Exploring Emerging Technologies in the HPC Co-Design Space

Jeffrey Vetter

Presented to
10th International Conf on Parallel Processing and Applied Mathematics
Warsaw
9 September 2013

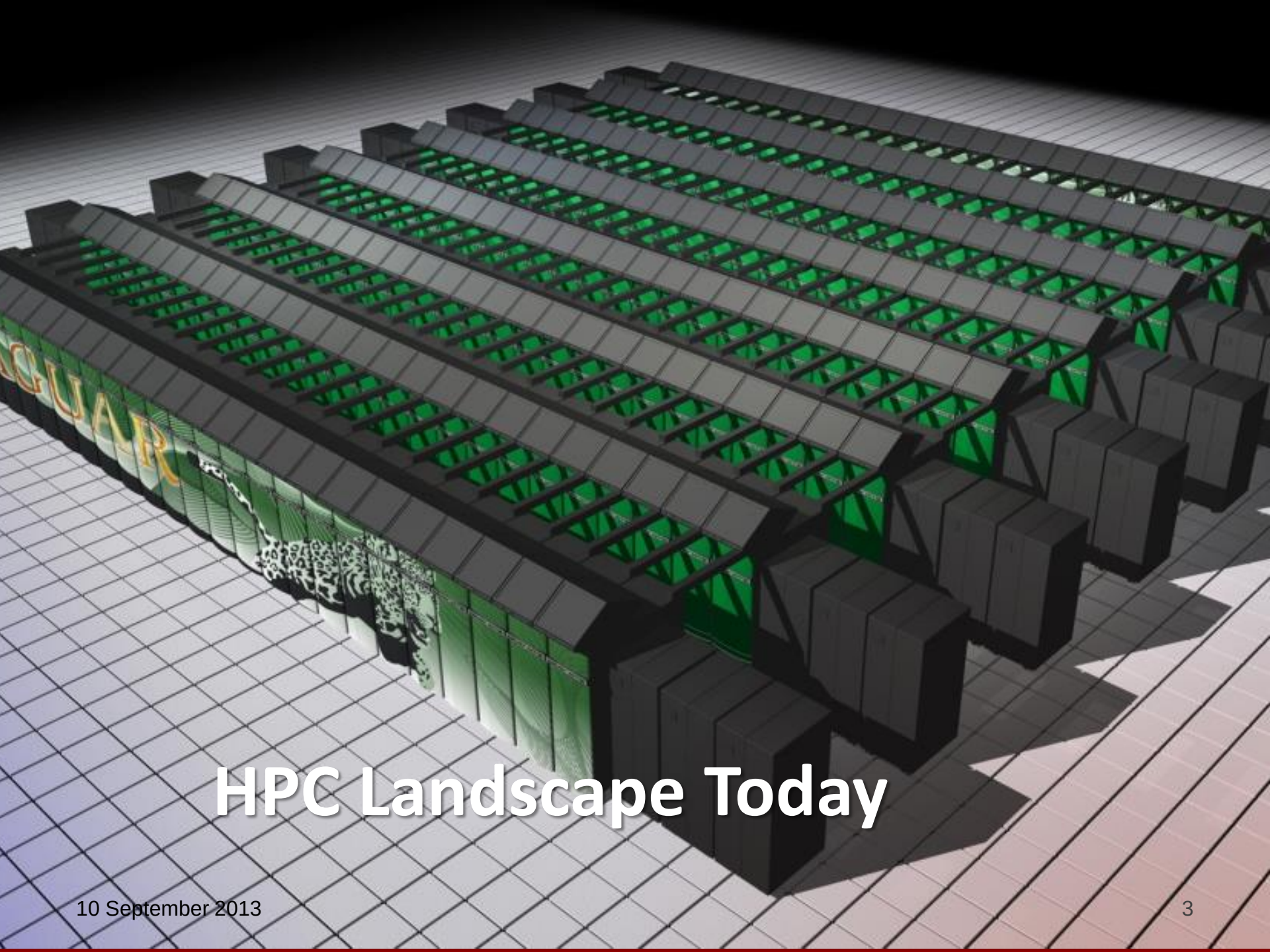
OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Georgia Tech College of Computing
Computational Science and Engineering

http://ft.ornl.gov ♦ vetter@computer.org

Presentation in a nutshell

- Our community expects major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - Major shifts and uncertainty in architectures, software, applications
  - Applications will have to change in response to design of processors, memory systems, interconnects, storage
- Technologies particularly pertinent to addressing some of these challenges
  - Heterogeneous computing
  - Nonvolatile memory
- DOE has initiated Codesign Centers that bring together all stakeholders to develop integrated solutions
- Aspen is a new approach to model characteristics of applications and emerging architectures
  - This structure allows easy development, sharing, verification of models
  - Automated exploration of design spaces



HPC Landscape Today

Contemporary HPC Architectures

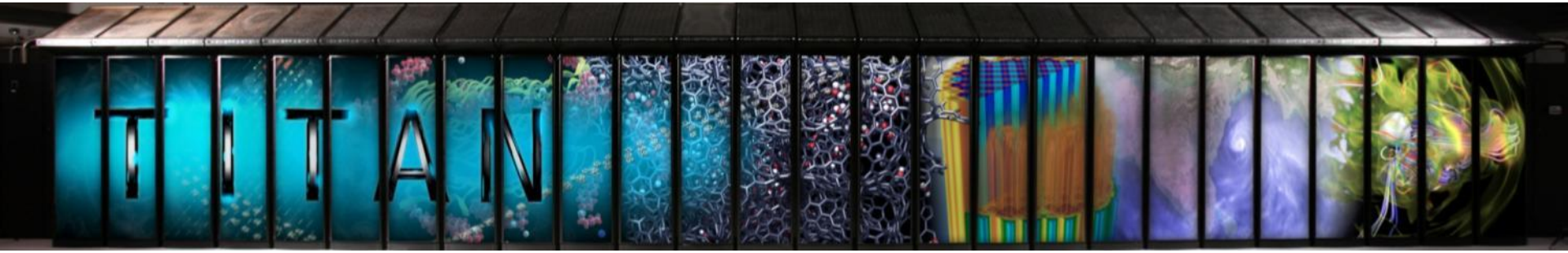
Date	System	Location	Comp	Comm	Peak (PF)	Power (MW)
2009	Jaguar; Cray XT5	ORNL	AMD 6c	Seastar2	2.3	7.0
2010	Tianhe-1A	NSC Tianjin	Intel + NVIDIA	Proprietary	4.7	4.0
2010	Nebulae	NSCS Shenzhen	Intel + NVIDIA	IB	2.9	2.6
2010	Tsubame 2	TiTech	Intel + NVIDIA	IB	2.4	1.4
2011	K Computer	RIKEN/Kobe	SPARC64 VIIIfx	Tofu	10.5	12.7
2012	Titan; Cray XK6	ORNL	AMD + NVIDIA	Gemini	27	9
2012	Mira; BlueGeneQ	ANL	SoC	Proprietary	10	3.9
2012	Sequoia; BlueGeneQ	LLNL	SoC	Proprietary	20	7.9
2012	Blue Waters; Cray	NCSA/UIUC	AMD + (partial) NVIDIA	Gemini	11.6	
2013	Stampede	TACC	Intel + MIC	IB	9.5	5
2013	Tianhe-2	NSCC-GZ (Guangzhou)	Intel + MIC	Proprietary	54	~20

TH-2 System

- 54 Pflop/s Peak!
- Compute Nodes have 3.432 Tflop/s per node
  - 16,000 nodes
  - 32000 Intel Xeon cpus
  - 48000 Intel Xeon phis (57c/phi)
- Operations Nodes
  - 4096 FT CPUs as operations nodes
- Proprietary interconnect TH2 express
- 1PB memory (host memory only)
- Global shared parallel storage is 12.4 PB
- Cabinets: 125+13+24 = 162 compute/communication/storage cabinets
  - ~750 m2
- NUDT and Inspur

TH-2 (w/ Dr. Yutong Lu)

DOE's "Titan" Hybrid System: Cray XK7 with AMD Opteron and NVIDIA Tesla processors



4,352 ft²

SYSTEM SPECIFICATIONS:

- Peak performance of 27.1 PF
  - 24.5 GPU + 2.6 CPU
- 18,688 Compute Nodes each with:
  - 16-Core AMD Opteron CPU
  - NVIDIA Tesla "K20x" GPU
  - 32 + 6 GB memory
- 512 Service and I/O nodes
- 200 Cabinets
- 710 TB total system memory
- Cray Gemini 3D Torus Interconnect
- 8.9 MW peak power


Looking Forward to Exascale

Notional Exascale Architecture Targets

(From Exascale Arch Report 2009)

System attributes	2001	2010	"2015"		"2018"	
System peak	10 Tera	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	~0.8 MW	6 MW	15 MW		20 MW	
System memory	0.006 PB	0.3 PB	5 PB		32-64 PB	
Node performance	0.024 TF	0.125 TF	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	16	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	416	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW		1.5 GB/s	150 GB/sec	1 TB/sec	250 GB/sec	2 TB/sec
MTTI		day	O(1 day)		O(1 day)	

http://science.energy.gov/ascr/news-and-resources/workshops-and-conferences/grand-challenges/

Constraint: Facilities and Power

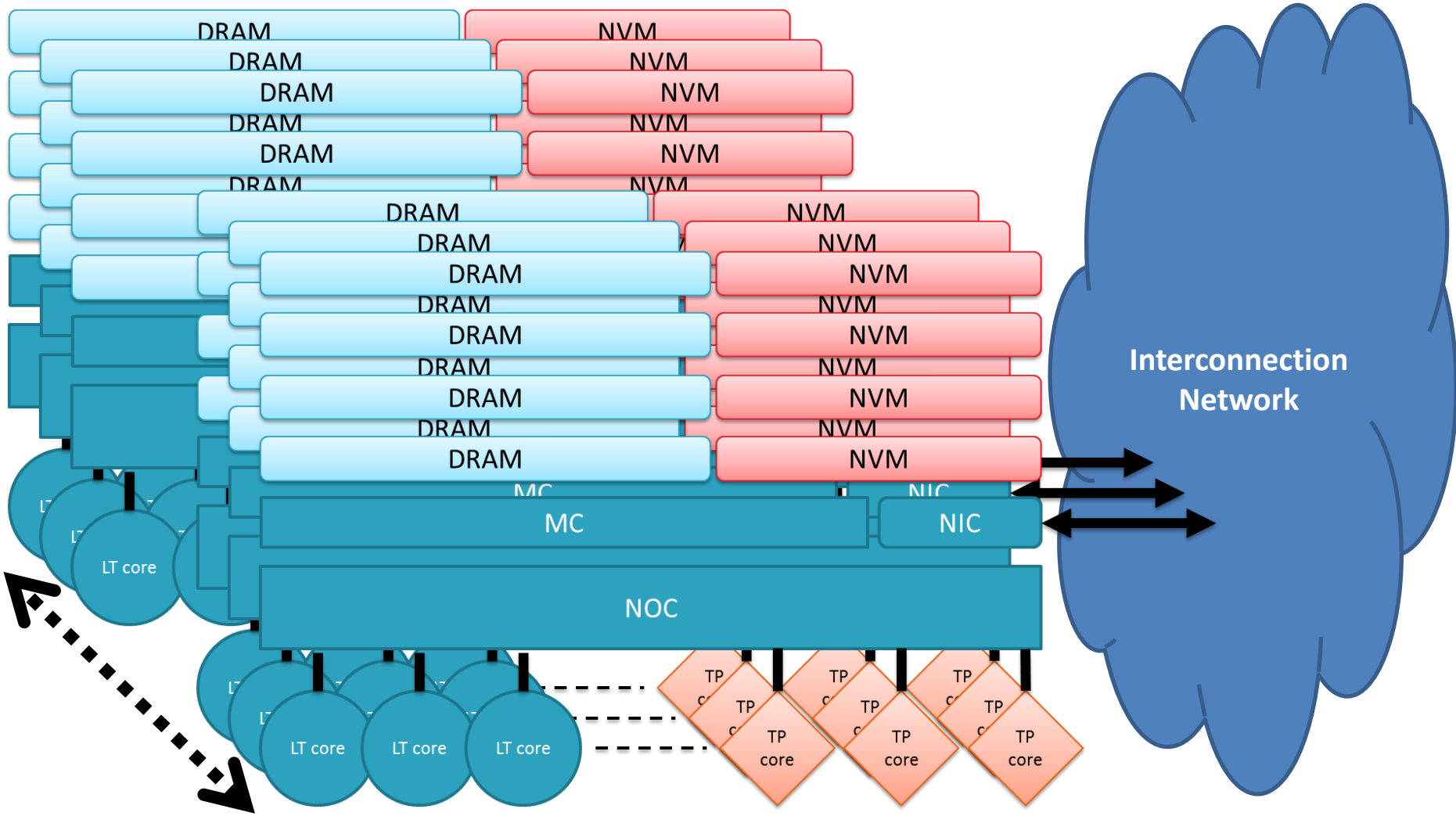
Critical Concerns for Future Systems

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Tf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 CPUs	1,000 CPUs	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

Table 1: Potential Exascale Computer Design for 2018 and its relationship to current HPC designs. [58]

- **Small memory capacity has profound impact on other features**
- **Feeding the core(s)**
- **Poor efficiencies**
- **Small messages, I/O**

Notional Future Architecture



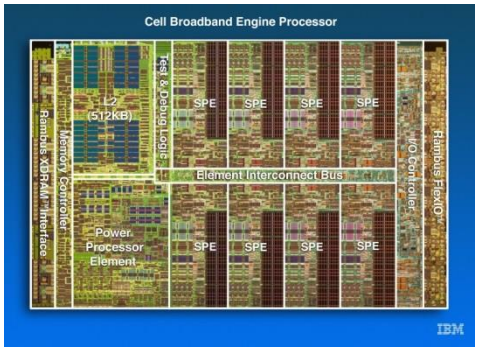
Emerging Architectures

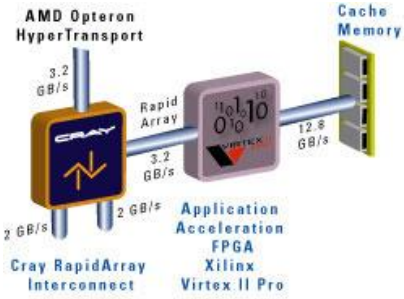
‘The root of all evil’ – Anonymous application scientist

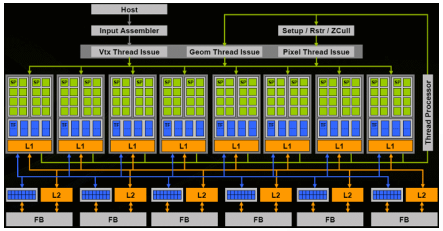
Recent Experimental Computing Systems

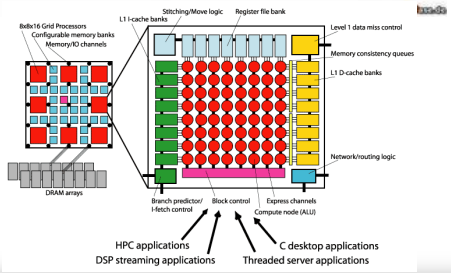
- The past decade has started the trend away from traditional architectures
- Mainly driven by facilities costs and successful (sometimes heroic) application examples
- Examples
  - Cell, GPUs, FPGAs, SoCs, etc
- Many open questions
  - Understand technology challenges
  - Evaluate and prepare applications
  - Recognize, prepare, enhance programming models

