Thoughts on system software for next-generation hardware

Pete Beckman

Director, Exascale Technology and Computing Institute (ETCi)

Argonne National Laboratory

Senior Fellow, Computation Institute

University of Chicago

What’s Happening in Exascale?
(do we care?)

INTERNATIONAL
EXASCALE
SOFTWARE PROJECT

ROADMAP

Jack Dongarra
Pete Beckman
Terry Moore
Jean-Claude Andre
Jean-Yves Berthou
Taisuke Boku
Franck Cappello
Barbara Chapman
Xuebin Chi

Alok Choudhary
Sudip Dosanjh
Al Geist
Bill Gropp
Robert Harrison
Mark Hereld
Michael Heroux
Adolfy Hoisie
Koh Hotta

Yutaka Ishikawa
Fred Johnson
Sanjay Kale
Richard Kenway
Bill Kramer
Jesus Labarta
Bob Lucas
Barney Maccabe
Satoshi Matsuoka

Paul Messina
Bernd Mohr
Matthias Mueller
Wolfgang Nagel
Hiroshi Nakashima
Michael E. Papka
Dan Reed
Mitsuhisa Sato
Ed Seidel

SPONSORS

Office of Science
U.S. Department of Energy

NSF

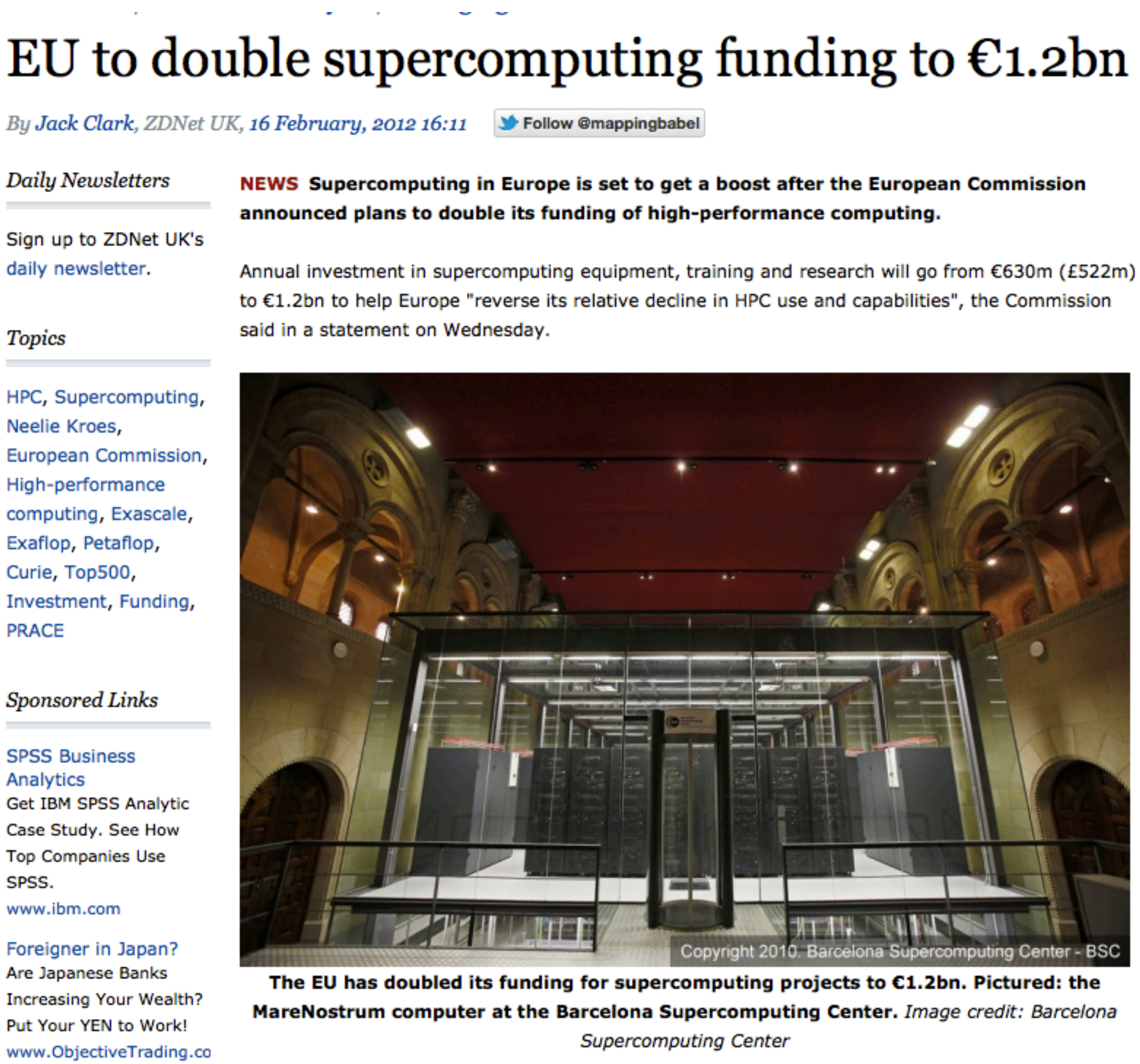
ANR cea CERFACS CRAY THE SUPERCOMPUTER COMPANY EDF FUJITSU INRIA

GENCI NVIDIA RIKEN 東京大学 THE UNIVERSITY OF TOKYO 筑波大学

Published in the International Journal of High Performance Computing Applications

Build an international plan for coordinating research for the next generation open source software for scientific high-performance computing

EU Announced Funding…



EU to double supercomputing funding to €1.2bn

By *Jack Clark*, ZDNet UK, *16 February, 2012 16:11* Follow @mappingbabel

Daily Newsletters

Sign up to ZDNet UK's daily newsletter.

Topics

HPC, Supercomputing, Neelie Kroes, European Commission, High-performance computing, Exascale, Exaflop, Petaflop, Curie, Top500, Investment, Funding, PRACE

NEWS Supercomputing in Europe is set to get a boost after the European Commission announced plans to double its funding of high-performance computing.

Annual investment in supercomputing equipment, training and research will go from €630m (£522m) to €1.2bn to help Europe "reverse its relative decline in HPC use and capabilities", the Commission said in a statement on Wednesday.

Copyright 2010. Barcelona Supercomputing Center - BSC

The EU has doubled its funding for supercomputing projects to €1.2bn. Pictured: the MareNostrum computer at the Barcelona Supercomputing Center. *Image credit: Barcelona Supercomputing Center*

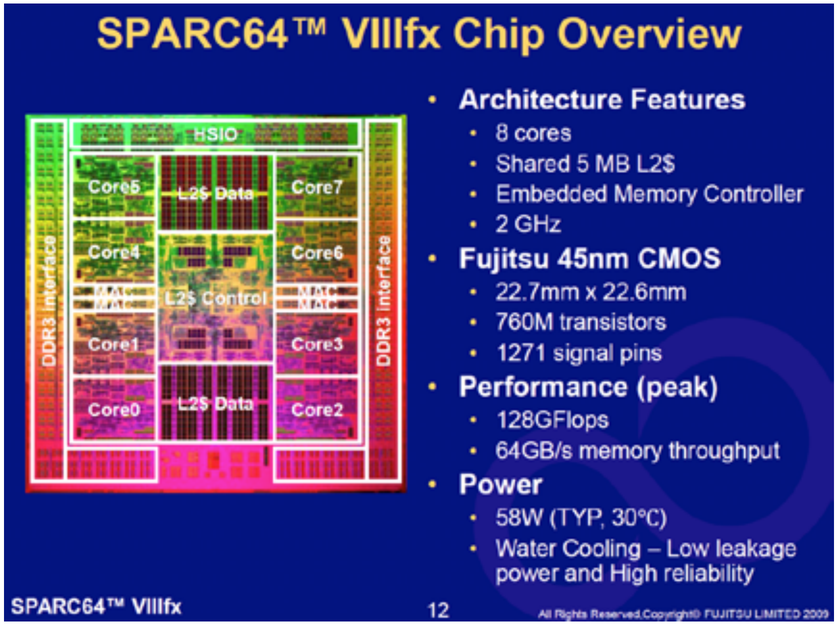
Three Exascale Platform Projects Started in Oct-2011 to Explore European Prototype Architectures

- Goal: jumpstart exascale platforms for Europe
- Joint funding: EC + (some) member states
- Immediate investment modest; $63M total across 3 years ($21M/year)
  - **Mont-Blanc** Project (14.5M€ total)
    - European: ARM (UK), STMicro (France/Italy), BULL (France)
    - + research teams from labs / universities
  - **DEEP** Project (18.5M€ total)
    - EU / US: EXTOLL(German), Intel (US)
    - + research teams from labs / universities
  - **CRESTA** Project (12M€ total)
    - Vampir (German), Cray (UK), Allinea (UK)
    - + research teams from labs / universities
- EESI Plan requests significant, sustained investments in 2 or 3 tracks for 2012
  - 500M€ - 1000M€ over 10 years

Kobe Japan:
Advanced Instituted for Computational Science

Japan: Current #1: The “K” Computer

The heart of the K computer consists of 80,000 Fujitsu’s SPARC64 VIIIfx CPUs



SPARC64™ VIIIfx Chip Overview

HSIO
Core5
L2$ Data
Core7
Core4
Core6
MAC
L2$ Control
MAC
Core1
Core3
Core0
L2$ Data
Core2
DDR3 interface
DDR3 interface

- **Architecture Features**
  - 8 cores
  - Shared 5 MB L2$
  - Embedded Memory Controller
  - 2 GHz
- **Fujitsu 45nm CMOS**
  - 22.7mm x 22.6mm
  - 760M transistors
  - 1271 signal pins
- **Performance (peak)**
  - 128GFlops
  - 64GB/s memory throughput
- **Power**
  - 58W (TYP, 30°C)
  - Water Cooling – Low leakage power and High reliability

