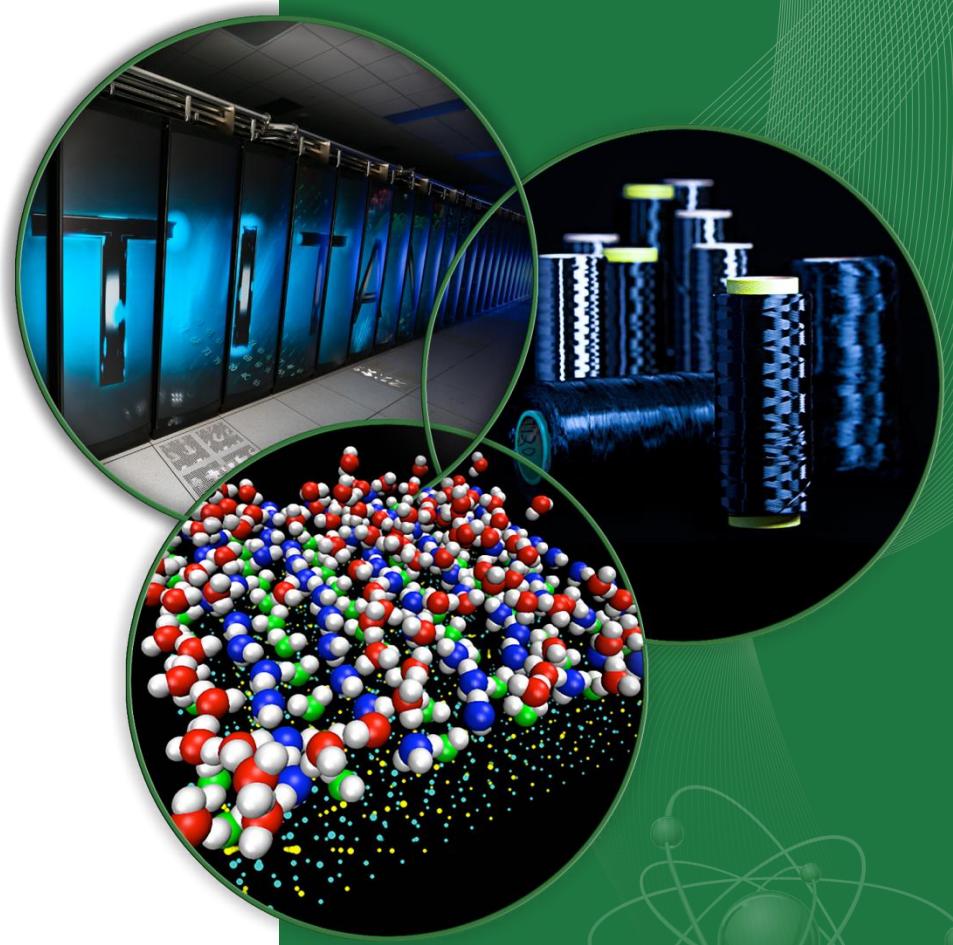
Exploring Emerging Technologies in the Extreme Scale HPC Co-Design Space

Jeffrey S. Vetter



11th International Conference on Parallel Processing and Applied Mathematics (PPAM)

Krakow

8 Sep 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

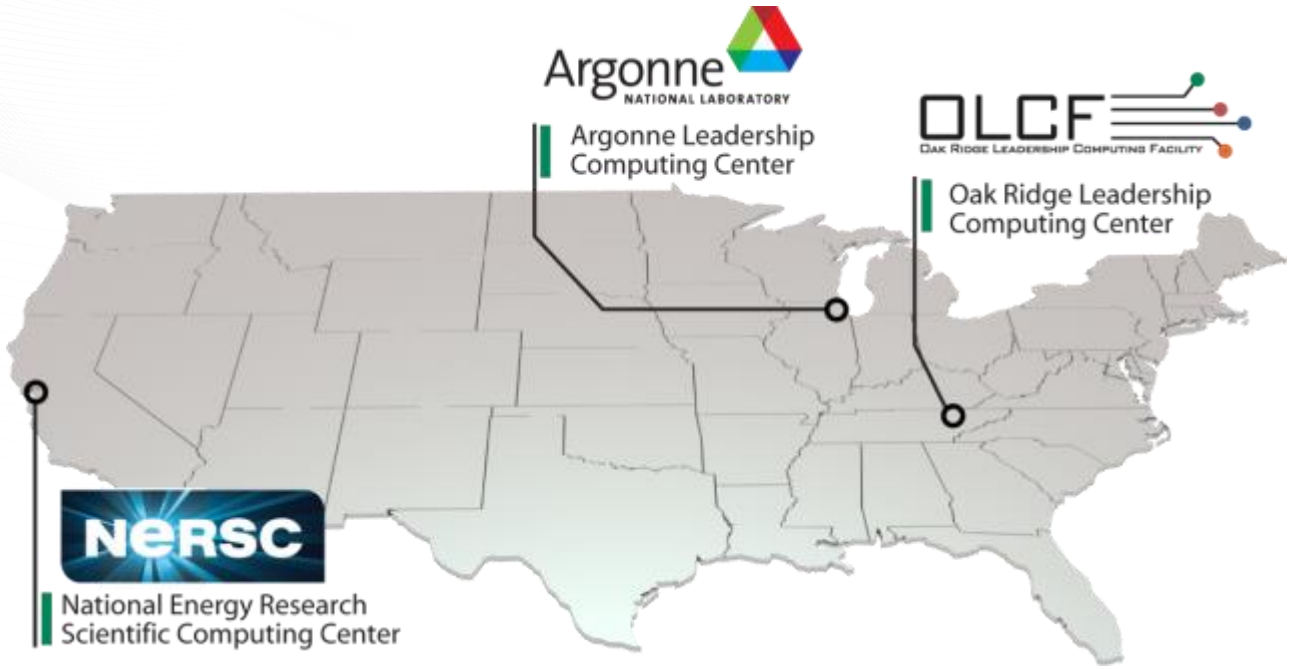
http://ft.ornl.gov ♦ vetter@computer.org

OAK RIDGE National Laboratory

Overview

- Our community has major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - New technologies emerging to address some of these challenges
    - Heterogeneous computing
    - Multimode memory systems including nonvolatile memory
  - Not just HPC: Most uncertainty in at least two decades
  - Exascale includes even more diversity and uncertainty
- We need performance prediction and portability 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
- OpenARC research compiler is a vehicle for
  - Understanding how to automate platform specific optimizations
  - Developing performance portable code

DOE's Office of Science Computation User Facilities



- DOE is leader in open High-Performance Computing
- Provide the world's most powerful computational tools for open science
- Access is free to researchers who publish
- Boost US competitiveness
- Attract the best and brightest researchers

NERSC
Edison is 2.57 PF

ALCF
Mira is 10 PF

OLCF
Titan is 27 PF

OAK RIDGE National Laboratory

HPC Trends


ELLIOTT
ELECTRONIC COMPUTER
WESTWOOD


the 10-Megabyte Computer System
Only $5995
COMPLETE
New From IMSAI
• 10-Megabyte Hard Disk
• 5¼" Dual-Density Floppy Disk Back-up
• 8-Bit Microprocessor (Optional 16-bit Microprocessor)
• Memory-Mapped Video Display Board
• Disk Controller
• Standard 64K RAM (Optional 256K RAM)
• 10-Slot S-100 Motherboard
• 28-Amp Power Supply
• 12" Monitor
• Standard Intelligent 62-Key ASCII Keyboard (Optional Intelligent 86-Key ASCII Extended Keyboard)
• 132-Column Dot-Matrix Printer
• CP/M* Operating System
You Read It Right ... All for $5995!
IMSAI ...Thinking ahead for the 80's
415/635-7615
Computer Division of the Fischer-Freitas Corporation
910 81st Avenue, Bldg. 14 • Oakland, CA 94621
*CP/M is a trademark of Digital Research. Imsai is a trademark of the Fischer-Freitas Corporation

Exascale architecture targets circa 2009

2009 Exascale Challenges Workshop in San Diego

Attendees envisioned two possible architectural swim lanes:

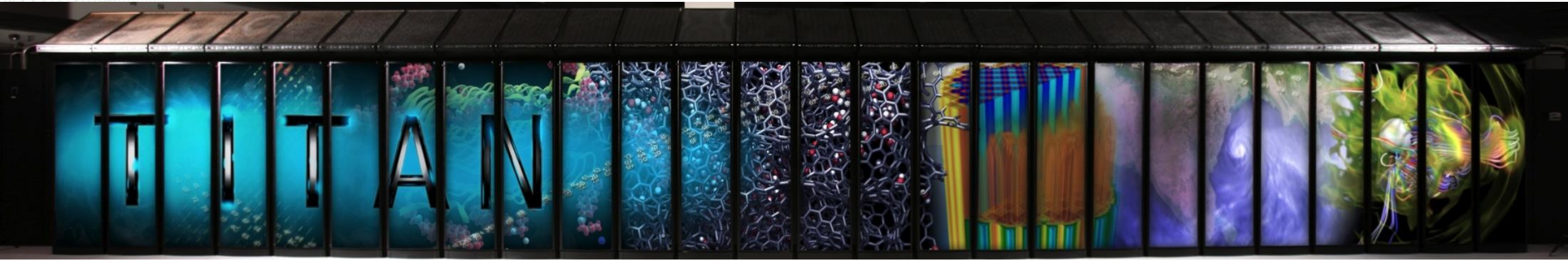
1. Homogeneous many-core thin-node system
2. Heterogeneous (accelerator + CPU) fat-node system

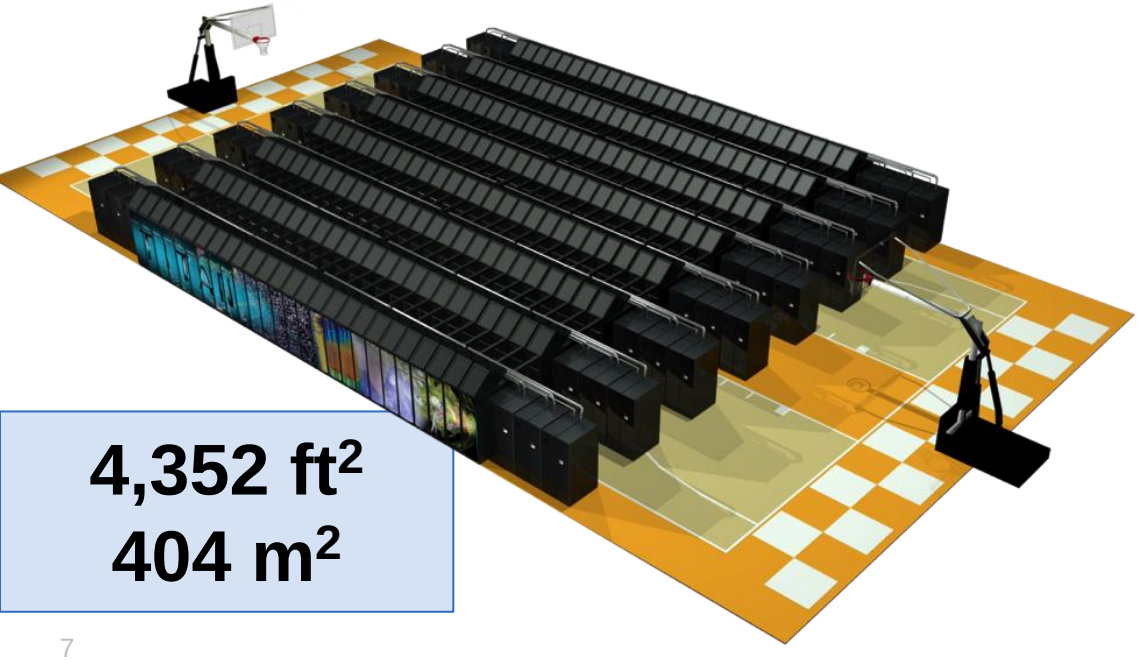
System attributes	2009	"Pre-Exascale"		"Exascale"	
System peak	2 PF	100-200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Storage	15 PB	150 PB		500 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	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)	

ASCR Computing At a Glance

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrades	
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®

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





4,352 ft²
404 m²

SYSTEM SPECIFICATIONS:

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

OAK RIDGE National Laboratory

2017 OLCF Leadership System

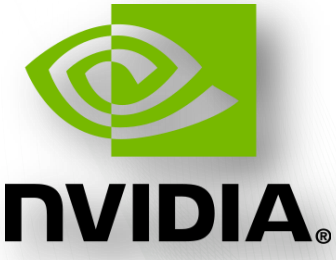
Hybrid CPU/GPU architecture

Vendor: IBM (Prime) / NVIDIA™ / Mellanox Technologies®

At least 5X Titan's Application Performance

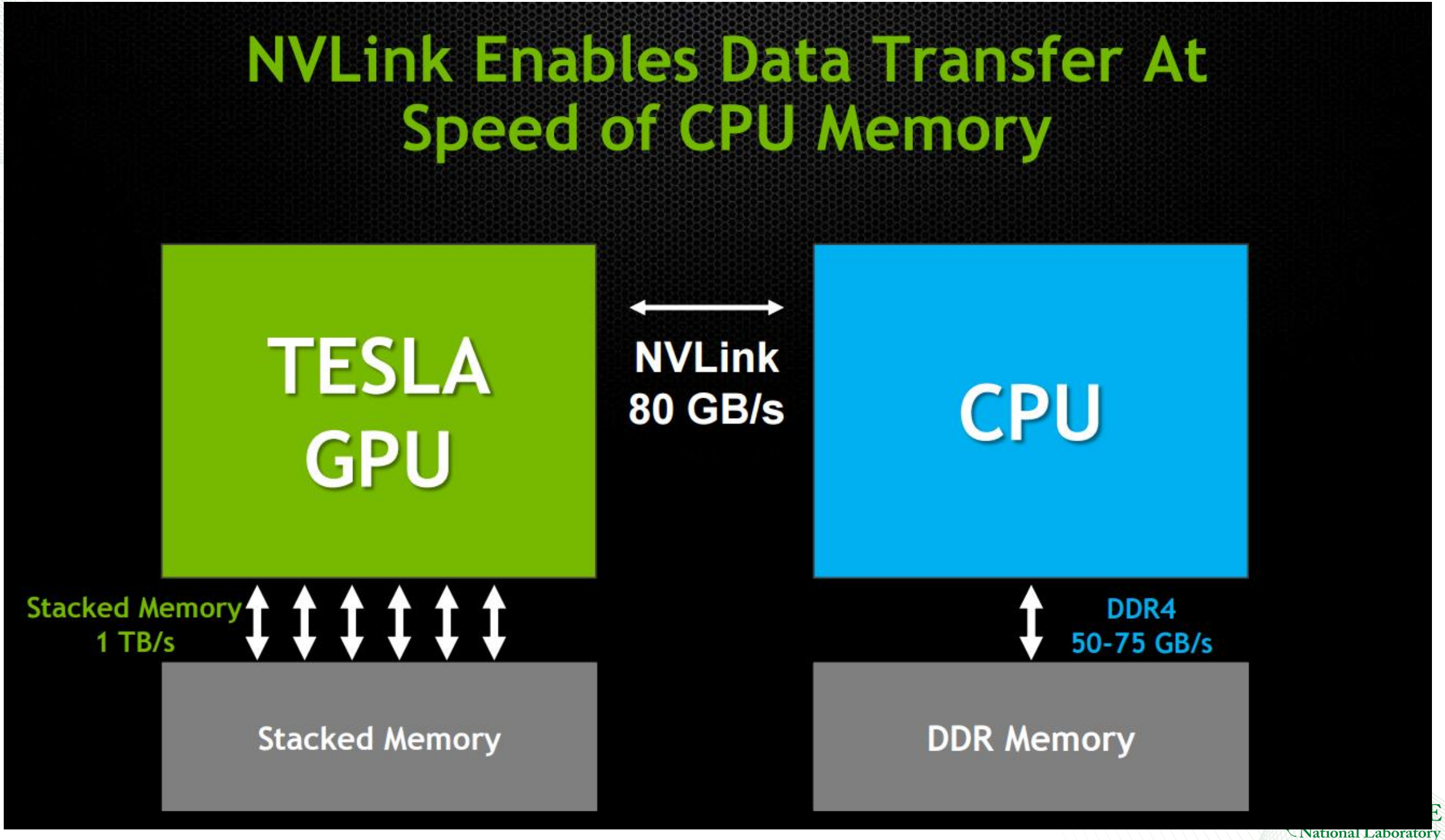
Approximately 3,400 nodes, each with:

- Multiple IBM POWER9 CPUs and multiple NVIDIA Tesla® GPUs using the NVIDIA Volta architecture
- CPUs and GPUs completely connected with high speed NVLink
- Large coherent memory: over 512 GB (HBM + DDR4)
  - all directly addressable from the CPUs and GPUs
- An additional 800 GB of NVRAM, either a burst buffer or as extended memory
- Over 40 TF peak performance



Dual-rail Mellanox® EDR-IB full, non-blocking fat-tree interconnect

IBM Elastic Storage (GPFS™) - 1TB/s I/O and 120 PB disk capacity.

