資料１

エクサに向けたプロジェクト（米国編）

東京工業大学
学術国際情報センター
松岡　聡

文部科学省
第２回HPCI検討ワーキンググループ　プレゼン資料
2012年5月30日(水)

Some slides and/or Contents Courtesy of Peter Beckman @ ANL, Bill Harrod @ DoE

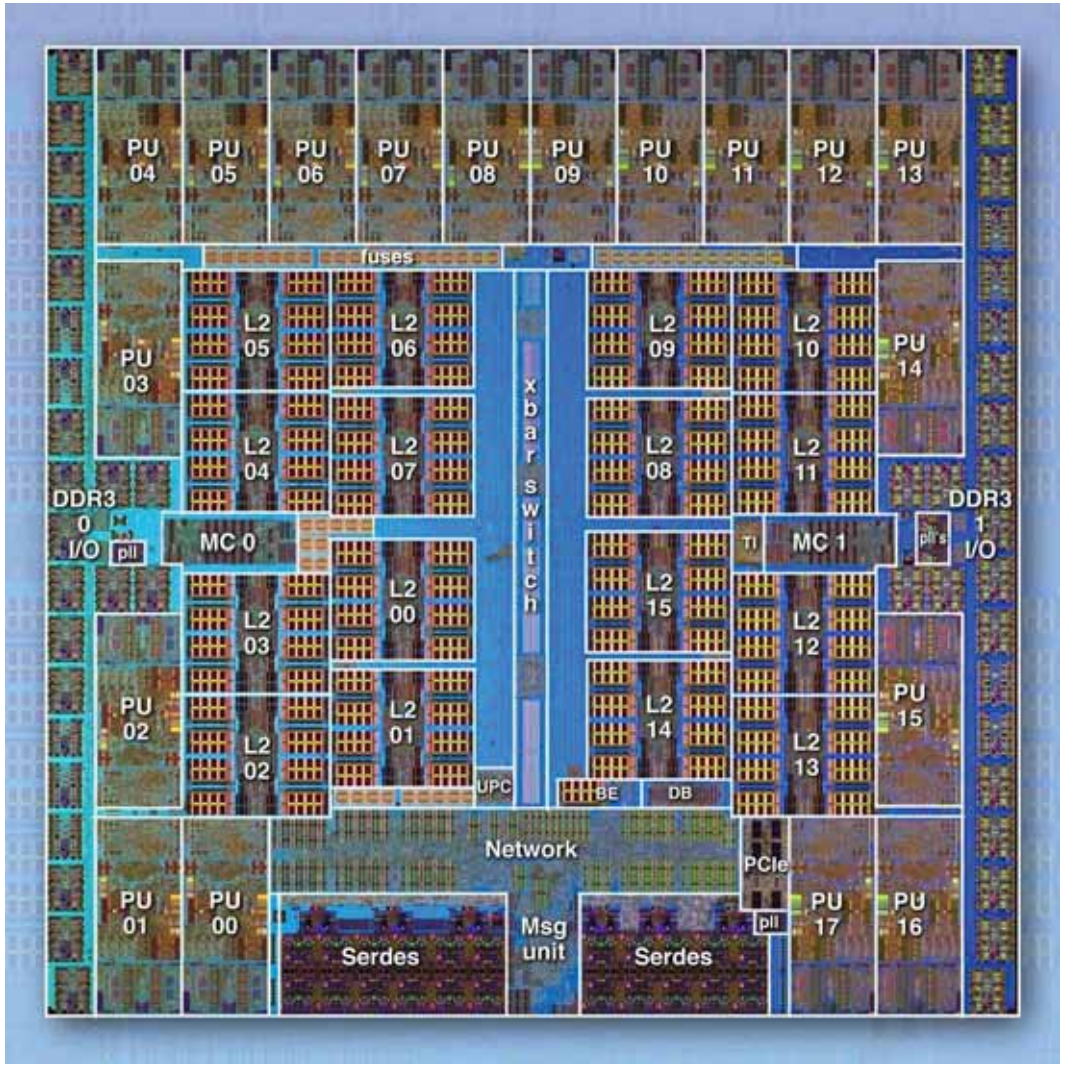
Mira: Argonne's Newest GREEN Supercomputer

- Blue Gene/Q System
  - 48 racks
  - 786,432 cores
  - 786 TB of memory
  - Peak flop rate: 10 PF
- Half size of LLNL Sequoia

