in national nuclear security, climate, energy and science enabled by predictive exascale simulations**
  - **U.S. industry leadership in information technology lead by aggressive exascale technology development**
  - **Competitive 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**

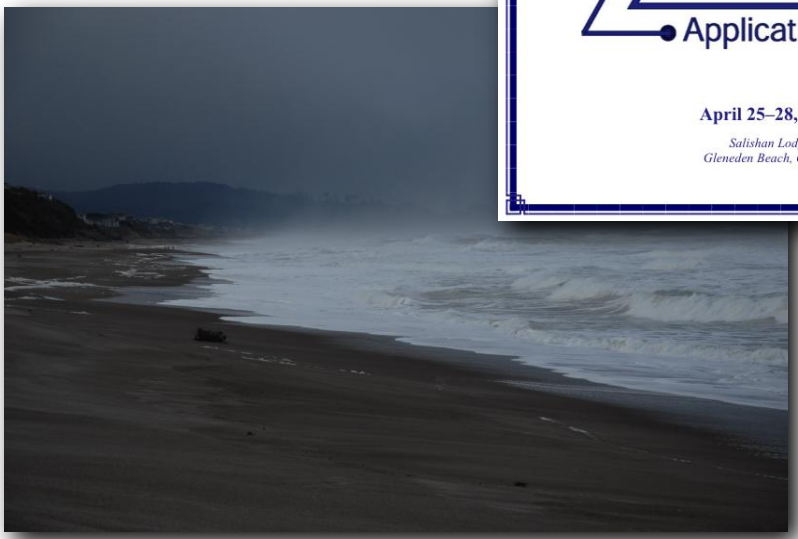
“You can run but you can’t hide.”

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 essential 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.*

A statistical view of the technical issues from Gleneden Beach, OR





System architecture targets are aggressive in schedule and scope.

System attributes	2010	"2		"2018"	
System peak	2 PF/s	200 P		≥ 1 Exaflop/sec	
Power	6 MW	1		≤ 20 MW	
System memory	0.3 PB			64 PB	
Node performance	125 GF/s	500 G		1 TF/s	10 TF/s
Node memory BW (consistent with 0.4 B/F)	25 GB/s	200 G		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		100 GB/s	1 TB/sec
Mean time before application failure	days			≥ 24 hours	
IO	0.2 TB/s			60 TB/s	

PEAK?
LINPACK?
AVERAGE?

WHAT DO YOU MEAN BY "MEMORY"?

WHAT DO YOU MEAN BY "FAILURE"?

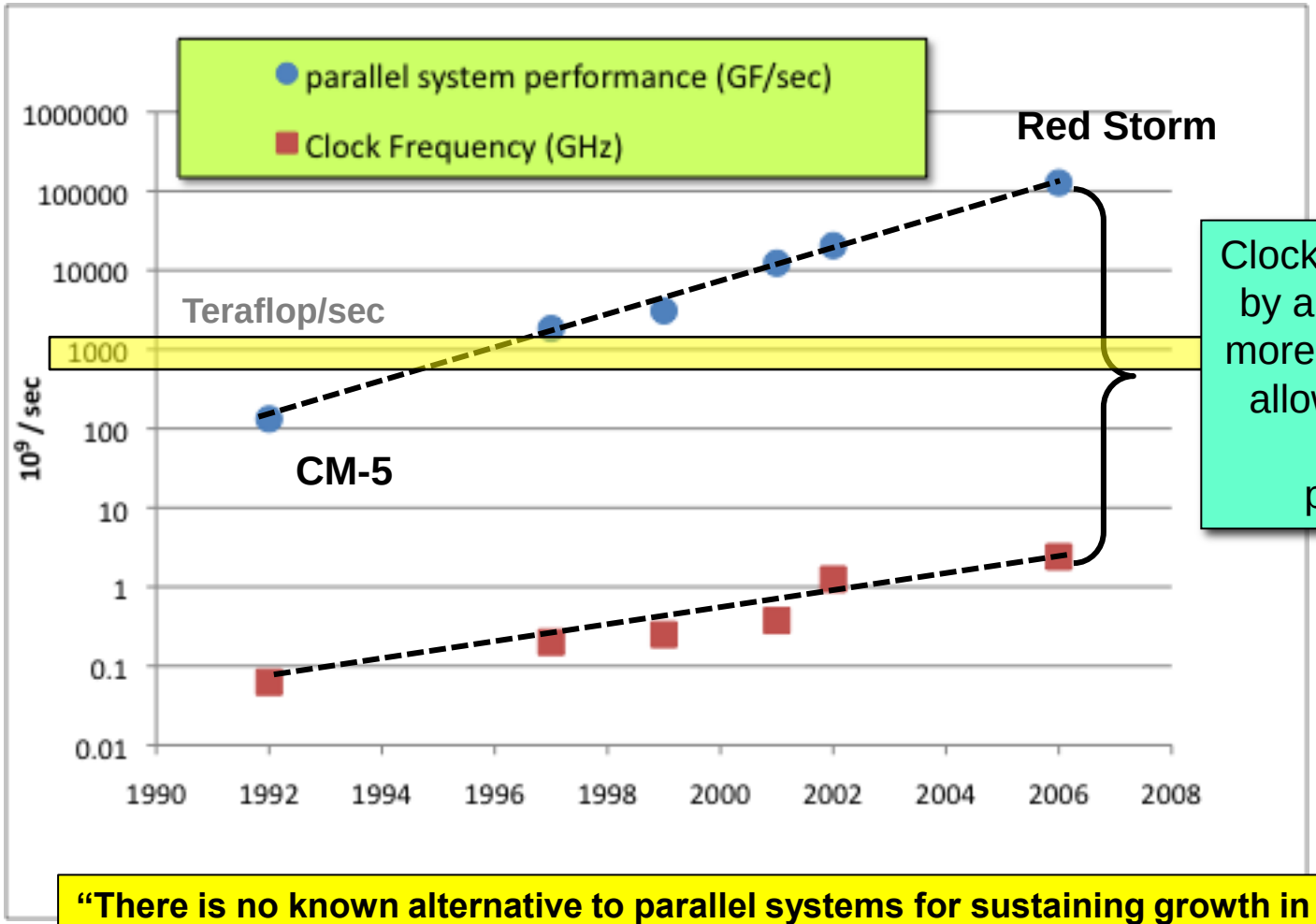
"The Future of Computing Performance: Game Over or Next Level?" NRC

