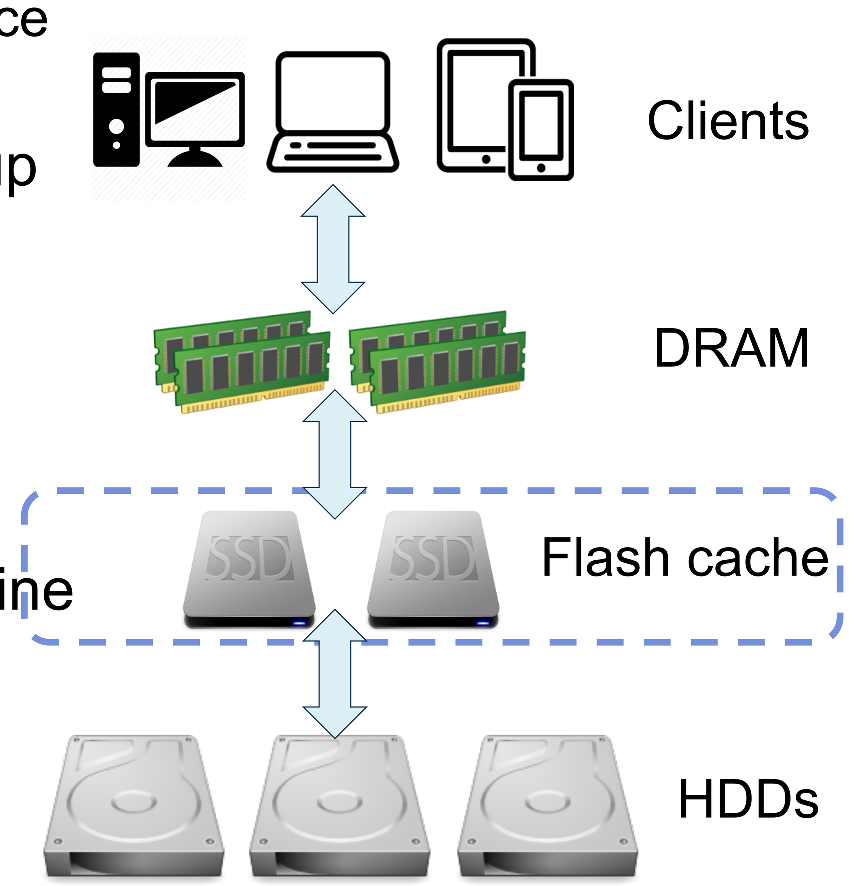
Erasing Belady's Limitations: In Search of Flash Cache Offline Optimality

Yue Cheng[1,2], Fred Douglis[2], Philip Shilane[2], Michael Trachtman[2], Grant Wallace[2], Peter Desnoyers[3], Kai Li[4]
[1]Virginia Tech, [2]EMC Corporation, [3]Northeastern University, [4]Princeton University

