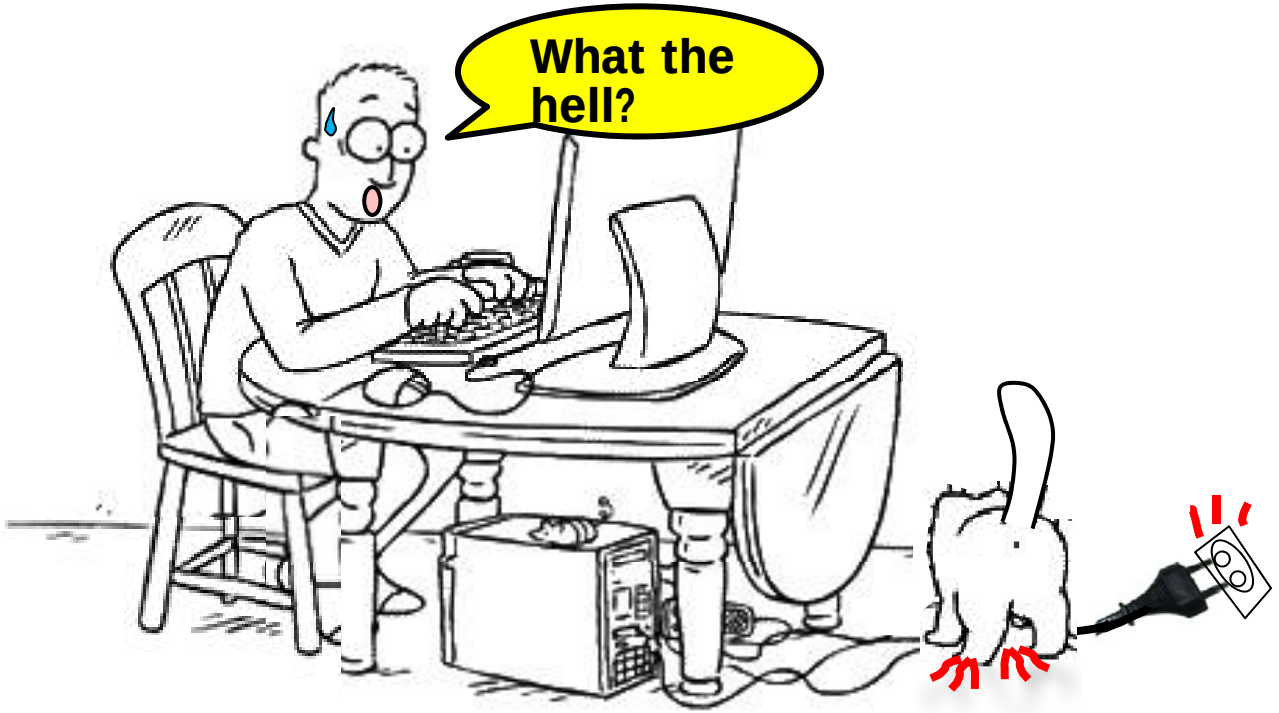
Unioning of the Buffer Cache and Journaling Layers with Non-volatile Memory

Hyokyung Bahn (Ewha University)

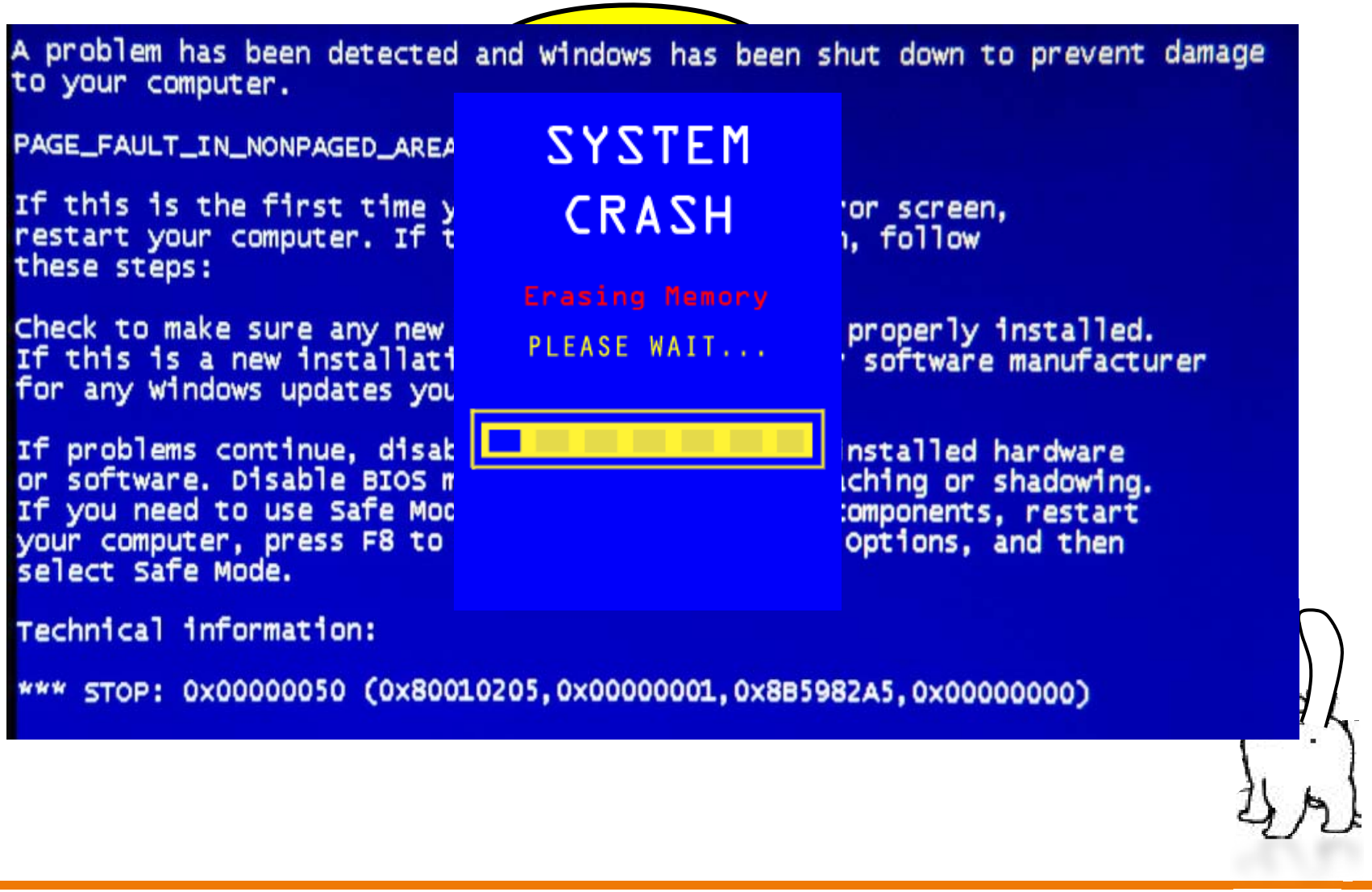
Contents

- Reliability issues in storage systems
  - Consistency problem
  - Journaling techniques
- Consistency problem with non-volatile memory
  - Non-volatile memory technology overview
  - Data inconsistency with non-volatile memory
- Unioning of the Buffer cache and Journaling layers
  - In-place commit and system recovery of UBJ
  - Cache performance of UBJ
  - Performance evaluation
- Conclusion

