



On the Path to Exascale

Sudip Dosanjh

**Co-Director, Science Partnership for Extreme-scale Computing
Co-Director, The Alliance for Computing at the Extreme Scale**

Sandia National Laboratories

**Presentation at
Advanced HPC Systems
Cetraro, Italy
June 29, 2011**



Current Status of the Exascale Effort

- **\$126 M in the President's FY12 budget (\$90 M in Office of Science, \$36 M in NNSA)**
- **Office of Science's ASCR has funded projects in**
 - **Advanced Architectures and Critical Technologies for Exascale**
 - **Scientific Data Management and Analysis at Extreme Scale**
 - **X-stack Software Research**
- **ASCR co-design centers are being planned**
- **ASC is identifying Exascale co-design centers and is developing an Exascale program plan**
- **ASCR kickoff meeting was in San Diego (early March), ASC Exascale workshop was in San Francisco (late March)**
- **7 DOE laboratories are working together on an Exascale RFI**
- **Expect that two laboratory/industry teams will be funded**
 - **Science Partnership for Extreme-scale Computing or SPEC (LANL, ORNL, Sandia)**
 - **Argonne, LBNL, LLNL, PNNL**



The Science Partnership for Extreme-scale Computing (SPEC) builds on previous collaborations

- **The Los Alamos/Sandia Alliance for Computing at the Extreme Scale**
 - ACES is deploying the Cielo Petascale capability platform for NNSA
 - MPI-only codes must run well
 - 1.33 PF Cray system with Gemini interconnect
 - 2 GB/core, ~130,000 cores
 - Panasas file system
- **Oak Ridge and Sandia have collaborated since the mid 90s**
 - Intel Paragon
 - Collaboration with Cray, Red Storm led to Jaguar
 - Institute for advanced Architectures and Algorithms (IAA)
 - ASCR CS/math institute
- **The Oak Ridge/Los Alamos Hybrid Multicore Consortium (HMC)**



SPEC has been very been very active since its inception

- **Initial meeting at SOS14 in March, 2010**
- **Weekly Tri-lab telecons**
- **Four way NDAs signed with 7 companies**
- **MOU signed by laboratory directors – November, 2010**
 - **Co-directors are Jeff Nichols, Andy White and Sudip Dosanjh**
- **Numerous meetings with potential industry partners**
 - **>30 meetings with computer companies (dozens of SPEC-industry telecons as well)**
- **Defining a SPEC technology roadmap that will advance the HPC ecosystem**
- **SPEC co-design effort on climate modeling**

Exascale Strategy

- **Create viable Exascale industry partnerships that advance the HPC ecosystem**
- **Build a broad coalition of support**
- **Identify cross-cutting issues and technologies (e.g., memory, silicon-photonics, programming models, file systems)**
- **Use competition to identify the best technical solutions**
- **Develop mechanisms to enable co-design (includes technical and IP considerations)**

Industry discussion points include:

- **Pre-Exascale systems must be representative of the Exascale systems**
 - **Programming continuity (i.e., no revolutionary programming change between pre-Exascale and Exascale systems)**
- **Constraints**
 - **1 EF**
 - **Specify a performance goal for targeted DOE applications (e.g., an average with a minimum)**
 - **Power must be <20 MW**
 - **>64 PB of memory (may be multiple levels)**
 - **Mean time between job interrupts on the order of a day**
 - **System cost < \$200M**
 - **R&D cost < ??**
- **Co-design methodology and IP**
- **Performance portability across different systems through a common programming model and architectural abstraction**





A few technical observations from our discussions...



Heterogeneous multicore nodes are in our future



[Home](#) > [Newsroom](#) > [News Stories](#) >

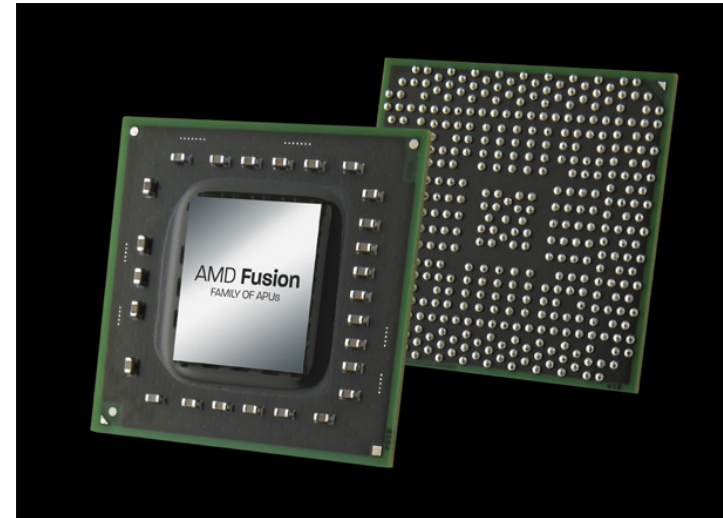
Intel News Release

[383](#)

Intel Unveils New Product Plans for High-Performance Computing

[retweet](#)

Intel® Many Integrated Core Chips to Extend Intel's Role in Accelerating Science and Discovery



NVIDIA Announces "Project Denver" to Build Custom CPU Cores Based on ARM Architecture, Targeting Personal Computers to Supercomputers

NVIDIA Licenses ARM Architecture to Build Next-Generation Processors That Add a CPU to the GPU



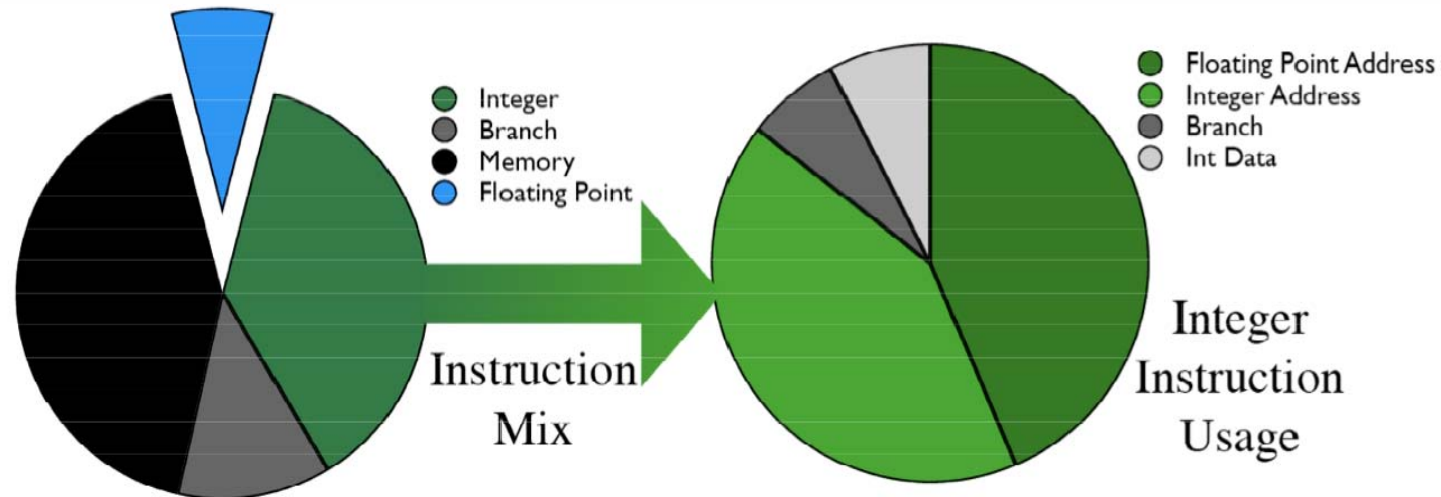


Later this decade an Exascale Node might be:

- 10 TF
- CPU cores -- 10
- GPU
 - Cores – 1000
 - Threads – 100/core
- Fast integrated memory
 - Capacity – 100GB
 - Bandwidth – 1-2 TB/s
- DRAM
 - Capacity – 300 GB
 - Bandwidth – 100 GB/s
- Interconnect
 - ~100 GB/s
- Not clear if mobile devices will require dependability (correctness and reliability)



Memory dominated now FLOPS can be overprovisioned



- Most of DOE's Applications (e.g., climate, fusion, shock physics, ...) spend most of their instructions accessing memory or doing integer computations, not floating point
- Additionally, most integer computations are computing memory Addresses
- Advanced development efforts are focused on accelerating memory subsystem performance for both scientific and informatics applications



Many talks at this workshop have discussed the importance of minimizing power

Power dissipation $\sim CV^2f + L$

C = Capacitance (decreases on a per transistor basis, but the number of transistors increase)

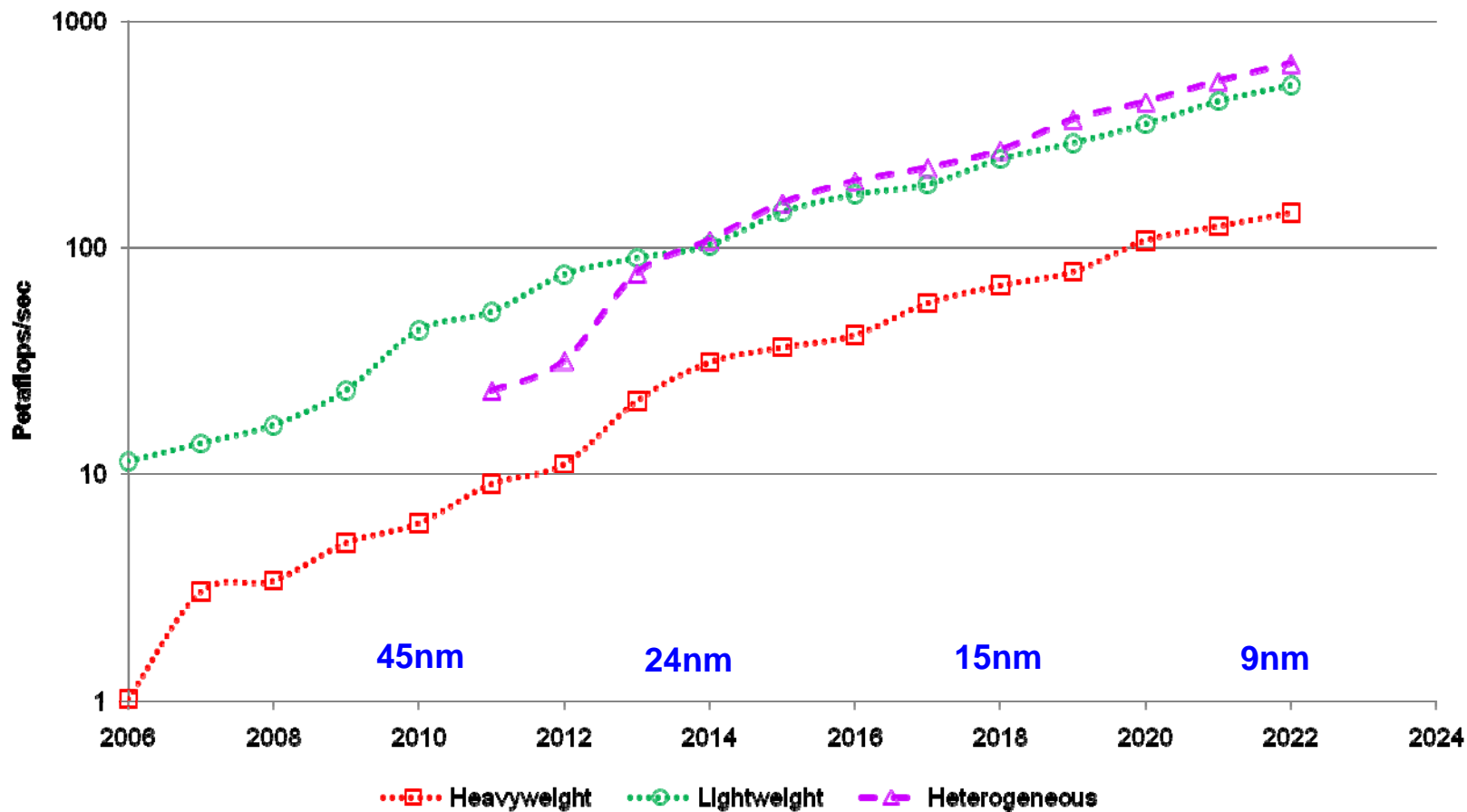
V = Voltage

L = Leakage

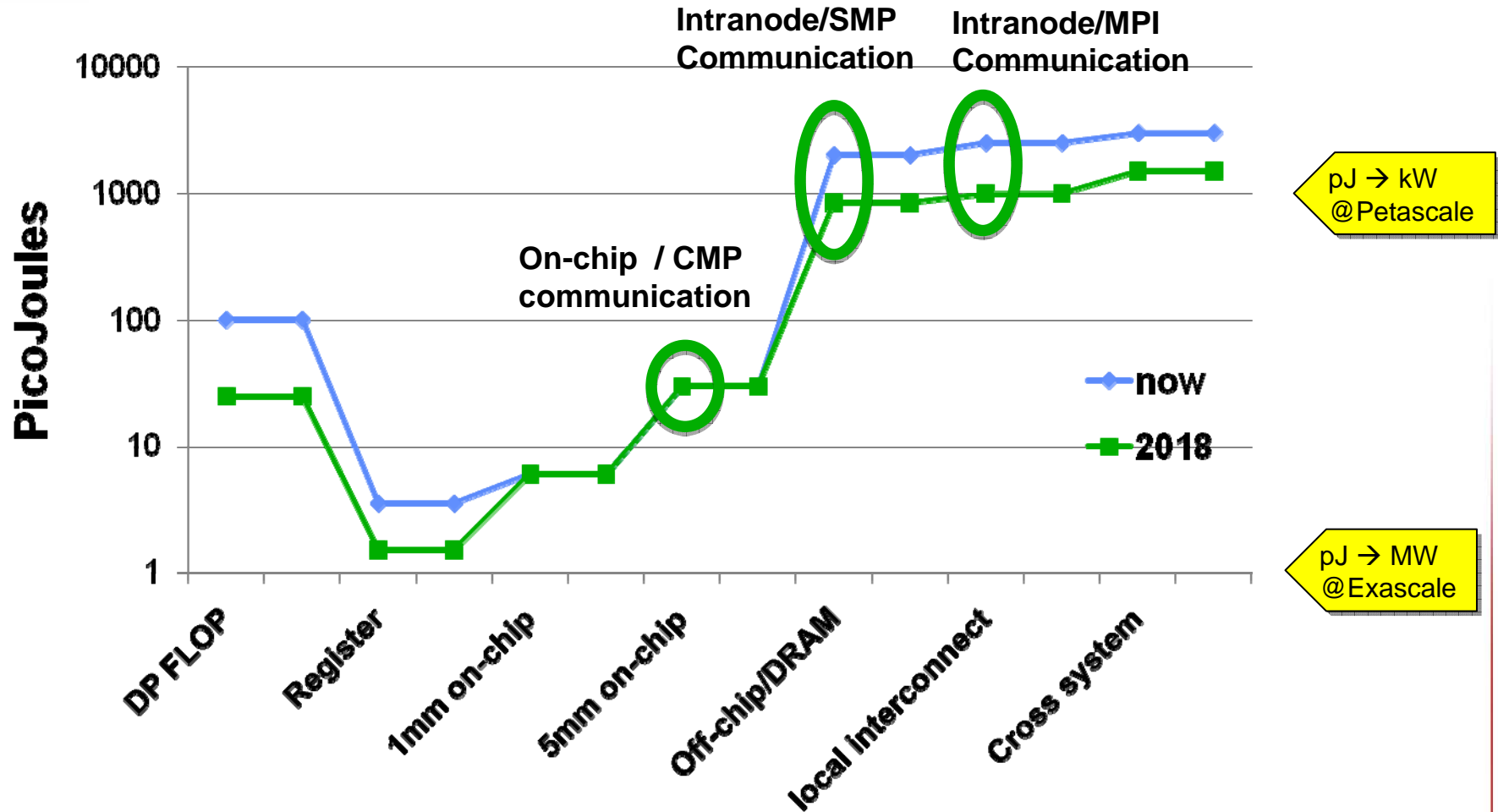


Meeting the 20 MW power goal will be a challenge

Performance Projections - 20MW



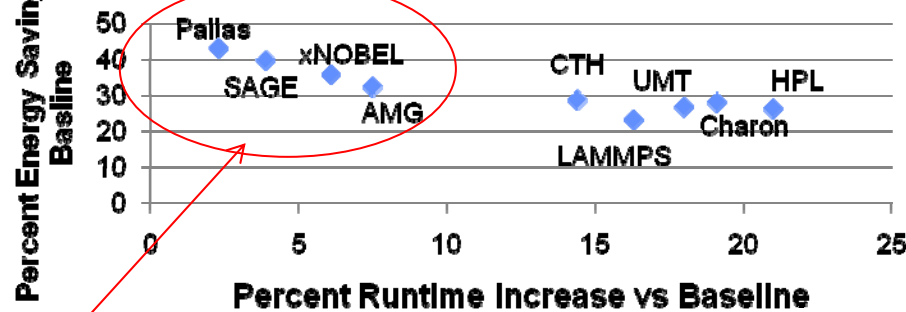
We need to reduce the pJs required to move a bit and applications will need to manage locality



We will need to actively manage Energy/Power

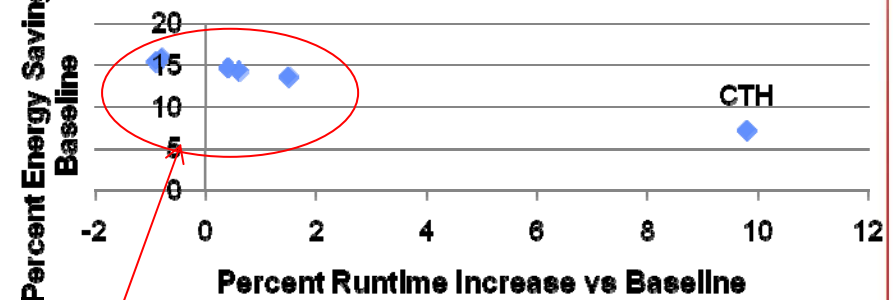