Popular architectures since ~2004



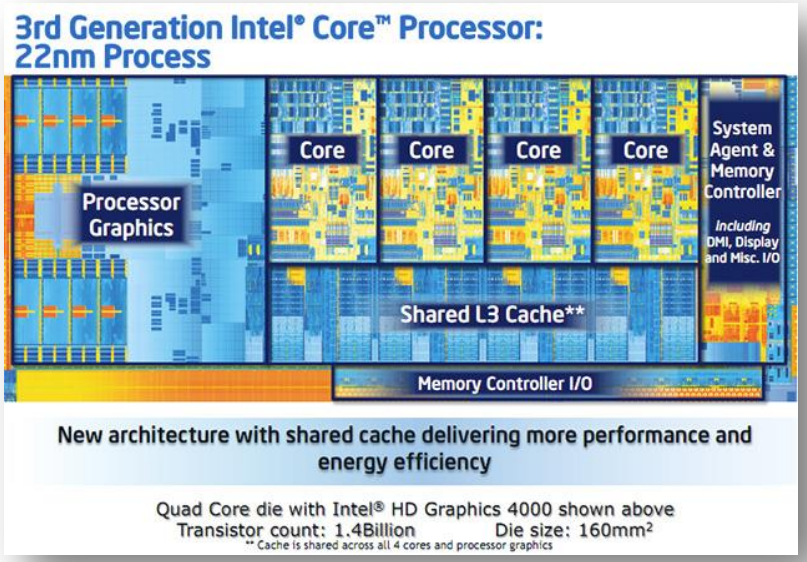


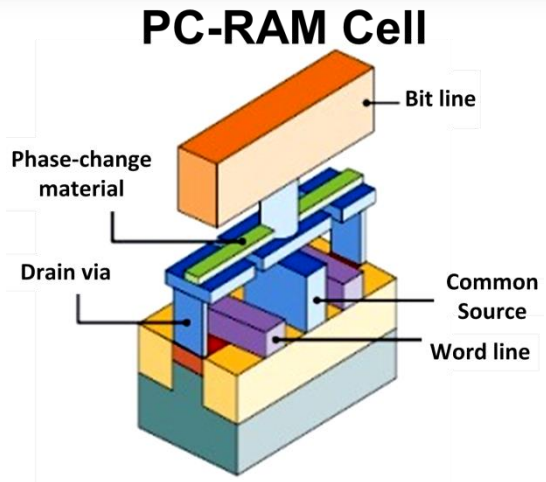




Emerging Computing Architectures – Future Possibilities

- Heterogeneous processing
  - Many cores
  - Fused, configurable memory
- Memory
  - 2.5D and 3D Stacking
  - New devices (PCRAM, ReRAM)
- Interconnects
  - Collective offload
  - Scalable topologies
- Storage
  - Active storage
  - Non-traditional storage architectures (key-value stores)
- Improving performance and programmability in face of increasing complexity
  - Power, resilience

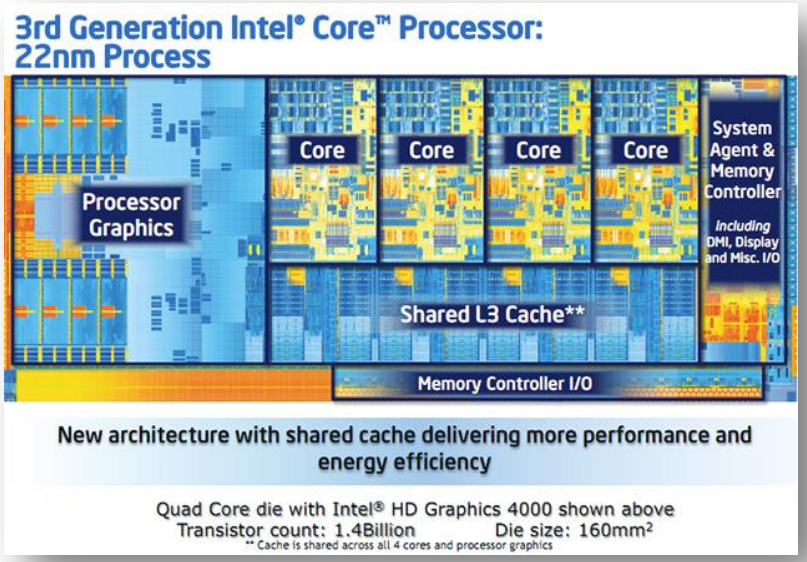


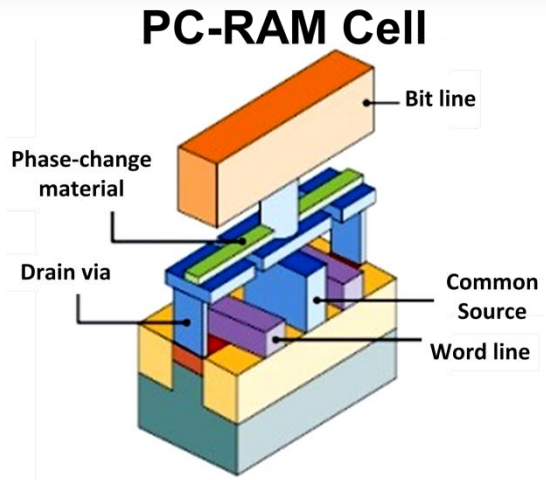


HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.

Emerging Computing Architectures – Future Possibilities

- Heterogeneous processing
  - Many cores
  - Fused, configurable memory
- Memory
  - 3D Stacking
  - New devices (PCRAM, ReRAM)
- Interconnects
  - Collective offload
  - Scalable topologies
- Storage
  - Active storage
  - Non-traditional storage architectures (key-value stores)
- Improving performance and programmability in face of increasing complexity
  - Power, resilience



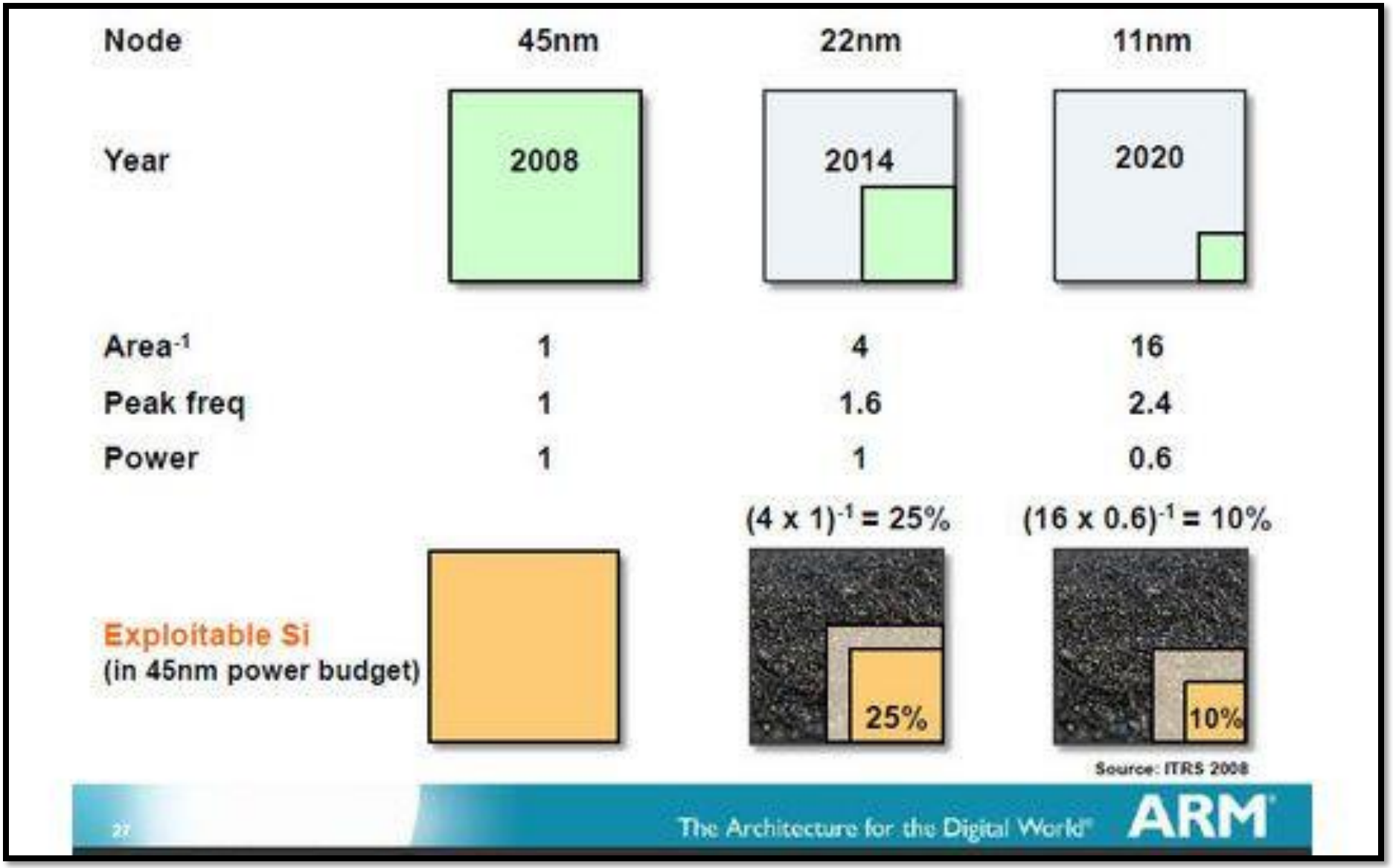


HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.

Heterogeneous Computing

You could not step twice into the same river. -- Heraclitus

Opportunity: Dark Silicon Will Make Heterogeneity and Specialization More Relevant



Source: ARM

http://keeneland.gatech.edu

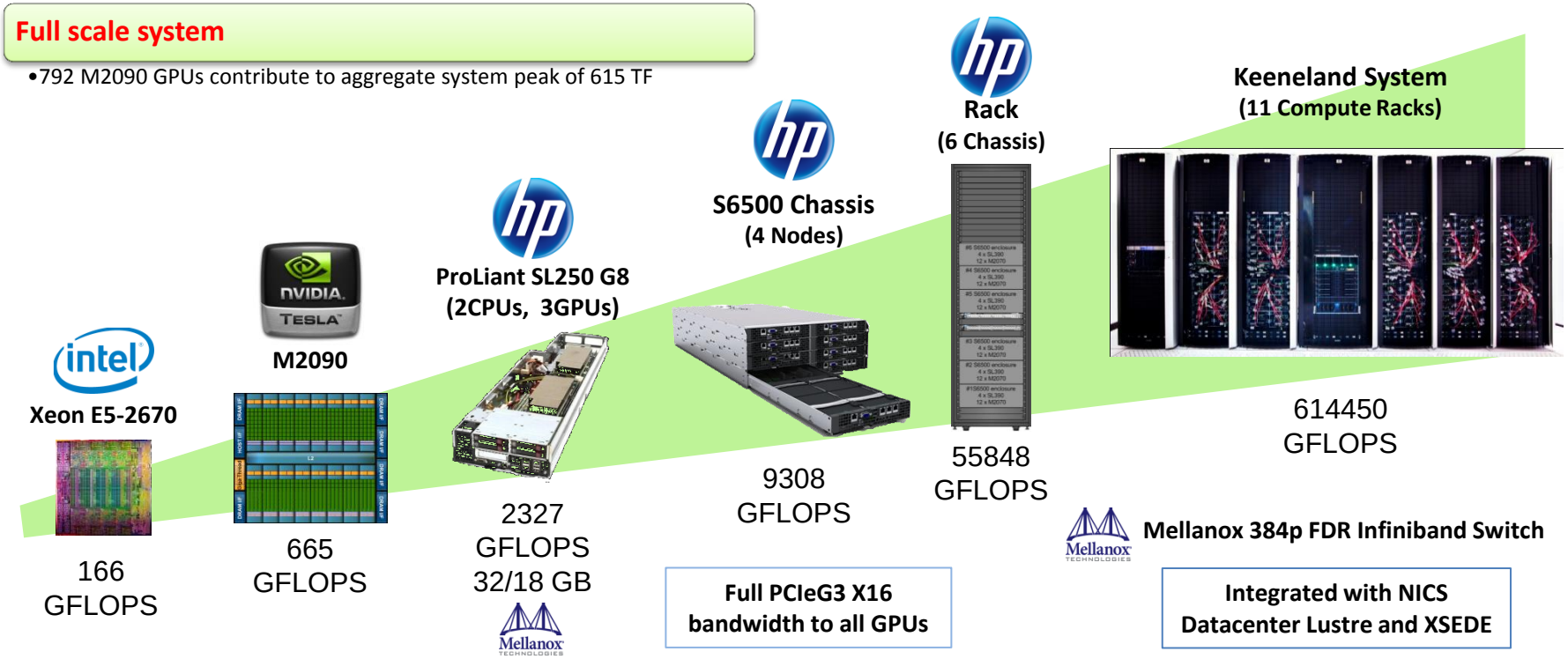
Keeneland – Full Scale System

Initial Delivery system installed in Oct 2010

- 201 TFLOPS in 7 racks (90 sq ft incl service area)
- 902 MFLOPS per watt on HPL (#12 on Green500)
- Upgraded April 2012 to 255 TFLOPS
- Over 200 users, 100 projects using KID

Full scale system

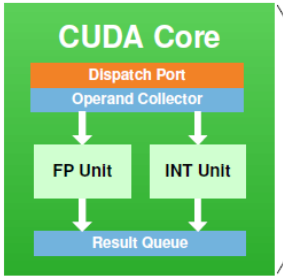
- 792 M2090 GPUs contribute to aggregate system peak of 615 TF

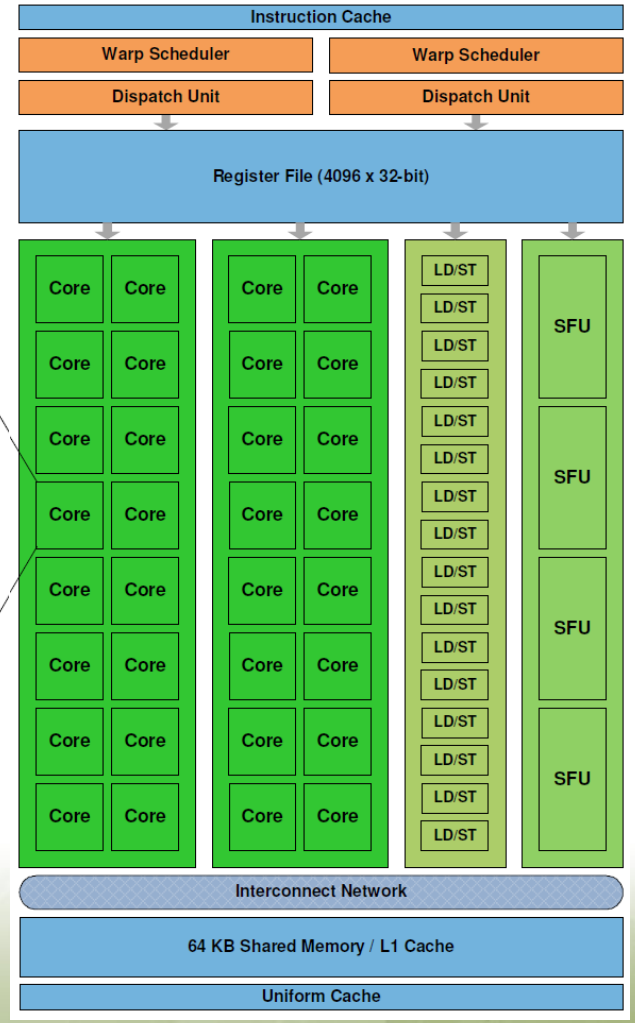


J.S. Vetter, R. Glassbrook *et al.*, *"Keeneland: Bringing heterogeneous GPU computing to the computational science community," IEEE Computing in Science and Engineering, 13(5):90-5, 2011, http://dx.doi.org/10.1109/MCSE.2011.83.*