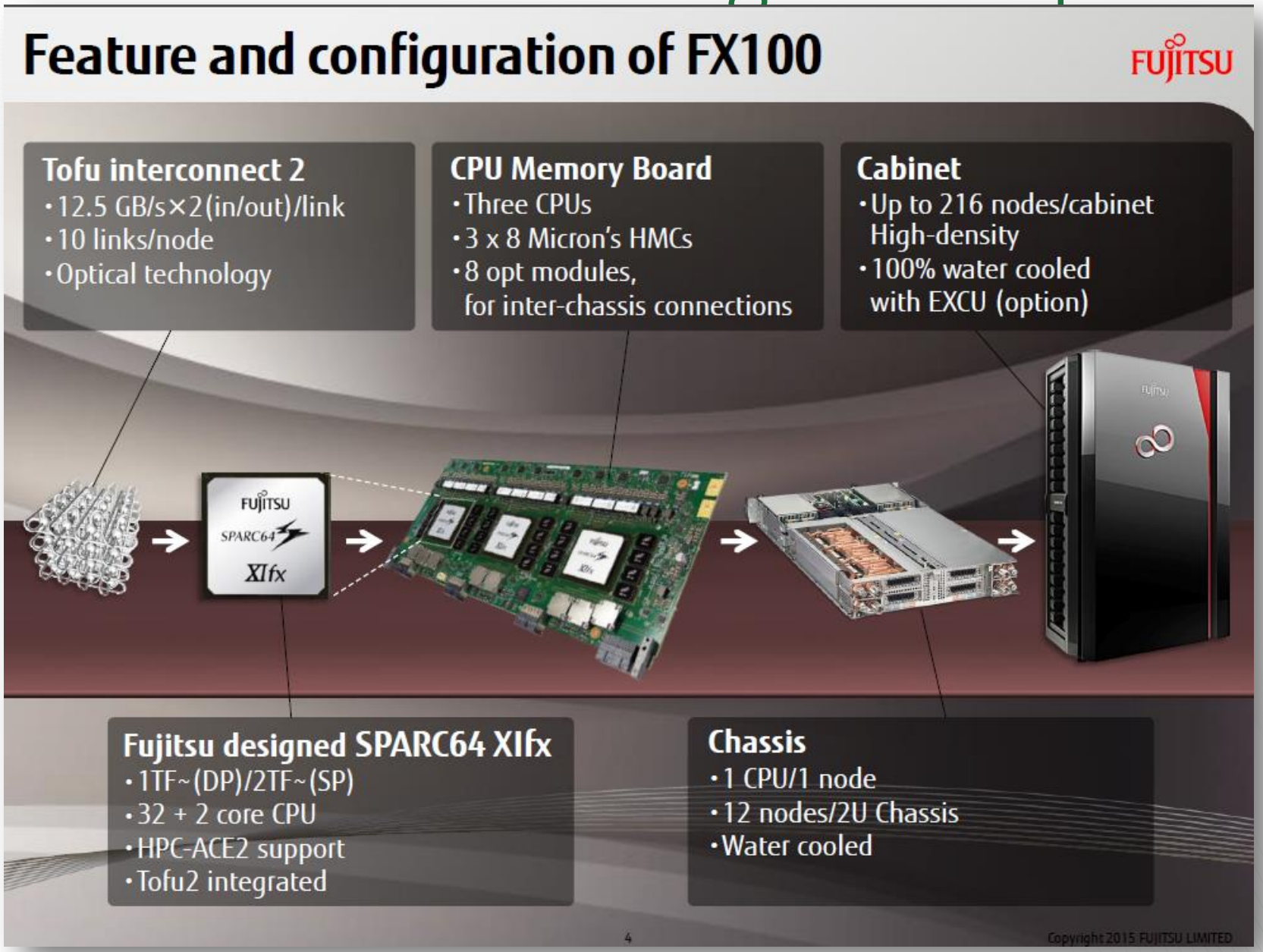


https://hpcuserforum.com/presentations/santafe2014/NVidia%20Disruptive.pdf

OLCF-5 Projections

				OLCF-5
Date	**2009**	**2012**	**2017**	**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**

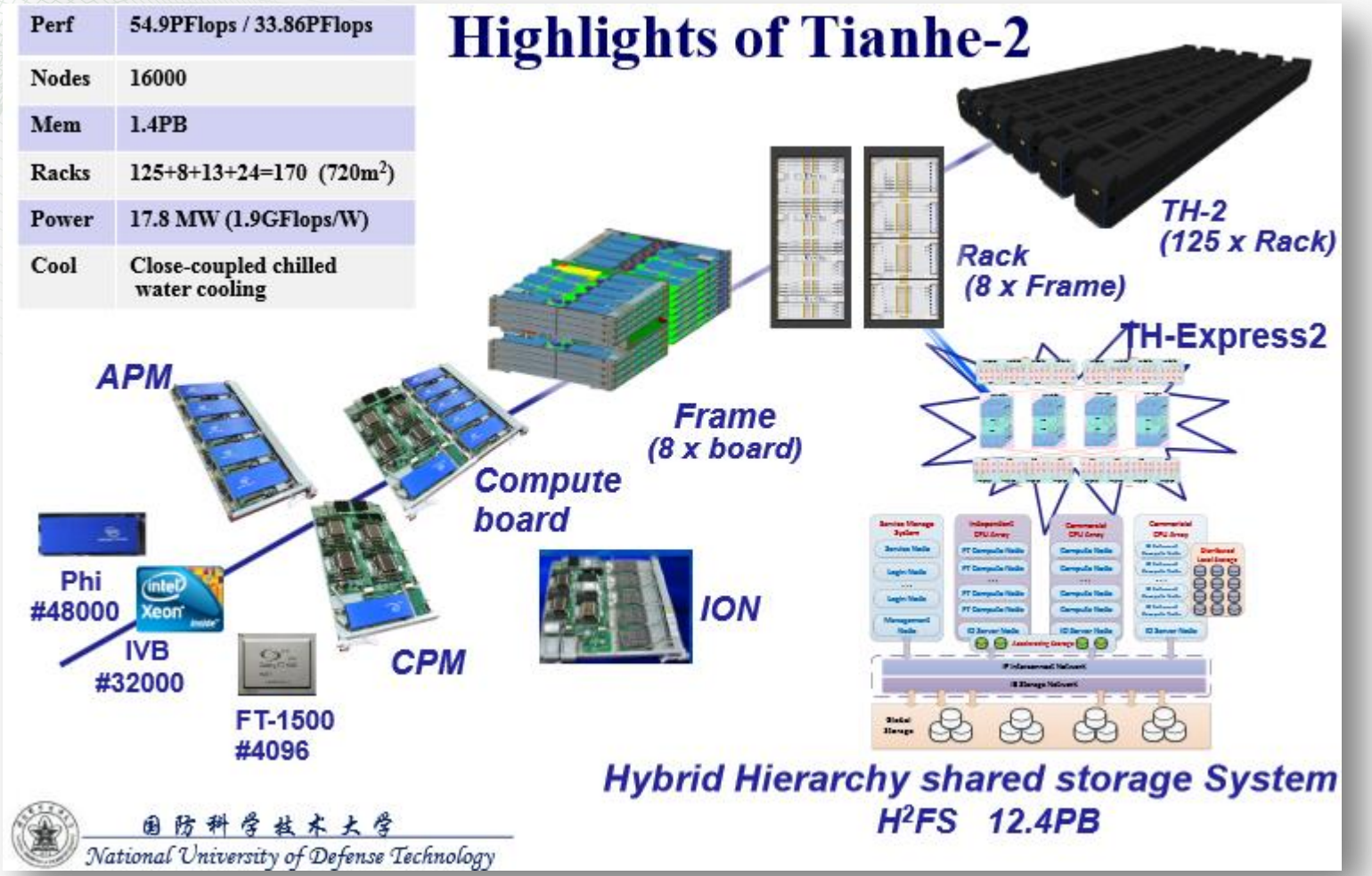
International Progress: Japan



Slide courtesy of Fujitsu, RIKEN

International Progress: China

Perf	54.9PFlops / 33.86PFlops
Nodes	16000
Mem	1.4PB
Racks	125+8+13+24=170 (720m^2)
Power	17.8 MW (1.9GFlops/W)
Cool	Close-coupled chilled water cooling



China May Develop Two 100 Petaflop Machines Within a Year

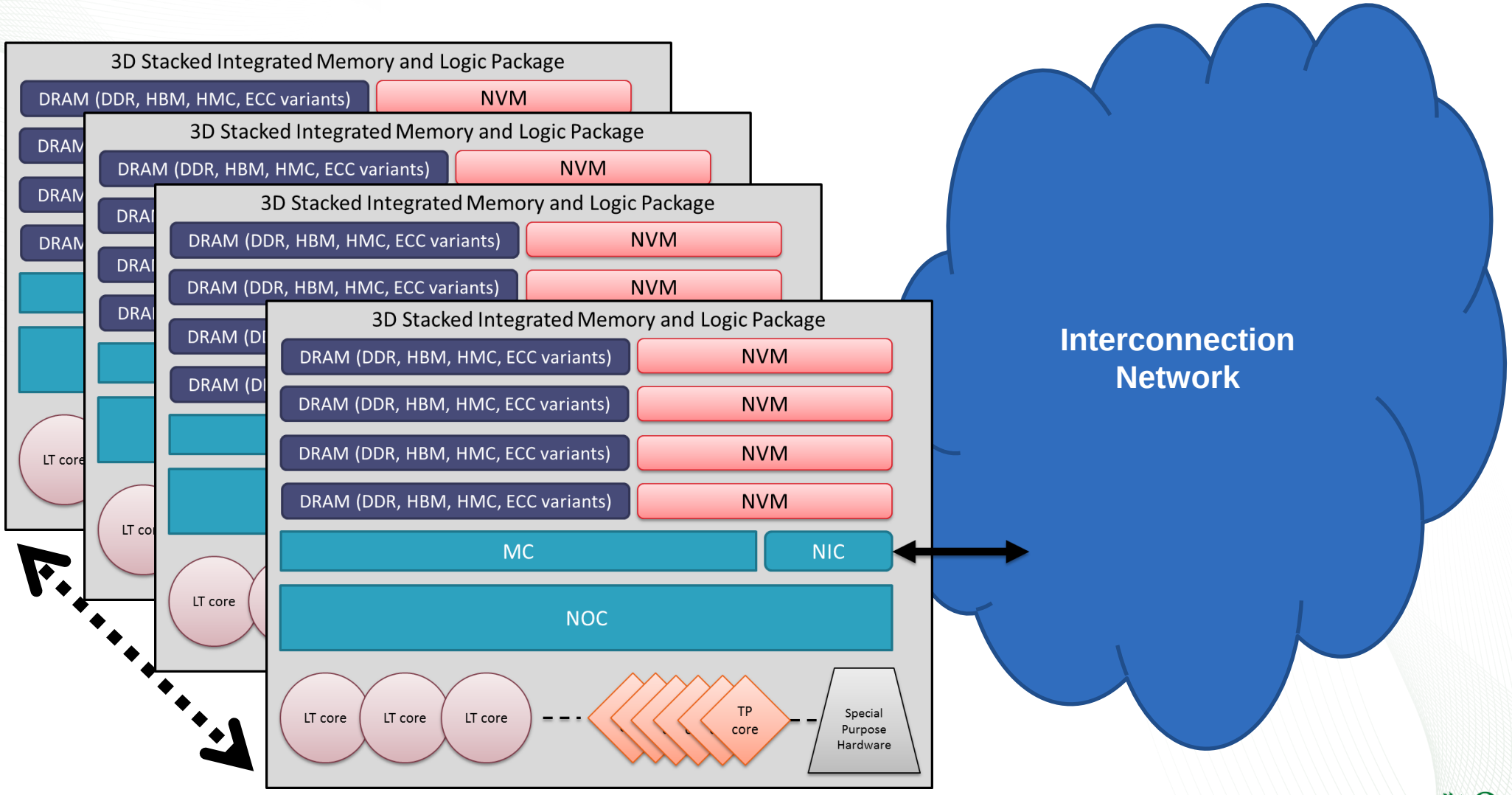
August 26, 2015 by Rich Brueckner 1 Comment

In this special guest feature from Scientific Computing World, Tom Wilkie looks at the next generation of supercomputers coming to China.

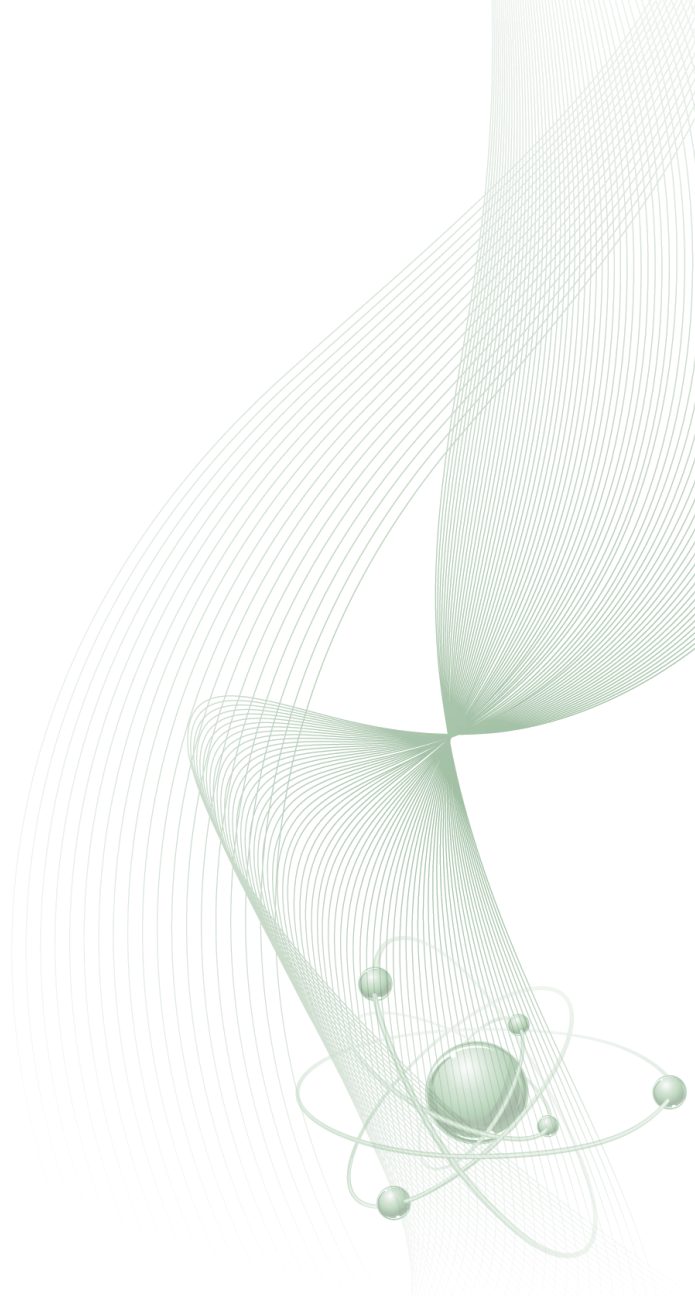
Within the next 12 months, China expects to be operating not one but two 100 Petaflop computers, each containing (different) Chinese-made processors, and both coming online about a year before the United States' 100 Petaflop machines being developed under the Coral initiative.

Ironically, the CPU for one machine appears very similar to a technology abandoned by the USA in 2007, and the US Government, through its export embargo, has encouraged China to develop its own accelerator for the other machine.

Notional Future Architecture



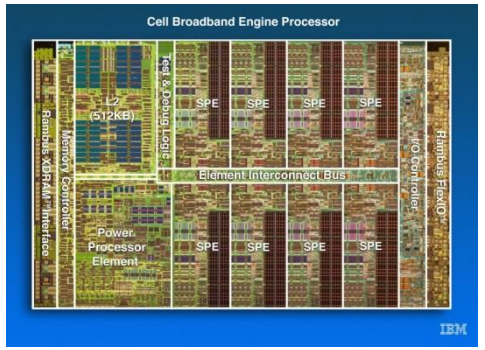
Heterogeneous computing is here to stay

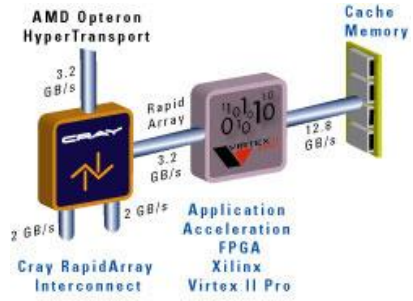


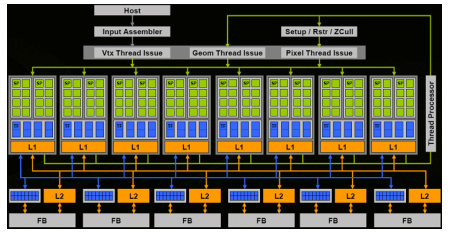
Earlier Experimental Computing Systems (past decade)

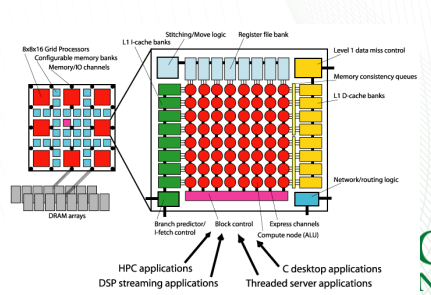
- The past decade has started the trend away from traditional 'simple' architectures
- Examples
  - Cell, GPUs, FPGAs, SoCs, etc
- Lessons learned?
- Mainly driven by facilities costs and successful (sometimes heroic) application examples

Popular architectures since ~2004





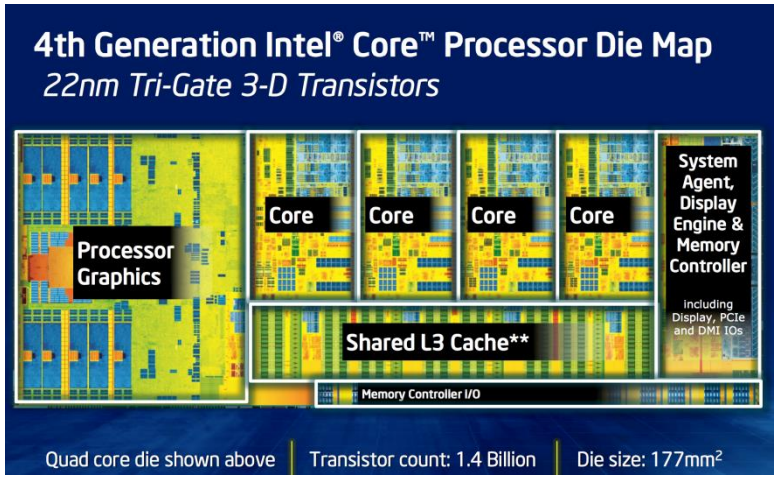


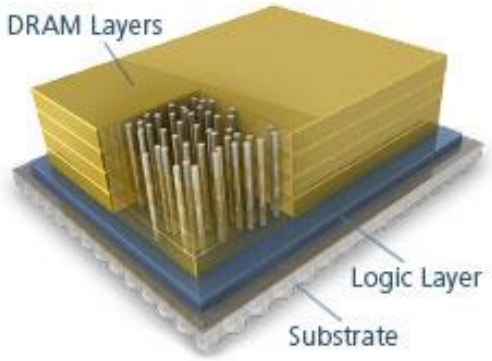


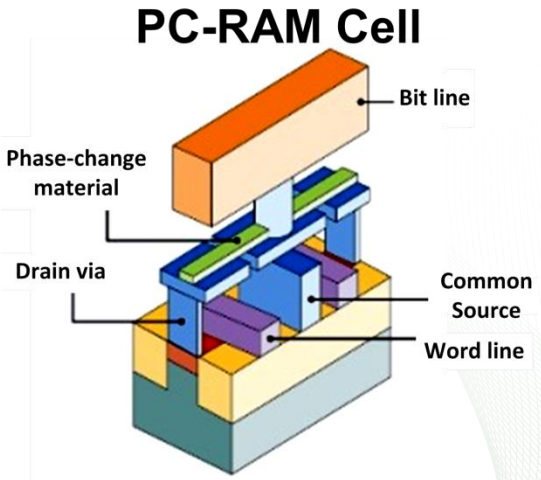
OAK RIDGE National Laboratory

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



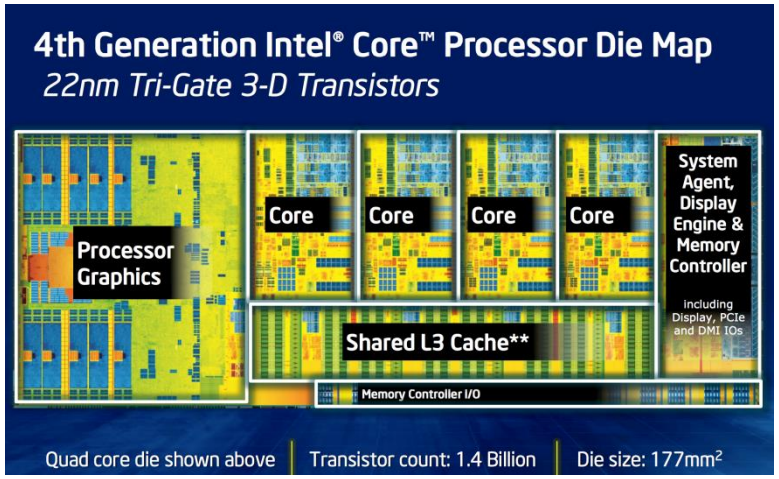


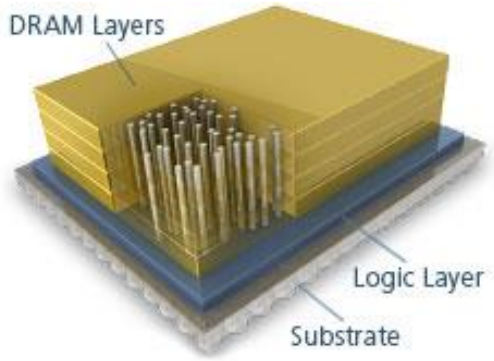


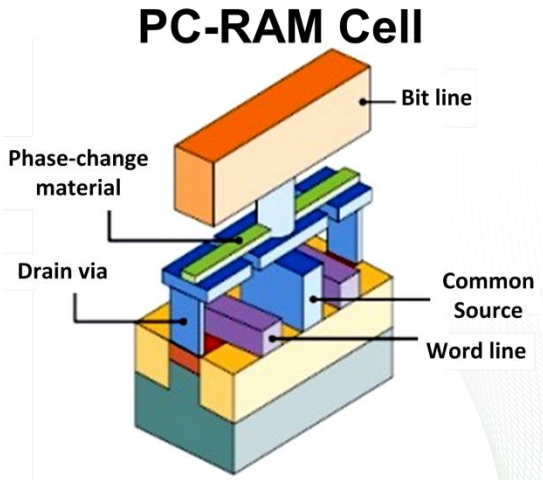
HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.

