



# Flash Memory for Buffer Caches

Emulating Memory Hierarchies  
Using a Modified Linux Kernel

John C. Koob, Duncan G. Elliott, Bruce F. Cockburn  
University of Alberta, Edmonton, Canada

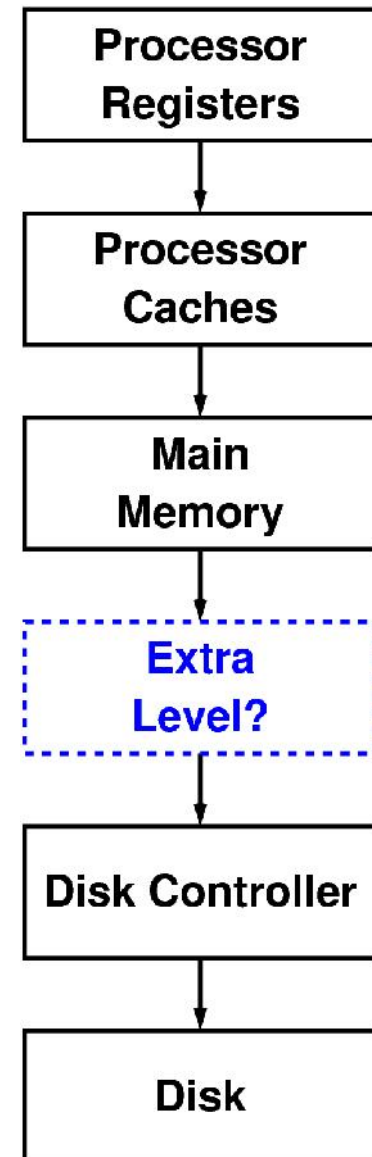


# Outline

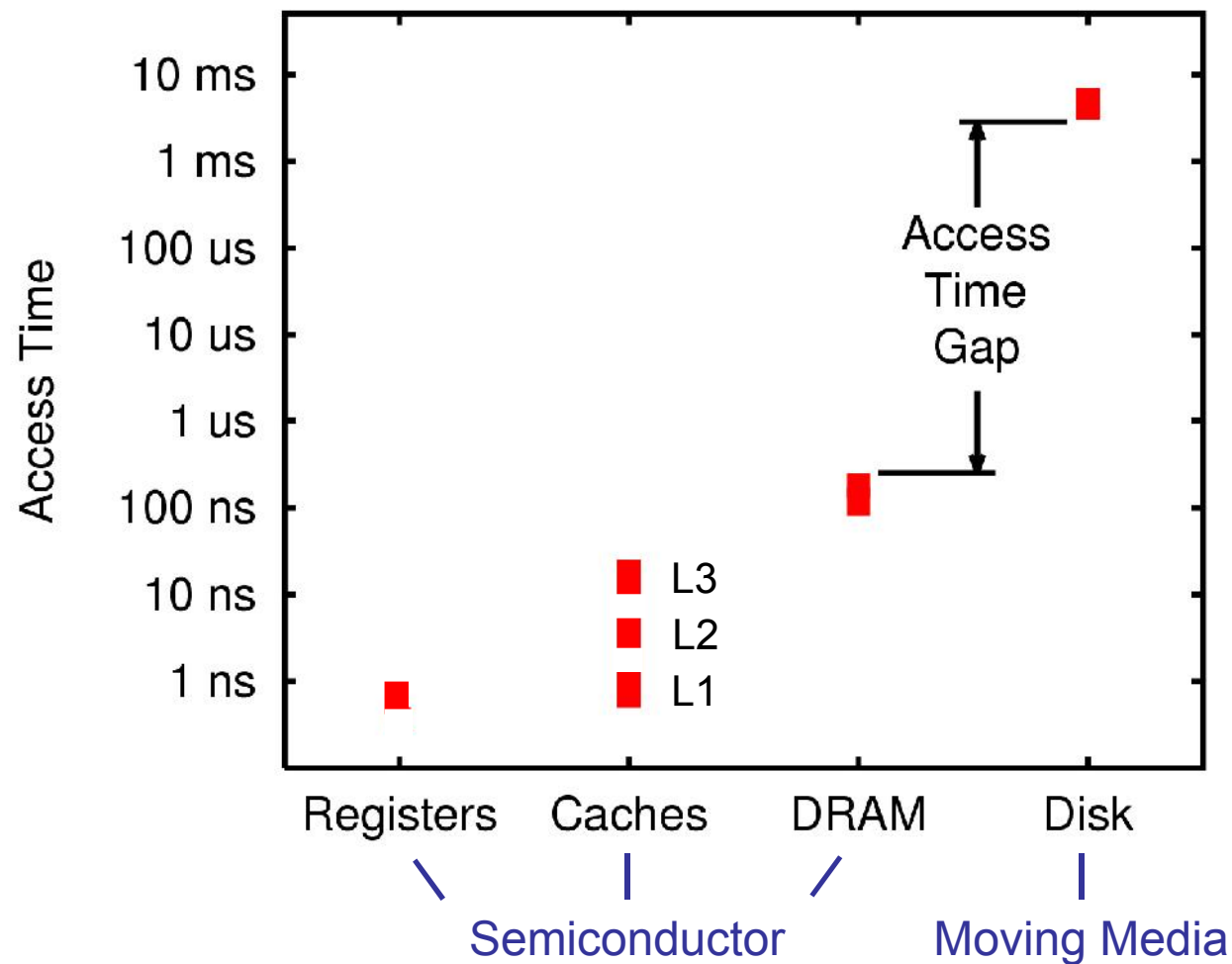
- Introduction
- Motivation
- Background
- Extended Memory Hierarchies
- Experimental Platform
- Emulation Results
- Conclusions

# Memory Hierarchy

- Computer memory is hierarchical
  - Exploit locality of reference in code
- Upper levels: Small, fast, pricey
- Lower levels: Large, slow, cheap
- A well-designed hierarchy:
  - A large, fast, economical memory

