A Solution to the Network Challenges of Data Recovery in Erasure-coded Distributed Storage Systems:
A Study on the Facebook Warehouse Cluster

Berkeley
University of California

facebook

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

Outline

- Introduction: Erasure coding in data centers
  - Low storage, high fault-tolerance
  - High download & disk IO during recovery
- Measurements from Facebook warehouse cluster in production
- Proposed alternative: Piggybacked-RS codes
  - Same storage overhead & fault tolerance
  - 30% reduction in download & disk IO

Outline

- Introduction: Erasure coding in data centers
  - Low storage, high fault-tolerance
  - High download & disk IO during recovery
- Measurements from Facebook warehouse cluster in production
- Proposed alternative: Piggybacked-RS codes
  - Same storage overhead & fault tolerance
  - 30% reduction in download & disk IO

Need for Redundant Storage

- Frequent unavailability in data-centers
  - commodity components fail frequently
  - software glitches, maintenance shutdowns, power failures

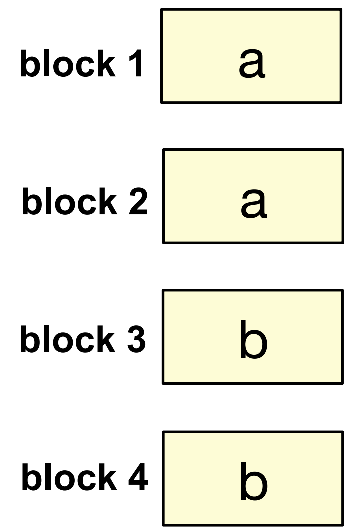
- Redundancy gives more reliability and availability

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

Popular approach: Replication

- Multiple copies of data across machines
- E.g., GFS, HDFS store 3 replicas by default

- Typically stored across different racks



a, b: data blocks

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

Petabyte Scale data: Replication expensive

- Moderately sized data: storage is cheap

  ⇒ replication viable

- Multiple tens of PBs

  ⇒ aggregate storage no longer cheap

  ⇒ replication is expensive

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

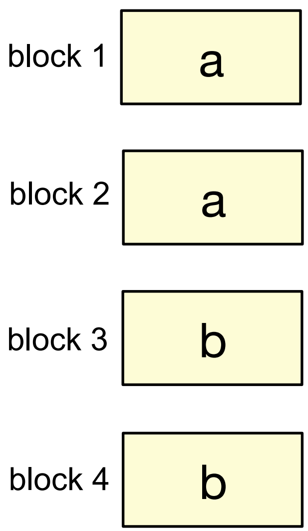
Erasure Codes

	Replication	Reed-Solomon (RS) code	
block 1	a	a	data blocks
block 2	a	b	data blocks
block 3	b	a+b	parity blocks
block 4	b	a+2b	parity blocks
Redundancy	2x	2x	

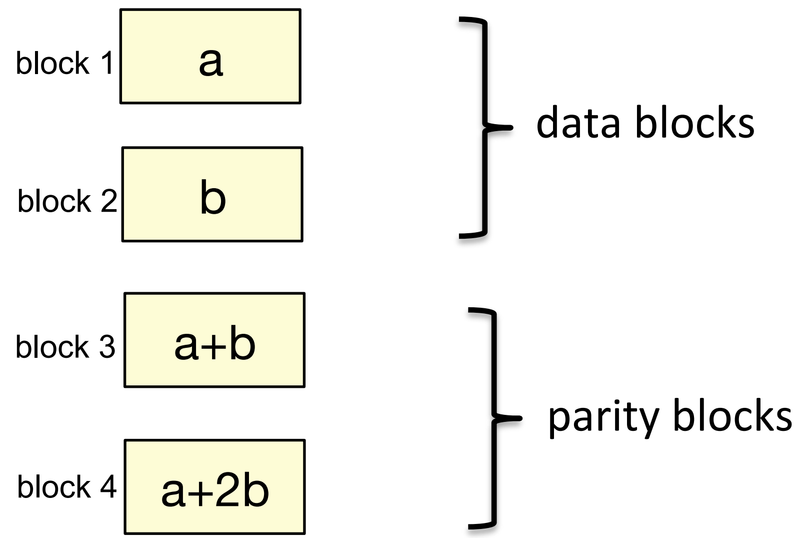
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Erasure Codes

Replication



Reed-Solomon (RS) code

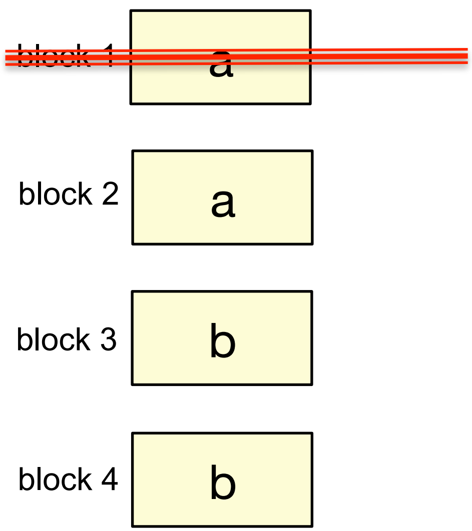


	Replication	Reed-Solomon (RS) code
Redundancy	2x	2x
First order comparison:	tolerates any one failure	tolerates any two failures

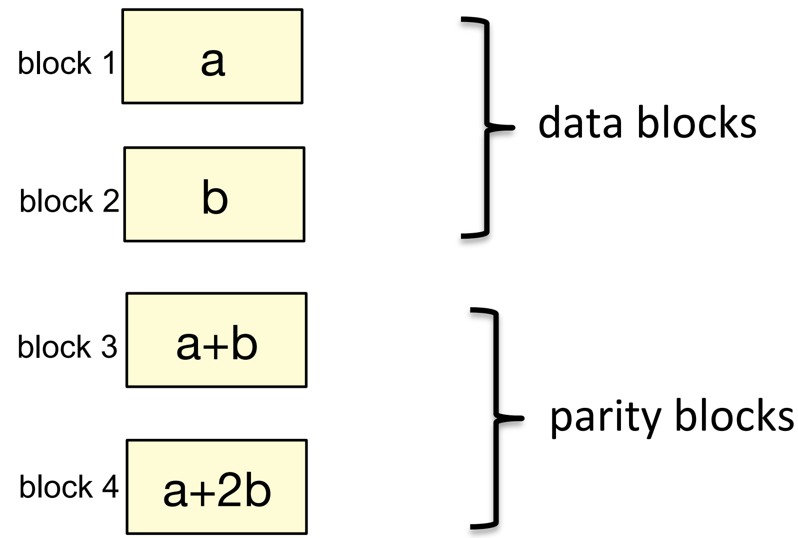
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Erasure Codes

Replication



Reed-Solomon (RS) code

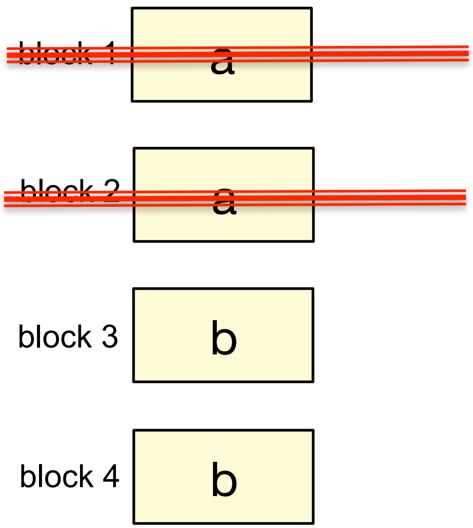


	Replication	Reed-Solomon (RS) code
Redundancy	2x	2x
First order comparison:	tolerates any one failure	tolerates any two failures

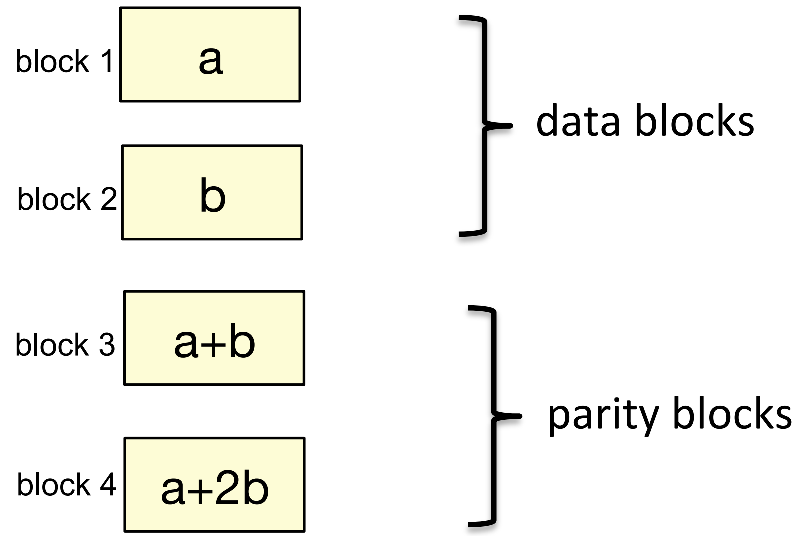
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Erasure Codes

Replication



Reed-Solomon (RS) code

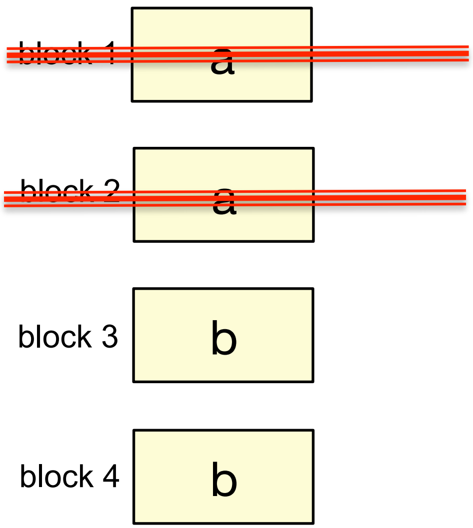


	Replication	Reed-Solomon (RS) code
Redundancy	2x	2x
First order comparison:	tolerates any one failure	tolerates any two failures

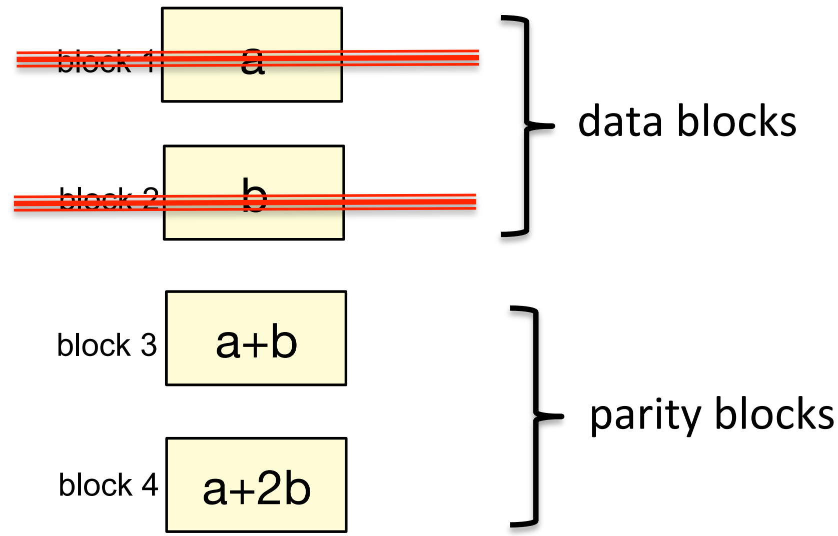
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Erasure Codes

