Warped Mirrors for Flash

Yiying Zhang

Andrea C. Arpaci-Dusseau

Remzi H. Arpaci-Dusseau

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON


Pacifica Airlines
flight 2340
SFO
TPE
Showtimes today

Flash-based SSDs in Storage Systems

- Using commercial SSDs in storage layer
  - Good performance
  - Easy to use
  - Relatively cheap

- Usage
  - MySpace, Facebook, Amazon, etc.
  - All-flash storage, e.g., Pure Storage

- **What about reliability?**

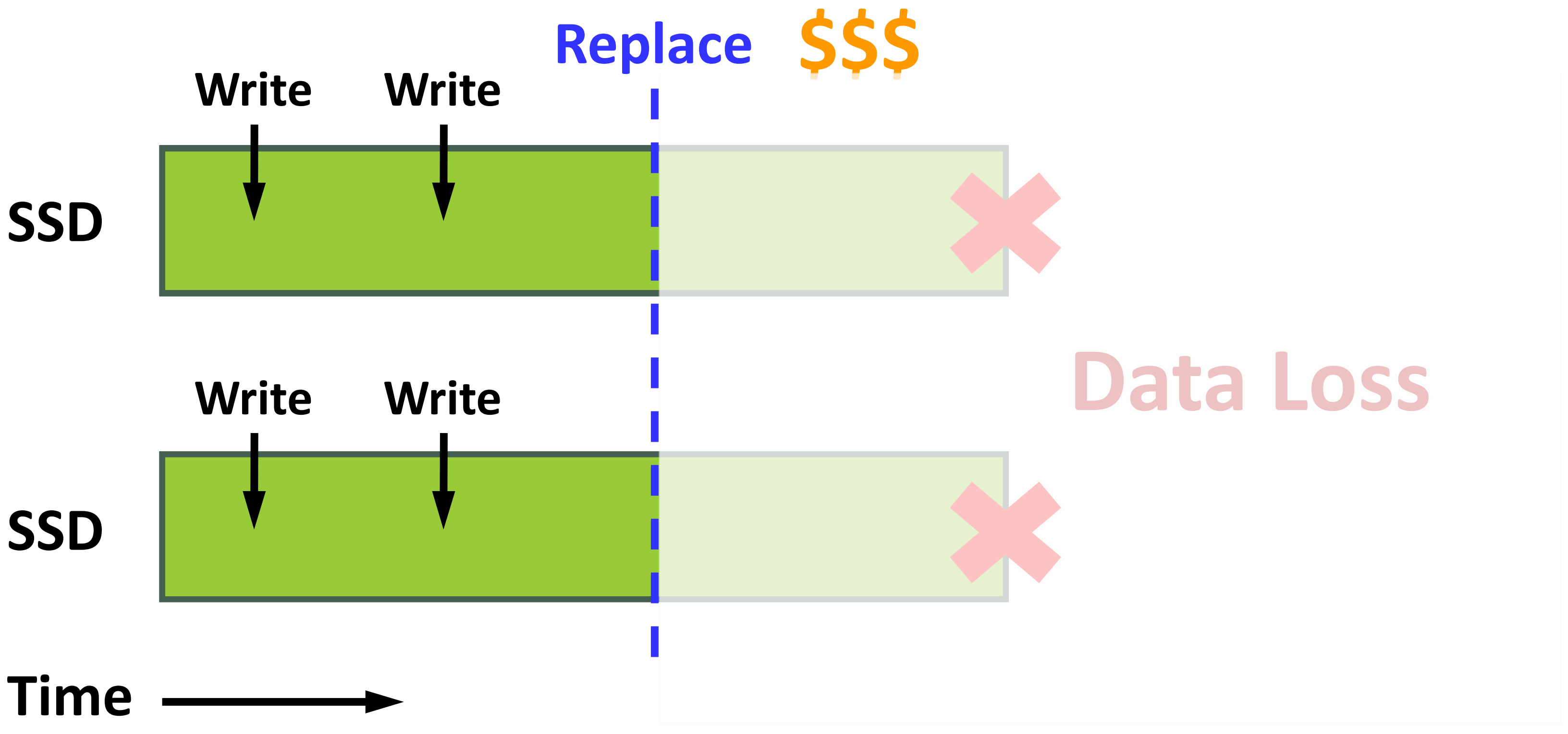
Flash-based SSD Reliability

- Flash wears out with erases
  - More writes => more erases
  - FTL and wear leveling help

