

Energy-Aware Algorithms at Exascale

Padma Raghavan

Computer Science and Engineering
Institute for CyberScience

Research Supported by NSF, DoD and IBM

Path To Exascale, November 17, 2008



Outline

- I. Exascale architectures
 - Impacts on algorithm & s/w design
- II. Energy-Aware Scaling
 - Examples of algorithm & s/w redesigns
- Summary

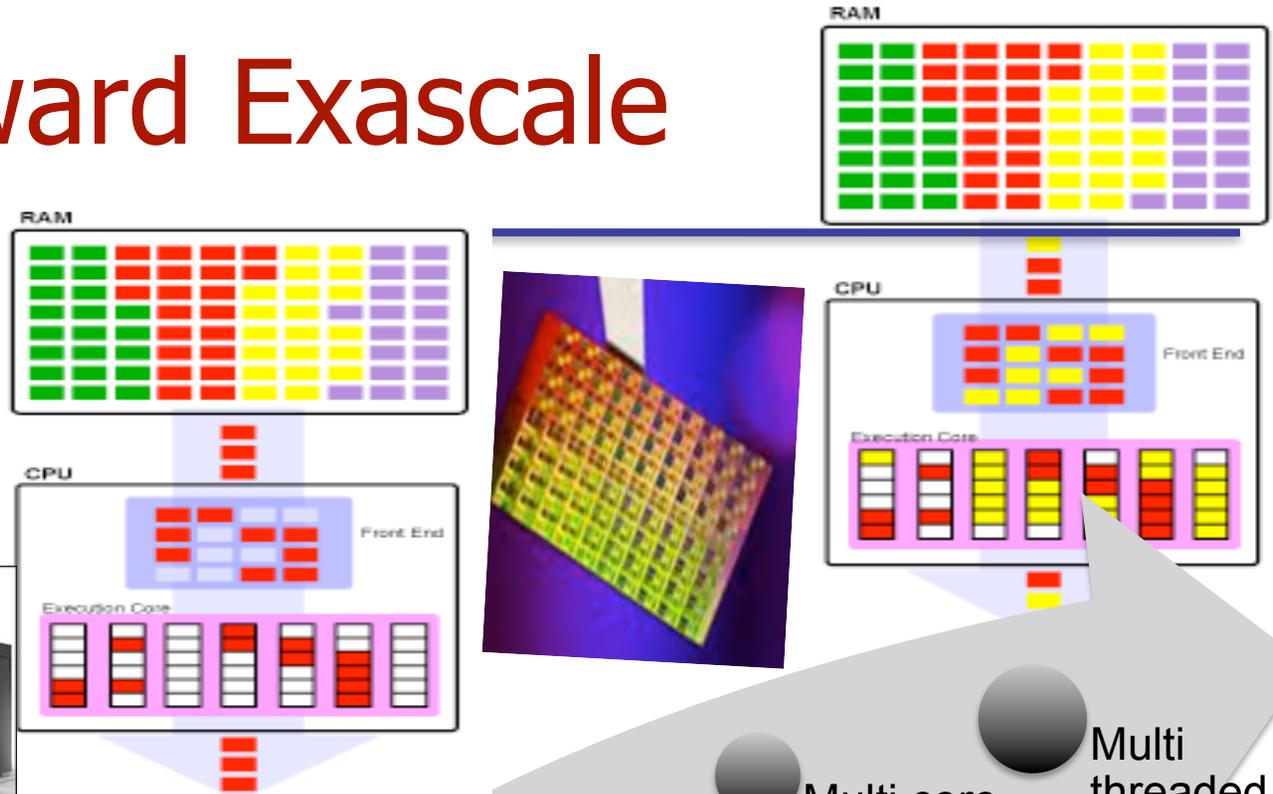
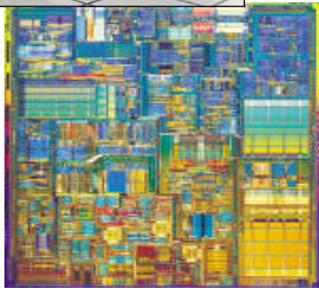
I: Exascale Architectures

- Then, now and beyond
 - From **fast, hot** ...
 - To **parallel, cooler**
 - To **billion-way parallel, heterogeneous, unreliable**



Toward Exascale

Future systems will consist of millions of nodes



Single node
P = 1

Multi node
x 1,000,000
150MW

ILP
x 4
150MW

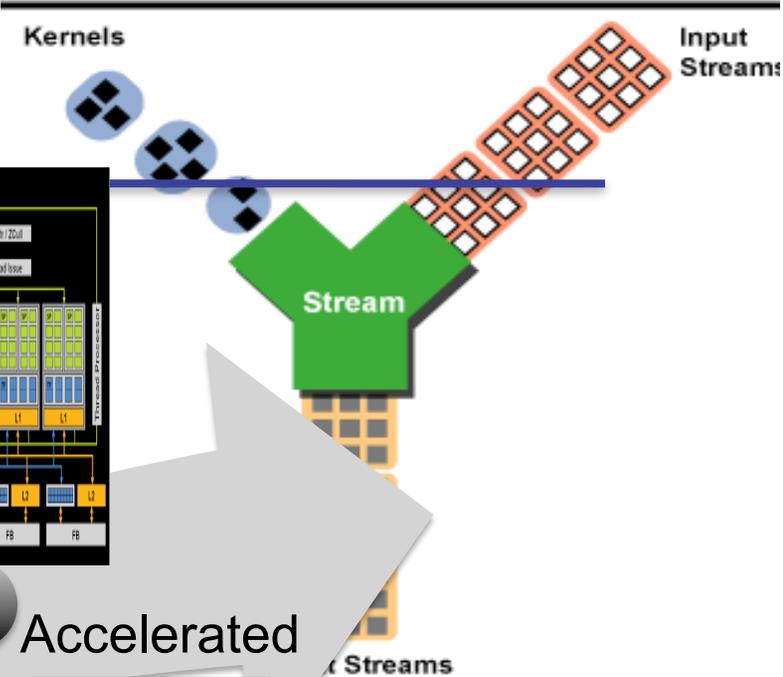
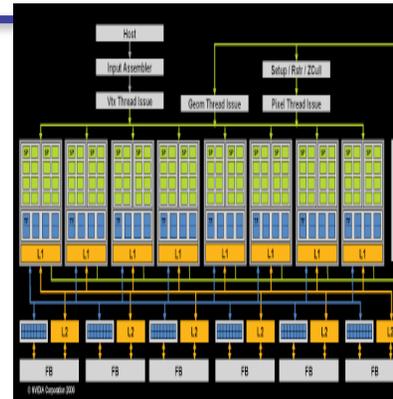
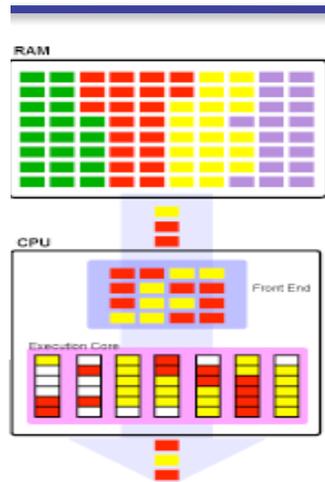
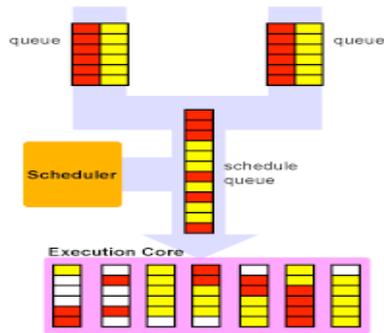
Multi core
x 100
150MW