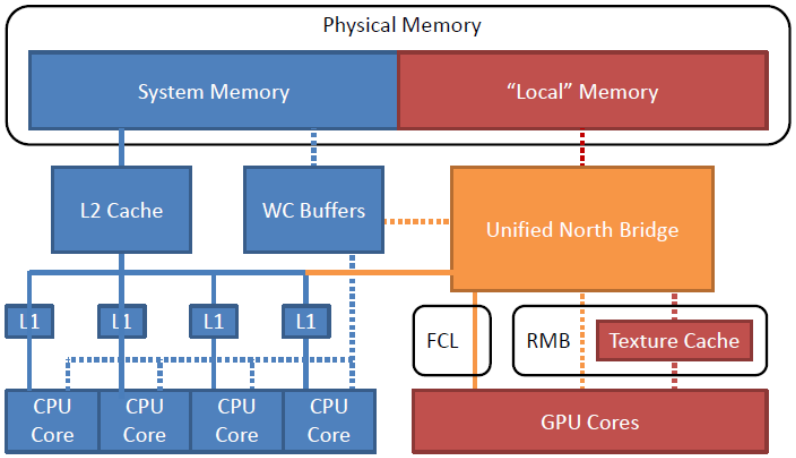
NVIDIA Fermi - M2090

- 3B transistors in 40nm
- 512 CUDA Cores
  - New IEEE 754-2008 floating-point standard
    - FMA
    - 8× the peak double precision arithmetic performance over NVIDIA's last generation GPU
  - 32 cores per SM, 21k threads per chip
- 384b GDDR5, 6 GB capacity
  - 178 GB/s memory BW
- C/M2090
  - 665 GigaFLOPS DP, 6GB
  - ECC Register files, L1/L2 caches, shared memory and DRAM





Fused memory hierarchy: AMD Llano



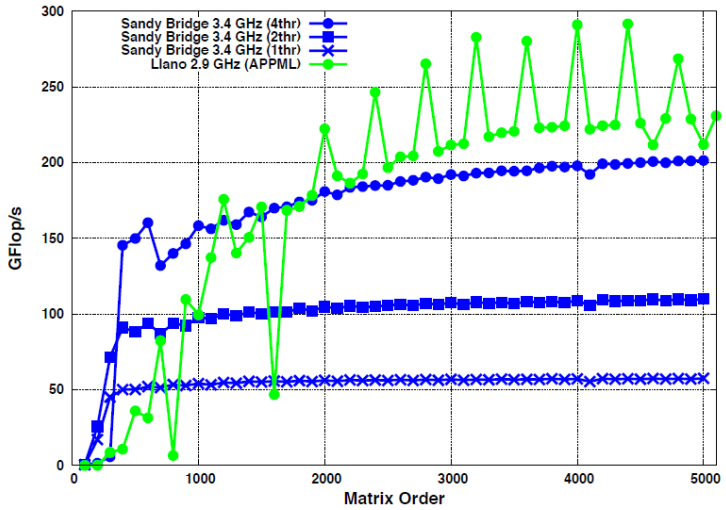
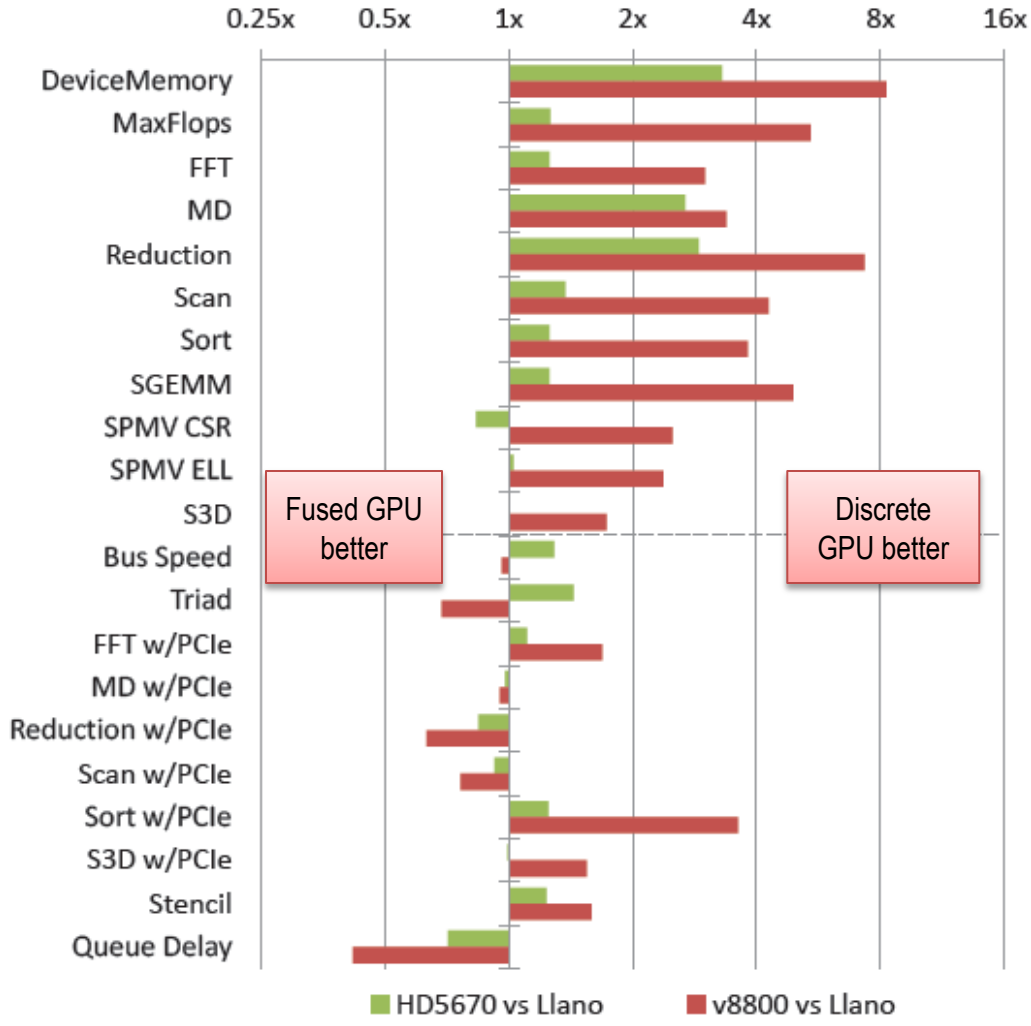
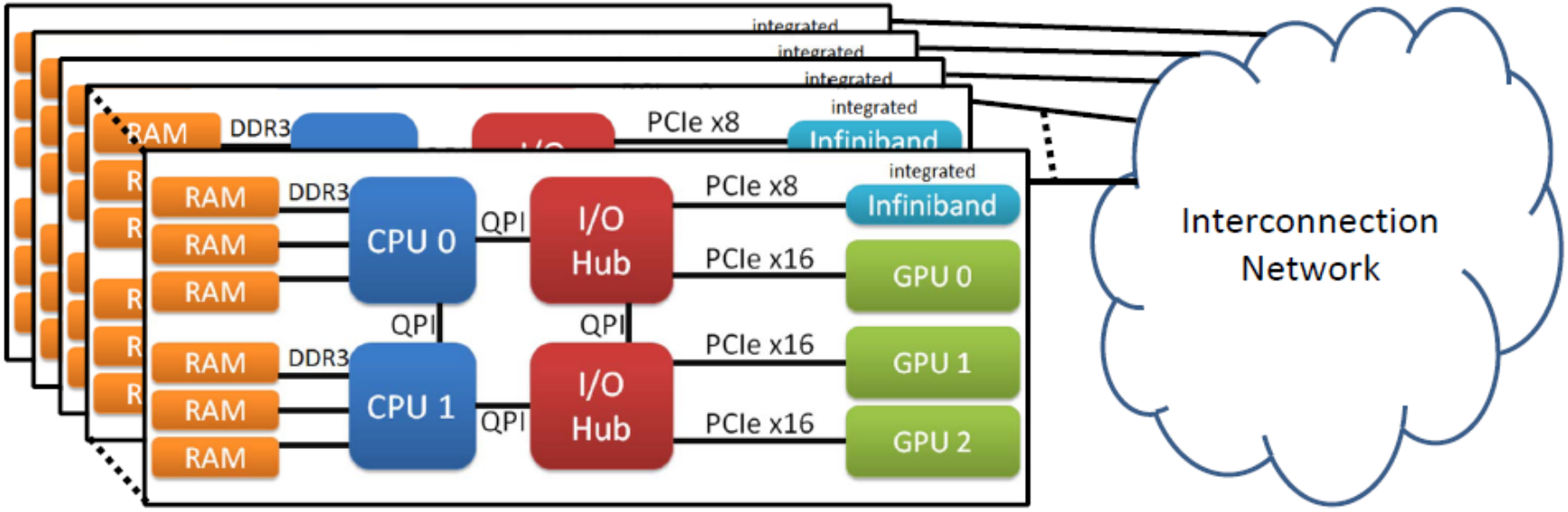


Figure 3: SGEMM Performance (one, two, and four CPU threads for Sandy Bridge and the OpenCL-based AMD APPML for Llano's fGPU)



K. Spafford, J.S. Meredith, S. Lee, D. Li, P.C. Roth, and J.S. Vetter, "The Tradeoffs of Fused Memory Hierarchies in Heterogeneous Architectures," in ACM Computing Frontiers (CF). Cagliari, Italy: ACM, 2012. Note: Both SB and Llano are consumer, not server, parts.

Applications must use a mix of programming models for these architectures



MPI

- Low overhead
- Resource contention
- Locality

OpenMP, Pthreads

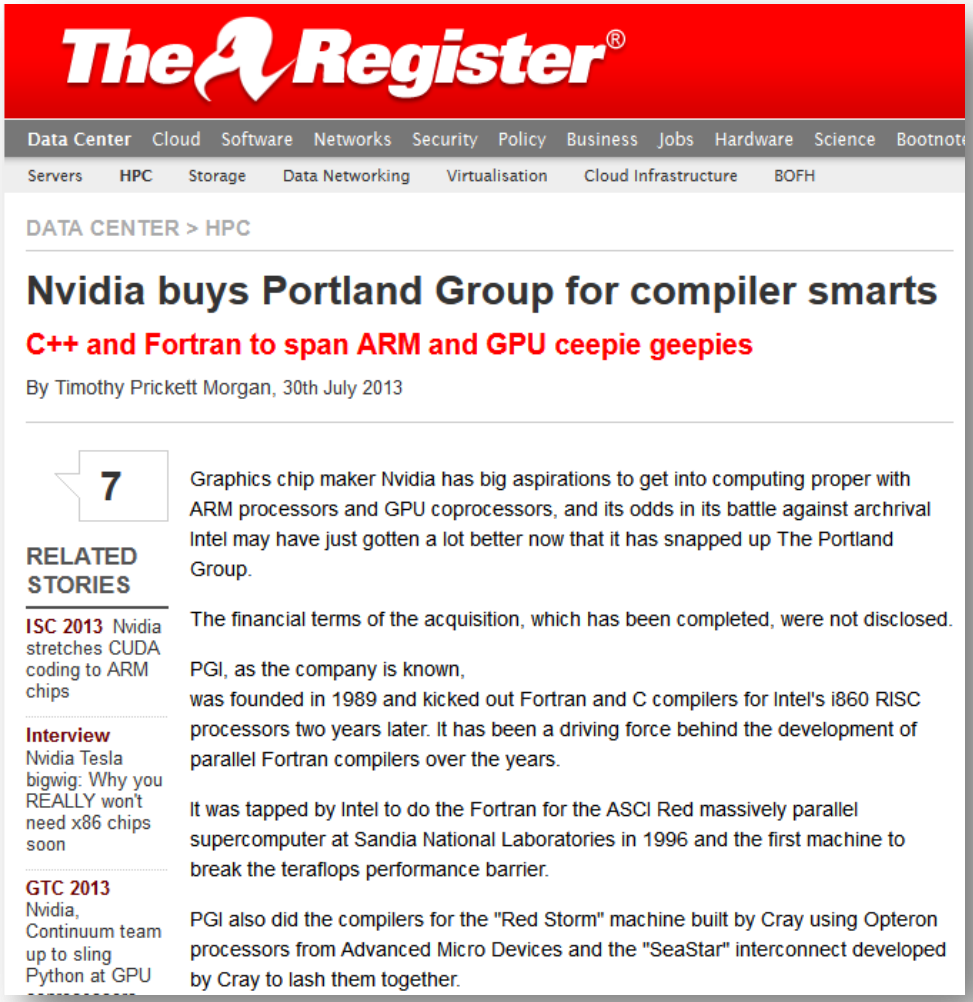
- SIMD
- NUMA

OpenACC, CUDA, OpenCL

- Memory use, coalescing
- Data orchestration
- Fine grained parallelism
- Hardware features

Critical Implications for Software, Apps, Developers

- Functional portability
- Performance portability
- Fast moving research, standards, products
- Incompatibilities among models
- Rewrite your code every 5 years
- Jobs!



The Register®

Data Center | Cloud | Software | Networks | Security | Policy | Business | Jobs | Hardware | Science | Bootnote

Servers | HPC | Storage | Data Networking | Virtualisation | Cloud Infrastructure | BOFH

DATA CENTER > HPC

Nvidia buys Portland Group for compiler smarts

C++ and Fortran to span ARM and GPU ceepie geepies

By Timothy Prickett Morgan, 30th July 2013

7

RELATED STORIES

ISC 2013 Nvidia stretches CUDA coding to ARM chips

Interview Nvidia Tesla bigwig: Why you REALLY won't need x86 chips soon

GTC 2013 Nvidia, Continuum team up to sling Python at GPU

Graphics chip maker Nvidia has big aspirations to get into computing proper with ARM processors and GPU coprocessors, and its odds in its battle against archrival Intel may have just gotten a lot better now that it has snapped up The Portland Group.

The financial terms of the acquisition, which has been completed, were not disclosed.

PGI, as the company is known, was founded in 1989 and kicked out Fortran and C compilers for Intel's i860 RISC processors two years later. It has been a driving force behind the development of parallel Fortran compilers over the years.

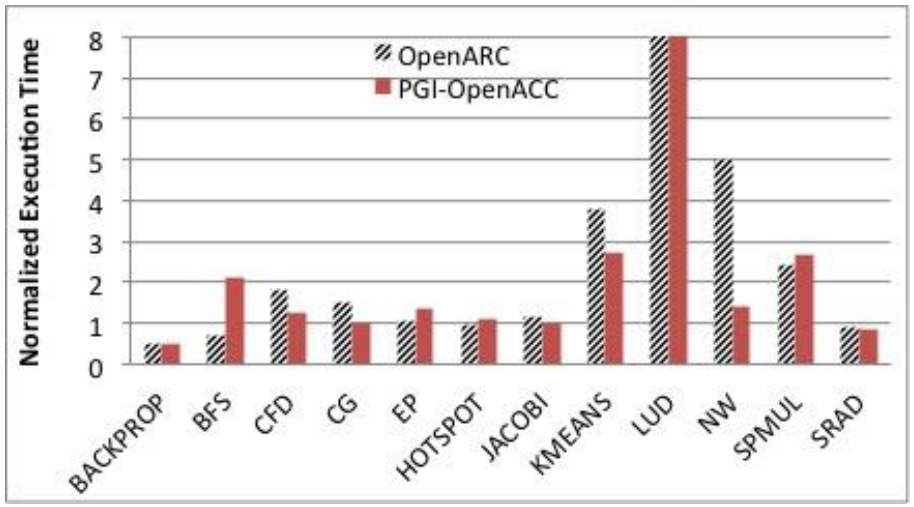
It was tapped by Intel to do the Fortran for the ASCI Red massively parallel supercomputer at Sandia National Laboratories in 1996 and the first machine to break the teraflops performance barrier.

PGI also did the compilers for the "Red Storm" machine built by Cray using Opteron processors from Advanced Micro Devices and the "SeaStar" interconnect developed by Cray to lash them together.

