CRAID
Online RAID Upgrades Using Dynamic **Hot Data** Reorganization

Alberto Miranda
Toni Cortes

Barcelona Supercomputing Center
Technical University of Catalonia

Introduction

- **RAID**: building block for large-scale storage
  - Performance, reliability and capacity
  - Acceptable cost
- Client requirements ↑ with time
- **Solution: add disks to upgrade RAID**
  - Significant data migration

Upgrade challenges

- Uniformity of distribution
    - B blocks into n disks → B/n per disk
- Minimal (ideal) data migration (IM)
    - m additional disks → $B \cdot m / (n+m)$ blocks to migrate
- Online rebalancing
    - 24/7 services

Motivation

- Existing methods with ideal migration
  - Uniformity ↓ after several upgrades [Semi-RR, Goel02]
  - Limited performance scalability [GSR, Chang07]
  - No support for parities yet [FastScale, Zheng11]
- Rebalancing the ideal amount of data can be too **expensive** in some situations!

Motivation

- Existing methods with ideal migration
  - Uniformity ↓ after several upgrades [Semi-RR, Goel02]
  - Limited performance scalability [GSR, Chang07]
  - No support for parities yet [FastScale, Zheng11]
- Rebalancing the ideal amount of data can be too **expensive** in some situations!

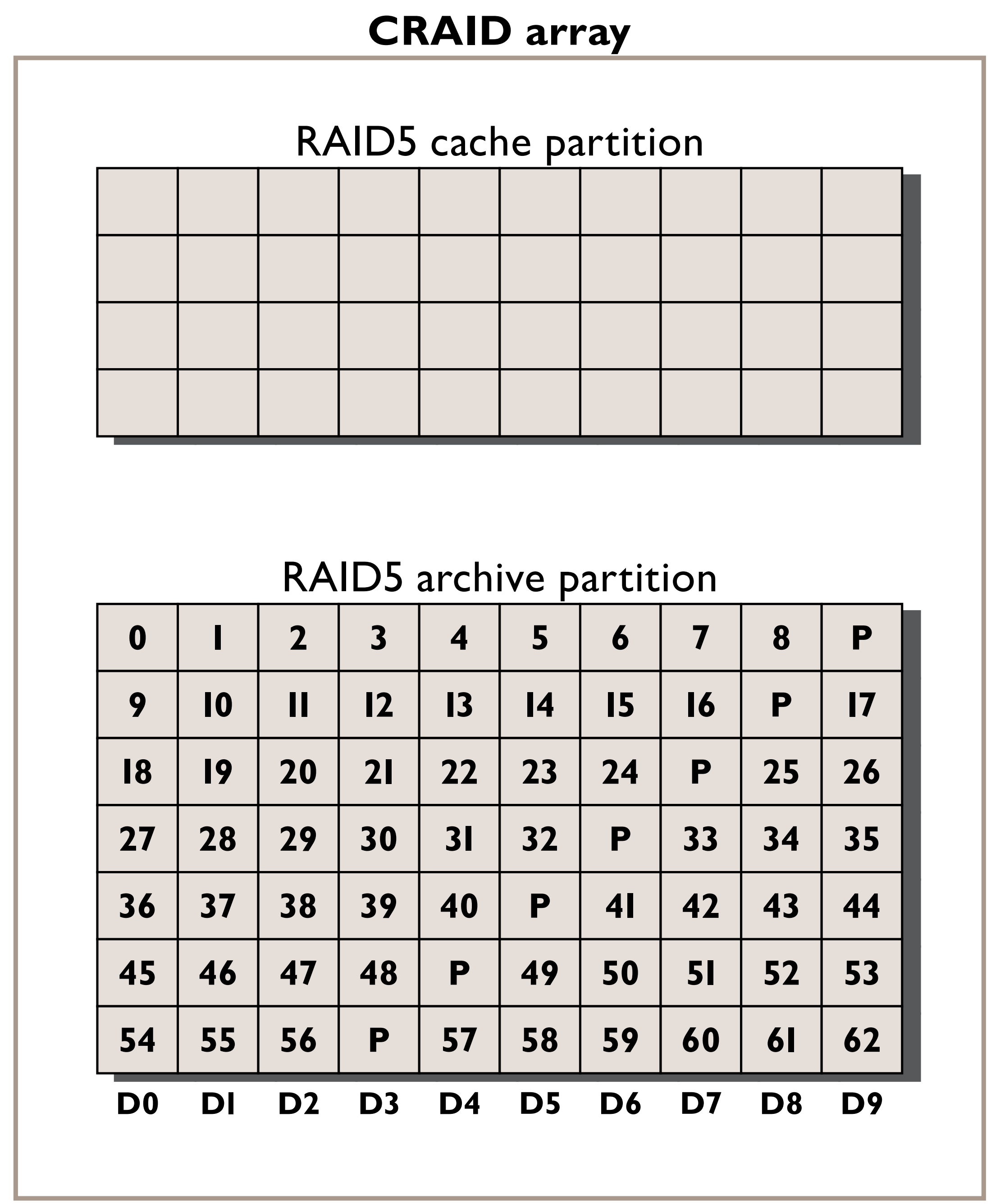
Can we migrate less than IM and still keep RAID's benefits?

Motivation

- Do we really need ALL DATA ideally distributed?
- Not all data is created equal
  - Frequently accessed data is a small % of total
  - Long-term locality
  - Really popular data can receive ↑90% accesses
- Rebalance only **hot data** in real time
  - Ideal RAID distribution of data actually in use
  - Upgrade overhead ↓ (hot migration < ideal migration)

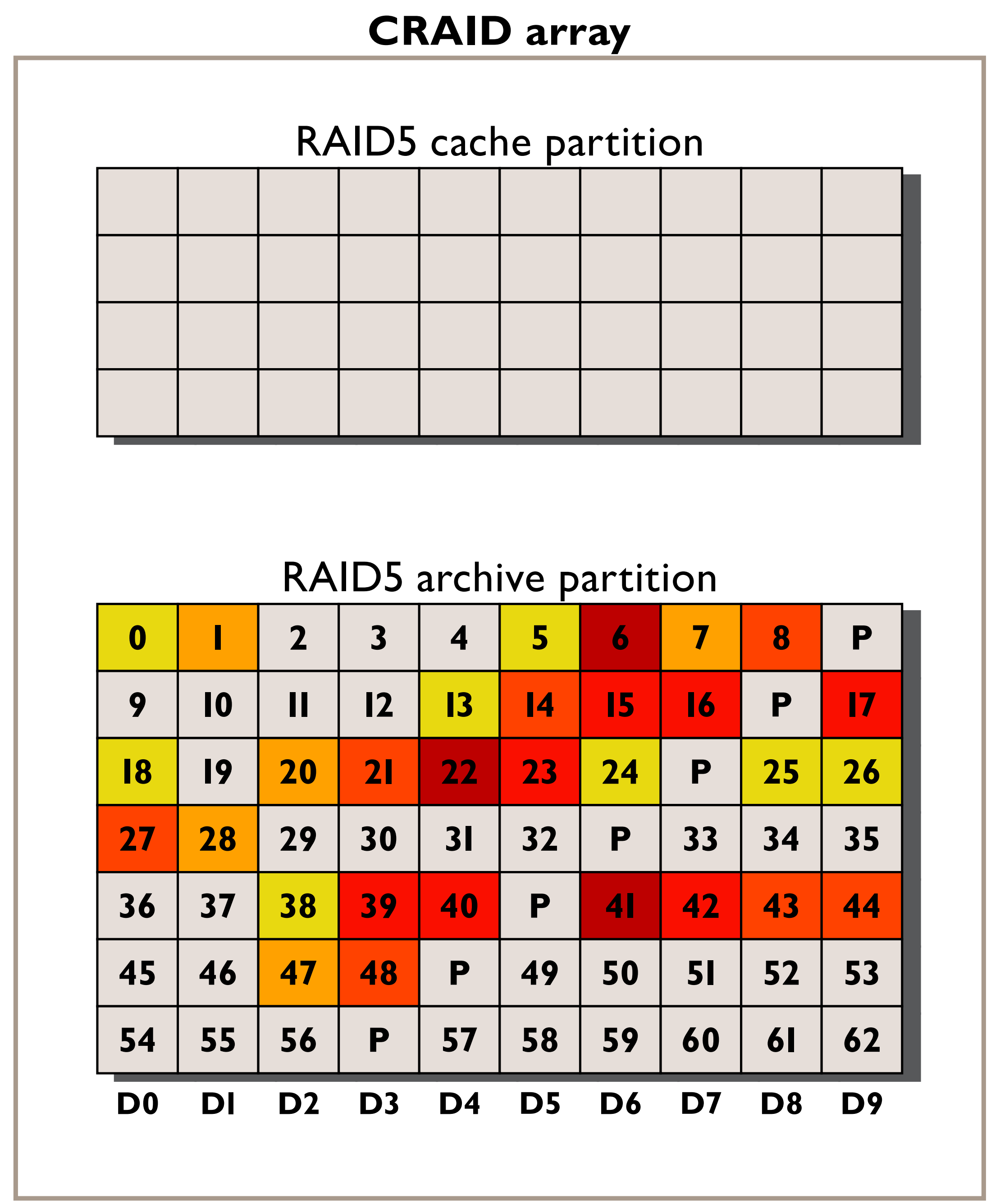
CRAID: Overview

- A **cache partition (CP)** is built from a % of all disks
- Original data remains in an **archive partition (AP)**



