RAIDP:
ReplicAtion with Intra-Disk Parity

Eitan Rosenfeld, Aviad Zuck, Nadav Amit, Michael Factor, Dan Tsafrir

Today's Datacenters

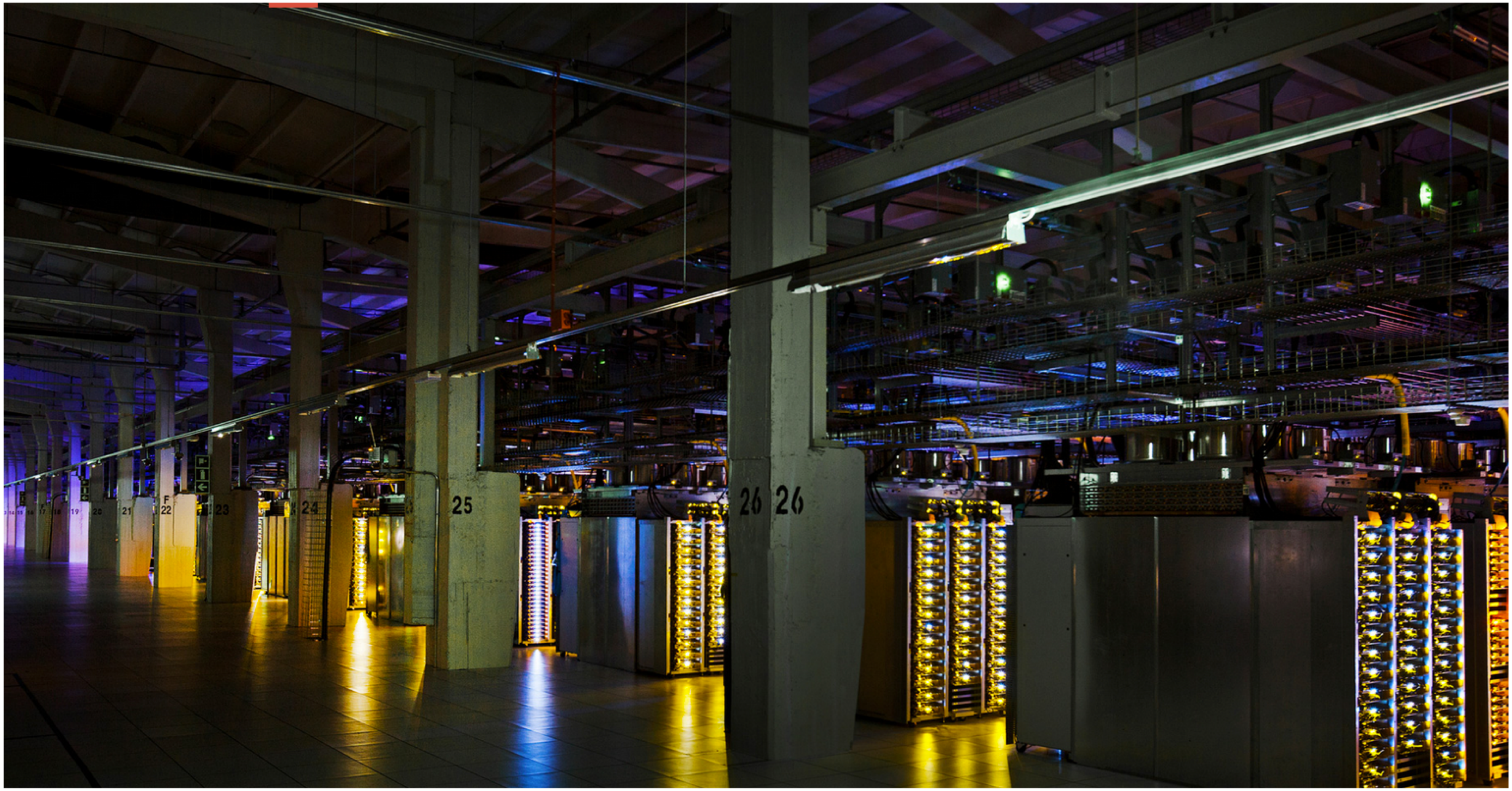


Image Source: http://www.google.com/about/datacenters/gallery/#/tech/14

Problem: Disks fail

- So storage systems use redundancy when storing data
- Two forms of redundancy:
  - Replication, or
  - Erasure codes

