Automatic Data Migration for Reducing Energy Consumption in Multi-Bank Memory Systems

V. De La Luz, M. Kandemir

The Pennsylvania State University

I. Kolcu

UMIST

Overview

- Introduction.
- DRAM Architecture and Operating Modes.
- Data Migration.
- Results.
- Conclusions.

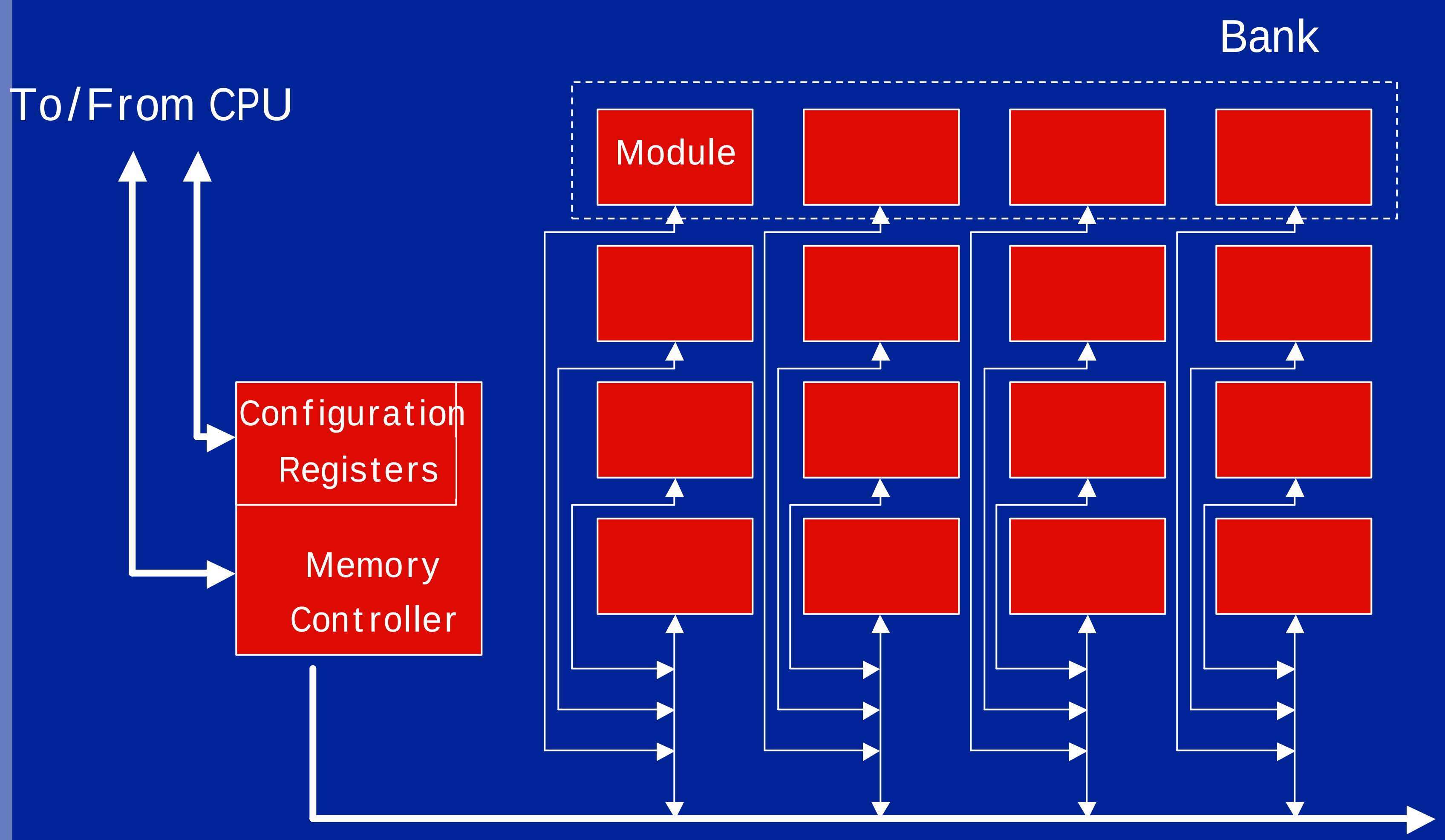
Introduction

- Memory is an energy bottleneck in current systems.
- Embedded applications are becoming more data-centric.
- In a multi-bank memory systems, idle banks can be transitioned to low-power modes saving important amounts of energy.