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

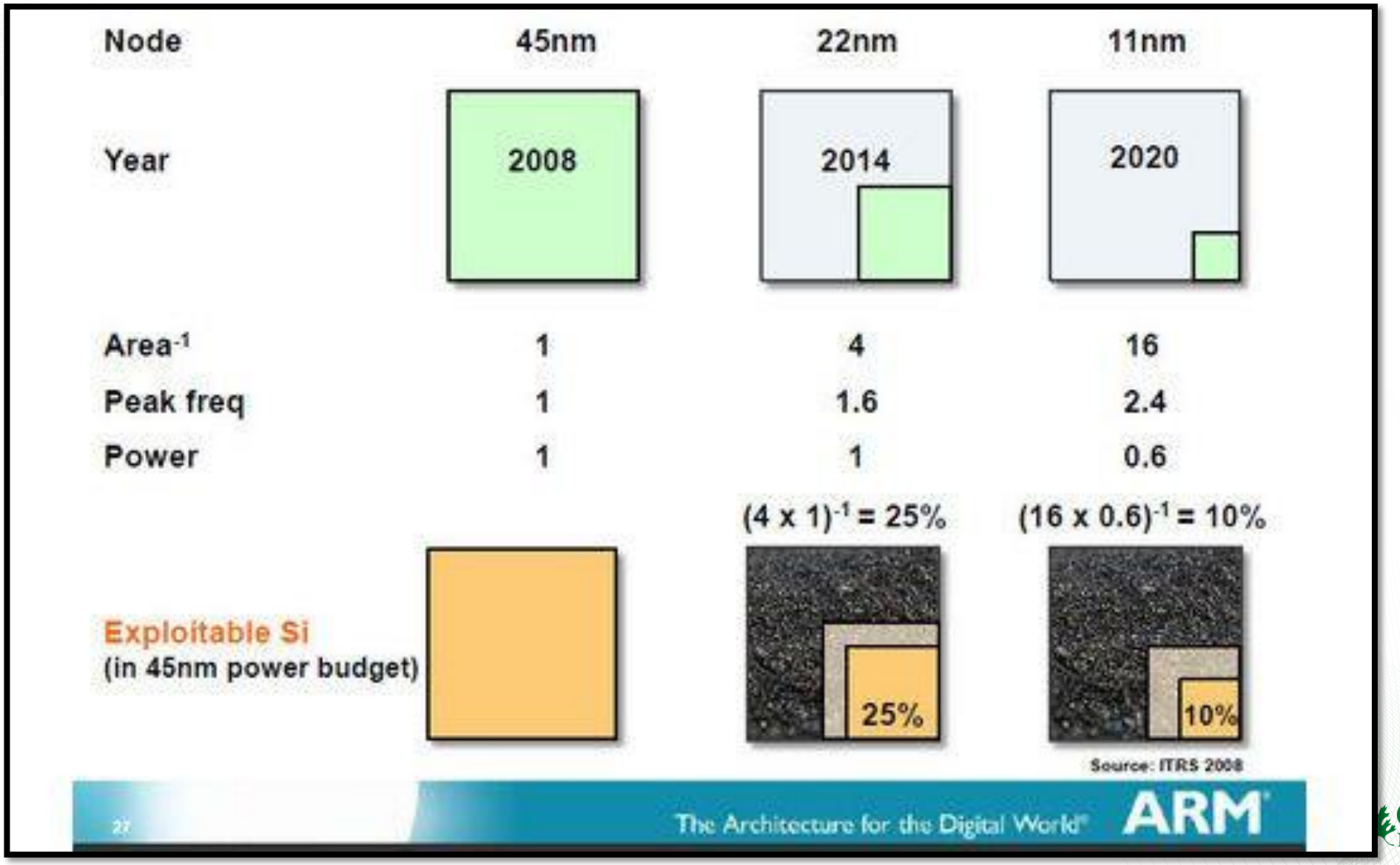






HPC (mobile, enterprise, embedded) computer design is more fluid now than in the past two decades.

Dark Silicon Facilitates Heterogeneity and Specialization



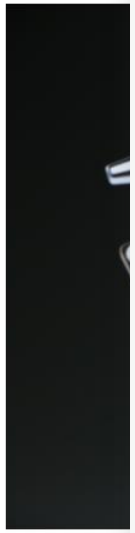
Source: ARM

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



Nvidia CEO Jen-H

Nvidia

Nvidia and IBM

units, letting GP

The fatter pipe

compared to 16

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

123 Comments

+ Add Comme

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

"SEATTLE" 64-BIT ARM SERVER PROCESSOR

FIRST 28NM AR

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

SAMPLI

12 | CHANGING INFRASTR

Around 15 mon

2014. Less thar

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/

Corporation ALTR, +0.00% and IBM IBM, +0

FPGA-based acceleration platform that cohe

CPU leveraging IBM's Coherent Accelerator

reconfigurable hardware accelerator feature

FPGA and processor which significantly impr

flexibility in high-performance computing (H

and IBM are presenting several POWER8 sys

using FPGAs at SuperComputing 2014.

Working together through the OpenPOWER

A-SERIES REDEFINES COMPUTE

Kaveri

4 "Steamroller" CPU Cores

8 GCN GPU Cores

Multi

AMD Tr

techn

UV

VC

Conne

PCIe

PCIe

Displ

DisplayPort 1.2

HDMI 1.4a

Avago Agrees to Buy Broadcom for $37 Billion

By MICHAEL J. de la MERCED and CHAD BRAY MAY 28, 2015

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

l has started open-sourcing the Linux driver for Broadwell's GPU. Broadwell is the 14nm die shrink of Intel's

nd while the CPU

xpected to

dwell's GPU looks

es (Iris) GPU

a modified

GPU, and the

18

Intel Mates FPGA With Future Xeon Server Chip

June 18, 2014 by Timothy Prickett Morgan

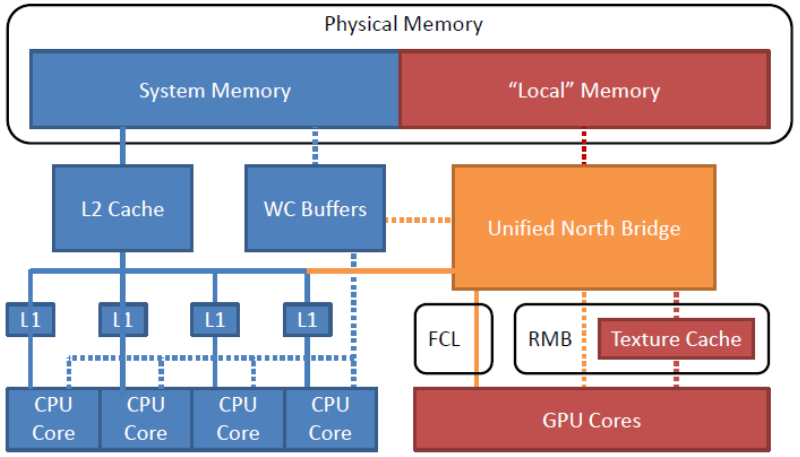
Intel is taking field programmable gate arrays seriously as a means of accelerating applications and has crafted a hybrid chip that marries an FPGA to a Xeon E5 processor and puts them in the same processor socket.

Intel to acquire Altera for $54 a share

Monday, 1 Jun 2015 | 8:33 AM ET

REUTERS

Tighter Integration will expand workload possibilities



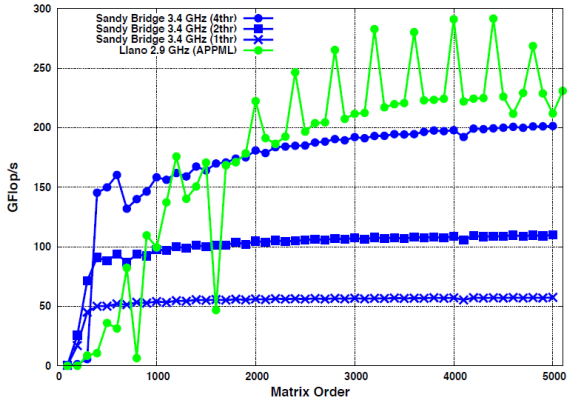
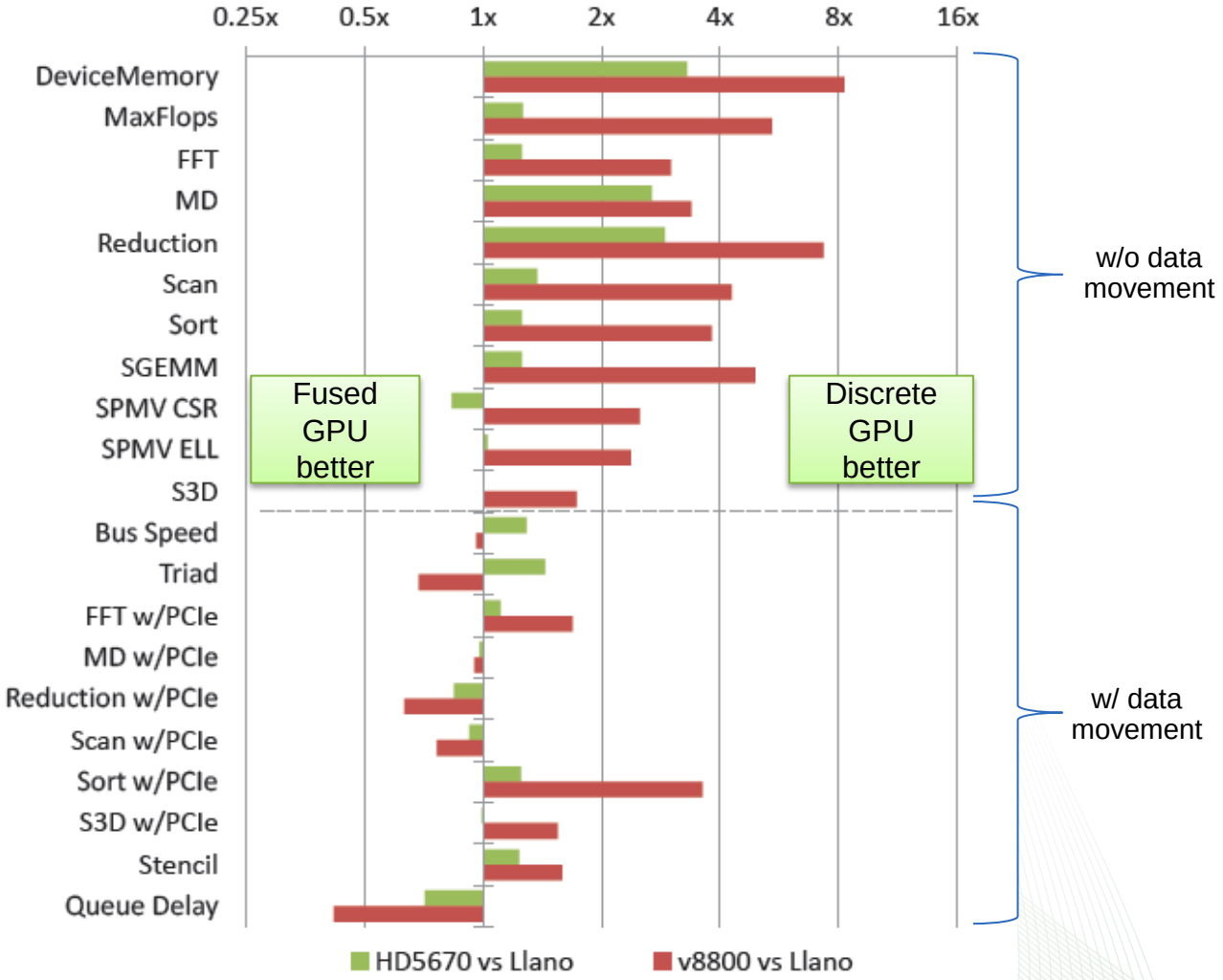


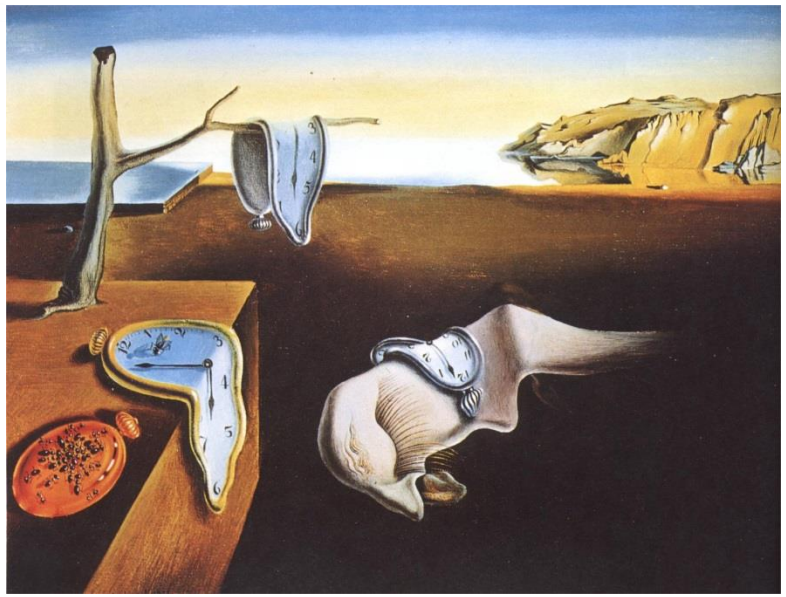
Figure 3: SGEMM Performance (one, two, and four CPU threads for Sandy Bridge and the OpenCL-based AMD APPML for Llano's fGPU)



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

New and Improved Memory Systems are the Next Big Thing

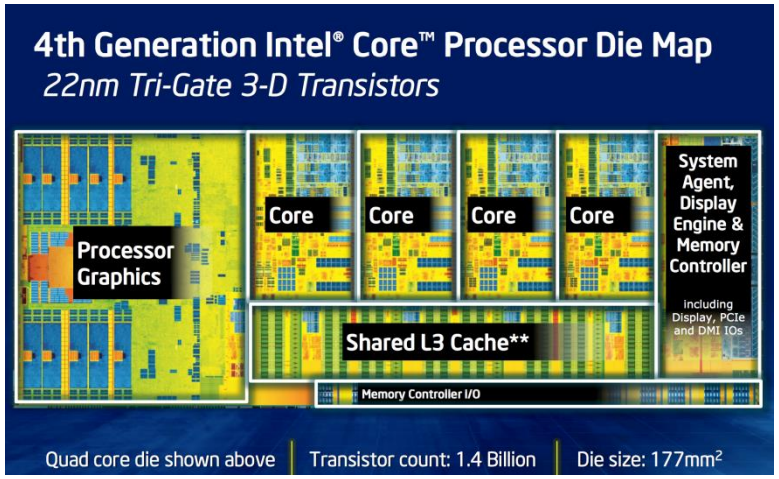


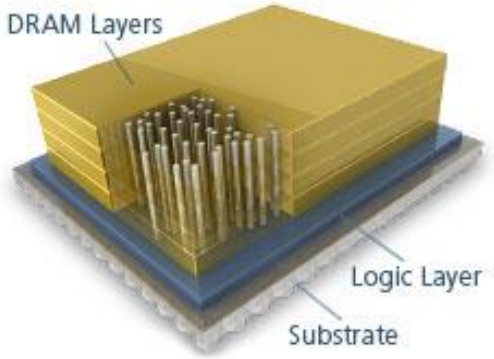
http://www.wikipaintings.org/en/salvador-dali/the-persistence-of-memory-1931

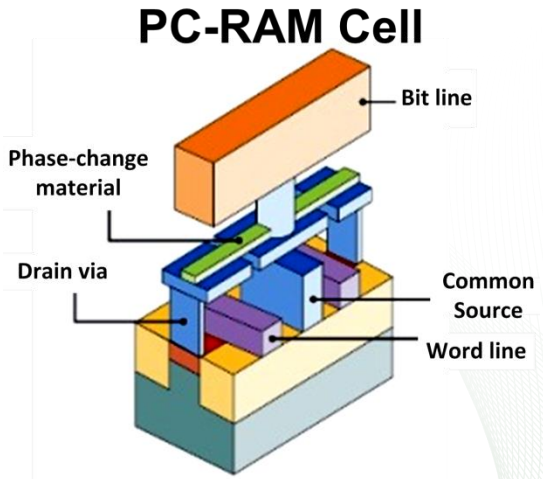
OAK RIDGE National Laboratory

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)
  - New interfaces
- 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.

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

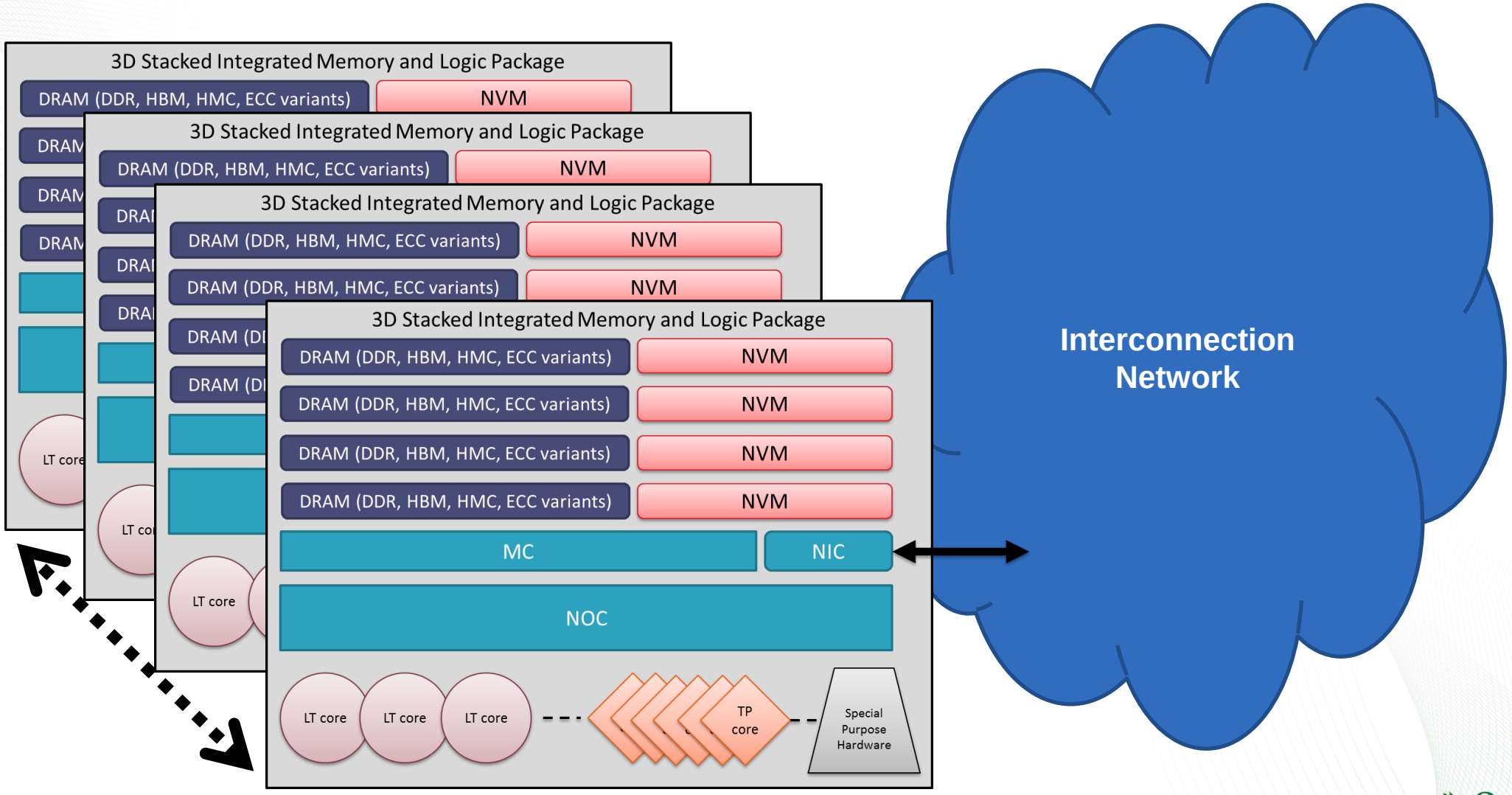
System attributes	2009	"Pre-Exascale"		"Exascale"	
System peak	2 PF	100-200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Storage	15 PB	150 PB		500 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	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)	

