A "Hitchhiker's" Guide to Fast and Efficient Data Reconstruction in Erasure-coded Data Centers

K. V. Rashmi, Nihar Shah, D. Gu,
H. Kuang, D. Borthakur, K. Ramchandran

Need for Redundant Storage in Data Centers

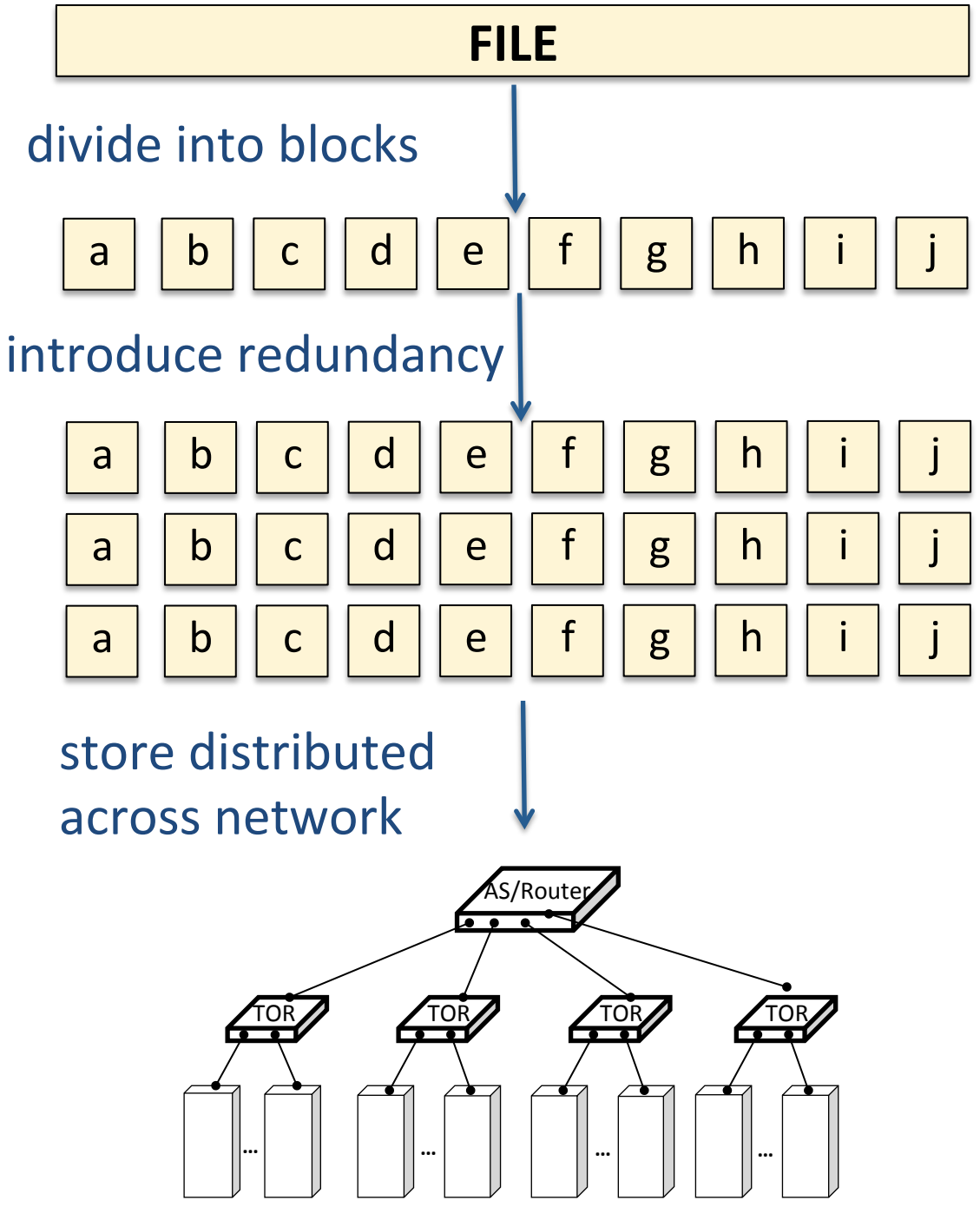
- Frequent unavailability events in data centers
  - unreliable components
  - software glitches, maintenance shutdowns, power failures, etc.
- Redundancy necessary for reliability and availability

Popular Approach for Redundant Storage: Replication

- Distributed file systems used in data centers store multiple copies of data on different machines
- Machines typically chosen on different racks
  - to tolerate rack failures

E.g., Hadoop Distributed File System (HDFS) stores 3 replicas by default

HDFS


FILE
divide into blocks
a b c d e f g h i j
introduce redundancy
a b c d e f g h i j
a b c d e f g h i j
a b c d e f g h i j
store distributed
across network
AS/Router
TOR
TOR
TOR
TOR

Massive Data Sizes: Need Alternative to Replication

- Small to moderately sized data: disk storage is inexpensive
  - replication viable
- No longer true for massive scales of operation
  - e.g., Facebook data warehouse cluster stores multiple tens of Petabytes (PBs)

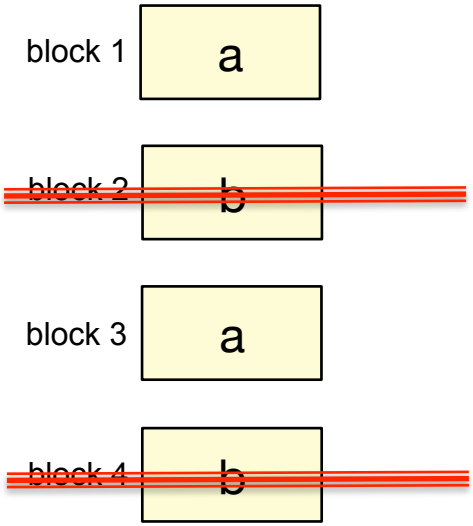
"Erasure codes" are an alternative

Erasure Codes in Data Centers

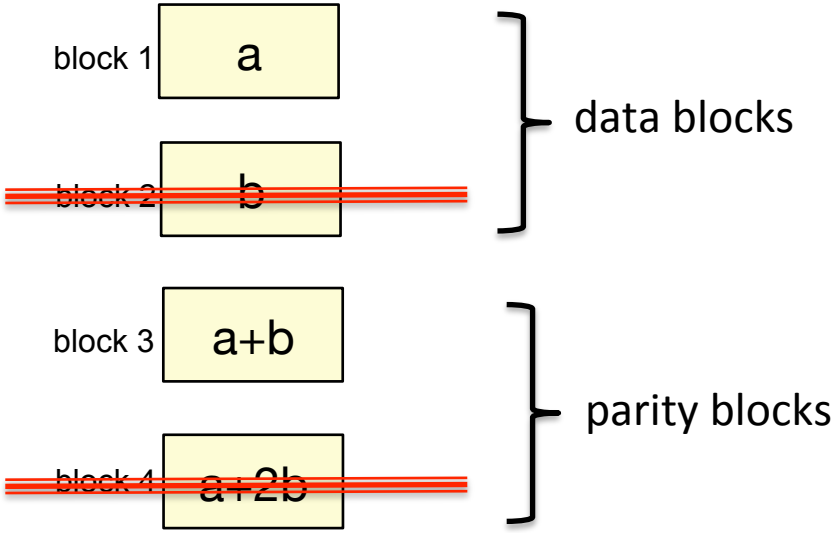
- Facebook data warehouse cluster
  - uses Reed-Solomon (RS) codes instead of 3-replication on a portion of the data
  - ***savings of multiple Petabytes of storage space***

Erasure Codes

Replication



Reed-Solomon (RS) code



	Replication	Reed-Solomon (RS) code
Overhead	2x	2x
Fault tolerance:	tolerates any one failure	tolerates any two failures

In general, erasure codes provide orders of magnitude higher reliability at much smaller storage overheads