- One way to improve SSD reliability
- Redundancy or RAID

Assume failure independence

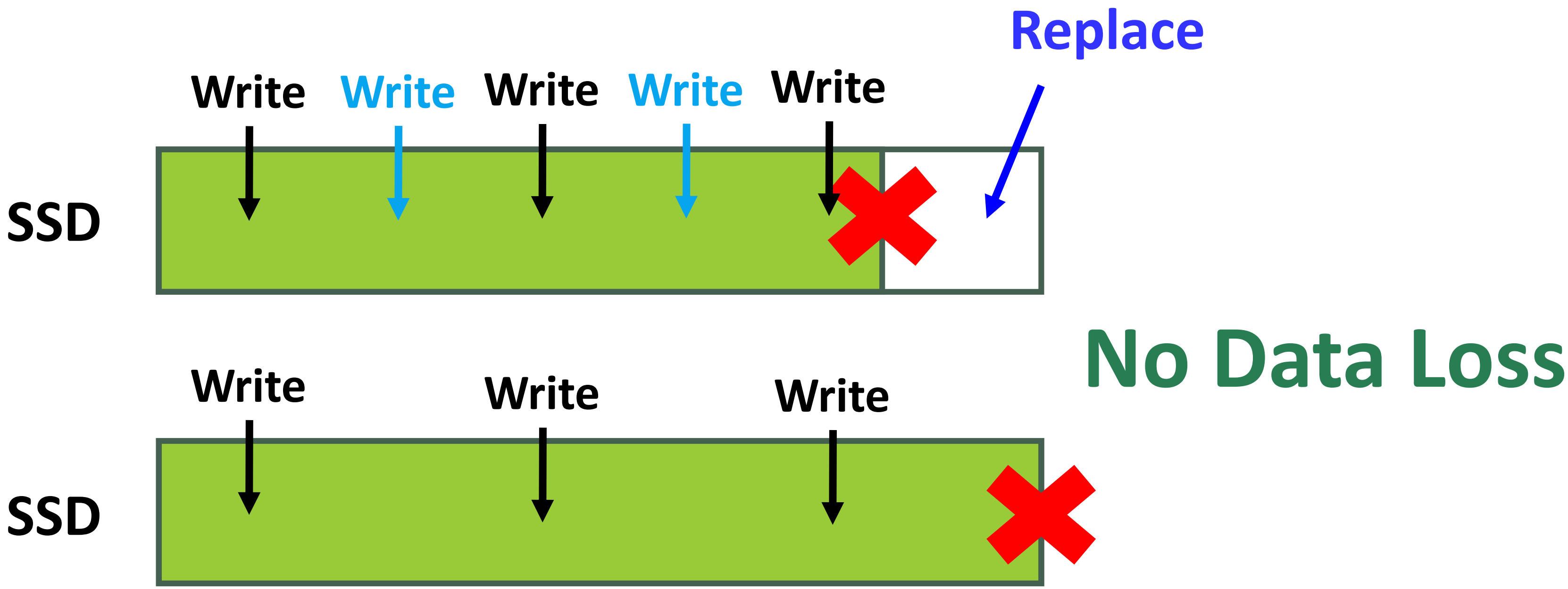
What About Flash-based Array?



Correlated failure !

WaM - Warped Mirrors for Flash

- Write more to one SSD to induce **earlier failure**



- Focus on mirrors (RAID1)

