

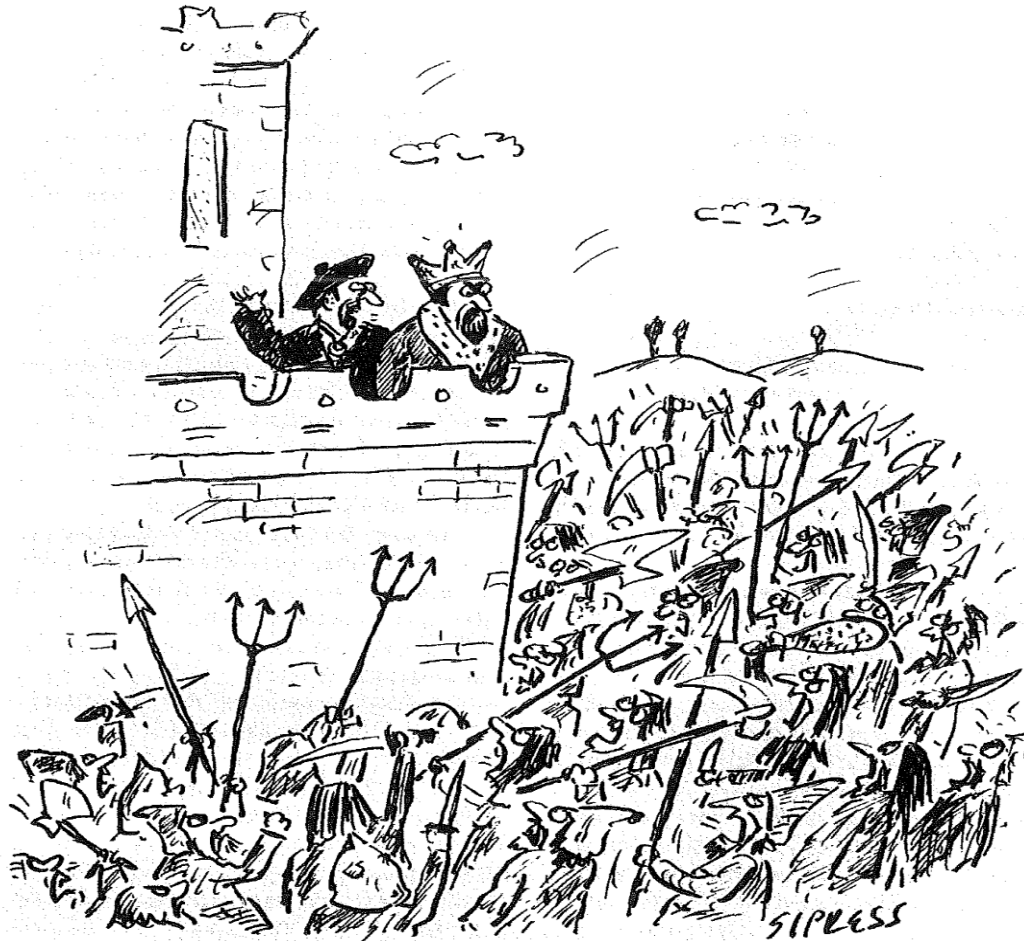


Mark D. Hill, Univ. of Wisconsin-Madison 12/2013 @ NSF CISE Distinguished Lecture

A talk in 2 ¼ parts:

- **21st Century Computer Architecture (whitepaper)**
- Efficient Virtual Memory for Big Memory Servers
- Opportunistic Virtual Cache (short, optional)

Lecture delayed by Gov't shutdown



"If they don't quit complaining you could threaten them with democracy."

The New Yorker, 9/16/2013

21st Century Computer Architecture

A CCC community white paper

<http://cra.org/ccc/docs/init/21stcenturyarchitecturewhitepaper.pdf>

- Participants & Process
- Information & Commun. Tech's Impact
- Semiconductor Technology's Challenges
- Computer Architecture's Future
- Pre-Competitive Research Justified

White Paper Participants

Sarita Adve, U Illinois *

David H. Albonesi, Cornell U

David Brooks, Harvard U

Luis Ceze, U Washington *

Sandhya Dwarkadas, U Rochester

Joel Emer, Intel/MIT

Babak Falsafi, EPFL

Antonio Gonzalez, Intel/UPC

Mark D. Hill, U Wisconsin *,**

Mary Jane Irwin, Penn State U *

David Kaeli, Northeastern U *

Stephen W. Keckler, NVIDIA/U Texas

Christos Kozyrakis, Stanford U

Alvin Lebeck, Duke U

Milo Martin, U Pennsylvania

José F. Martínez, Cornell U

Margaret Martonosi, Princeton U *

Kunle Olukotun, Stanford U

Mark Oskin, U Washington

Li-Shiuan Peh, M.I.T.

Milos Prvulovic, Georgia Tech

Steven K. Reinhardt, AMD

Michael Schulte, AMD/U Wisconsin

Simha Sethumadhavan, Columbia U

Guri Sohi, U Wisconsin

Daniel Sorin, Duke U

Josep Torrellas, U Illinois *

Thomas F. Wenisch, U Michigan *

David Wood, U Wisconsin *

Katherine Yelick, UC Berkeley/LBNL *

“*” contributed prose; “**” effort coordinator

Thanks of CCC, Erwin Gianchandani & Ed Lazowska for guidance and Jim Larus & Jeannette Wing for feedback

White Paper Process

- Late March 2012
 - CCC contacts coordinator & forms group
- April 2012
 - Brainstorm (meetings/online doc)
 - Read related docs (PCAST, NRC Game Over, ACAR1/2, ...)
 - Use online doc for intro & outline then parallel sections
 - Rotated authors to revise sections
- May 2012
 - Brainstorm list of researcher in/out of comp. architecture
 - Solicit researcher feedback/endorsement
 - Do distributed revision & redo of intro
 - Release May 25 to CCC & via email

Kudos to participants on executing on a tight timetable

\$15M NSF XPS 2/2013

Exploiting Parallelism and Scalability (XPS)

PROGRAM SOLICITATION

NSF 13-507



National Science Foundation

Directorate for Computer & Information Science & Engineering
Division of Computing and Communication Foundations
Division of Information & Intelligent Systems
Division of Computer and Network Systems

Office of Cyberinfrastructure

Full Proposal Deadline(s) (due by 5 p.m. proposer's local time):

February 20, 2013

Award Information

Anticipated Type of Award: Standard Grant or Continuing

Estimated Number of Awards: 20

Approximately 20 awards of up to \$750,000 for periods up to the availability of funds.

Anticipated Funding Amount: \$15,000,000

\$15,000,000 is anticipated to be awarded, subject to availability.

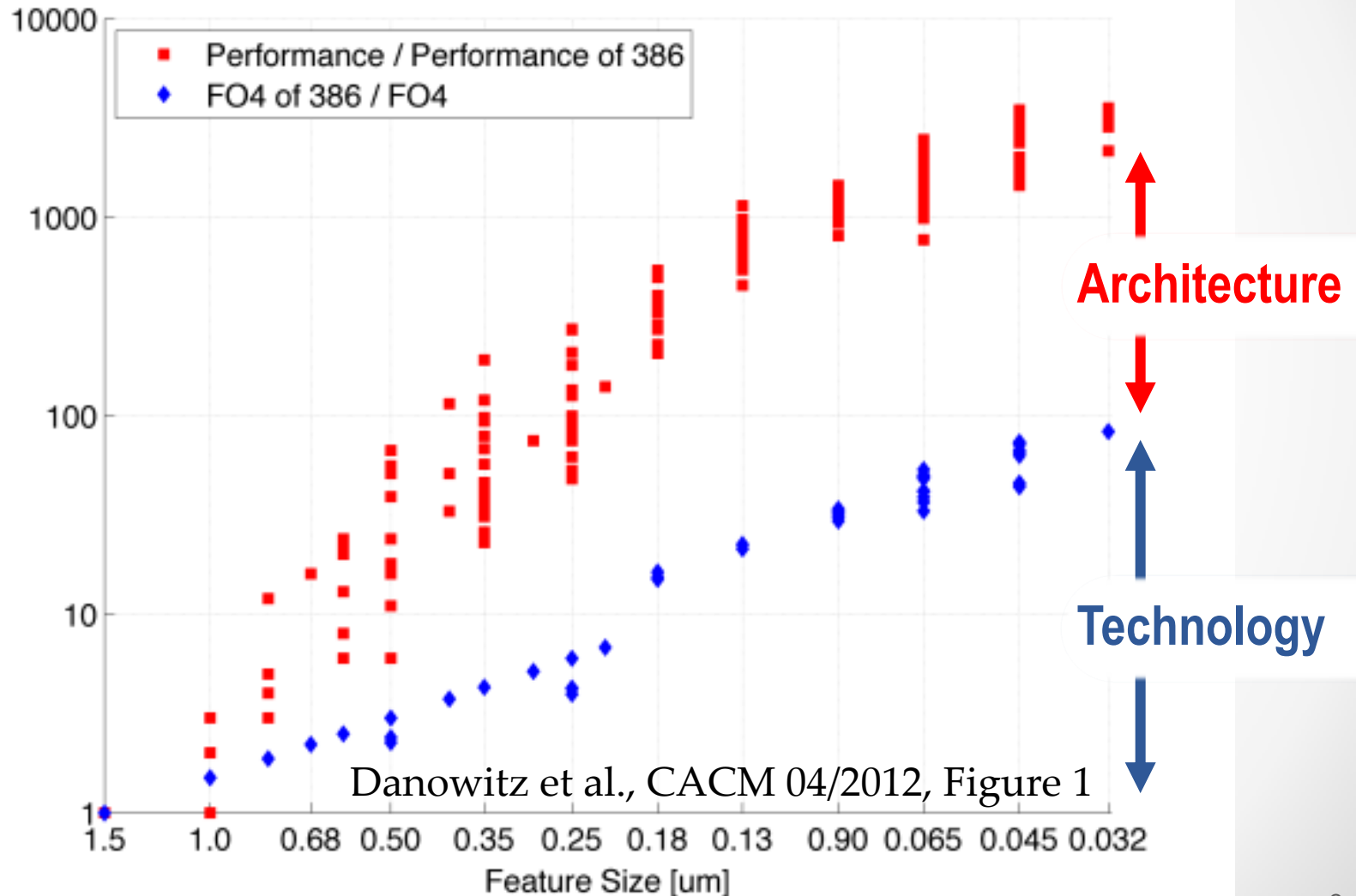
At the same time, a main driver of continued performance improvement is ending: semiconductor technology is facing fundamental physical limits. High processor performance has plateaued. Two recent reports, "21st Century Computer Architecture" commissioned by the Computing Community Consortium (<http://cra.org/ccc/docs/init/21stcenturyarchitecturewhitepaper.pdf>) and the 2011 NRC report on "The Future of Computing Performance: Game Over or Next Level?" (http://www.nap.edu/reports.php?record_id=12980) highlight this development and its impact on science, the economy, and society. The reports pose the question of how to enable the computational systems that will support emerging applications without the benefit of near-perfect performance scaling from hardware improvements. NSF's *Advanced Computing Infrastructure: Vision and Strategic Plan* (<http://www.nsf.gov/pubs/2012/nsf12051/nsf12051.pdf>) published in February 2012 describes strategies that address this challenge for NSF and the research community. The XPS program is part of the larger NSF CIF21 framework.

+ \$15M for 2/2014

20th Century ICT Set Up

- Information & Communication Technology (ICT) Has Changed Our World
 - <long list omitted>
- Required innovations in algorithms, applications, programming languages, ... , & system software
- Key (invisible) enablers (cost-)performance gains
 - Semiconductor technology (“Moore’s Law”)
 - Computer architecture (~80x per Danowitz et al.)

Enablers: Technology + Architecture



21st Century ICT Promises More



Data-centric personalized health care



Computation-driven scientific discovery



"You never call, and the federal government will back me up on that."

Human network analysis

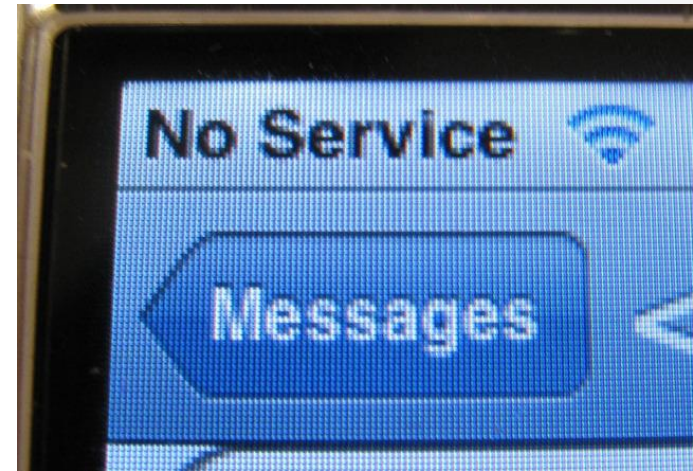


Much more: known & unknown

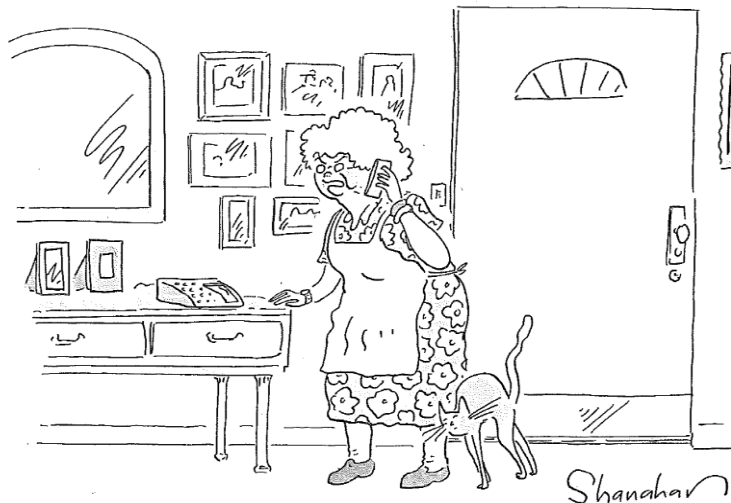
21st Century App Characteristics



BIG DATA



ALWAYS ONLINE



"You never call, and the federal government will back me up on that."

