f4: Facebook's Warm BLOB Storage System

f4: Facebook's Warm BLOB Storage System

Subramanian Muralidhar*, Wyatt Lloyd†*, Sabyasachi Roy*, Cory Hill*, Ernest Lin*, Weiwen Liu*, Satadru Pan*, Shiva Shankar*, Viswanath Sivakumar*, Linpeng Tang‡*, Sanjeev Kumar*

*Facebook Inc., †University of Southern California, ‡Princeton University

Abstract

Facebook's corpus of photos, videos, and other Binary Large OBjects (BLOBs) that need to be reliably stored and quickly accessible is massive and continues to grow. As the footprint of BLOBs increases, storing them in our traditional storage system, Haystack, is becoming increasingly inefficient. To increase our storage efficiency, from a throughput perspective. Yet, triple replication also provided important fault tolerance guarantees.

Our newer f4 BLOB storage system provides the same fault tolerance guarantees as Haystack but at a lower effective-replication-factor. f4 is simple, modular, scalable, and fault tolerant; it handles the request rate of BLOBs we store it in; it responds to requests with

Disclaimer:

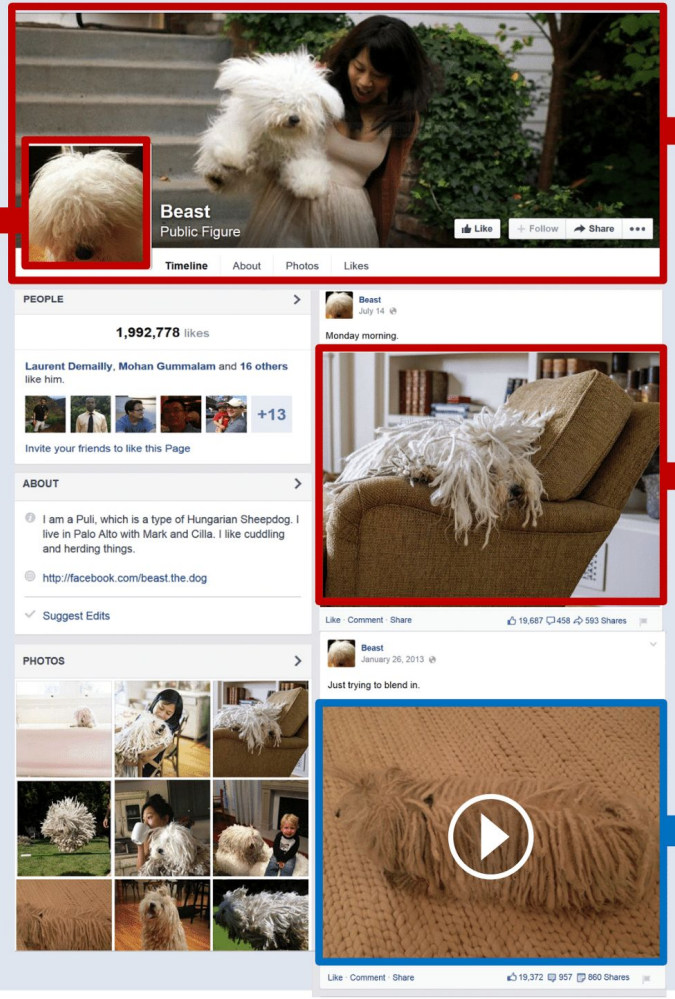
Some of the slides were borrowed from Facebook’s OSDI presentation

BLOBs@FB

Immutable
&
Unstructured

Diverse

A LOT of them!!



Profile Photo

Cover Photo

Feed Photo

Feed Video

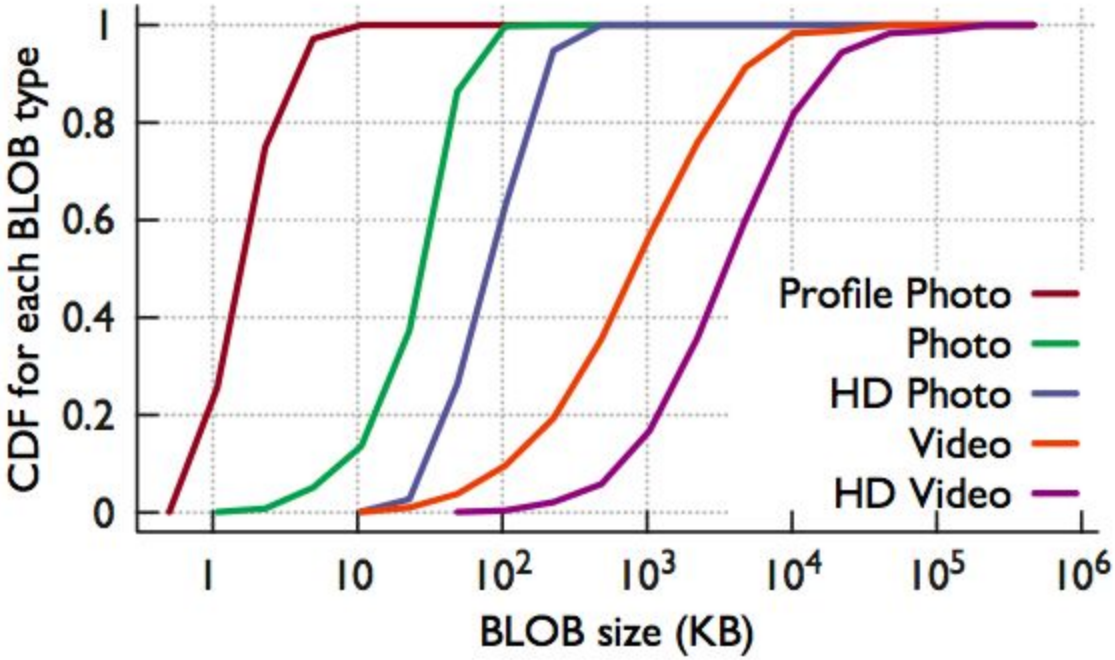
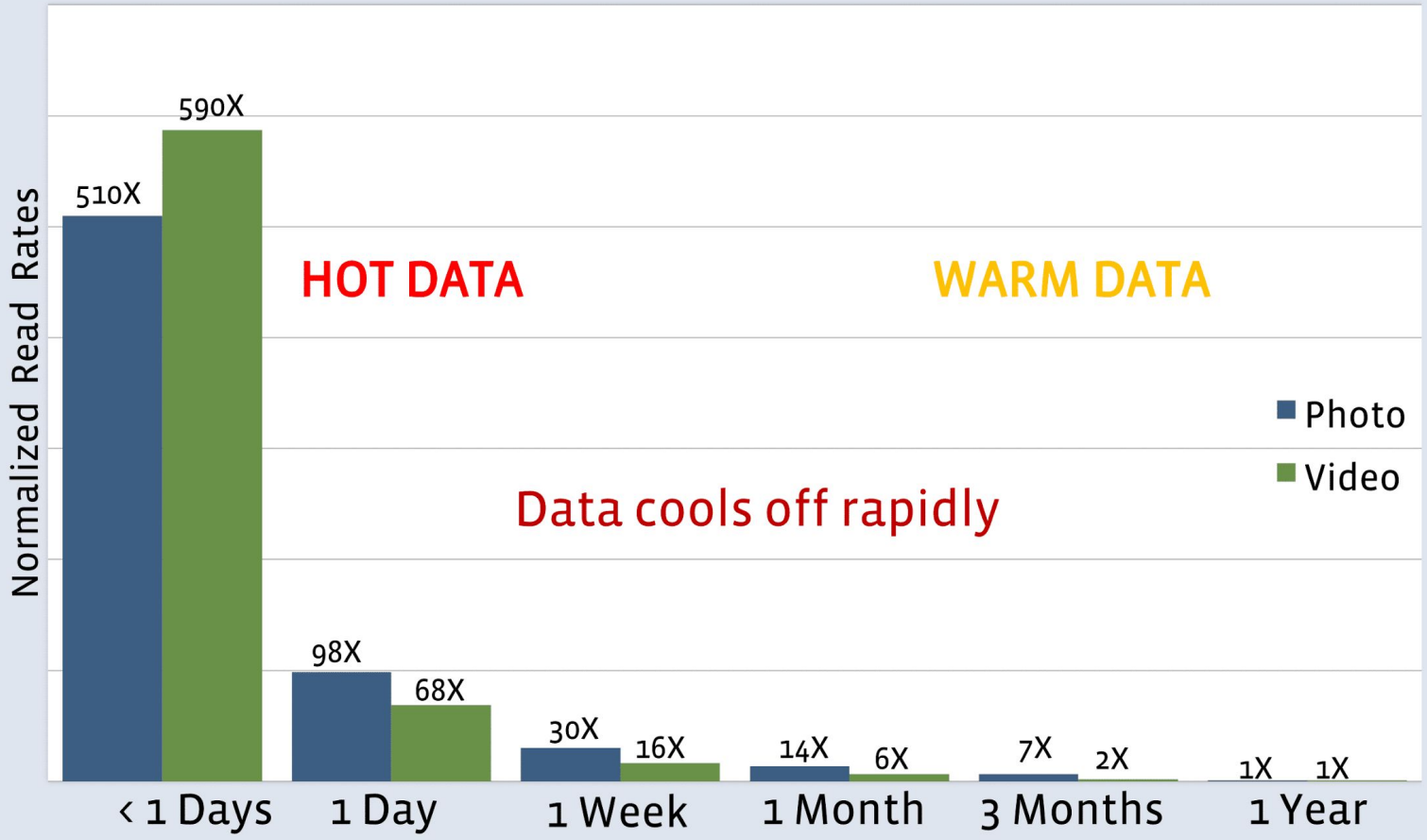