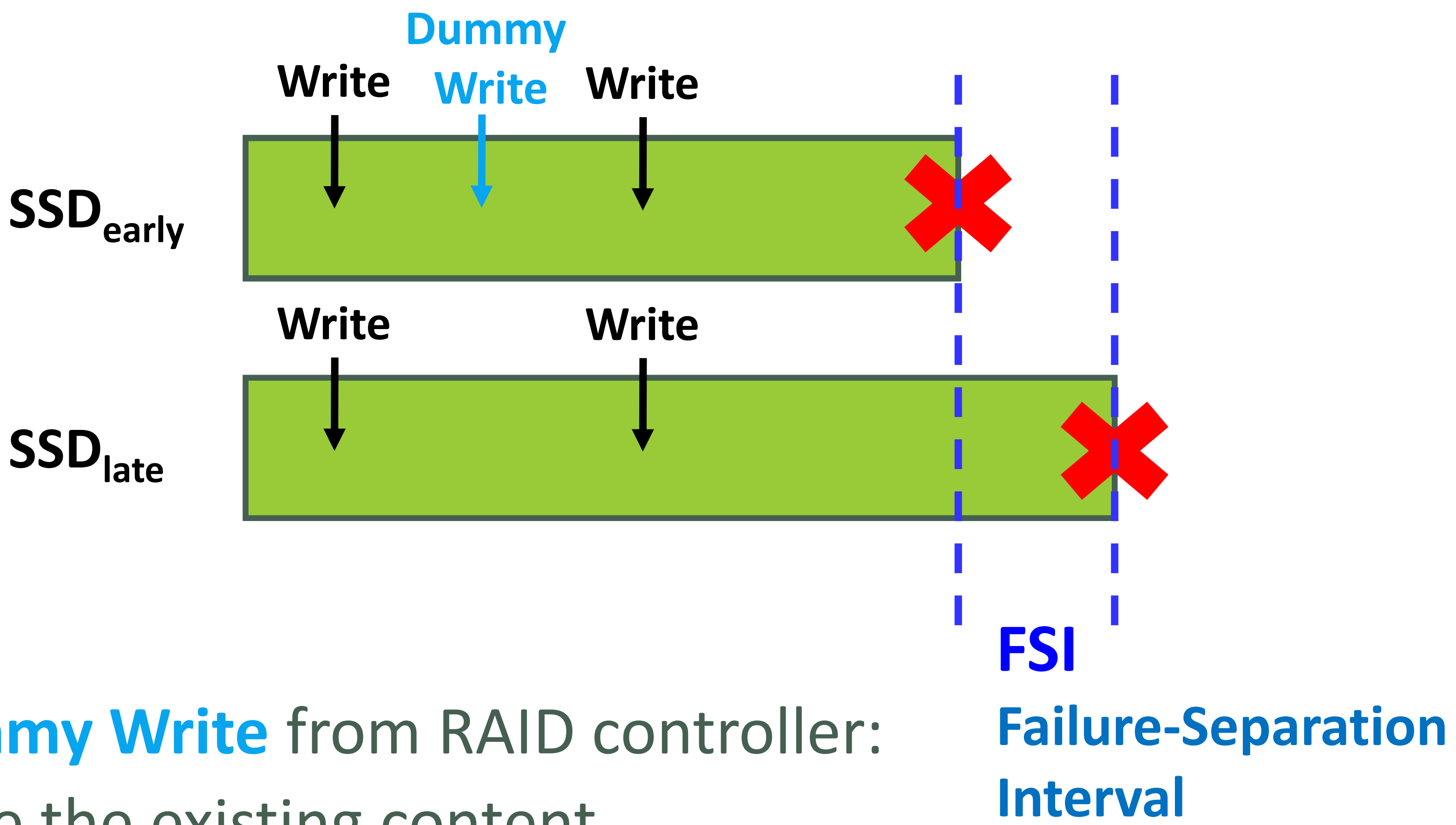
WaM Benefits

- Reliability achieved by **failure separation**
- Configurable
  - Approximated **model** + correcting method
- Low monetary cost
  - **1-2 cents** per hour for mirrors using WaM
  - **47-94%** of fixed-time replacement every one year
- Small performance overhead
  - **10%** more resp time for **52hr-159day** separation

Outline

- Introduction
- WaM design and model
- Evaluation results
- Conclusion

Basic Solution - Adding Dummy Writes



Dummy Write from RAID controller:

Write the existing content

From last write or a random page

Failure Separation Interval

- FSI: window for detection and reconstruction
  - Set by administrator at initialization time
  - Can be adjusted

- Choosing FSI
  - Long enough for recovery
  - Short to avoid high performance cost

How many dummy writes to add given an FSI?

Challenges

- Subverting FTL
  - No knowledge of underlying FTL
- Achieving near-perfect FSI
  - FSI cannot be shorter than target (reliability)
  - Performance overhead should be minimized

WaM Model

- Model based on
  - Target FSI length
  - SSD properties
  - Workload properties

- Goal
  - Find dummy write percentage for a target FSI

WaM Model – Dummy Write Percentage

- Ratio of erases between two mirrored SSDs

$$R_{erase} = \frac{N^{early}_{erases}}{N^{late}_{erases}}$$

Number of erases issued by SSD_{early}

Number of erases issued by SSD_{late}

- Dummy write percentage P_{dummy}

$$R_{erase} = 1 + P_{dummy}$$

$$P_{dummy} = \boxed{R_{erase}} - 1$$

WaM Model – Num Erases Remaining

Maximum number of erases of an SSD block (SSD_{late})

$$N_{remaining}^{late} = N_{worn} - N_{erases}^{late}$$

Number of erases with SSD_{late} when SSD_{early} dies

$$N_{erases}^{late} = \frac{N_{worn}}{R_{erase}}$$

SSD_{early}

$$\boxed{N_{remaining}^{late}} = N_{worn} - \frac{N_{worn}}{R_{erase}}$$

WaM Model – Num Erases during Time

$$N_{I/Os} = \boxed{\frac{T}{T_r + T_i}}$$

Workload dependent

Avg Response Time Avg Idle Time

$$N_{erases}^{total}(T) = \frac{T}{T_r + T_i} \times P_{writes} \times \boxed{\frac{S_{page}}{S_{block}}}$$