Jeffrey Vetter, ORNL
Seyong Lee, ORNL

OpenARC: Open Accelerator Research Compiler

- Problem
  - Directive-based GPU programming models provide abstraction over complex language syntax of low-level GPU programming and diverse architectural details. However, too much abstraction puts significant burdens on programmers regarding debugging and performance optimizations.
- Solution
  - OpenARC is *an open-sourced, very High-level Intermediate Representation (HIR)-based, extensible compiler framework*, where various performance optimizations, traceability mechanisms, fault tolerance techniques, etc., can be built for better debuggability/performance/resilience on the complex accelerator computing.
- Impact
  - OpenARC is the first open source compiler *supporting full OpenACC features*.
  - HIR with a rich set of directives in OpenARC provides a powerful research framework for *various source-to-source translation and instrumentation experiments*, even for porting Domain-Specific Languages (DSLs).
  - Additional OpenARC directives with its built-in tuning tools allow users to control overall OpenACC-to-GPU translation in a fine-grained, but still abstract manner.

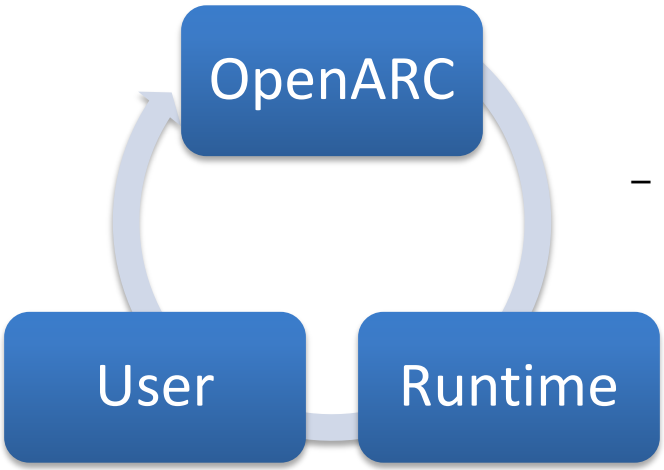


Performance of OpenARC and PGI-OpenACC compilers relative to manual CUDA versions (Lower is better.)

S. Lee and J.S. Vetter, "Early Evaluation of Directive-Based GPU Programming Models for Productive Exascale Computing," in SC12: ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis. Salt Lake City, Utah, USA: IEEE press, 2012.

Optimization and Interactive Program Verification with OpenARC

- Problem
  - *Too much abstraction* in directive-based GPU programming!
    - Debuggability
      - Difficult to diagnose logic errors and performance problems at the directive level
    - Performance Optimization
      - Difficult to find where and how to optimize
- Solution
  - Directive-based, interactive GPU program verification and optimization
    - OpenARC compiler:
      - Generates runtime codes necessary for *GPU-kernel verification* and *memory-transfer verification and optimization*.
    - Runtime
      - Locate trouble-making kernels by comparing execution results at kernel granularity.
      - Trace the runtime status of CPU-GPU coherence to detect incorrect/missing/redundant memory transfers.
    - Users
      - Iteratively fix/optimize incorrect kernels/memory transfers based on the runtime feedback and apply to input program.

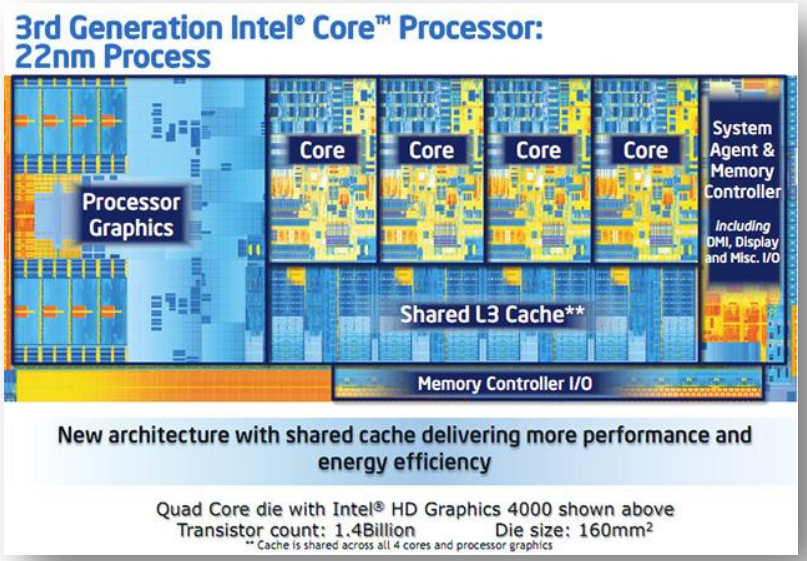


Iteratively find where and how to fix/optimize

Clause	Description
accglobal(list)	contains global symbols
accexplicitshared (list)	contains user-specified shared symbols
accreadonly(list)	contains R/O shared symbols
kernelConfPt (kernel)	indicates where to put kernel-configuration statements
gangconf(list)	contains sizes of each gang loop in nested gang loops
iterspace(exp)	contains iteration size of the loop

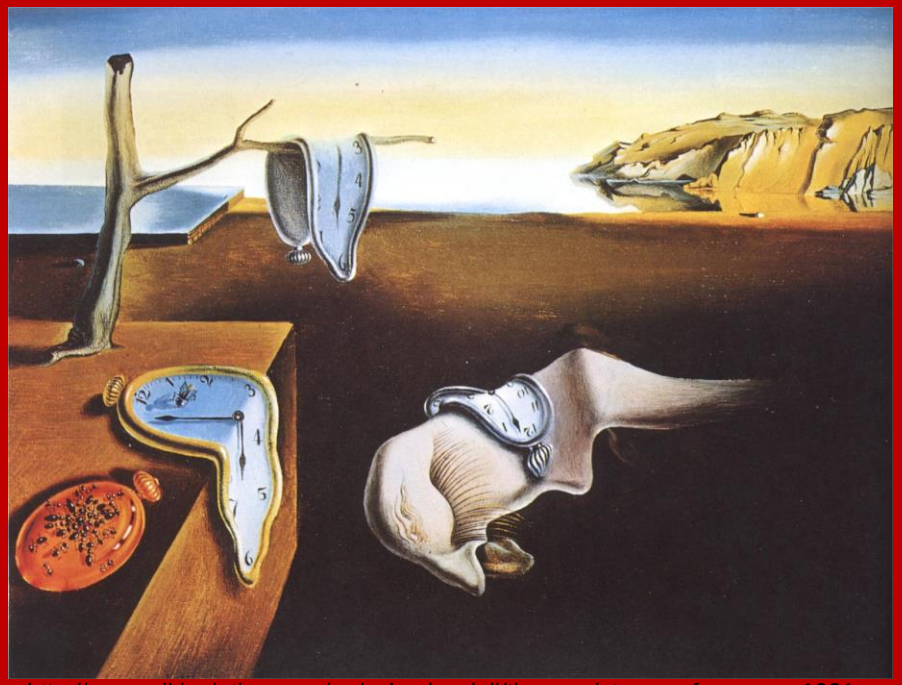
Future Directions in Heterogeneous Computing

- Over the next decade: Heterogeneous computing will continue to increase in importance
- Manycore
- Hardware features
  - Transactional memory
  - Random Number Generators
    - MC caveat
  - Scatter/Gather
  - Wider SIMD/AVX
- Synergies with BIGDATA, mobile markets, graphics
- Top 10 list of features to include *from application perspective. Now is the time!*



- Inform vendors about our priorities
- Inform applications teams to new features and gather their requirements

Memory Systems



The Persistence of Memory

http://www.wikipaintings.org/en/salvador-dali/the-persistence-of-memory-1931

Notional Exascale Architecture Targets

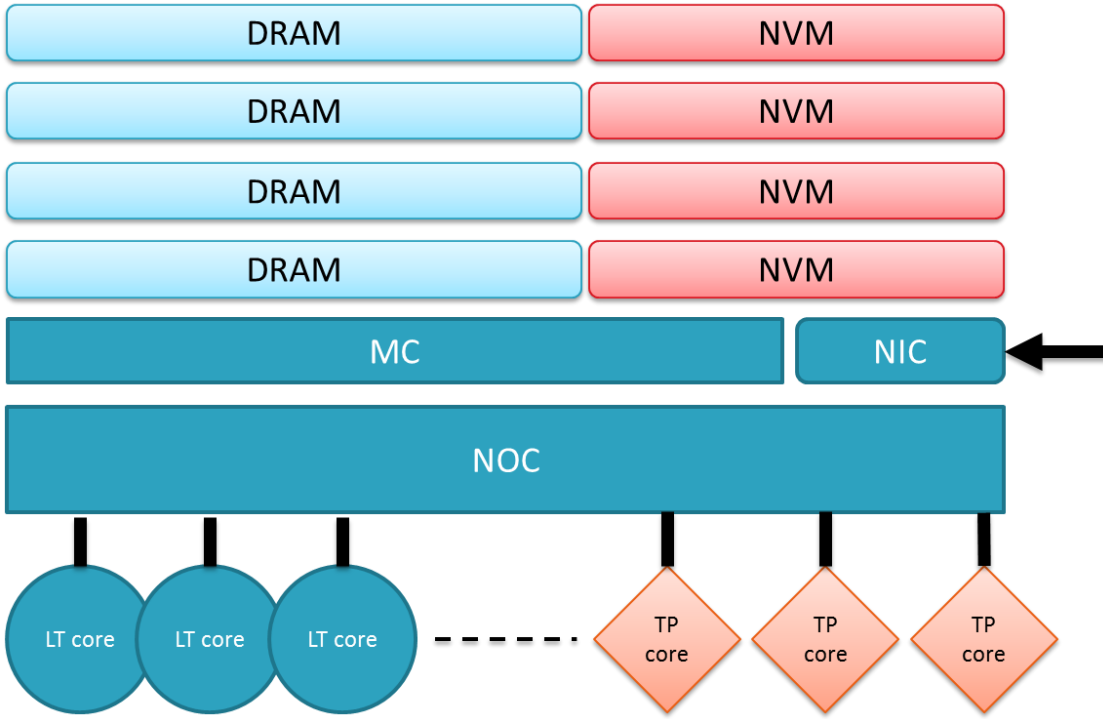
(From Exascale Arch Report 2009)

System attributes	2001	2010	"2015"		"2018"	
System peak	10 Tera	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	~0.8 MW	6 MW	15 MW		20 MW	
System memory	0.006 PB	0.3 PB	5 PB		32-64 PB	
Node performance	0.024 TF	0.125 TF	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	16	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	416	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW		1.5 GB/s	150 GB/sec	1 TB/sec	250 GB/sec	2 TB/sec
MTTI		day	O(1 day)		O(1 day)	

http://science.energy.gov/ascr/news-and-resources/workshops-and-conferences/grand-challenges/

Notional Future Node Architecture

- NVM to increase memory capacity
- Mix of cores to provide different capabilities
- Integrated network interface
- Very high bandwidth, low latency to on-package locales



Blackcomb: Hardware-Software Co-design for Non-Volatile Memory in Exascale Systems

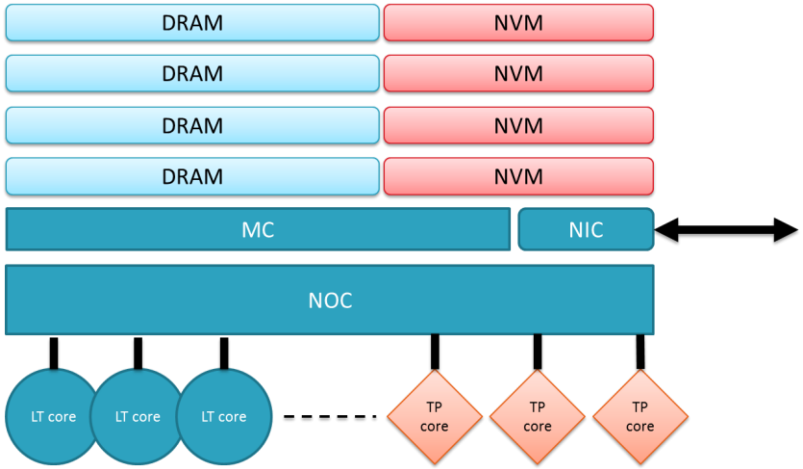
Jeffrey Vetter, ORNL
Robert Schreiber, HP Labs
Trevor Mudge, University of Michigan
Yuan Xie, Penn State University

http://ft.ornl.gov/trac/blackcomb

FWP #**ERKJU59**

Objectives

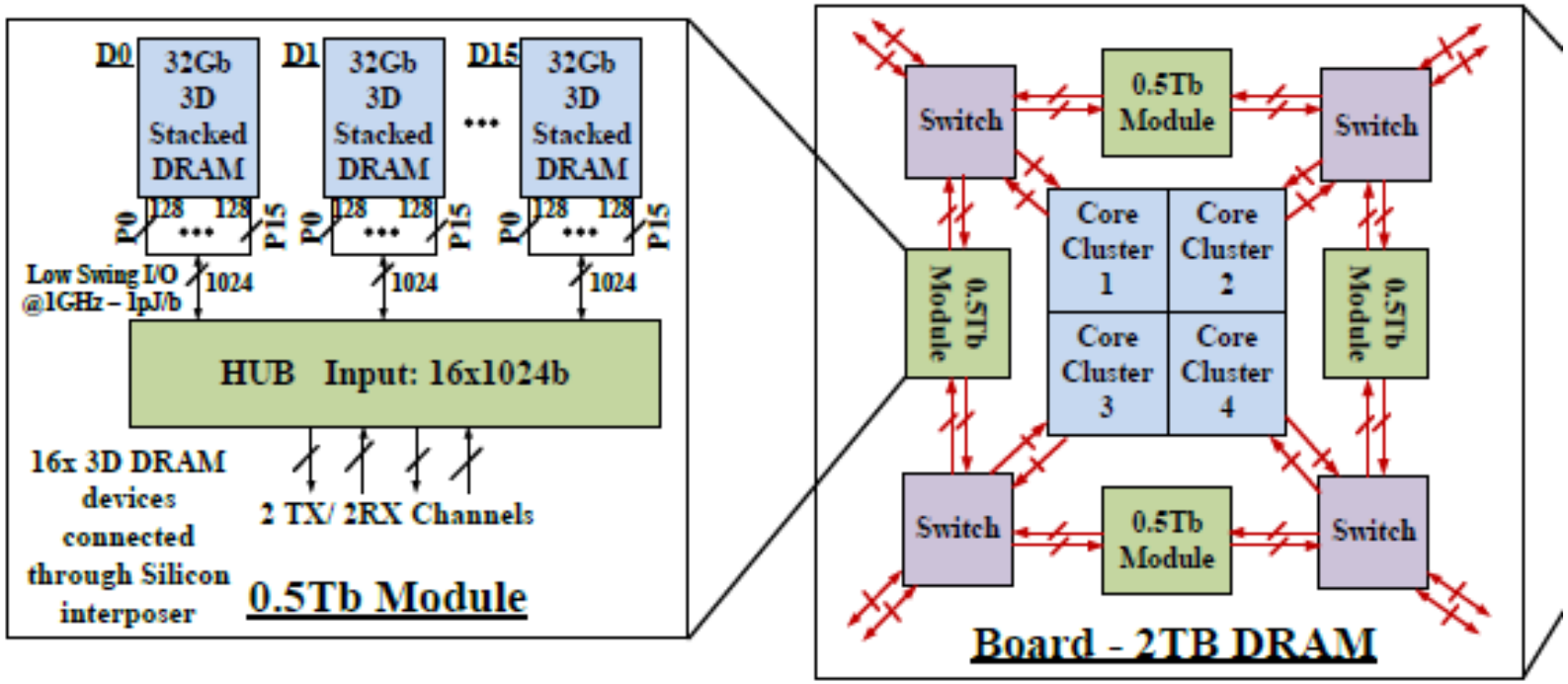
- Rearchitect servers and clusters, using nonvolatile memory (NVM) to overcome resilience, energy, and performance walls in exascale computing:
  - Ultrafast checkpointing to nearby NVM
  - Reoptimize the memory hierarchy for exascale, using new memory technologies
  - Replace disk with fast, low-power NVM
  - Enhance resilience and energy efficiency
  - Provide added memory capacity