A man working hard …

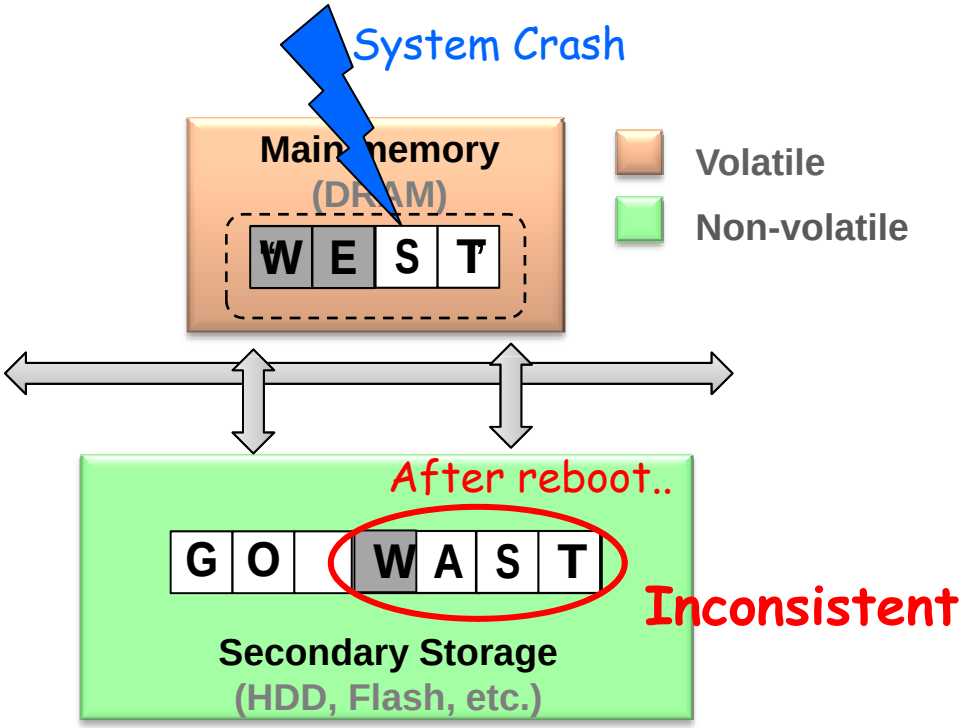


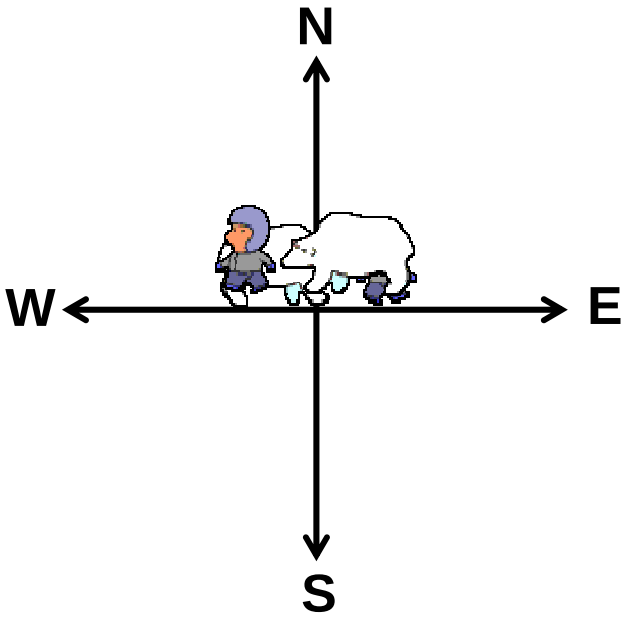
A man working hard …



What happens?

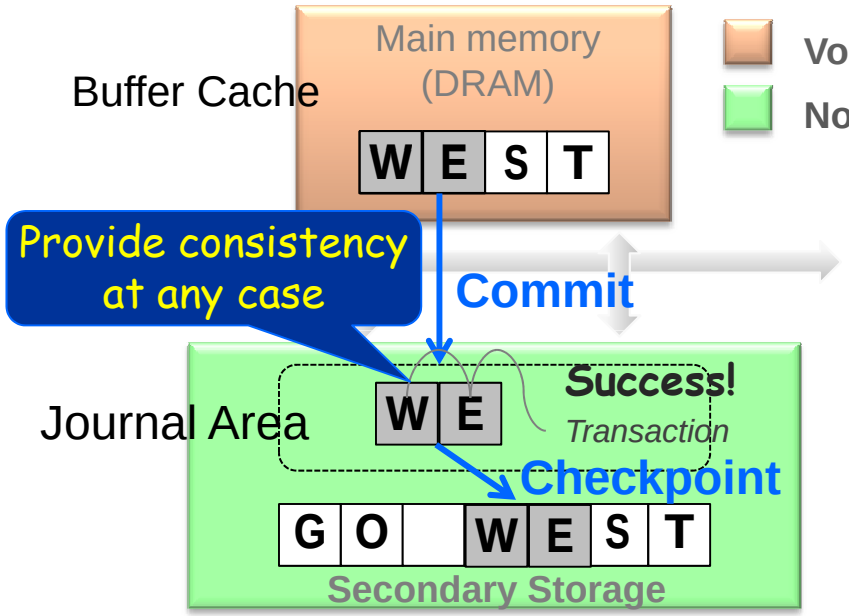
- Sudden power failure may incur file system inconsistency in hierarchical memory systems

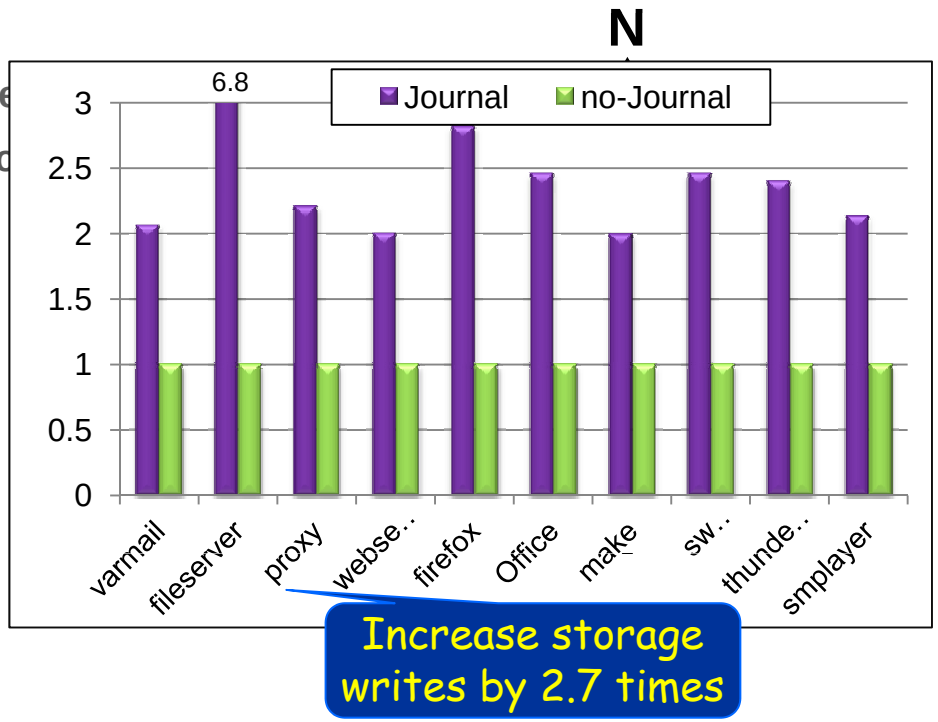




How to solve this problem?