Flash Page Size

Knowledge of SSD parameters

Write Percentage

Flash Erase Block Size

$$N_{erases}^{perblock}(T) = \frac{T}{T_r + T_i} \times P_{writes} \times \boxed{\frac{S_{page}}{S_{block}} \times \frac{1}{N_{ssd}}}$$

Perfect wear leveling

Num of Erase Blocks in SSD

WaM Model – Final Steps

$$N_{remaining}^{late} = N_{erase}^{perblock}(\mathrm{FSI})$$

$$N_{worn} - \frac{N_{worn}}{R_{erase}} = \frac{FSI}{T_r + T_i} \times P_{writes} \times \frac{N_{page}}{N_{block}} \times \frac{1}{N_{ssd}}$$

$$R_{erase} = \frac{N_{worn}}{N_{worn} - \frac{FSI}{T_r + T_i} \times P_{writes} \times \frac{N_{page}}{N_{block}} \times \frac{1}{N_{ssd}}}$$

$$P_{dummy} = R_{erase} - 1$$

Assumptions and Limitations

- Device parameters
  - From device vendor or detect with tool
- Workload changes
  - Adjust model as workloads change
- Imperfect or no wear leveling
- Incorrect SSD lifetime

Violations: FSI too short or too long

Achieving Target FSI

$$R_{delay} = \frac{N^{late}_{remaining\_target}}{N^{late}_{remaining\_actual}}$$

- If FSI too short
  - Delay writes to the surviving SSD

- If FSI too long
  - Performance cost
  - Adjust in future WaM modeling

Recovery

- When the first SSD (SSD_{early}) fails
  - Replace with a new SSD
  - Reconstruct the data
- Replacing the second SSD (SSD_{late})
  - At the same time when first SSD fails (no reliability risk, slightly higher cost)
  - When it fails (higher reliability risk, slightly low cost)

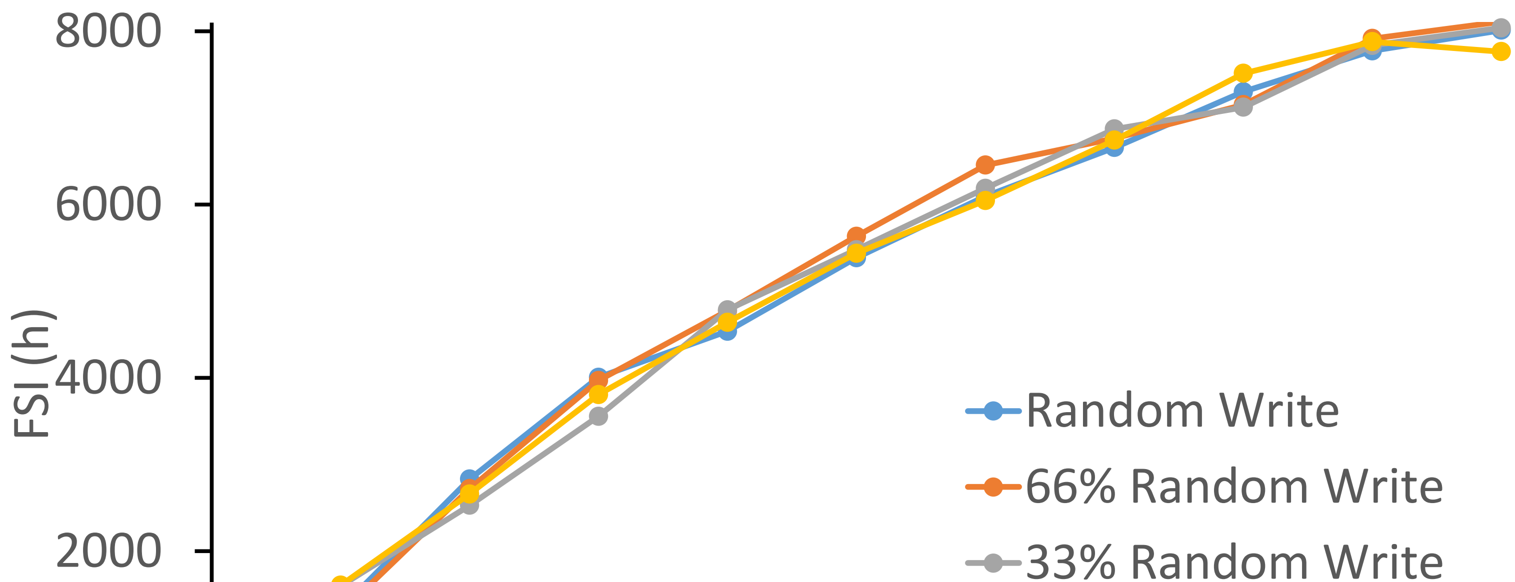
Outline

- Introduction
- WaM design and model
- Evaluation results
- Conclusion

Evaluation Environment

- Simulation based on Disksim + SSD extension
- A mirror pair of two 80GB SSDs
- Workloads
  - Microbenchmark
  - Macrobenchmark
  - Trace
  - No idle time

Can Failures Be Separated with Dummy Writes? And How?