SPARC64™ VIIIfx 12 All Rights Reserved,Copyright© FUJITSU LIMITED 2009

864 Cabinets
10PFlops
1PB

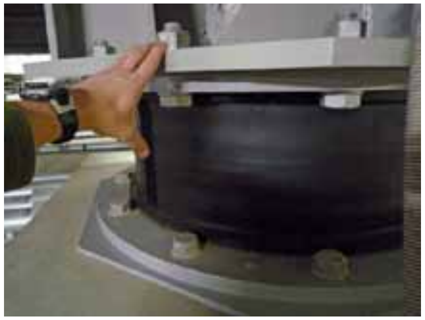
24 Boards /
Cabinet

Fujitsu SPARC64™IXfx

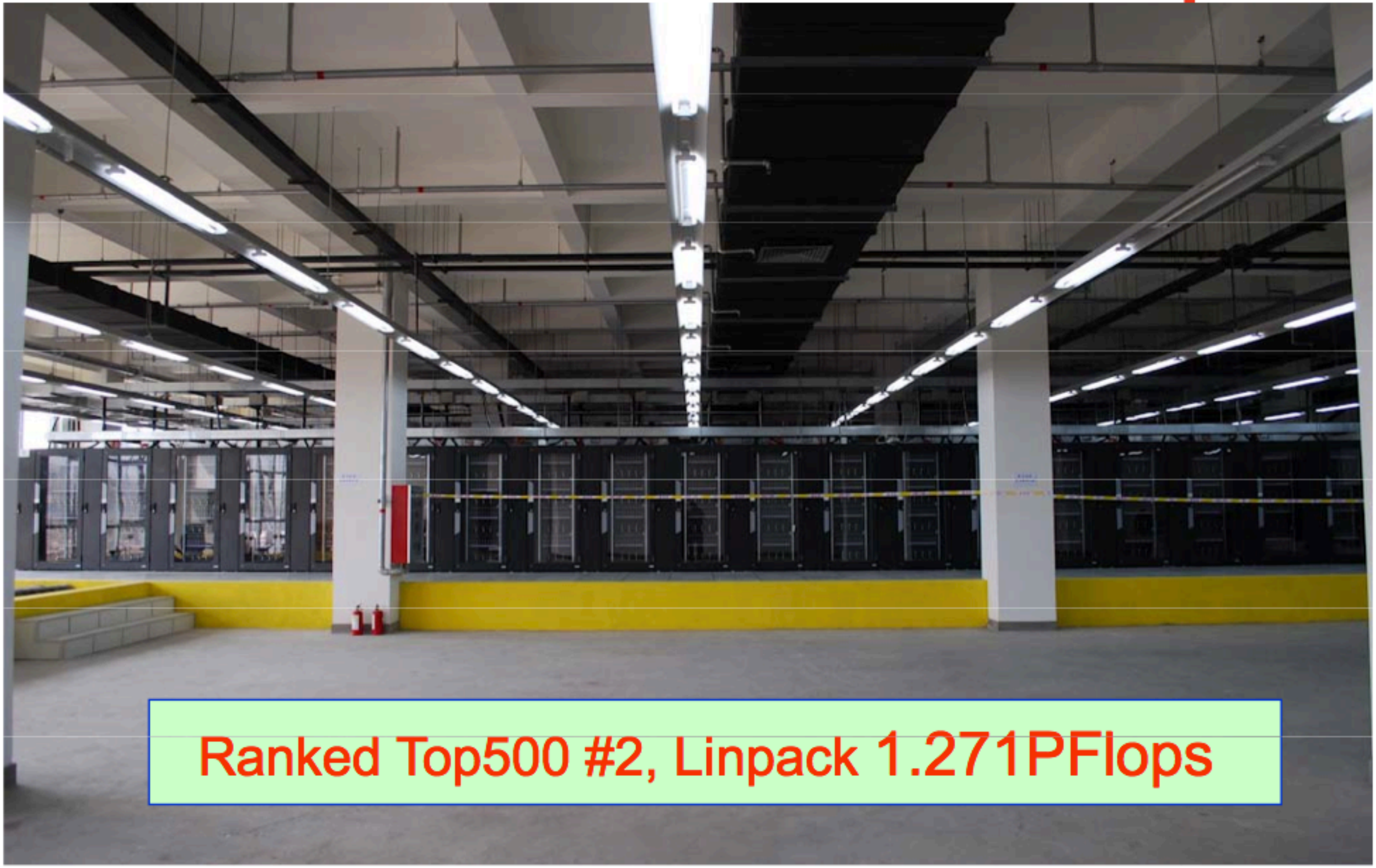
Sept 2011: New chip announced

An amazing accomplishment, with unique and advanced system software

京
K computer
京
K computer
FUJITSU

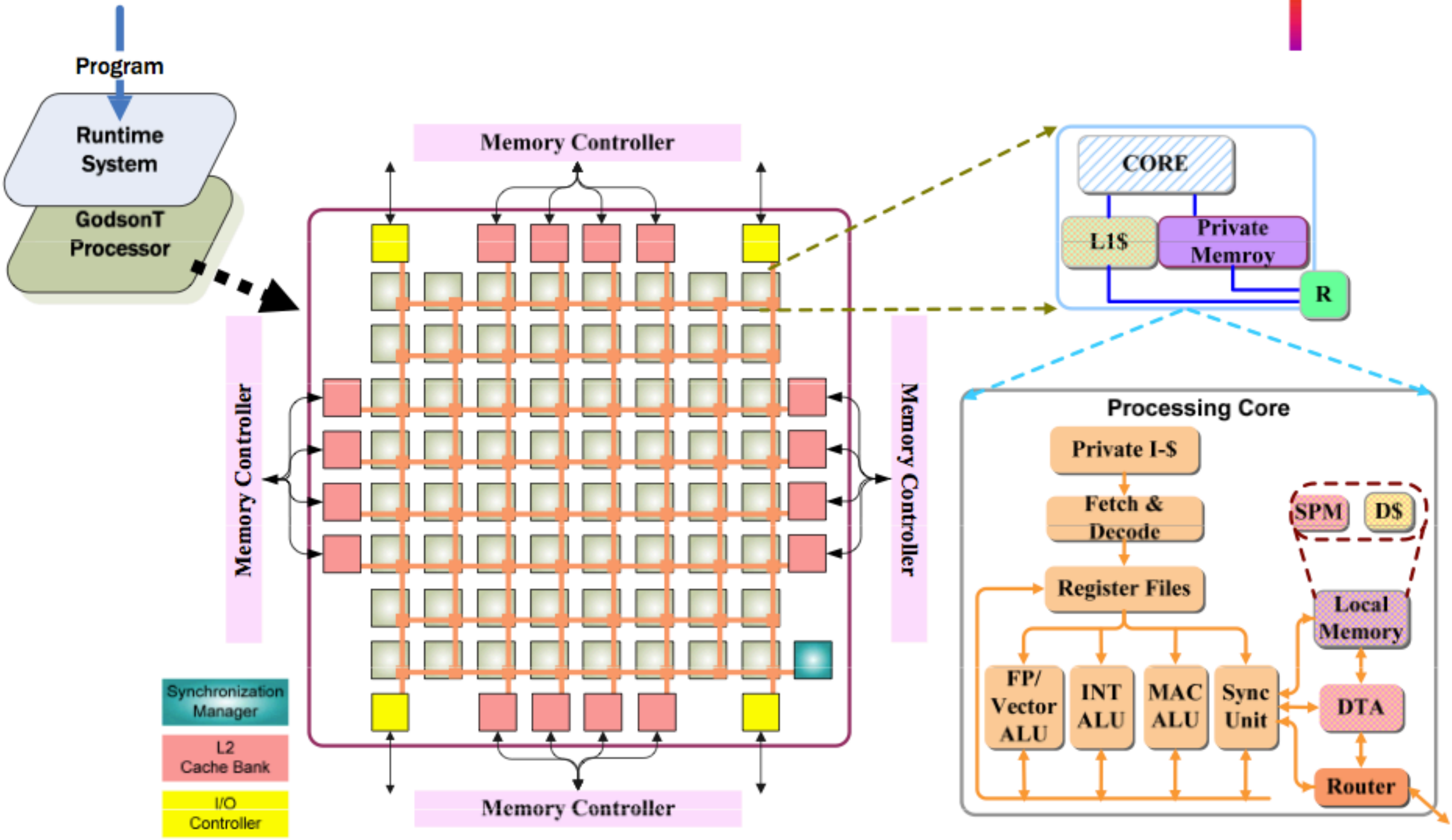


Dawning Nebulae: 3PFlops (2010)



