Processors Have Evolved
Is Memory Stuck?

Al Davis

University of Utah

&

Hewlett-Packard Laboratories

Motivation

- **Some things are clear**
  - **historical observation**
    - » **"Memory is the bottleneck …" John Von Neumann, 1945**
      - **this has been perpetually true**
  - **today's situation: processors have changed**
    - » **multi- and many-core processor era in play**
    - » **memory controllers have moved onto the processor die**
      - **multiple memory controllers connected – e.g. Nehalem QPI**
    - » **core count/socket going up**
      - **some pundits predict the "new Moore's Law"**
    - » **socket pin count & pin bandwidth growing much more slowly (ITRS)**
      - **signal integrity and high-speed (SERDES) I/O power problems**
    - » **power & heat are fundamental barriers to performance improvement**
- **Memory evolution has been too slow for the HPC community**
  - **Von Neumann's prediction remains**
  - **question is what should change?**
    - » **what follows is a brief look at almost everything**

Memory and Storage

- **Goal = more, more, more!!**
  - **ideally we want**
    - » **reduced latency (some can be hidden with SMT tactics)**
    - » **improved bandwidth commensurate with processor performance growth**
      - **without a commensurate increase in energy consumption**
    - » **increased capacity per socket**
    - » **reduced cost of everything**
- **Reality**
  - **lots of constraints – pins, pin bandwidth, thermals, power**
    - » **the goal list = conflicting constraints**
  - **BUT – new technology is emerging**
    - » **NVM, photonics, and architecture improvements**
    - » **question is how to use these to achieve MOST of the goals**
      - **reduced cost expectations will likely need to be compromised**
- **Focus today**
  - **memory controllers, DRAM & interfaces, NVM today and future technology**

Baselines

Operation	Energy (pJ)
64b Floating FMA (2 ops)	100
64b Integer Add	1
Write 64b DFF	0.5
Read 64b Register (64 x 32 bank)	3.5
Read 64b RAM (64 x 2K)	25
Read tags (24 x 2K)	8
Move 64b 1mm	6
Move 64b 20mm	120
Move 64b off chip	256
Read 64b from DRAM	2000

Source: Dally 2009

Operation	Energy (pJ)	DP FLOPs	Insts*
I$ Fetch	33	0.67	2.0
Register Access (3W)	10.5	0.2	0.6
Access 3 D$	100	2	6
Access 3 L2 D$	460	9	27
Access 3 off chip	762	15	45
Access 3 from DRAM	6000	120	360

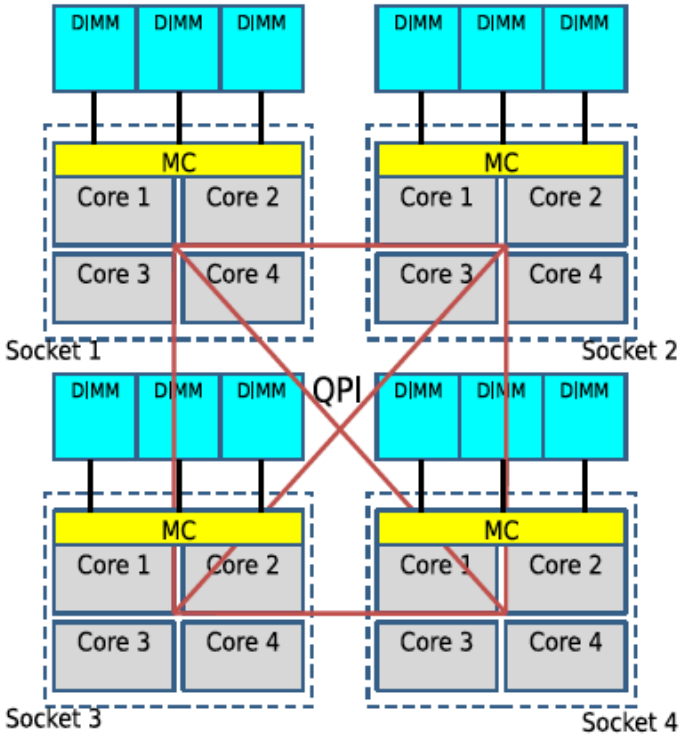
Memory Hasn't Changed Much

- **DRAM device optimization**
  - **industry focus: minimize cost/bit & high volume markets**
    - » **persistent problem for the low-volume HPC community**
  - **slowly evolving interface standards: JEDEC, RAMBUS**
- **Significant change CMP memory access patterns**
  - **access pattern is increasingly random**
    - » **DRAM's optimized for row-locality 4-8KB row buffers**
      - **wasted energy due to "over fetch" for 64B $_line**
      - **open row scheduling becomes dubious**
      - **server vendors moving to default closed-row policies**
        - **e.g. IBM**
      - **latency hit w/ current interfaces for closed row access**
    - » **it's only going to get worse as core counts grow**
  - **signal integrity issues limit capacity & per pin bandwidth**
    - » **FB and BoB interfaces – latency hit**
    - » **Inphi's iMB (isolation mem. buff) → RL-DIMMs**

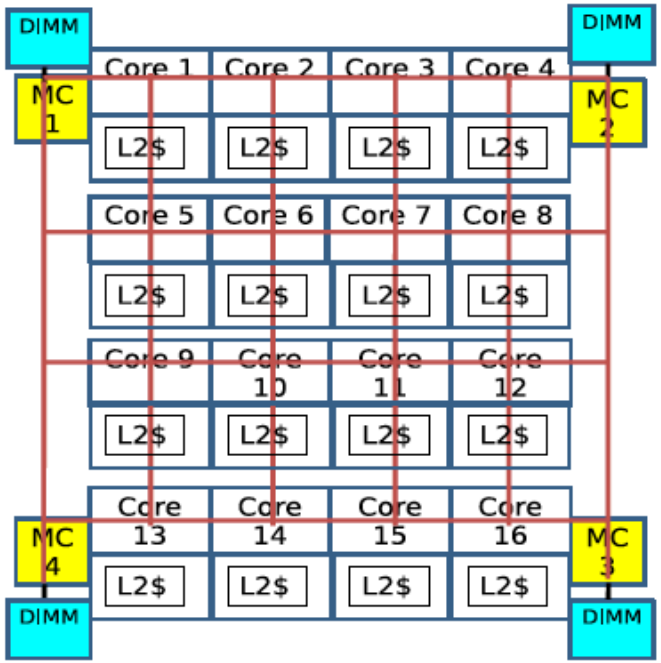
Main Memory Bandwidth

- How much do you need for a balanced system?
  - NSF Blue-Ribbon Advisory panel report in 2003 and a Gordon Bell et. al article in IEEE Computer, Jan. 2006 are in rough agreement
    - » 1 Byte/Flop main memory capacity
    - » 1 Byte/Flop/s main memory bandwidth
  - useful as a guideline BUT note
    - » existing supercomputer systems don't meet this metric
  - difference however between HPC & "THE CLOUD"??
    - » the Byte/Flop metric has been called conservative for the commercial side
    - » right choice based on application mix
- Power can't be ignored however
  - PetaFlop machine → PetaByte/s to the main memory
    - » n pico-joules through NOC, MC, off-chip to DRAM, DRAM and back → n WATTS – n in hundreds now - VERY SCARY!!

Changing Memory Controller Landscape



Today's Intel Nehalem
(AMD similar w/ HT)



Likely Future

MC Managed Timing Parameters

Parameter	Description
tAL	added latency to column accesses for posted CAS
tBURST	data burst duration on the data bus
tCAS	interval between CAS and start of data return
tCCD	column command delay - determined by internal burst timing
tCMD	time command is on bus from MC to device
tCWD	column write delay, CAS write to write data on the bus from the MC
tFAW	rolling temporal window for how long four banks can remain active
tOST	interval to switch ODT control from rank to rank
tRAS	row access command to data restore interval
tRC	interval between accesses to different rows in same bank = tRAS+tRP
tRCD	interval between row access and data ready at sense amps
tRFC	interval between refresh and activation commands
tRP	interval for DRAM array to be precharged for another row access
tRRD	interval between two row activation commands to same DRAM device
tRTP	interval between a read and a precharge command
tRTRS	rank to rank switching time
tWR	write recovery time - interval between end of write data burst and a precharge command
tWTR	interval between end of write data burst and start of a column read command

Complex!!!

Access Dependent Timing Equations

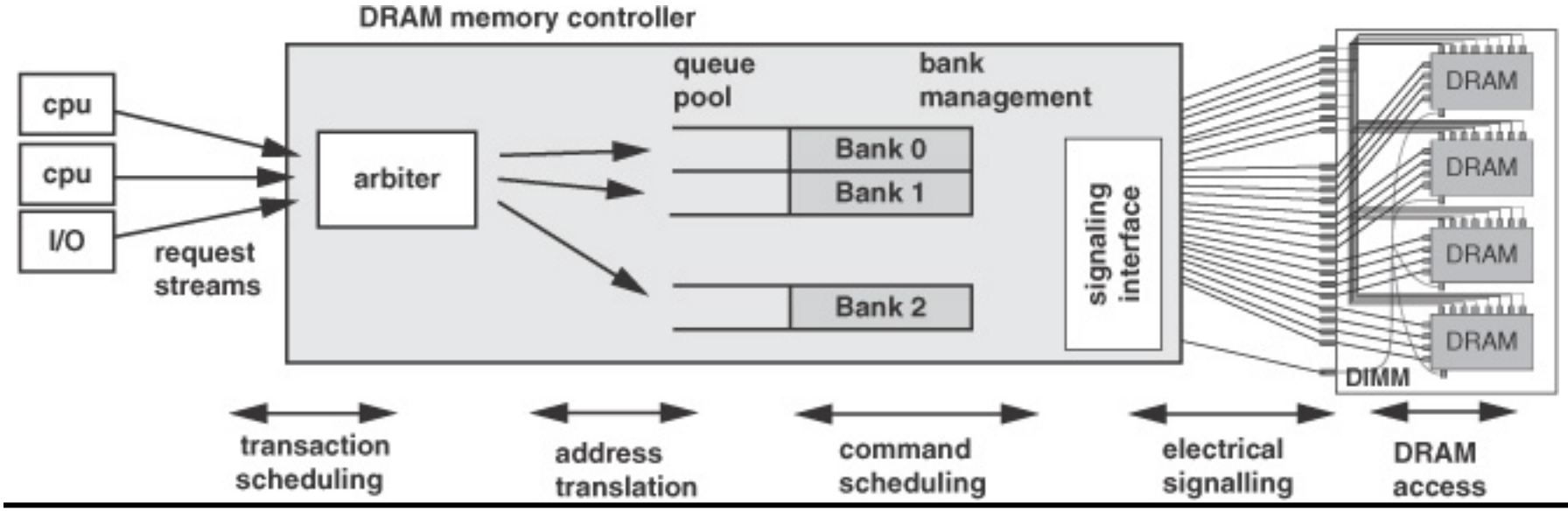
A=row access
R=col_rd
W=col_wr
P=precharge
F=Refresh
s=same
d=different
a=any

Prev	Next	Rank	Bank	Min. Timing	Notes
A	A	s	s	tRC	
A	A	s	d	tRRD	plus tFAW for 5th RAS same rank
P	A	s	d	tRP	
F	A	s	s	tRFC	
A	R	s	s	tRCD-tAL	tAL=0 unless posted CAS
R	R	s	a	Max(tBURST, tCCD)	tBURST of previous CAS, same rank
R	R	d	a	tBURST+tRTRS	tBURST prev. CAS diff. rank
W	R	s	a	tCWD+tBURST+tWTR	tBURST prev CASW same rank
W	R	d	a	tCWD+tBURST+tRTRS-tCAS	tBURST prev CASW diff rank
A	W	s	s	tRCD-tAL	
R	W	a	a	tCAS+tBURST+tRTRS-tCWD	tBURST prev. CAS any rank
W	W	s	a	Max(tBURST, tCCD)	tBURST prev CASW same rank
W	W	d	a	tBURST+tOST	tBURST prev CASW diff rank
A	P	s	s	tRAS	
R	P	s	s	tAL+tBURST+ tRTP-tCCD	tBURST of previous CAS, same rank
W	P	s	s	tAL+tCWD+ tBURST+tWR	tBURST prev CASW same rank
F	F	s	a	tRFC	
P	F	s	a	tRFC	

Eye chart
Details less important

Basic MC Components

- **Note**
  - **as memory access cost increases w.r.t. compute on CPU's**
    - » **combining transaction and command scheduling is important**
  - **address translation targets rank and bank**
    - » **transaction turned into a series of DRAM commands**
      - **optimization options occur with interleaved transactions**
        - **while still respecting device timing restrictions**
      - **goal: maximize data bus utilization**



Ideas for Memory Controller Improvement

Micro-pages (ASPLOS 2010)

Predictor Based Row Policy Management

Multiple MC/Socket issues (PACT 2010)

DRAM Access Inefficiencies - I

- **Over fetch due to large row-buffers.**
  - **8 KB read into row buffer for a 64 byte cache line.**
  - **Row-buffer utilization for a single request < 1%.**

- **Why are row buffers so large?**
  - **Large arrays minimize cost-per-bit.**
  - **Striping a cache line across multiple chips (arrays) improves data transfer bandwidth.**

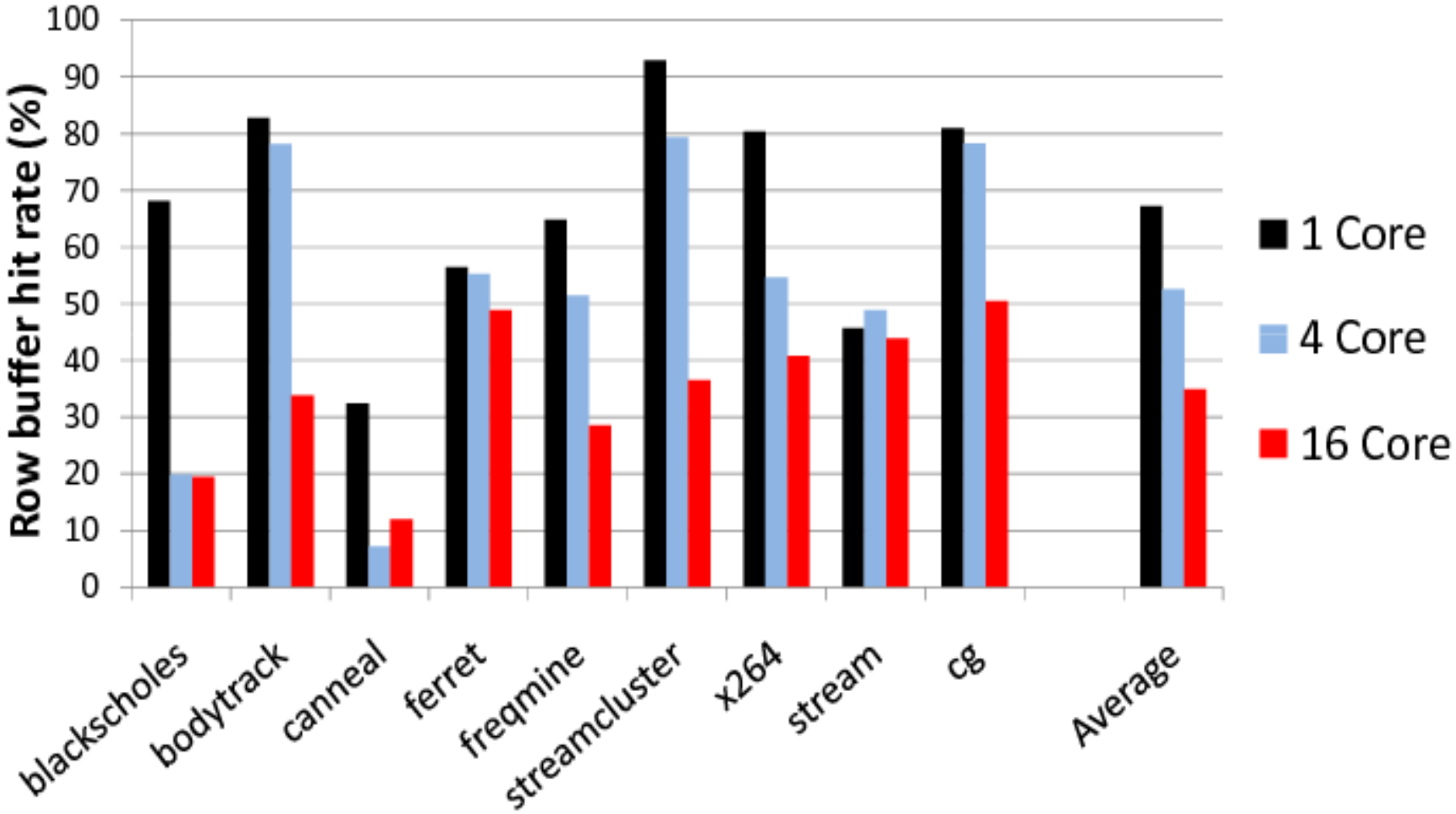
DRAM Access Inefficiencies - II

- **Open page policy**
  - Row buffers kept open with the hope that subsequent requests will be row buffer hits.
- **FR-FCFS request scheduling (First-Ready FCFS)**
  - Memory controller schedules requests to open row -buffers first.