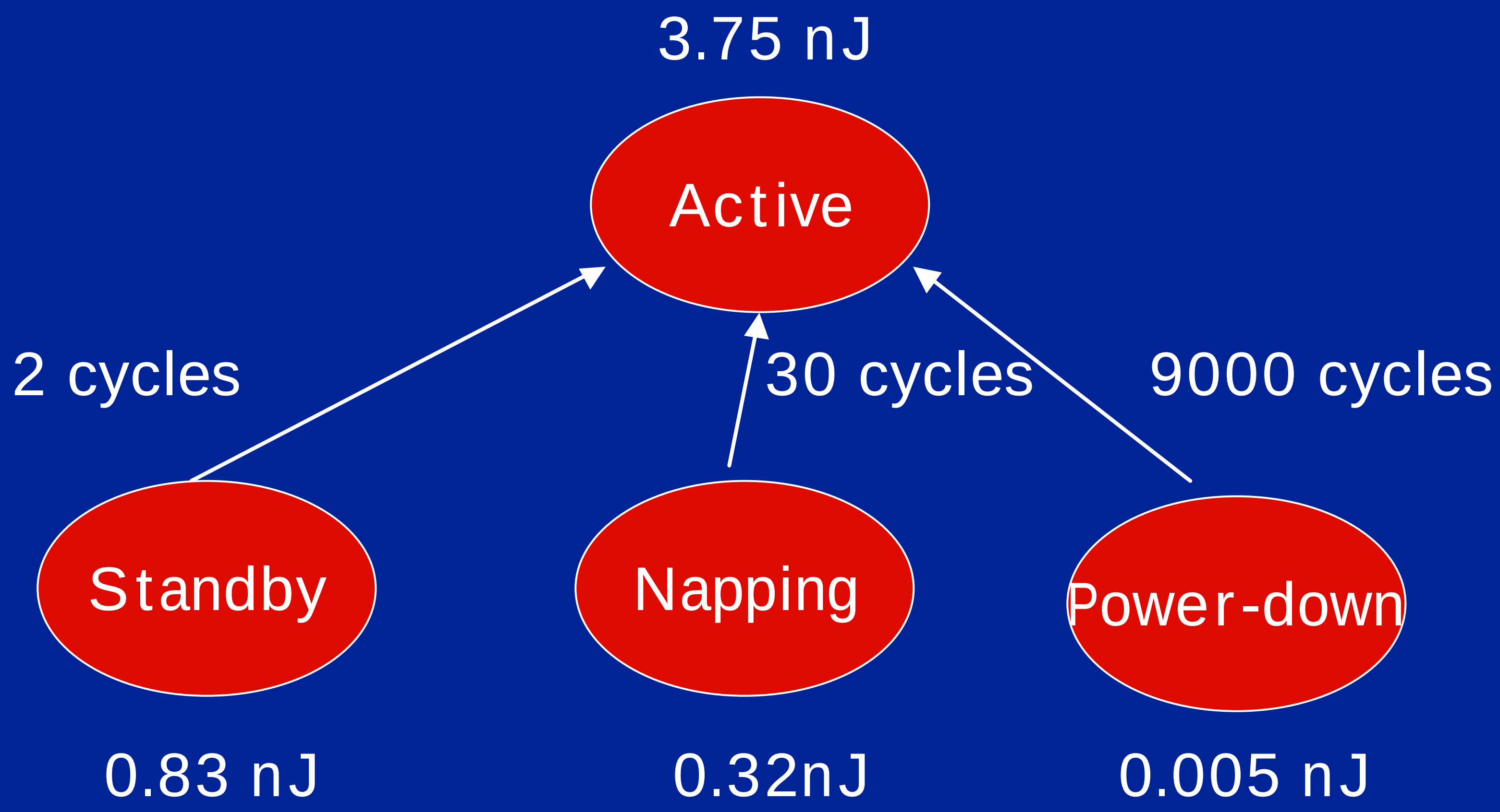
Introduction (2)

