NVMW 22

TECHNION | The Henry and Marilyn Taub Faculty of Computer Science

Offline and Online Algorithms for SSD Management

Tomer Lange, Joseph (Seffi) Naor, Gala Yadgar

Computer Science Department

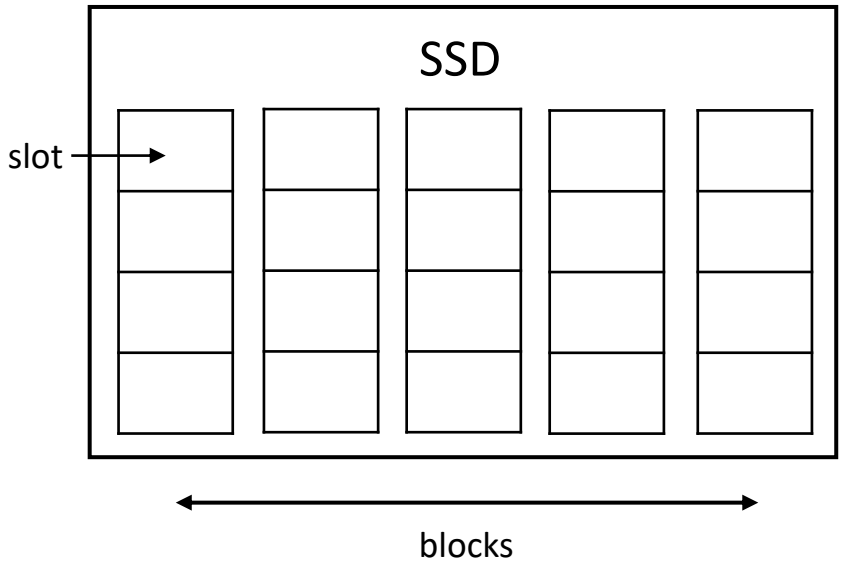
Technion – Israel Institute of Technology

Lange, Naor and Yadgar. Offline and Online Algorithms for SSD Management. Proc. ACM Meas. Anal. Comput. Systems, 2021.

Background

SSD's limitations

- Out-of-place page writes
- Writes in page granularity
- Cleanings in block granularity
- Upon cleaning, valid pages must be rewritten
- Page rewrites are bad
  - Reduce endurance
  - Harm performance

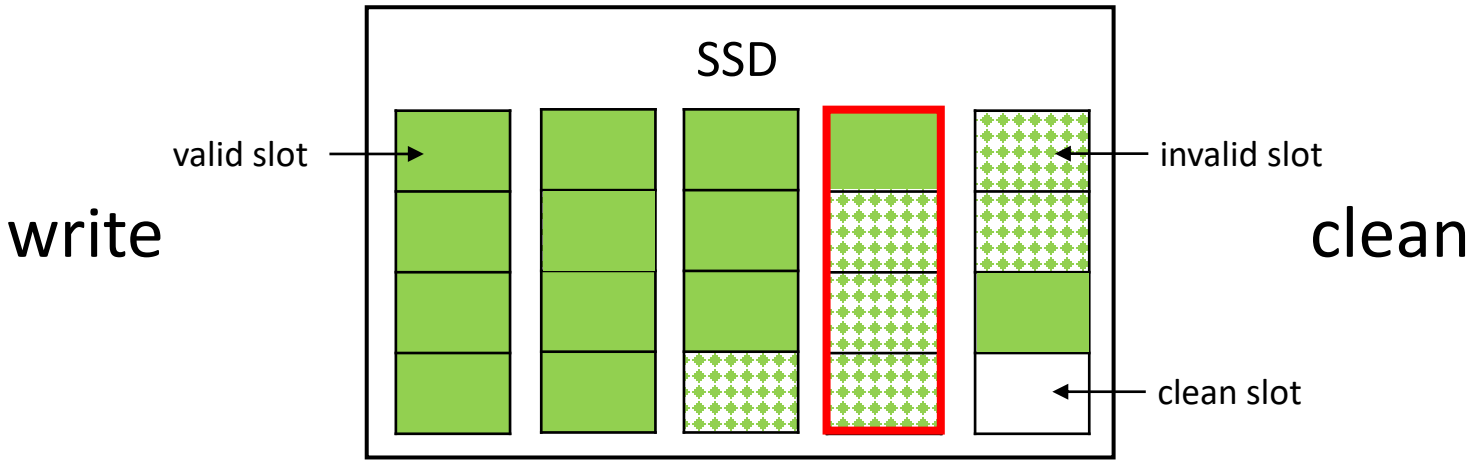


Contribution

- ✓ Formal definition of the SSD management problem
- ✓ Worst-case analysis of existing and new algorithms
- ✓ Novel, efficient approach for reducing WA
  - ➢ In both offline and online settings

