15-750: Algorithms in the Real World

Algorithms for coding

(Error Correcting Codes)

Announcement:

Midterm exam on March 1.

More details in the Piazza post.

Welc**e t* t*is clas* o* c*d*ng .
Y*u a** in f*r a f*n rid*!

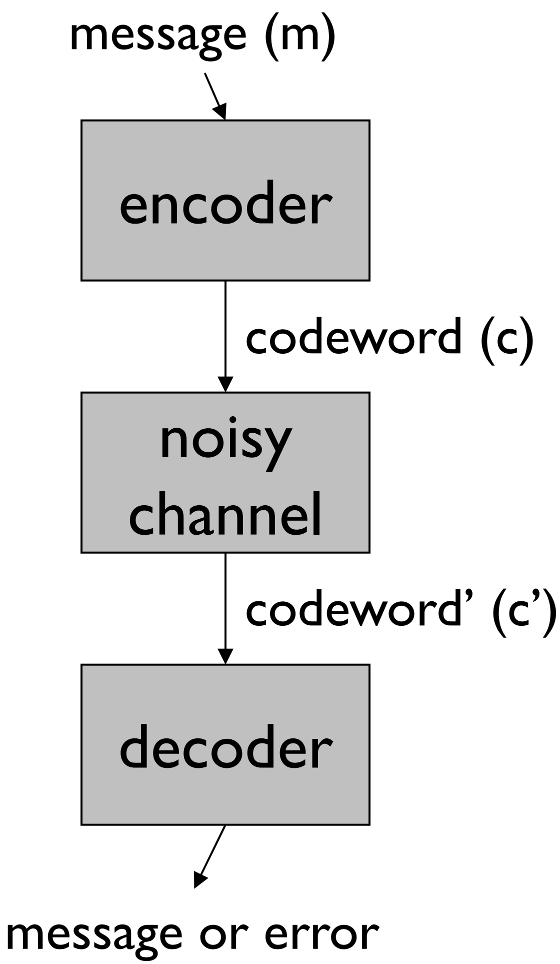
What do these sentences say?

Why did this work?

Redundancy!

Codes are clever ways of judiciously adding redundancy to enable recovery under “noise”.

General Model



"Noise" introduced by the channel:

- changed fields in the codeword vector (e.g. a flipped bit).
  - Called **errors**
- missing fields in the codeword vector (e.g. a lost byte).
  - Called **erasures**

How the decoder deals with errors and/or erasures?

- **detection** (only needed for errors)
- **correction**

Applications

Numerous applications:

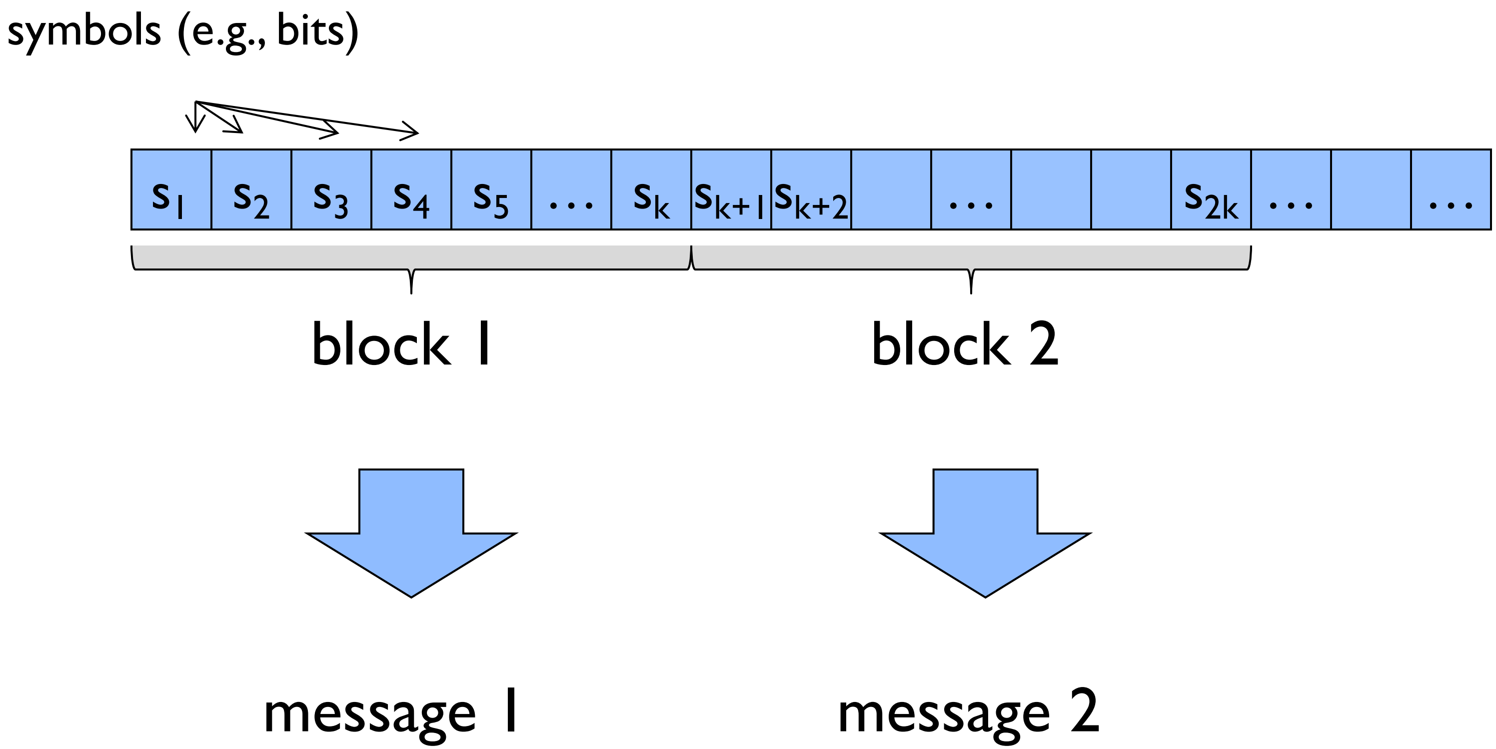
Some examples

- **Storage**: Hard disks, cloud storage, NAND flash…
- **Wireless**: Cell phones, wireless links,
- **Satellite and Space**: TV, Mars rover, …

Reed-Solomon codes are by far the most used in practice.