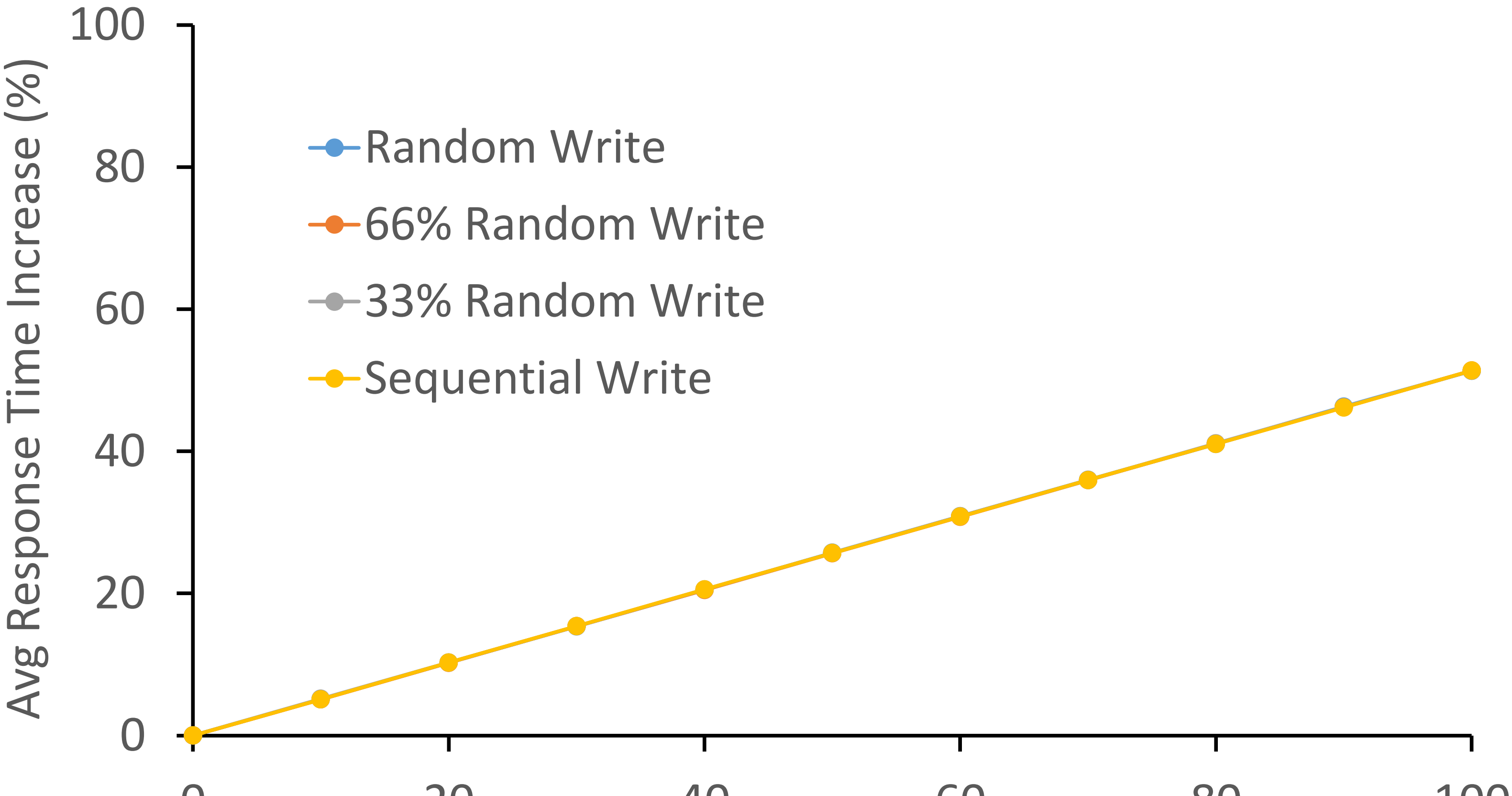


Failures can be separated with dummy writes

More dummy writes -> longer separation