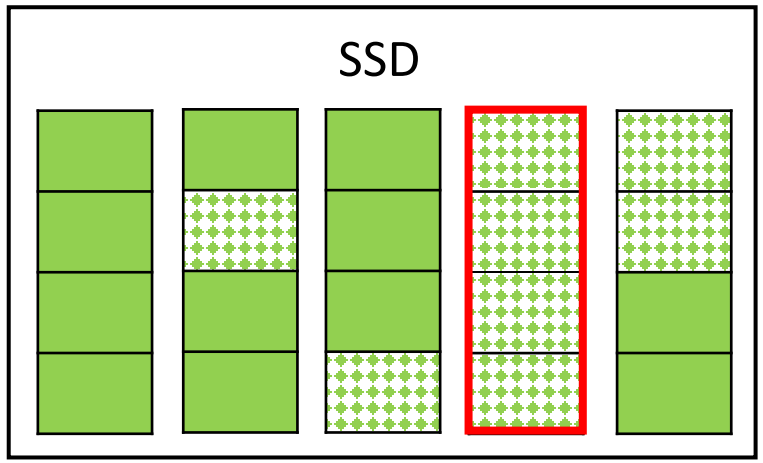
Problem Statement

- At every time t a new write request arrives
- The requested page should be **written** to a clean slot
- Upon cleaning, every valid page must be **rewritten** to a clean slot
- Goal: minimize WA



Problem Statement

- At every time t a new write request arrives
- The requested page should be **written** to a clean slot
- Upon cleaning, every valid page must be **rewritten** to a clean slot
- Goal: minimize WA



clean

Classical Algorithms

	Victim selection	Placement strategy	Performance
Greedy	Min-valid block	–	Optimal under uniform distribution (Yang et al., SIGMETRICS 2015)
Hot/cold	Greedy cleaning within each area	Allocate different blocks for different temperatures	Better than Greedy on skewed distributions (Desnoyers, SYSTOR 2012)

Myopic Algorithms

Definition

Myopic algorithm is an algorithm that has no information about the future.

Definition

The **spare factor** is defined by $\alpha = \frac{|physical\_capacity| - |logical\_capacity|}{|physical\_capacity|}$.

Theorem 1

For any request sequence, $WA_{Greedy} \leq \frac{1}{\alpha}$.

Theorem 2

No deterministic myopic algorithm provides guarantee better than $\frac{1}{\alpha}$.

Theorem 3

No randomized myopic algorithm provides guarantee better than $\frac{1}{2\alpha}$ (Yao's principle).

Approach

Definition

The **death time** of a page is the time it will be accessed next.

- Optimally, each block should group pages with **consecutive death times** (He et al., EuroSys 2017)
  - ➢ Blocks are invalidated sequentially; no rewrites needed
- Unfortunately, we mostly cannot maintain optimal layout

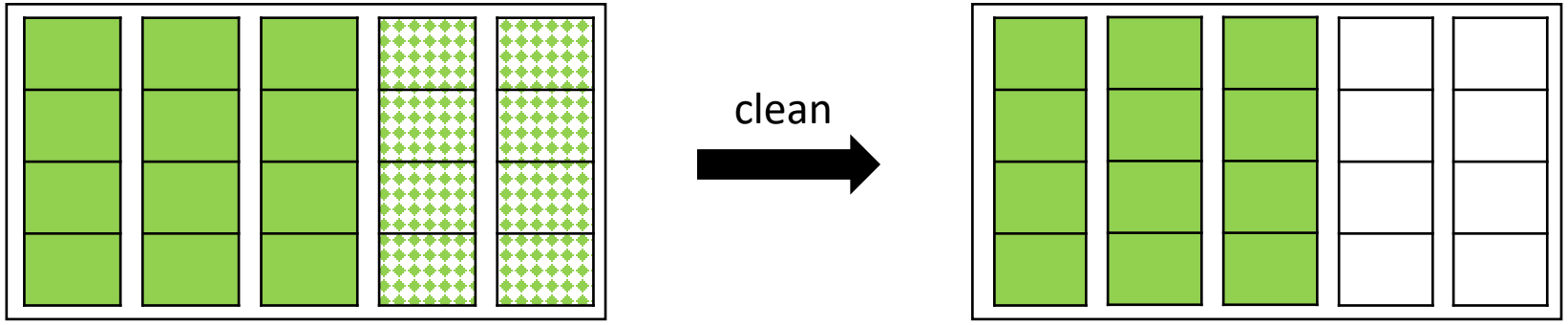
Our approach

- Start with optimal layout
- Serve new requests while keeping layout “almost optimal”
- When layout is “too far” from optimal – **reorder**