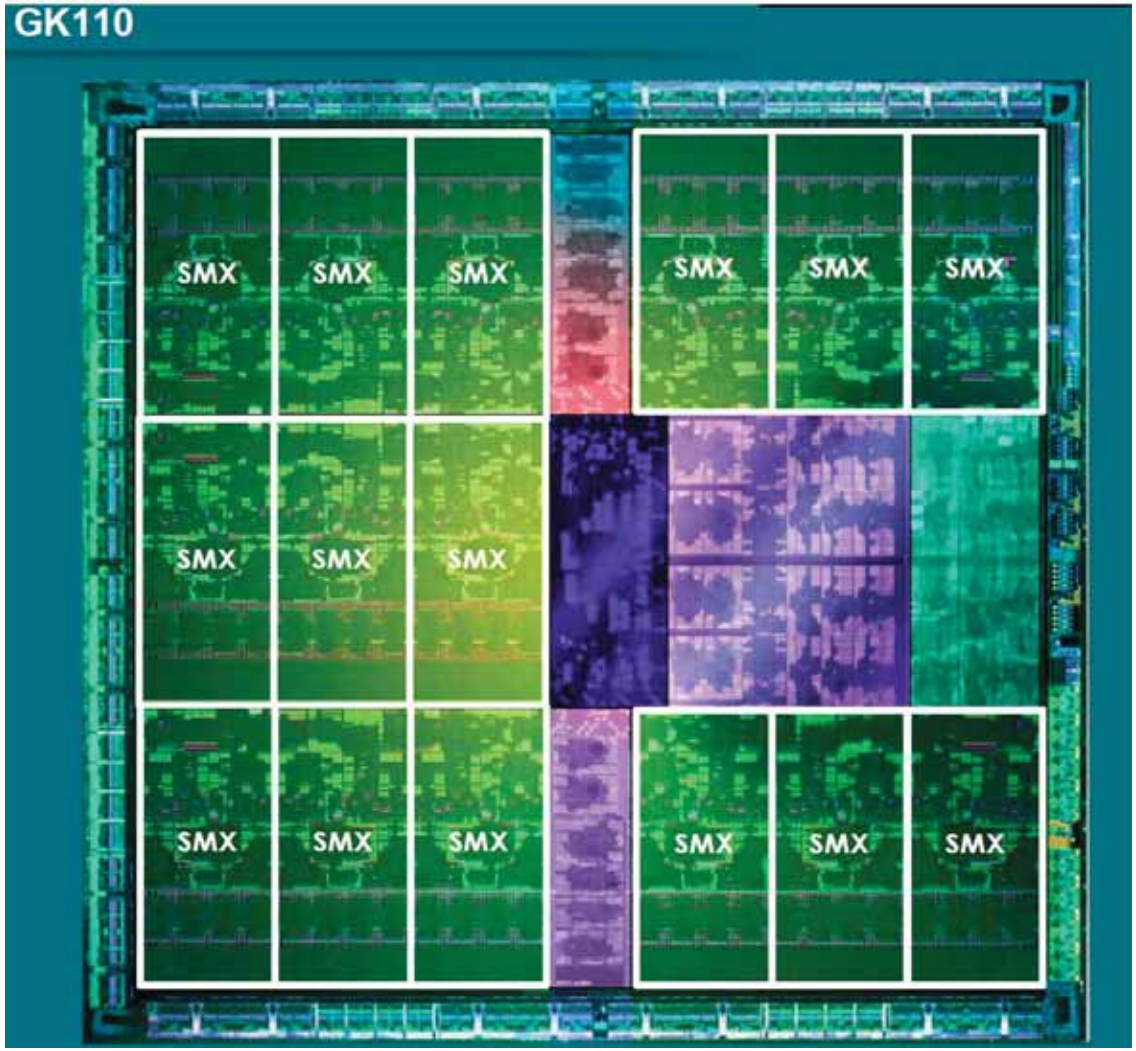
- Storage System
  - ~30 PB capability
    - 240GB/s bandwidth (GPFS)

BlueGene/Q Compute chip

System-on-a-Chip design : integrates processors, memory and networking logic into a single chip



- **360 mm² Cu-45 technology (SOI)**
  - ~ 1.47 B transistors
- **16 user + 1 service PPC processors**
  - plus 1 redundant processor
  - all processors are symmetric
  - each 4-way multi-threaded
  - 64 bits
  - 1.6 GHz
  - L1 I/D cache = 16kB/16kB
  - L1 prefetch engines
  - each processor has Quad FPU (4-wide double precision, SIMD)
  - peak performance 204.8 GFLOPS @ 55 W
- **Central shared L2 cache: 32 MB**
  - eDRAM
  - multiversioned cache – will support transactional memory, speculative execution.
  - supports atomic ops
- **Dual memory controller**
  - 16 GB external DDR3 memory
  - 1.33 Gb/s
  - 2 * 16 byte-wide interface (+ECC)
- **Chip-to-chip networking**
  - Router logic integrated into BQC chip.
- **External IO**
  - PCIe Gen2 interface



NVIDIA “Kepler2” GK110
28nm TSMC, ~600m2?
7.1 Billion Transistors
2880 CUDA Cores, 15 SMXs
4.? TFLOPS SFP / 1.? TFLOPS DFP
> 200GB/s Memory BW
6~XXGB GDDR5 Memory
PCIe3 Interface
GPU Direct3 – Direct PCIe transfer to IB and other HCAs (no CPU memory buffering)
Hyper-Q multi-job queuing
Hardware-assisted dynamic parallelism
GPU Virtualization
CUDA5 & OpenACC directive-based programming

C.f. Power7: 1.2B, BG/Q: 1.5B, SPARC IX fx: 1.9B, Sandy-Bridge EP: 2.3B, GF110: 3.0B, GK110: 7.1B

18,000 Kepler2s in ORNL Jaguar -> Titan (Cray XK6) Upgrade

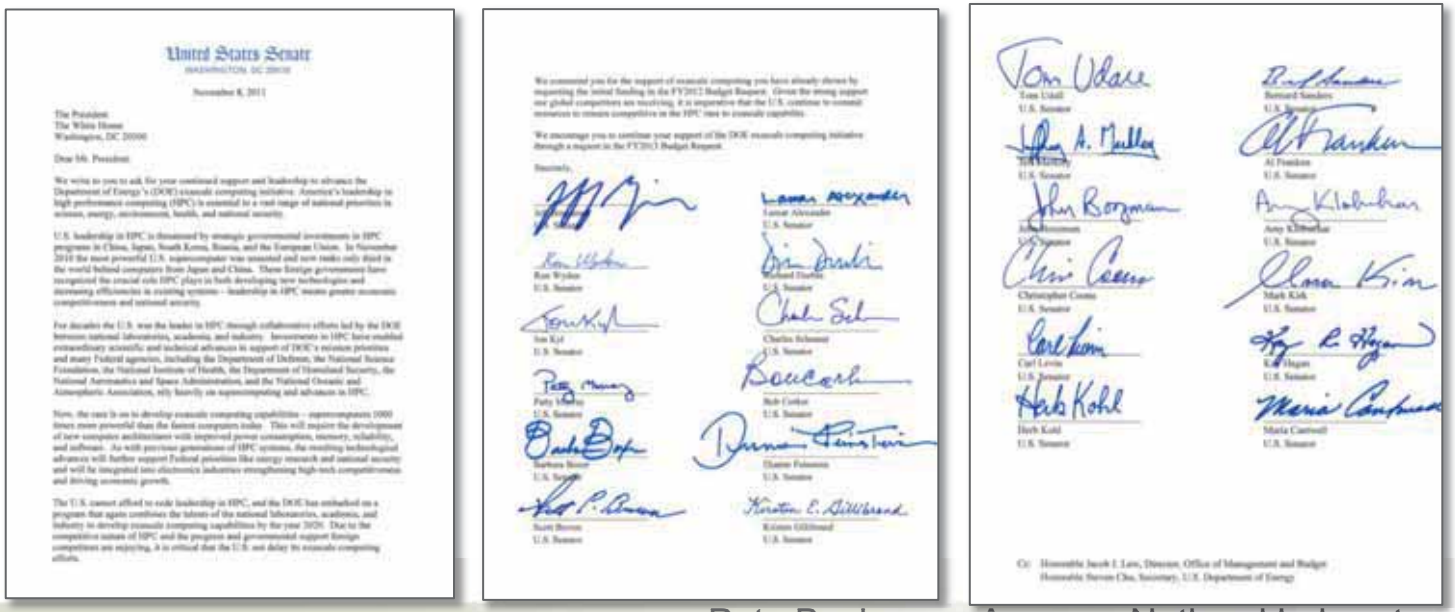
Exascale is International
Discussions and plans almost 5 years in the making…