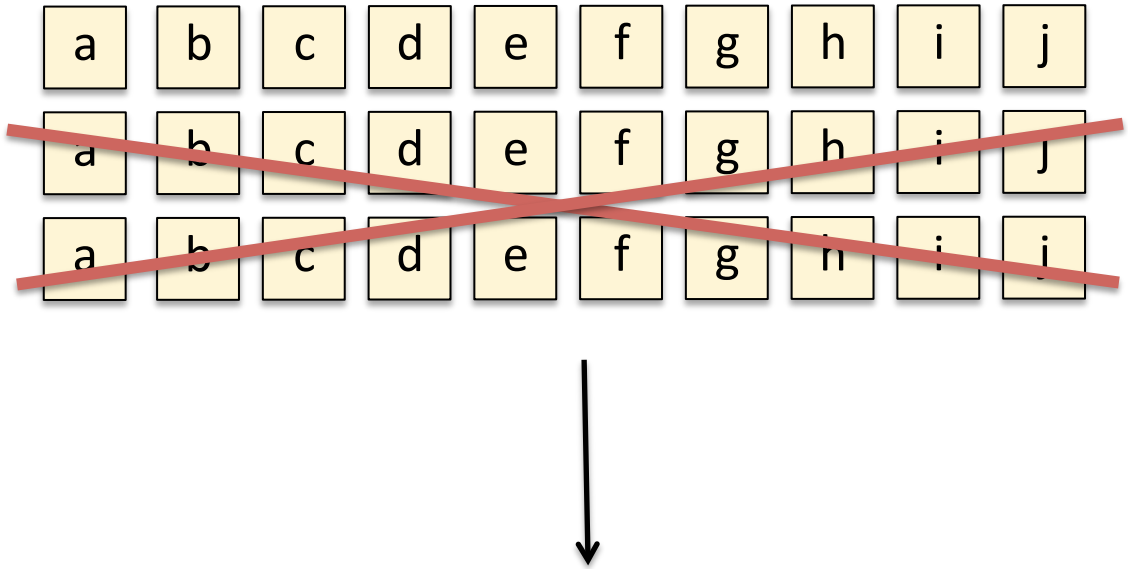
Outline

- Erasure Codes in Data Centers
  - HDFS
- Impact on the data center network
  - Problem description
- Our system: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

Outline

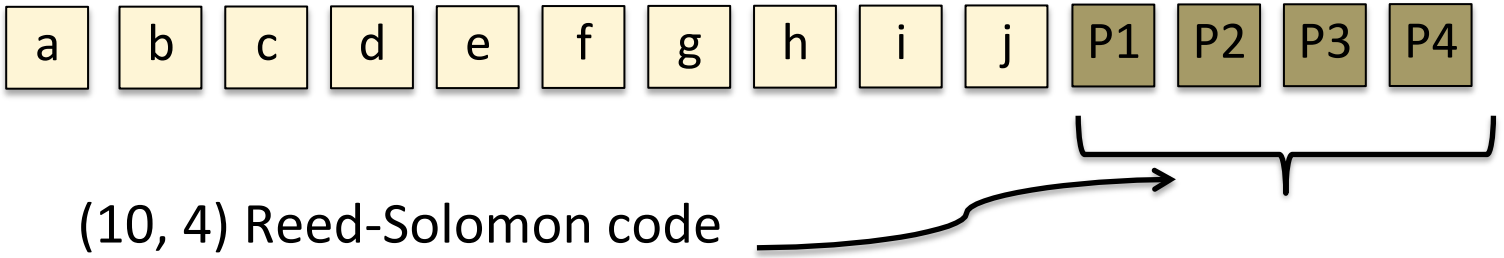
- Erasure Codes in Data Centers
  - HDFS
- Impact on the data center network
  - Problem description
- Our solution: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

Erasure codes in Data Centers: HDFS-RAID



Overhead: 3x

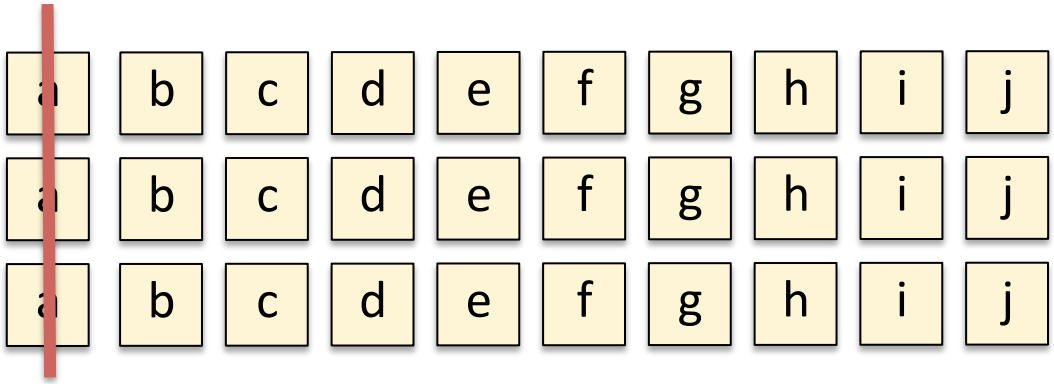
Overhead: 1.4x



(10, 4) Reed-Solomon code

Borthakur, "HDFS and Erasure Codes (HDFS-RAID)"
Fan, Tantisiriroj, Xiao and Gibson, "DiskReduce: RAID for Data-Intensive Scalable Computing", PDSW 09

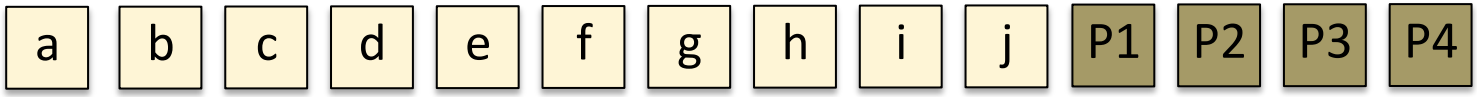
Erasure codes in Data Centers: HDFS-RAID



Overhead: 3x

Cannot tolerate many 3-failures

Overhead: 1.4x



(10, 4) Reed-Solomon code

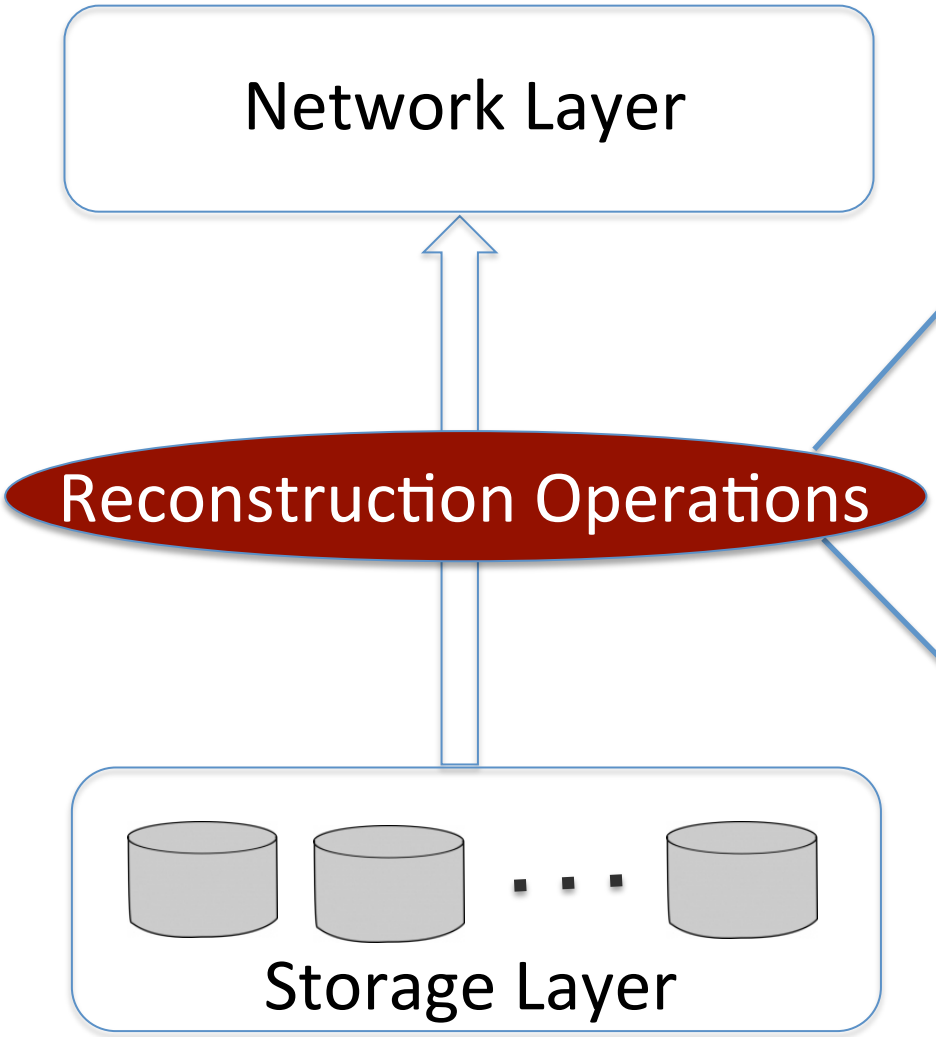
- Any 10 blocks sufficient
- Can tolerate any 4-failures