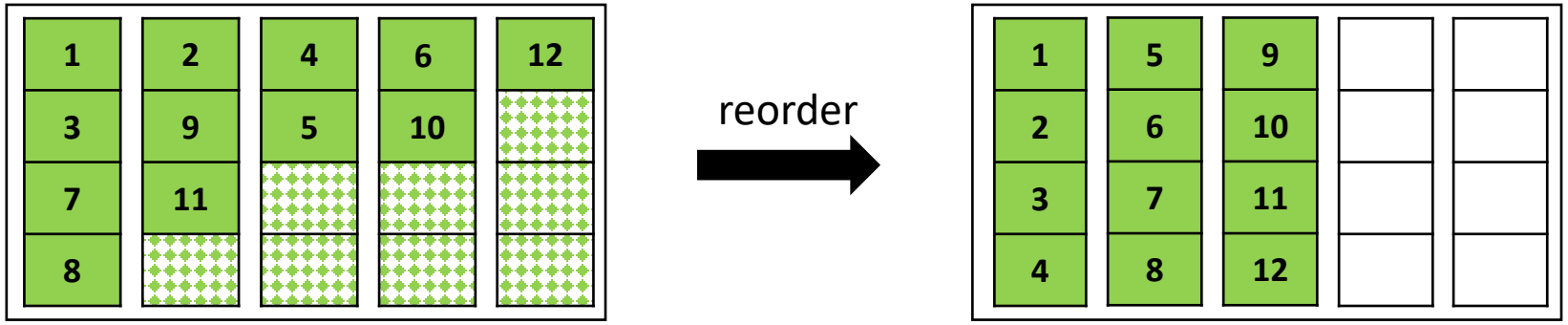
Offline Algorithm

Victim selection

- If completely invalid blocks exist, clean all of them



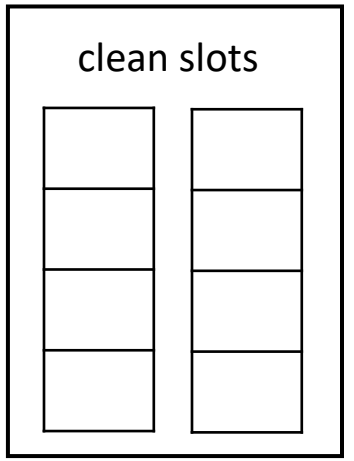
- Otherwise, **reorder** all pages by death times



Offline Algorithm

Placement

- Place new pages sorted by death times, until no clean slot is available



reordering step is rare

Theorem

For any request sequence, $WA_{offline} \leq 1 + O\left(\frac{1}{B}\right)$.

#blocks

Computational Complexity

- Is our algorithm optimal?
- Consider the following (easier) **decision problem**:

> Input: request sequence σ
>
> Question: can σ be served **without rewrites**?

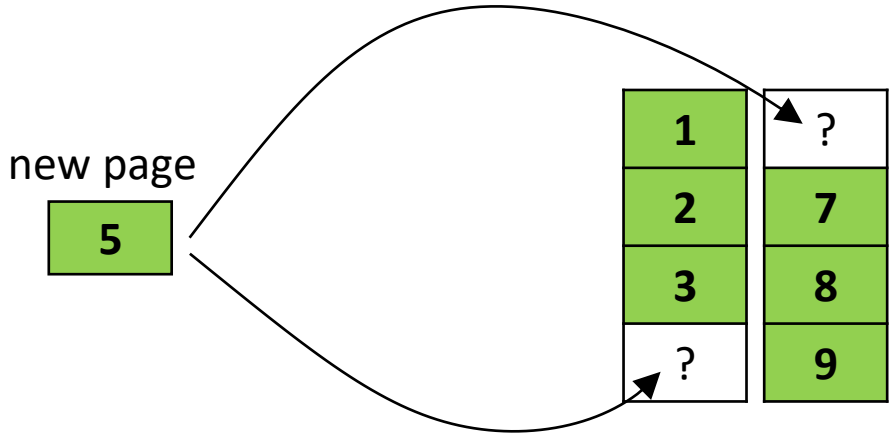
- Special case of *fragmented coloring* on interval graphs
  - ➢ Generalizes standard vertex coloring
- Computational status is open

Online Setting with Predictions

Assumption: each request arrives with a **predicted** death time

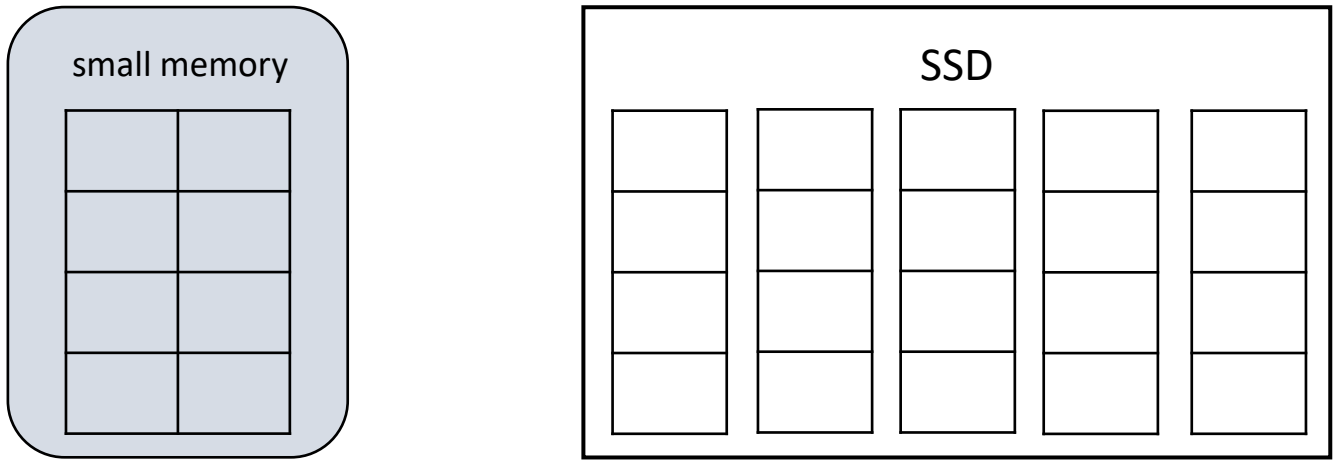
Placement strategy: sort pages by predicted death times

- **Problem:** in which block to place page with death time = 5?