Ranked Top500 #2, Linpack 1.271PFlops

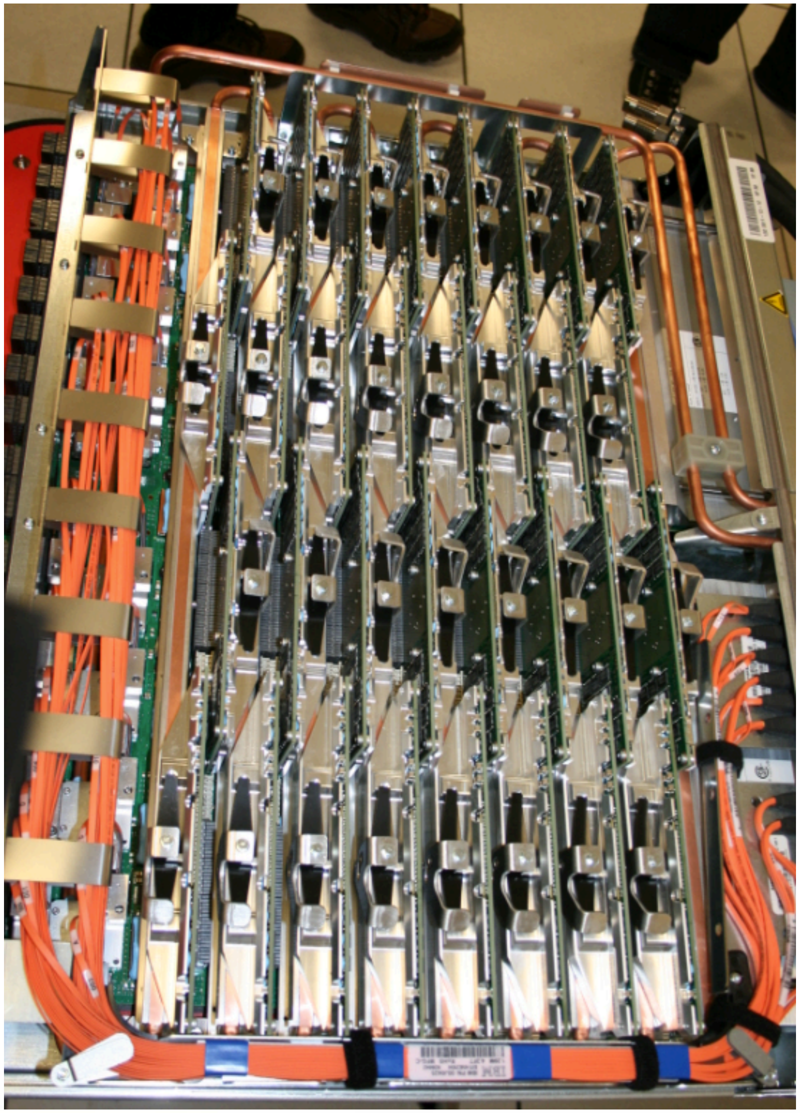
Architecture Overview of Godson-T



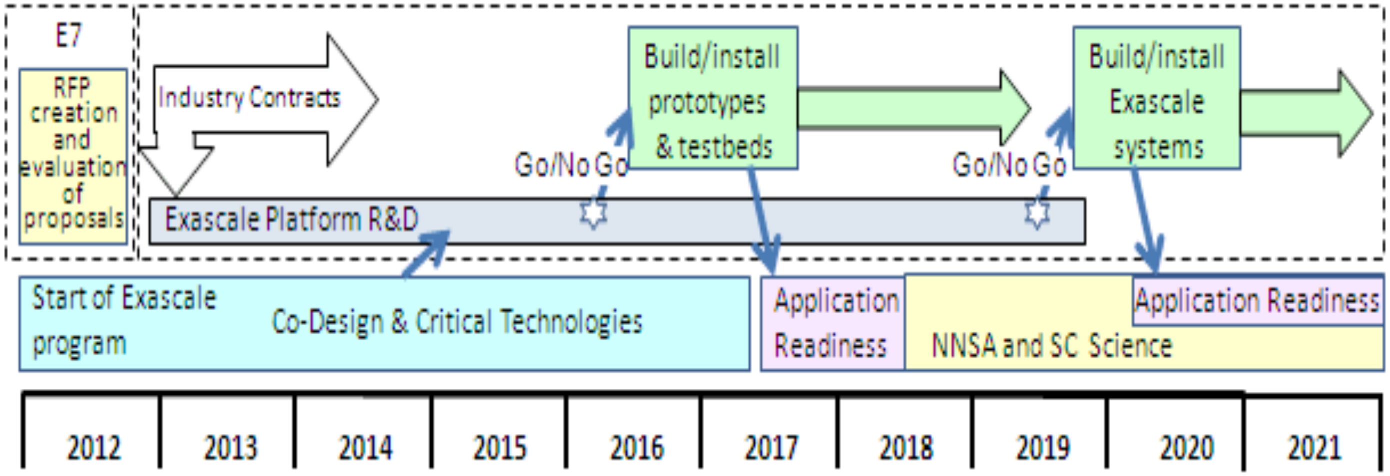
New at Argonne: BLUE GENE/Q

- *Mira* - Blue Gene/Q System
  - 48 racks
  - 48K 1.6 GHz nodes
  - 768K cores & 786TB RAM
  - 384 I/O nodes
  - Peak: 10PF
- Storage
  - ~35 PB capacity, 240GB/s bandwidth (GPFS)
  - Disk storage upgrade planned in 2015
    - Double capacity and bandwidth
- New Visualization Systems
  - Initial system in 2012
  - Advanced visualization system in 2014
    - State-of-the-art server cluster with latest GPU accelerators
    - Provisioned with the best available parallel analysis and visualization software

BG/Q installed and running!
A GREEN Solution: Co-Designed with IBM



USA: Exascale RFI: Deep NDAs with Companies to Explore Computing Technology for 2020



E7

RFP creation and evaluation of proposals

Industry Contracts

Build/install prototypes & testbeds

Build/install Exascale systems

Go/No Go

Go/No Go

Exascale Platform R&D

Start of Exascale program

Co-Design & Critical Technologies

Application Readiness

Application Readiness

NNSA and SC Science

2012 2013 2014 2015 2016 2017 2018 2019 2020 2021

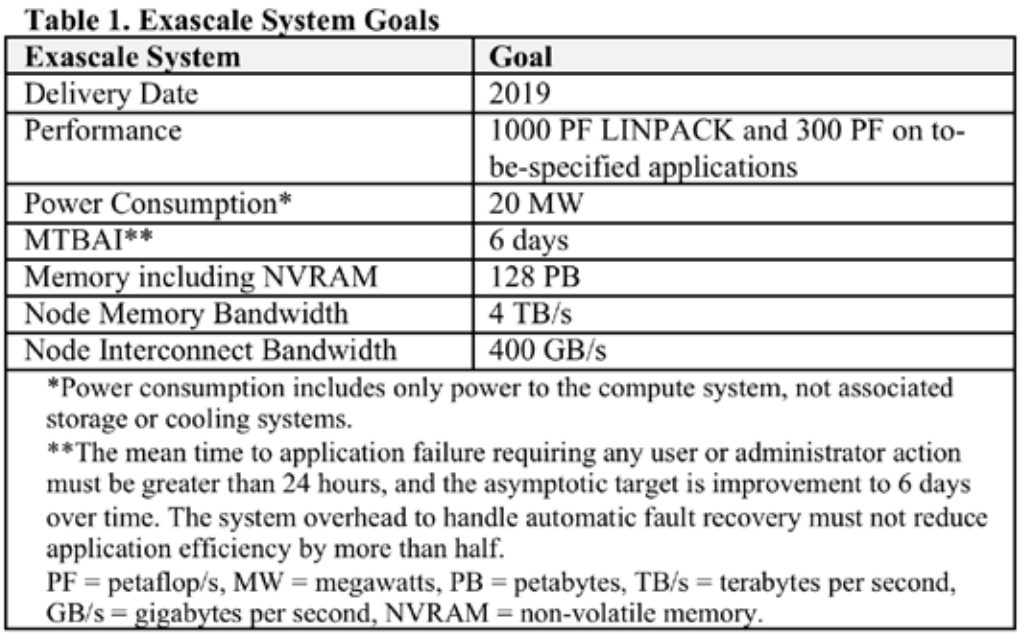


Table 1. Exascale System Goals

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.

What Did We Learn?
Maybe the Obvious… CPUs are Changing…

- **Parallelism** within a node is dramatically increasing
  - System software will change
- **Dynamic power management** is critical to performance
  - System software will change
- **Distributed memory**: cache coherence not power efficient
  - System software will change
- **Deep memory hierarchies**: 3D local RAM and NVRAM
  - System software will change
- **Faults** may increase
  - System software will change

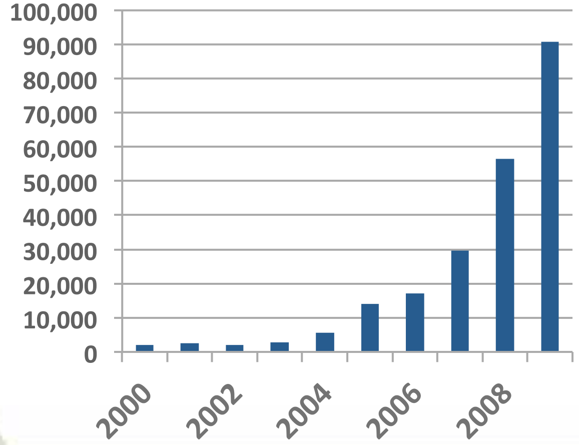
Phones lead, desktops follow?

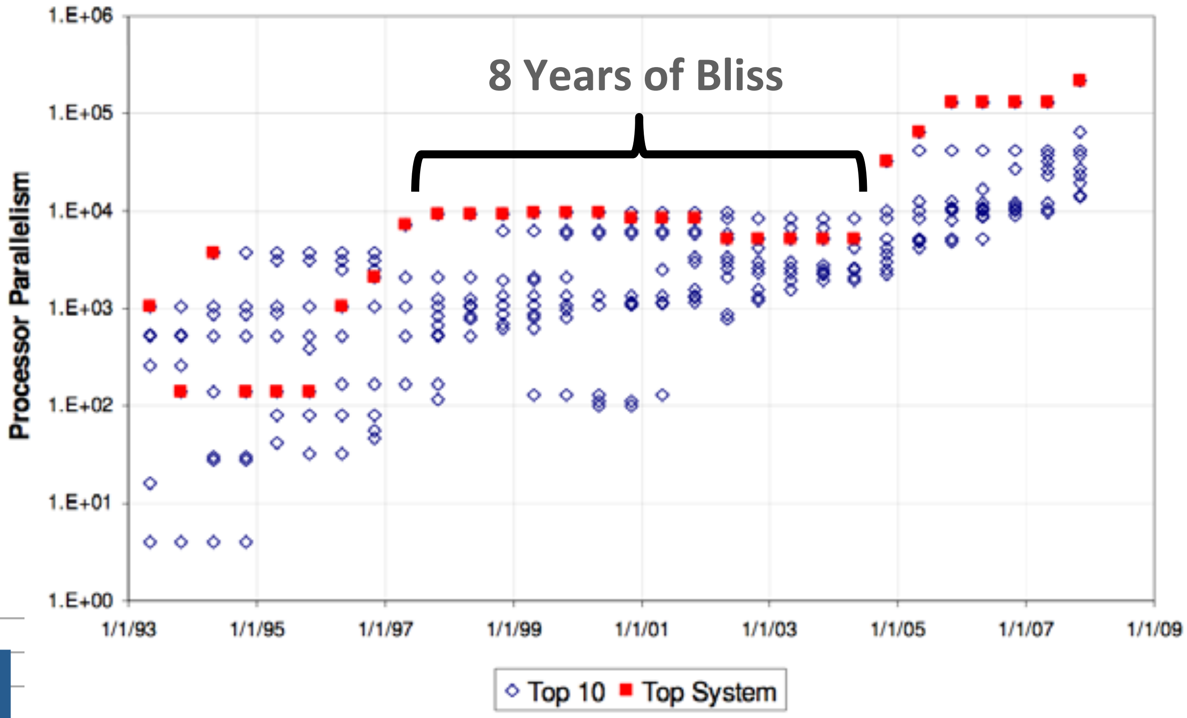
Parallelism

Parallelism Has Suddenly Exploded
“The core is the new transistor” (new Moore’s law)

Raspberry Pi: $25

- 700MHz ARM11
- $25



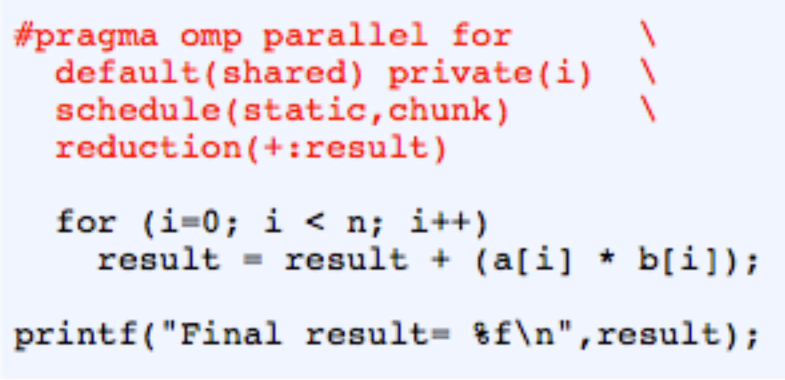


8 Years of Bliss

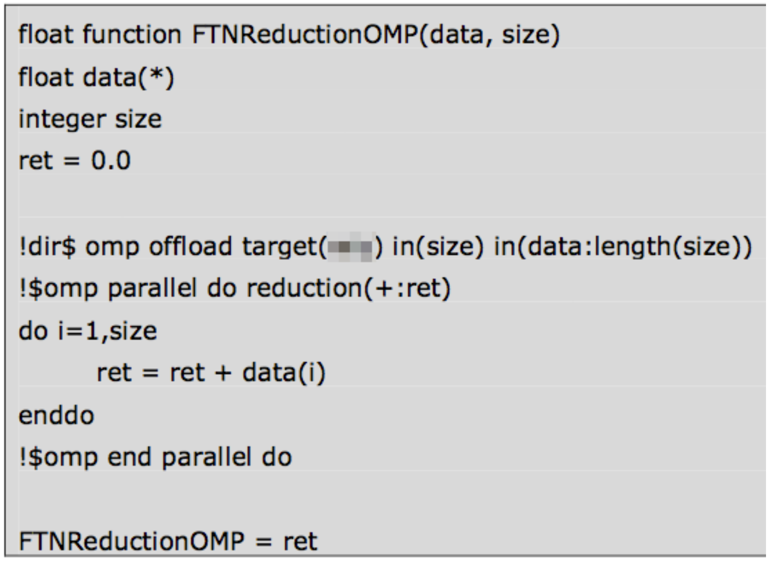
Source: DARPA Exascale Report

With Intranode Parallelism Exploding, How Do We Write Programs?