"The U.S. Computing Industry has been adept at taking advantage of increases in computing performance, allowing the United States to be a moving and therefore elusive target – innovating and improvising faster than anyone else."

- Invest in research in and development of algorithms that can exploit parallel processing.
- Invest in research in and development of programming methods that will enable efficient use of parallel systems …
- Focus long-term efforts on rethinking of the canonical computing "stack" …
- Invest in research on and development of parallel architectures driven by applications, …
- Invest in research and development to make computer systems more power efficient at all levels of the system …

"There is no known alternative to parallel systems for sustaining growth in computing performance; however, no compelling programming paradigms for general parallel systems have yet emerged."

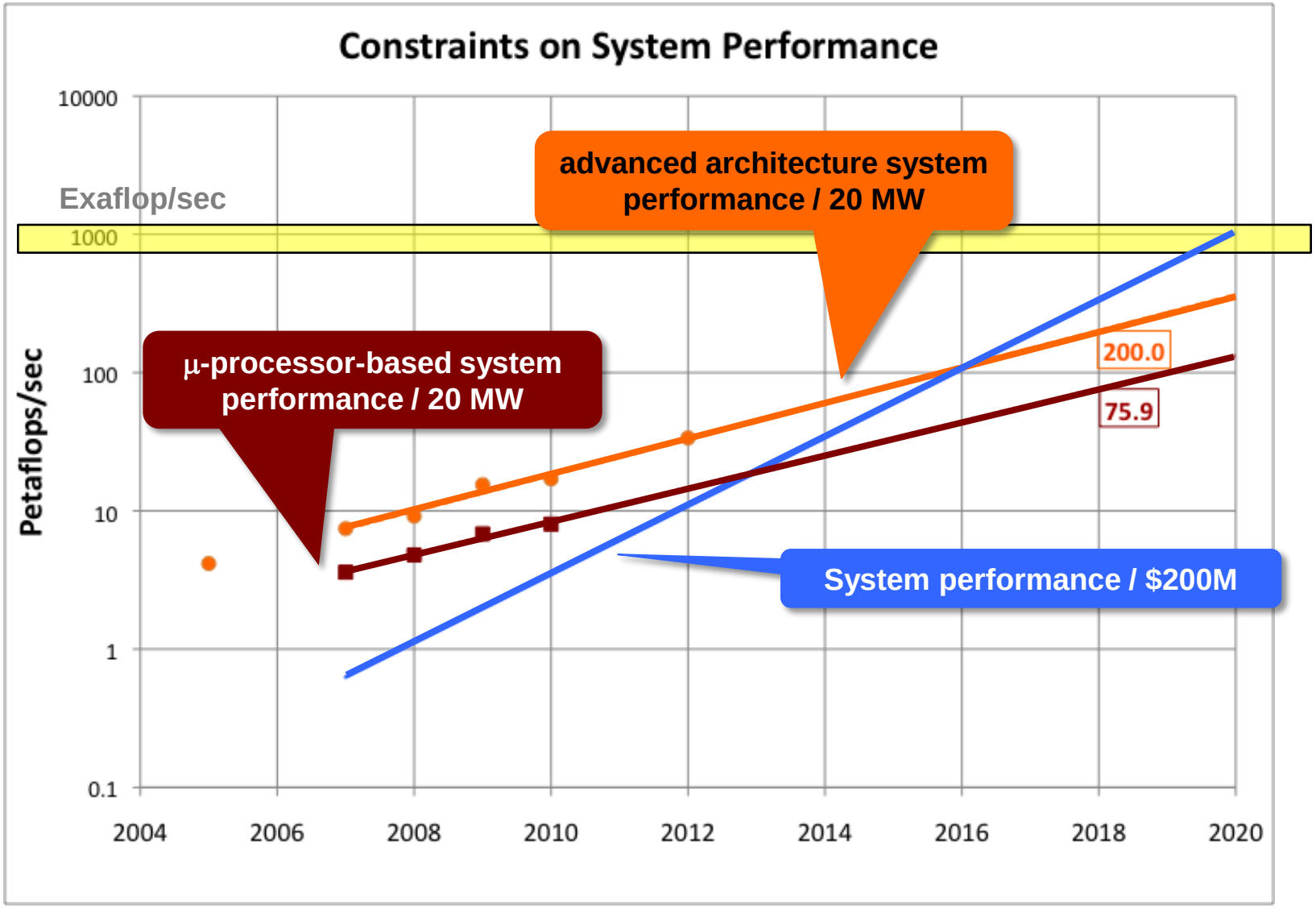
The last factor of 1000 was made possible by HPCC, ASCI and the marketplace.