Replication



Reed-Solomon (RS) code

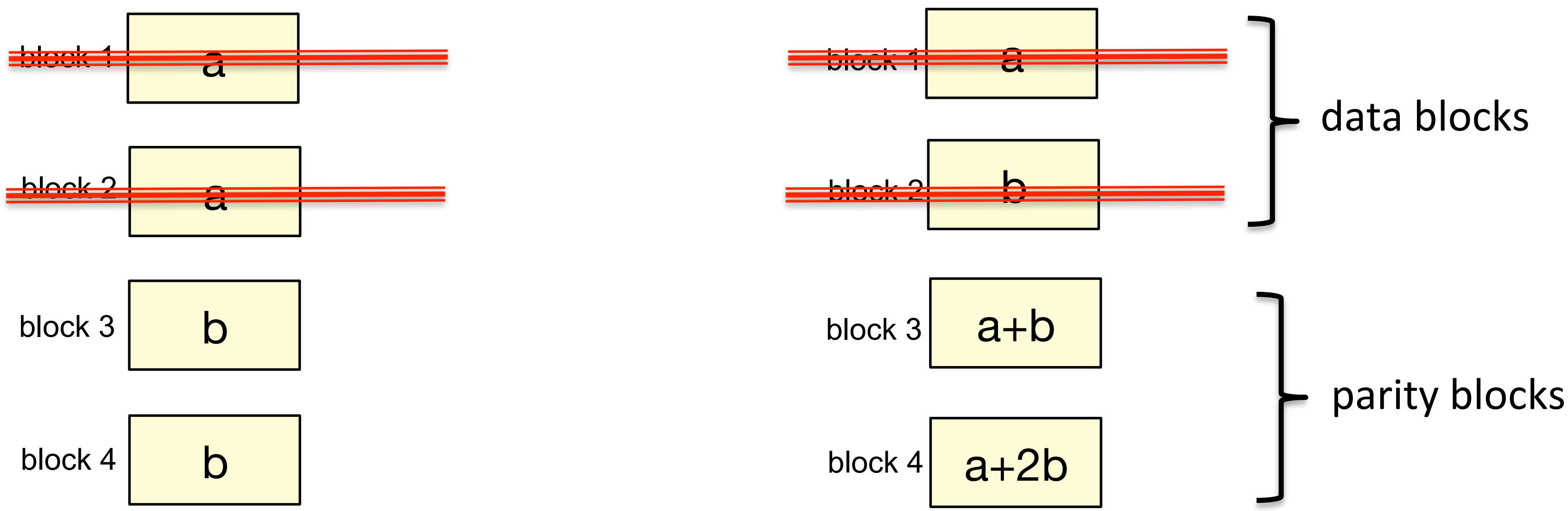


	Replication	Reed-Solomon (RS) code
Redundancy	2x	2x
First order comparison:	tolerates any one failure	tolerates any two failures

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

Erasure Codes

	Replication	Reed-Solomon (RS) code



	Replication	Reed-Solomon (RS) code
Redundancy	2x	2x
First order comparison:	tolerates any one failure	tolerates any two failures
In general:	lower MTTDL, high storage requirement	order of magnitude higher MTTDL with much lesser storage

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

Erasure Codes

Using RS codes instead of 3-replication on less-frequently accessed data has led to savings of multiple Petabytes in the Facebook Warehouse cluster

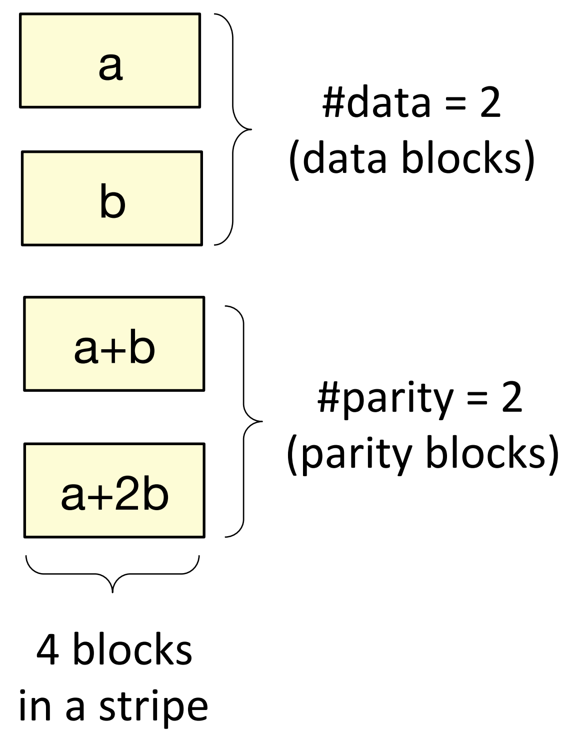
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Reed-Solomon (RS) Codes

- (#data, #parity) RS code:
  - tolerates failure of any #parity blocks
  - these (#data + #parity) blocks constitute a "stripe"

- Facebook warehouse cluster uses a (10, 4) RS code

Example: (2, 2) RS code



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

Why RS codes ?

- Maximum possible fault-tolerance for storage overhead
  - storage-capacity optimal
  - "maximum-distance-separable (MDS)" (in coding theory parlance)

- Flexibility in choice of parameters
  - Supports any #data and #parity

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

Why RS codes ?

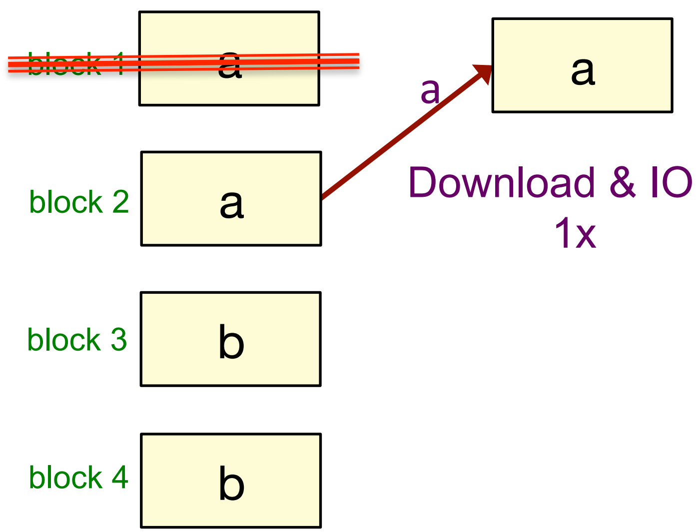
- Maximum possible fault-tolerance for storage overhead
  - storage-capacity optimal
  - "maximum-distance-separable (MDS)" (in coding theory parlance)
- Flexibility in choice of parameters
  - Supports any #data and #parity

However…
result in increased download and disk IO during data recovery

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

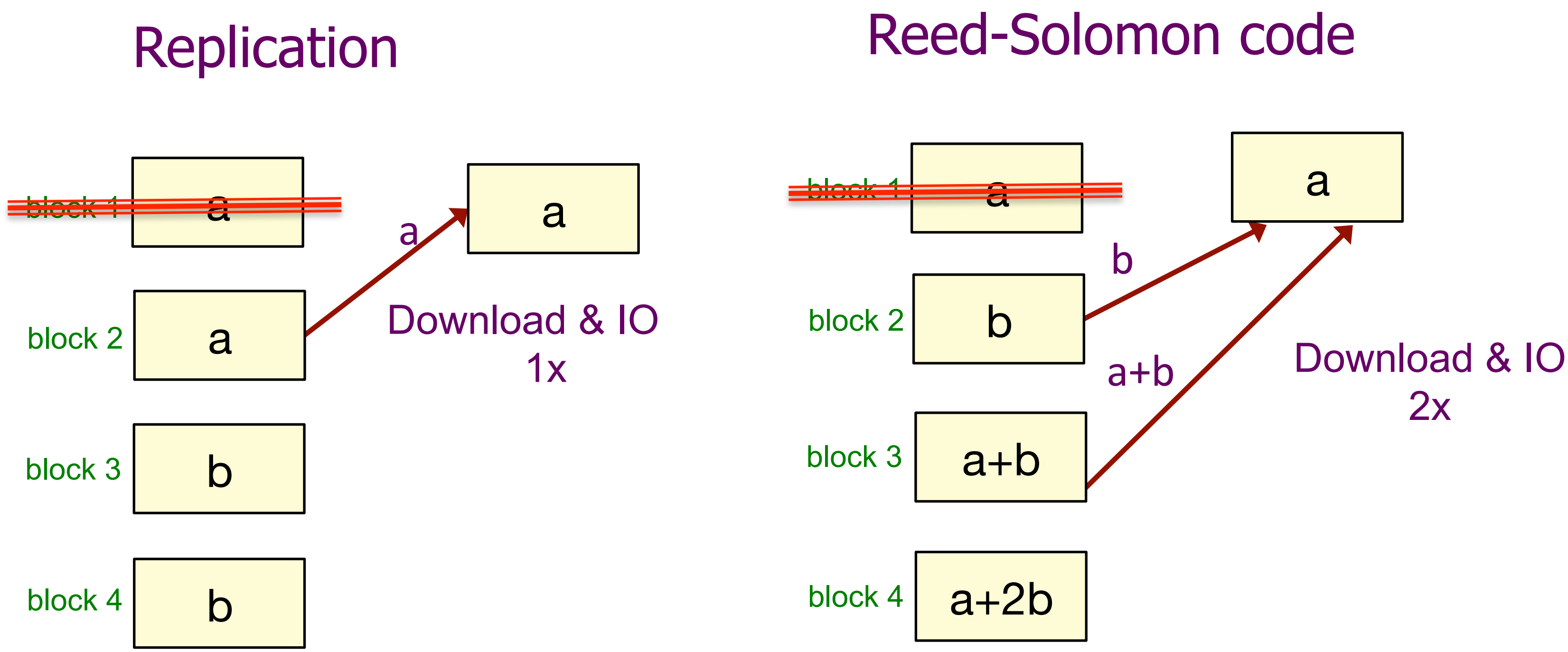
Data Recovery: Increased download & disk IO

