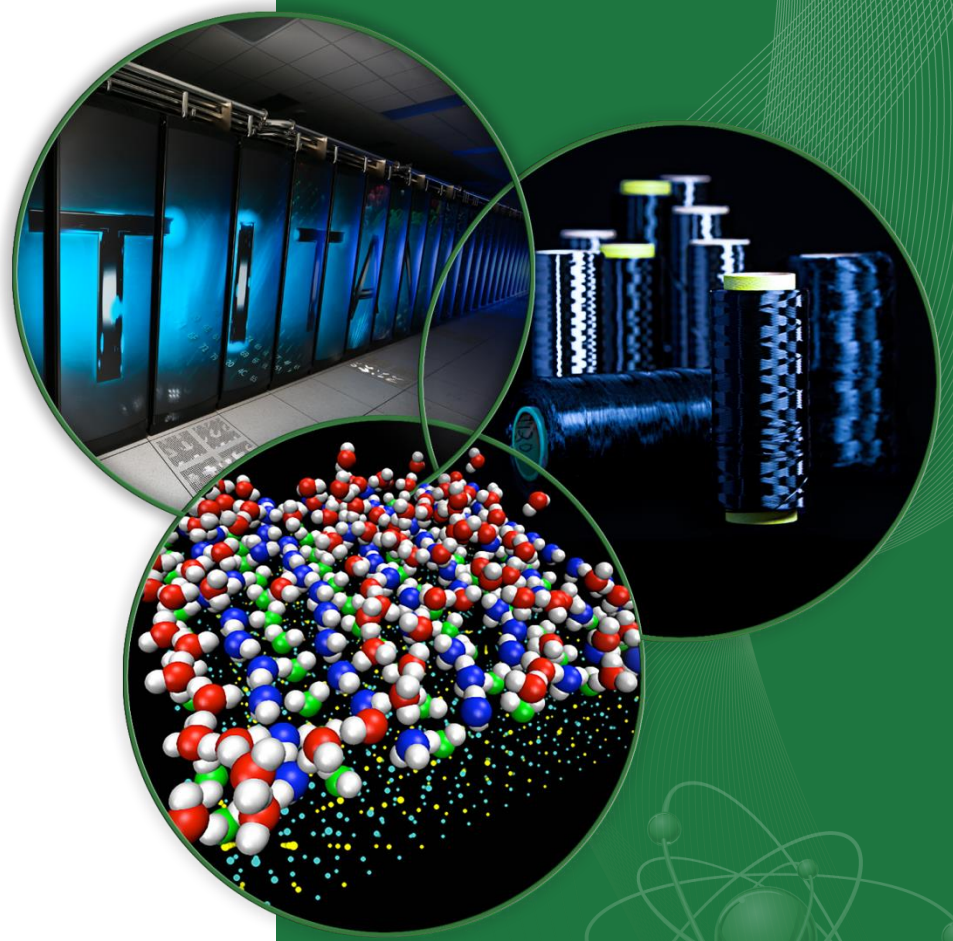
Architecture trends, performance prediction and co-design tools

Jeffrey S. Vetter



US-Japan Joint Institute for Fusion Theory Workshop on Innovations and Co-designs of Fusion Simulations towards Extreme Scale Computing

Nagoya

20 Aug 2015

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

Georgia Tech College of Computing
Computational Science and Engineering

ORNL is managed by UT-Battelle
for the US Department of Energy

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

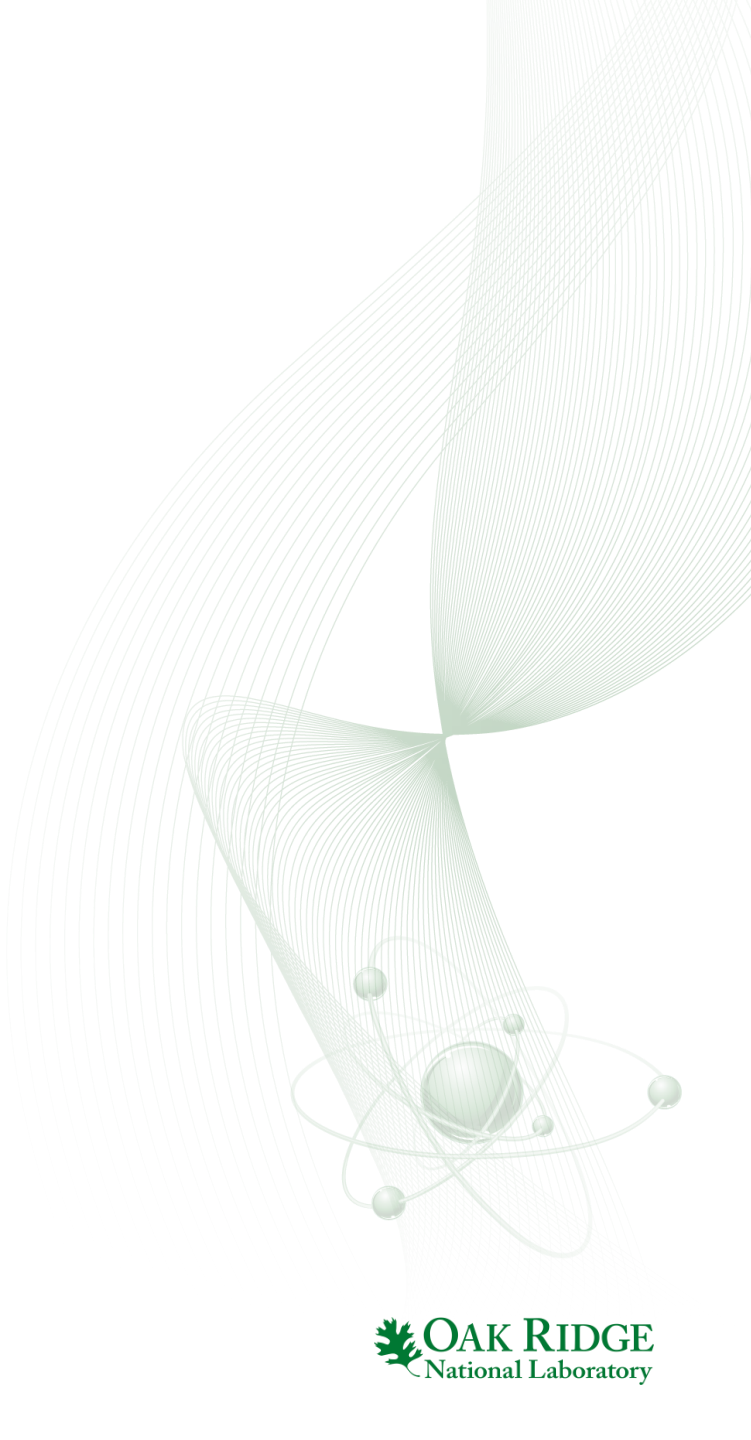
OAK RIDGE National Laboratory

Overview

- Quick status review of US HPC
- Our community faces major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - Architectural Trends: New technologies emerging to address some of these challenges
    - Heterogeneous computing
    - Nonvolatile memory
- Not just HPC: Most uncertainty in at least two decades
- Codesign
  - We need performance prediction and engineering tools now more than ever!
  - Aspen is a tool for structured design and analysis
    - Co-design applications and architectures for performance, power, resiliency
    - Automatic model generation
    - Scalable to distributed scientific workflows
    - DVF – a new twist on resiliency modeling

HPC in USA

- NSCI announced
- DOE Exascale moving into project planning phase

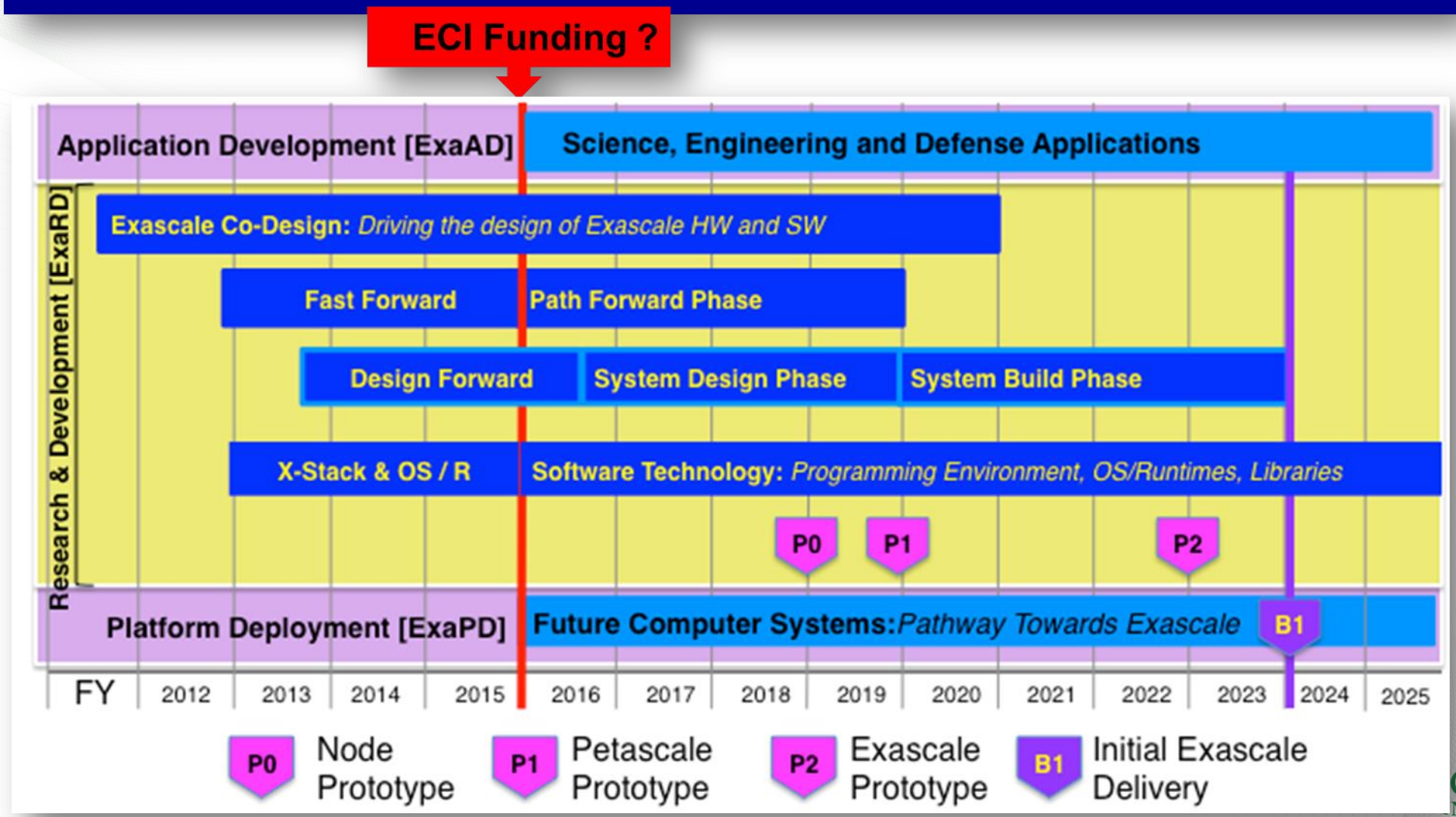


National Strategic Computing Initiative - 5 Themes

- Create systems that can apply exaflops of computing power to exabytes of data
  - Applications readiness and requirements
- Keep the US at the forefront of HPC capabilities
- Improve HPC application developer productivity
- Make HPC readily available
- Establish hardware technology for future HPC systems
  - Neuromorphic, beyond CMOS, Quantum, etc

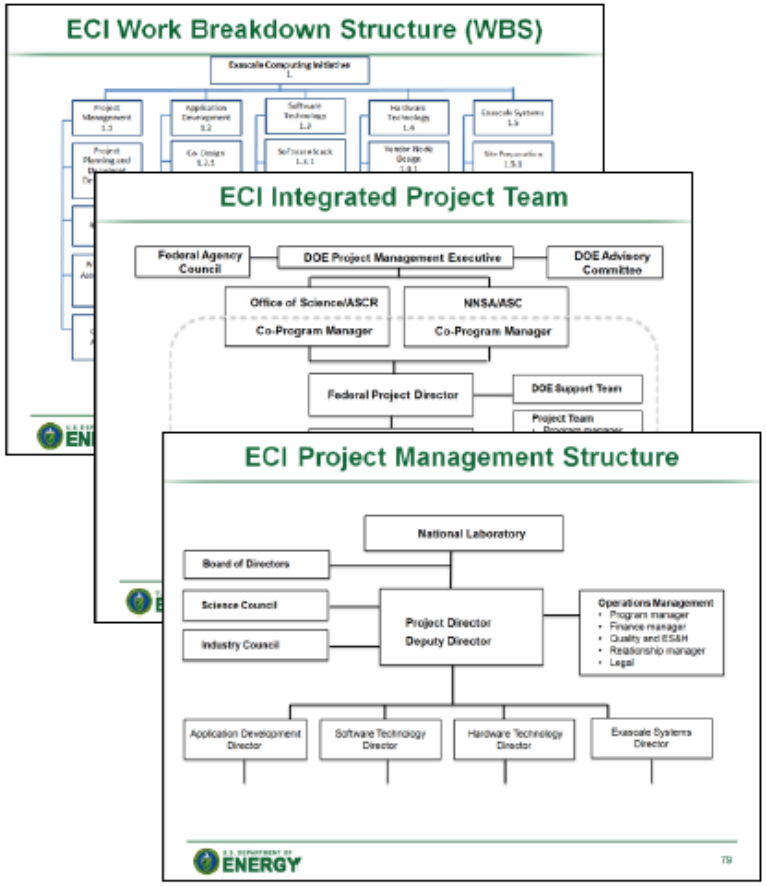
Executive order, 29 July 2015
https://www.whitehouse.gov/sites/default/files/microsites/ostp/nsci_fact_sheet.pdf

ECI Project Schedule



Courtesy Bill Harrod, Dec 2014

“Projectizing” the Exascale Initiative



- The exascale initiative will follow established DOE review and decision protocols for its execution
- A project office has been established at ORNL with representation from the major participating laboratories (ANL, LANL, LBNL, LLNL, ORNL, SNL)
- An Integrated Project Team (IPT) has been established, analogous to execution of previous, large, Office of Science projects
- The IPT is refining the work breakdown structure (WBS) and is preparing required project documentation (e.g., critical-decision packages, preliminary project execution plans, etc.)
- A top-level WBS activity has been established to develop and implement exascale applications, based on a labs-wide request for information

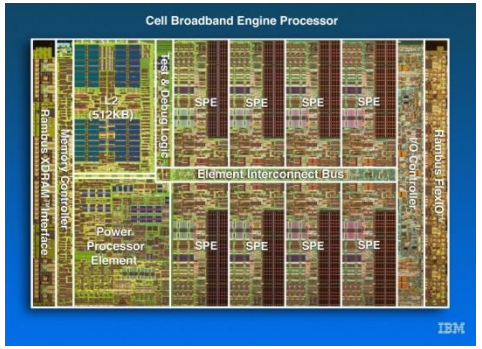
Architectural Trends

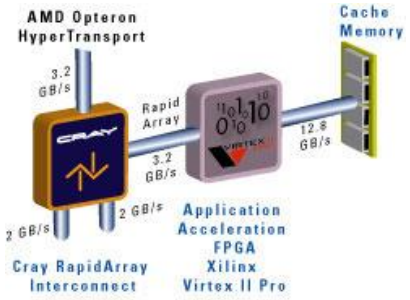
- Heterogeneous Computing
- NV Memory
- Optical interconnect, Silicon photonics
- Storage systems (key, value)