CRAID: Overview

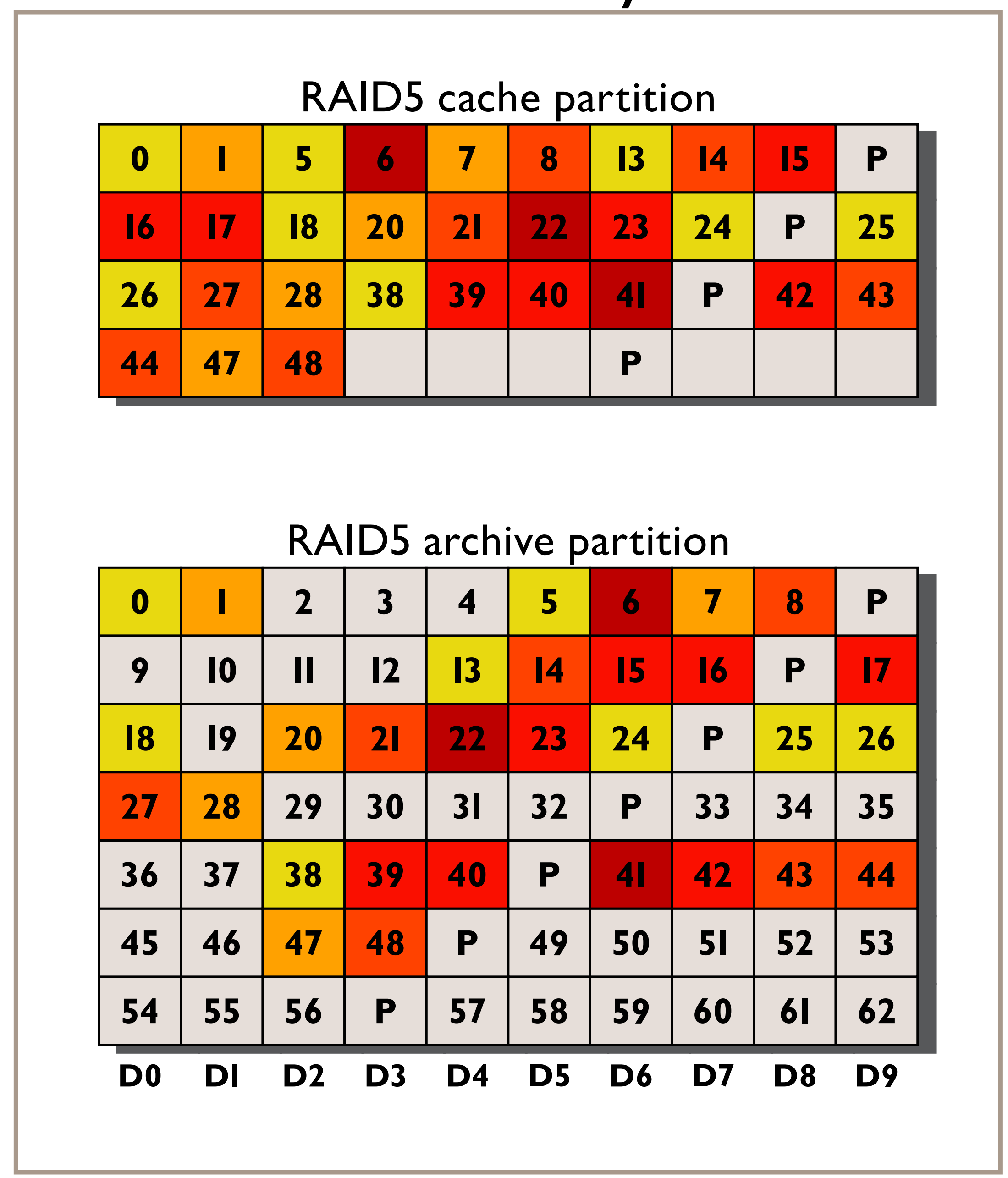
- A **cache partition (CP)** is built from a % of all disks
- Original data remains in an **archive partition (AP)**
- Hot data is **tracked** …



CRAID: Overview

- A **cache partition (CP)** is built from a % of all disks
- Original data remains in an **archive partition (AP)**
- Hot data is **tracked** …
- … and **copied** to CP
- When CP is full:
  - **replace** w/ hotter data
  - **update original** (if needed)

CRAID array

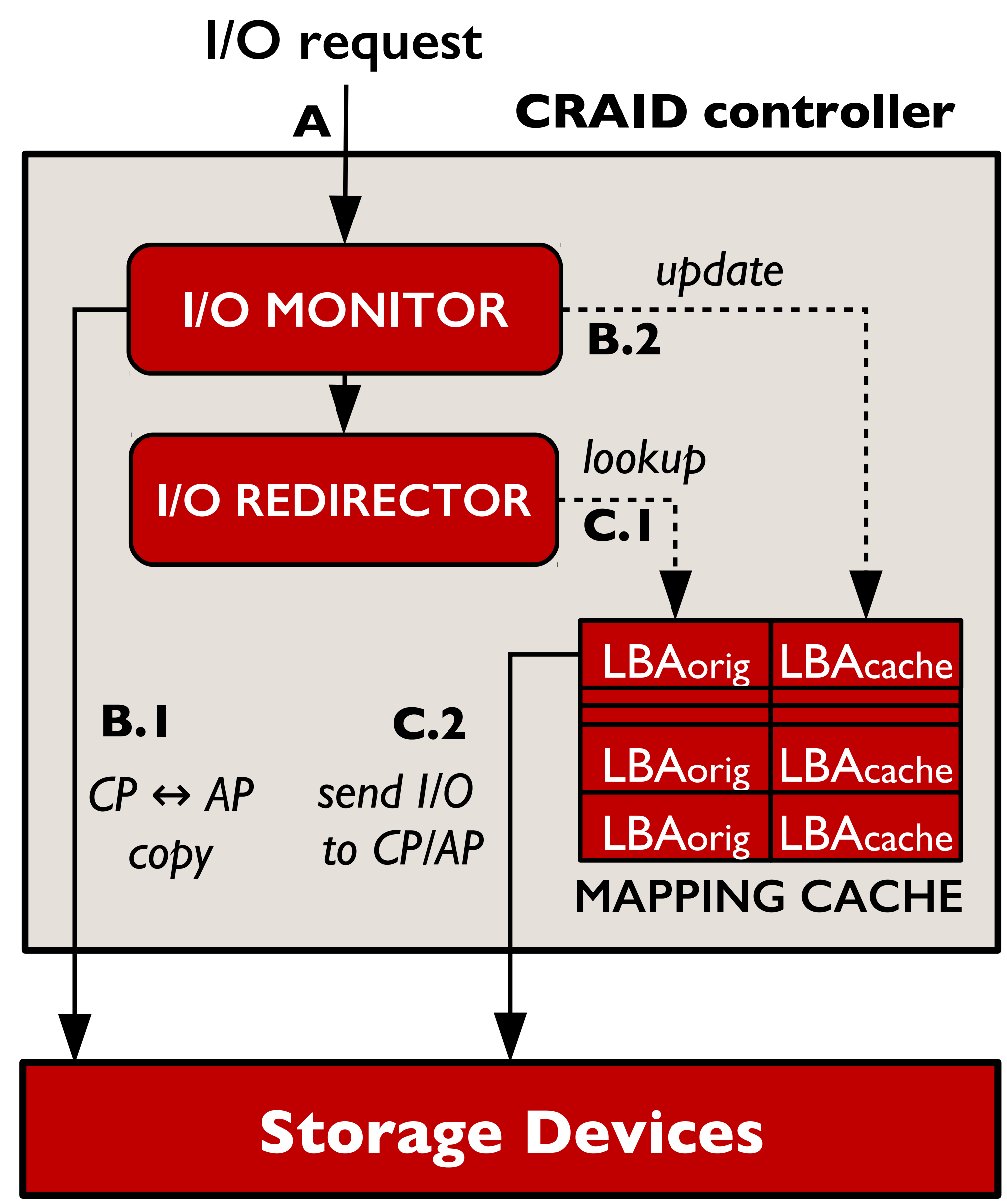


CRAID: Advantages

- Large cache with small % per disk
  - Important data is resident longer
- Disk-based cache → persistent optimizations
- Clustering of hot data → sequentialization of access
- Benefits gained with in-place devices