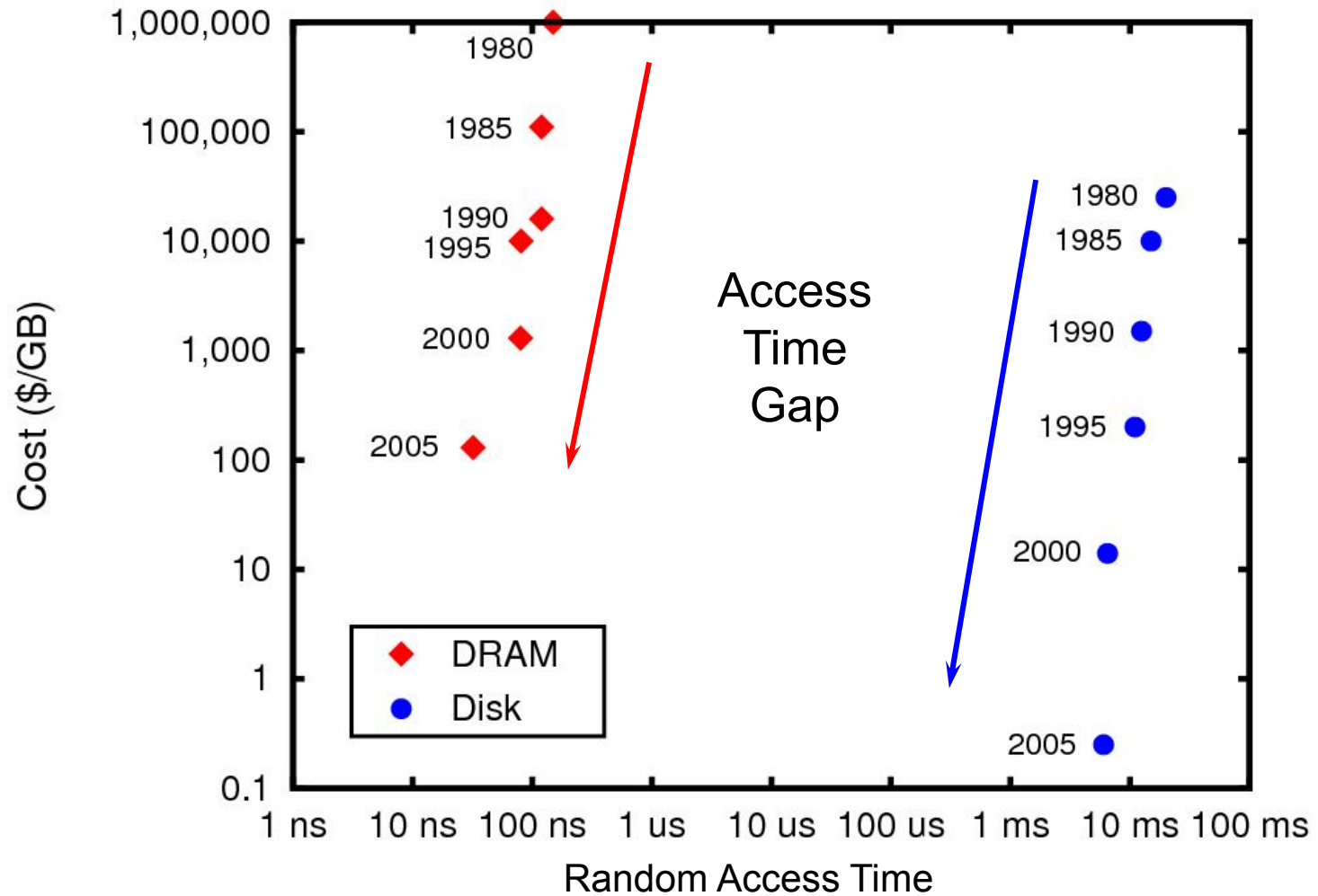


# Motivation (1)



Source: Hennessy & Patterson, *Computer Architecture*, 2003

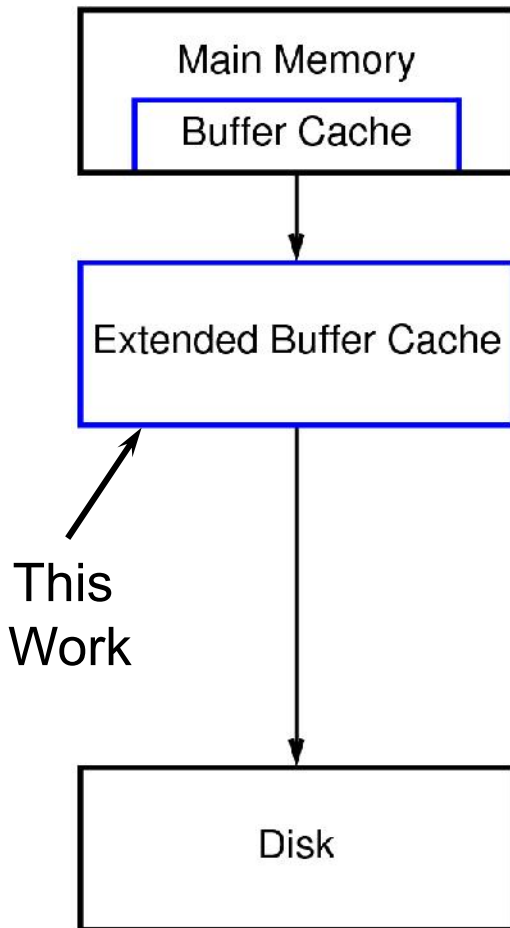
# Motivation (2)



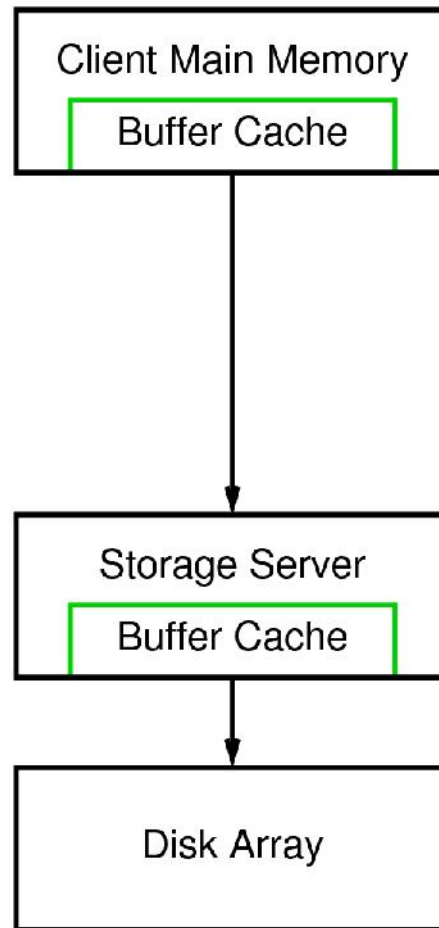
Source: Hennessy & Patterson, *Computer Architecture*, 2006