Figure 2: Size distribution for five BLOB types.


Normalized Read Rates
510X
590X
98X
68X
30X
16X
14X
6X
7X
2X
1X
1X
HOT DATA
WARM DATA
Data cools off rapidly
Photo
Video
< 1 Days
1 Day
1 Week
1 Month
3 Months
1 Year

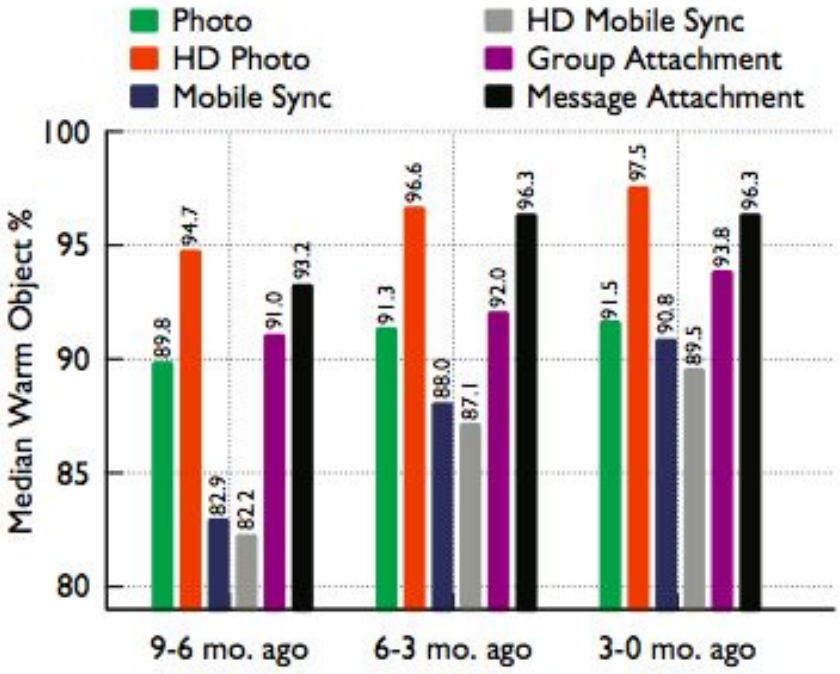


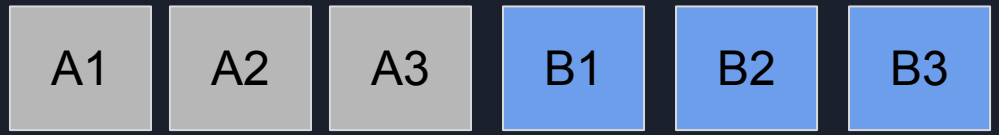
Figure 5: Median percentage of each type that was warm 9-6 months ago, 6-3 months ago, 3 months ago to now. The remaining percentage of each type is hot.

Requirements for “hot” data storage

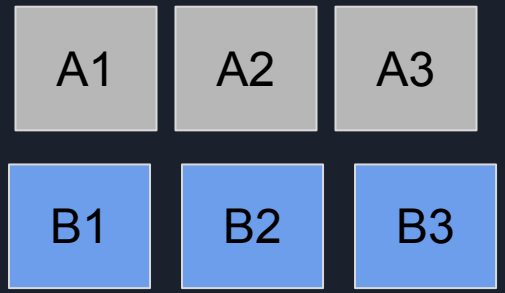
- handle different types of failures
- be able to handle incoming load