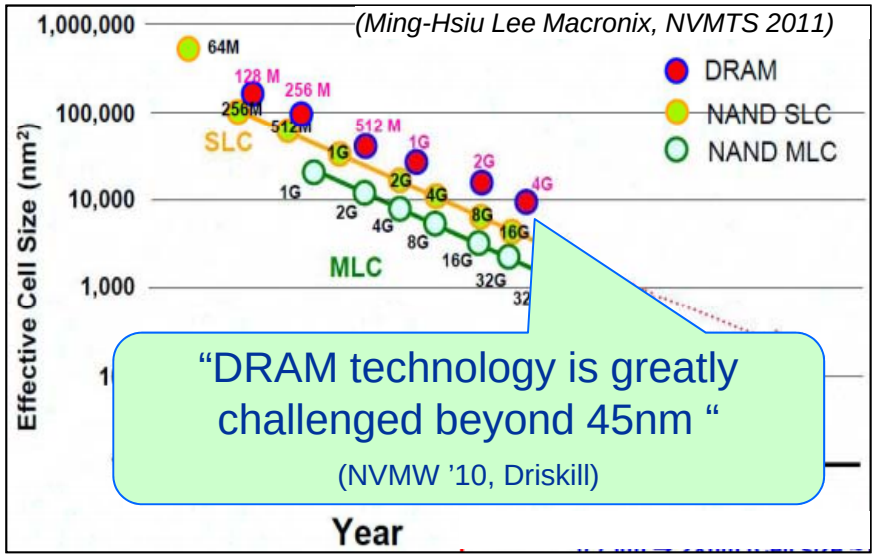
- Prevent data inconsistency by journaling techniques
  - ext3, ext4, ReiserFS, XFS, btrFS
- Frequent commit increases write traffic to storage significantly, leading to the performance degradation





Non-volatile memory promises to replace DRAM in main memory

1. Scaling Limit of DRAM



"DRAM technology is greatly challenged beyond 45nm "

(NVMW '10, Driskill)

2. Power consumption

As much as 40% of the total system energy is consumed by the main memory subsystem in a mid-range IBM eServer machine. (Querish, ISCA 2009)

Replacing DRAM with STT-RAM in data centers can reduce power by up to 75%

(NVMW '10, Driskill)

3. Demand for fast memory access

As critical applications are becoming more data-centric, memory performance is fast becoming the key bottleneck

Non-volatile Memory Technology

Source: T. Perez, C. A. F. D Rose, Technical Report, PUCRS, 2010

	SRAM	DRAM	Disk	NAND Flash		PCRAM	RRAM (Memristor)	MRAM (STT-RAM)
Maturity	Product	Product	Product	Product		Advanced development	Early development	Advanced development
Cell Size	>100 F^2	6-8 F^2	(2/3) F^2	4-5 F^2		8-16 F^2	>5 F^2	37 F^2
Read Latency	<10 ns	10-60 ns	8.5 ms	25 µs		48 ns	<10 ns	<10 ns
Write Latency	<10 ns	10-60 ns	9.5 ms	200 µs		40-150 ns	~10 ns	12.5 ns
Energy per bit access	>1 pJ	2 pJ	100-1000 mJ	10 nJ		100 pJ	2 pJ	0.02 pJ
Static Power	Yes	Yes	Yes	No		No	No	No
Endurance	$>10^{15}$	$>10^{15}$	$>10^{15}$	10^4		10^8	10^5	$>10^{15}$
Nonvolatility	No	No	Yes	Yes		Yes	Yes	Yes
	Current Memory Technologies					Emerging NVM Technologies		

Scalability ☐ Low-power ☐ High-performance ☐

Non-volatile Memory Technology

Source: T. Perez, C. A. F. D Rose, Technical Report, PUCRS, 2010

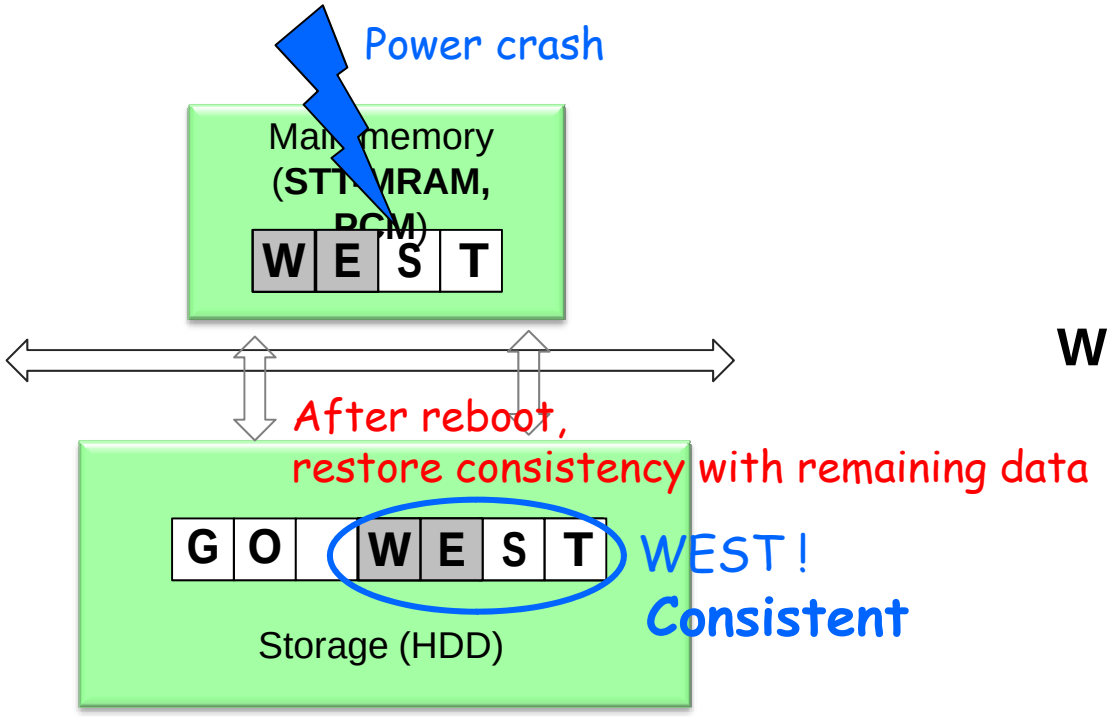
	SRAM	DRAM	Disk	NAND Flash	PCRAM	RRAM (Memristor)	MRAM (STT-RAM)
Maturity	Product	Product	Product	Product	Advanced development	Early development	Advanced development
Cell Size	>100 F^2	6-8 F^2	(2/3) F^2	4-5 F^2	8-16 F^2	>5 F^2	37 F^2
Read Latency	<10 ns	10-60 ns	8.5 ms	25 µs	48 ns	<10 ns	<10 ns
Write Latency	<10 ns	10-60 ns	9.5 ms	200 µs	40-150 ns	~10 ns	12.5 ns
Energy per bit access	>1 pJ	2 pJ	100-1000 mJ	10 nJ	100 pJ	2 pJ	0.02 pJ
Static Power	Yes	Yes	Yes	No	No	No	No
Endurance	$>10^{15}$	$>10^{15}$	$>10^{15}$	10^4	10^8	10^5	$>10^{15}$
Nonvolatility	No	No	Yes	Yes	Yes	Yes	Yes
	Current Memory Technologies				Emerging NVM Technologies		