	Access Latency	Access Energy
Row-buffer Hit	~ 75 cycles	~ 18 nJ
Row-buffer Miss	~ 225 cycles	~ 38 nJ

- **Diminishing locality in multi-cores.**

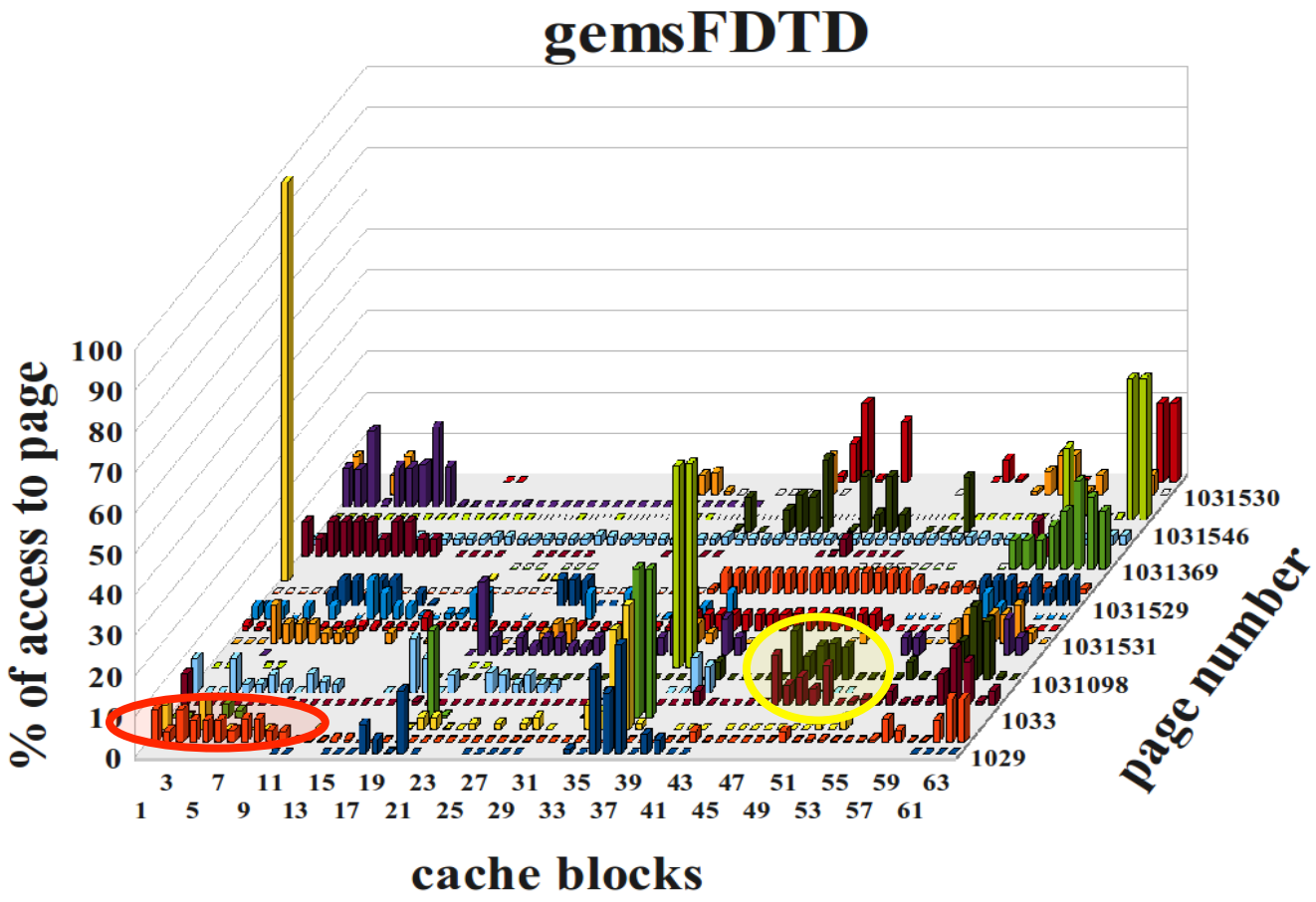
DRAM Row-buffer Hit-rates



With increasing core/thread counts, DRAM row-buffer hit-rates decrease.

DRAM Row-Buffer Hit Rates

Within a relatively small time interval – row hit rate is relatively small



Basic Idea

- **Micro-pages**
  - finer granularity tracking than OS page size
  - in this case 1KB
- **Identify "hot" micro-pages**
  - via memory controller accounting and OS daemon
- **Reserve DRAM rows for hot micro-pages**
  - book keeping overhead is < 0.1% of main memory capacity
- **EPOCH based accounting**
  - expose EPOCH length to OS for flexibility
- **Promote cold micro-pages to superpage**
  - to extend TLB reach in lightly used areas

Two Approaches

- SW Reduced OS Page Size (ROPS)
  - reduce page size to 1KB
  - migrate hot pages via DRAM copy
- Hardware assisted migration (HAM)
  - add level of address indirection
    - » initial data placement via typical first touch policy
    - » maintain a mapping table
    - » copy hot micropages to reserved rows
    - » populate/update mapping table every 50M cycle epoch

Results

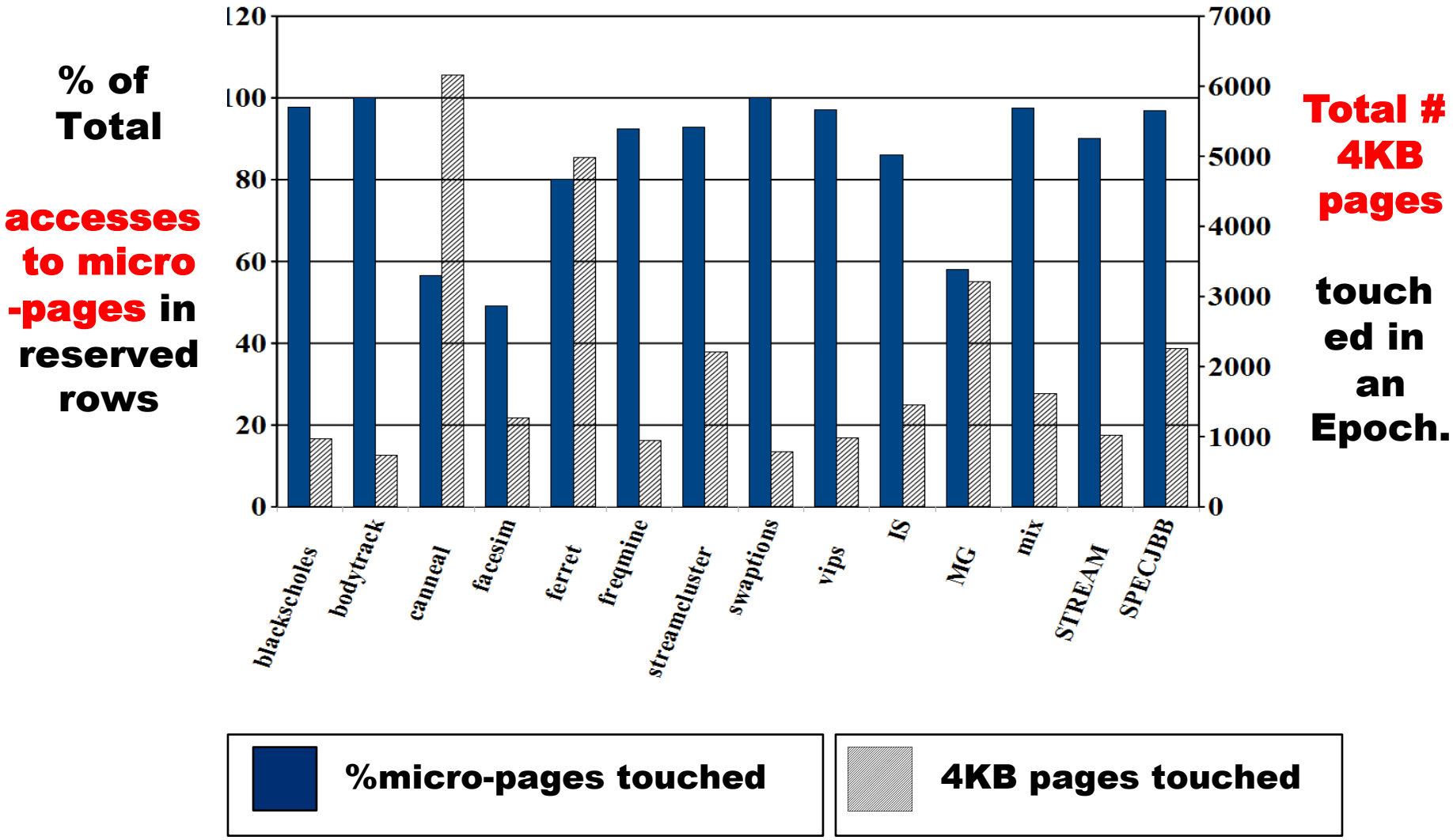
- **Schemes evaluated**
  - **baseline**
  - **profiled oracle**
    - » **best-effort estimate of what will happen next epoch based on previous profile run**
  - **epoch based ROPS & HAM**
- **Simulation platform**
  - **SIMICS**
  - **DRAMsim based DRAM timing**
    - » **timing and energy parameters from Micron datasheets**

Simulation Parameters

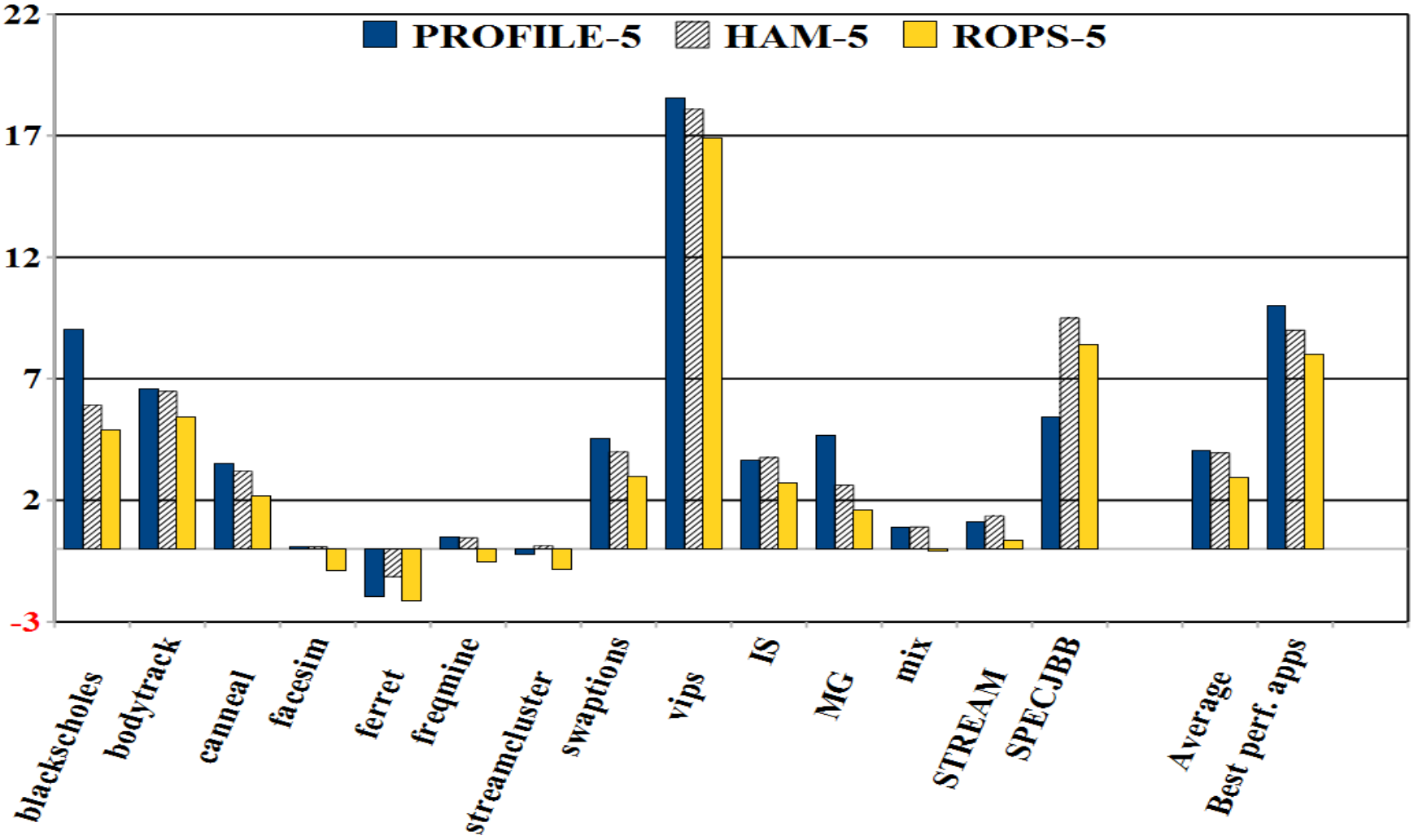
CPU	**4-core Out-of-Order CMP, 2 GHz freq.**
L1 Inst. and Data Cache	**Private, 32 KB/2-way, 1-cycle access**
L2 Unified Cache	**Shared, 128 KB/8-way, 10-cycle access**
Total DRAM Capacity	**4 GB**
DIMM Configuration	**8 DIMMs, 1 rank/DIMM, 64 bit channel, 8 devices/DIMM**
Active Row-Buffers per DIMM	**4**
DIMM-Level Row-Buffer Size	**8 KB**

Note – more diverse parameter set underway

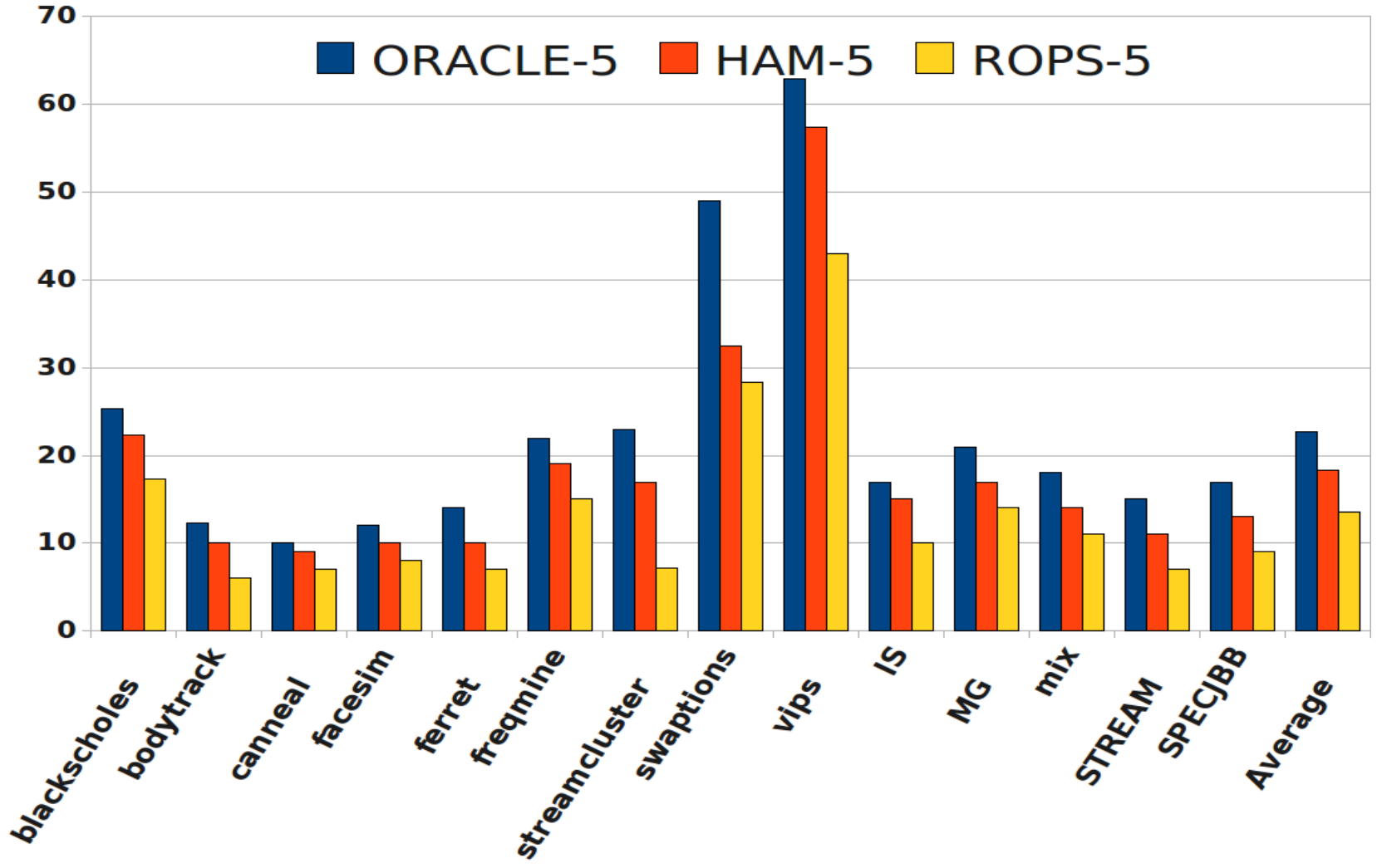
Micro-Page vs. OS Page Accessed per Epoch