Exascale architecture targets
defined at 2009 Exascale Challenges Workshop in San Diego

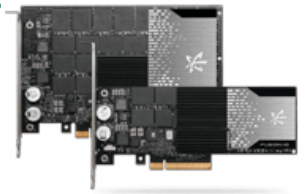
Where we are going "off the tracks" is
data capacity, and movement between nodes and from node to storage
Summit: Interconnect BW= 25 GB/s, I/O BW= 1 TB/s

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)	

Notional Future Architecture



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

Like 17 Tweet 7

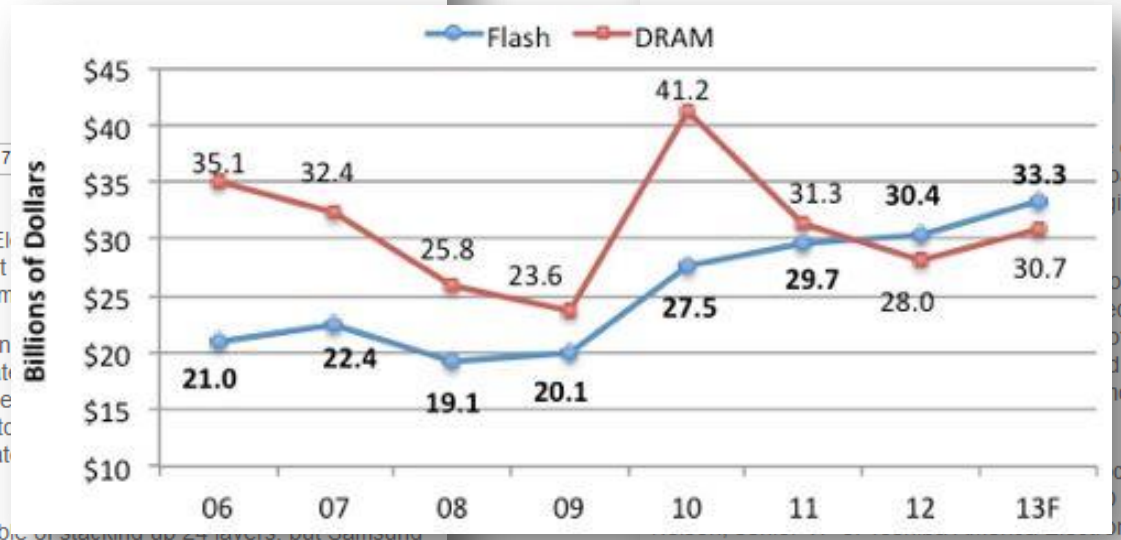
LONDON — Samsung El
production of a 128 Gbit
multiple layers, and claim

The memory is based on
conventional floating gat
In the vertical arrangeme
reliability between a facto
conventional floating-gat
in a press release.

The technology is capabl
did not disclose how many layers it had used in its 128 Gbit vertical NAND, nor whether the memory cells are multilevel cell or whether it had relaxed the design geometry from the leading edge in 2D memory, which stands at about 19 or 16 nm.

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

NO RATINGS
LOGIN TO RATE

+1 1

economic ser
partner Toshib
ies to produce

both 2-bit-pe
ctures with
f 2014. The 1
will be used a
novable cards to

cess techi
NAND, sa
nic Comp
memory business unit. Toshiba's 15nm process works i
conjunction with improved peripheral circuitry technolo
chips that achieve the same write speed as chips forme
second generation 19nm process technology, but boos
transfer rate to 533 megabits a second -- 1.3 times fas
employing a high-speed interface.

Nelson said there is room to advance floating gates before moving

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 Mellor

h after the tech titan's CTO
mristor drives come 2018.

limiting the maximum capacity
the memories' reliability drops
dies.

M technology that is supposed
imed at an HP Discover event
D hits its limit in five years. He
acity by the end of the decade.

Forbes / Tech

JUL 28, 2015 @ 2:46 PM 7,391 VIEWS

Intel And Micron Jointly Announce Game-Changing 3D XPoint Memory Technology

China's Tsinghua Unigroup plans $23B bid for Micron Technology

CNBC.com staff | @CNBC

Monday, 13 Jul 2015 | 8:41 PM ET

CNBC

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

Comparison of emerging memory technologies

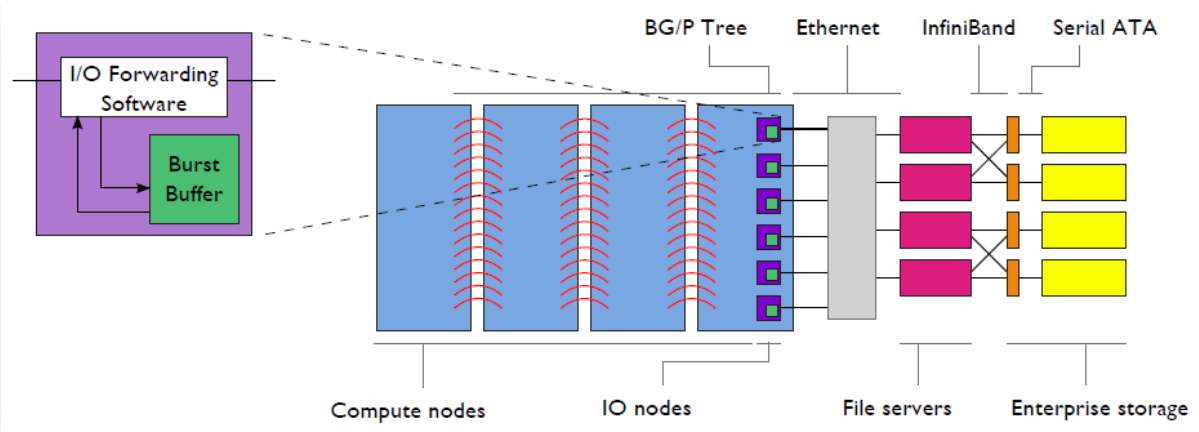
	SRAM	DRAM	eDRAM	2D NAND Flash	3D NAND Flash	PCRAM	STTRAM	2D ReRAM	3D ReRAM
Data Retention	N	N	N	Y	Y	Y	Y	Y	Y
Cell Size (F^2)	50-200	4-6	19-26	2-5	<1	4-10	8-40	4	<1
Minimum F demonstrated (nm)	14	25	22	16	64	20	28	27	24
Read Time (ns)	< 1	30	5	10^4	10^4	10-50	3-10	10-50	10-50
Write Time (ns)	< 1	50	5	10^5	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									

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

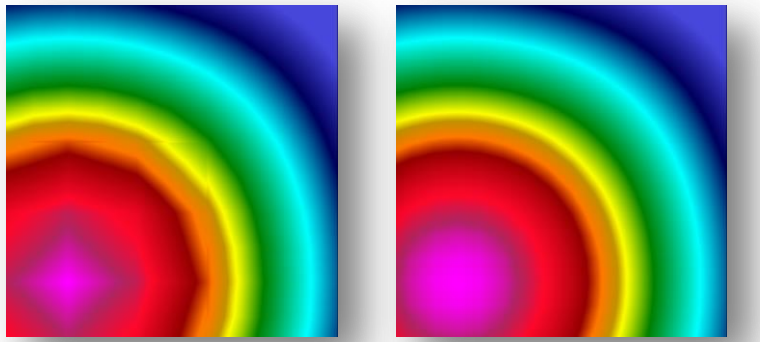
Opportunities for NVM in Emerging Systems

- Burst Buffers

[Liu, et al., MSST 2012]



- In situ visualization



http://ft.ornl.gov/eavl

- In-mem tables

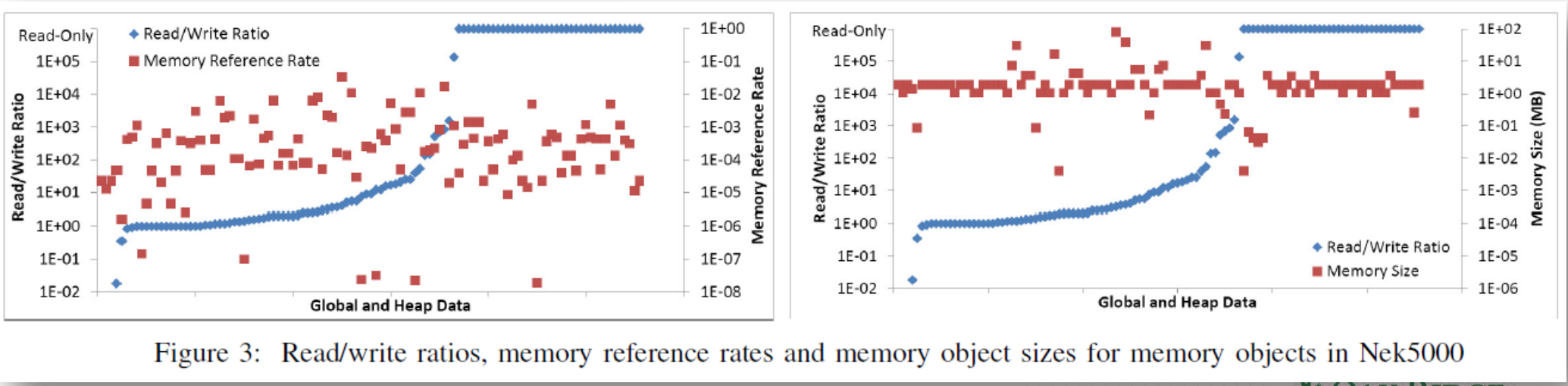
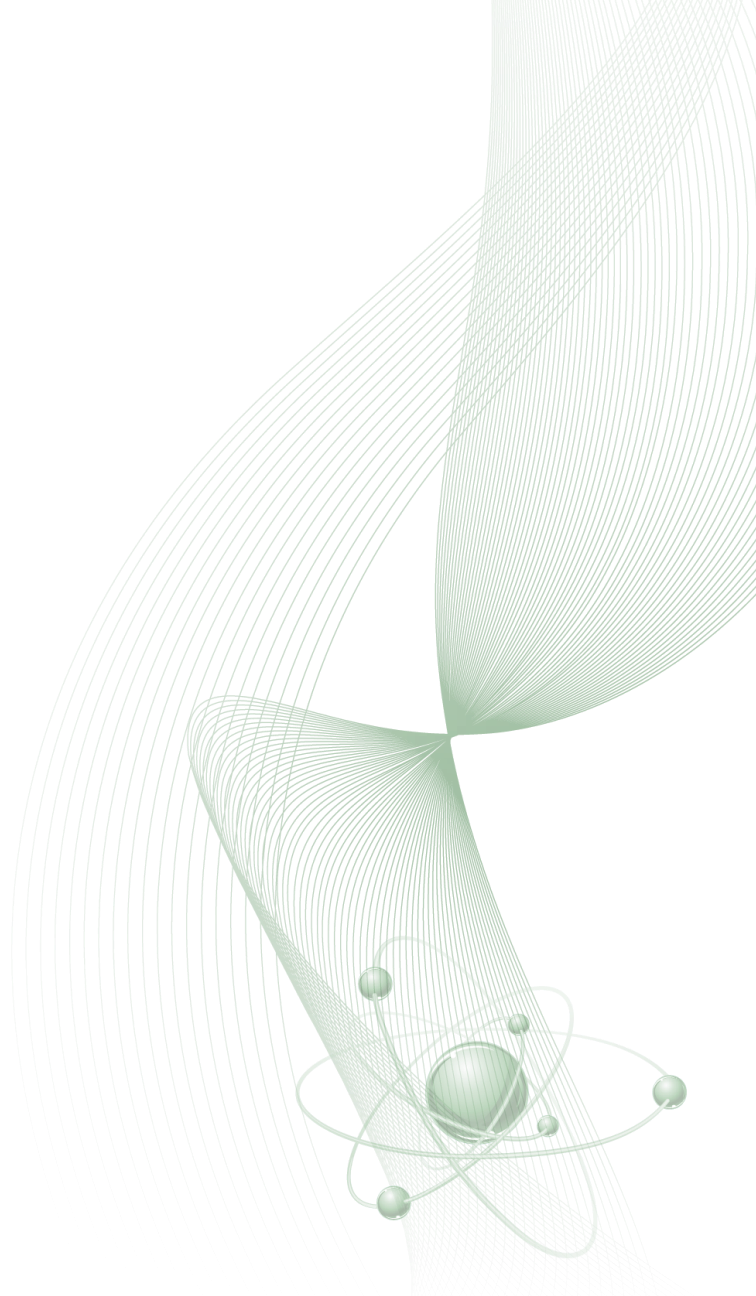


Figure 3: Read/write ratios, memory reference rates and memory object sizes for memory objects in Nek5000

J.S. Vetter and S. Mittal, "Opportunities for Nonvolatile Memory Systems in Extreme-Scale High-Performance Computing," *Computing in Science & Engineering, 17(2):73-82, 2015, 10.1109/MCSE.2015.4.*

Architectural Uncertainty Demands Methods for Performance Prediction and Portability

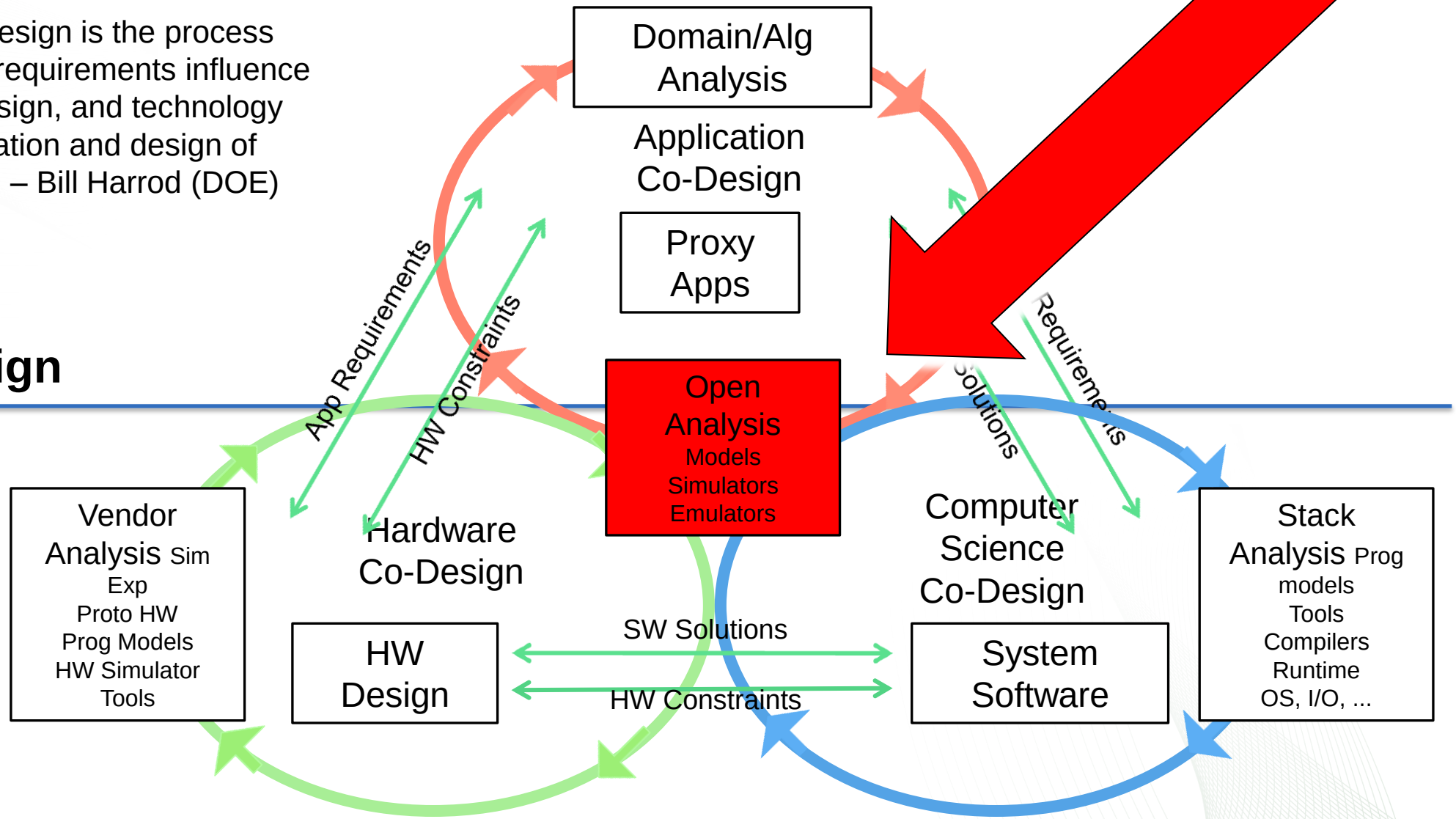
- Performance Prediction – Aspen
- Performance Portability – OpenARC



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

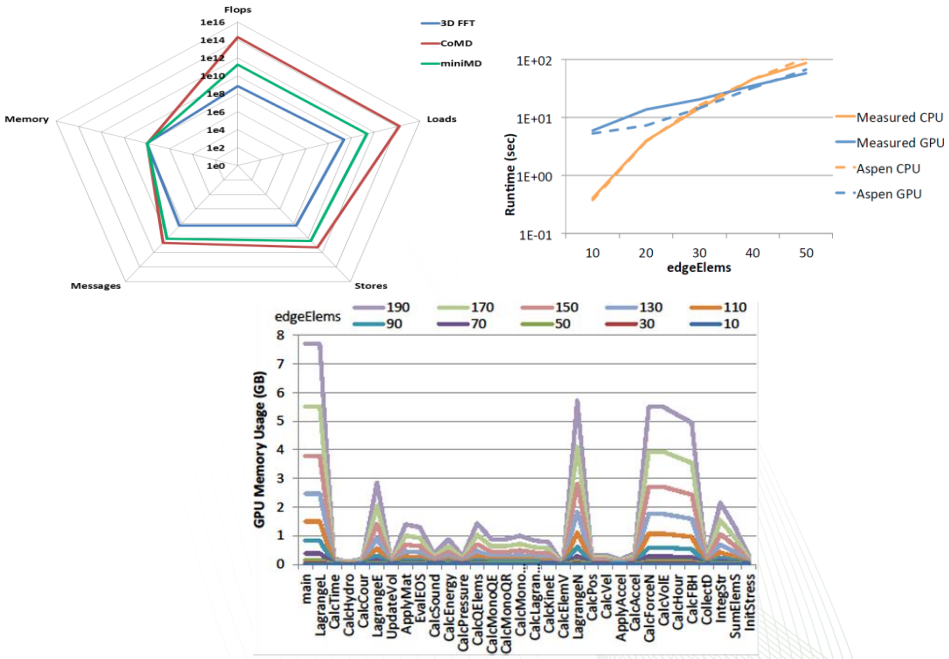
- Interactive tools for graphs, queries
- Design space exploration
- Workload Generation
- Feedback to Runtime Systems