Refresher: RAID 6

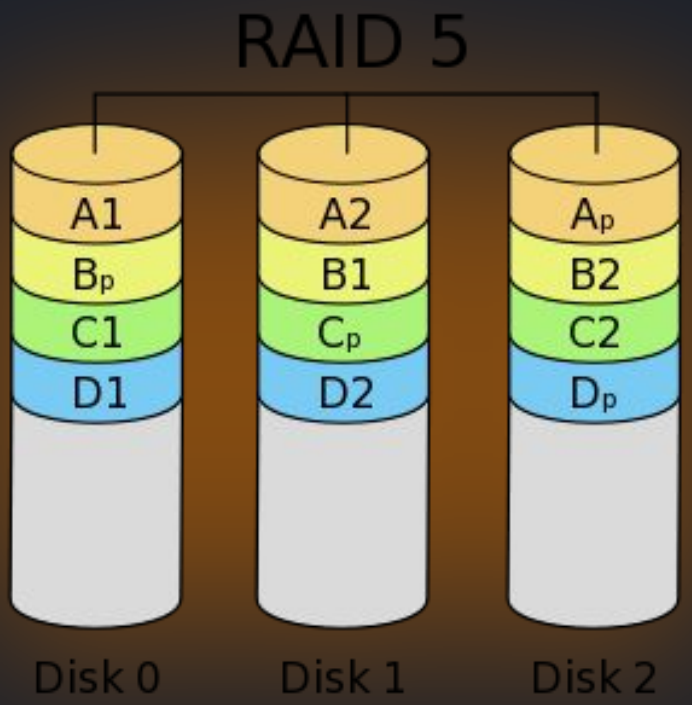
Refresher: RAID 6



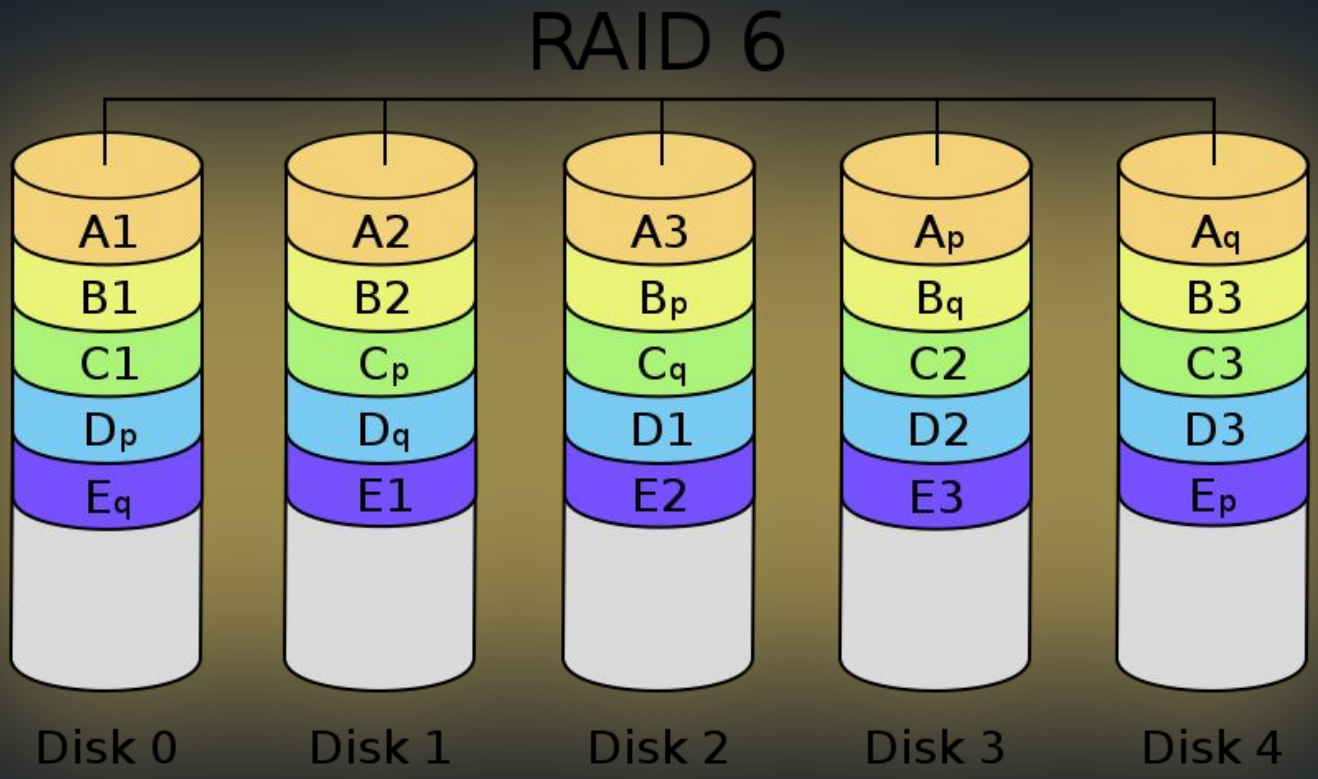
Refresher: RAID 6



Refresher: RAID 6

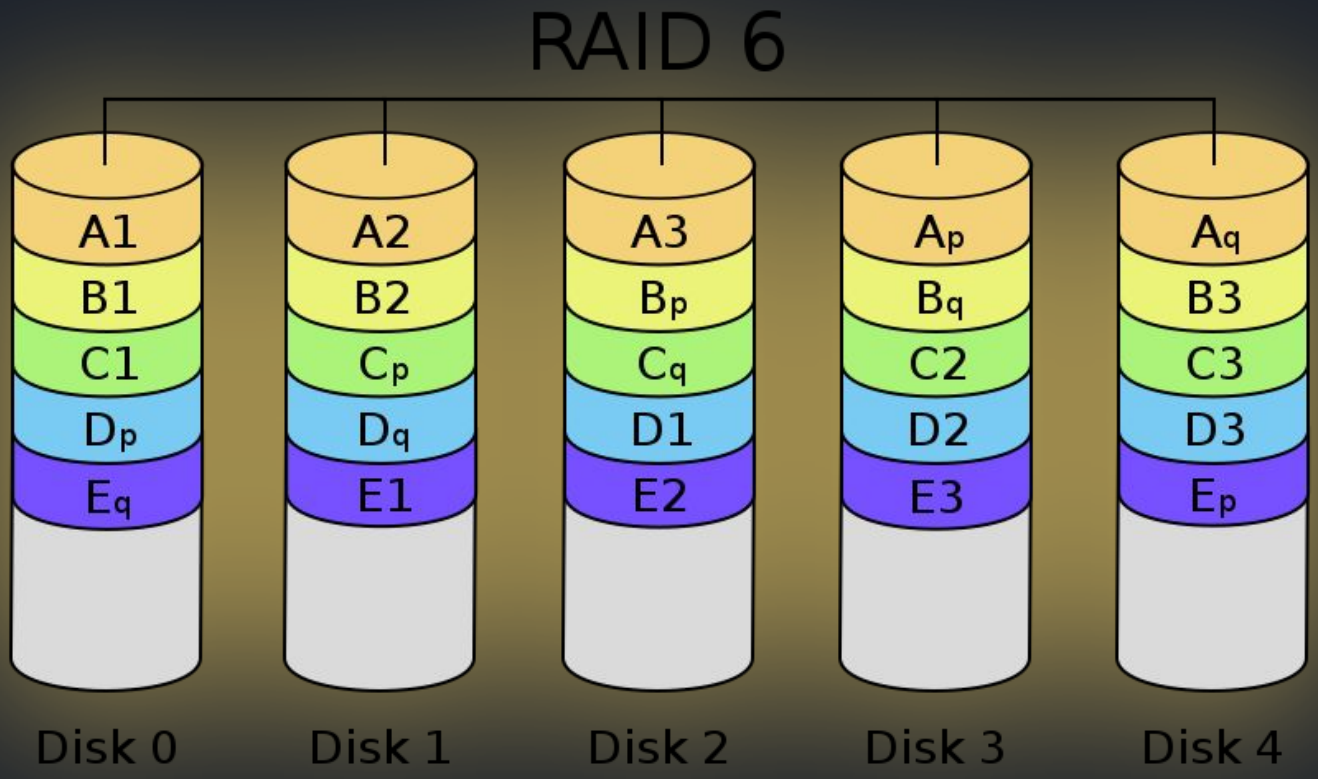


Refresher: RAID 6



$$\mathbf{Q} = \bigoplus_i g^i D_i = g^0 \mathbf{D}_0 \oplus g^1 \mathbf{D}_1 \oplus g^2 \mathbf{D}_2 \oplus \ldots \oplus g^{n-1} \mathbf{D}_{n-1}$$