SECURE/PRIVATE



**Whither enablers of future
(cost-)performance gains?** • 10

Technology's Challenges 1/2

Late 20 th Century	The New Reality
Moore's Law — 2× transistors/chip	Transistor count still 2× BUT...
Dennard Scaling — ~constant power/chip	Gone. Can't repeatedly double power/chip

Classic CMOS Dennard Scaling: the Science behind Moore's Law

(Finding 2)

THE NATIONAL ACADEMIES

Source: Future of Computing Performance:
Game Over or Next Level?,
National Academy Press, 2011

Scaling:

Voltage: V/α

Oxide: t_{ox}/α

Wire width: W/α

Gate width: L/α

Diffusion: x_d/α

Substrate: αN_A

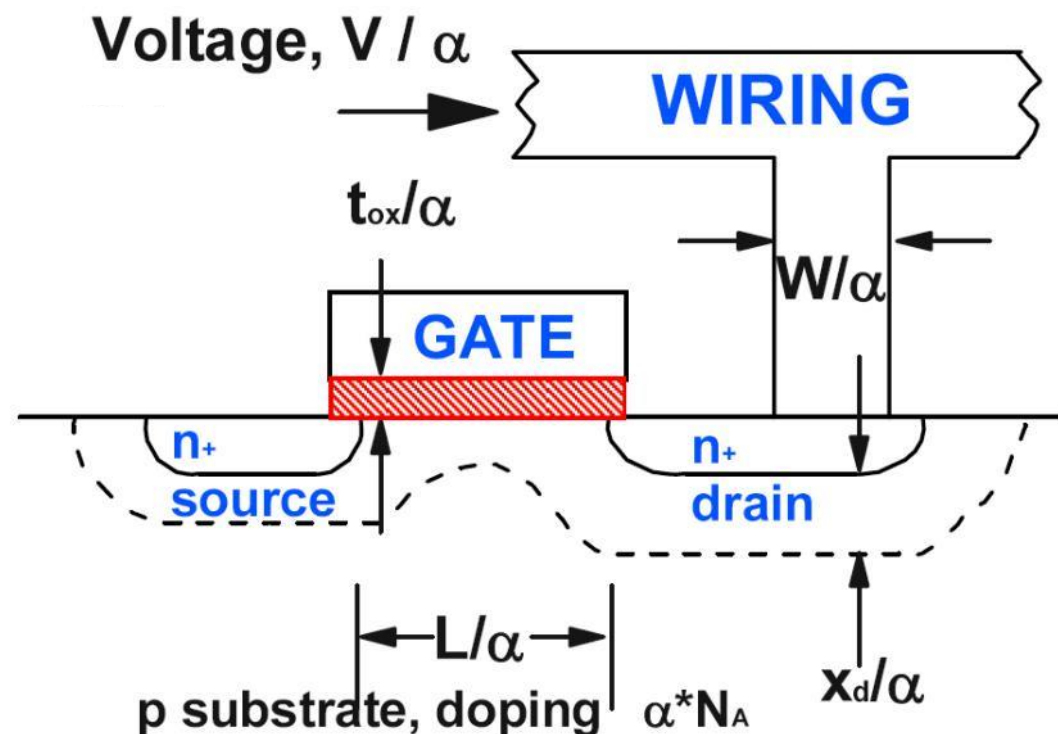
Results:

Higher Density: $\sim \alpha^2$

Higher Speed: $\sim \alpha$

Power/ckt: $1/\alpha^2$

Power Density: $\sim \text{Constant}$



R. H. Dennard et al.,
IEEE J. Solid State Circuits, (1974).

Post-classic CMOS Dennard Scaling

THE NATIONAL ACADEMIES

Post Dennard CMOS Scaling Rule

Scaling:

Voltage: ~~V/α~~ V

Oxide: t_{ox}/α

Wire width: W/α

Gate width: L/α

Diffusion: x_d/α

Substrate: αN_A

Results:

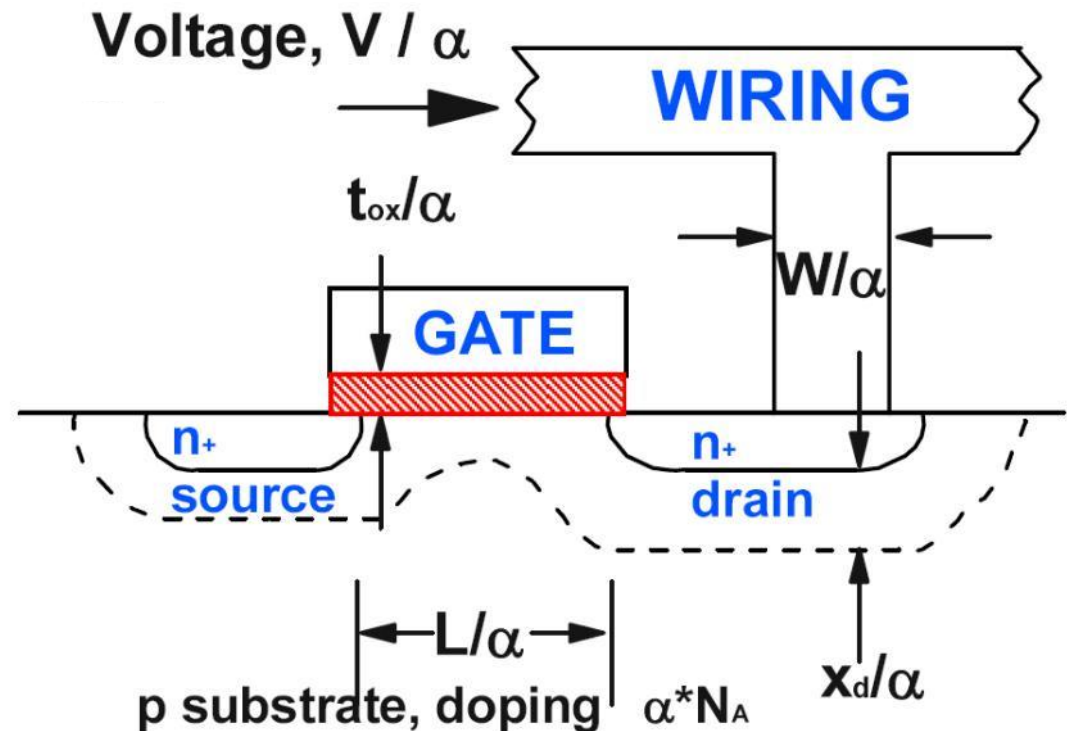
Higher Density: $\sim \alpha^2$

Higher Speed: $\sim \alpha$

Power/ckt: ~~$1/\alpha^2$~~ 1

Power Density: ~~$\sim \text{Constant}$~~ α^2

*Chips w/ higher power (no), smaller (⊗),
dark silicon (⊙), or other (?)*



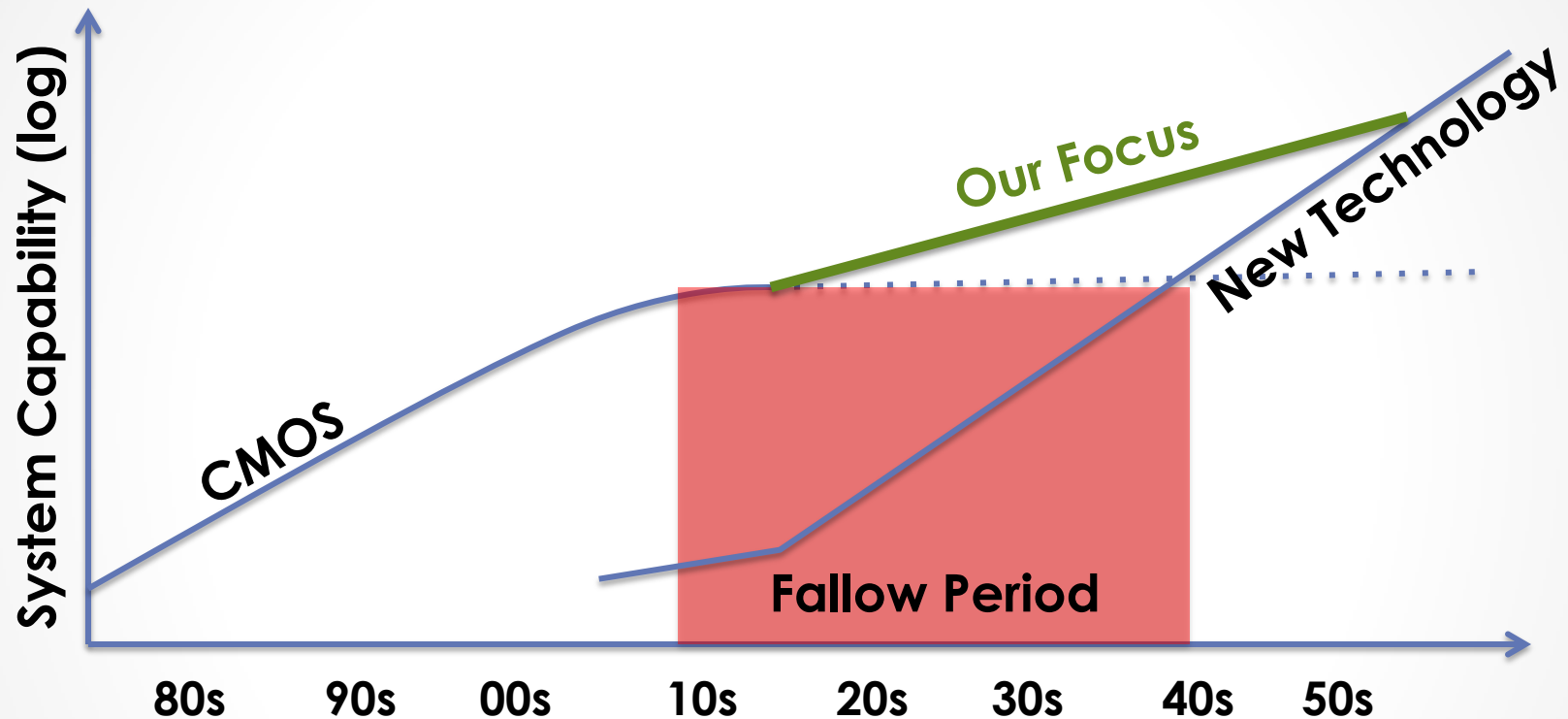
R. H. Dennard et al.,
IEEE J. Solid State Circuits, (1974).

Technology's Challenges 2/2

Late 20 th Century	The New Reality
Moore's Law — 2× transistors/chip	Transistor count still 2× BUT...
Dennard Scaling — ~constant power/chip	Gone. Can't repeatedly double power/chip
Modest (hidden) transistor unreliability	Increasing transistor unreliability can't be hidden
Focus on computation over communication	Communication (energy) more expensive than computation
1-time costs amortized via mass market	One-time cost much worse & want specialized platforms

How should architects step up as technology falters?

“Timeline” from DARPA ISAT



Source: Advancing Computer Systems without Technology Progress,
ISAT Outbrief (http://www.cs.wisc.edu/~markhill/papers/isat2012_ACSWTP.pdf)

Mark D. Hill and Christos Kozyrakis, DARPA/ISAT Workshop, March 26-27, 2012.

Approved for Public Release, Distribution Unlimited

The views expressed are those of the author and do not reflect the official policy or position of the
Department of Defense or the U.S. Government.

21st Century Comp Architecture

20th Century

Single-chip in
generic
computer

Performance
via invisible
instr.-level
parallelism

Predictable
technologies:
CMOS, DRAM,
& disks

21st Century





MORGAN & CLAYPOOL PUBLISHERS

The Datacenter as a Computer

*An Introduction to the Design
of Warehouse-Scale Machines
Second Edition*


Luiz André Barroso
Jimmy Clidaras
Urs Hölzle

**SYNTHESIS LECTURES ON
COMPUTER ARCHITECTURE**

Mark D. Hill, *Series Editor*

**Available for Free:
Search on
“Synthesis Lectures on
Computer Architecture”**

21st Century Comp Architecture

20 th Century	21 st Century	
Single-chip in generic computer	Architecture as Infrastructure: Spanning sensors to clouds Performance plus security, privacy, availability, programmability, ...	
Performance via invisible instr.-level parallelism		
Predictable technologies: CMOS, DRAM, & disks		

21st Century Comp Architecture

20th Century

Single-chip in
generic
computer

Performance
via invisible
instr.-level
parallelism

Predictable
technologies:
CMOS, DRAM,
& disks



21st Century Comp Architecture

20th Century

Single-chip in
generic
computer

Performance
via invisible
instr.-level
parallelism

Predictable
technologies:
CMOS, DRAM,
& disks

21st Century



21st Century Comp Architecture

20 th Century	21 st Century	
Single-chip in stand-alone computer	Architecture as Infrastructure: Spanning sensors to clouds Performance plus security, privacy, availability, programmability, ...	Cross-Cutting: Break current layers with new interfaces
Performance via invisible instr.-level parallelism	Energy First <ul style="list-style-type: none"> • Parallelism • Specialization • Cross-layer design 	
Predictable technologies: CMOS, DRAM, & disks	New technologies (non-volatile memory, near-threshold, 3D, photonics, ...) Rethink: memory & storage, reliability, communication	

What Research Exactly?