Motivation

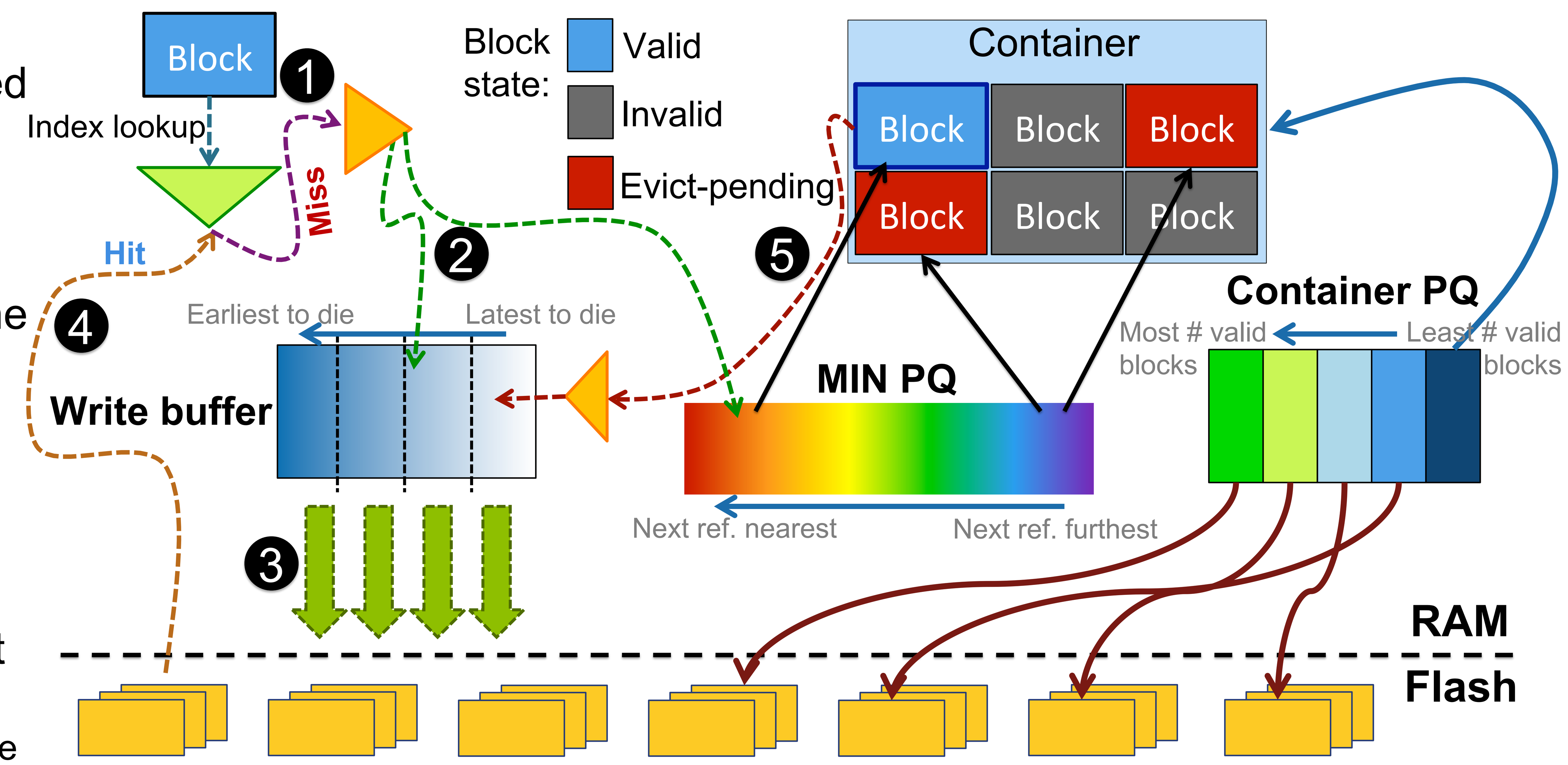
- Using SSDs for an HDD cache
  - SSD cache btw. DRAM & HDD
    - Goal: to balance performance against endurance
  - Nitro [ATC'14], CacheDedup [FAST'16]
  - RIPQ [FAST'15]
  - Pannier [Middleware'15]
- Are we doing well?
  - Comparison against an offline **"best case"**
  - But what is the offline optimal for flash cache?



Container-optimized Heuristic

① On a **read miss**, check if the block will be read before evicted
② Pack a block into in-RAM write buffer and insert it to MIN PQ*
③ Disperse the write buffer into containers and write them to the flash cache
④ On a **read hit**, read the block from the flash cache
⑤ At garbage collection, copy-forward (CF) the valid block to the write buffer by checking if it will be reread before evicted