Multi threaded
x 4
150MW

Fixed power per chip
More cores, threads per chip
1.6 billion-way parallelism

Demands algorithm redesign

Acceleration & Power for fit some (but not all) apps



Single node

- P = 1
- E = 150W

Multi node

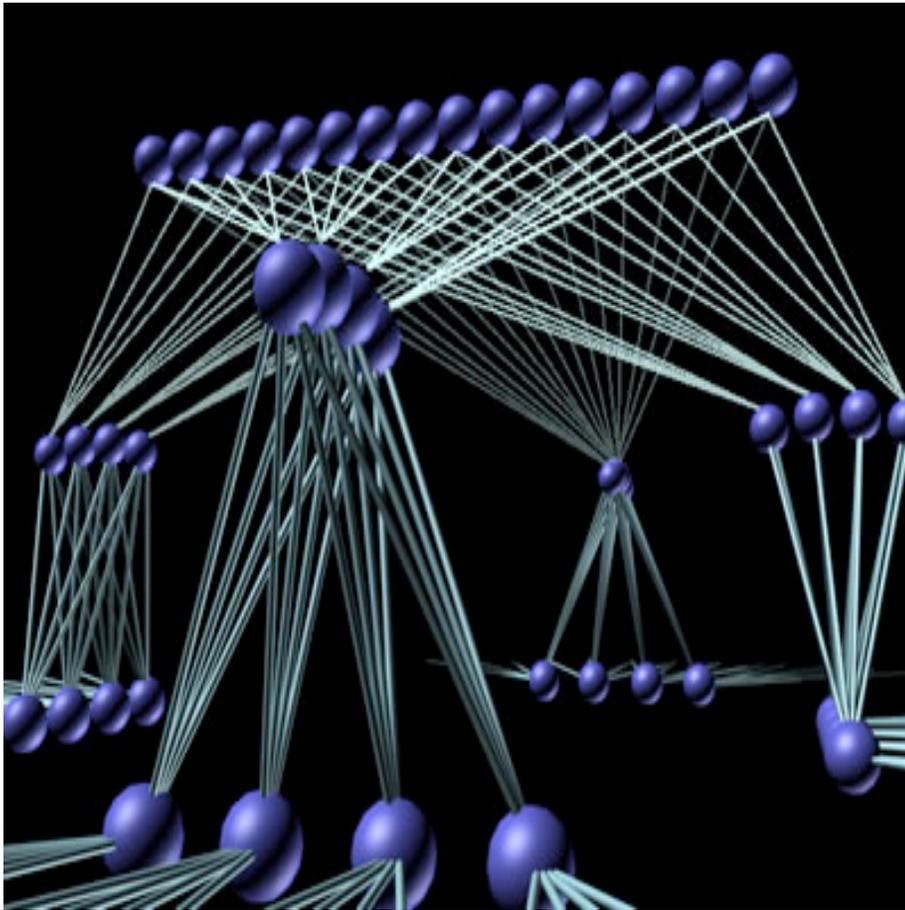
- X 1000,000,
- E=150 MW
- 4-core multicore peak performance – 93GFLOPS

Accelerated

- x 30,720
- Gforce 8800 peak performance 933 GFLOPS
- E = 300 MW
- Ex: GPU accelerated is 5 x faster /watt
- Will not work for all apps but can benefit some

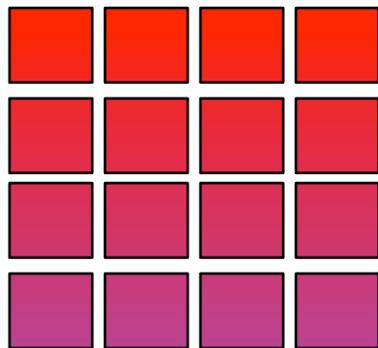
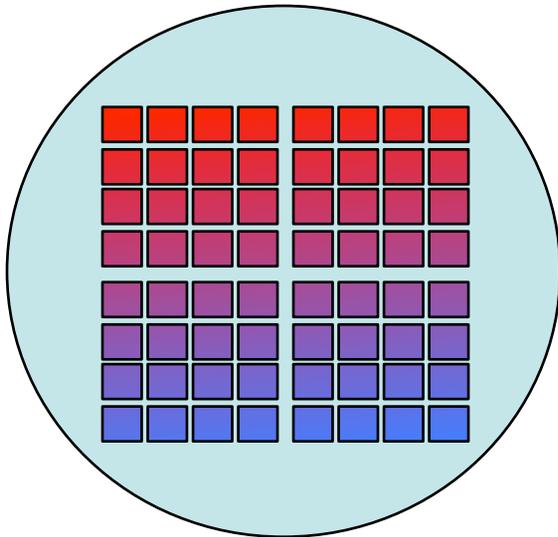


Importance of Network Power



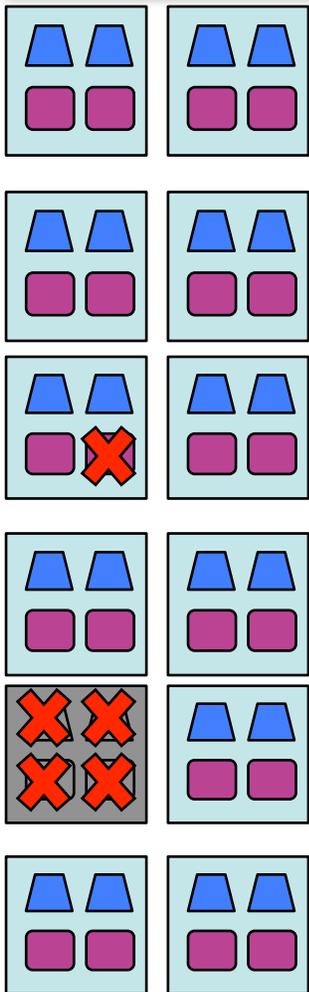
- Network power will increase in importance
 - As high as 60-70% system
 - DVFS and link throttling options to save energy
- Needs algorithm /software redesign

Process Variability



- Manufacturing is imperfect
- Die for 4 chips @ 16 cores
 - Top fast, high leak
 - Bottom slow, low leak
 - Variations within chip
- Reorganize computations to model variations
- Schedule and load balance for performance and energy
- Algorithms/software will have to model these variations

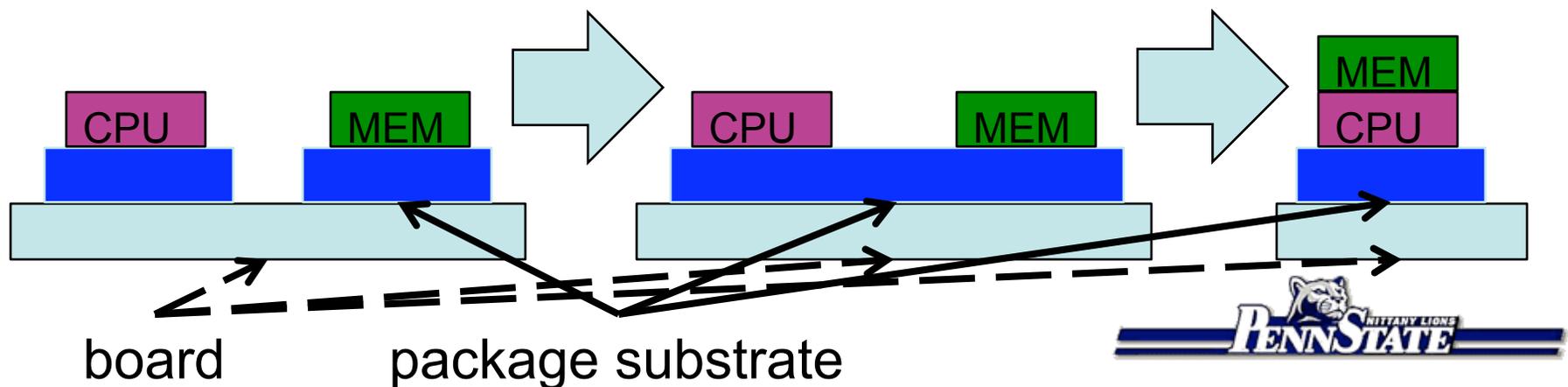
Failures & Soft-Errors



- Components will fail in 100 core chips
 - Not cost effective to throw out chip
- Use cores in diminished capacity
 - Example: failure of one functional unit
 - Disable core if unusable
- Soft errors (bit flips) in low V regimes & algorithm correctness
 - Algorithms/software have to be redesigned to be adaptive

Caches & Memory

- Caches not useful for applications w/o reuse or long reuse distances
- Alternatives such as user programmable memories that are power efficient
- Options for data-staging to mask latency
- **Algorithm/software reorganization**



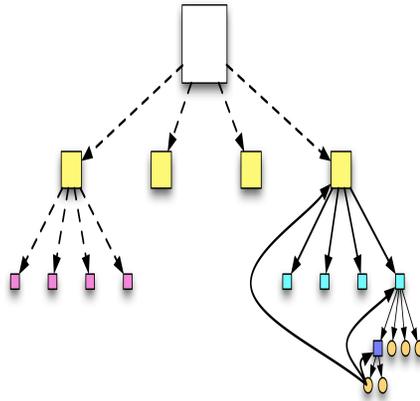
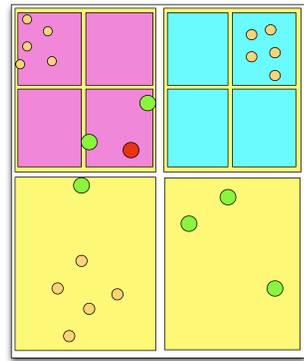
II: Energy-Aware Scaling

- Exploiting concurrency & managing power for $O(N)$ sparse graph/matrix
 - Algorithms/library design for billion way parallelism and dynamic adaptivity for energy efficiency
 - Cross node & network
 - At node

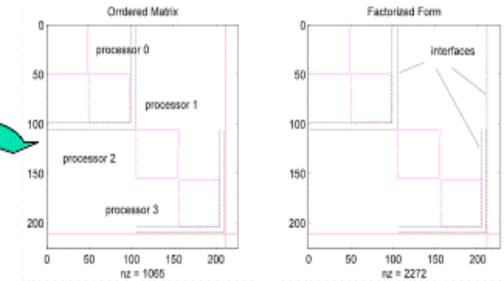
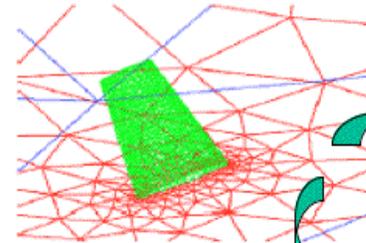