- Research areas in white paper (& backup slides)
 1. Architecture as Infrastructure: Spanning Sensors to Clouds
 2. Energy First
 3. Technology Impacts on Architecture
 4. Cross-Cutting Issues & Interfaces
- Much more research developed by future PIs!
- Example from our work in 2nd part of talk [ISCA 2013]
 - Cloud workloads(memcached) use vast memory (100 GB to TB) wasting up to 50% execution time
 - A cross-cutting OS-HW change eliminates this waste

Pre-Competitive Research Justified

- **Retain (cost-)performance enabler to ICT revolution**
- Successful companies cannot do this by themselves
 - Lack needed long-term focus
 - Don't want to pay for what benefits all
 - Resist transcending interfaces that define their products
- Corroborates
 - Future of Computing Performance: Game Over or Next Level?, National Academy Press, 2011
 - DARPA/ISAT Workshop Advancing Computer Systems without Technology Progress with outbrief
http://www.cs.wisc.edu/~markhill/papers/isat2012_ACSWTP.pdf

21st Century Computer Architecture

A CCC community white paper

<http://cra.org/ccc/docs/init/21stcenturyarchitecturewhitepaper.pdf>

- Participants & Process
- Information & Commun. Tech's Impact
- Semiconductor Technology's Challenges
- Computer Architecture's Future
- Pre-Competitive Research Justified

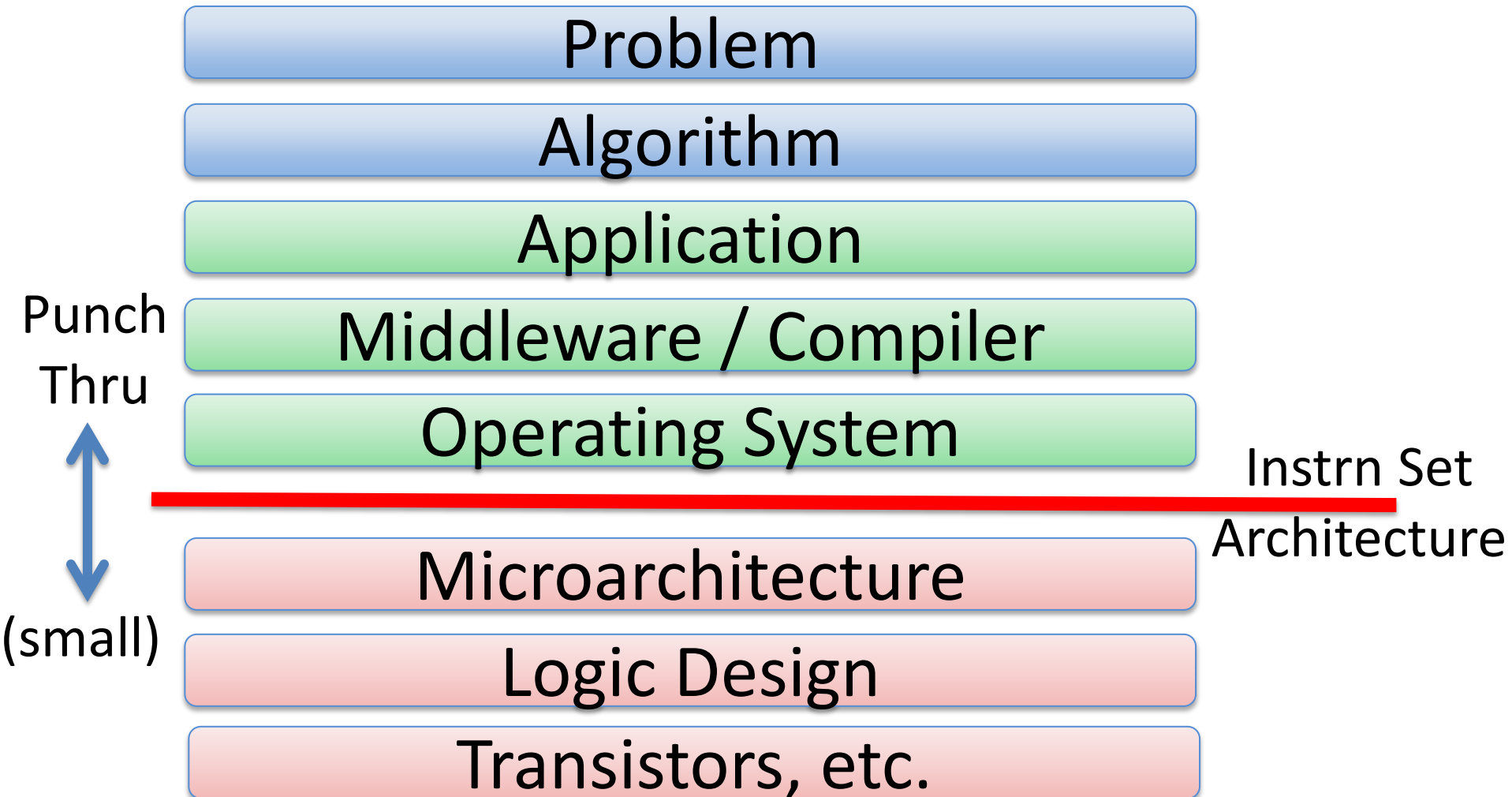


Mark D. Hill, Univ. of Wisconsin-Madison 10/2013 @ NSF CISE Distinguished Lecture

A talk in 2 ¼ parts:

- 21st Century Computer Architecture (whitepaper)
- **Efficient Virtual Memory for Big Memory Servers**
- Opportunistic Virtual Cache (short, optional)

A View of Computer Layers





Efficient Virtual Memory for Big Memory Servers

Arkaprava Basu, Jayneel Gandhi, Jichuan Chang*,
Mark D. Hill, Michael M. Swift

* HP Labs

Q: “Virtual Memory was invented in a time of scarcity. Is it still good idea?”

--- Charles Thacker, 2010 Turing Award Lecture

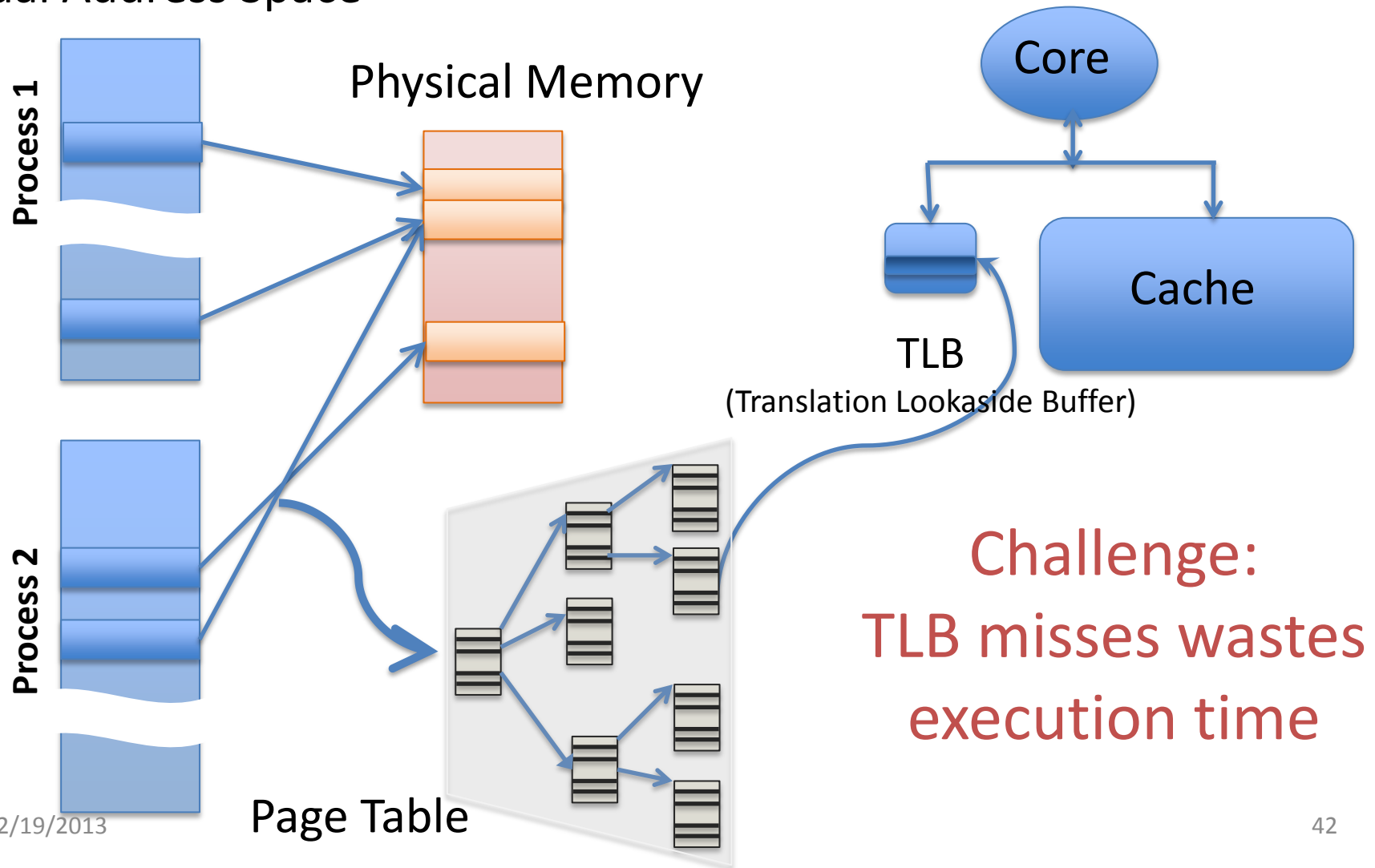
A: As we see it, OFTEN but not ALWAYS.

Executive Summary

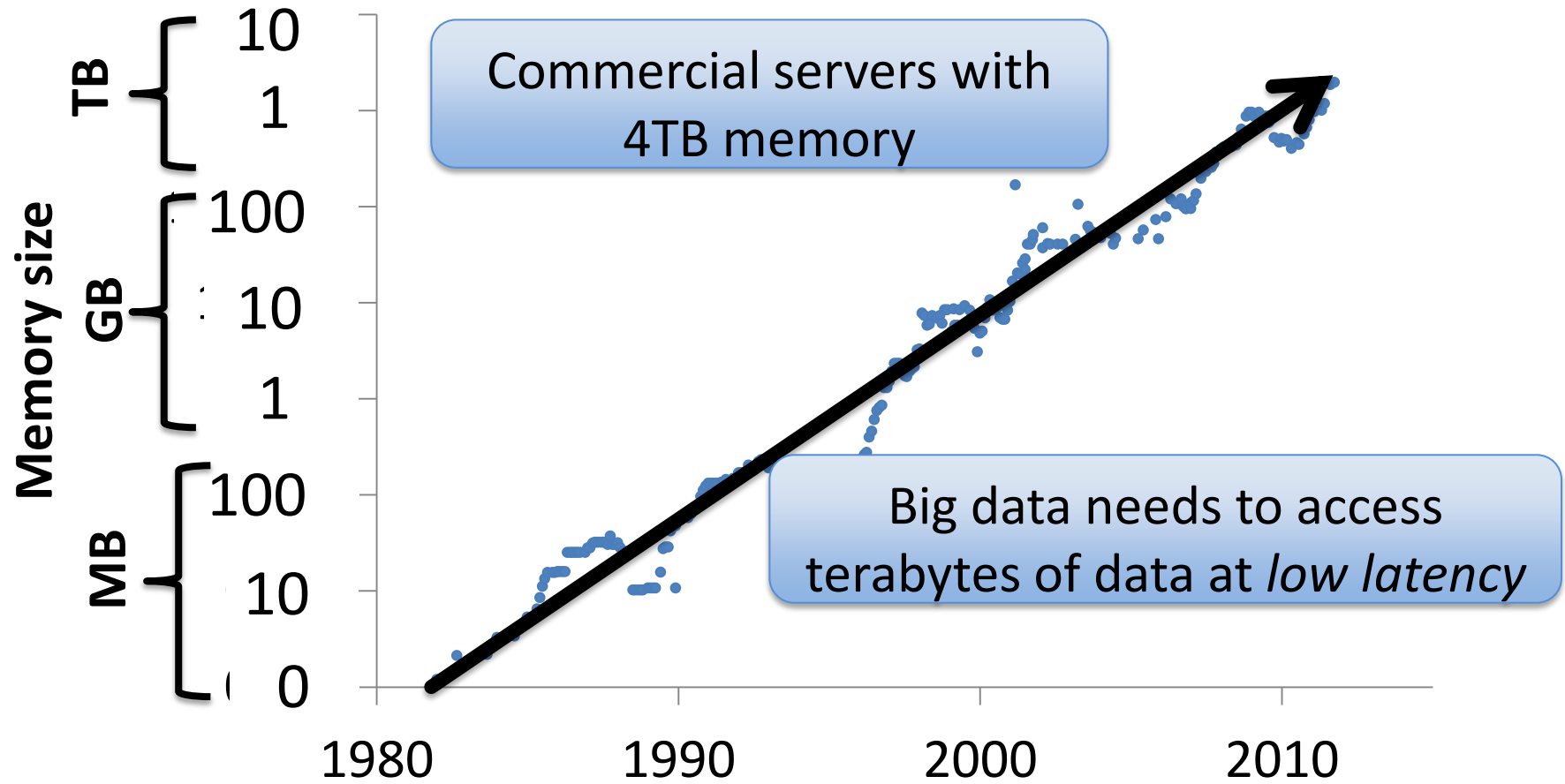
- Big memory workloads important
 - graph analysis, memcached, databases
- Our analysis:
 - TLB misses burns up to 51% execution cycles
 - Paging not needed for almost all of their memory
- Our proposal: **Direct Segments**
 - Paged virtual memory *where needed*
 - Segmentation (No TLB miss) *where possible*
- Direct Segment often eliminates 99% DTLB misses

Virtual Memory Refresher

Virtual Address Space



Memory capacity for \$10,000*



*Inflation-adjusted 2011 USD, from: jcmit.com

TLB is Less Effective

- TLB sizes hardly scaled

Year	1999	2001	2008	2012
L1-DTLB entries	72 (Pent. III)	64 (Pent. 4)	96 (Nehalem)	100 (Ivy Bridge)



- Low access locality of server workloads

[Ramcloud'10, Nanostore'11]

↑ Memory Size + → TLB size + ↓ Low locality
→ ↑ TLB miss latency overhead

Experimental Setup

- Experiments on Intel Xeon (Sandy Bridge) x86-64
 - Page sizes: 4KB (Default), 2MB, 1GB

	4 KB	2 MB	1GB
L1 DTLB	64 entry, 4-way	32 entry, 4-way	4 entry, fully assoc.
L2 DTLB	512 entry, 4-way		

- 96GB installed physical memory
- Methodology: Use hardware performance counter

Big Memory Workloads

graph500[?]

