Carnegie Mellon University SFU SIMON FRASER UNIVERSITY ETH zürich SAFARI

Demystifying Complex Workload–DRAM Interactions:

An Experimental Study

Saugata Ghose, Tianshi Li,
Nastaran Hajinazar, Damla Senol Cali, Onur Mutlu

June 27, 2019

Why Study Workload–DRAM Interactions?

- Manufacturers are developing many new types of DRAM
  - DRAM limits performance, energy improvements: new types may overcome some limitations
  - Memory systems now serve a very diverse set of applications: can no longer take a one-size-fits-all approach
- So which DRAM type works best with which application?
  - Difficult to understand intuitively due to the complexity of the interaction
  - Can't be tested methodically on real systems: new type needs a new CPU
- We perform a wide-ranging experimental study to uncover the combined behavior of workloads and DRAM types
  - 115 prevalent/emerging applications and multiprogrammed workloads
  - 9 modern DRAM types: DDR3, DDR4, GDDR5, HBM, HMC, LPDDR3, LPDDR4, Wide I/O, Wide I/O 2

Insights From Our Extensive Experimental Study

SAFARI

1. DDR4 worse than DDR3 for most desktop/scientific applications
2. HMC worse than DDR3 for apps with high spatial locality
3. HMC good for highly memory intensive multiprogrammed workloads
4. LPDDRx performance gap over DDRx worsens as the intensity increases
5. LPDDR4 saves power over DDR3 without losing much performance for high-intensity multiprogrammed workloads
6. Multithreaded programs with irregular accesses become throughput-bound as the input problem size increases
7. Server/cloud applications don't benefit from high-throughput DRAM
8. LPDDRx saves energy for server/cloud apps without a large performance penalty
9. Intensive multimedia apps benefit from high-throughput DRAM with wide rows
10. Network accelerators experience high queuing latencies, benefit from parallelism
11. GDDR5 is more energy efficient than DDR3 for high-throughput accelerators
12. OS routines benefit most from low-latency DRAM that exploits spatial locality

Insights From Our Extensive Experimental Study

- 12 key observations on the combined DRAM–workload behavior
- Most significant experimental observations
  - **Newer is not always better:**
    DDR4, HMC (newer DRAM types) do not outperform the older DDR3 type for many families of applications
  - **Power savings don't always sacrifice performance:**
    Some low-power DRAM types perform close to or better than standard-power DRAM types when bandwidth demand is very high
  - **Heterogeneous systems cannot rely on a single DRAM type:**
    The ideal DRAM for a heterogeneous system depends heavily on the predominant functions performed by the system (e.g., multimedia, network processing, compute)
  - **Increasing memory latency to increase parallelism can be harmful:**
    OS routines exhibit extremely high spatial locality, and can't benefit from the high parallelism provided by many newer DRAM types

Background: DRAM Basics

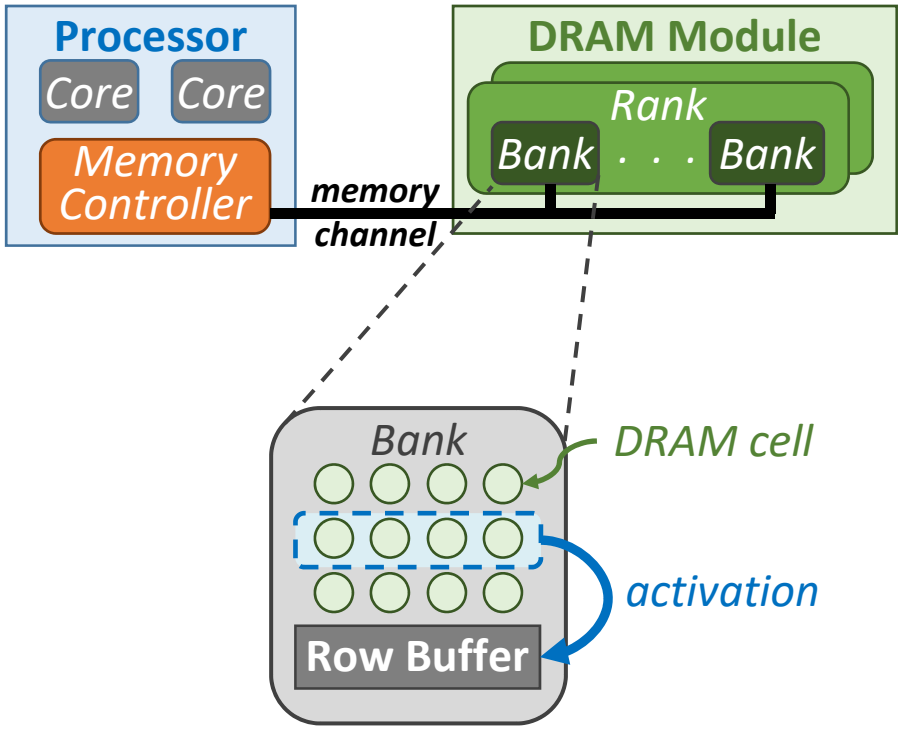
Characterization Methodology & Metrics

Major Observations from Our Study

Key Takeaways and Conclusion

Simplified View: DRAM Organization & Operation

- A **memory channel** connects the processor with DRAM
  - Controller issues commands
  - Many systems have multiple independent channels
- DRAM has multiple **banks**
  - Conceptually: 2D array of data
  - Banks can operate in parallel
  - Many banks share one channel
- Data is stored in **DRAM cells**
  - A cell stores charge in a capacitor to represent a one-bit data value
  - Before reading/writing, a full row of cells (e.g., 8kB in DDR3) must be **activated** (opened)

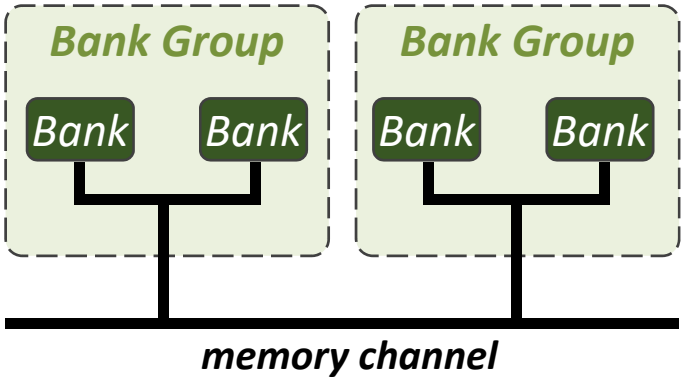


Modern DRAM Types: Comparison to DDR3

SAFARI

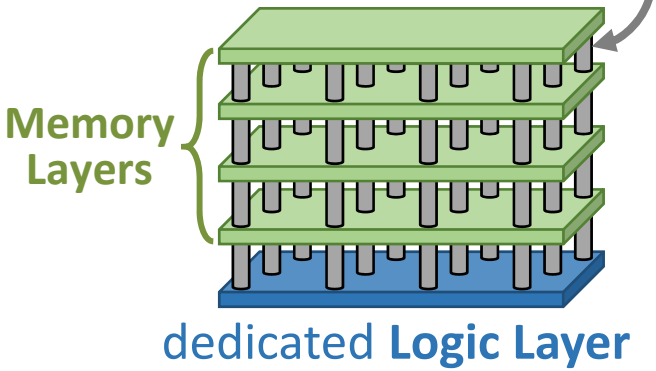
DRAM Type	Banks per Rank	Bank Groups	3D-Stacked	Low-Power
DDR3	8			
DDR4	16	✓	*increased latency*	
GDDR5	16	✓	*increased area/power*	
HBM High-Bandwidth Memory	16		✓	
HMC Hybrid Memory Cube	256	*narrower rows, higher latency*	✓	
Wide I/O	4		✓	✓
Wide I/O 2	8		✓	✓
LPDDR3	8			✓
LPDDR4	16			✓

- Bank groups



- 3D-stacked DRAM

high bandwidth with Through-Silicon Vias (TSVs)



Background: DRAM Basics

Characterization Methodology & Metrics

Major Observations from Our Study

Key Takeaways and Conclusion

Characterization Methodology

- Modified version of Ramulator
  - New shared cache model: comes within 6.1% of gem5 (a detailed, rigorously validated out-of-order processor simulator)
  - New HMC model *(to be released in July)*
  - Open source: available at https://github.com/CMU-SAFARI/ramulator/
- Energy: modeled for available memories using DRAMPower
- Applications
  - **87 different applications, which we group into six diverse families**
    - » Desktop/scientific
    - » Server/cloud
    - » Multimedia acceleration
    - » Network acceleration
    - » General-purpose GPU (GPGPU)
    - » Common OS routines
  - Includes single-threaded and multi-threaded workloads
  - **28 multiprogrammed workloads**
  - Open source: available at https://github.com/CMU-SAFARI/MemBen/

Architectural Parameters

- 9 DRAM types
  - Standard power
    - » DDR3-2133
    - » DDR4-3200
    - » GDDR5-7000
    - » HBM
    - » HMC 2.0
  - Low power
    - » LPDDR3-2133
    - » LPDDR4-3200
    - » Wide I/O
    - » Wide I/O 2
- 4GB of DRAM total, typically distributed across 4 channels
- Processor cores
  - Single-thread/multiprogrammed: 4 cores, 4.0 GHz
  - Multithreaded: 20 cores, 2 threads per core, 2.2 GHz
  - Out-of-order cores, 128-entry ROB, 4-wide issue
- 64kB/256kB private L1/L2 caches
- 2MB of shared last-level cache (LLC) per core

Metrics to Capture Parallelism and Contention

- **Parallelism**
  - Existing metrics (e.g., memory-level parallelism, bank-level parallelism) are typically from the perspective of an application
  - We want to measure how well hardware parallelism is being utilized
  - New metric: **bank parallelism utilization (BPU)**

$$\frac{\sum_i \#\text{ active banks in cycle } i}{\#\text{ cycles memory is active}}$$

- **Contention**
  - Dictates the time that a read/write request must wait
  - Refined metric: **row buffer locality (RBL)**
    - » **Row hit:** no time
    - » **Row miss:** activation latency
    - » **Row conflict:** activation latency *plus* latency of requests to already-open row

Outline

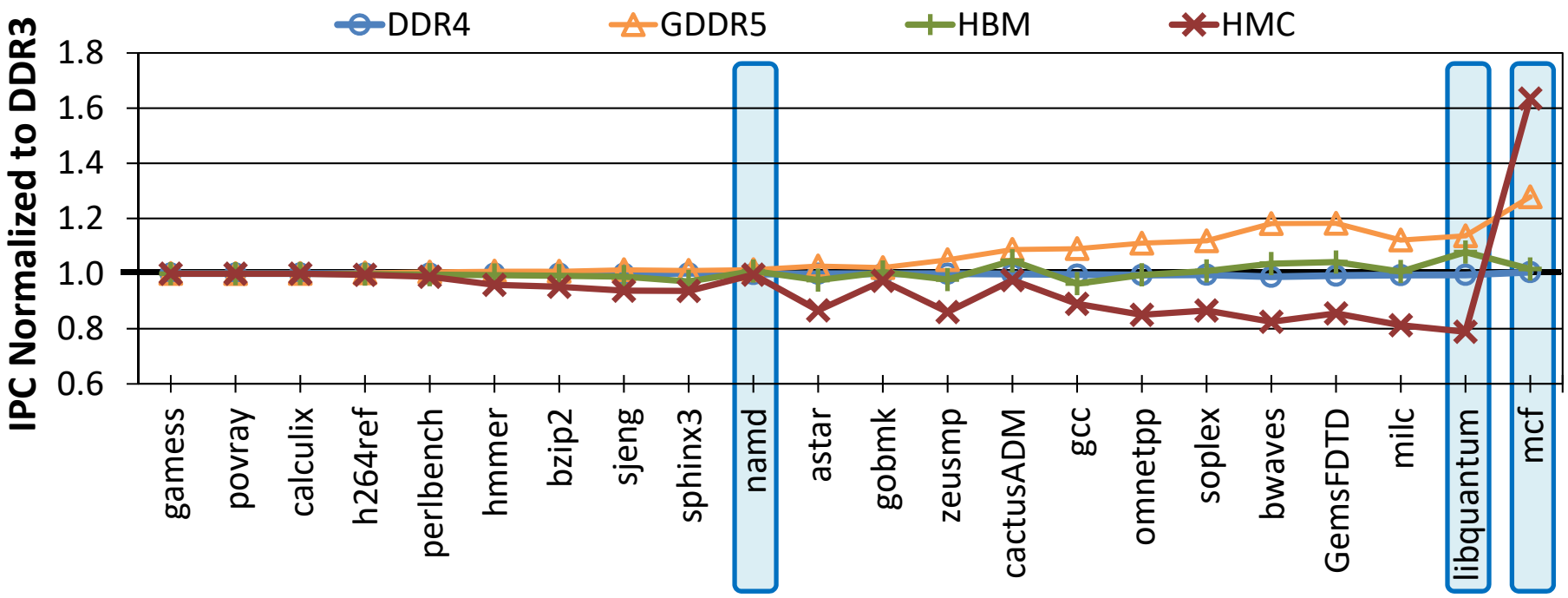
Background: DRAM Basics

Characterization Methodology & Metrics

Major Observations from Our Study

Key Takeaways and Conclusion

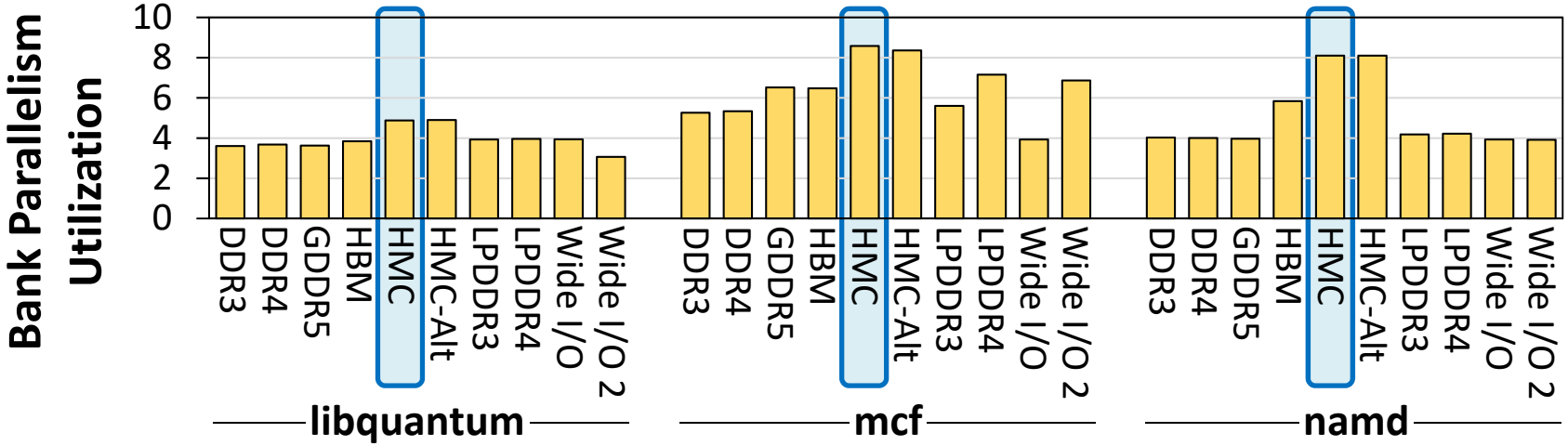
1. Newer Is Not Always Better: Desktop/Scientific