Modest % Change in Performance



% Reduction in Memory System Power



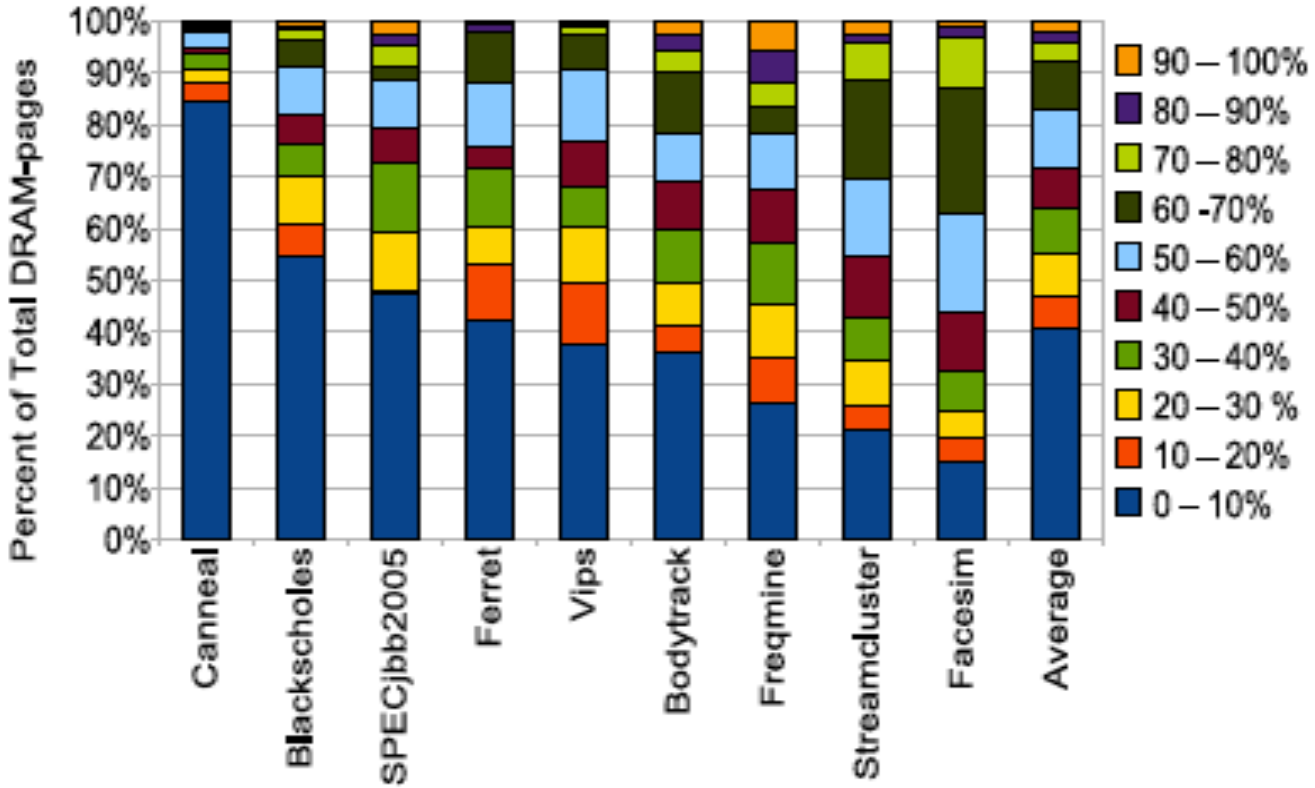
Micro-Page Conclusions

- On average, for applications with room for improvement and with our best performing scheme
  - Average performance ↑ 9% (max. 18%)
  - Average memory energy consumption ↓ 18% (max. 62%).
  - Average row-buffer utilization ↑ 38%
- Hardware assisted migration offers better returns due to fewer overheads of TLB shoot-down and misses.
- Ongoing Work
  - try other grain, epoch sizes w/ multi-MC & more cores

Predictor Based Row Buffer Management

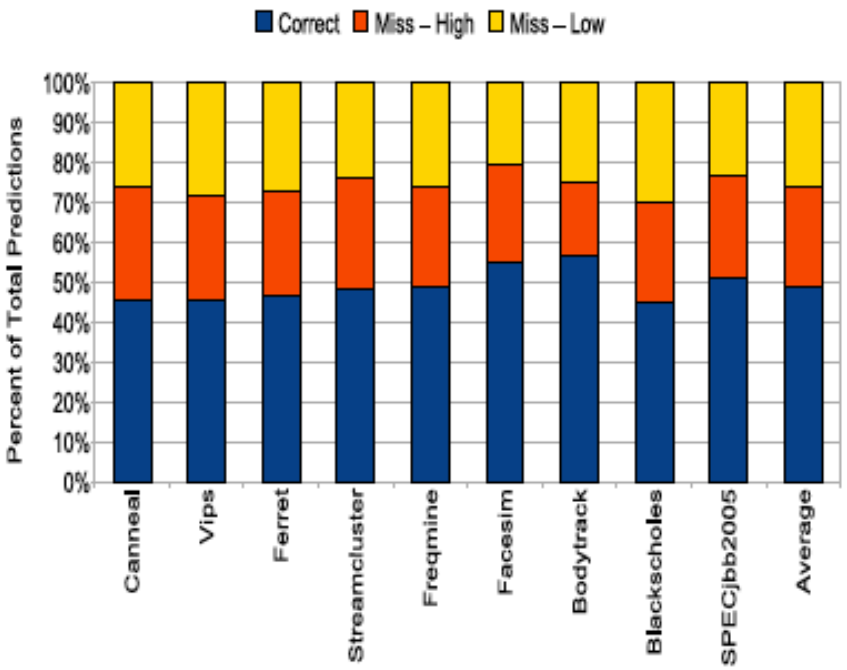
- **Basic idea for MC scheduling policy**
  - **neither open row or closed row MC scheduling is likely optimal for all DRAM rows**
    - » **particularly in multi-core workloads**
  - **similar to branch prediction idea**
  - **DRAM row accesses tend to follow a pattern**
    - » **use a dynamic predictor to determine when to close a row**
  - **2 variants**
    - » **keep track of time**
      - **Kahn patent 6799241 selects between 1k, 2k, 5k cycles for all pages**
      - **TBP – dynamically adjusts time on a per-page basis**
    - » **keep track of access count**
      - **ABP – bases prediction on # of accesses**

For 8 cores & 8 threads

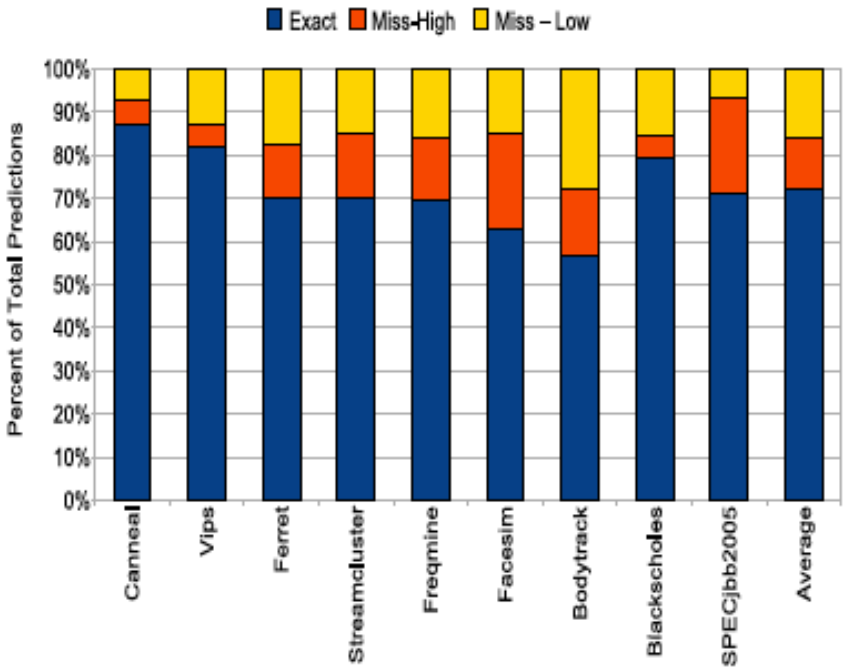


Tracks all DRAM-rows accessed

Per Row Based Prediction



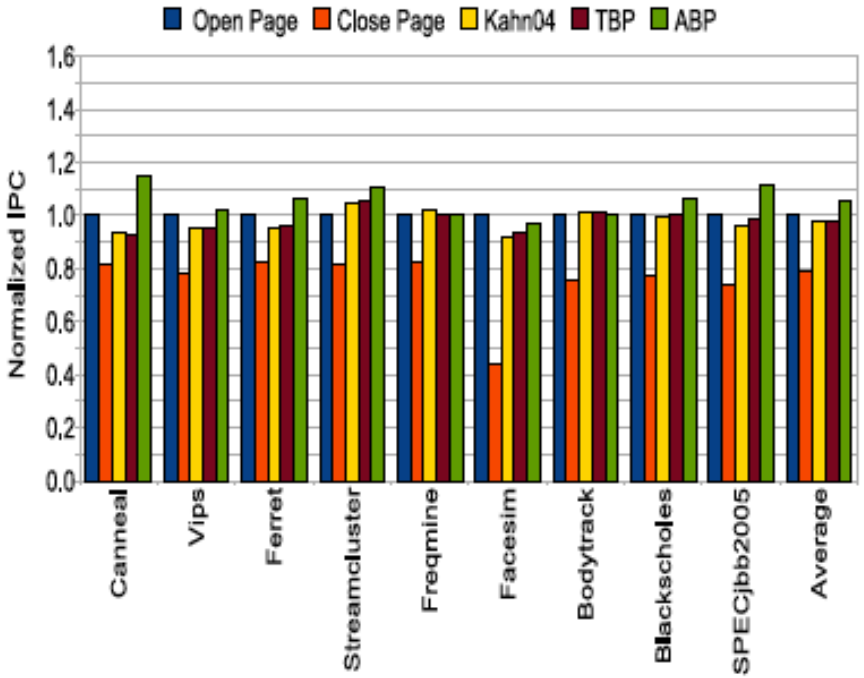
a. Time-based predictor (TBP)



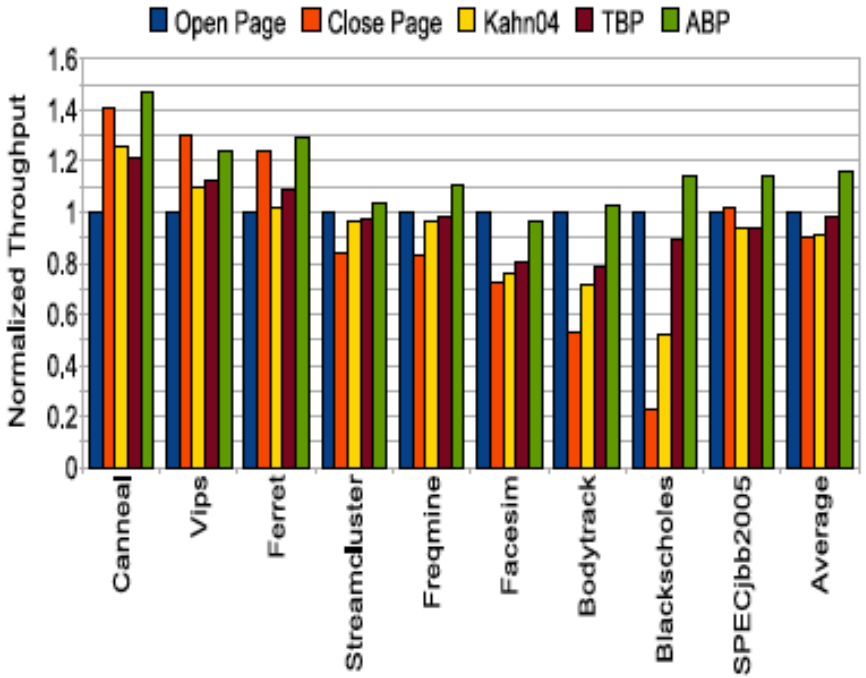
b. Access-based predictor (ABP)