Established and Emerging Memory Technologies – A Comparison

	SRAM	DRAM	eDRAM	NAND Flash	PCRAM	STTRAM	ReRAM (1T1R)	ReRAM (Xpoint)
Data Retention	N	N	N	Y	Y	Y	Y	Y
Cell Size (F^2)	50-200	4-6	19-26	2-5	4-10	8-40	6-20	1- 4
Read Time (ns)	< 1	30	5	10^4	10-50	10	5-10	50
Write Time (ns)	< 1	50	5	10^5	100-300	5-20	5-10	10-100
Number of Rewrites	10^{16}	10^{16}	10^{16}	10^4-10^5	10^8-10^{12}	10^{15}	10^8-10^{12}	10^6-10^{10}
Read Power	Low	Low	Low	High	Low	Low	Low	Medium
Write Power	Low	Low	Low	High	High	Medium	Medium	Medium
Power (other than R/W)	Leakage	Refresh	Refresh	None	None	None	None	Sneak

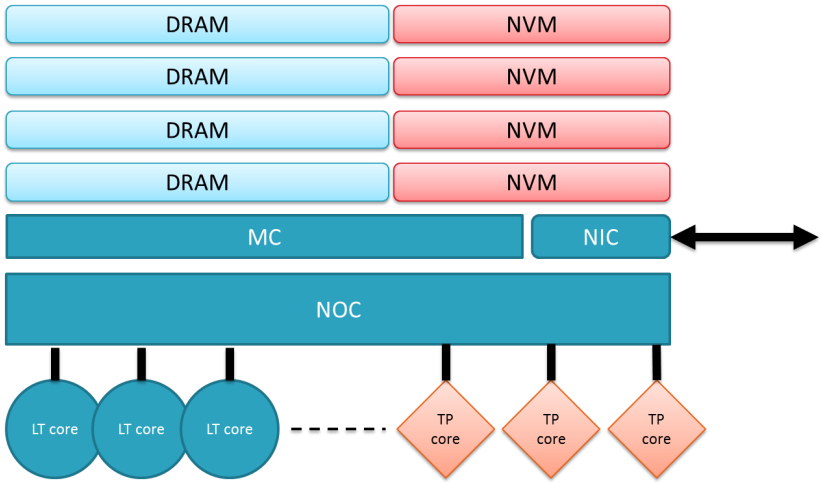
Tradeoffs in Exascale Memory Architectures

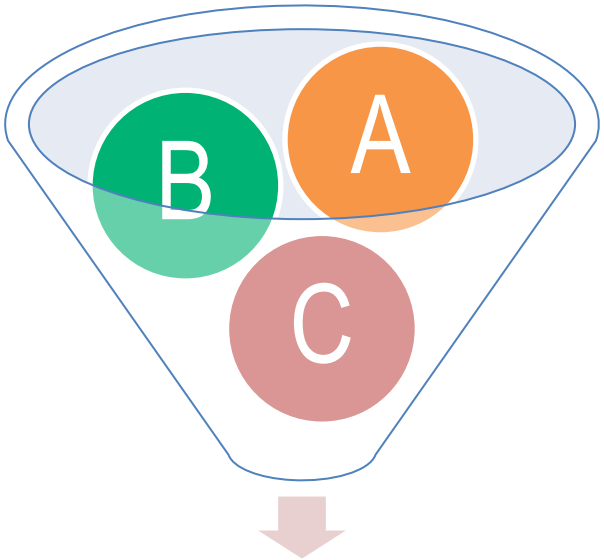


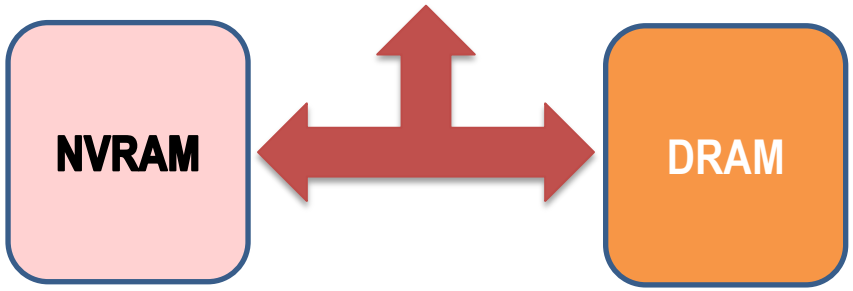
- Understanding the tradeoffs
  - ECC type, row buffers, DRAM physical page size, bitline length, etc

Blackcomb team, "Optimizing DRAM Architectures for Energy-Efficient, Resilient Exascale Memories," (*to appear) SC13, 2013*

New hybrid memory architectures: What is the ideal organizations for our applications?

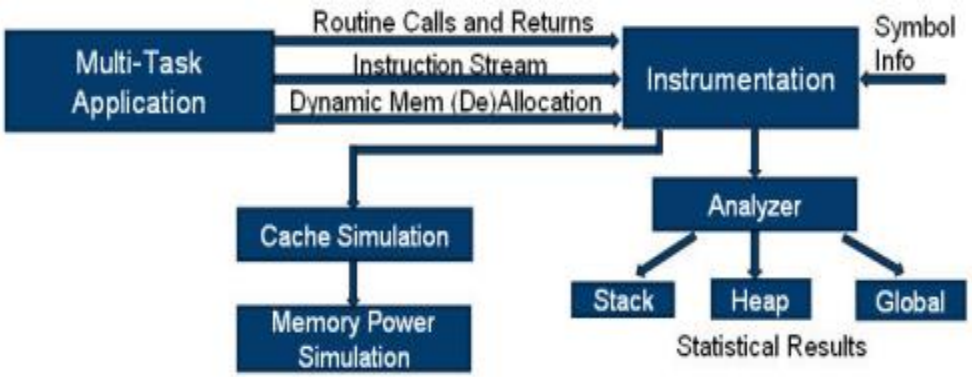






Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications

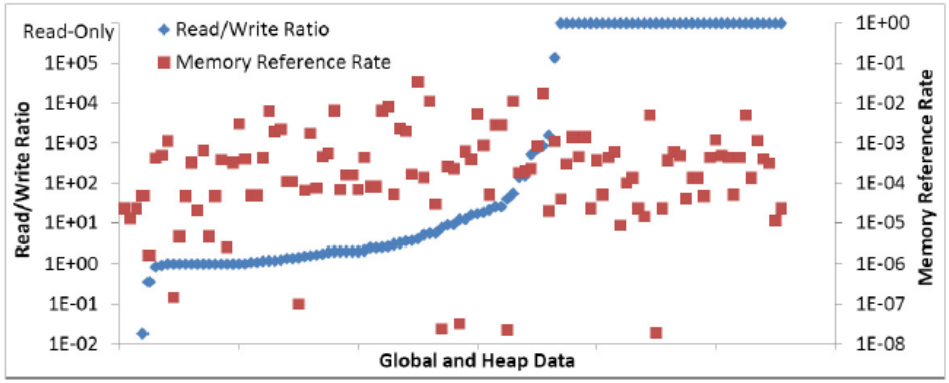
- **Problem**
  - Do specific memory workload characteristics of scientific apps map well onto NVRAMs' features?
  - Can NVRAM be used as a solution for future Exascale systems?



- **Solution**
  - Develop a binary instrumentation tool to investigate memory access patterns related to NVRAM
  - Study realistic DOE applications (Nek5000, S3D, CAM and GTC) at fine granularity
- **Impact**
  - Identify large amount of commonly existing data structures that can be placed in NVRAM to save energy
  - Identify many NVRAM-friendly memory access patterns in DOE applications
  - Received attention from both vendor and apps teams

D. Li, J.S. Vetter, G. Marin, C. McCurdy, C. Cira, Z. Liu, and W. Yu, "Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications," in *IEEE International Parallel & Distributed Processing Symposium (IPDPS). Shanghai: IEEEE, 2012*

Measurement Results



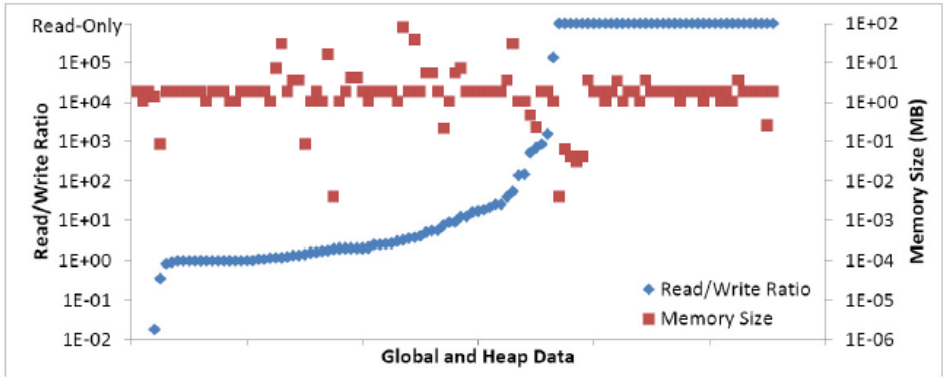
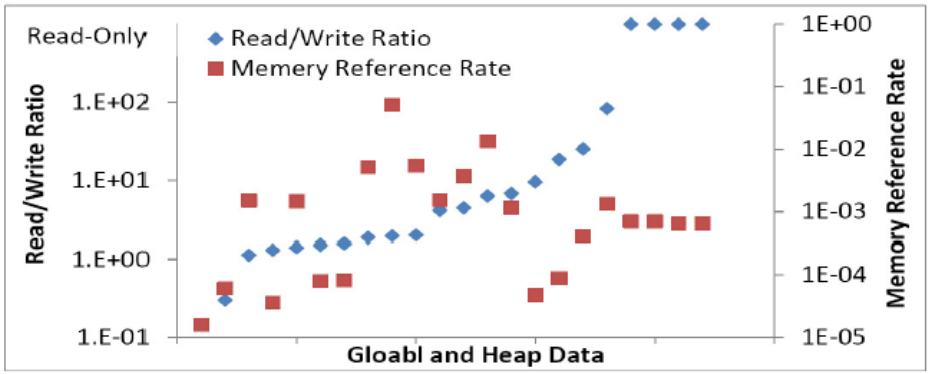


Figure 3: Read/write ratios, memory reference rates and memory object sizes for memory objects in Nek5000



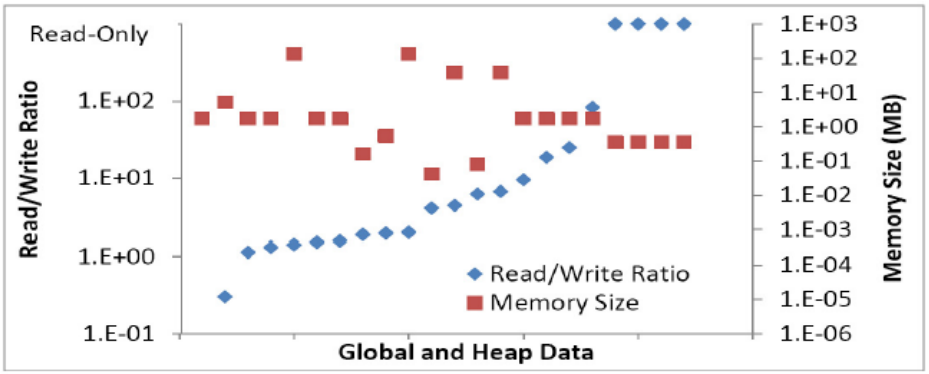
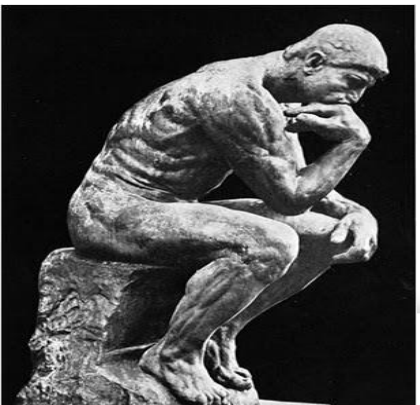
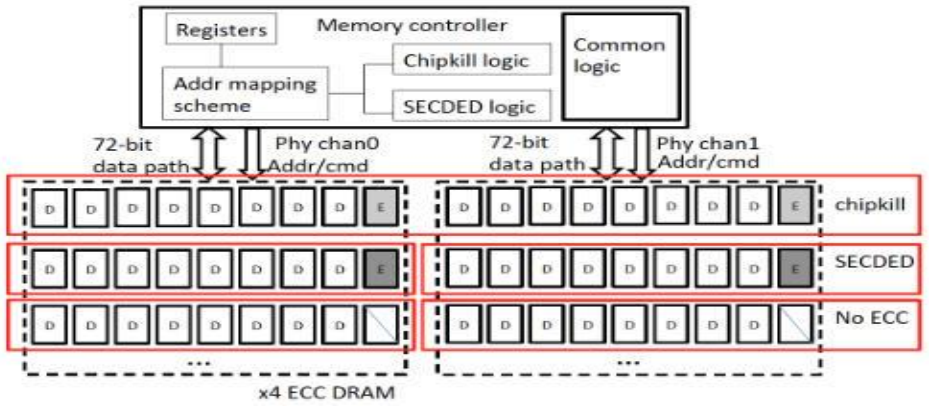


Figure 6: Read/write ratios, memory reference rates and memory object sizes for memory objects in S3D

Rethinking Algorithm-Based Fault Tolerance

- Algorithm-based fault tolerance (ABFT) has many attractive characteristics
  - Can reduce or even eliminate the expensive periodic checkpoint/rollback
  - Can bring negligible performance loss when deployed in large scale
  - No modifications from architecture and system software
- However
  - ABFT is completely opaque to any underlying hardware resilience mechanisms
  - These hardware resilience mechanisms are also unaware of ABFT
  - Some data structures are over-protected by ABFT and hardware





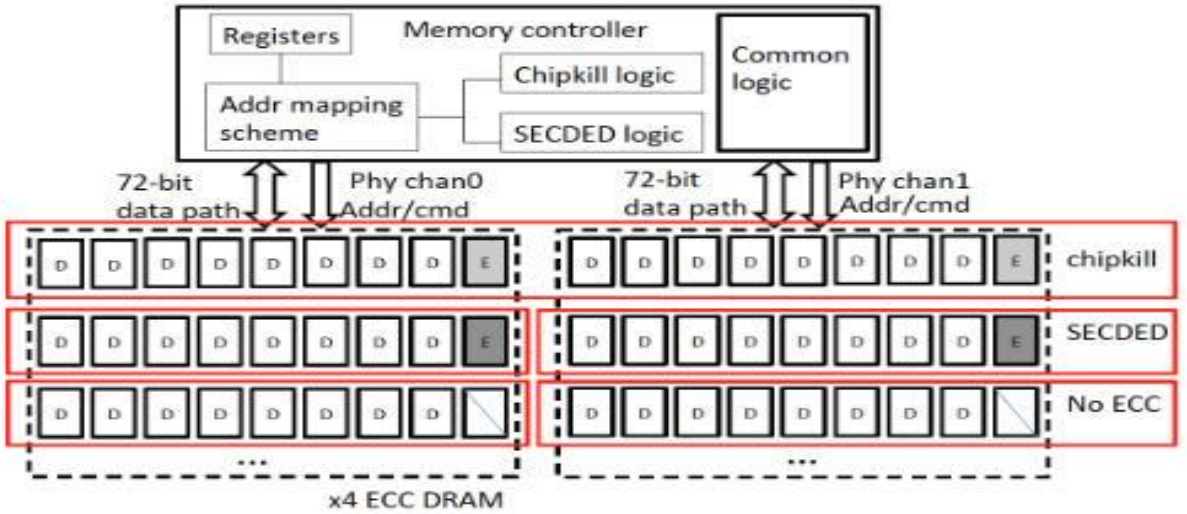
D. Li, C. Zizhong, W. Panruo, and S. Vetter Jeffrey, "Rethinking Algorithm-Based Fault Tolerance with a Cooperative Software-Hardware Approach," Proc. International Conference for High Performance Computing, Networking, Storage and Analysis (SC13), 2013, pp. (to appear),

We consider ABFT using a holistic view from both software and hardware

- We investigate how to integrate ABFT and hardware-based ECC for main memory
- ECC brings energy, performance and storage overhead
- The current ECC mechanisms cannot work
  - There is a significant semantic gap for error detection and location between ECC protection and ABFT
- We propose an explicitly-managed ECC by ABFT
  - A cooperative software-hardware approach
  - We propose customization of memory resilience mechanisms based on algorithm requirements.