Online Setting with Predictions

- **Proposed solution:** a small, fast additional memory
- First aggregate pages in memory, then move to SSD



Online Setting with Predictions

Assumption: each request arrives with a **predicted** death time

- Challenge: predictions might be erroneous
- Prediction error: $\eta = \sum_t |y(p,t) - \hat{y}(p,t)|$

real death time — predicted death time

- Placement strategy: “believe predictions”
- What about victim selection?

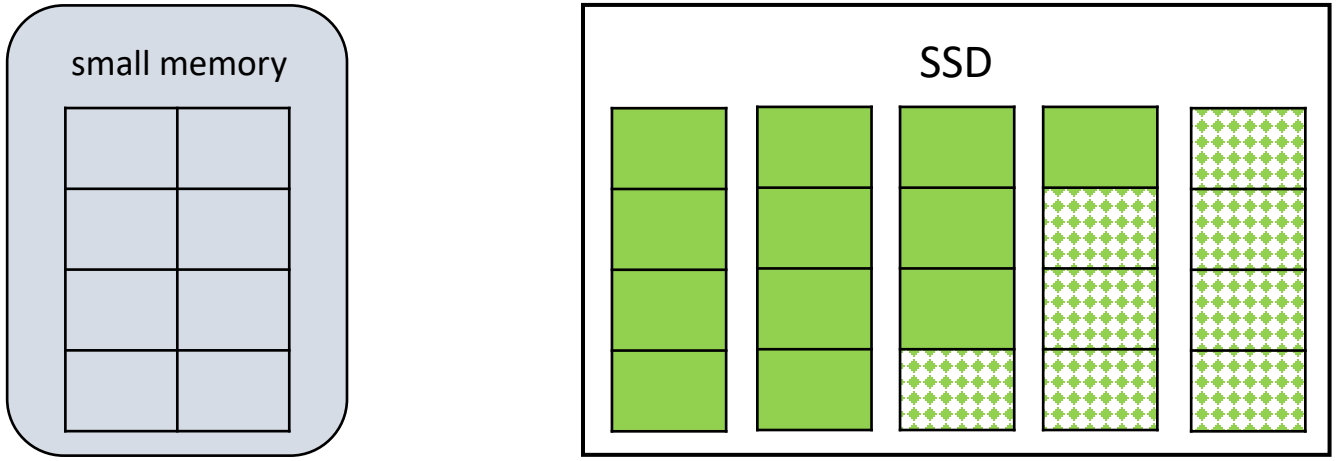
Online Setting with Predictions

Victim selection strategy

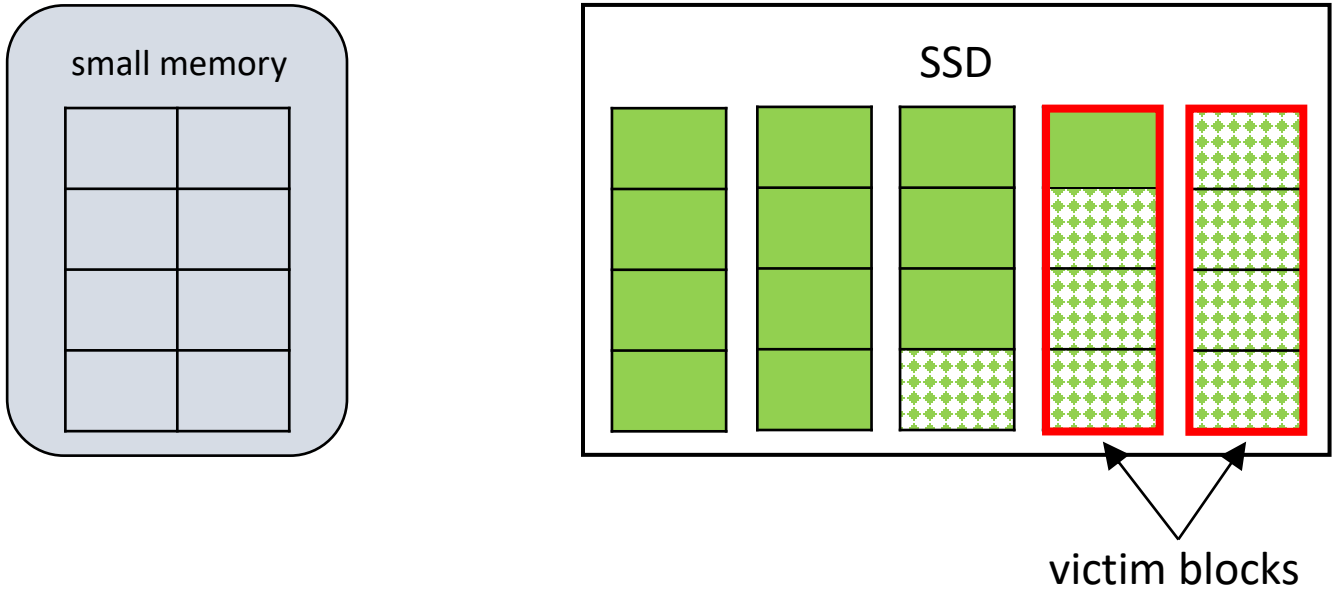
❌
- Option 1: consider predictions
  - For example: avoid cleaning a “hot” block
  - Problem: large prediction error can be disastrous