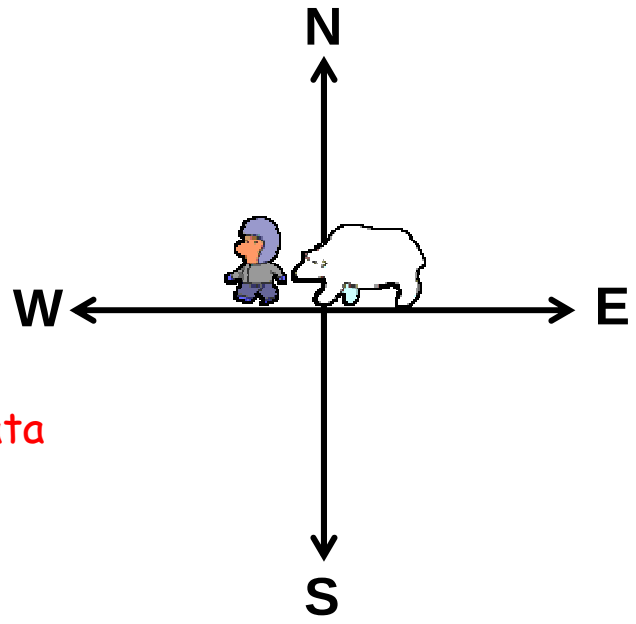
Scalability ☑ Low-power ☑ High-performance ☑

(Optimistic expectations)

What if non-volatile main memory?

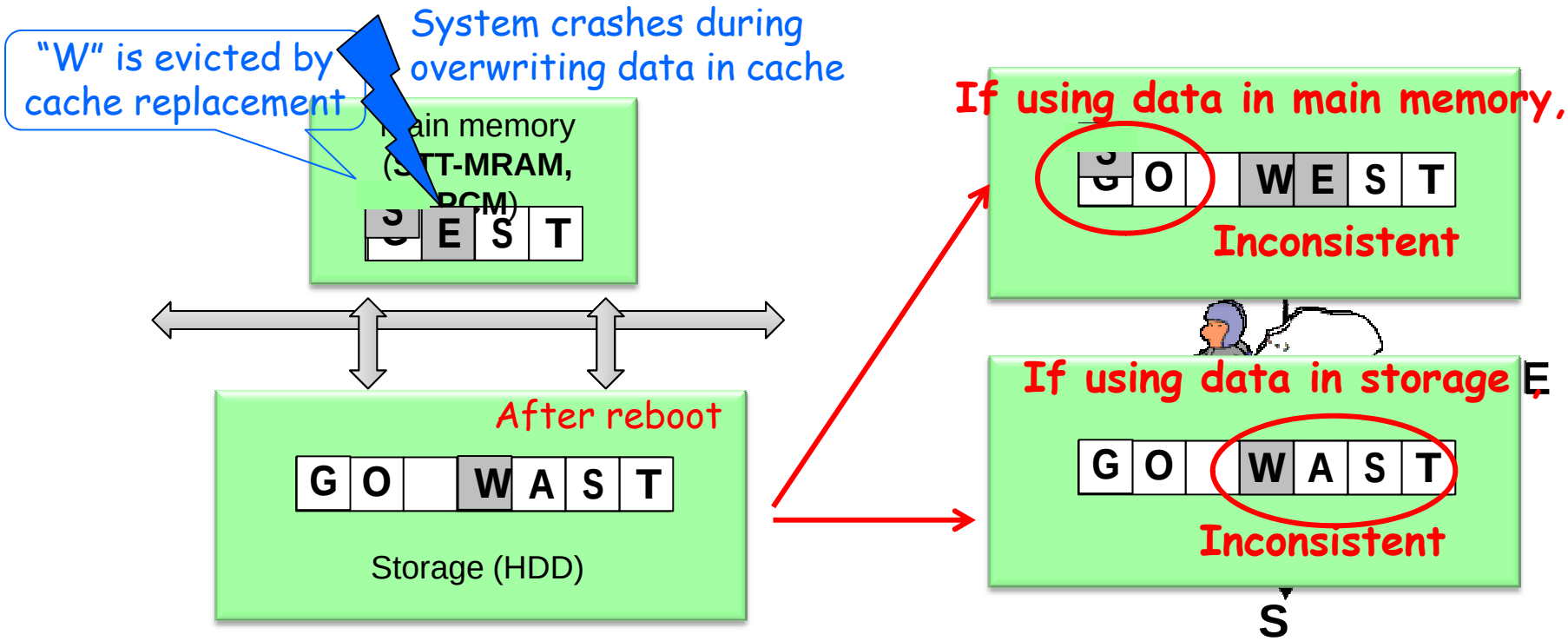
- Non-volatile memory seems to provide data consistency as data survive after crashes





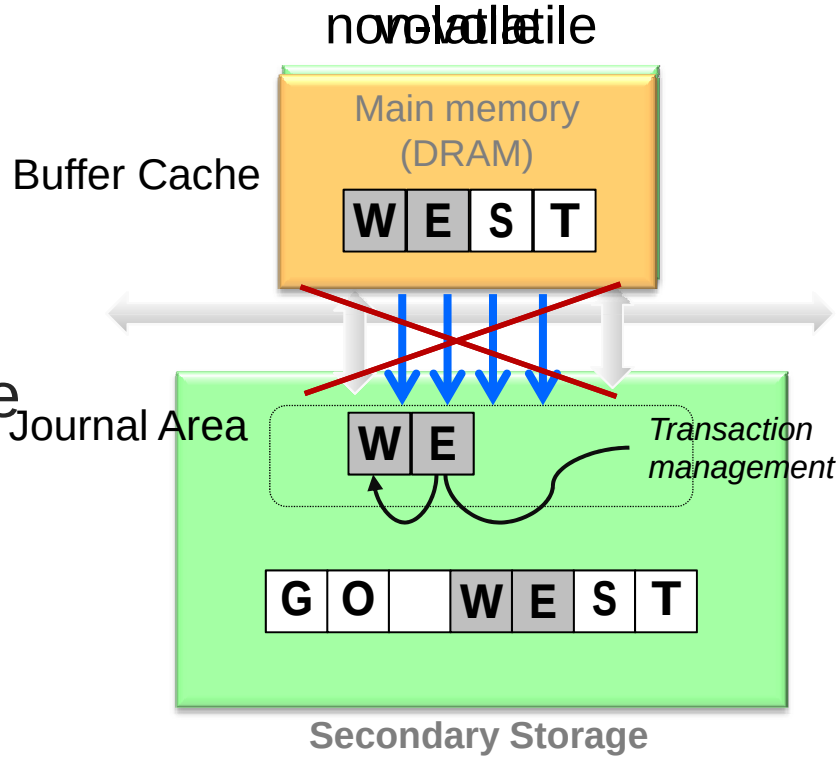
What if non-volatile main memory?