# Extended Memory Hierarchies

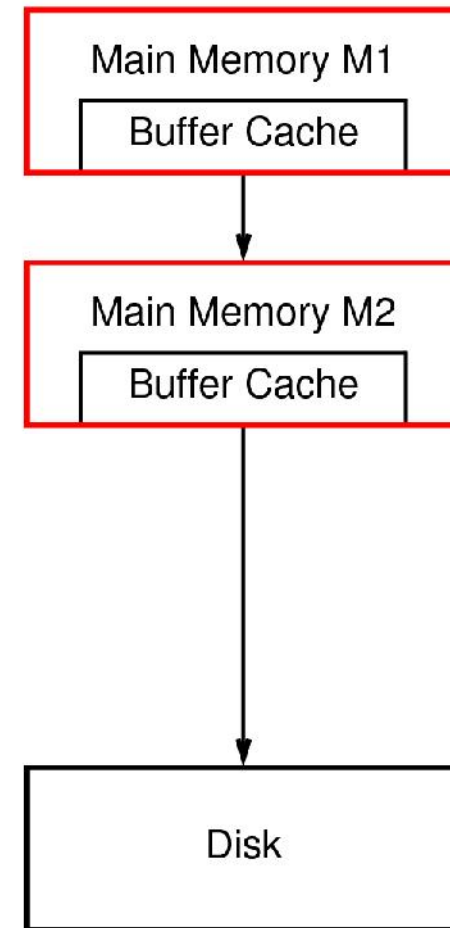
**Extended Buffer Cache**



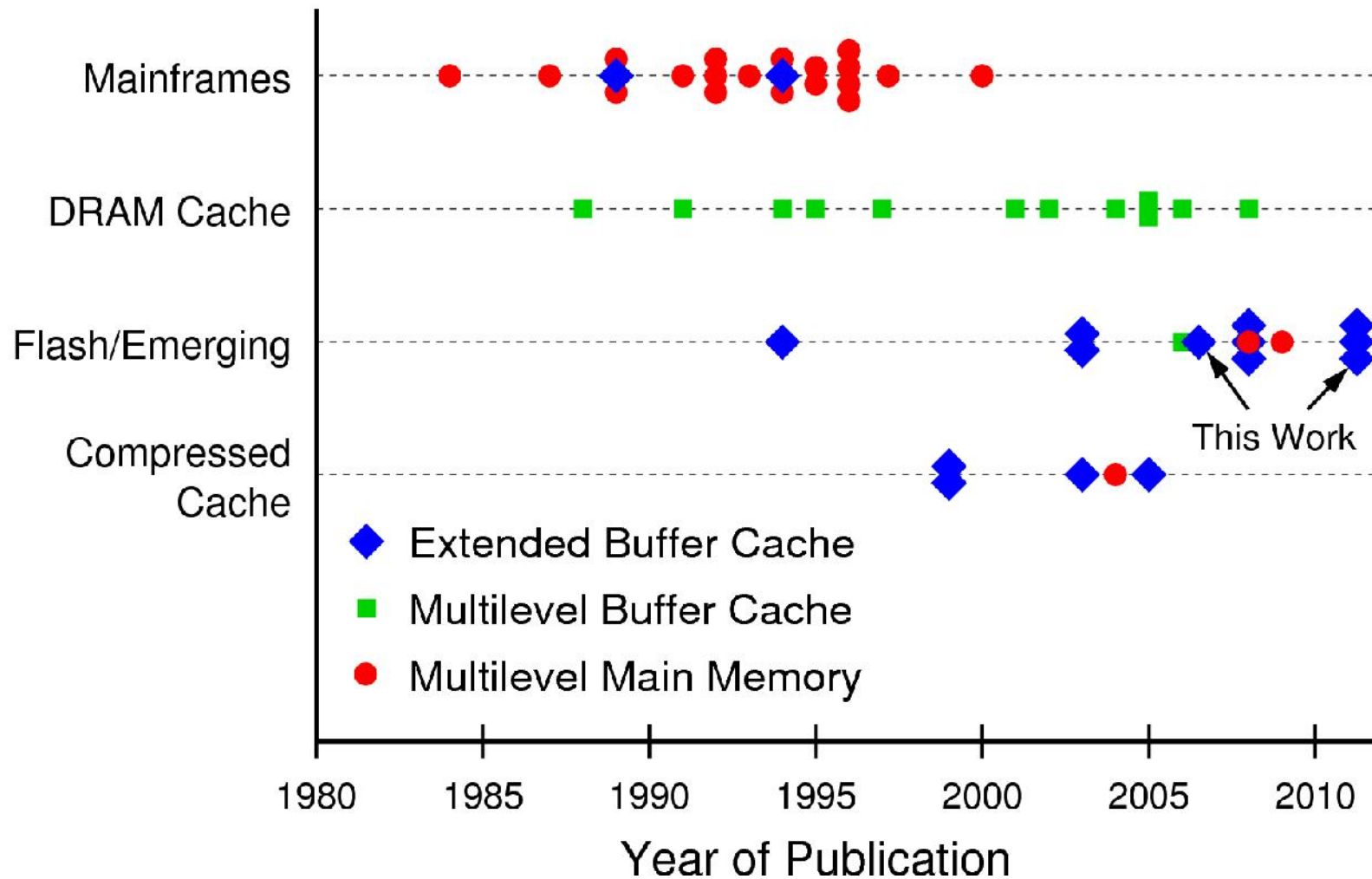
**Multilevel Buffer Cache**



**Multilevel Main Memory**



# Reported Extended Hierarchies



## How to fill the access time gap?

Hierarchy Level	Typical Technology	Access Time (ns)	Typical Size (MB)
Registers	SRAM	0.25	0.0005
Caches	SRAM	1.0	< 16
Main Mem.	DRAM	10-100	< 64,000
Extra Level	?	?	?
Disk	Magnetic	10,000,000	1,000,000