Replication vs. Erasure Coding

a=2

b=3

Replication vs. Erasure Coding

Replication

a=2

b=3

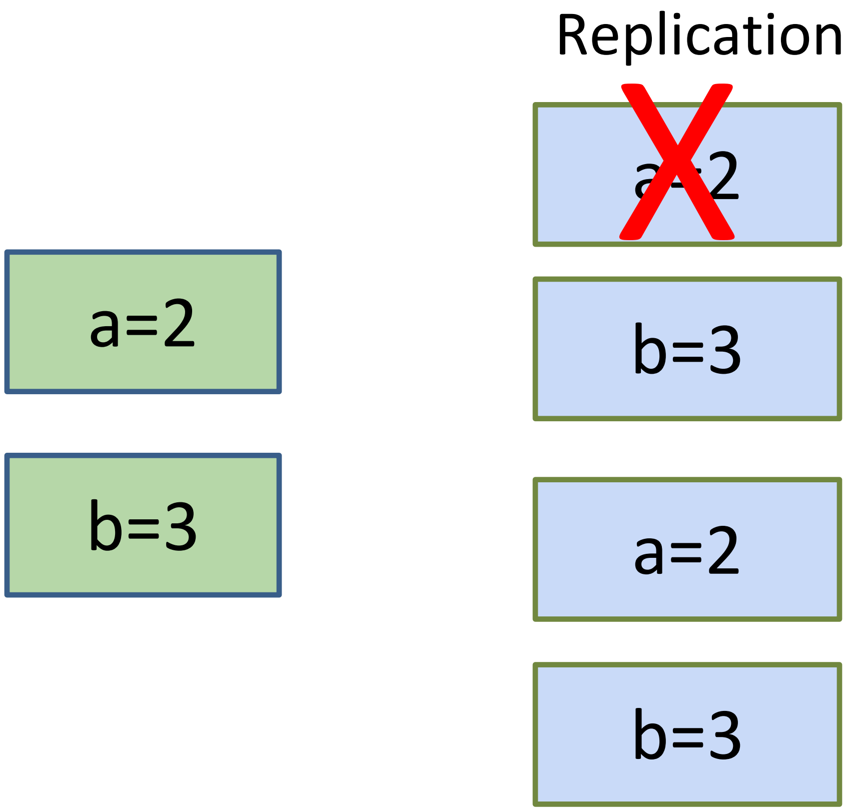
a=2

b=3

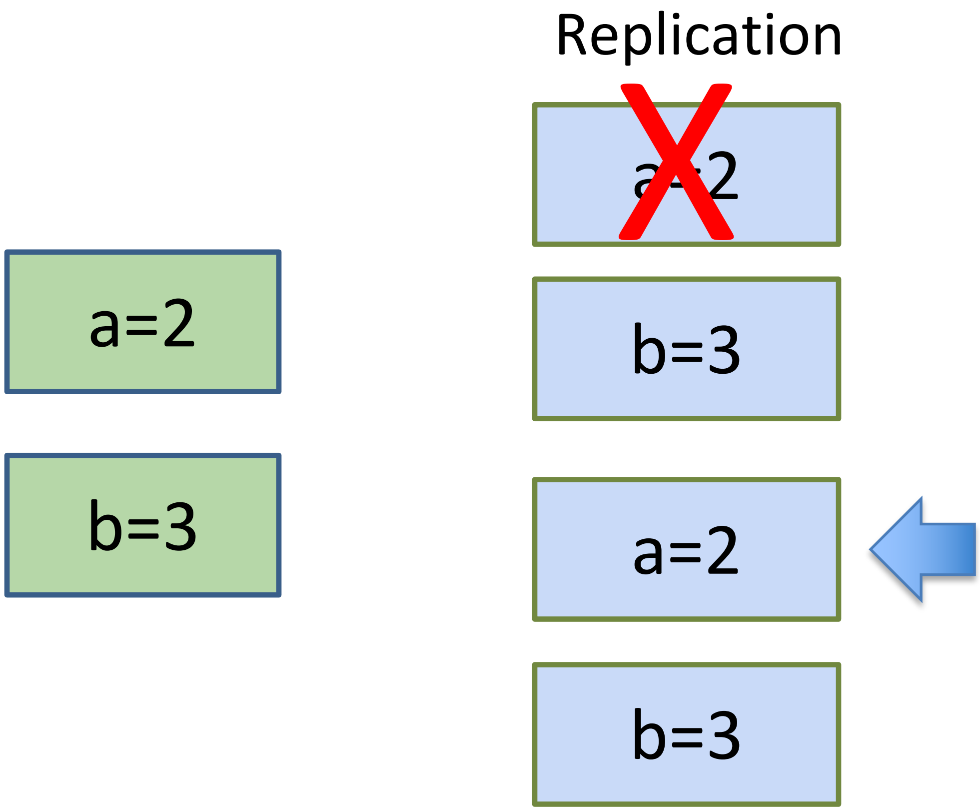
a=2

b=3

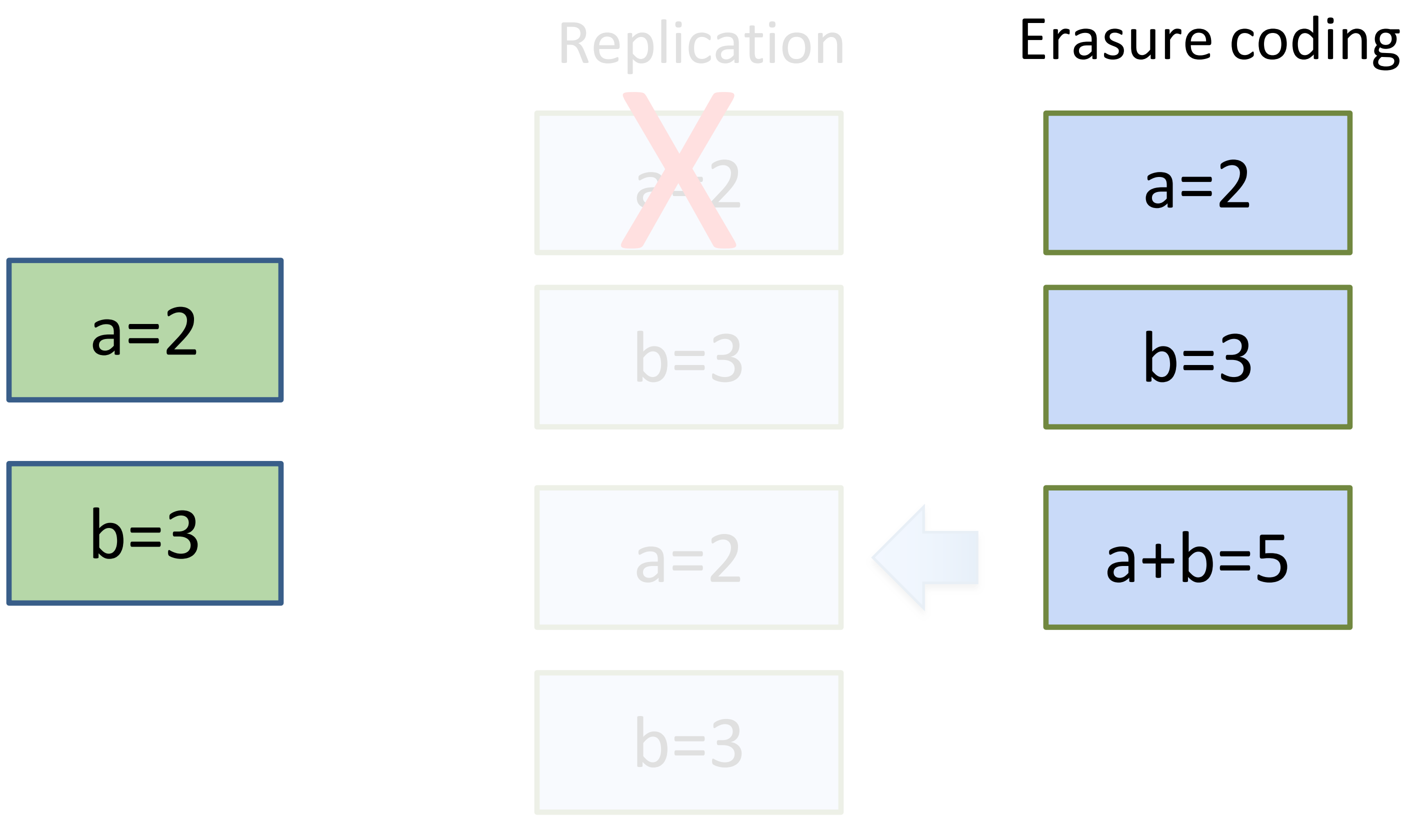
Replication vs. Erasure Coding



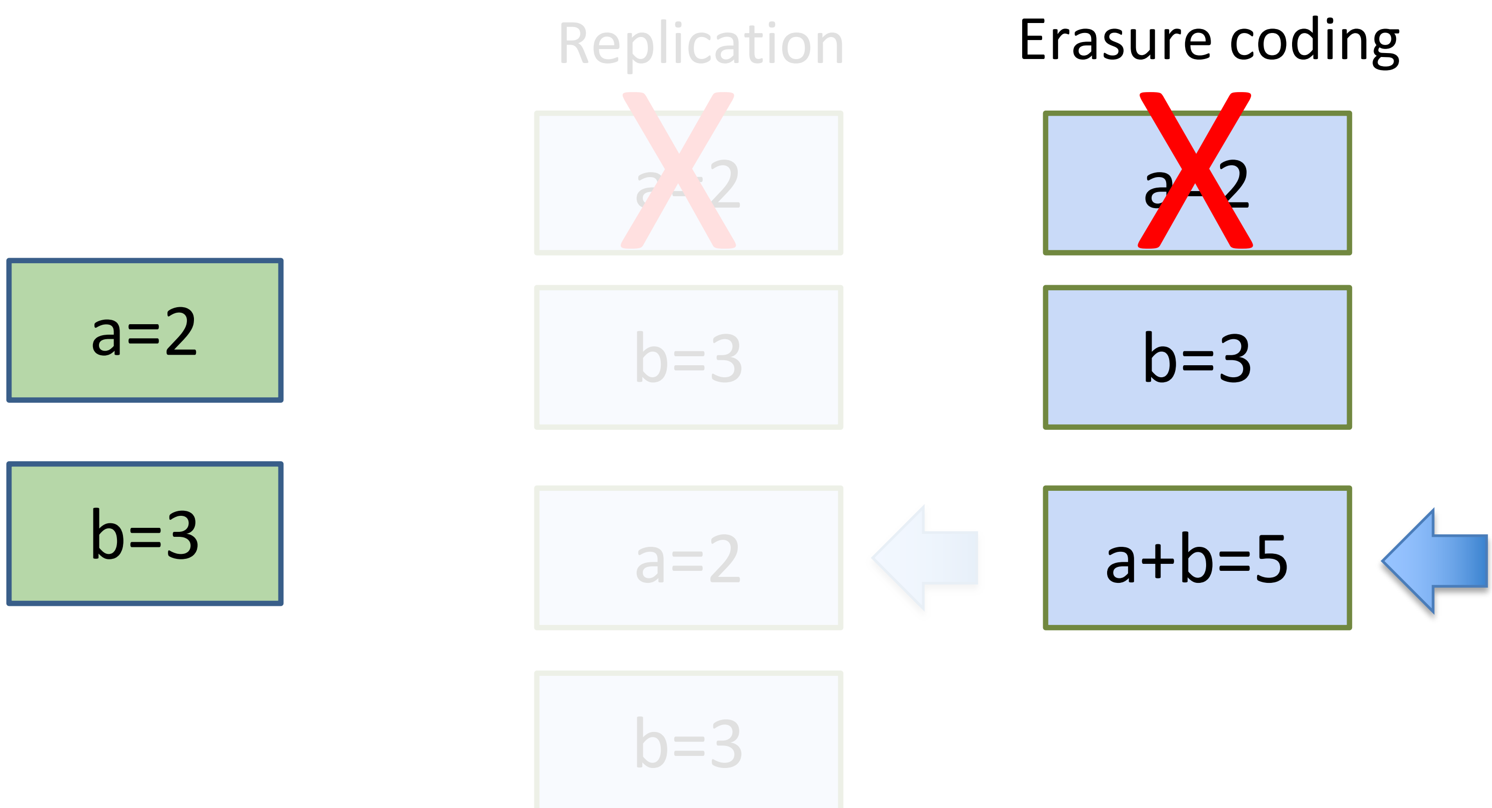
Replication vs. Erasure Coding



Replication vs. Erasure Coding



Replication vs. Erasure Coding



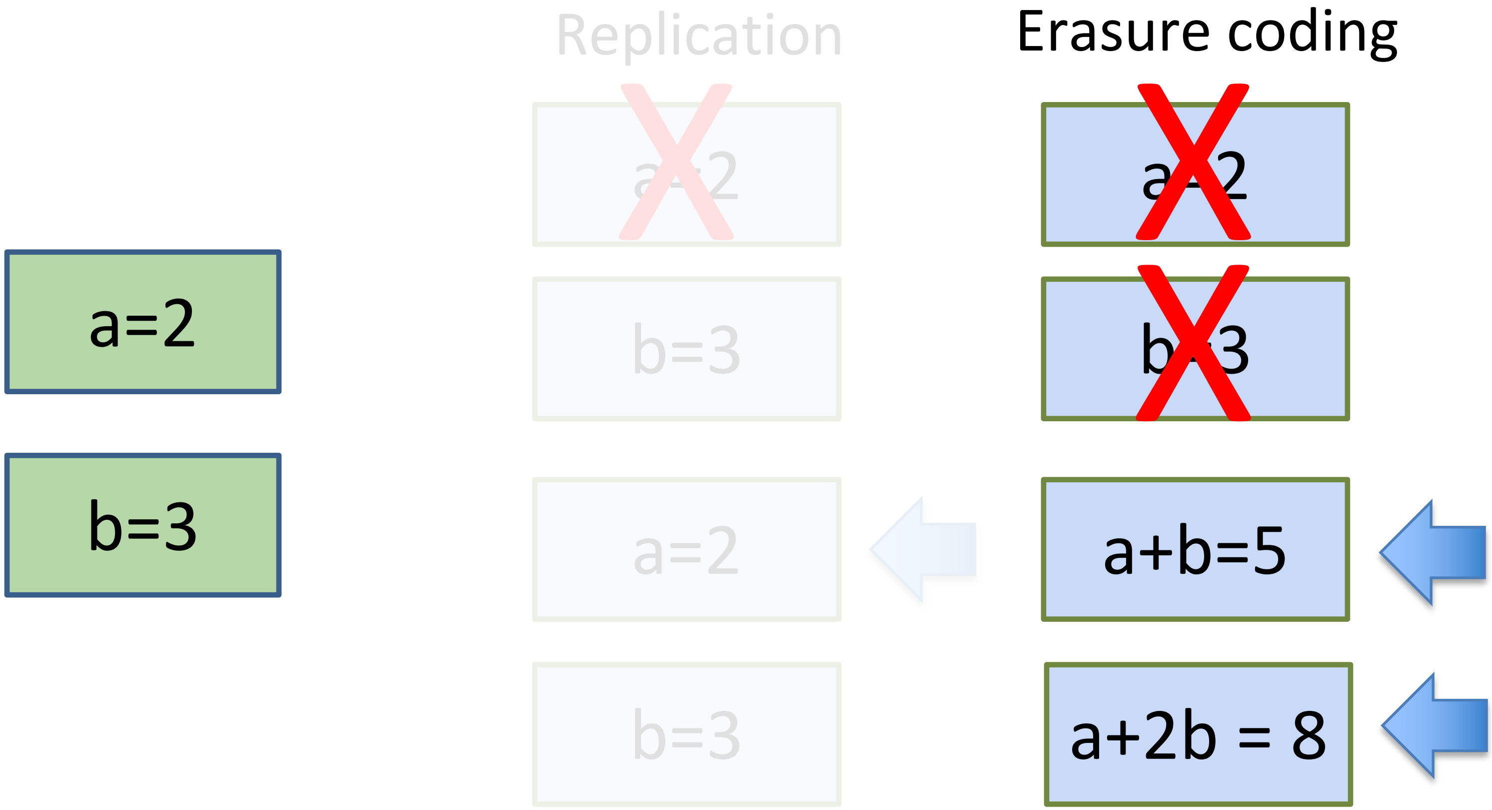
Replication vs. Erasure Coding

a=2

b=3

Replication	Erasure coding
~~a=2~~	~~a=2~~
b=3	b=3
a=2	a+b=5
b=3	a+2b = 8

Replication vs. Erasure Coding



Many modern systems replicate **warm** data

- Amazon's storage services