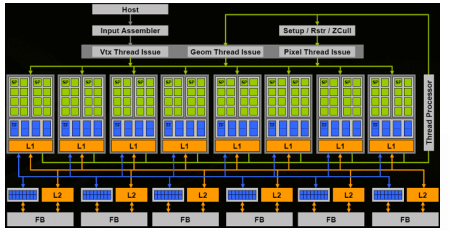
Earlier Experimental Computing Systems (past decade)

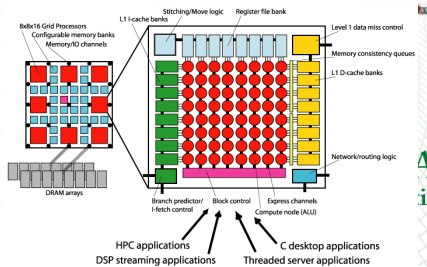
- The past decade has started the trend away from traditional 'simple' architectures
- Examples
  - Cell, GPUs, FPGAs, SoCs, etc
- Mainly driven by facilities costs and successful (sometimes heroic) application examples

Popular architectures since ~2004



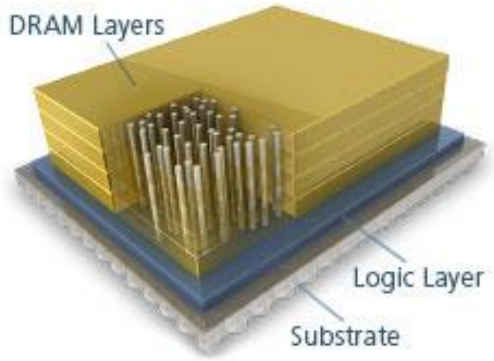


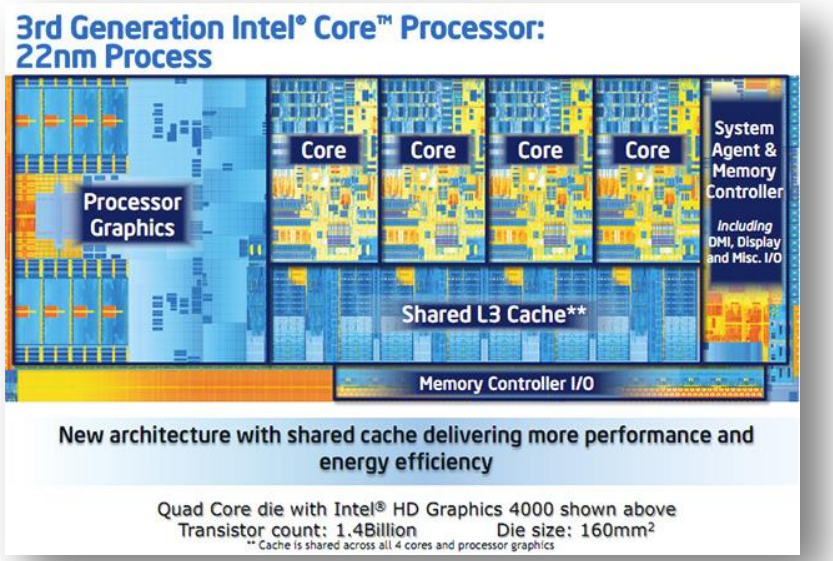


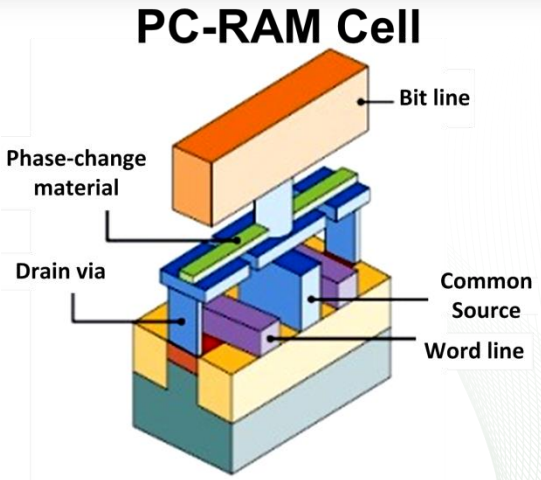


Emerging Computing Architectures – Future

- Heterogeneous processing
  - Latency tolerant cores
  - Throughput cores
  - Special purpose hardware (e.g., AES, MPEG, RND)
  - Fused, configurable memory
- Memory
  - 2.5D and 3D Stacking
  - HMC, HBM, WIDEIO2, LPDDR4, etc
  - 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

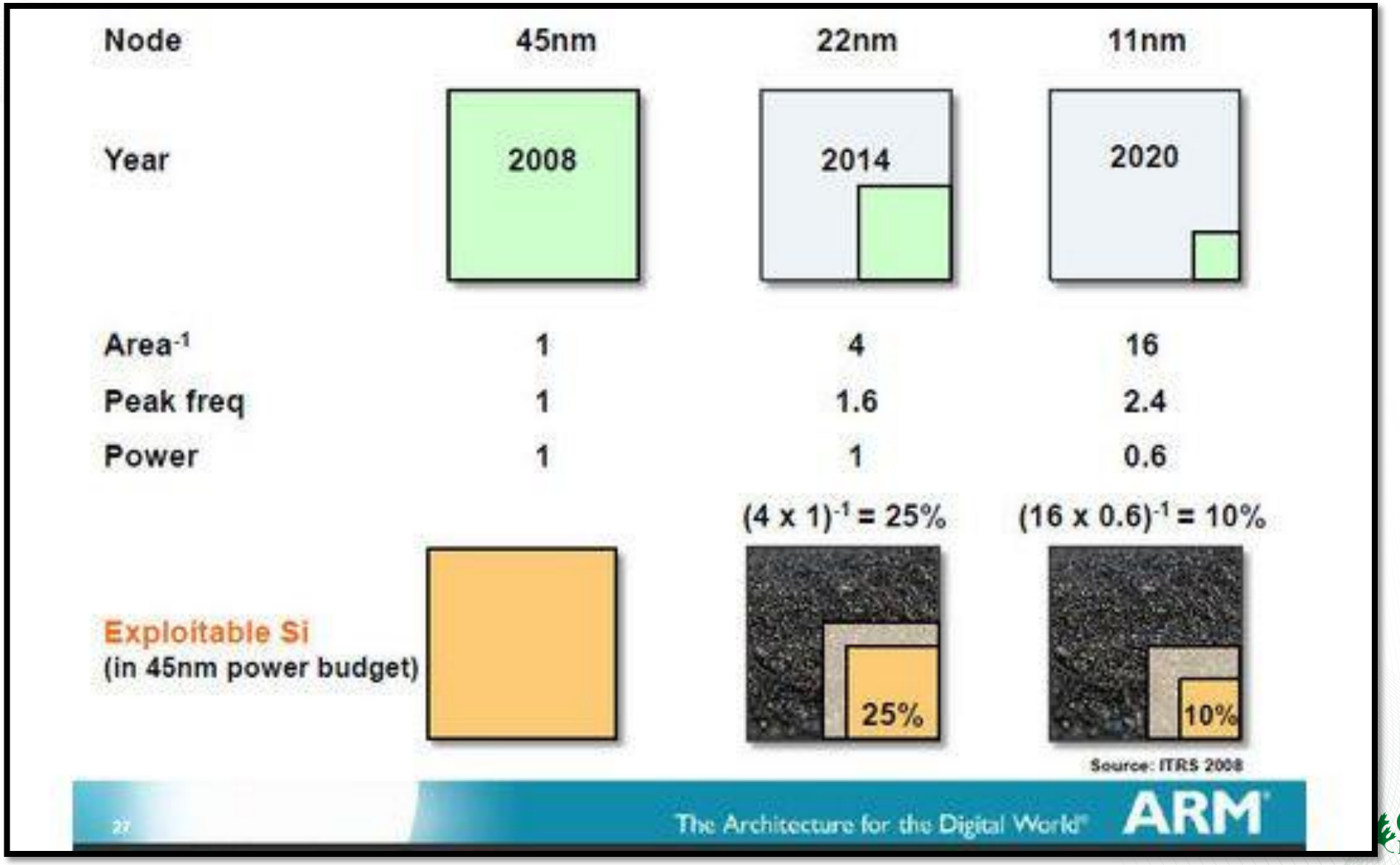






HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.

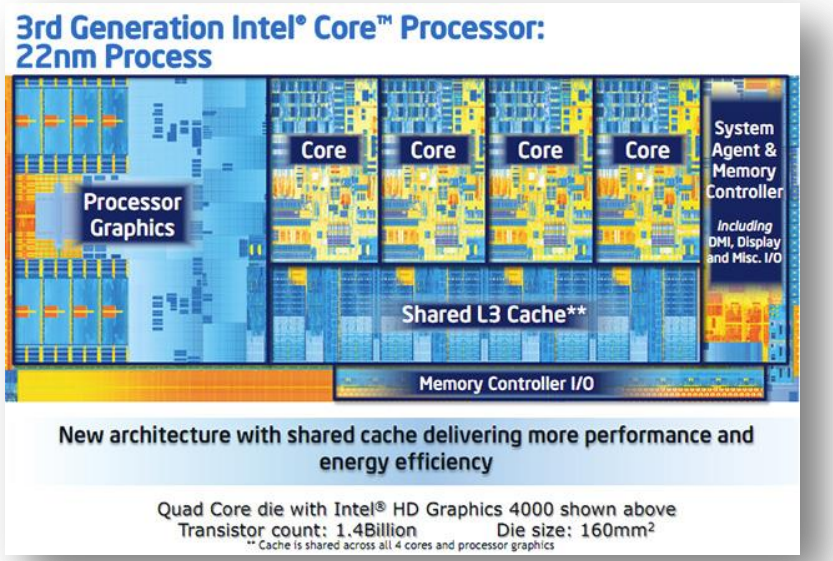
Dark Silicon Will Make Heterogeneity and Specialization More Relevant

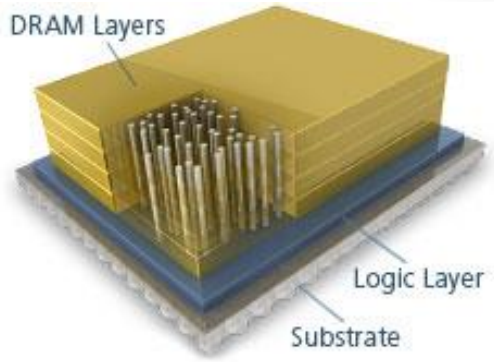


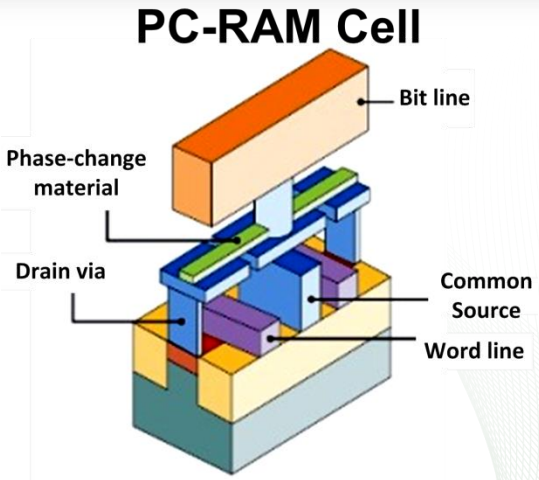
Source: ARM

Emerging Computing Architectures – Now

- Heterogeneous processing
  - Latency tolerant cores
  - Throughput cores
  - Special purpose hardware (e.g., AES, MPEG, RND)
- Memory
  - Fused, configurable memory
  - 2.5D and 3D Stacking
  - HMC, HBM, WIDEIO2, LPDDR4, etc
  - New devices (PCRAM, ReRAM)
- Interconnects
  - Collective offload
  - Scalable topologies
- Storage
  - Active storage
  - Non-traditional storage architectures (key-va stores)
- Improving performance and programmability in face of increasing complexity







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

Recent announcements

Nvidia and IBM create GPU interconnect for faster supercomputing

"NVLink" shares up to 80GB of data per second between CPUs and GPUs.

by Jon Brodkin - M

It Begins: AMD Announces Its First ARM Based Server SoC, 64-bit/8-core Opteron A1100

by Anand Lal Shimpi on January 28, 2014 6:35 PM EST

Posted in CPUs | IT Computing | Enterprise | enterprise CPUs | AMD | Opteron | Opteron A1100 | ARM

123 Comments

+ Add A Comme

"SEATTLE" 64-BIT ARM SERVER PROCESSOR

FIRST 28NM AR

AMD

- Indus
- prove
- The
- Serv

CPU

- 2-4x
- impr
- 8 co
- Base
- Exte
- CPU
- Serv
- Lega
- Stora

SAMPLI

12 | CHANGING INFRASTRU

Around 15 mon
2014. Less than
A1100: a 64-bit

The Opteron A1
talking about ha
away entirely, b
bets going on. E
process at Glob

Nvidia Jetson TK1 mini supercomputer is up for pre-order

Will ship on 15 May

By Lee Bell

Fri May 02 2014, 11

computers attempt
self-driving cars.

Speaking at the GP
Hsun Huang descril
can run, but at a sl

With a total perfor
Raspberry Pi board
in the US - a numb
launched at CES in

"The Jetson TK1 als
comes with a whole
it could be classifyi

Parameters are loa
recognises objects,

MarketWatch

PRESS RELEASE

Altera and IBM Unveil FPGA-accelerated POWER Systems with Coherent Shared Memory

By

Published: Nov 17, 2014 8:00 a.m. ET

8 13 Aa

POWER8 Systems that Leverage Reprogrammable FPGA Accelerators Gain Significant Improvements in System Performance, Efficiency and Flexibility

NEW ORLEANS, Nov. 17, 2014 /PRNewswire/ -- **SuperComputing 2014** -- Altera Corporation ALTR, +0.00% and IBM IBM, +0.00% today unveiled the industry's first FPGA-based acceleration platform that coherently connects an FPGA to a POWER8 CPU leveraging IBM's Coherent Accelerator Processor Interface (CAPI). The reconfigurable hardware accelerator features shared virtual memory between the FPGA and processor which significantly improves system performance, efficiency and flexibility in high-performance computing (HPC) and data center applications. Altera and IBM are presenting several POWER8 systems that are coherently accelerated using FPGAs at SuperComputing 2014.