- DDR4 performs near identically to DDR3 most of the time
- HMC performs *worse* than DDR3 in many (but not all) cases

Why? Let's focus on three representative applications

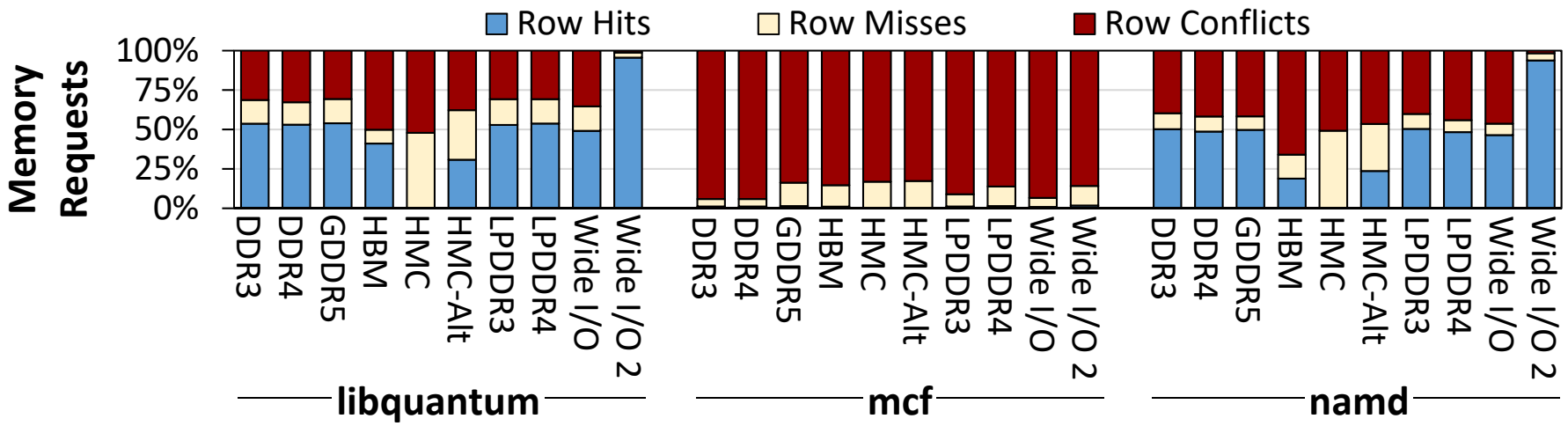
1. Newer Is Not Always Better: BPU



Performance does not improve when BPU is constant

- *libquantum* (HMC slower): cannot even take advantage of the 32 banks available in DDR3 – additional banks don't help
- *mcf* (HMC faster): makes use of more banks due to its much higher memory intensity (70 LLC misses per kiloinstruction)
- *namd* (HMC same): HMC has higher BPU due to the shorter rows, but increased latency cancels out performance benefits

1. Newer Is Not Always Better: Row Buffer Locality

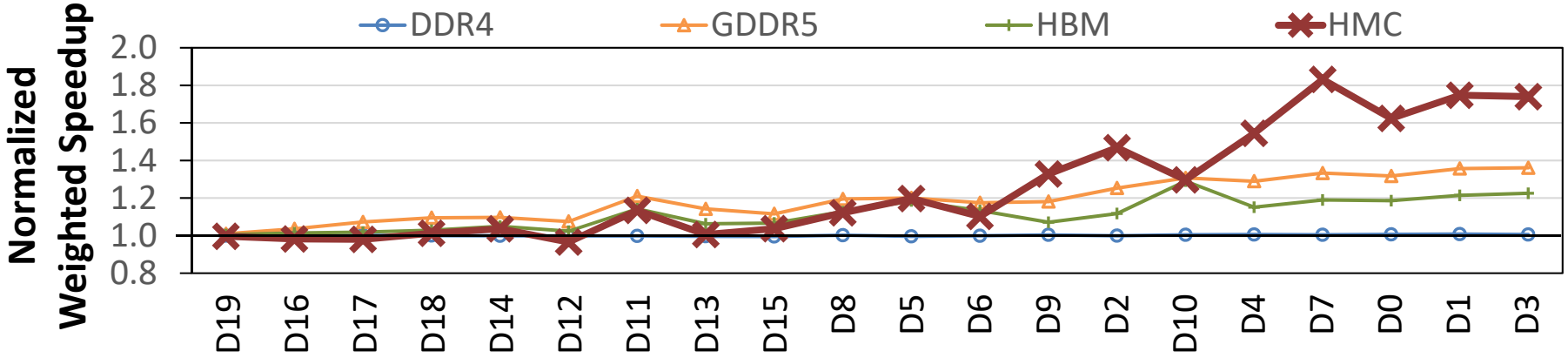


HMC works best when there are not many row hits

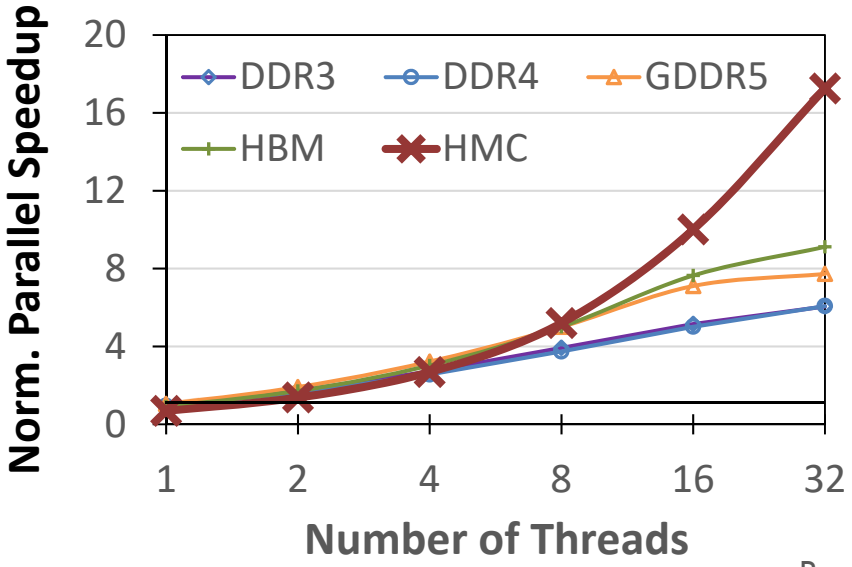
- HMC's poor spatial locality means that read/write requests are rarely row hits – leads to higher latency
- HMC only benefits applications that exhibit few row hits and high memory intensity

When Does HMC Perform Better?

- Multiprogrammed workloads
  (e.g., desktop/scientific: average speedup of 17.0% over DDR3)



- Multithreaded applications with large problem sizes and high thread counts
  (right: *miniFE*, 64x64x64)

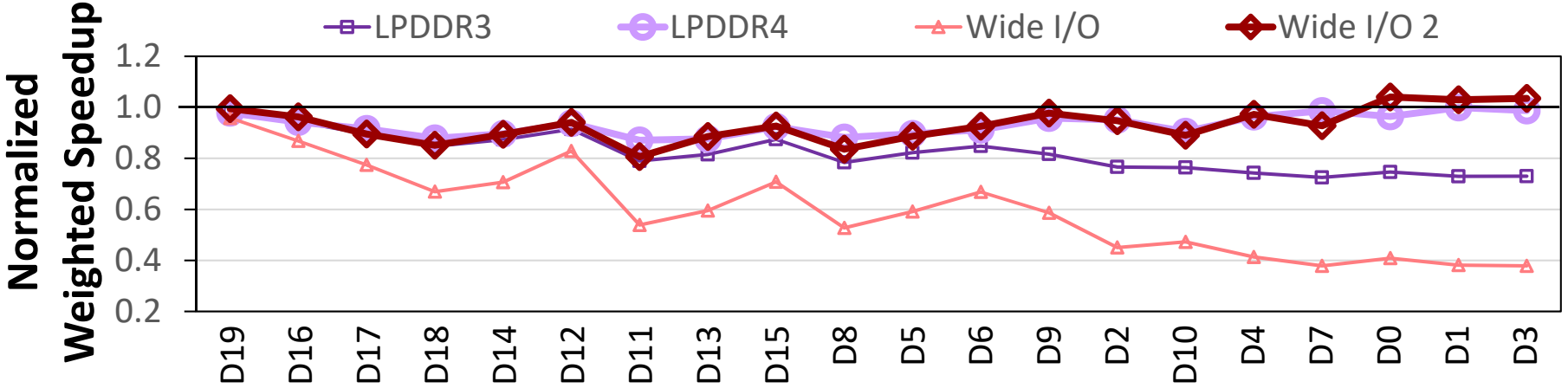


- Network accelerators

2. Low Power Does Not Always Hurt Performance

SAFARI

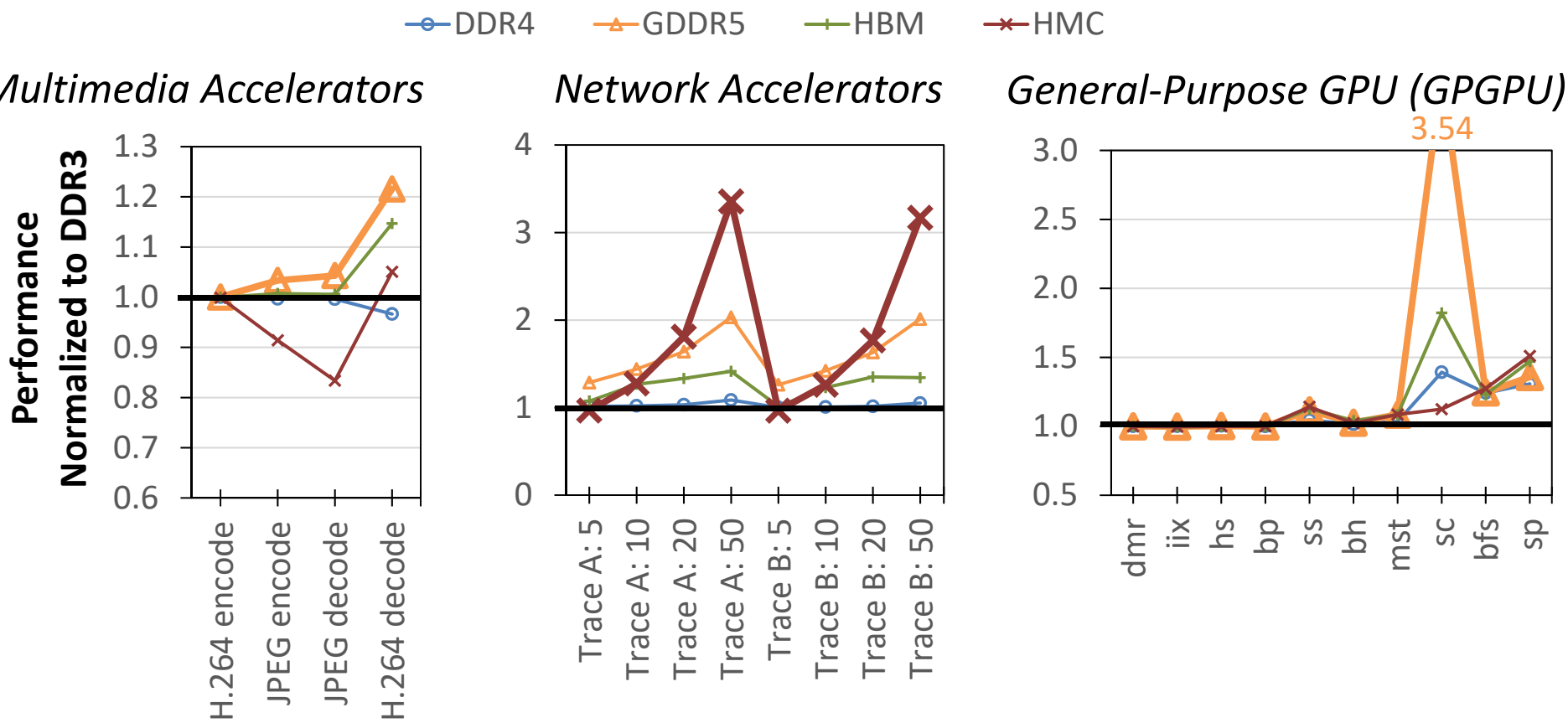
- For many families of applications, using low-power DRAM types can hurt the performance significantly
- Some exceptions, such as multiprogrammed desktop/scientific workloads (below) and multimedia accelerators



Some low-power DRAM types perform well when the bandwidth demand is very high

- LPDDR4: 68.2% less energy than DDR3, only 7.0% slower

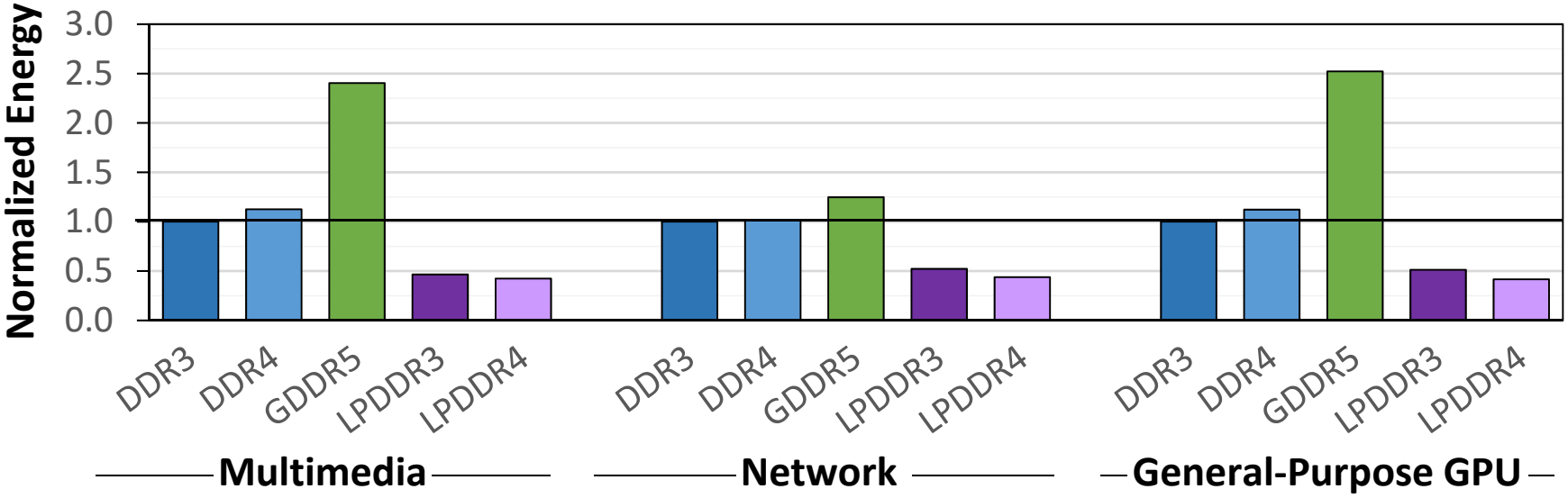
3. No One Best Type for Heterogeneous Systems