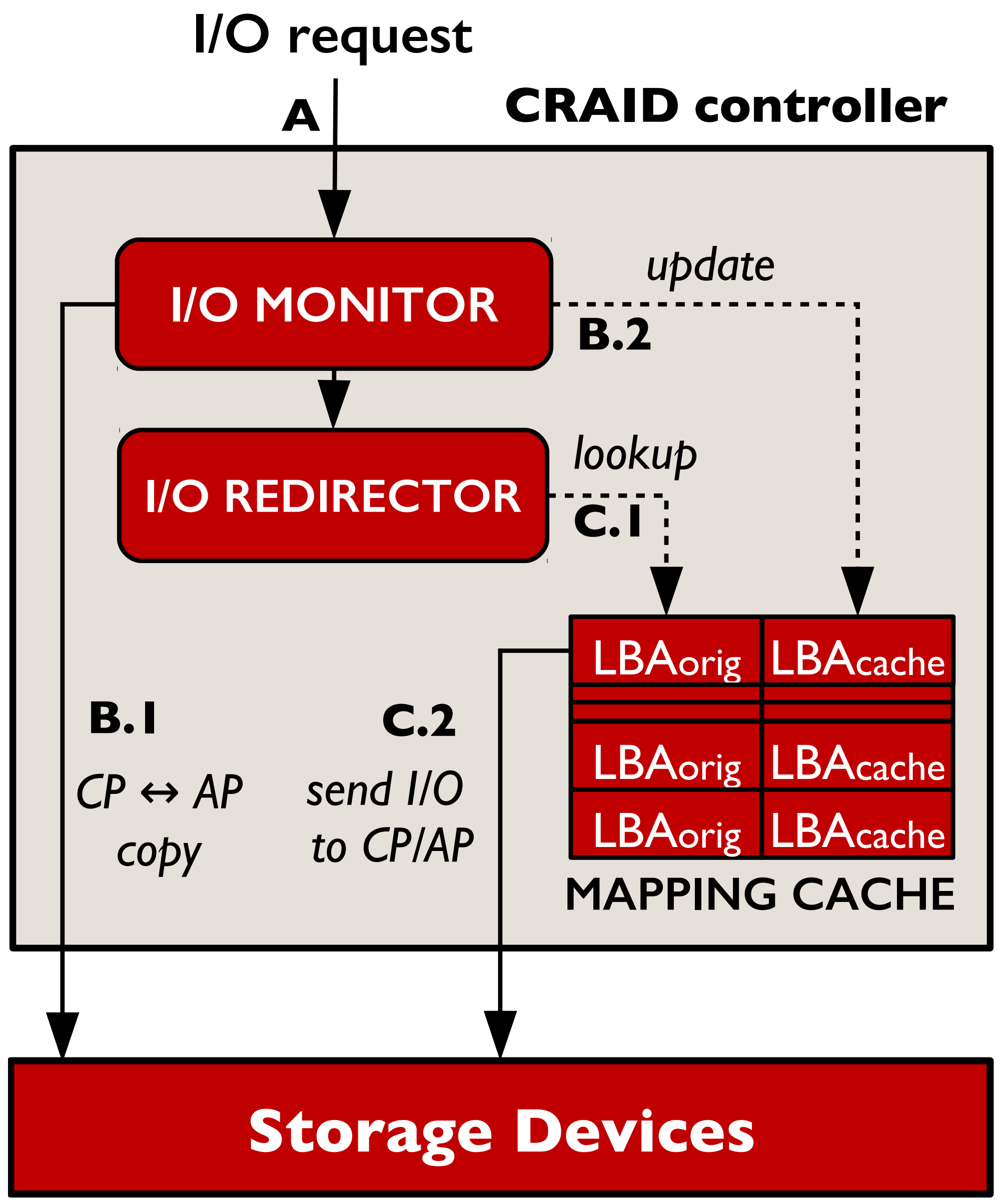
CRAID: Architecture

- I/O Monitor
- Mapping Cache
- I/O Redirector



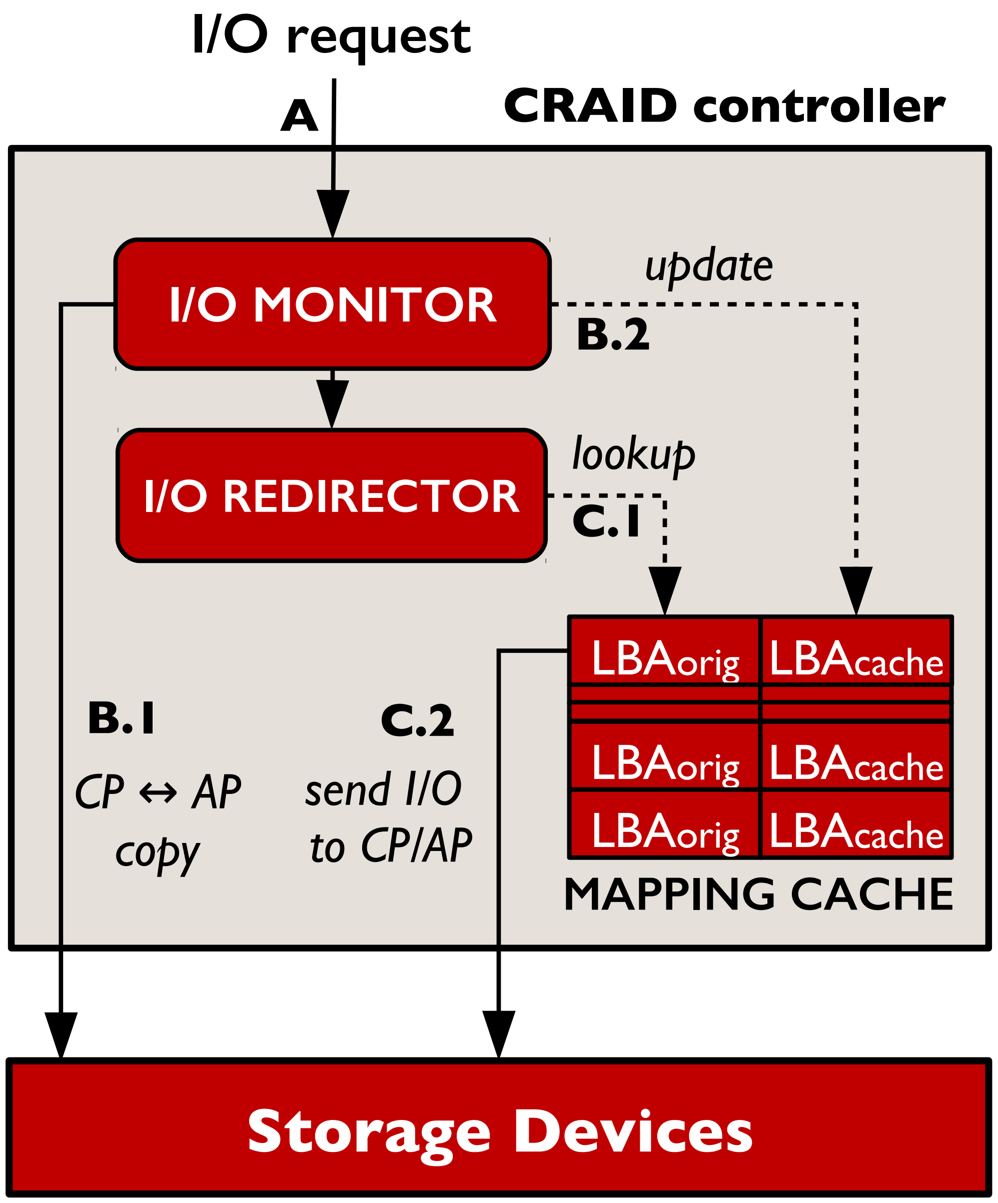
I/O Monitor

- Analyzes REQS to identify hot data
- Schedules I/Os to/from partitions
- Chooses which data to replace
  - LRU, LFUDA, GDSF, ARC, WLRU
  - Simple, with high success rates



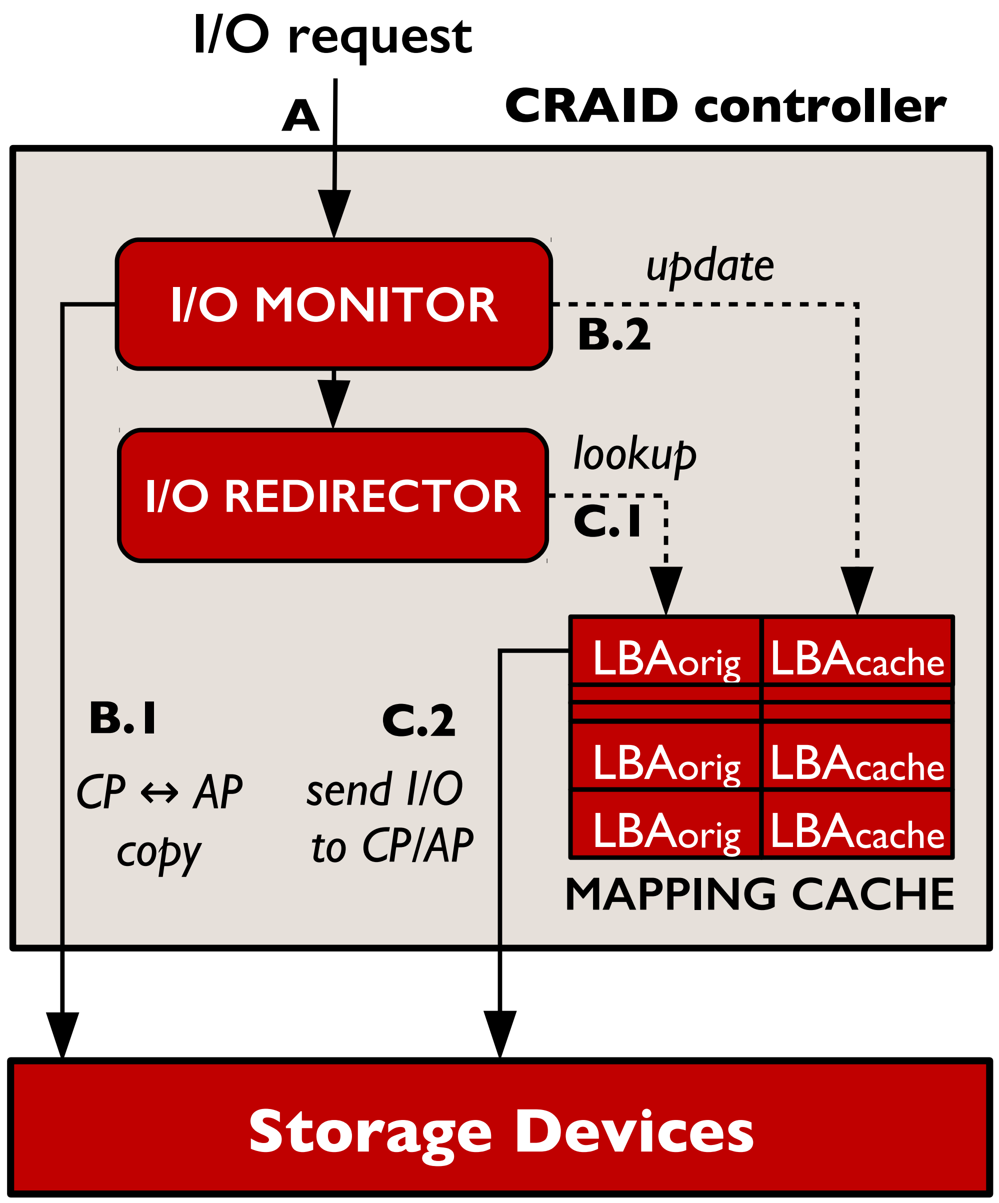
Mapping Cache

- Translates original LBAs to CP LBAs
- Lookup must be **fast**!
  - Tree structure O(log ||CP||)
- Memory usage: ≈ 5.9MB/GB
- Failure resilience: persistent log of dirty blocks and translations