memcached^{??}

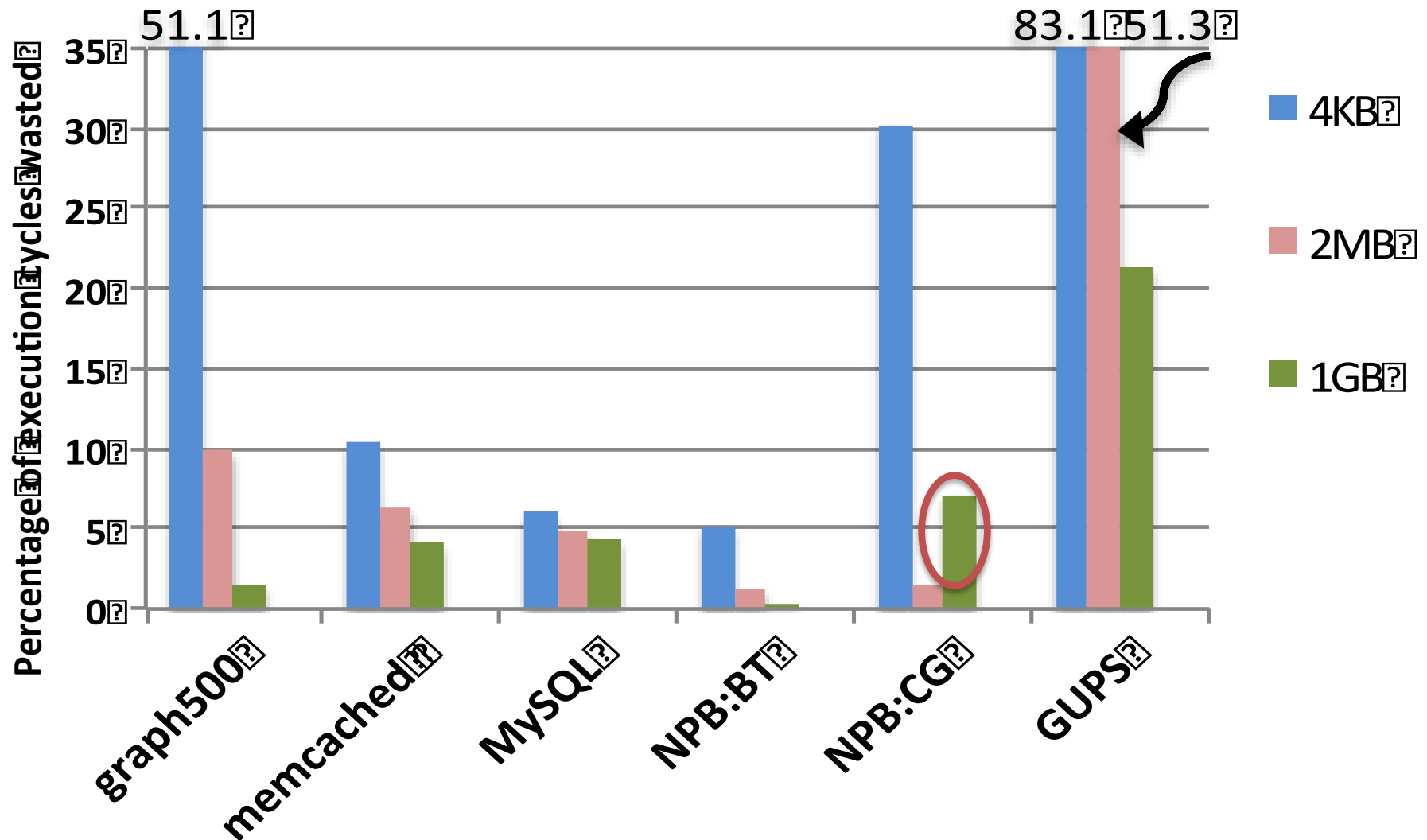
MySQL[?]

NPB:BT[?]

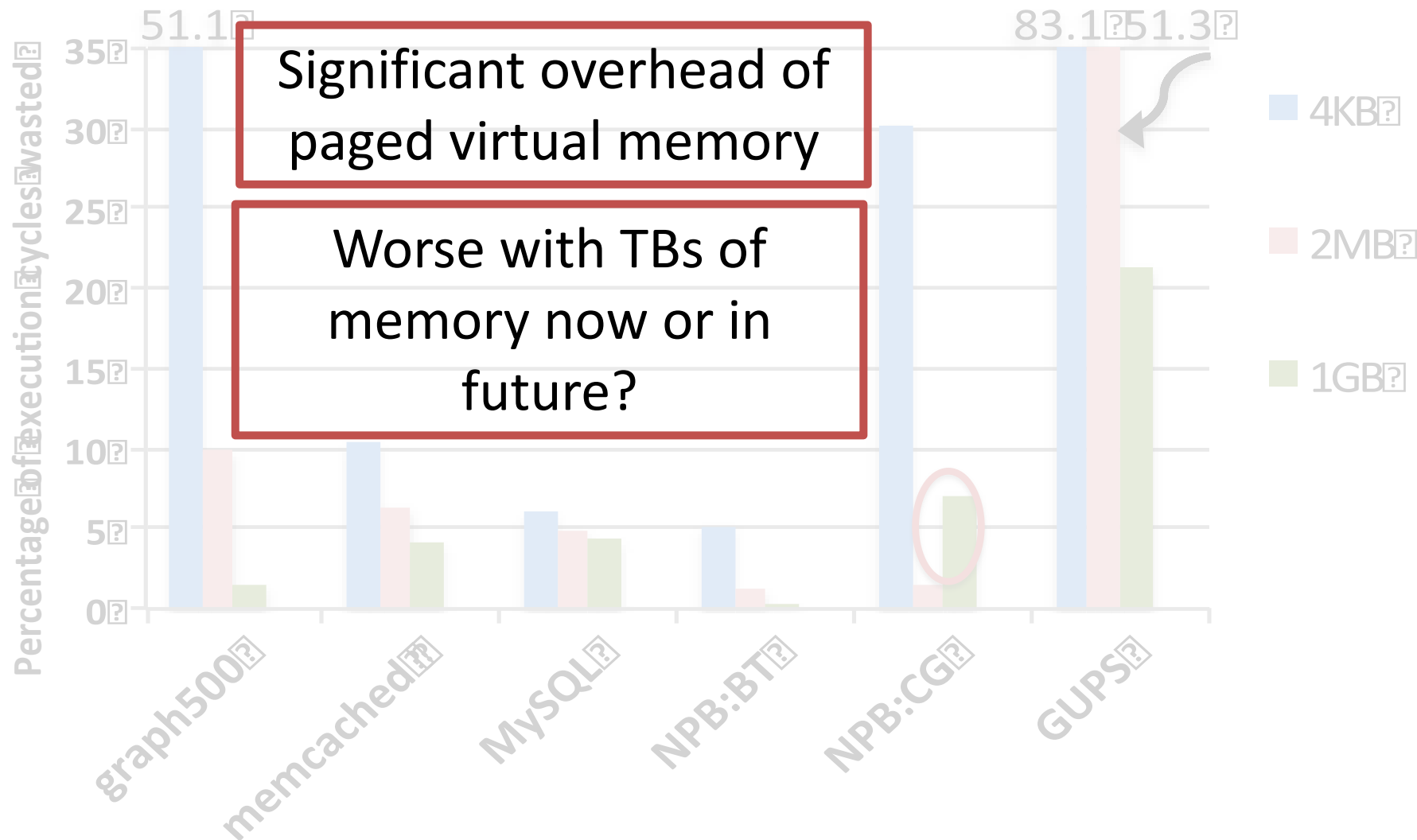
NPB:CG[?]

GUPS[?]

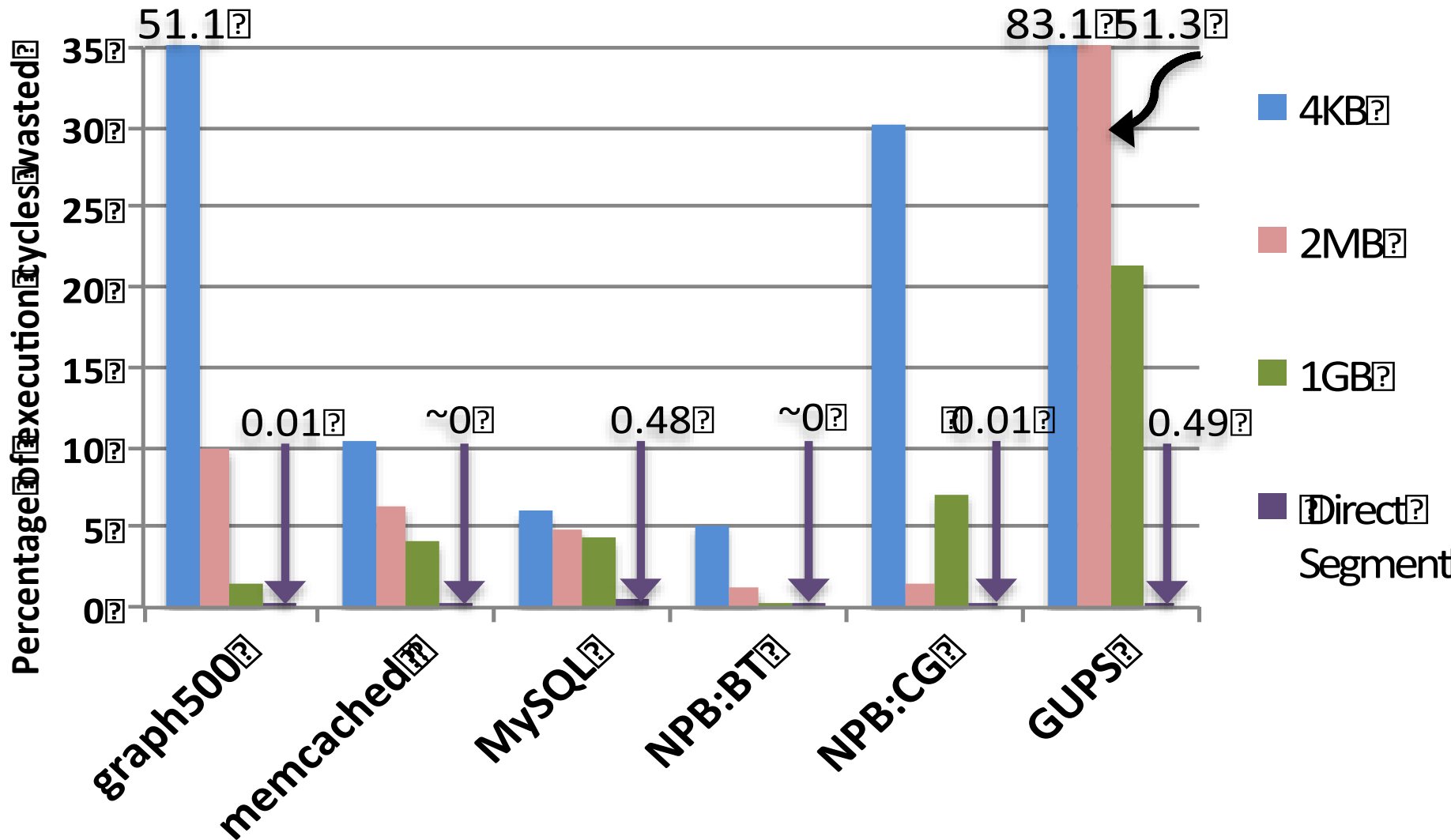
Execution Time Overhead: TLB Misses




Execution Time Overhead: TLB Misses



Execution Time Overhead: TLB Misses

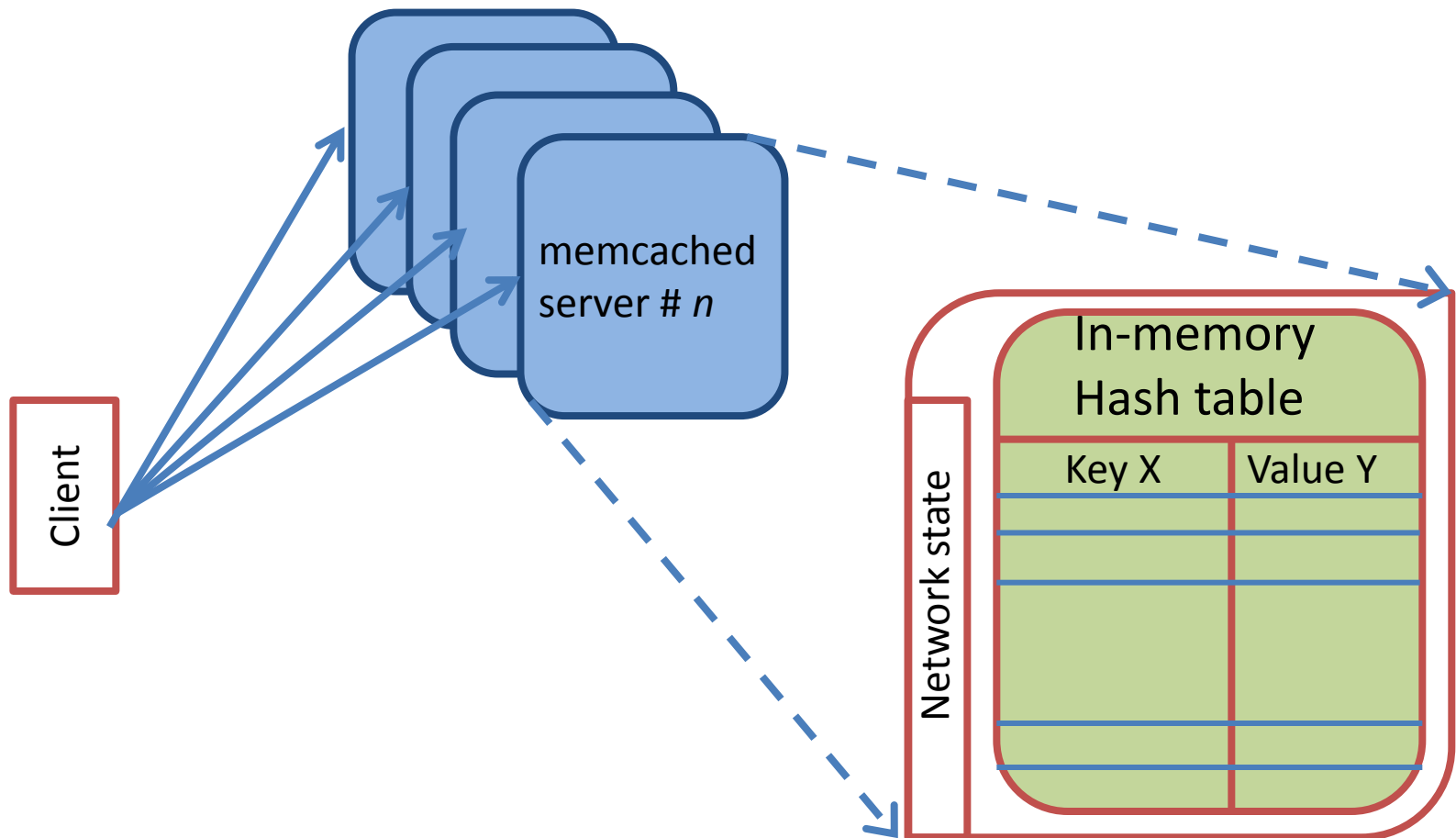


Roadmap

- Introduction and Motivation
- Analysis: Big memory workloads 
- Design: Direct Segment
- Evaluation
- Summary

How is Paged Virtual Memory used?