Wear leveling homogenize workloads

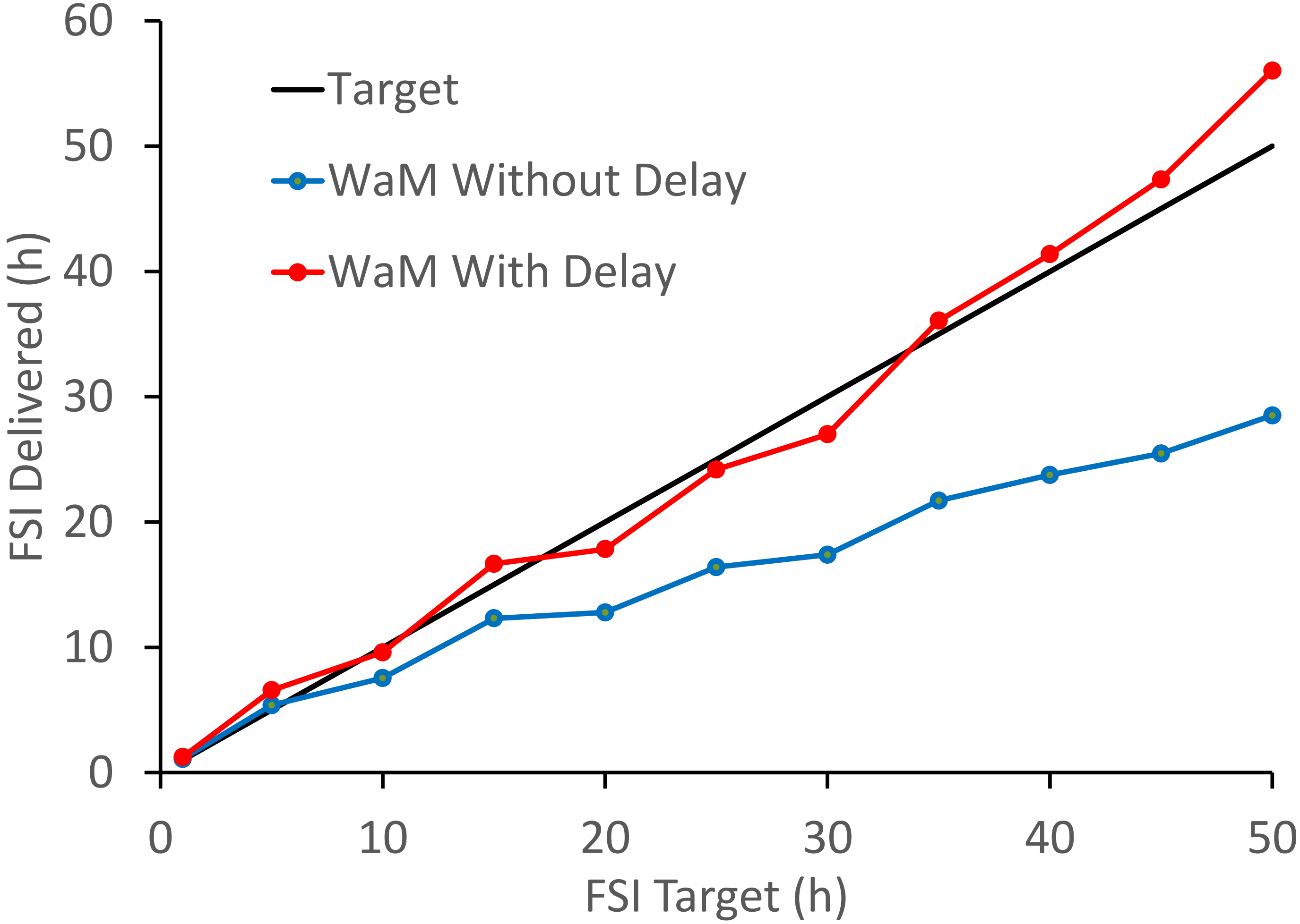
What Is the Performance Overhead?