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

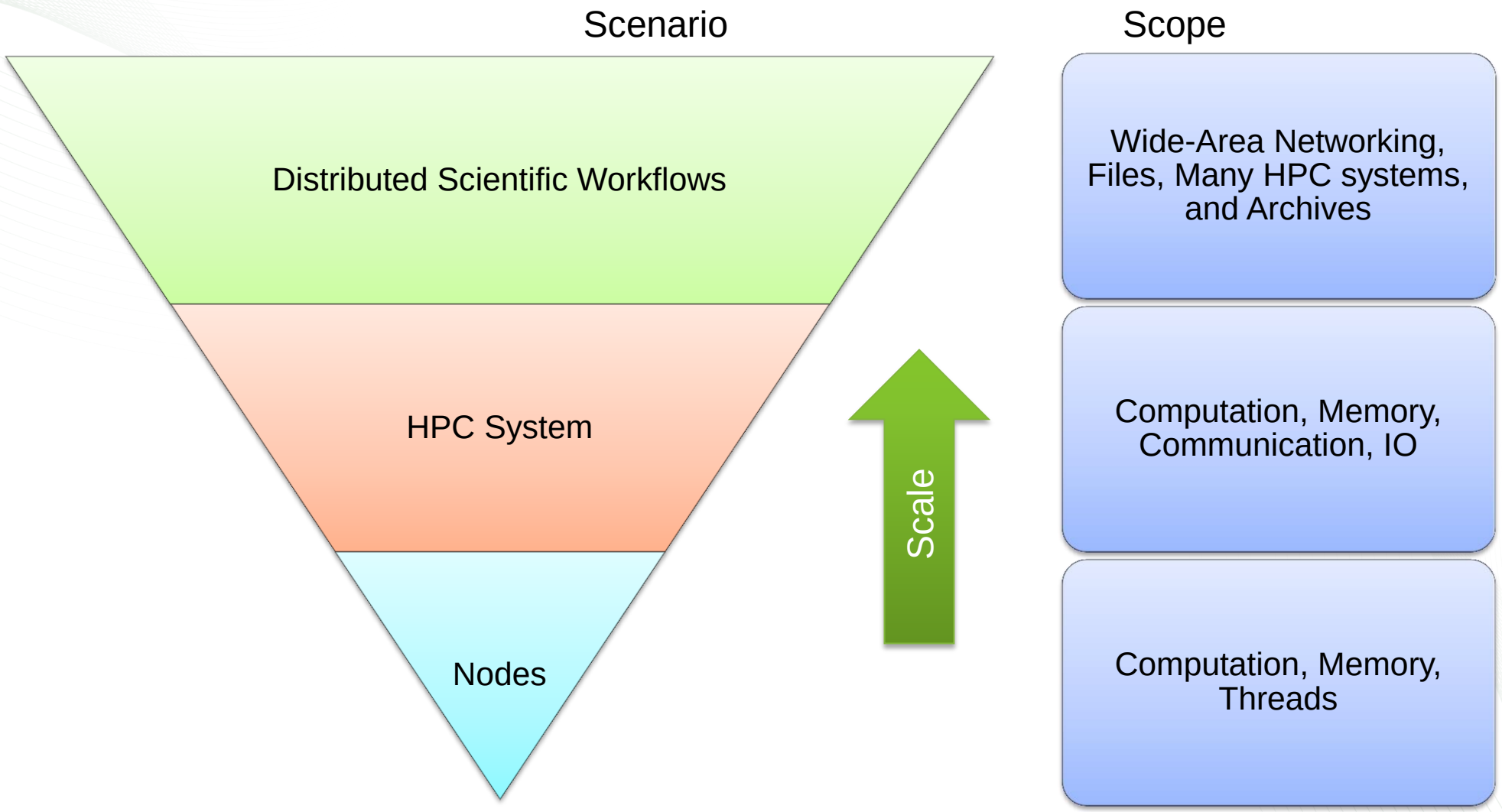
1 contributor

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)
```

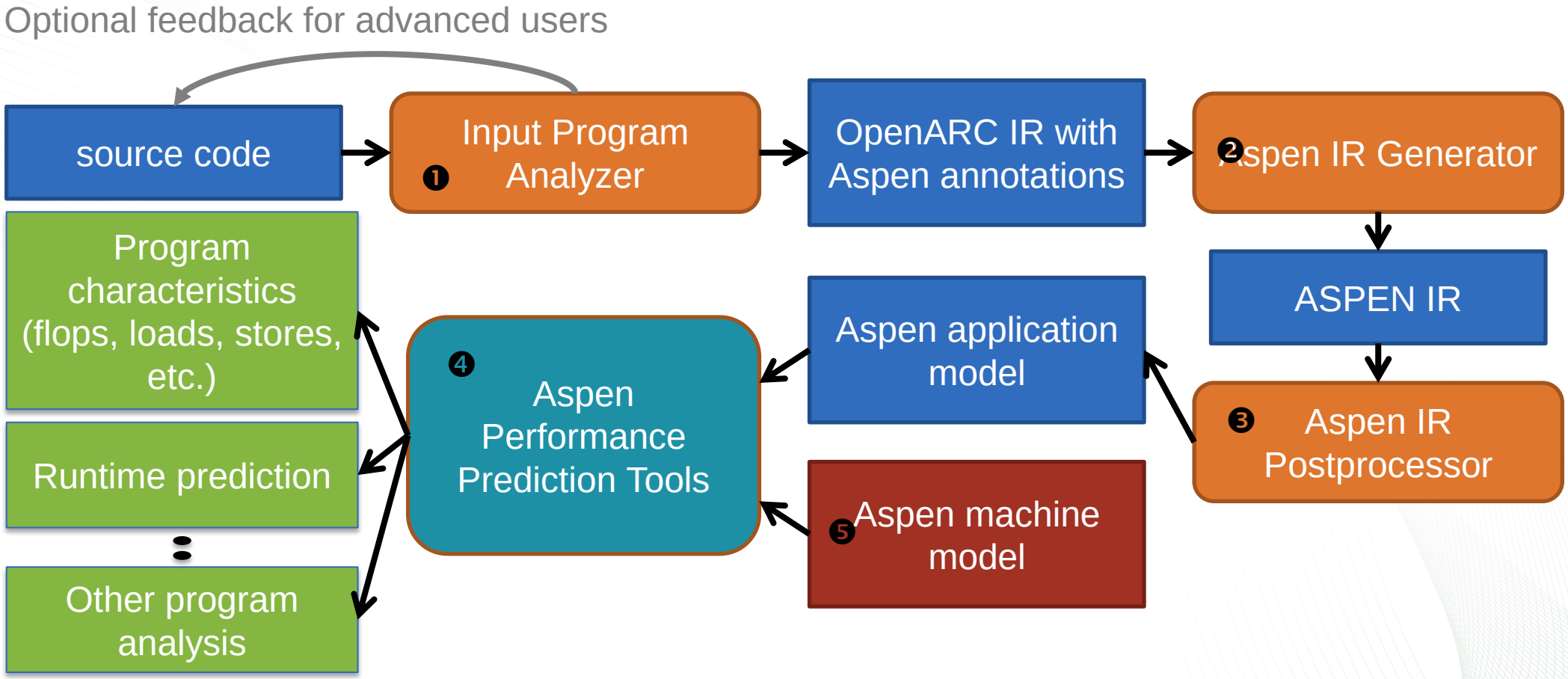
```
kernel CalcMonotonicQGradients {
  execute [numElems]
  {
    loads [8 * indexWordSize] from nodelist
    // Load and cache position and velocity.
    loads/caching [8 * wordSize] from x
    loads/caching [8 * wordSize] from y
    loads/caching [8 * wordSize] from z

    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
  }
}
```

Aspen allows Multiresolution Modeling



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.*

MM example generated from COMPASS

```
1  int N = 1024;
2  void matmul(float *a, float *b, float *c){ int i, j, k ;
3  #pragma acc kernels loop gang copyout(a[0:(N*N)]) \
4  copyin(b[0:(N*N)],c[0:(N*N)])
5    for (i=0; i<N; i++){
6  #pragma acc loop worker
7      for (j=0; j<N; j++) { float sum = 0.0 ;
8        for (k=0; k<N; k++) {sum+=b[i*N+k]*c[k*N+j];}
9        a[i*N+j] = sum; }
10   } //end of i loop
11 } //end of matmul()
12 int main() {
13   int i; float *A = (float*) malloc(N*N*sizeof(float));
14   float *B = (float*) malloc(N*N*sizeof(float));
15   float *C = (float*) malloc(N*N*sizeof(float));
16   for (i = 0; i <  N*N; i++)
17   {  A[i] = 0.0F; B[i] = (float) i; C[i] = 1.0F; }
18 #pragma aspen modelregion label(MM)
19   matmul(A,B,C);
20   free(A); free(B); free(C); return 0;
21 } //end of main()
```

```
1  model MM {
2    param floatS = 4; param N = 1024
3    data A as Array((N*N), floatS)
4    data B as Array((N*N), floatS)
5    data C as Array((N*N), floatS)
6    kernel matmul {
7      execute matmul2_intracommIN
8      {  intracomm [floatS*(N*N)] to C as copyin
9        intracomm [floatS*(N*N)] to B as copyin }
10     map matmul2 [N] {
11       map matmul3 [N] {
12         iterate [N] {
13           execute matmul5
14           {  loads [floatS] from B as stride(1)
15             loads [floatS] from C; flops [2] as sp, simd }
16         } //end of iterate
17         execute matmul6 { stores [floatS] to A as stride(1) }
18       } // end of map matmul3
19     } //end of map matmul2
20     execute matmul2_intracommOUT
21     { intracomm [floatS*(N*N)] to A as copyout }
22   } //end of kernel matmul
23   kernel main { matmul() }
24 } //end of model MM
```

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)
    ......
```

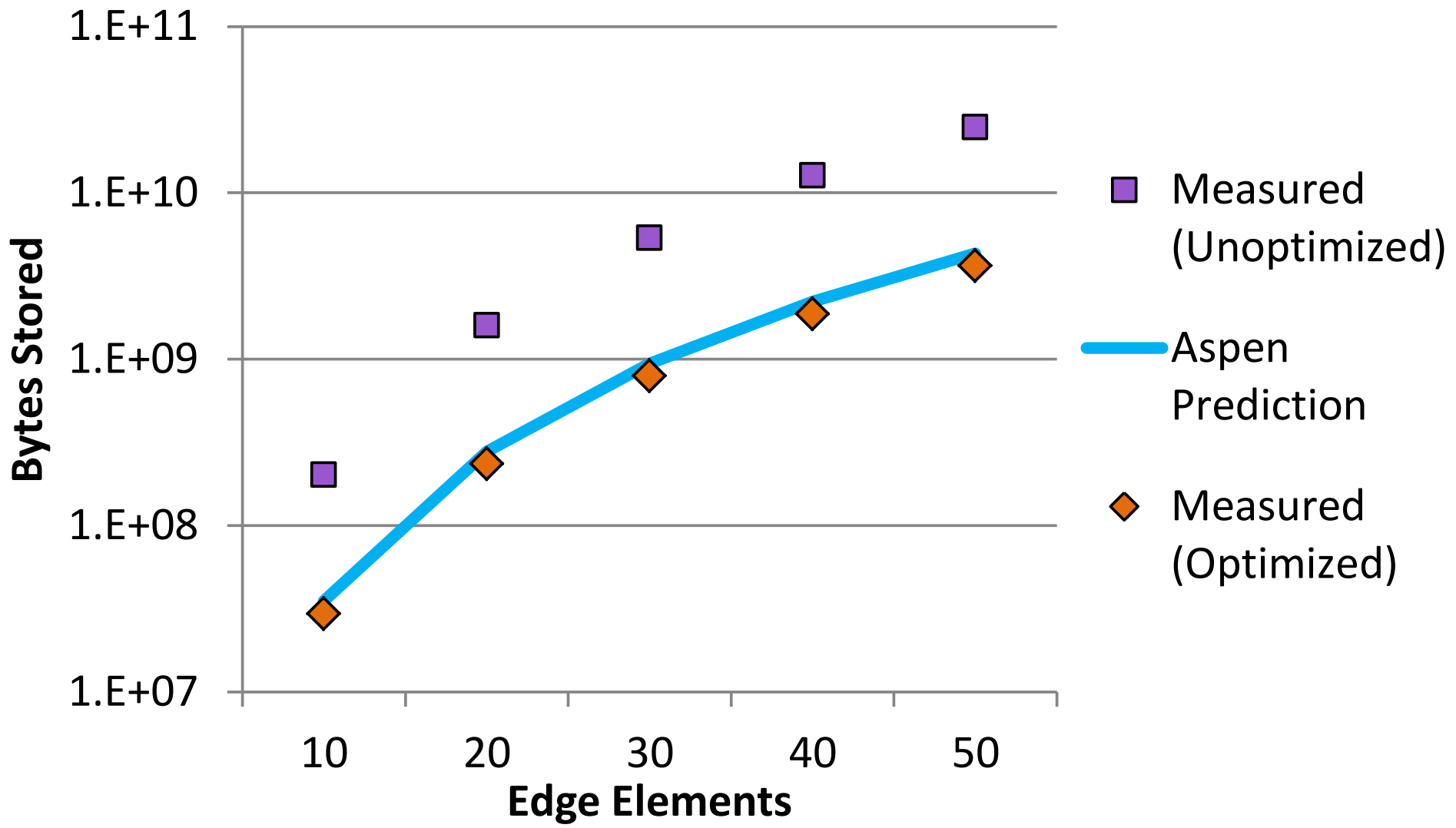
- Input LULESH program: 3700 lines of C codes
- Output Aspen model: 2300 lines of Aspen codes

Model 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.

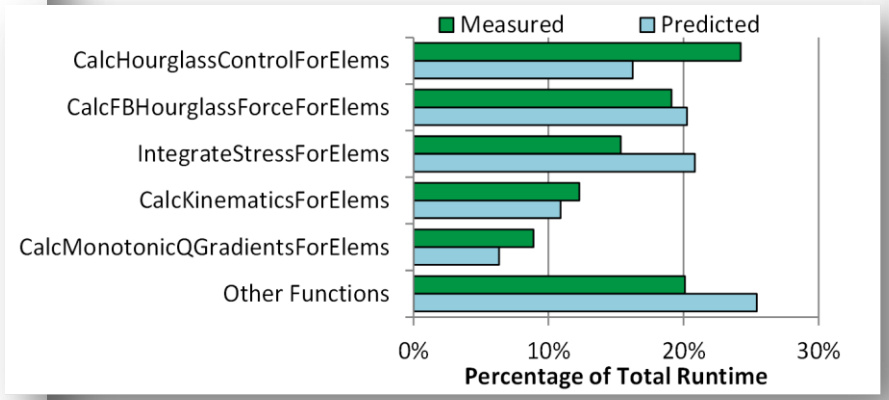
Model Scaling Validation (LULESH)



Example Queries

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.



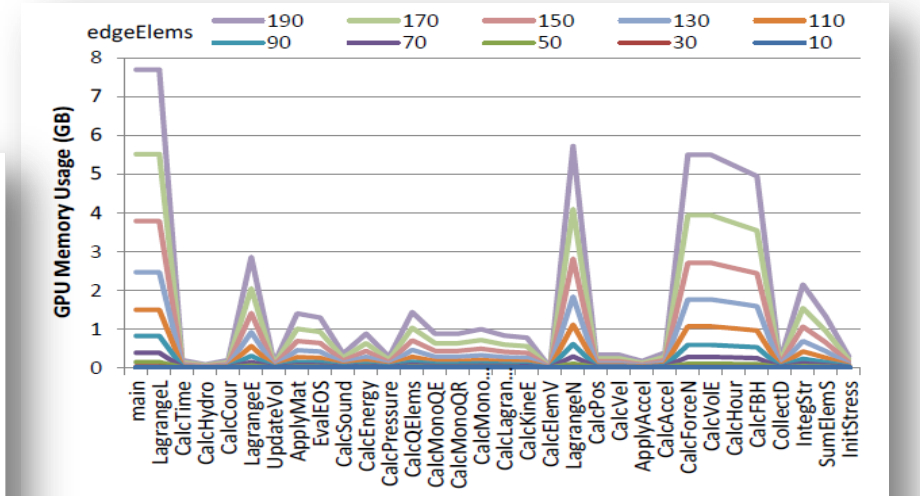


Fig. 8: GPU Memory Usage of each Function in LULESH, where the memory usage of a function is inclusive; value for a parent function includes data accessed by its child functions in the call graph.

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

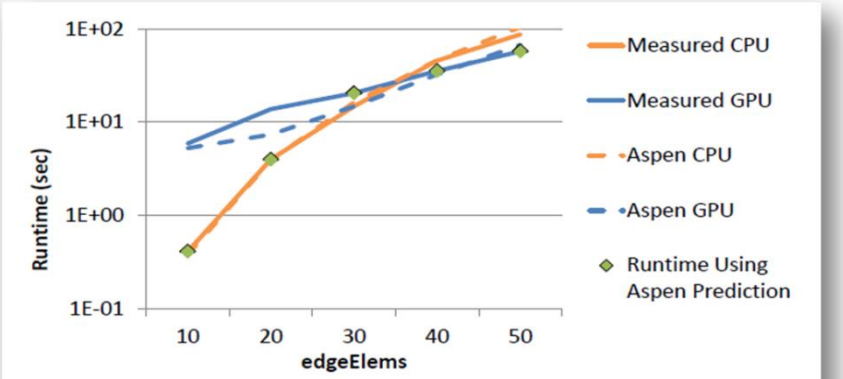


Fig. 7: Measured and predicted runtime of the entire LULESH program on CPU and GPU, including measured runtimes using the automatically predicted optimal target device at each size.