Working together through the OpenPOWER Foundation, Altera and IBM are

A-SERIES REDEFINES COMPUTE

Kaveri

4 "Steamroller" CPU Cores

8 GCN GPU Cores

Multimedia
AMD TrueAudio technology
UVD
VCE

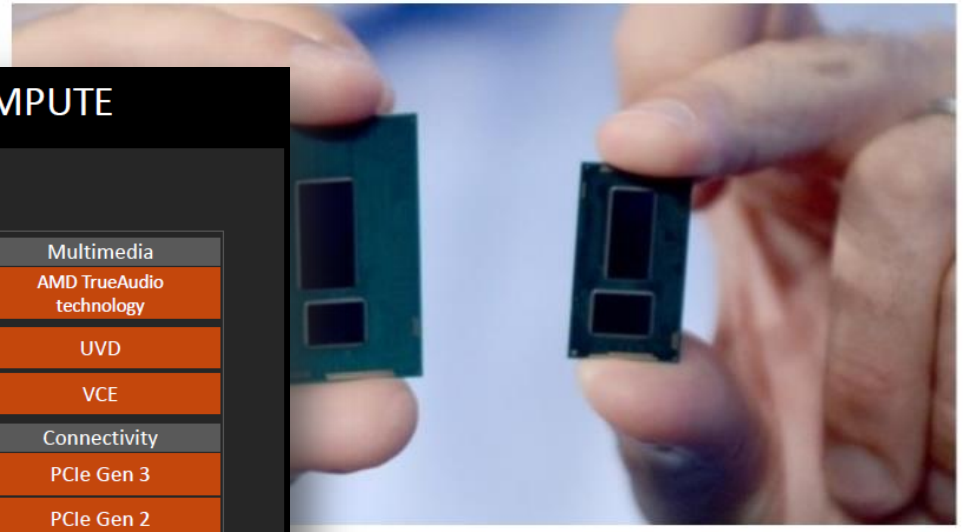
Connectivity
PCIe Gen 3
PCIe Gen 2

Display
DisplayPort 1.2
HDMI 1.4a

lying AMD's "Kaveri" APU for Hete

Intel's 14nm Broadwell GPU takes shape, indicates major improvements over Haswell

By Sebastian Anthony on November 5, 2013 at 10:21 am | 16 Comments



Ahead of its 2014 launch, Intel has started open-sourcing the Linux driver for Broadwell's GPU. Broadwell is the 14nm die shrink of Intel's

18

nd while the CPU
xpected to
dwell's GPU looks
es (Iris) GPU
a modified
GPU, and the

Intel Mates FPGA With Future Xeon Server Chip

June 18, 2014 by Timothy Prickett Morgan

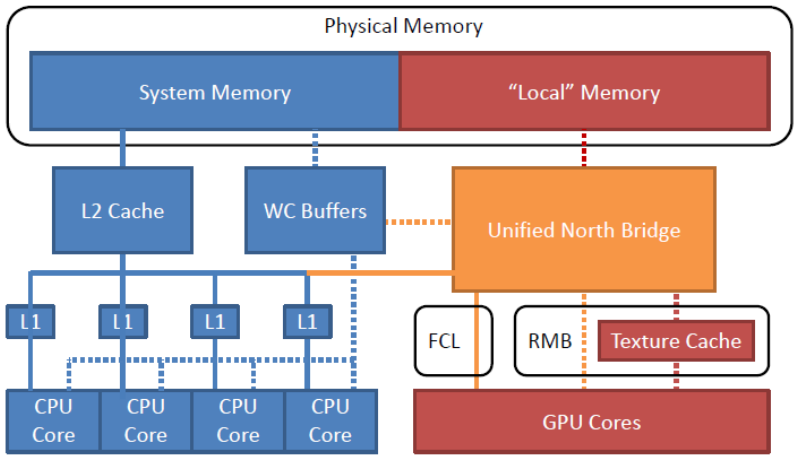
intel inside

Intel is taking field programmable gate arrays seriously as a means of accelerating applications and has crafted a hybrid chip that marries an FPGA to a Xeon E5 processor and puts them in the same processor socket.

Contemporary Heterogeneous Architectures

Property	CUDA	GCN	MIC
Programming models	CUDA, OpenCL	OpenCL, C++ AMP	OpenCL, Cilk, TBB, LEO, OpenMP
Thread Scheduling	Hardware	Hardware	Software
Programmer Managed Cache	Yes	Yes	No
Global Synchronization	No	No	Yes
L2 Cache Type	Shared	Private per core	Private per core
L2 Total Size	Up to 1.5MB	Up to 0.5 MB	25MB
L2 Line-size	128	64	64
L1 Data Cache	Read-only + Read-write	Read-only	Read-write
Native Mode	No	No	Yes

Tight Integration will expand workload possibilities



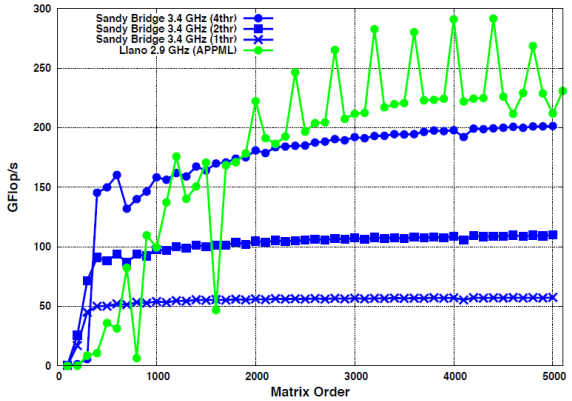
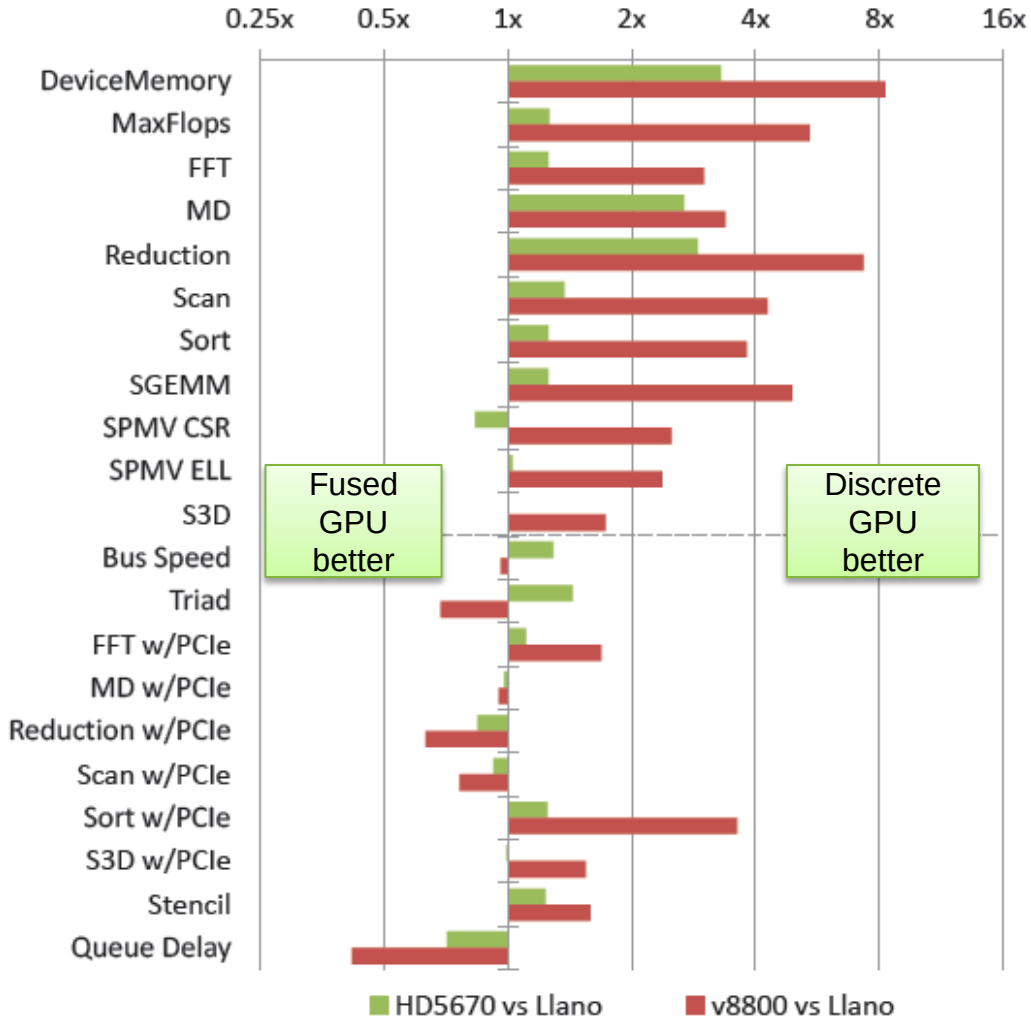


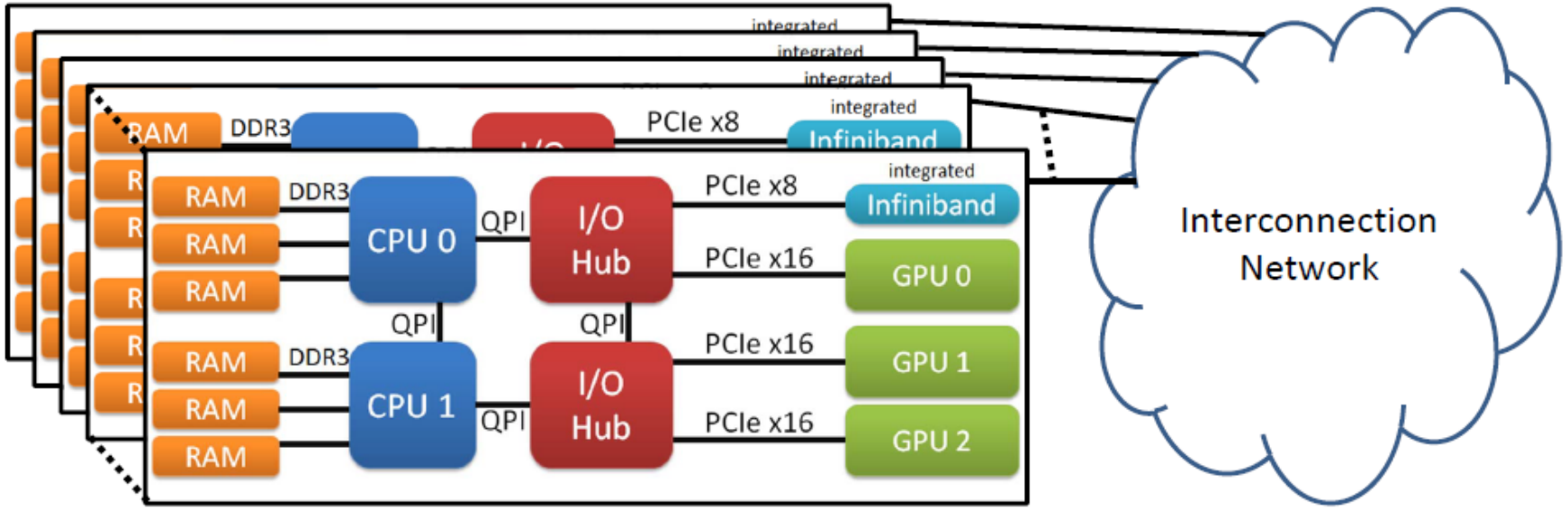
Figure 3: SGEMM Performance (one, two, and four CPU threads for Sandy Bridge and the OpenCL-based AMD APPML for Llano's fGPU)



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.

https://github.com/vetter/shoc

Challenges: Today's programming model



MPI

Low overhead

Resource contention

Locality

OpenMP, Pthreads

SIMD

NUMA

OpenACC, CUDA, OpenCL, OpenMP4, …

Memory use, coalescing

Data orchestration

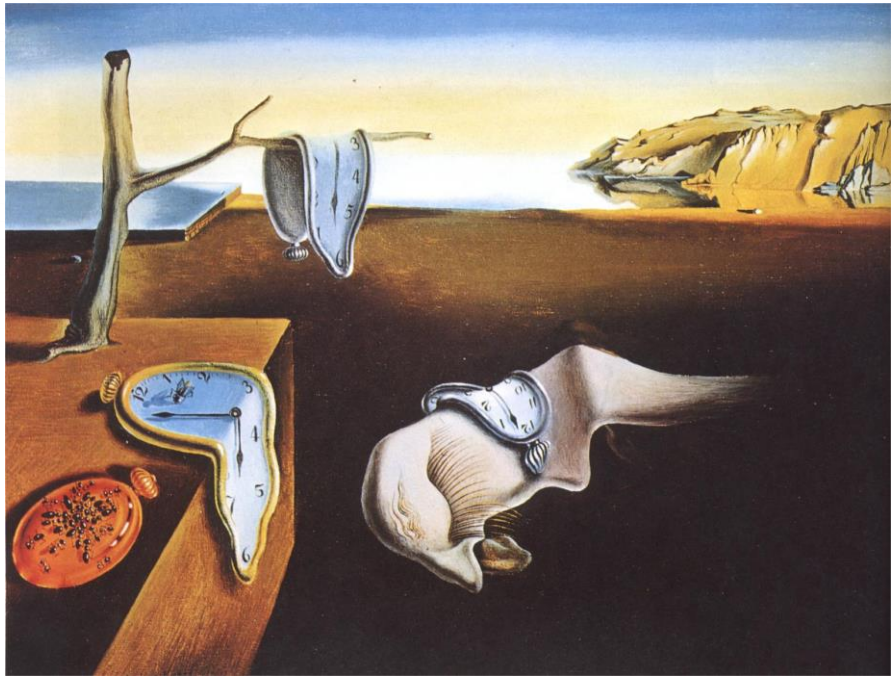
Fine grained parallelism

Hardware features

Heterogeneous Computing – Key Messages

- Heterogeneous computing is not new, and it will continue to play a role in HPC systems
- Tighter integration of heterogeneous cores will provide new opportunities for application acceleration
- Programmer productivity will be a major challenge
  - Performance portability
  - Need better, portable programming models for upcoming diverse systems

New and Improved Memory Systems are the Next Big Thing

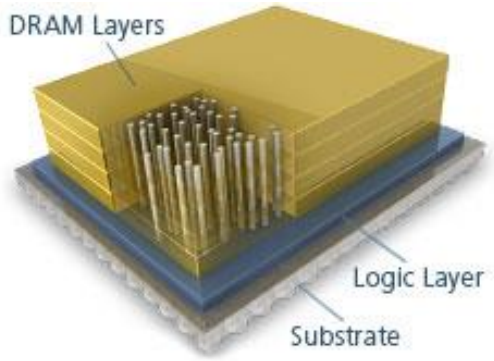


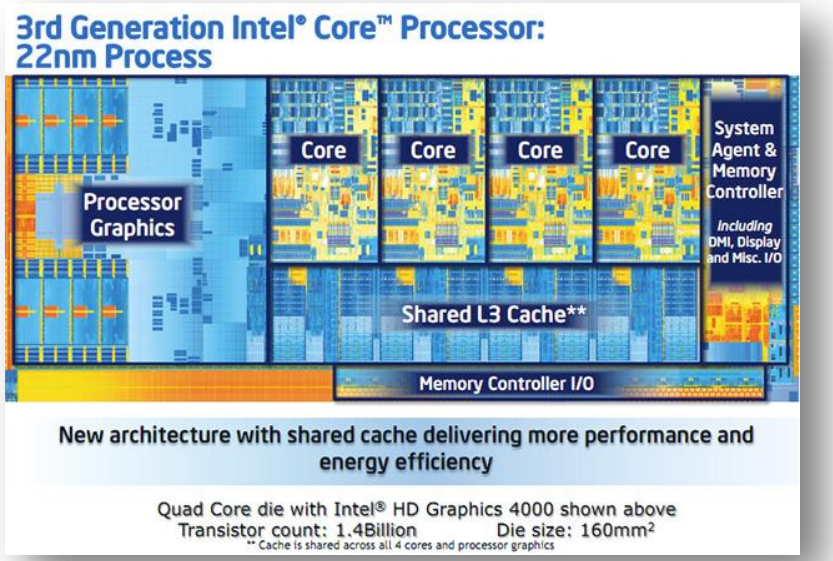
http://www.wikipaintings.org/en/salvador-dali/the-persistence-of-memory-1931

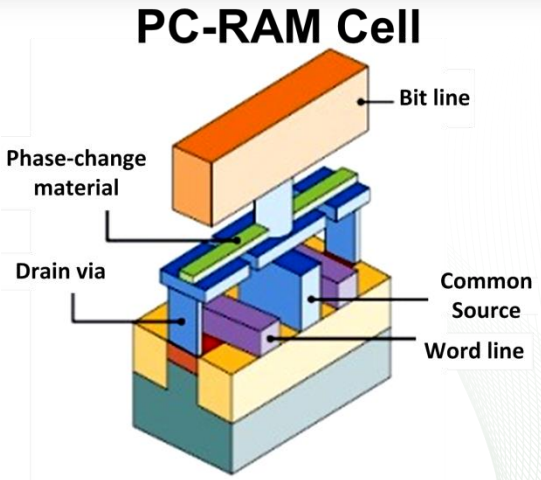
OAK RIDGE National Laboratory

Emerging Computing Architectures – Future

- Heterogeneous processing
  - Latency tolerant cores
  - Throughput cores
  - Special purpose hardware (e.g., AES, MPEG, RND)
  - Fused, configurable memory
- Memory
  - 2.5D and 3D Stacking
  - HMC, HBM, WIDEIO2, LPDDR4, etc
  - 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.