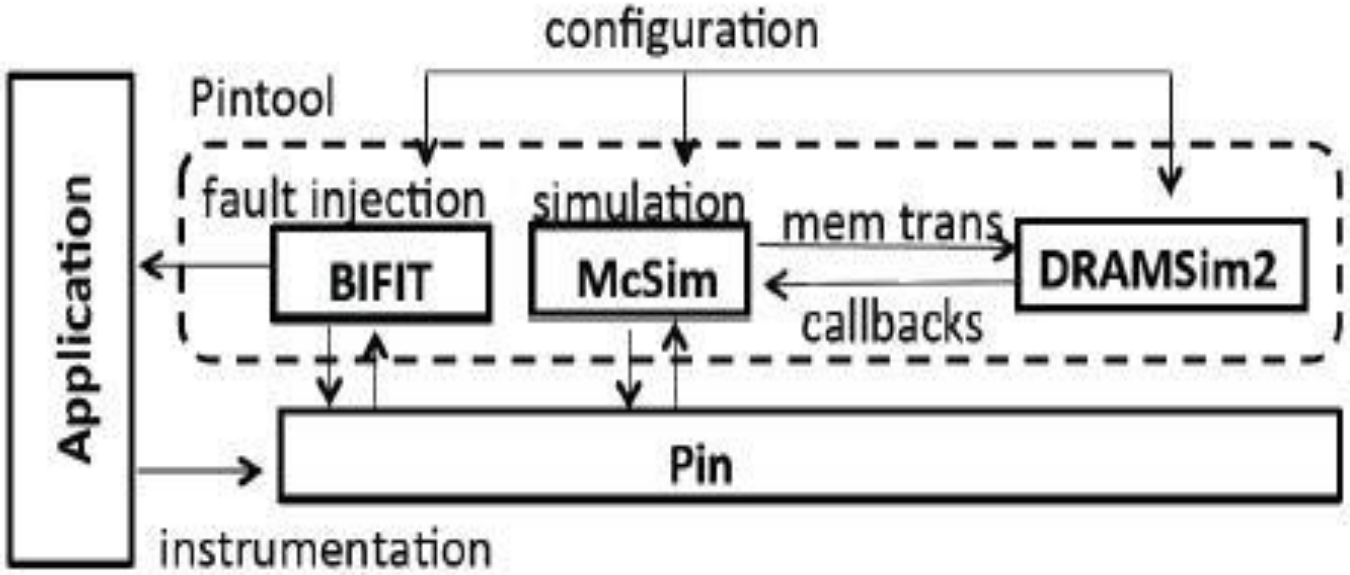
System Designs

- Architecture
  - Enable co-existence of multiple ECC
  - Introduce a set of ECC registers into the memory controller (MC)
  - MC is in charge of detecting, locating, and reporting errors
- Software
  - The users control which data structures should be protected by which relaxed ECC scheme by ECC control APIs.
  - ABFT can simplify its verification phase, because hardware and OS can explicitly locate corrupted data



Evaluation

- We use four ABFT (FT-DGEMM, FT-Cholesky, FT-CG and FT-HPL)



- We save up to 25% for system energy (and up to 40% for dynamic memory energy) with up to 18% performance improvement

Future Directions in Next Generation Memory

- Next decade will also be exciting for memory technology
- New devices
  - Flash, ReRam, STTRAM will challenge DRAM
  - Commercial markets already driving transition
- New configurations
  - 2.5D, 3D stacking removes recent JEDEC constraints
  - Storage paradigms (e.g., key-value)
  - Opportunities to rethink memory organization
- Logic/memory integration
  - Move compute to data
  - Programming models

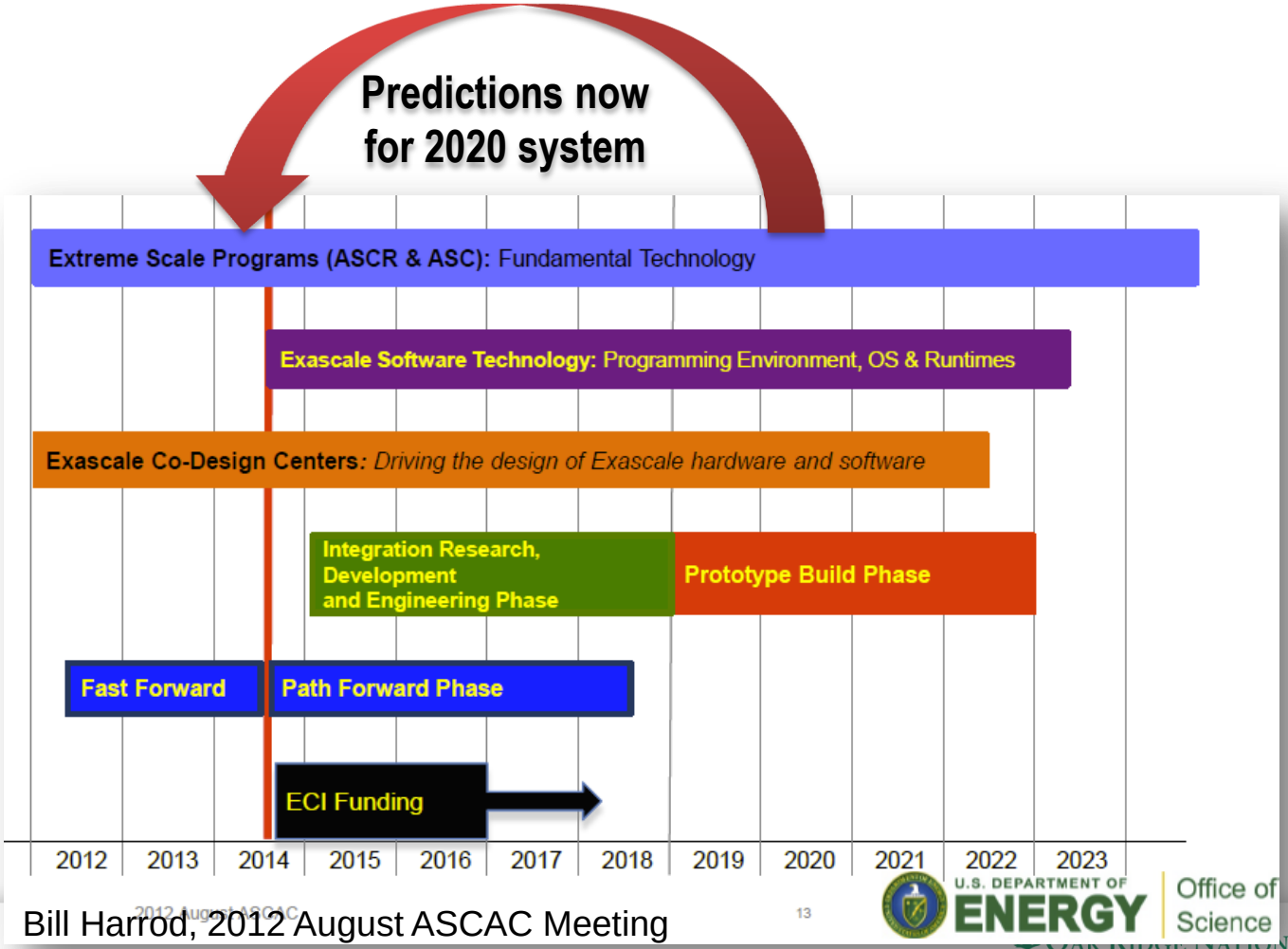
	SRAM	DRAM	eDRAM	NAND Flash	PCRAM	STTRAM	ReRAM (1T1R)	ReRAM (Xpoint)
Data Retention	N	N	N	Y	Y	Y	Y	Y
Cell Size (F^2)	50-200	4-6	19-26	2-5	4-10	8-40	6-20	1- 4
Read Time (ns)	< 1	30	5	10^4	10-50	10	5-10	50
Write Time (ns)	< 1	50	5	10^5	100-300	5-20	5-10	10-100
Number of Rewrites	10^{16}	10^{16}	10^{16}	10^4-10^5	10^8-10^{12}	10^{15}	10^8-10^{12}	10^6-10^{10}
Read Power	Low	Low	Low	High	Low	Low	Low	Medium
Write Power	Low	Low	Low	High	High	Medium	Medium	Medium
Power (other than R/W)	Leakage	Refresh	Refresh	None	None	None	None	Sneak

- Refactor our applications to make use of this new technology
- Add HPC programming support for these new technologies
- Explore opportunities for improved resilience, power, performance

Co-designing Future Extreme Scale Systems

Predictive Performance

- Empirical measurement is necessary but we must investigate future applications on future architectures using future software stacks



Bill Harrod, 2012 August ASCAC Meeting

Holistic View of HPC

Applications	Programming Environment	System Software	Architectures

• Materials
• Climate
• Fusion
• National Security
• Combustion
• Nuclear Energy
• Cybersecurity
• Biology
• High Energy Physics
• Energy Storage
• Photovoltaics
• National Competitiveness
 	
• Usage Scenarios
  • Ensembles
  • UQ
  • Visualization
  • Analytics
	• Domain specific
	  • Libraries
	  • Frameworks
	  • Templates
	  • Domain specific languages
	  • Patterns
	  • Autotuners
	
	• Platform specific
	  • Languages
	  • Compilers
	  • Interpreters/Scripting
	  • Performance and Correctness Tools
	  • Source code control
		• Resource Allocation
		• Scheduling
		• Security
		• Communication
		• Synchronization
		• Filesystems
		• Instrumentation
		• Virtualization
			• Processors
			  • Multicore
			  • Graphics Processors
			  • Vector processors
			  • FPGA
			  • DSP
			• Memory and Storage
			  • Shared (cc, scratchpad)
			  • Distributed
			  • RAM
			  • Storage Class Memory
			  • Disk
			  • Archival
			• Interconnects
			  • Infiniband
			  • IBM Torrent
			  • Cray Gemini, Aires
			  • BGL/P/Q
			  • 1/10/100 GigE

Holistic View of HPC – Past 15 years

Applications	Programming Environment	System Software	Architectures

• Materials
• Climate
• Fusion
• National Security
• Combustion
• Nuclear Energy
• Cybersecurity
• Biology
• High Energy Physics
• Energy Storage
• Photovoltaics
• National Competitiveness
 	
• Usage Scenarios
  • Ensembles
  • UQ
  • Visualization
  • Analytics
	• Domain specific
	  • Libraries
	  • Frameworks
	  • Templates
	  • Domain specific languages
	  • Patterns
	  • Autotuners
	
	• Platform specific
	  • Languages
	  • Compilers
	  • Interpreters/Scripting
	  • Performance and Correctness Tools
	  • Source code control
		• Resource Allocation
		• Scheduling
		• Security
		• Communication
		• Synchronization
		• Filesystems
		• Instrumentation
		• Virtualization
			• Processors
			  • Multicore
			  • Graphics Processors
			  • Vector processors
			  • FPGA
			  • DSP
			• Memory and Storage
			  • Shared (cc, scratchpad)
			  • Distributed
			  • RAM
			  • Storage Class Memory
			  • Disk
			  • Archival
			• Interconnects
			  • Infiniband
			  • IBM Torrent
			  • Cray Gemini, Aires
			  • BGL/P/Q
			  • 1/10/100 GigE

Holistic View of HPC – Going Forward
Many more technologies, programming models

Applications	Programming Environment	System Software	Architectures

• Materials
• Climate
• Fusion
• National Security
• Combustion
• Nuclear Energy
• Cybersecurity
• Biology
• High Energy Physics
• Energy Storage
• Photovoltaics
• National Competitiveness
 	
• Usage Scenarios
  • Ensembles
  • UQ
  • Visualization
  • Analytics
	• Domain specific
	  • Libraries
	  • Frameworks
	  • Templates
	  • Domain specific languages
	  • Patterns
	  • Autotuners
	• Platform specific
	  • Languages
	  • Compilers
	  • Interpreters/Scripting
	  • Performance and Correctness Tools
	  • Source code control
		• Resource Allocation
		• Scheduling
		• Security
		• Communication
		• Synchronization
		• Filesystems
		• Instrumentation
		• Virtualization
			• Processors
			  • Multicore
			  • Graphics Processors
			  • Vector processors
			  • FPGA
			  • DSP
			• Memory and Storage
			  • Shared (cc, scratchpad)
			  • Distributed
			  • RAM
			  • Storage Class Memory
			  • Disk
			  • Archival
			• Interconnects
			  • Infiniband
			  • IBM Torrent
			  • Cray Gemini, Aires
			  • BGL/P/Q
			  • 1/10/100 GigE

Holistic View of HPC – Going Forward
Large design space –> uncertainty!

Performance, Resilience, Power, Programmability

Applications	Programming Environment	System Software	Architectures

• Materials
• Climate
• Fusion
• National Security
• Combustion
• Nuclear Energy
• Cybersecurity
• Biology
• High Energy Physics
• Energy Storage
• Photovoltaics
• National Competitiveness
 	
• Usage Scenarios
  • Ensembles
  • UQ
  • Visualization
  • Analytics
	• Domain specific
	  • Libraries
	  • Frameworks
	  • Templates
	  • Domain specific languages
	  • Patterns
	  • Autotuners
	• Platf
	  • Langu
	  • Compilers
	  • Int
	  • Pe
	  Correct
	  • Source
		• Resource Allocation
		• Scheduling
		• Security
		• Comm nication
		• Sy nization
		• ems
		• ent
			• Processors
			  • Multicore
			  • Graphics Processors
			  • Vector processors
			  • FPGA
			  • DSP
			• Memory and Storage
			  • Shared (cc, scratchpad)
			  • Distributed
			  • RAM
			  • Storage Class Memory
			  • Disk
			  • Archival
			• Interconnects
			  • Infiniband
			  • IBM Torrent
			  • Cray Gemini, Aires
			  • BGL/P/Q
			  • 1/10/100 GigE

Large design space is challenging for apps, software, and architecture scientists.