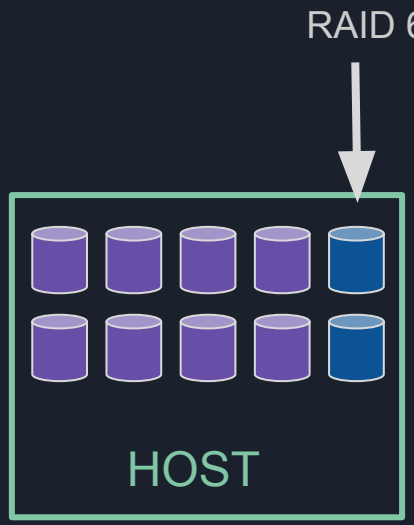
Refresher: RAID 6



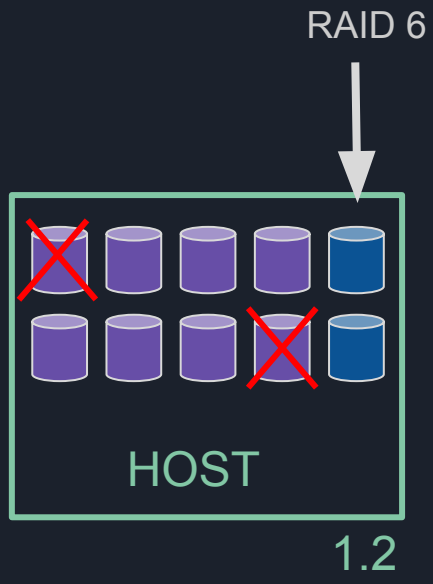
Fault tolerance

Fault tolerance

Fault tolerance



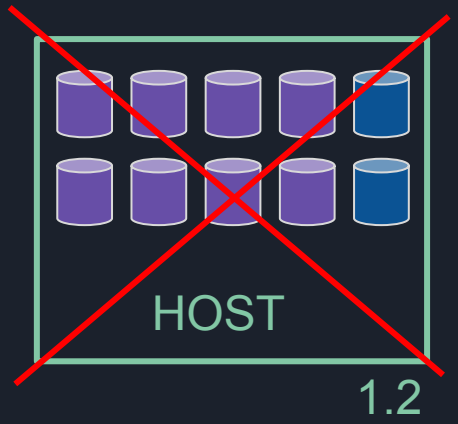
Fault tolerance



Fault tolerance

1.2

Fault tolerance

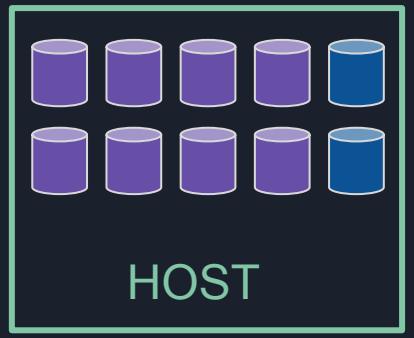


Fault tolerance

Fault tolerance