✔
- Option 2: clean min-valid blocks, **regardless of predictions**

Online Algorithm: Illustration

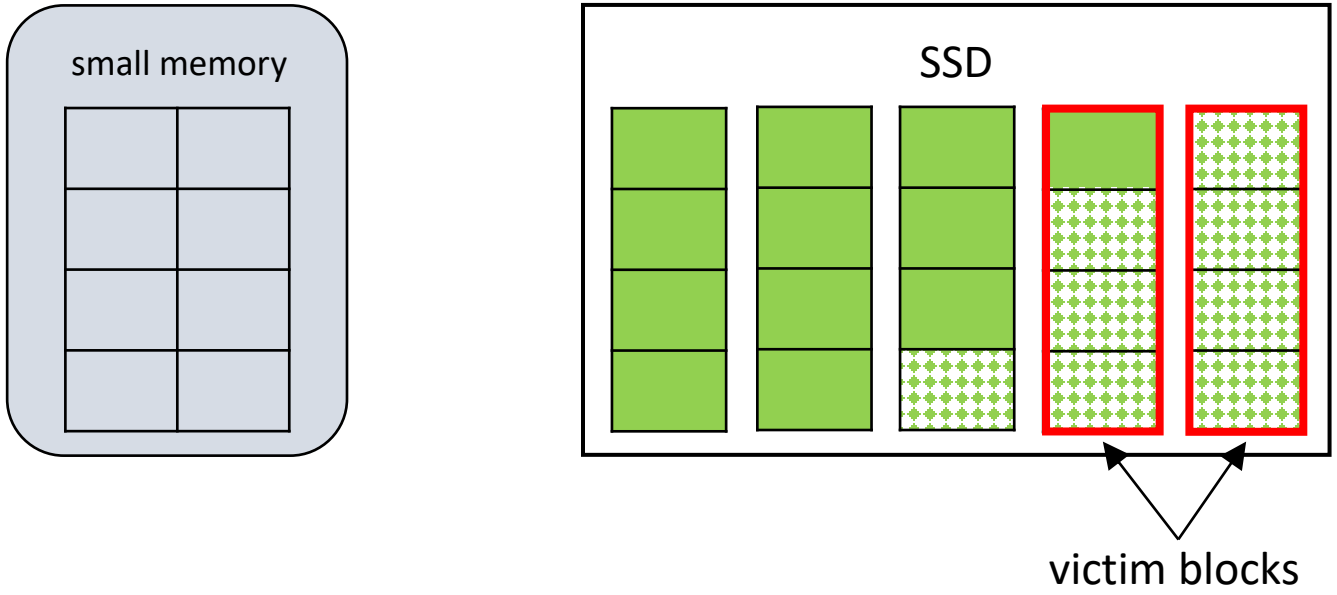


Online Algorithm: Illustration



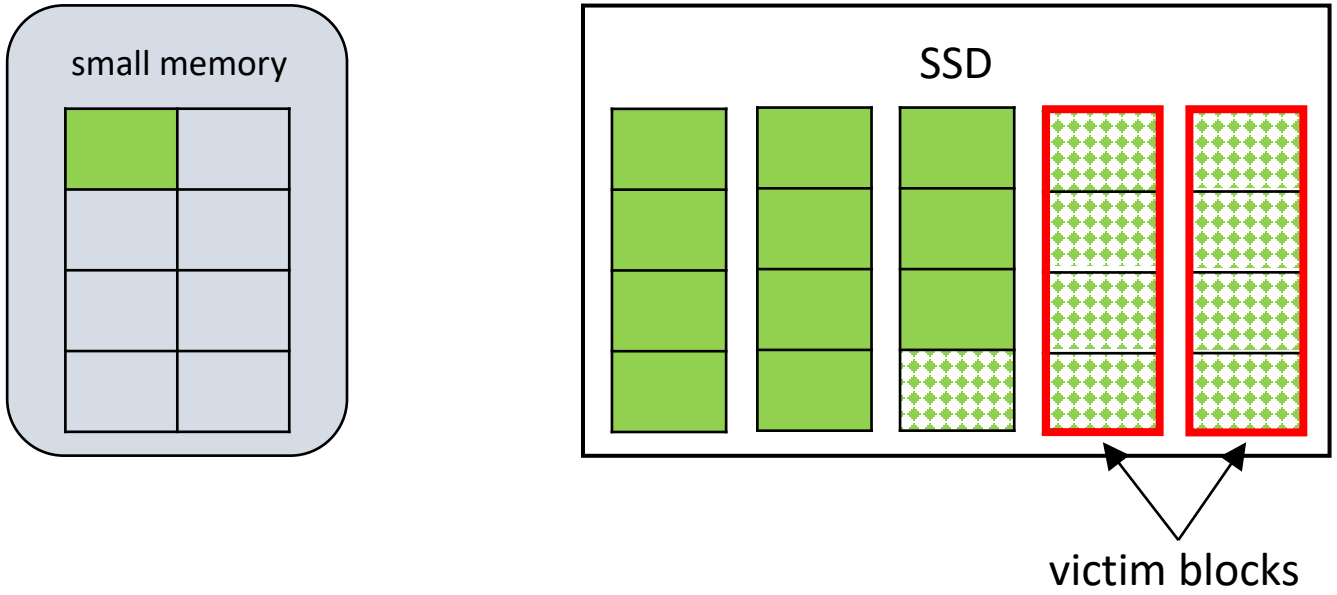
Select min-valid blocks with capacity equal to the memory capacity