Borthakur, "HDFS and Erasure Codes (HDFS-RAID)"
Fan, Tantisiriroj, Xiao and Gibson, "DiskReduce: RAID for Data-Intensive Scalable Computing", PDSW 09

Outline

- Erasure Codes in Data Centers
  - HDFS
- Impact on the data center network
  - Problem description
- Our system: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

Impact on Data Center Network

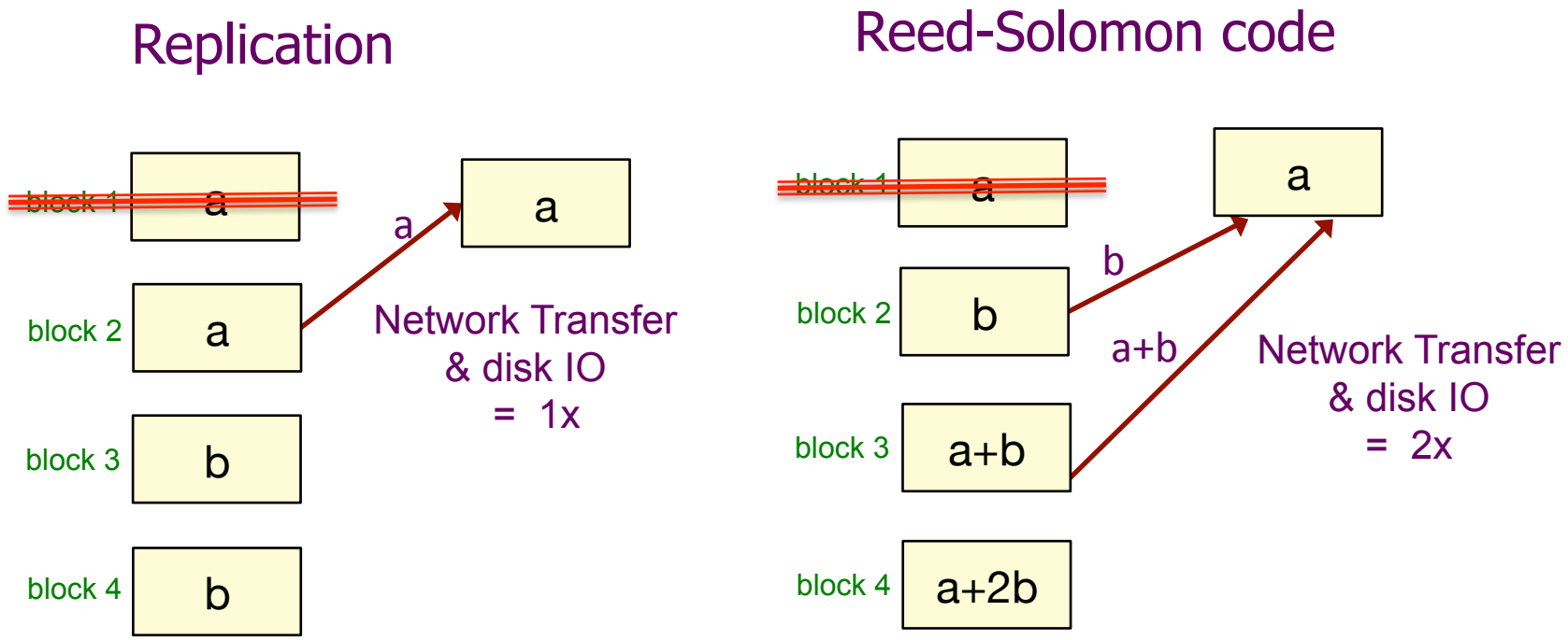


- Degraded Reads
  - requesting currently unavailable data
  - on-the-fly reconstruction
- Recovery
  - periodically replace unavailable blocks
  - to ensure desired level of reliability

Impact on Data Center Network

RS codes significantly increase network usage during reconstruction

Impact on Data Center Network

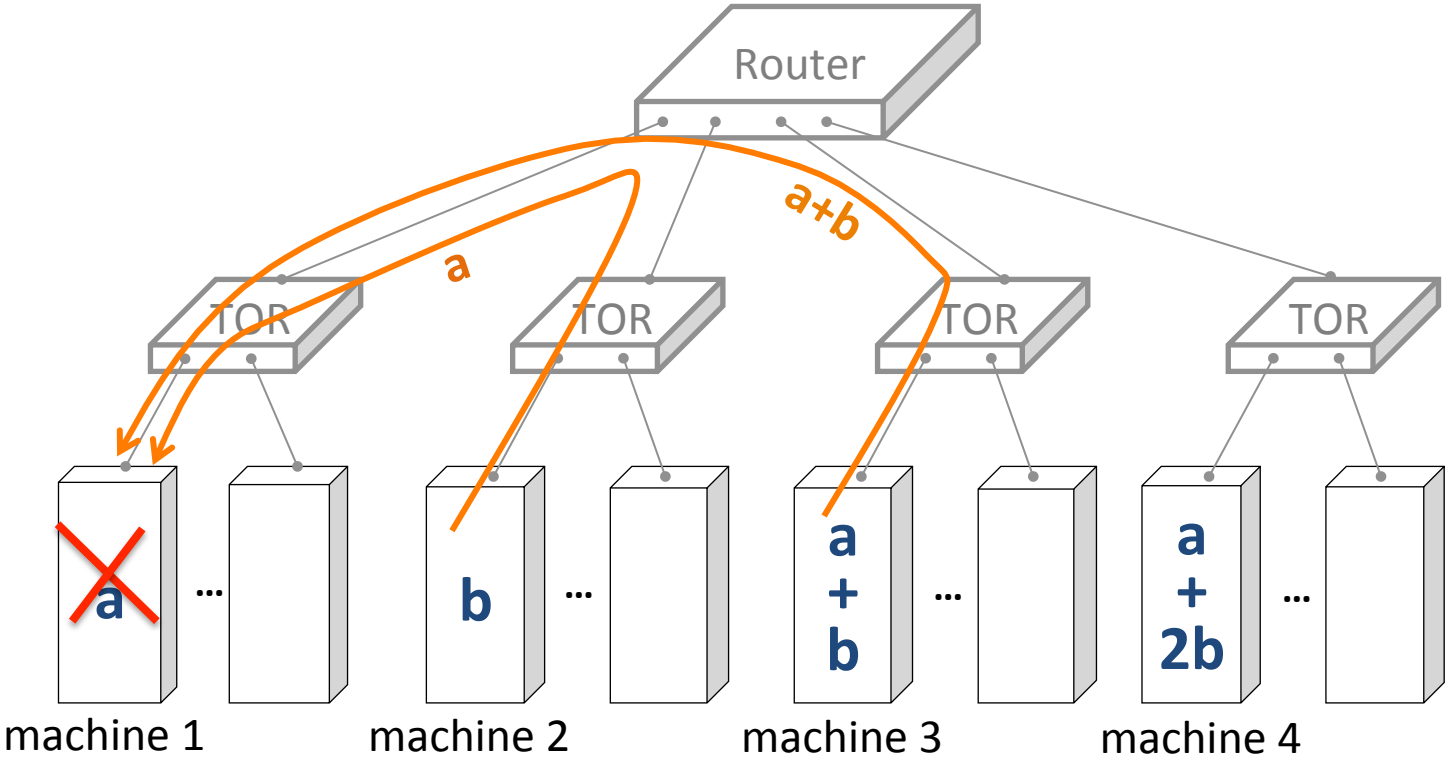


Network transfer & disk IO

= (#data-blocks) x (size of data to be reconstructed)