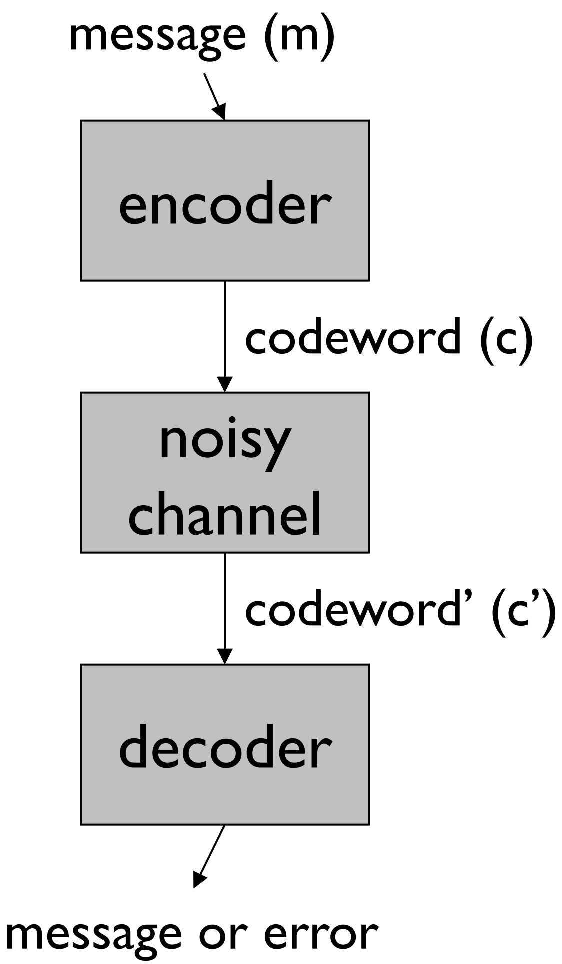
Low density parity check codes (LDPC) codes used for 4G (and 5G) communication and NAND flash

Block Codes



Other kind: convolutional codes (we won't cover it)…

Block Codes



- Each message and codeword is of fixed size
- Notation:

"dimension of the code"

k = |m|
length of the message

"length of the code"

n = |c|
length of the codeword

C = "code" = set of codewords

Simple Examples

3-Repetition code: k=1, n=3

Message		Codeword
0	->	000
1	->	111

- How many **erasures** can be recovered?
- How many **errors** can be **detected**?
- Up to how many **errors** can be **corrected**?

Errors are much harder to deal with than erasures.

Why?

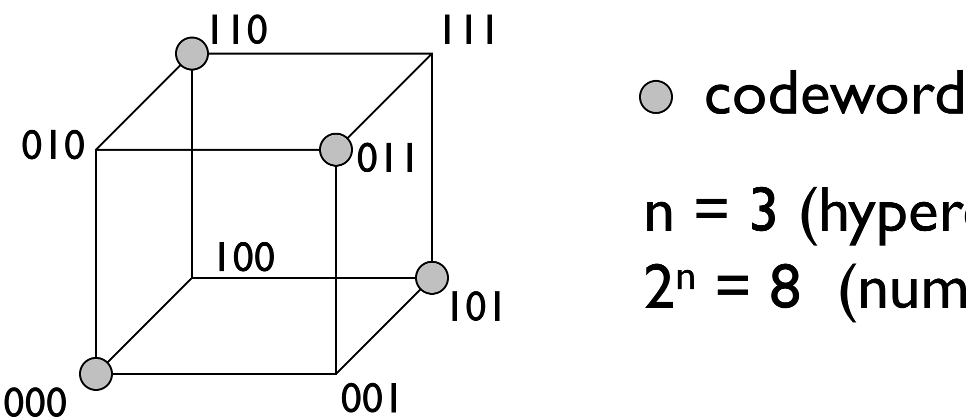
Need to find out **where** the errors are!

Simple Examples

Single parity check code: k=2, n=3

Message		Codeword
00	->	000
01	->	011
10	->	101
11	->	110

Consider codewords as vertices on a hypercube.