A grand challenge for the 21st century

Development of an Exascale Computing System is a Grand Challenge for the 21st Century

"[Development of] An "exascale" supercomputer capable of a million trillion calculations per second – dramatically increasing our ability to understand the world around us through simulation and slashing the time needed to design complex products such as therapeutics, advanced materials, and highly-efficient autos and aircraft."

Sept 20th 2009
EXECUTIVE OFFICE OF THE PRESIDENT NATIONAL ECONOMIC COUNCIL OFFICE OF SCIENCE AND TECHNOLOGY POLICY



U.S. DEPARTMENT OF ENERGY

Office of Science

Current Exascale Programs (2011-12)

- Advanced Architectures and Critical Technologies for Exascale
  - 6 projects focused on power management, memory management, and reducing the cost of data movement
- R&E Prototpyes
- X-Stack Software Research
  - 10 projects focused on operating systems, fault tolerance, programming challenges, performance optimization, etc.
- Scientific Data Management and Analysis at Extreme Scale
  - 10 projects spanning file systems and I/O, data triage, feature detection and data analysis, and visualization

U.S. DEPARTMENT OF ENERGY

Office of Science

Exascale Co-Design Centers

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

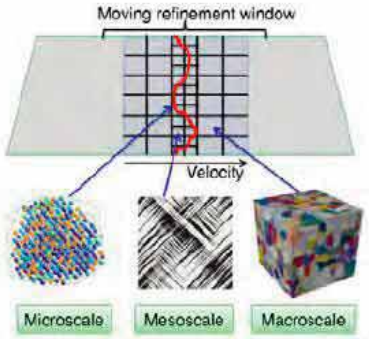
Director: Timothy Germann (LANL)

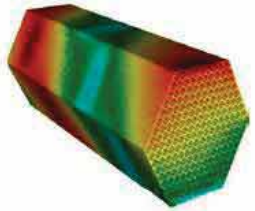
Center for Exascale Simulation of Advanced Reactors (CESAR)

Director: Robert Rosner (ANL)

Combustion Exascale Co-Design Center (CECDC)

Director: Jacqueline Chen (SNL)

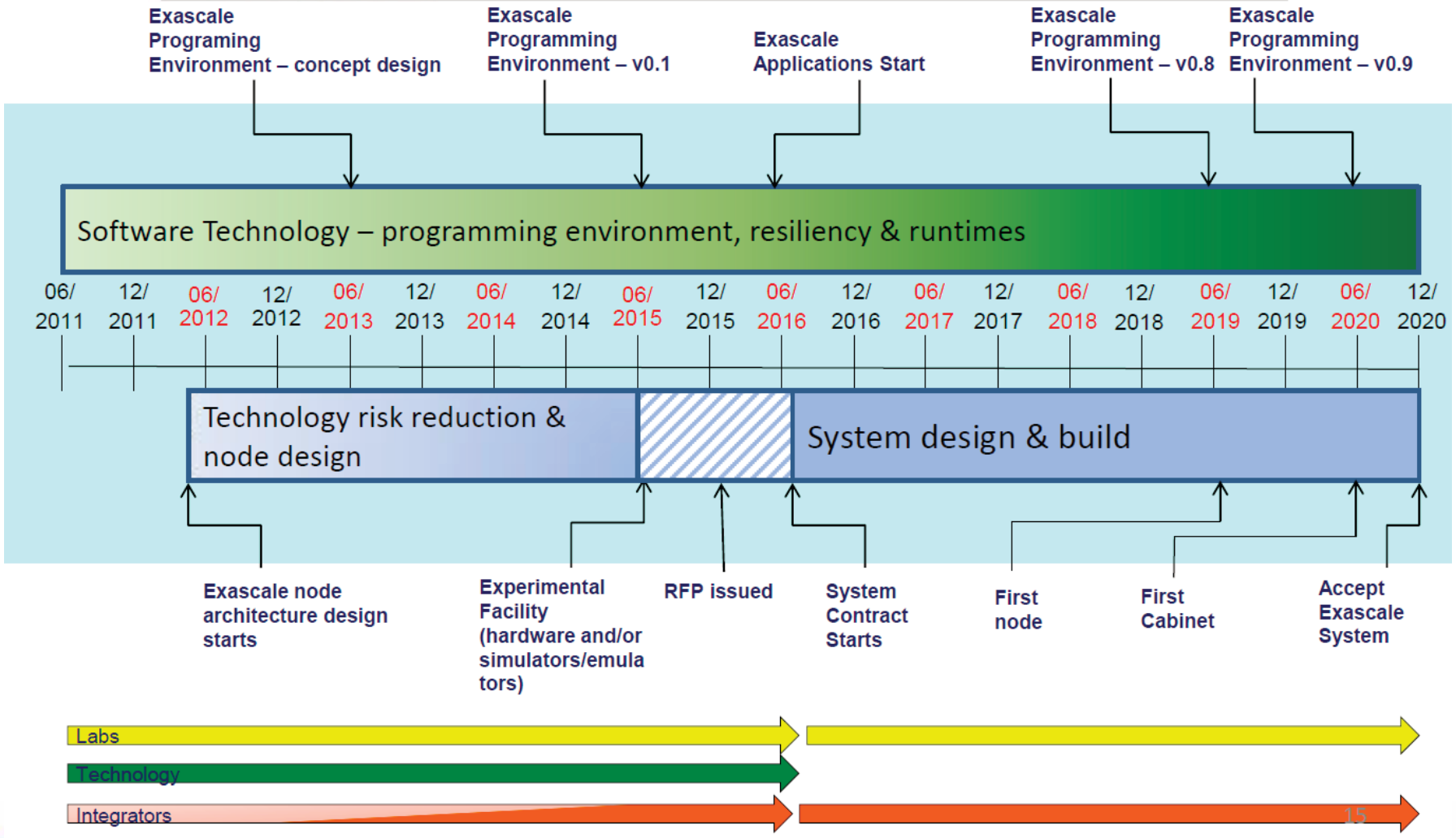




	ExMatEx (Germann)	CESAR (Rosner)	CECDC (Chen)
National Labs	LANL	ANL	SNL
	LLNL	PNNL	LBNL
	SNL	LANL	LANL
	ORNL	ORNL	ORNL
		LLNL	LLNL
			NREL
University & Industry Partners	Stanford	Studsvik	Stanford
	CalTech	TAMU	GA Tech
		Rice	Rutgers
		U Chicago	UT Austin
		IBM	Utah
		TerraPower	
		General Atomic	
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