GDDR5 for multimedia acceleration, GPGPU

HMC for network acceleration

3. Heterogeneous Systems: Energy



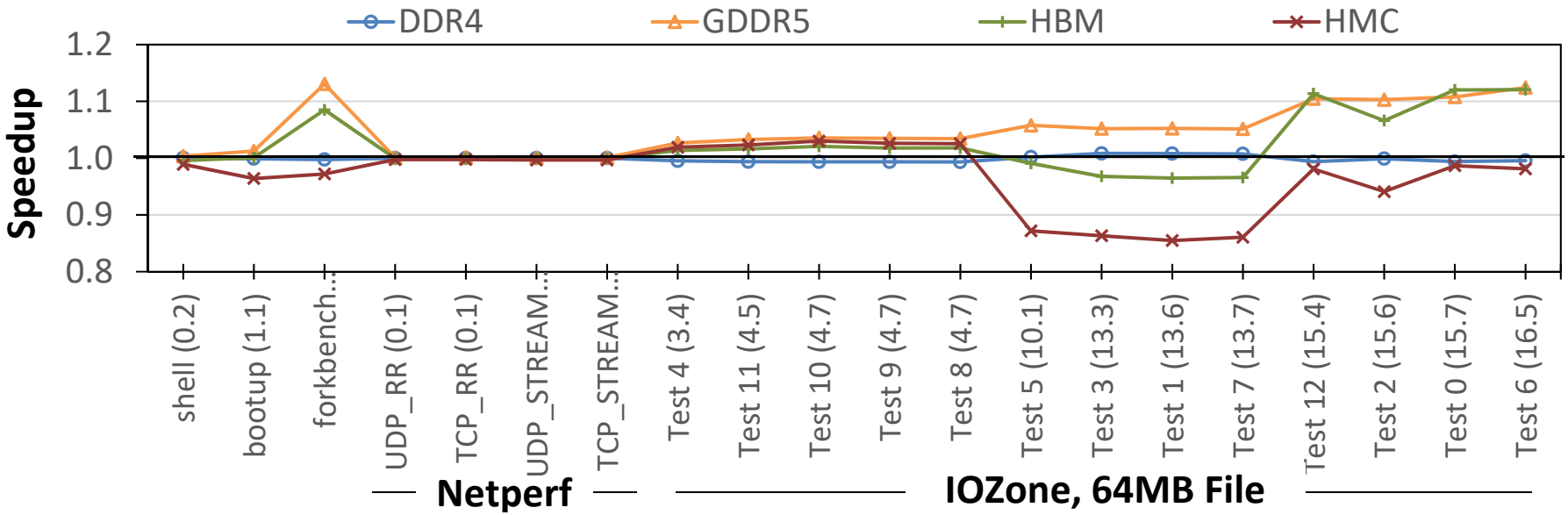
GDDR5 is much more energy efficient than DDR3 for accelerators with high memory throughput

- Example applications
  - Network accelerator workloads
  - Several GPGPU applications: *sc*, *bfs*, *sp*

4. Need for Lower Access Latency: Performance

SAFARI

- New DRAM types often increase access latency in order to provide more banks, higher throughput
- Many applications can't make up for the increased latency
  - Especially true of common OS routines (e.g., file I/O, process forking)

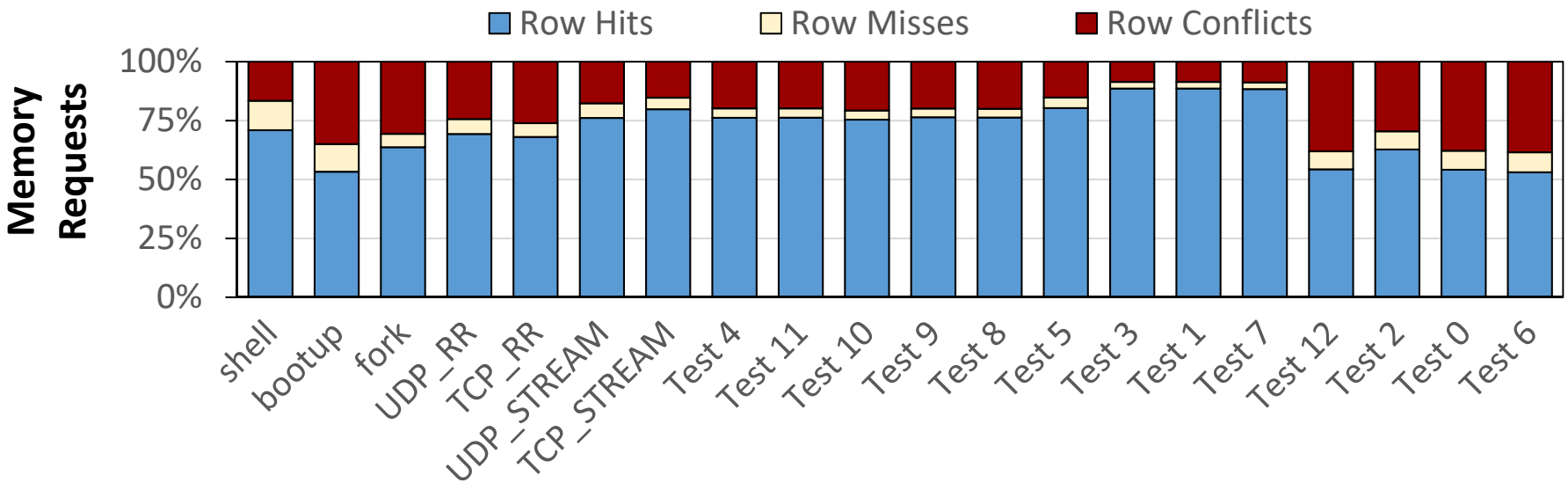


- A variety of desktop/scientific, server/cloud, GPGPU applications

Several applications don't benefit from more parallelism

4. Need for Lower Access Latency: Insight

- Many applications still exhibit high spatial locality
- We can see an example of this from the row buffer locality of the common OS routines



Requests often go to a single open row:
few opportunities to exploit bank parallelism

Outline

Background: DRAM Basics

Characterization Methodology & Metrics

Major Observations from Our Study

Key Takeaways and Conclusion

Key Takeaways

1. DRAM latency remains a critical bottleneck for many applications

2. Bank parallelism is not fully utilized by a wide variety of our applications

3. Spatial locality continues to provide significant performance benefits if it is exploited by the memory subsystem

4. For some classes of applications, low-power memory can provide energy savings without sacrificing significant performance

Conclusion

- Manufacturers are developing many new types of DRAM
  - DRAM limits performance, energy improvements: new types may overcome some limitations
  - Memory systems now serve a very diverse set of applications: can no longer take a one-size-fits-all approach
  - Difficult to intuitively determine which DRAM–workload pair works best
- We perform a wide-ranging experimental study to uncover the combined behavior of workloads, DRAM types
  - 115 prevalent/emerging applications and multiprogrammed workloads
  - 9 modern DRAM types
- 12 key observations on DRAM–workload behavior

Open-source tools: https://github.com/CMU-SAFARI/ramulator

Full paper: https://arxiv.org/pdf/1902.07609

Carnegie Mellon University SFU SIMON FRASER UNIVERSITY ETH zürich SAFARI

Demystifying Complex Workload–DRAM Interactions:

An Experimental Study

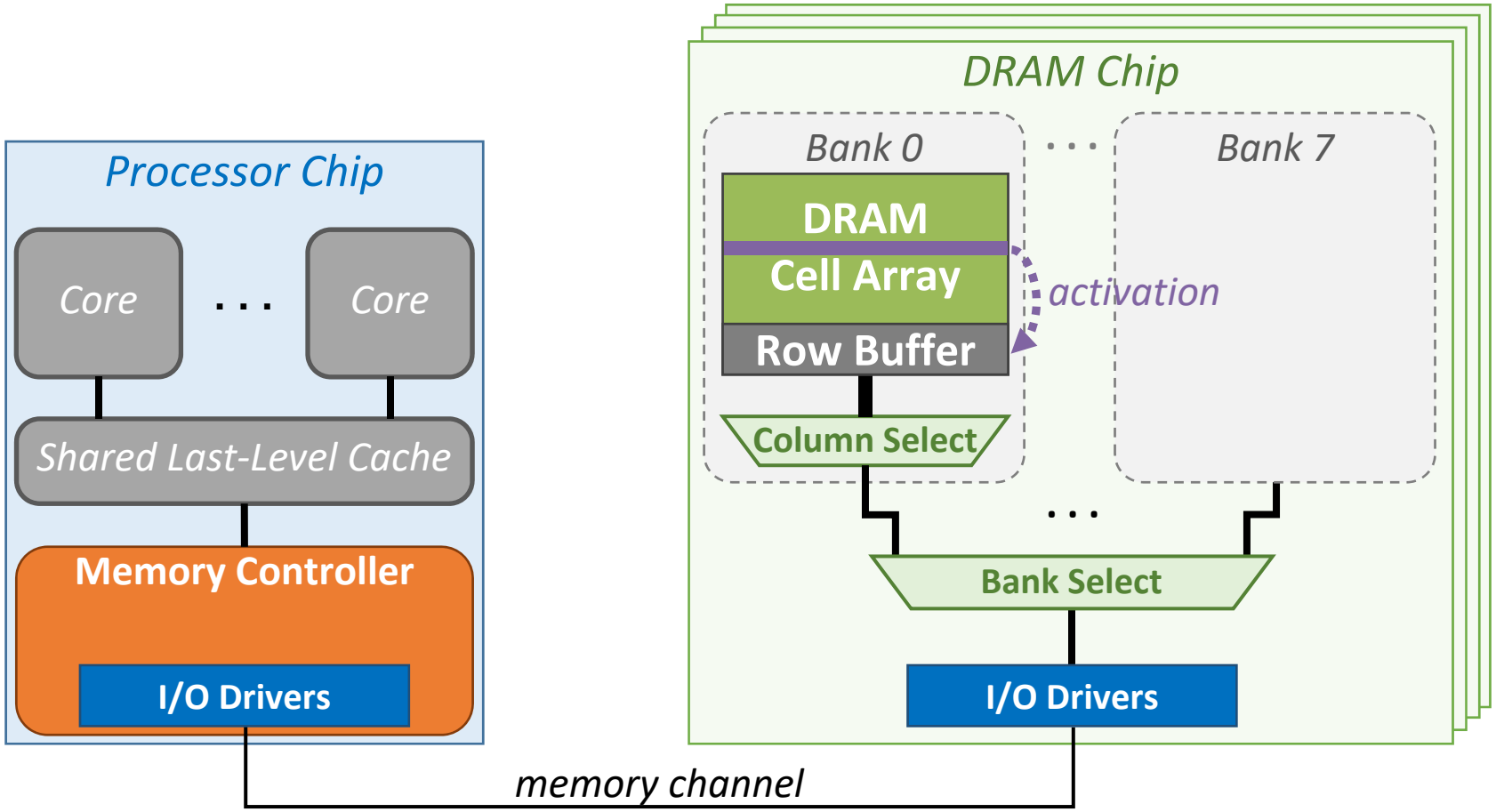
Saugata Ghose, Tianshi Li,
Nastaran Hajinazar, Damla Senol Cali, Onur Mutlu

Open-source tools: https://github.com/CMU-SAFARI/ramulator

Full paper: https://arxiv.org/pdf/1902.07609

Backup Slides

Simplified DRAM Organization and Operation



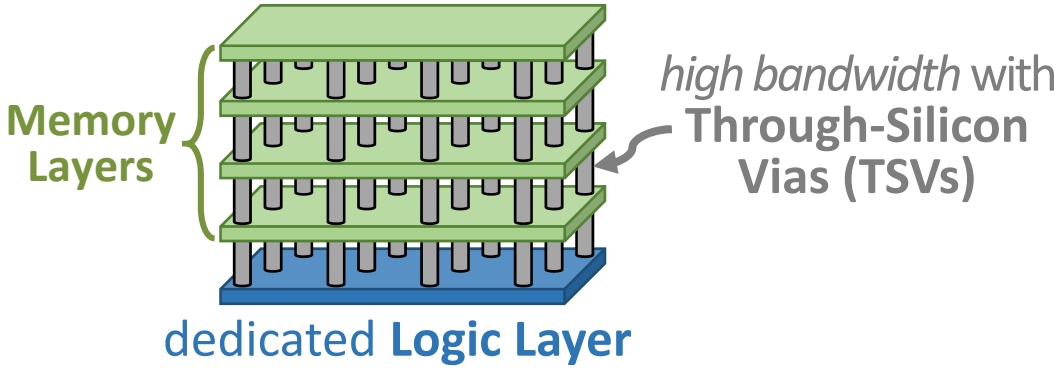
- Fundamental DRAM commands: activate, read, write, precharge
- One row of DRAM: 8 kB
- One cache line of data: 64 B

Modern DRAM Types: Double Data Rate DRAM

- DDR3
  - **Double Data Rate:** bursts of data sent on memory channel at both positive and negative clock edge
  - 8 banks per rank
- DDR4
  - More banks per rank using **bank groups** – results in 11-14% **higher latency**
  - 16 banks per rank
- GDDR5
  - Also uses bank groups, but **consumes more area/energy** instead of increasing latency
  - Sends double the data per cycle compared to DDR3 using a **faster clock**
  - 16 banks per rank

Modern DRAM Types: 3D-Stacked DRAM

- 3D die stacking enables new DRAM types with high capacity, much higher bandwidth



- High-Bandwidth Memory (HBM)
  - Instead of GDDR5's faster clocks, HBM uses **multiple channels per module** to increase throughput
  - 16 banks per rank
- Hybrid Memory Cube (HMC)
  - Partitions 3D-stacked DRAM into **vaults:** small vertical slices of DRAM with multiple banks
  - Contains many **more banks**, but a row is 97% **narrower** than in DDR3: **cannot exploit as much spatial locality**
  - 256 banks per rank

Modern DRAM Types: Low-Power DRAM

SAFARI

- **LPDDR3 and LPDDR4**
  - Low-power variants of DDR3/DDR4
  - Increased memory latency, limited capacity
  - LPDDR3: 8 banks per rank
  - LPDDR4: 16 banks per rank

- **Wide I/O and Wide I/O 2**
  - Low-power 3D-stacked DRAM
  - Fewer channels than HBM
  - Wide I/O: 4 banks per rank
  - Wide I/O 2: 8 banks per rank

Key DRAM Properties

DRAM Type	Standard Power				
	DDR3	DDR4	GDDR5	HBM	HMC
Data Rate (MT/s)	2133	3200	7000	1000	2500
Clock Frequency (MHz)	1067	1600	1750	500	1250
Maximum Bandwidth (GBps)	68.3	102.4	224.0	128.0	320.0
Channels/Ranks per Channel	4/1	4/1	4/1	8/1	1/1
Banks per Rank	8	16	16	16	256 (32 vaults)
Channel Width (bits)	64	64	64	128	32
Row Buffer Size	8KB	8KB	8KB	2KB	256B
Row Hit/Miss Latencies (ns)	15.0/26.3	16.7/30.0	13.1/25.1	18.0/32.0	16.8/30.4