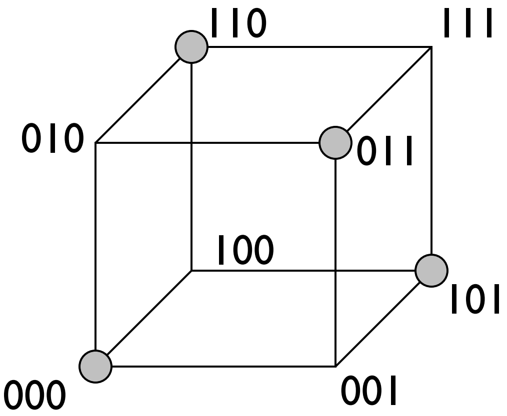


n = 3 (hypercube dimensionality)

$2^n = 8$ (number of nodes)

Simple Examples

Single parity check code: k=2, n=3



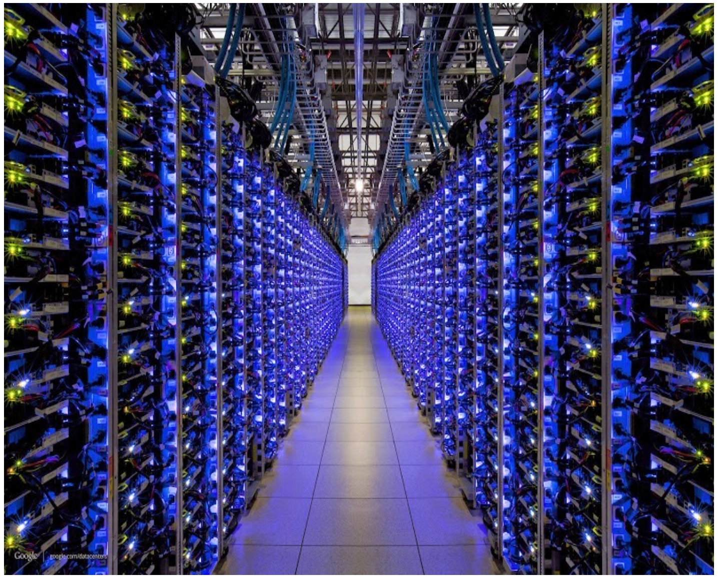
- How many **erasures** can be recovered?
- How many **errors** can be **detected**?
- Up to how many **errors** can be **corrected**?

Systematic codes

Definition: A **Systematic code** is one in which the message symbols appear in the codeword in uncoded form

message	codeword
000	**000**000
001	**001**011
010	**010**101
011	**011**110
100	**100**110
101	**101**101
110	**110**011
111	**111**000