- Ran applications at high processor counts to determine if energy savings persisted at production scales
- Ran two studies
 - Varying processor frequency
 - Varying network bandwidth
- A single, static change can have a significant payoff
 - No need to resort to more frequent adjustments as an application runs
 - Optimal setting per application can be set before job starts

**Tradeoff of Reduced Energy Consumption versus Runtime
Reduced Processor Frequency**



These apps are candidates for running at lower frequency levels

**Tradeoff of Reduced Energy Consumption versus Runtime
Reduced Network Bandwidth**



All but CTH are candidates of running at reduced network bandwidth





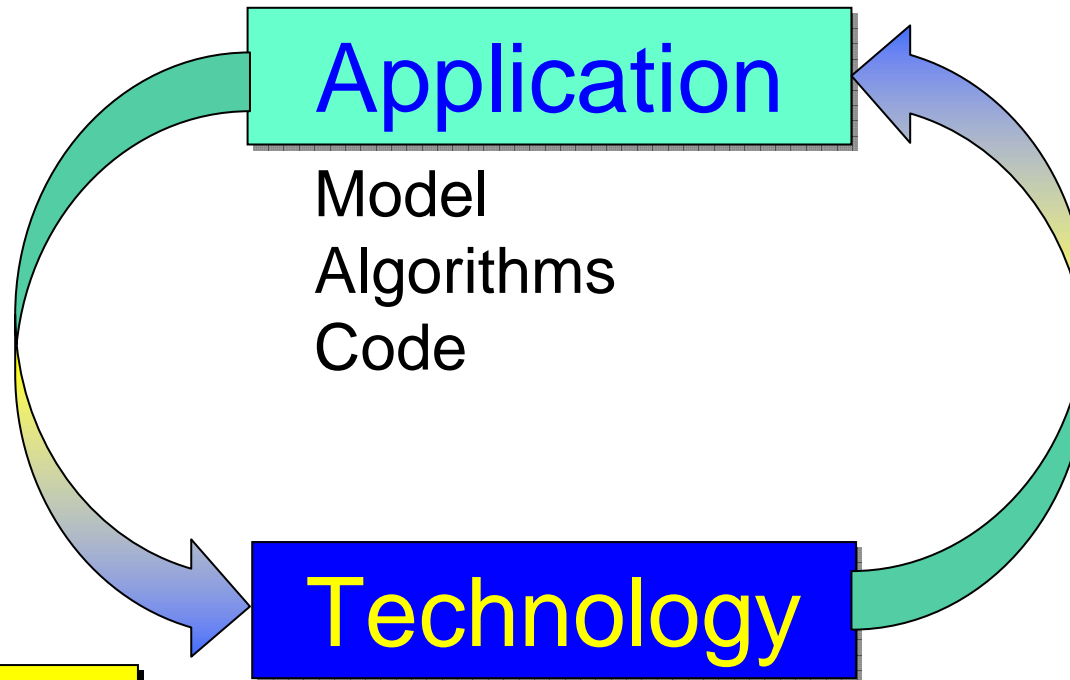
Need for Co-design

- **Need a new methodology to enable algorithms R&D for supercomputers that don't yet exist, are much different from today and are not well-defined**
 - **Analyze the performance of current algorithms on current systems**
 - **Predict the performance of current algorithms on future systems**
 - Usually able to extrapolate when changes are evolutionary (higher clock speed, faster memory, larger L2 cache, improved interconnect)
 - **Predict the performance of new algorithms on future systems**
- **Reaching Exascale will require architectures R&D**
 - **Need to provide feedback on choices, prioritize investments**
- **Overcoming the Exascale challenges will require changes to both hardware and software**