- ✓ However, inconsistency can occur with non-volatile main memory

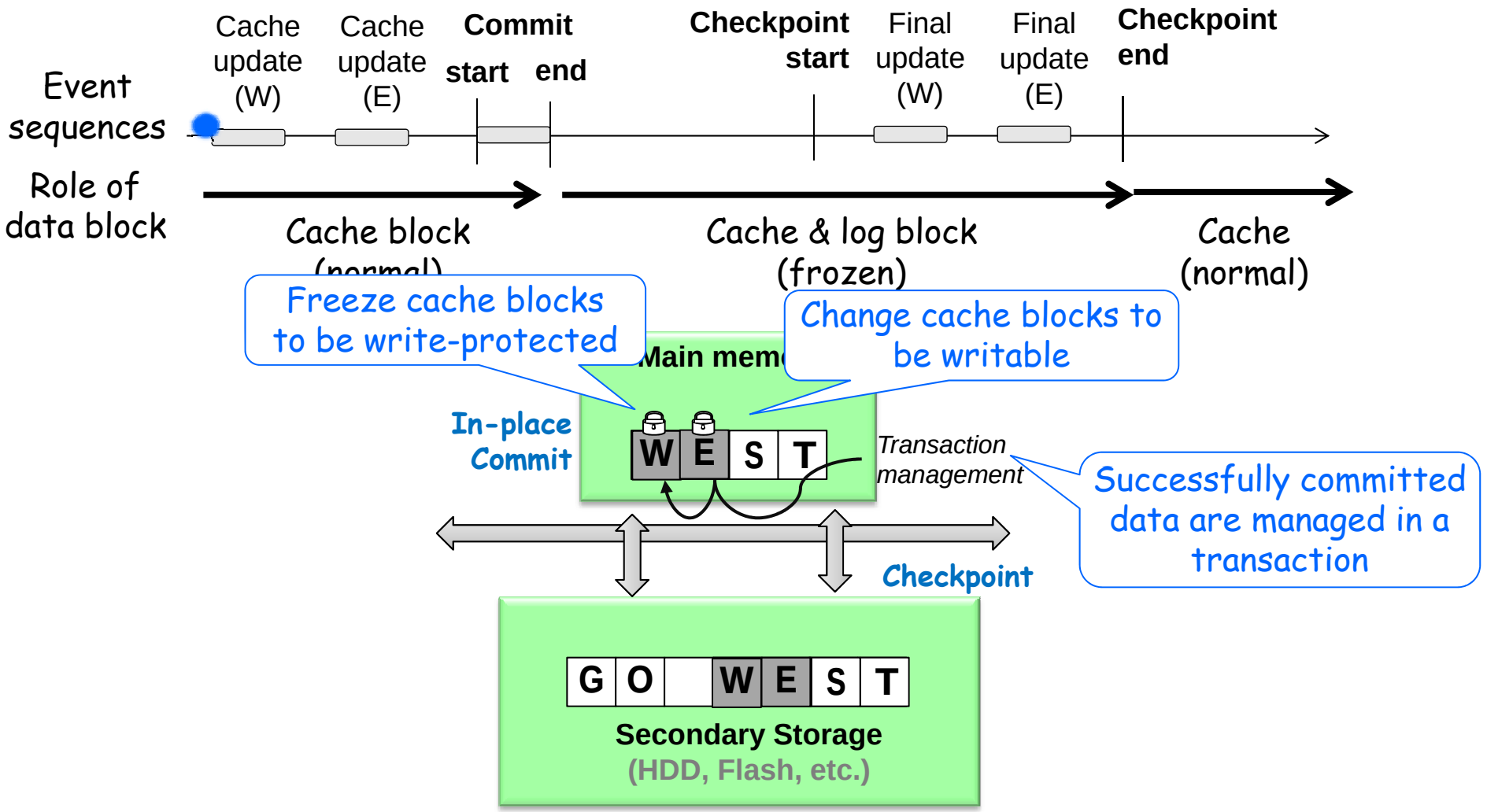


Unioning of Buffer cache and Journaling Layers (UBJ)

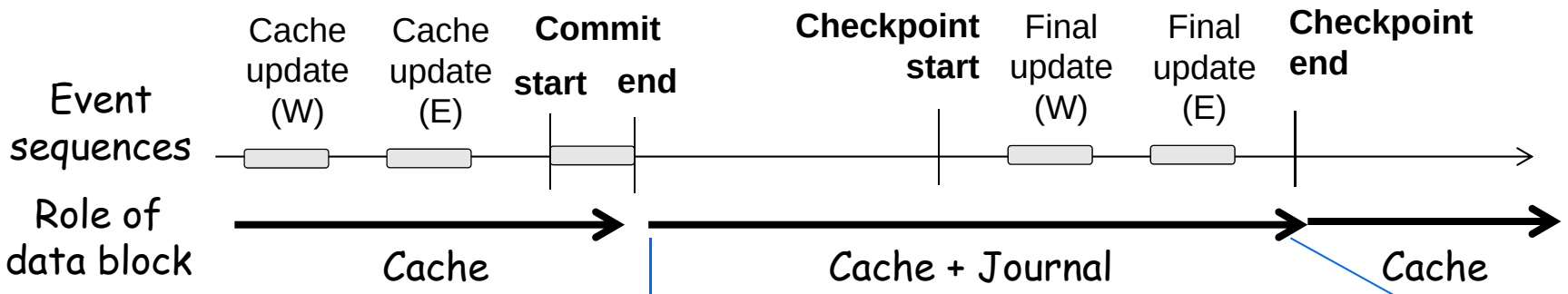
- Goal is providing data consistency without sacrificing performance
- Simply adopting non-volatile memory does not suffice
- Novel buffer cache architecture called “UBJ”
- Subsume functions of caching and journaling by using a data block for dual purposes
- Make a journaling effect just by changing the status of cache block instead of storage writes