- Google File System (GFS)
- Facebook's Haystack
- Windows Azure Storage (WAS)
- Microsoft's Flat Datacenter Storage (FDS)
- HDFS (open-source file-system for Hadoop)
- Cassandra
- ...

Why is replication advantageous for warm data?

Better for **reads:**

1. Load balancing ✓
2. Parallelism ✓
3. Avoids degraded reads ✓

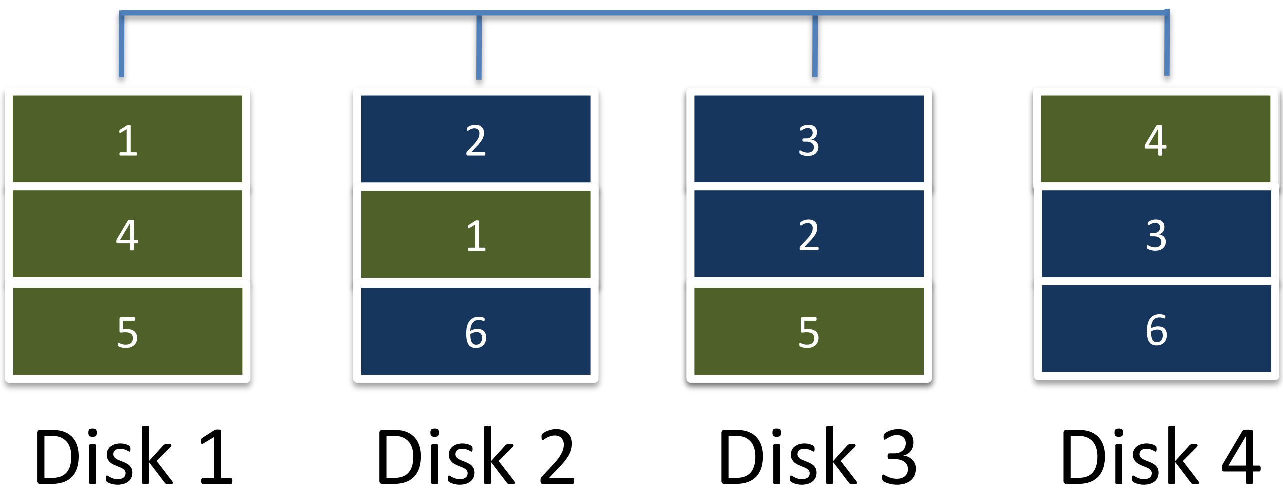
Better for **writes**:

4. Lower sync latency ✓

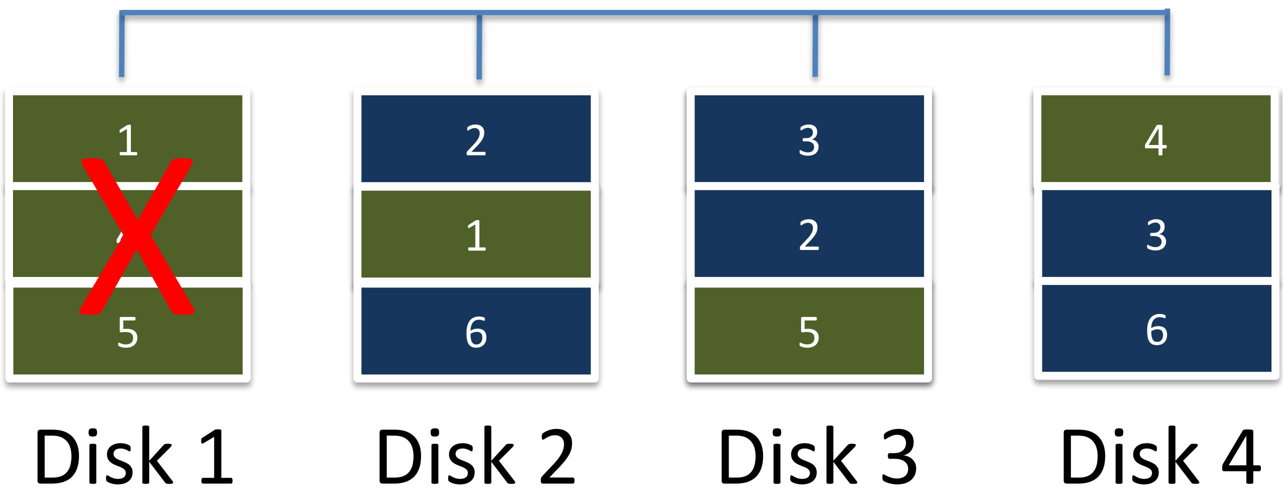
Better for **reads and writes**:

5. Increased sequentiality ✓
6. Avoids the CPU processing used for encoding ✓
7. Lower repair traffic ✓

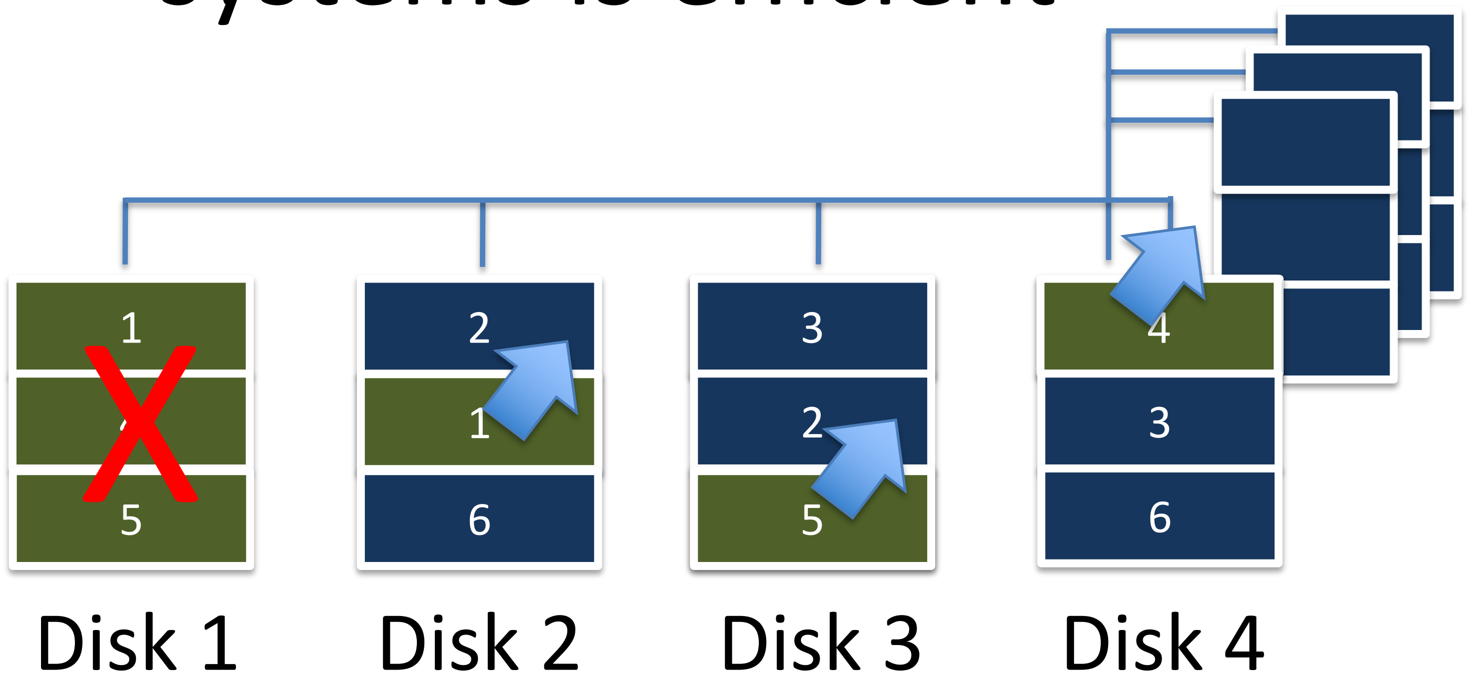
Recovery in replication based systems is efficient