Large-scale distributed storage systems



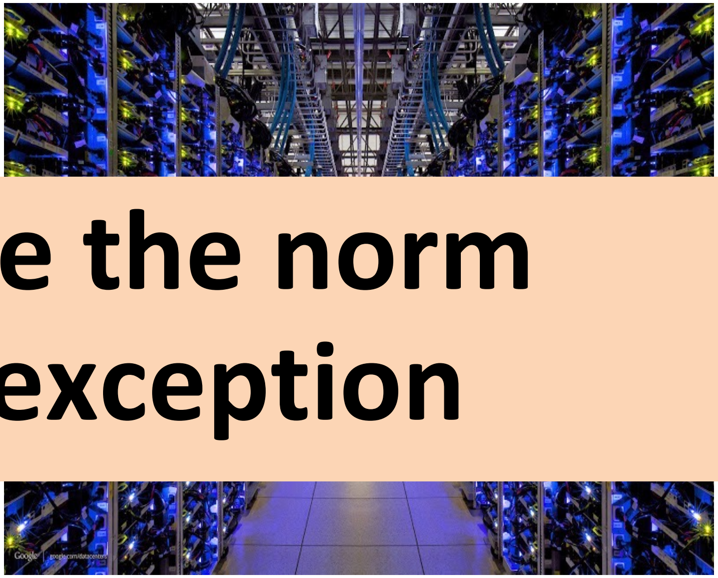
1000s of interconnected servers

100s of petabytes of data

- Commodity components
- Software issues, power failures, maintenance shutdowns

Large-scale distributed storage systems

1000s of interconnected servers

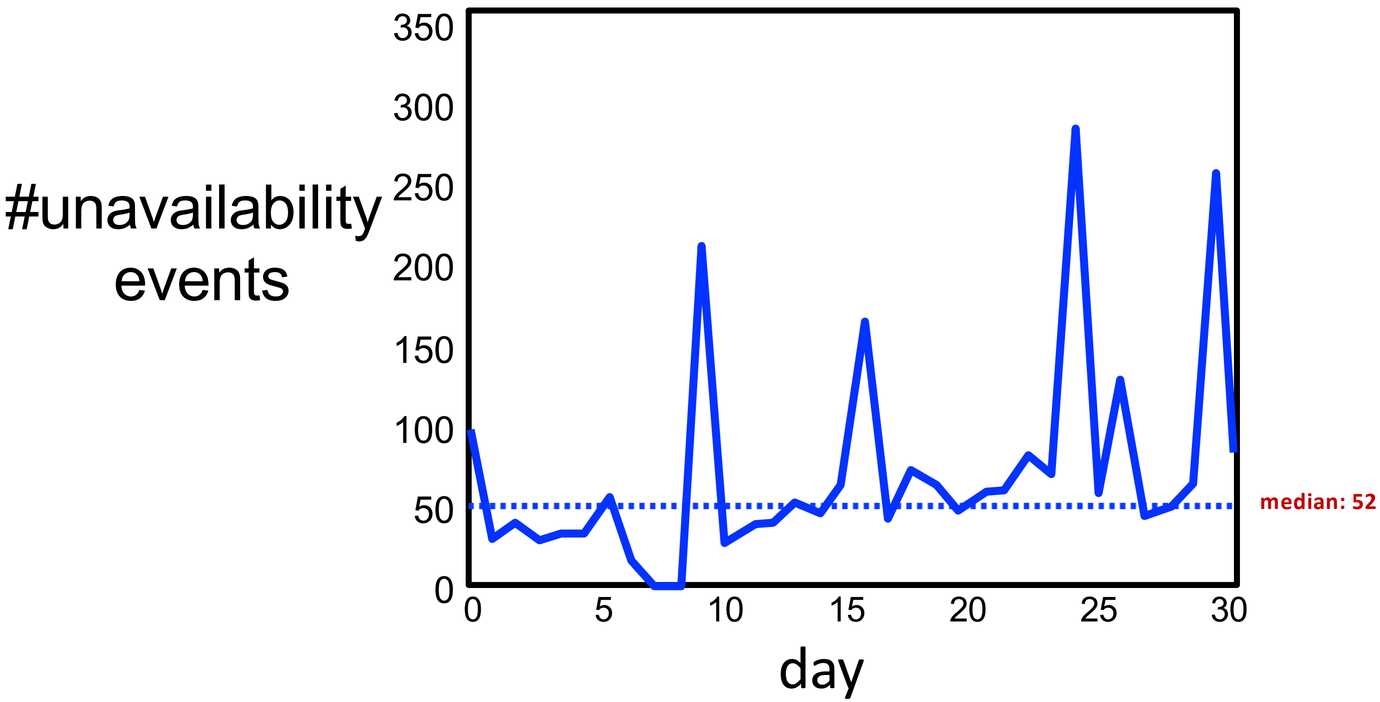


Unavailabilities are the norm rather than the exception

- Commodity components
- Software issues, power failures, maintenance shutdowns

Facebook analytics cluster in production: unavailability statistics

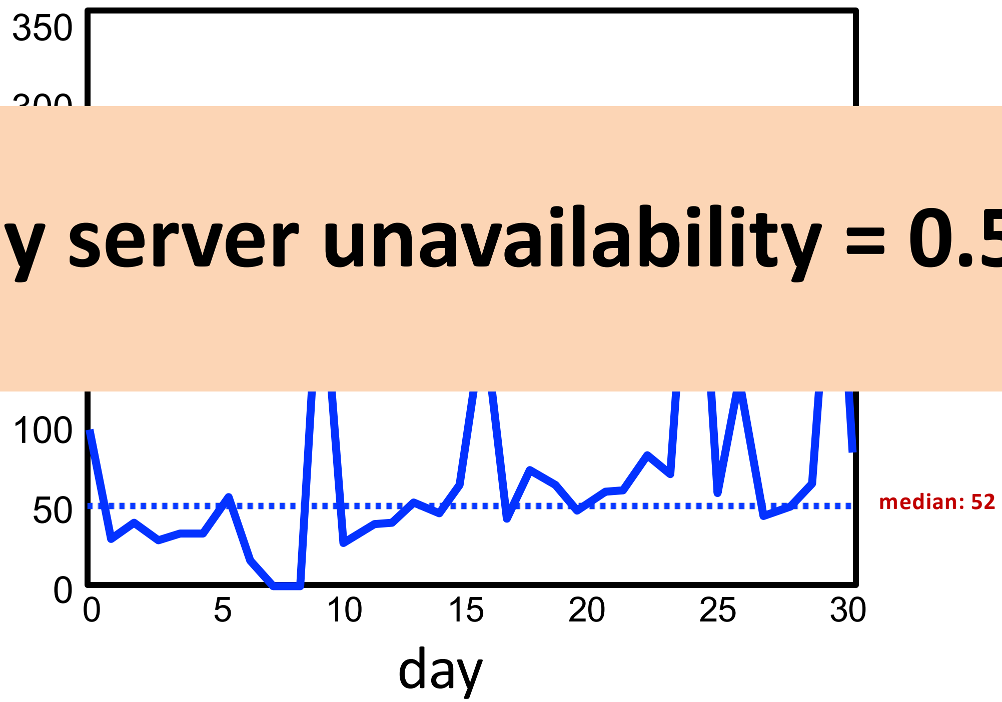
- Multiple thousands of servers
- Unavailability event: server unresponsive for > 15 min



[Rashmi, Shah, Gu, Kuang, Borthakur, Ramchandran, USENIX HotStorage 2013 and ACM SIGCOMM 2014]

Facebook analytics cluster in production: unavailability statistics

- Multiple thousands of servers
- Unavailability event: server unresponsive for > 15 min



Daily server unavailability = 0.5 - 1%

[Rashmi, Shah, Gu, Kuang, Borthakur, Ramchandran,
USENIX HotStorage 2013 and ACM SIGCOMM 2014]