I/O Redirector

- Intercepts I/Os and sends them to appropriate partition
- Lazy updates: writes directly to CP



CRAID: Rebalancing

- Done by I/O Monitor when new devices appear
  - Update originals of dirty blocks
  - Invalidate CP
  - Begin filling w/ currently hot working set
    - ***Gradual, on-line rebalancing!***

CRAID: Rebalancing

- **Pros:**
  - New disks used at T=0
  - Long sequential chains of related blocks
- **Cons:**
  - Invalidation of the CP = work lost

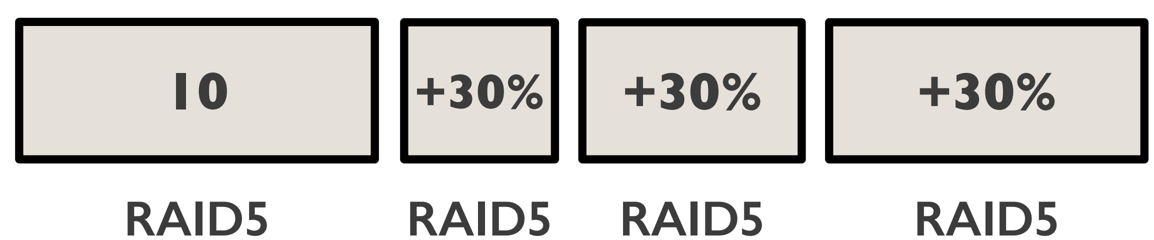
Evaluation

- Trace-based simulation (Disksim)
  - 7 datasets in paper (research, NFS, servers, …)
    - Here 'webusers' (FIU) and 'proj' (MSRC)
  - 1 week trace time
  - 50 disks (+5 SSDs in some cases)
  - 128KB block size
  - WLRU algorithm (best hit/miss % tradeoff)
  - CP cold at start