Replication



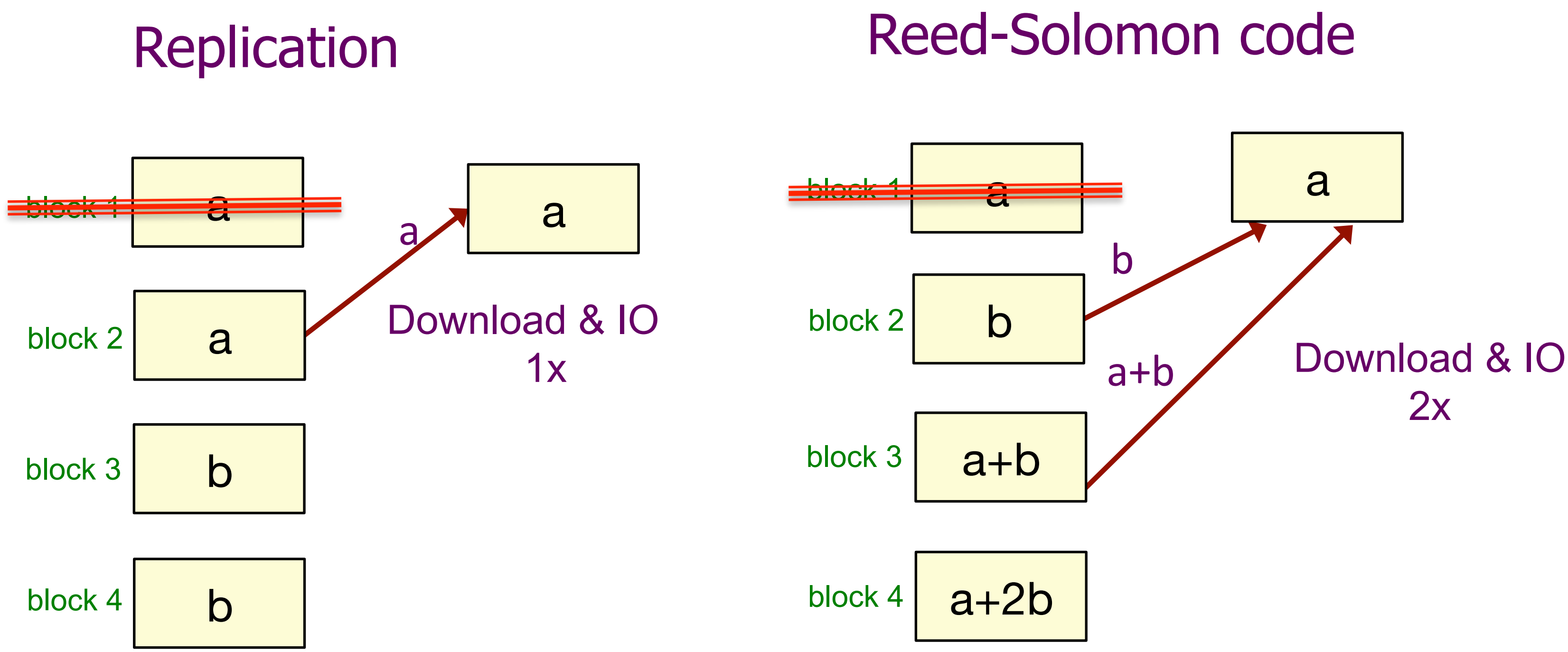
Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Data Recovery: Increased download & disk IO



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

Data Recovery: Increased download & disk IO

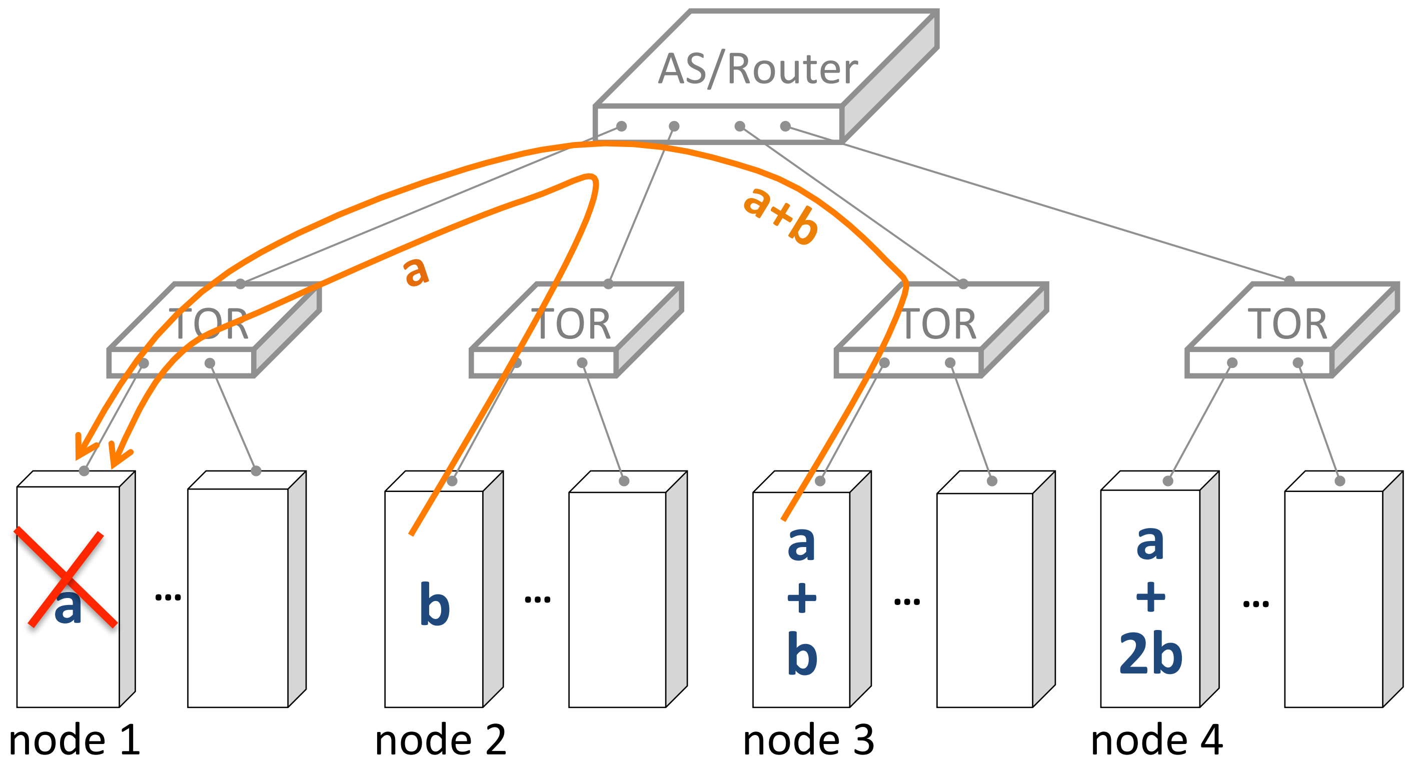


In general…

Download & IO required = #data x (size of data to be recovered)

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

Data Recovery: Burden on TOR switches



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

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

Outline

- Introduction: Erasure coding in data centers
  - Low storage, high fault-tolerance
  - High download & disk IO during recovery
- Measurements from Facebook warehouse cluster in production
- Proposed alternative: Piggybacked-RS codes
  - Same storage overhead & fault tolerance
  - 30% reduction in download & disk IO

Brief System Description

- HDFS cluster with multiple thousands of nodes
- Multiple tens of PBs and growing
- Data immutable until deleted

Reducing storage requirements is of high importance

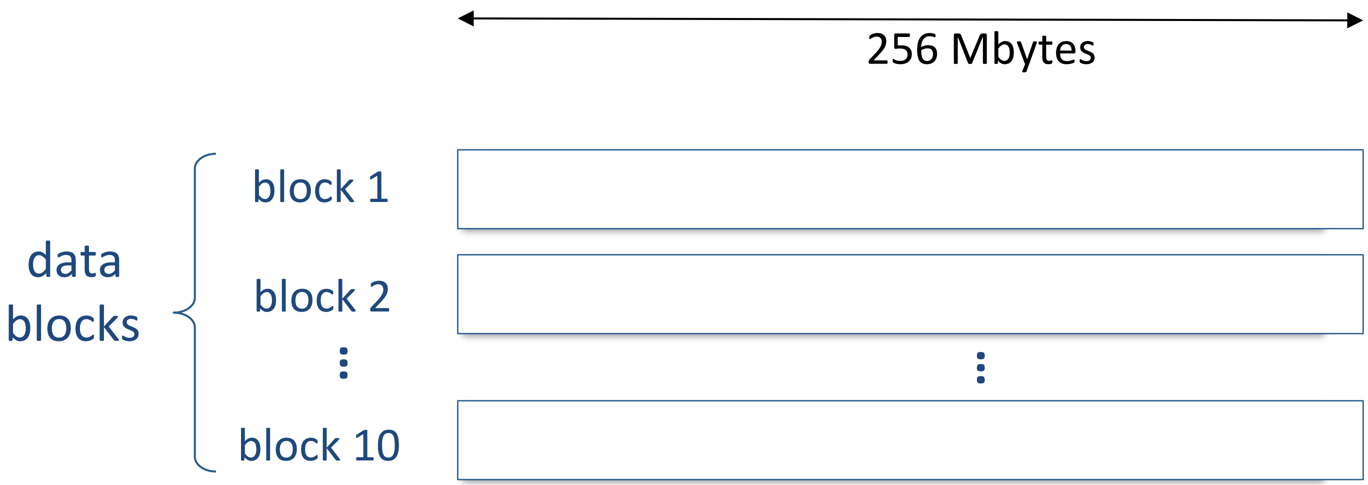
Brief System Description

- HDFS cluster with multiple thousands of nodes
- Multiple tens of PBs and growing
- Data immutable until deleted

Reducing storage requirements is of high importance

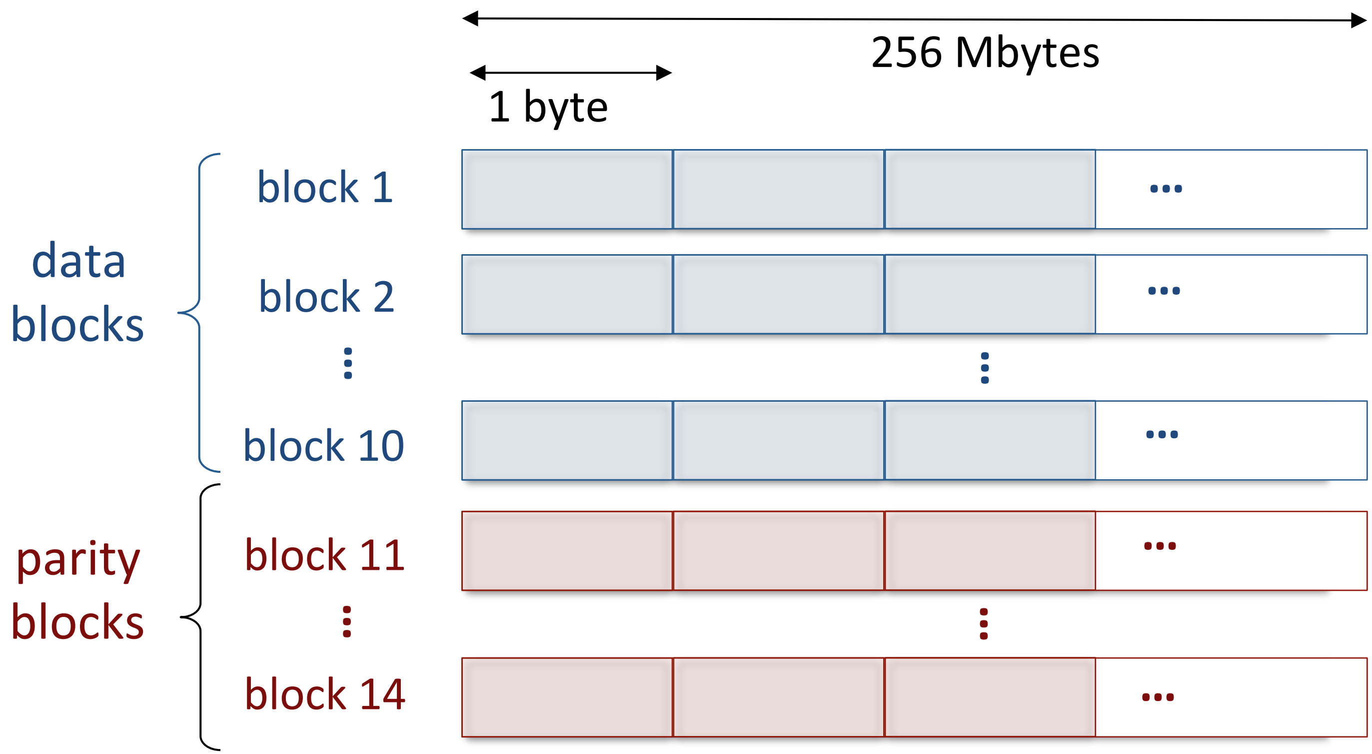
- Uses (10, 4) RS code to reduce storage requirements
  - on less-frequently accessed data
- Multiple PBs of RS coded data