An example: memcached servers

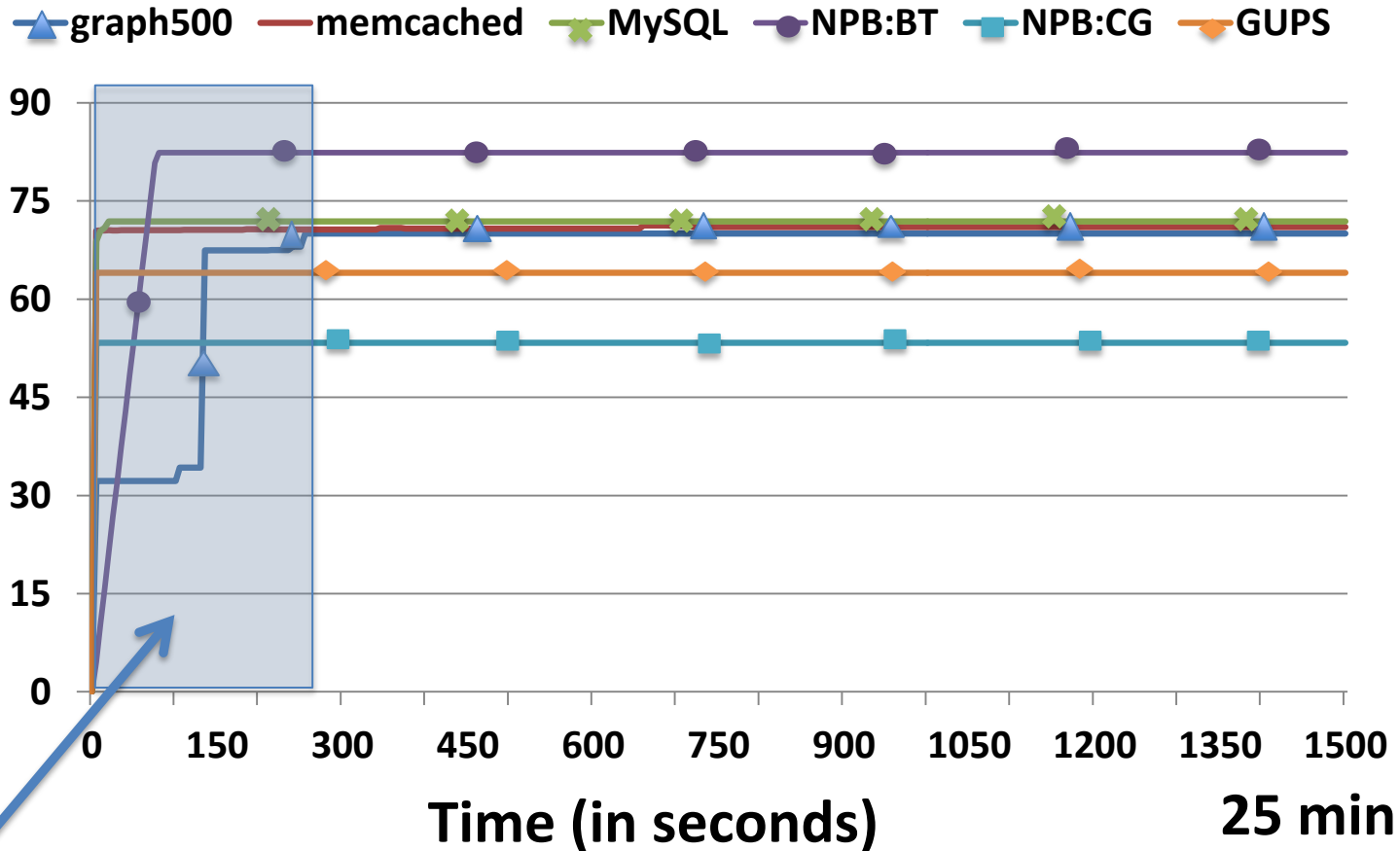


Big Memory Workloads' Use of Paging

Paged VM Feature	Our Analysis	Implication
Swapping	~0 swapping	Not essential
Per-page protection	~99% pages read-write	Overkill
Fragmentation reduction	Little OS-visible fragmentation (next slide)	Per-page (re)-allocation less important

Memory Allocation Over Time

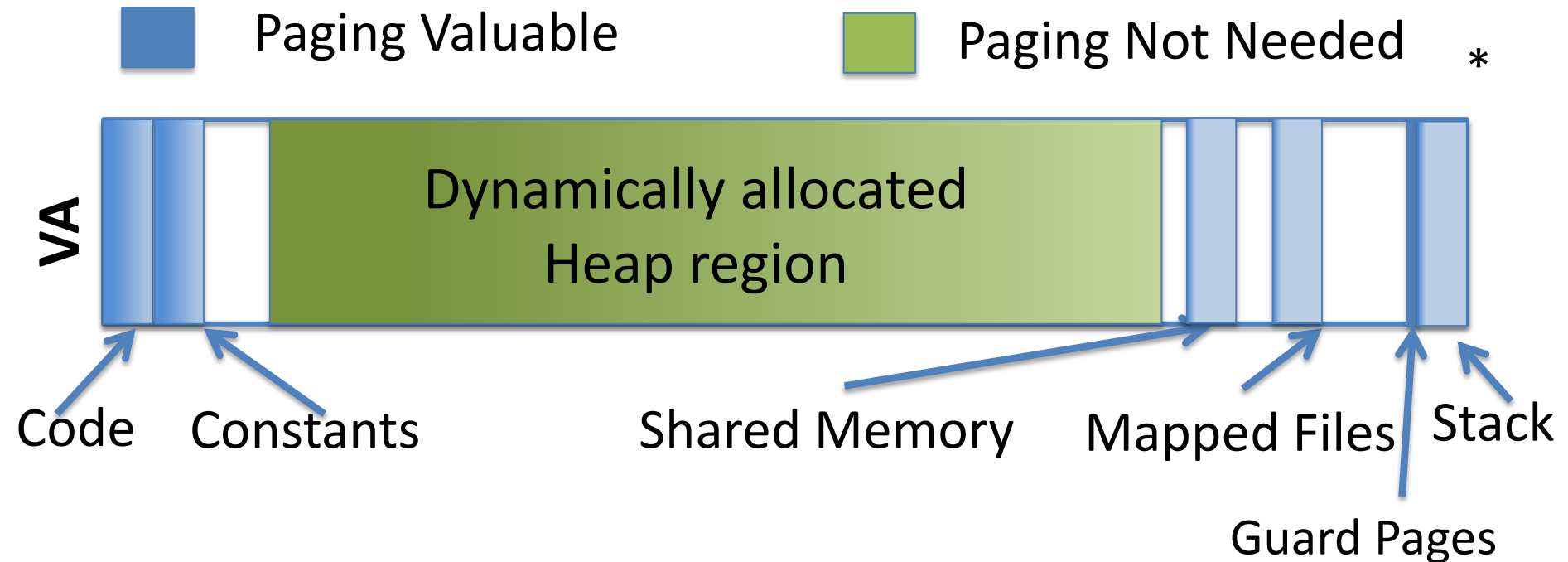
Allocated Memory (in GB)



Warm-up

Most of the memory allocated early


Where Paged Virtual Memory Needed?



Paged VM **not** needed for **MOST** memory

* Not to scale

Roadmap

- Introduction and Motivation
- Analysis: Big Memory Workloads
- **Design: Direct Segment** 
 - Idea
 - Hardware
 - Software
- Evaluation
- Summary

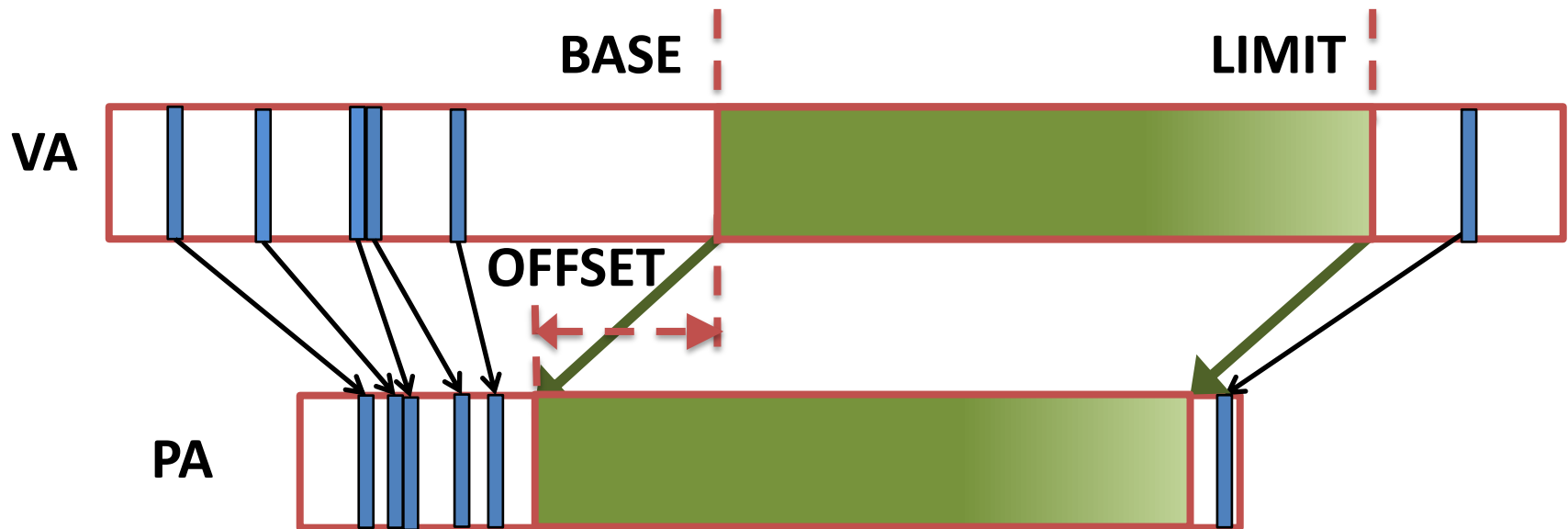
Idea: Two Types of Address Translation

- A Conventional paging
 - All features of paging
 - All cost of address translation
 - B Simple address translation
 - Protection but **NO** (easy) swapping
 - **NO** TLB miss
- OS/Application decides where to use which
[=> Paging features where needed]