More dummy writes -> worse performance

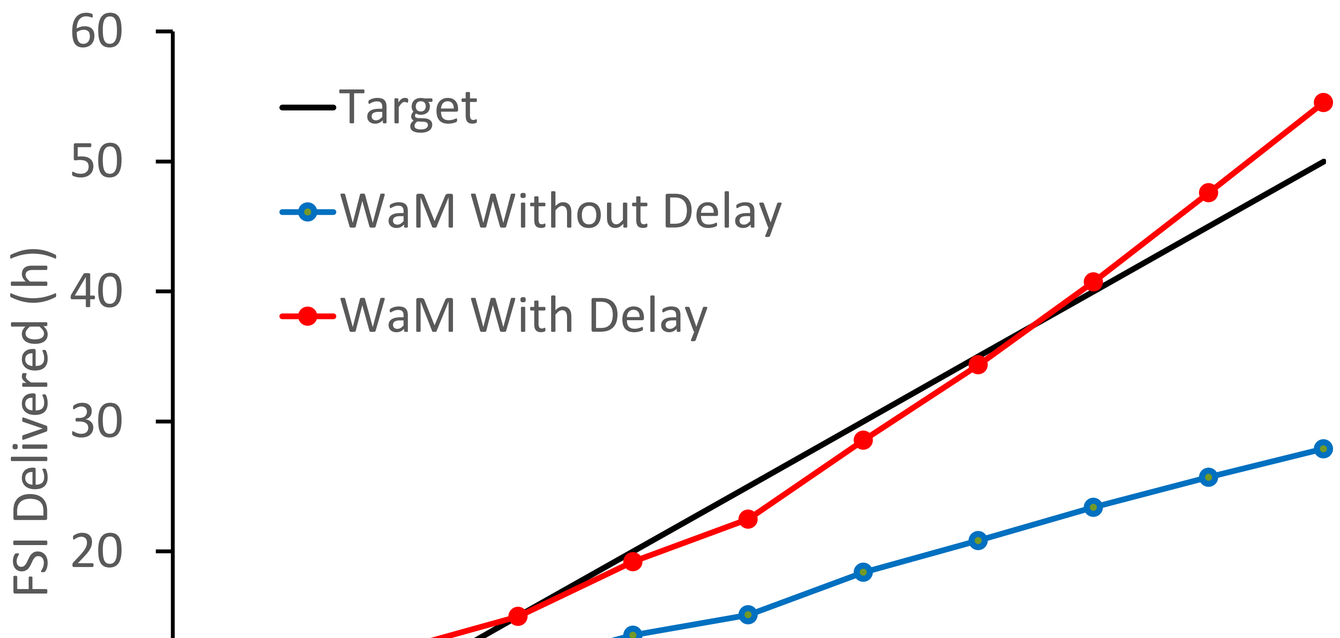
Can the Correct FSI Be Achieved?

- Sequential workload



Can the Correct FSI Be Achieved?

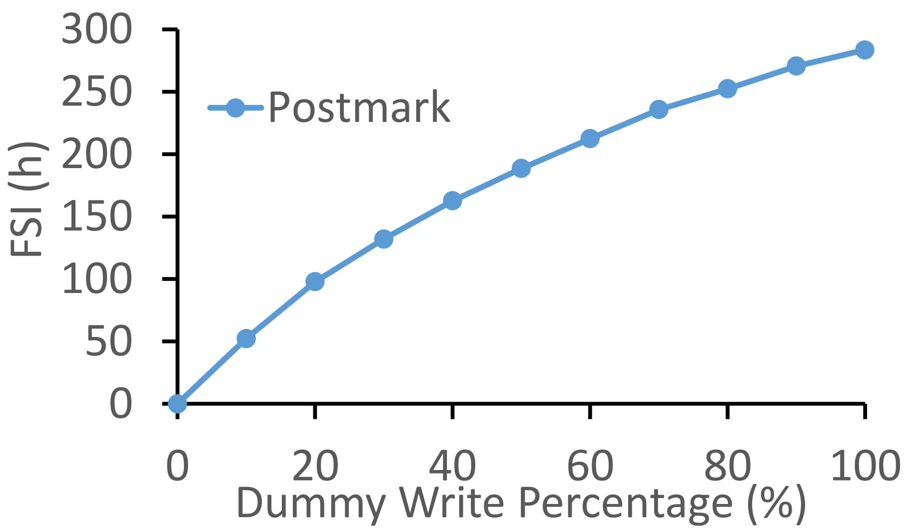
- Random workload

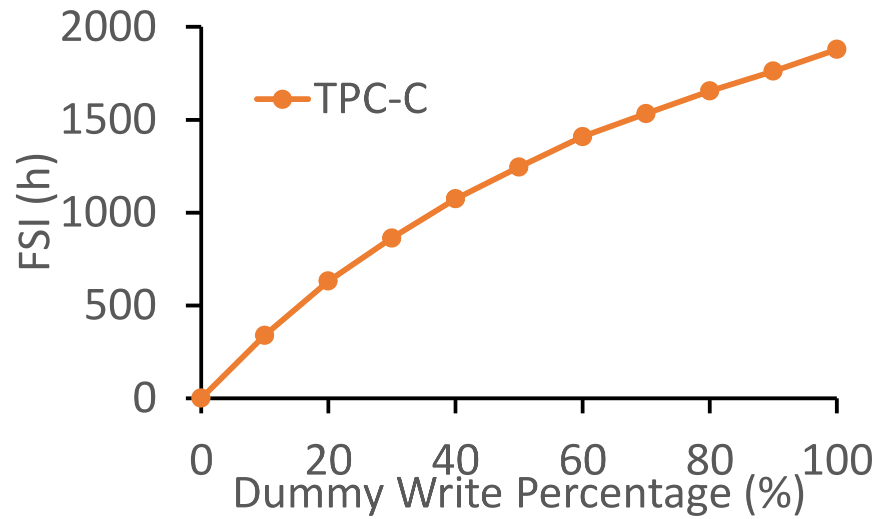


WaM model can be inaccurate

Target FSI can be delivered with delaying

How about Real Workloads? - FSI





FSI and dummy write relationship as expected

Larger FSI with read-intensive workloads

Dummy Write Percentage (%)