Sparse/Irregular Data -Driven Computations

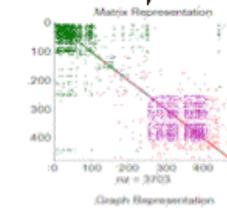
Interoperable, Sparse Data Structures and Transformations



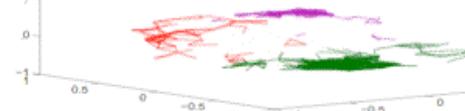
Discretizing with adaptive meshes



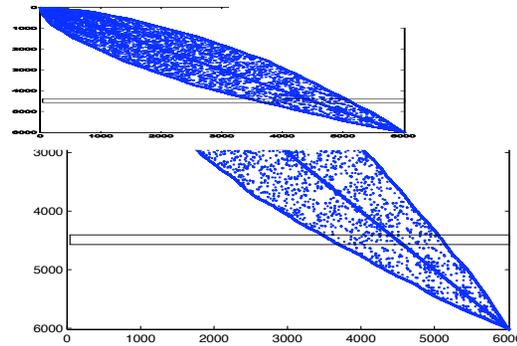
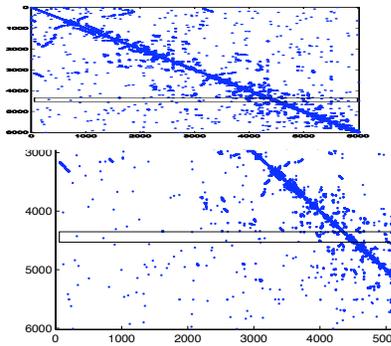
Graphs, networks, clusters in low dimensional space



Sparse matrices, automatically ordered to reveal hierarchical structure

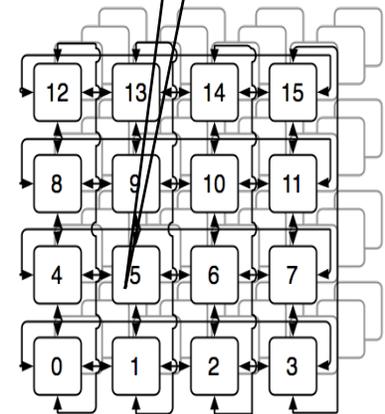
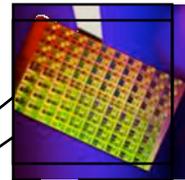
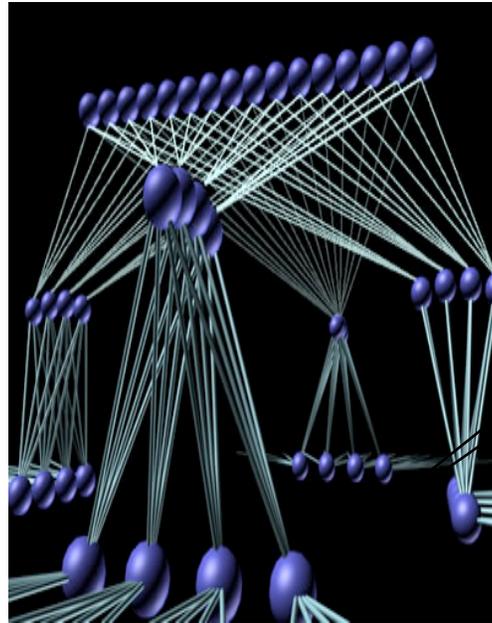
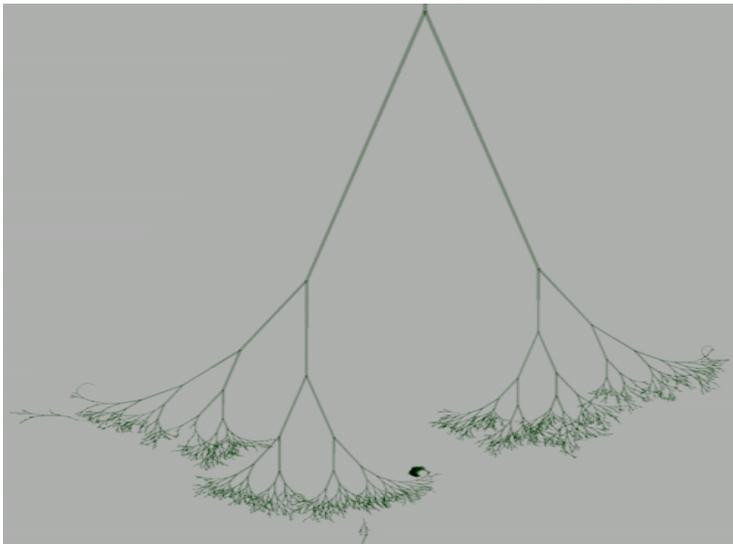
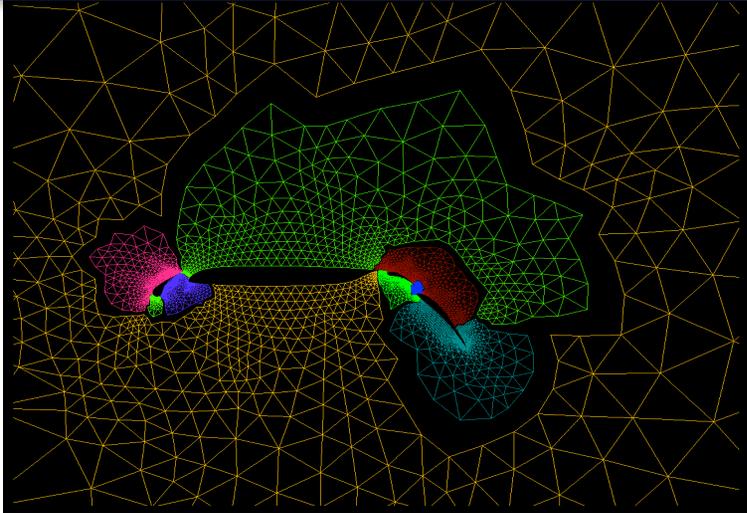


Sparse structures enable linear scaling of computational cost with problem size, parallelism, ..



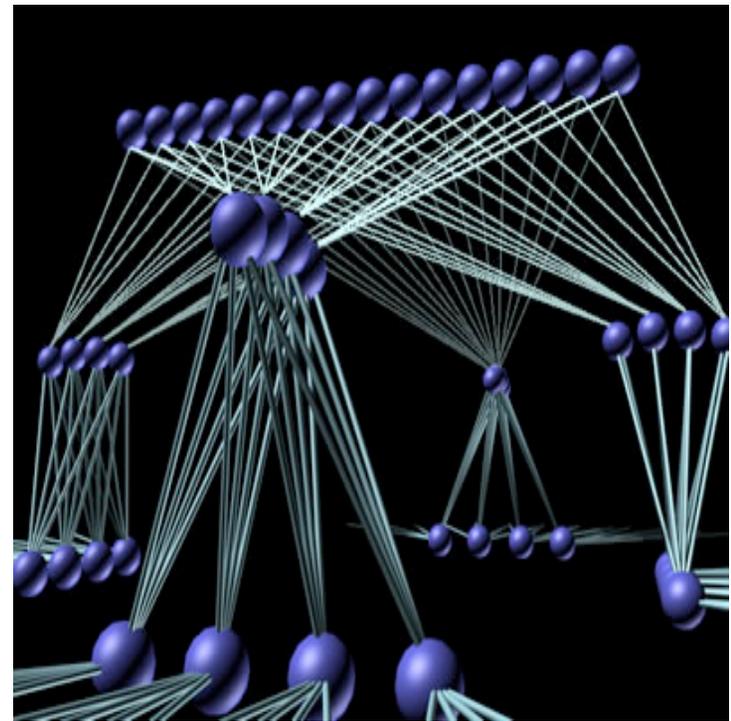
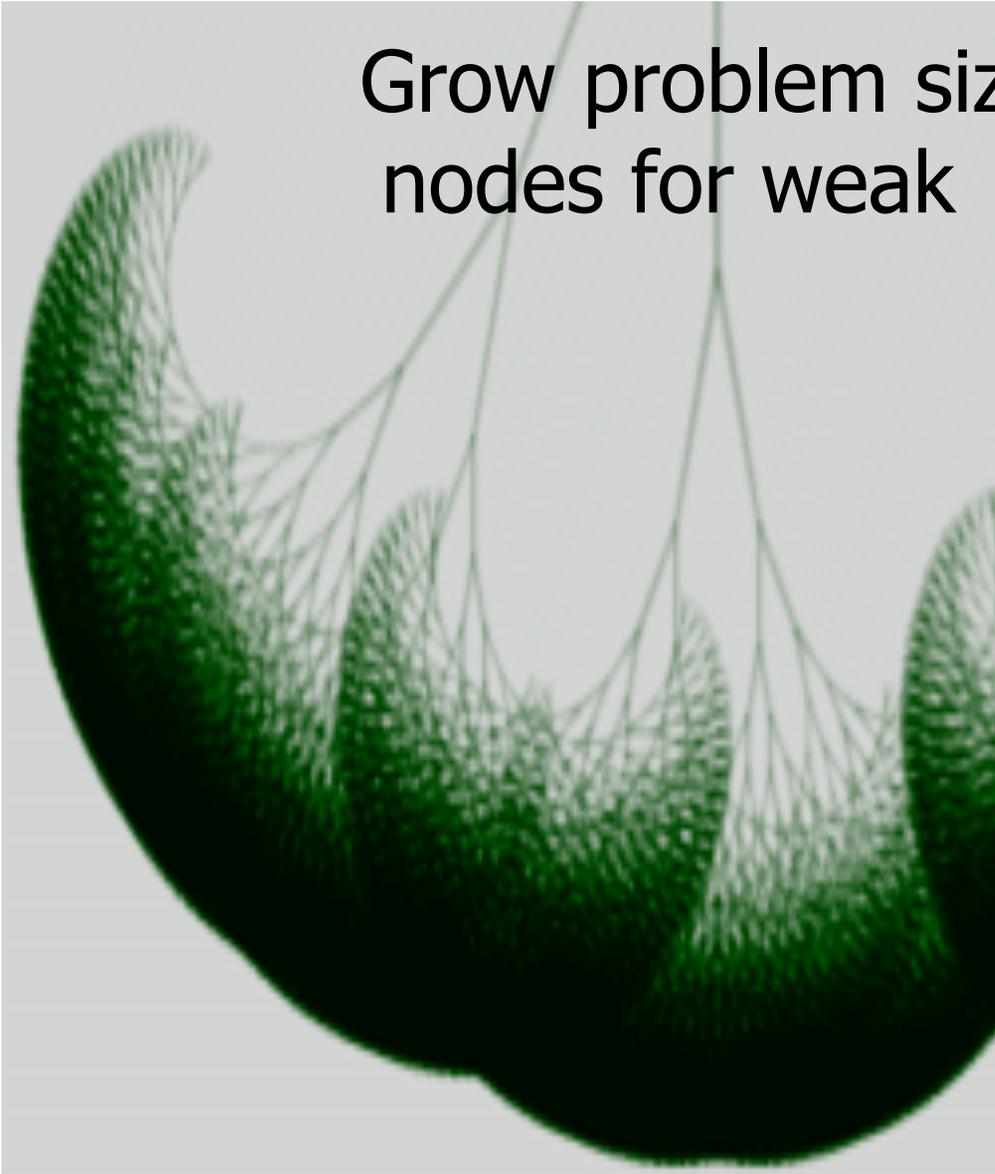
Application requirements ->
algorithm selection + tuning
-> H/W, S/W adaptivity