Hardware: Direct Segment

1 Conventional Paging

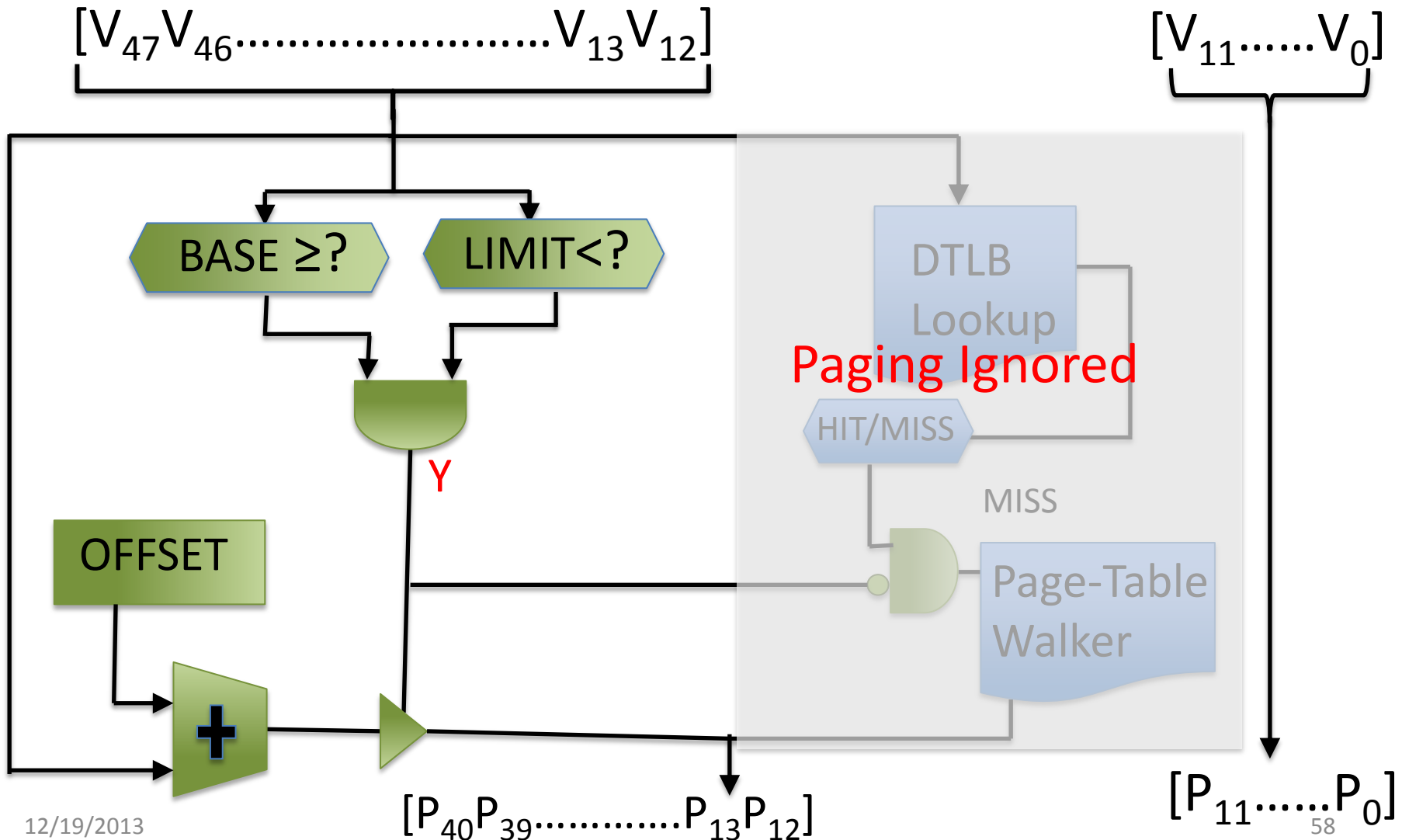
2 Direct Segment



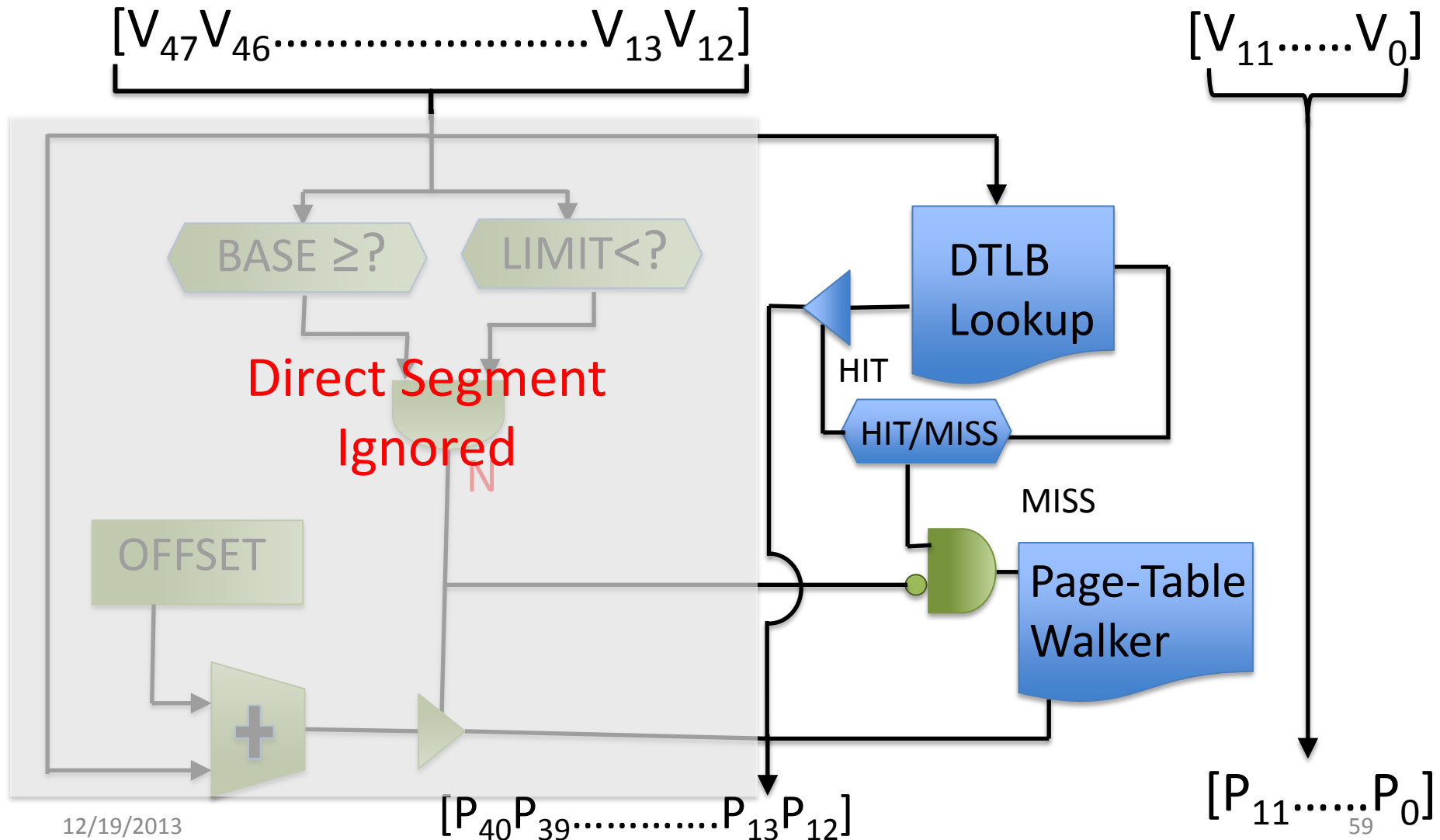
Why Direct Segment?

- Matches big memory workload needs
- NO TLB lookups => NO TLB Misses

H/W: Translation with Direct Segment

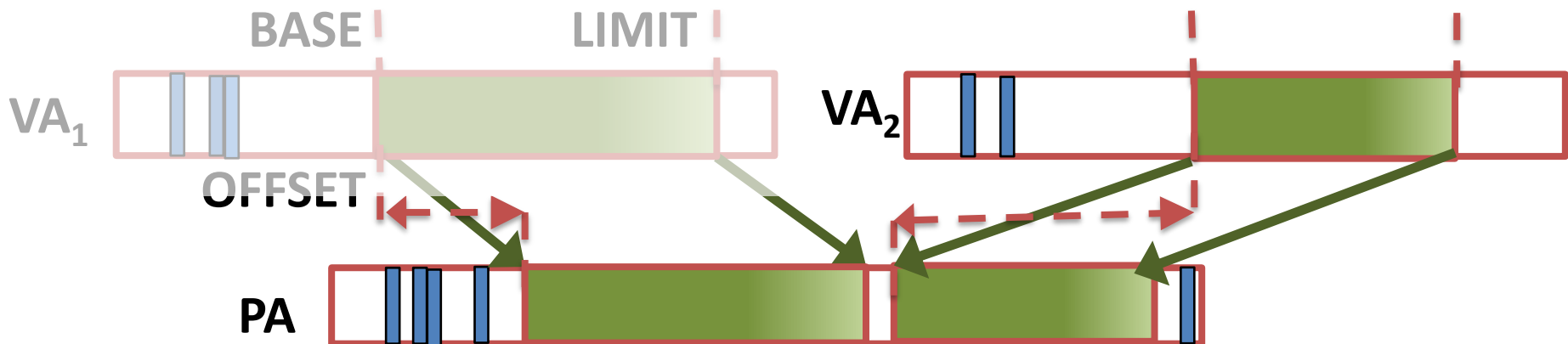


H/W: Translation with Direct Segment



S/W: **1** Setup Direct Segment Registers

- Calculate register values for processes
 - BASE = Start VA of Direct Segment
 - LIMIT = End VA of Direct Segment
 - OFFSET = BASE – Start PA of Direct Segment
- Save and restore register values

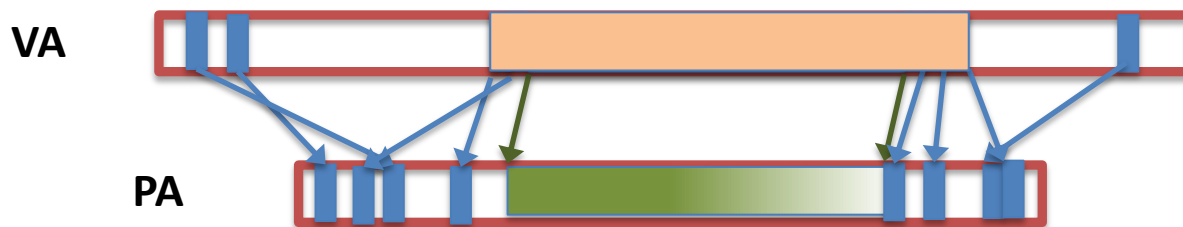


S/W Provision Physical Memory

- Create contiguous physical memory
 - Reserve at startup
 - Big memory workloads cognizant of memory needs
 - e.g., memcached's object cache size
 - Memory compaction
 - Latency insignificant for long running jobs
 - 10GB of contiguous memory in < 3 sec
 - 1% speedup => 25 mins break even for 50GB compaction

S/W: 3 Abstraction for Direct Segment

- Primary Region
 - Contiguous VIRTUAL address not needing paging
 - Hopefully backed by Direct Segment
 - But all/part can use base/large/huge pages



- What allocated in primary region?
 - All anonymous read-write memory allocations
 - Or only on explicit request (e.g., *mmap* flag)

Segmentation Not New

ISA/Machine	Address Translation
Multics	Segmentation on top of Paging
Burroughs B5000	Segmentation without Paging
UltraSPARC	Paging
X86 (32 bit)	Segmentation on top of Paging
ARM	Paging
PowerPC	Segmentation on top of Paging
Alpha	Paging
X86-64	Paging only (mostly)


Direct Segment:

NOT on top of paging.

NOT to replace paging.

NO two-dimensional address space: keeps linear address space.

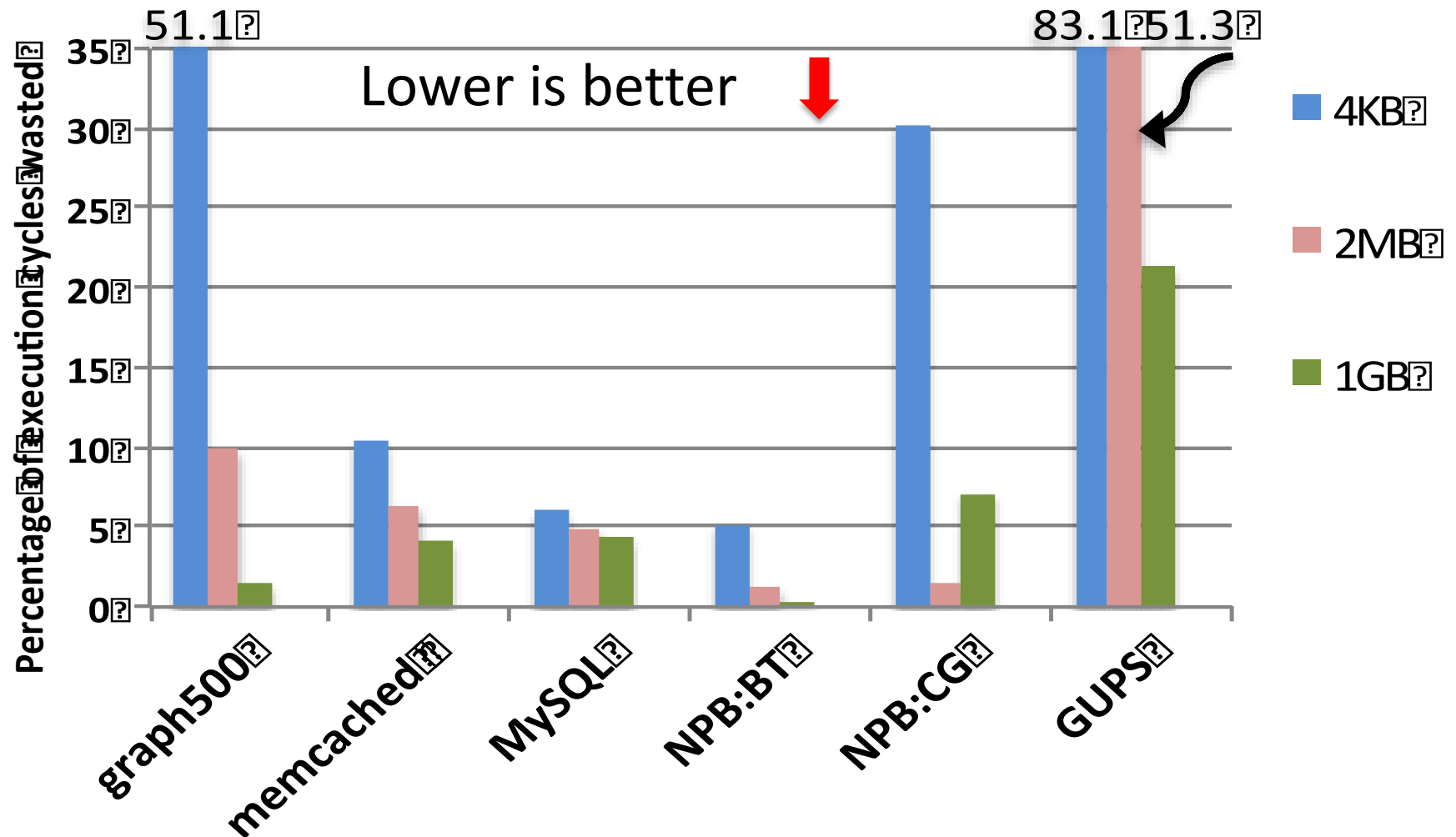
Roadmap

- Introduction and Motivation
- Analysis: Big Memory Workloads
- Design: Direct Segment
- **Evaluation** 
 - Methodology
 - Results
- **Summary**