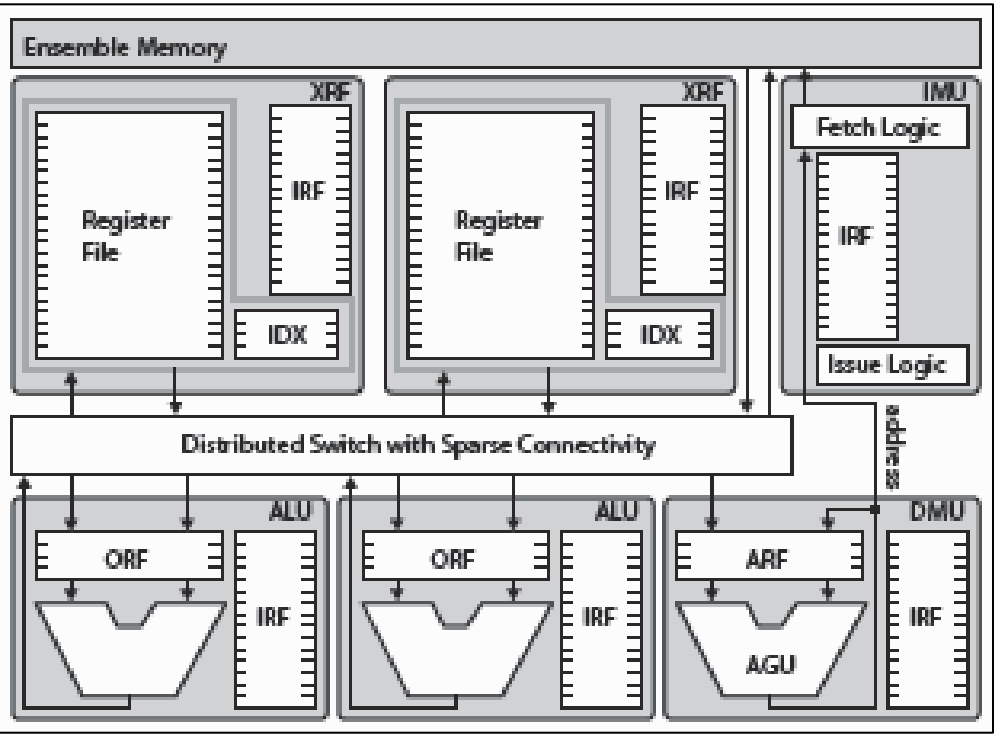
Clock speed increased by a factor of 40 and more parallelism (25x) allowed a 1000-fold increase in performance.

"There is no known alternative to parallel systems for sustaining growth in computing performance; however, no compelling programming paradigms for general parallel systems have yet emerged." NRC, 2011

Power will be a significant constraint on system performance.