Workings of UBJ



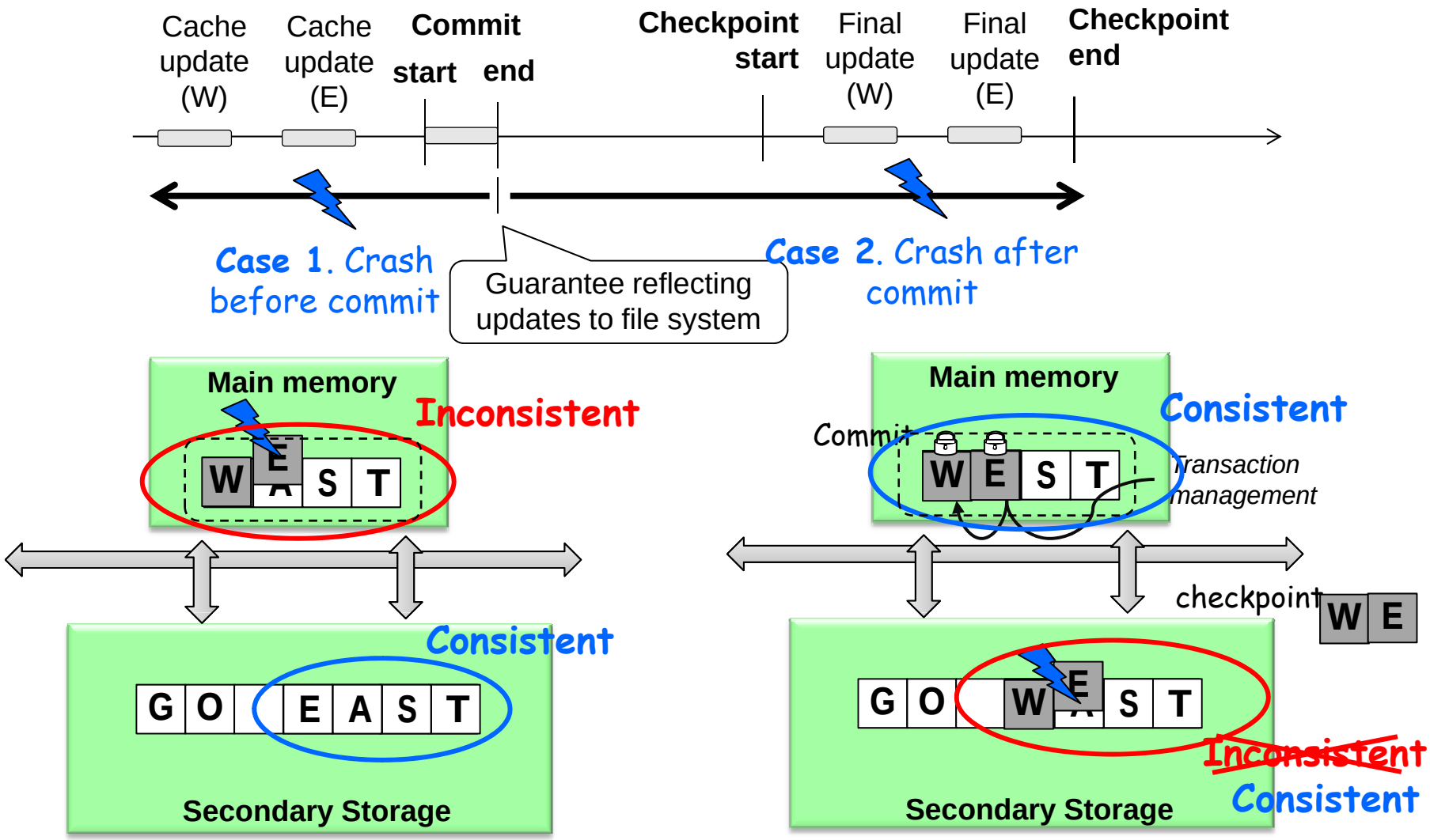
Workings of UBJ



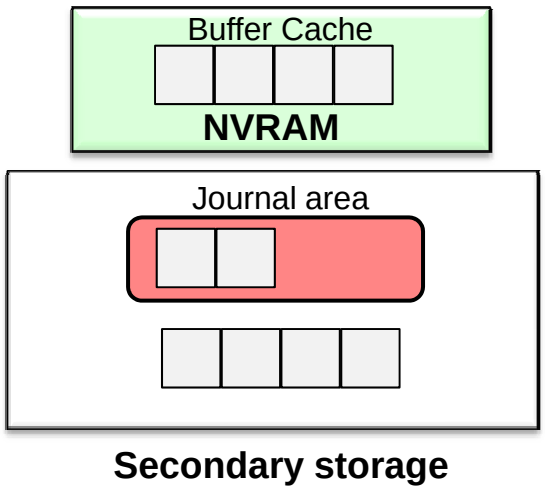
Journal data

- ✓ Managed by a transaction
- ✓ Protected from partial updates due to cache replacement
- ✓ Perform write requests via copy-on-write to protect commit data safely
- ✓ Serve read requests as cache blocks