DRAM Type	Low Power			
	LPDDR3	LPDDR4	Wide I/O	Wide I/O 2
Data Rate (MT/s)	2133	3200	266	1067
Clock Frequency (MHz)	1067	1600	266	533
Maximum Bandwidth (GBps)	68.3	51.2	17.0	34.1
Channels/Ranks per Channel	4/1	4/1	4/1	4/2
Banks per Rank	8	16	4	8
Channel Width (bits)	64	64	128	64
Row Buffer Size	8KB	4KB	2KB	4KB
Row Hit/Miss Latencies (ns)	21.6/40.3	26.9/45.0	30.1/38.9	22.5/41.3

Detailed Methodology Details

Processor	x86-64 ISA, 128-entry instruction window, 4-wide issue single-thread/multiprogrammed: 4 cores, 4.0 GHz multithreaded: 20 cores, 2 threads per core, 2.2 GHz
Caches	per-core L1: 64 kB, 4-way set associative per-core L2: 256 kB, 4-way set associative shared L3: 2 MB for every core, 8-way set associative
Memory Controller	32/32-entry read/write request queues, FR-FCFS [117, 142], cache line interleaving

- **Workload traces collected using Pin, Bochs**
  - Private caches fully modeled
  - Bochs captures all system calls – important for OS routines, server/cloud
  - Multithread traces capture per-trace interactions (based on [Pelley+ ISCA 2014])
- **GPGPU results run on a full GPGPU-Sim+Ramulator platform**

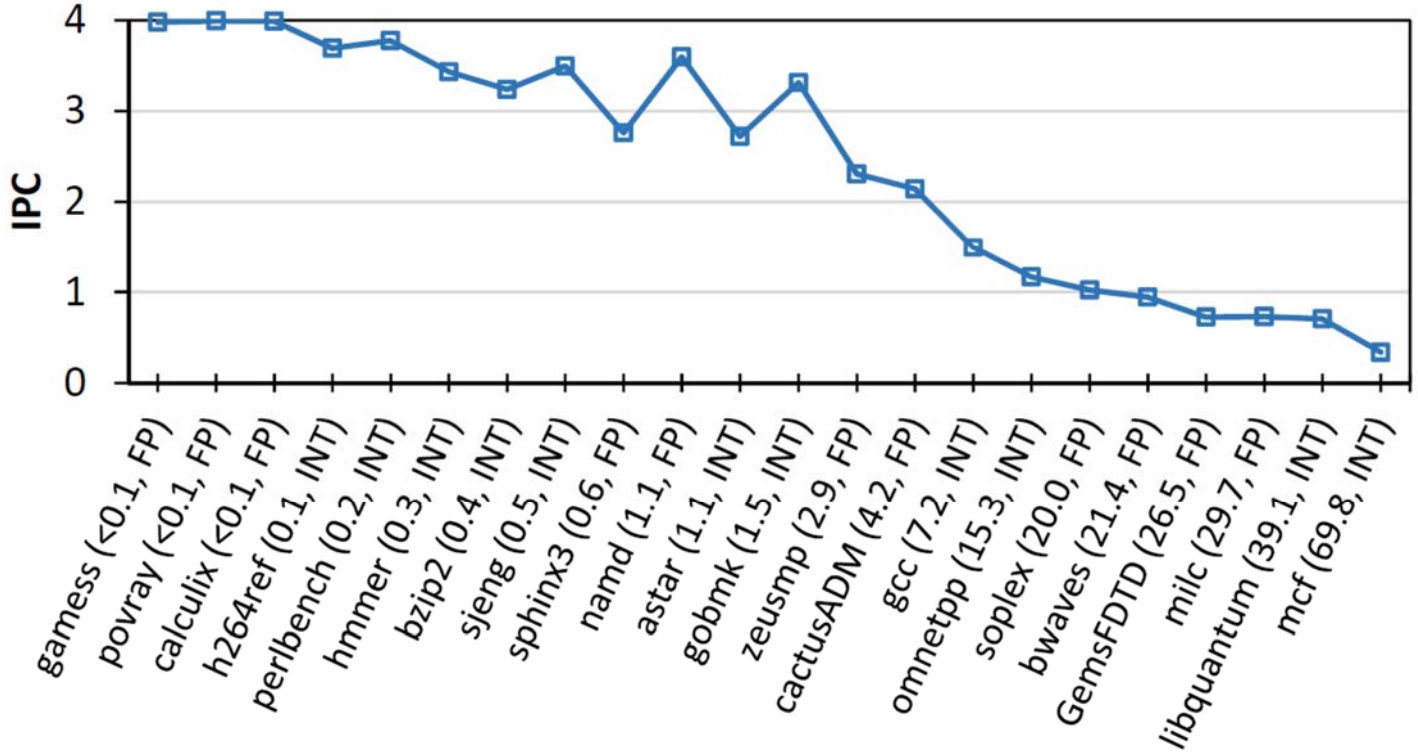
Validating a Trace-Based Ramulator Approach

- We compare
  - gem5 (full-system, cycle accurate CPU simulator) + unmodified Ramulator
  - Modified Ramulator with shared cache model (trace-driven)

- Run all SPEC CPU2006 applications
- Checking trends: normalize each performance value to one application (*gamess*)

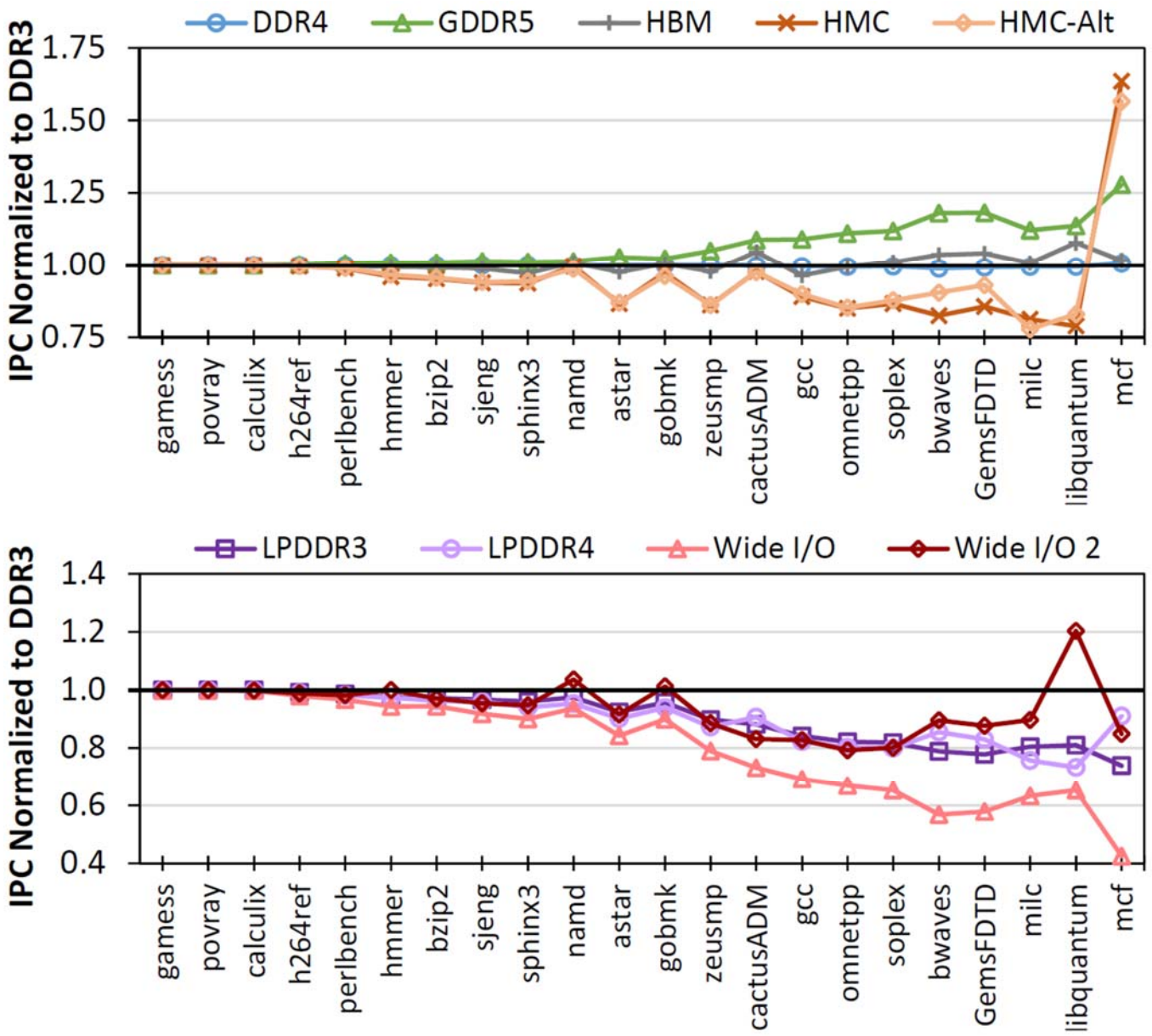
- Average error of Ramulator vs. gem5+Ramulator: 6.1%

Desktop/Scientific: Single-Thread IPC/MPKI



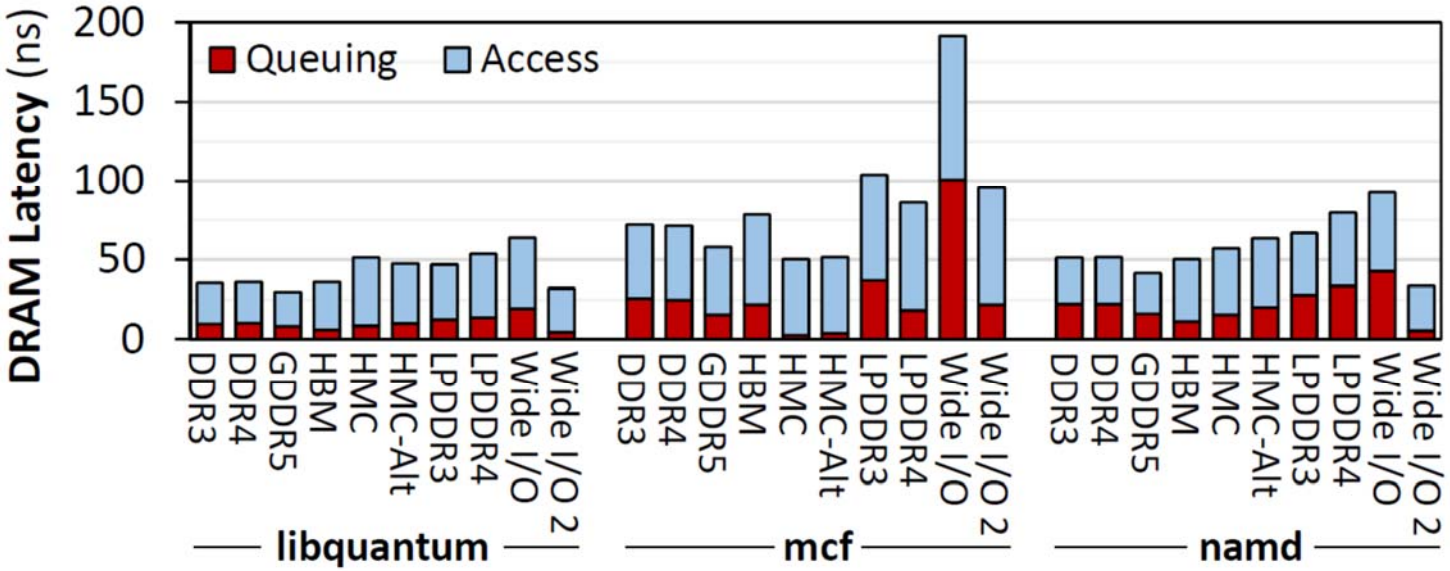
Desktop/Scientific: Single-Thread Performance

SAFARI



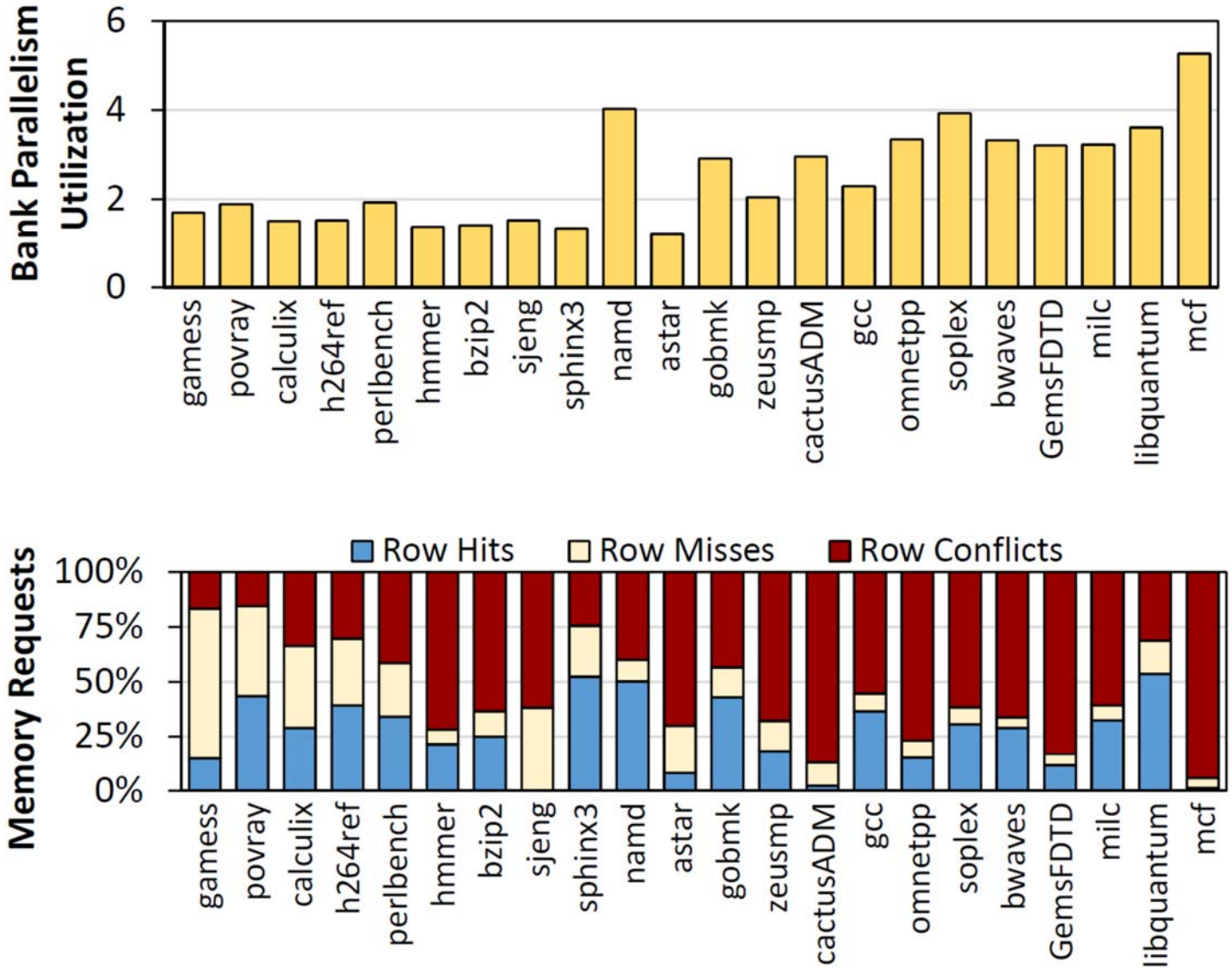
Desktop/Scientific: Single-Thread DRAM Latency