Servers unavailable

⬇

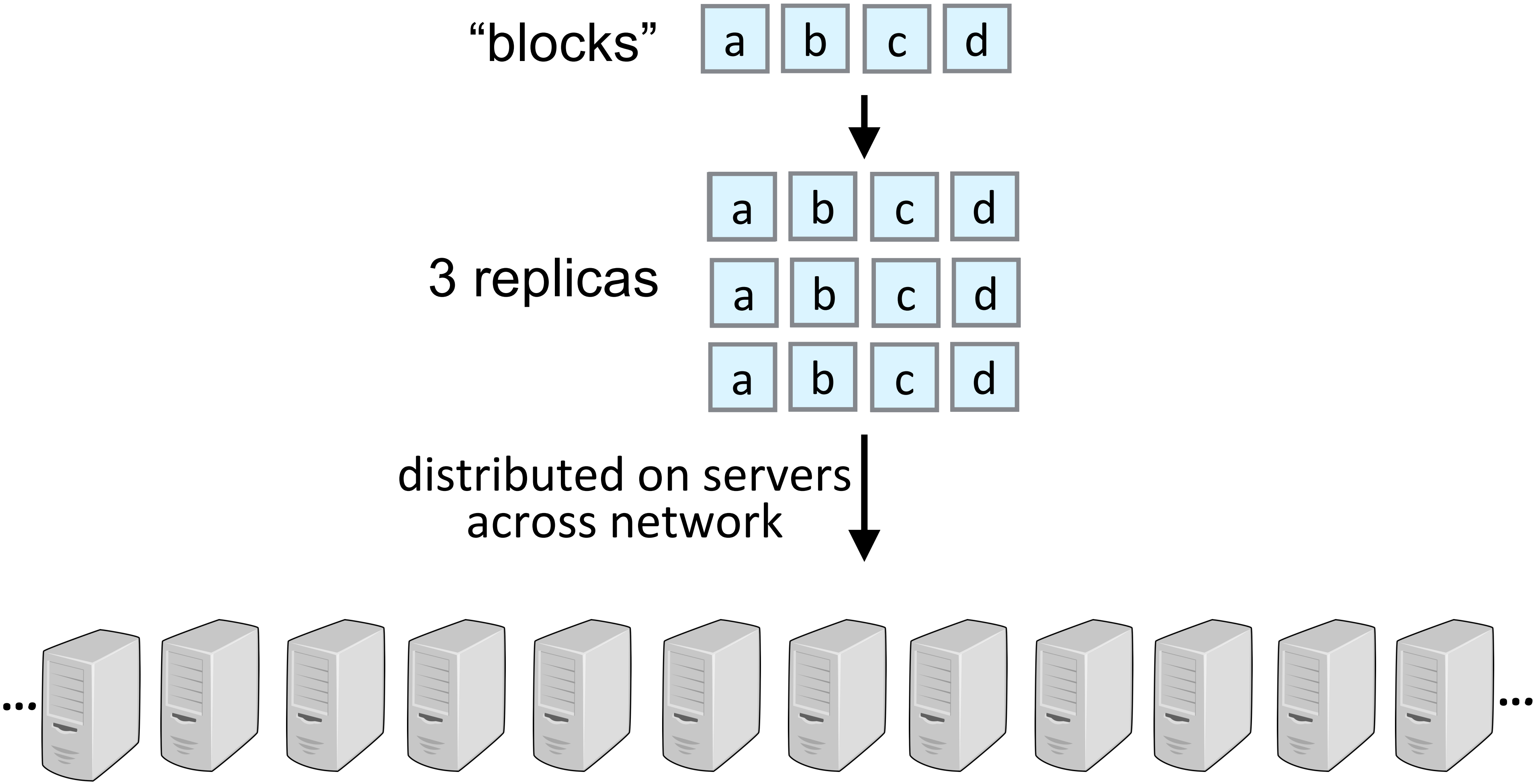
Data inaccessible

⬇ Applications cannot wait,
Data cannot be lost

Data needs to be stored in a redundant fashion

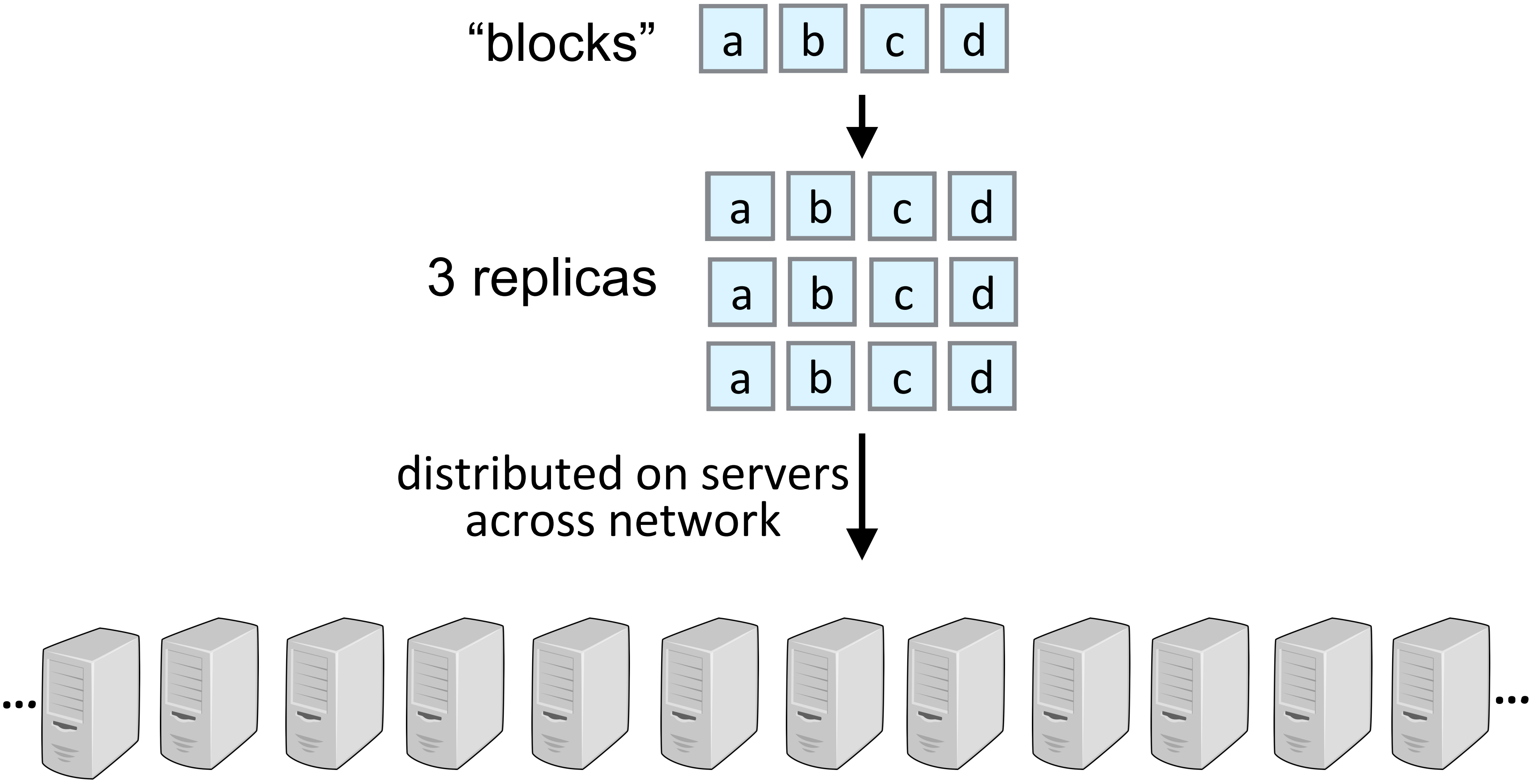
Traditional approach: Replication

- Storing **multiple copies** of data: Typically 3x-replication



Traditional approach: Replication