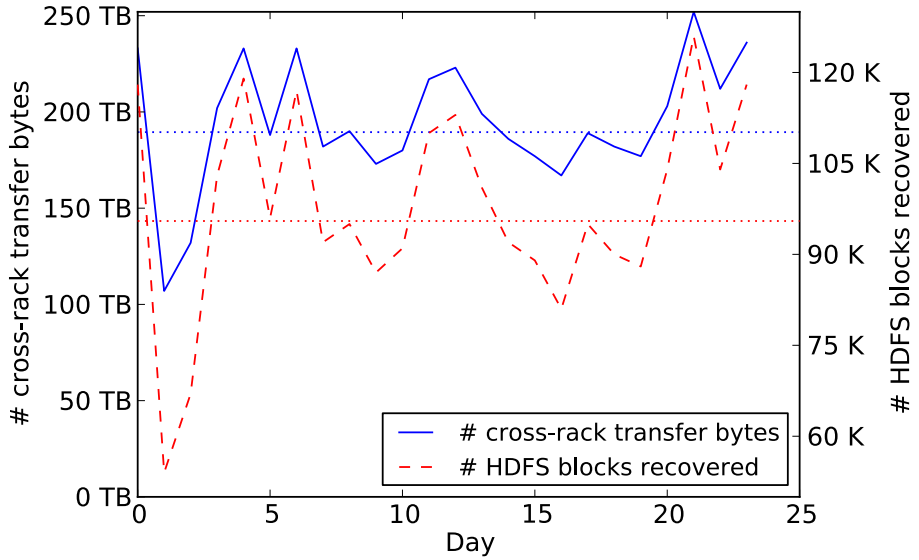
In (10, 4) RS, it is 10x

Impact on Data Center Network



Burdens the already oversubscribed Top-of-Rack and higher level switches

Impact on Data Center Network: Facebook Data Warehouse Cluster



- Multiple PB of Reed-Solomon encoded data
- Median of 180 TB transferred across racks per day for RS reconstruction ≈ 5 times that under 3-replication

Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster", Usenix HotStorage Workhsop 2013

RS codes: The Good and The Bad

- Maximum possible fault-tolerance for given storage overhead
  - storage-capacity optimal
  - ("*maximum-distance-separable*" in coding theory parlance)
- Flexibility in choice of parameters
  - Supports any number of data and parity blocks

- Not designed to handle reconstruction operations efficiently
  - negative impact on the network

RS codes: The Good and The Bad

Goal

- Maximum possible fault-tolerance for given storage overhead
  - storage-capacity optimal
  - ("*maximum-distance separable*" in coding-theory parlance)
- Flexibility in choice of parameters
  - Supports any number of data and parity blocks

Maintain

- Not designed to handle reconstruction operations efficiently
  - negative impact on the network

Improve

Goal

To build a system with:

Maintain

- Same (optimal) storage requirement and fault tolerance
- Same (complete) flexibility in choice of design parameters

Improve

- Reduced data transfer across network and reduced IO from disk during reconstruction

Hitchhiker

Is a system with:

Maintain

- Same (optimal) storage requirement and fault tolerance ☑
- Same (complete) flexibility in choice of design parameters ☑

Improve

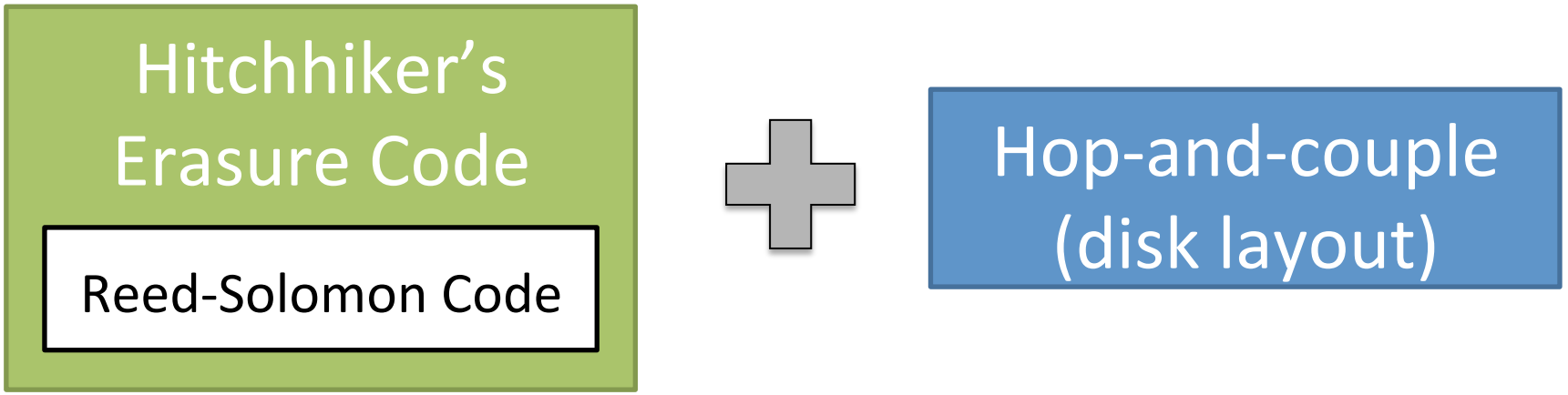
- 25 to 45% less network transfers and disk IO during reconstruction ☑

Outline

- Erasure Codes in Data Centers
  - HDFS
- Impact on the data center network
  - Problem description
- Our system: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