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

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



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

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

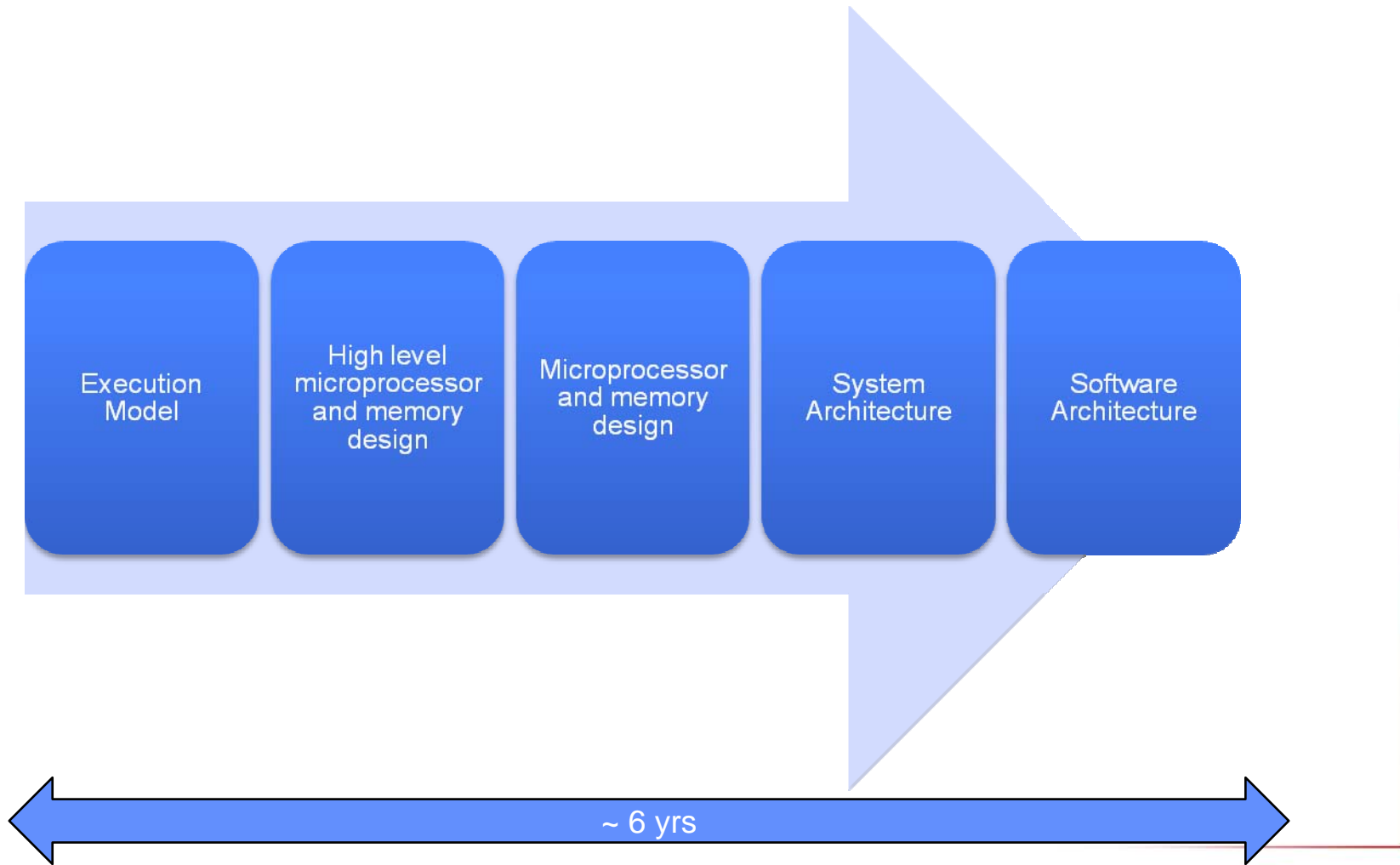
- ⊕ architecture
- ⊕ programming model
- ⊕ resilience
- ⊕ power