- Storing **multiple copies** of data: Typically 3x-replication



Traditional approach: Replication

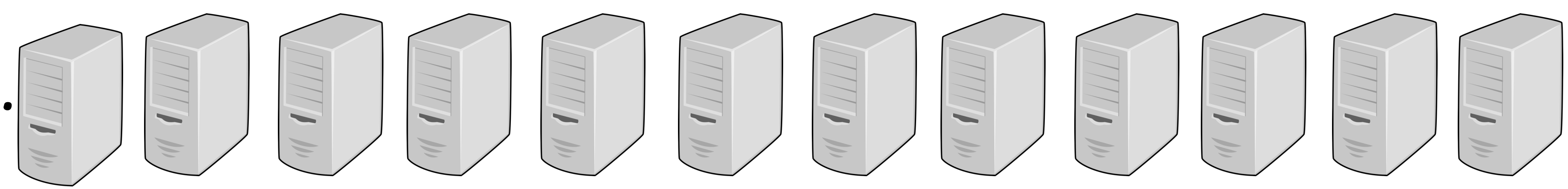
- Storing **multiple copies** of data: Typically 3x-replication

Too expensive for large-scale data

3 replicas

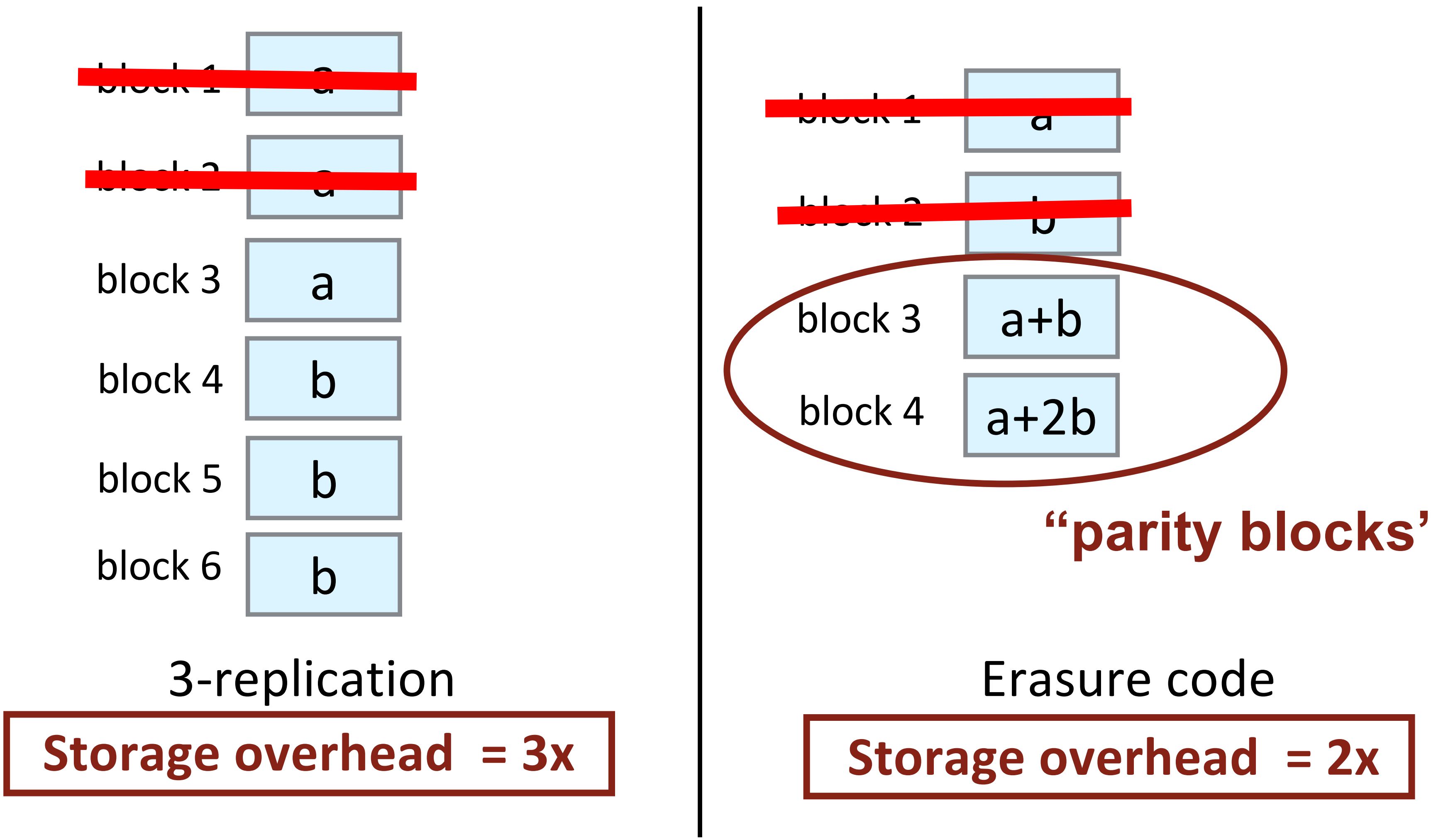
a	b	c	d

Better alternative: codes!