Evaluation

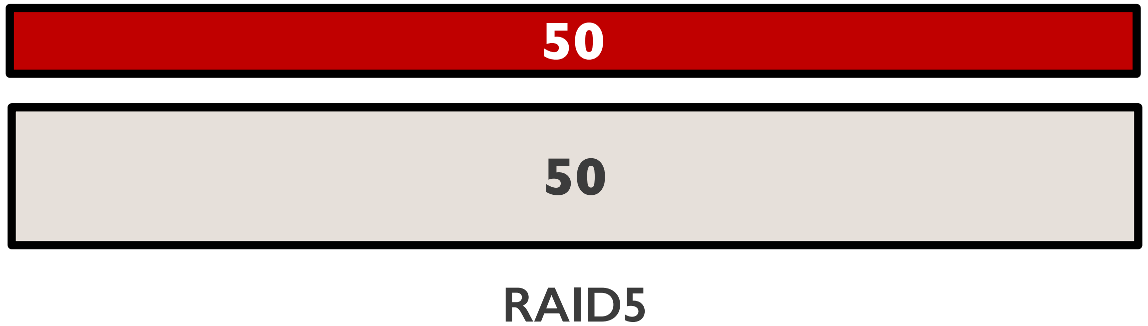
- **RAID5:** ideal, 50 disks, restriped

- **RAID5$^+$:** ideal, 10 base disks + 30% growth, restriped

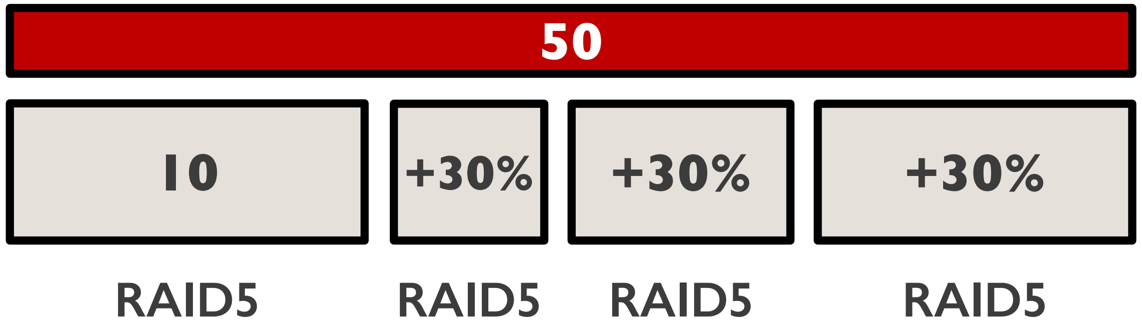


Evaluation

- **CRAID5:** RAID5 (CP) + RAID5 (AP), not restriped

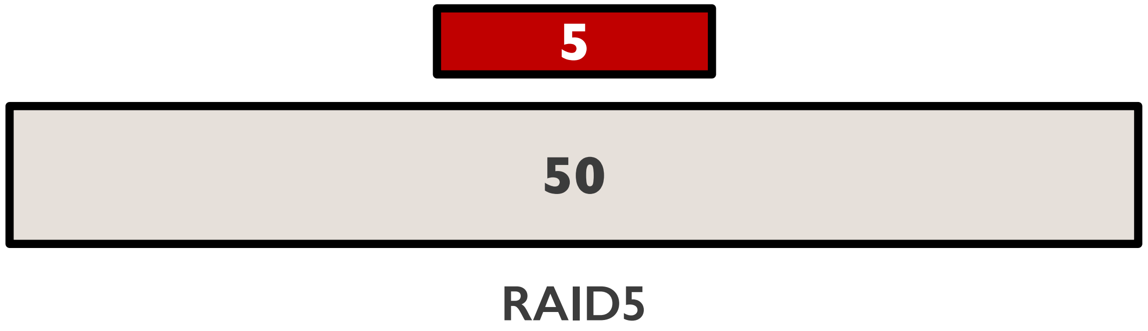


- **CRAID5⁺:** RAID5 (CP) + RAID5⁺ (AP), not restriped

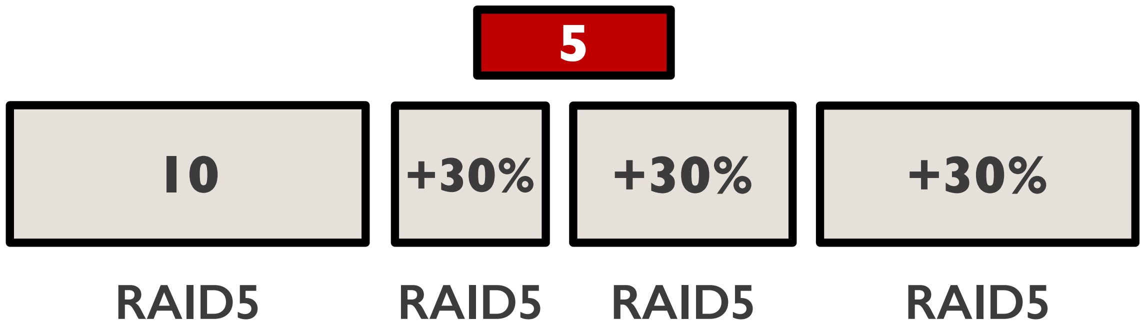


Evaluation

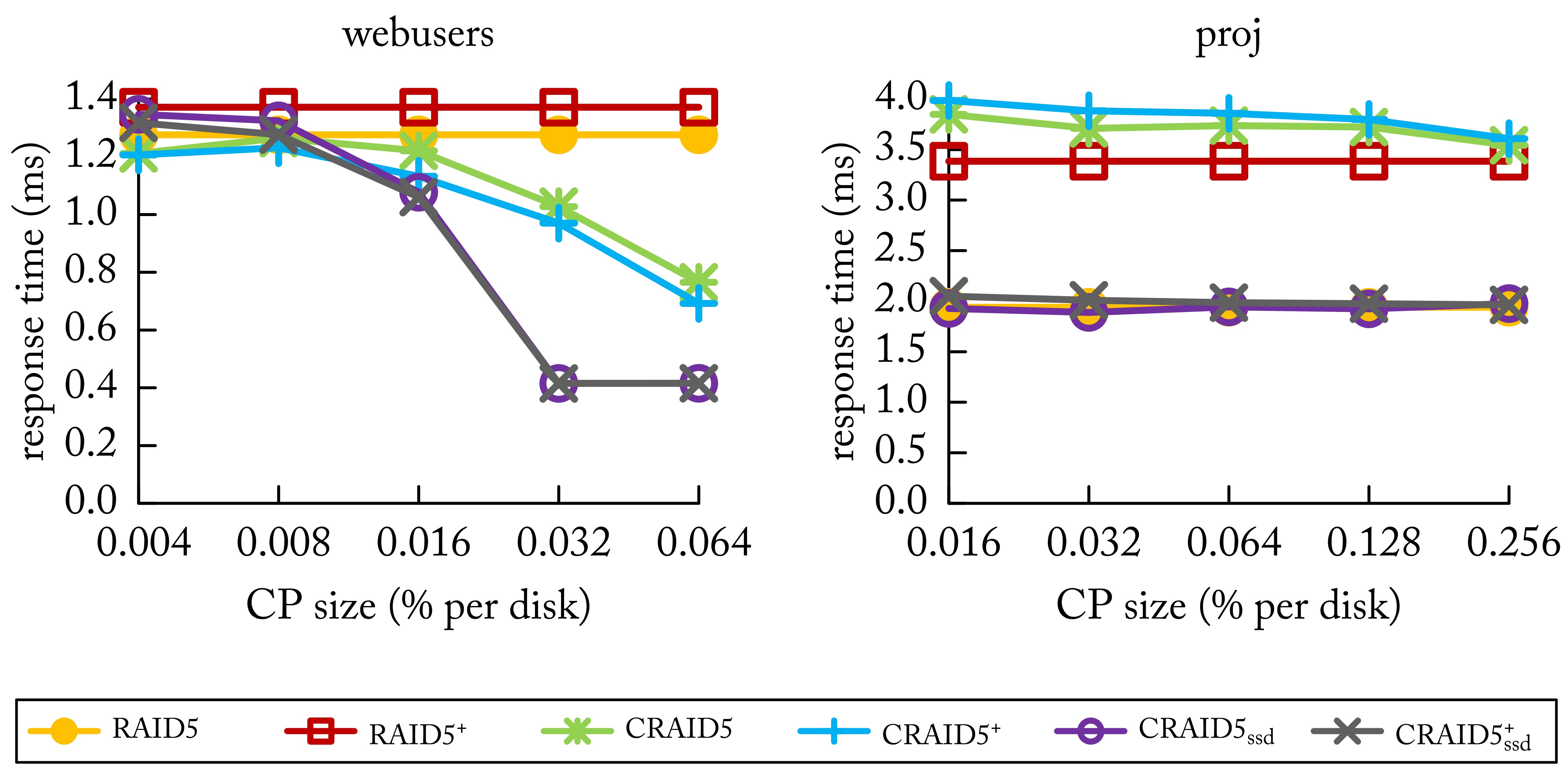
- **CRAID5$_{ssd}$**: SSD dedicated CP, HDD AP (RAID5)



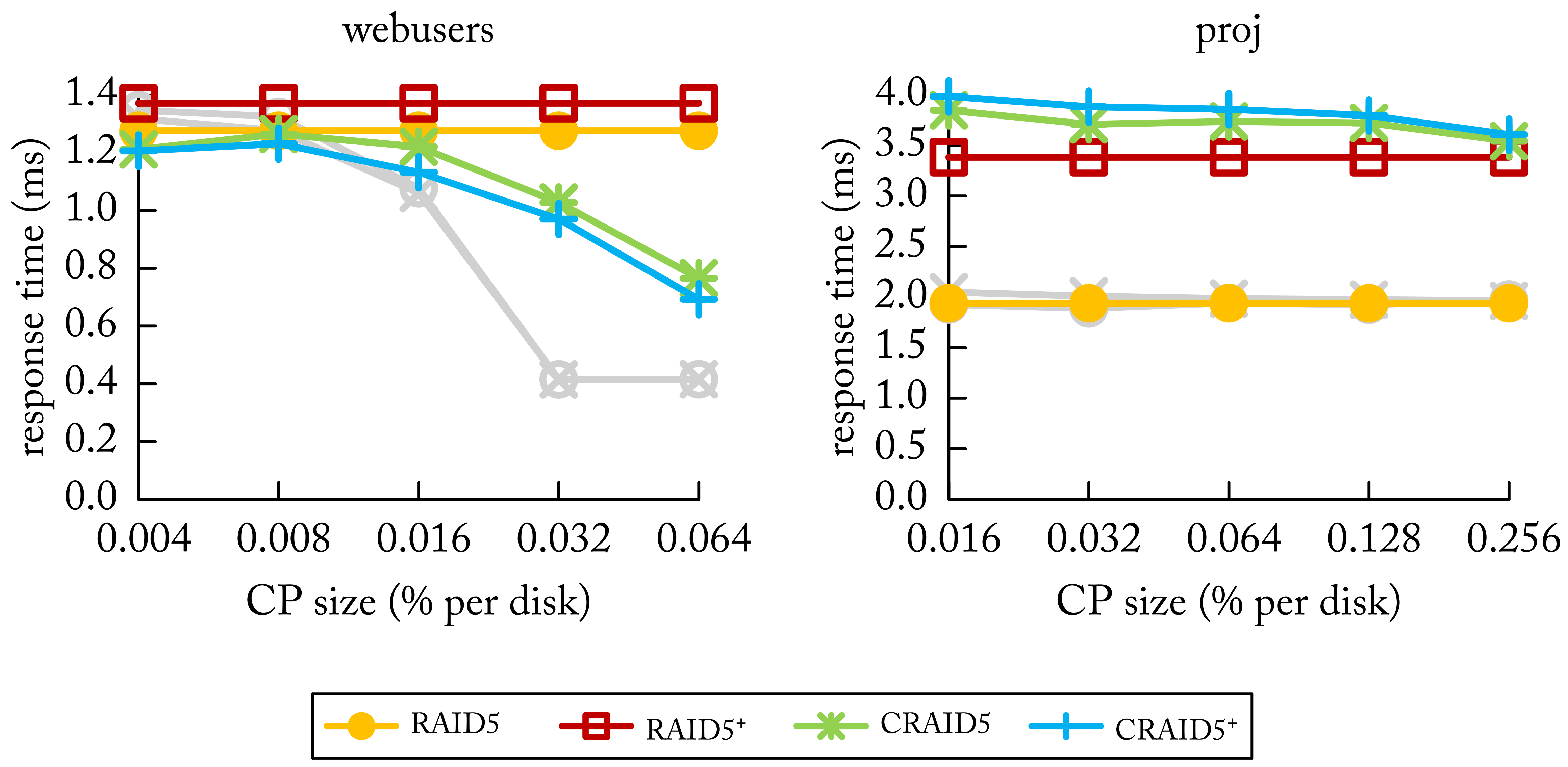
- **CRAID5$^{+}_{ssd}$**: SSD dedicated CP, HDD AP (RAID5^{+})



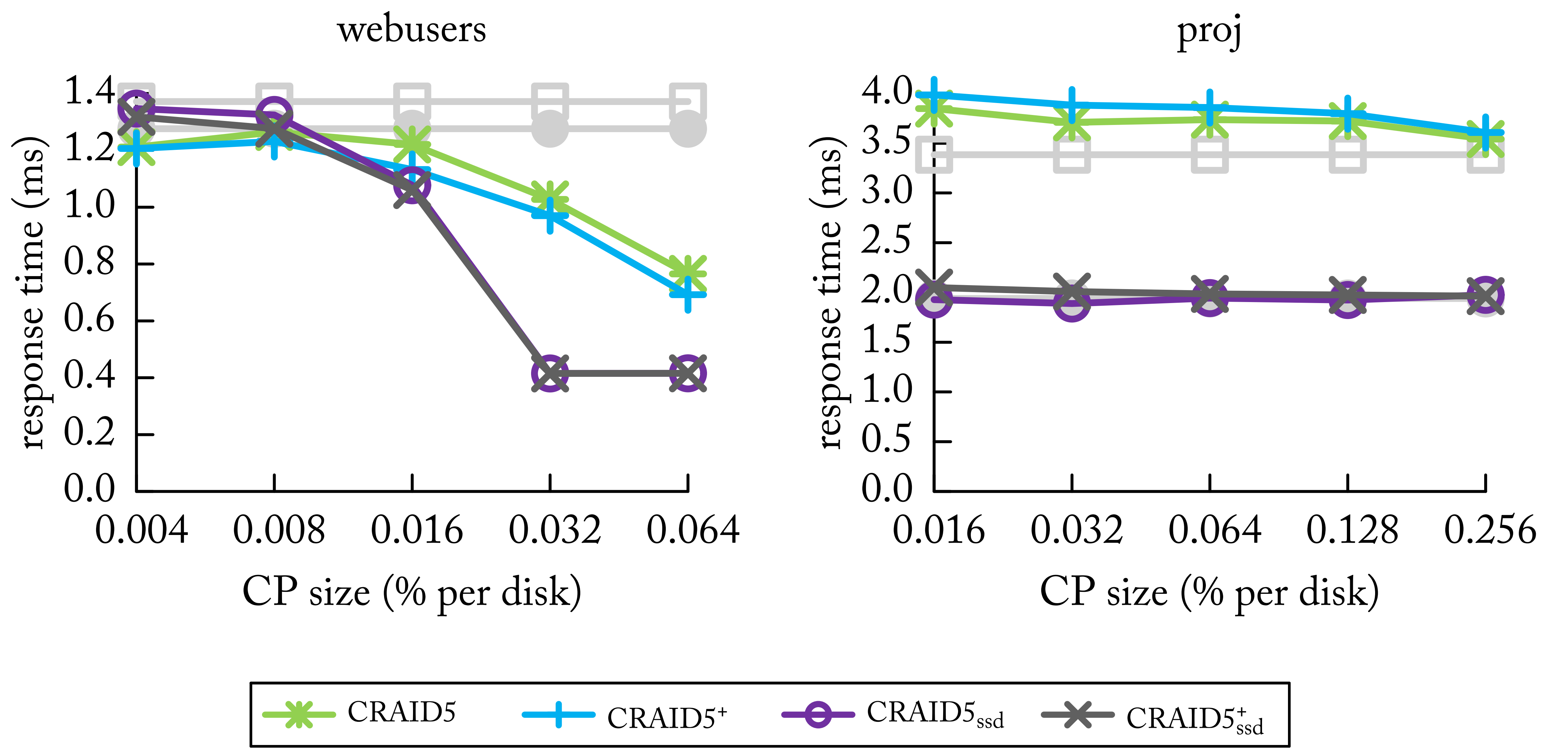
Response time (reads)