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 Exascale Architecture Targets

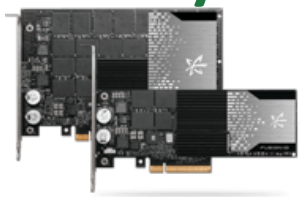
(From Exascale Arch Report 2009)

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)	

Parallel I/O ??

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

NVRAM Technology Continues to Improve – Driven by Market Forces



designlines MEMORY

News & Analysis

3D NAND Production Starts at Samsung

Peter Clarke

8/6/2013 08:05 AM EDT

16 comments

NO RATINGS
1 saves
LOGIN TO RATE

Like 17 Tweet

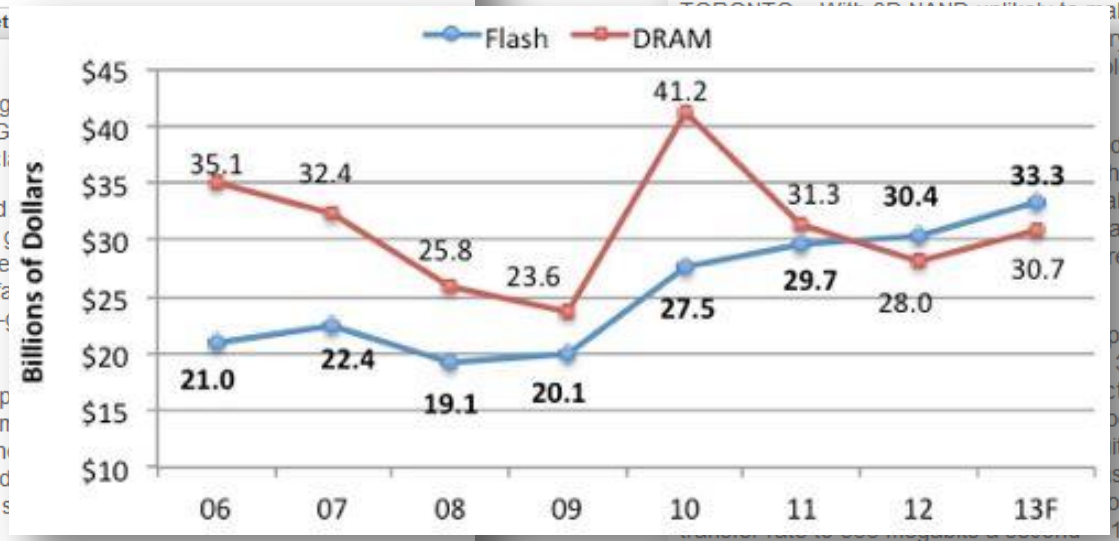
LONDON — Samsung
production of a 128 G
multiple layers, and cl

The memory is based
conventional floating
In the vertical arrange
reliability between a fa
conventional floating-
in a press release.

The technology is cap
did not disclose how n
vertical NAND, nor wh
whether it had relaxed
in 2D memory, which s

The company did say that the memory would provide improvements in performance and area ratio, and a V-NAND chip is suitable for a wide range of consumer and commercial applications including embedded NAND storage and solid-state drives.

The V-NAND component has the same memory capacity as a 128



designlines MEMORY

News & Analysis

3D NAND Transition: 15nm Process Technology Takes Shape

Gary Hilson

5/13/2014 08:15 AM EDT

5 comments

NO RATINGS
LOGIN TO RATE

Like 15 Tweet 6 Share 6 +1 1

ake economic sense until
ry partner Toshiba both
logies to produce NAND

d to both 2-bit-per-cell
hitectures with
alf of 2014. The 15nm
and will be used across
removable cards to

process technology, and
3D NAND, said Scott
tronic Components'
ocess works in
itry technology to create
s chips formed with
gy, but boost the data
1.3 times faster -- by
employing a high-speed interface.

Nelson said there is room to advance floating gates before moving

http://www.eetasia.com/STATIC/ARTICLE_IMAGES/201212/EEOL_2012DEC28_STOR_MFG_NT_01.jpg

Original URL: http://www.theregister.co.uk/2013/11/01/hp_memristor_2018/

HP 100TB Memristor drives by 2018 – if you're lucky, admits tech titan

Universal memory slow in coming

By Chris
Posted in

Blocks a
Martin Fi

In five yea
of the tec
off a cliff

The HP a
to have D
in Las Ve
also shov

Intel Announces Optane Storage Brand For 3D XPoint Products

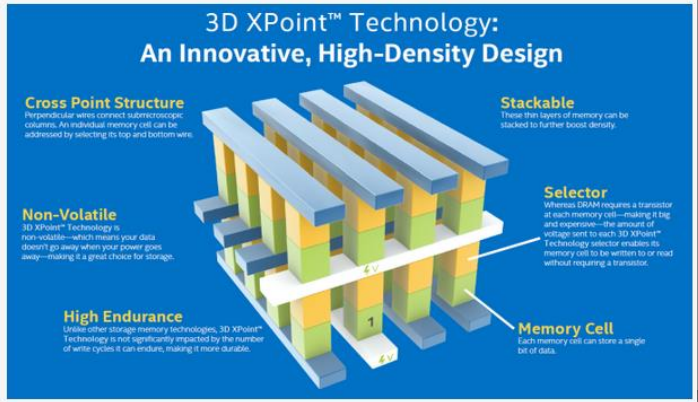
42 Comments

+ Add A Comment

by Ryan Smith on August 18, 2015 2:15 PM EST

Posted in Storage Intel 3D XPoint Memory

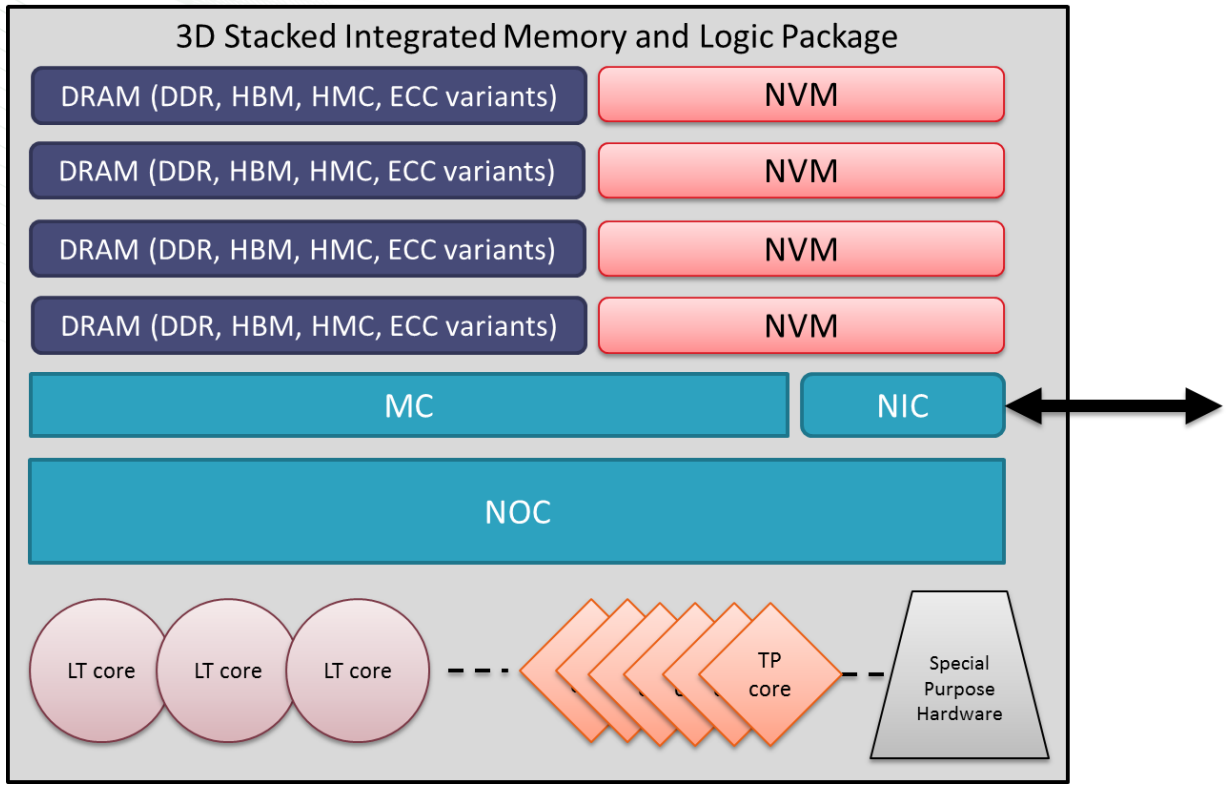
At IDF 2015 this year Intel has announced that their forthcoming 3D XPoint technology based products will be sold under a new brand for the company, Optane.

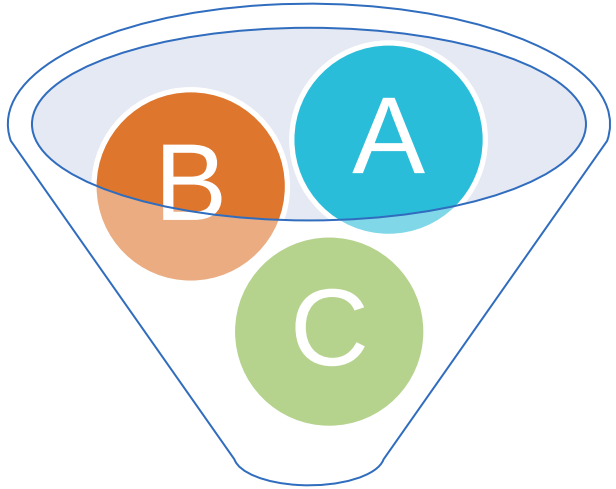


Comparison of emerging memory technologies

	SRAM	DRAM	eDRAM	2D NAND Flash	3D NAND Flash	PCRAM	STTRAM	2D ReRAM	3D ReRAM
Data Retention	N	N	N	Y	Y	Y	Y	Y	Y
Cell Size (F^2)	50-200	4-6	19-26	2-5	<1	4-10	8-40	4	<1
Minimum F demonstrated (nm)	14	25	22	16	64	20	28	27	24
Read Time (ns)	< 1	30	5	10^4	10^4	10-50	3-10	10-50	10-50
Write Time (ns)	< 1	50	5	10^5	10^5	100-300	3-10	10-50	10-50
Number of Rewrites	10^{16}	10^{16}	10^{16}	10^4-10^5	10^4-10^5	10^8-10^{10}	10^{15}	10^8-10^{12}	10^8-10^{12}
Read Power	Low	Low	Low	High	High	Low	Medium	Medium	Medium
Write Power	Low	Low	Low	High	High	High	Medium	Medium	Medium
Power (other than R/W)	Leakage	Refresh	Refresh	None	None	None	None	Sneak	Sneak
Maturity									

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





Natural separation of applications objects?