Irrecoverability:
- 9 disk failures
- 3 host failures

1.2

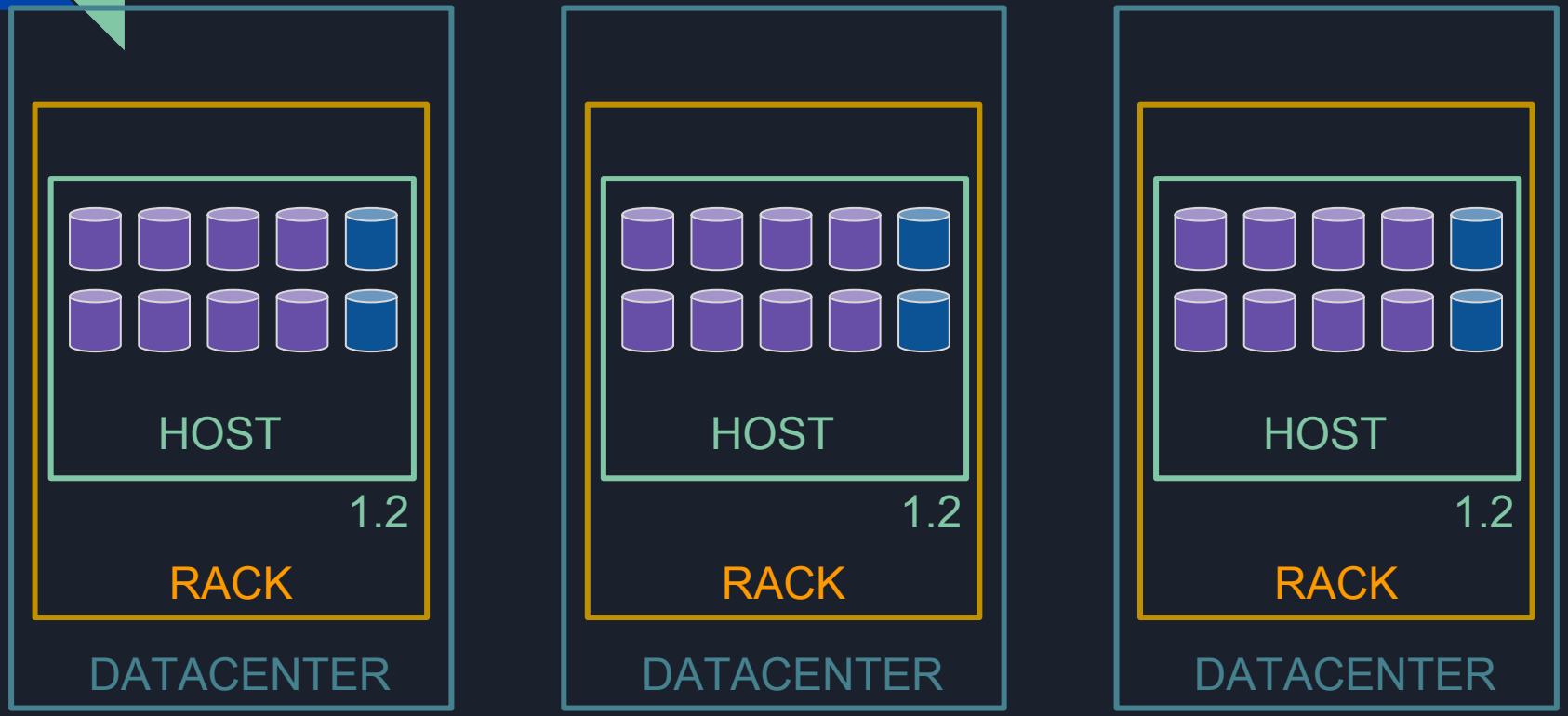


1.2

1.2

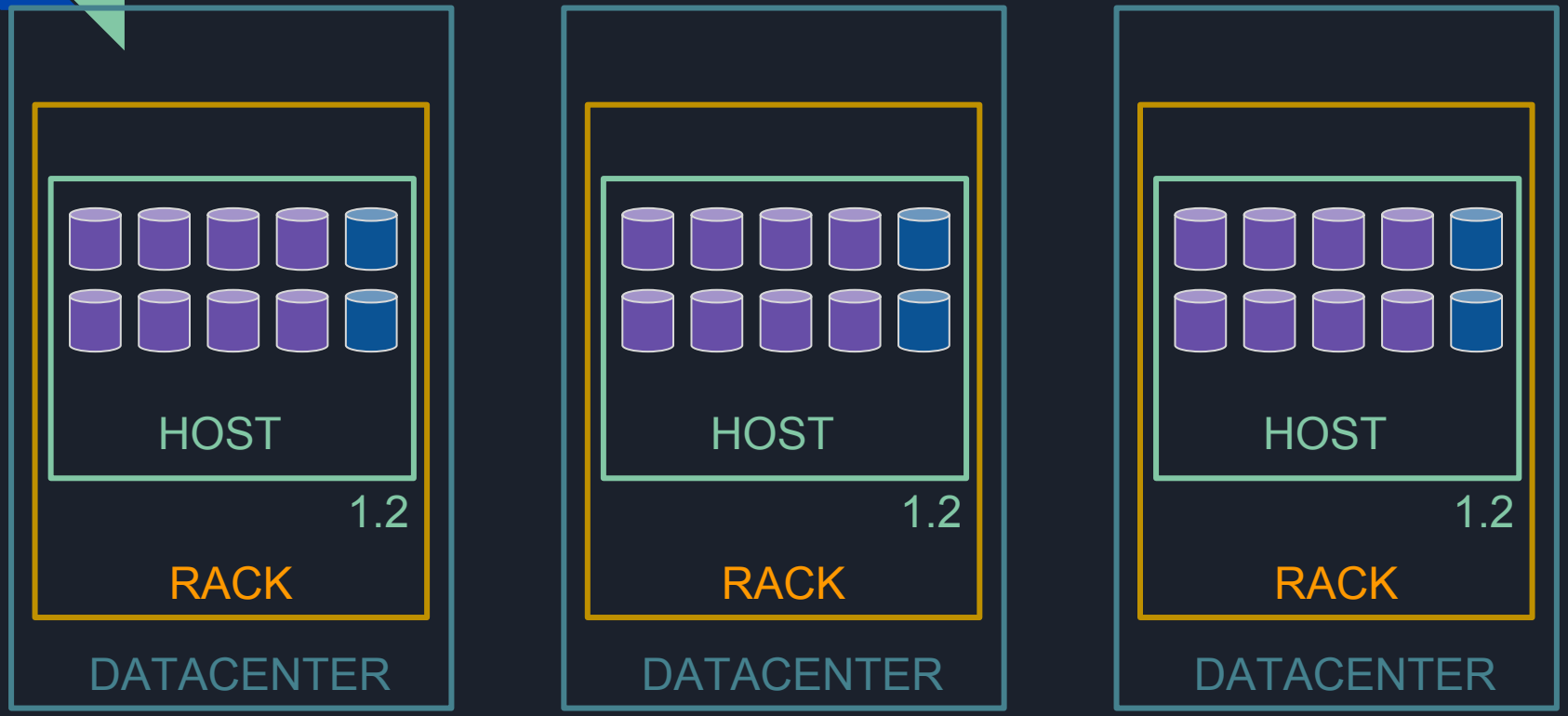
Fault tolerance

Irrecoverability:
- 3 rack failures
- 3 DC failures



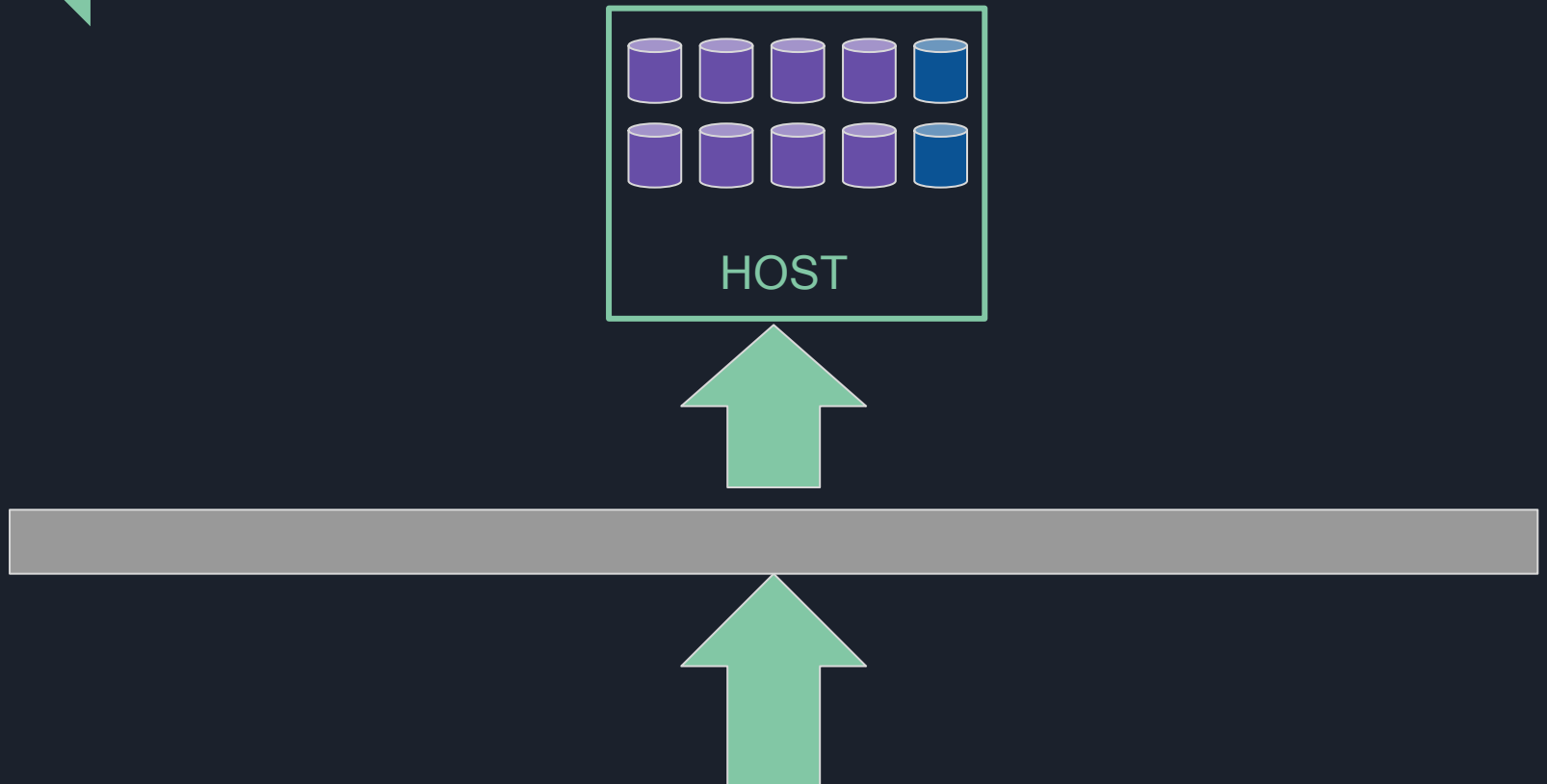
Fault tolerance

Replication factor:
1.2 * 3 = 3.6