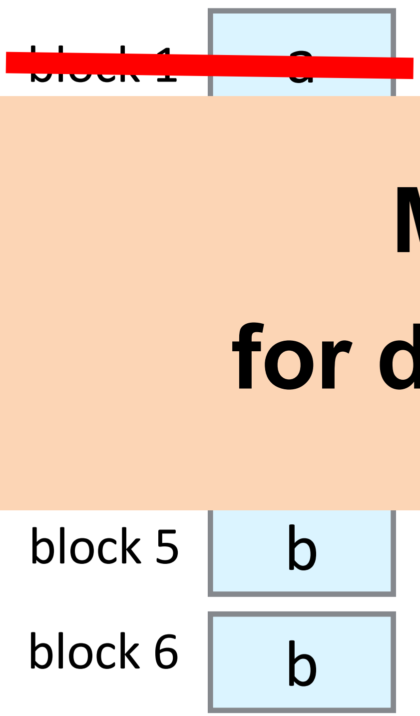
Two data blocks to be stored: a and b

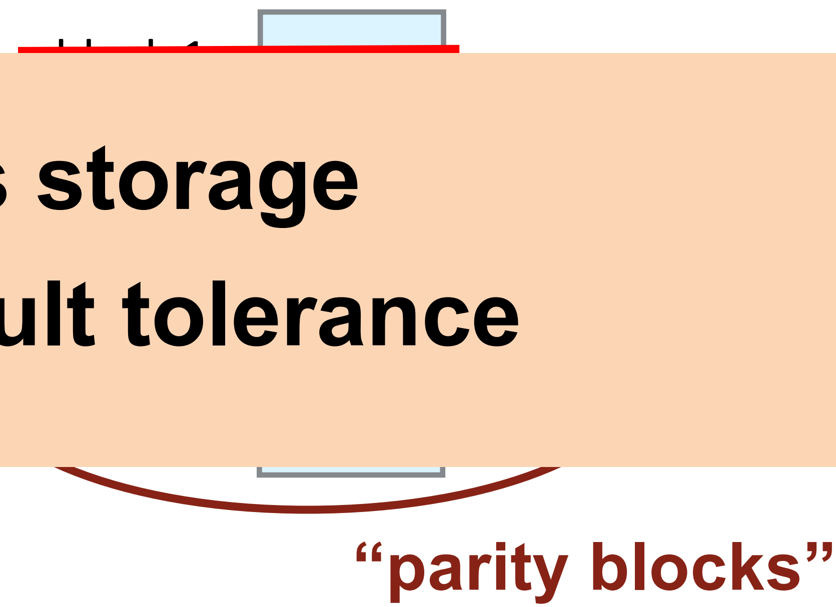
Tolerate any 2 failures



Two data blocks to be stored: a and b

Tolerate any 2 failures





Much less storage
for desired fault tolerance

3-replication

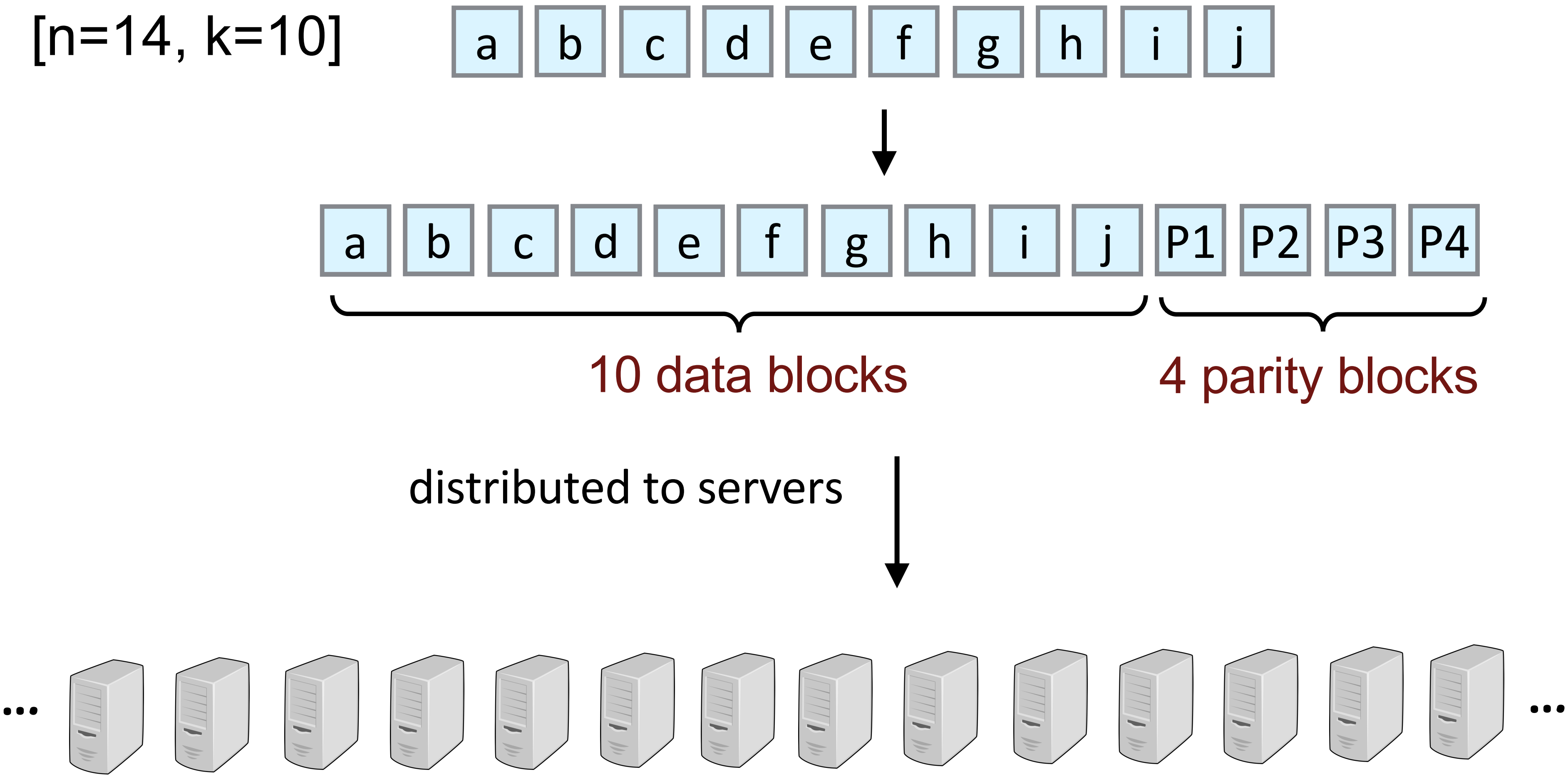
Storage overhead = 3x

Erasure code

Storage overhead = 2x

Erasure codes: how are they used in distributed storage systems?

Example:
[n=14, k=10]



Almost all large-scale storage systems today employ erasure codes

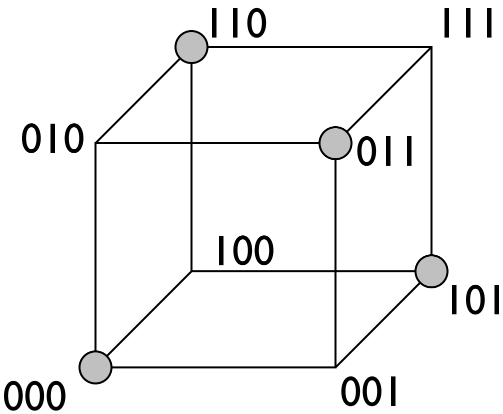
Facebook, Google, Amazon, Microsoft...

"Considering trends in data growth & datacenter hardware, we foresee HDFS **erasure coding** being an **important feature in years to come**"

- Cloudera Engineering (September, 2016)

Simple Examples

Single parity check code: k=2, n=3



- How many **erasures** can be recovered?
- How many **errors** can be **detected**?
- Up to how many **errors** can be **corrected**?

Erasure correction = 1, error detection = 1, error correction = 0

Cannot even correct single error. Why?

Codewords are too “close by”

Let’s formalize this notion of distance..