Partition and Map to MPP Nodes



Cross Node Scaling

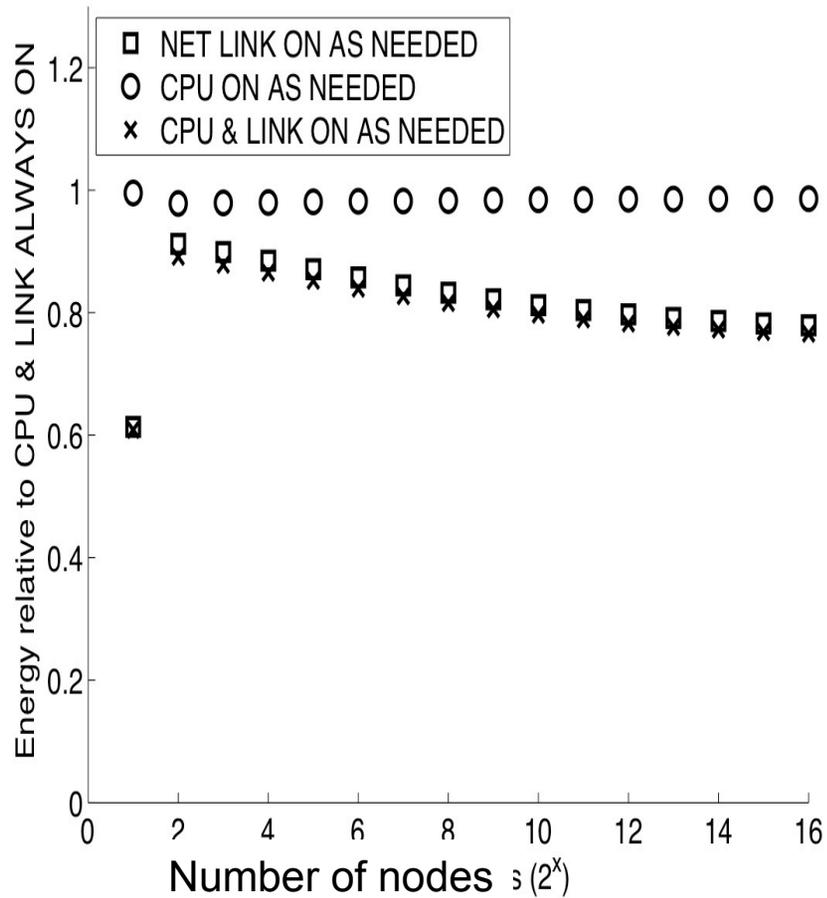
Grow problem size with number of nodes for weak (iso-efficient) scaling