Access based is better

Relative Throughput



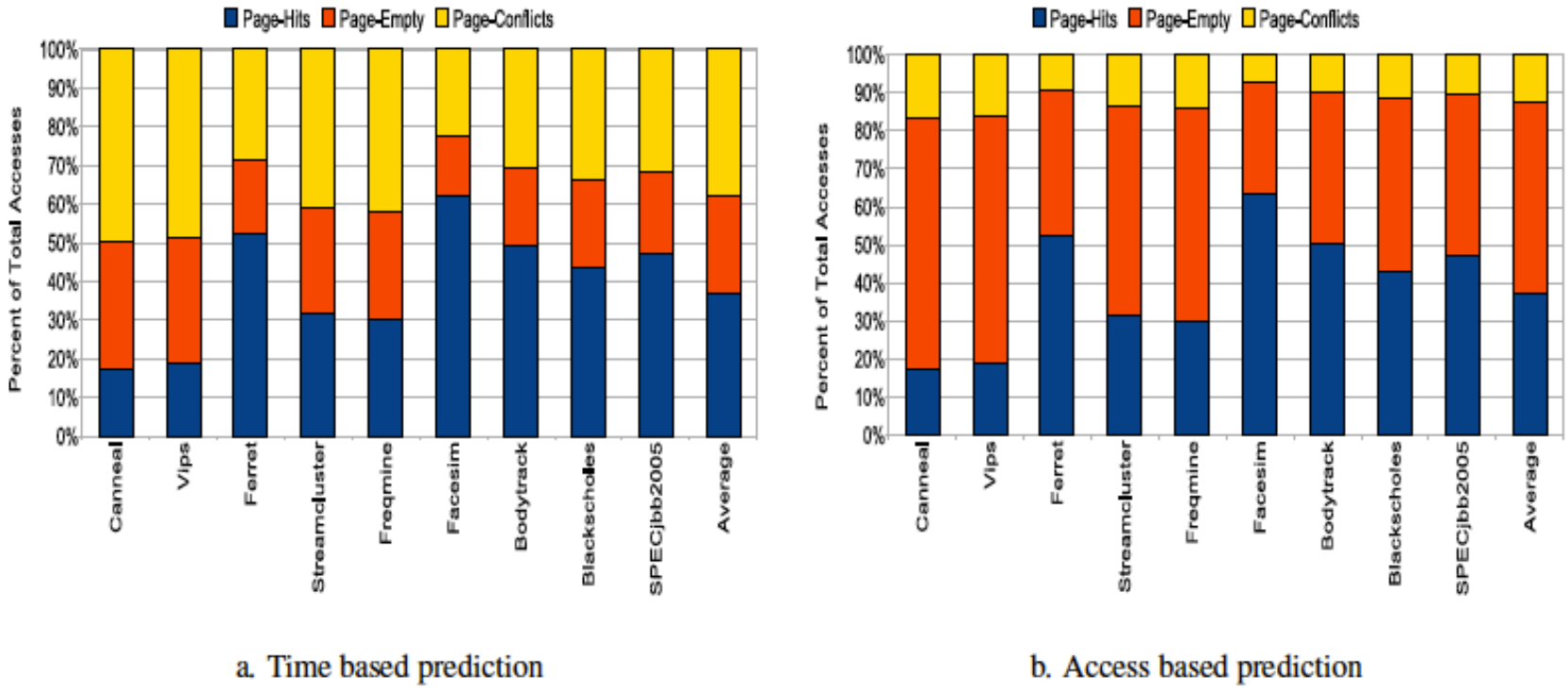
a. Relative Throughput, Single Core



b. Relative Throughput, 8 Cores/8 Threads

Throughput variation increases with core/thread count

Row/Page Hits vs. Misses

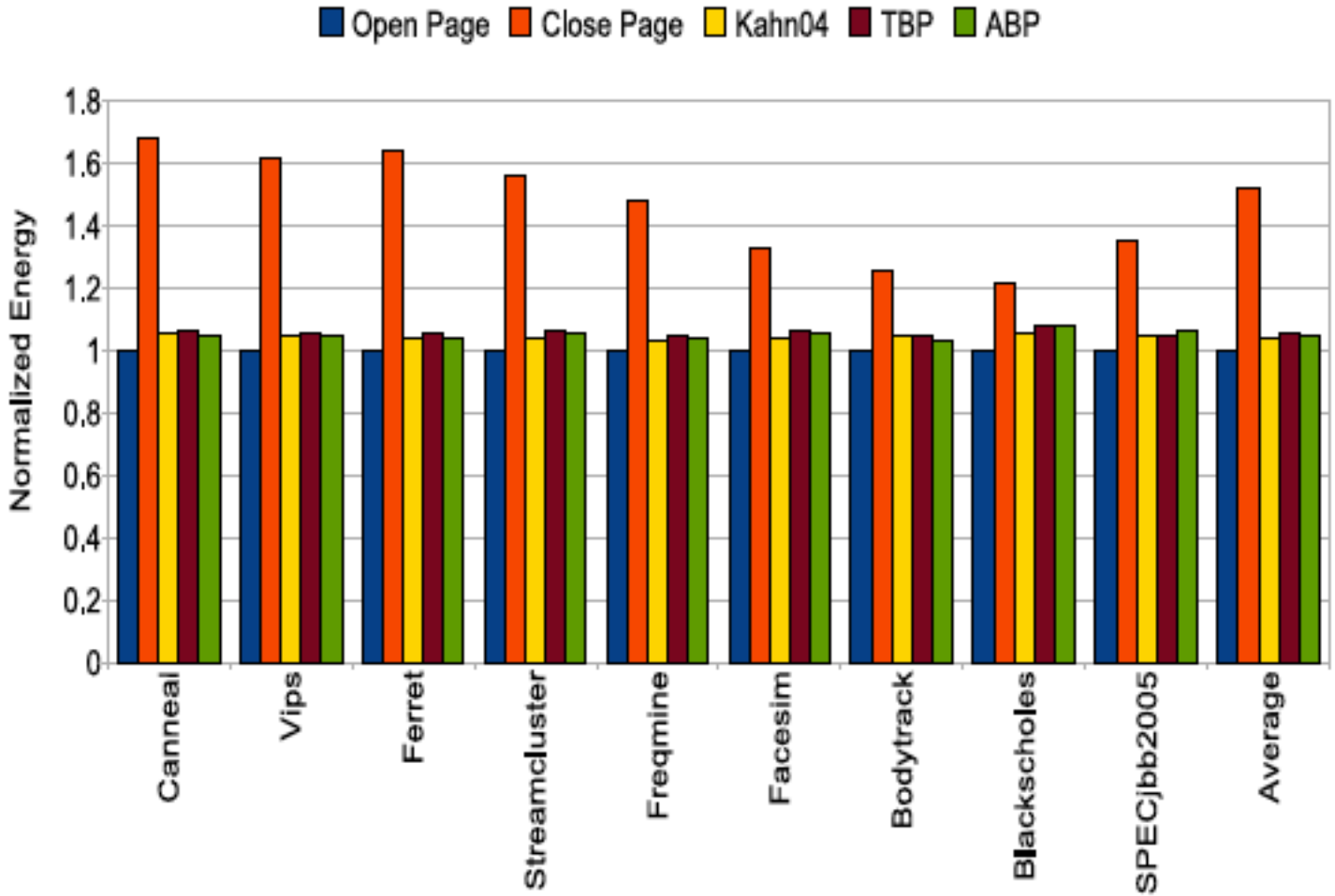


a. Time based prediction

b. Access based prediction

Page conflicts are much worse than page empty misses – hence ABP performance is better for all workloads

Energy is also Important



8 core system – simulation and actual measurements within 5%

Controller Based Prediction

- Initial results promising
- Downside – further work required
  - refine the tactics
    - » simulation needs to consider much larger core/thread counts
      - and more diverse app set
      - average isn't the right metric (kudo's to Jim Smith)
    - » plus consider multiple MC issues
  - area and power impact of memory controller
    - » these results track DRAM energy and performance
    - » but MC model is weak for both delay and power

Multiple MC Problems

- Pin limitations imply an increase in queuing delay
  - Almost 8x increase in queuing delays from single core/one thread to 16 cores/16 threads
- Multi-core implies an increase in row-buffer interference
  - Increasingly randomized memory access stream
  - Row-buffer hit rates bound to go down
- Longer on- and off-chip wire delays imply an increase in NUMA factor
  - NUMA factor defined as slowest/fastest access
- NUMA factor already at 1.5 today

Problems - II

- **DRAM access time in systems with multiple on-chip MCs is governed by**
  - **Distance between requesting core and responding MC.**
  - **Load on the on-chip interconnect.**
  - **Average queuing delay at responding MC**
  - **Bank and rank contention at target DIMM**
  - **Row-buffer hit rate at responding MC**

Bottomline : Intelligent management of data is required

Adaptive First Touch Policy

- **Basic idea : Assign each new virtual page to a DRAM (physical) page belonging to MC (j) that minimizes the following cost function –**

$$cost_j = \alpha \times load_j + \beta \times rowhits_j + \lambda \times distance_j$$

Measure of Queuing Delay

Measure of Locality at DRAM

Measure of Physical Proximity

Constants α, β and λ can be made programmable

Dynamic Page Migration Policy

- **Programs change phases!!**
  - **Can completely stop touching new pages**
  - **Can change the frequency of access to a subset of pages**
- **Leads to imbalance in MC accesses**
  - **For long running programs with varying working sets, AFT can lead to some MCs getting overloaded**

Solution : Dynamically migrate pages between MCs at runtime to decrease imbalance

Catch-22: energy cost & time to migrate in order to save time and energy later.

Dynamic Page Migration Policy - Challenges

- **Selecting recipient MC**

$$cost_k = \Lambda \times distance_k + \Gamma \times rowhits_k$$

Move pages to a physically proximal MC

Minimize interference at recipient MC

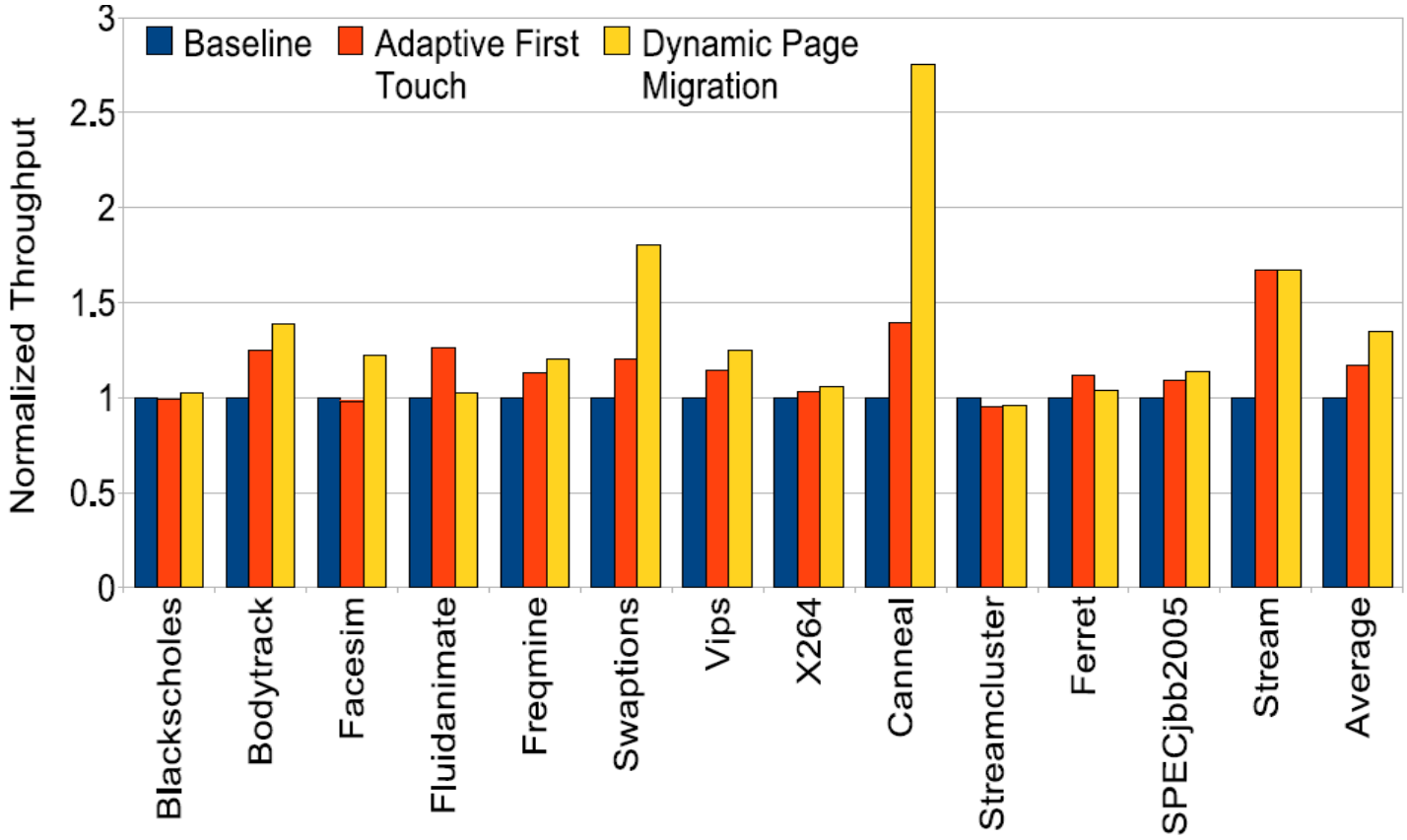
  - **Move pages to MC with *least* value of cost function**
- **Selecting *N* pages to migrate**
  - **Empirically select the best possible value**
  - **Can also be made programmable**

Methodology

- Simics based simulation platform
- DRAMSim based DRAM timing.
- DRAM energy figures from CACTI 6.5
- Baseline : Assign pages to closest MC

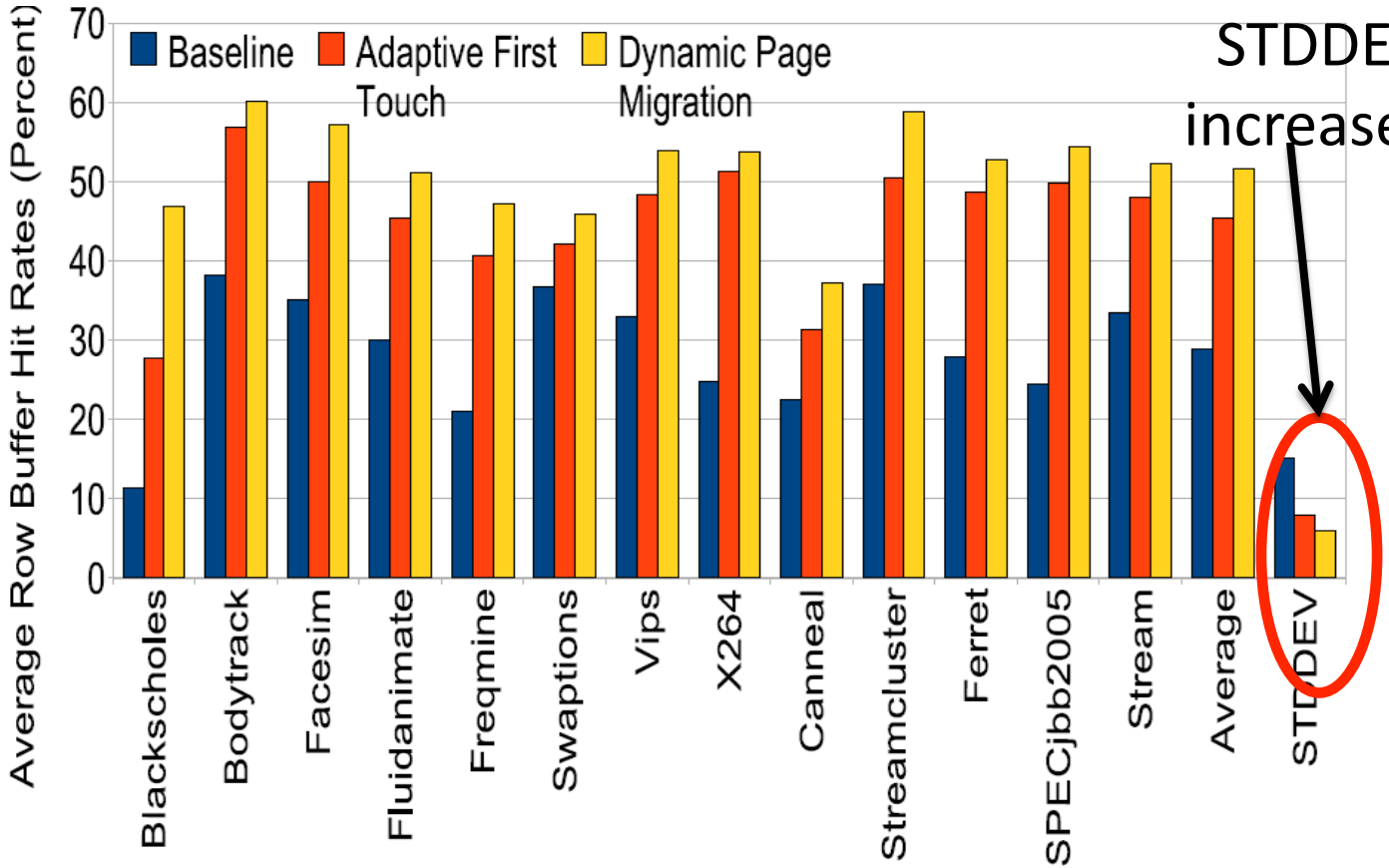
CPU	16-core Out-of-Order CMP, 3 GHz freq.
L1 Inst. and Data Cache	Private, 32 KB/2-way, 1-cycle access
L2 Unified Cache	Shared, 2 MB KB/8-way, 4x4 S-NUCA, 3 cycle bank access
Total DRAM Capacity	4 GB
DIMM Configuration	8 DIMMs, 1 rank/DIMM, 64 bit channel, 8 devices/DIMM
$\alpha, \beta, \lambda, \Lambda, \Gamma$	10, 20, 100, 100, 100