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*

Observations: Numerous characteristics of applications are a good match for byte-addressable NVRAM

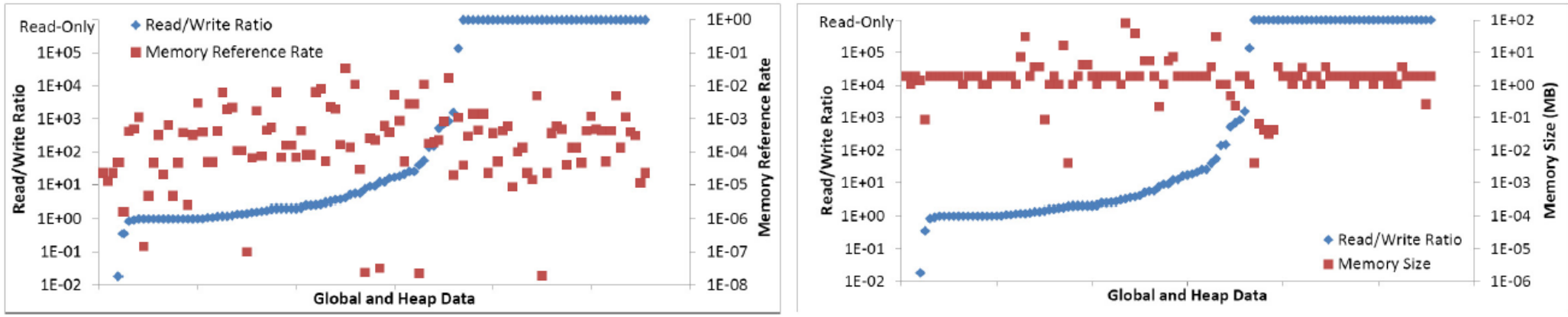


Figure 3: Read/write ratios, memory reference rates and memory object sizes for memory objects in Nek5000

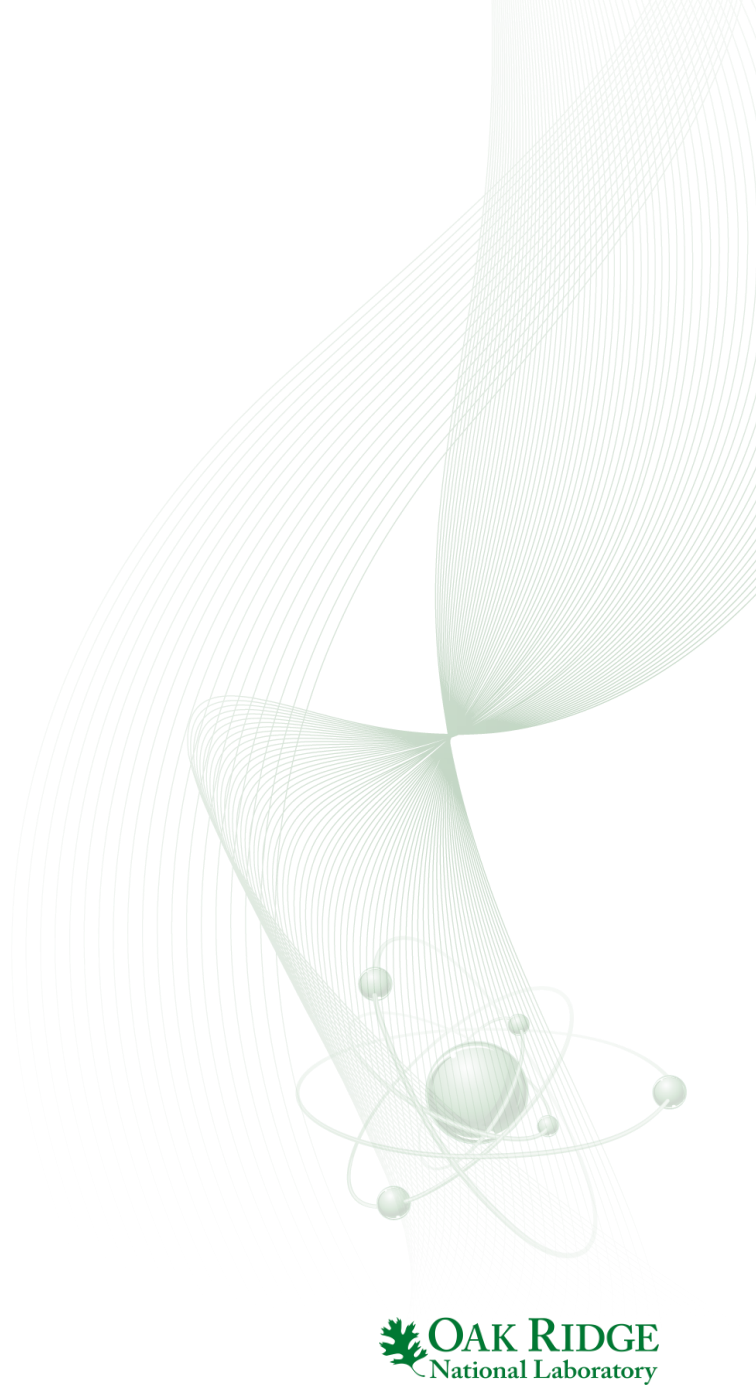
- Many lookup, index, and permutation tables
- Inverted and 'element-lagged' mass matrices
- Geometry arrays for grids
- Thermal conductivity for soils
- Strain and conductivity rates
- Boundary condition data
- Constants for transforms, interpolation
- …

Memory Technologies – Key Messages

- New memory technologies are emerging based on both supply and demand
- NVM will be in future systems, but it is not clear at which levels of the hierarchy, it will be most common
  - Need experience with endurance, performance, and usage models
  - HDD replacement to cache peer
  - How can we use persistence effectively?
- Strategy for exposing this feature is an open research question
  - Hide in memory controller
  - OS managed placement
  - User control

28

Architectural Summary



Exascale architecture targets circa 2009

2009 Exascale Challenges Workshop in San Diego

Attendees envisioned two possible architectural swim lanes:

1. Homogeneous many-core thin-node system
2. Heterogeneous (accelerator + CPU) fat-node system

System attributes	2009	"Pre-Exascale"		"Exascale"	
System peak	2 PF	100-200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Storage	15 PB	150 PB		500 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	500,000	50,000	1,000,000	100,000
Node interconnect BW	1.5 GB/s	150 GB/s	1 TB/s	250 GB/s	2 TB/s
IO Bandwidth	0.2 TB/s	10 TB/s		30-60 TB/s	
MTTI	day	O(1 day)		O(0.1 day)	

Exascale architecture targets

defined at 2009 Exascale Challenges Workshop in San Diego

Where we are going "off the tracks" is
data movement between nodes and from node to storage

Summit: Interconnect BW= 25 GB/s, I/O BW= 1 TB/s

System attributes	2009	"Pre-Exascale"		"Exascale"	
System peak	2 PF	100-200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Storage	15 PB	150 PB		500 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	500,000	50,000	1,000,000	100,000
Node interconnect BW	1.5 GB/s	**150 GB/s**	**1 TB/s**	250 GB/s	2 TB/s
IO Bandwidth	0.2 TB/s	**10 TB/s**		30-60 TB/s	
MTTI	day	O(1 day)		O(0.1 day)	

ASCR Computing Upgrade Summary

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrades	
Name Planned Installation	**Edison**	**TITAN**	**MIRA**	**Cori 2016**	**Summit 2017-2018**	**Theta 2016**	**Aurora 2018-2019**
System peak (PF)	2.6	27	10	> 30	150	>8.5	180
Peak Power (MW)	2	9	4.8	< 3.7	10	1.7	13
Total system memory	357 TB	710TB	768TB	~1 PB DDR4 + High Bandwidth Memory (HBM)+1.5PB persistent memory	> 1.74 PB DDR4 + HBM + 2.8 PB persistent memory	>480 TB DDR4 + High Bandwidth Memory (HBM)	> 7 PB High Bandwidth On-Package Memory Local Memory and Persistent Memory
Node performance (TF)	0.460	1.452	0.204	> 3	> 40	> 3	> 17 times Mira
Node processors	Intel Ivy Bridge	AMD Opteron Nvidia Kepler	64-bit PowerPC A2	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	Multiple IBM Power9 CPUs & multiple Nvidia Voltas GPUS	Intel Knights Landing Xeon Phi many core CPUs	Knights Hill Xeon Phi many core CPUs
System size (nodes)	5,600 nodes	18,688 nodes	49,152	9,300 nodes 1,900 nodes in data partition	~3,500 nodes	>2,500 nodes	>50,000 nodes
System Interconnect	Aries	Gemini	5D Torus	Aries	Dual Rail EDR-IB	Aries	2nd Generation Intel Omni-Path Architecture
File System	7.6 PB 168 GB/s, Lustre®	32 PB 1 TB/s, Lustre®	26 PB 300 GB/s GPFS™	28 PB 744 GB/s Lustre®	120 PB 1 TB/s GPFS™	10PB, 210 GB/s Lustre initial	150 PB 1 TB/s Lustre®

OLCF-5 What's exascale look like?

Date	2009	2012	2017	OLCF-5 2022
System	**Jaguar**	**Titan**	**Summit**	**Exascale**
System peak	2.3 Peta	27 Peta	150+ Peta	**1-2 Exa**
System memory	0.3 PB	0.7 PB	2-5 PB	**10-20 PB**
NVM per node	none	none	800 GB	**~2 TB**
Storage	15 PB	32 PB	120 PB	**~300 PB**
MTTI	days	days	days	**O(1 day)**
Power	7 MW	9 MW	10 MW	**~20 MW**
Node architecture	CPU 12 core	CPU + GPU	X CPU + Y GPU	**X loc + Y toc**
System size (nodes)	18,700	18,700	3,400	**How fat?**
Node performance	125 GF	1.5 TF	40 TF	**depends (X,Y)**
Node memory BW	25 GB/s	25 - 200 GB/s	100 – 1000 GB/s	**10x fast vs slow**
Interconnect BW	1.5 GB/s	6.4 GB/s	25 GB/s	**4x each gen**
IO Bandwidth	0.2 TB/s	1 TB/s	1 TB/s	**flat**

Codesign

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)

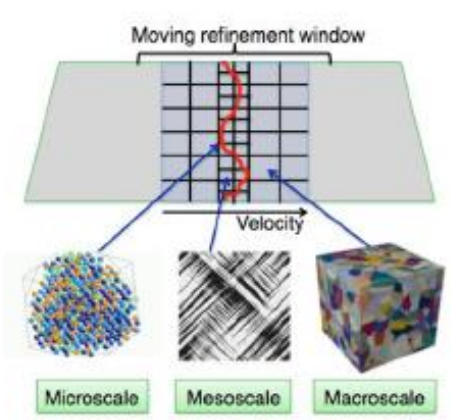
Center for Exascale Simulation of Advanced Reactors (CESAR)

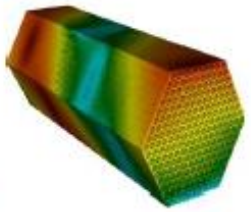
Andrew Siegel (ANL) r (ANL)

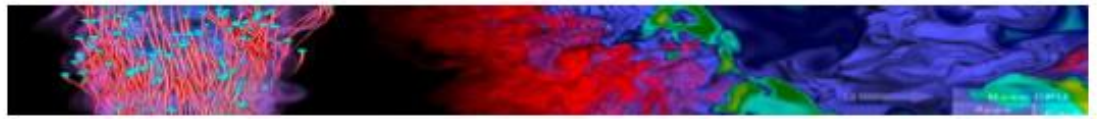
Center for Exascale Simulation of Combustion in Turbulence (EXaCT)

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	







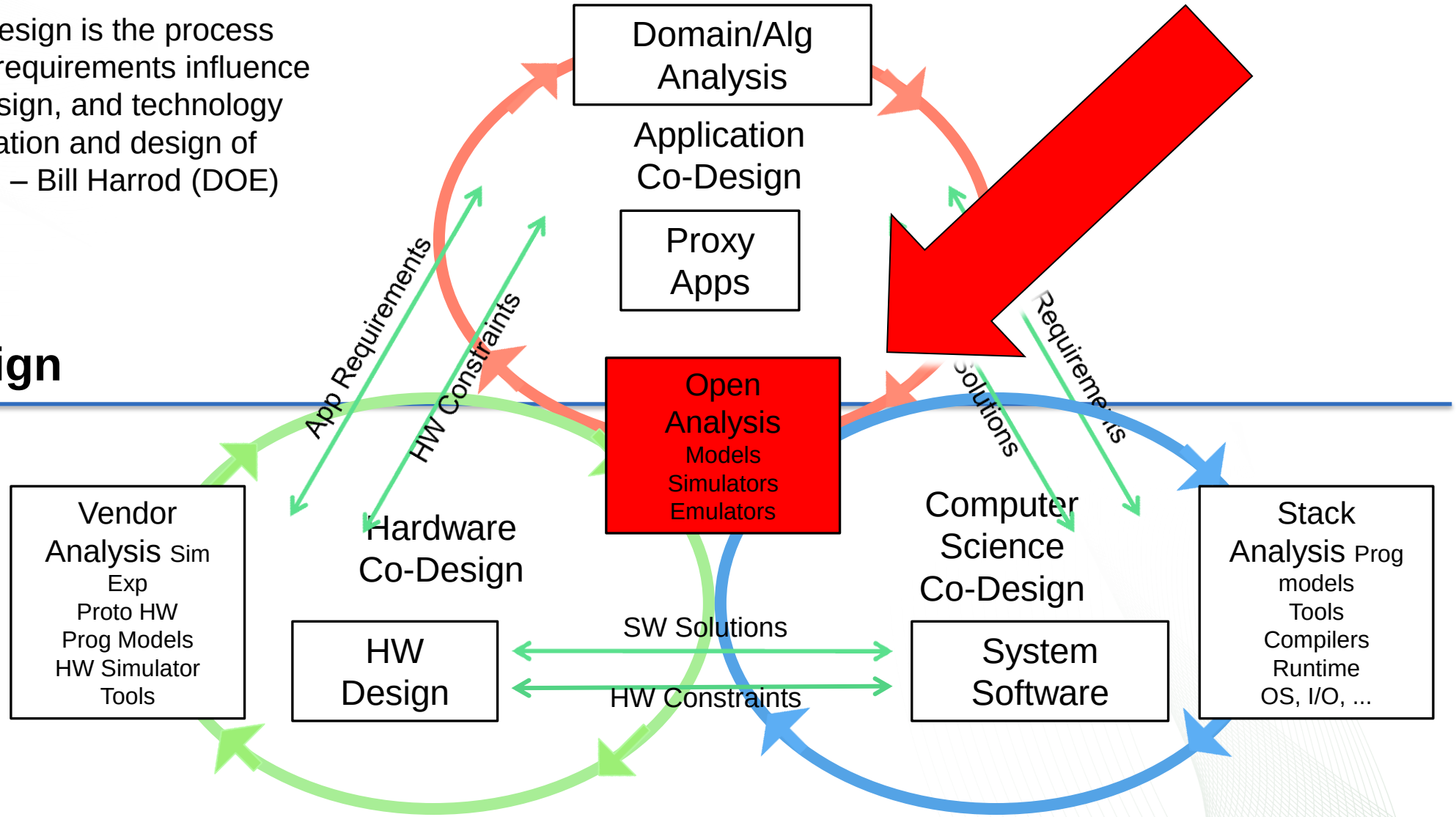
Each project is $4M/yr for 5 years, subject to satisfactory progress as gauged by frequent reviews

OAK RIDGE National Laboratory

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: Abstract Scalable Performance Engineering Notation

Model Creation

- Static analysis via compiler, tools
- Empirical, Historical
- Manual (for future applications)

Representation in Aspen

- **Modular**
- **Sharable**
- **Composable**
- **Reflects prog structure**

E.g., MD, UHPC CP 1, Lulesh, 3D FFT, CoMD, VPFFT, …

Model Uses

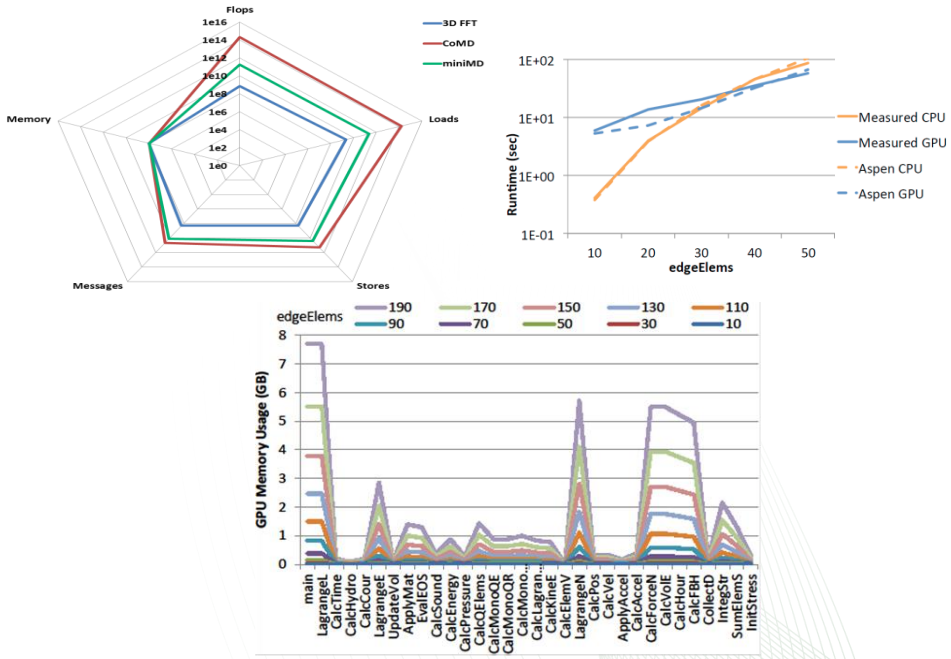
- Interactive tools for graphs, queries
- Design space exploration
- Workload Generation
- Feedback to Runtime Systems