- In order to reduce energy, we can modify the data layout to improve temporal affinity.
- We present a dynamic data migration scheme which co-locates data elements automatically to take advantage of variations in inter-array access patterns.

DRAM Memory Architecture



Memory Operating Modes



Data Migration

- The idea is to bring together arrays that originally were mapped in different banks but that are accessed together recently.
- We expect that if two data sets are accessed together in the past, they will continue to be accessed together in the future.

Data Migration (2)

- Migrating just portions of an array make it more difficult to address that array later on.
- We migrate only entire arrays.
- In order to implement data migration we need to:
  - Detect which arrays are accessed together.
  - Decide when to migrate.

Detecting Temporal Affinity

- We need to know which data sets are accessed together.
- At regular intervals (sampling threshold), the data accessed are sampled.
  An affinity table is updated.
- Table entries are triplets (A, B, ‘count’), meaning that arrays A, B are accessed together count times.

When to migrate?

- We need to select a suitable period (migration threshold) for any given application, given that access patterns vary from one application to another.
- A small threshold might incur high overhead.
- A large threshold, on the other hand, might miss opportunities for migration.

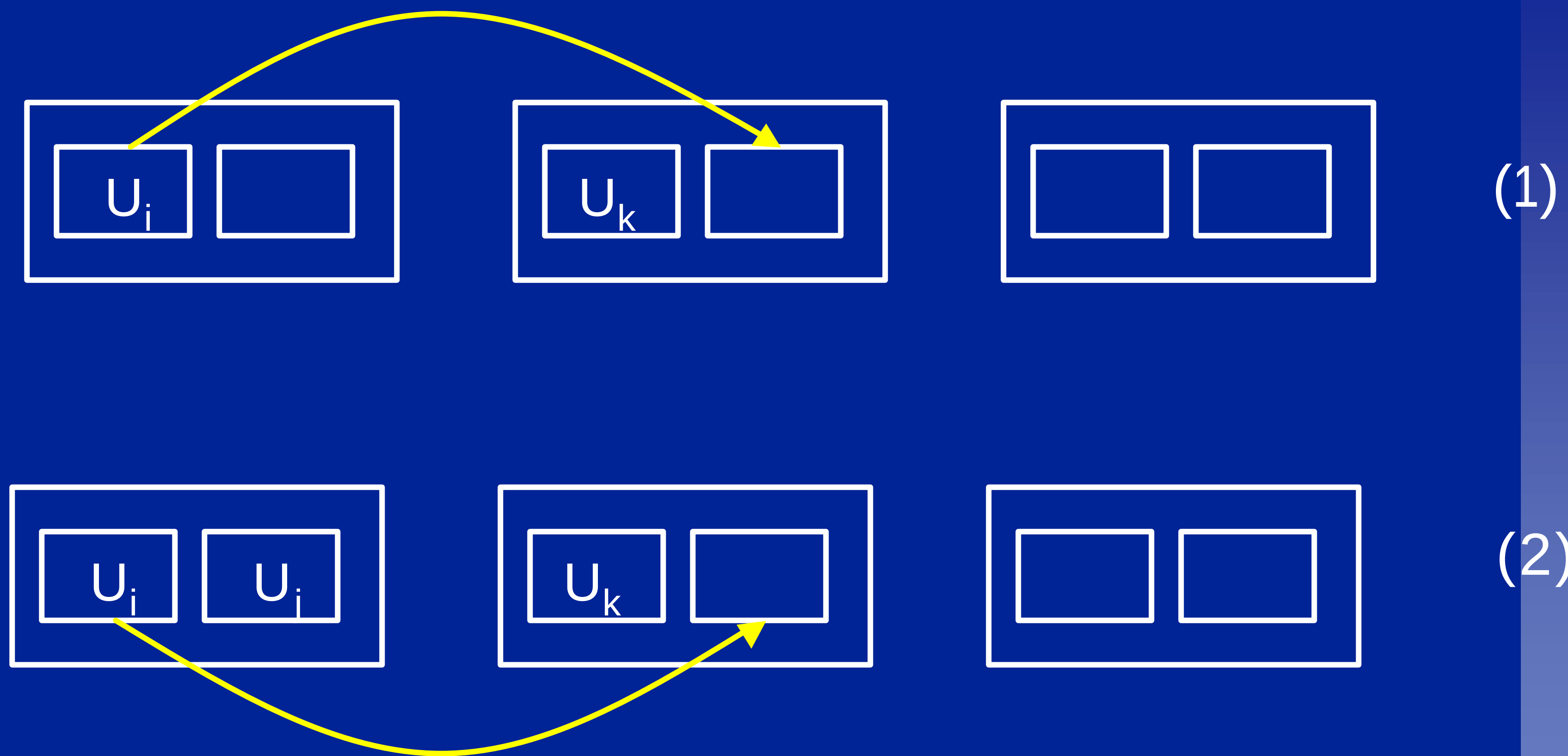
Tracing Temporal Affinity

- We try to prevent old affinity relations from dominating recent relations.
- When we increment count values of currently accessed arrays, we can also decrement the count values of all other (not currently accessed) arrays.
- In this way, we reward most recent affinity relations and punish old affinity relations.

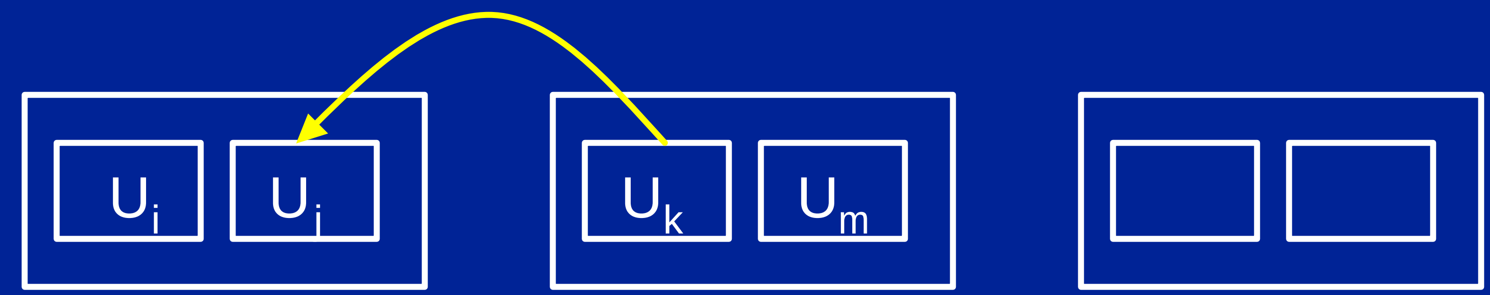
What arrays to migrate?

- Two approaches
  - mFT
    - The m triplets with the highest counts are considered for migration.
  - nLT
    - Consider only the triplets if and only if its count value is larger than n.