Brief System Description



Rashmi et al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster”

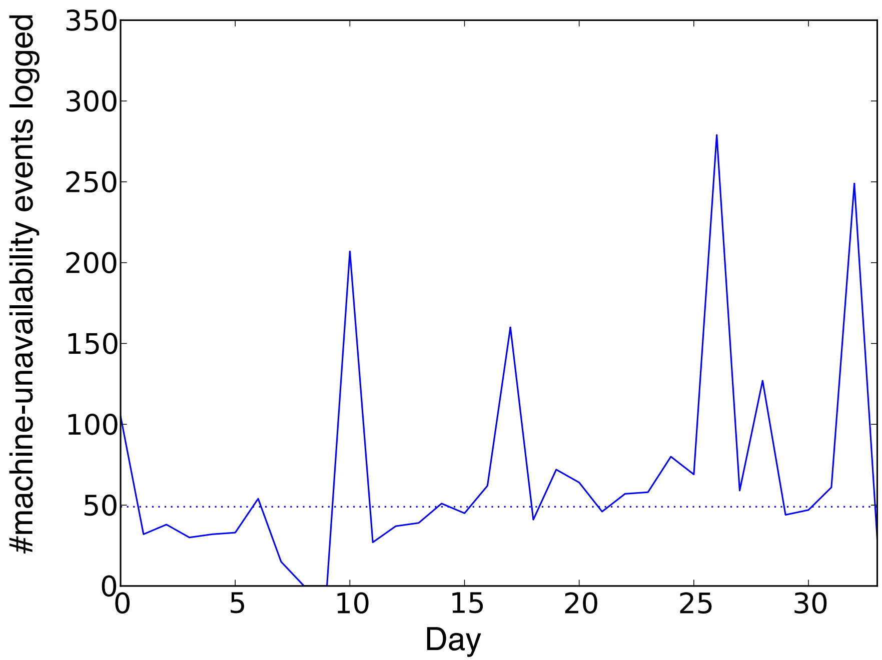
Brief System Description



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

Machine Unavailability Events

- From HDFS Name-Node logs
- Logged when no heart-beat for > 15min
- Blocks marked unavailable, periodic recovery process



Median of ≈50 machine-unavailability events logged per day

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

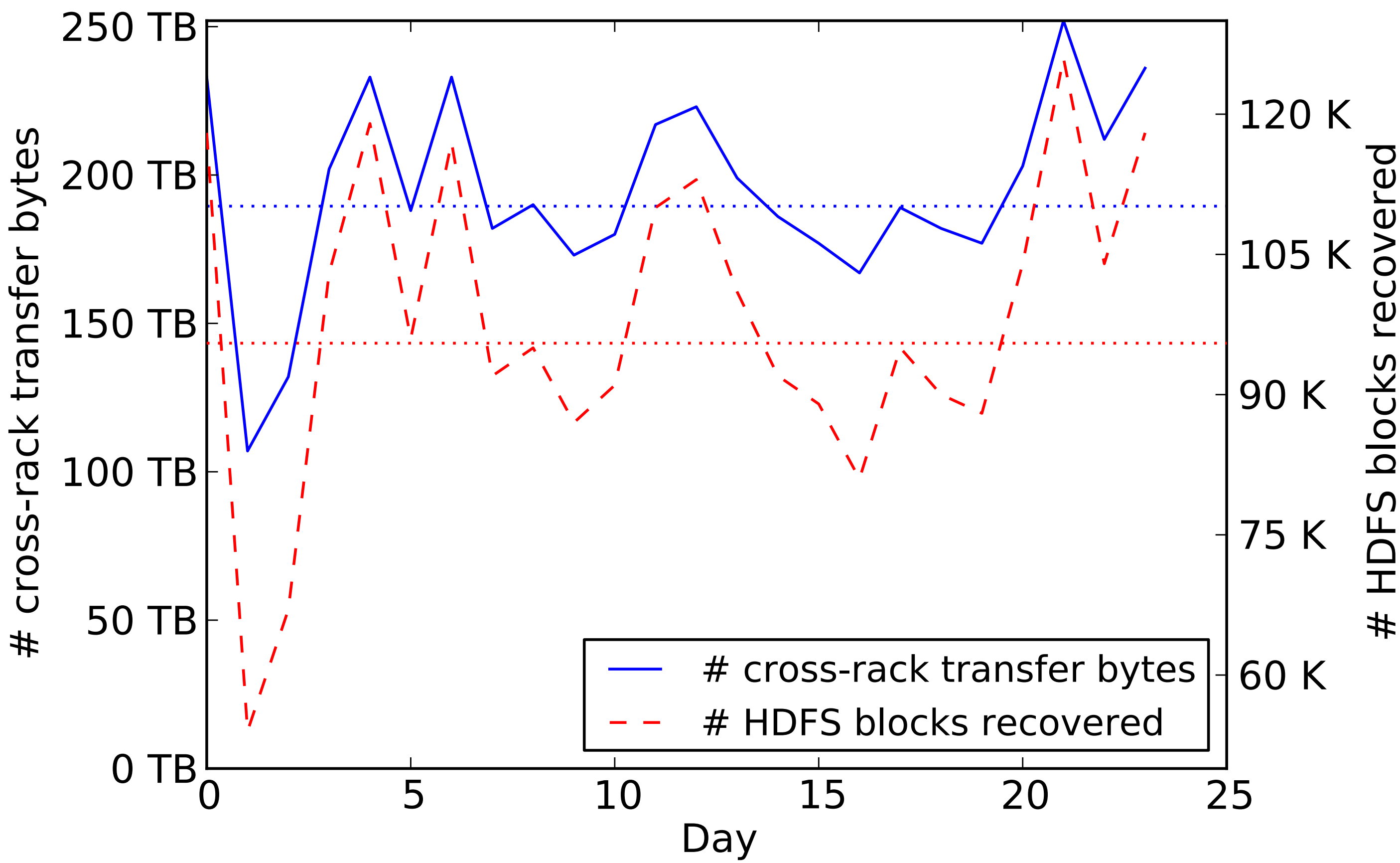
Missing blocks per stripe

# blocks missing in stripe	% of stripes with missing blocks
1	98.08
2	1.87
3	0.036
4	9×10^{-6}
≥ 5	9×10^{-9}

Dominant scenario: Single block recovery

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

#Blocks Recovered & Cross-rack Transfers



- Median of 180 TB transferred across racks per day for recovery operations
- Around 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"

Outline

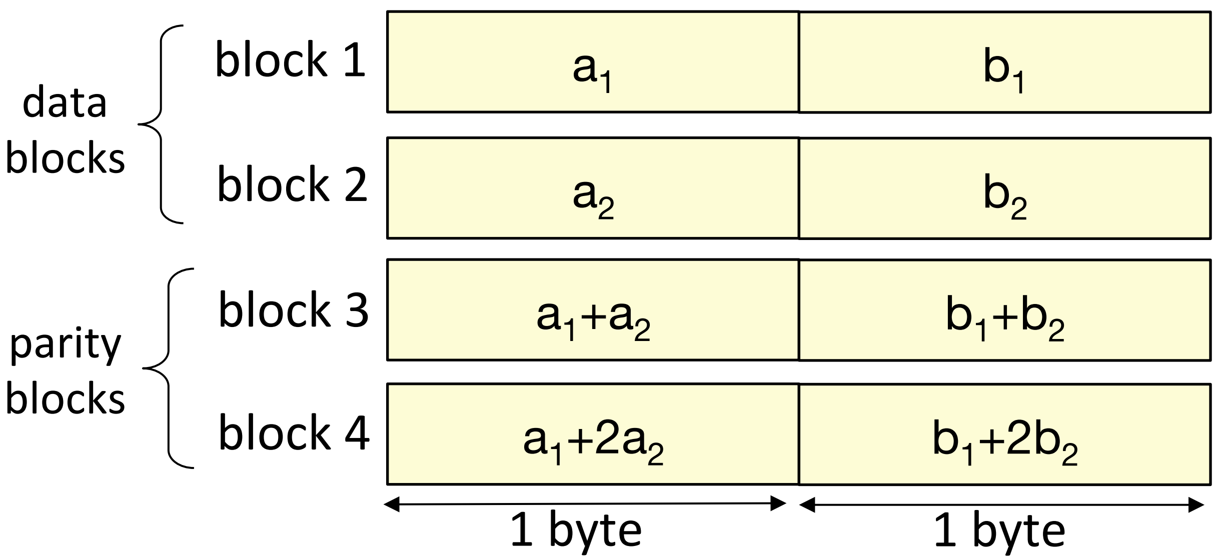
- Introduction: Erasure coding in data centers
  - Low storage, high fault-tolerance
  - High download & disk IO during recovery

- Measurements from Facebook warehouse cluster in production

- Proposed alternative: Piggybacked-RS codes
  - Same storage overhead & fault tolerance
  - 30% reduction in download & disk IO

Piggybacking: Toy Example

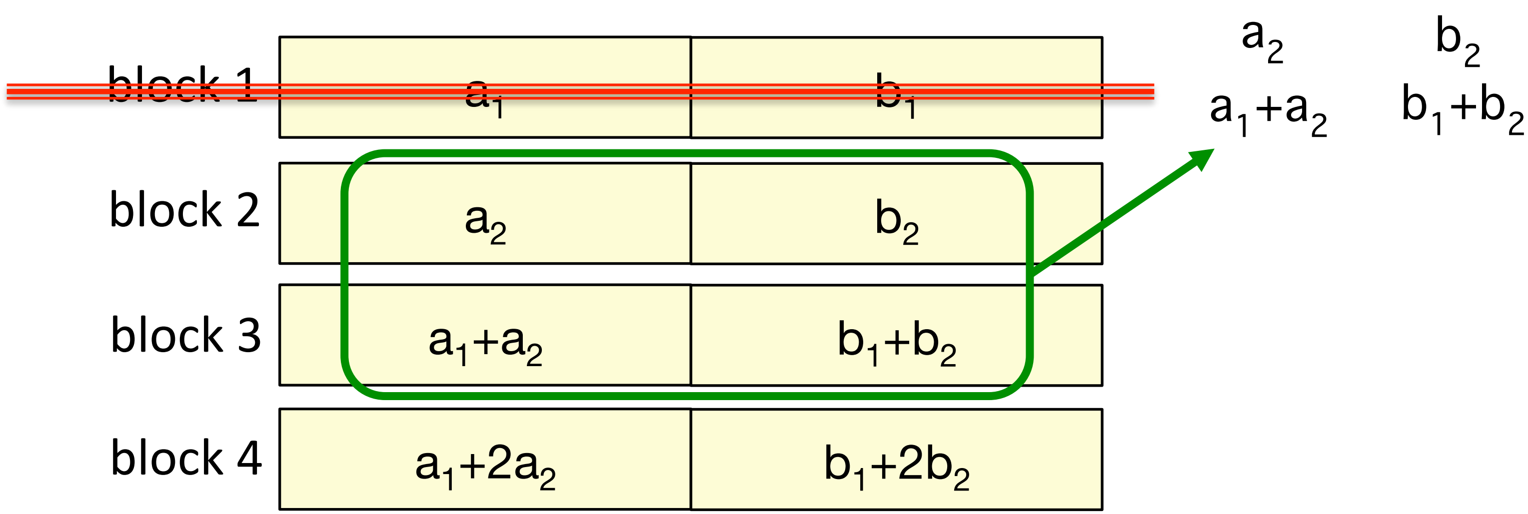
Step 1: Take a (2, 2) Reed-Solomon code



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

Piggybacking: Toy Example

(In (2,2) RS code: recovery download & IO = 4 bytes)



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

Piggybacking: Toy Example

Step 2: Add 'piggybacks' to parity nodes

block 1	a_1	b_1
block 2	a_2	b_2
block 3	a_1+a_2	b_1+b_2
block 4	a_1+2a_2	$b_1+2b_2+a_1$

No additional storage!