Source code

```
static inline
void CalcMonotonicQGradientsForElems(Index_t p_nodelist[T_NUMELEM8],
        Real_t p_x[T_NUMNODE], Real_t p_y[T_NUMNODE], Real_t p_z[T_NUMNODE],
        Real_t p_xd[T_NUMNODE], Real_t p_yd[T_NUMNODE],Real_t p_zd[T_NUMNODE],
        Real_t p_volo[T_NUMELEM], Real_t p_vnew[T_NUMELEM],
        Real_t p_delx_zeta[T_NUMELEM], Real_t p_delv_zeta[T_NUMELEM],
        Real_t p_delx_xi[T_NUMELEM], Real_t p_delv_xi[T_NUMELEM],
        Real_t p_delx_eta[T_NUMELEM], Real_t p_delv_eta[T_NUMELEM])
{
    Index_t i;
    Index_t numElem = m_numElem;
#pragma acc parallel loop independent present(p_vnew, p_nodelist, p_x, p_y, p_z, p_xd, \
p_yd, p_zd, p_volo, p_delx_xi, p_delx_eta, p_delx_zeta, p_delv_xi, p_delv_eta,\
p_delv_zeta)
    for (i = 0 ; i < numElem ; ++i ) {
      const Real_t ptiny = 1.e-36 ;
      Real_t ax,ay,az ;
      Real_t dxv,dyv,dzv ;

      const Index_t *elemToNode = &p_nodelist[8*i];
      Index_t n0 = elemToNode[0] ;
      Index_t n1 = elemToNode[1] ;
      Index_t n2 = elemToNode[2] ;
      Index_t n3 = elemToNode[3] ;
      Index_t n4 = elemToNode[4] ;
      Index_t n5 = elemToNode[5] ;
      Index_t n6 = elemToNode[6] ;
      Index_t n7 = elemToNode[7] ;

      Real_t x0 = p_x[n0] ;
```

Aspen code

```
kernel CalcMonotonicQGradients {
  execute [numElems]
  {
    loads [8 * indexWordSize] from nodelist
    // Load and cache position and velocity.
    loads/caching [8 * wordSize] from x
    loads/caching [8 * wordSize] from y
    loads/caching [8 * wordSize] from z

    loads/caching [8 * wordSize] from xvel
    loads/caching [8 * wordSize] from yvel
    loads/caching [8 * wordSize] from zvel

    loads [wordSize] from volo
    loads [wordSize] from vnew
    // dx, dy, etc.
    flops [90] as dp, simd
    // delvk delxk
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delv_xeta
    // delxi delvi
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delx_xi
    // delxj and delvj
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delv_eta
  }
}
```



Researchers are using Aspen for parallel applications, scientific workflows, capacity planning, power, quantum computing, etc

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*

Manual Example of LULESH

branch: master ▾ aspen / models / lulesh / **lulesh.aspen**

jsmeredith on Sep 20, 2013 adding models

1 contributor

336 lines (288 sloc) | 9.213 kb Raw | Blame | History

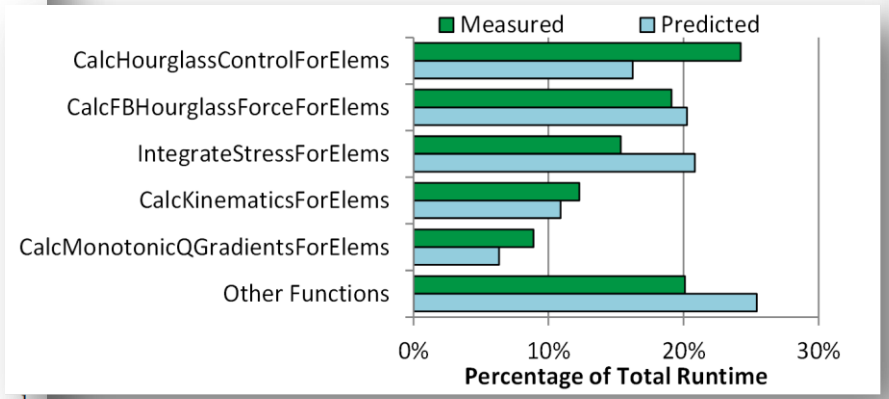
```
//
// lulesh.aspen
//
// An ASPEN application model for the LULESH 1.01 challenge problem.  Based
// on the CUDA version of the source code found at:
//  https://computation.llnl.gov/casc/ShockHydro/
//
param nTimeSteps = 1495

// Information about domain
param edgeElems = 45
param edgeNodes = edgeElems + 1

param numElems = edgeElems^3
param numNodes = edgeNodes^3

// Double precision
param wordSize = 8

// Element data
data mNodeList as Array(numElems, wordSize)
data mMatElemList as Array(numElems, wordSize)
data mNodeList as Array(8 * numElems, wordSize) //  8 nodes per element
data mlxim as Array(numElems, wordSize)
data mlxip as Array(numElems, wordSize)
data mletam as Array(numElems, wordSize)
data mletap as Array(numElems, wordSize)
data mzetam as Array(numElems, wordSize)
data mzetap as Array(numElems, wordSize)
data melemBC as Array(numElems, wordSize)
data mE as Array(numElems, wordSize)
data mP as Array(numElems, wordSize)
```

```
kernel CalcMonotonicQGradients {
  execute [numElems]
  {
    loads [8 * indexWordSize] from nodelist
    // Load and cache position and velocity.
    loads/caching [8 * wordSize] from x
    loads/caching [8 * wordSize] from y
    loads/caching [8 * wordSize] from z

    loads/caching [8 * wordSize] from xvel
    loads/caching [8 * wordSize] from yvel
    loads/caching [8 * wordSize] from zvel

    loads [wordSize] from volo
    loads [wordSize] from vnew
    // dx, dy, etc.
    flops [90] as dp, simd
    // delvk delxk
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delv_xeta
    // delxi delvi
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delx_xi
    // delxj and delvj
    flops [9 + 8 + 3 + 30 + 5] as dp, simd
    stores [wordSize] to delv_eta
  }
}
```

Example Uses: Resource Exploration

Benchmark	Runtime Order
BACKPROP	$H * O + H * I$
BFS	$nodes + edges$
CFD	$nelr * ndim$
CG	$nrow + ncol$
HOTSPOT	$sim_time * rows * cols$
JACOBI	$m\_size * m\_size$
KMEANS	$nAttr * nClusters$
LAPLACE2D	n^2
LUD	$matrix\_dim^3$
MATMUL	$N * M * P$
NW	max_cols^2
SPMUL	$size + nonzero$
SRAD	$niter * rows * cols$

Table 2: Order analysis, showing Big O runtime for each benchmark in terms of its key parameters.



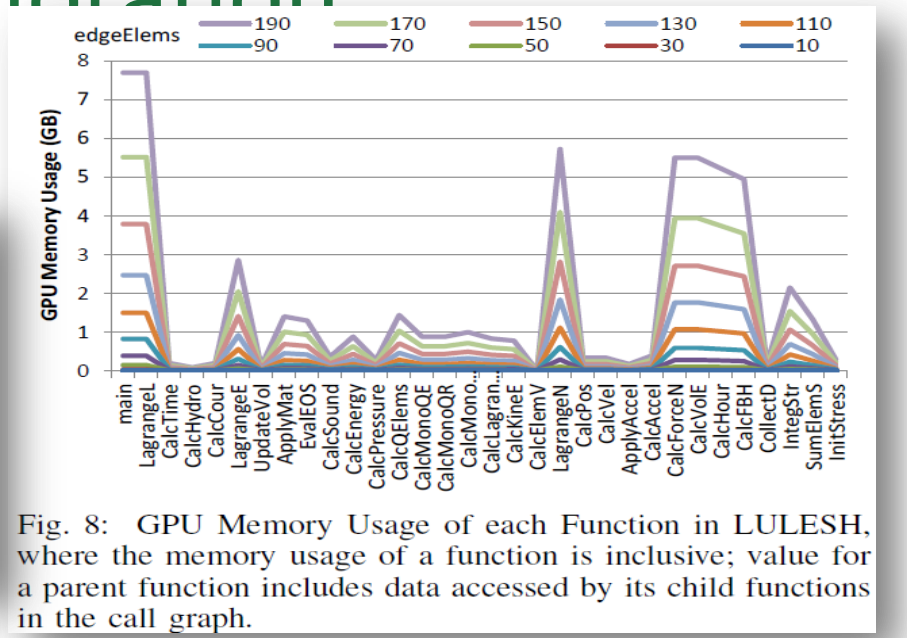


Fig. 8: GPU Memory Usage of each Function in LULESH, where the memory usage of a function is inclusive; value for a parent function includes data accessed by its child functions in the call graph.

Method Name	FLOPS/byte
InitStressTermsForElems	0.03
CalcElemShapeFunctionDerivatives	0.44
SumElemFaceNormal	0.50
CalcElemNodeNormals	0.15
SumElemStressesToNodeForces	0.06
IntegrateStressForElems	0.15
CollectDomainNodesToElemNodes	0.00
VoluDer	1.50
CalcElemVolumeDerivative	0.33
CalcElemFBHourglassForce	0.15
CalcFBHourglassForceForElems	0.17
CalcHourglassControlForElems	0.19
CalcVolumeForceForElems	0.18
CalcForceForNodes	0.18
CalcAccelerationForNodes	0.04
ApplyAccelerationBoundaryCond	0.00
CalcVelocityForNodes	0.13
CalcPositionForNodes	0.13
LagrangeNodal	0.18
AreaFace	10.25
CalcElemCharacteristicLength	0.44
CalcElemVelocityGrandient	0.13
CalcKinematicsForElems	0.24
CalcLagrangeElements	0.24
CalcMonotonicQGradientsForElems	0.46

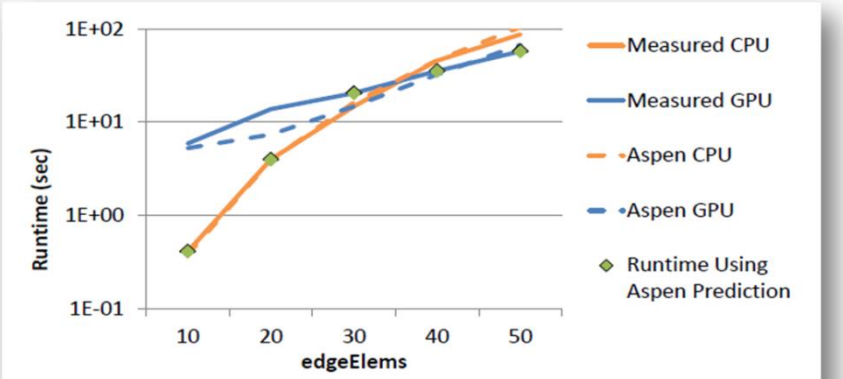


Fig. 7: Measured and predicted runtime of the entire LULESH program on CPU and GPU, including measured runtimes using the automatically predicted optimal target device at each size.

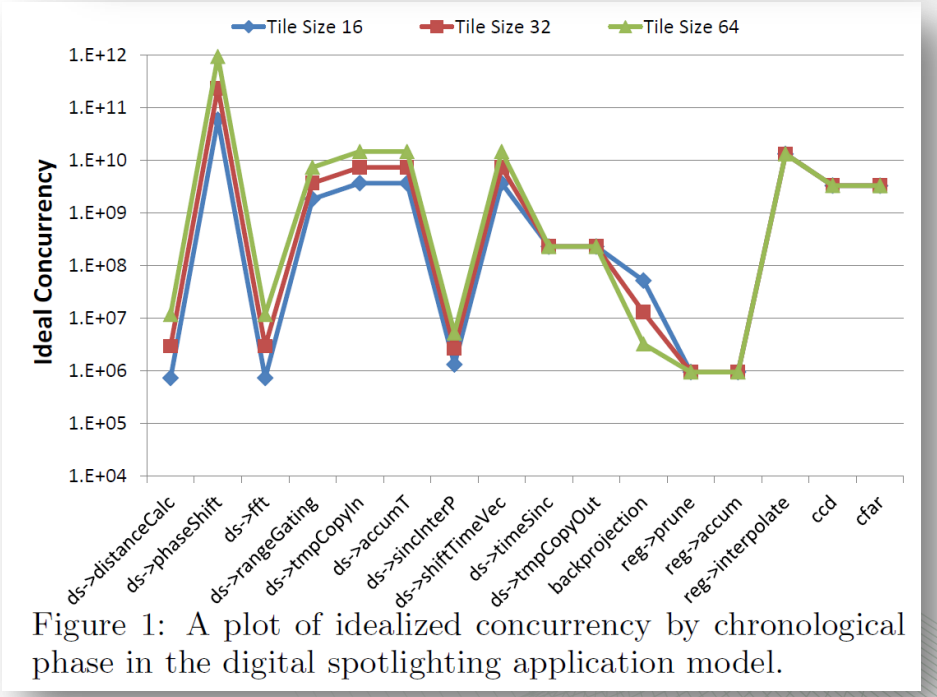
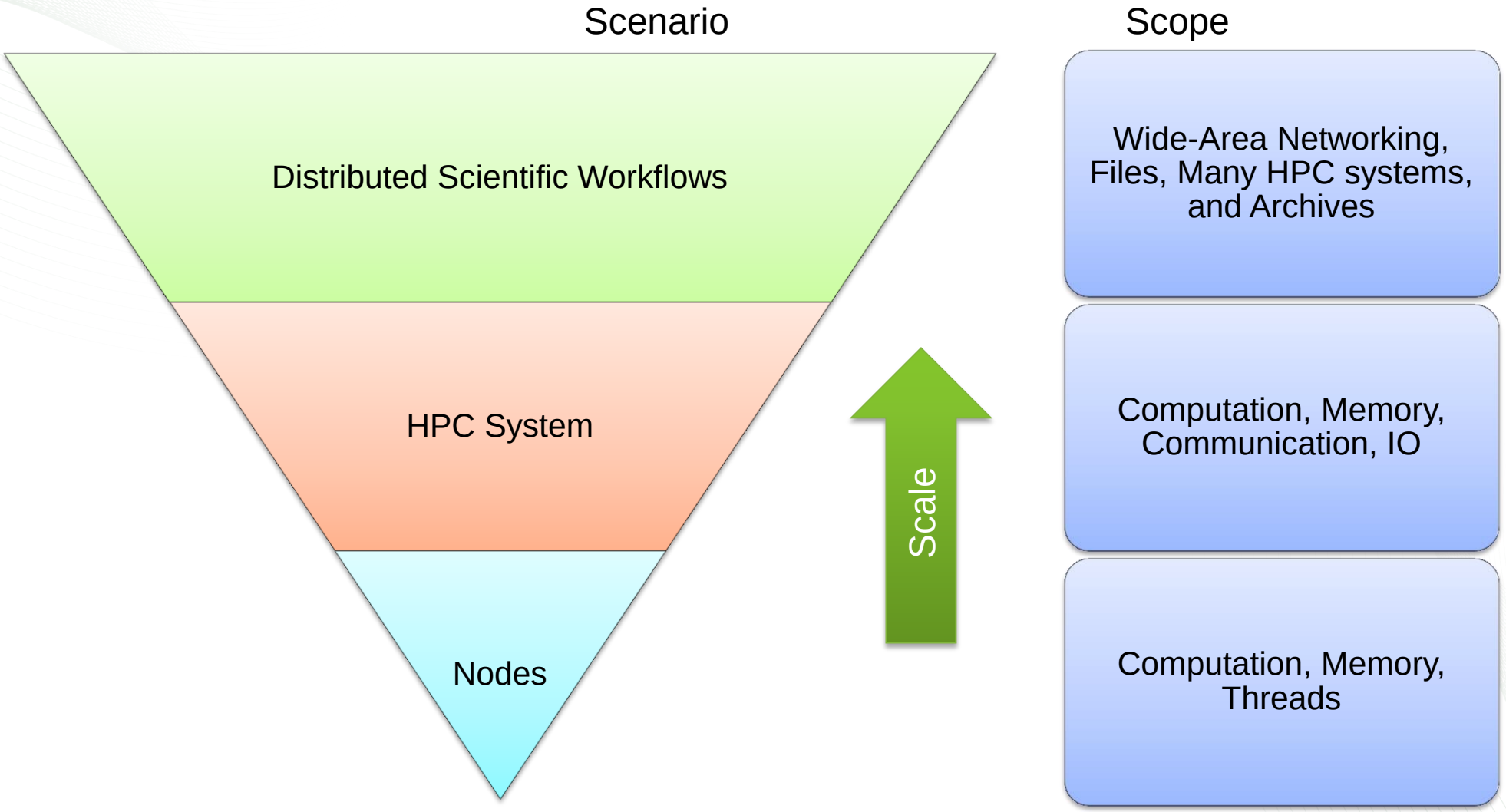