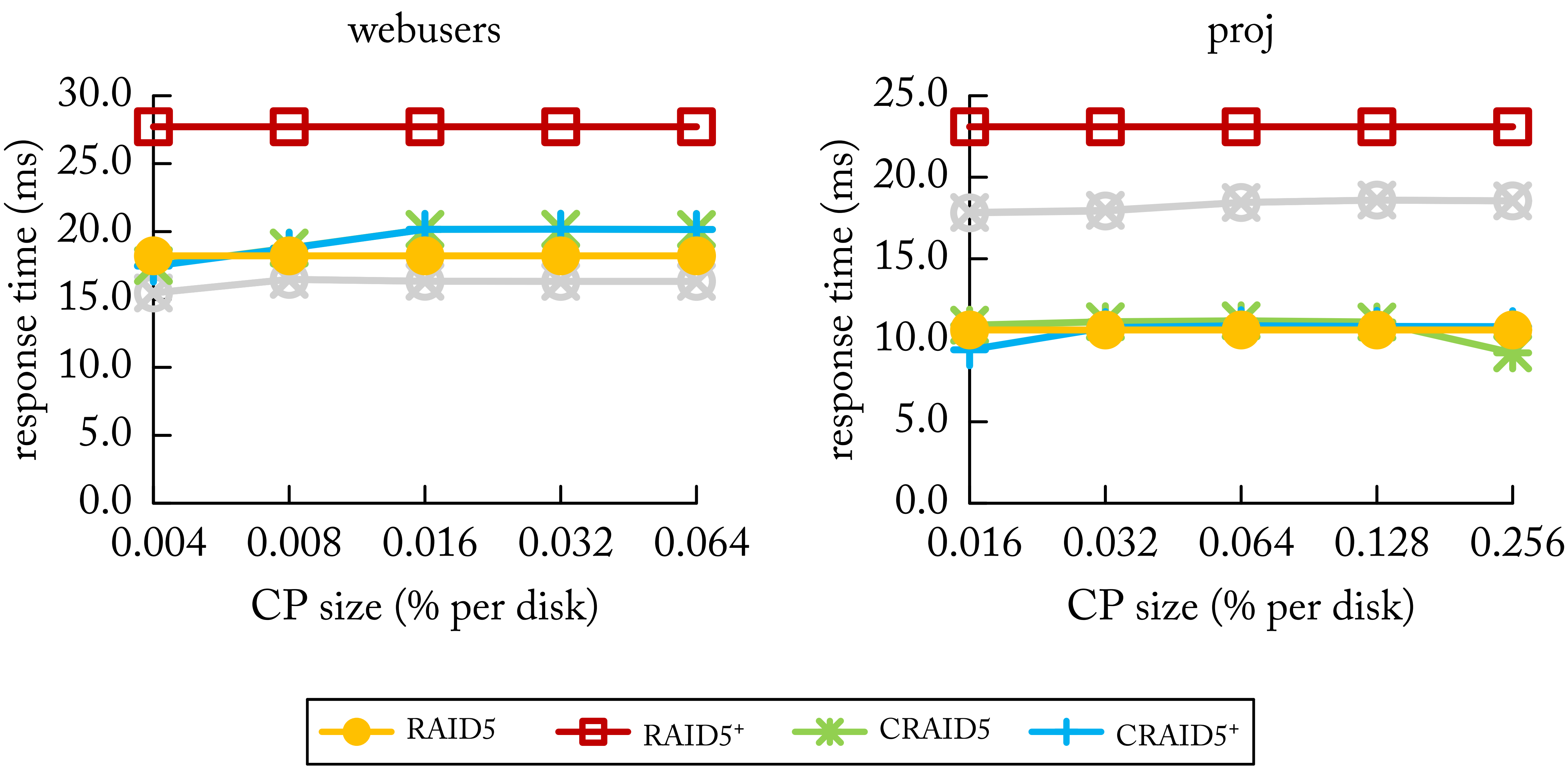
Response time (reads)



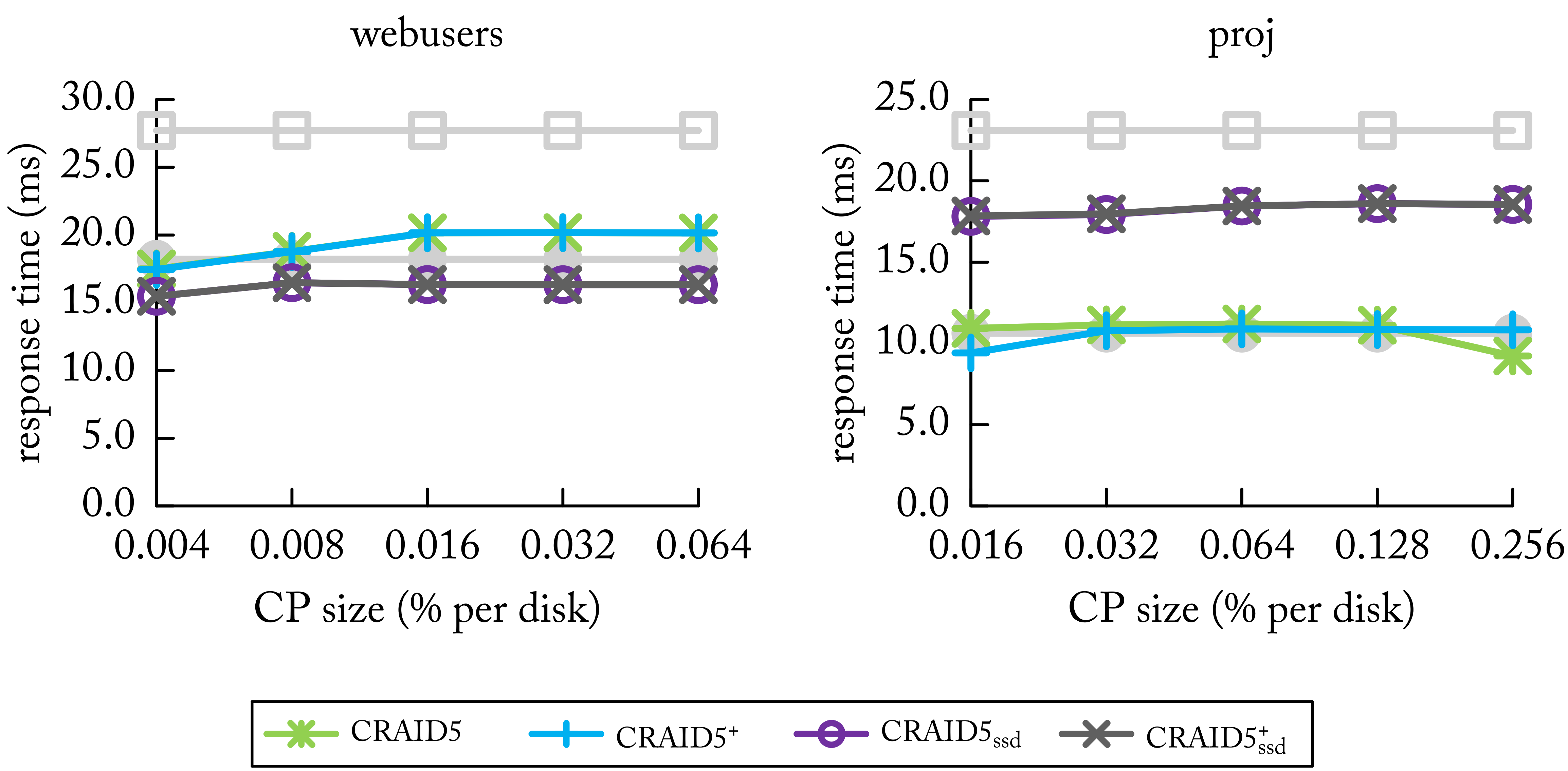
Response time (reads)