Network Energy & Weak Scaling

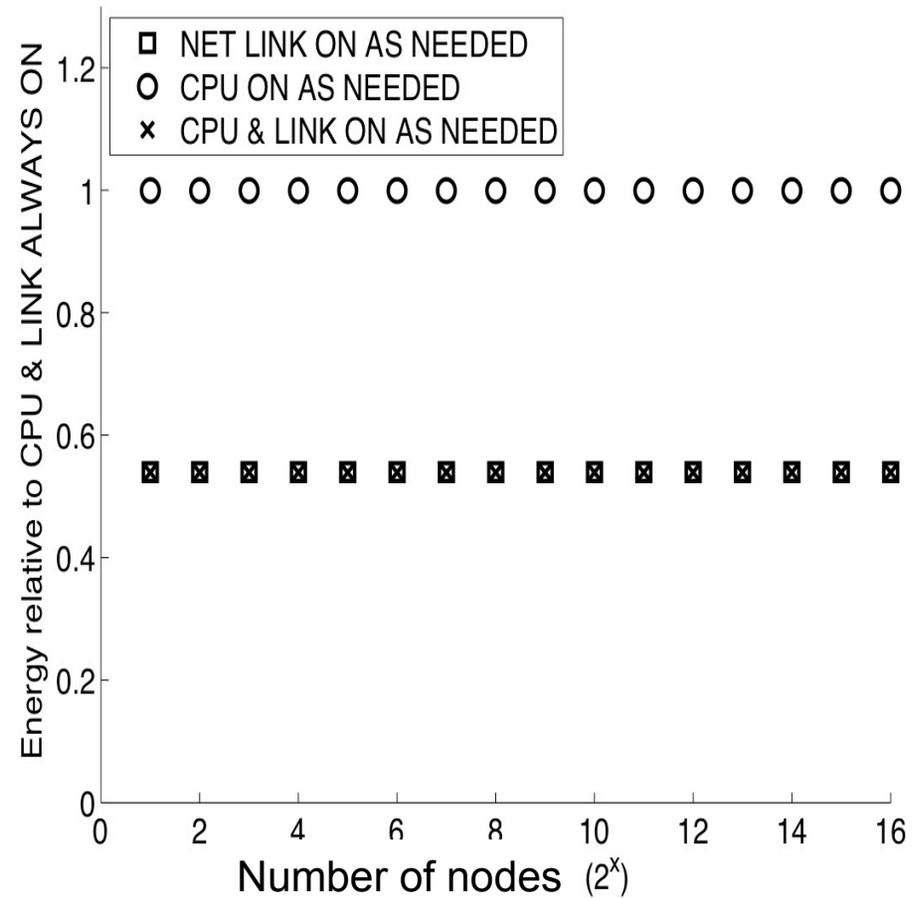
FFT

Low Power Processor – FFT – Weak Scaling



Mat-Vec

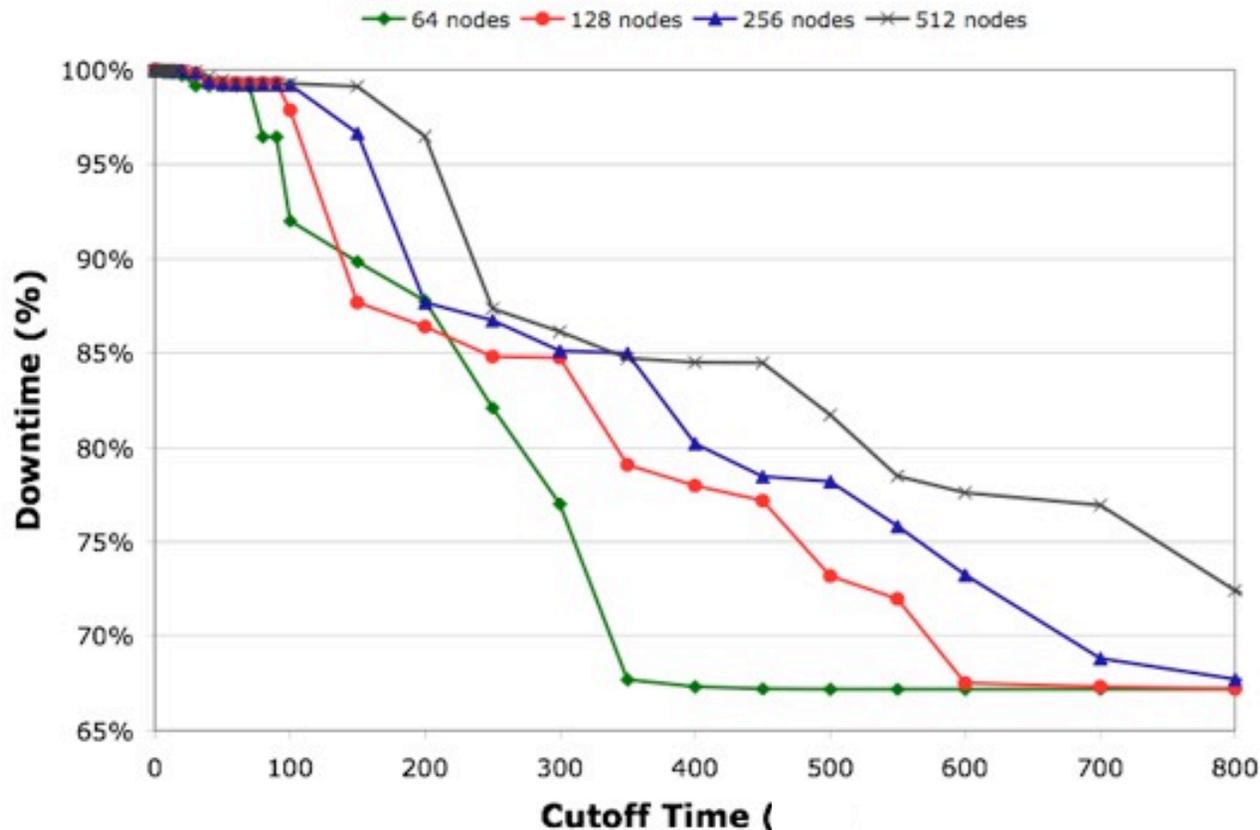
Low Power Processor – SMVM – Weak Scaling



Link Shutdown in Collective

Reduce Operation, 512 node torus

Link Shutdown Opportunity vs. Timer
65%-100% Scale



- Many links remain unused. For reduce, it's 66%
- Implement simple link shutdown (LS) hardware in the net
- Library code X LS hardware can utilize link shutdown



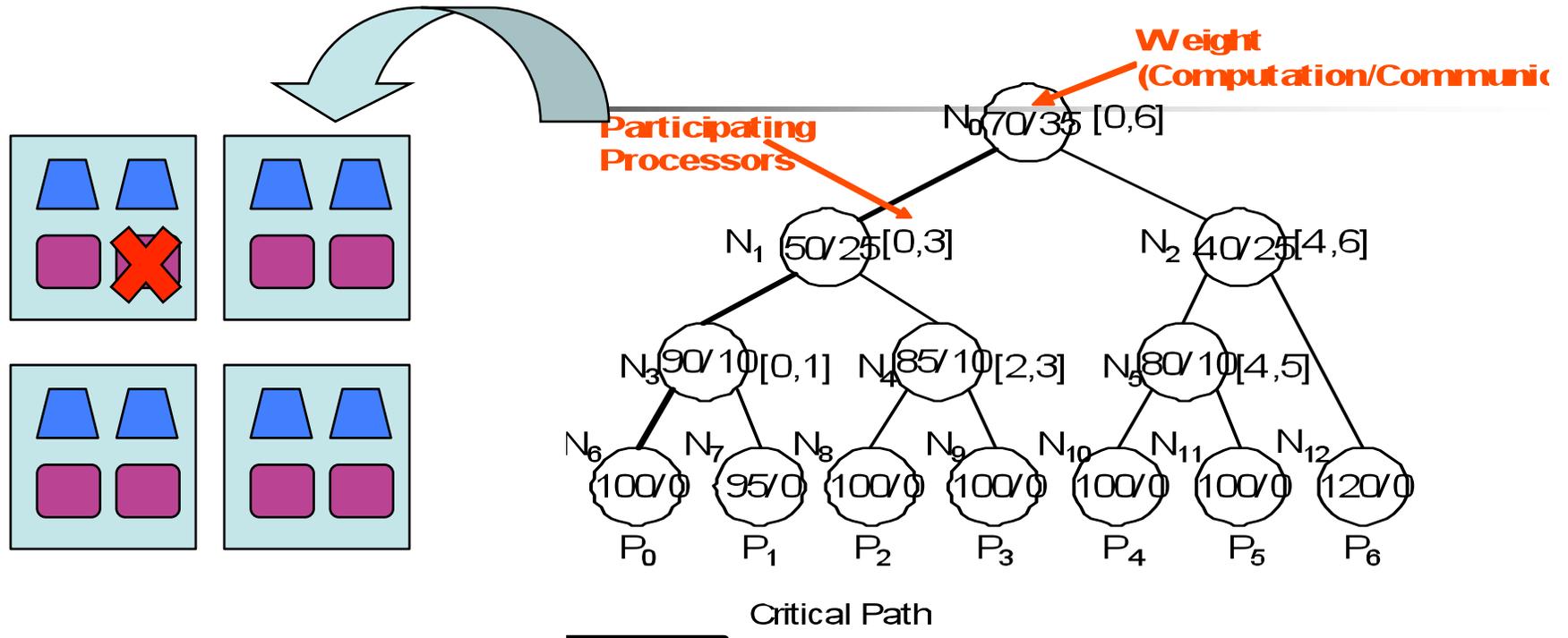
II: Energy-Aware Algorithms

- Node:
 - Scheduling for energy and reliability



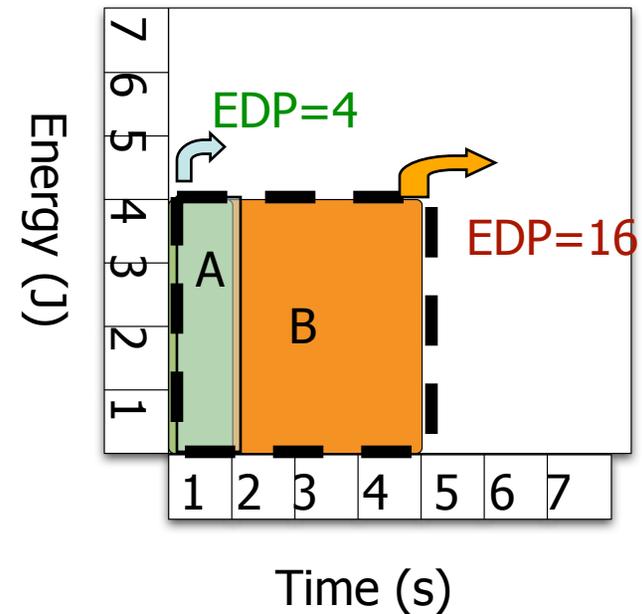
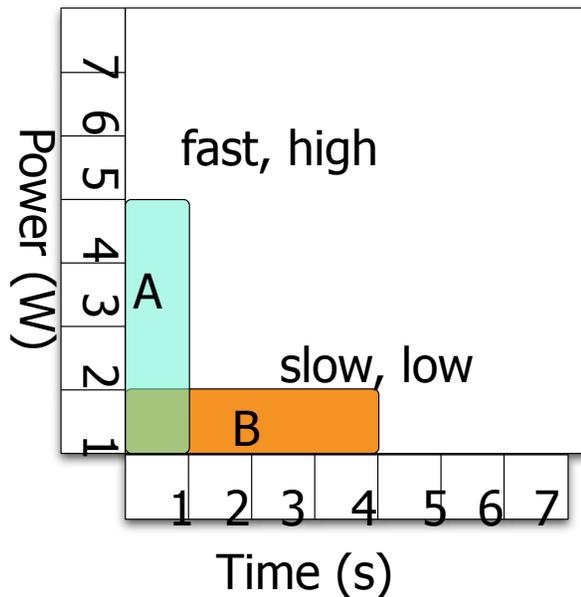
At Node Efficiency

- Fixed problem energy & performance efficiency at a node is key
- Critical path scheduling for performance, energy
- Model core variations for load balance