# Memory Technology Comparison

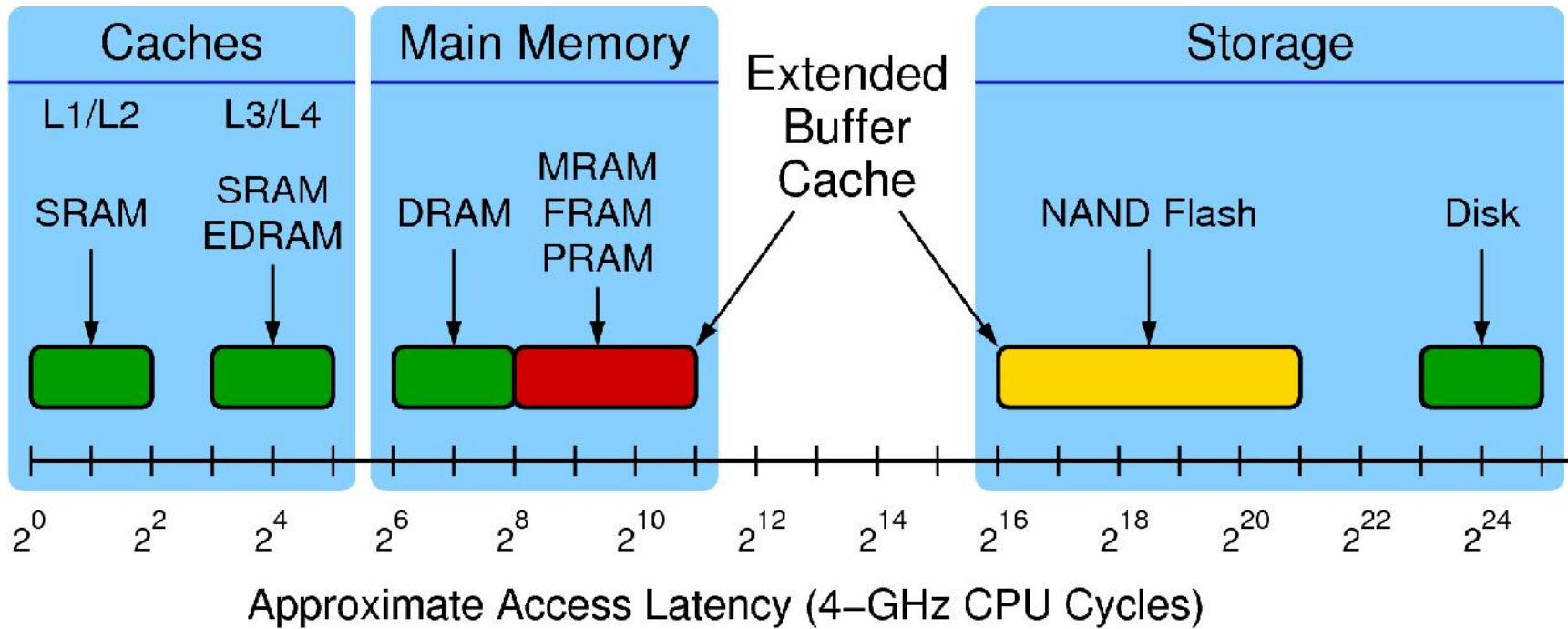
Memory Technology	Cell Area (F <sup>2</sup> )	Cell Endur.	Byte Addr	Non-Volatile
DRAM	6-10	10 <sup>15</sup>	Yes	No
MLDRAM	6-10*	10 <sup>15</sup>	Yes	No
MRAM	6-20	10 <sup>15</sup>	Yes	Yes
FRAM	15-34	10 <sup>15</sup>	Yes	Yes
PRAM	6-12*	10 <sup>7</sup>	Yes	Yes
NOR Flash	10*	10 <sup>5</sup>	Reads	Yes
NAND Flash	5*	10 <sup>5</sup>	No	Yes

\* Products with MLC increases effective density

*Sources: Tabrizi, Non-volatile STT-RAM, Flash Memory Summit, 2009.*

*Jung, FRASH: Storage Class Memory, Trans. Storage, 2010.*

# Targeted Latency Range



- Conventional computer memory and storage
- Non-volatile technology in new role
- Emerging technology (limited production)



# Cost of Backward Compatibility

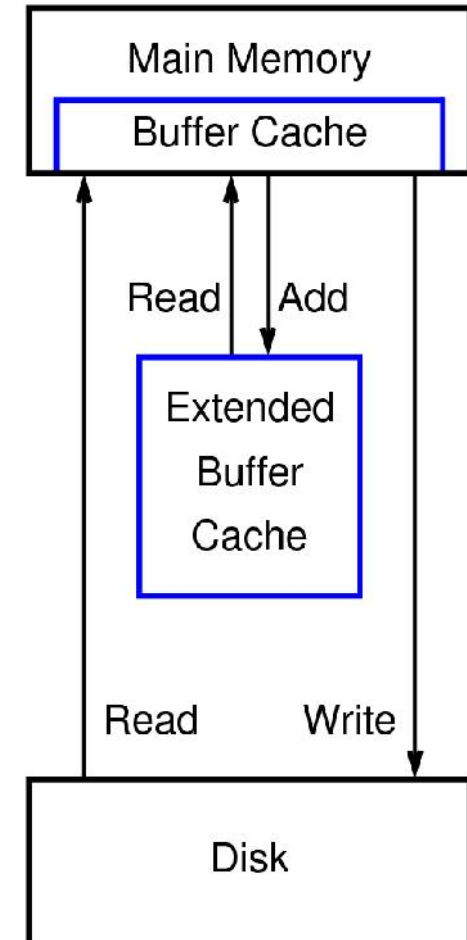
- Compatibility with disk is convenient
  - Hybrid drives and SSDs are backward compatible
  - Usually transparent to OS
  
- Performance loss due to compatible design
  - Loss at interfaces between subsystems\*
  - OS still optimized for slow disk devices
  
- IDEA: Get the most out of flash
  - Move flash from I/O bus to memory bus
  - Optimize OS for existence of flash

\*Source: Jacob, *Memory Systems: Cache, DRAM, Disk*, 2008

# Implemented Hierarchy

- Minimize impact on OS
  - Linux is carefully optimized
  - Extend existing data structures
  - Do not degrade performance
- Extended Buffer Cache
  - Organized into 4-KB pages
  - Filled with page cache evictions
  - Avoid disk I/O on page cache miss
  - Similar to L2ARC from Solaris 10