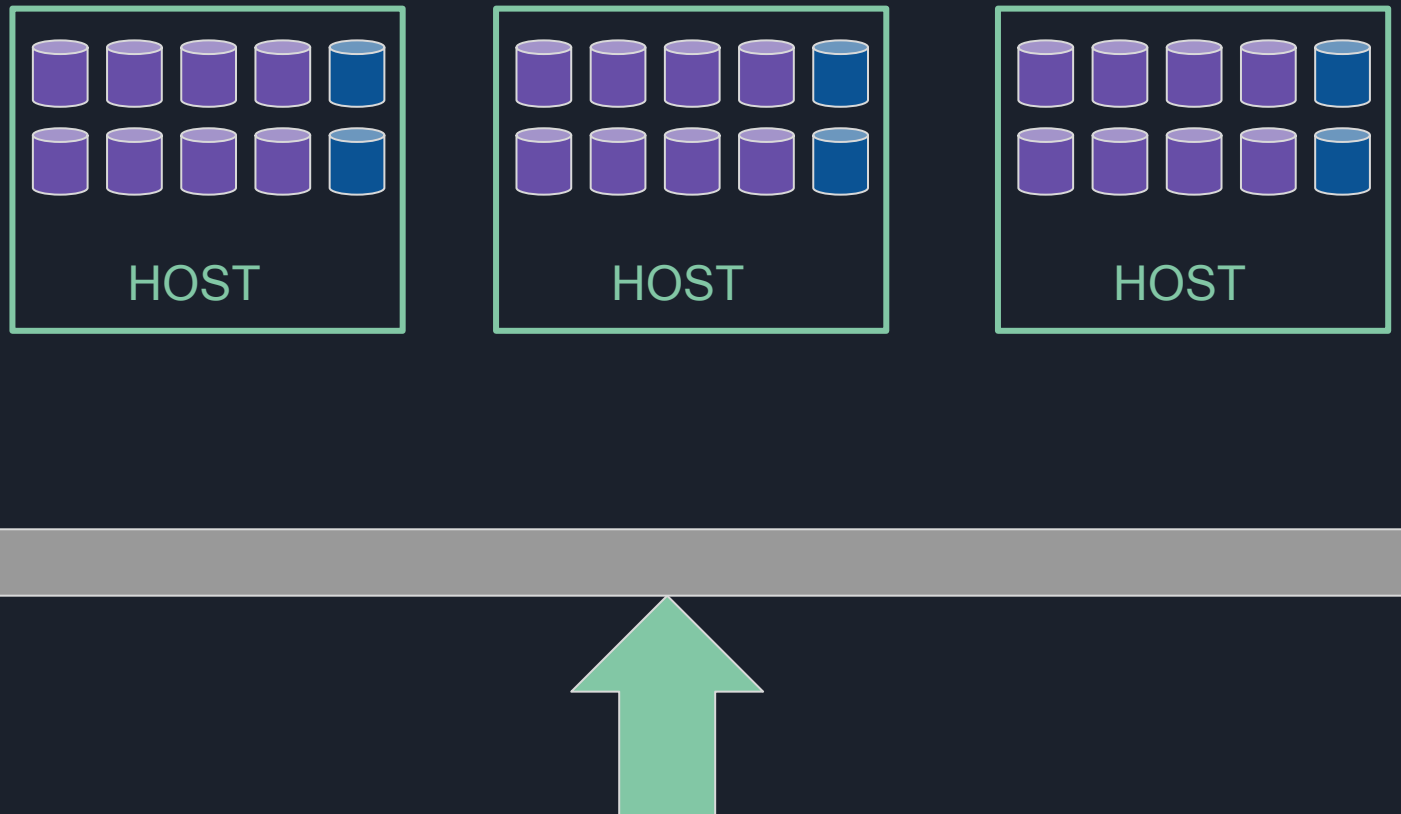
Handling load

Handling load

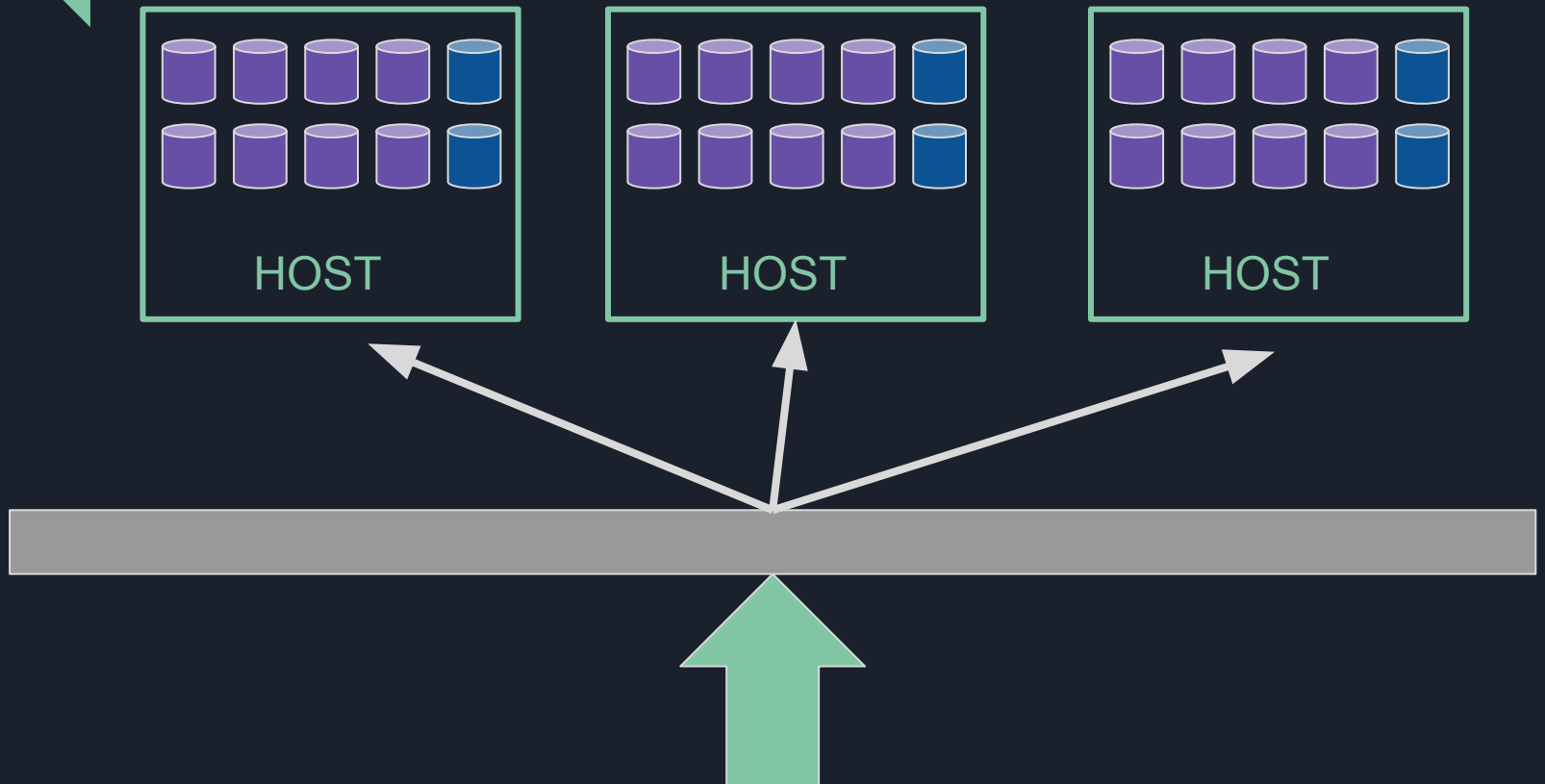


Handling load

Handling load



Handling load



Warm data

Can we reduce effective replication factor by relaxing throughput requirement while still being reliable?

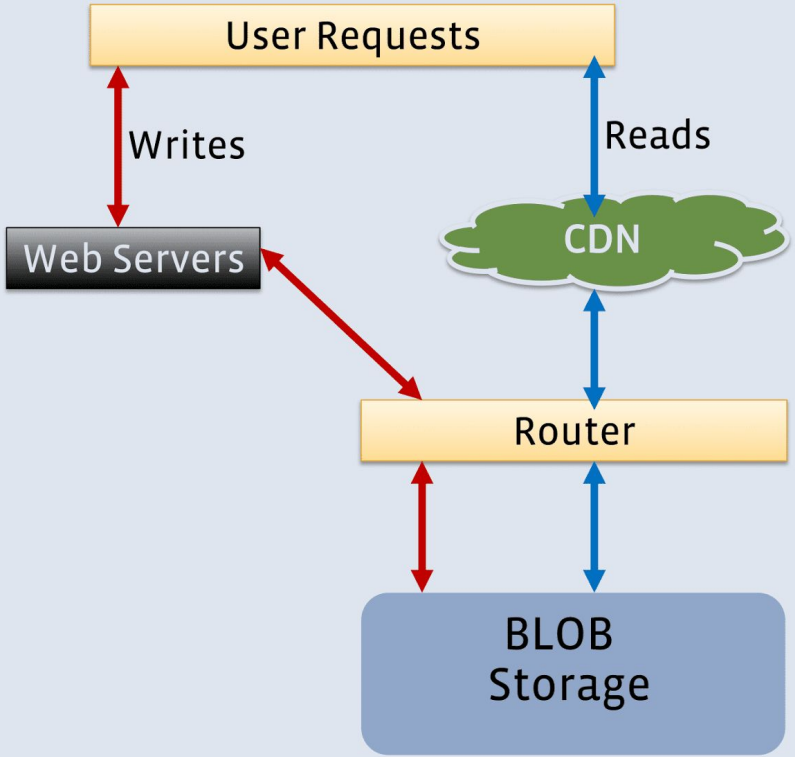
Warm data