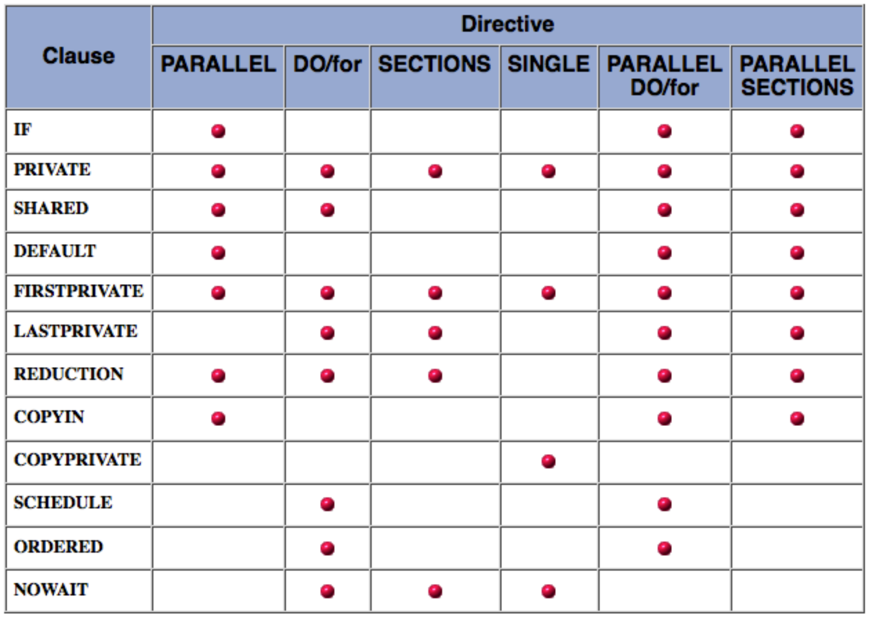
In-Socket Parallel Programming is a Mess:



```
#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);
```