Slide courtesy of Karen Pao, DOE

U.S. DEPARTMENT OF ENERGY

Three Exascale Co-Design Centers selected after intense competition

Exascale Co-Design Center for Materials in Extreme Environments (ExMatEx)

Director: Timothy Germann (LANL)

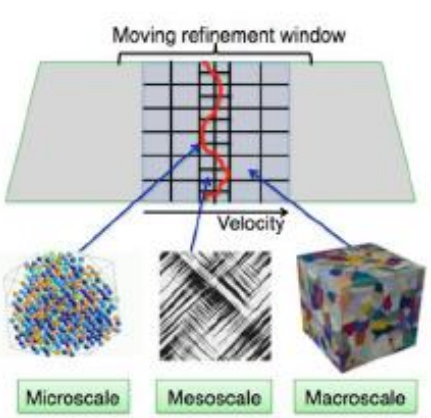
Center for Exascale Simulation of Advanced Reactors (CESAR)

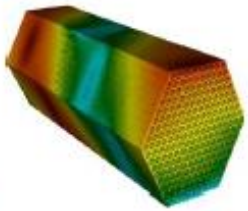
Director: Robert Rosner (ANL)

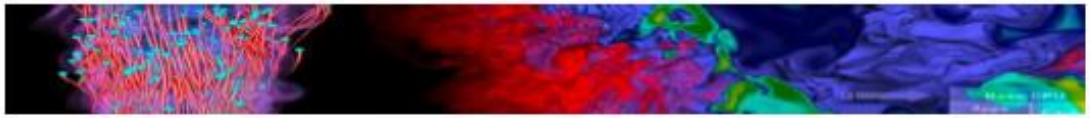
Center for Exascale Simulation of Combustion in Turbulence (EXaCT)

Director: Jacqueline Chen (SNL)

	ExMatEx (Germann)	CESAR (Rosner)	EXaCT (Chen)
National Labs	LANL	ANL	SNL
	LLNL	PNNL	LBNL
	SNL	LANL	LANL
	ORNL	ORNL	ORNL
		LLNL	LLNL
			NREL
University & Industry Partners	Stanford	MIT	Stanford
	CalTech	TAMU	GA Tech
		Rice	Rutgers
		U Chicago	UT Austin
		IBM	Utah
		TerraPower	
		General Atomic	
		Areva	







Each project is $4M/yr for 5 years, subject to satisfactory progress as gauged by frequent reviews

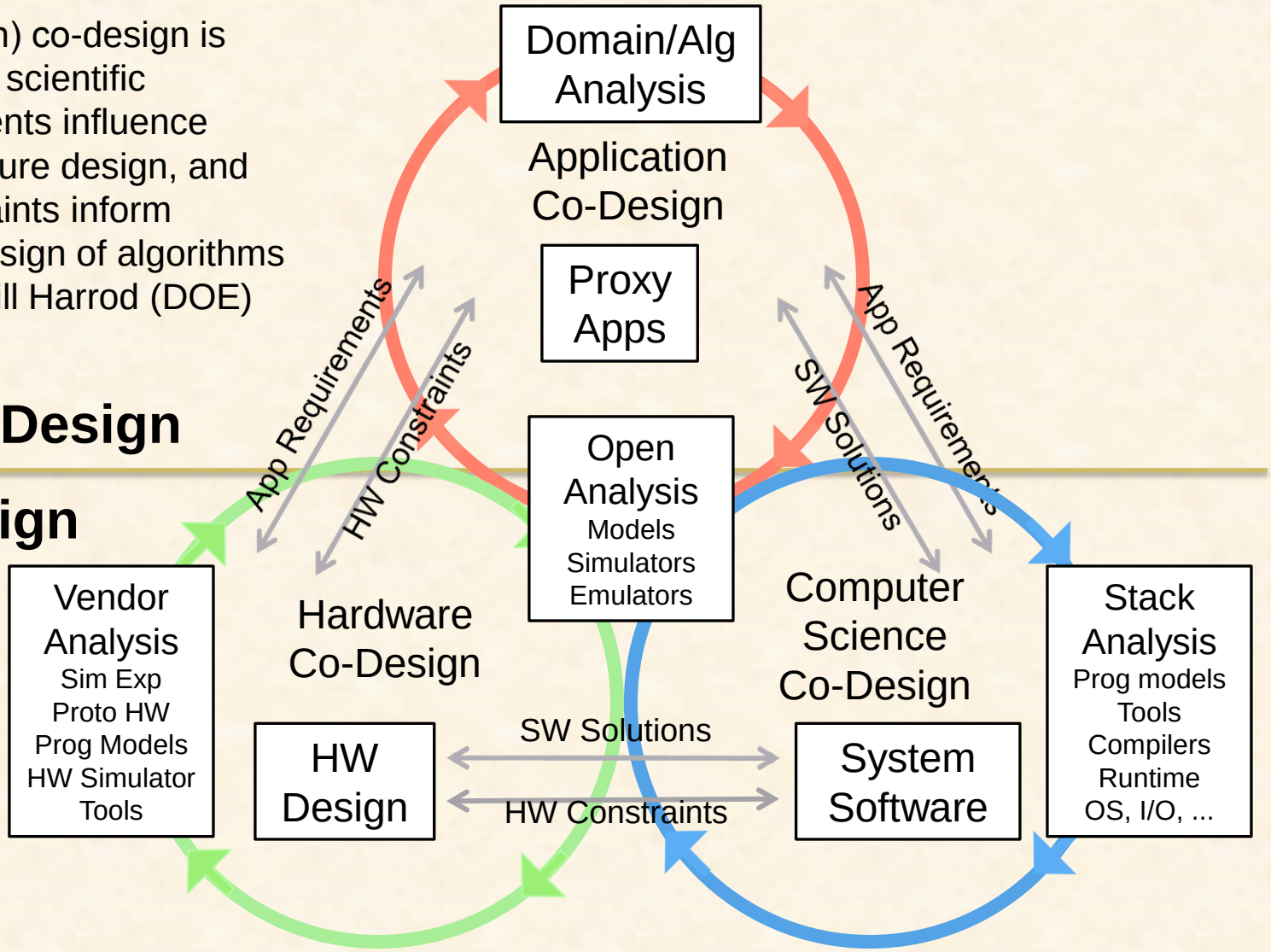
Slide courtesy of ExMatEx Co-design team.

Workflow within the Exascale Ecosystem

"(Application driven) co-design is the process where scientific problem requirements influence computer architecture design, and technology constraints inform formulation and design of algorithms and software." – Bill Harrod (DOE)

Application Design

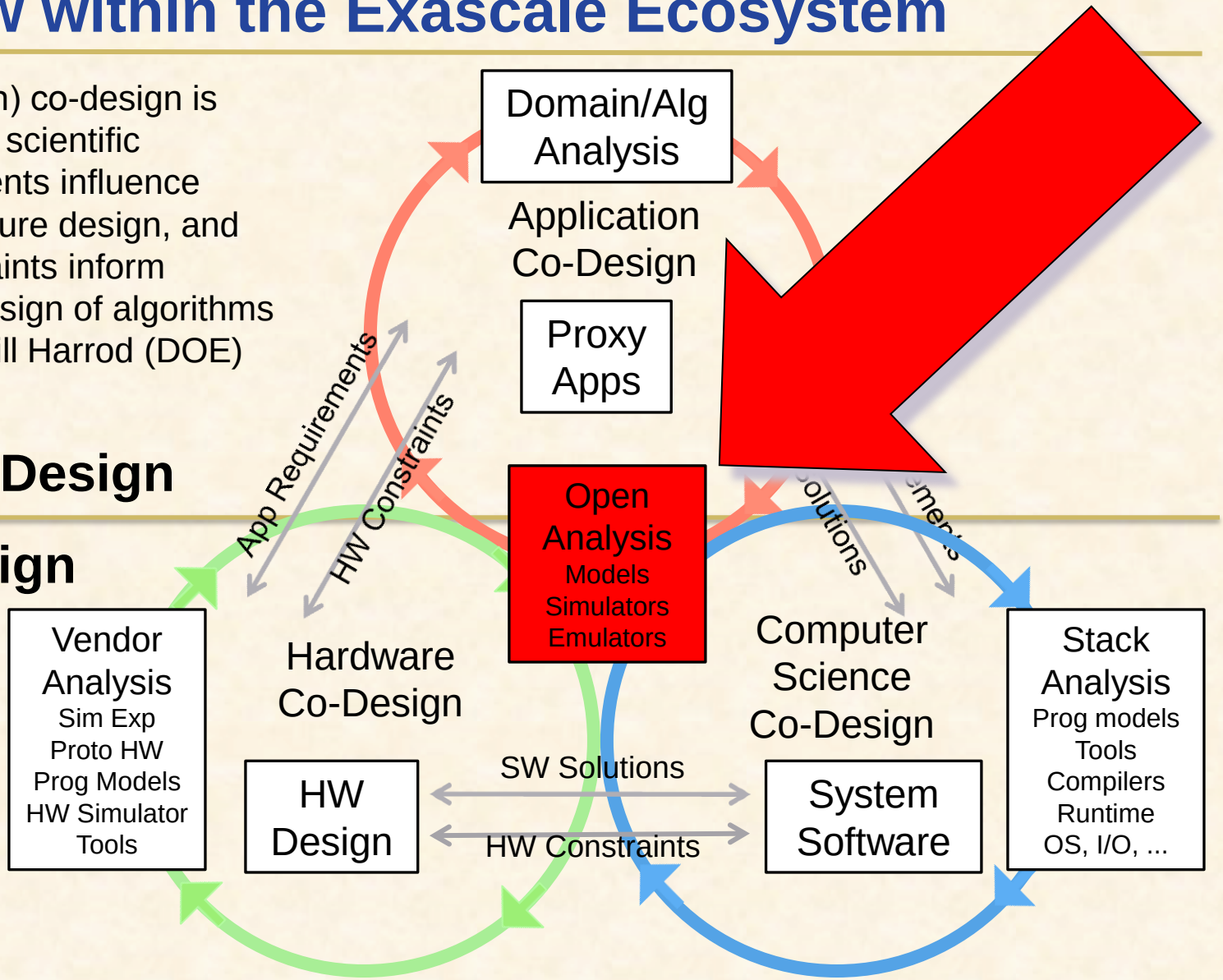
System Design



Slide courtesy of ExMatEx Co-design team.

Workflow within the Exascale Ecosystem

"(Application driven) co-design is the process where scientific problem requirements influence computer architecture design, and technology constraints inform formulation and design of algorithms and software." – Bill Harrod (DOE)



Prediction Techniques Ranked

	Speed	Ease	Flexibility	Accuracy	Scalability
Ad-hoc Analytical Models	1	3	2	4	1
Structured Analytical Models	1	2	1	4	1
Simulation – Functional	3	2	2	3	3
Simulation – Cycle Accurate	4	2	2	2	4
Hardware Emulation (FPGA)	3	3	3	2	3
Similar hardware measurement	2	1	4	2	2
Node Prototype	2	1	4	1	4
Prototype at Scale	2	1	4	1	2
Final System	-	-	-	-	-

Prediction Techniques Ranked

	Speed	Ease	Flexibility	Accuracy	Scalability
Ad-hoc Analytical Models	1	3	2	4	1
Structured Analytical Models	1	2	1	4	1
Aspen	1	1	1	4	1
Simulation – Functional	3	2	2	3	3
Simulation – Cycle Accurate	4	2	2	2	4
Hardware Emulation (FPGA)	3	3	3	2	3
Similar hardware measurement	2	1	4	2	2
Node Prototype	2	1	4	1	4
Prototype at Scale	2	1	4	1	2
Final System	-	-	-	-	-

Aspen – Design Goals

- Abstract Scalable Performance Engineering Notation
  - Create a deployable, extensible, and highly semantic representation for analytical performance models
  - Design and implement a new language for analytical performance modeling
  - Use the language to create machine-independent models for important applications and kernels
- Models are composable

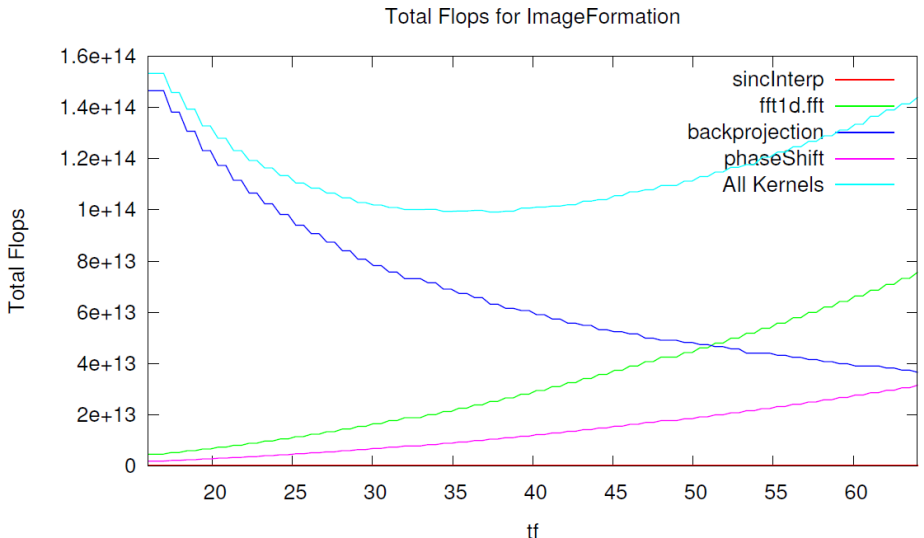
```
1 kernel localFFT {
2   exposes parallelism [n^2]
3   requires flops [5 * n * log2(n)] as dp,
        complex, simd
4   requires loads [a * n * max(1, log(n)/
        log(Z)) * wordSize] from fftVolume
5 }
```

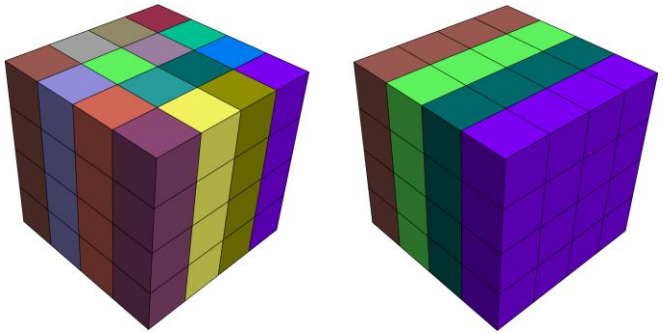
Listing 2. Aspen statements for the local 1D FFTs

K. Spafford and J.S. Vetter, "Aspen: A Domain Specific Language for Performance Modeling," in *SC12: ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis, 2012*

Aspen – Design Goals (2)

- Develop a suite of analysis tools which operate on the models and produce key performance metrics like
  - available parallelism, memory capacity, arithmetic intensity, and message volume
- Not goals for Aspen
  - Replace simulators by making highly detailed models
  - Solve all the problems of analytical modeling
    - Cache policies
    - Network contention
- *Constructed models for important apps and mini-apps: MD, UHPC CP 1, Lulesh, 3D FFT, CoMD, VPFFT, …*





Aspen Model Walkthrough: 3DFFT

Pencil v. Slab Decomposition

Based on earlier analytical models by 1/ Gahvari and 2/ Czechowski

3DFFT

```
// Dimension of cubic 3D Volume
param n = 8192
param a = 6.3
param wordSize = 16 // Double Complex Words
param dataPerProc = (n^3 * wordSize) / P
data fftVolume [n^3 * wordSize]
```

```
control pencil {
  localFFT -> transpose // in X
  exchange
  localFFT -> transpose // in Y
  exchange
  localFFT -> transpose // in Z
}
```