SAFARI

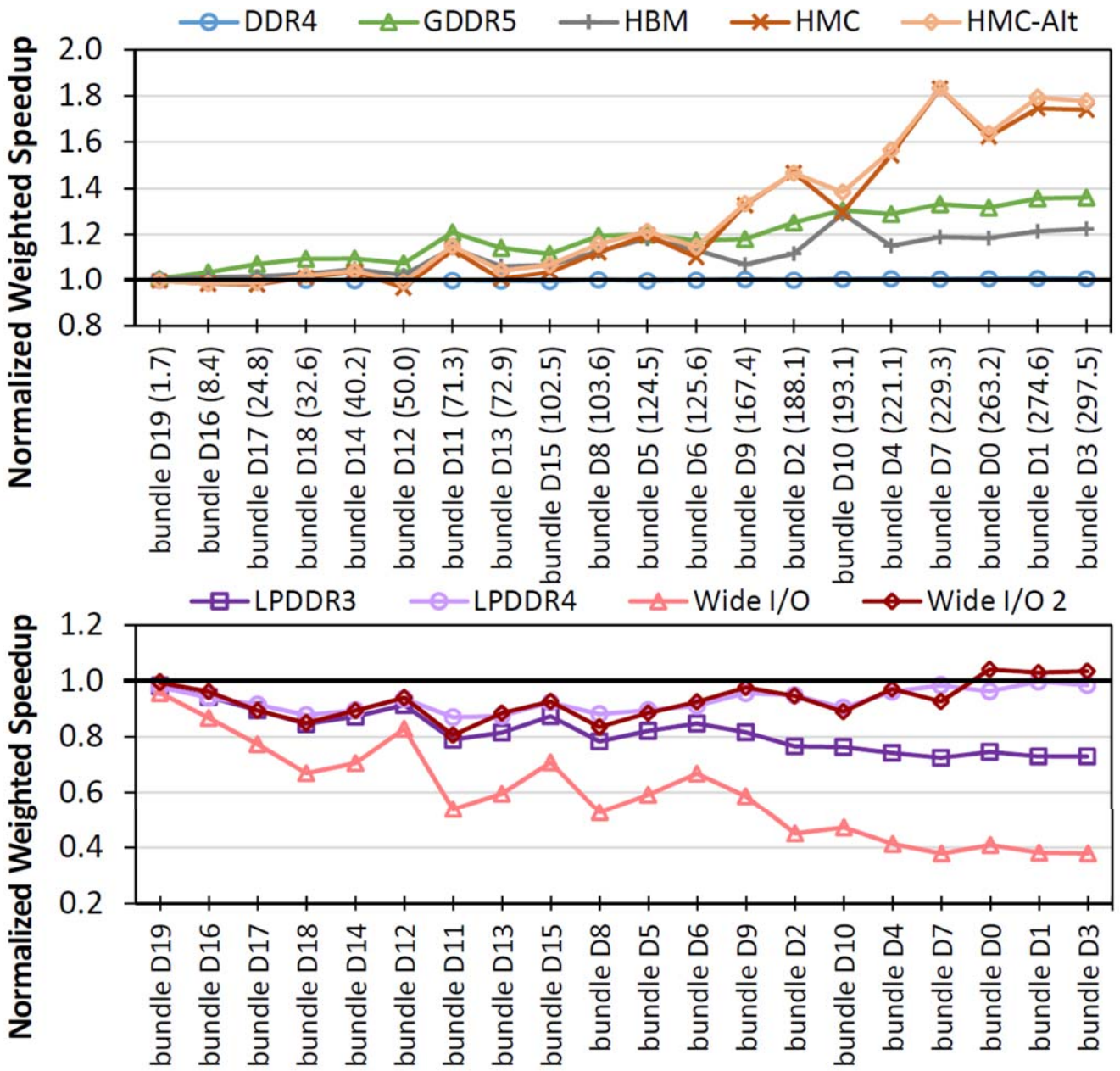


Desktop/Scientific: DDR3 BPU/RBL

SAFARI

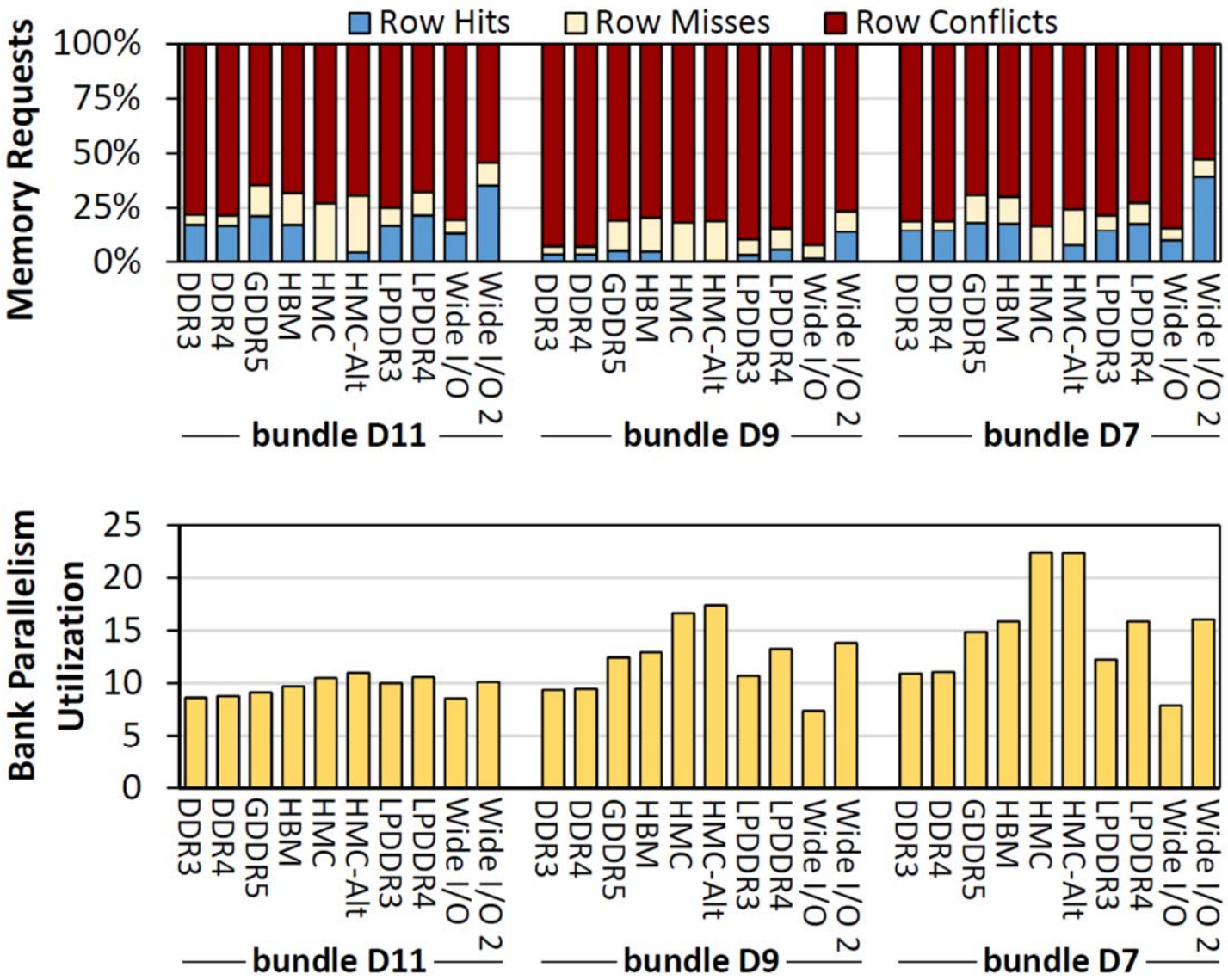


Desktop/Scientific: Multiprogrammed Performance

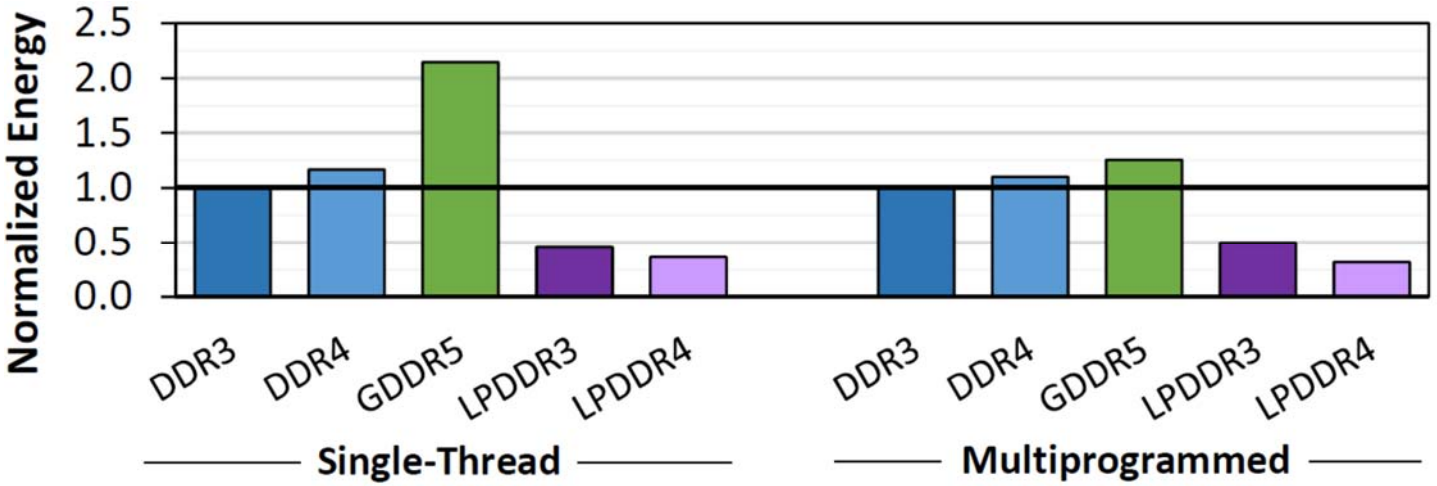


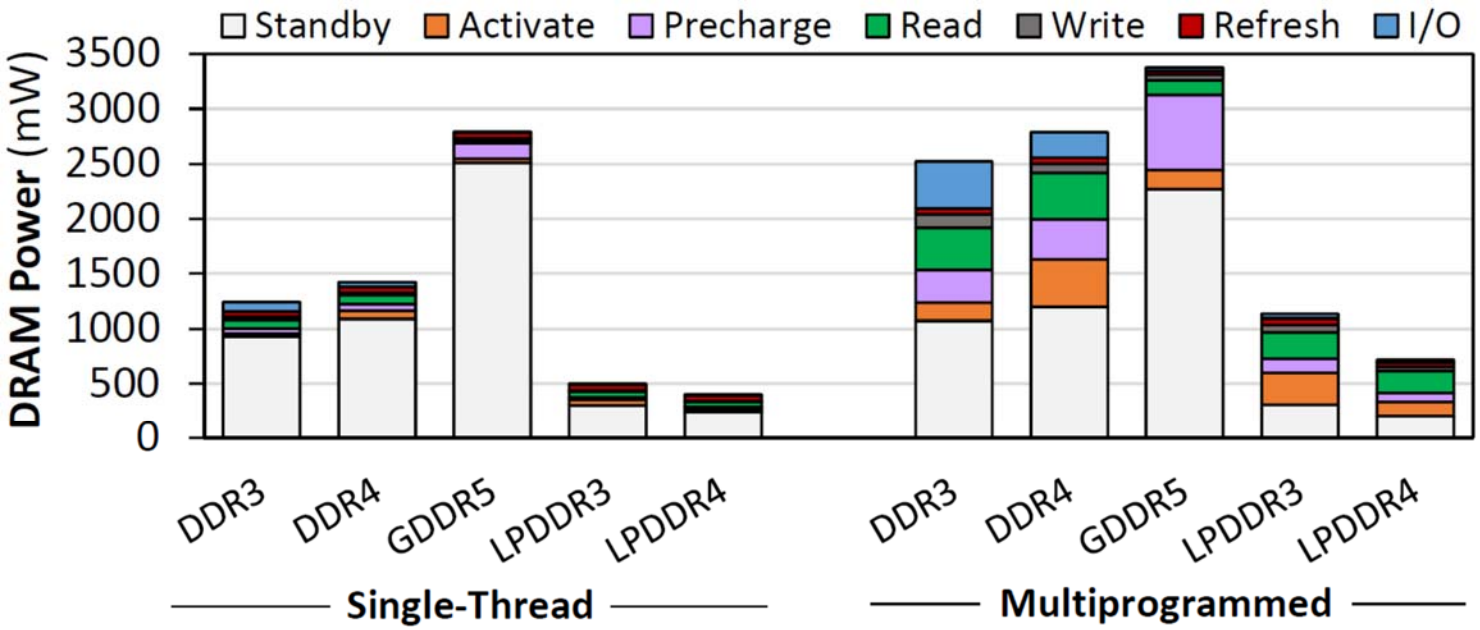
Desktop/Scientific: Multiprogrammed RBL/BPU

SAFARI



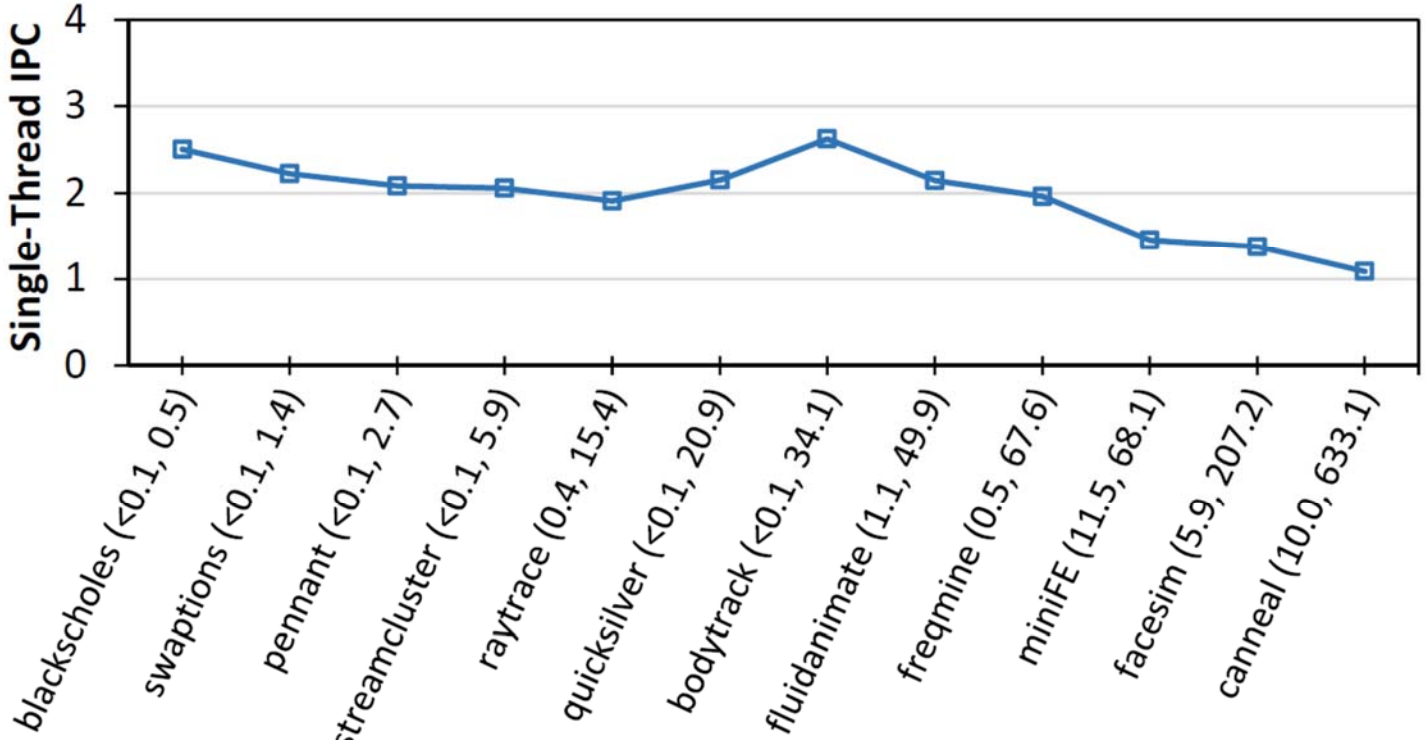
Desktop/Scientific: Energy and Power Breakdown