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

Fault-Tolerance (toy example)

Same fault tolerance as RS code:
can tolerate failure of any 2 nodes

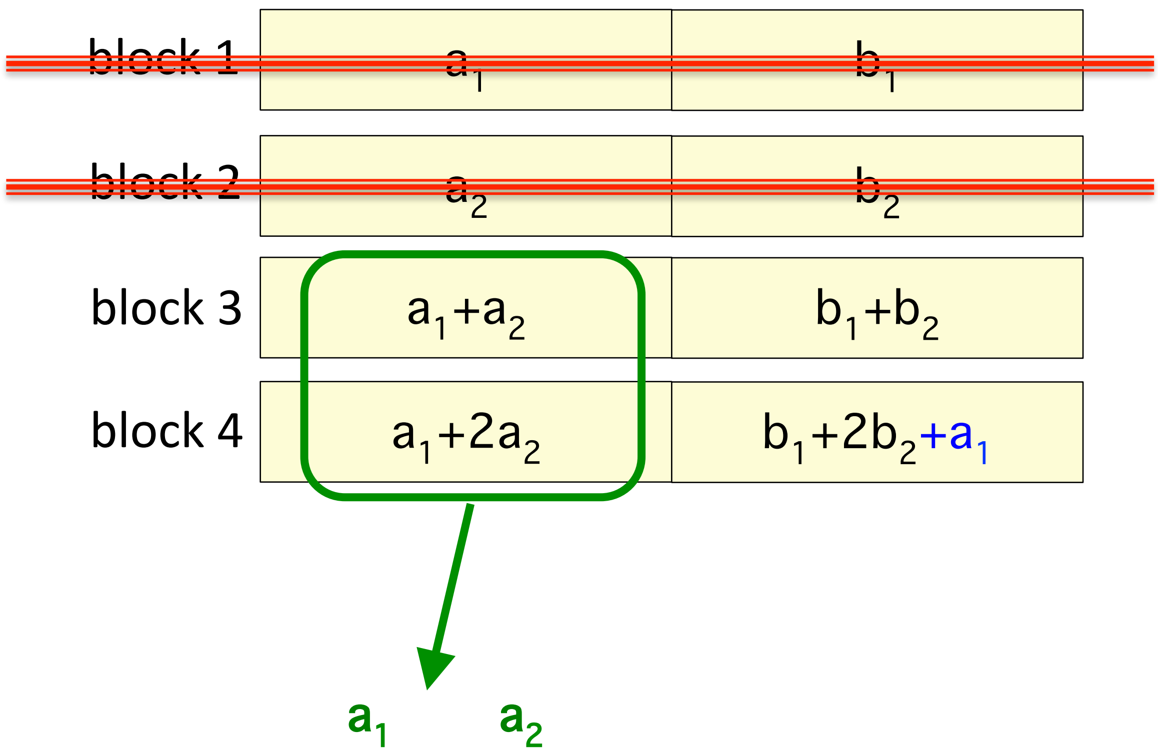
~~block 1~~	~~a_1~~	~~b_1~~
~~block 2~~	~~a_2~~	~~b_2~~
block 3	a_1+a_2	b_1+b_2
block 4	a_1+2a_2	$b_1+2b_2+a_1$

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

Fault-Tolerance (toy example)

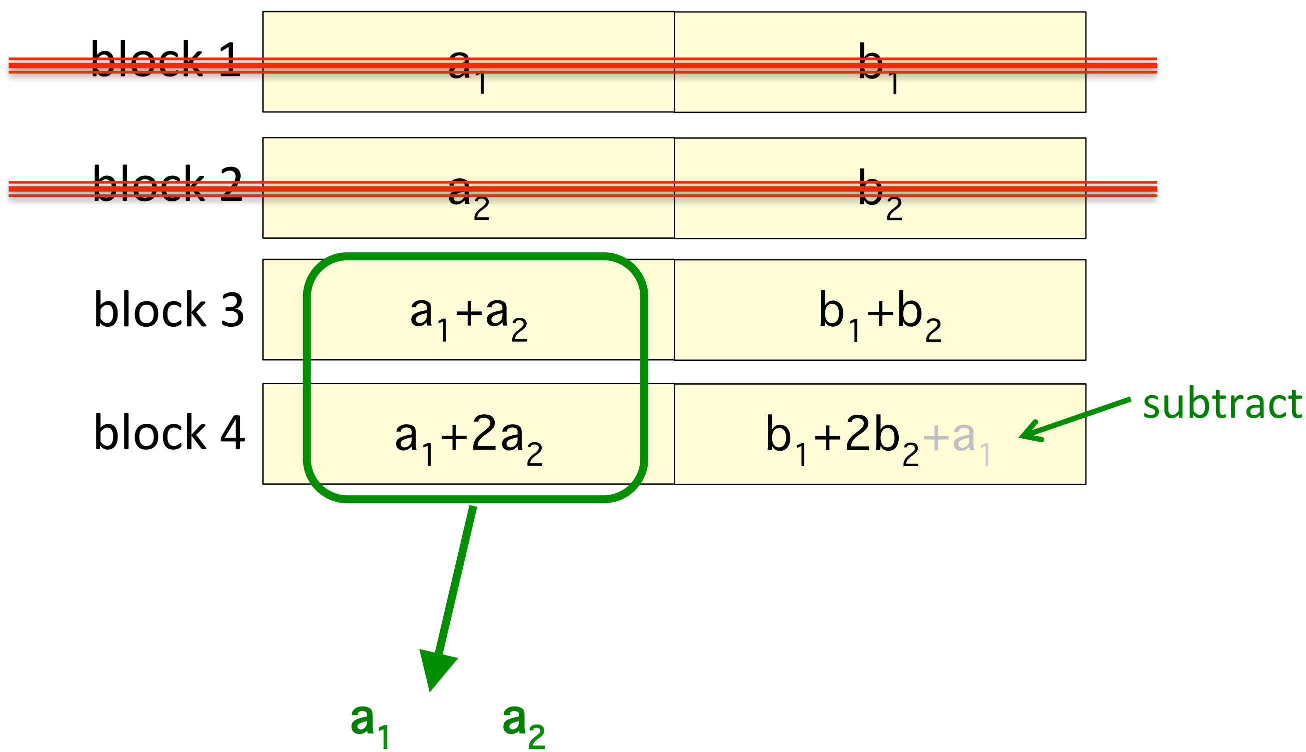
Same fault tolerance as RS code:

can tolerate failure of any 2 nodes



Fault-Tolerance (toy example)

Same fault tolerance as RS code:
can tolerate failure of any 2 nodes

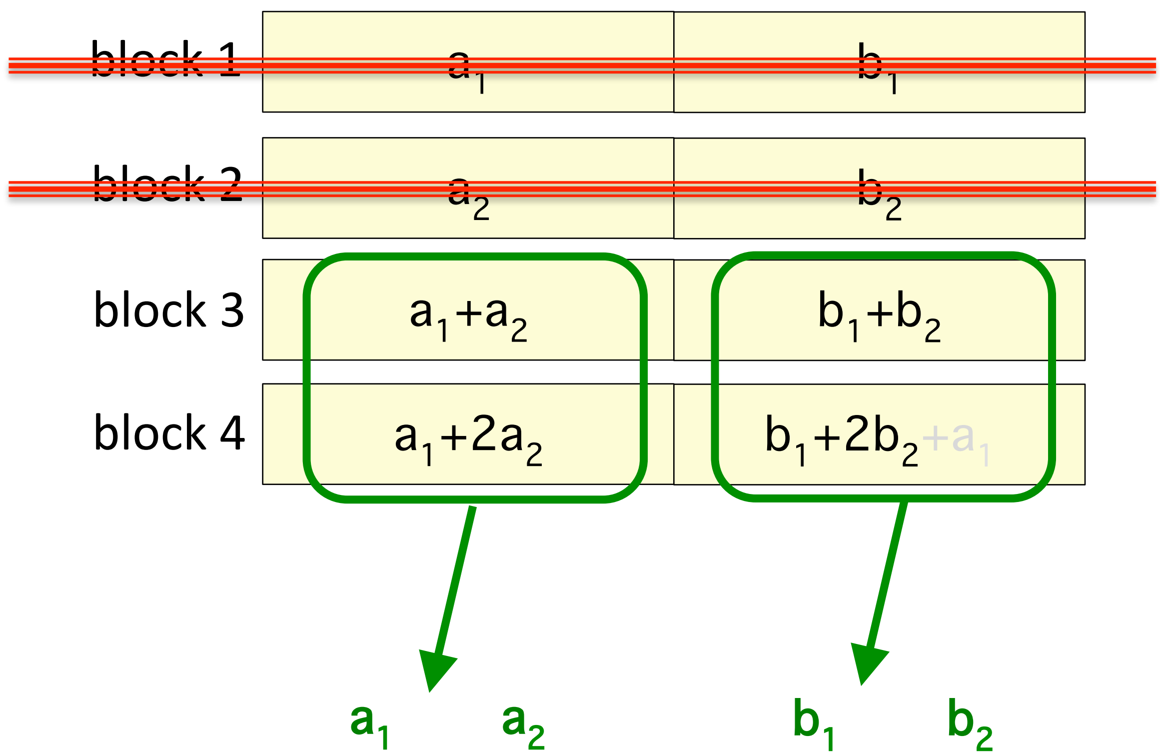


Rashmi et al., "A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster"

Fault-Tolerance (toy example)

Same fault tolerance as RS code:

can tolerate failure of any 2 nodes



Rashmi et al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster”

Recovery (toy example)