Results - Throughput



AFT : ⇧17.1% , Dynamic Page Migration : ⇧34.8%

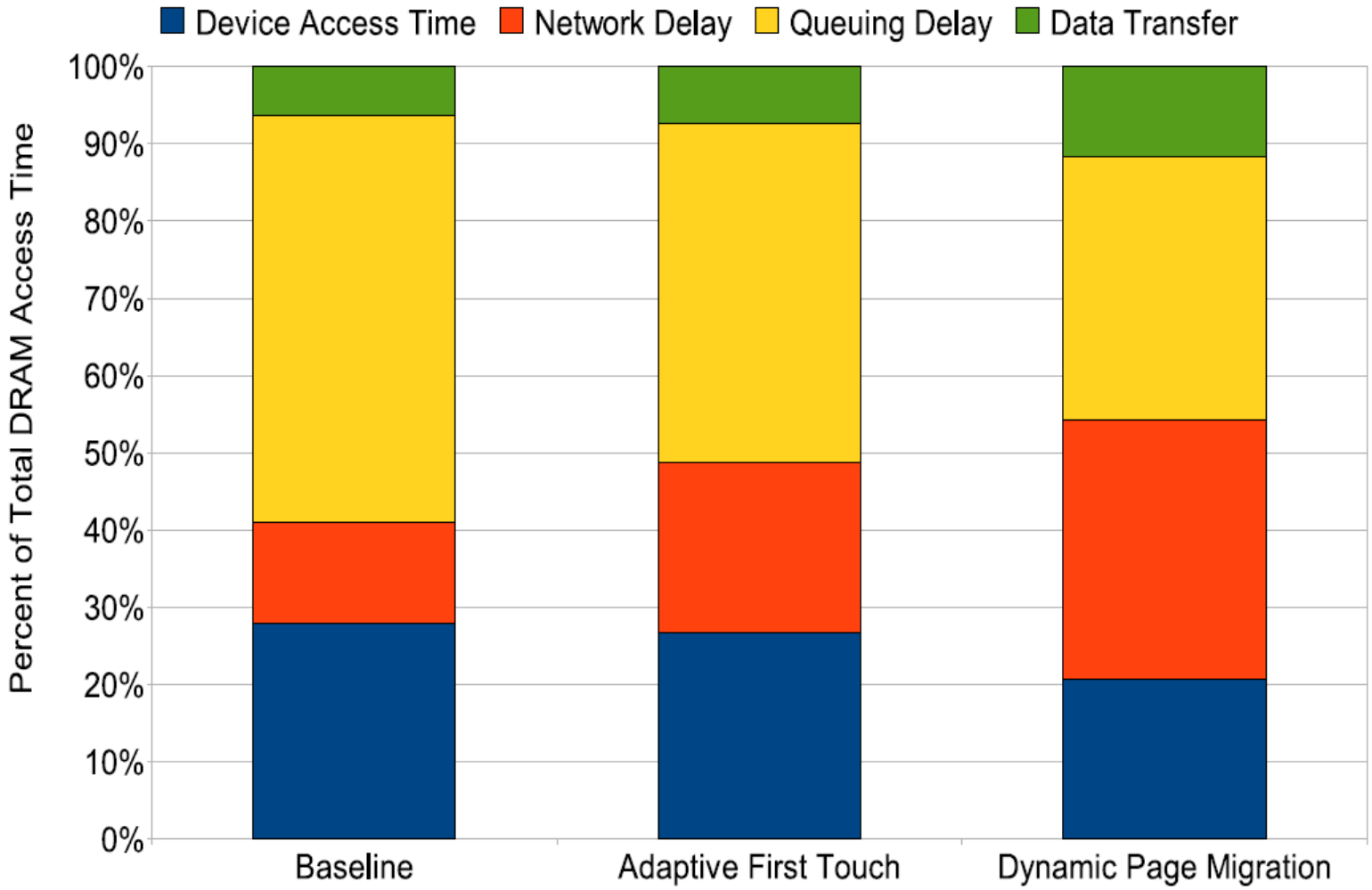
Results – DRAM Locality



STDDEV Down, increased fairness

AFT : ⇧ 16.6% , Dynamic Page Migration : ⇧ 22.7%

Results – Reasons for Benefits



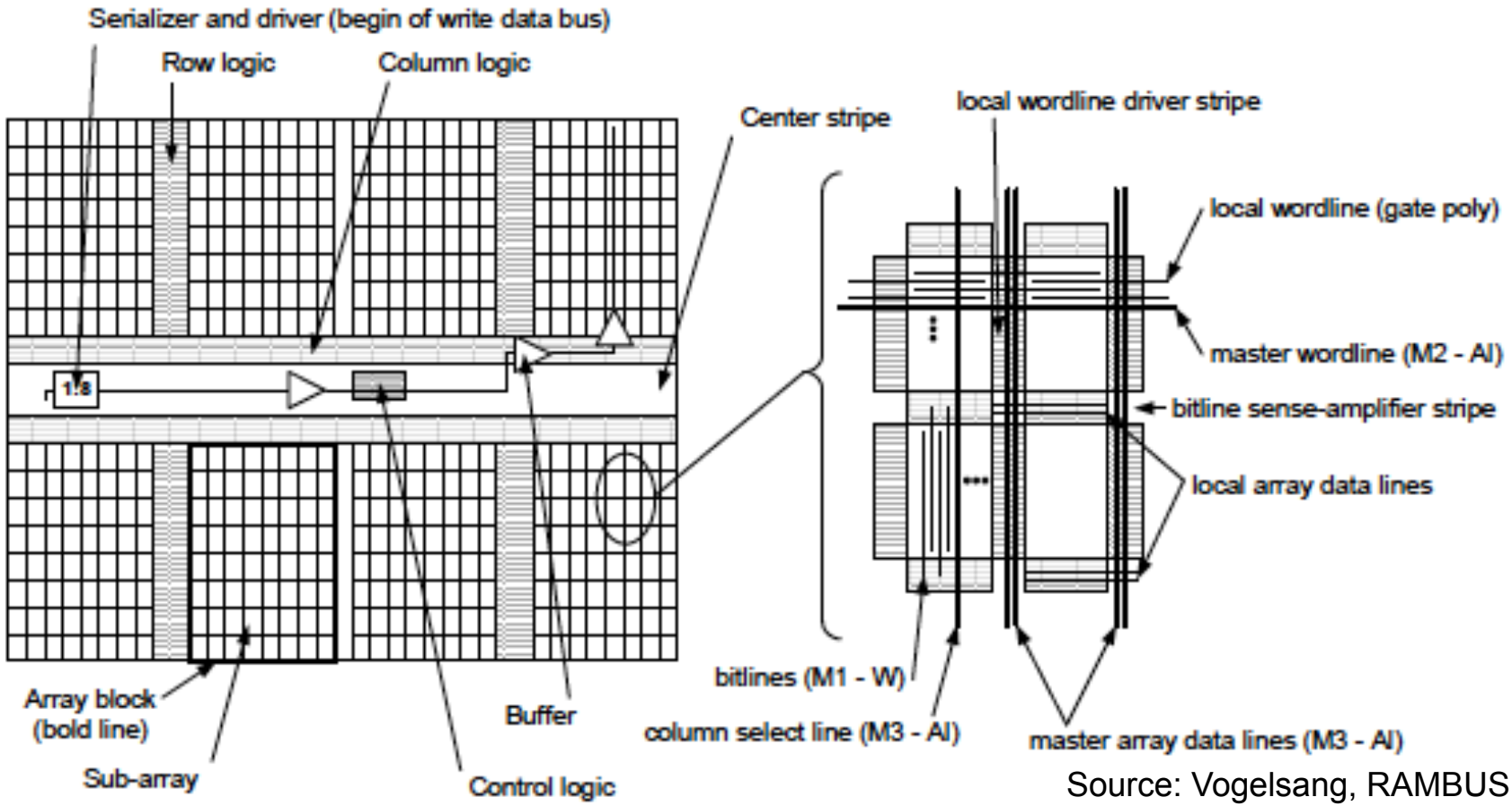
Multiple MC Summary

- Multiple, on-chip MCs will be common in future CMPs, with multiple cores sharing one MC
  - Intelligent data mapping will need to be done to reduce average memory access delay
- Adaptive First Touch policy
  - Increases performance by 17.1%
  - Decreases DRAM energy consumption by 14.1%
- Dynamic page migration, improvement on AFT
  - Further improvement over AFT by 17.7%, 34.8% over baseline.
  - Increases energy consumption by 5.2%

Rethinking DRAM & Dimm Micro-architecture

ISCA 2010

Typical Layout



Source: Vogelsang, RAMBUS
to appear MICRO 2010

Optimize cost/bit in cheap process (no Cu, 3 metal process)

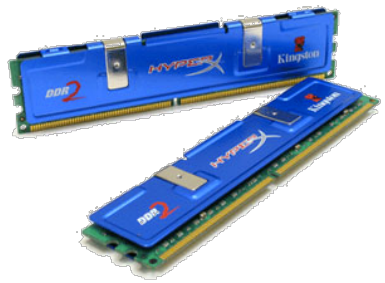
Mainstream DRAM Issues

- 2 types of circuits
  - on pitch
    - » bit and word repeat distance, bit-line sense amps, local word -line drivers, word and column logic close to the mats
  - off-pitch – the other stuff
    - » typically limited by signal and power wiring constraints
- Most costly changes
  - #1: bit-line sense amp stripe
  - #2: local word-line driver stripe
  - #3: column logic
  - #4: row logic

History of Disruptive Changes to date

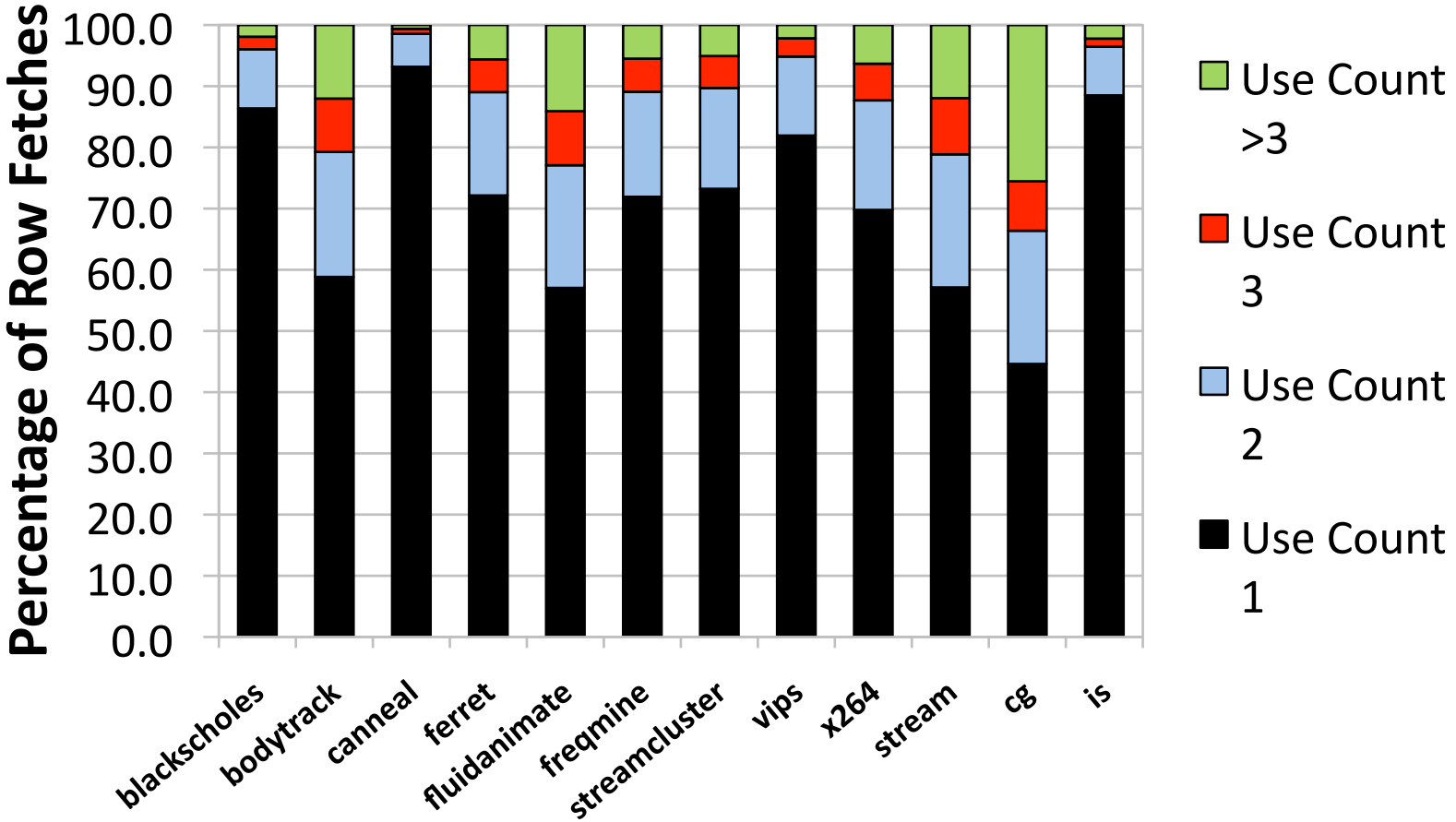
Technology transition	Disruptive change	Background
Range from 250nm to 200nm to 140nm to 110nm	Stitched wordline to segmented wordline	Minimum feature size of aluminum wiring no longer feasible. The time when different vendors did this transition has a large spread
110nm to 90nm	Increase in number of cells per bitline and / or local wordline	Leads to smaller die size. Better control of technology and design make step possible.
110nm to 90nm	Introduction of dual gate oxide	Allows lower voltage operation and better performance of standard logic transistors.
90nm to 75nm	Introduction of p+ gate doping of PMOS transistors	Buried channel pfet performance not sufficient for standard logic of high data rate DRAMs.
90nm to 75nm	Introduction of 3-dimensional access transistor	Planar transistor device length got too short for threshold voltage control.
75nm to 65nm	Cell architecture 8f2 folded bitline to 6f2 open bitline	Leads to smaller die size. Better control of technology and design make step possible.
55nm to 44nm	Cu metallization	Lower resistance and / or capacitance in wiring for improved performance and / or power reduction.
40nm to 36nm	Cell architecture 6f2 to 4f2 with vertical access transistor	Leads to smaller die size. Better control of technology and design expected to make step possible.
36nm to 31nm	High-k dielectric gate oxide	Better subthreshold behavior and reduced gate leakage.

Memory Trends



- **Energy**
  - large scale systems attribute 25-40% of total power to the memory subsystem
  - capital acquisition costs = operating costs over 3 years
  - energy is a first-order design constraint
- **Access patterns**
  - increasing socket, core, and thread counts
  - increasingly random access stream
  - locality increasingly non-existent

Memory Trends



Related Work

- **Overfetch**
  - Ahn et al. (SC '09), Ware et al. (ICCD '06), Sudan et al. (ASPLOS '10)
- **DRAM Low-power modes**
  - Hur et al. (HPCA '08), Fan et al. (ISLPED '01), Pandey et al. (HPCA '06)
- **DRAM Redesign**
  - Loh (ISCA '08), Beamer et al. (ISCA '10)
- **Chipkill mechanisms**
  - Yoon and Erez (ASPLOS '10)

Consider

- Rethink DRAM design for modern constraints
  - Low-locality, reduced energy consumption, optimize TCO
- Selective Bitline Activation (SBA)
  - Minimal design changes
  - Considerable dynamic energy reductions for small latency and area penalties
- Single Subarray Access (SSA)
  - Significant changes to memory interface
  - Large dynamic and static energy savings
- Chipkill-level reliability
  - Reduced energy and storage overheads for reliability

Selective Bitline Activation

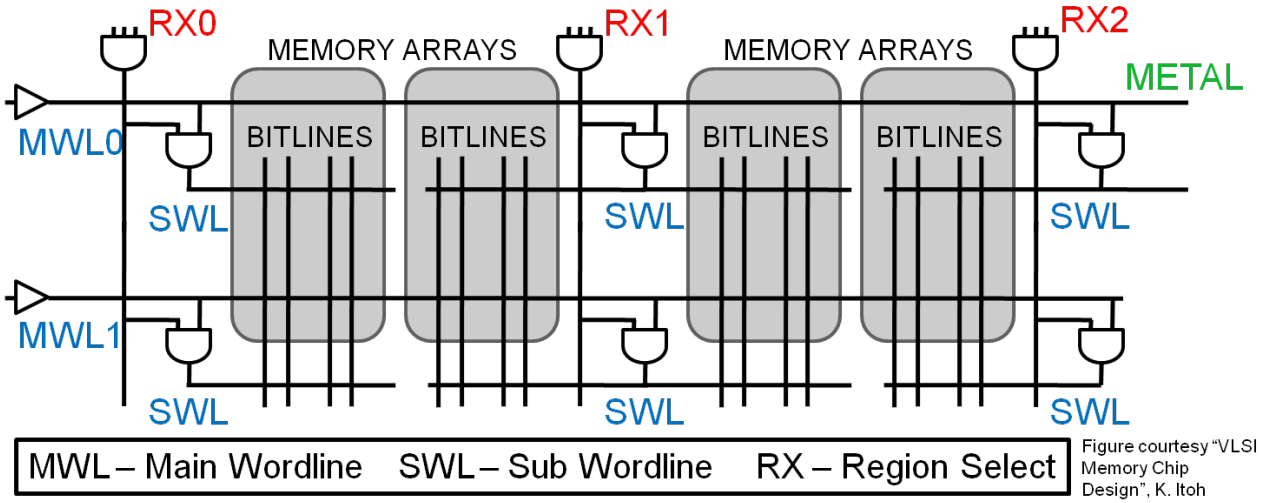
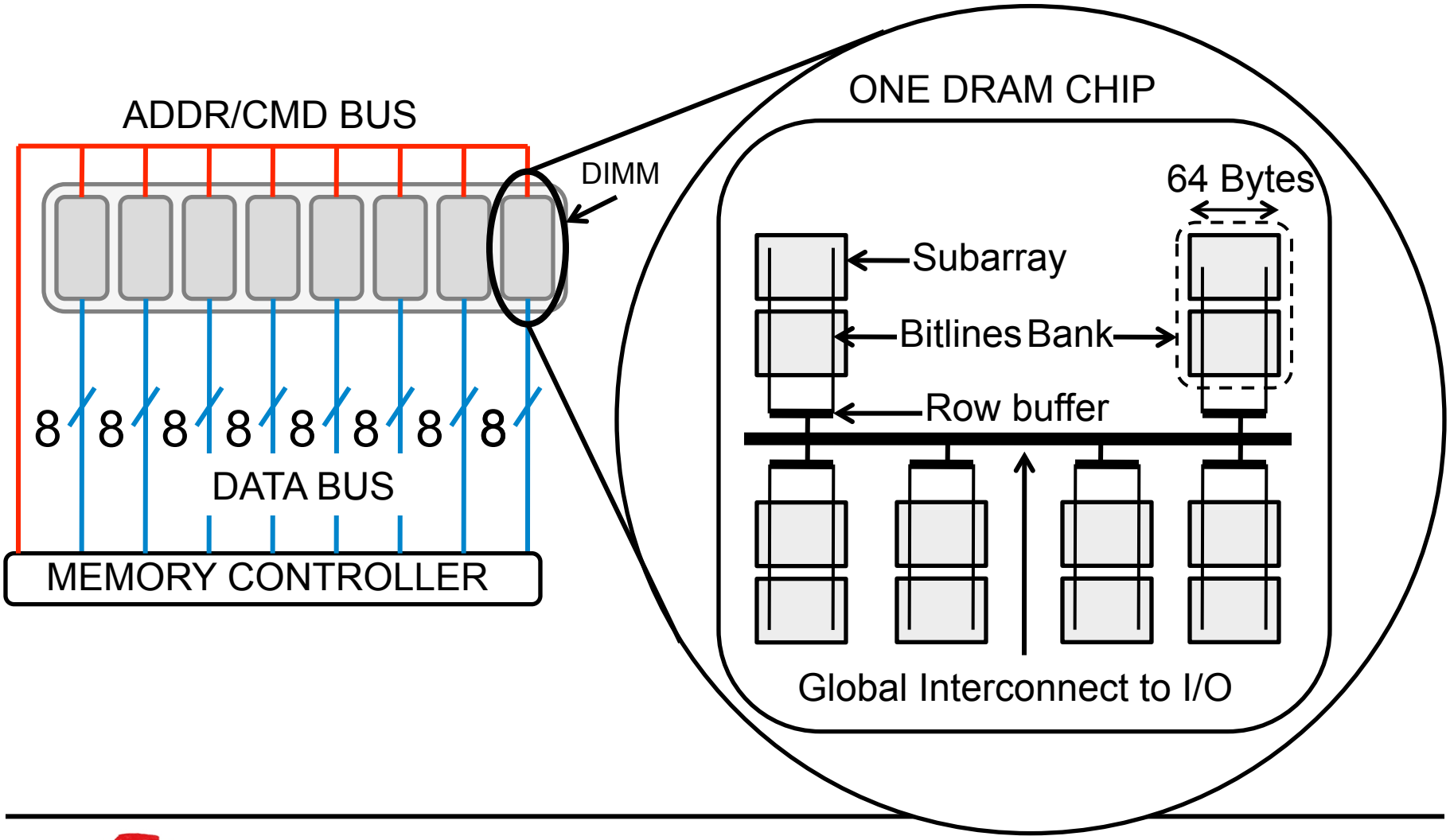