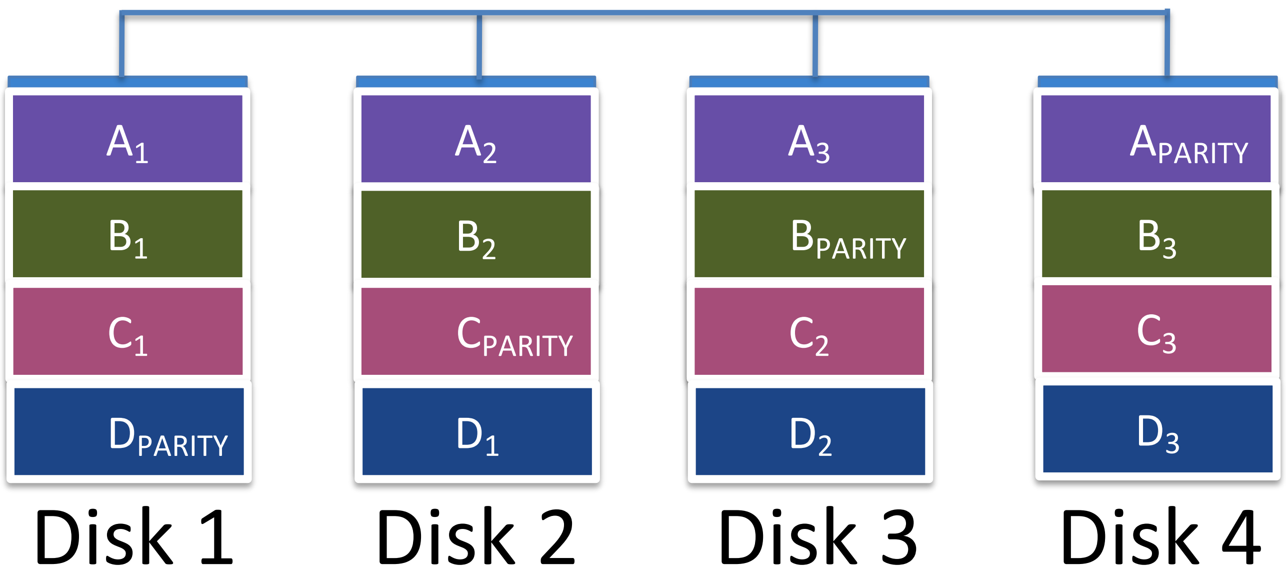
Recovery in replication based systems is efficient



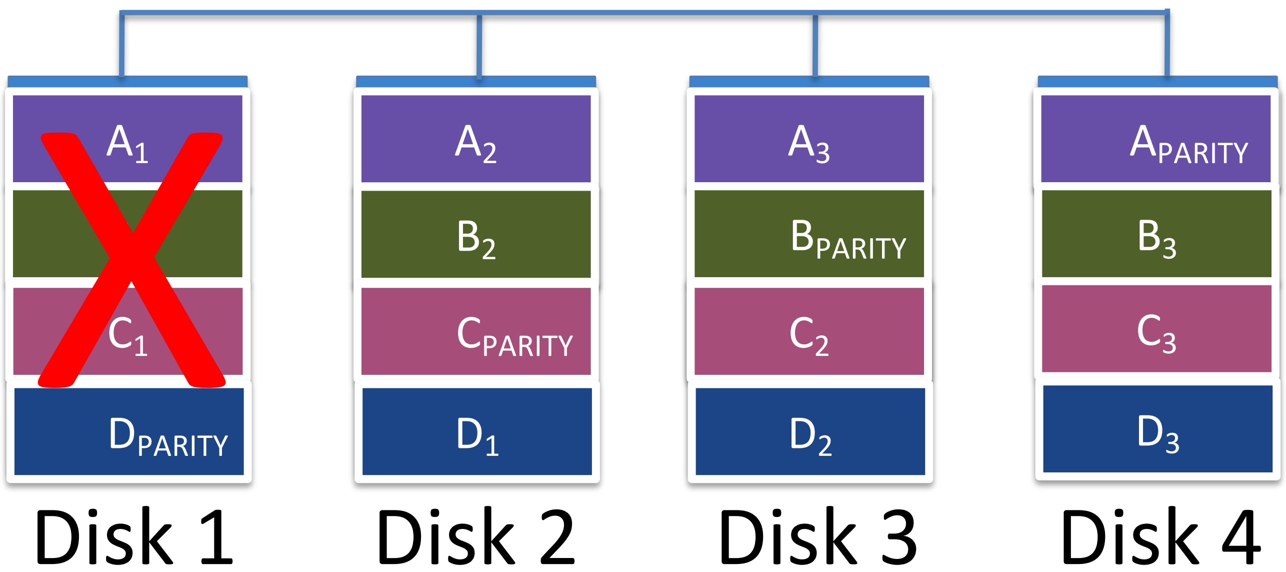
Recovery in replication based systems is efficient



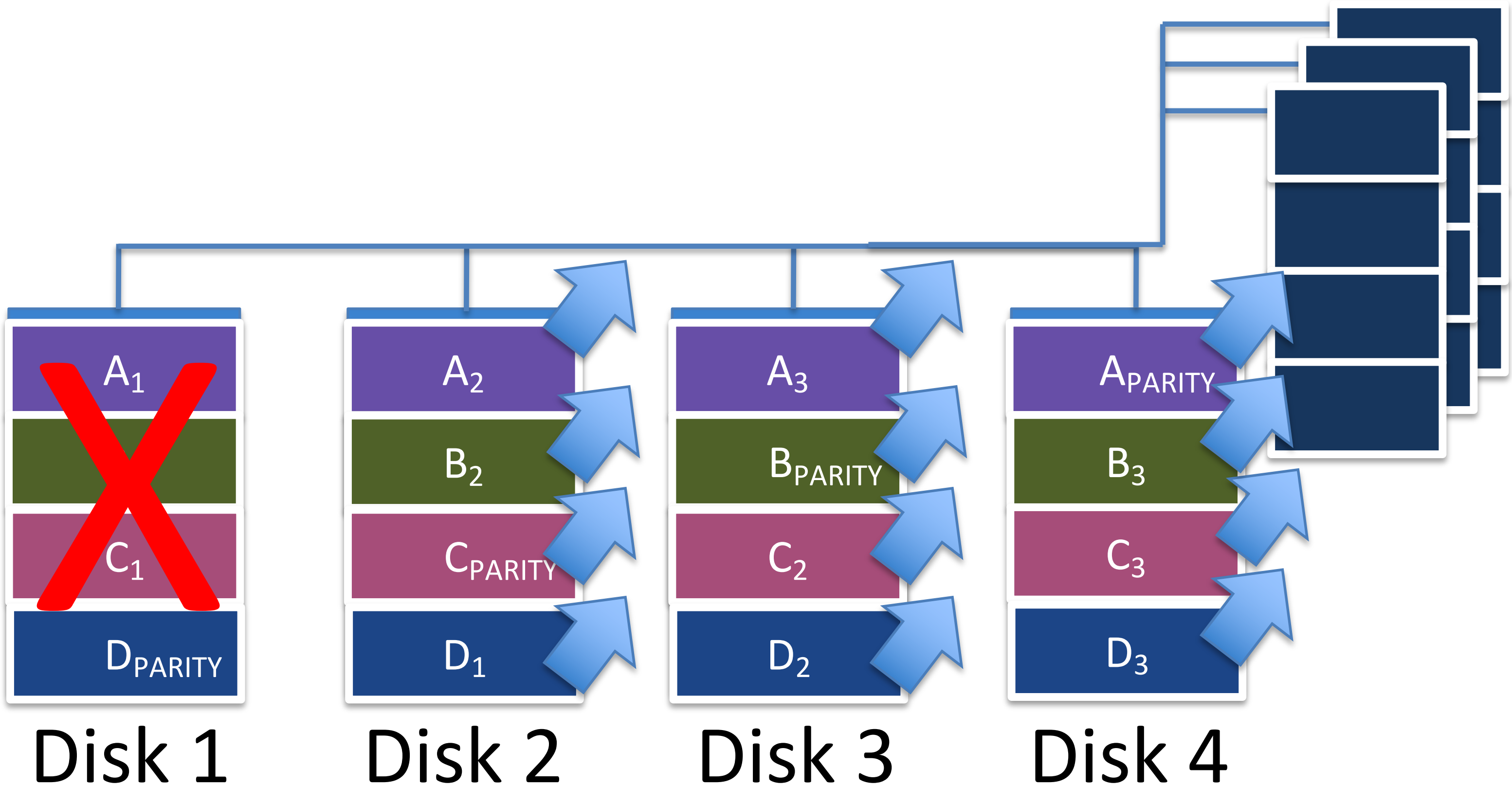
Erasure coding, on the other hand…



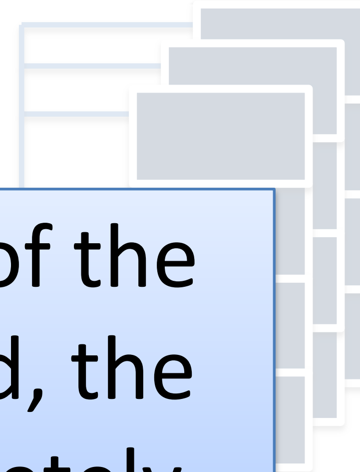
Erasure coding, on the other hand…



Erasure coding, on the other hand...