Online Algorithm: Illustration



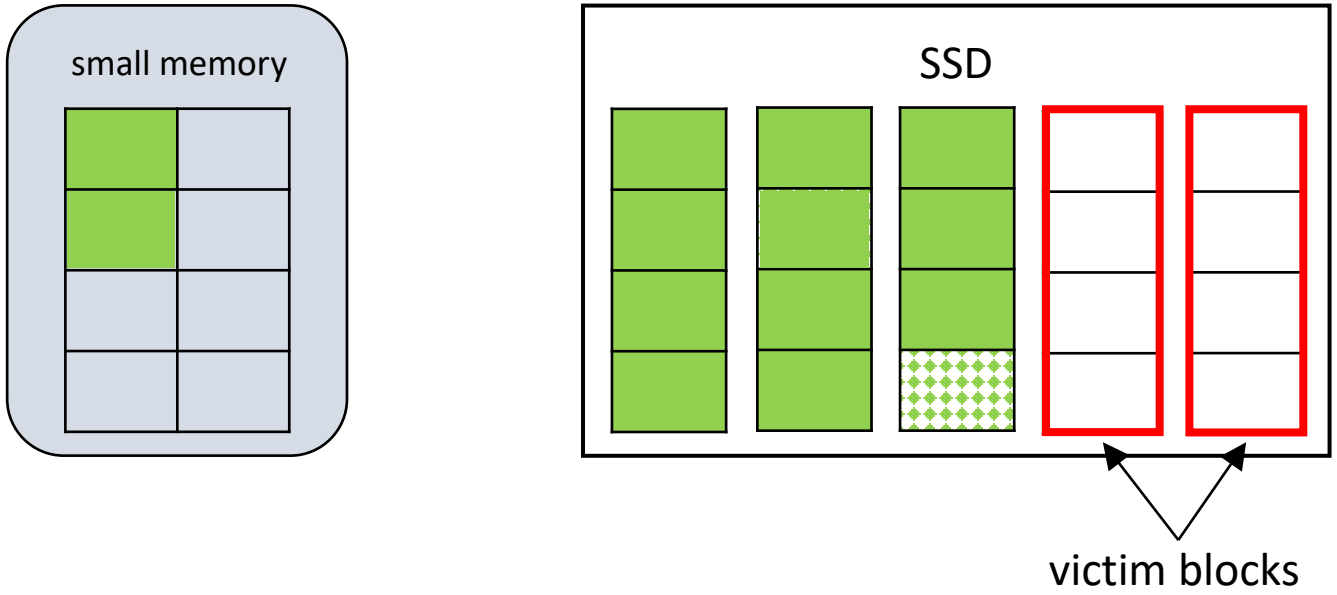
Retrieve valid pages to memory

Online Algorithm: Illustration



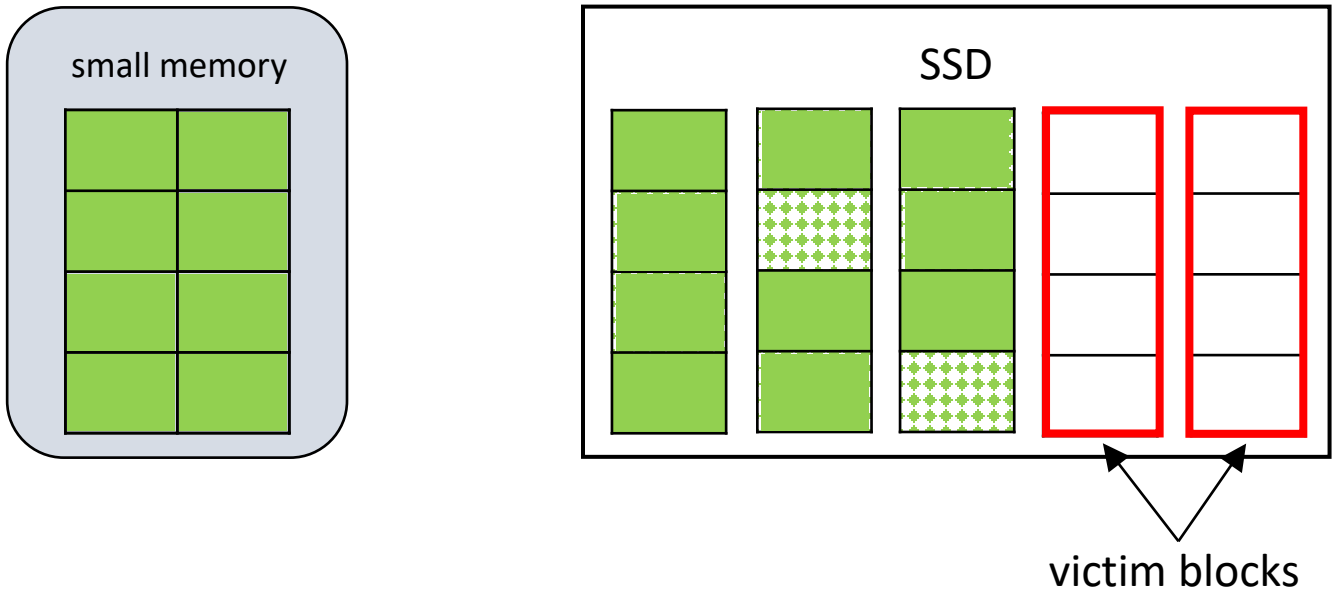
Clean victim blocks

Online Algorithm: Illustration



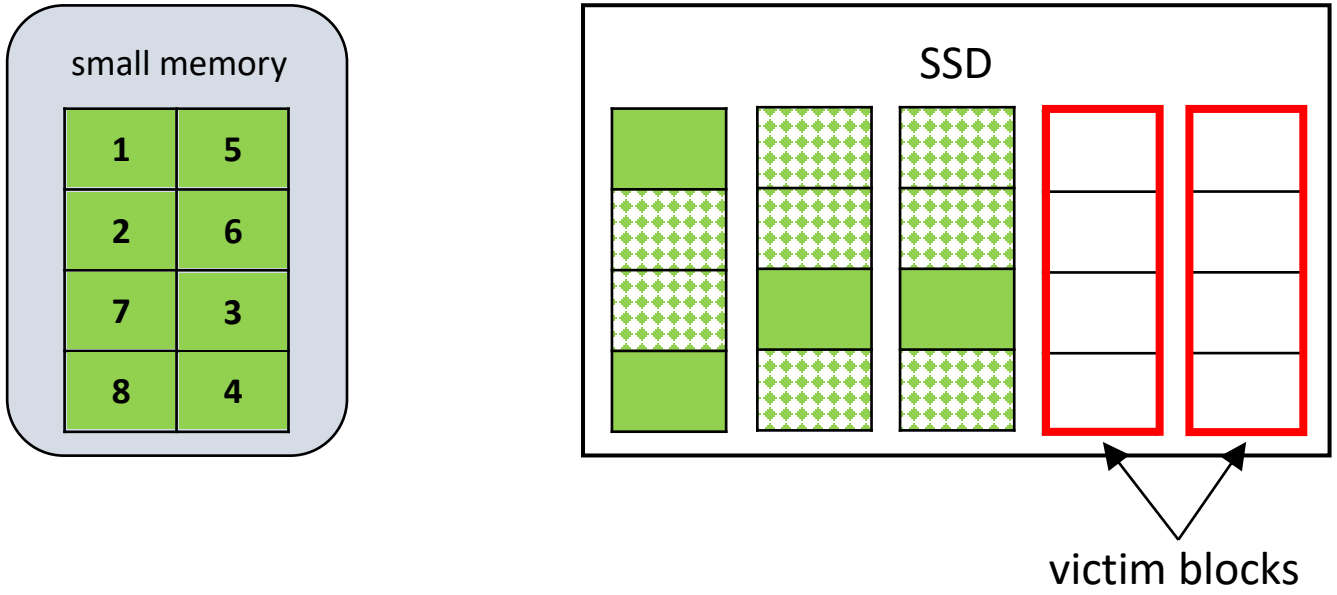
Serve next requests until
memory becomes full

Online Algorithm: Illustration



Serve next requests until
memory becomes full