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

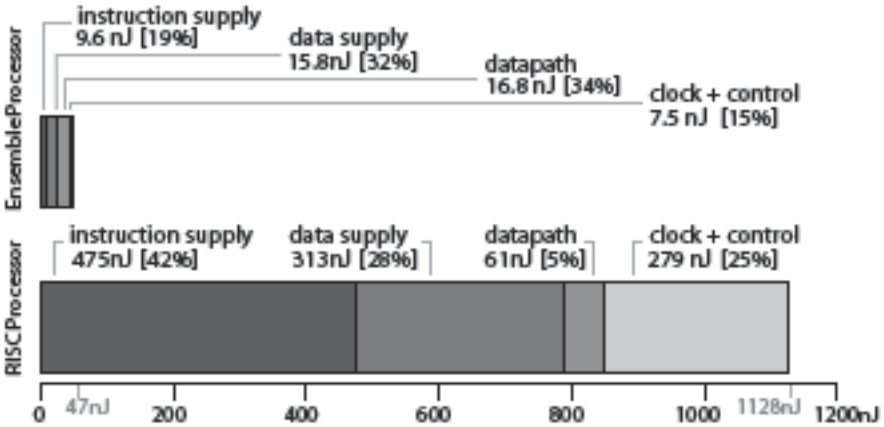


ENSEMBLE PROCESSOR

TABLE I

PROCESSOR CONFIGURATIONS AND DETAILS

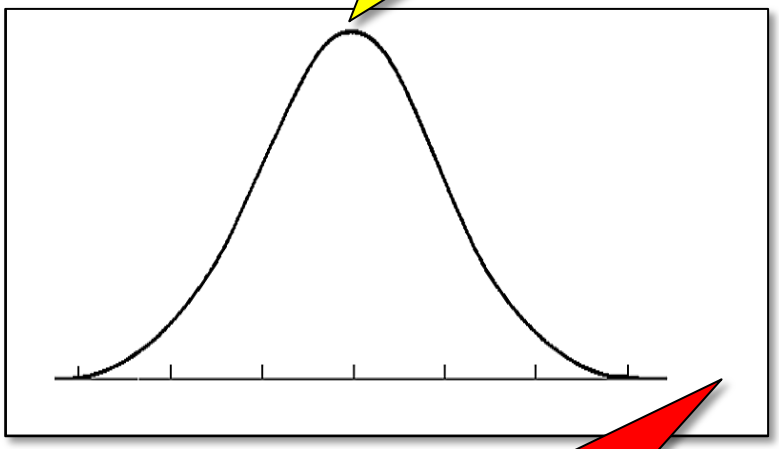
Ensemble Processor			
Technology	TSMC CL013G (V_{DD}=1.2V)		
Clock Frequency	200 MHz		
Average Power	28 mW		
Multipliers	16-bit + 40-bit acc.	16.5 pJ/op	
IRFs	64 128-bit registers	16 pJ/read	18 pJ/write
XRFs	32 32-bit registers	14 pJ/read	8.7 pJ/write
ORFs	8 32-bit registers	1.3 pJ/read	1.8 pJ/write
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

There are a number of factors affecting resilience @ exascale

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



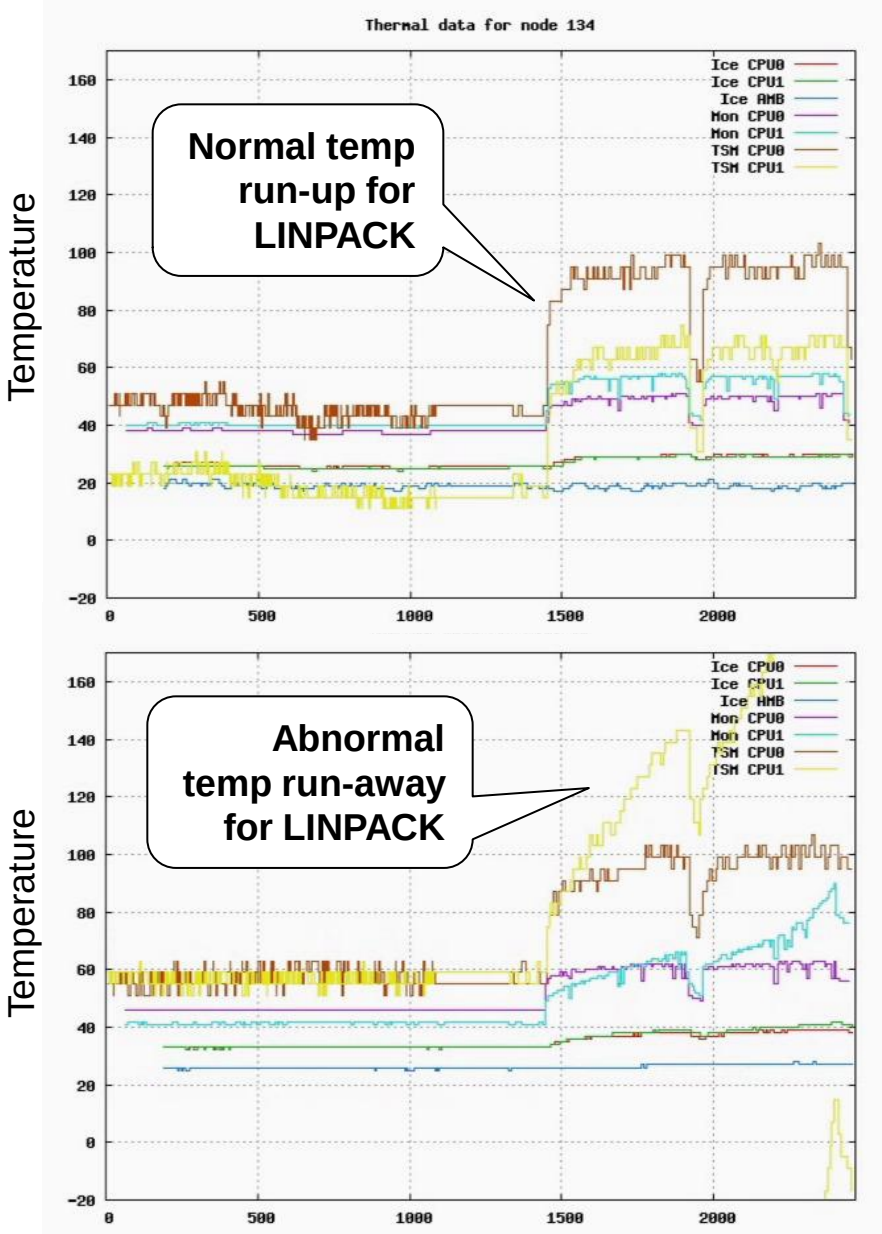
Number of operations
ensure that system will sample the tails of the probability distributions