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

!dir$ omp offload target([redacted]) 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
```

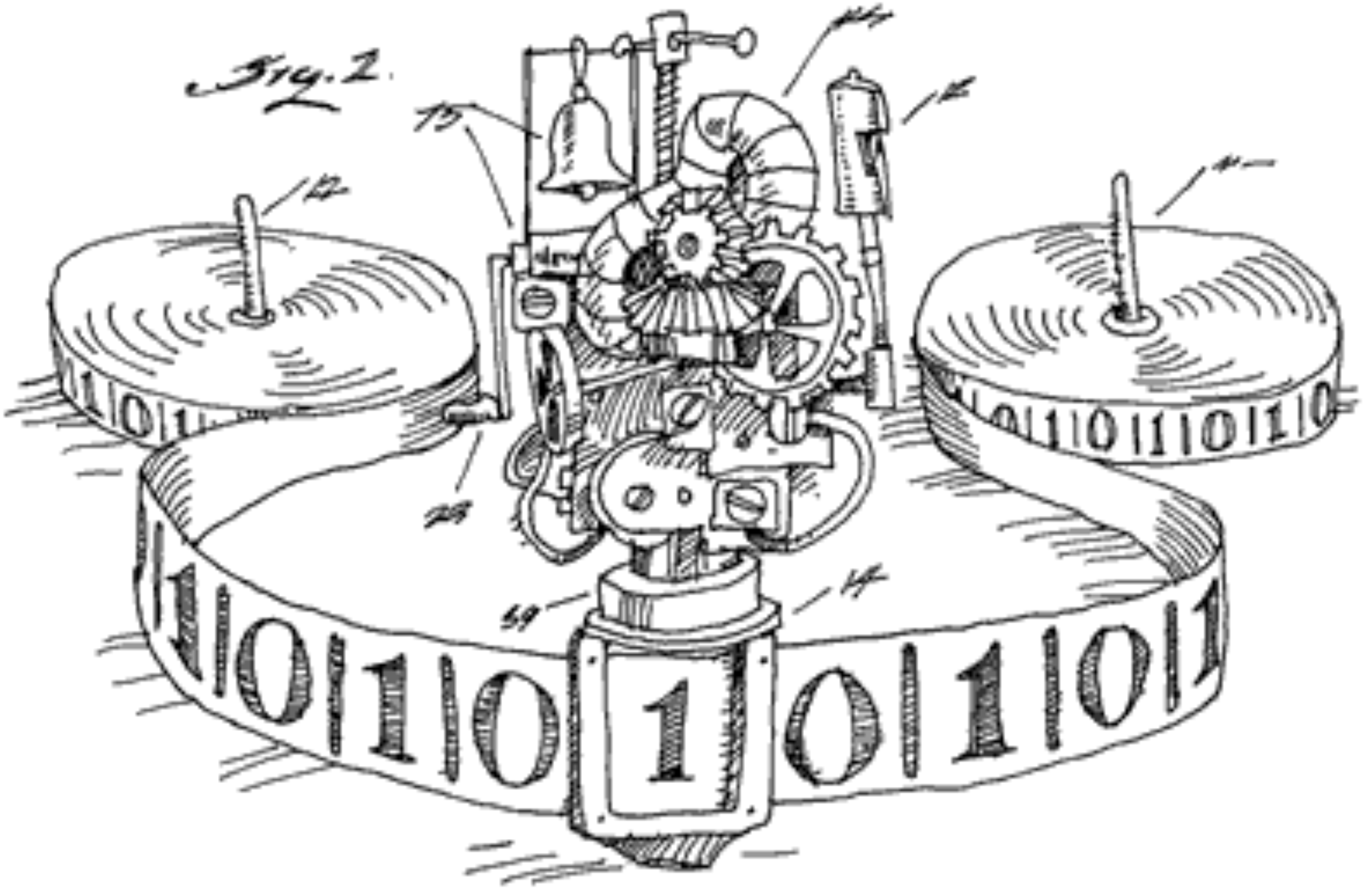


Clause	Directive					
	PARALLEL	DO/for	SECTIONS	SINGLE	PARALLEL DO/for	PARALLEL SECTIONS
IF	●				●	●
PRIVATE	●	●	●	●	●	●
SHARED	●	●			●	●
DEFAULT	●				●	●
FIRSTPRIVATE	●	●	●	●	●	●
LASTPRIVATE		●	●		●	●
REDUCTION	●	●	●		●	●
COPYIN	●				●	●
COPYPRIVATE				●		
SCHEDULE		●			●	
ORDERED		●			●	
NOWAIT		●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

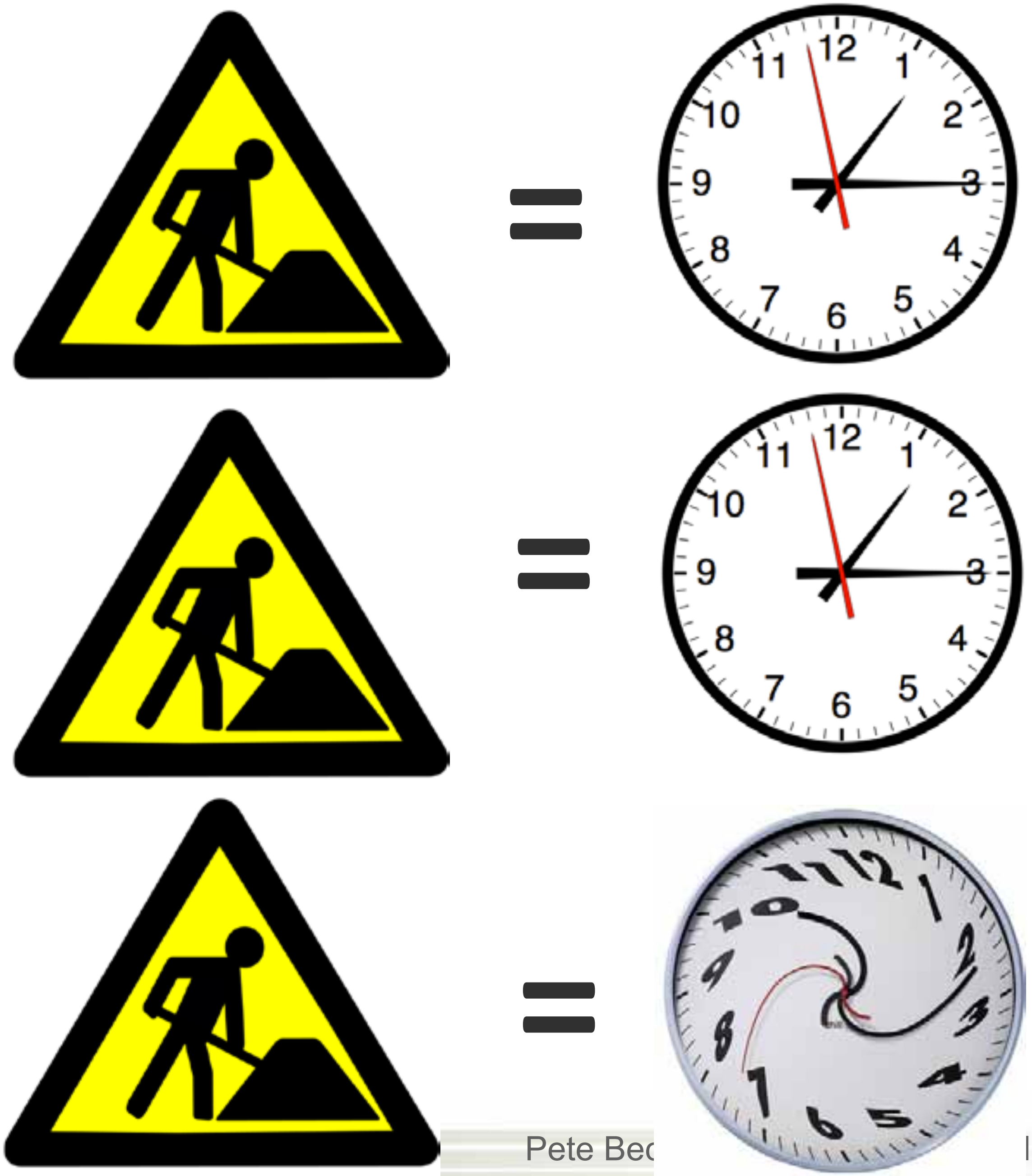
System Software Challenges:

- We do not yet have a good in-socket parallel programming model
- New Programming Models & Languages Needed (OpenMP is a mess)
- Memory mgmt for deeper hierarchies (3D scratchpad, cache, memory)
- OS that controls threads, tasks, and power
- How do we represent heterogeneous HW?

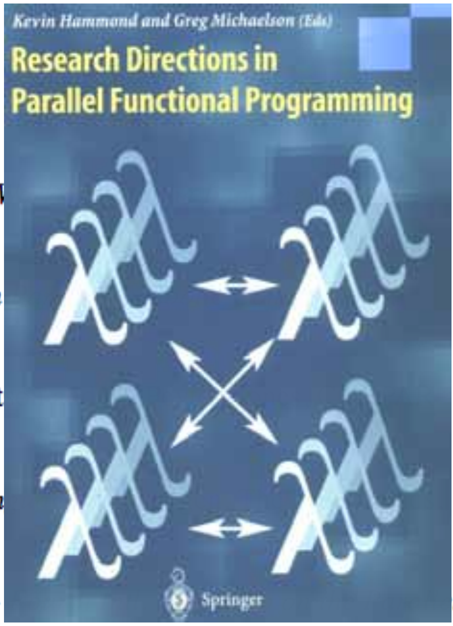
Rethinking the sequential abstract machine….



Rethinking the parallel abstract machine….



Returning to our Roots: Graphs

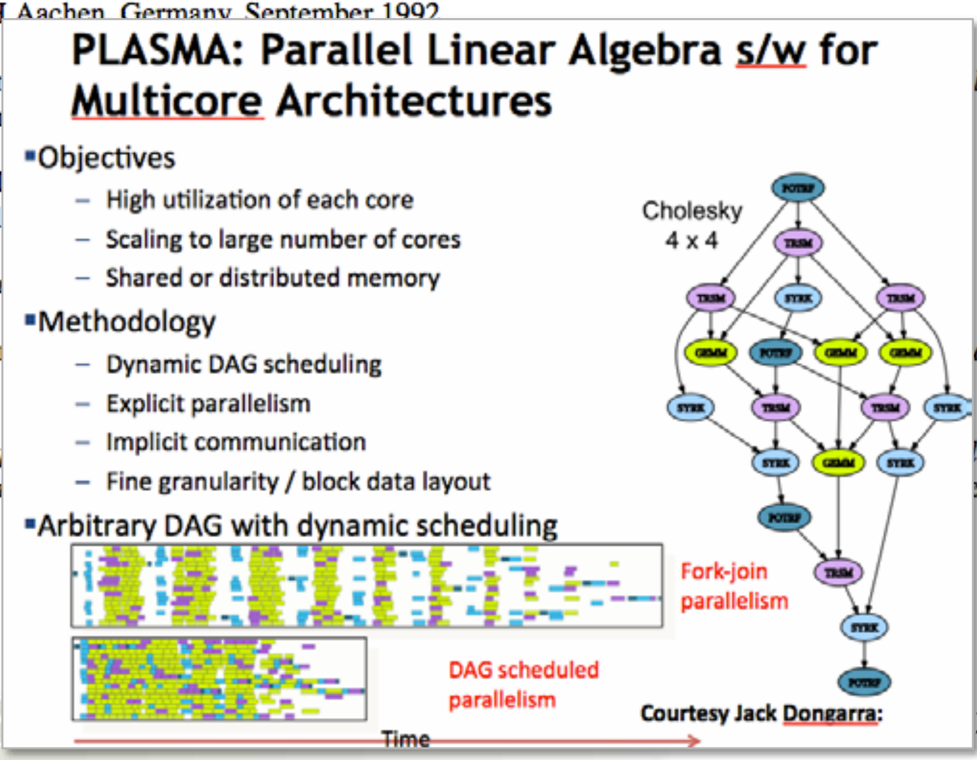

Kevin Hammond and Greg Michaelson (Eds)

Research Directions in Parallel Functional Programming

Springer

1999

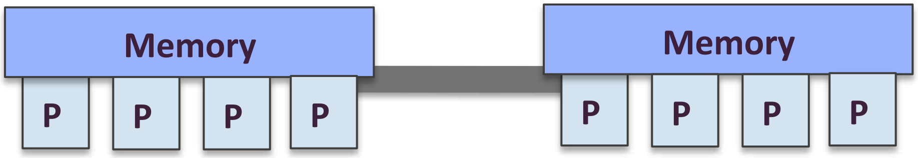
- *GRAPH for PVM: Graph Reduction for Distributed Hardware* , H-W Loidl, K Hammond, In *IFL'94 --- Intl. W*
  Functional Languages , Norwich, England, Sep. 7--9, 1994.
- *Parallel Functional Programming: An Introduction* , K Hammond, In *PASCO'94 --- First Intl. Symposium on*
  Hagenberg/Linz, Austria, 26-28 September. World Scientific.
- *Automatic spark strategies and granularity for a parallel functional-language reducer*, K Hammond, JS Matt
  1994, Linz, Austria, Springer LNCS 854, Sept 1994, pp521-532.
- *Getting a GRIP*, K Hammond, In *IFL'93 --- Intl. Workshop on the Parallel Implementation of Functional Lan*
  September 1993.
- Profiling Scheduling Strategies on the GRIP Multiprocessor, K Hammond and SL Peyton Jones, In *IFL'92 ---* *l*
  Implementation of Functional Languages, pp. 73-98, RWTH Aachen, Germany, September 1992.
- A Parallel Functional Database for GRIP, G Akerholt, K Har *l*
  Implementation of Functional Languages, pp. 7-30, Southan
- Some Early Experiments on the GRIP Parallel Reducer, K H
  Implementation of Functional Languages, pp. 51-72, June, 1
- *Parallel Implementations of Functional Programming Langu*
- *High-Performance Parallel Graph Reduction*, SL Peyton Jo *rope*,
  pp. 193-206. LNCS 365, Springer Verlag 1989.