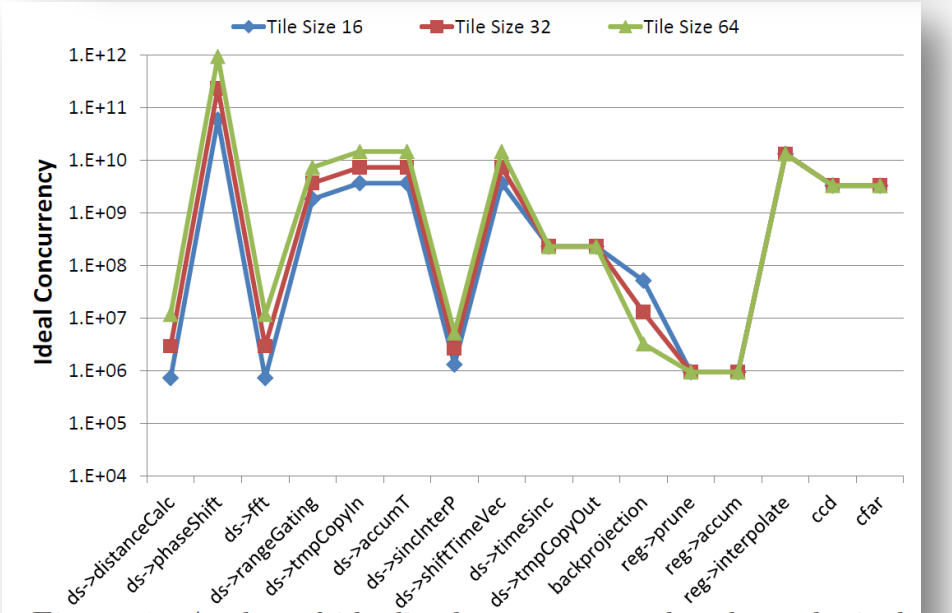
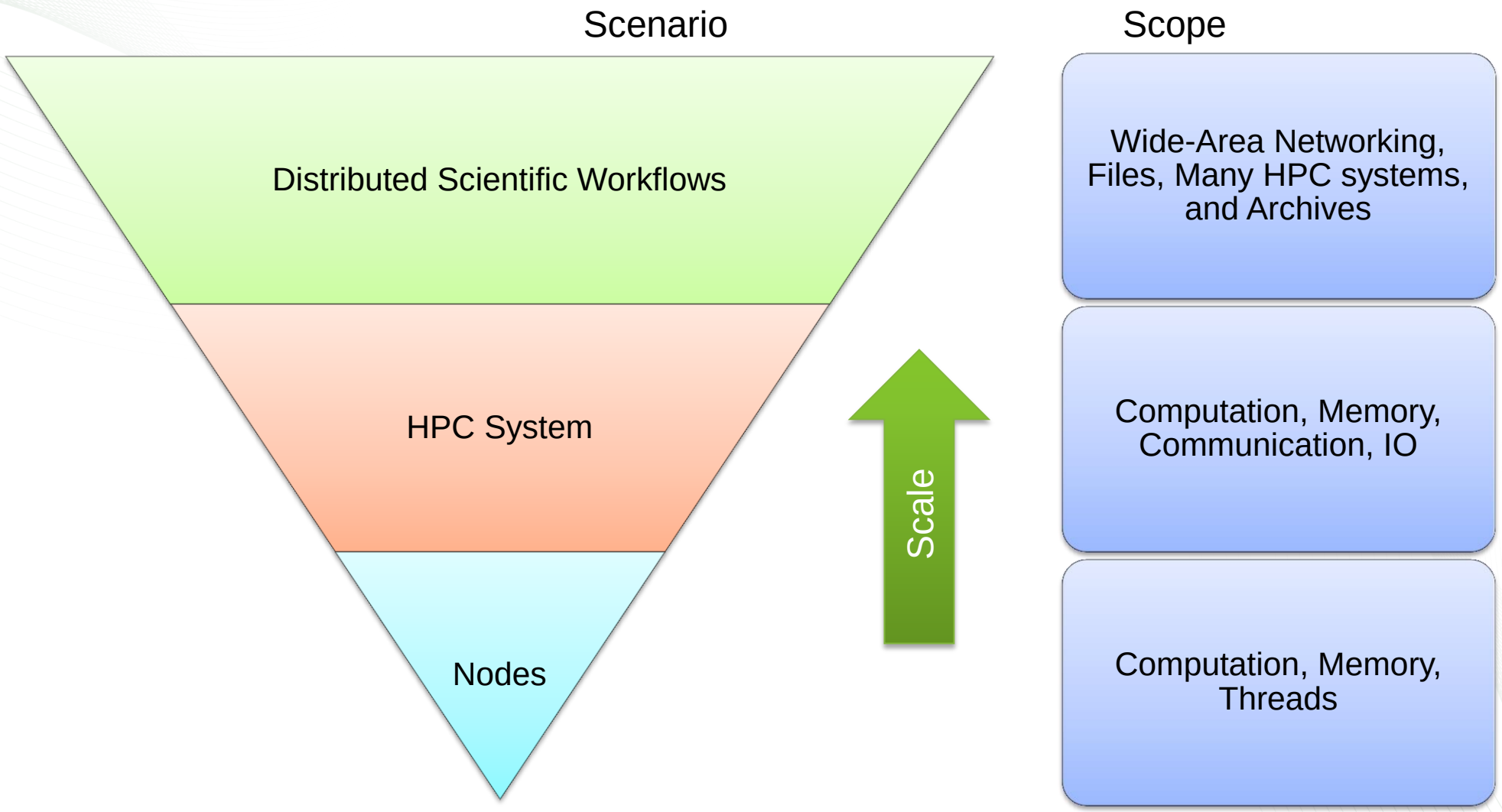
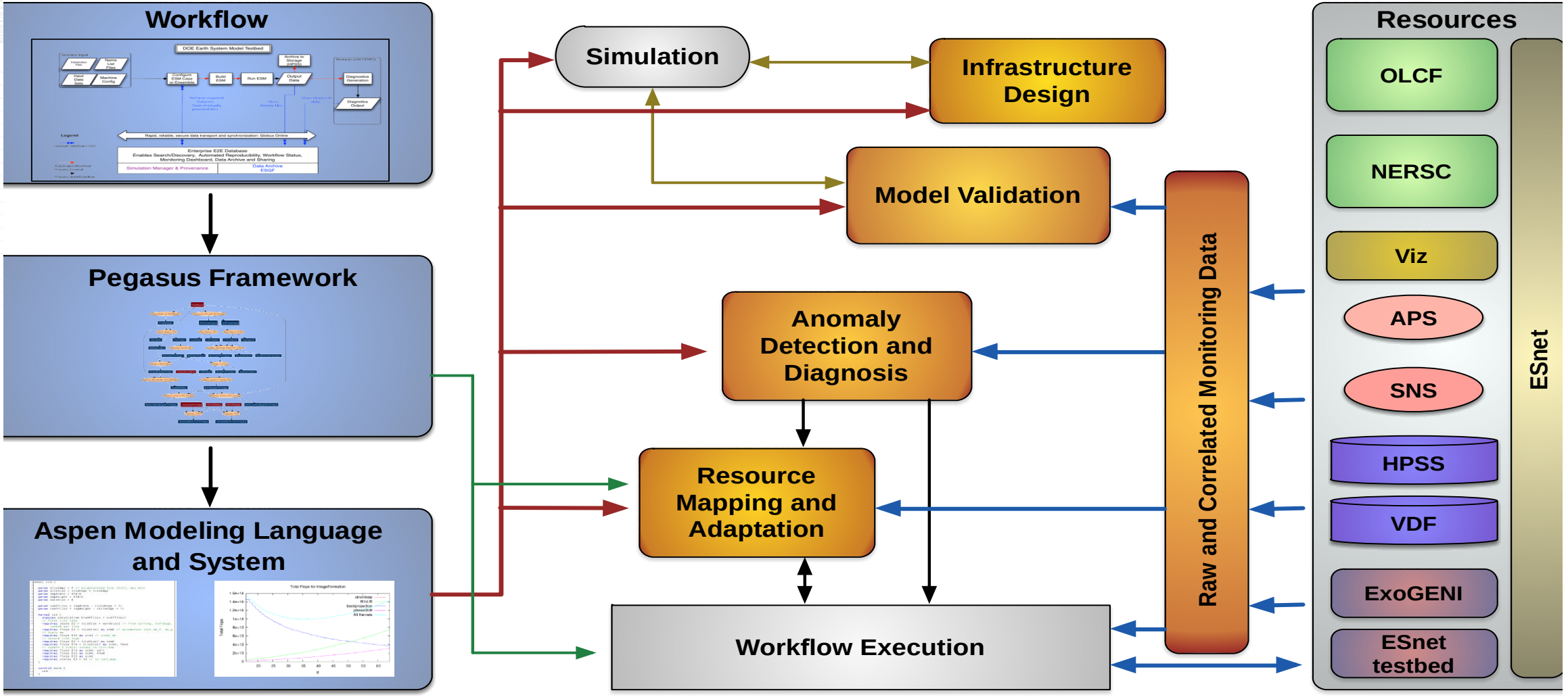


Figure 1: A plot of idealized concurrency by chronological phase in the digital spotlighting application model.

Aspen allows Multiresolution Modeling



PANORAMA Overview



E. Deelman, C. Carothers *et al., "PANORAMA: An Approach to Performance Modeling and Diagnosis of Extreme Scale Workflows," International Journal of High Performance Computing Applications, (to appear), 2015,*

Workflov ACME Climate Modelin

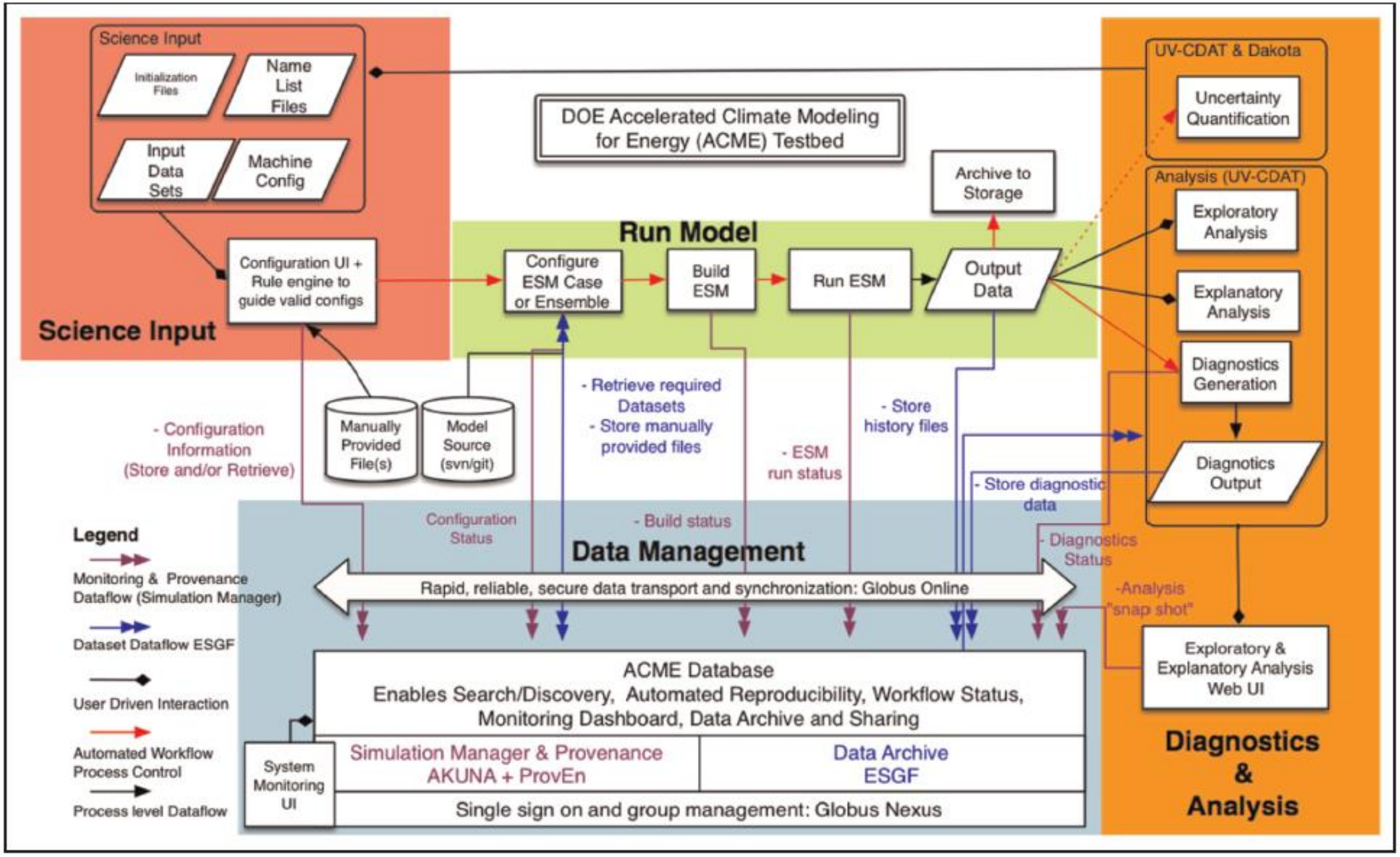
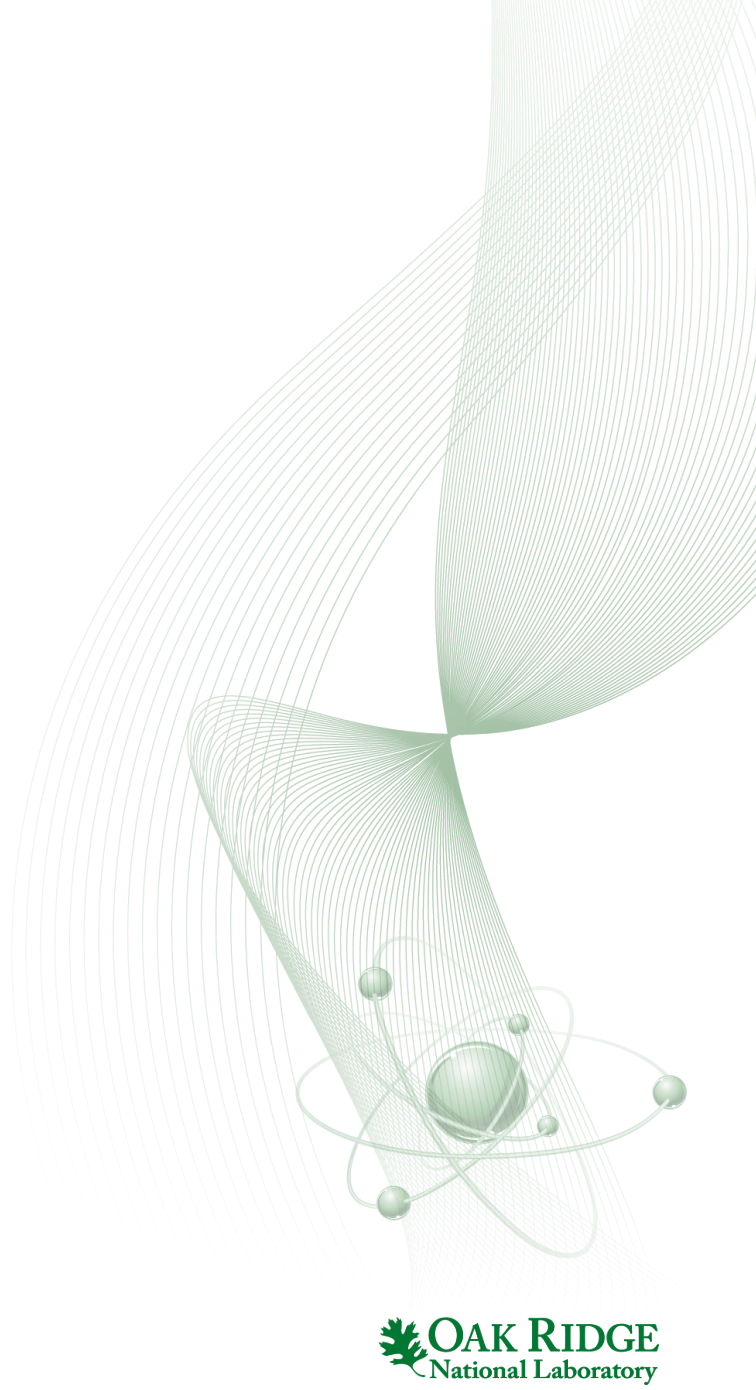
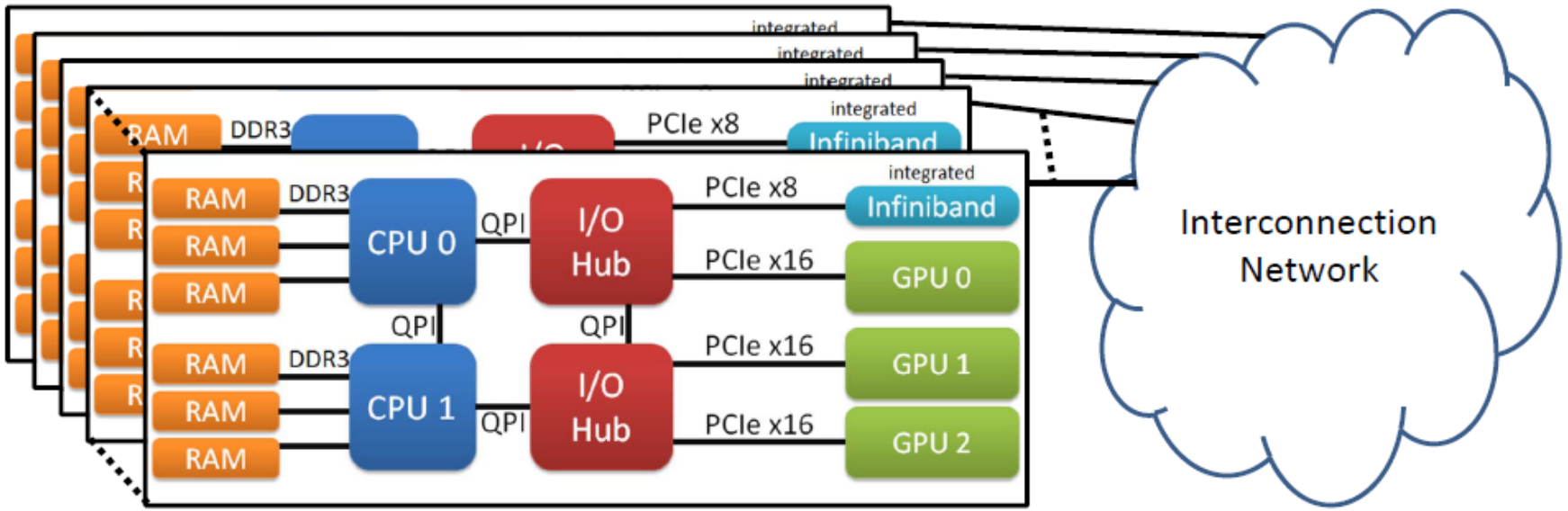


Figure 3: The complete Accelerated Climate Modeling for Energy (ACME) includes many interacting components distributed across DOE labs.

Enabling Performance Portability



Today's programming model



MPI

- Low overhead
- Resource contention
- Locality

OpenMP, Pthreads

- SIMD
- NUMA

OpenACC, CUDA, OpenCL, OpenMP4, …

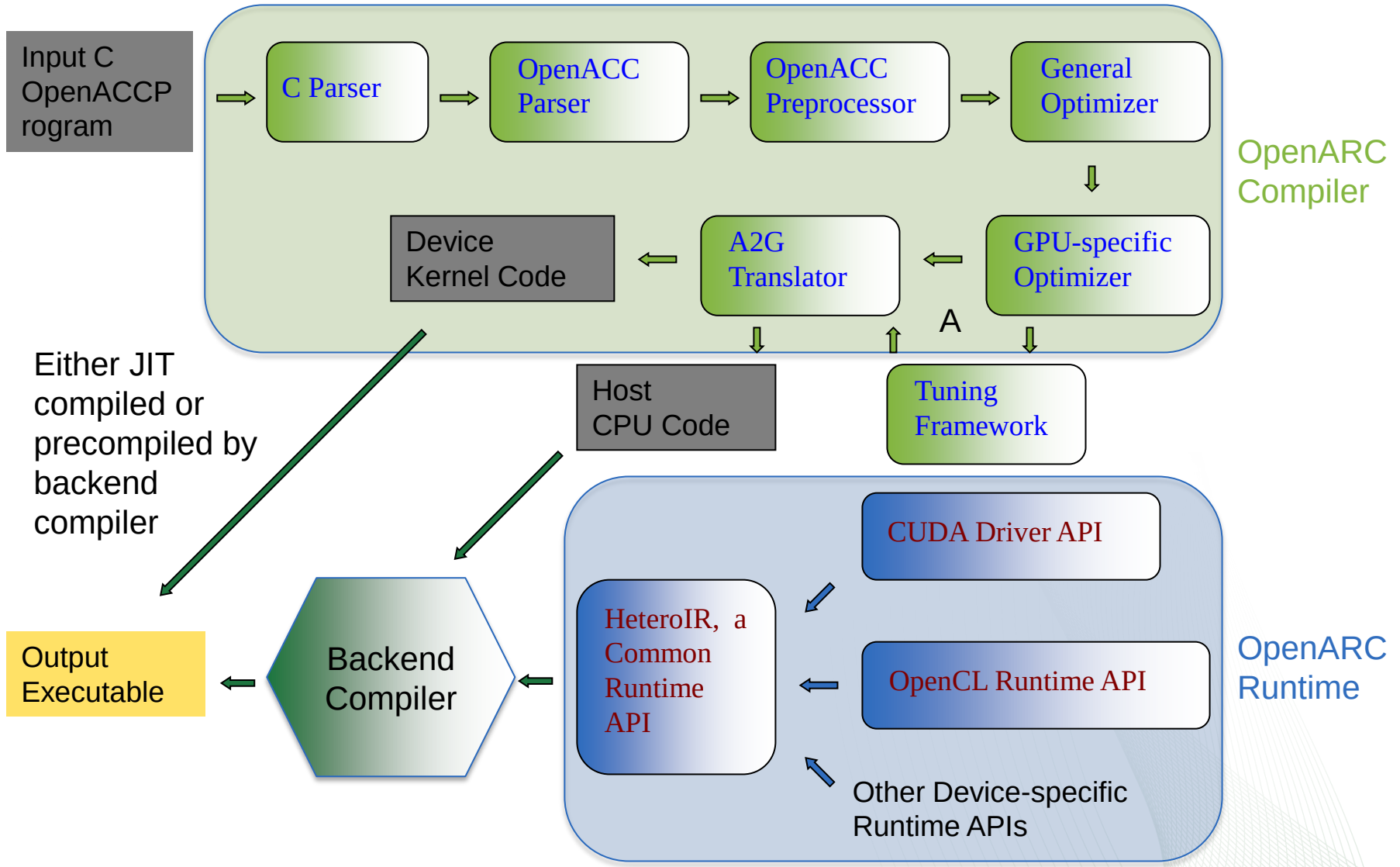
- Memory use, coalescing
- Data orchestration
- Fine grained parallelism
- Hardware features