## Low-Impact Hierarchy





# Experimental Platform

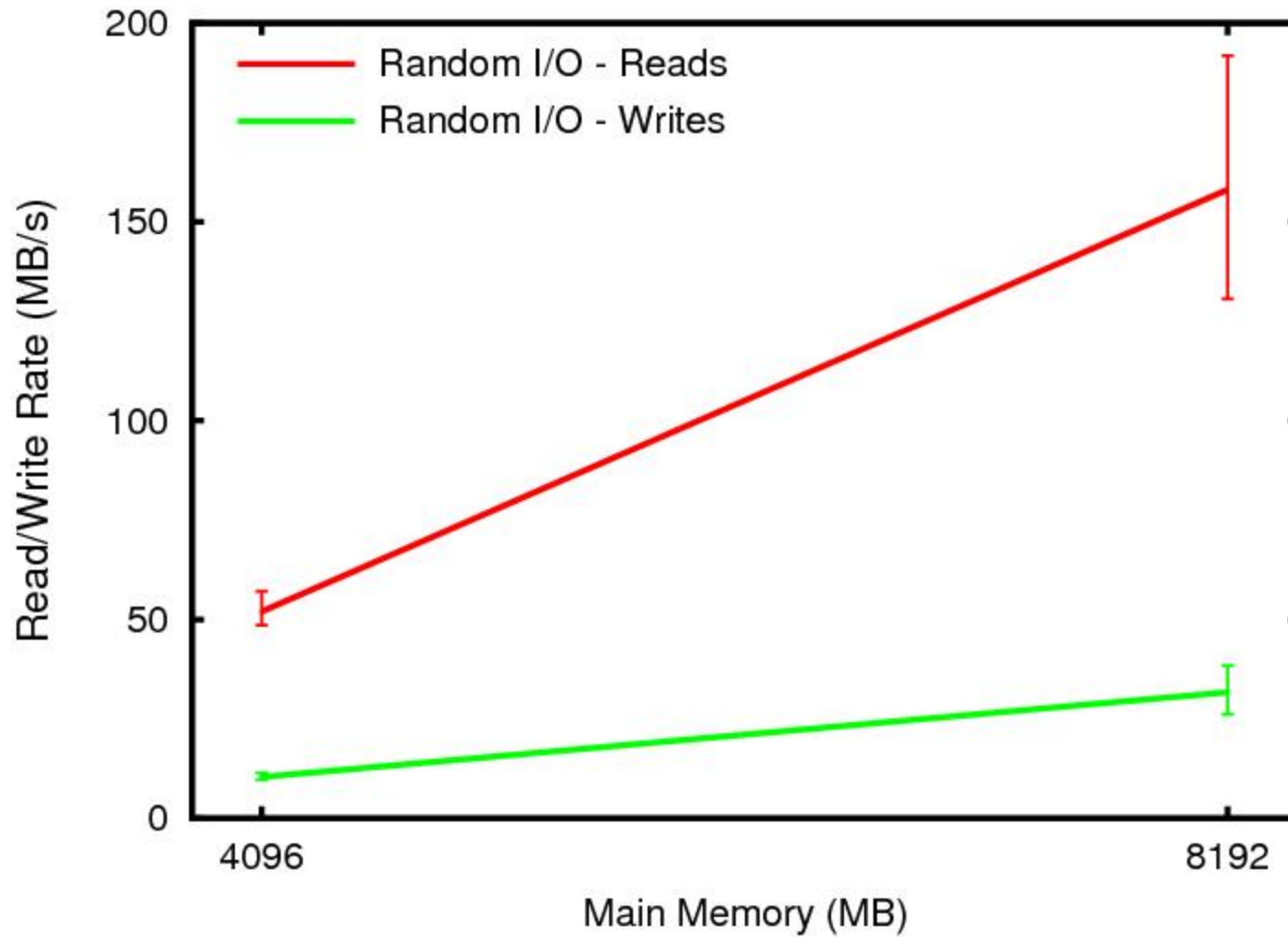
- Flash Memory Emulation Platform
  - Modify Linux 2.6.32 operating system kernel
  - Reserve a portion of DRAM to *emulate* flash
  - Use tunable slowdown factors to model flash access
  - Accurate measurements possible in instrumented kernel
  
- Test System Specifications
  - System                      Sun Fire X2200 M2 server
  - Processors                 Two Dual-Core Opterons 2.6 GHz
  - Page Cache                Shared among cores
  - Memory                     32 GB DDR2 SDRAM
  - Hard Disk                  500 GB SATA



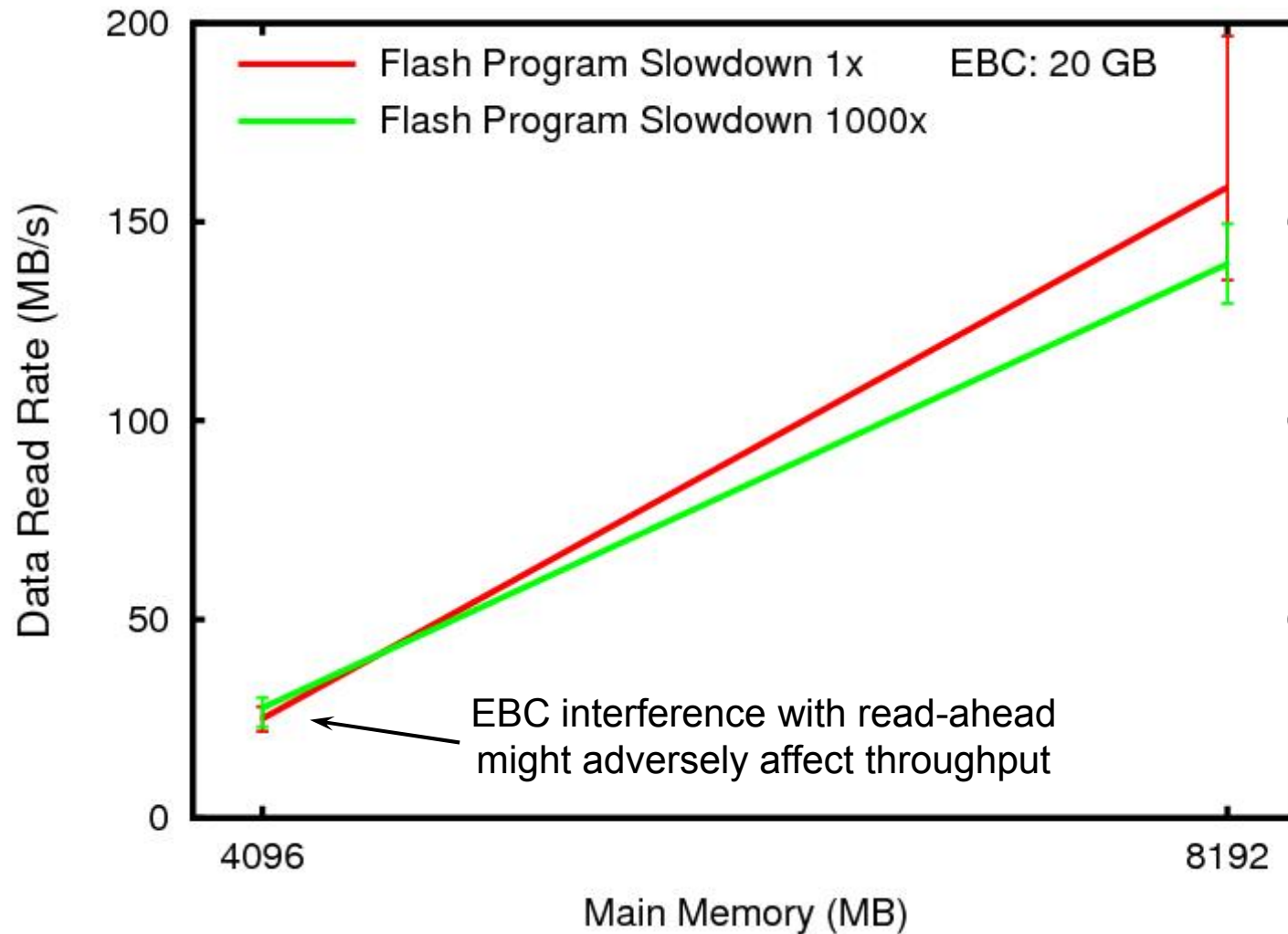
# Synthetic Benchmark

- Postmark file system benchmark
  - Performs random file I/O on a set of files
  - Targeted working set of approximately 6 GB
  
- Configuration of *Postmark 1.5*
  - Number of files created: 10,000
  - Number of transactions: 50,000
  - File sizes: 400,000B - 800,000B
  - Favor create over delete
  - Favor read over append