Can we reduce effective replication factor by relaxing throughput requirement while still being reliable?

Can we somehow trade more time for less space?

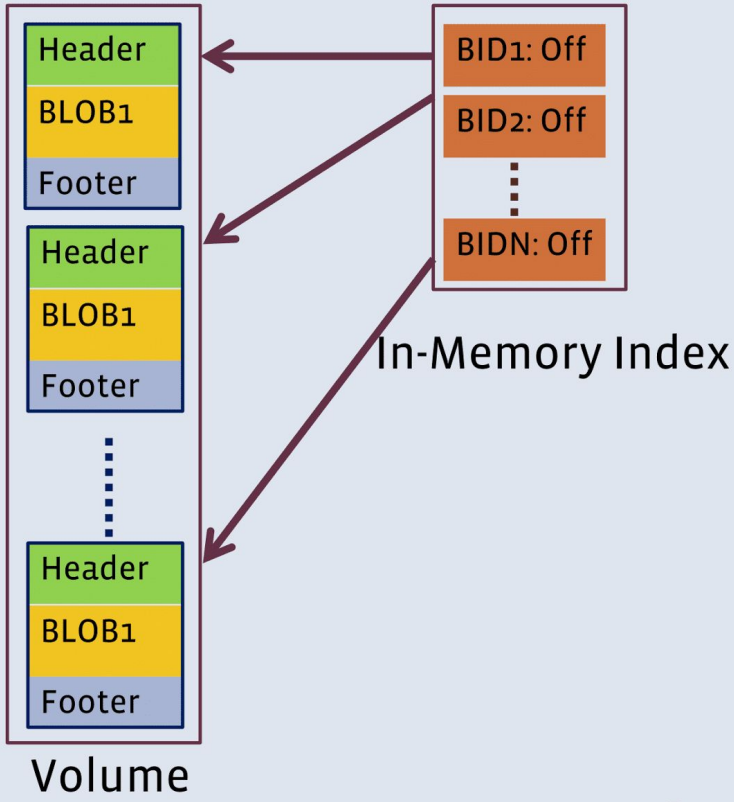
Background: Data serving

- CDN protects storage
- Router abstracts storage
- Web tier adds business logic



Background: Haystack [OSDI2010]