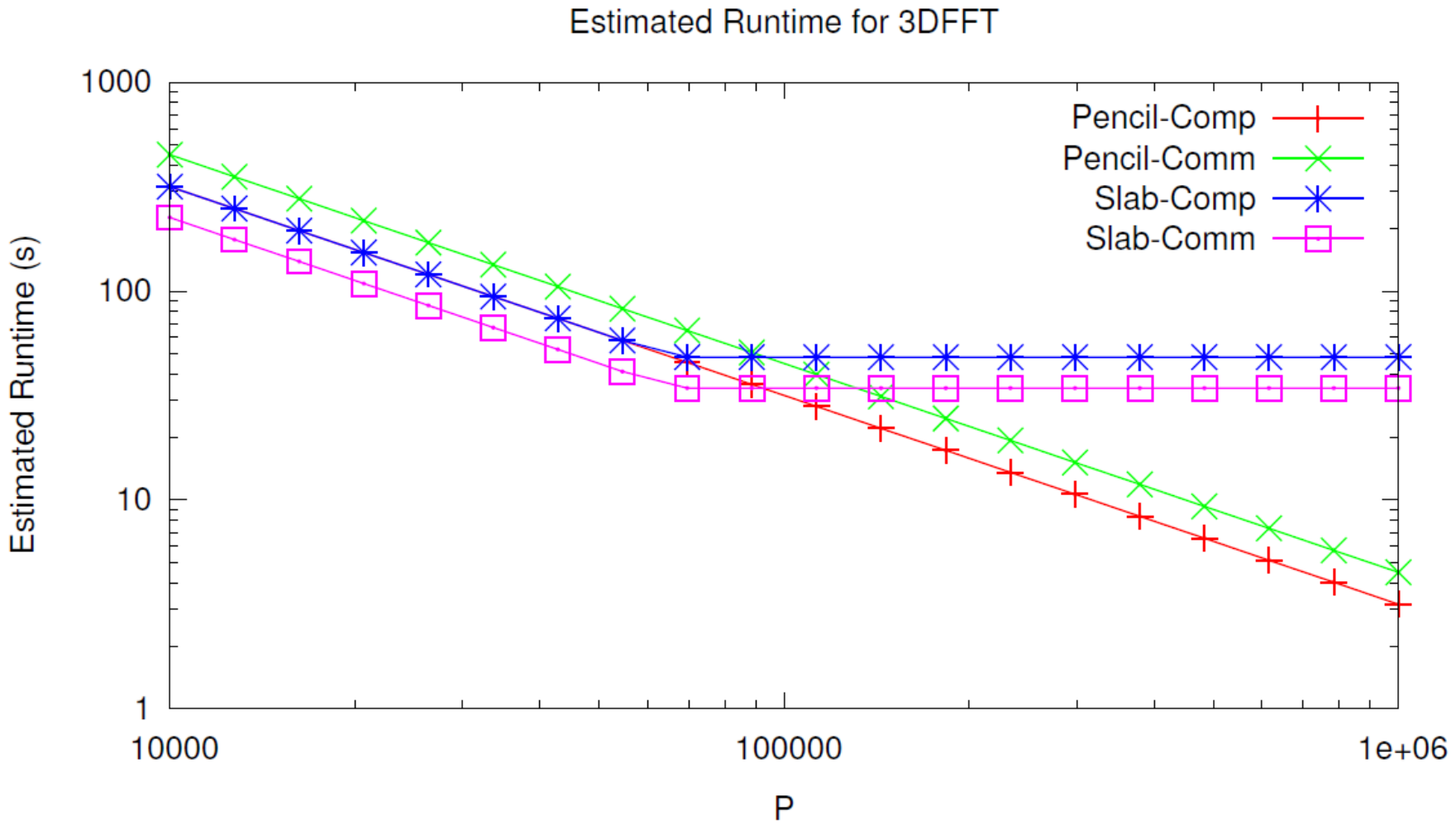
```
control slab {
  localFFT -> transpose // in X
  localFFT -> transpose // in Y
  exchange
  localFFT -> transpose // in Z
}
```

```
kernel localFFT {
  exposes parallelism [n^2]
  requires flops [5 * n * log2(n)] as dp, simd
  requires loads [a * (n*wordSize) * max(1, log(n*wordSize)/log(Z))]
    from fftVolume
}
```

```
kernel exchange {
  exposes parallelism [P]
  requires messages [(n^3 * wordSize) / P] as
      allToAll
}
```

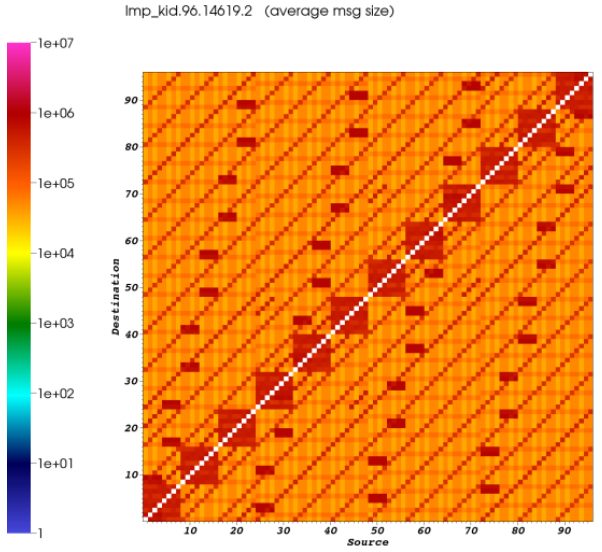
3DFFT: Slab vs. Pencil Tradeoff Ideal Parallelism

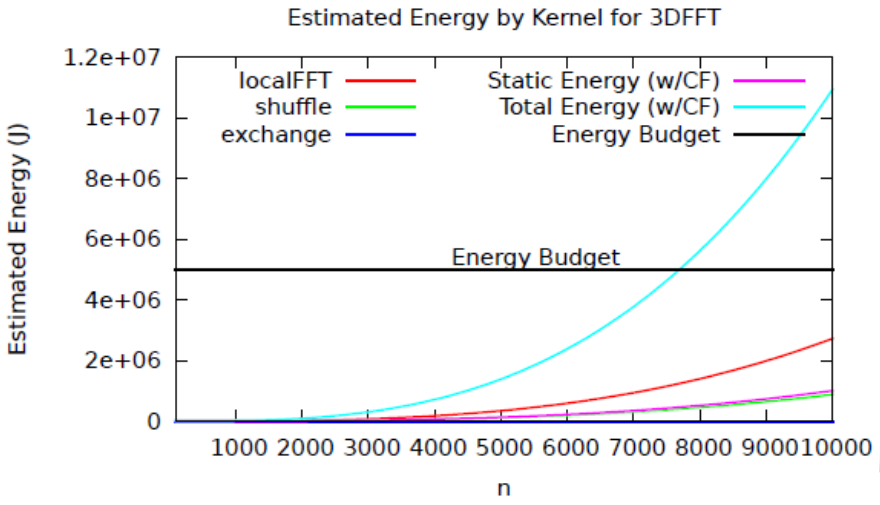
- Insights become obvious with Aspen



Future Directions in Predictive Performance

- Predictive techniques for performance, (resiliency, power) are more important now than over the last two decades!
  - Emerging architectures
  - Application paradigms (e.g., UQ) have important differences from traditional usage scenarios
- We need deployable, flexible methodologies for driving the decisions in architectures, software, and applications.
  - Range of predictive techniques must allow 10+ years foresight
  - Multiple resolutions and timescales
  - Analytical models, simulation, prototypes must be part of an overall strategy to achieve our goal
- Oxbow focuses on today's concrete details
- Aspen fills an important gap in our methods





Summary

- Our community expects major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - Major shifts and uncertainty in architectures, software, applications
  - Design of processors, memory systems, interconnects, storage
- Technologies particularly pertinent to addressing some of these challenges
  - Heterogeneous computing
  - Nonvolatile memory
- DOE has initiated Codesign Centers that bring together all stakeholders to develop integrated solutions
- Aspen is a new approach to model characteristics of applications and architectures
  - This structure allows easy development, sharing, verification of models

Contributors and Recent Sponsors

- Future Technologies Group: http://ft.ornl.gov
  - Publications: https://ft.ornl.gov/publications

- Department of Energy Office of Science
  - Vancouver Project: https://ft.ornl.gov/trac/vancouver
  - Blackcomb Project: https://ft.ornl.gov/trac/blackcomb
  - ExMatEx Codesign Center: http://codesign.lanl.gov
  - Cesar Codesign Center: http://cesar.mcs.anl.gov/
  - SciDAC: SUPER, SDAV http://science.energy.gov/ascr/research/scidac/scidac-institutes/
  - CS Efforts: http://science.energy.gov/ascr/research/computer-science/
- DOE 'Application' offices
- National Science Foundation Keeneland Project: http://keeneland.gatech.edu
- NVIDIA CUDA Center of Excellence at Georgia Tech
- Other sponsors
  - ORNL LDRD, NIH, AFRL, DoD
  - DARPA (HPCS, UHPC, AACE)

Q & A

More info: vetter@computer.org

Recent Publications from FTG (2012-3)

[1] F. Ahmad, S. Lee, M. Thottethodi, and T.N. VijayKumar, "MapReduce with Communication Overlap (MaRCO)," *Journal of Parallel and Distributed Computing, 2012, http://dx.doi.org/10.1016/j.jpdc.2012.12.012.*

[2] C. Chen, Y. Chen, and P.C. Roth, "DOSAS: Mitigating the Resource Contention in Active Storage Systems," in *IEEE Cluster 2012, 2012, 10.1109/cluster.2012.66.*

[3] A. Danalis, P. Luszczek, J. Dongarra, G. Marin, and J.S. Vetter, "BlackjackBench: Portable Hardware Characterization," *SIGMETRICS Performance Evaluation ReviewSIGMETRICS Performance Evaluation Review, 40, 2012,*

[4] A. Danalis, C. McCurdy, and J.S. Vetter, "Efficient Quality Threshold Clustering for Parallel Architectures," in *IEEE International Parallel & Distributed Processing Symposium (IPDPS). Shanghai: IEEEE, 2012, http://dx.doi.org/10.1109/IPDPS.2012.99.*

[5] J.M. Dennis, J. Edwards, K.J. Evans, O. Guba, P.H. Lauritzen, A.A. Mirin, A. St-Cyr, M.A. Taylor, and P.H. Worley, "CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model," *International Journal of High Performance Computing Applications, 26:74–89, 2012, 10.1177/1094342011428142.*

[6] J.M. Dennis, M. Vertenstein, P.H. Worley, A.A. Mirin, A.P. Craig, R. Jacob, and S.A. Mickelson, "Computational Performance of Ultra-High-Resolution Capability in the Community Earth System Model," *International Journal of High Performance Computing Applications, 26:5–16, 2012, 10.1177/1094342012436965.*

[7] K.J. Evans, A.G. Salinger, P.H. Worley, S.F. Price, W.H. Lipscomb, J. Nichols, J.B.W. III, M. Perego, J. Edwards, M. Vertenstein, and J.-F. Lemieux, "A modern solver framework to manage solution algorithm in the Community Earth System Model," *International Journal of High Performance Computing Applications, 26:54–62, 2012, 10.1177/1094342011435159.*

[8] S. Lee and R. Eigenmann, "OpenMPC: Extended OpenMP for Efficient Programming and Tuning on GPUs," *International Journal of Computational Science and Engineering, 8(1), 2013,*

[9] S. Lee and J.S. Vetter, "Early Evaluation of Directive-Based GPU Programming Models for Productive Exascale Computing," in *SC12: ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis. Salt Lake City, Utah, USA: IEEE press, 2012, http://dl.acm.org/citation.cfm?id=2388996.2389028, http://dx.doi.org/10.1109/SC.2012.51.*

[10] D. Li, B.R. de Supinski, M. Schulz, D.S. Nikolopoulos, and K.W. Cameron, "Strategies for Energy Efficient Resource Management of Hybrid Programming Models," *IEEE Transaction on Parallel and Distributed SystemsIEEE Transaction on Parallel and Distributed Systems, 2013, http://dl.acm.org/citation.cfm?id=2420628.2420808,*

[11] D. Li, D.S. Nikolopoulos, and K.W. Cameron, "Modeling and Algorithms for Scalable and Energy Efficient Execution on Multicore Systems," in *Scalable Computing: Theory and Practice, U.K. Samee, W. Lizhe et al., Eds.: Wiley & Sons, 2012,*

[12] D. Li, D.S. Nikolopoulos, K.W. Cameron, B.R. de Supinski, E.A. Leon, and C.-Y. Su, "Model-Based, Memory-Centric Performance and Power Optimization on NUMA Multiprocessors," in *International Symposium on Workload Characterization. San Diego, 2012, http://www.computer.org/csdl/proceedings/iiswc/2012/4531/00/06402921-abs.html*

[13] D. Li, J.S. Vetter, G. Marin, C. McCurdy, C. Cira, Z. Liu, and W. Yu, "Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications," in *IEEE International Parallel & Distributed Processing Symposium (IPDPS). Shanghai: IEEEE, 2012, http://dl.acm.org/citation.cfm?id=2358563, http://dx.doi.org/10.1109/IPDPS.2012.89.*

Recent Publications from FTG (2012-3)

[14] D. Li, J.S. Vetter, and W. Yu, "Classifying Soft Error Vulnerabilities in Extreme-Scale Scientific Applications Using a Binary Instrumentation Tool," in *SC12: ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis. Salt Lake City, 2012, http://dl.acm.org/citation.cfm?id=2388996.2389074, http://dx.doi.org/10.1109/SC.2012.29.*

[15] Z. Liu, B. Wang, P. Carpenter, D. Li, J.S. Vetter, and W. Yu, "PCM-Based Durable Write Cache for Fast Disk I/O," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). Arlington, Virginia, 2012, http://www.computer.org/csdl/proceedings/mascots/2012/4793/00/4793a451-abs.html*

[16] G. Marin, C. McCurdy, and J.S. Vetter, "Diagnosis and Optimization of Application Prefetching Performance," in *ACM International Conference on Supercomputing (ICS). Euguene, OR: ACM, 2013*

[17] J.S. Meredith, S. Ahern, D. Pugmire, and R. Sisneros, "EAVL: The Extreme-scale Analysis and Visualization Library," in *Proceedings of the Eurographics Symposium on Parallel Graphics and Visualization (EGPGV), 2012*

[18] J.S. Meredith, R. Sisneros, D. Pugmire, and S. Ahern, "A Distributed Data-Parallel Framework for Analysis and Visualization Algorithm Development," in *Proceedings of the 5th Annual Workshop on General Purpose Processing with Graphics Processing Units. New York, NY, USA: ACM, 2012, pp. 11–9, http://doi.acm.org/10.1145/2159430.2159432, 10.1145/2159430.2159432.*

[19] A.A. Mirin and P.H. Worley, "Improving the Performance Scalability of the Community Atmosphere Model," *International Journal of High Performance Computing Applications, 26:17–30, 2012, 10.1177/1094342011412630.*

[20] P.C. Roth, "The Effect of Emerging Architectures on Data Science (and other thoughts)," in *2012 CScADS Workshop on Scientific Data and Analytics for Extreme-scale Computing. Snowbird, UT, 2012, http://cscads.rice.edu/workshops/summer-2012/data-analytics*

[21] K. Spafford, J.S. Meredith, S. Lee, D. Li, P.C. Roth, and J.S. Vetter, "The Tradeoffs of Fused Memory Hierarchies in Heterogeneous Architectures," in *ACM Computing Frontiers (CF). Cagliari, Italy: ACM, 2012, http://dl.acm.org/citation.cfm?id=2212924, http://dx.doi.org/10.1145/2212908.2212924.*

[22] K. Spafford and J.S. Vetter, "Aspen: A Domain Specific Language for Performance Modeling," in *SC12: ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis, 2012, http://dl.acm.org/citation.cfm?id=2388996.2389110, http://dx.doi.org/10.1109/SC.2012.20.*

[23] C.-Y. Su, D. Li, D.S. Nikolopoulos, M. Grove, K.W. Cameron, and B.R. de Supinski, "Critical Path-Based Thread Placement for NUMA Systems," *ACM SIGMETRICS Performance Evaluation ReviewACM SIGMETRICS Performance Evaluation Review, 40, 2012, http://dl.acm.org/citation.cfm?id=2381056.2381079,*

[24] V. Tipparaju and J.S. Vetter, "GA-GPU: Extending a Library-based Global Address Space Programming Model for Scalable Heterogeneous Computing Systems," in *ACM Computing Frontiers (CF), 2012, http://dx.doi.org/10.1145/2212908.2212918.*

[25] J.S. Vetter, *Contemporary High Performance Computing: From Petascale Toward Exascale, vol. 1, 1 ed. Boca Raton: Taylor and Francis, 2013, http://j.mp/RrBdPZ,*

[26] J.S. Vetter, R. Glassbrook, K. Schwan, S. Yalamanchili, M. Horton, A. Gavrilovska, M. Slawinska, J. Dongarra, J. Meredith, P.C. Roth, K. Spafford, S. Tomov, and J. Wynkoop, "Keeneland: Computational Science using Heterogeneous GPU Computing," in *Contemporary High Performance Computing: From Petascale Toward Exascale, vol. 1, CRC Computational Science Series, J.S. Vetter, Ed., 1 ed. Boca Raton: Taylor and Francis, 2013, pp. 900,*

[27] W. Yu, X. Que, V. Tipparaju, and J.S. Vetter, "HiCOO: Hierarchical cooperation for scalable communication in Global Address Space programming models on Cray XT systems," *Journal of Parallel and Distributed ComputingJournal of Parallel and Distributed Computing, 2012.*