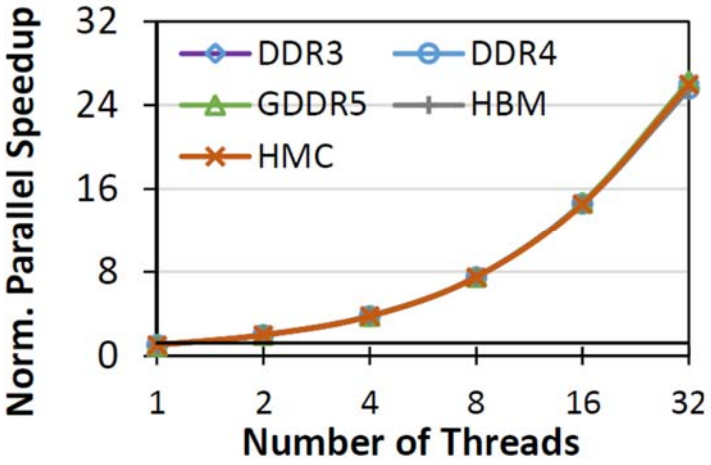


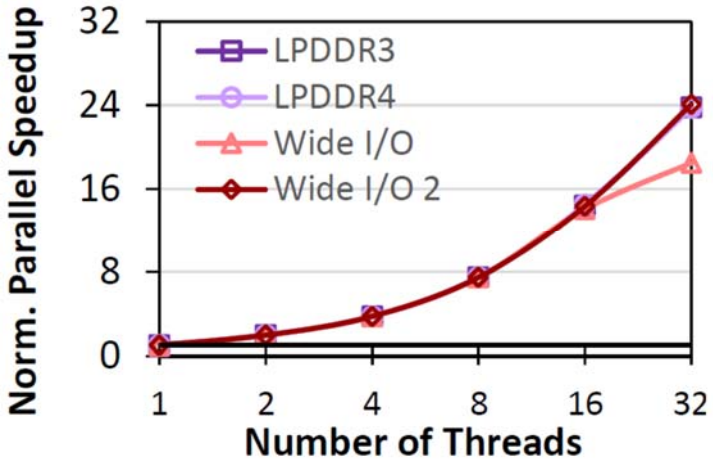
Desktop/Scientific: Multithreaded IPC/MPKIs

SAFARI



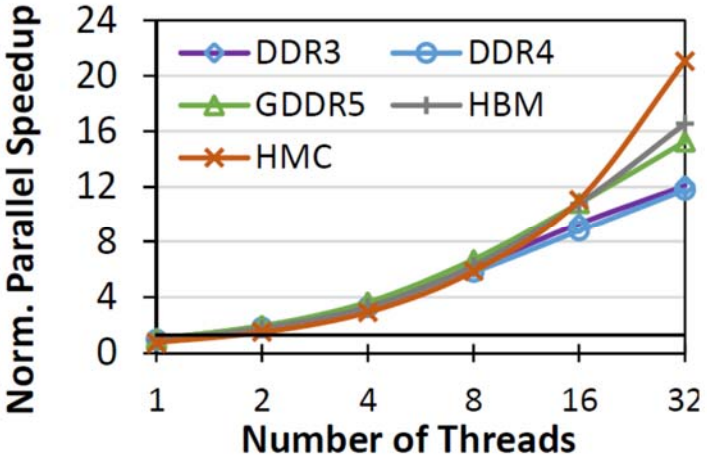
Desktop/Scientific: *quicksilver* Performance

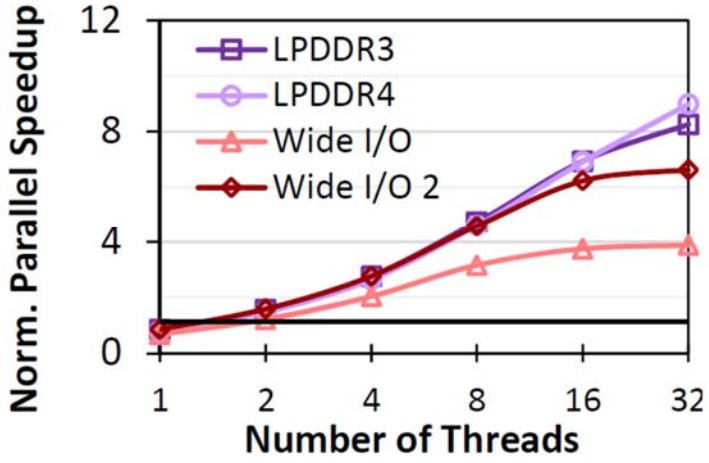




Desktop/Scientific: *facesim* Performance

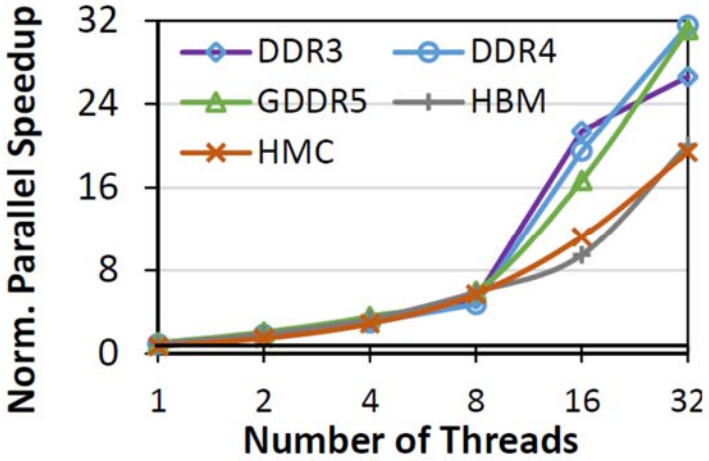
SAFARI

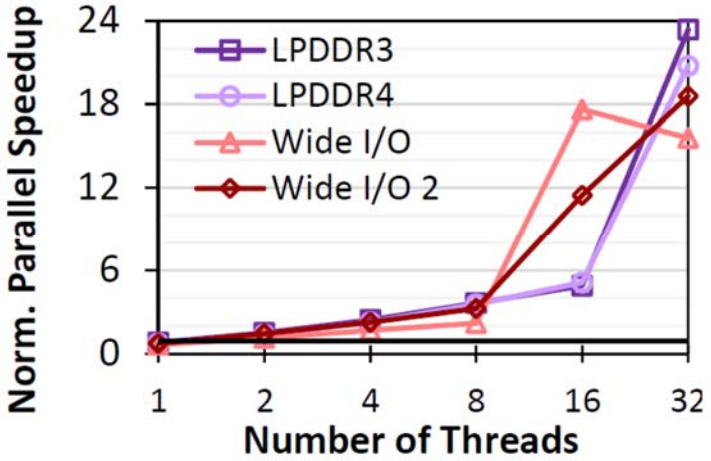




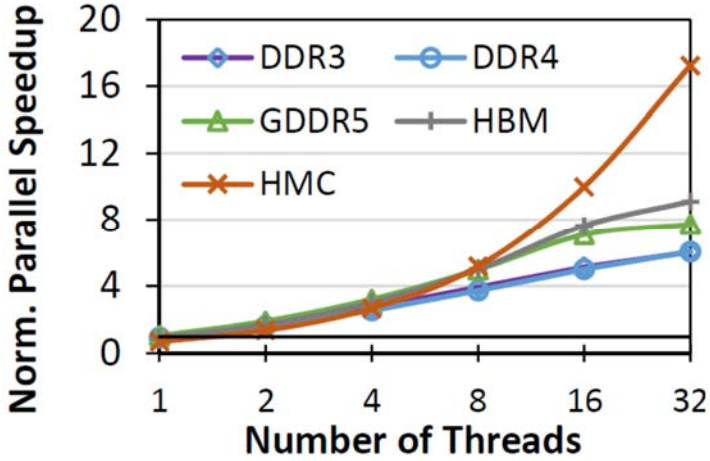
Desktop/Scientific: *miniFE* Performance