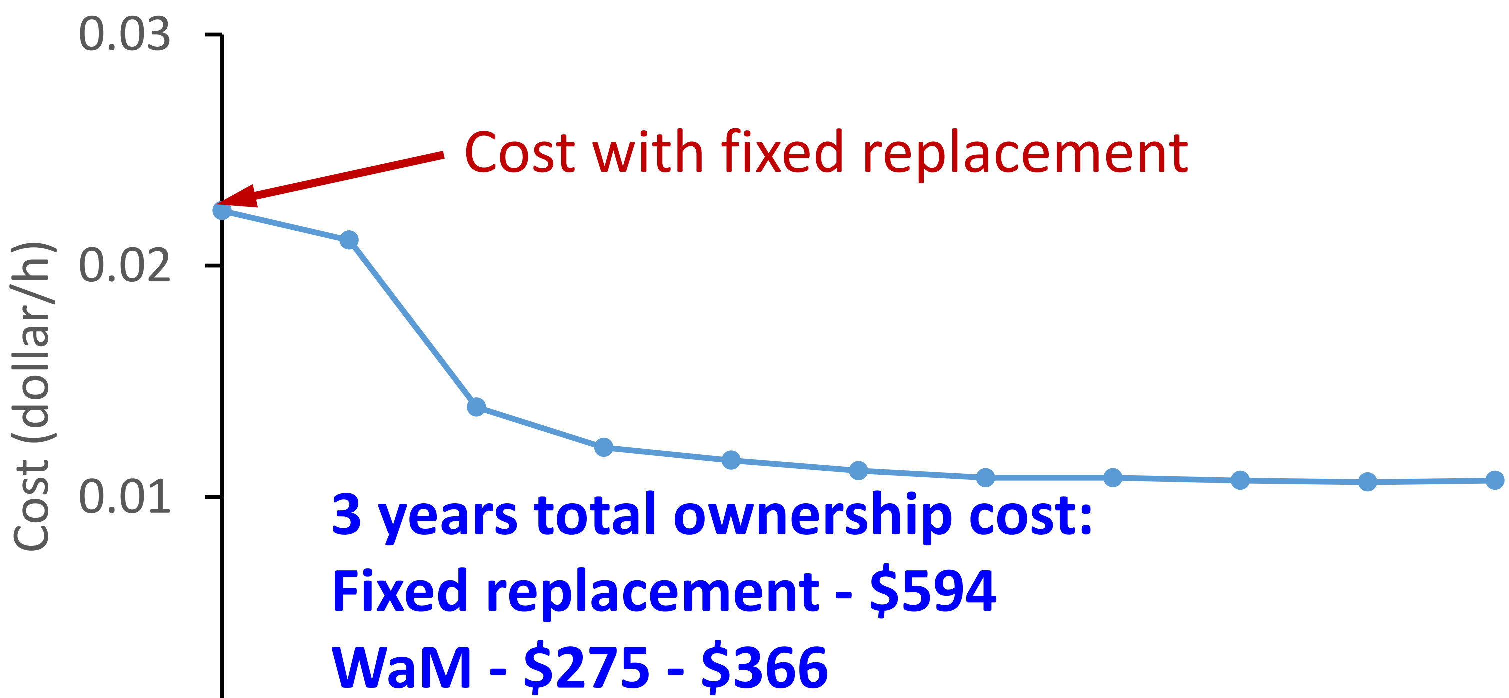
How about Real Workloads? - Performance

Higher overhead with write-intensive workloads

Performance overhead is small for typical FSI

What is the Monetary Cost?

- WaM: cost of SSD + sys-admin check each FSI interval
- Fixed replacement: replace SSD after one year



WaM costs lower than fixed-time replacement

Summary of Results

- Failures are separated with desired FSI

- Model is approximated
- Achieves desired FSI with delaying

- Small performance overhead

- Low monetary cost

Outline

- Introduction
- WaM design and model
- Evaluation results
- Conclusion

Conclusion

- Correlated failure of flash-based RAID
- Separate failures by carefully adding dummy writes and delaying writes
- Other techniques for failure separation
  - Wear our one SSD to some extent before using
  - Stagger SSDs with different ages in a RAID
  - Vendor control when SSDs in RAID fail

Conclusion

- Applying existing solutions directly to new devices may not work
- WaM is a simple solution to guarantee failure separation and pushes aggressive use of SSDs
- Other techniques may work well
- WaM model can be useful

Thank You

Questions?

WISDOM http://wisdom.cs.wisc.edu/home

ADSL http://research.cs.wisc.edu/adsl