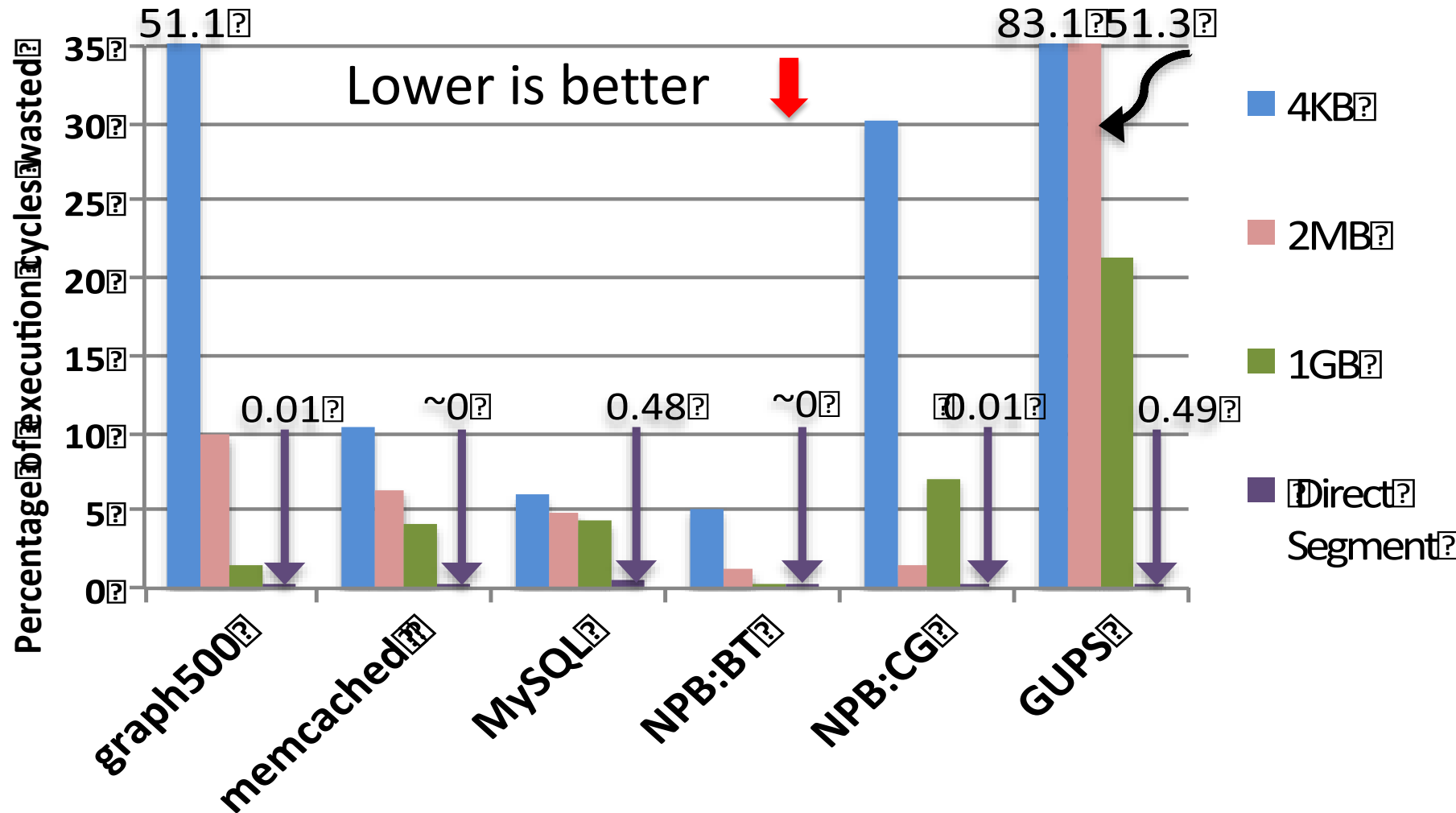
Methodology

- Primary region implemented in Linux 2.6.32
- Estimate performance of non-existent direct-segment
 - Get fraction of TLB misses to direct-segment memory
 - Estimate performance gain with linear model
- Prototype simplifications (design more general)
 - One process uses direct segment
 - Reserve physical memory at start up
 - Allocate r/w anonymous memory to primary region

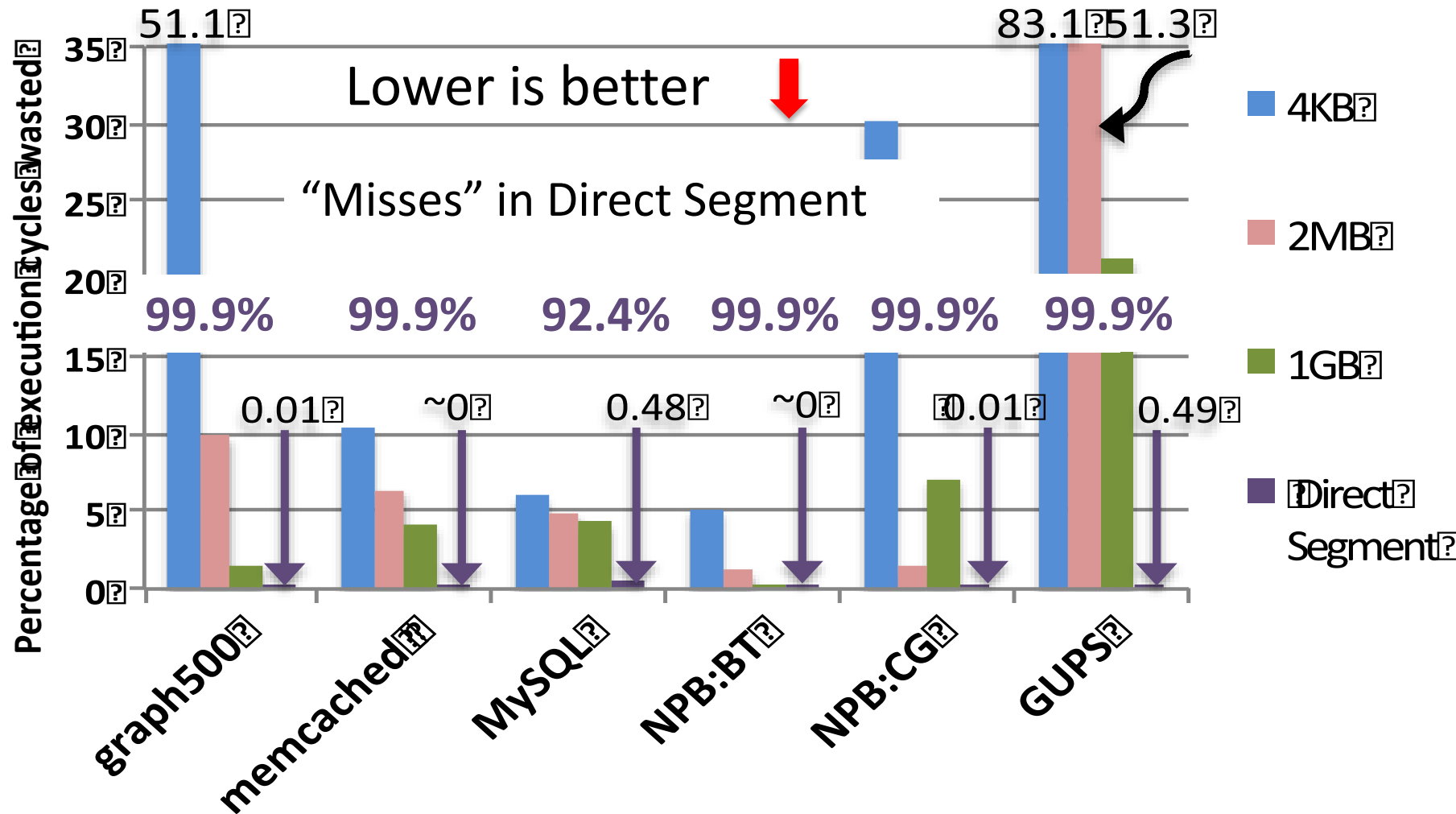
Execution Time Overhead: TLB Misses



Execution Time Overhead: TLB Misses



Execution Time Overhead: TLB Misses



(Some) Limitations

- Does not (yet) work with Virtual Machines
 - Can be extended but memory overcommit challenging
- Less suitable for sparse virtual address space
- One direct segment
 - Our workloads did not justify more

Summary

- Big memory workloads
 - Incurs high TLB miss cost
 - Paging not needed for almost all memory
- Our proposal: **Direct Segment**
 - Paged virtual memory *where needed*
 - Segmentation (NO TLB miss) *where possible*
- Bonus: Whither TLB Energy?



Mark D. Hill, Univ. of Wisconsin-Madison 10/2013 @ NSF CISE Distinguished Lecture

A talk in 2 ¼ parts:

- 21st Century Computer Architecture (whitepaper)
- Efficient Virtual Memory for Big Memory Servers
- **Opportunistic Virtual Cache (short, optional)**