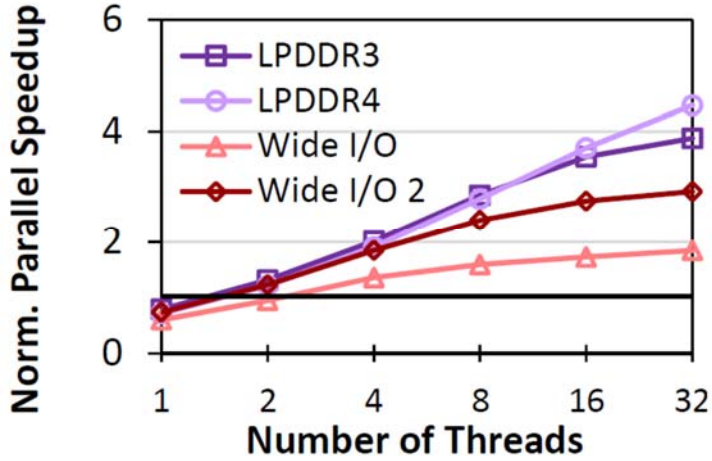
- Input size: 32x32x32



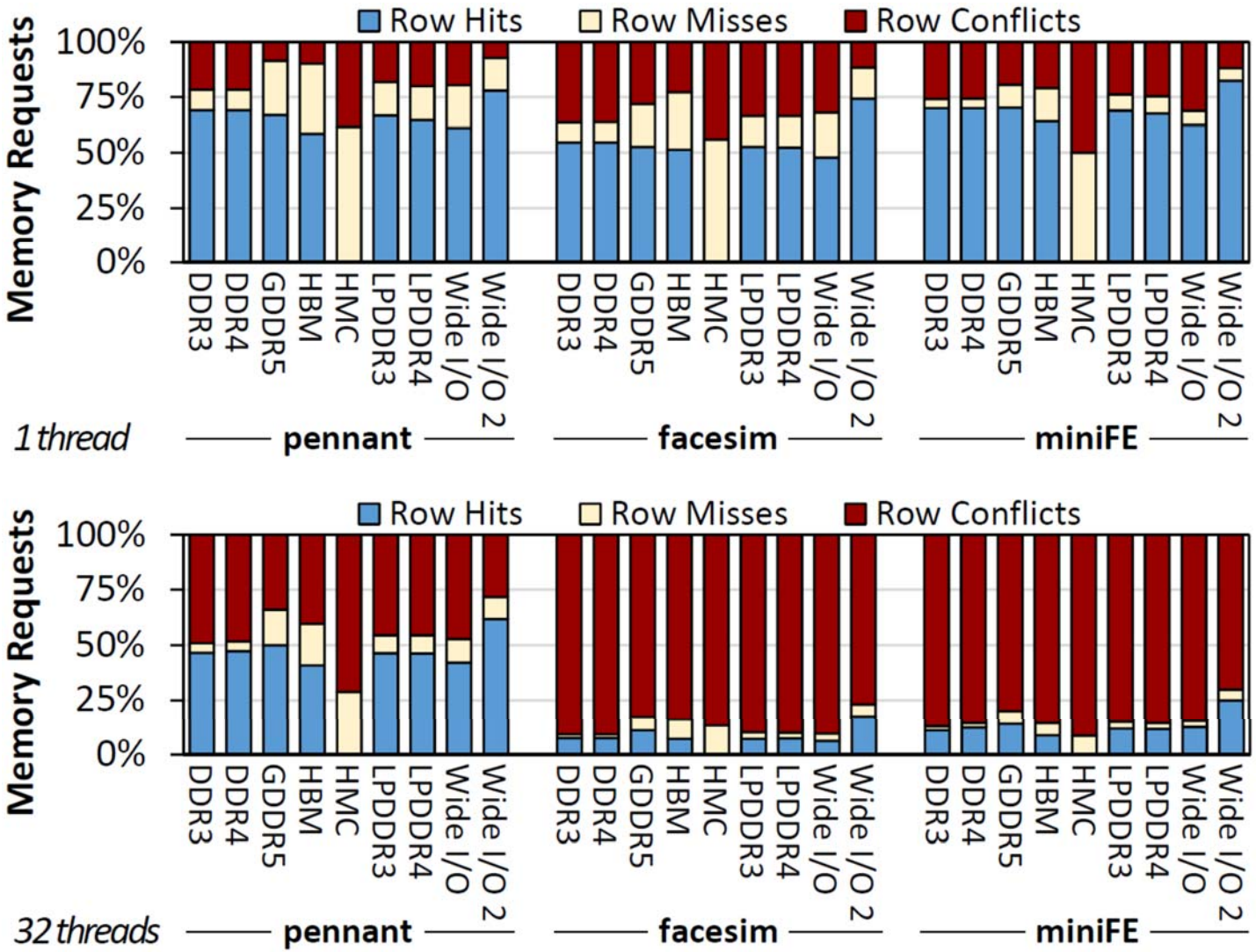


- Input size: 64x64x64

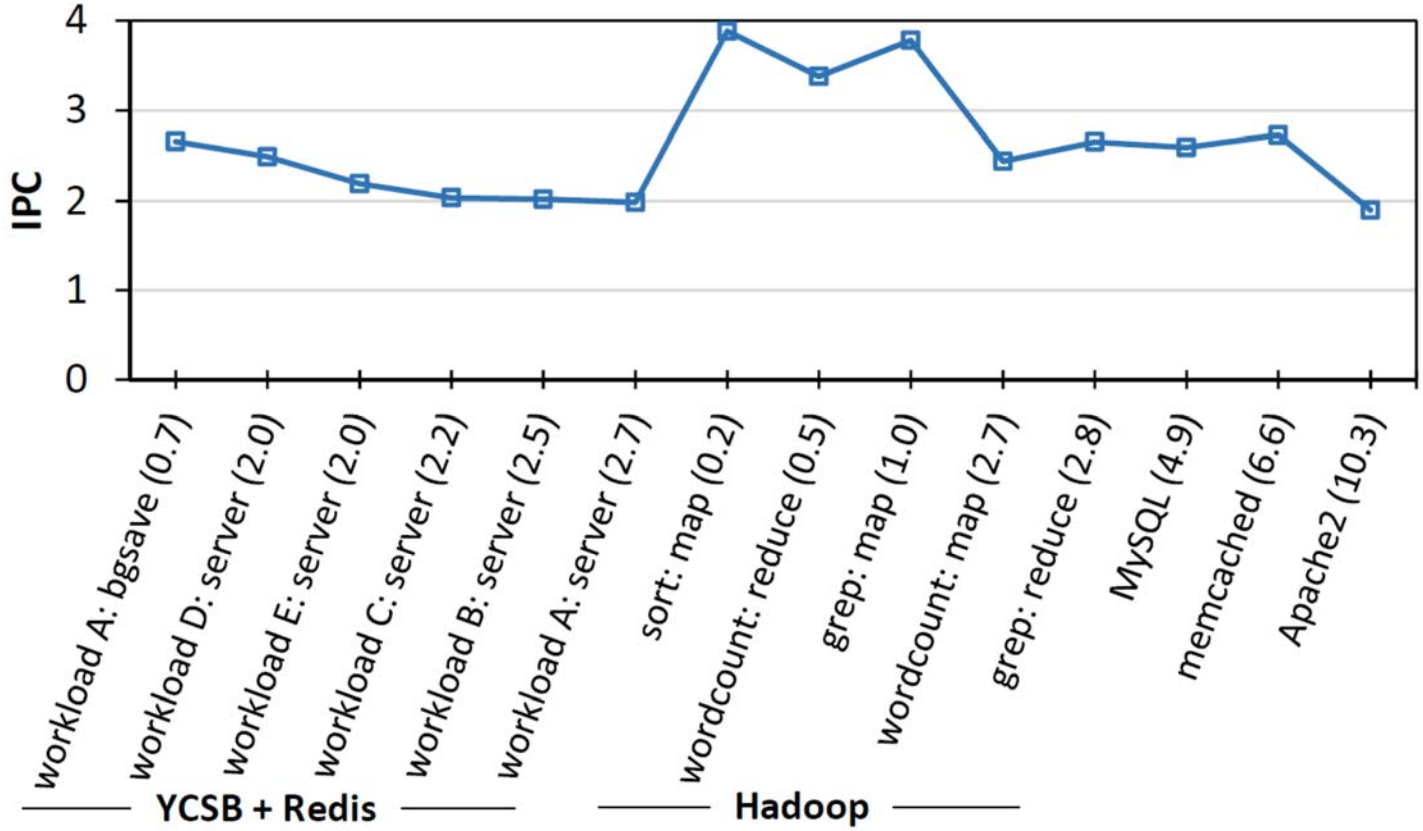




Desktop/Scientific: Multithreaded RBL

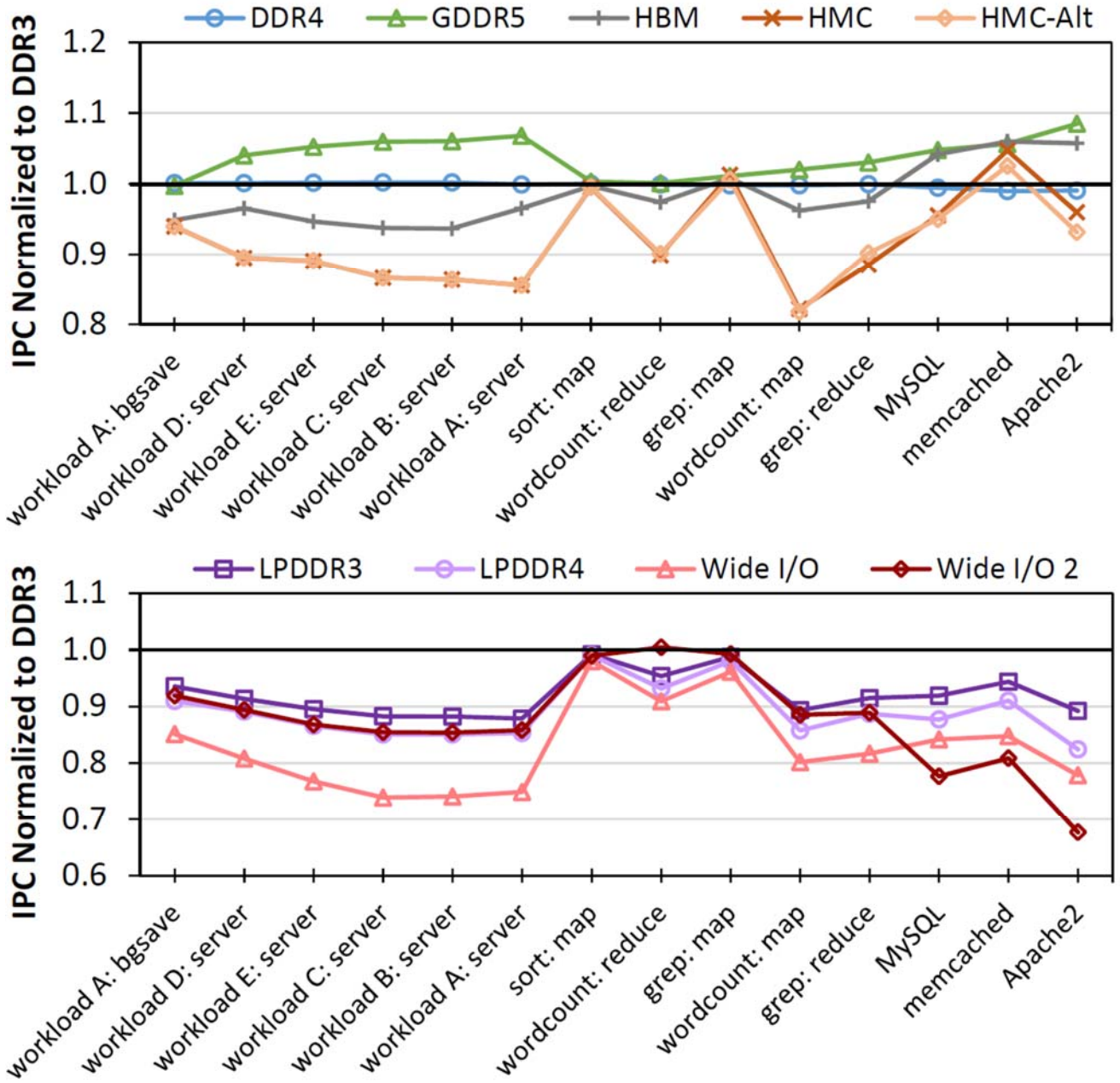


Server/Cloud: Single-Thread IPC

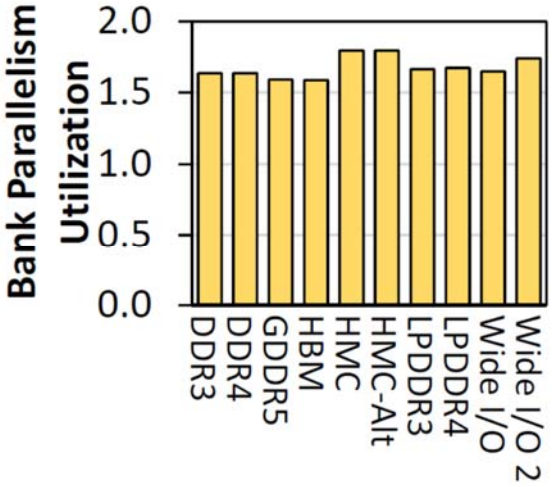


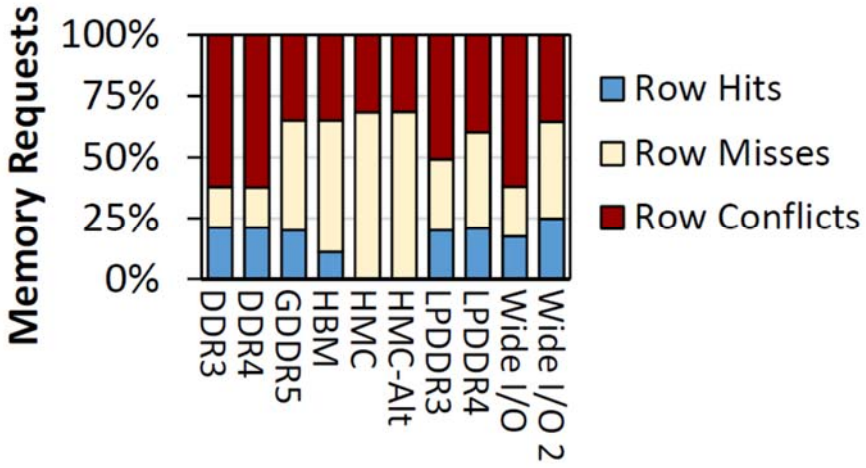
Server/Cloud: Single-Thread Performance

SAFARI



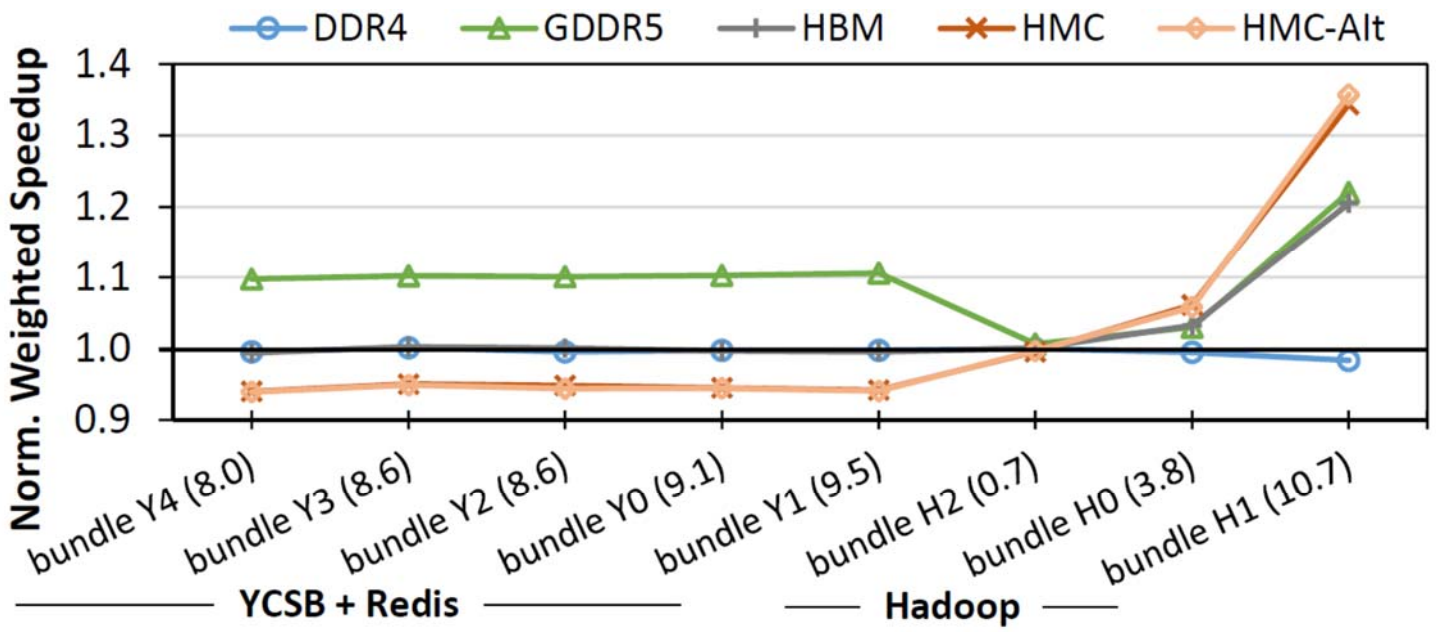
Server/Cloud: *workload A server* BPU/RBL

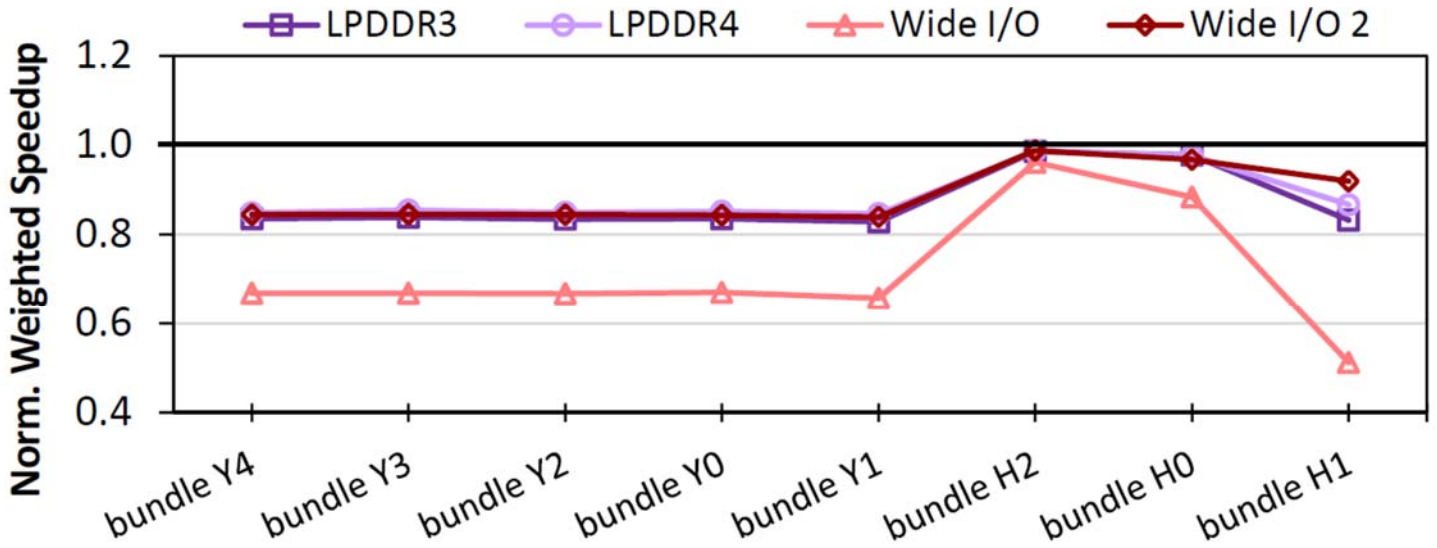




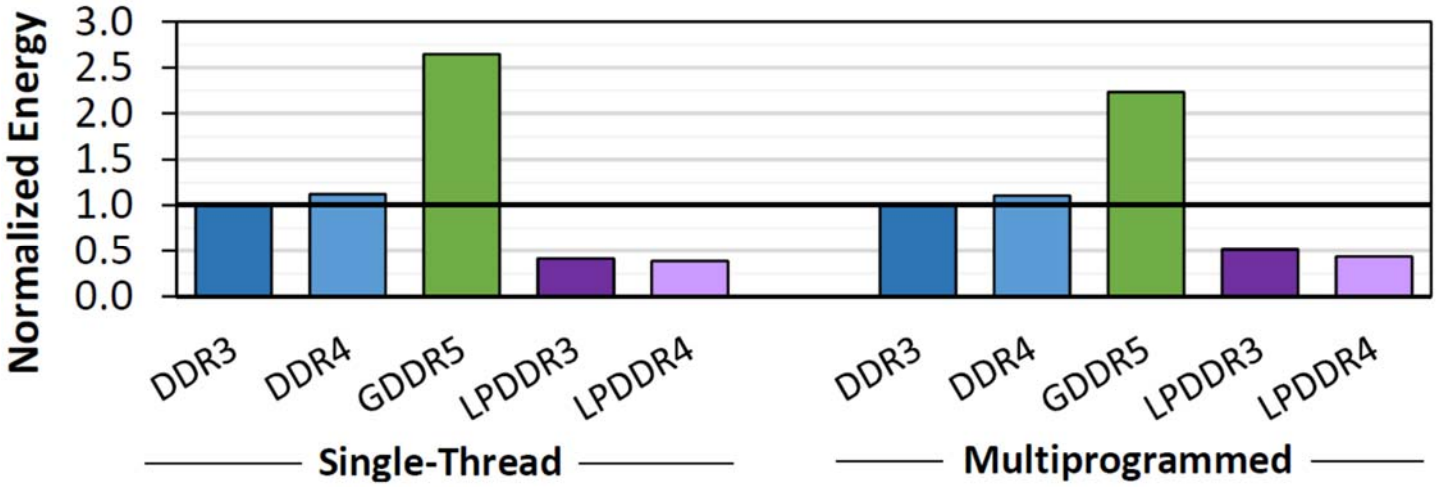
Server/Cloud: Multiprogrammed Performance