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

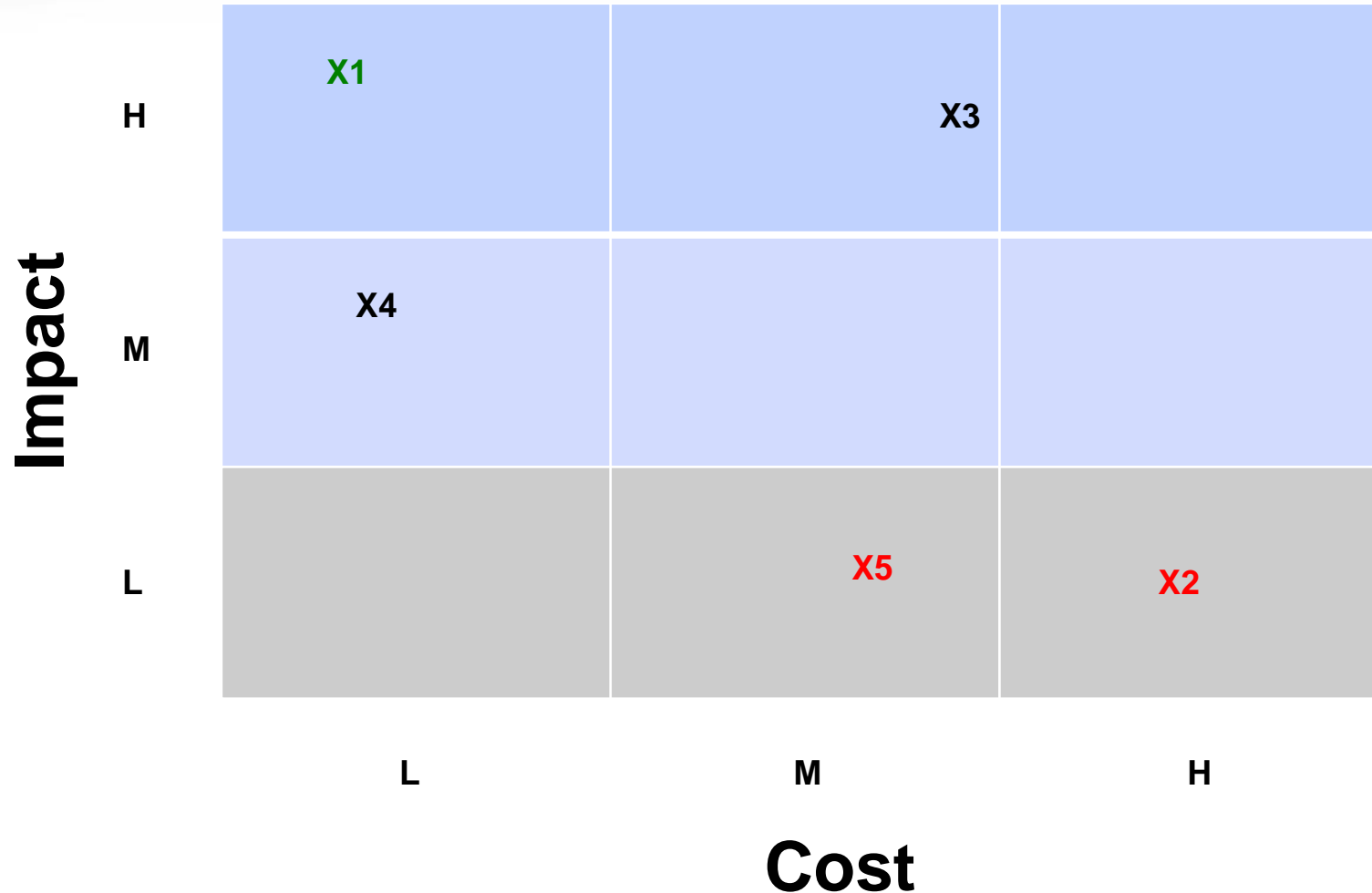




It is urgent to begin soon for co-design to have an impact

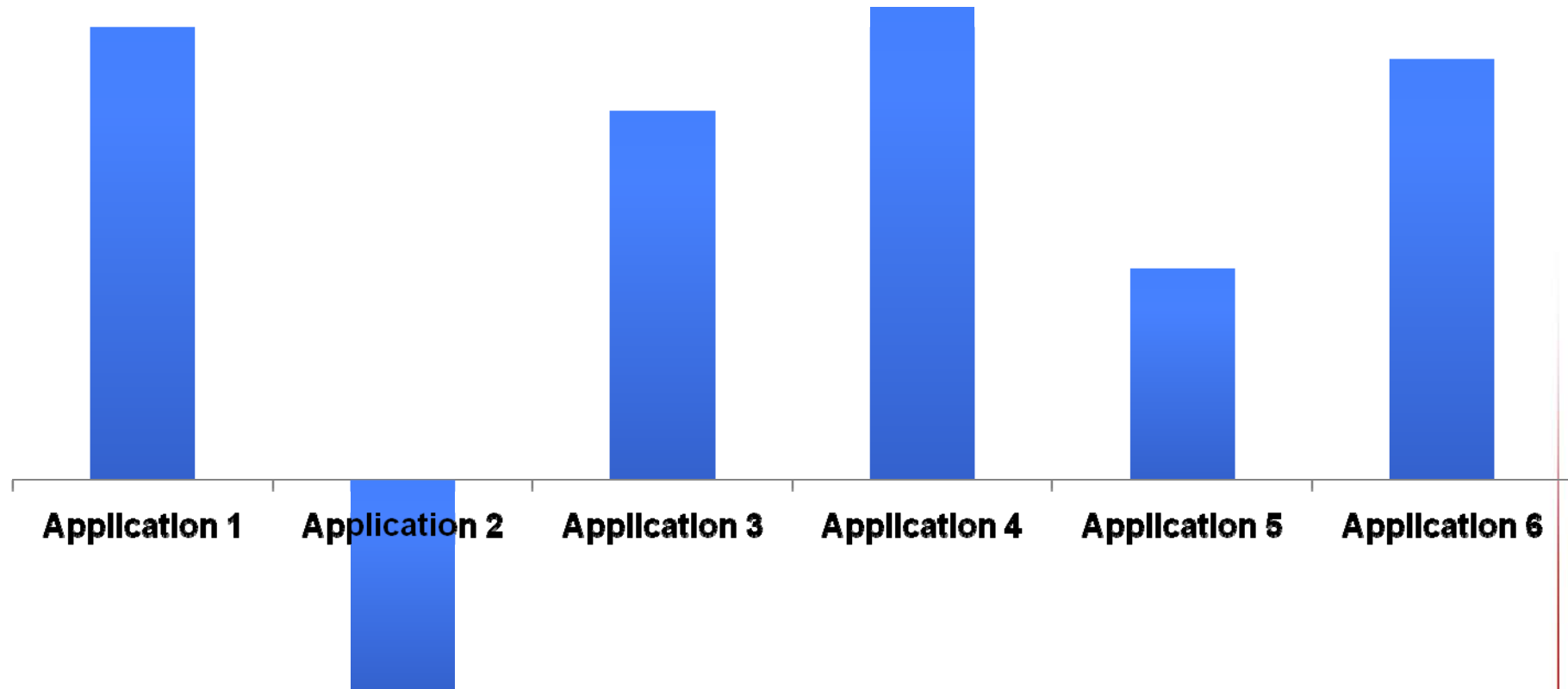


We will need data to make decisions at key points in the design process




Determine the benefit of X_n architectural choices that have a given cost (Si area, energy, R&D)

**We will work with industry partners,
co-design teams and DOE to make decisions**



Is there a weighting? A minimum?



We have the potential to influence many elements of an Exascale system

- **Elements we might influence**
 - **Cores/node, threads/core, scheduling width/thread**
 - **Memory capacity and bandwidth**
 - **Logic in memory subsystem (improve effective bandwidth)**
 - **Interconnect performance**
 - **Dependability**
- **However, we must understand and leverage industry roadmaps**





Technical progress on co-design...