- **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		

The rate and effect of undetected (aka silent) errors must be better understood.

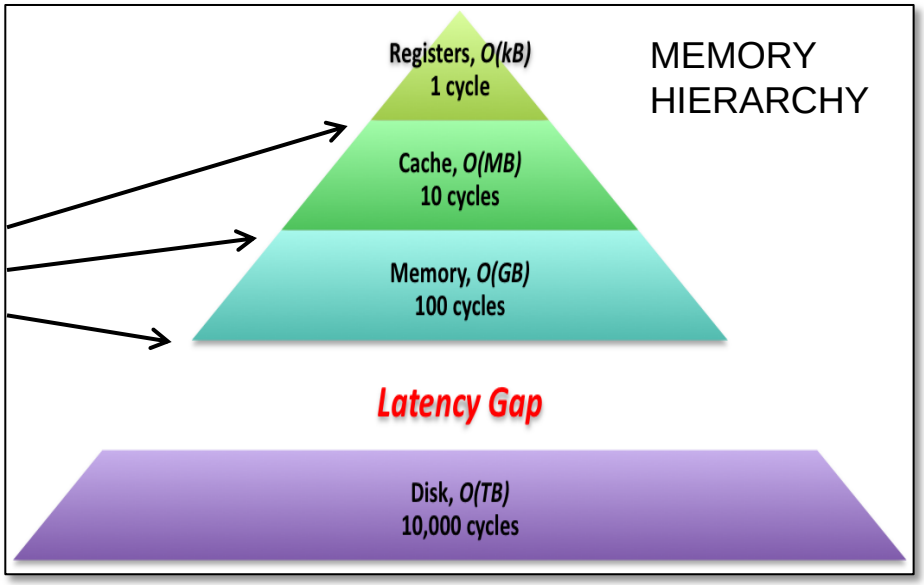
- During acceptance, RR experienced intermittent, but relatively frequent (20 microhertz) silent errors on HPL
- The issue was eventually tracked to an entire MPI transfer filled with zeroes
  - But data on the sending side was confirmed to be correct
- Root cause was a policy misunderstanding between
  - System: *when I move pinned memory, I will tell you*
  - MPI: *you won't move pinned memory, so I won't listen*