SAFARI



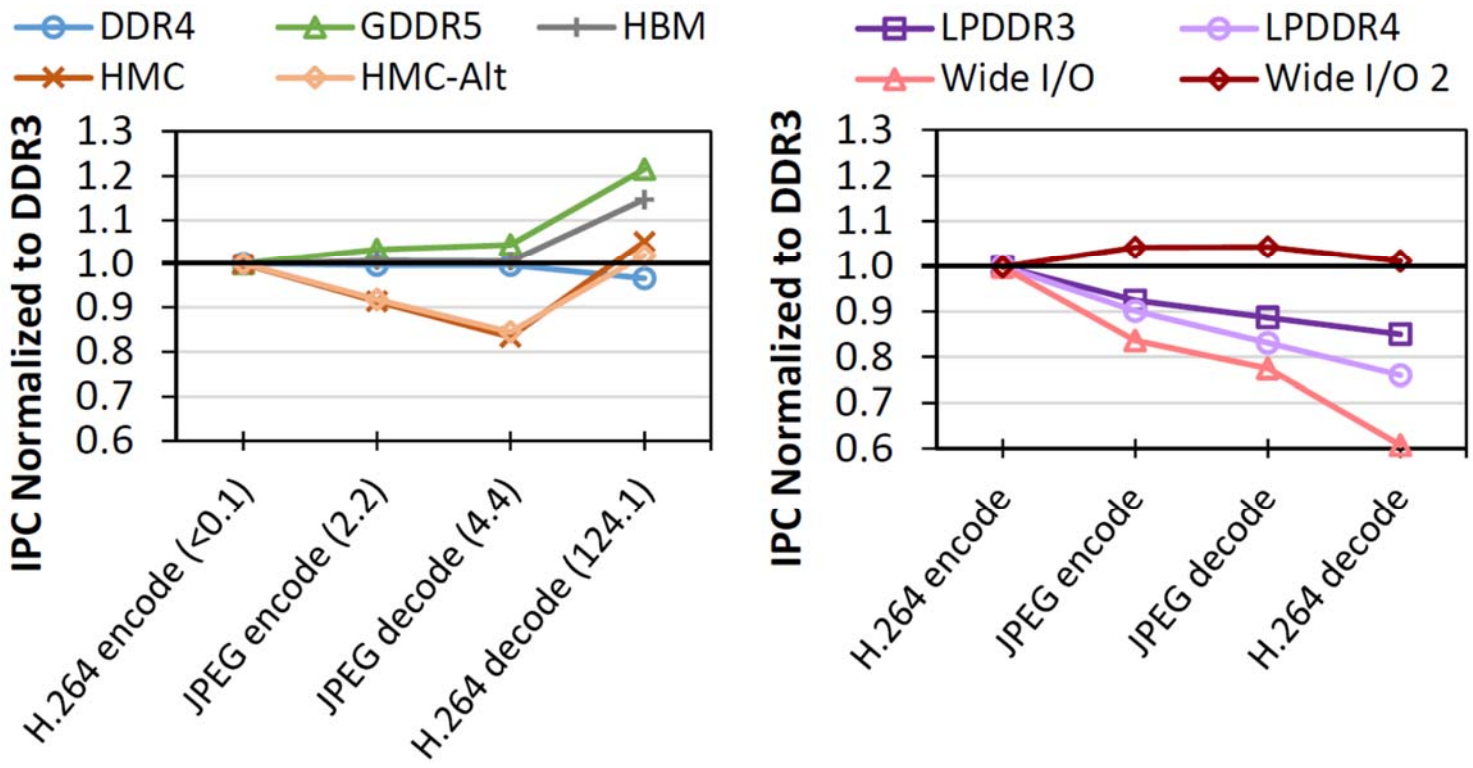


Server/Cloud: Energy



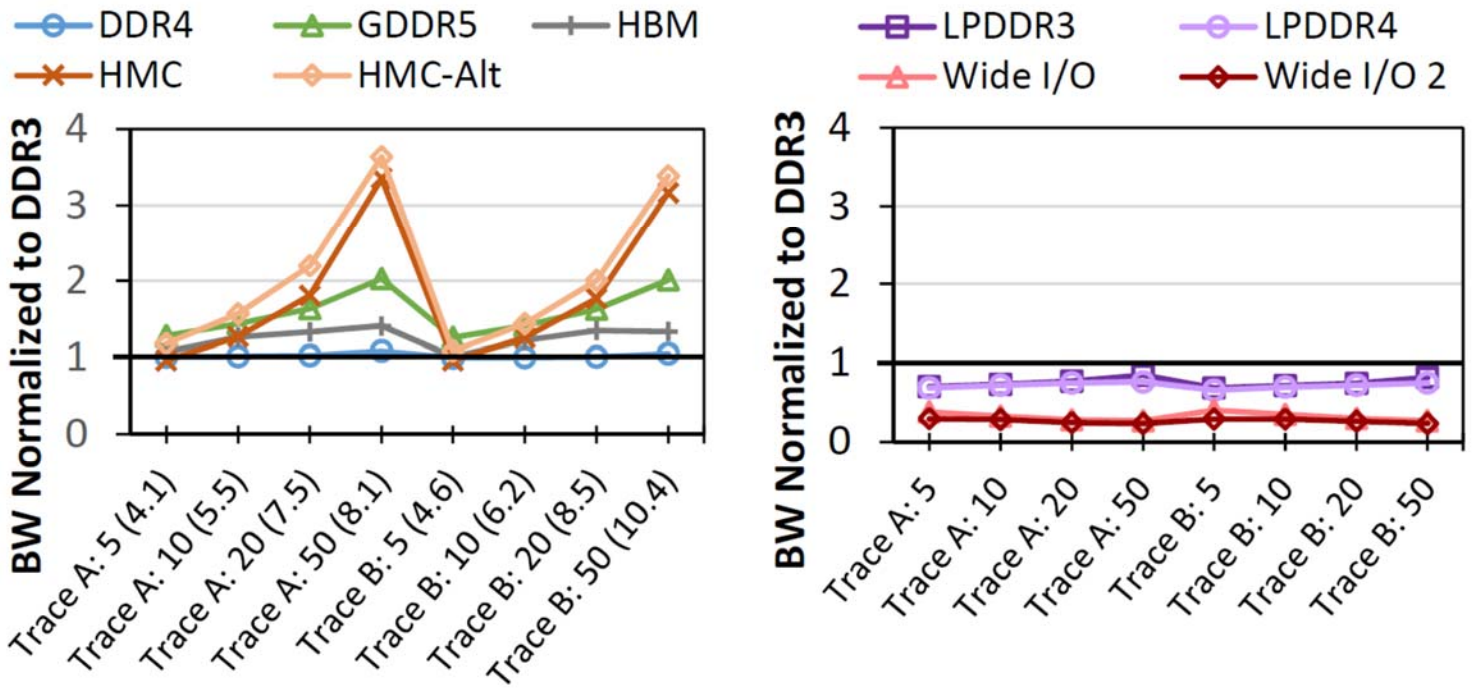
Heterogeneous: Multimedia Accel. Performance

SAFARI



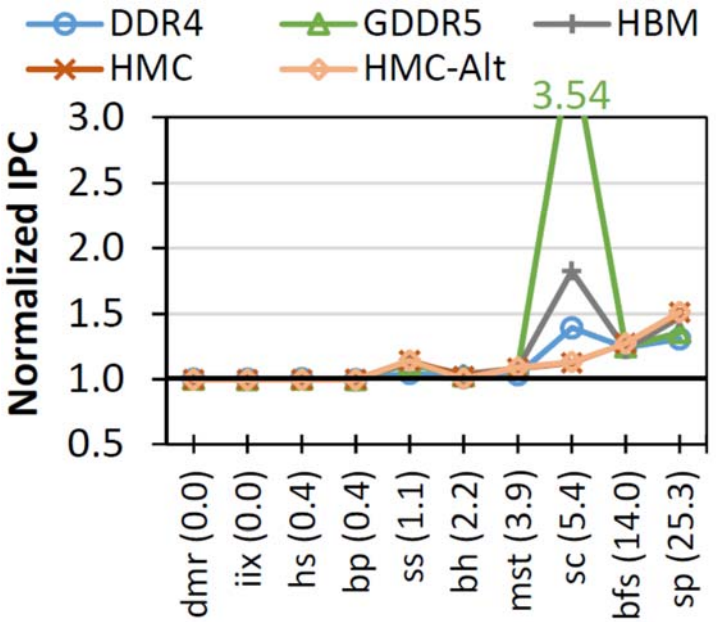
Heterogeneous: Network Accel. Performance

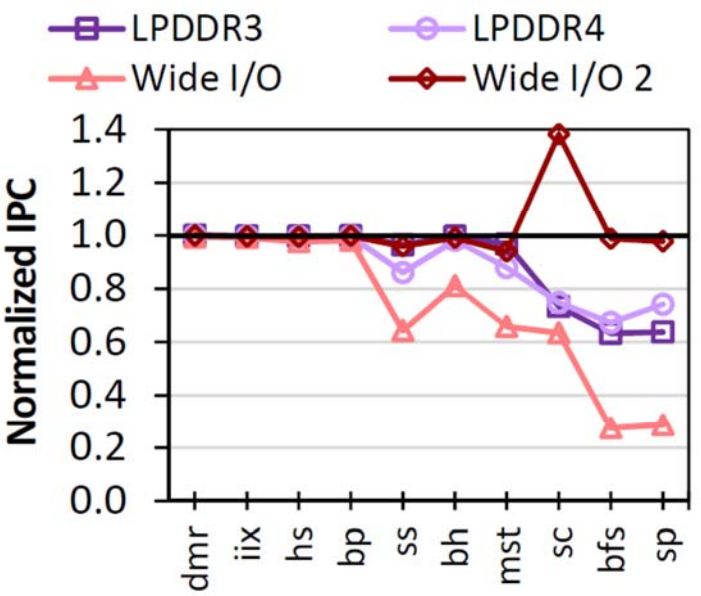
SAFARI



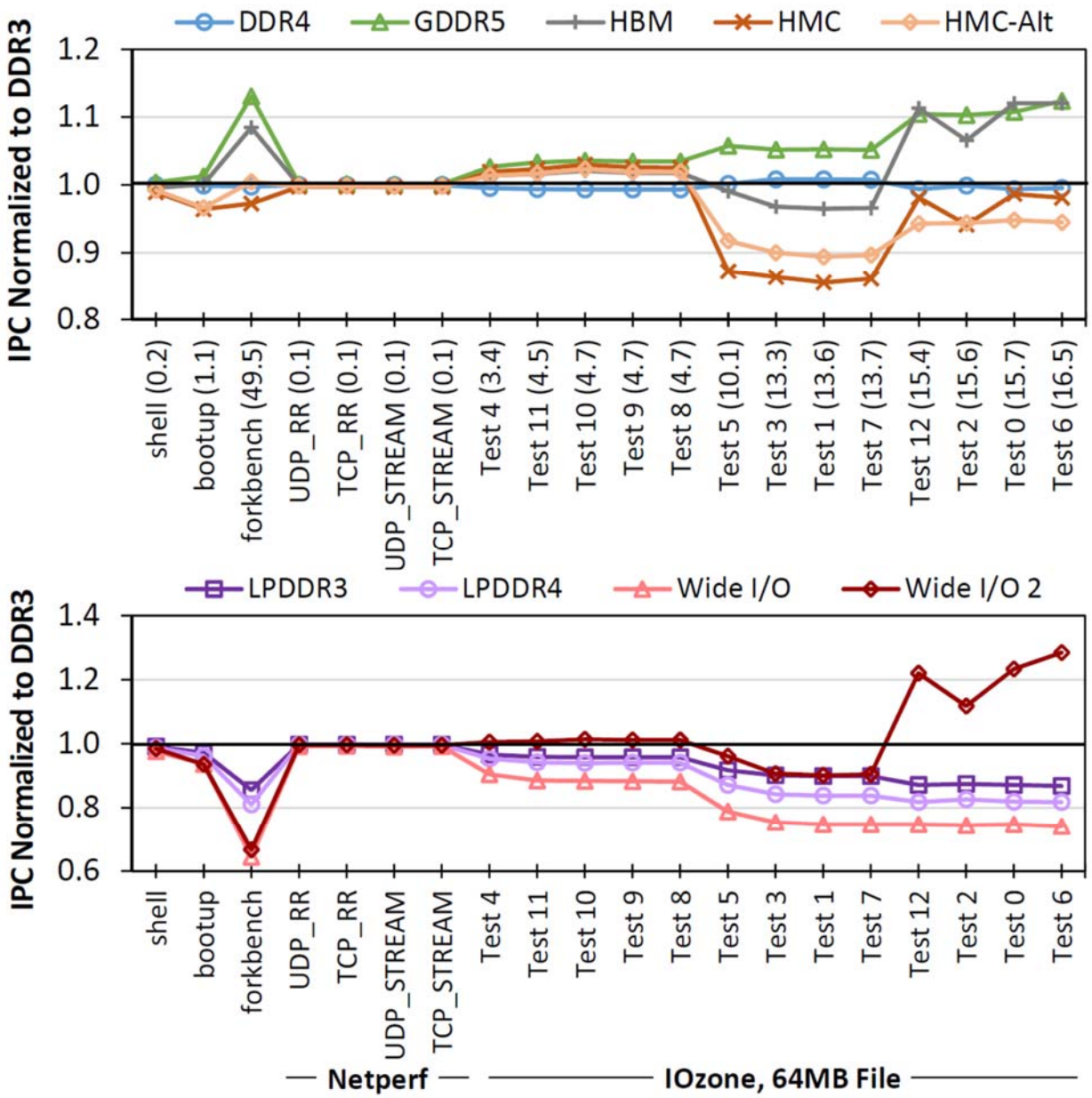
Heterogeneous: GPGPU Performance

SAFARI





OS Routines: Performance



OS Routines: Energy