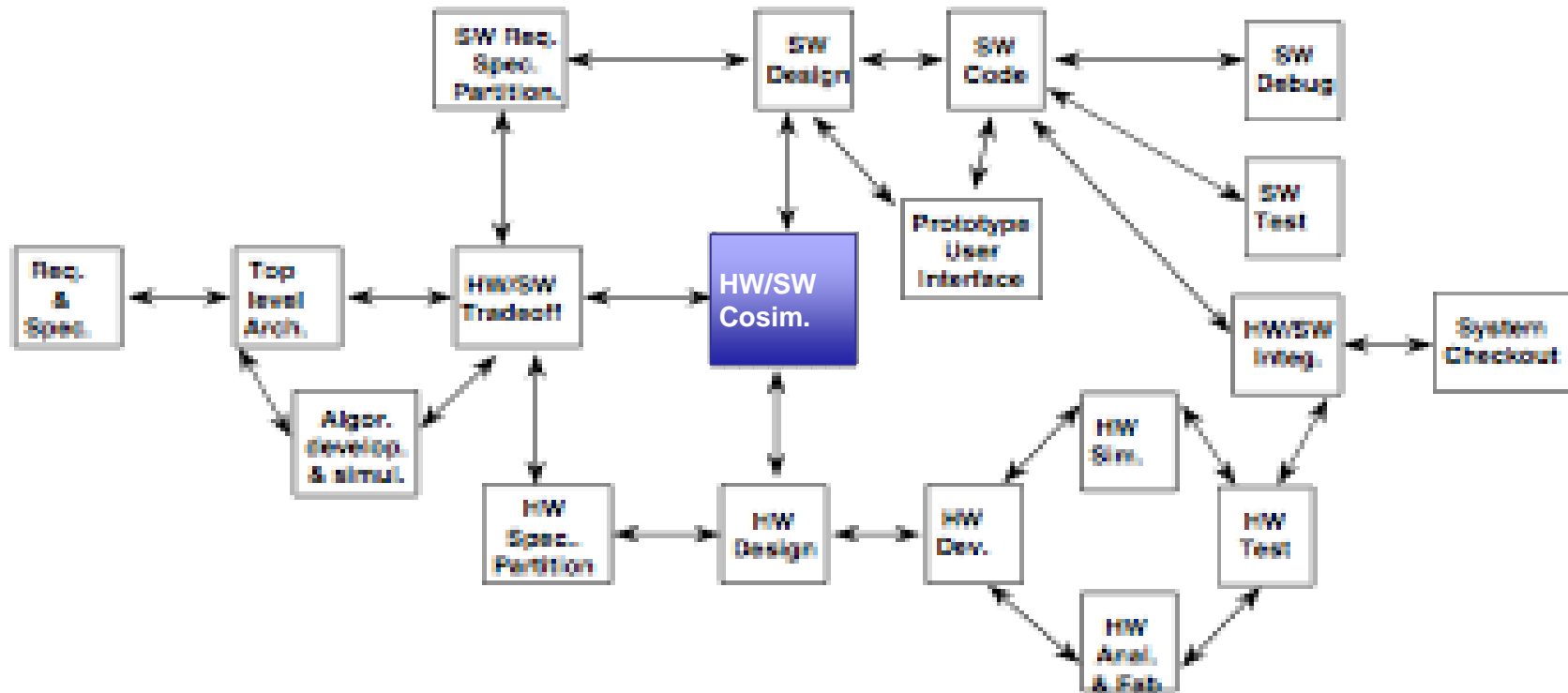


Why has co-design not been used more extensively in HPC?

- Leveraging of COTs technology
 - Almost all leadership systems have some custom components but HPC has benefited from the ability to leverage commercial technology
- ~15-20 years of architectural and programming model stability
 - Bulk synchronous processing + explicit message passing
- Lack of Adequate Simulation Tools → **Structural Simulation Toolkit**
 - Often use Byte to Flop ratios and Excel spreadsheets
 - Industry simulation tools are proprietary
- HPC applications are very complex → **MiniApps**
 - May contain a million of lines of code



Lockheed Martin Co-design Methodology



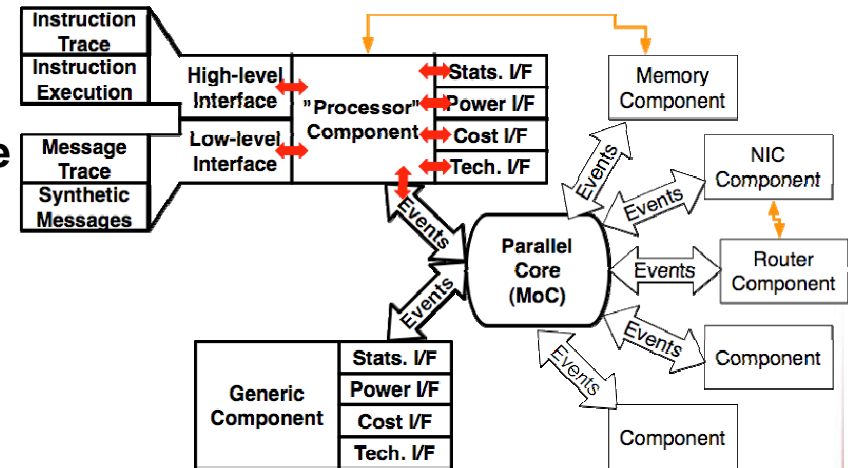
Hierarchical co-simulation is a key for co-design

Structural Simulation Toolkit

- Parallel
- Parallel Discrete Event core with conservative optimization over MPI
- Holistic
- Integrated Tech. Models for power
- McPAT, Sim-Panalyzer
- Multiscale
- Detailed and simple models for processor, network, and memory
- Current Release (2.0) at

<http://www.cs.sandia.gov/sst/>

- Includes parallel simulation core, configuration, power models, basic network and processor models, and interface to detailed memory model

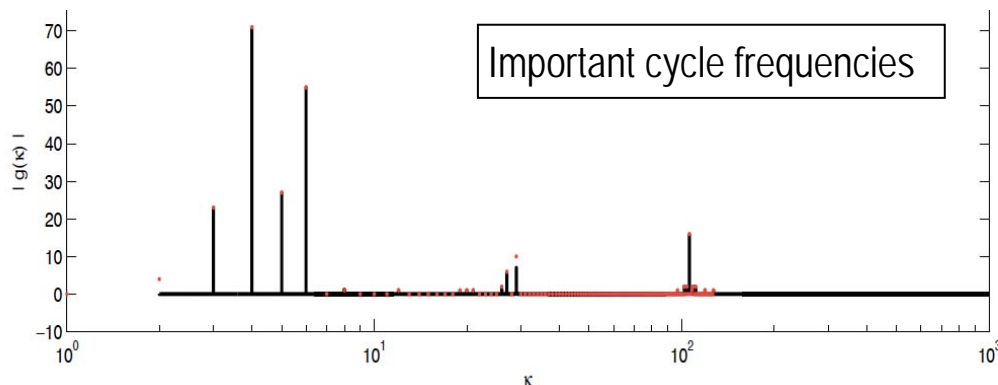
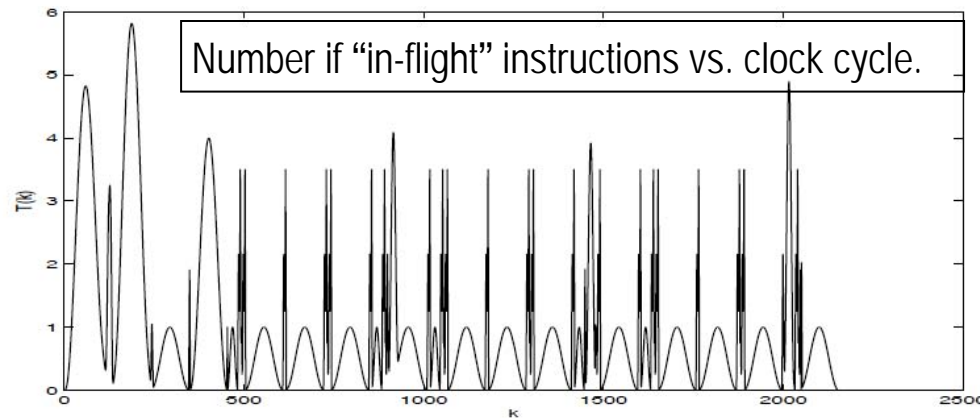


SST is providing architectural insights to algorithms developers

- **Input: SST Trace for SpMV.**
- **Lots of instruction stream data.**
- **Model: Use restricted \sin^2 function to mark start/finish of each instruction.**
- **Use FFTs to analyze behavior.**

Trace fragment from SpMV inner loop

j	I_j	issue	complete	κ
59	bc	737	741	4
60	lwz	738	744	6
61	lfd	740	746	6
62	addi	742	746	4
63	addi	742	746	4
64	rlwinm	743	746	3
65	lfdx	744	850	106
66	fmadd	849	854	5
67	bc	850	854	4
68	lwz	851	857	6
69	lfd	853	859	6
70	addi	855	859	4
71	addi	855	859	4
72	rlwinm	856	859	3
73	lfdx	857	886	29
74	fmadd	885	890	5
75	bc	886	890	4
76	lwz	887	893	6
77	lfd	889	895	6
78	addi	891	895	4
79	addi	891	895	4
80	rlwinm	892	895	3
81	lfdx	893	899	6
82	fmadd	898	903	5
83	bc	899	903	4