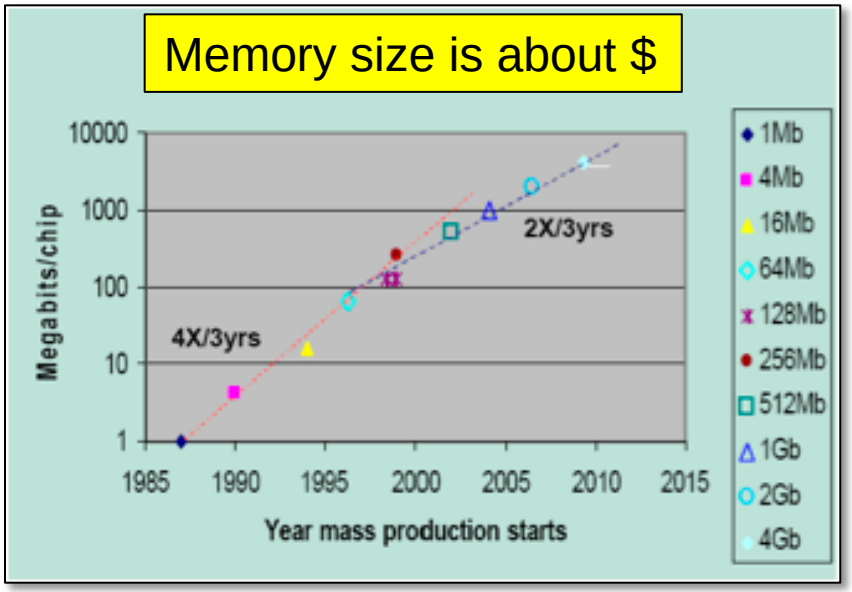


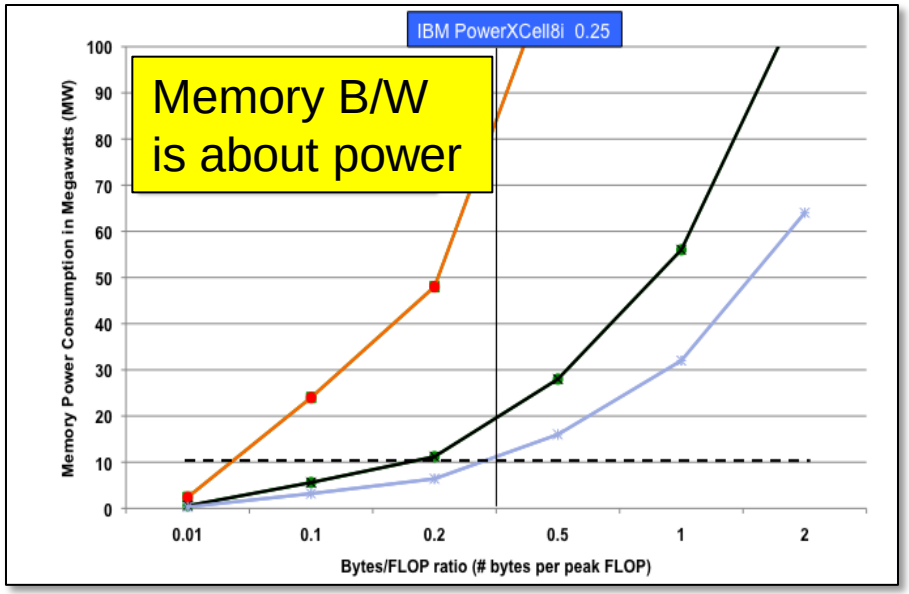
Memory size, bandwidth and hierarchy will be challenges by 2018, if not sooner.

The memory hierarchy will be much richer at the end of the decade than it is now:

- *Software managed caches or scratch pad memory*
- *Very fast 3D stacked memory*
- *NVRAM for check-pointing and extended memory*



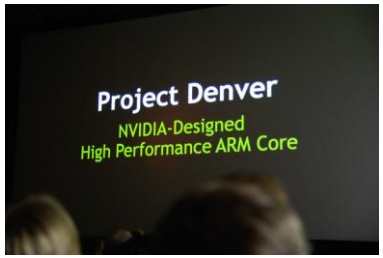




Architecture will be responsible for some problems and some solutions.

AMD
The future is fusion

AMD:
Delivering heterogeneous computing



NVIDIA:
ARM CPU integrated with GPU

INTEL
Many Integrated Core architecture

Use parallelism to increase performance

"swim lane" #1
many cores

"swim lane" #2
many threads

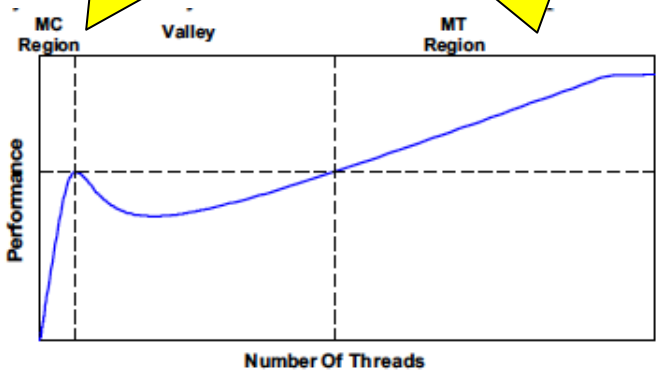
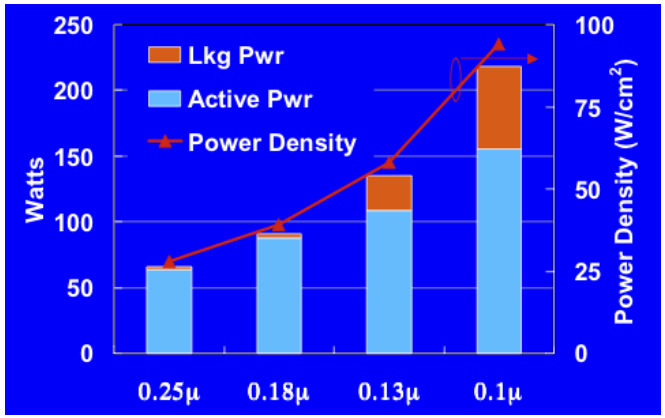


Fig. 1. Performance of a unified many-core (MC) many-thread (MT) machine exhibits three performance regions, depending on the number of threads in the workload.