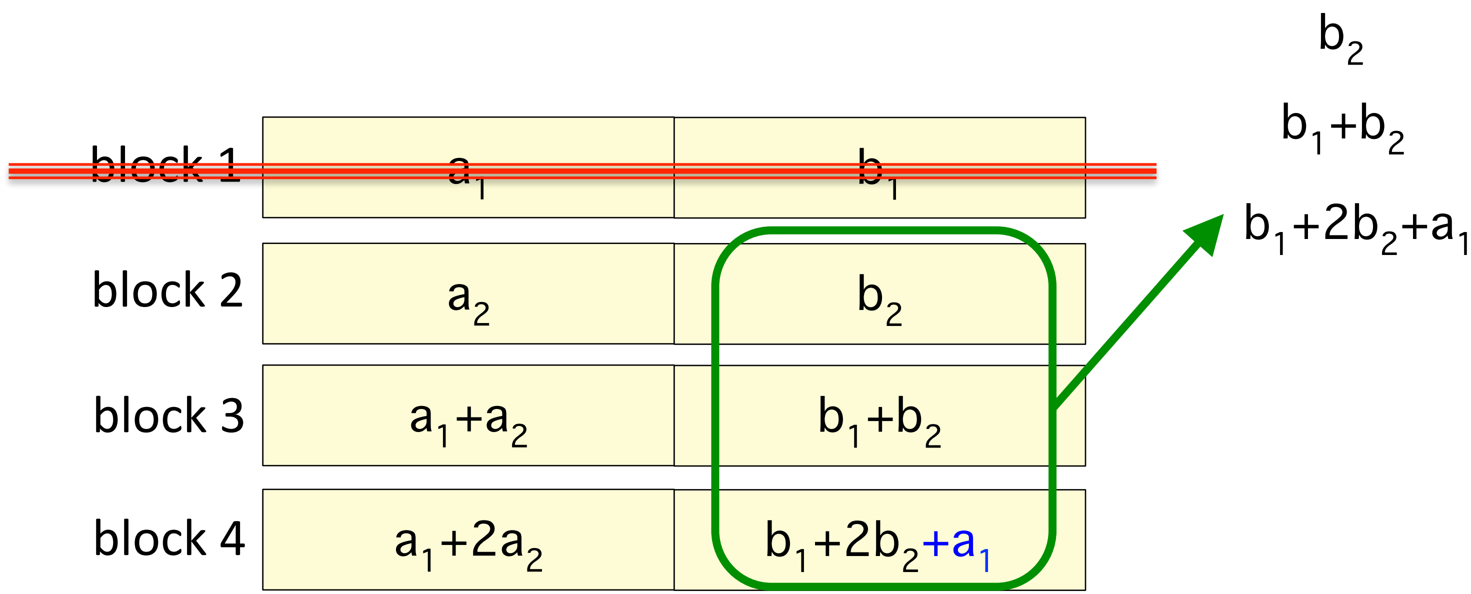
Download & IO only 3 bytes
(instead of 4 bytes as in RS)

~~block 1~~	~~a_1~~	~~b_1~~
block 2	a_2	b_2
block 3	a_1+a_2	b_1+b_2
block 4	a_1+2a_2	$b_1+2b_2+a_1$

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

Recovery (toy example)

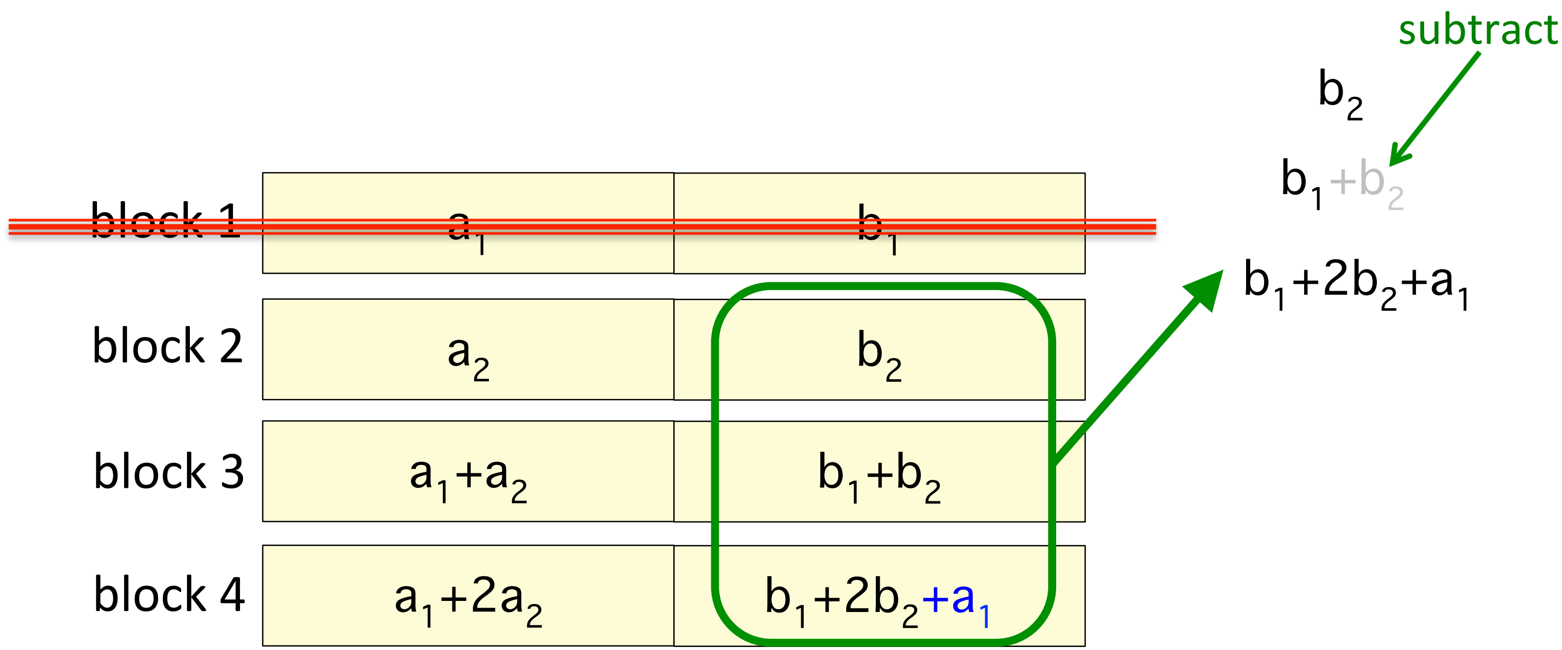
Download & IO only 3 bytes
(instead of 4 bytes as in RS)



Rashmi et al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster”

Recovery (toy example)

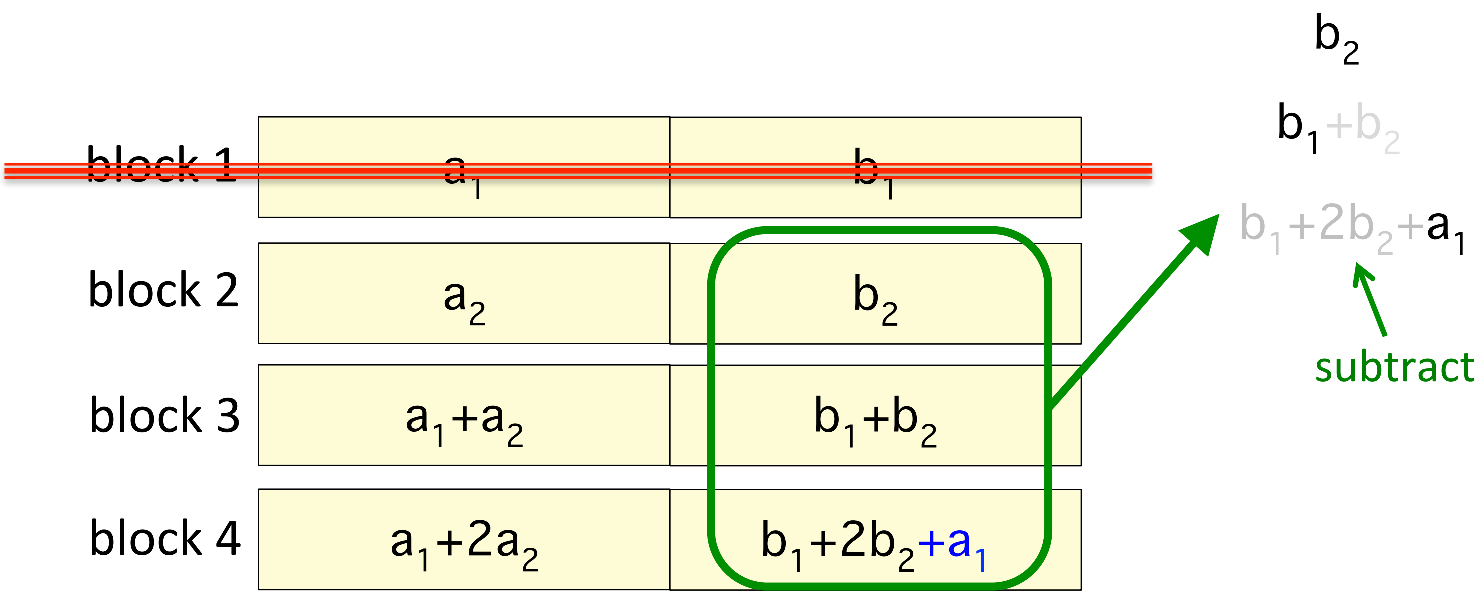
Download & IO only 3 bytes
(instead of 4 bytes as in RS)



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

Recovery (toy example)

Download & IO only 3 bytes
(instead of 4 bytes as in RS)



General Piggybacking Recipe

To construct a Piggybacked-RS code:

- Step 1: Take RS code with identical parameters
- Step 2: Add carefully designed functions from one byte stripe on to another
  - retains same fault-tolerance and storage overhead
  - piggyback functions designed to reduce amount of download and IO for recovery

General theory and algorithms:

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 (ISIT) 2013.

(10,4) Piggybacked-RS

alternative to

(10,4) RS currently used in HDFS

(10,4) Piggybacked-RS code

Step 1: Take a (10, 4) Reed-Solomon code

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10})$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10})$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10})$
	← 1 byte →	← 1 byte →

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

(10,4) Piggybacked-RS code

Step 2: Add `Piggybacks’

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10})$ + $f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_7,a_8,a_9,0)$
	← 1 byte →	← 1 byte →

Rashmi et al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster”

(10,4) Piggybacked-RS code

Tolerates any 4 block failures

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\dots,a_{10})$	$f_1(b_1,\dots,b_{10})$
block 12	$f_2(a_1,\dots,a_{10})$	$f_2(b_1,\dots,b_{10}) + f_4(a_1,a_2,a_3,0,\dots,0)$
block 13	$f_3(a_1,\dots,a_{10})$	$f_3(b_1,\dots,b_{10}) + f_4(0,\dots,0,a_4,a_5,a_6,0,\dots,0)$
block 14	$f_4(a_1,\dots,a_{10})$	$f_4(b_1,\dots,b_{10}) + f_4(0,\dots,0,a_7,a_8,a_9,0)$

Rashmi et al., “A Solution to the Network Challenges of Data Recovery in Erasure-coded Storage: A Study on the Facebook Warehouse Cluster”

(10,4) Piggybacked-RS code

Tolerates any 4 block failures

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_7,a_8,a_9,0)$

recover $a_1,\ldots,a_{10}$
like in RS

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

(10,4) Piggybacked-RS code

Tolerates any 4 block failures

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10})$ + $f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_7,a_8,a_9,0)$

recover $a_1,\ldots,a_{10}$
like in RS

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

(10,4) Piggybacked-RS code

Tolerates any 4 block failures

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_1(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_1(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_1(0,\ldots,0,a_7,a_8,a_9,0)$

recover $a_1,\ldots,a_{10}$ like in RS

subtract piggybacks (functions of $a_1,\ldots,a_{10}$)

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

(10,4) Piggybacked-RS code

Tolerates any 4 block failures

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_1(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_1(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_1(0,\ldots,0,a_7,a_8,a_9,0)$

recover $a_1,\ldots,a_{10}$ like in RS

subtract piggybacks (functions of $a_1,\ldots,a_{10}$)

recover $b_1,\ldots,b_{10}$ like in RS

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

(10,4) Piggybacked-RS code

Efficient data-recovery

~~block 1~~	~~a_1~~	~~b_1~~
⋮		
block 2	a_2	b_2
block 3	a_3	b_3
	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10})$ + $f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10})$ + $f_4(0,\ldots,0,a_7,a_8,a_9,0)$