Exascale Reverse Timeline

U.S. DEPARTMENT OF ENERGY
Office of Science



U.S. DEPARTMENT OF ENERGY

Office of Science

Anticipated Future Programs

- Programming Models, Languages, Compilers, and Tools
  - Minimize exposure of system complexity
  - Extreme concurrency
  - Heterogeneous system
  - Minimize data movement
- X-Stack
  - Strong focus on runtimes for efficiency and resiliency
  - Self-aware OS
- Exascale Architectures
  - Abstract machine models for design space exploration, utilizing simulation
  - Driven by DOE selected applications
- Extreme Scale Solver Algorithms
  - Fine grain parallelism
  - Data movement & locality

Exascale… Xstack (new call) $45mil? 2012-2015

2011

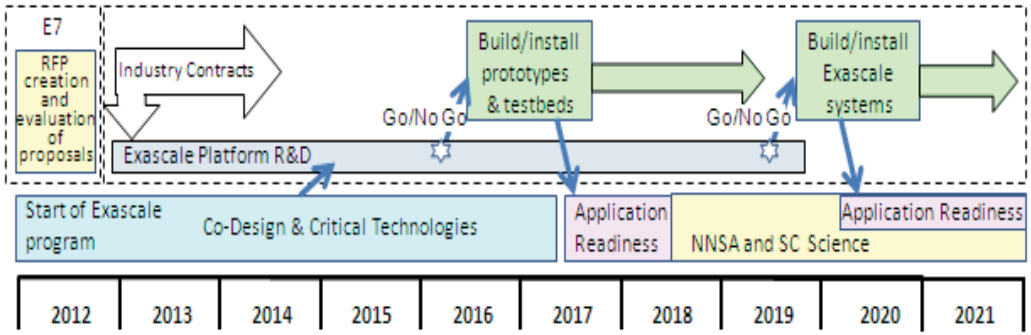


Table 1. Exascale System Goals

Exascale System	Goal
Delivery Date	2019
Performance	1000 PF LINPACK and 300 PF on to-be-specified applications
Power Consumption*	20 MW
MTBAI**	6 days
Memory including NVRAM	128 PB
Node Memory Bandwidth	4 TB/s
Node Interconnect Bandwidth	400 GB/s

*Power consumption includes only power to the compute system, not associated storage or cooling systems.
**The mean time to application failure requiring any user or administrator action must be greater than 24 hours, and the asymptotic target is improvement to 6 days over time. The system overhead to handle automatic fault recovery must not reduce application efficiency by more than half.
PF = petaflop/s, MW = megawatts, PB = petabytes, TB/s = terabytes per second, GB/s = gigabytes per second, NVRAM = non-volatile memory.

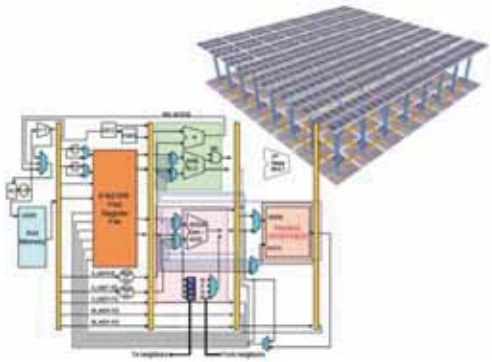
2012

- RFP was due May 11th
- Process
- Memory
- Storage and I/O
- Good responses

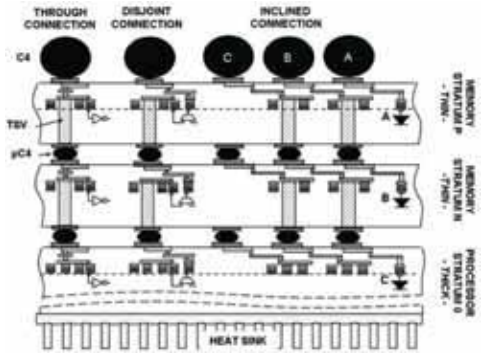
The 4 Issues for Exascale Software

- Memory & Interconnect
- Low Power
- Parallelism
- Fault

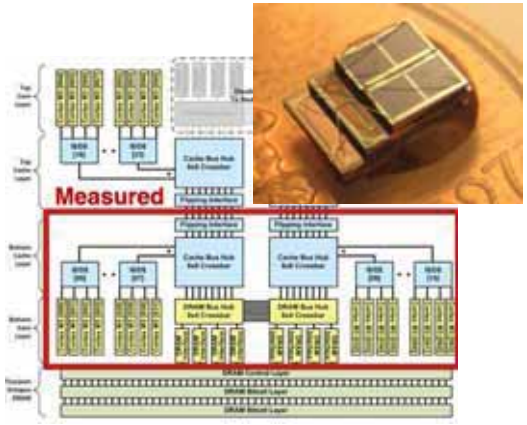
3D Chip Stacking: Fast, Close, (relatively) Small