Figure courtesy "VLSI Memory Chip Design", K. Itoh

- **Activate only those bitlines corresponding to the requested cache line – reduce dynamic energy**
  - **Some area overhead depending on access granularity – note #1 cost adder**
  - **we pick 16 cache lines for 12.5% area overhead**
- **Requires no changes to the interface and minimal MC scheduling changes**

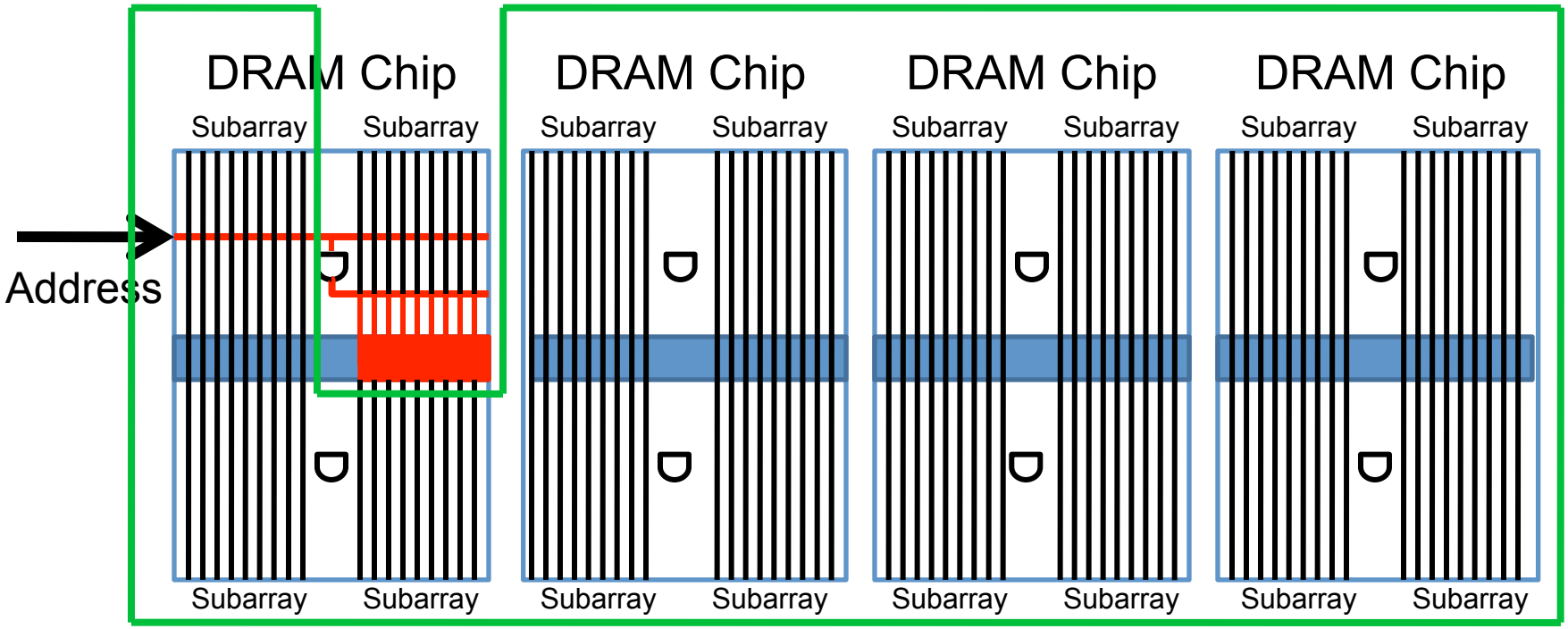
SSA Architecture



SSA Basics

- Entire DRAM chip divided into small subarrays
- Width of each subarray is exactly one cache line
- Fetch *entire* cache line from a single subarray in a single DRAM chip – SSA
- Groups of subarrays combined into "banks" to keep peripheral circuit overheads low
- Close page policy and "posted-RAS" similar to SBA
- Data bus to processor essentially split into 8 narrow buses

SSA Operation



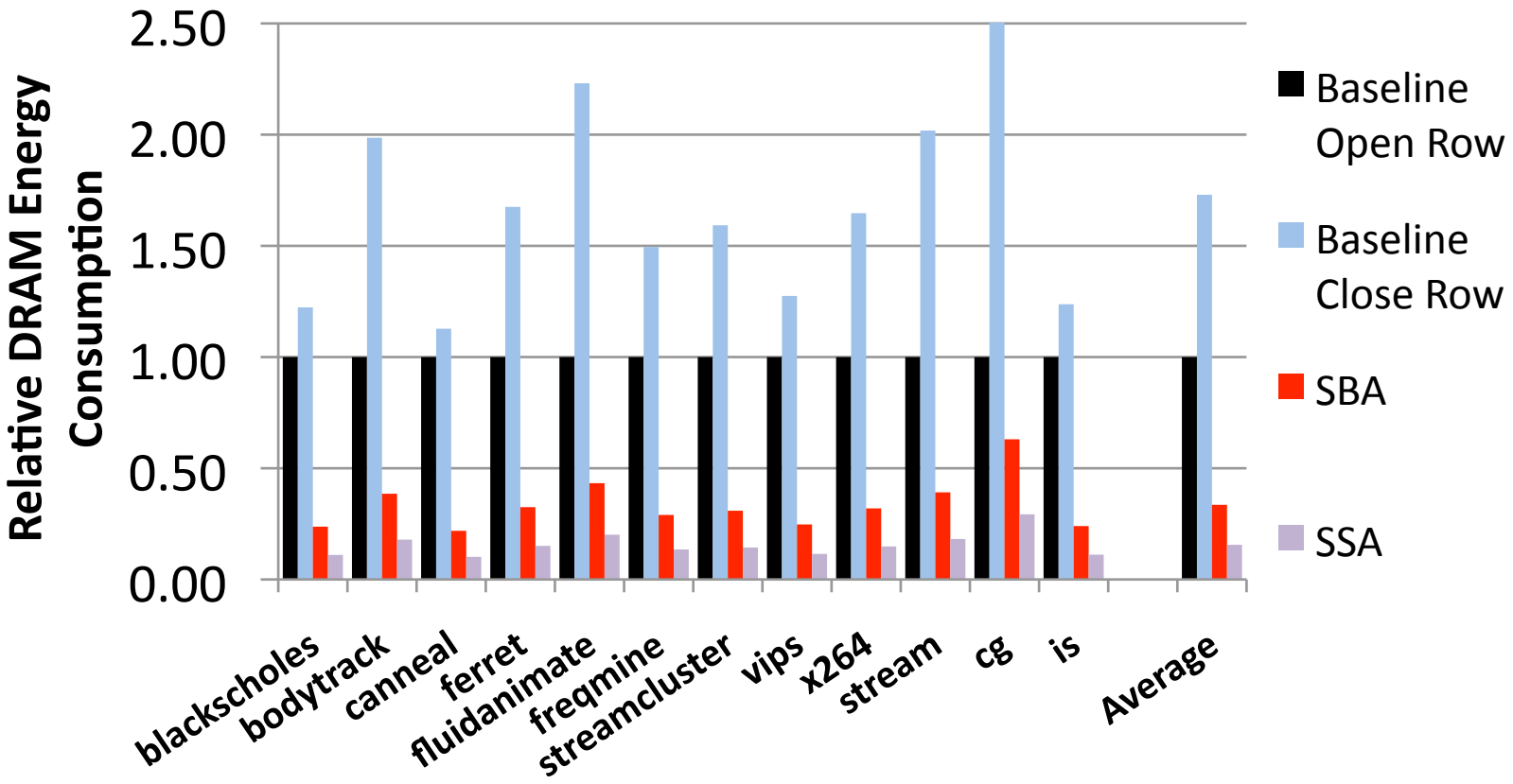
SSA Impact

- **Energy reduction**
  - Dynamic – fewer bitlines activated
  - Static – smaller activation footprint – more and longer spells of inactivity – better power down
- **Latency impact**
  - Limited pins per cache line – serialization latency
  - Higher bank-level parallelism – shorter queuing delays
- **Area increase**
  - More peripheral circuitry and I/O at finer granularities – area overhead (< 5%)

Methodology

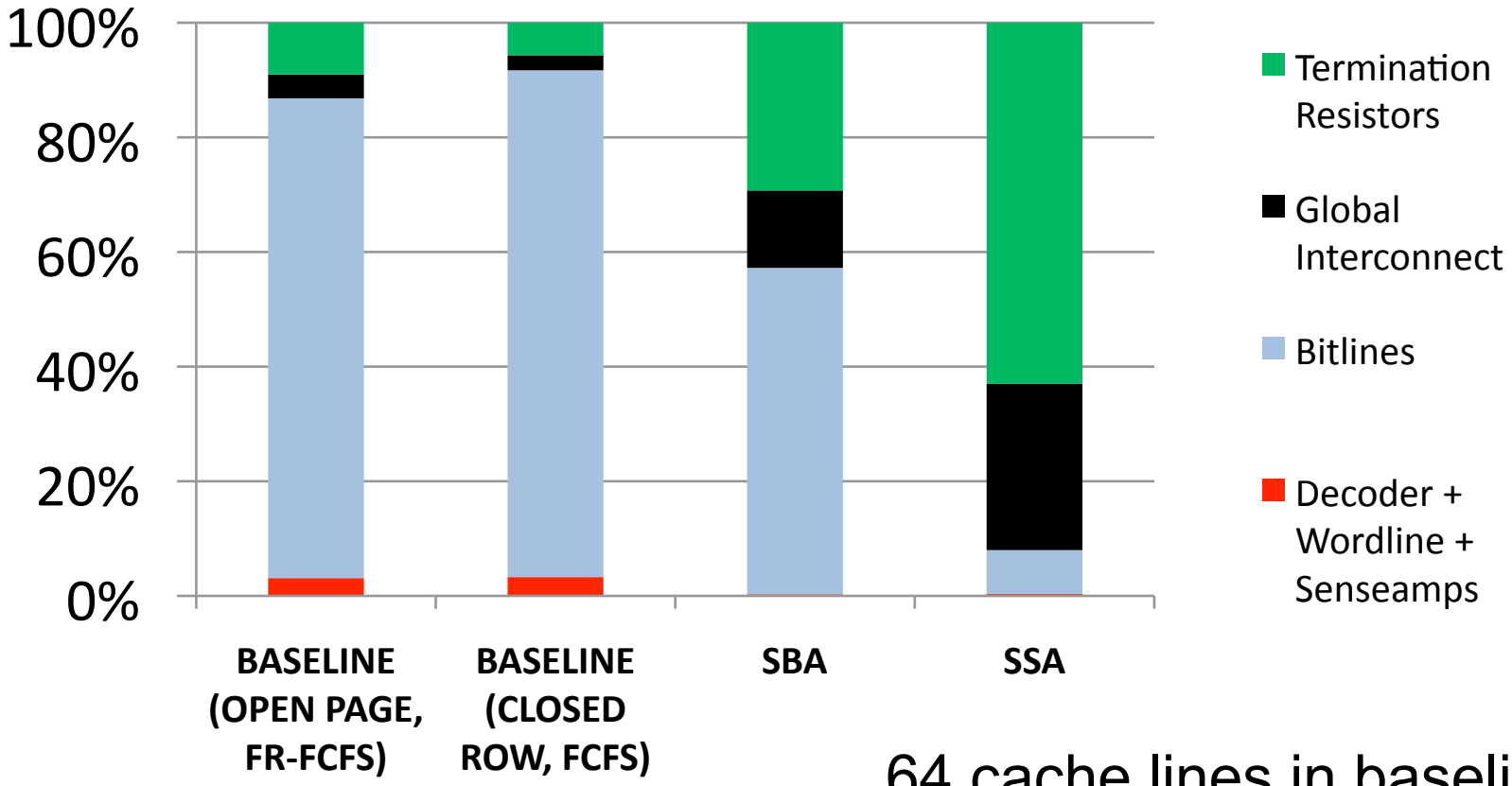
- Simics based simulator
  - 'ooo-micro-arch' and 'trans-staller'
- FCFS/FR-FCFS scheduling policies
- Address mapping and DRAM models from DRAMSim
- DRAM data from Micron datasheets
- Area/Energy numbers from heavily modified CACTI 6.5
- PARSEC/NAS/STREAM benchmarks
- 8 single-threaded OOO cores, 32 KB L1, 2 MB L2
- 2GHz processor, 400MHz DRAM

Dynamic Energy Reduction



Moving to close page policy – 73% energy increase on average
Compared to open page, 3X reduction with SBA, 6.4X with SSA