*PQ=Priority queue



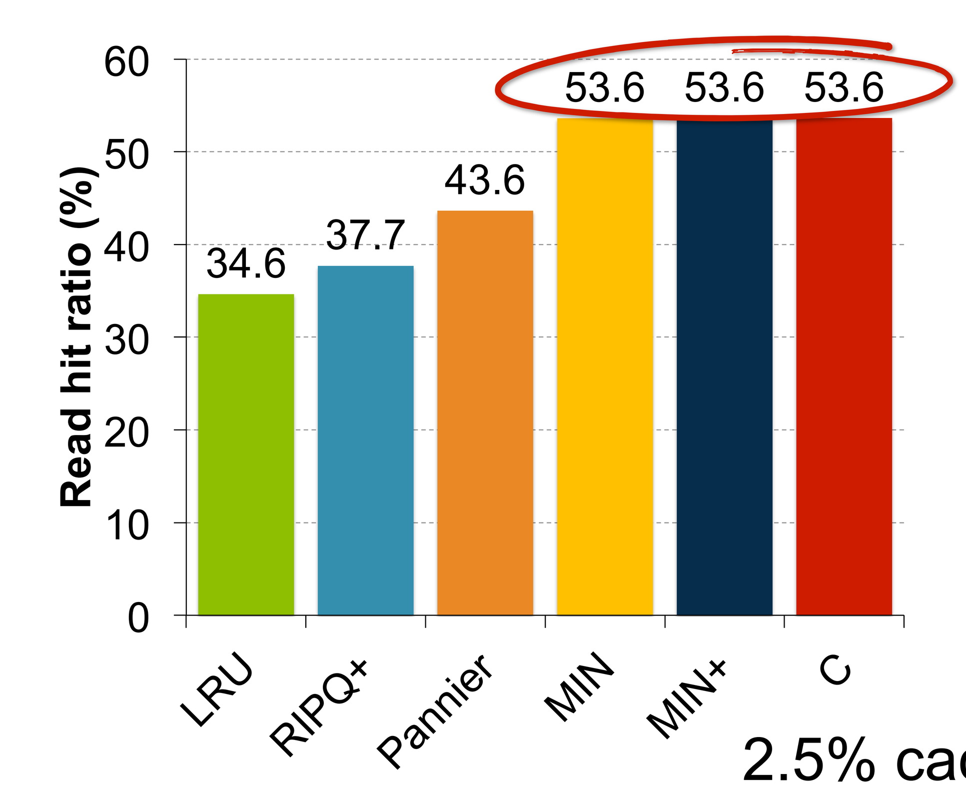
Background: Offline Optimality

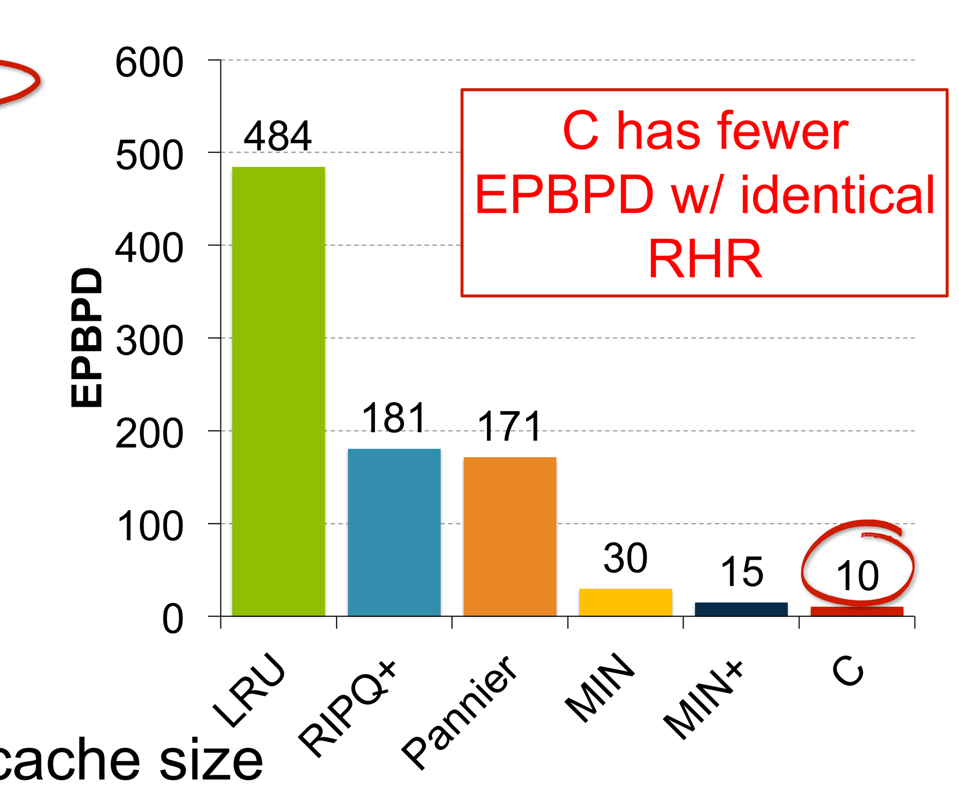
- Belady's MIN: A simple offline caching alg. **when the next access is known**
  - Evicts the item that will be used furthest in the future
  - Yields the optimal read hit ratio (RHR)
  - Ignorant of minimizing erasures/block/day (EPBPD)
- MIN is not able to provide the optimal erasures in the context of flash caching
  - MIN inserts items that **won't actually be read**

Comparing Algorithms

Policy	Description	O*	C⁺
LRU	Least recently used	✗	✗
RIPQ+	Static web content	✗	✗
Pannier	Handles divergent containers	✗	✓
MIN	Don't insert data w/ furthest next ref	✓	✗
MIN+	Don't insert data evicted w/o read	✓	✓
C	Our container-optimized heuristic	✓	✓