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

(10,4) Piggybacked-RS code

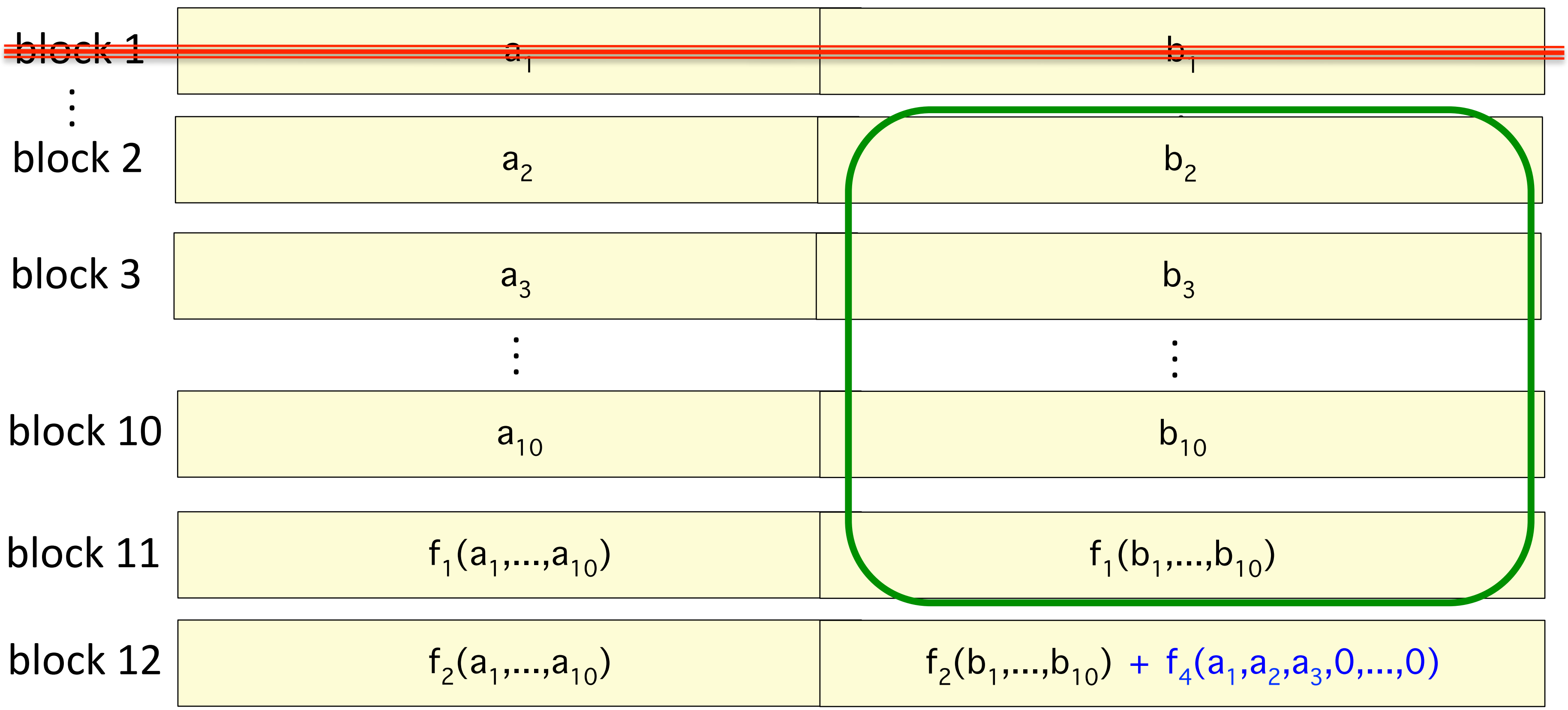
Efficient data-recovery

~~block 1~~	~~a_1~~	~~b_1~~
⋮		
block 2	a_2	b_2
block 3	a_3	b_3
	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10})$ + $f_4(a_1,a_2,a_3,0,\ldots,0)$