Measuring Energy Efficiency

Same code on two different systems A and B

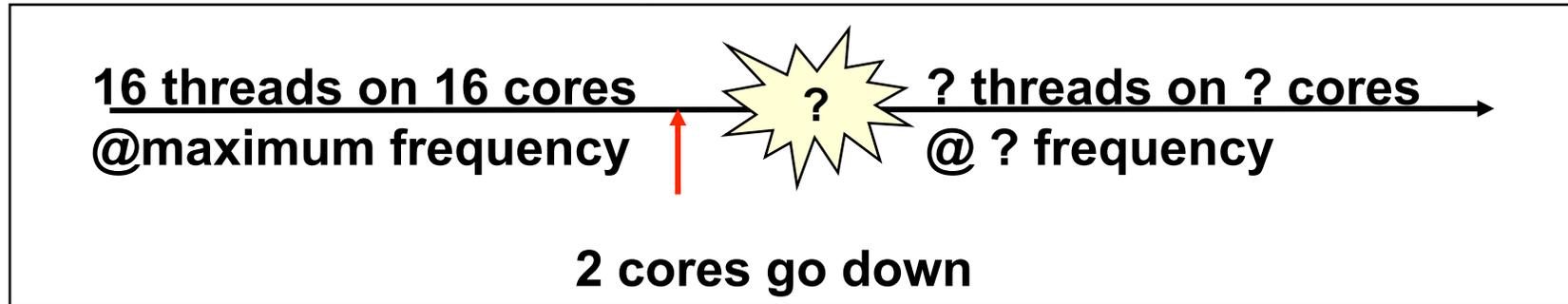


Equal energy (PDP)
does not differentiate A
from B

Energy**D**elay**P**roduct
(**E**nergy X **T**ime) is lower
for faster system A

Energy-Aware Adaptation to Failures

Program Execution



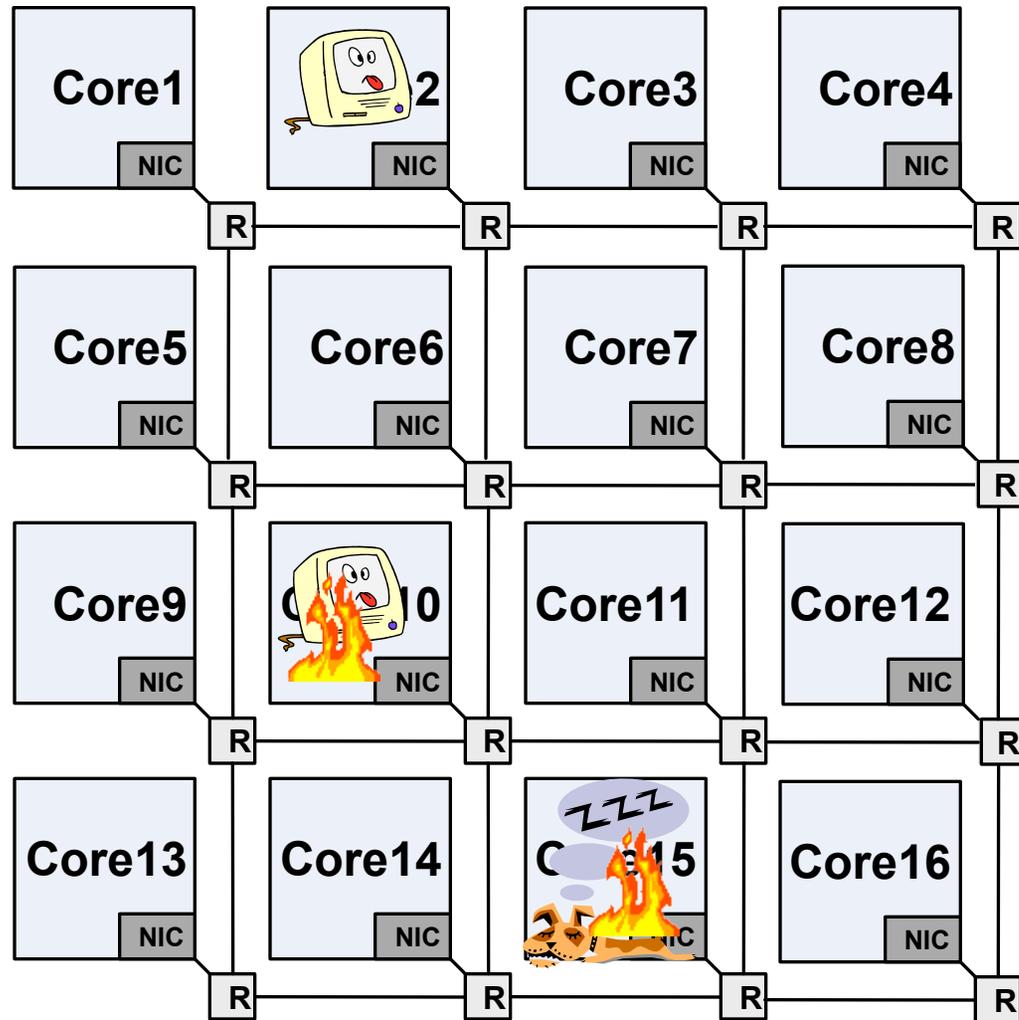
Scenarios

1. Change number of cores
2. Change number of cores and number of threads
3. Change number of cores, number of threads, and voltage/frequency levels

Mechanism

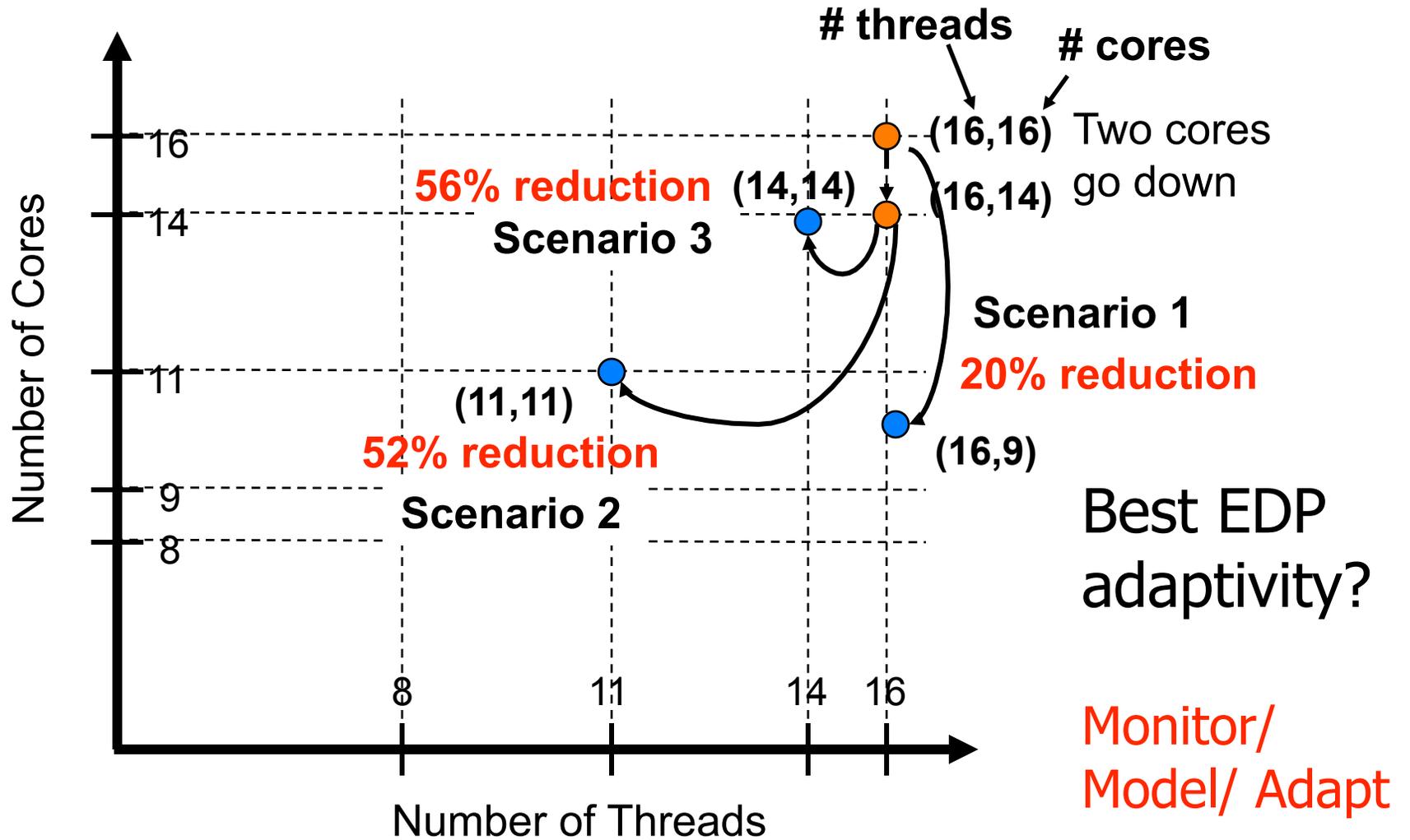
Helper thread + dynamic scheduling
Function-based adaptivity

Resiliency Issues in Multicore



- Run away leakage on idle cores
- Thermal emergencies
- Transient errors

EDP Landscape for Multigrid



[Ding, Kandemir, Raghavan, Irwin, IPDPS'08]

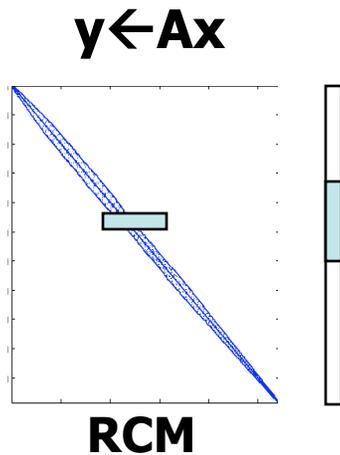
II: Energy-Aware Algorithms

- Node:
 - Data staging and user-programmable memories



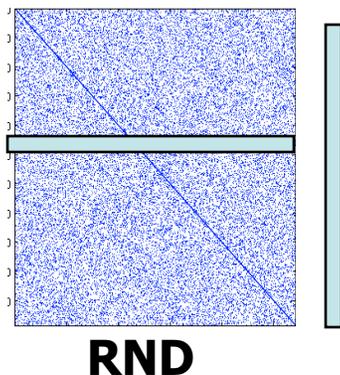
Data Staging in Multicores

- Efficiency: performance, power



Data when and where it can be computed upon (data locality)