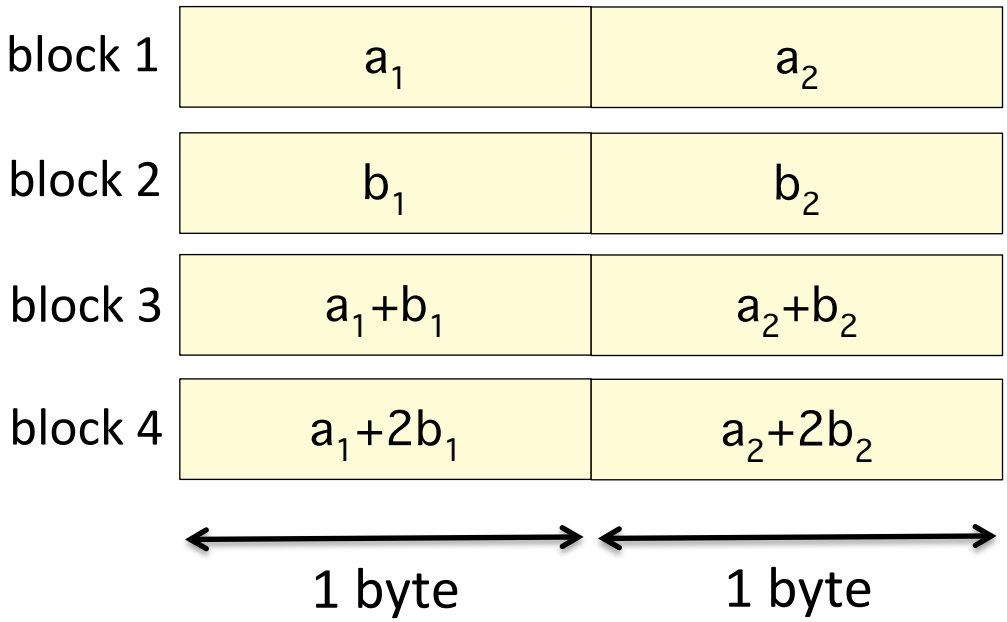
At an Abstract Level

HITCHHIKER



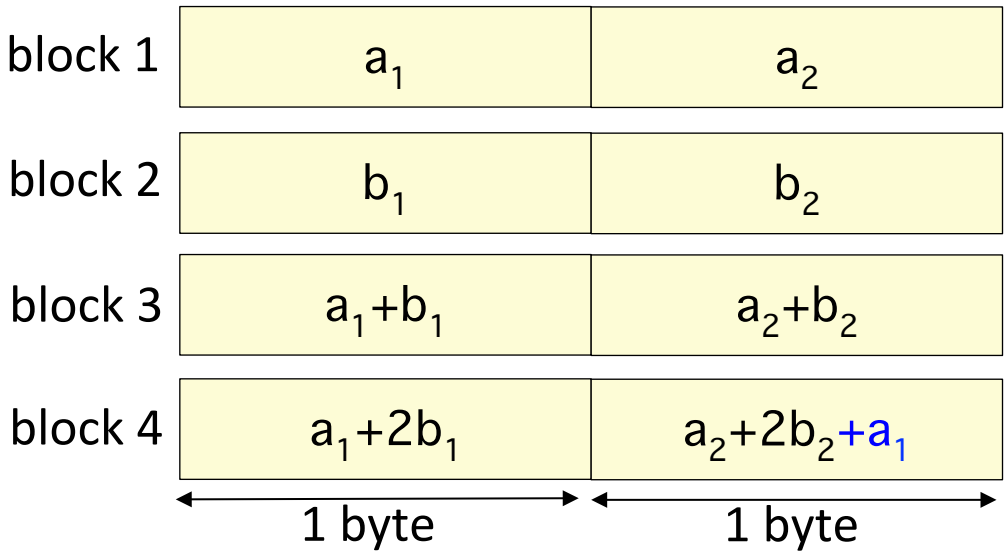
Hitchhiker's Erasure Code: Toy Example

Start with the RS code



Intermediate Code

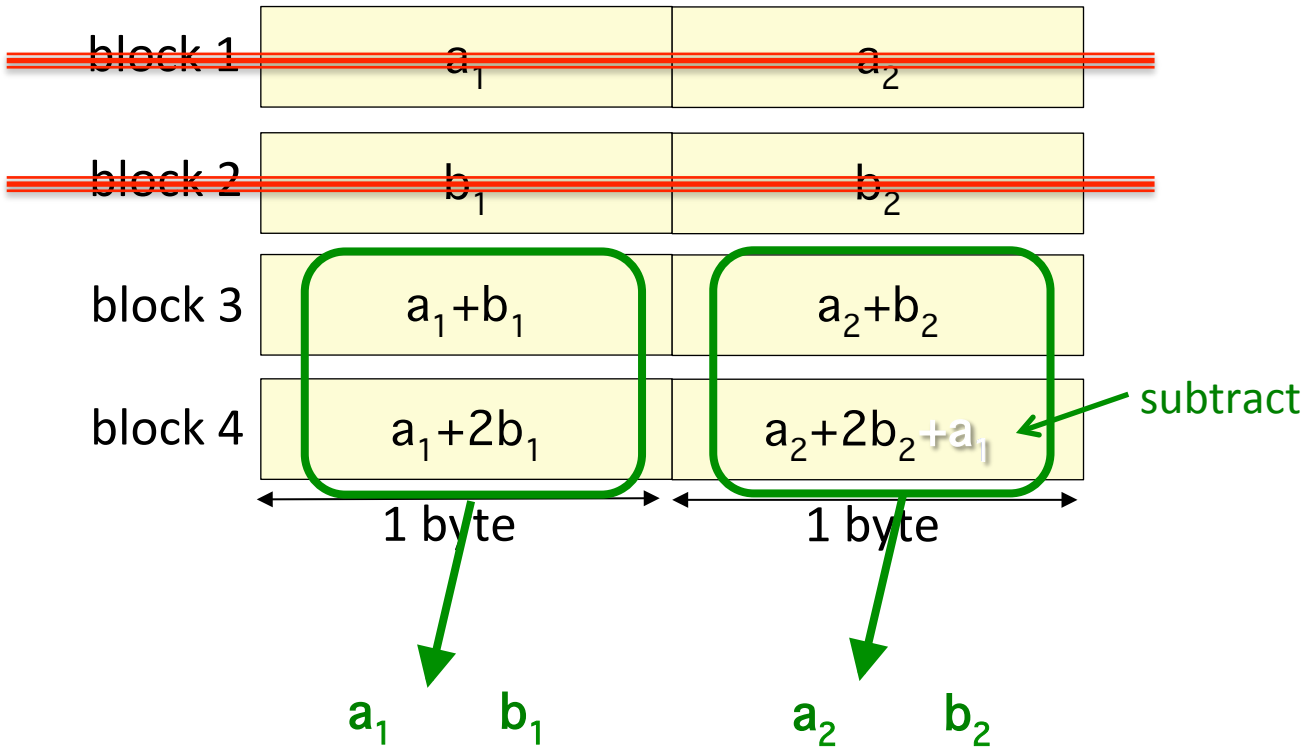
Add information from first group on to parities of the second group