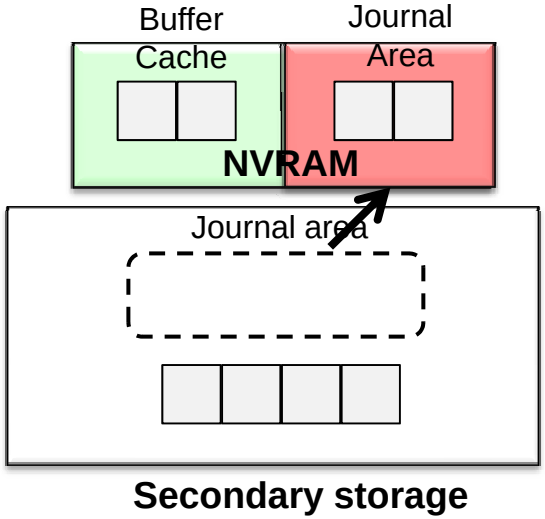
System recovery of UBJ



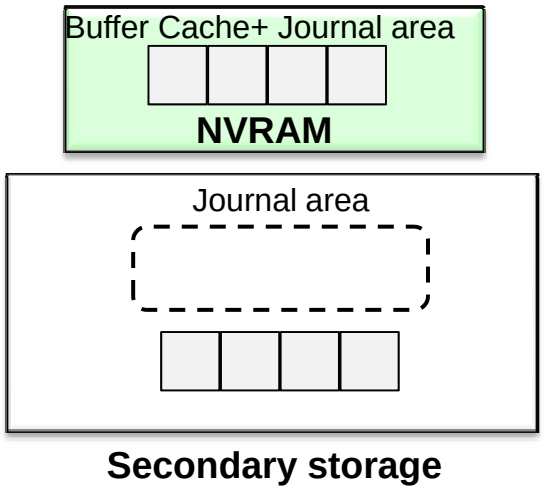
Cache performance of UBJ



1. Original buffer cache

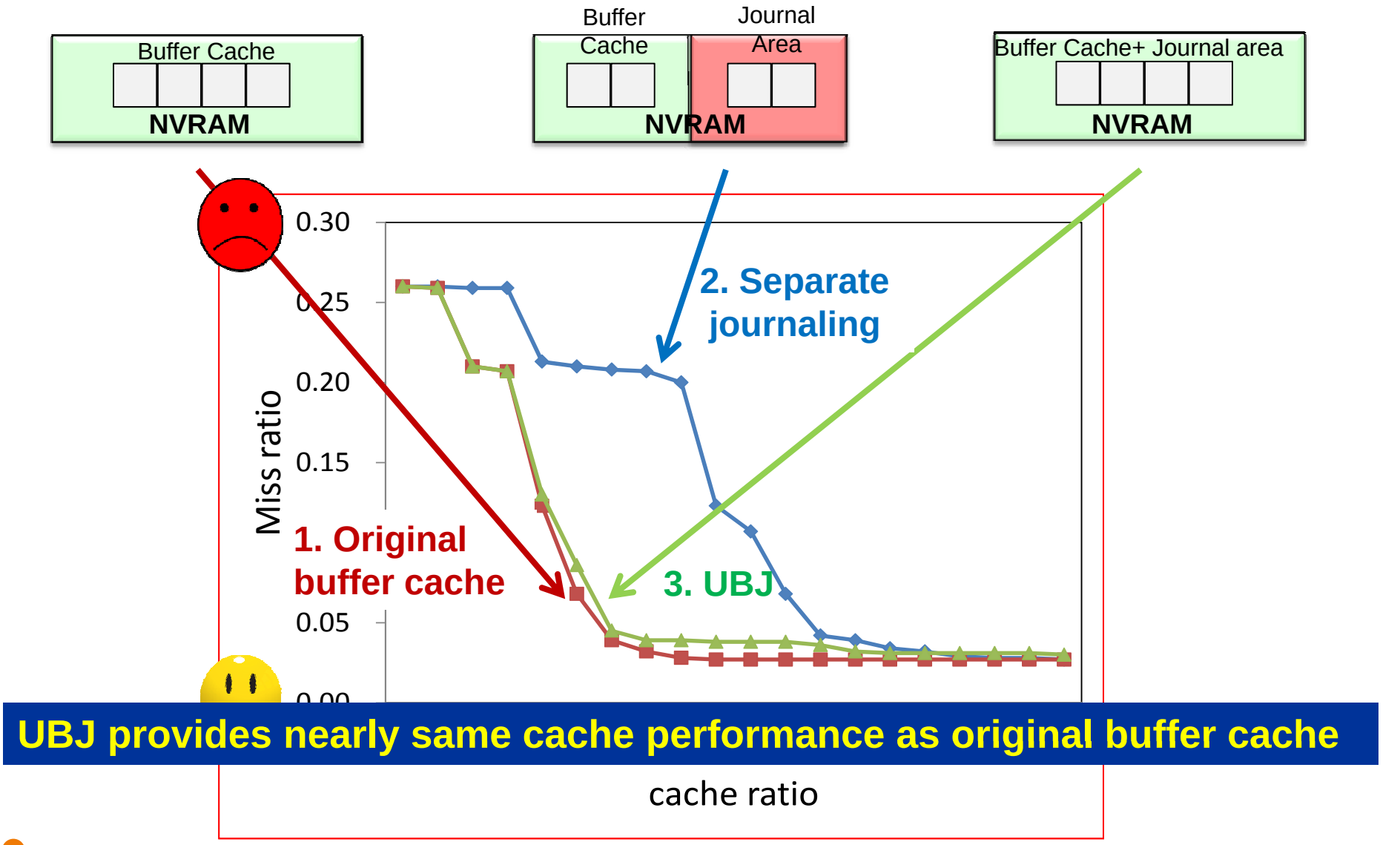


2. Separate journaling



3. UBJ

Cache performance of UBJ



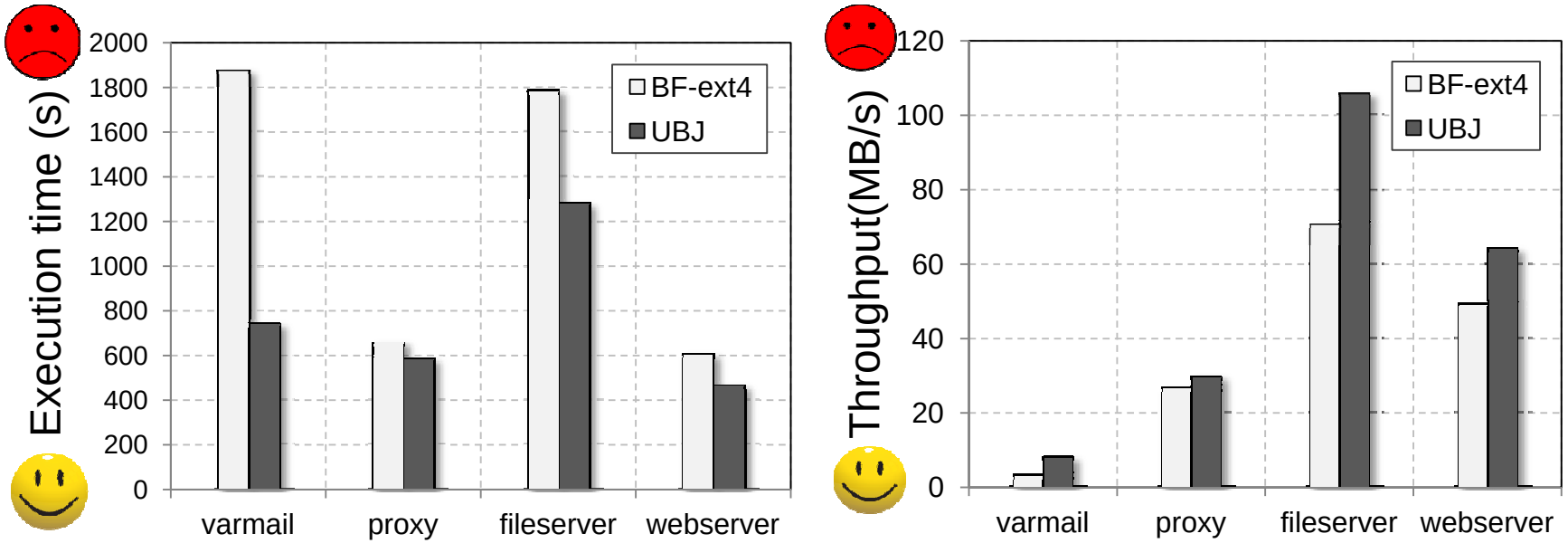
UBJ provides nearly same cache performance as original buffer cache

Performance Evaluation

- Prototype of UBJ on Linux 2.6.38
- Compare with ext4 in journal-mode
  - logs both data and metadata
- Intel Core i3-2100 CPU
  - 3.1GHz and 4GB of DDR2-800 memory
- Emulate non-volatile memory with DRAM
- Three benchmarks
  - Filebench, IOzone, Postmark