Erasure coding, on the other hand…



Facebook "estimate[s] that if 50% of the cluster was Reed-Solomon encoded, the repair network traffic would completely saturate the cluster network links"

Modern replicating systems **triple-replicate warm data**

- Amazon's DynamoDB
- Facebook's Haystack
- Google File System (GFS)
- Windows Azure Storage (WAS)
- Microsoft's Flat Datacenter Storage (FDS)
- HDFS (open-source file-system for Hadoop)
- Cassandra
- …

Bottom Line

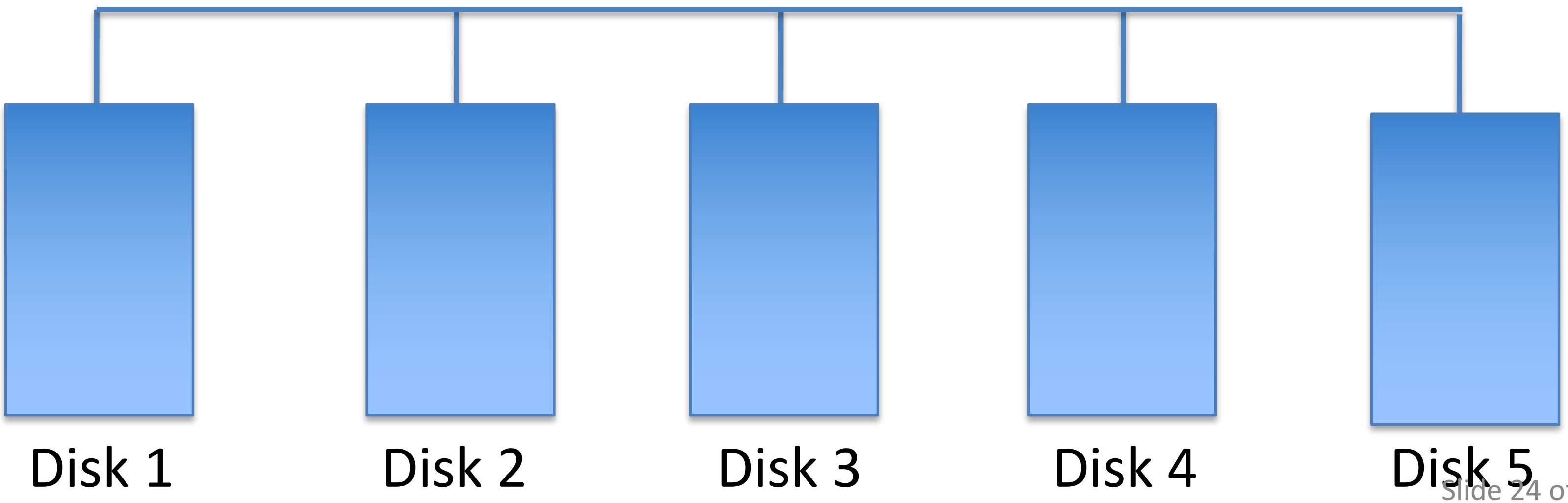
- **Replication** is used for **warm data** only
    - It's expensive! (Wastes storage, energy, network)
- **Erasure coding** used for the rest (**cold data**)

Our goal: Quickly recover from two simultaneous disk failures *without resorting to a third replica* for warm data

RAIDP - ReplicAtion with Intra-Disk Parity

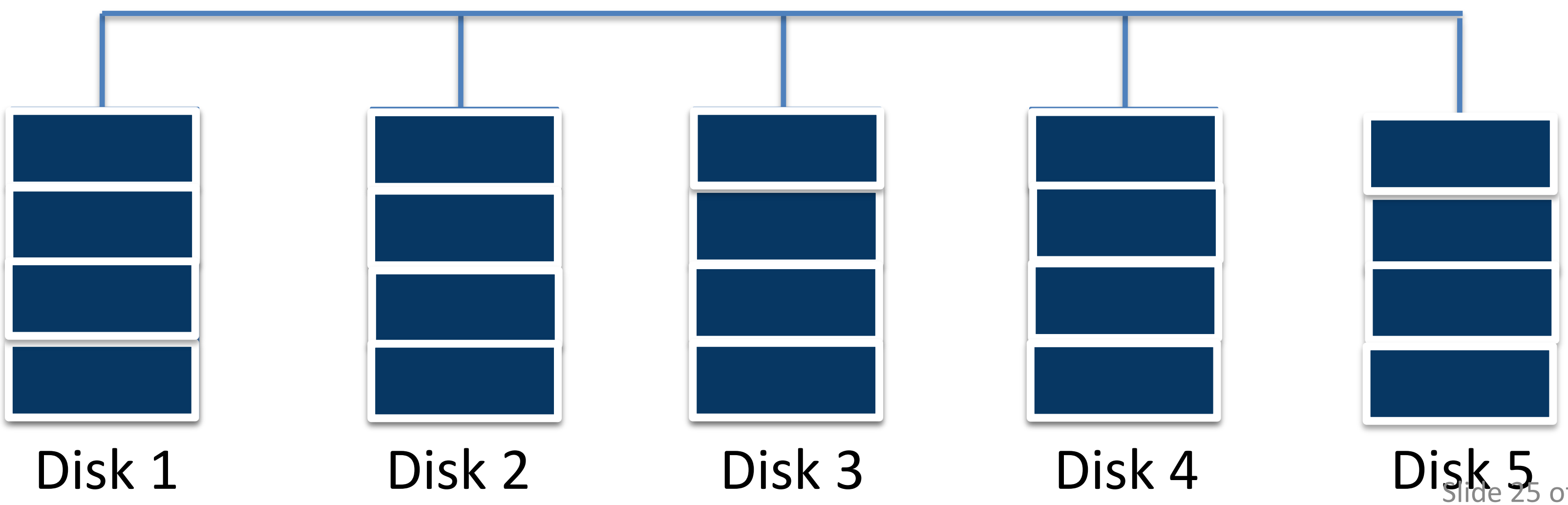
- **Hybrid** storage system for warm data with only *two** copies of each data object.
- **Recovers quickly** from a simultaneous failure of any two disks
- Largely enjoys the aforementioned 7 advantages of replication