No extra storage

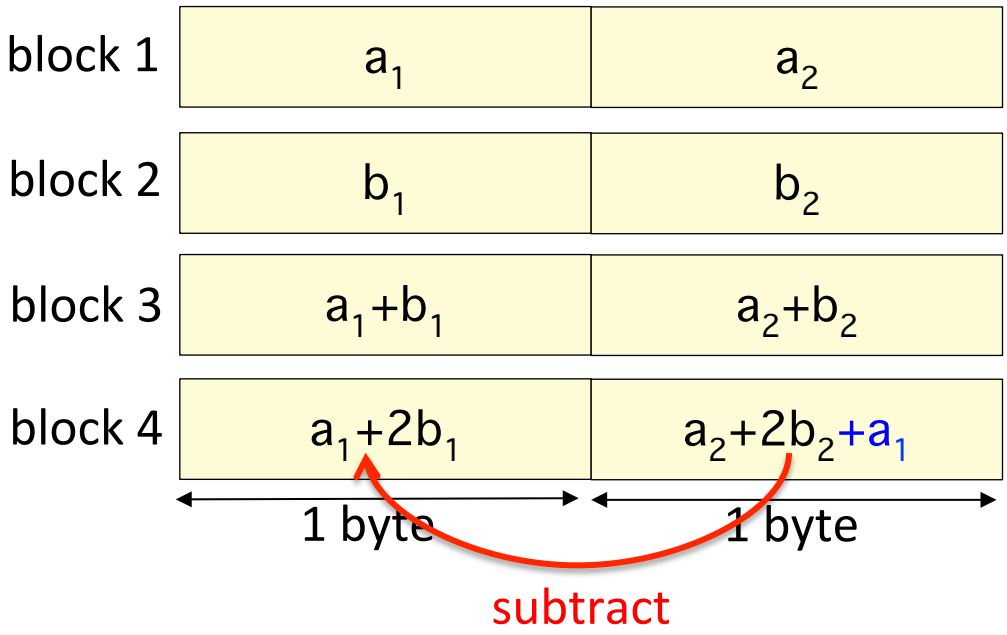
Storage-optimality of Intermediate Code

Retains failure tolerance of RS codes:
can tolerate failure of any 2 nodes



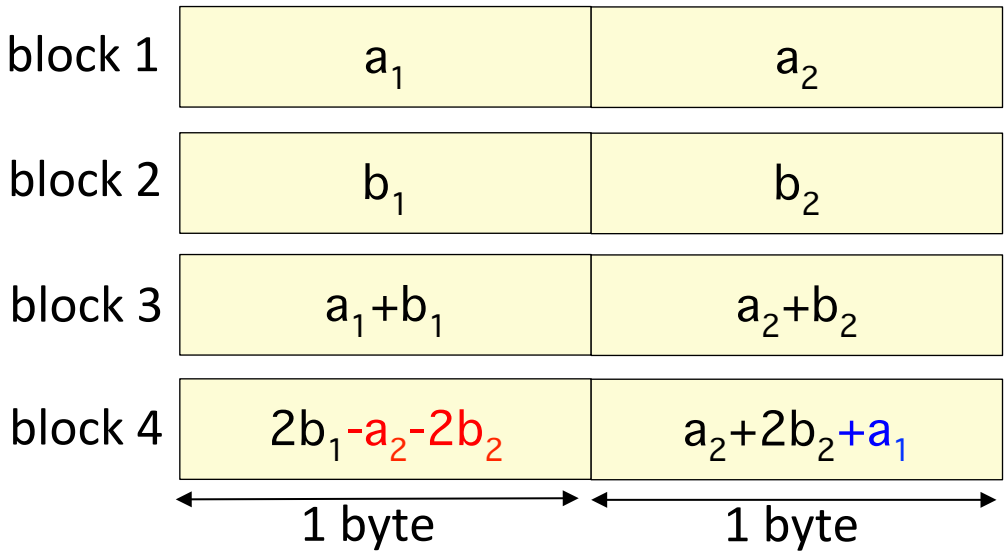
Final Code

Invertible operation *within* a block



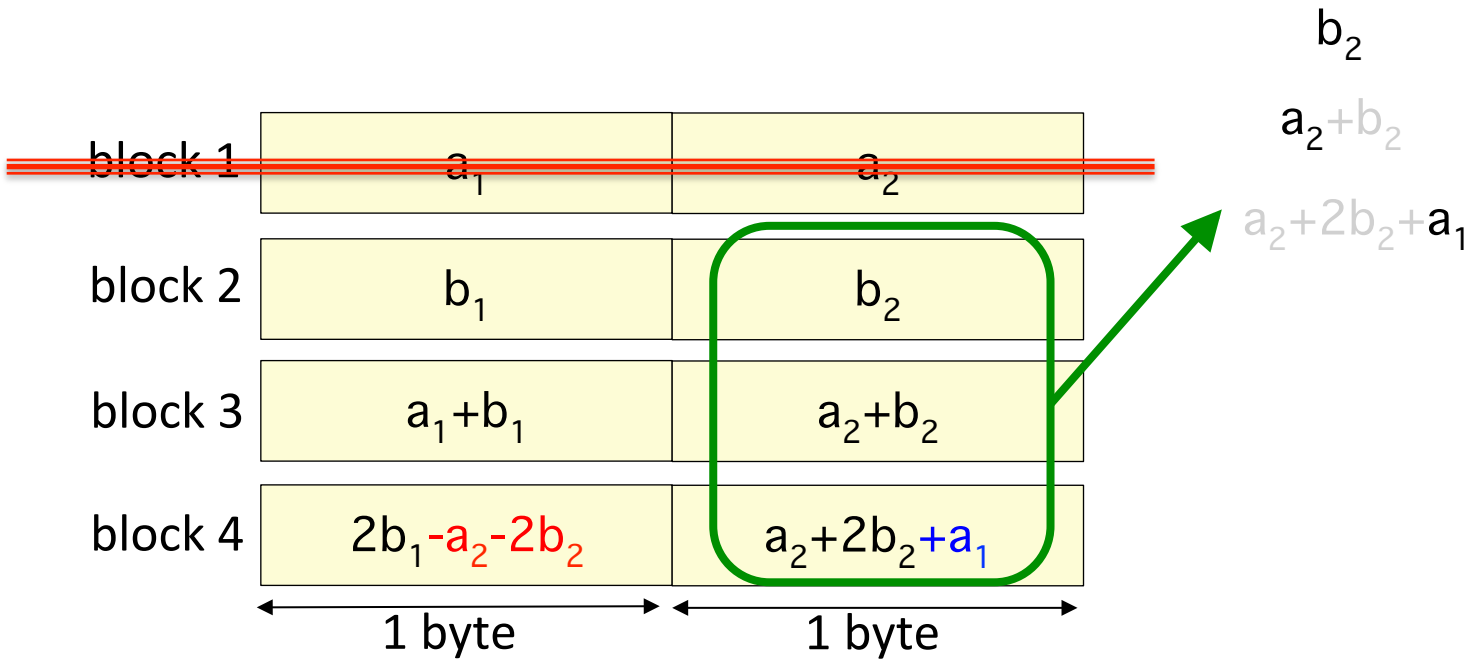
Final Code

Invertible operations *within* blocks do not change storage or fault tolerance



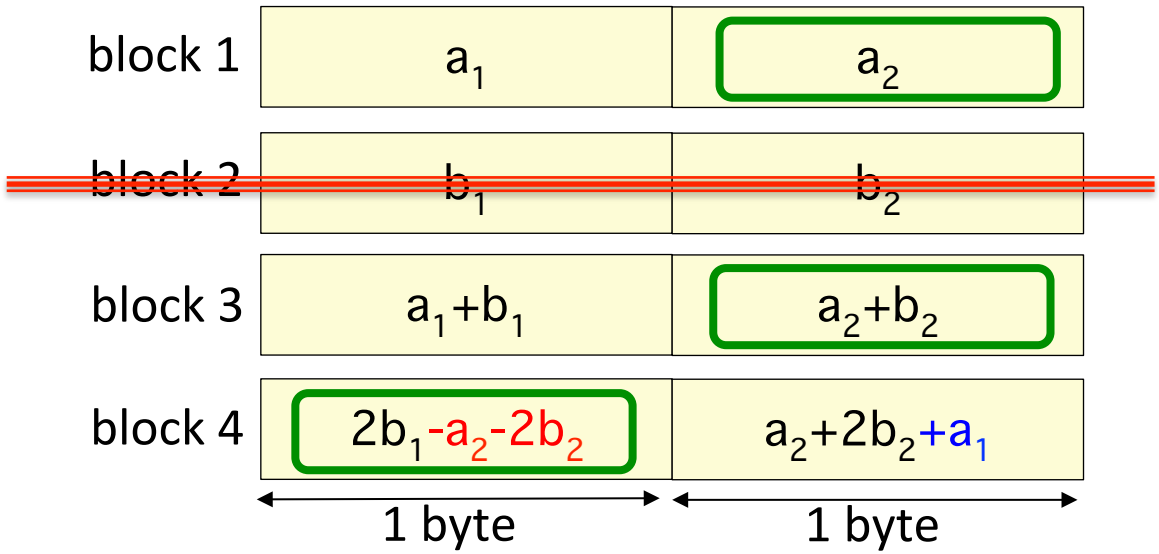
Efficient Reconstruction

Data transferred: only 3 bytes
(instead of 4 bytes as in RS)



Efficient Reconstruction

Data transferred: only 3 bytes
(instead of 4 bytes as in RS)



Hitchhiker's Erasure Code

- Builds on top of RS codes
- Uses our theoretical framework of "Piggybacking"*
- Three versions
  - XOR
  - XOR+
  - non-XOR