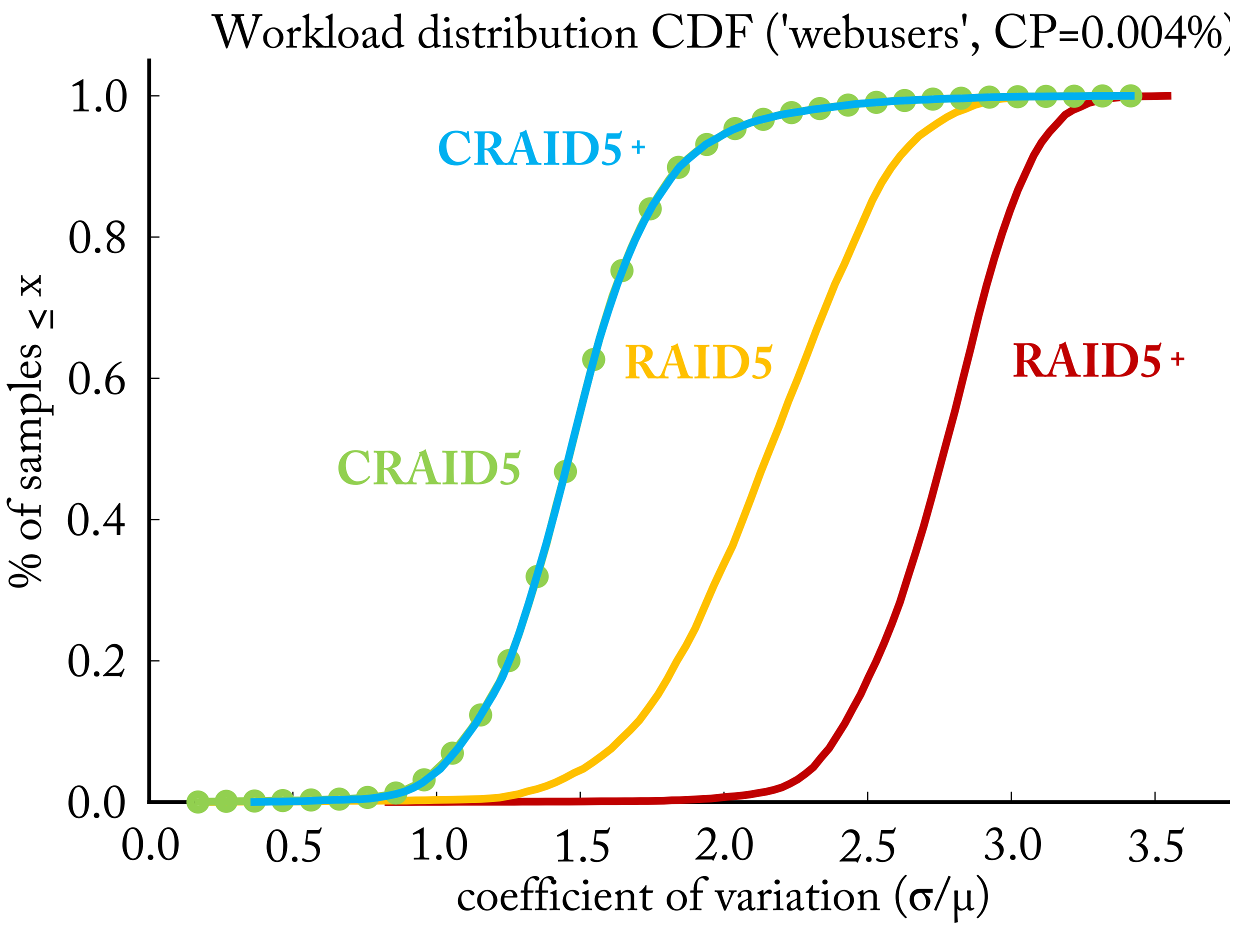
Response time (writes)

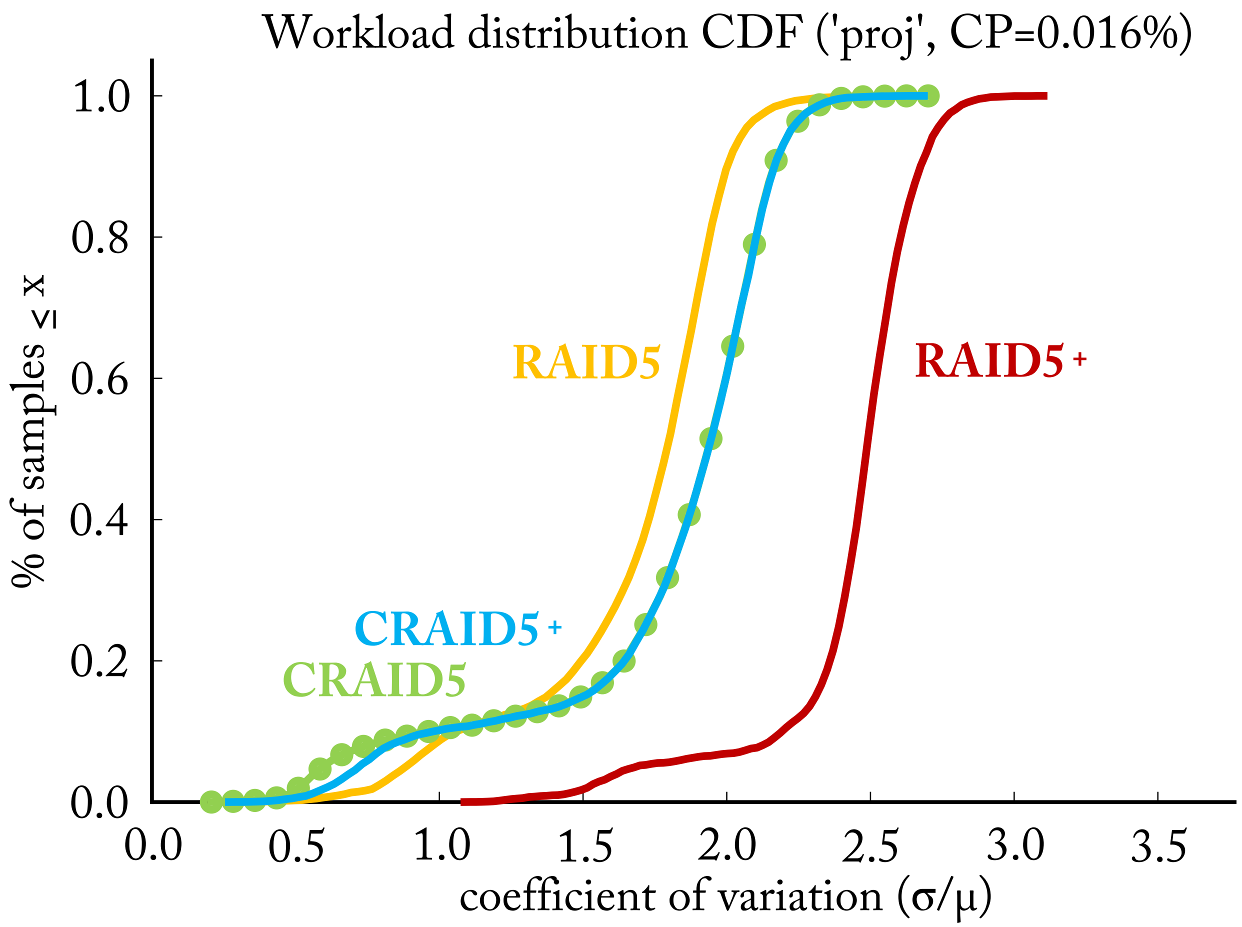


Response time (writes)