- *GRIP --- a High-Performance Architecture for Parallel Gra* *l. Conf.*
  on Functional Programming Languages and Computer Arch rlag
  1987.


PLASMA: Parallel Linear Algebra s/w for Multicore Architectures

- Objectives
  - High utilization of each core
  - Scaling to large number of cores
  - Shared or distributed memory
- Methodology
  - Dynamic DAG scheduling
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout
- Arbitrary DAG with dynamic scheduling

Courtesy Jack Dongarra:

Reinventing Programming Models?

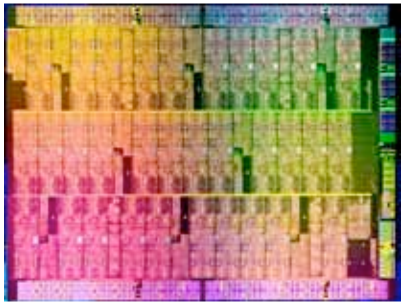


- In this new world, we must reinvent our abstract machine
  - Programmers have focused on “cores”, dividing work across cores
- We can’t program to an exponentially changing component... (num cores)
  - Only trees handle exponentially growing resources...
- We must return to higher-level models
  - Coherence domains, sea of ALUs
- Programming model cannot be based on parallelism after the fact (openMP)
  - Charm++, CILK? Concurrent Collections? Functional Programming?
- System Software Challenge:
  - Explore new abstract machine and programming languages, and run-time systems

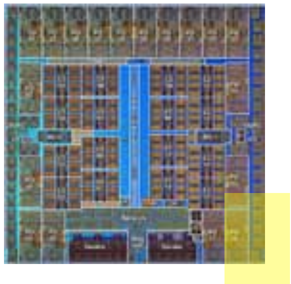
Intranode Power Constraints and Cache Coherence

Within the Node, What Else is Changing?

How Will System Software Manage CPUs?
How Will They Be Programmed?



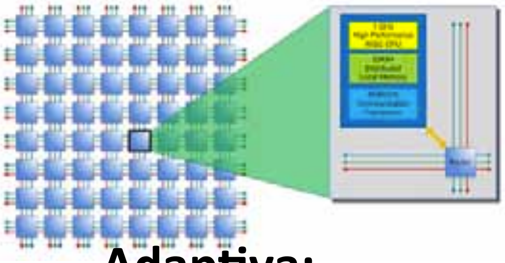
Intel: Knight’s Ferry



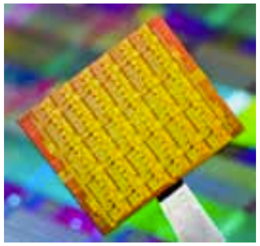
#18

IBM: BG/Q

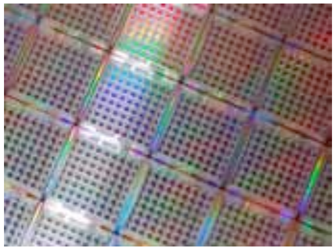
Power Constrained Memory Consistency



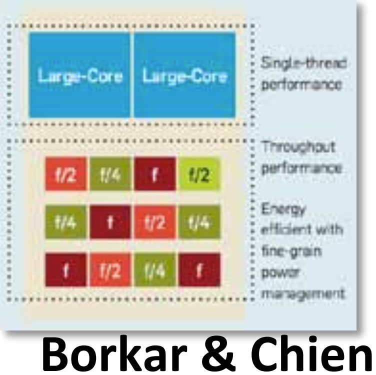
Adaptiva:



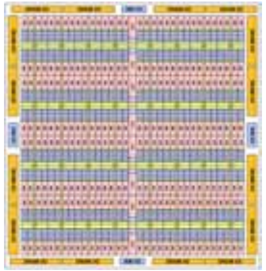
Intel: SCC



Tilera: GX


Large-Core
Large-Core
Single-thread performance
f/2
f/4
f
f/2
f/4
f
f/2
f/4
f
f/2
f/4
f
Throughout performance
Energy efficient with fine-grain power management

Borkar & Chien



Dally: Echelon

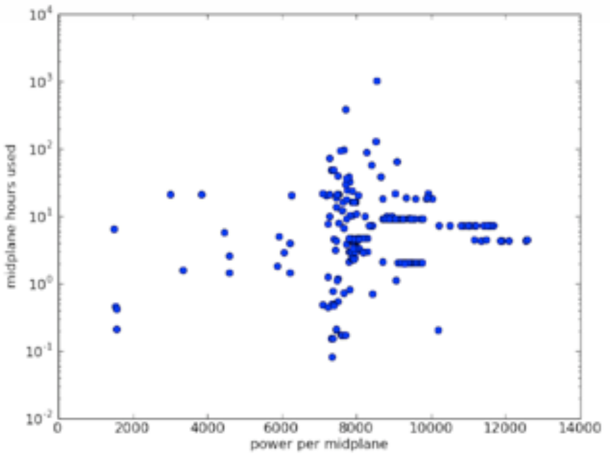
?

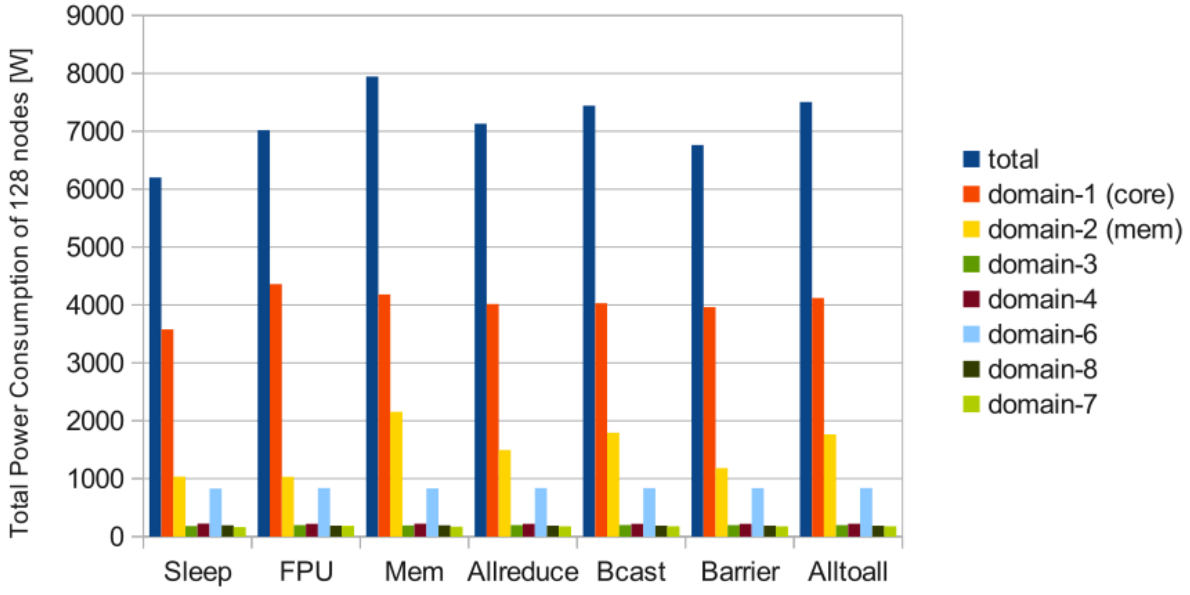
Power, Parallelism, Coherence, Fault, Storage

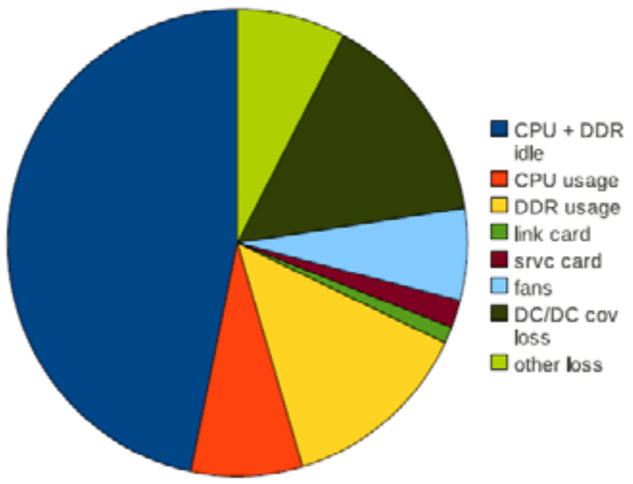
System Software Challenges:

- Power must be a managed resource
  - Dark Silicon: More functional units than can run at full speed
  - Variable speed subcomponents
  - New: Optimize perf for Thermal Design Point (TDP)
- Restructured node architecture
  - Massive levels of in-package parallelism
  - Variable coherence domains and intrasocket messaging
  - Heterogeneous multi-core (graphics, compression, etc)
    - Programming model for this?
- Complex fault behavior