Reducing Memory Reference Energy with *Opportunistic* Virtual Caching

Arkaprava Basu

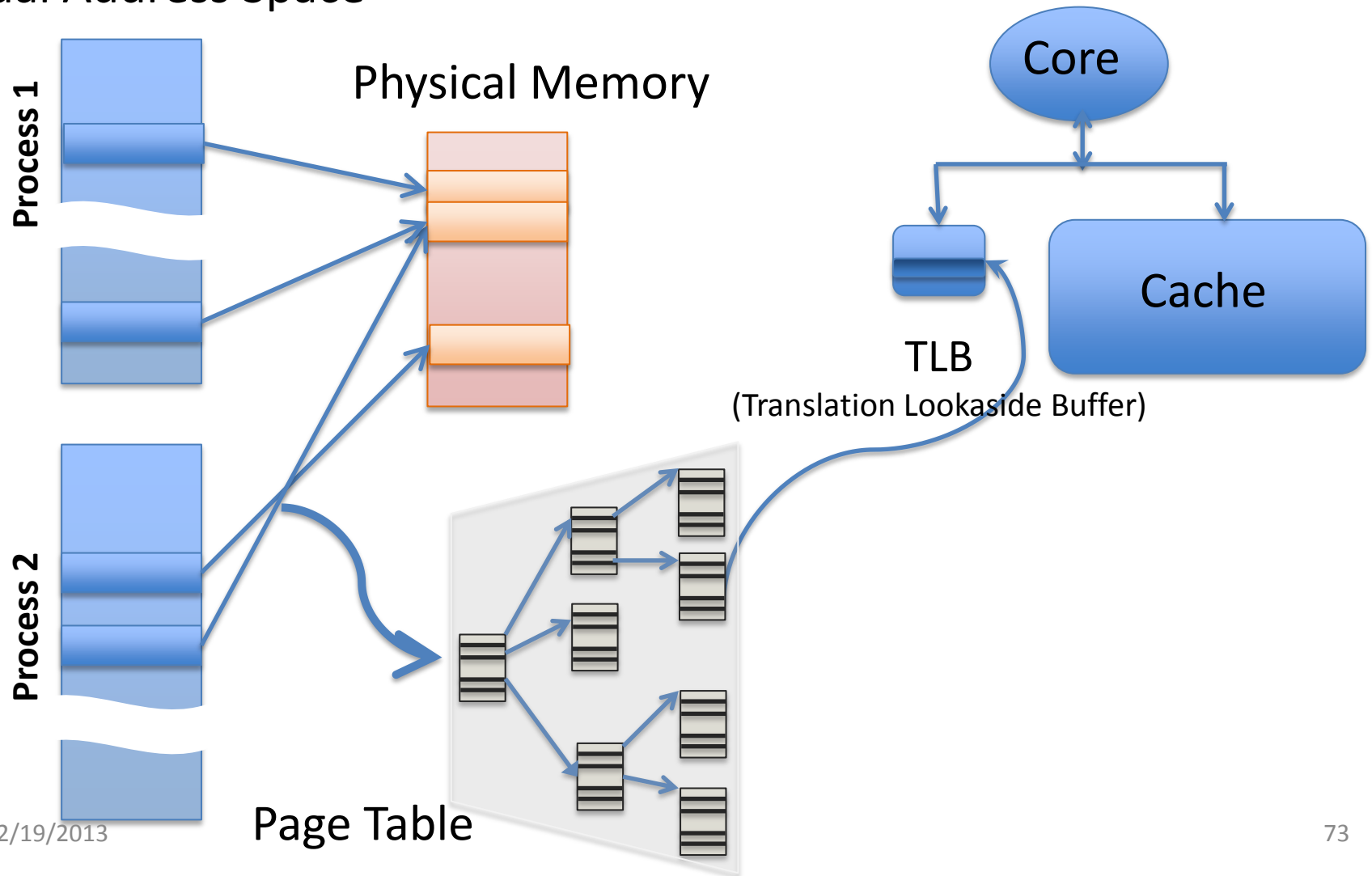
Mark D. Hill

Michael M. Swift

University of Wisconsin-Madison

Virtual Memory Refresher

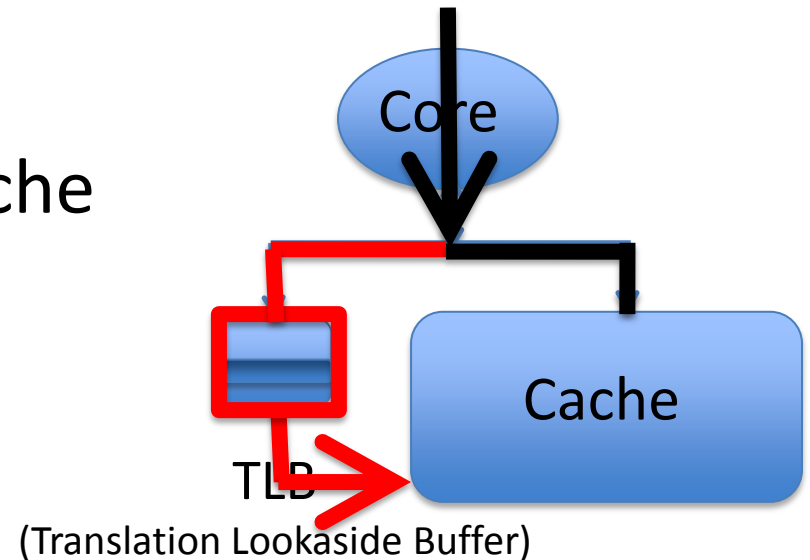
Virtual Address Space



Standard Physical Cache

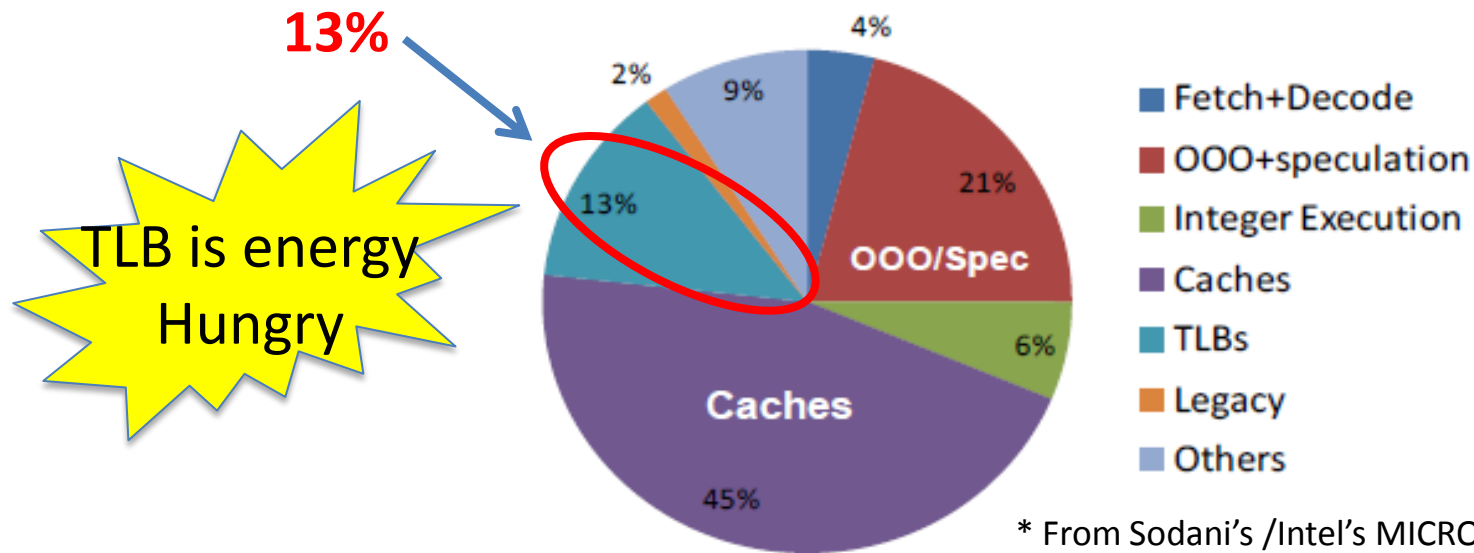
TLB access on **every** memory reference in parallel w/ L1 cache

- Enables physical address hit/miss check
- Hides TLB **latency**



BUT DOES NOT HIDE TLB ENERGY!

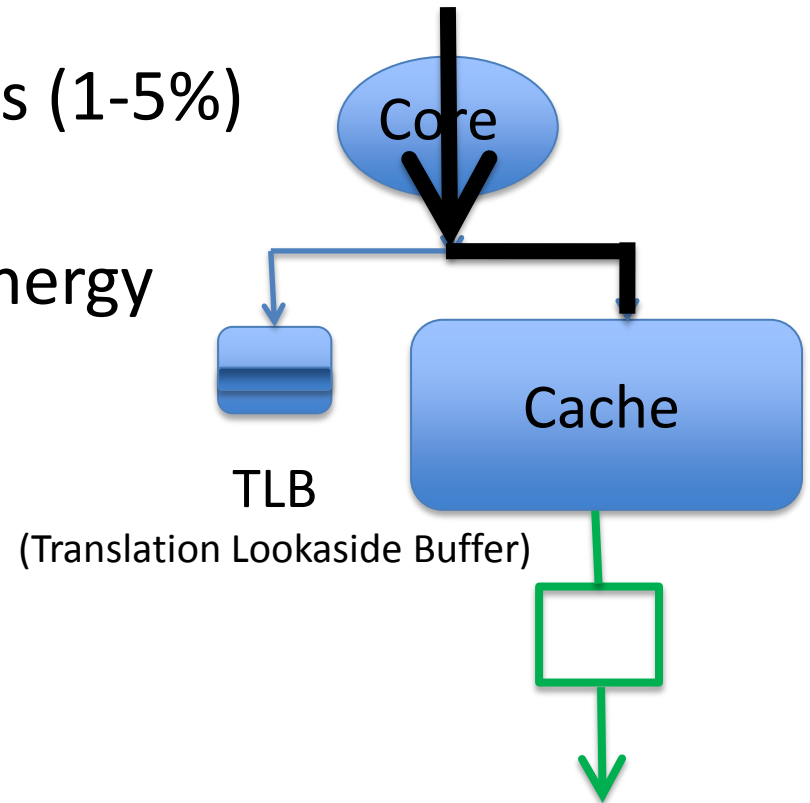
A 2nd Virtual Memory Problem: Energy



How can we avoid this energy?

Old Idea: Virtual Cache

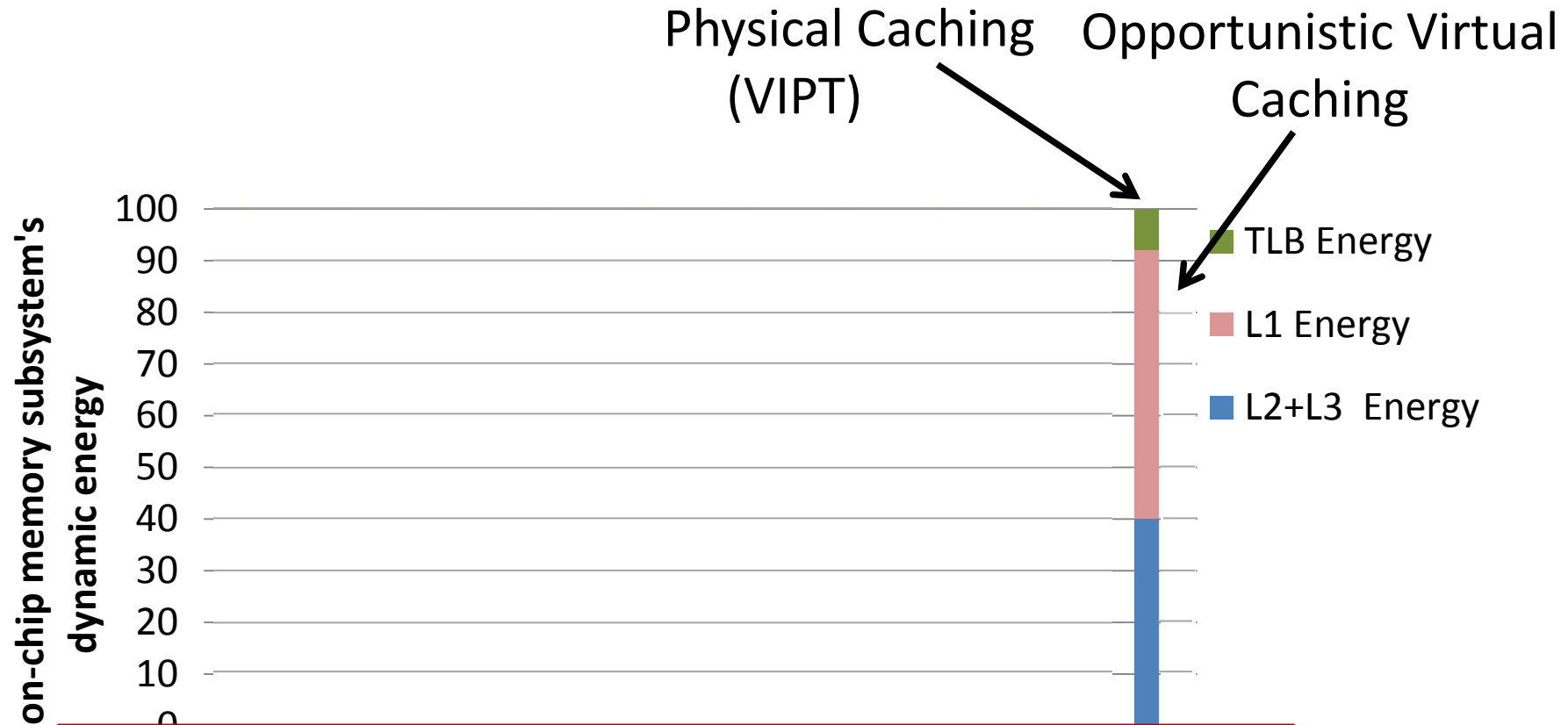
- TLB access after L1 cache miss (1-5%)
- Greatly reduces TLB access energy
- But breaks compatibility
 - e.g., read-write synonyms
- OUR ANALYSIS:
Read-write synonyms rare



Opportunistic Virtual Cache

- A New Hardware Cache [ISCA 2012]
 - Caches w/ virtual addresses for almost-all blocks
 - Caches w/ physical addresses only for compatibility
- OS decides (cross-layer design)
- Saves energy
 - Avoids most TLB lookups
 - L1 cache can use lower associativity (subtle)

Dynamic Energy Savings?



On average ~20% of the on-chip memory subsystem's dynamic energy is reduced



Mark D. Hill, Univ. of Wisconsin-Madison 10/2013 @ NSF CISE Distinguished Lecture

A talk in 2 ¼ parts:

- 21st Century Computer Architecture (whitepaper)
- Efficient Virtual Memory for Big Memory Servers
- **Opportunistic Virtual Cache (short, optional)**

My Students, Colleagues, & I Thank NSF

NSF has been my principal support since 1988

1. Presidential Young Investigator Award [1989]

...

19. WasteNot: Streamlining Virtual Memory for Modern Systems [2013]

Made tax dollars go further w/ donations

ATI, AMD, Bell Labs, Compaq, Cray, DEC, Google, HAL, HP, IBM, Intel, Lucent, Microsoft, Oracle, Qualcomm, Silicon Graphics, Sun, & Texas Inst.

Thank You!

The Design of Secondary Caches, 1989-1991, Two-year National Science Foundation grant (CCR-8902536).

Presidential Young Investigator Award: Cache Memory Design, 1989-1994, Five-year grant and matching funds from the National Science Foundation (MIPS-8957278). PYI matching funds have been donated by A.T.&T. Bell Laboratories, Cray Research, Digital Equipment Corporation, Sun Microsystems, and Texas Instruments.

A High Speed Data Acquisition System for Research in Parallel Computing, 1990-1991, National Science Foundation equipment grant (CDA-8920777), partially matched by the University of Wisconsin Graduate School and A.T.&T. Bell Laboratories, Co-PI with Mary Vernon.

PRISM: A Laboratory for Research in Future High-Performance Parallel Computing, 1991-1995, National Science Foundation institutional infrastructure grant (CDA-9024618), partially matched by the University of Wisconsin Graduate School, Co-PI with Michael Carey, Charles Dyer, Robert Meyer, Barton Miller, and Mary Vernon (project coordinator).

Cooperative Shared Memory and the Wisconsin Wind Tunnel, 1993-1996, National Science Foundation MIPS Experimental Systems (MIPS-9225097), Co-PI with James Larus and David Wood.

Cooperative Shared Memory and the Wisconsin Wind Tunnel (Supplement), 1994-1996, National Science Foundation MIPS Experimental Systems (MIPS-9225097), Co-PI with James Larus and David Wood.

Tornado: Fine-Grain Distributed Shared Memory for SMP Clusters, 1996-1999, National Science Foundation MIPS Experimental Systems (MIPS-9625558), Co-PI with David Wood, James Larus, and Pei Cao.

MIDSHIP: Managing Image Data for Scalable High Performance, 1996-2001, National Science Foundation institutional infrastructure grant (CDA-9623632), partially matched by the University of Wisconsin Graduate School, Project co-director with Jeffery Naughton and co-PI with nine other faculty.

Multifacet: Exploiting Prediction and Speculation in Multiprocessor Memory Systems, 1999-2002, National Science Foundation CISE Experimental Partnerships (EIA-9971256), Co-PI with David A. Wood with Investigators Pei Cao, Anne Condon, and Charles Fischer.

Exploiting the Critical Path in the Design and Performance Analysis of Modern Processors, 2001-2004, National Science Foundation (CCR-0105721), Co-PI with Rastislav Bodik.

SafetyNet: Synergistic Support for Availability, Designability, Programmability, & Performance, 9/2002-8/2007, National Science Foundation CISE ITR (EIA/CNS-0205286), Co-PI with David A. Wood with Investigator is Rastislav Bodik.

Advanced Architectures and Technologies for Chip Multiprocessors, 9/2003-8/2008, National Science Foundation CISE ITR (CCR-0324878), Co-PI with David A. Wood.

CRI: MASSIV Cluster for Designing Chip Multiprocessors, 6/2006-5/2010, National Science Foundation CNS Computing Research Infrastructure (CNS-0551401), Co-PIs Gurindar S. Sohi and David A. Wood.

CSR—AES: Deconstructing Transactional Memory: System Support for Robust Concurrent Programming, 7/2007-6/2010, National Science Foundation (CNS-0720565), Co-PIs Michael M. Swift and David A. Wood.

SHF:Small: Managing Non-Determinism in Multithreaded Software and Hardware Multithreaded Record, Replay, and Execution, 8/2009-7/2012, National Science Foundation (CCF-0916725), Co-PI David A. Wood.

SHF:Small: Power Husbanding via Architectural Techniques (PHAT), 8/2010-7/2013, National Science Foundation (CCF-1017650), Co-PI with David A. Wood PI.

CSR: Small: Codesign of Accelerator Interface Software and Hardware, 8/2011-7/2014, National Science Foundation (CNS-1117280), Co-PI with David A. Wood PI and Michael Swift co-PI.

SHF:Small: Energy-Optimized Memory Hierarchies, 8/2012-7/2015, National Science Foundation (CCF-1218323), Co-PI with David A. Wood PI.

CSR:Medium: WasteNot: Streamlining Virtual Memory for Modern Systems, 9/2013-8/2016, National Science Foundation (CNS-1302260), Co-PI with David A. Wood; PI is Michael M. Swift.