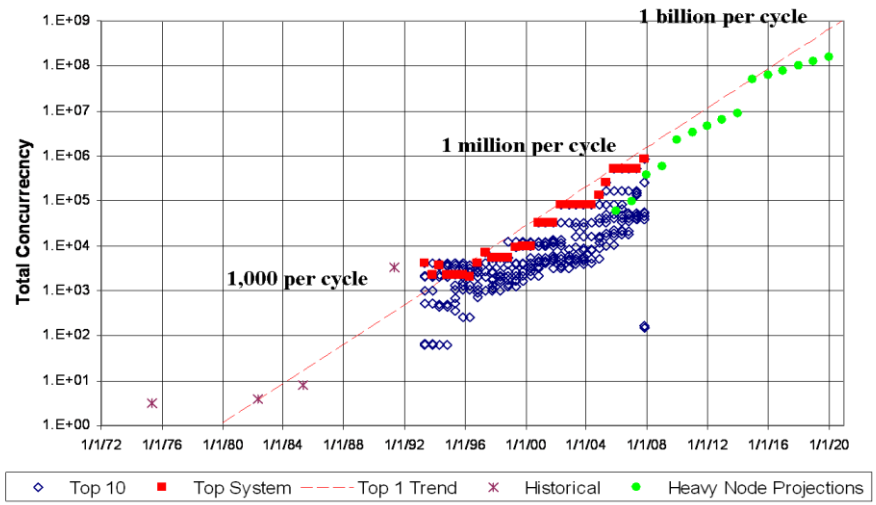
Manage on-chip power consumption



Chip power density = # gates * gate capacitance * frequency * voltage^2

Programming models and environments require early investment. Already too late?

- **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

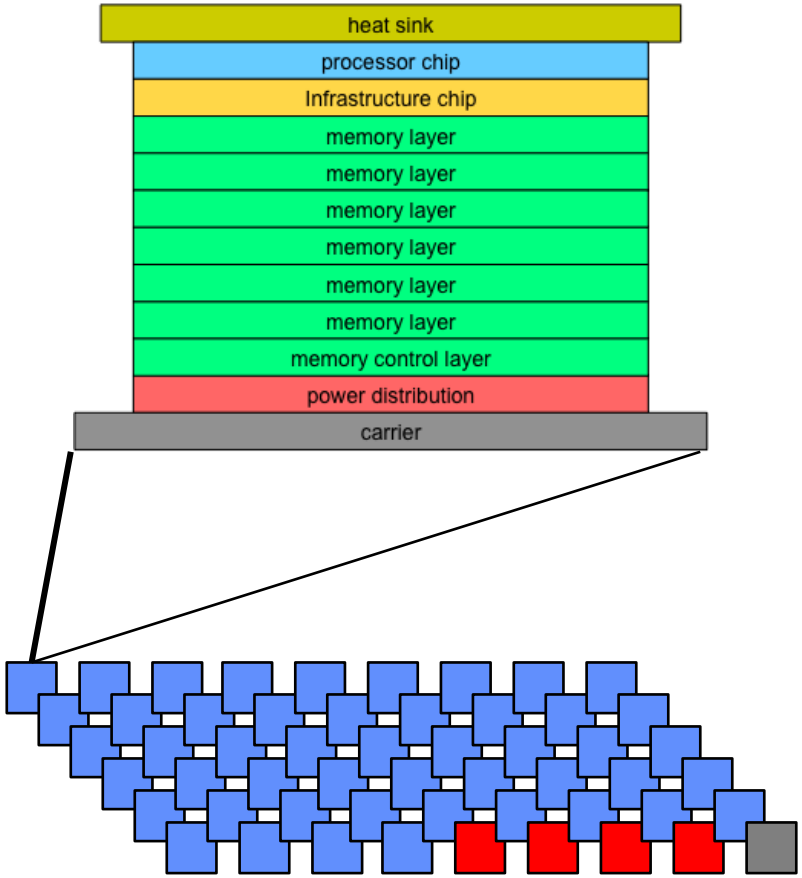


How much parallelism must be handled by the program?
From Peter Kogge (on behalf of Exascale Working Group), "Architectural *Challenges* at the Exascale Frontier", June 20, 2008

- **Technology Investments**
  - 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