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

(10,4) Piggybacked-RS code

Efficient data-recovery

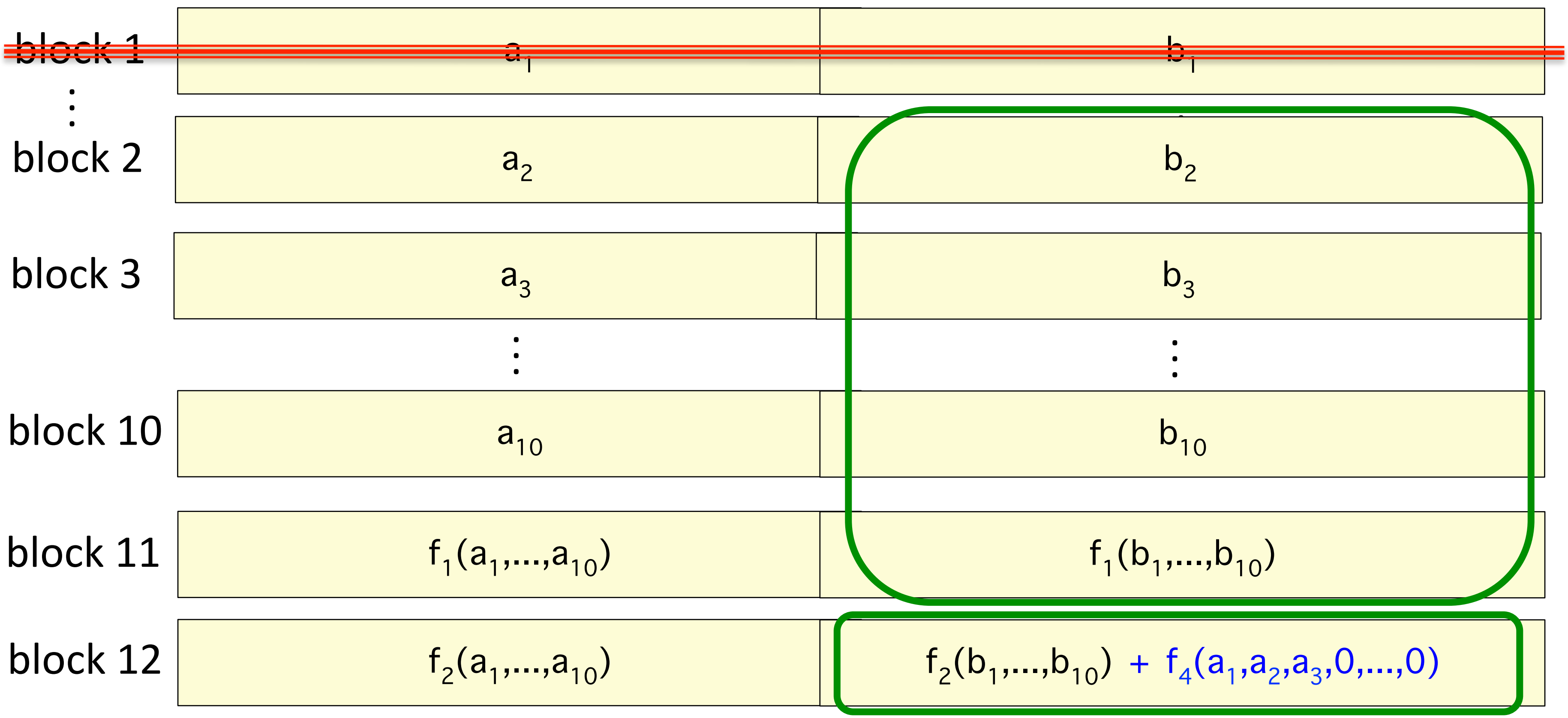


recover $b_1,\ldots,b_{10}$
like in RS

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

(10,4) Piggybacked-RS code

Efficient data-recovery

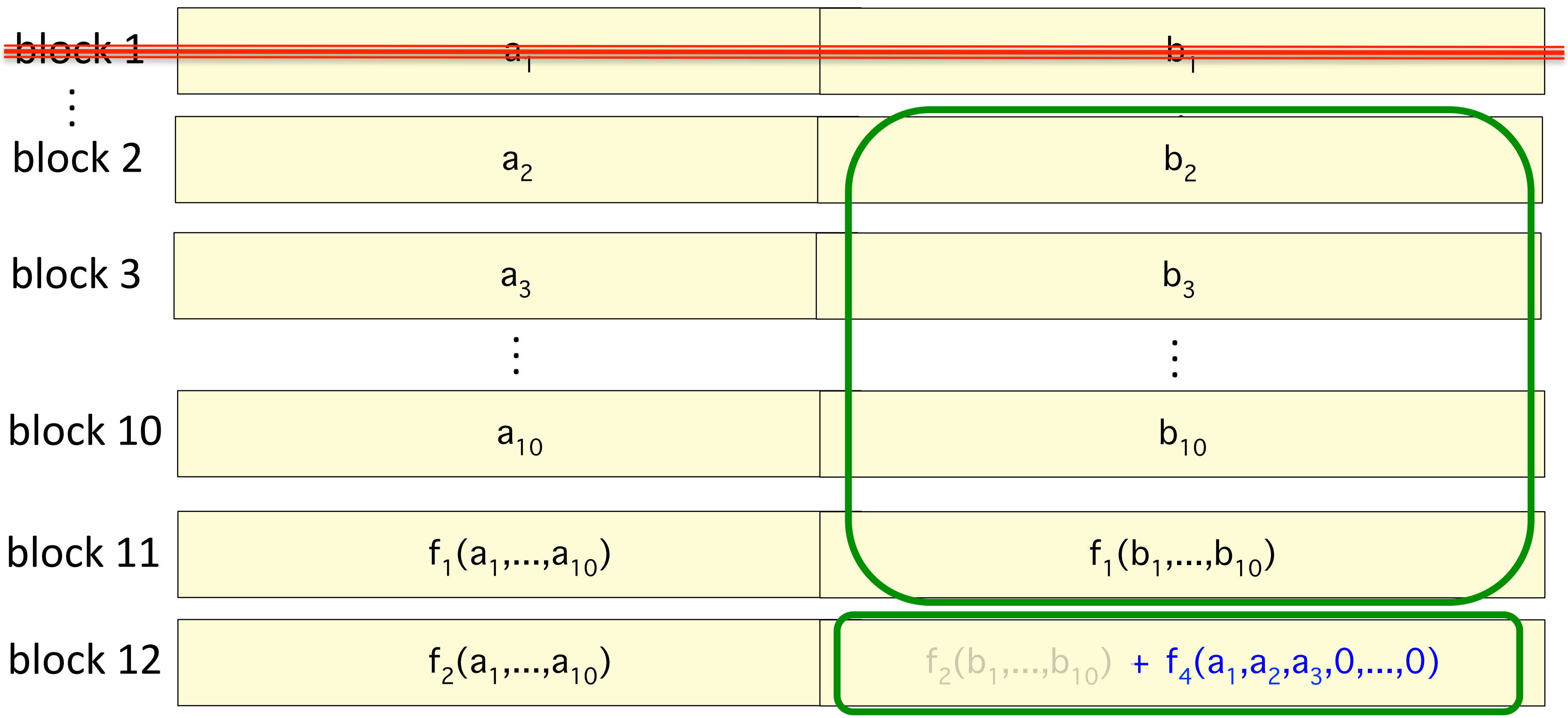


recover $b_1,\ldots,b_{10}$
like in RS

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

(10,4) Piggybacked-RS code

Efficient data-recovery



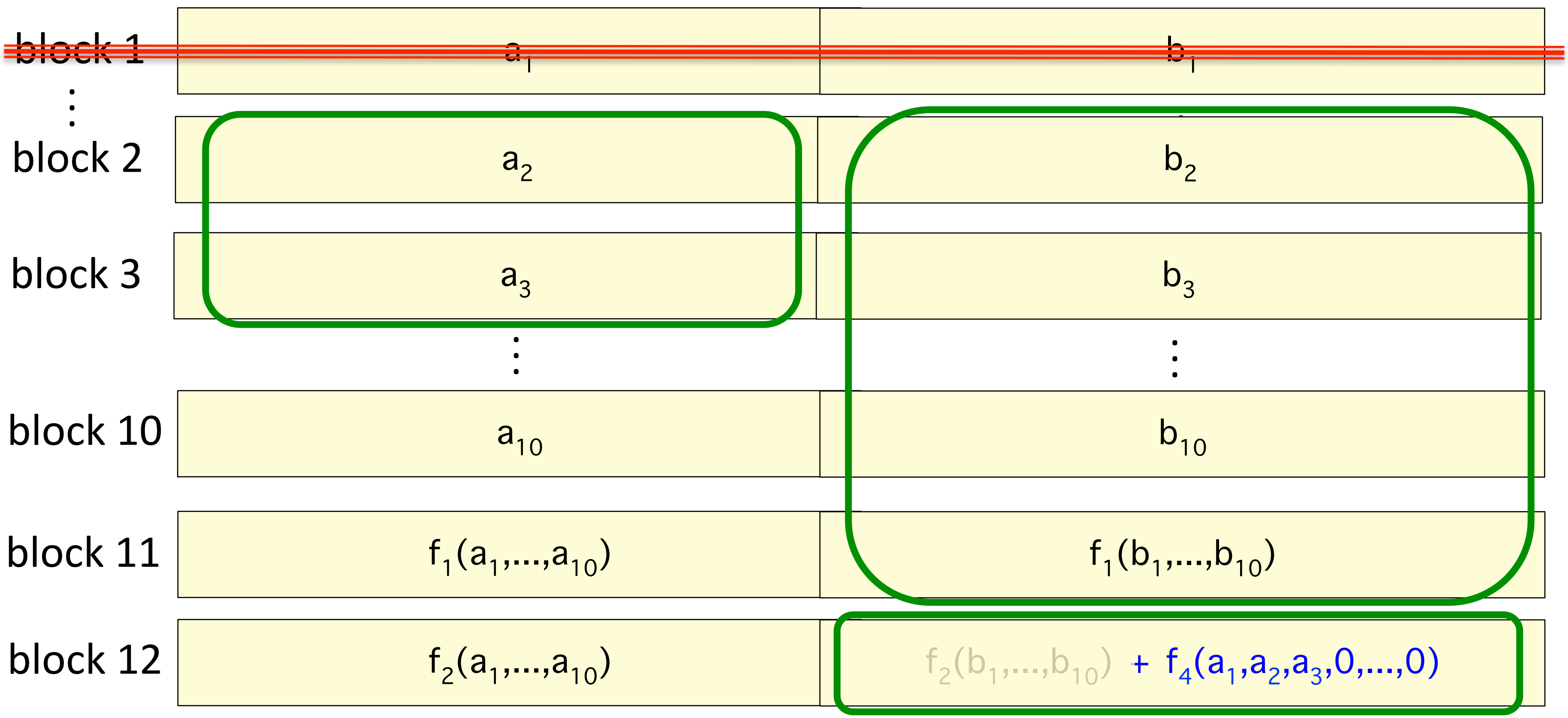
recover $b_1,\ldots,b_{10}$ like in RS

subtract $f_2(b_1,\ldots,b_{10})$

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

(10,4) Piggybacked-RS code

Efficient data-recovery



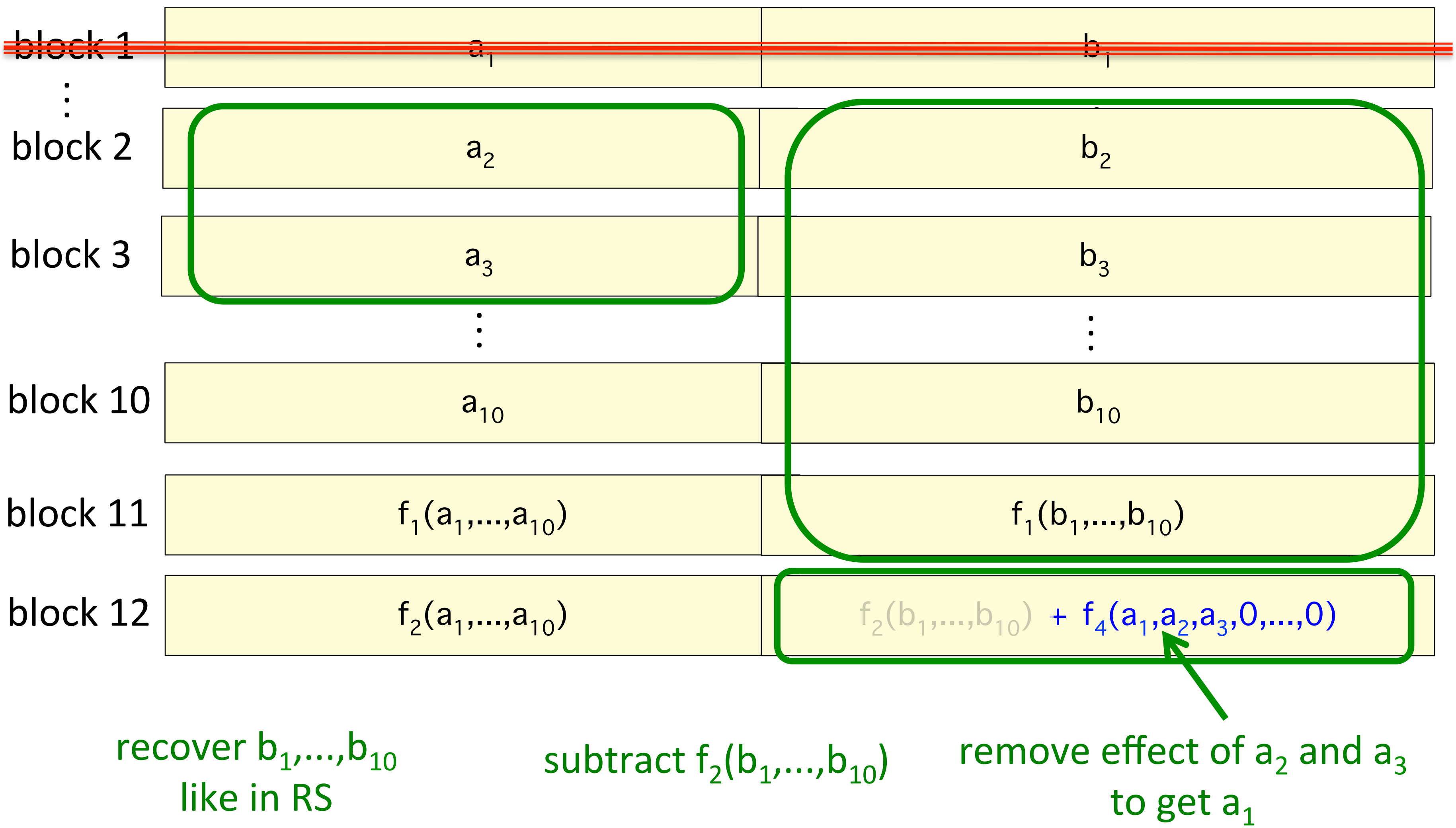
recover $b_1,\ldots,b_{10}$ like in RS

subtract $f_2(b_1,\ldots,b_{10})$

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

(10,4) Piggybacked-RS code

Efficient data-recovery



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

(10,4) Piggybacked-RS code

<u>Download & IO:</u>

20 in RS

13 in Piggybacked-RS

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

(10,4) Piggybacked-RS code

Efficient data-recovery

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_7,a_8,a_9,0)$

Repair of blocks 1,2,3

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

(10,4) Piggybacked-RS code

Efficient data-recovery

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\dots,a_{10})$	$f_1(b_1,\dots,b_{10})$
block 12	$f_2(a_1,\dots,a_{10})$	$f_2(b_1,\dots,b_{10}) + f_4(a_1,a_2,a_3,0,\dots,0)$
block 13	$f_3(a_1,\dots,a_{10})$	$f_3(b_1,\dots,b_{10}) + f_4(0,\dots,0,a_4,a_5,a_6,0,\dots,0)$
block 14	$f_4(a_1,\dots,a_{10})$	$f_4(b_1,\dots,b_{10}) + f_4(0,\dots,0,a_7,a_8,a_9,0)$

Repair of blocks 4,5,6

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

(10,4) Piggybacked-RS code

Efficient data-recovery

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_7,a_8,a_9,0)$

Repair of blocks 7,8,9

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

(10,4) Piggybacked-RS code

Efficient data-recovery

block 1	a_1	b_1
⋮	⋮	⋮
block 10	a_{10}	b_{10}
block 11	$f_1(a_1,\ldots,a_{10})$	$f_1(b_1,\ldots,b_{10})$
block 12	$f_2(a_1,\ldots,a_{10})$	$f_2(b_1,\ldots,b_{10}) + f_4(a_1,a_2,a_3,0,\ldots,0)$
block 13	$f_3(a_1,\ldots,a_{10})$	$f_3(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_4,a_5,a_6,0,\ldots,0)$
block 14	$f_4(a_1,\ldots,a_{10})$	$f_4(b_1,\ldots,b_{10}) + f_4(0,\ldots,0,a_7,a_8,a_9,0)$

Repair of block 10

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

Expected Performance

- Storage efficiency and reliability
  - no additional storage vs RS
  - same fault-tolerance vs RS

Expected Performance

- Storage efficiency and reliability
  - no additional storage vs RS
  - same fault-tolerance vs RS
- Reduced recovery download & disk IO
  - 30% less for single block recoveries in stripe
  - potential reduction >50TB cross-rack traffic per day

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

Expected Performance

- Storage efficiency and reliability
  - no additional storage vs RS
  - same fault-tolerance vs RS
- Reduced recovery download & disk IO
  - 30% less for single block recoveries in stripe
  - potential reduction >50TB cross-rack traffic per day
- Recovery time: expect faster recovery
  - need to connect to more nodes
  - system limited by disk and network bandwidth
  - corroborated by preliminary experiments
  - hence, expect higher MTTDL

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

Related Work: Measurements

- Existing Studies
  - Availability studies:
    Schroeder & Gibson 2007, Jiang et al. 2008, Ford et al. 2010 etc.
  - Comparisons between replication and erasure codes:
    Rodrigues & Liskov 2005, Weatherspoon & Kubiatowicz 2002 etc.

- Our focus
  - Increased network traffic due to increased downloads during recovery of erasure-coded data
  - Measurements from Facebook warehouse cluster in production

Related Work: Codes for Efficient Data Recovery

- Huang et al. (Windows Azure) 2012, Sathiamoorthy et al. (Xorbas) 2013
  - add additional parities: need extra storage
- Hu et al. (NCFS) 2011
  - Network file system using 'repair-by-transfer' codes (Shah et al.): need extra storage
- Khan et al. (Rotated-RS) 2012
  - #parity ≤ 3 (also, #data ≤ 36)
- Xiang et al., Wang et al. (Optimized RDP & EVENODD) 2010
  - #parity <=2
- **Our solution: Piggybacked-RS**
  - no additional storage: storage-capacity optimal
  - any #data & #parity
  - as good as or better than Rotated-RS, optimized RDP & EVENODD

Summary and Future Work

- Erasure codes require higher download & IO for recovery
- Measurements from Facebook warehouse cluster in production
- Piggybacked-RS: alternative to RS
  - no additional storage required; same fault-tolerance as RS
  - 30% reduction in download & disk IO for recovery
- Future Work
  - implementation in HDFS (in progress at UC Berkeley)
  - empirical evaluation