Online Algorithm: Illustration



Move pages from memory to victim blocks, sorted by death times

Online Algorithm: Performance

Theorem

For any request sequence, $WA_{online}(\eta) \leq \min\left(1 + O\left(\frac{\eta}{T}\right), \frac{1}{\alpha}\right) + O\left(\frac{1}{B_M}\right)$.

avg. error

memory size (in blocks)

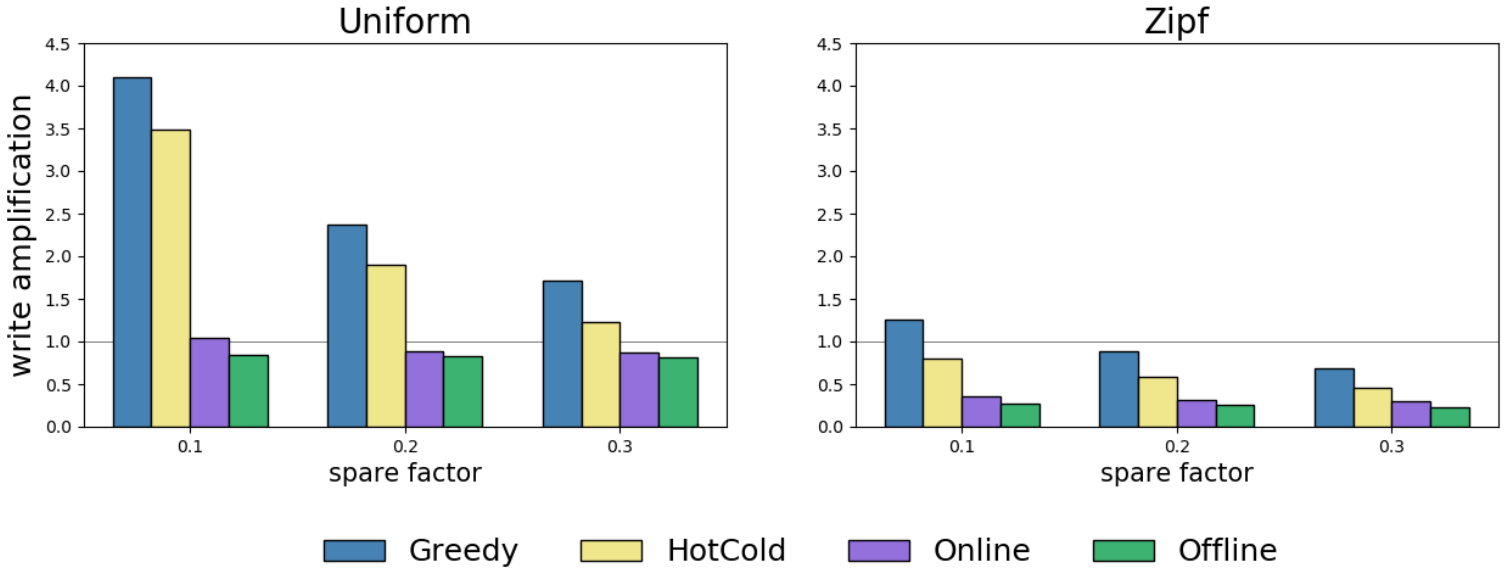
- ✓ Performance improves as error decreases
- ✓ Never worse than Greedy