Contributors to energy consumption

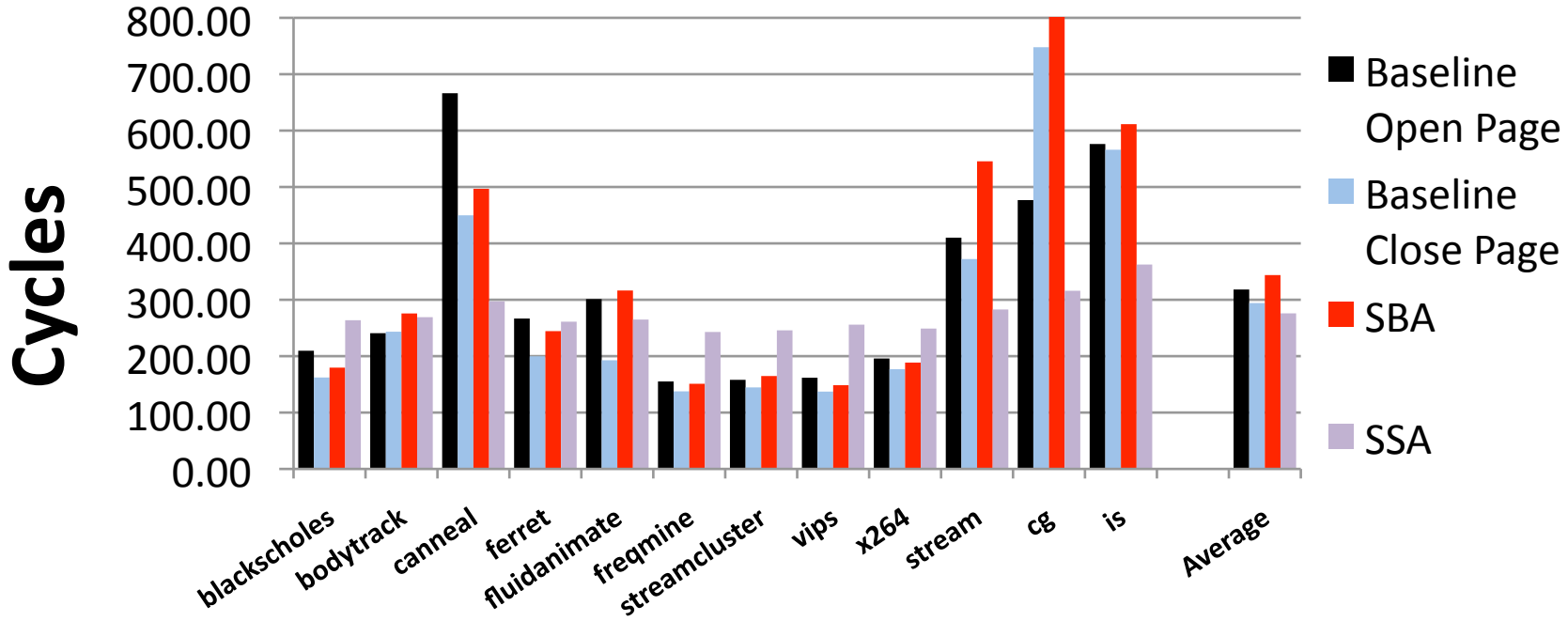


64 cache lines in baseline
16 cache lines in SBA
1 cache line in SSA

Static Energy – Power down modes

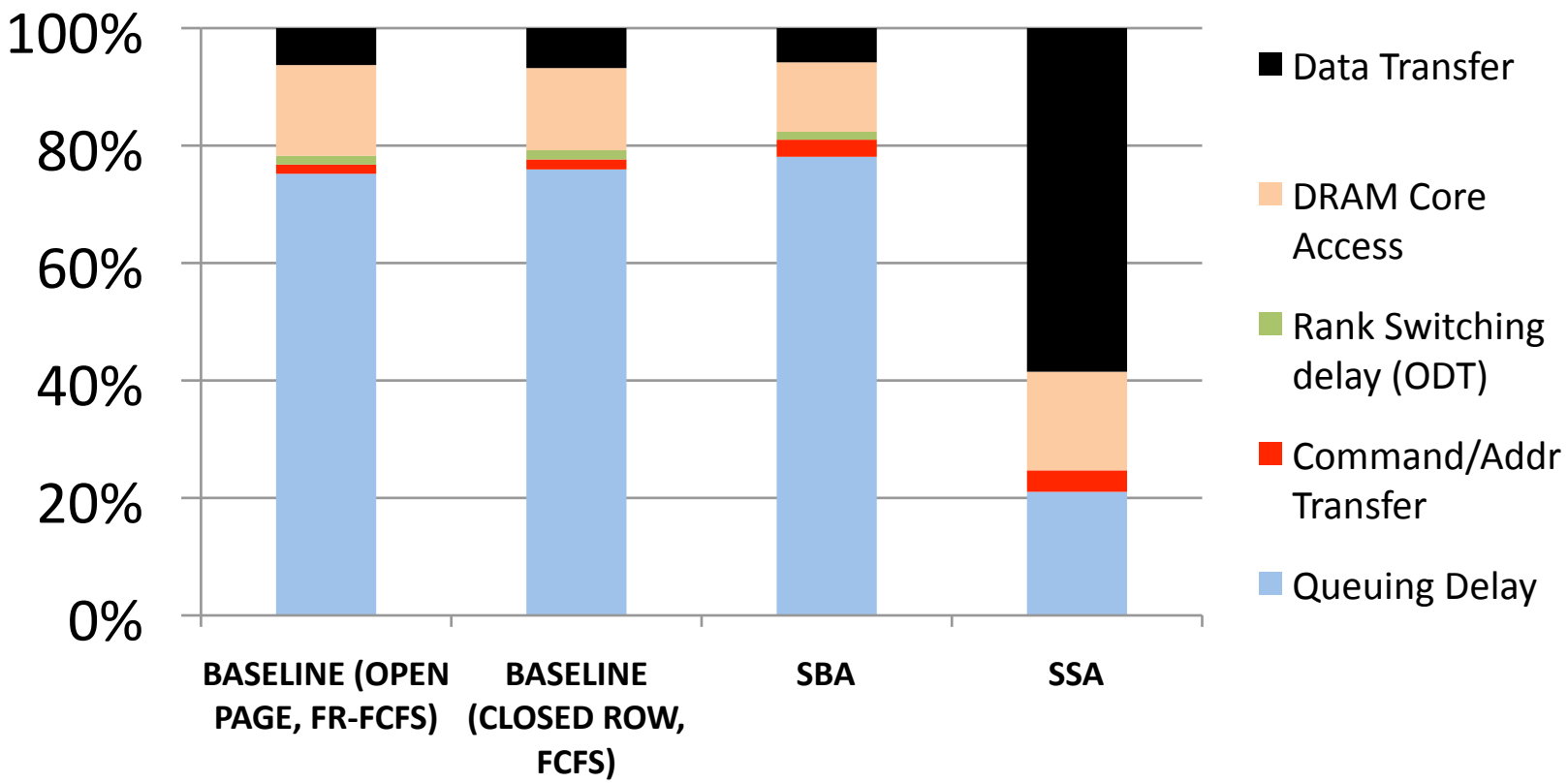
- Current DRAM chips already support several low-power modes
- Consider the low-overhead power down mode: 5.5X lower energy, 3 cycle wakeup time
- For a constant 5% latency increase
  - 17% low-power operation in the baseline
  - 80% low-power operation in SSA

Latency Characteristics



- Impact of Open/Close page policy – app. dependent: 17% decrease or 28% increase
- Posted-RAS adds about 10%
- Serialization/Queuing delay balance in SSA - 30% decrease for half the apps or 40% increase for the other half

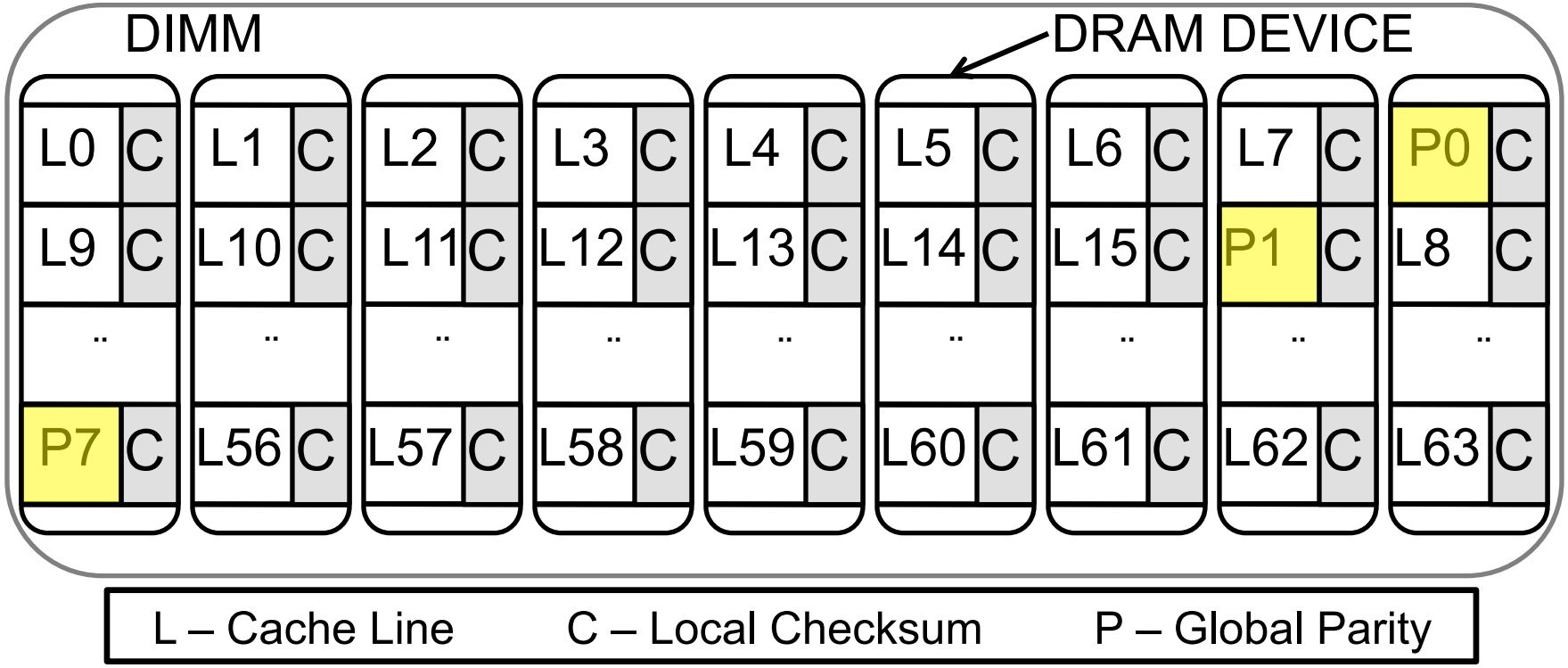
Contributors to Latency



DRAM Reliability

- Many server applications require chipkill-level reliability – failure of an entire DRAM chip
- One example of existing systems
  - 64-bit word requires 8-bit ECC
  - Each of these 72 bits must be read out of a different chip, else a chip failure will lead to a multi-bit error in the 72-bit field – unrecoverable!
  - Reading 72 chips - significant overfetch!
- Chipkill even more of a concern for SSA since entire cache line comes from a single chip

Proposed Solution



Approach similar to RAID-5

Chipkill design

- Two-tier error protection
- Tier - 1 protection – self-contained error detection
  - 8-bit checksum/cache line – 1.625% storage overhead
  - Every cache line read is now slightly longer
- Tier -2 protection – global error correction
  - RAID-like striped parity across 8+1 chips
  - 12.5% storage overhead
- Error-free access (common case)
  - 1 chip reads
  - 2 chip writes – leads to some bank contention
  - 12% IPC degradation
- Erroneous access
  - 9 chip operation

Main Memory Bandwidth Problem

The Main Memory Bandwidth Problem

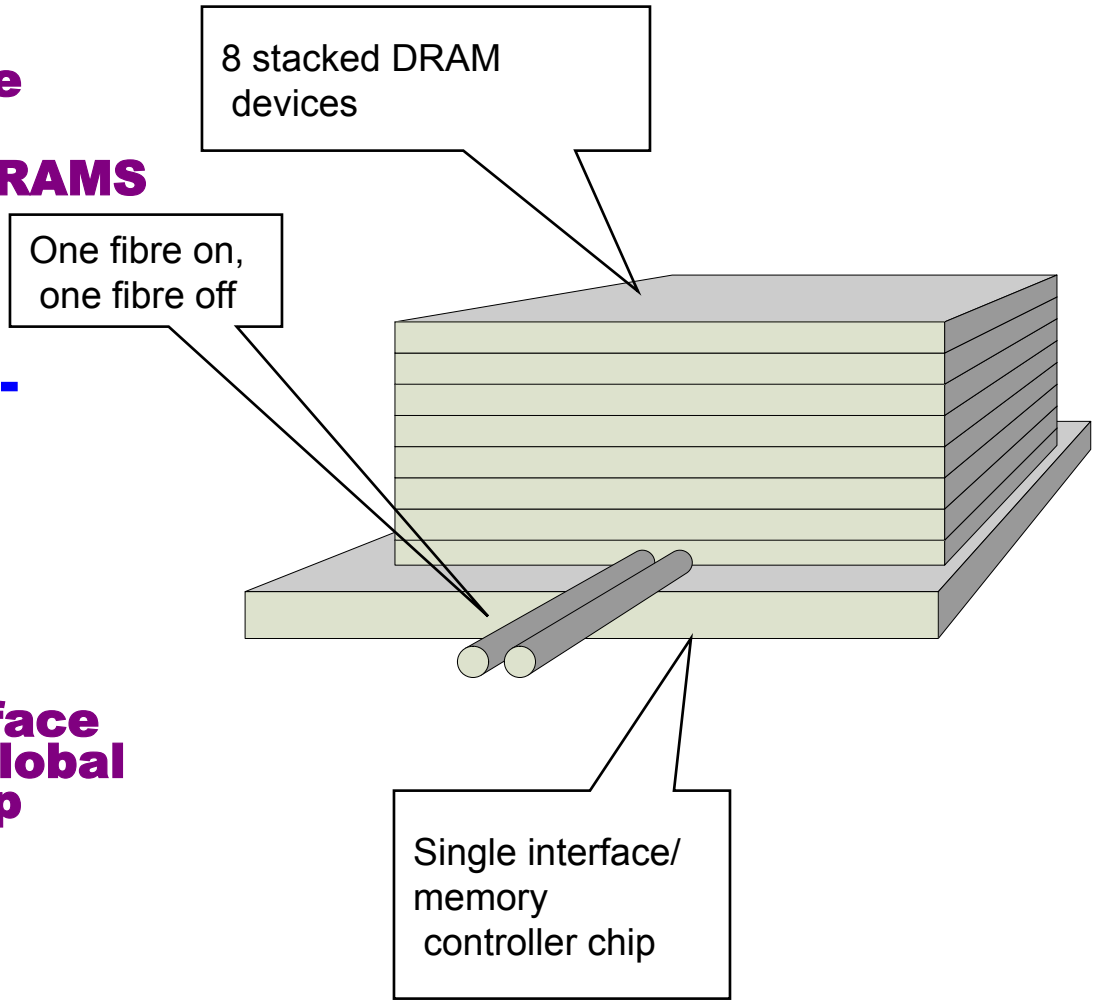
- Increased performance → increased main memory pressure
  - caches mitigate but don't eliminate this basic fact
- Wire based interconnect has limited promise
  - pin B/W and pin count growth limit socket to memory B/W
    - » signal integrity issues also limits capacity
    - » length dependent energy consumption is problematic
- Alternatives
  - LR-Dimm, BoB, etc. helps with capacity but not bandwidth
  - RL-Dimm helps with latency but not bandwidth
- Solutions?
  - I only see one that is likely at this point
  - nano-photonics
    - » Moray's talk describes the devices
    - » helps w/ B/W, energy, and capacity but not latency

What About Photonic NOC's

- Numerous recent advances
  - HP, IBM, Columbia, Luxtera, UCSB, Cornell, MIT, Infinera …
- Basic advantages
  - rewrite the bandwidth power rule
  - bandwidth per lane not limited by signal integrity issues
    - » wave division multiplexing
      - 8 λ available now
      - 64 λ predicted achievable in 16 nm (Vantrease et. al ISCA '08)
  - waveguide loss is very low
    - » power consumed at OE & EO endpoints
    - » length independent
  - activity factor influence is minor
- Optical bit-transport-energy (BTE) now
  - 2-3x better than wires on chip
  - 10-30x better than wires off chip

OCMM with stacked DRAM

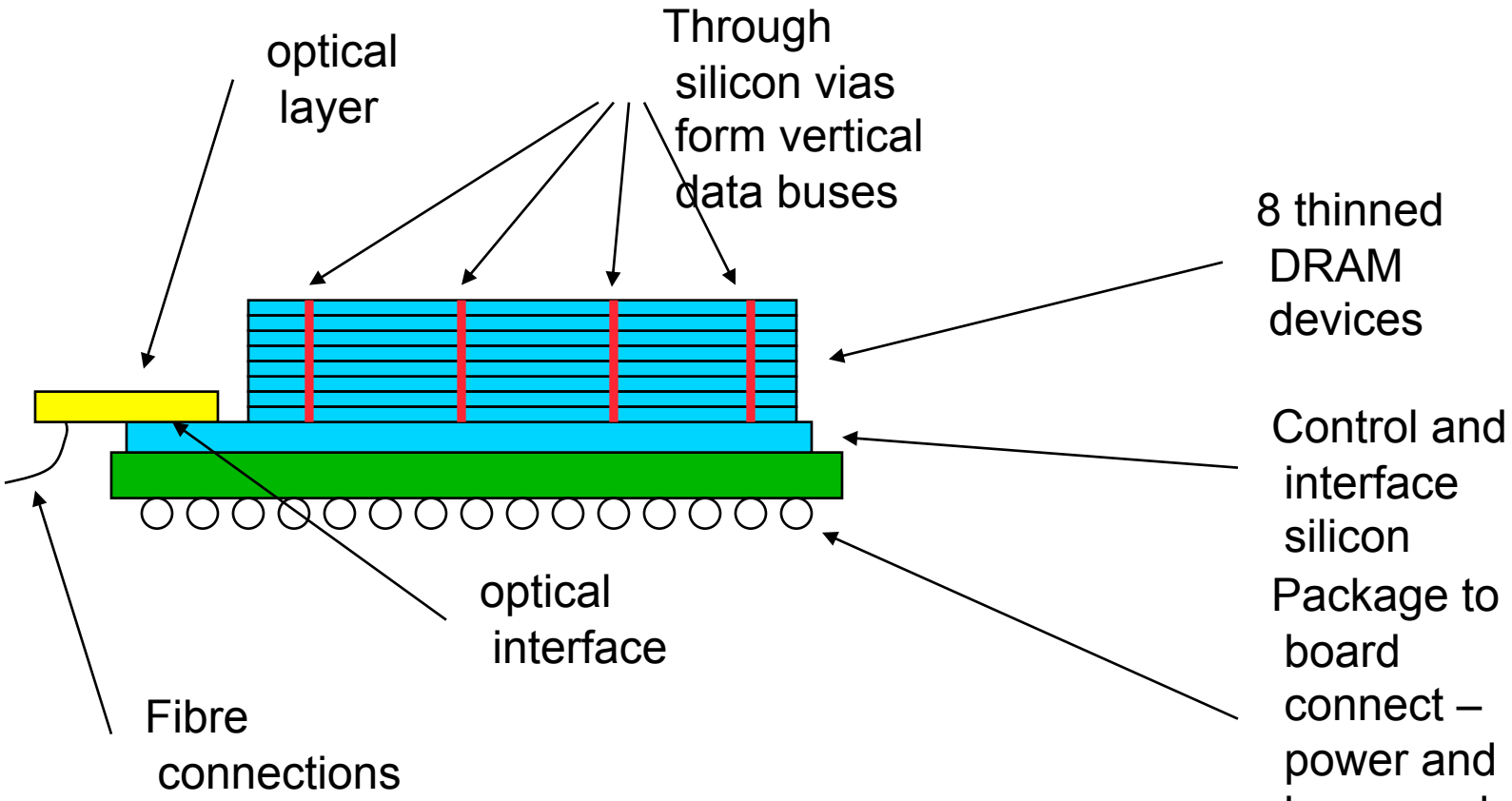
- 8 stacked DRAMs
  - Stacked devices are separate RANKS
  - All lines between DRAMS shared except ChipSelect
- Single controller
- Controller functions:-
  - Global IO
  - Optical interface + buffering
  - Refresh control
- Either
  - Single optical interface point + electrical global IO on interface chip
  - Multiple optical interfaces points + reduced global IO