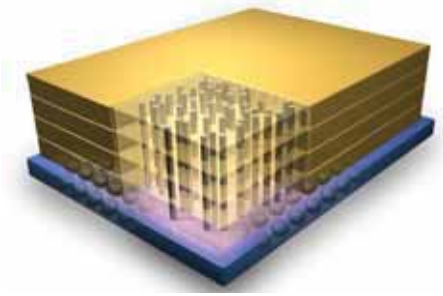
Georgia Tech



IBM

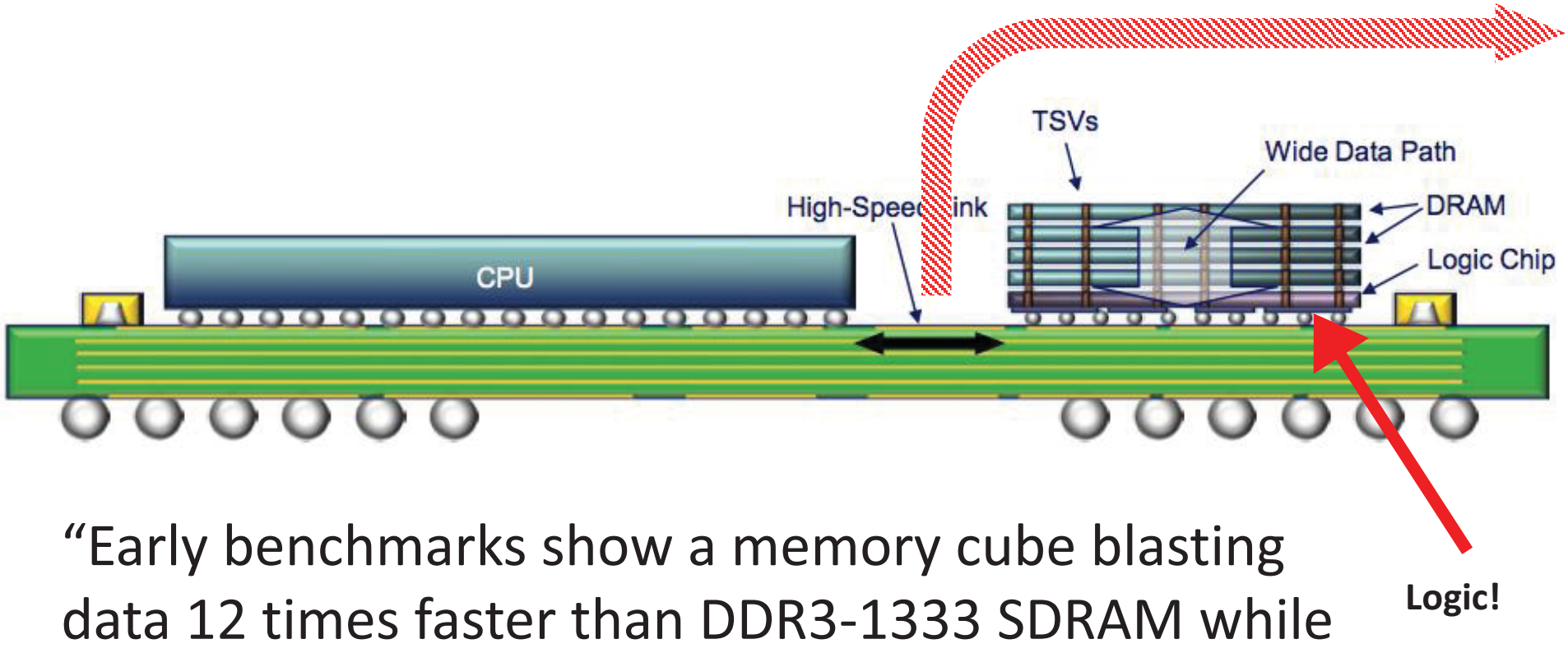


Univ of Michigan



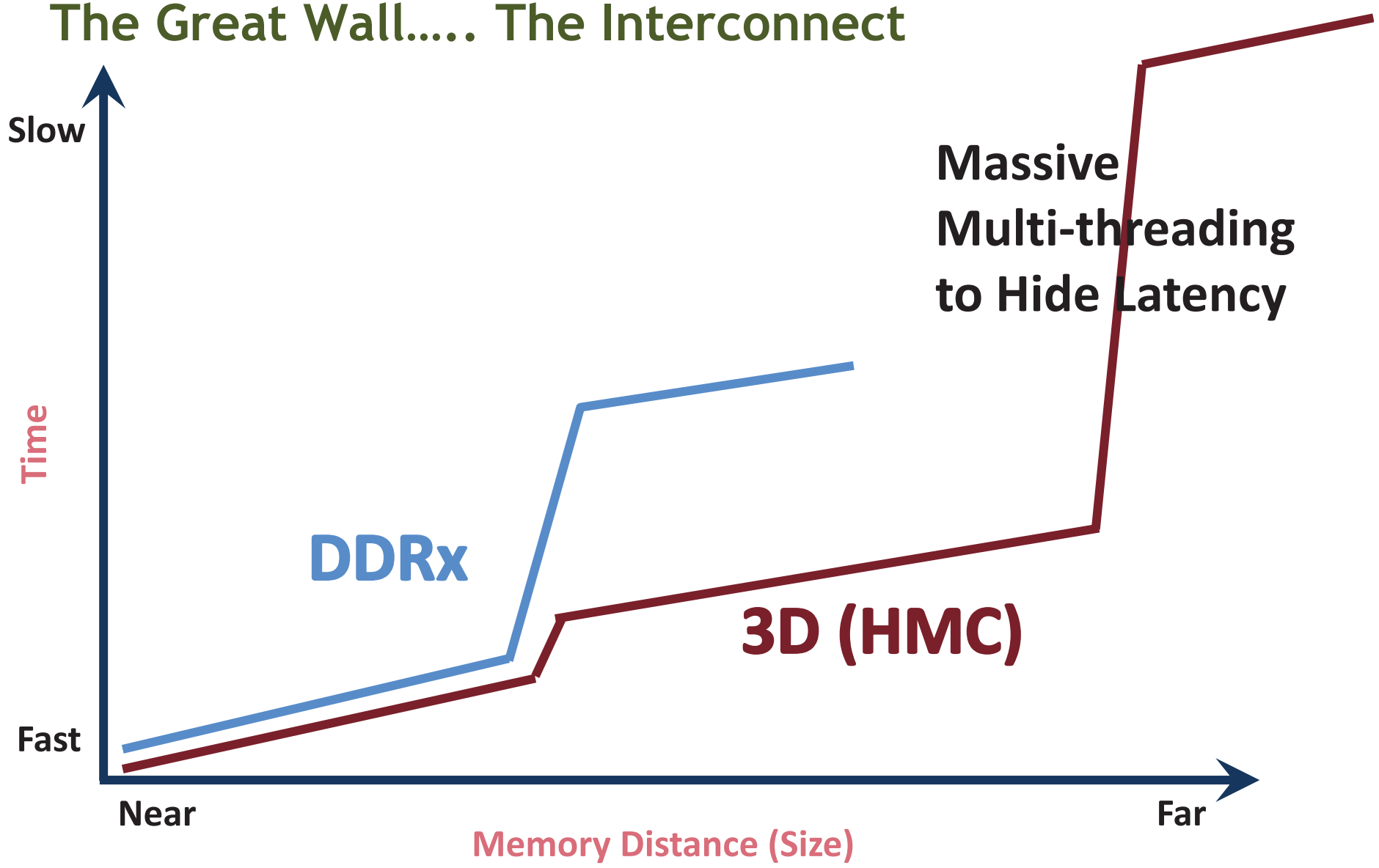
Micron HMC

Micron Hybrid Memory Cube



"Early benchmarks show a memory cube blasting data 12 times faster than DDR3-1333 SDRAM while using only about 10 percent of the power."

The Great Wall….. The Interconnect



Impact on System Software: Memory / Interconnect

- Intra-node data movement
  - *Data movement dominates power*
    - Explicit core-to-core data movement
    - MPI for intranode?
  - Programmable memory logic functions
- Next-gen message layer (to hide latency):
  - Redesigned for massive multithreading
    - Not just message rates, but pending requests
  - Implementation must become parallel
- OS/R: lightweight active messages & threads
- Design Question: Interconnect to Memory or CPU?

Parallelism

Raspberry Pi: $25

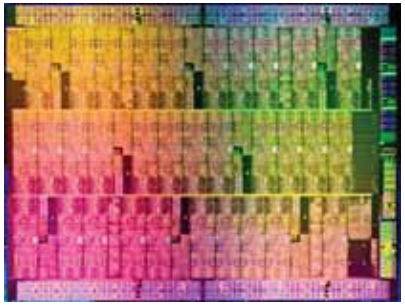
- 700MHz ARM11
- $25
- 4 cores

On-chip Parallelism Exploding
"The core is the new Mhz"

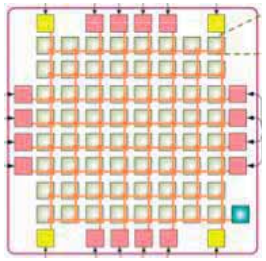
- 2008: largest system had O(100K) cores
- Today (2012)

LLNL BG/Q	1600K cores
RIKEN K	705K cores
Jülich BG/P	295K cores
ORNL XT5	224K cores
ANL BG/P	164K cores

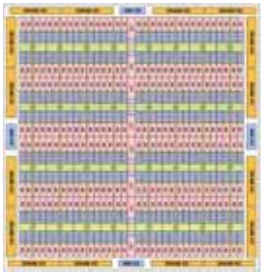
Key Changes:
Coherency, Power Management, Specialization



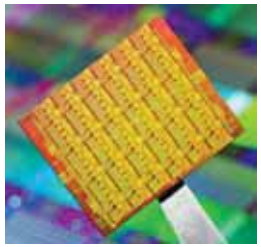
Intel: MIC



Godson T

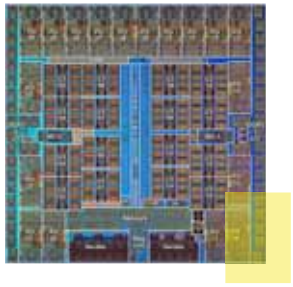