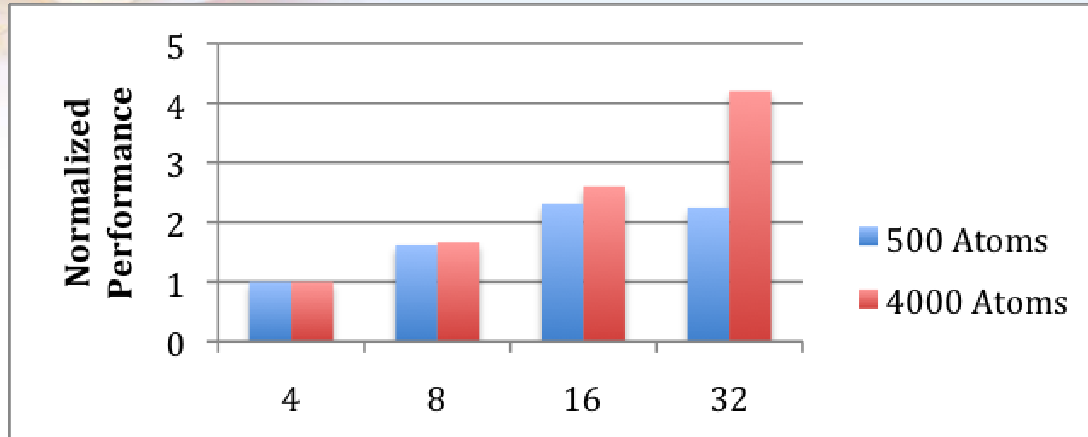


Component Validation

- **Strategy: component validation in parallel with system-level validation**
- **Current components validated at different levels, with different methodologies**
- **Validation in isolation**
- **What is needed**
 - **Uniform validation methodology (apps)**
 - **System (multi-component) level validation**

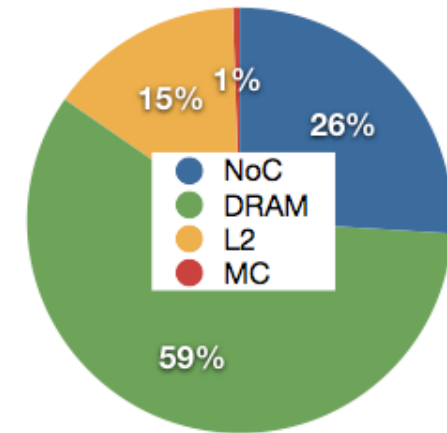
Component	Method	Error
DRAMSim	RTL Level validation against Micron	Cycle
Generic Proc	Simple scalar SPEC92 Validation	~5%
NMSU	Comparison vs. existing processors on SPEC	<7%
RS Network	Latency/BW against SeaStar 1.2, 2.1	<5%
MacSim	Comparison vs. Existing GPUs	Ongoing <10% expected
Zesto	Comparison vs several processors, benchmarks	4-5%
McPAT	Comparisons against existing processors	10-23%

Sample Results –Node Level

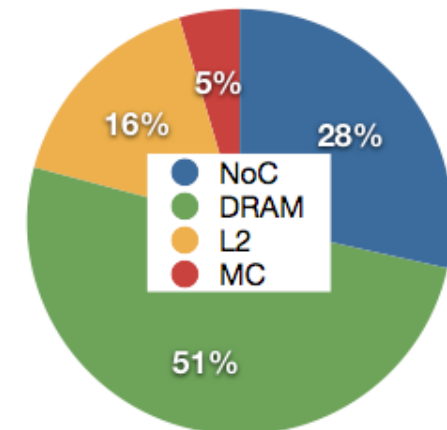


SST Simulation of MD code shows diminishing returns for threading on small data sets