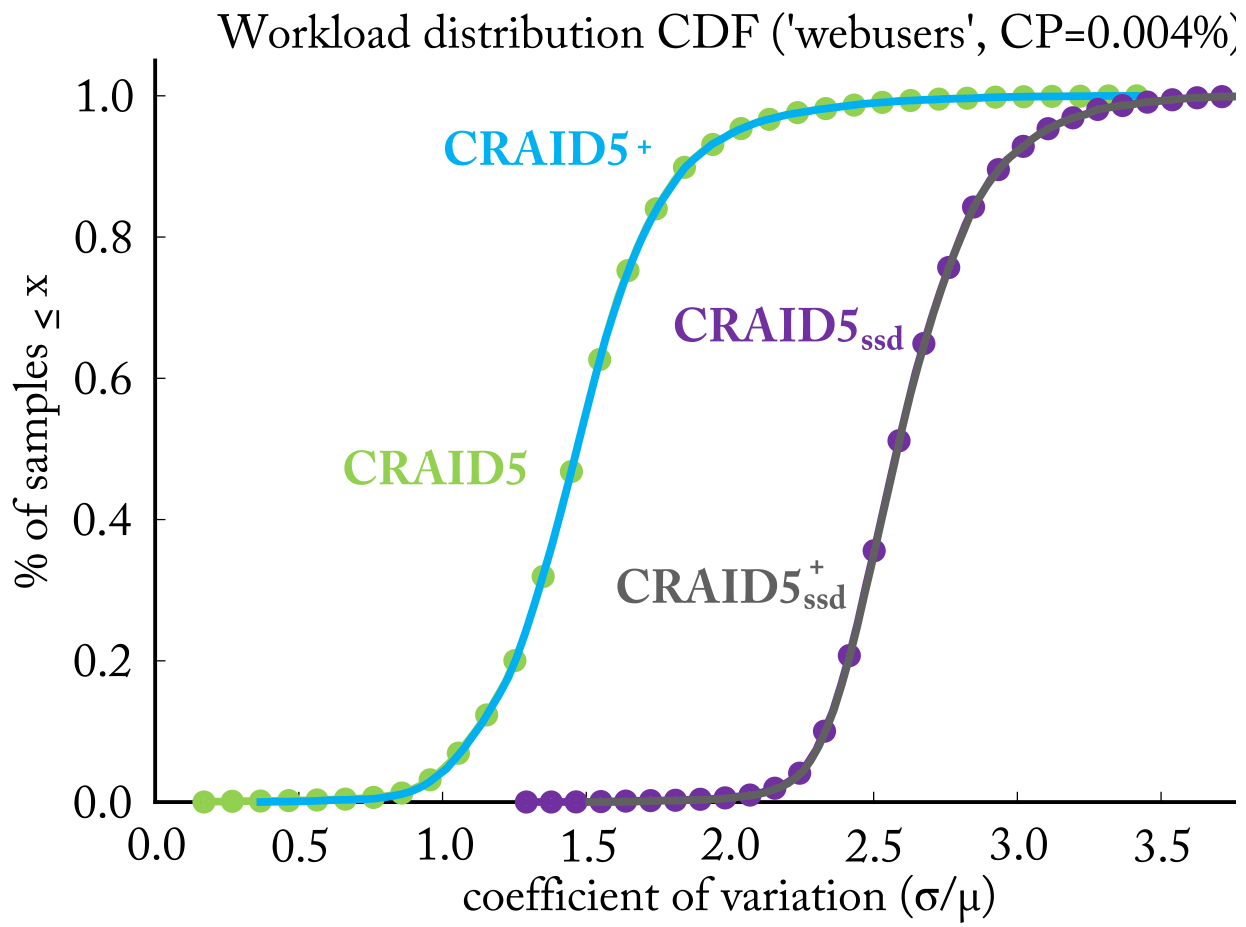


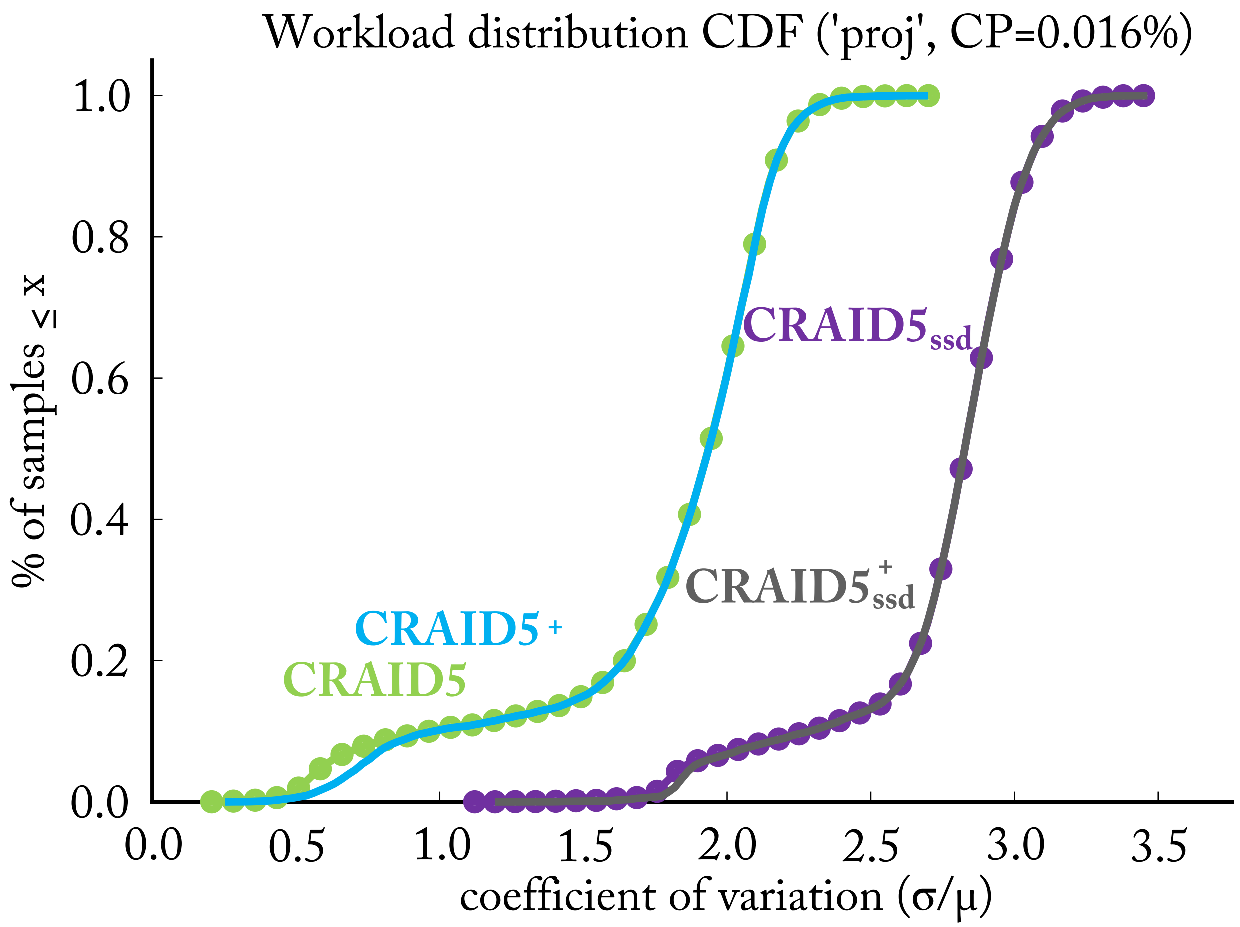
Workload distribution (rw)





Workload distribution (rw)





Conclusions

- Low upgrade overhead
  - Increased locality and sequentiality of hot data
  - Improved read/write performance
- Good workload distribution (close to ideal RAID)
- Good results w/ 1.28% available capacity
- Alternative to SSD caching for less $
  - Should work with full-SSD RAIDs

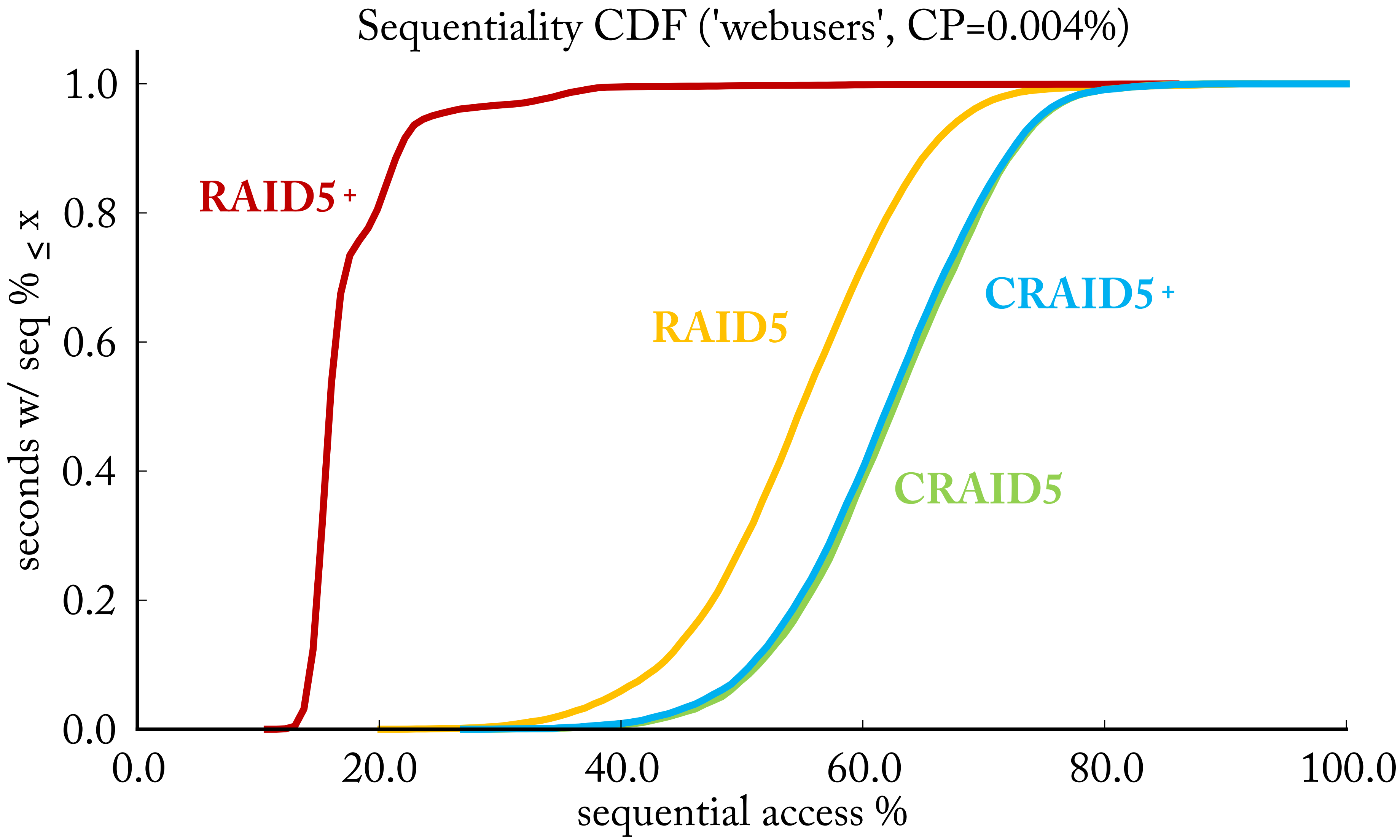
Future work

- Abandon simulations → HW prototype
- RAID6 and EC support
- Explore smarter prediction/rebalancing algorithms
- Extend beyond RAID storage

Q&A?

THANK YOU!

Sequentiality



More information

- Long-term locality of hot data:
  - *"Analyzing Long-Term Access Locality to Find Ways to Improve Distributed Storage Systems"*
    PDP2012
    A. Miranda, T. Cortes
- CRAID over RAID0
  - *"Performance Optimized Lustre"*
    PRACE-2IP white paper (2012)
    E. Artiaga, A. Miranda
    http://www.prace-ri.eu/IMG/pdf/D12-4_2ip.pdf