Figure 1: A plot of idealized concurrency by chronological phase in the digital spotlighting application model.

Aspen allows Multiresolution Modeling

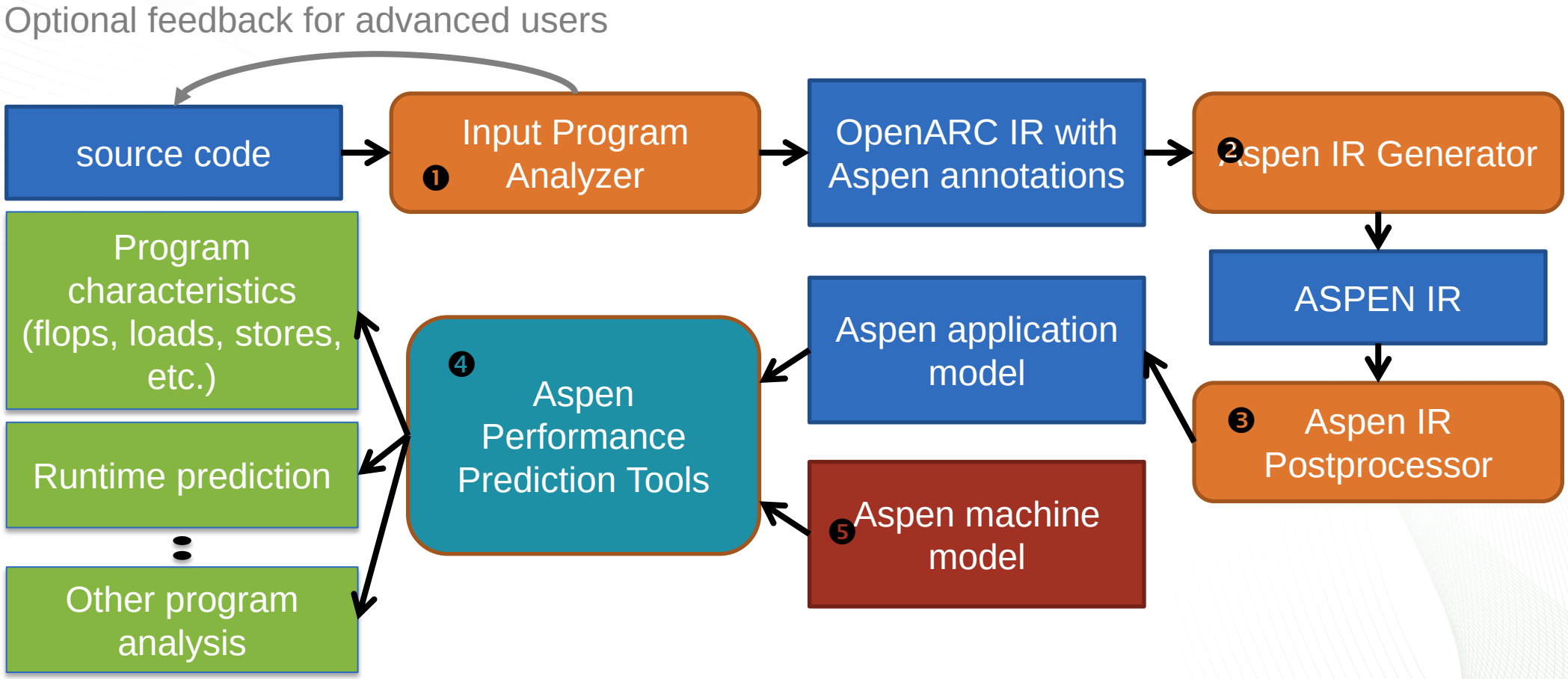


Node Scale Modeling with COMPASS

OAK RIDGE National Laboratory

COMPASS System Overview

- Detailed Workflow of the COMPASS Modeling Framework



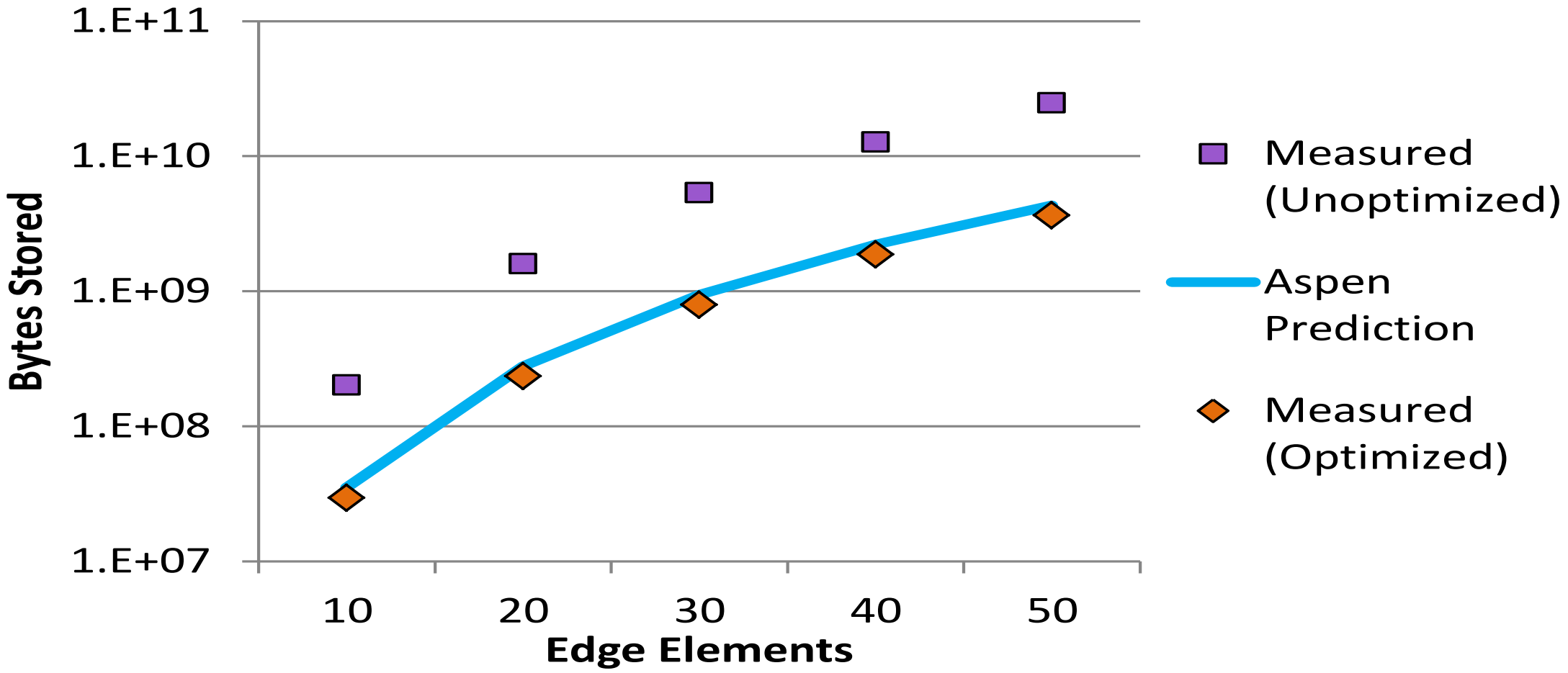
S. Lee, J.S. Meredith, and J.S. Vetter, "COMPASS: A Framework for Automated Performance Modeling and Prediction," in *ACM International Conference on Supercomputing (ICS). Newport Beach, California: ACM, 2015, 10.1145/2751205.2751220.*

Example: LULESH (10% of 1 kernel)

```
kernel IntegrateStressForElems
{
  execute [numElem_CalcVolumeForceForElems]
  {
    loads [((1*aspen_param_int)*8)] from elemNodes as stride(1)
    loads [((1*aspen_param_double)*8)] from m_x
    loads [((1*aspen_param_double)*8)] from m_y
    loads [((1*aspen_param_double)*8)] from m_z
    loads [(1*aspen_param_double)] from determ as stride(1)
    flops [8] as dp, simd
    flops [8] as dp, simd
    flops [8] as dp, simd
    flops [8] as dp, simd
    flops [3] as dp, simd
    flops [3] as dp, simd
    flops [3] as dp, simd
    flops [3] as dp, simd
    stores [(1*aspen_param_double)] as stride(0)
    flops [2] as dp, simd
    stores [(1*aspen_param_double)] as stride(0)
    flops [2] as dp, simd
    stores [(1*aspen_param_double)] as stride(0)
    flops [2] as dp, simd
    loads [(1*aspen_param_double)] as stride(0)
    stores [(1*aspen_param_double)] as stride(0)
    loads [(1*aspen_param_double)] as stride(0)
    stores [(1*aspen_param_double)] as stride(0)
    loads [(1*aspen_param_double)] as stride(0)
    ......
```

- Input LULESH program: 3700 lines of C codes
- Output Aspen model: 2300 lines of Aspen codes

Model Scaling Validation (LULESH)



Model Validation

	FLOPS	LOADS	STORES
MATMUL	15%	<1%	1%
LAPLACE2D	7%	0%	<1%
SRAD	17%	0%	0%
JACOBI	6%	<1%	<1%
KMEANS	0%	0%	8%
LUD	5%	0%	2%
BFS	<1%	11%	0%
HOTSPOT	0%	0%	0%
LULESH	0%	0%	0%

0% means that prediction fell between measurements from optimized and unoptimized runs of the code.

Key Messages

- Get ready for Heterogeneous Computing and Nonvolatile memory
- Explore programming models to provide performance portability
- Develop and use new tools for performance prediction and planning
  - Codesign needs a common language for describing and using resources.
  - Codesign needs to focus on long term, not short term

Acknowledgements

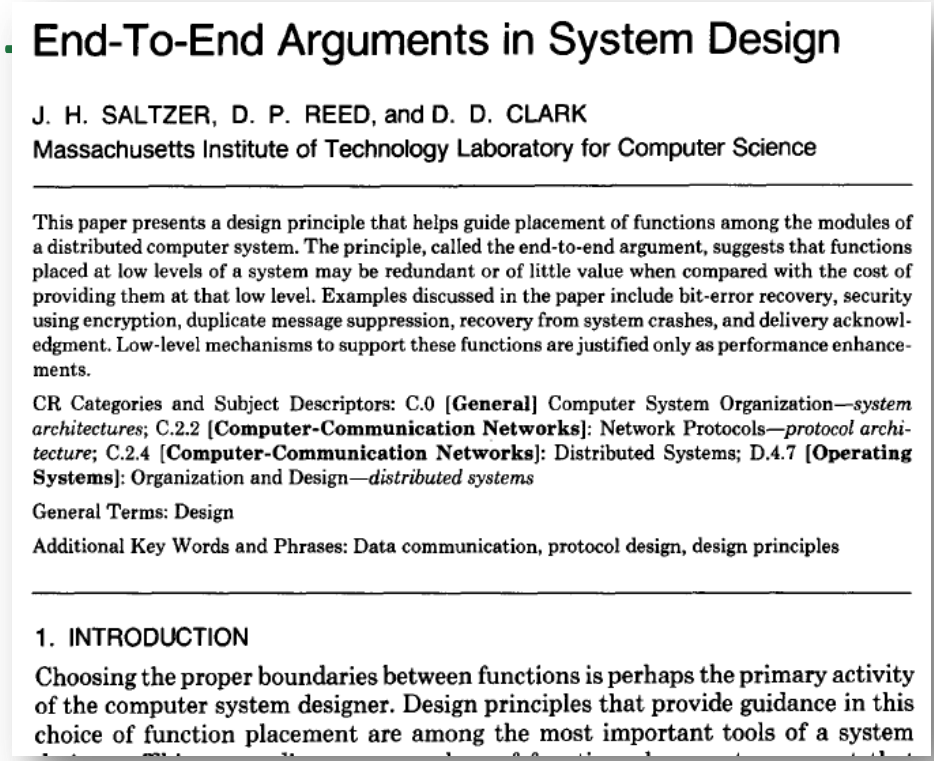
- Contributors and Sponsors
  - Future Technologies Group: http://ft.ornl.gov
  - US Department of Energy Office of Science
    - DOE Vancouver Project: https://ft.ornl.gov/trac/vancouver
    - DOE Blackcomb Project: https://ft.ornl.gov/trac/blackcomb
    - DOE ExMatEx Codesign Center: http://codesign.lanl.gov
    - DOE Cesar Codesign Center: http://cesar.mcs.anl.gov/
    - DOE Exascale Efforts: http://science.energy.gov/ascr/research/computer-science/
  - Scalable Heterogeneous Computing Benchmark team: http://bit.ly/shocmarx
  - US National Science Foundation Keeneland Project: http://keeneland.gatech.edu
  - US DARPA
  - NVIDIA CUDA Center of Excellence