GUPS Memory Power Breakdown

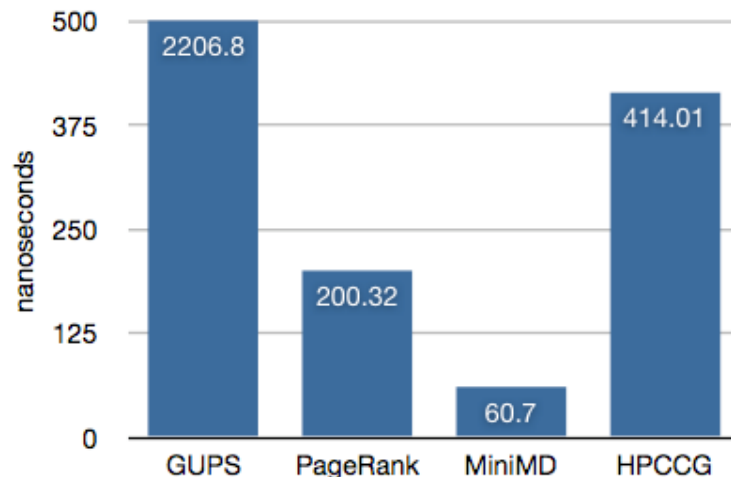


MiniMD Memory Power Breakdown



Power analysis help prioritize technology investments

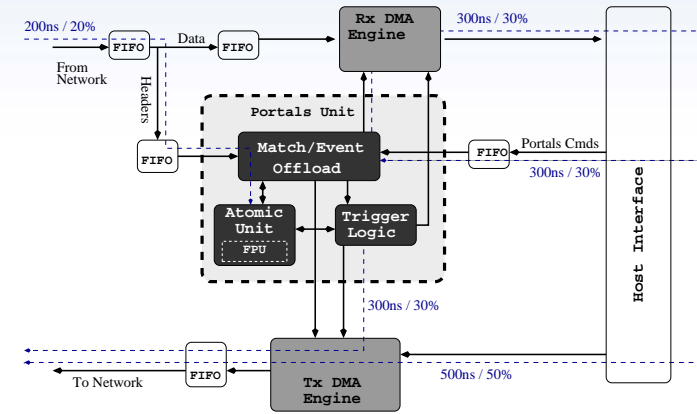
Avg. Memory Latency



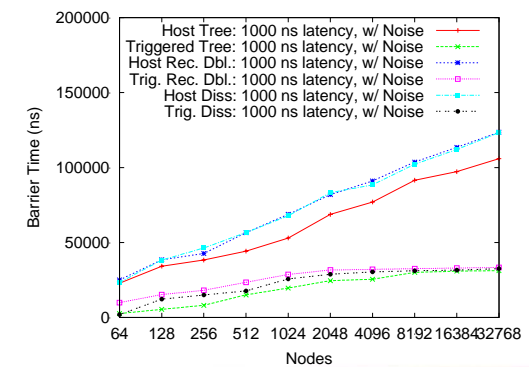
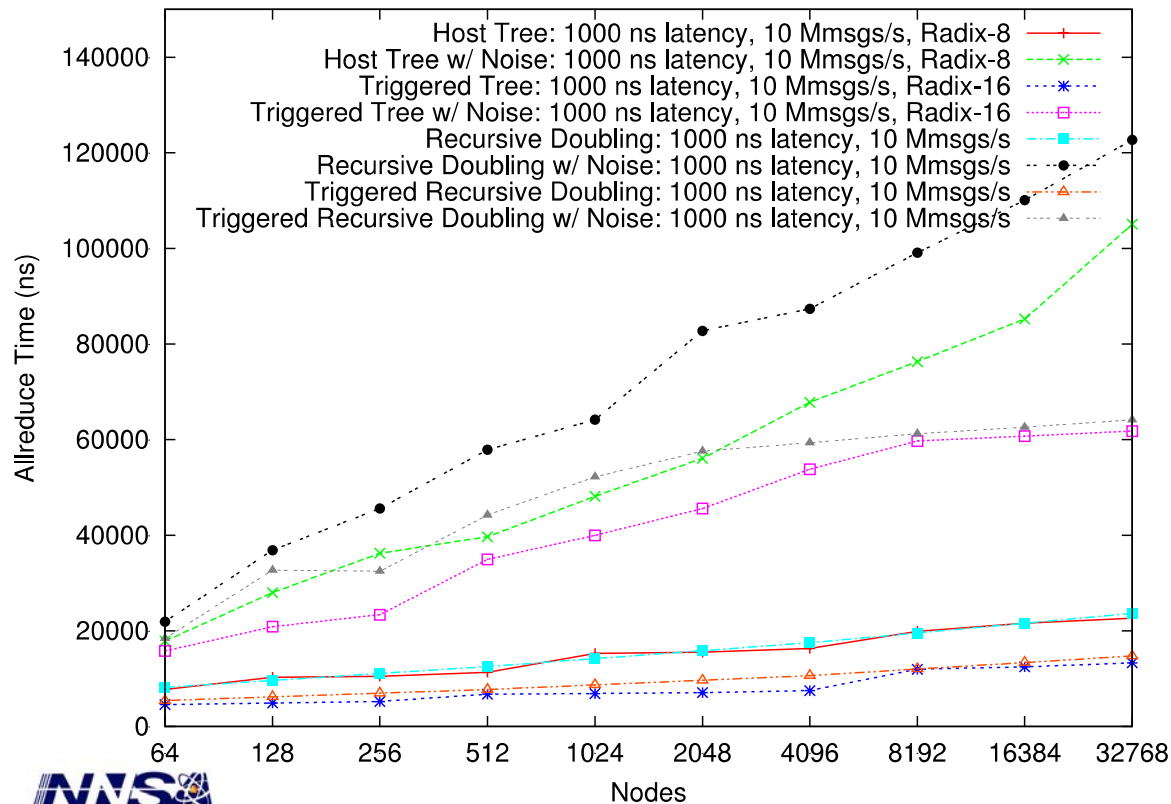
Detailed component simulation highlights bottlenecks

Sample Results – System Level

Simulation of new network API semantics (triggered operations) enabling flexible collective offload shows advantages in latency and noise tolerance



Simulation uses validated Red Storm router model coupled with a block-level NIC model (shown above) and a high level processor model



Miniapps: Specs

- **Size:** O(1K) lines.
- **Focus:** Proxy for key app performance issue.
- **Availability:** Open Source.
- **Scope of allowed change:** Any and all.
- **Intent:** Co-design: From HW registers to app itself.
- **Developer & owner:** *Application team.*
- **Lifespan:** *Until it's no longer useful.*



Mantevo* Project

* Greek: augur, guess, predict, presage



- Multi-faceted application performance project.
- **Started 4 years ago.**
- Two types of packages:
 - **Miniapps:** Small, self-contained programs.
 - **MiniFE/HPCCG:** unstructured implicit FEM/FVM.
 - **phdMesh:** explicit FEM, contact detection.
 - **MiniMD:** MD Force computations.
 - **MiniXyce:** Circuit RC ladder.
 - **CTH-Comm:** Data exchange pattern of CTH.
 - **Minidrivers:** Wrappers around Trilinos packages.
 - **Beam:** Intrepid+FEI+Trilinos solvers.
 - **Epetra Benchmark Tests:** Core Epetra kernels.
 - **Dana Knoll working on new one.**
- Open Source (LGPL)
- Staffing: Application & Library developers.



Charon Complexity

- **SLOCCOUNT (tool from David A. Wheeler)**
 - **Charon physics:** 191,877 SLOC.
 - **Charon + nevada framework** 414,885 SLOC
 - **Charon_TPL** 4,022,296 SLOC
- **Library dependencies:**
 - **25 Trilinos package.**
 - **15 other TPLs.**
- **Requires “heroic effort” to build**
- **MPI-only, no intranode parallelism**
- **Export controlled**



MiniFE Complexity

- **SLOCCOUNT:**
 - **Main code:** 6,469 SLOC
 - **Optional libraries (from Trilinos):** 37,040 SLOC
- **Easy to build:**
 - **Multiple targets:**
 - Internode: MPI or not.
 - Intranode: Serial, Pthreads, OpenMP, TBB, CUDA.
 - **Dialable properties:**
 - Compute load imbalance.
 - Communication imbalance.
 - Data types: float, double, mixed.
- **Open source**



Does MiniFE Predict Charon Behavior?

Processor Ranking: 8 MPI tasks; 31k DOF/core

- Charon steady-state drift-diffusion BJT
- Nehalem (Intel 11.0.081 –O2 –xsse4.2; all cores of dual-socket quadcore)
- 12-core Magny-Cours (Intel 11.0.081 –O2; one socket, 4 MPI tasks/die)
- Barcelona (Intel 11.1.064 –O2; use two sockets out of the quad-socket)
- 2D Charon (3 DOF/node) vs. 3D MiniFE; match DOF/core and NNZ in matrix row
- Charon LS w/o or w/ ps: GMRES linear solve without/with ML precondition setup time
- Try to compare MiniFE “assembling FE”+”imposing BC” time with Charon equivalent

MiniFE

	CG	FE assem+BC
1	Nehalem	Nehalem
2	MC(1.7)	MC(1.7)
3	Barc(2.7)	Barc(1.8)

Charon

	LS w/o ps	LS w/ ps	Mat+RHS
1	Nehalem	Nehalem	Nehalem
2	MC(1.7)	MC(1.8)	MC(1.46)
3	Barc(2.8)	Barc(2.5)	Barc(1.52)

Number in parenthesis is factor greater than #1 time



MiniFE Predict Charon? Multicore Efficiency Dual-Socket 12-core Magny-Cours : 124k DOF/core

- Charon steady-state drift-diffusion BJT; Intel 11.0.081 -O2
- Weak scaling study with 124k DOF/core
- 2D Charon (3 DOF/node) vs. 3D MiniFE; match DOF/core and NNZ in matrix row
- Efficiency: ratio of 4-core time to n-core time (expressed as percentage)
- Charon LS w/o or w/ ps: GMRES linear solve without/with ML precondition setup time
- 100 Krylov iterations for both MiniFE and Charon (100 per Newton step)

MiniFE

cores	CG eff
4	Ref
8	89
12	73
16	61
20	54
24	45

Charon

cores	LS w/o ps eff	LS w/ ps eff
4	Ref	Ref
8	87	89
12	74	78
16	61	66
20	49	54
24	40	45



Conclusions

- **A DOE Exascale Effort has support from the President and U.S. Congress**
- **Plan to start in FY12**
- **There is technical progress on co-design, a key element of the strategy**
 - **Hardware/software co-simulation tools are being developed**
 - **MiniApps are being used to reduce complexity by a factor of 1000**