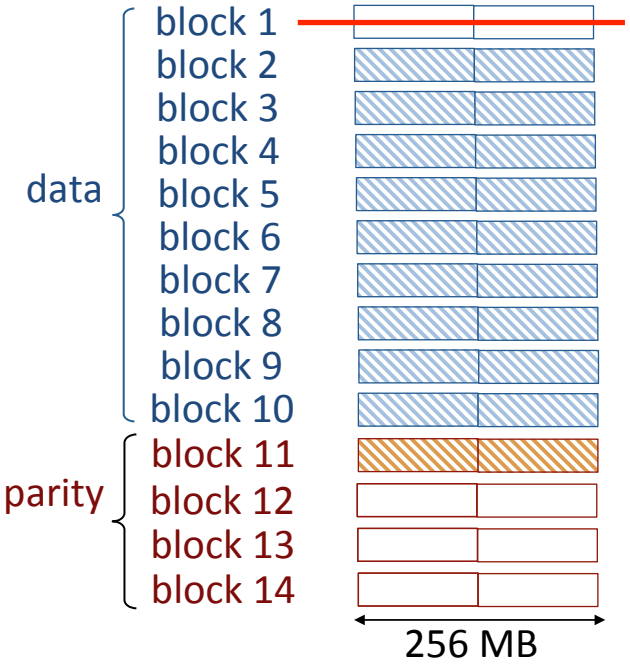
* K.V. Rashmi, Nihar Shah, K. Ramchandran, "*A Piggybacking Design Framework for Read-and Download-efficient Distributed Storage Codes*", in IEEE International Symposium on Information Theory, 2013.

Hop-and-couple (disk layout)

- Way of choosing which bytes to mix
  - couples bytes farther apart in block
  - to minimize fragmentation of reads during reconstruction
- Translate savings in network-transfer to savings in disk-IO as well
  - By making reads contiguous

RS vs Hitchhiker from the Network's Perspective…

Data Transfer during Reconstruction in RS-based System

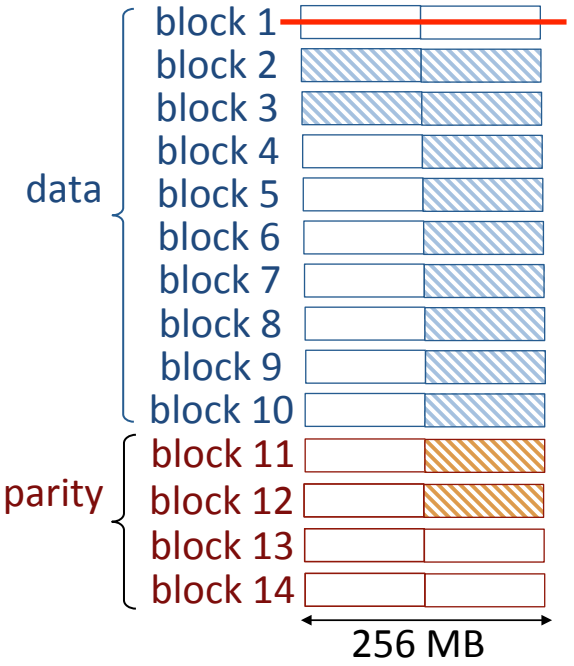


Transfer: 10 full blocks

Connect to 10 machines

Data Transfer during Reconstruction in Hitchhiker

Reconstruction of data blocks 1-9:

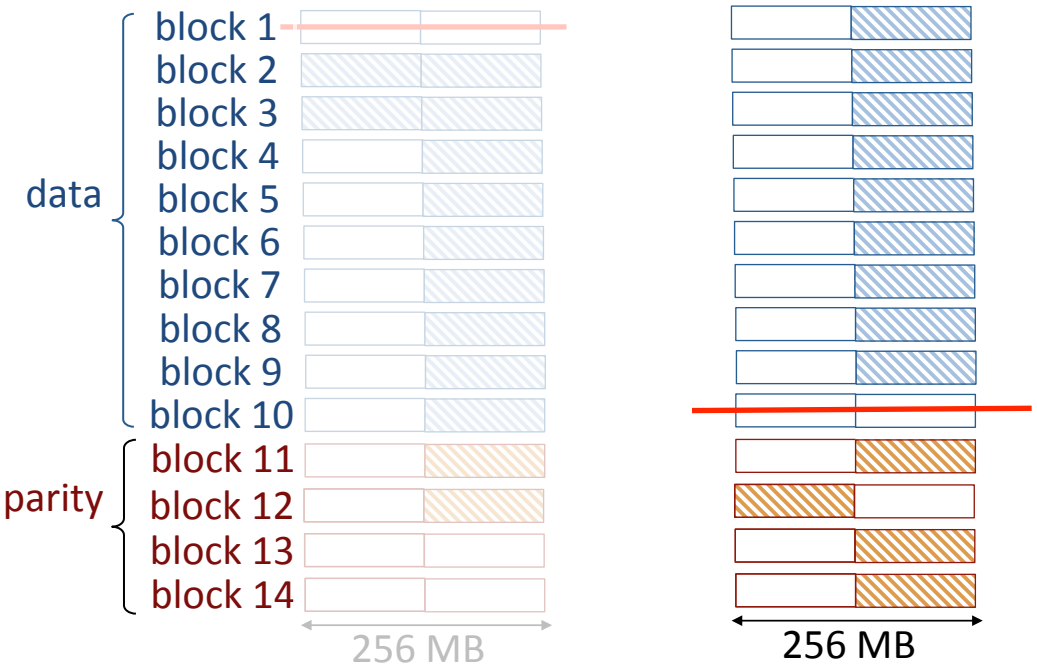


Transfer: 2 full blocks + 9 half blocks (= 6.5 blocks total)

Connect to 11 machines

Data Transfer during Reconstruction in Hitchhiker

Reconstruction of block 10:



Transfer: 13 half blocks (= 6.5 blocks total)

Connect to 13 machines

Outline

- Erasure Codes in Data Centers
  - HDFS
- Impact on the data center network
  - Problem description
- Our system: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

Implementation & Evaluation Setup (1)

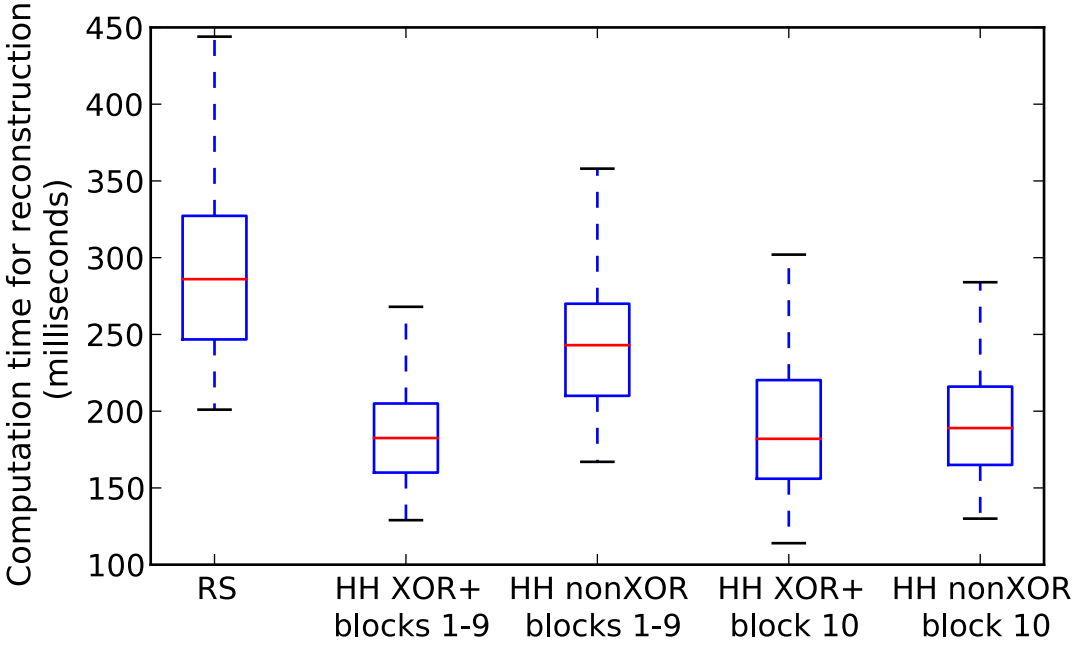
- Implemented on top of HDFS-RAID
  - erasure coding module in HDFS based on RS
  - used in the Facebook data warehouse cluster