Simple optimization

- Use memory **also as a cache**
  - ➢ Cache writes “for free”
  - ➢ Note: WA may be smaller than 1

Experimental Evaluation (SSD simulator)

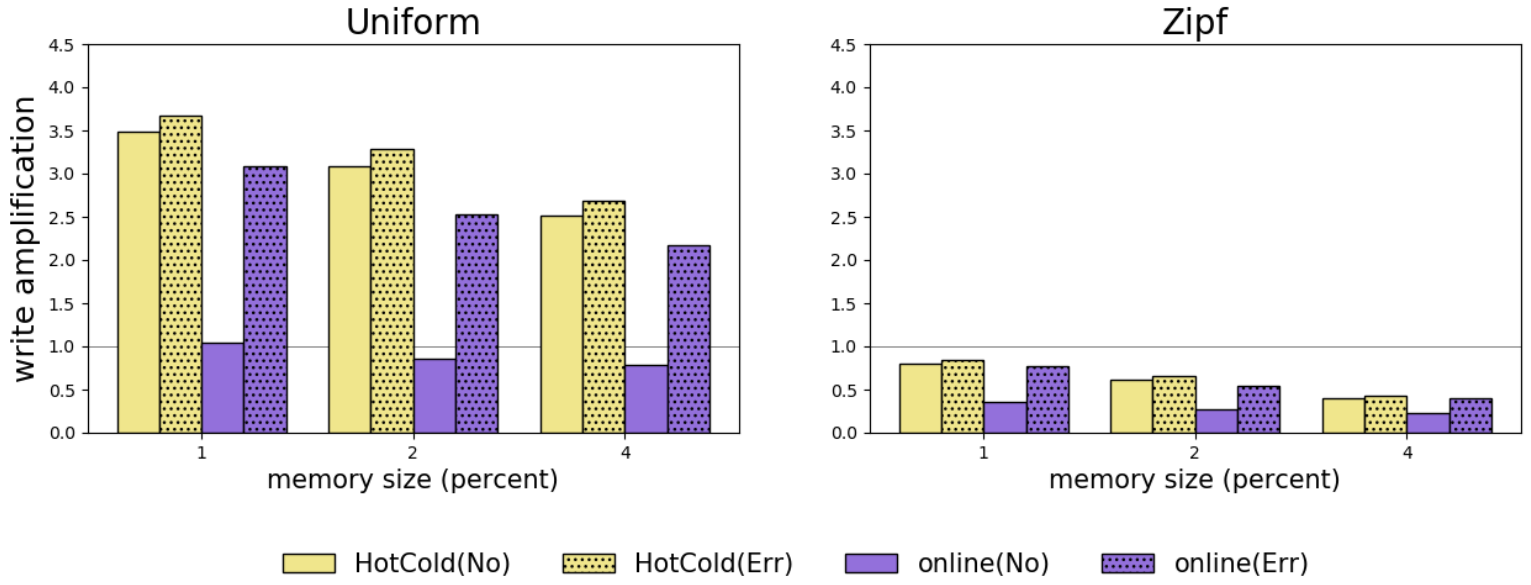
- Memory size: 1% of SSD
  - ➢ Greedy and Hot/cold: use memory as a cache (LRU)
- Exact predictions ($\eta = 0$)
  - ➢ Hot/cold: use predictions to determine temperature



Experimental Evaluation (SSD simulator)

real death time

- Exact and erroneous predictions
  - ➢ Error: predicted death time is distributed uniformly in $[t, y(t)]$
- Spare factor = 0.1



See full version paper for additional results (including real traces)

Conclusion

- Theoretical analysis of existing and new algorithms
- New algorithms that group pages by death times
  - ✓ Offline algorithm
  - ✓ Online algorithm with predictions
- Start “bridging the gap” between systems and theory communities

Thank You