  - Single core could experience fault
  - Need for fault domains

BG/P & BG/Q Power Experiments



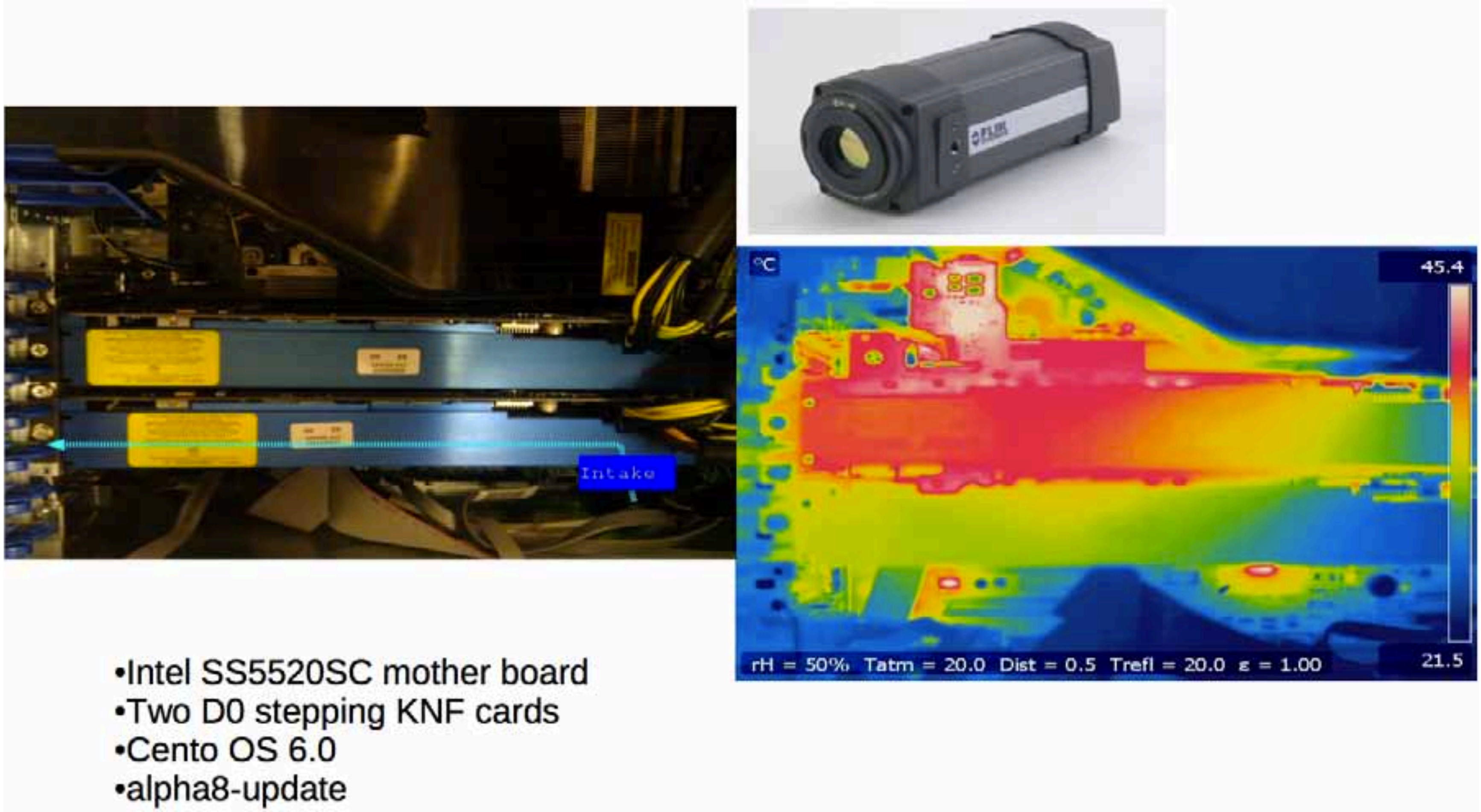




Comparison between CNK and Linux on sleep()

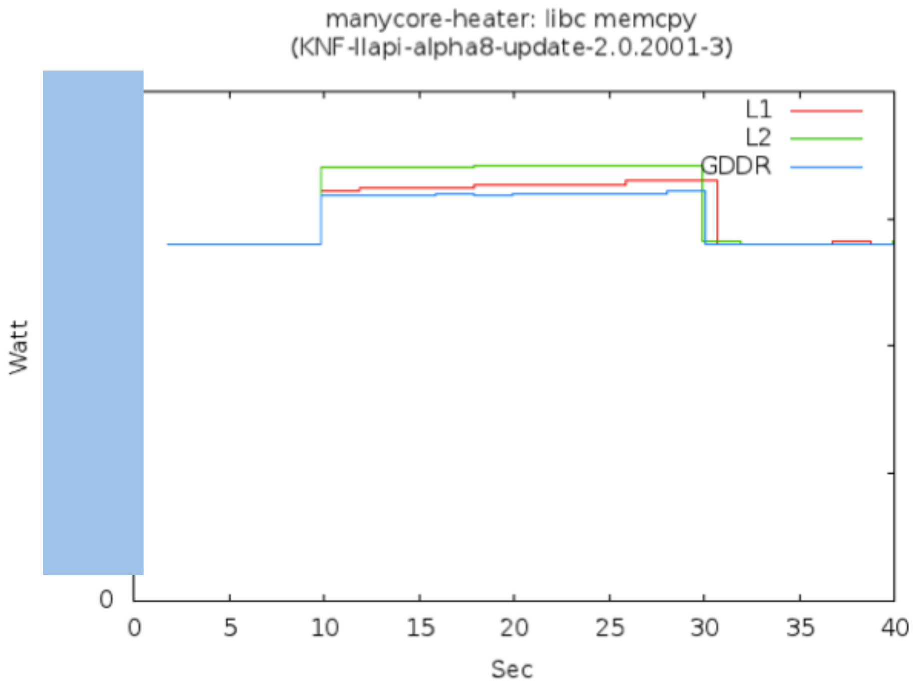
	CNK	Linux	%
KWatt	14.935	13.809	7.75

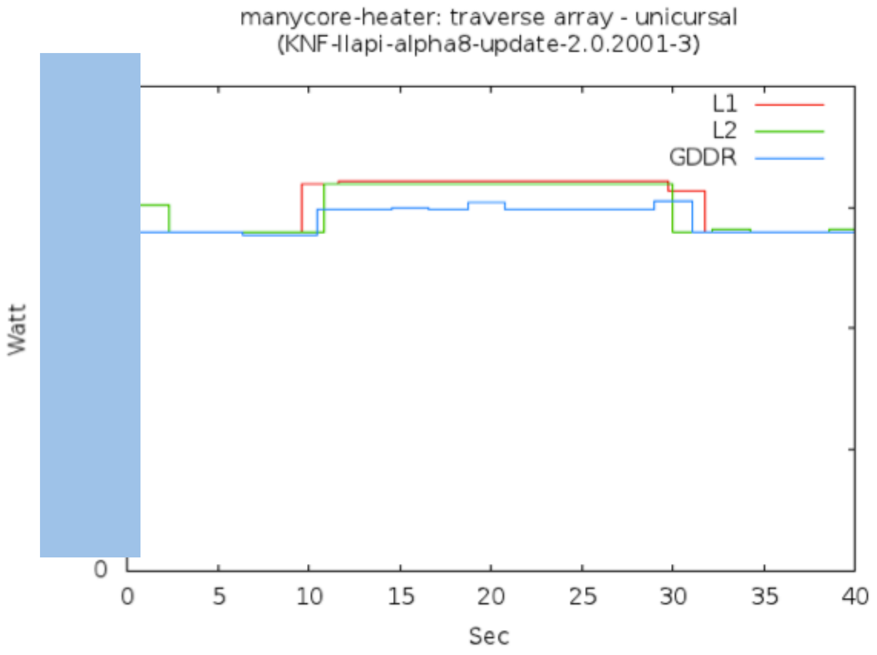
Exploring Power on Intel Knights Ferry



- •Intel SS5520SC mother board
- •Two D0 stepping KNF cards
- •Cento OS 6.0
- •alpha8-update

Seeking to Isolate Components





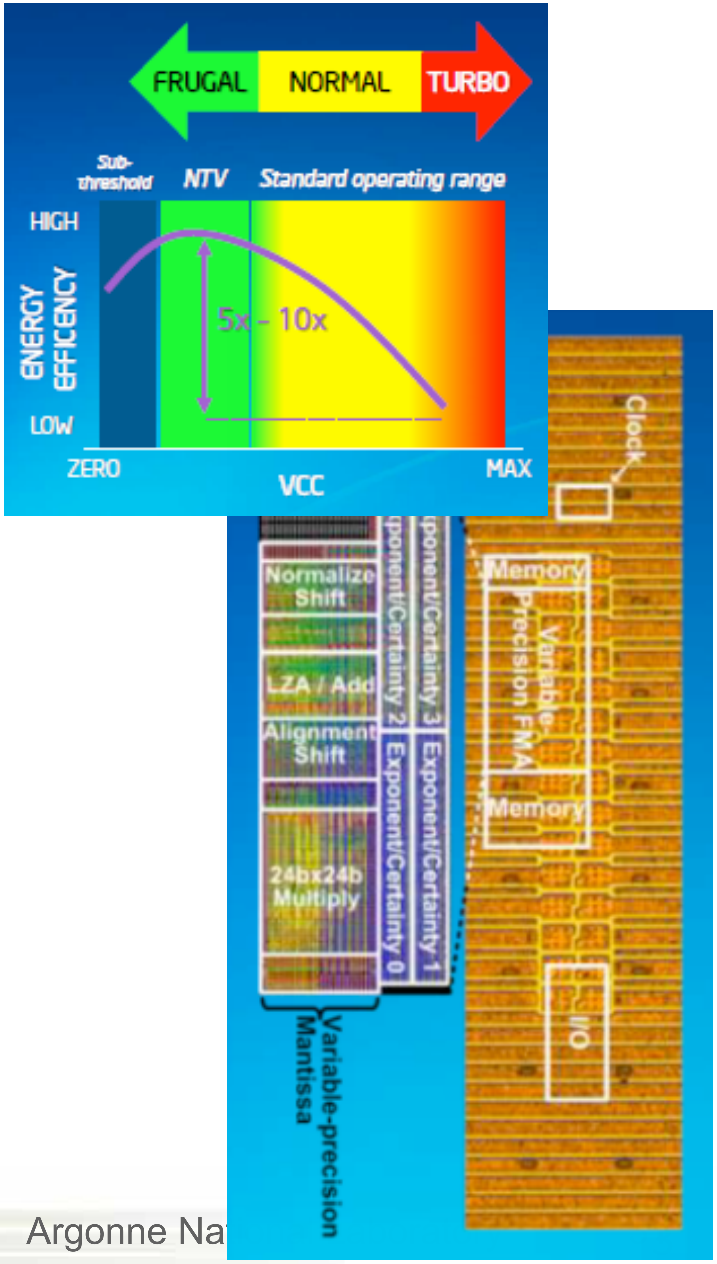
- Future manycore chips will permit many power modes and speeds per core
- System software needed to manage power
- Goal:
  - Create abstract machine model for power use (compiler, runtime, etc)
  - Create dynamic power-aware run-time system

Near Future Technologies

Intel: NTV (Near Threshold Voltage) circuits & variable precision floating point

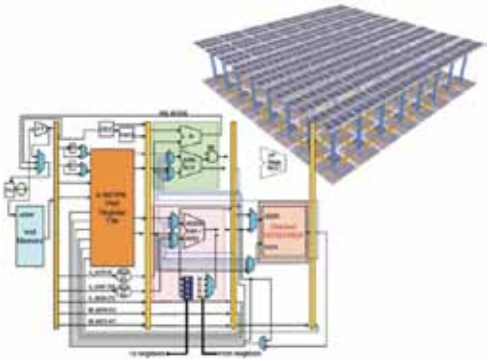
- In NTV range, 5 to 10 times more efficient
- Demonstrated chip that can go from 3Mhz to 915Mhz
- Prototype variable precision floating point system

- **System Software Challenges:**
  - **Manage speed over range of O(100)**
  - **Control precision based on required error bounds**

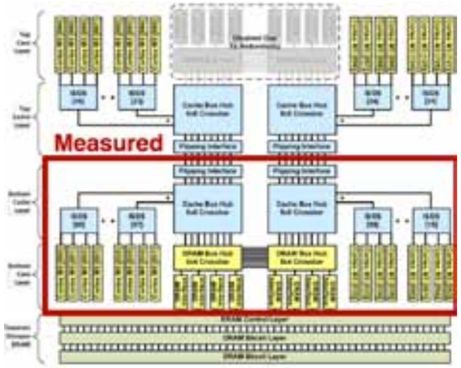


3D Chip Stacking:
Really Fast, Really Close, Really Small

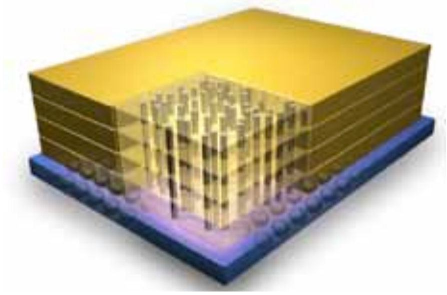
- On-chip RAM getting smaller WRT parallelism
- Bandwidth will be excellent
- Advanced memory operations possible
- Integrated NIC is the next step
- Explicit data movement within chip
- **System Software Challenges**
  - **Memory management, data movement**
  - **OS that controls threads, tasks, and power**



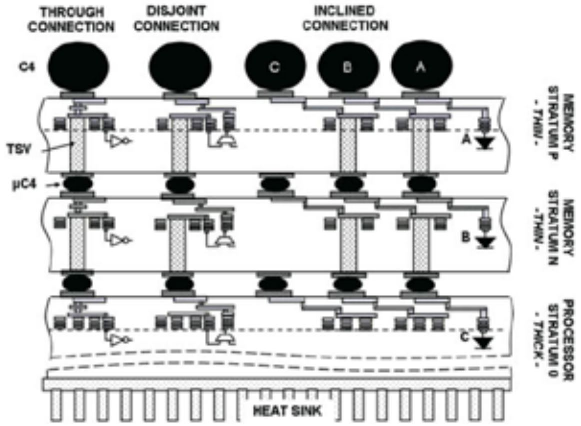
Georgia Tech



Univ of Michigan



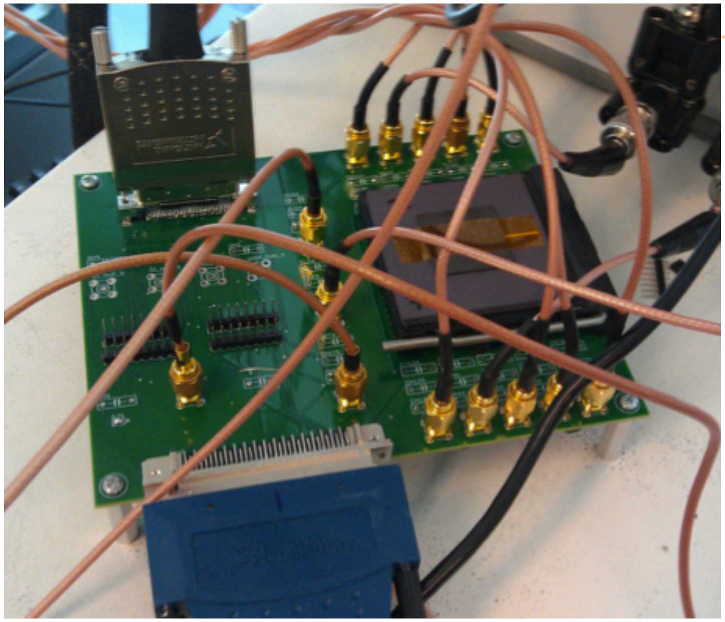
Micron HMC


THROUGH CONNECTION
DISJOINT CONNECTION
INCLINED CONNECTION
C4
C
B
A
TSV
μC4
MEMORY STRATUM P - THIN -
MEMORY STRATUM N - THIN -
PROCESSOR STRATUM 0 - THICK -
HEAT SINK

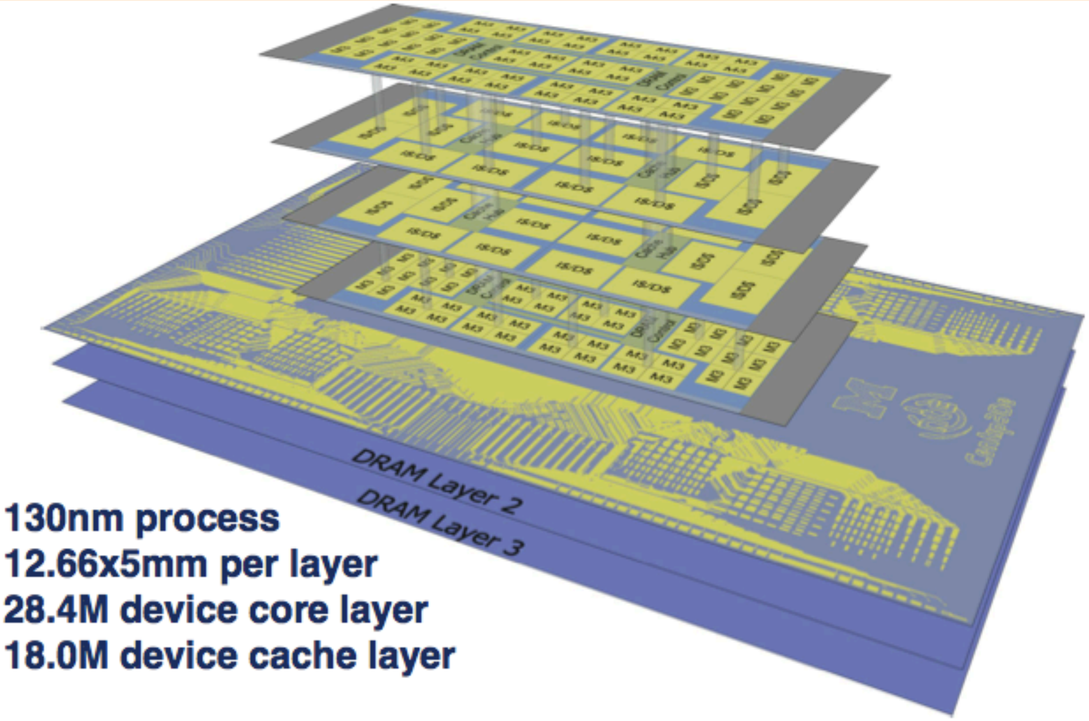
IBM

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

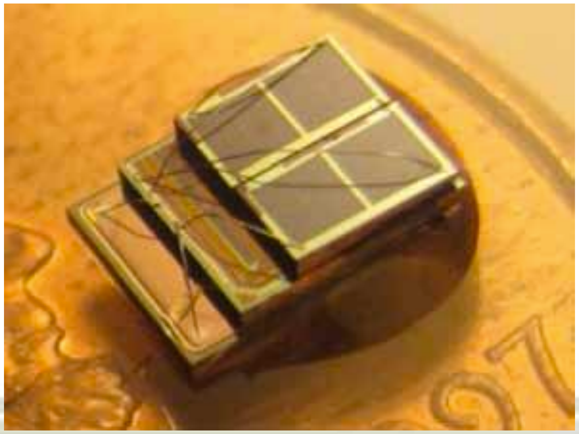
University of Michigan



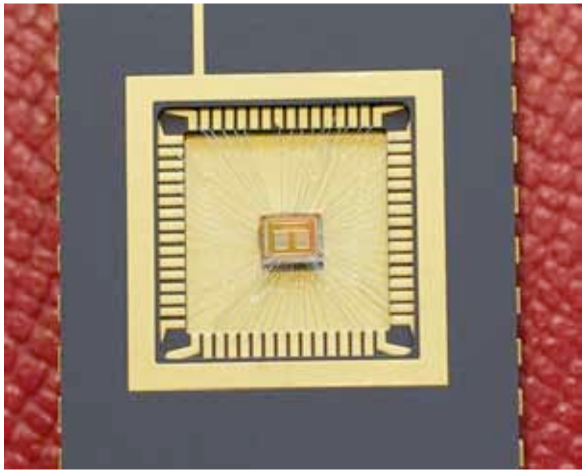
Centip3De System Overview



130nm process
12.66x5mm per layer
28.4M device core layer
18.0M device cache layer



Non-Volatile RAM May Replace Everything…

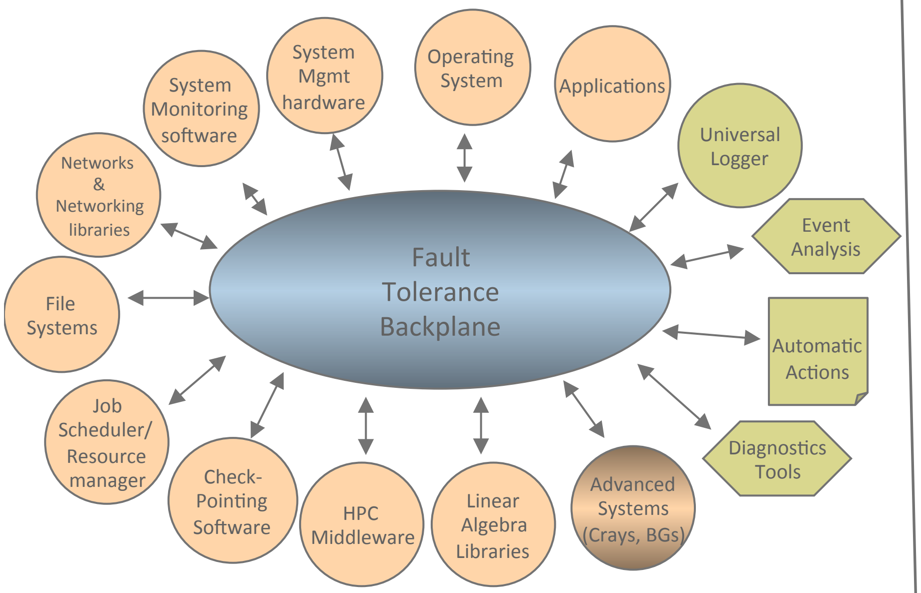


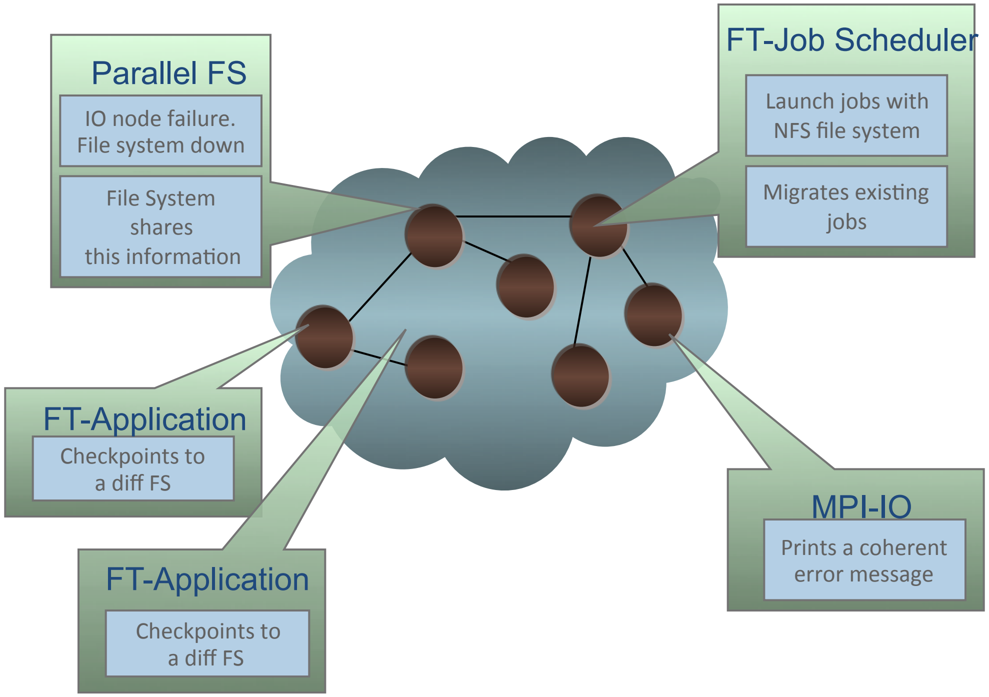
- Phase Change Memory (PCM) already being produced for some markets
  - More power to ***write*** data than to ***read*** data