- Deployed and tested on a 60 machine test cluster at Facebook
  - verified 35% reduction in the network transfers during reconstruction

Implementation & Evaluation Setup (2)

- Evaluation of timing metrics on the Facebook data warehouse cluster in production
  - under real-time production traffic and workloads
  - using Map-Reduce to run encoding and reconstruction jobs, just as HDFS-RAID

Decoding Time



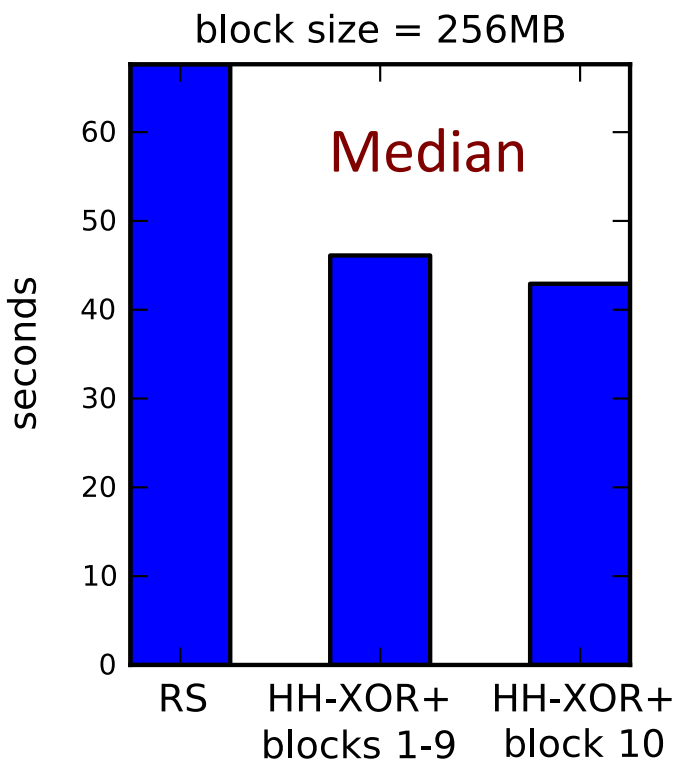
36% reduction

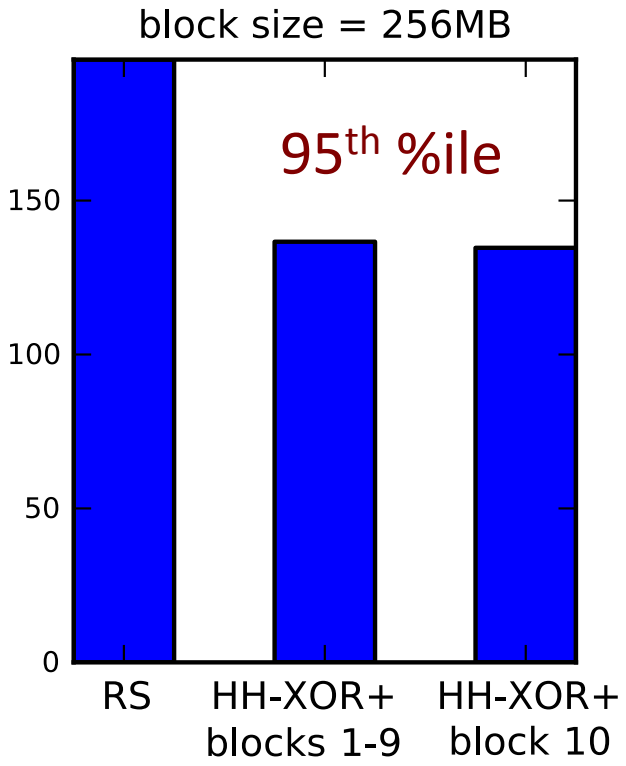
- RS decoding on only half portion of the blocks
- Faster computation for degraded reads and recovery

- XOR versions: 25% lesser than non-XOR

Read & Transfer Time

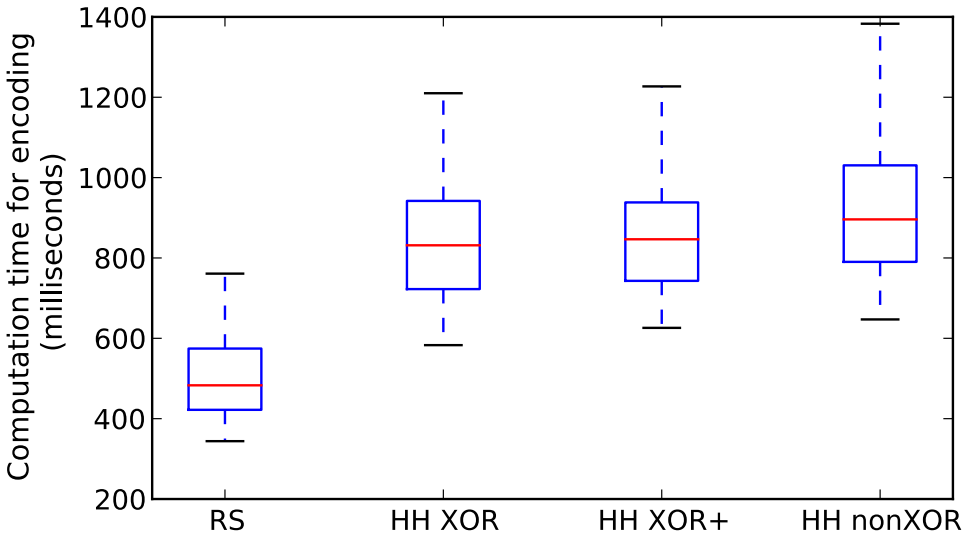
System	Data transfer	Connectivity (#machines)
RS	2.56 GB	10
HH blocks 1-9	1.67 GB	11
HH block 10	1.67 GB	13





- Read & transfer time 30% lower in Hitchhiker (HH)
- Similar reduction for other block sizes as well

Encoding Time



72% higher

Benefits outweigh higher encoding cost in many systems (e.g., HDFS):

- encoding is one time operation
- often run as a background job
- does not fall along any critical path

Outline

- Impact on the data center network
  - Problem description
- Our system: “Hitchhiker”
- Implementation and evaluation
  - Facebook data warehouse cluster
- Literature

Existing Systems

- Need additional storage
  - Huang et al. (Windows Azure) 2012, Sathiamoorthy et al. (Xorbas) 2013, Esmaili et al. (CORE) 2013
    - Add additional parities to reduce download
  - Hu et al. (NCFS 2011)

- Highly restricted parameters
  - Khan et al. (Rotated-RS) 2012: #parity ≤ 3
  - Xiang et al., Wang et al. 2010, Hu (NCCloud) et al. 2012: #parity ≤ 2
  - Hitchhiker performs as good or better for these restricted settings as well

Hitchhiker: Summary

Code metrics:

Storage requirement	Same (optimal)
Supported parameters	All
Fault tolerance	Same (optimal)

Reconstruction:

Network transfers	35% less
Disk IO	35% less
Data read and transfer time (median)	31.8% less
Data read and transfer time (95th %ile)	30.2% less
Computation time (median)	36.1% less

Encoding:

Encoding time (median)	72.1% more

Thanks!

Backup Slides

Hop-and-Couple

- Technique to pair bytes under Hitchhiker's erasure code
- Makes disk reads during reconstruction contiguous