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

OpenARC System Architecture

- OpenARC Compiler Framework
  - OpenACC
  - OpenMP4 Offload
- Provide common runtime APIs for various back-ends
  - CUDA
  - OpenCL
  - LLVM IR
- Open-Sourced
- High-Level Intermediate Representation (HIR)-Based
- Extensible



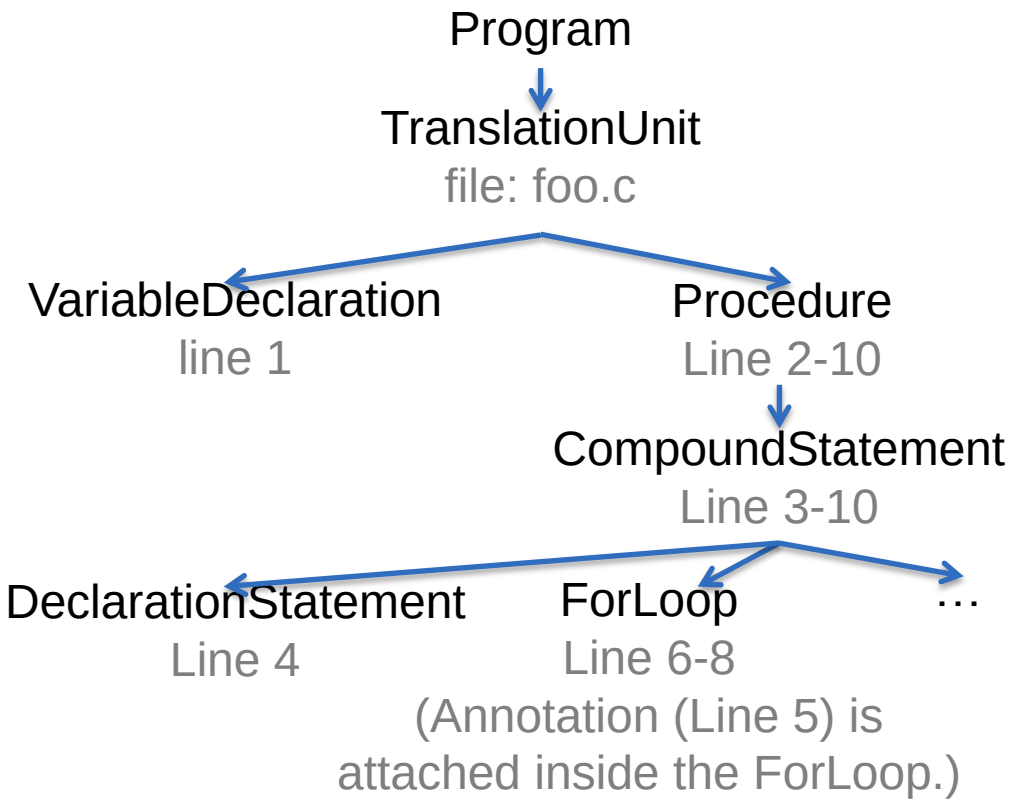
S. Lee and J.S. Vetter, "OpenARC: Open Accelerator Research Compiler for Directive-Based, Efficient Heterogeneous Computing," in *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC). Vancouver: ACM, 2014*

OpenARC High-Level Representation Example

- Input program

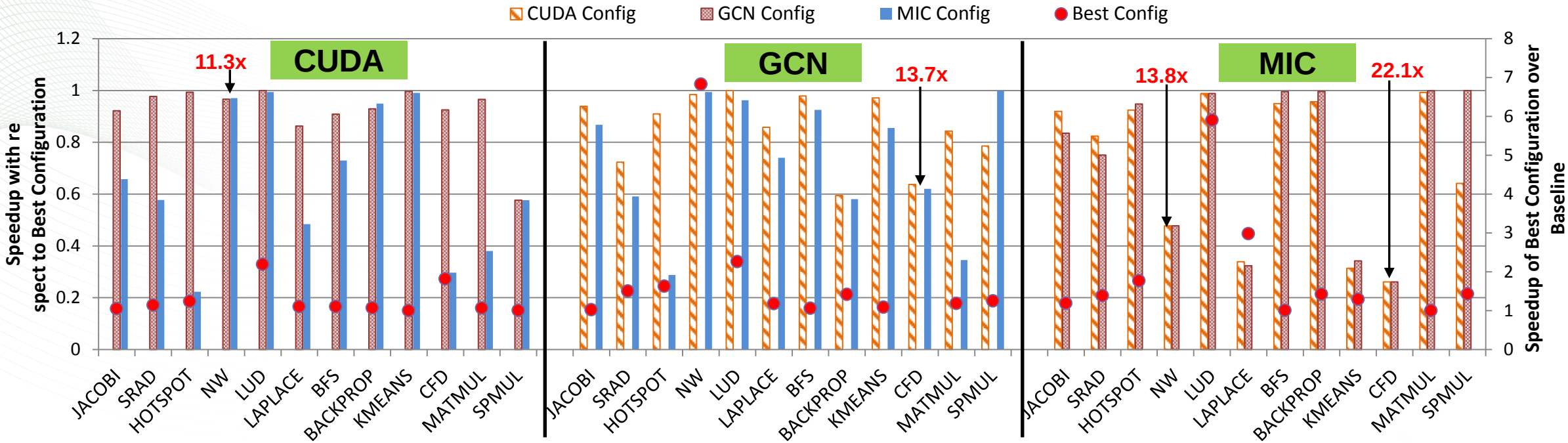
```
0  /* file: foo.c */
1    int k = 2;
2    int main(void)
3   {
4       int i, a[1024], b[1024];
5       #pragma acc kernels
     loop copyin(b) copyout(a)
6        for( i=0; i<1024; i++ ) {
7           a[i] = k*b[i];
8       }
9       …
10 }
```

- OpenARC IR



- Common Optimizations
  - Data transfer optimizations
  - Parallel loop swap
  - Tree-based reduction generation
  - Obtaining aligned memory access
  - Loop unrolling
- Architecture specific Optimizations
  - Texture memory loading → CUDA GPUs
  - Automatic shared memory loading → CUDA, GCN GPUs
  - Pitched memory allocation → CUDA GPUs

Overall Performance Portability



- Better perf. portability among GPUs
- Lesser across GPUs and MIC
- Main reasons
  - Parallelism arrangement
  - Compiler optimizations : e.g. device-specific memories, unrolling etc.

Performance Portability Matrix

Best Program version of	Executed on: CUDA	GCN	MIC
CUDA	100	84	65
GCN	91	100	67
MIC	58	68	100

A. Sabne, P. Sakhnagool *et al., "Evaluating Performance Portability of OpenACC," in 27th International Workshop on Languages and Compiler for Parallel Computing (LCPC)*
Portland, Oregon, 2014

Automating selection of optimizations based on machine model

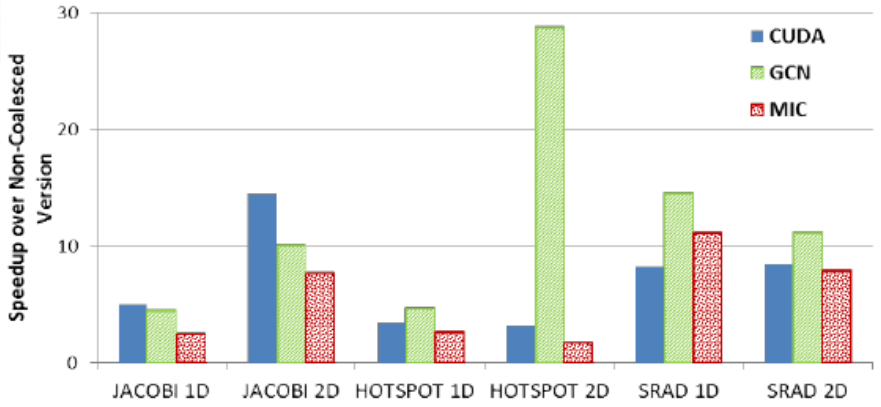


Figure 5: Memory Coalescing Benefits on Different Architectures : MIC is impacted the least by the non-coalesced accesses

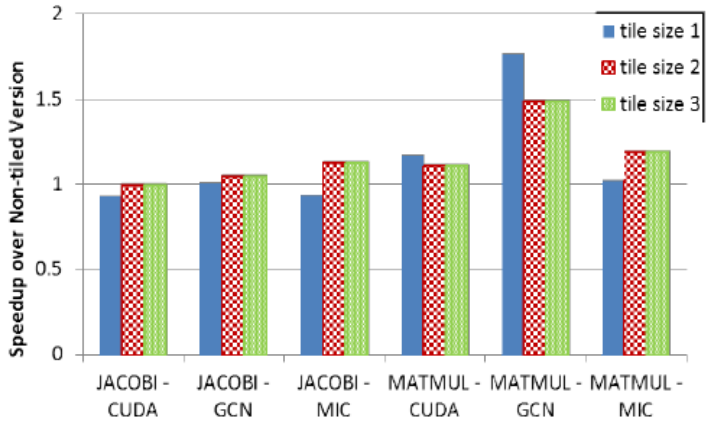


Figure 7: Impact of Tiling Transformation : *MATMUL* shows higher benefits than *JACOBI* owing to more contiguous accesses

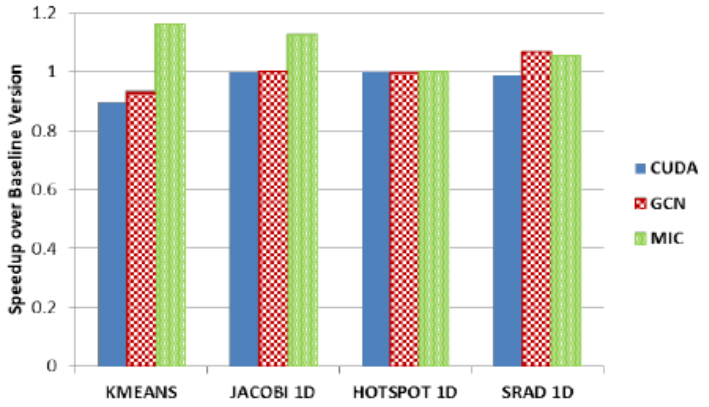


Figure 9: Effects of Loop Unrolling - MIC shows benefits on unrolling

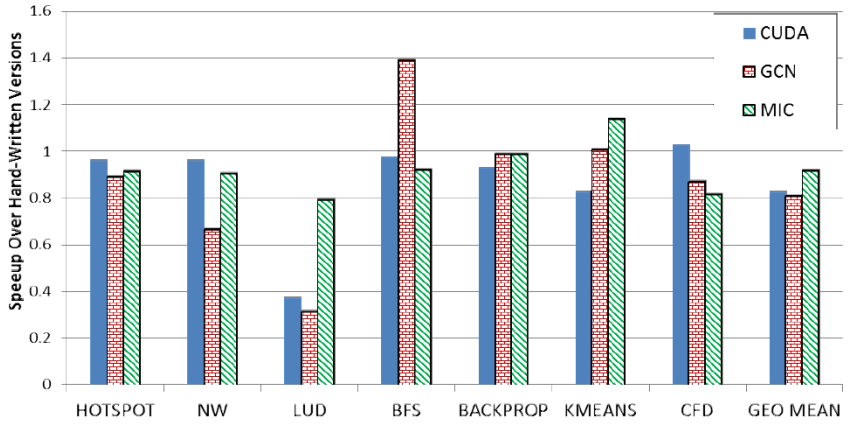


Fig. 11: Comparison of hand-written CUDA/OpenCL programs against auto-tuned OpenARC code versions : Tuned OpenACC programs perform reasonably well against hand-written codes

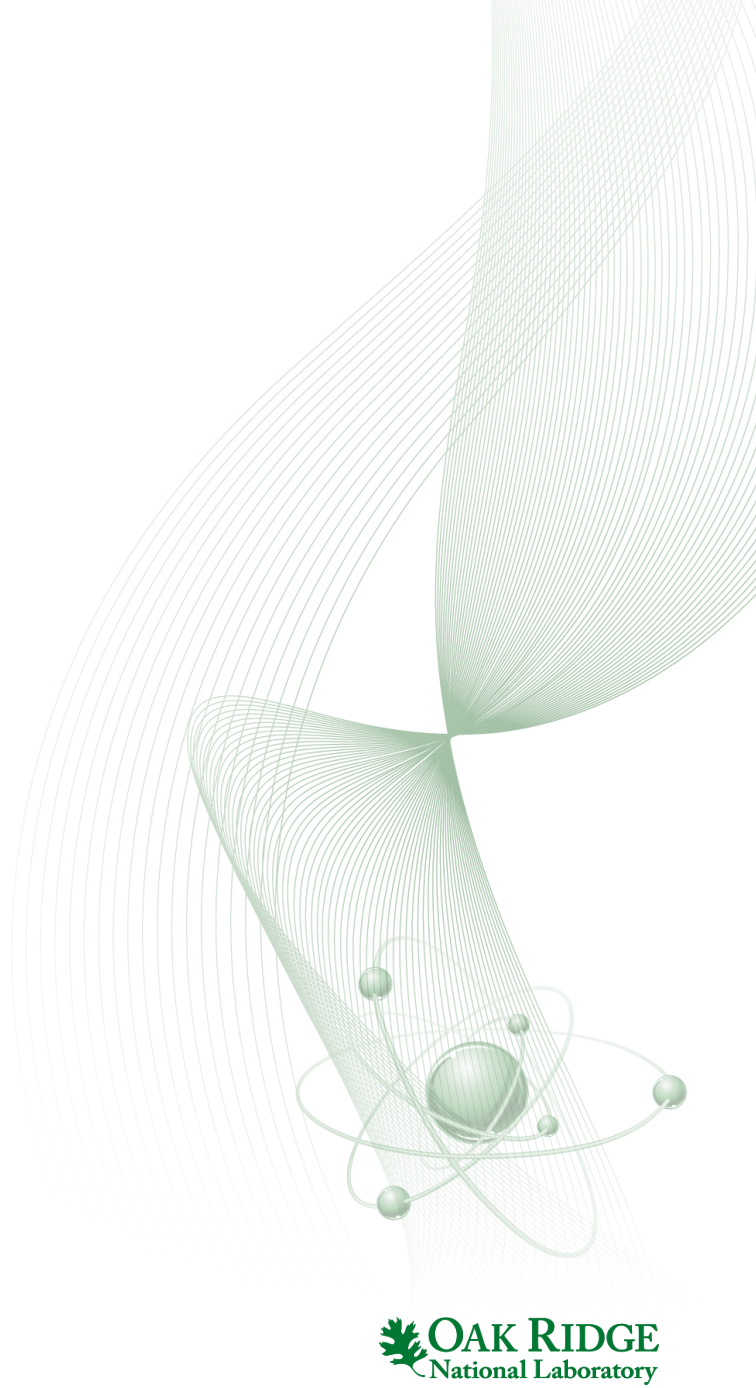
Overview

- Our community has major challenges in HPC as we move to extreme scale
  - Power, Performance, Resilience, Productivity
  - New technologies emerging to address some of these challenges
    - Heterogeneous computing
    - Multimode memory systems including nonvolatile memory
  - Not just HPC: Most uncertainty in at least two decades
  - Exascale includes even more diversity and uncertainty
- We need performance prediction and portability 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
- OpenARC research compiler is a vehicle for
  - Understanding how to automate platform specific optimizations
  - Developing performance portable code

Acknowle

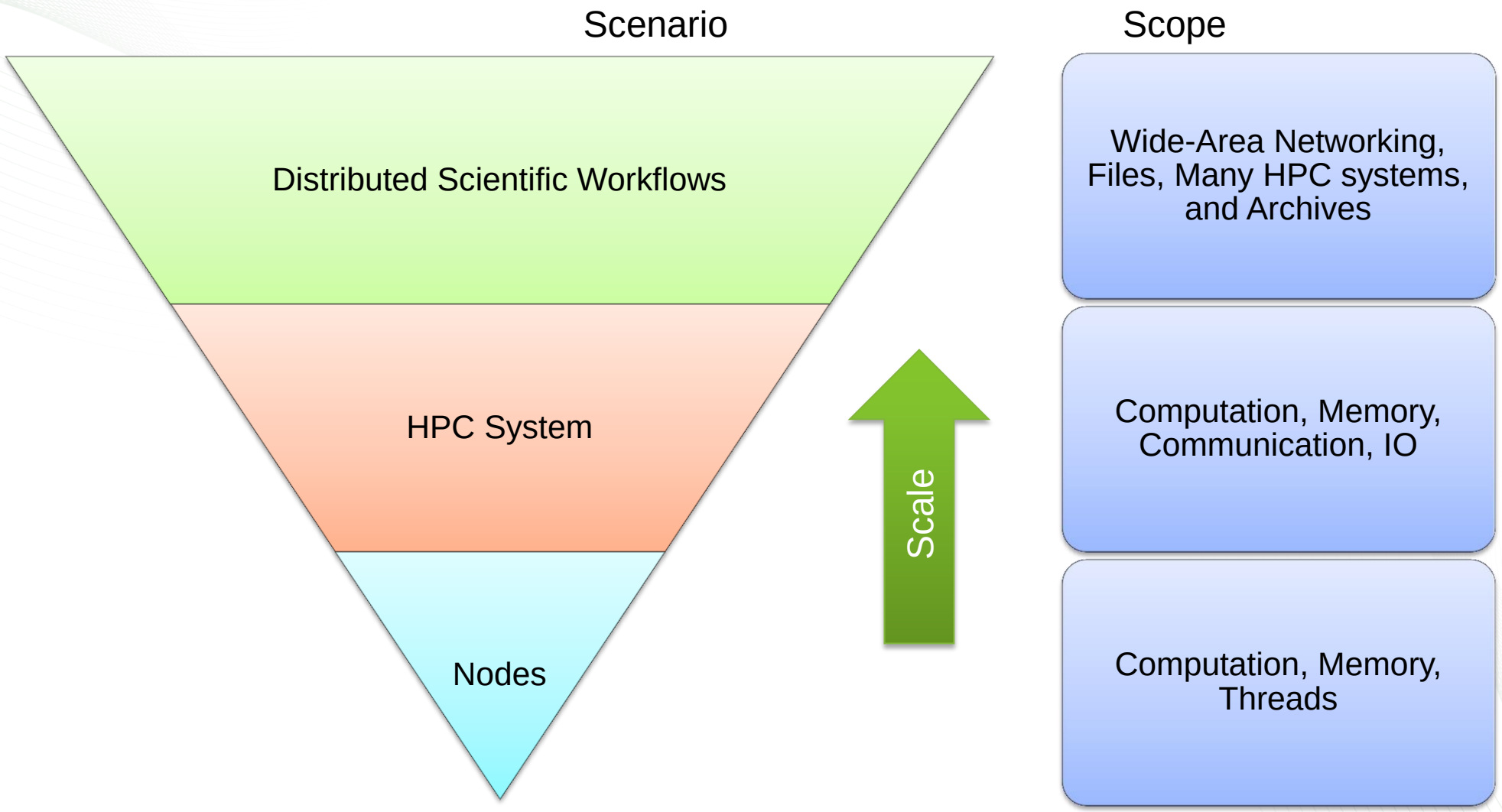
- Contributors and Sponsors
  - Future Technologies Group: http://ft.ornl.gov
  - US Department of Energy Office of Science
    - DOE Vancouver Project: https://ft.ornl.gov/trac/vancouver
    - DOE Blackcomb Project: https://ft.ornl.gov/trac/blackcomb
    - DOE ExMatEx Codesign Center: http://codesign.lanl.gov
    - 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