DRAM changes

- No Global IO wiring
  - Global IO is all the wiring that multiplexes the data from the various DRAM sub blocks. Uses significant time, area and power
- Through Silicon Vias pitch matched to sense amps /output buffers
- Non-array functions, (controller, refresh, etc.) migrated to interface chip where possible.
- Maybe no area overhead if vias take up similar space to global IO?
  - to be determined

Stacked OCM – single optical interface



Multiple interface mux select vertical bus to optical out

NVM and Memory/Storage Implications

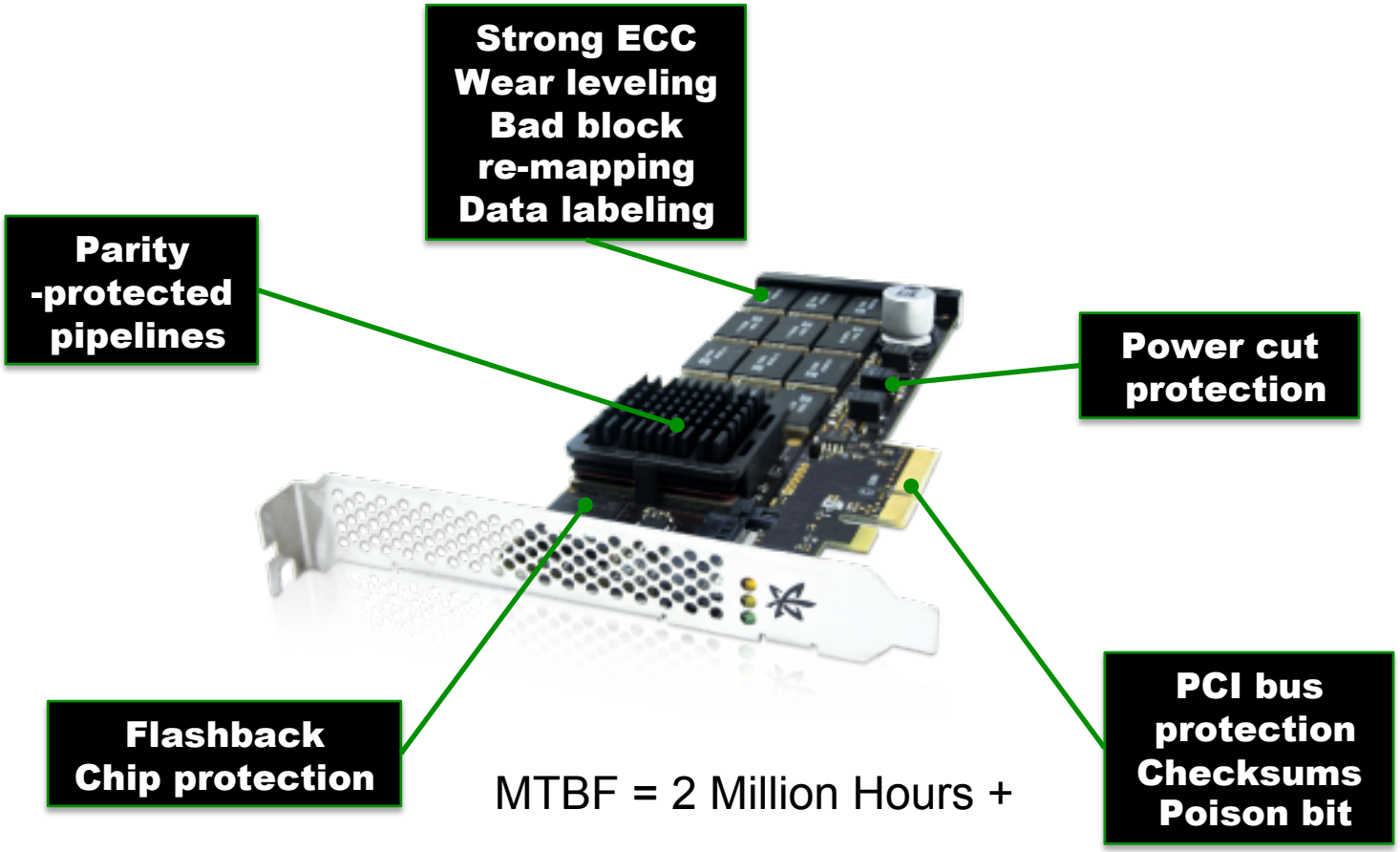
FLASH Dominates Today's Products

- **Mature development**
  - **ubiquitous use in embedded applications**
    - » **mobile phones, MP3 players, automotive**
  - **tiered storage option or solid state disk**
    - » **faster but more expensive than HDD**
    - » **naturally block oriented – similar to disk**
    - » **has some of the same problems**
      - **reliability**
      - **translation layer requirement in the controller**
        - **wear leveling required for FLASH however**
- **Interesting products from database perspective**
  - **ioFusion**
    - » **improved interface – PCIe rather than stodgy SATA**
  - **Oracle's Sun storage F5100 Flash array**
  - **Flash SO-DIMMs starting to appear**
    - » **JEDEC protocol compliant (a somewhat forced choice)**

ioFusion

- Don't call their stuff SSD
  - they prefer to be ioMemory
- Comparison
  - disk: 15K rpmm, random seek 3.5ms, 500 IOPS
    - » big culprit is SATA interface
  - ioMemory
    - » 30 us latency (SLC Nand)
    - » 140K IOPS
- LLNL using purpose built ioFusion boards
  - checkpointing
    - » reduce checkpoint time by 100x
      - basic difference in IOPS

ioFusion



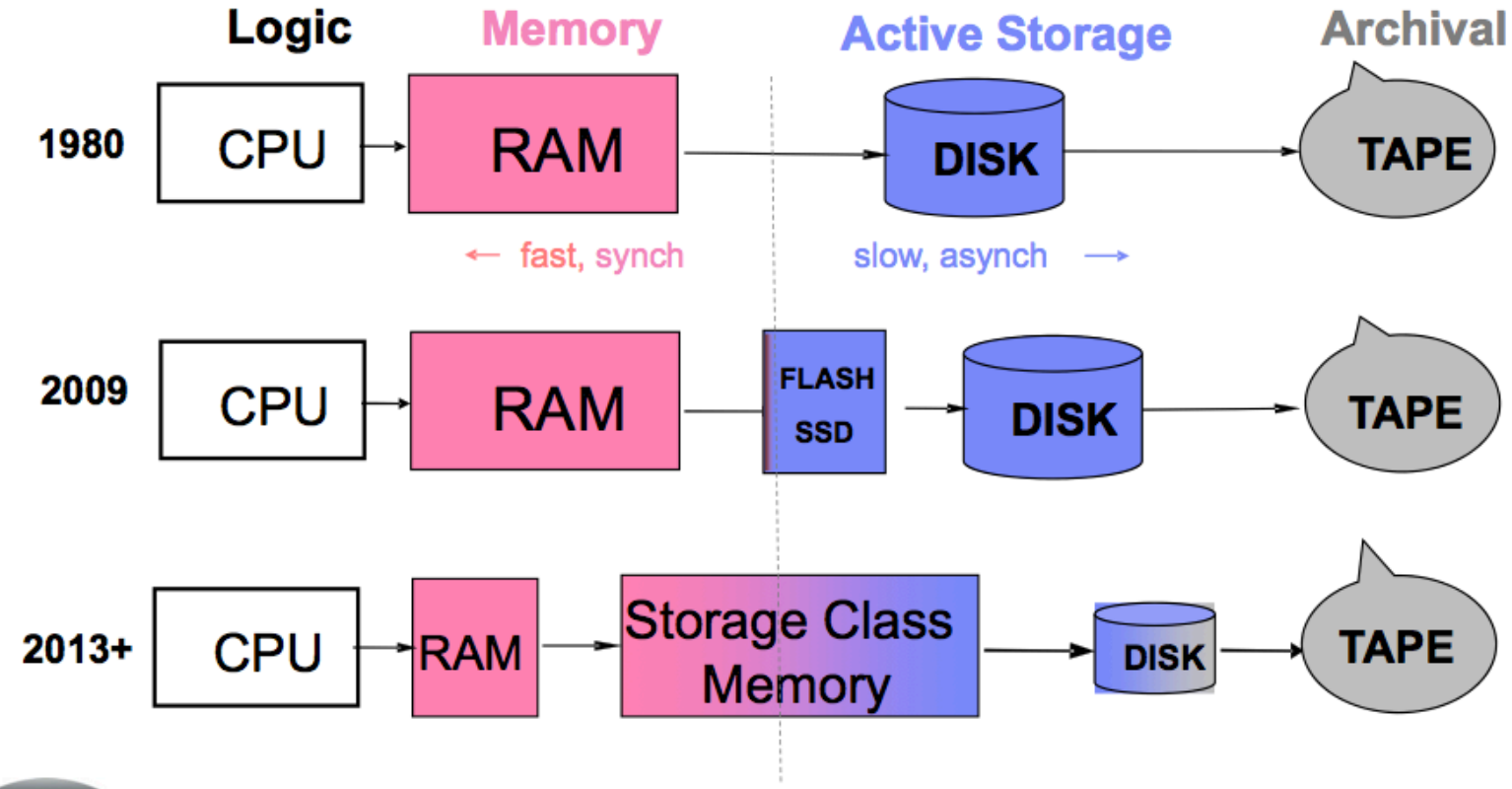
MTBF = 2 Million Hours +

F5100 Factoids

- Oracle's ZFS file system management
  - optimized for Flash storage
- Key specs
  - > 1M IOPS
  - 2 TB in 1U form factor
    - » 300 watts however
- Claimed benefits
  - accelerates DB apps by 2x
  - I/O service times 15x faster
  - 100x less power compared to same capacity HDD's
    - » somewhat specious claim
    - » guess: must be some normalized performance assumption here

System Evolution

Education SNIA



Check out SNIA Tutorial:

"The Future of Solid State Storage"

FLASH Issues

- Good for read mostly applications
- For heavy write traffic things are getting ugly
  - claimed 10^6 write cycles comes down
    - » 32 nm today: 10-20K reliable writes SLC
      - 5K for MLC
    - » 22 nm expected in 9 months
      - 1K writes for MLC, 4K for SLC expected
    - » NO PROJECTIONS past 22 nm
      - open question: is 22 nm end-of-life for FLASH?
  - 1 promising option (others: Intel eMLC)
    - » several Israel startups using analog "test interface"
      - initial results show 10-100x reliability improvement
    - » so maybe there is hope
      - but the digital FTL control layer grows analog hair
      - 1/5 of access latency (50ns) is device based
      - 4/5 is FTL (~200 ns)
      - Amdahl's law has something to say about this

2005 NVRAM Outlook

	Flash		FeRAM	MRAM	PCM	Probe Storage
Cell Type	NOR 1T	NAND 1T	1T/1C	1T/1R	1T/1R	AFM-based
Cell Size (F^2)	10	4 or 5	30-100	30-50	8-16	0.4 (no litho)
Endurance W/R	10^6/inf		10^12/10^12	>10^14/inf	10^12/inf	10^5-10^12/10^7-inf
Read Time (random)	60 ns	60 ns / serial	40 + 80 ns (read + write destructive read)	30 ns	60 ns	2-20ms
Write time (byte)	1 us	200 us / page		30 ns	10 ns	0.1-1 ms for each tip
Erase time (byte)	1 s / sector	2 ms / block		30 ns	150 ns	< 1 us /bit
Scalability	Fair	Fair	Poor	Poor	Good	Very Good
Scalability Limits	Tunnel oxide, HV		Capacitor	Current Density	Litho	None
Multi-bit capability	Yes		No	No	Yes	No
Relative cost/bit	Medium	Low	High	High	Medium	Very low
Maturity	Very high		Medium	Low	Low	Very low

Source: Pirovano ICMTD-2005

Last 5 Years

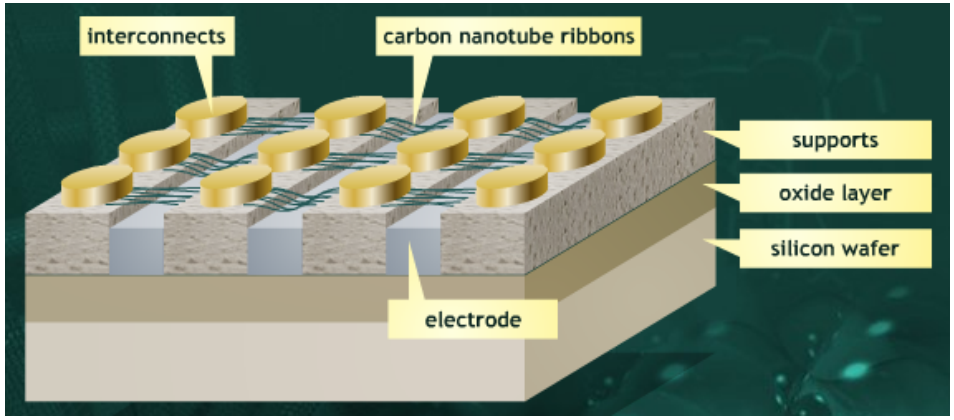
- **Improvements follow investment**
  - **STT-MRAM developed by Grandis**
    - » partners with Hynix in 2008
    - » Samsung, DARPA, NSF … jump on board
  - **PCRAM**
    - » heavy backing by IBM
    - » development by Toshiba & Samsung
      - Samsung shipping 512 Mb parts in FLASH compliant package
- **2 new technologies appear**
  - **CNT/NRAM – Nantero**
  - **Memristor – HP**
    - » 2010 partnership w/ Hynix just announced
  - **both show promise to beat STT and PCRAM options**
    - » smaller cells
    - » lower read and write currents
    - » faster read and write access times

2010 View (source Grandis)

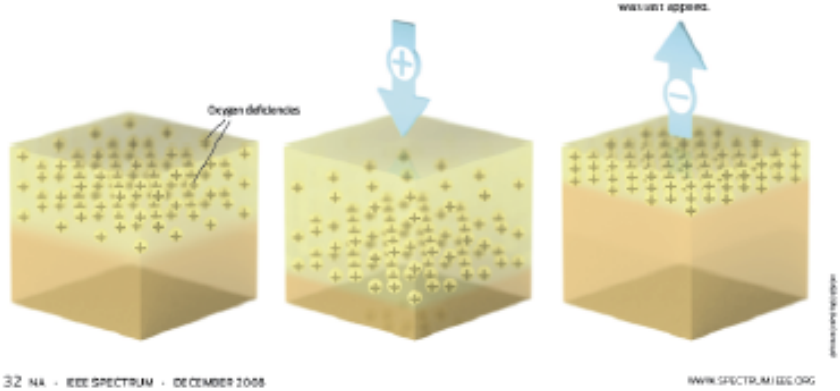
	SRAM	DRAM	Flash (NOR)	Flash (NAND)	FeRAM	MRAM	PRAM	RRAM	STT-RAM
Non-volatile	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cell size (F^2)	50–120	6–10	10	5	15–34	16–40	6–12	6–10	6–20
Read time (ns)	1–100	30	10	50	20–80	3–20	20–50	10–50	2–20
Write / Erase time (ns)	1–100	15	1 µs / 10 ms	1 ms / 0.1 ms	50 / 50	3–20	50 / 120	10–50	2–20
Endurance	10^{16}	10^{16}	10^5	10^5	10^{12}	$>10^{15}$	10^8	10^8	$>10^{15}$
Write power	Low	Low	Very high	Very high	Low	High	Low	Low	Low
Other power consumption	Current leakage	Refresh current	None	None	None	None	None	None	None
High voltage required	No	3 V	6–8 V	16–20 V	2–3 V	3 V	1.5–3 V	1.5–3 V	<1.5 V
	Existing products						*Prototype*		

New NVM Contenders

Nantero – CNT/NRAM



HP – Memristor – 2010 partnership w/ Hynix announced



- **Both have promise to be better than STT or PCRAM**
  - **too early to tell if or when this promise will prove true**

Conclusions

- Memory is a bottleneck
  - it will only be more true as socket throughput goes up & memory pressure increases
- Numerous options for improvement
  - improvements: MC, DRAM uArch, interfaces, …
    - » the change won't be cheap
    - » big challenge is what to expose to the OS or app SW?
      - SW folks need to be proactive here
  - NVM options abound
    - » what forms of tier'ing make sense?
    - » check-pointing is ideal candidate
      - ideal choice is cheap (energy & time) writes
        - OK for reads to be a bit more expensive – hopefully they are rare
- Biggest dark cloud on the horizon
  - cost

THANKS

Q & A