*O=Offline, ⁺C=Container-optimized





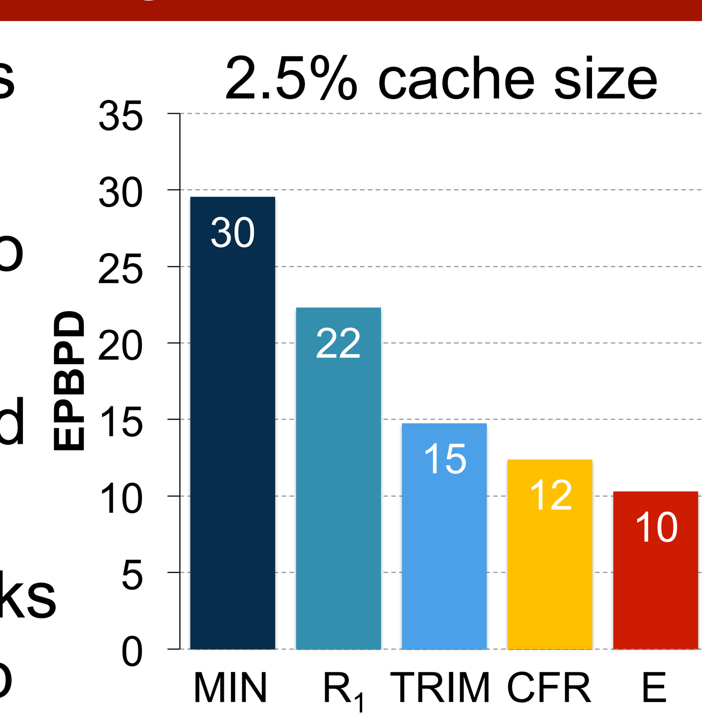
2.5% cache size

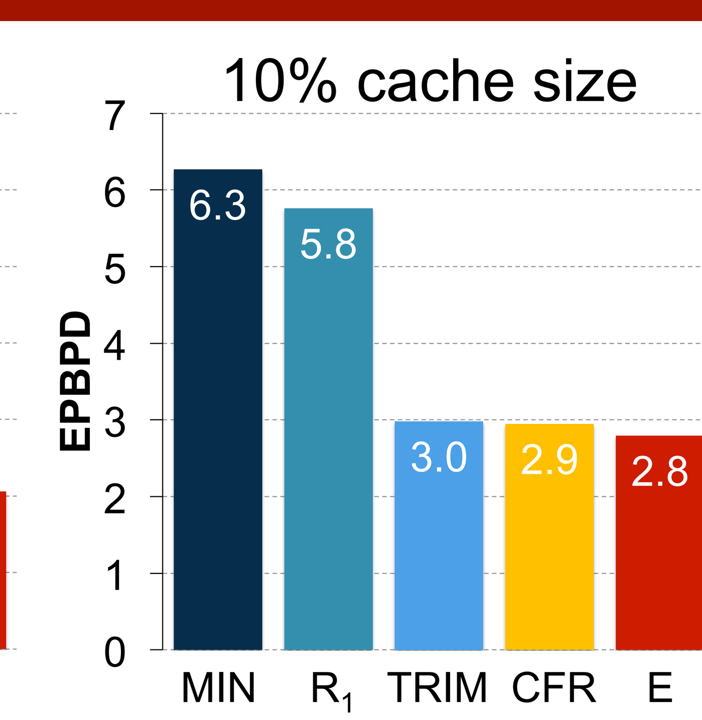
Offline Flash Caching

- Objectives
  - Minimize erasures s.t. maximal RHR
    - Never insert items if it does not increase RHR
  - Other objectives possible as discussed in the paper
- True optimal vs. Heuristic
  - Complexity of true optimal?
  - **Approximation** is the focus of this work

Evaluating Optimization Techniques

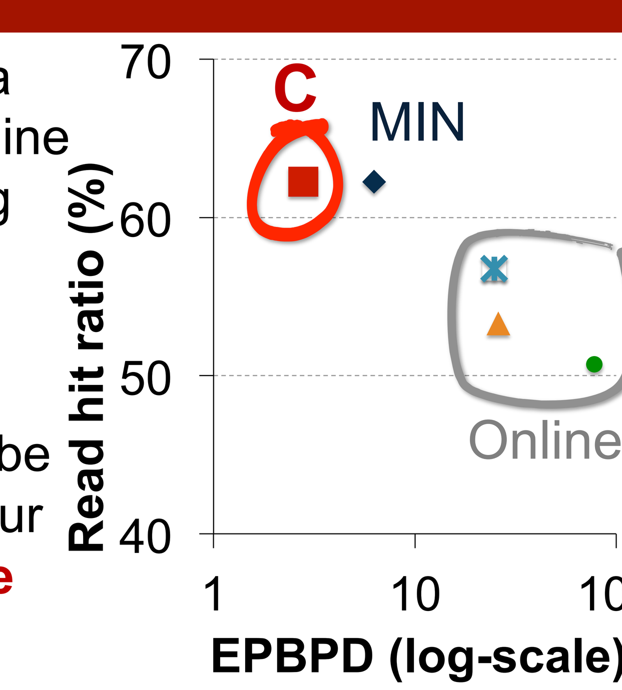
- **R_1**: Omit insertions w/ no reread
- **TRIM**: Notify GC to omit dead blocks
- **CFR**: Avoid wasted CF blocks
- **E**: Segregate blocks by evict timestamp





Conclusion

- Important to have a baseline for the offline optimal considering **both RHR and endurance**
- Additional optimizations may be possible to move our heuristic to the **true** optimal



VirginiaTech EMC² Northeastern PRINCETON UNIVERSITY