# Unmodified Kernel - Postmark

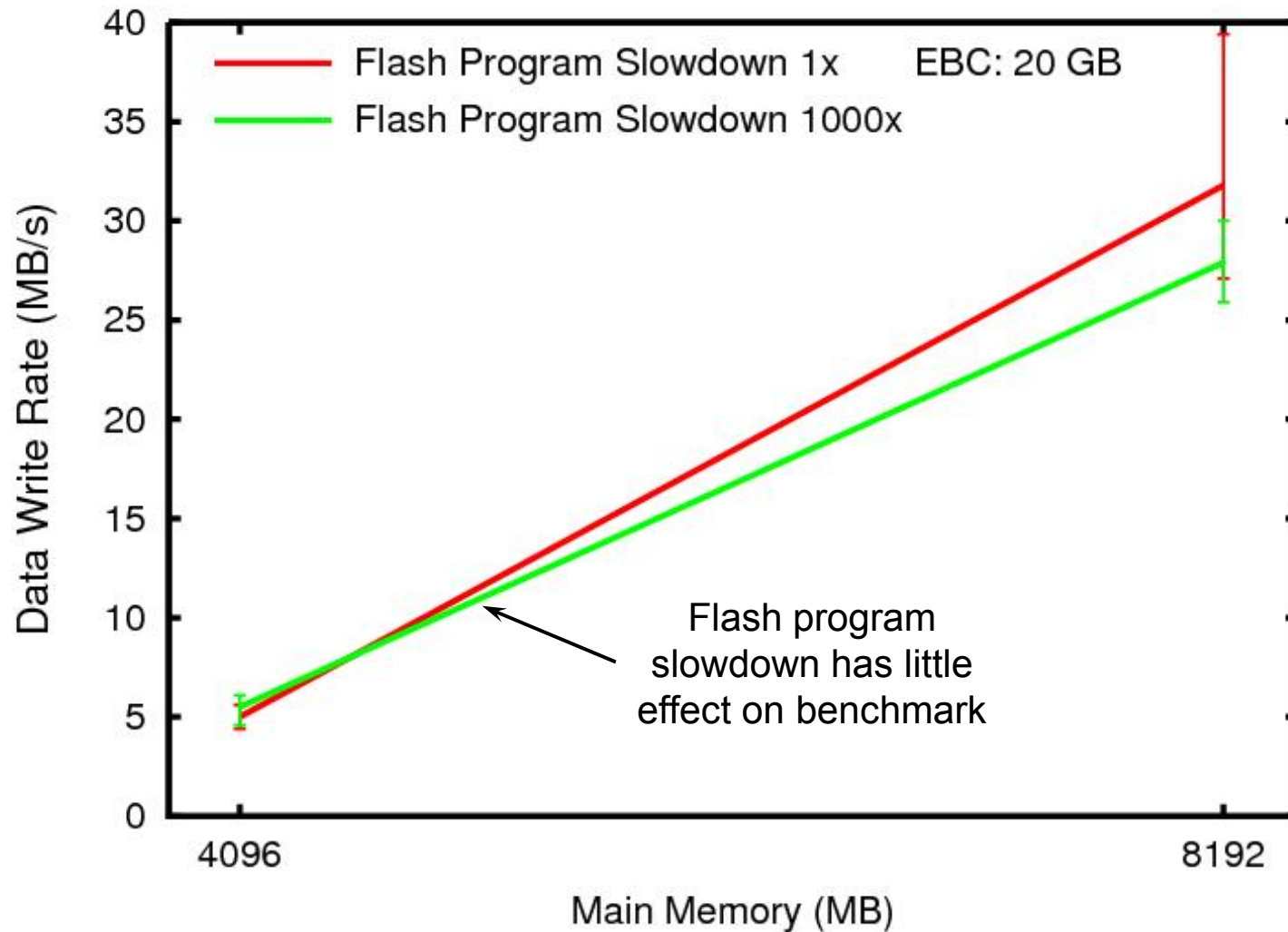


# Emulation - Postmark Read Rate

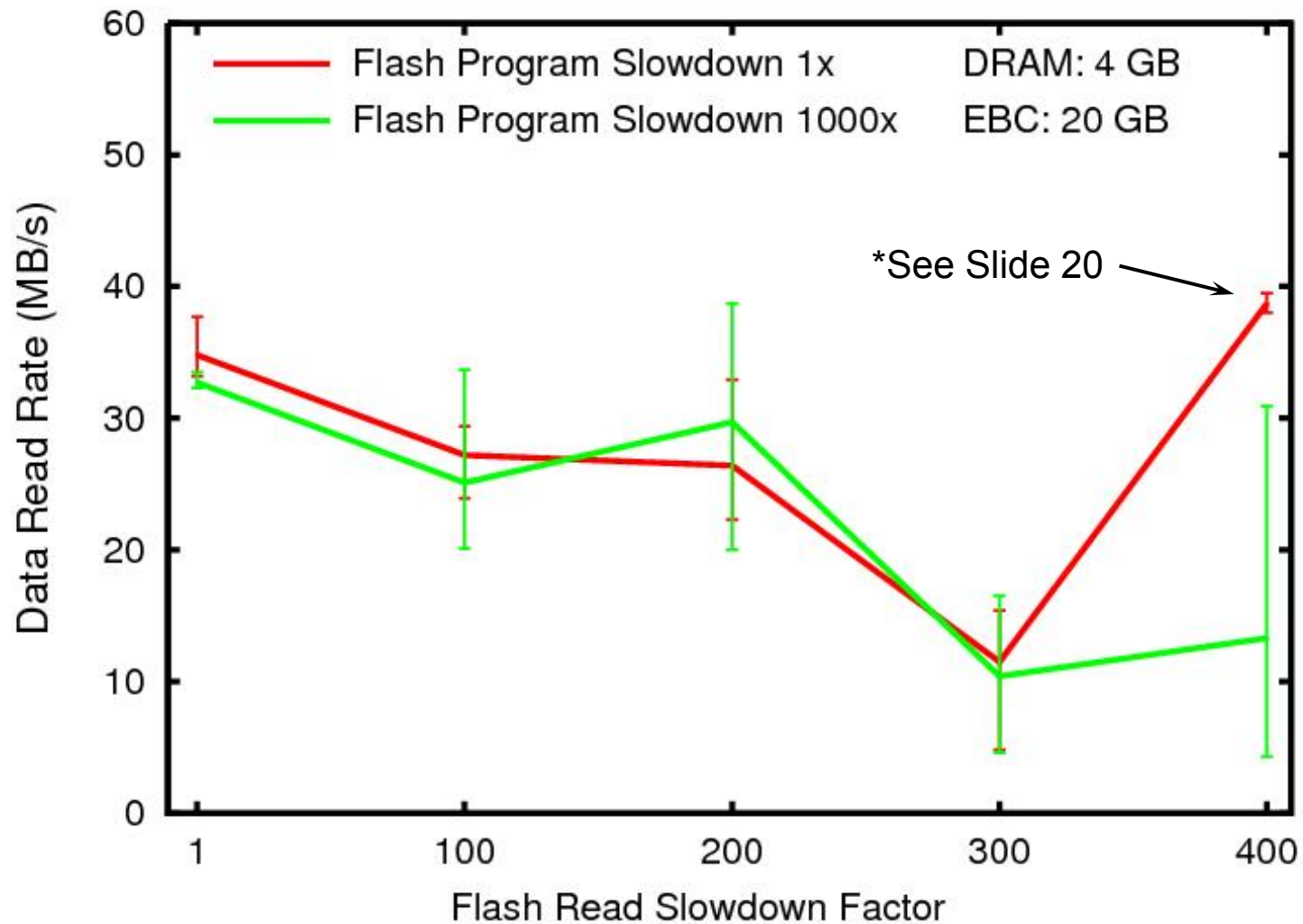




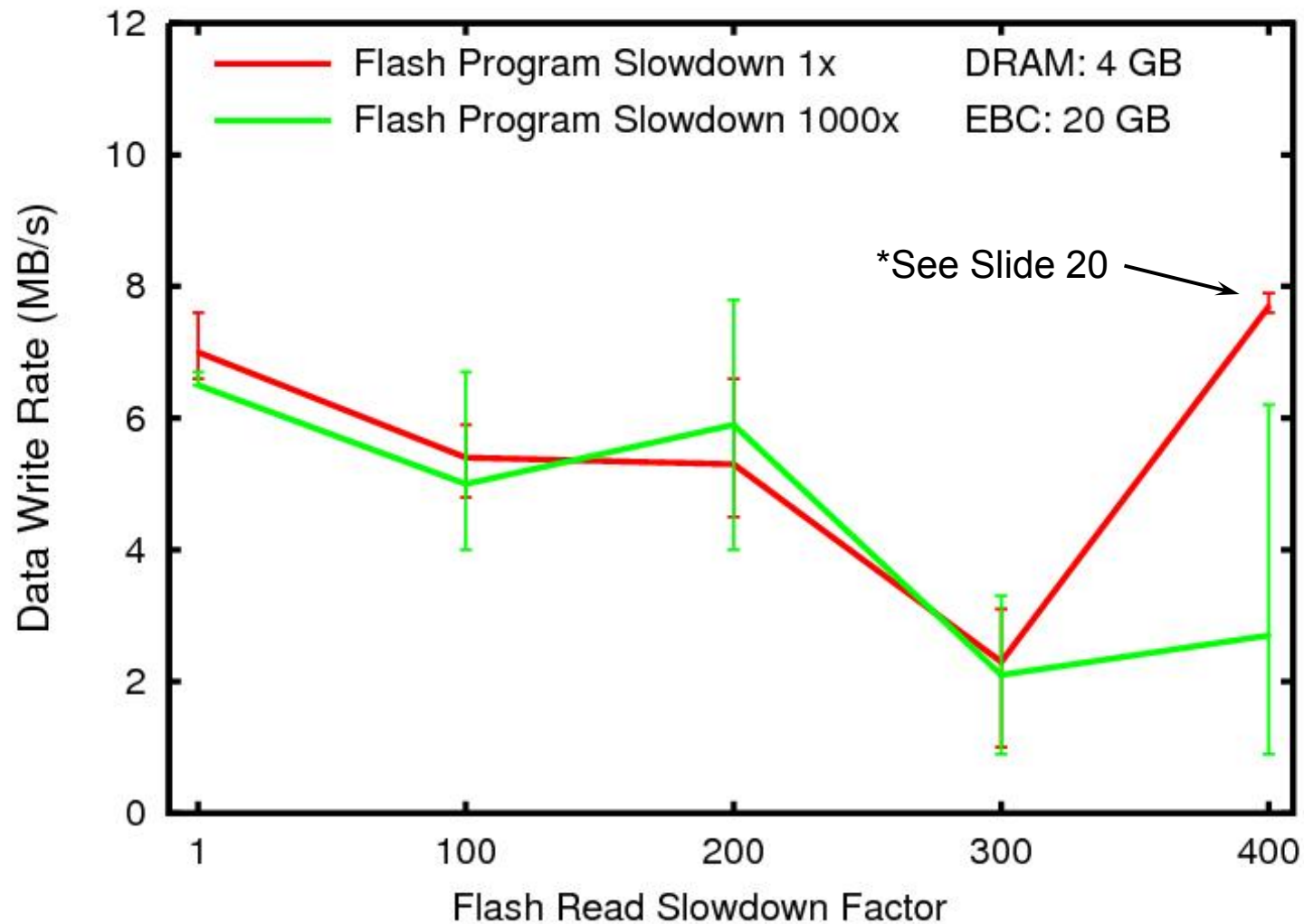
# Emulation - Postmark Write Rate



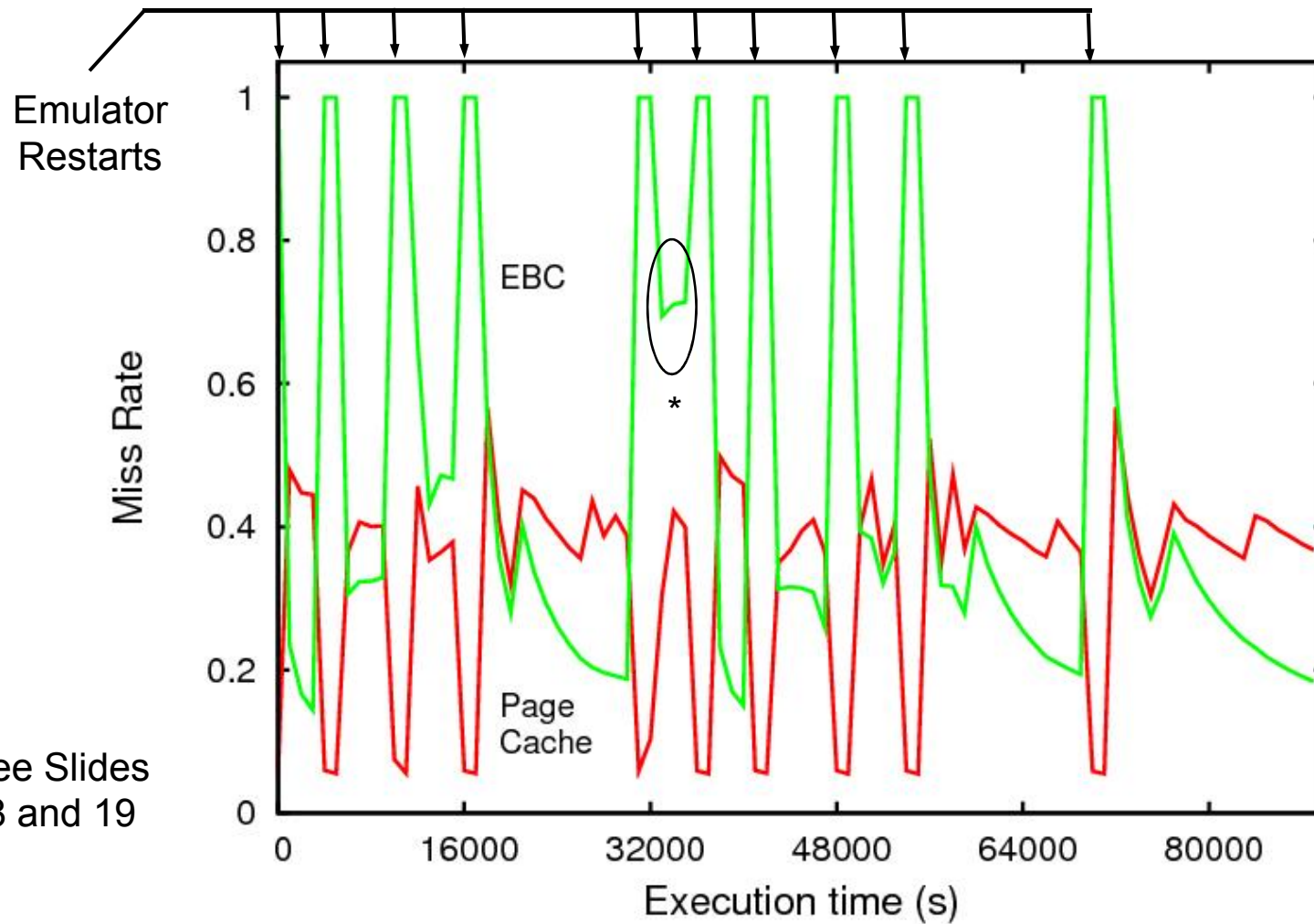
# Emulation - Postmark Read Rate



# Emulation - Postmark Write Rate



# Full Experiment Run - Miss Rates

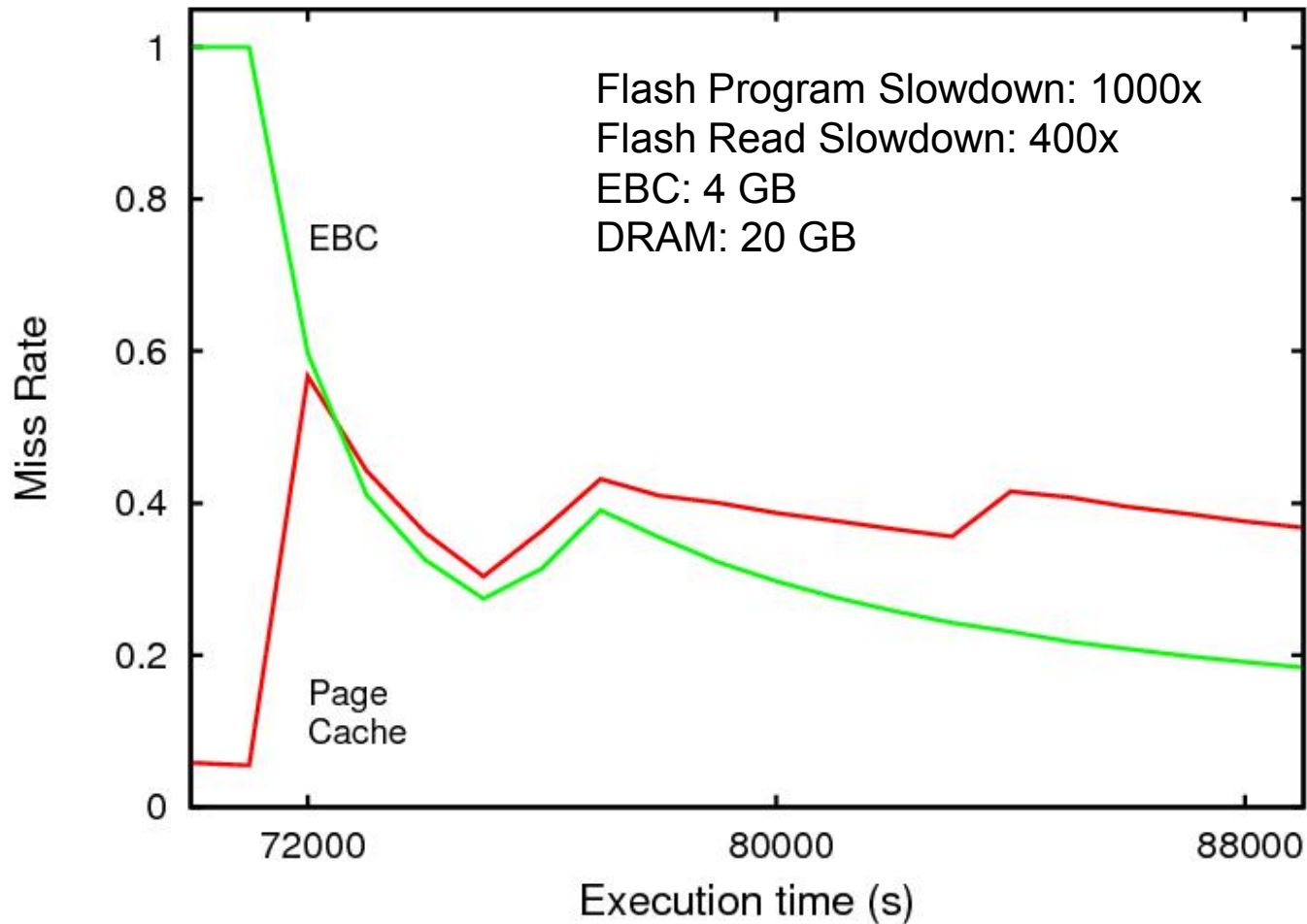


\*See Slides  
18 and 19



# Single Experiment - Miss Rates

Single experiment run with a series of five Postmark processes



## Conclusions

- An emulator to evaluate extended hierarchies
  - Avoid unnecessary performance losses
    - *Optimize subsystem interfaces*
    - *Identify performance impacts (e.g. read-ahead)*
  - Measure performance gains
    - *OS aware of extended hierarchy*
  
- New market opportunities
  - Integrate flash with OS memory management
  - Compatible with other emerging technologies

Updates to Results: <http://www.ece.ualberta.ca/~jkoob/research/research.html>



## Selected References

- Koob, et al., “An Empirical Evaluation of Semiconductor File Memory as a Disk Cache”, WMPI 2006.
- Hennessy & Patterson, *Computer Architecture*, 2003.
- Hennessy & Patterson, *Computer Architecture*, 2006.
- Jacob, *Memory Systems: Cache, DRAM, Disk*, 2008.
- Jung, “FRASH: Storage Class Memory”, *Trans. Storage*, 2010.
- Tabrizi, “Non-volatile STT-RAM”, Flash Memory Summit, 2009.