Performance Evaluation

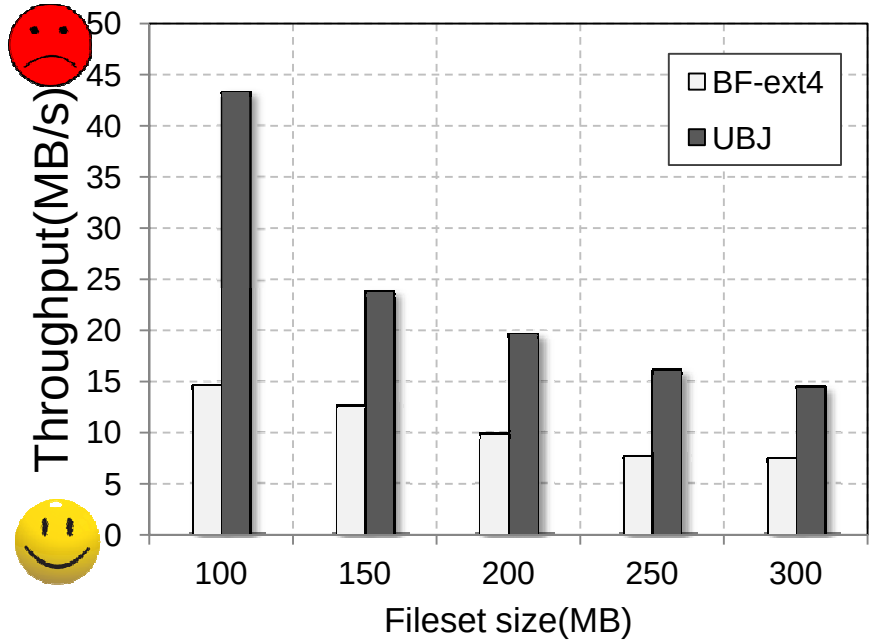
- Filebench



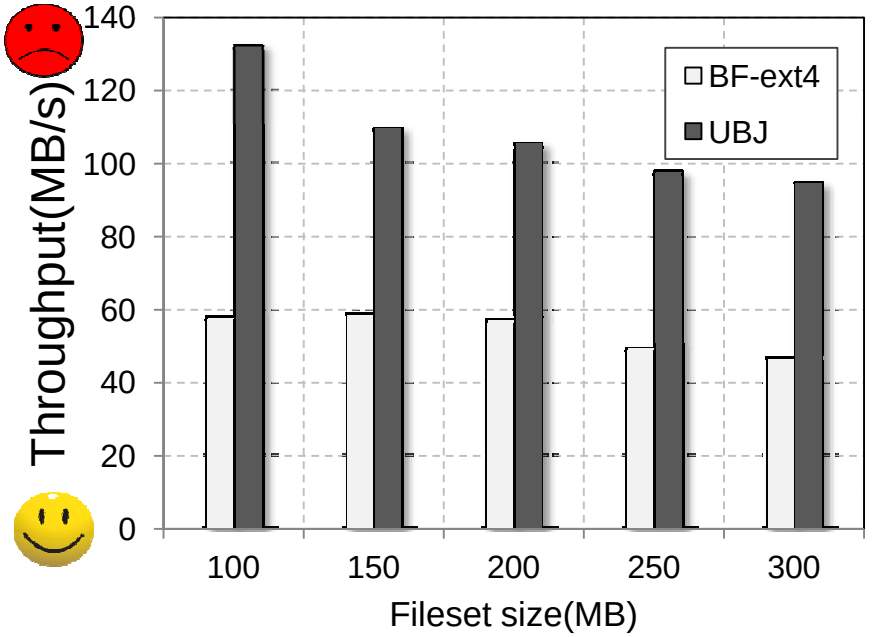
Improve execution time and throughput by 30.7% and 59.8% on average

Performance Evaluation

- Iozone



(a) Random write

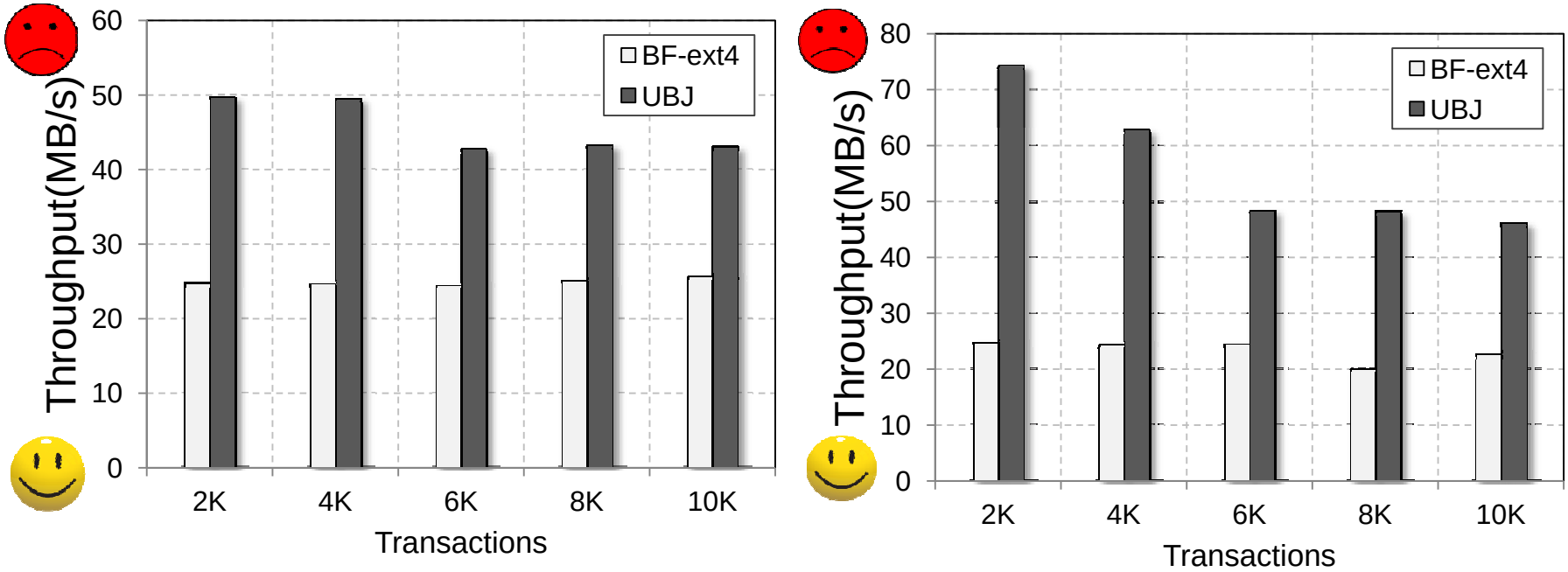


(b) Sequential write

Improve performance by 110% on average

Performance Evaluation

- Postmark



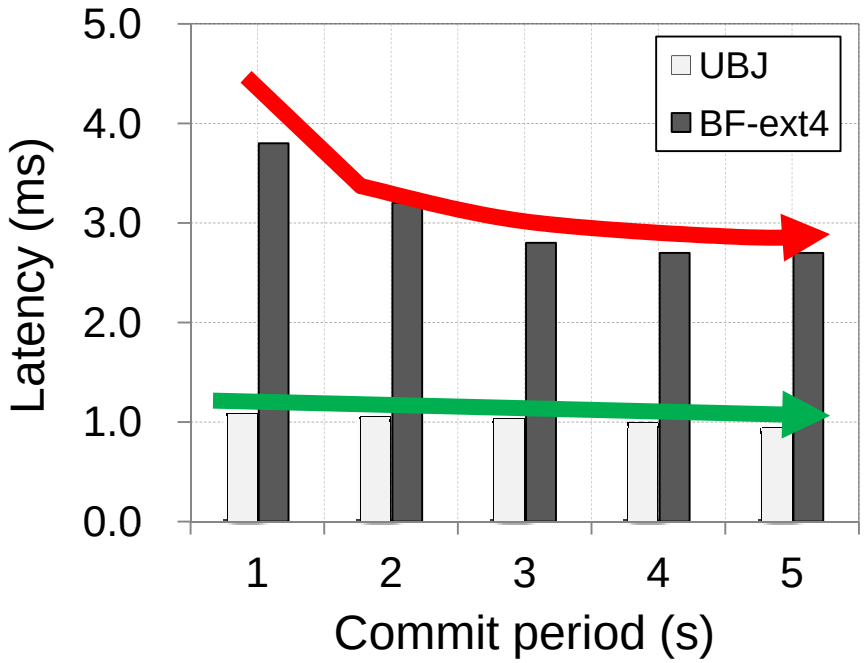
(a) Read

(b) Write

Improve performance by 109% on average

Performance Evaluation

- Effectiveness of UBJ on performance as the commit period changes



The latency of ext4 becomes small as the commit period is longer

The latency of UBJ is not sensitive to the commit period changes

Conclusion

- Novel non-volatile memory buffer cache architecture
- Subsumes the functions of caching and journaling
  - Buffer cache blocks ← → Journal logs
  - Notion of a frozen state
    - In-place Commit
  - Journal log block as well as a cache block
- Performance results
  - Implemented on Linux 2.6.38
  - Compared to ext4 in journal mode
  - Improve I/O performance by %76 and up to 240%