System Architecture



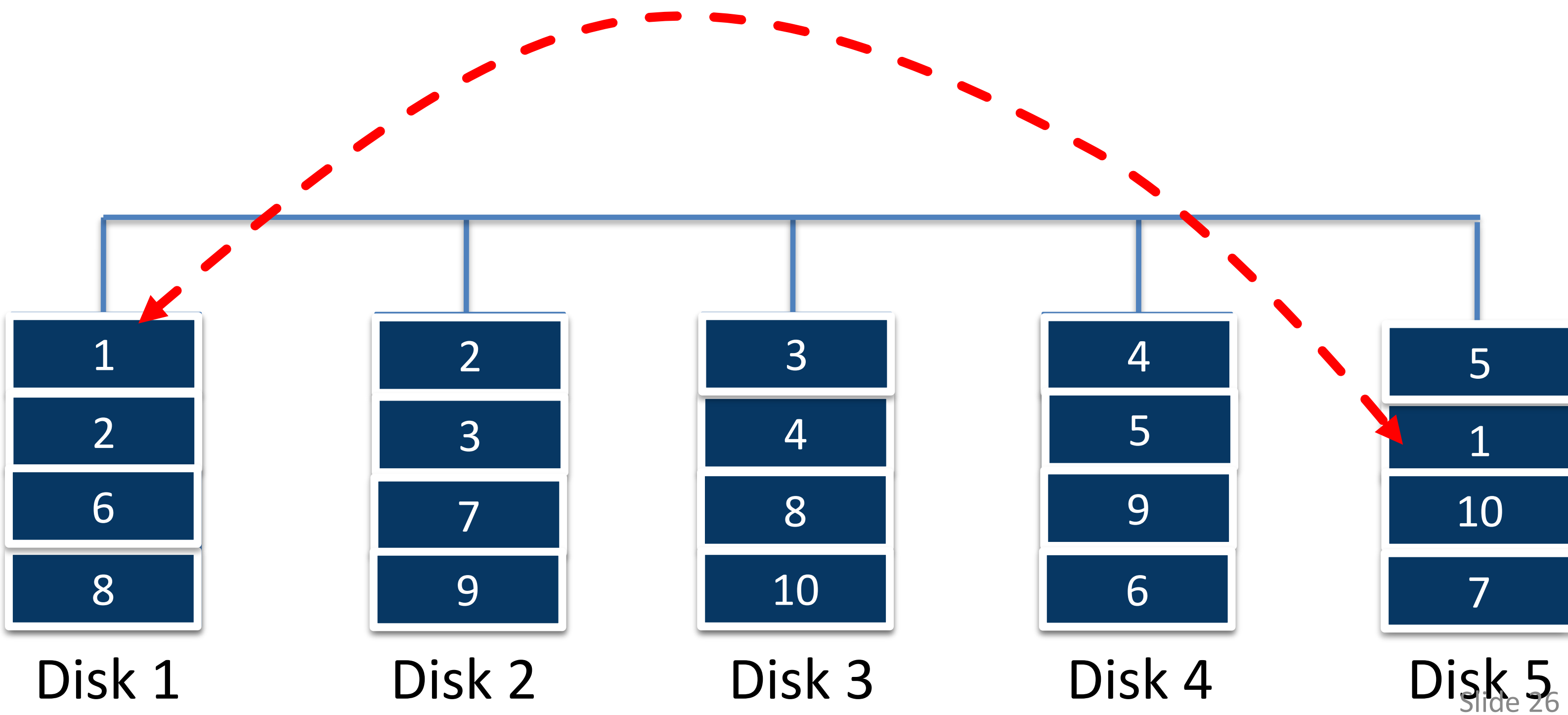
System Architecture

- Each of the N disks is divided into N-1 *superchunks*
  - e.g. 4GB each



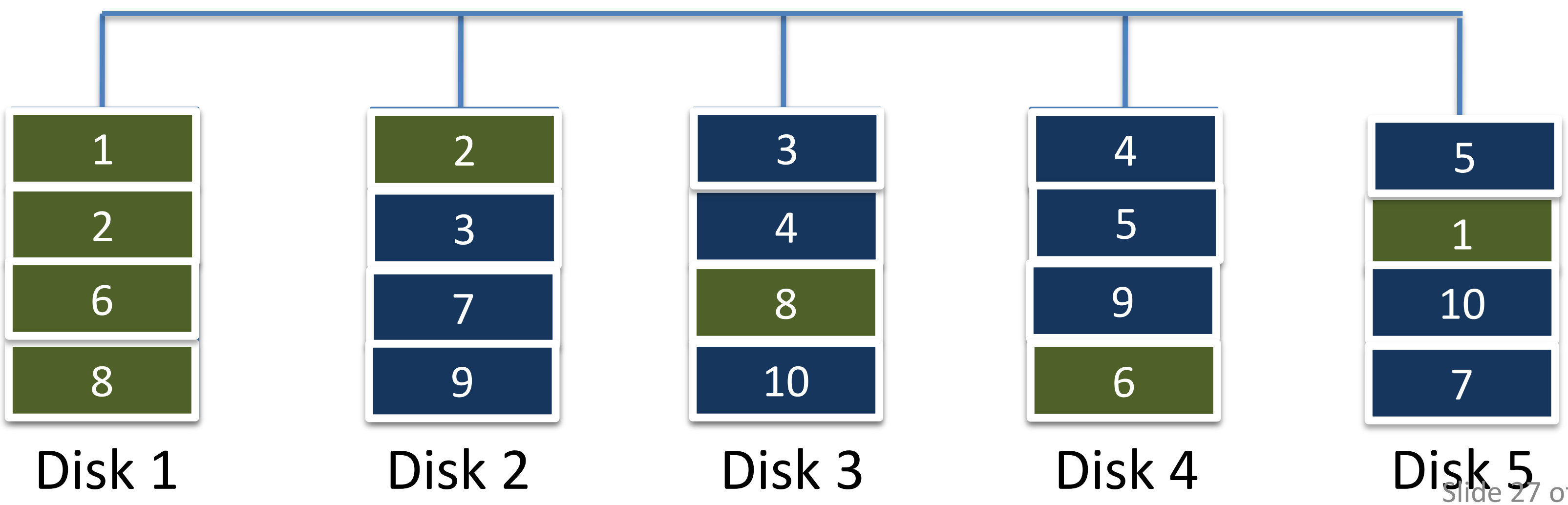
System Architecture

- Each of the N disks is divided into N-1 *superchunks*
  - e.g. 4GB each
- **1-Mirroring**: Superchunks must be 2-replicated

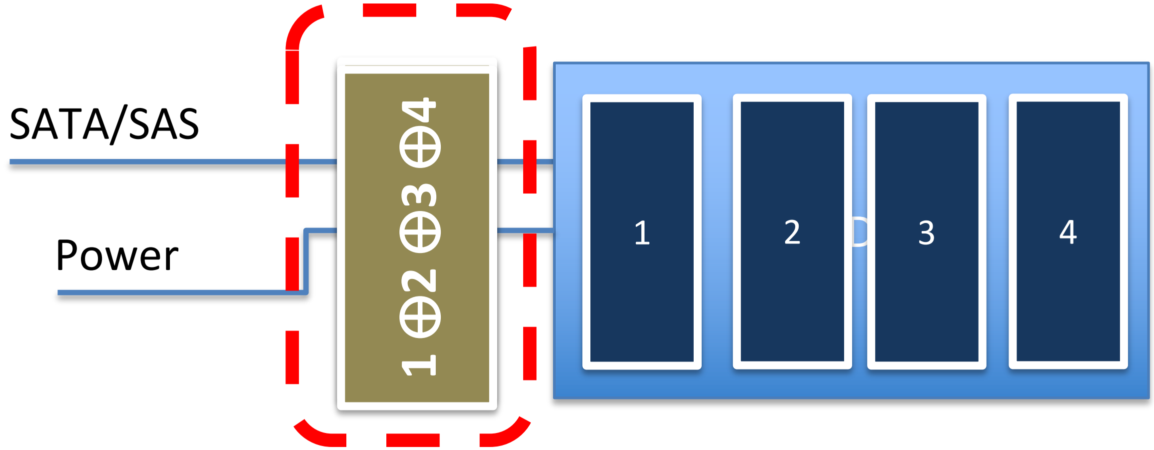


System Architecture

- Each of the N disks is divided into N-1 *superchunks*
  - e.g. 4GB each
- **1-Mirroring**: Superchunks must be 2-replicated
- **1-Sharing**: Any two disks share at most one superchunk

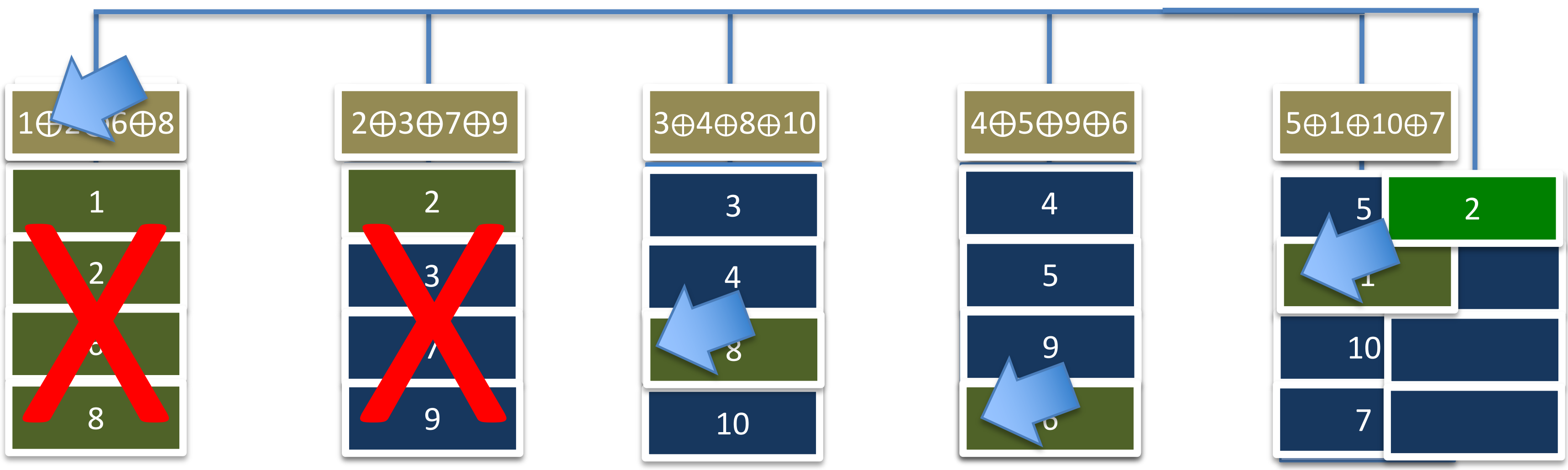


Introducing "disk add-ons"