Block Codes

Notion of distance between codewords: **Hamming distance**

$\Delta(\mathbf{x,y})$ = number of positions s.t. $x_i \neq y_i$

Minimum distance of a code

$\mathbf{d} = \min\{\Delta(x,y) : x,y \in C, x \neq y\}$

Code described as: $\mathbf{(n, k, d)_q}$ ←

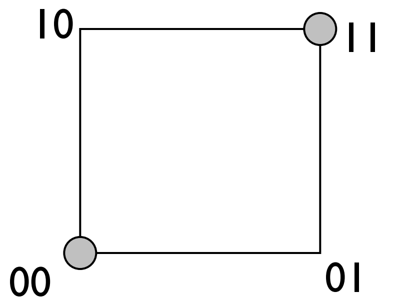
- Σ = alphabet
- $\mathbf{q} = |\Sigma|$ = alphabet size
- $\mathbf{C} \subseteq \Sigma^n$ (codewords)

Question:

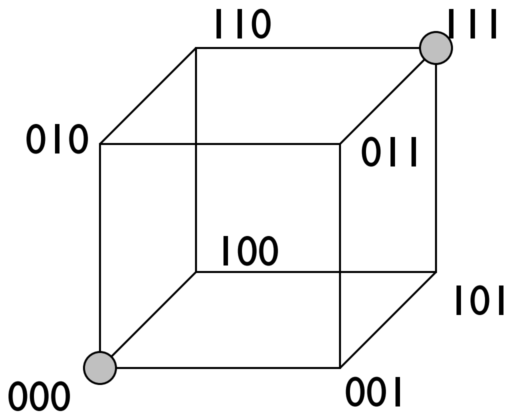
What alphabet did we use so far?

Error Correcting One Bit Messages

How many bits do we need to correct a one bit error on a one bit message?



2 bits
0 -> 00, 1-> 11
(n=2,k=1,d=2)



3 bits
0 -> 000, 1-> 111
(n=3,k=1,d=3)

In general need $d \geq 3$ to correct one error. Why?

Role of Minimum Distance

Theorem:

A code C with minimum distance “d” can:

1. detect any (d-1) errors
2. recover any (d-1) erasures
3. correct any $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors

Intuition: <board>

Stated another way:

For s-bit error detection $d \geq s + 1$

For s-bit error correction $d \geq 2s + 1$

To correct a erasures and b errors if $d \geq a + 2b + 1$