Dally: Echelon



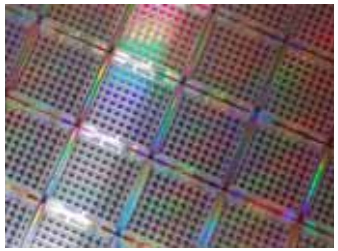
Intel: SCC

Chien: 10x10



IBM: BG/Q

#18



Tilera: GX

Power Constrained Memory Consistency

Extreme Specialization and Power Management

Static or Dynamic?

- We must switch to dynamic view of our parallel abstract machine
  - Automatic correction of faults
  - Explicit power management
  - Implicit power management
  - Contention
- Massive parallelism -> static is unscalable
- How will our programming change?

In-Socket Parallel Programming is a Mess:

```
#pragma omp parallel for        \
  default(shared) private(i)    \
  schedule(static,chunk)        \
  reduction(+:result)

  for (i=0; i < n; i++)
    result = result + (a[i] * b[i]);

printf("Final result= %f\n",result);
```

Clause	Directive					
	PARALLEL	DO/for	SECTIONS	SINGLE	PARALLEL DO/for	PARALLEL SECTIONS
IF	●				●	●
PRIVATE	●	●	●	●	●	●
SHARED	●	●			●	●
DEFAULT	●				●	●
FIRSTPRIVATE	●	●	●	●	●	●
LASTPRIVATE		●	●		●	●
REDUCTION	●	●	●		●	●
COPYIN	●				●	●
COPYPRIVATE				●		
SCHEDULE		●			●	
ORDERED		●			●	
NOWAIT		●	●	●		

```
float function FTNReductionOMP(data, size)
float data(*)
integer size
ret = 0.0

!dir$ omp offload target(mic) in(size) in(data:length(size))
!$omp parallel do reduction(+:ret)
do i=1,size
        ret = ret + data(i)
enddo
!$omp end parallel do

FTNReductionOMP = ret
```

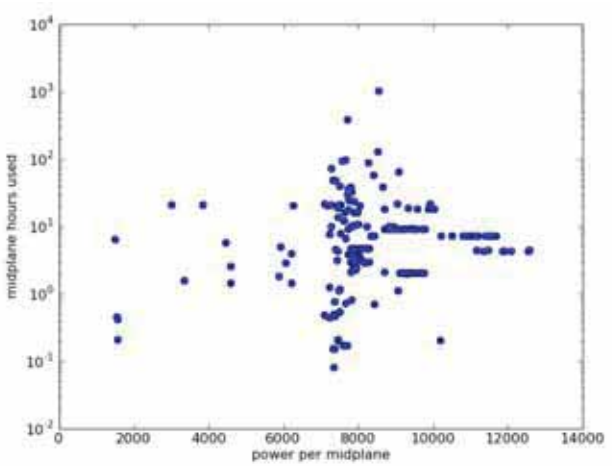
System Software Challenges:

- OpenMP is a mess
- OpenMP is not used by compiler writers
- OpenMP is not used by message libs
- Representation of deep memory
- New, more expressive programming model

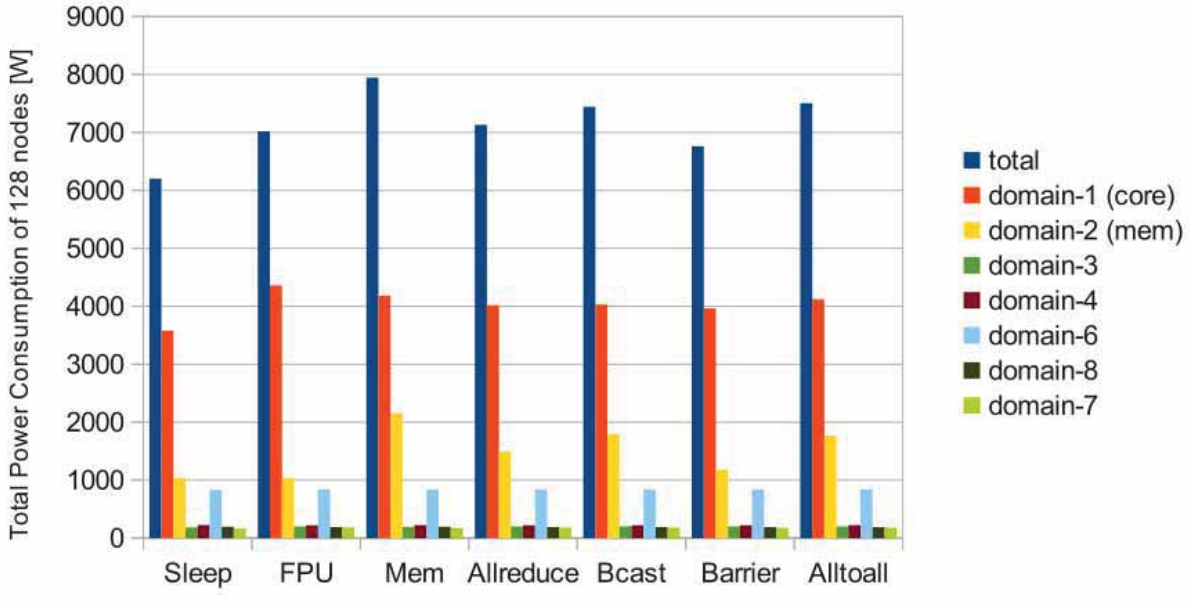
Impact on System Software: Parallelism

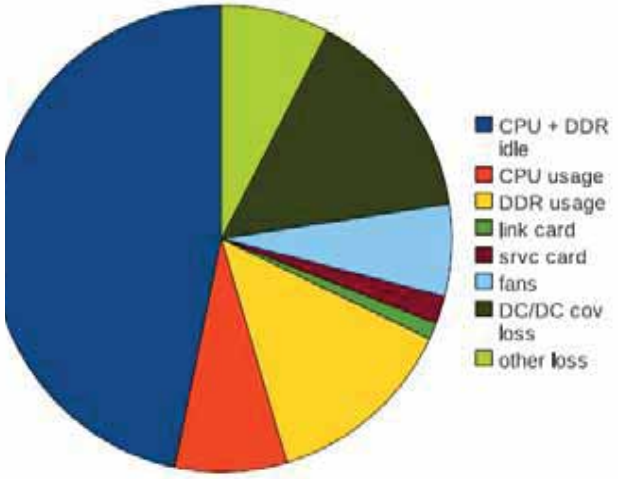
- Parallelism is growing exponentially in sockets
  - graphs/trees handle exponentially growing resources well
  - Fork/join and loop parallelism does not scale
- Systems are no longer static
  - graphs/trees handle load balancing well
- How will the community solve this?
  - Is a completely new programming model needed?
- OS/R redesign for massive numbers of dynamic threads, memory placement, and support for remote put/get