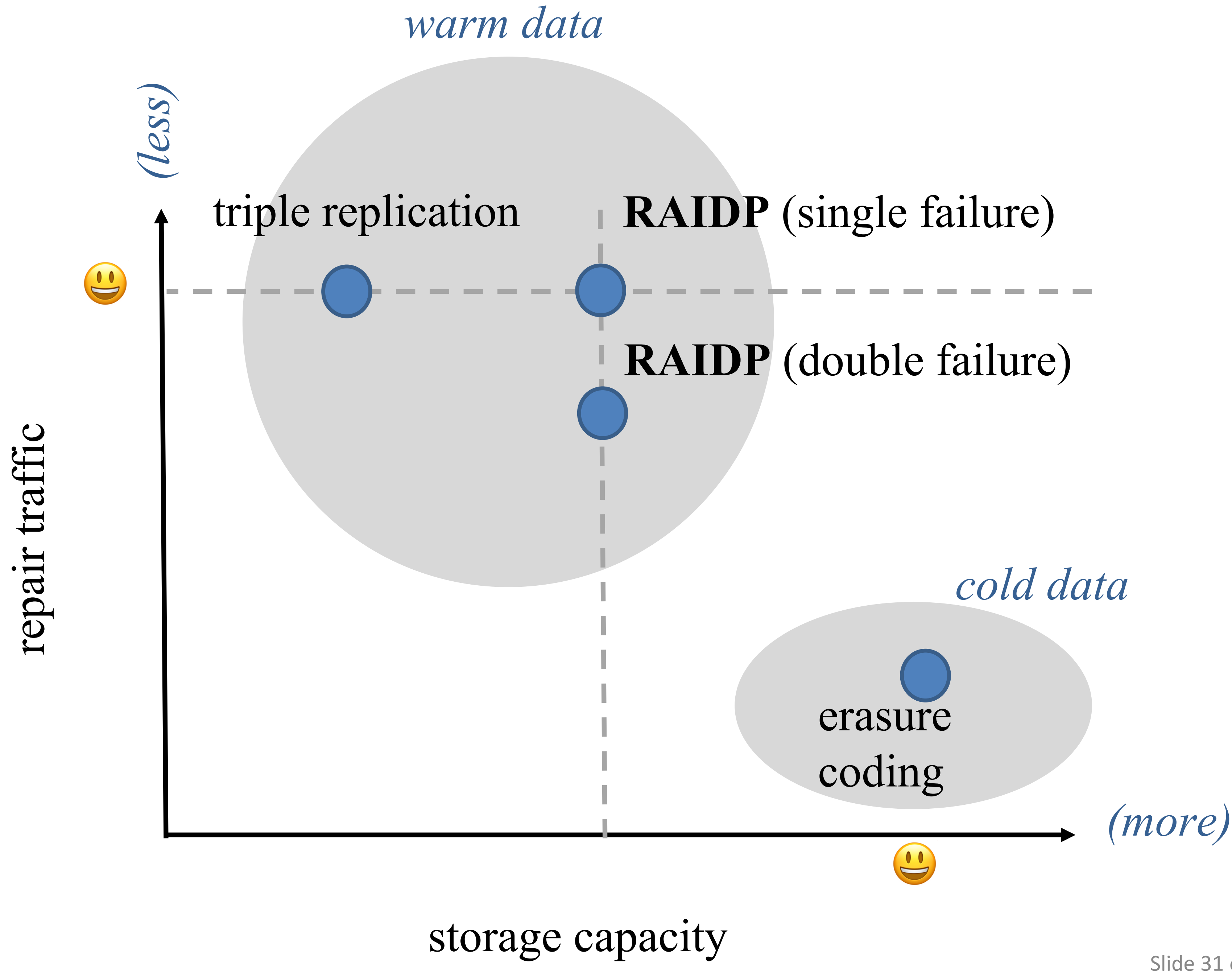
- Associated with a specific disk
  - Interposes all I/O to disk
  - Stores an erasure code of the local disk's superchunks
  - Fails separately from the associated disk

RAIDP Recovery



XOR Add-on 1 with the surviving superchunks from Disk 1.

$1\oplus2\oplus6\oplus8 \oplus 8 \oplus 6 \oplus 1 = 2$


warm data
(less)
triple replication
RAIDP (single failure)
RAIDP (double failure)
repair traffic
cold data
erasure
coding
(more)
storage capacity

Lstor Feasability

Goal: Replace a third replica disk with 2 Lstors

Lstors need to be cheap, fast, and fail separately from disk.

- **Storage:** Enough to maintain parity (~$9) [1]
- **Processing:** Microcontroller for local machine independence (~$5) [2]
- **Power:** Several hundred Amps for 2–3 min from small supercapacitor to read data from the Lstor

Commodity 2.5” 4TB disk for storing an additional replica costs $100:

66% more than a conservative estimate of the cost of two Lstors

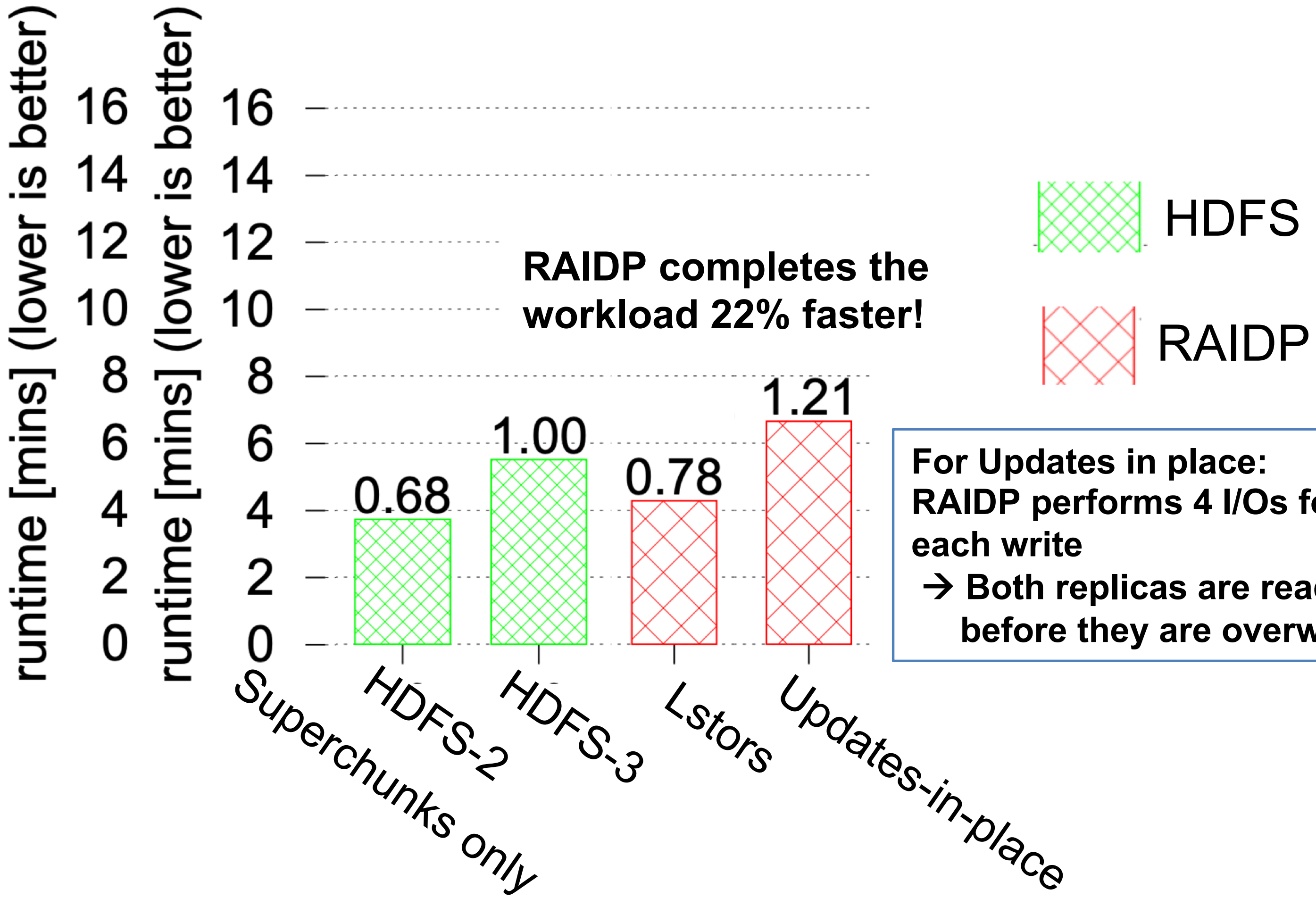
Implementation in HDFS

- RAIDP implemented in in Hadoop 1.0.4
  - Two variants:
    - Append-only
    - Updates-in-place
- 3K LOC extension to HDFS
  - Pre-allocated block files to simulate superchunks
  - Lstors simulated in memory
  - Added crash consistency and several optimizations

Evaluation

- RAIDP vs. HDFS with 2 and 3 replicas
- Tested on a 16-node cluster
  - Intel Xeon CPU E3-1220 V2 @ 3.10GHz
  - 16GB RAM
  - 7200 RPM disks
- 10Gbps Ethernet
- 6GB superchunks, ~800GB cluster capacity