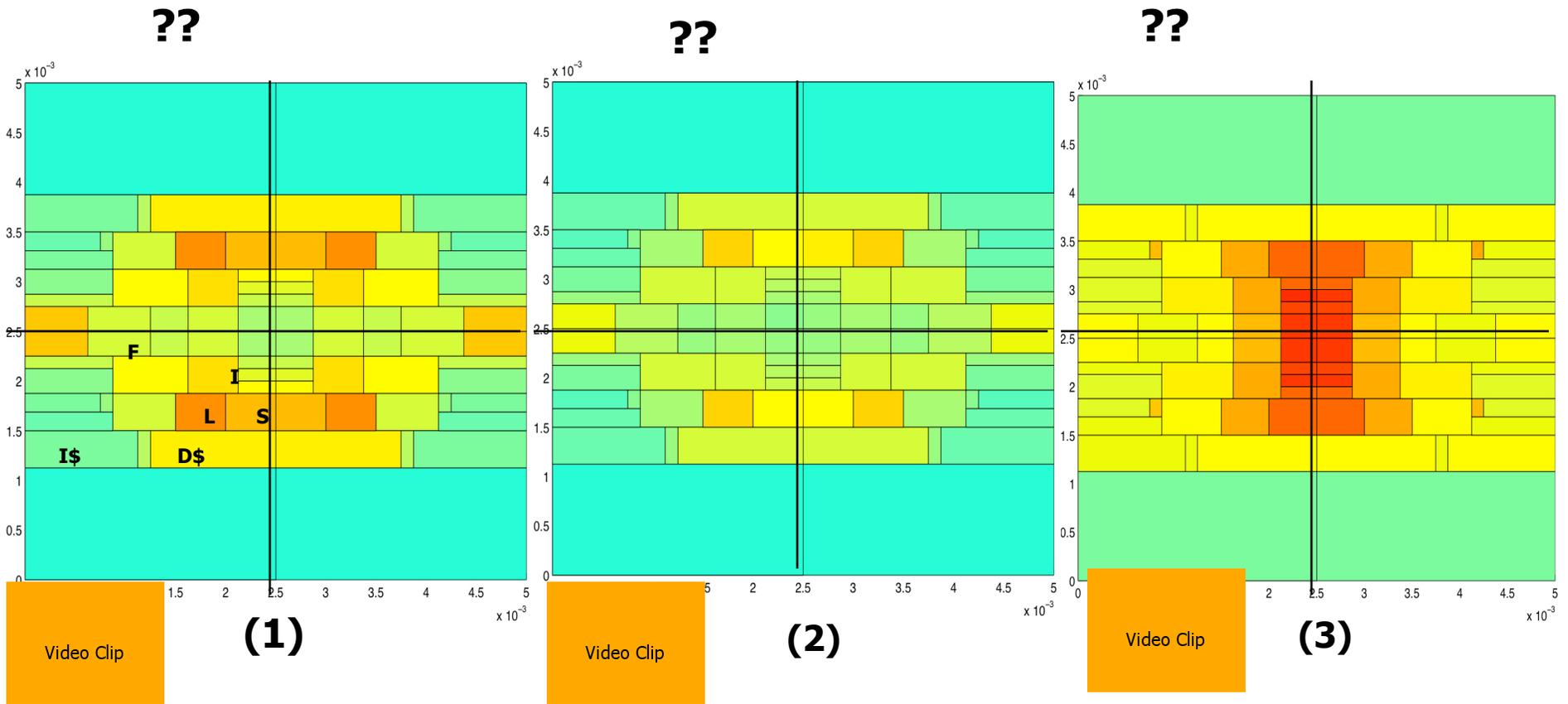
Power when and where it enables useful activity (power locality)



- Efficiency: Fraction relative to DGEMM for sparse matrix vector multiplication (SMV)
- SMV variants: CSR format: RCM, RND
RCM enhances locality in x ... Toledo, Yelick..

Temperature Evolution (4-core)

DGEMM, SMV_RCM, SMV_RND



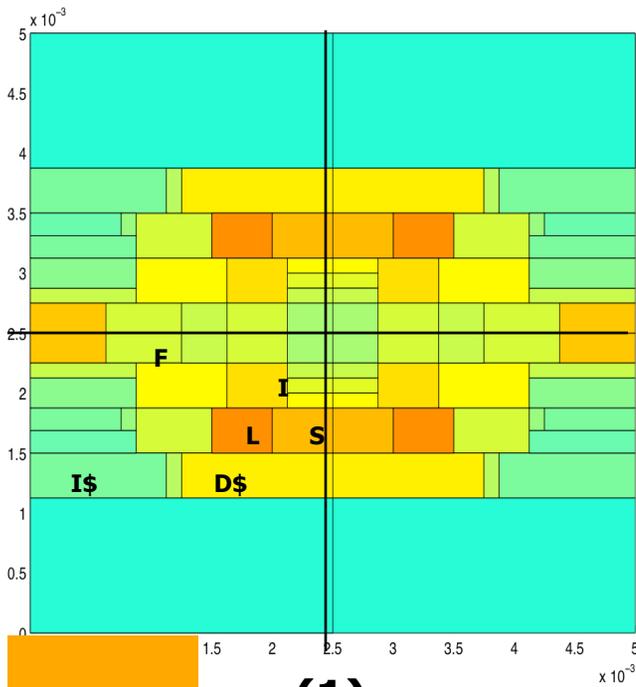
Temp: 0

65 C



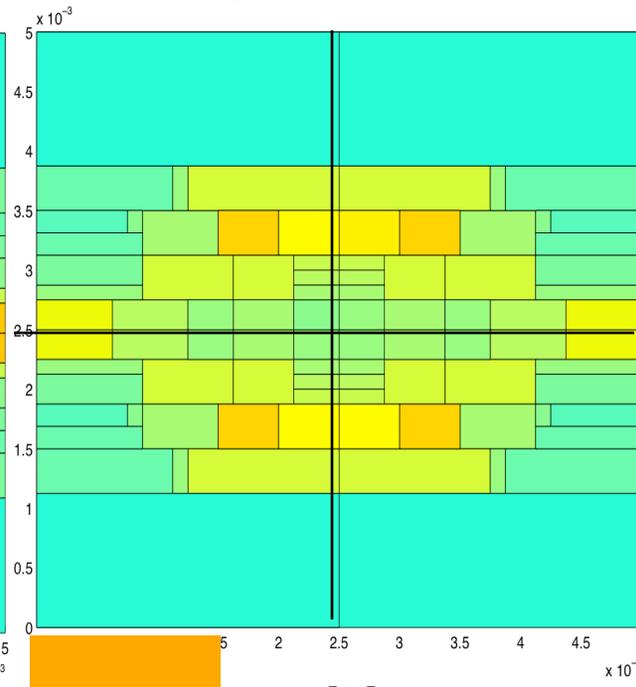
Temperature Evolution (4-core)

SMV_RCM



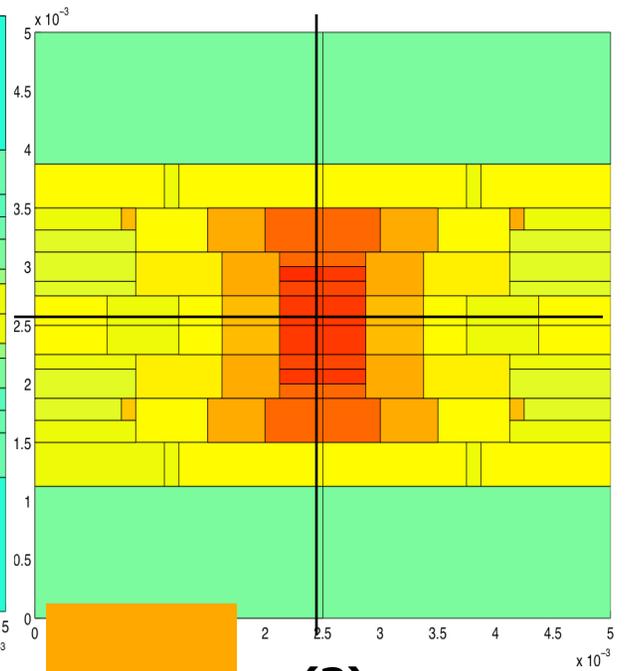
(1)

DGEMM



(2)