- Memresistor is probably further from deployment
- Both may be deployed either in 3D or as a burst buffer near chips
- System Software Challenges
  - New algorithms that read more than they store
  - Management of the deeper hierarchy
    - Memory pages? Block I/O?

Fault

Intranode and Internode

CIFTS project:
Exploring a Fault Tolerance Backplane




Parallel FS
IO node failure. File system down
File System shares this information
FT-Job Scheduler
Launch jobs with NFS file system
Migrates existing jobs
FT-Application
Checkpoints to a diff FS
MPI-IO
Prints a coherent error message
FT-Application
Checkpoints to a diff FS

The **Fault Tolerance Backplane** provides

- A scalable framework to exchange fault-related information
- Exposes a standard interface that can be used by any software to connect to the FTB
- Provides a common uniform event handling mechanism and event notification mechanism

Exciting Times

- **Parallelism** within a node is dramatically increasing
  - System software will change
- **Dynamic power management** is critical to performance
  - System software will change
- **Distributed memory**: cache coherence not power efficient
  - System software will change
- **Deep memory hierarchies**: 3D local RAM and NVRAM
  - System software will change
- **Faults** may increase
  - System software will change

Phones lead, desktops follow?

What Does This Mean for Computer Science? (and System Software)

- Parallelism: Sequential code is obsolete. Crazy amounts of parallelism
  - SIMD, Vector, MIMD, etc
  - We must revisit programming models, languages, invent new ways to express parallelism
  - Advanced run-time systems to manage tasks and dependencies
- Dynamic power management: first class object in system software
  - Performance is limited by Thermal Design Point (TDP)
  - New algorithms to improve performance within TDP… New analysis techniques
  - Power (speed) and dark silicon must be explicitly managed by system software
- Distributed memory: intranode programming must access remote data
  - Combined with parallelism, programming model will manage data movement
- Deep memory hierarchies: 3D RAM, NVRAM on node
  - I/O Forwarding inside the node
  - New models for deep memory hierarchy
- Faults: Distributed computing arrives within the node
- Variable precision floating point: new numerical analysis and library designs
  - Quantify precision and uncertainty
  - Library interfaces to specify precision
  - Hybrid algorithms based on precision, speed, power