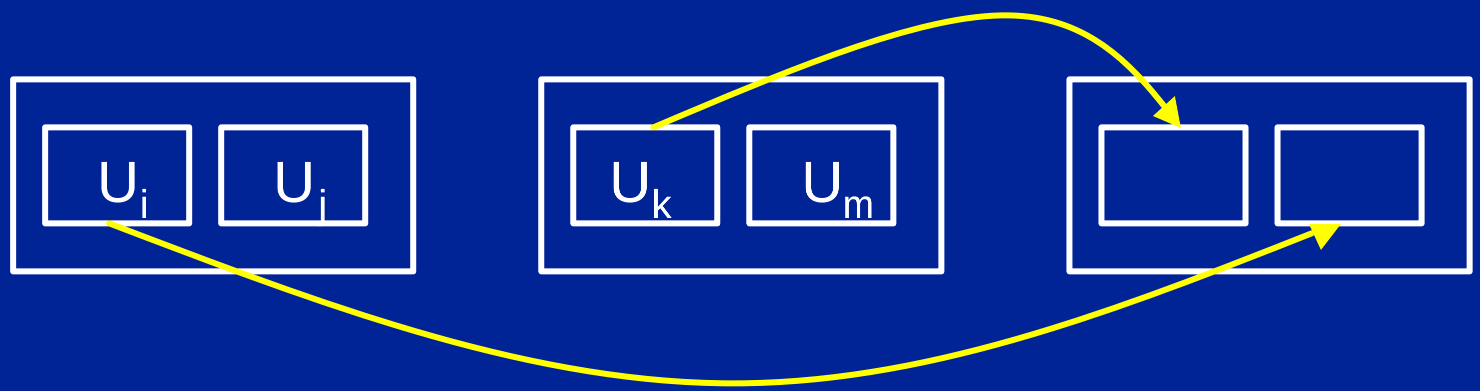
Migration: Some Cases



Migration: Some Cases (2)



(3)



(4)

Data Freezing and Defrosting

- What if an array is constantly migrated?
- We define a migration counter per array. If an array is migrated more than a certain value (freezing threshold), that array is frozen.
- After a period of time (defrosting threshold) migration counters for frozen arrays can be decremented.

Benchmarks

Benchmark	Input Size (MB)	Base Energy (mJ)	Overhead
adi	19.2	1,492	2.5%
apsi	42.4	499	16.2%
btrix	48.8	719	7.6%
bflux	33.5	98,501	4.8%
phods	32.9	37,785	2.8%
tomcatv	28.0	19,850	7.7%
vpenta	44.0	13,714	6.8%

Experimental Setup

- Sampling Threshold: 1000 cycles.
- Migration Threshold:100,000 to 1.5M cycles.
- Cache: 16KB, 2-way associative.
- Normalized results to the case when no migration was used but the arrays are given the best possible static layout.

Normalized Energy Consumption (1FT)

Benchmark	Sequential Mapping		Random Mapping	
	8 X 8 MB	4 X 16MB	8 X 8 MB	4 X 16MB
adi	92.5	96.0	92.5	96.0
apsi	68.4	61.4	68.2	61.1
btrix	105.0	100.0	105.0	100.0
eflux	92.8	94.3	90.5	91.8
phods	72.7	61.6	70.0	60.3
tomcatv	66.2	60.4	66.0	61.1
vpenta	93.9	83.7	92.8	81.7