End-to-end Resiliency Design using Aspen

OAK RIDGE National Laboratory

Resiliency Modeling wi

- End-to-End system design for Extreme-scale HPC
  - Why pay redundant costs for power, performance, etc?
- We introduce a new metric, the data vulnerability factor (DVF)
  - Quantifying vulnerability of data structures
  - Avoiding the isolation between application and hardware
- We measure DVF based on Aspen, a domain specific language for system modeling
- We categorize memory access patterns of scientific applications from a spectrum of computational domains
  - Dense linear algebra, Sparse linear algebra, N-body method, Structured grids, Spectral methods, and Monte Carlo
- We demonstrate the significance of DVF by two case studies
  - Algorithm optimization
  - Data protection quantification



End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK

Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**] Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

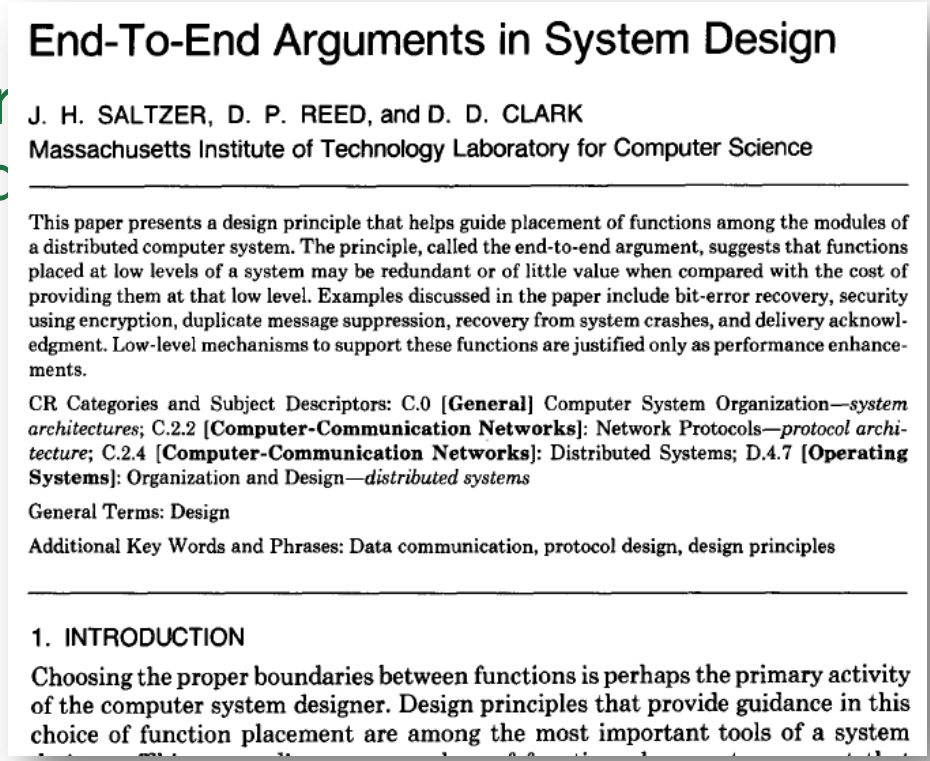
Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system

DVF_d	DVF for a specific data structure
FIT	Failure rate (i.e., failures per billion hours per Mbit)
T	Application execution time
S_d	Size of data structure
N_{error}	Number of errors that could occur to a specific data structure during application execution
N_{ha}	Number of accesses to hardware (the main memory in this work)
n	Number of major data structures in an application
DVF_a	DVF for the application

L. Yu, D. Li *et al., "Quantitatively modeling application resilience with the data vulnerability factor (Best Student Paper Finalist)," in SC14: International Conference for High Performance Computing, Networking, Storage and Analysis. New Orleans, Louisiana: IEEE Press, 2014, pp. 695-706, 10.1109/sc.2014.62.*

Data Vulnerability Factor
Why a new metric and method

- Analytical model of resiliency that includes important features of architecture and application
  - Fast
  - Flexible
- Balance multiple design dimensions
  - Application requirements
  - Architecture (memory capacity and type)
- Focus on main memory initially
- Prioritize vulnerabilities of application data



End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

CR Categories and Subject Descriptors: C.0 [**General**] Computer System Organization—*system architectures*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.7 [**Operating Systems**]: Organization and Design—*distributed systems*

General Terms: Design

Additional Key Words and Phrases: Data communication, protocol design, design principles

1. INTRODUCTION

Choosing the proper boundaries between functions is perhaps the primary activity of the computer system designer. Design principles that provide guidance in this choice of function placement are among the most important tools of a system

DVF_d	DVF for a specific data structure
FIT	Failure rate (i.e., failures per billion hours per Mbit)
T	Application execution time
S_d	Size of data structure
N_{error}	Number of errors that could occur to a specific data structure during application execution
N_{ha}	Number of accesses to hardware (the main memory in this work)
n	Number of major data structures in an application
DVF_a	DVF for the application

L. Yu, D. Li *et al., "Quantitatively modeling application resilience with the data vulnerability factor (Best Student Paper Finalist)," in SC14: International Conference for High Performance Computing, Networking, Storage and Analysis. New Orleans, Louisiana: IEEE Press, 2014, pp. 695-706, 10.1109/sc.2014.62.*

Workflow to calculate Data Vulnerability Factor

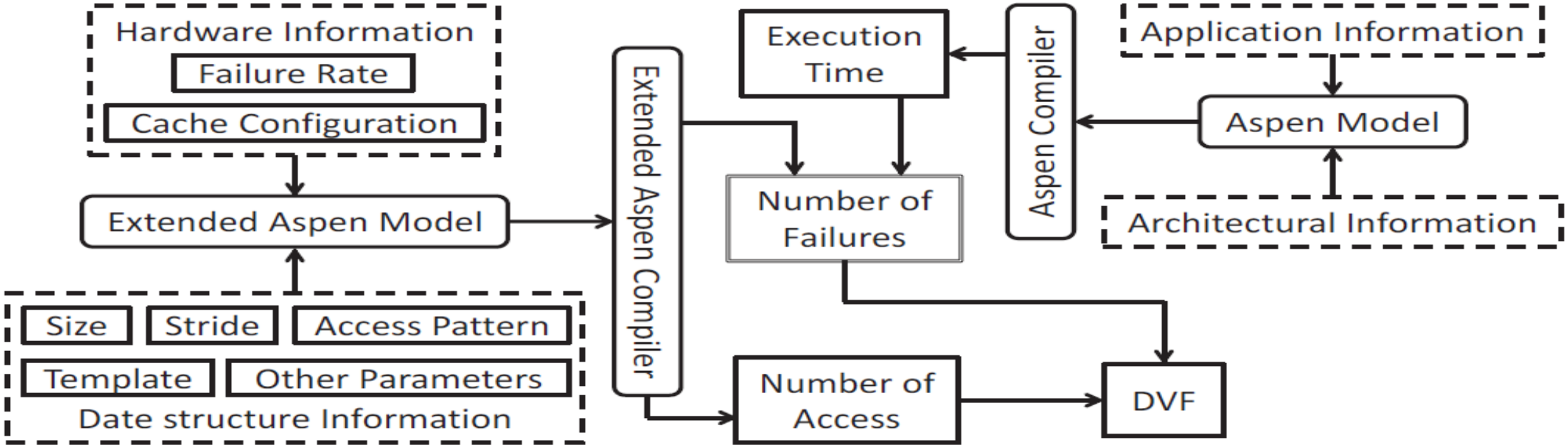


Fig. 3. The workflow to calculate DVF.

An Example of Aspen Program for DVF

```
procedure VM(A,B,C)
    for i ← 1, 1000 do
        C[i] ← C[i] + A[i*4] * B[i*8]
    end for
end procedure
```

Pseudocode

```
kernel vecmul {
    execute mainblock2 [1]
    {
    flops [2*(n^3)] as sp, fmad, simd
    access {1000} from {matA} as stream(4,16)
    access {4000} from {matB} as stream(4,32)
    access {8000} from {matC} as stream(4,4)
    }
}
```

Extended Aspen Statements

Extended Parser

```
Resilience Statements:
    Footprint Sizes:
        Int: 16,000
    Data Structures:
        Ident: matA
    Access Pattern: Stream
        Int: 4
        Int: 16
Resilience Statements:
    Footprint Sizes:
        Int: 16,000
    Data Structures:
        Ident: matA
    Access Pattern: Stream
        Int: 4
        Int: 16
Resilience Statements:
    Footprint Sizes:
        Int: 16,000
    Data Structures:
        Ident: matA
    Access Pattern: Stream
        Int: 4
        Int: 16
```

Syntax Tree

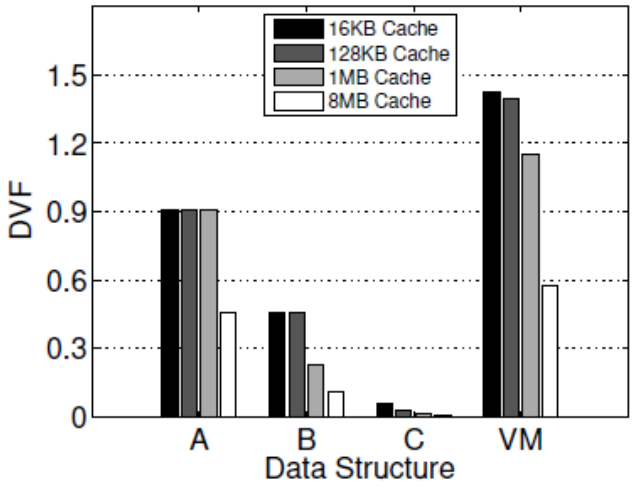
Extended Complier

```
Data structure A:
Number of errors: 30,400
Number of memory accesses: 51
DVF: 105504e+06
…
```

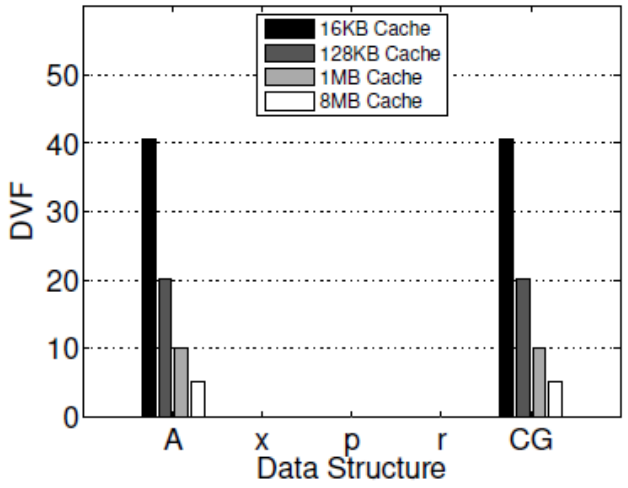
Resilience Modeling Results

DVF Results

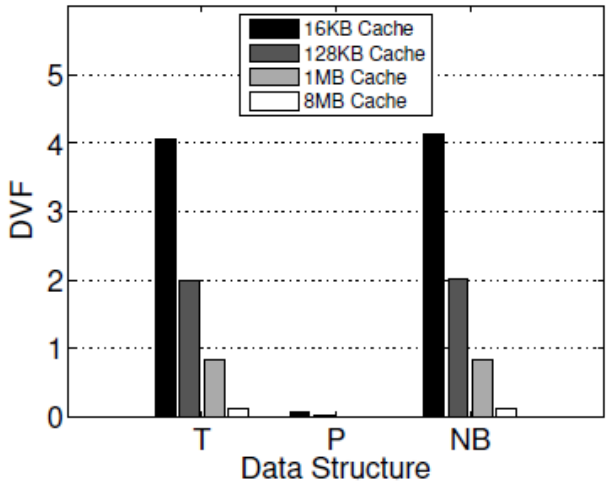
Provides insight for balancing interacting factors



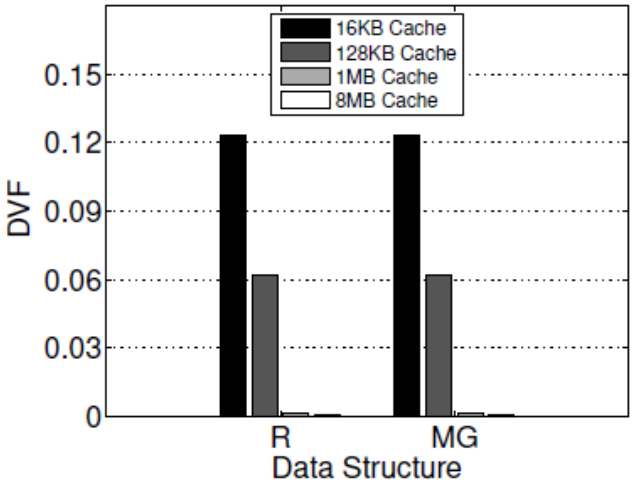
(a) Vector Multiplication



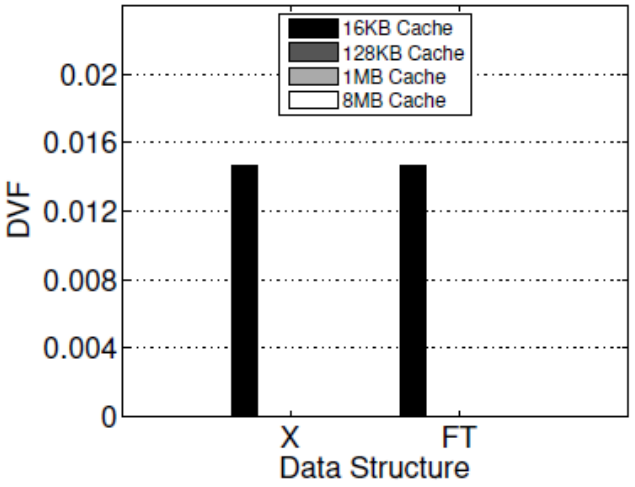
(b) Conjugate Gradient



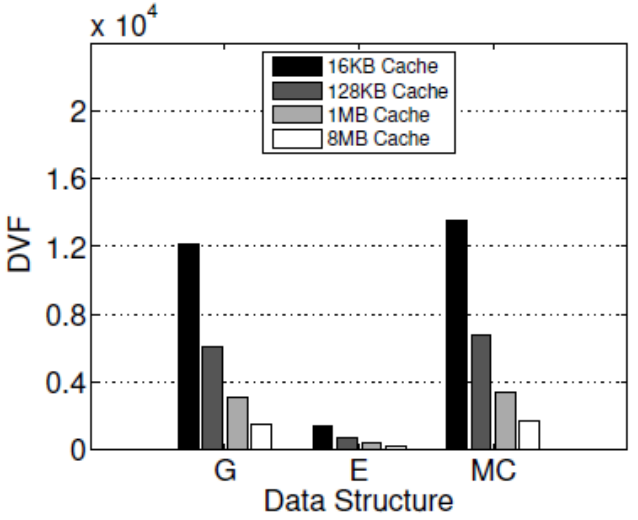
(c) Nbody (Barnes-hut)



(d) Multi-grid



(e) 1D FFT



(f) Monte Carlo

DVF: next steps

- Evaluated different architectures
  - How much no-ECC, ECC, NVM?
- Evaluate software and applications
  - ABFT
  - C/R
  - TMR
  - Containment domains
  - Fault tolerant MPI
- End-to-End analysis
  - Where should we bear the cost for resiliency?
    - Not everwhere!

58