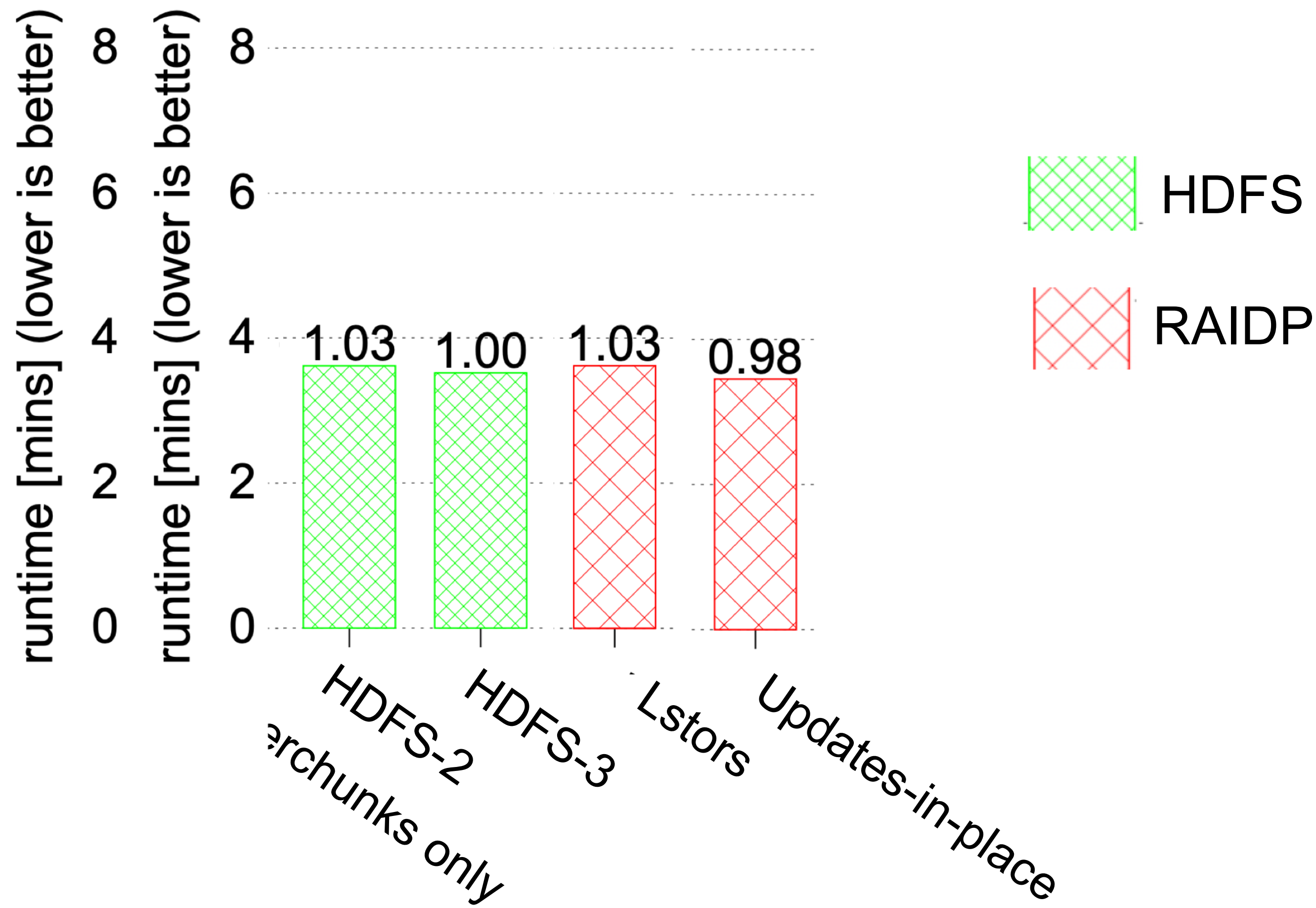
Hadoop write throughput (Runtime of writing 100GB)



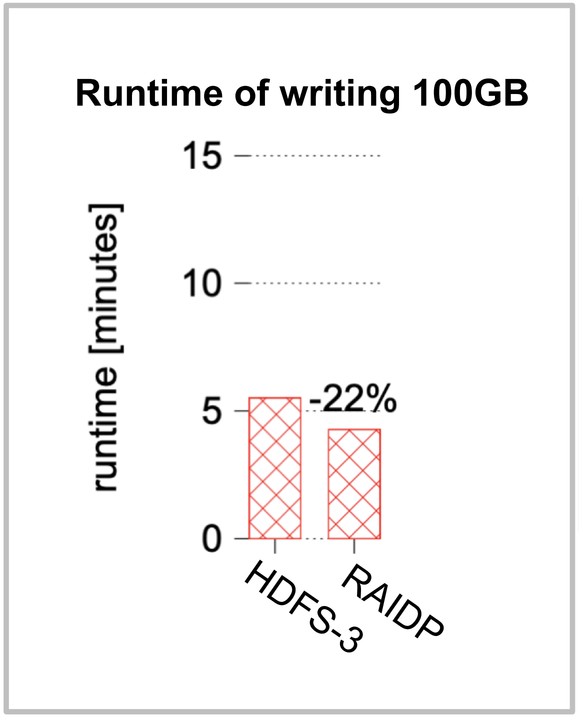
RAIDP completes the workload 22% faster!

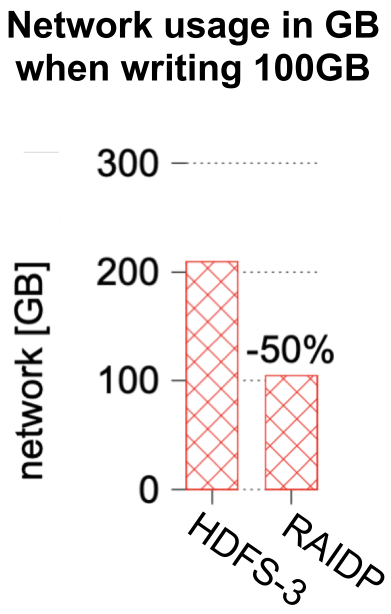
For Updates in place:
RAIDP performs 4 I/Os for each write
→ Both replicas are read before they are overwritten

Hadoop read throughput (Runtime of reading 100GB)

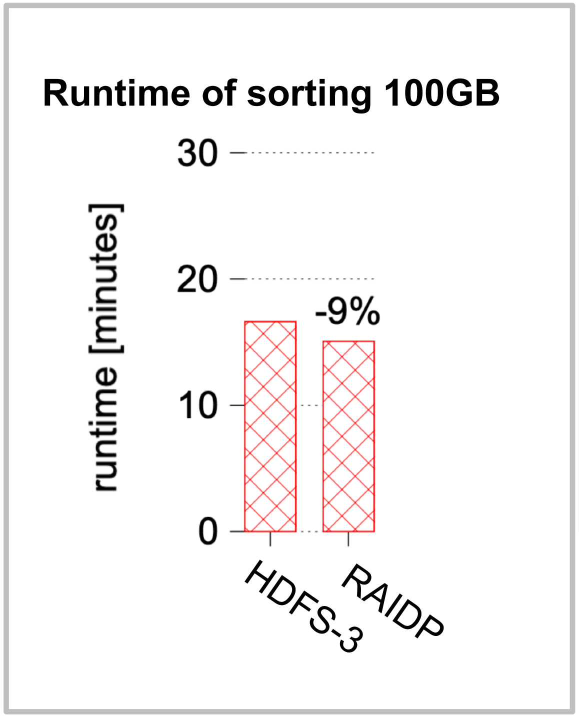


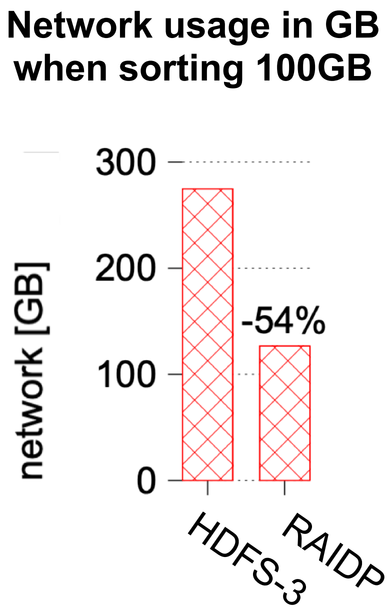
Write Runtime vs. Network Usage





TeraSort Runtime vs. Network Usage





Recovery time in RAIDP

System	1Gbps Network	10Gbps Network
RAIDP	**827 s**	**125 s**
RAID-6	**12,300 s**	**1,823 s**

16 node cluster with 6GB superchunk

RAIDP recovers 14x faster!

For erasure coding, such a recovery is required for **every** disk failure.
For RAIDP, such a recovery is only required after the 2nd failure.

Vision and Future work

- Survives two simultaneous failures with only two replicas
- Can be augmented to withstand more than two simultaneous failures
  - “Stacked” LSTORs
- Building Lstors instead of simulating them
- Equipping Lstors with network interfaces so that they can withstand rack failures
- Experiment with SSDs

Summary

- RAIDP achieves similar failure tolerance as 3-way replicated systems
  - Better performance when writing new data
  - Small performance hit during updates
- Yet:
  - Requires 33% less storage
  - Uses considerably less network bandwidth for writes
  - Recovery is much more efficient than EC

- Opens the way for storage vendors and cloud providers to use 2 (instead of 3, or more) replicas
  - Potential savings in size, energy, and capacity