SMV_RND



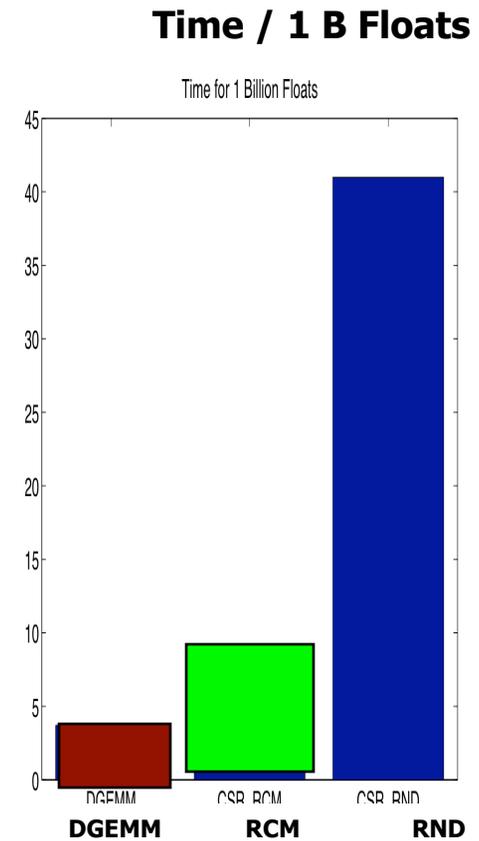
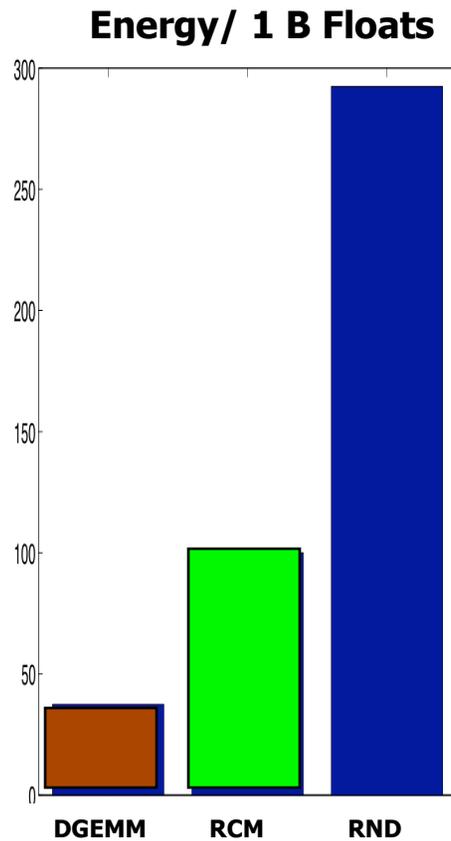
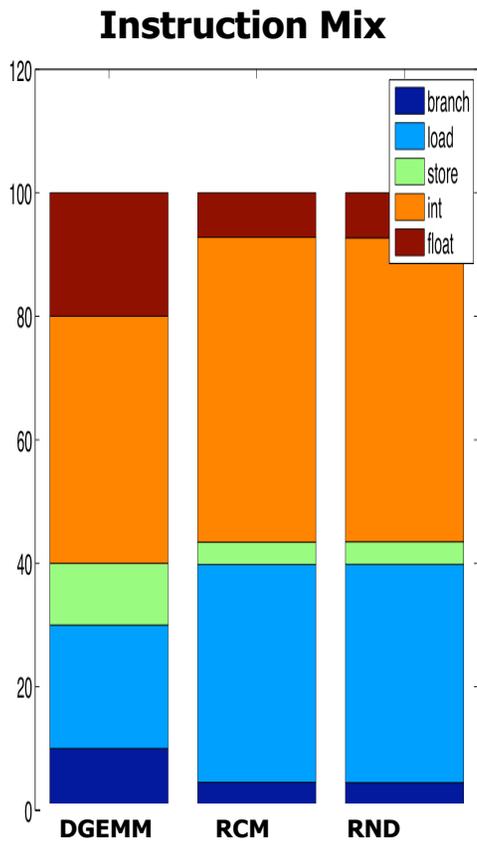
(3)

Temp: 0

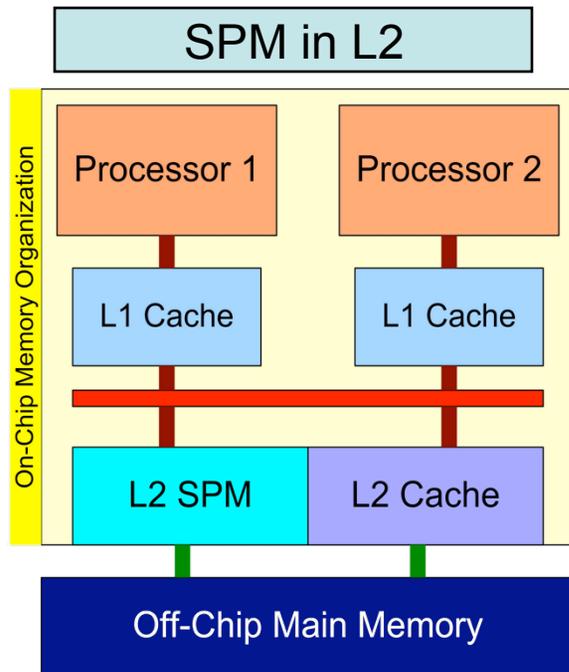
65 C



DGEMM, SMV Profiles

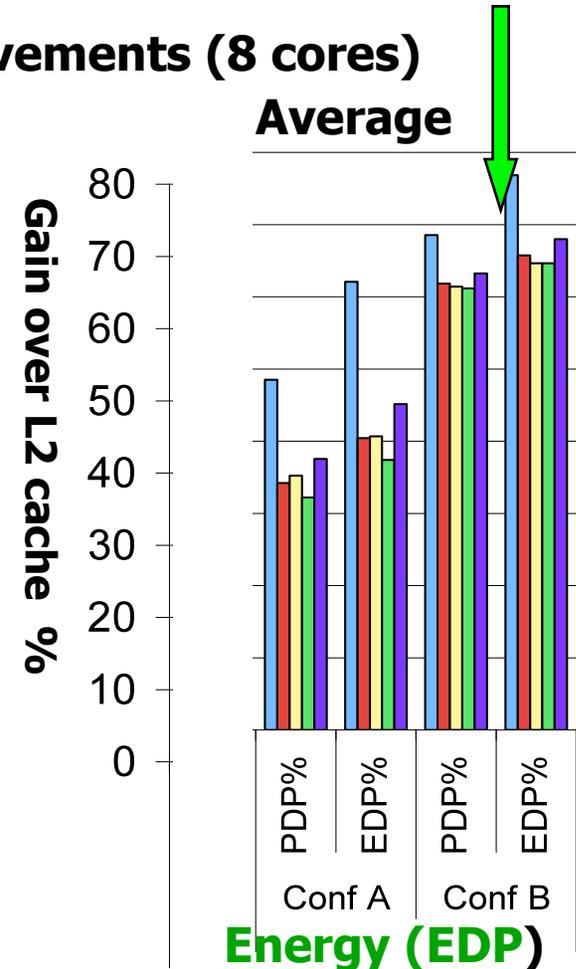
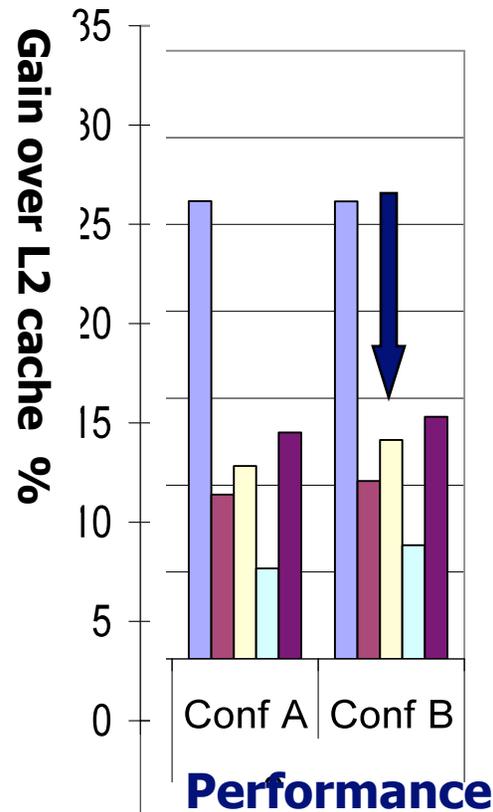


Scratch Pad Memory



- L2 memory split into:
1. **Cache + SPM** (Conf A)
 2. **Entirely SPM** (Conf B)
 3. SMV-CSR-RCM: A in SPM (Raghavan et al. in IPDPS 08)

L2 Relative Improvements (8 cores)



Summary

- Algorithm/software redesign needed for sparse/irregular data-driven kernels
 - Multi-level parallelism—ILP to Multi-node
 - Controlling network energy dynamically
 - At node scheduling for performance, power, h/w variations
 - Software control of data-staging
 - Reliability/correctness in soft-error regimes
- APIs/Abstractions/Languages for revealing/exploiting S/W & H/W features for cross-layer optimizations
- System support for state modeling and recovery



Acknowledgements

- Joint work with:
 - Konrad Malkowski, Bryan Cover, Yang Ding
 - Mary Jane Irwin and Mahmut Kandemir
- Support from:
 - NSF, DoD, IBM
 - Institute for CyberScience@PSU
- Thanks to:
 - Google Images, Wikipedia, ARSTechnica
 - Exascale organizers and attendees