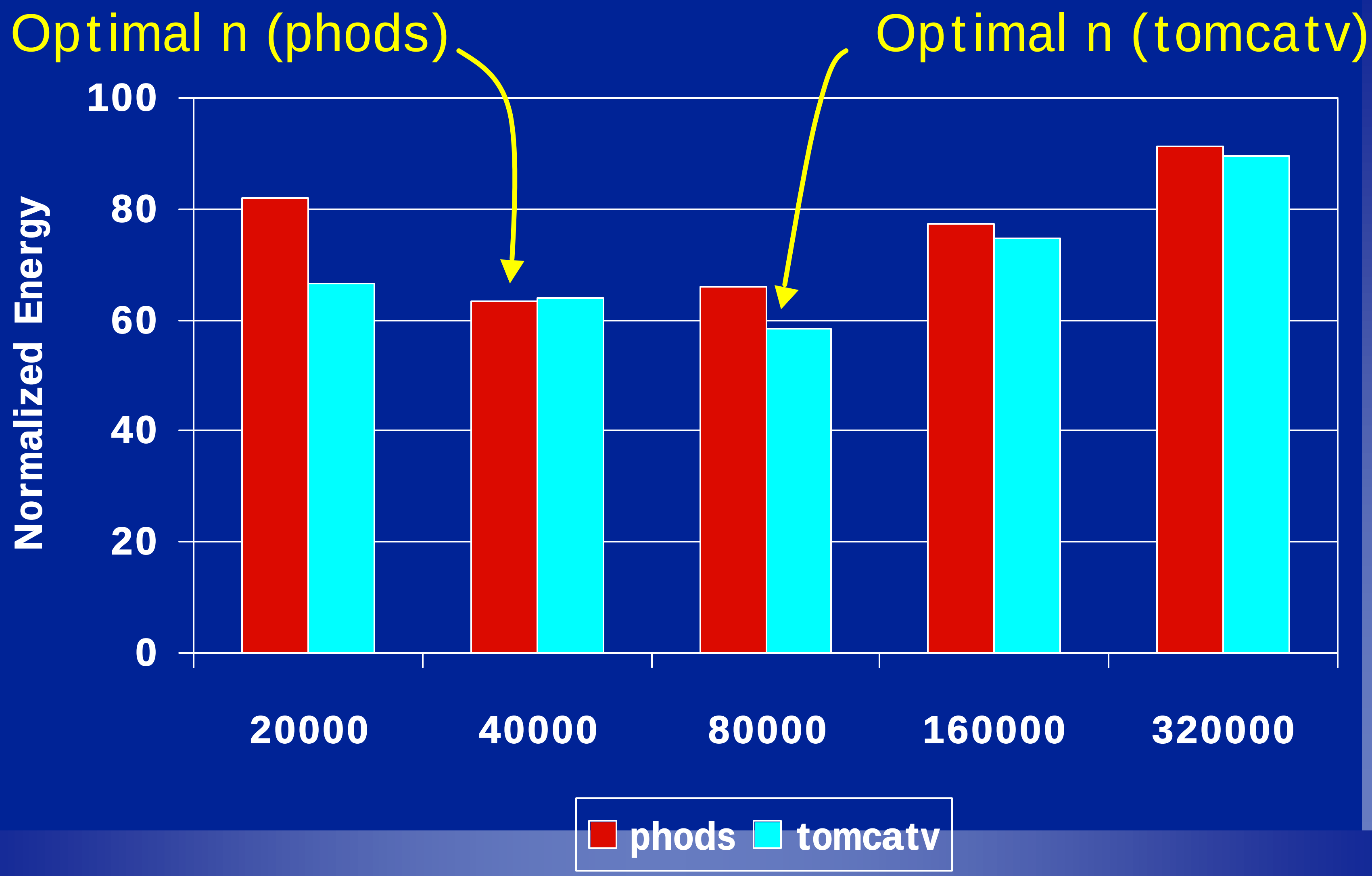
Sensitivity to Sampling and Migration Thresholds

ST2 = 2 * ST1, MT2 = 2 * MT1

Benchmark	Sampling Threshold		Migration Threshold	
	ST1	ST2	MT1	MT2
adi	-0.5%	-0.6%	0.0%	0.0%
apsi	-1.0%	-1.0%	18.6%	18.8%
eflux	0.0%	0.0%	-8.3%	-8.3%
tomcatv	-2.4%	-2.8%	19.6%	19.6%
vpenta	-0.8%	-0.8%	7.0%	7.0%

Normalized Energy Consumption (nLM)

(8 x 8 MB)



Conclusions

- We presented an automatic data migration strategy for low power. It tries to co-locate arrays with temporal affinity in the smallest number of banks.
- Our approach achieves good, promising results that encourage us to do further research on this area.

Microsystems Design Laboratory

The Pennsylvania State University

Dept. of Computer Science and Engineering

http://www.cse.psu.edu/~mdl

Energy Overhead of Migration