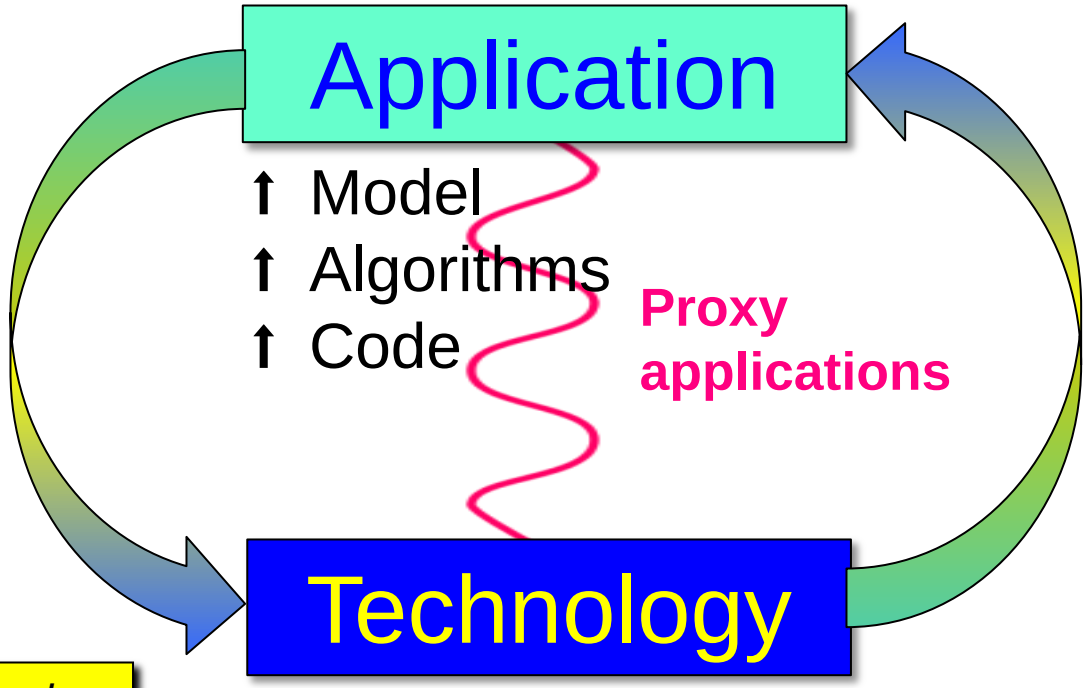
Programming models requires a dual approach.

- **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**



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?

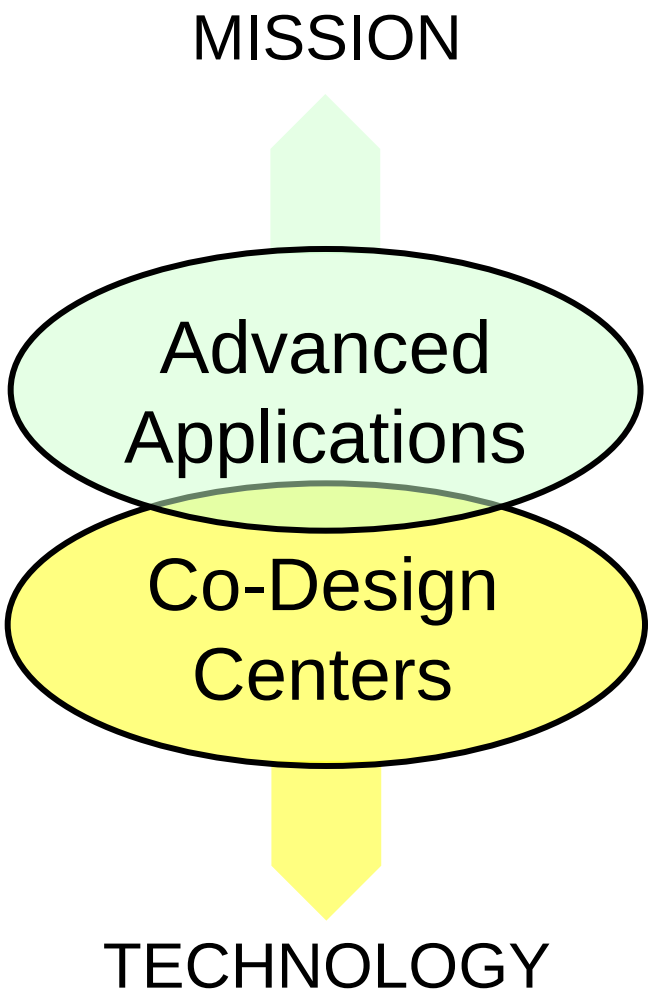
⊕ programming model
⊕ operating system
⊕ architecture

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.*

ASC and ASCR exascale responsibilities are not bijective.

- **Co-design and Advanced Applications are distinct activities**
  - **<u>Co-design</u>: commensurate with the notion of co-design centers. Focus on {mini, proxy, skeleton} apps to facilitate co-design among apps, software and hardware technology efforts**
  - **<u>Advanced Applications</u>: have no ASCR counterpart. Focus on advanced applications linking the co-design effort with ASC mission space.**



Applications and predictive science must transform with the technology.

- **Power will be the number one architectural constraint**
  - **Applications will be effected by power efficient architectures**
  - **Applications may be directly involved in managing system power**
  - **Load balancing will have a new dimension**
- **On-chip: ten thousand way parallelism, deeper/higher memory hierarchies, 100x more upsets/sec**
  - **New programming models, languages and run-time systems**
  - **Fault-aware applications and fault-tolerant algorithms**
- **Cheap flops, expensive data motion, very expensive I/O**
  - **Remap multi-physics and algorithms to maximize data reuse and locality**
  - **Data analysis on-the-fly and embedded UQ**
  - **Reformulate algorithms to trade flops for memory use**

… but this is not just a challenge; it is also an opportunity to transform our capability to do predictive science and engineering.