Low Power

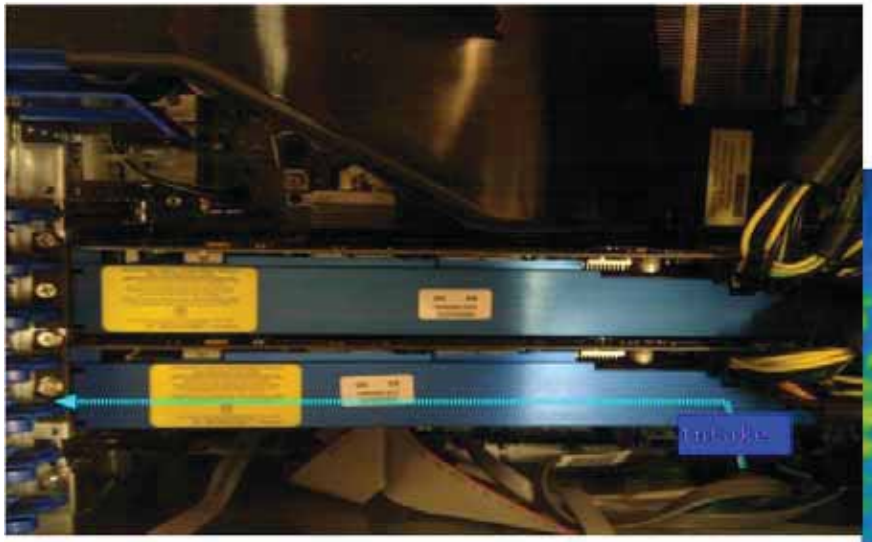


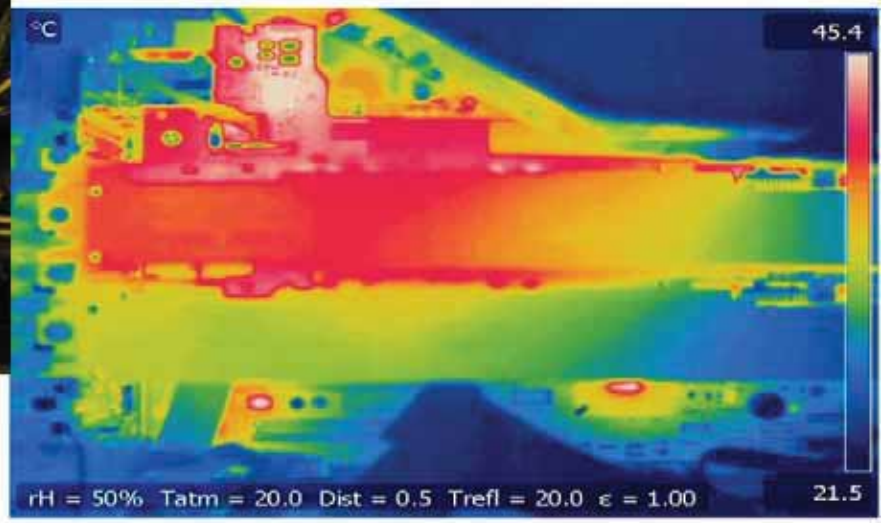
BG/P & BG/Q Power Experiments

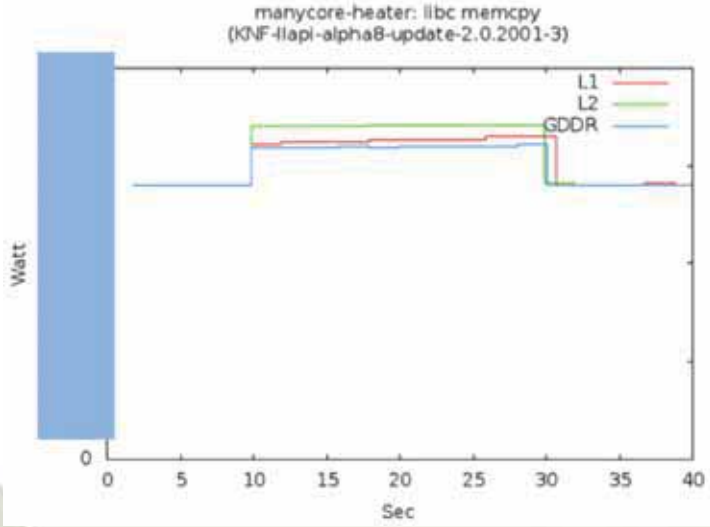




Exploring Power on Intel Knights Ferry





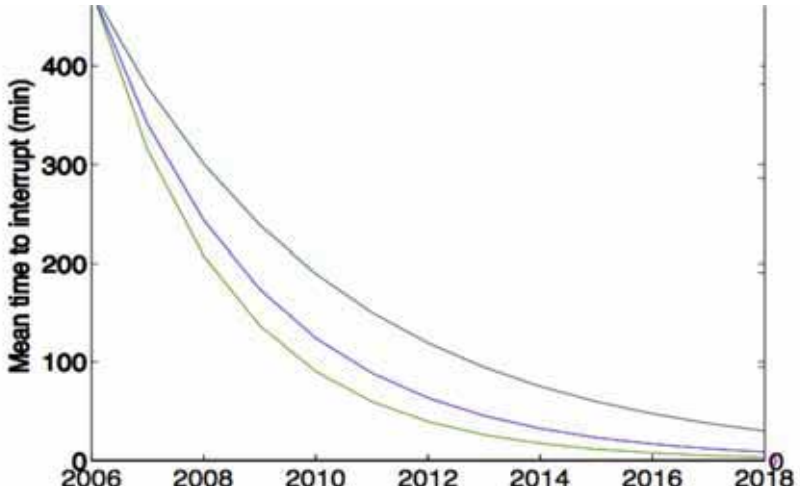


Impact on System Software: Low Power

- Power will become first-class managed resource
  - Dark Silicon: More functional units than can run at full speed
  - Variable speed subcomponents
  - New low-level interfaces for runtime systems
  - New algorithms to optimize performance for given Thermal Design Point (TDP)
- OS/R
  - Actively manage turning cores and memory on and off
  - Support for variable coherence domain to manage power
  - Heterogeneous (10x10) multi-core (graphics, compression, etc)
    - Programming model for this?
- Dynamic power-aware run-time system

Predictions are Hard

Example Prediction from 2007



"Over the past thirty years there have been several predictions of the eminent cessation of the rate of improvement in computer performance. Every such prediction was wrong. They were wrong because they hinged on unstated assumptions that were overturned by subsequent events.

What we do know:

- Driving down power increases faults
- Vendors have great market incentive to redesign for reliable hardware
- Our current HPC software is very fragile
- We should improve
- Build solutions at multiple layers

Summary: Exciting Times

- **Deep memory hierarchies**: 3D local RAM and NVRAM
- **Distributed memory**: cache coherence not power efficient
  - OS/R support dynamic selection of coherence domains
- **Parallelism** within a node is dramatically increasing
  - Current programming models are completely unprepared
- **Dynamic power management** is critical to performance
  - System software will develop APIs and new algorithms
- **Massive multithreading**: hide latency and provide dynamism
  - Overdecomposition, load balancing
- **Faults** may increase. Start building software now…

終わりに

- 米国はエクサスケールの「言いだしっぺ」、DoEを中心に計画促進
  - 政府としての超党派のエクサフロップスへのサポート
  - 今の10ペタ級の配備の遅れも2009年のリーマンショックなどの外因的要因が主で、2012年は世界の性能シェアにおいて記録的な割合を達成
- しかしながら、当初計画の2018年から2019-2020年への遅れ
  - 技術的要因: 電力・大規模化等の技術的制限により、ギガ⇒テラ、およびテラ⇒ペタより困難
  - 社会的要因:財政赤字による科学技術への投資の抑制⇒2014年度開始?
  - 研究的要因:データサイエンスの台頭、「計算するだけが能ではない」
- 一般IT、特にクラウド・IDCや組み込みとの相互レバレッジの発展
  - 水平展開・下方展開の重要さの強調⇒数は増加
  - 中心プレイヤーの変化:IBM, Cray等システムベンダーの縮小、Intel/NVIDIA/ARM等のチップベンダーの台頭、Amazon, Google, MS等IDC
  - メニーコアおよびベクトルプロセッサのコモディティ化・組込産業との連携も
  - システムソフトウェアの重要性が更に著しく:ハード開発中心主義からの脱却、Co-Designを重要視