Performance Modeling for Distributed Scientific Workflows

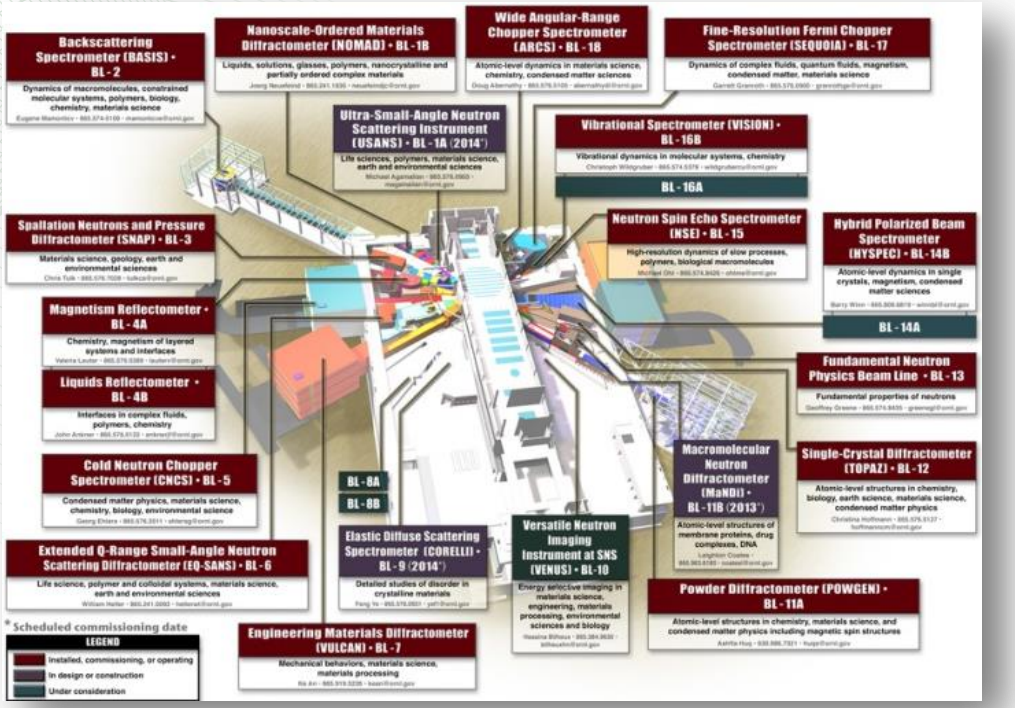


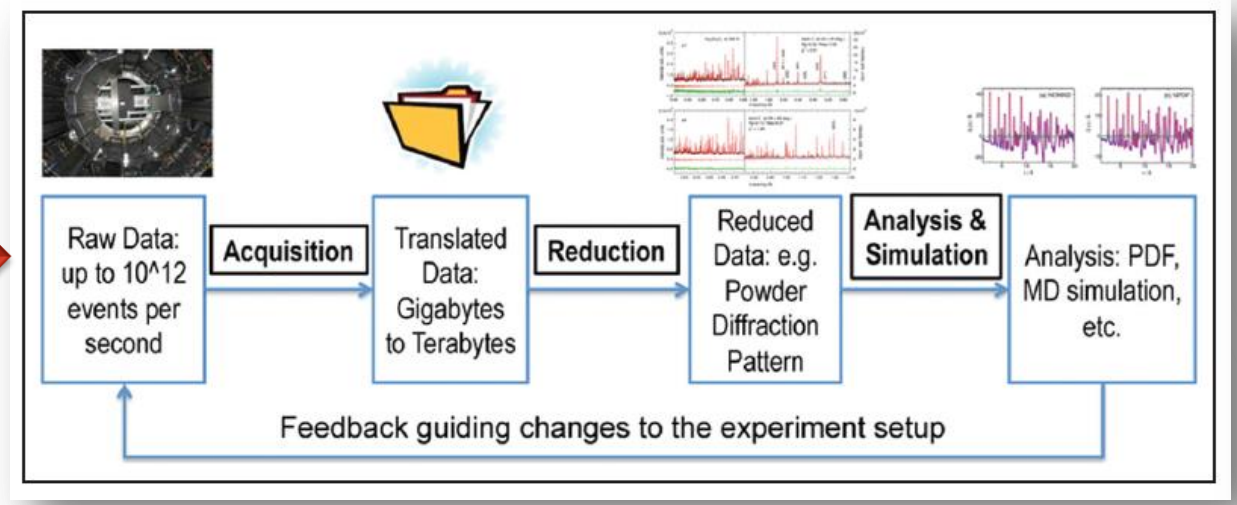
OAK RIDGE National Laboratory

Aspen allows Multiresolution Modeling



Workflow: SNS





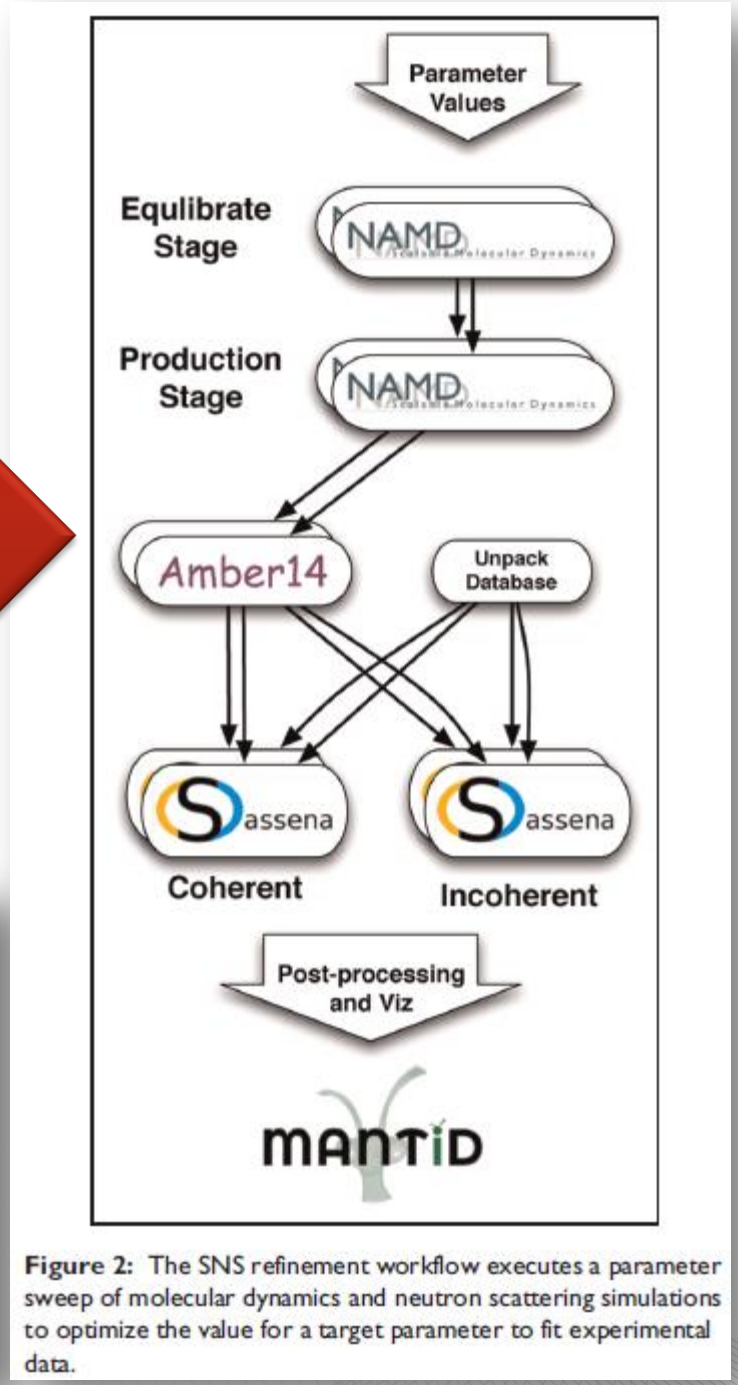
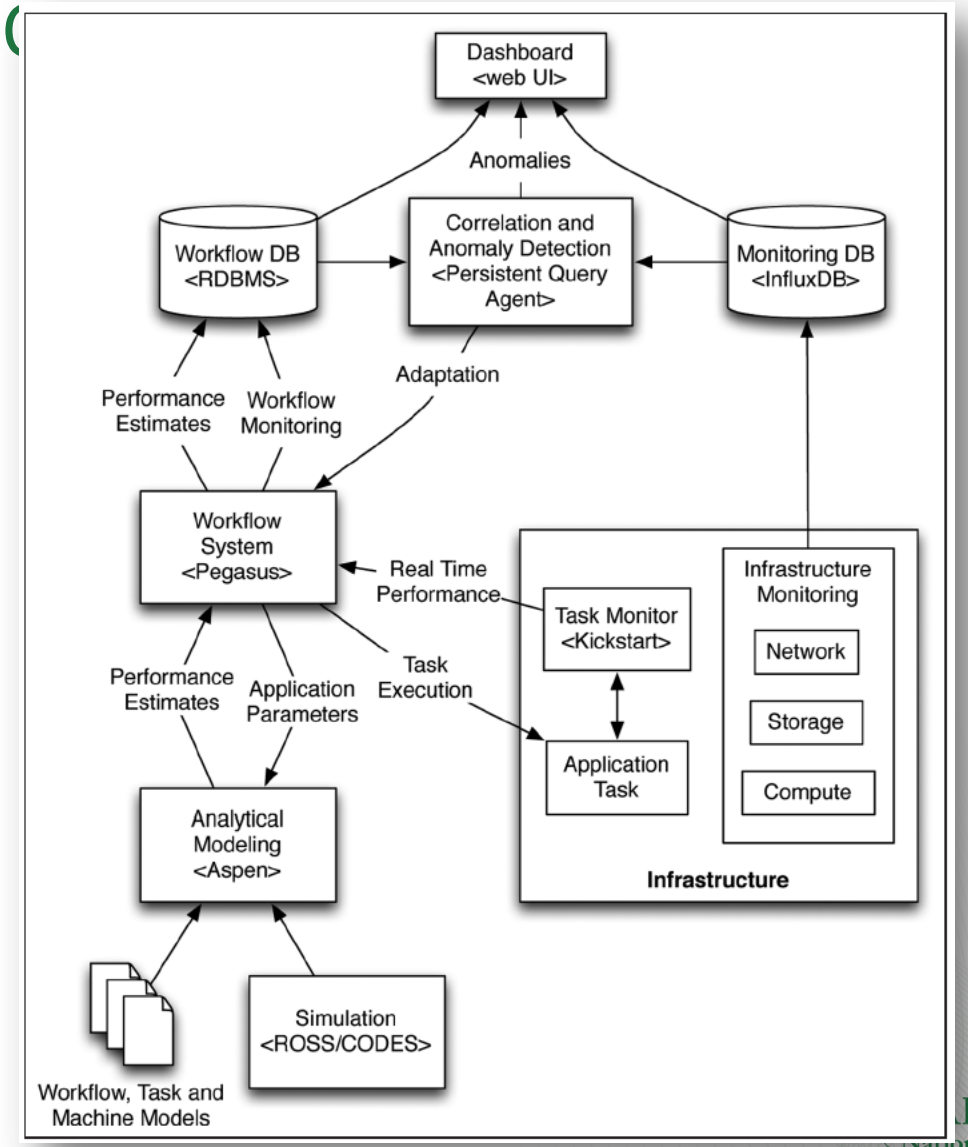


Figure 2: The SNS refinement workflow executes a parameter sweep of molecular dynamics and neutron scattering simulations to optimize the value for a target parameter to fit experimental data.

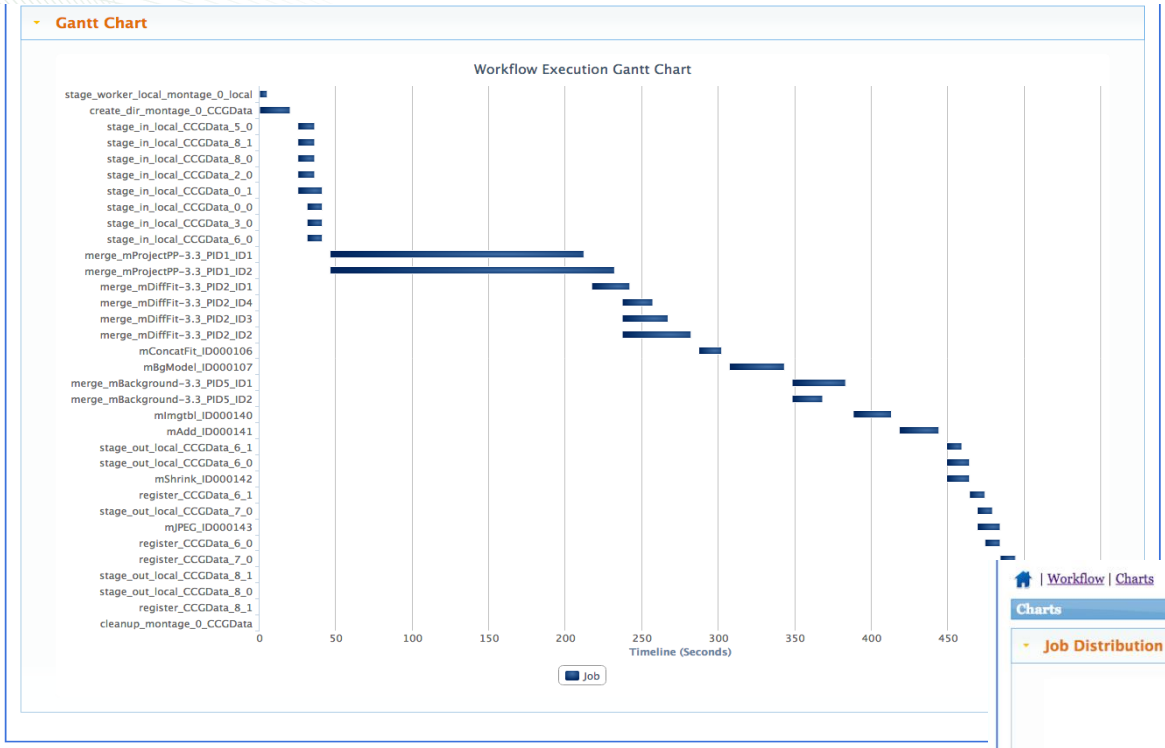
Automatically Generate Aspen from Pegasus DAX; Use Aspen Predictions to Inform/Monitor Decisi

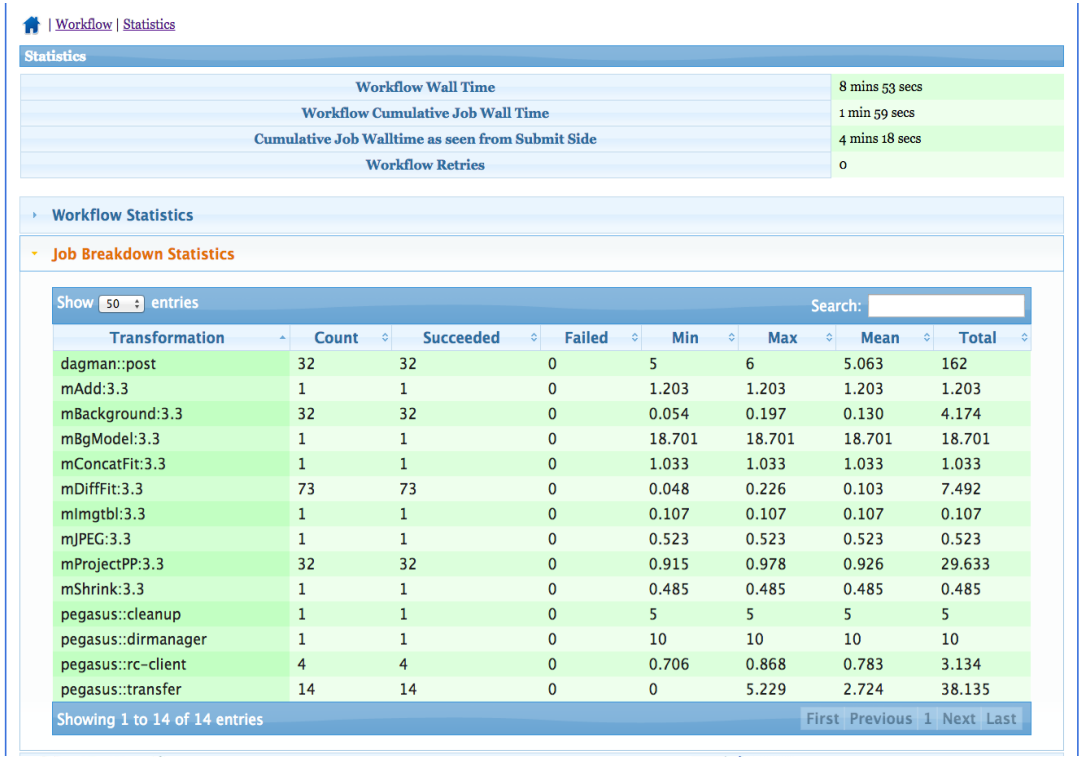
```
1  kernel main
2  {
3      par {
4          seq {
5              call namd_eq_200()
6              call namd_prod_200()
7          }
8          seq {
9              call namd_eq_290()
10             call namd-prod-290()
11         }
12     }
13     par {
14         call unpack_database()
15         call ptraj_200()
16         call ptraj-290()
17     }
18     par {
19         call sassena-incoh-200()
20         call sassena_coh_200()
21         call sassena_incoh_290()
22         call sassena-coh-290()
23     }
24 }
```

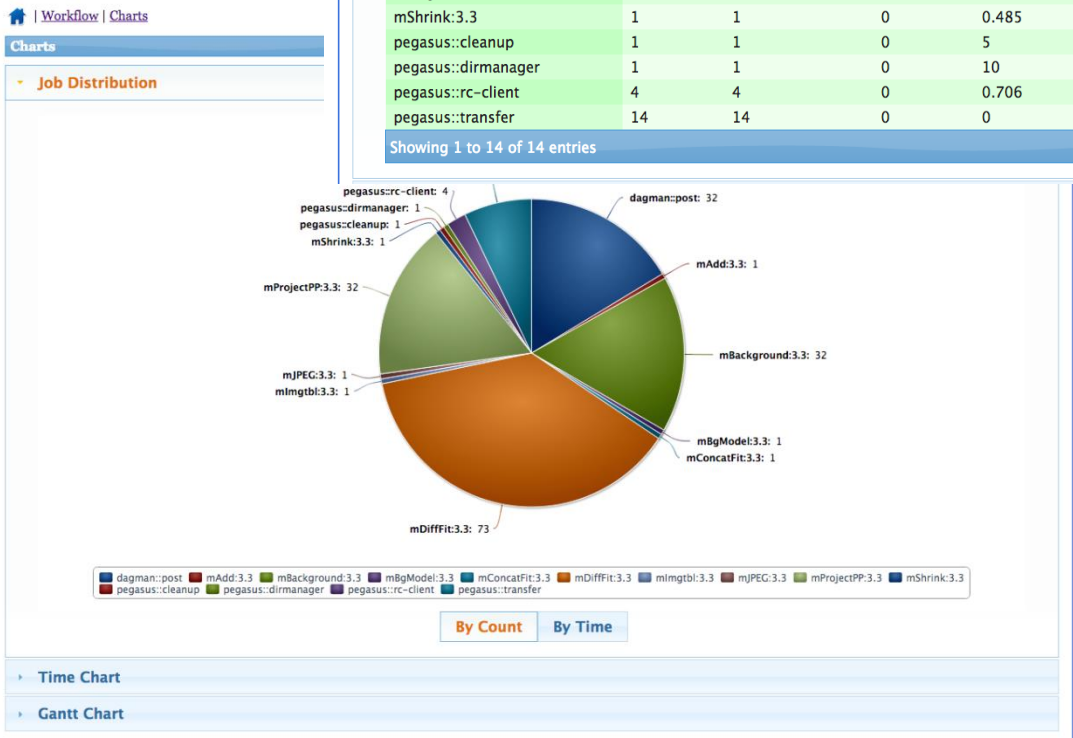
Listing 1: Automatically generated Aspen model for kample SNS workflow.



Workflow Monitoring Dashboard – *pegasus-dashboard*







Status, statistics, timeline of jobs

Helps pinpoint errors