- Volume is a series of BLOBs
- In-memory index



f4: Haystack on cells

f4: Haystack on cells

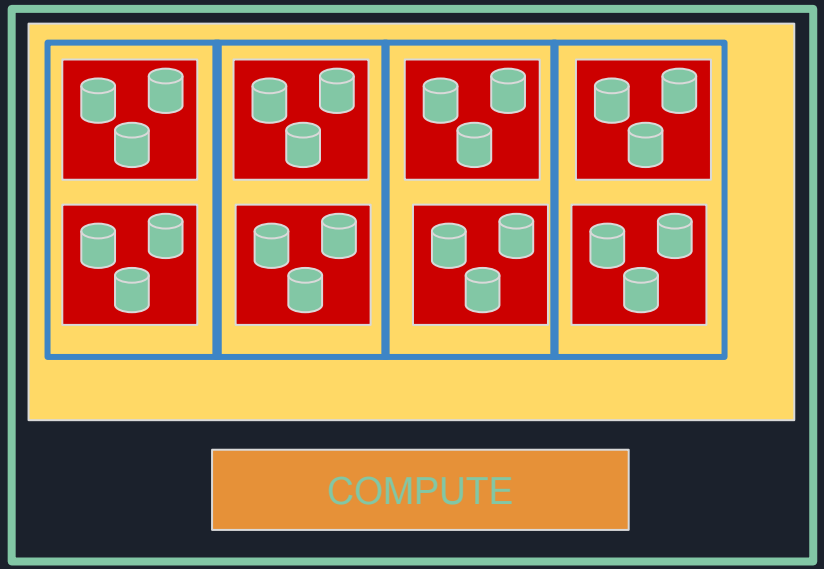
f4: Haystack on cells

Data + Index

CELL

f4: Haystack on cells

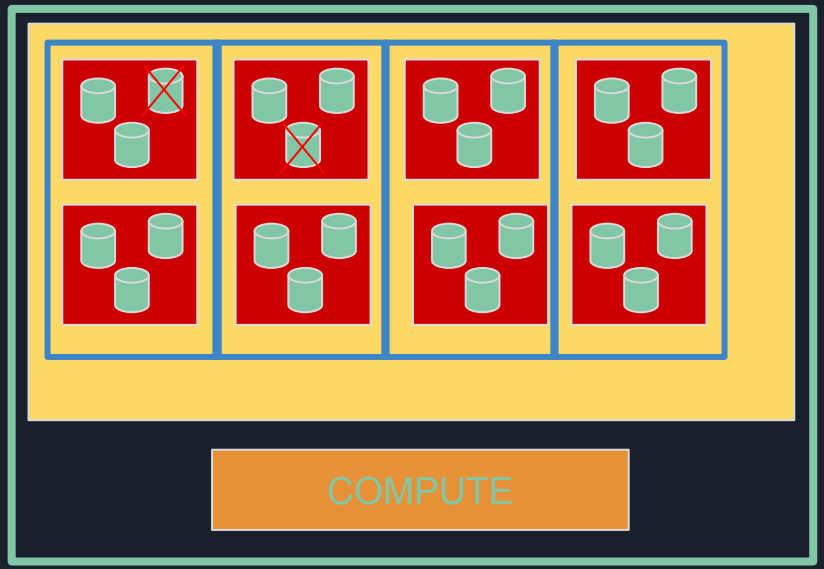
Data + Index



CELL

f4: Haystack on cells

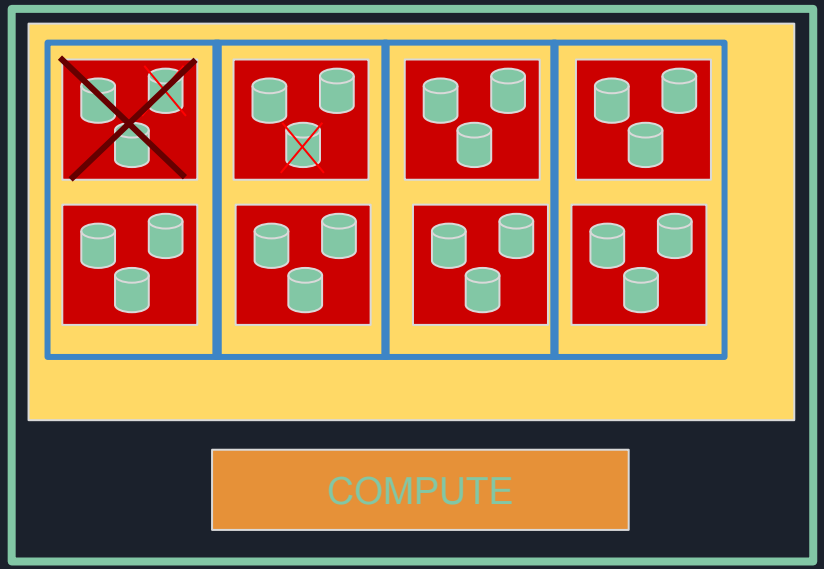
Data + Index



CELL

f4: Haystack on cells

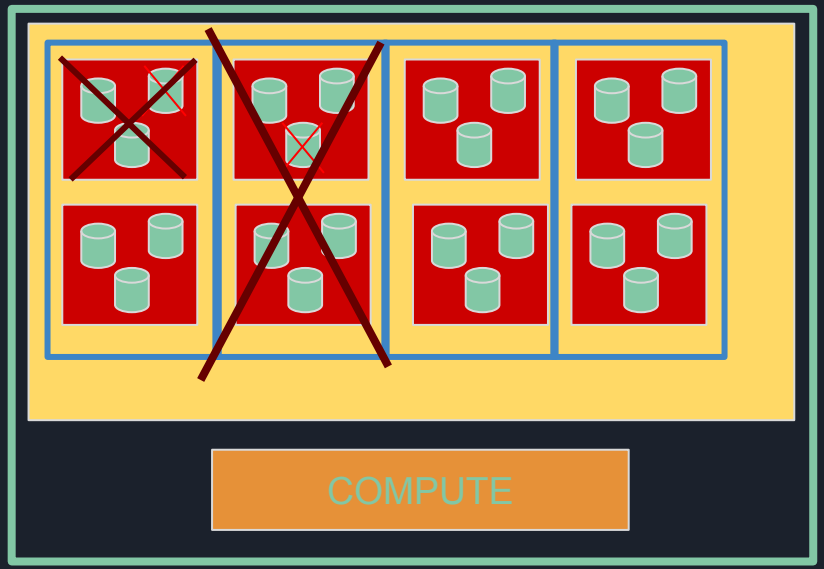
Data + Index



CELL

f4: Haystack on cells

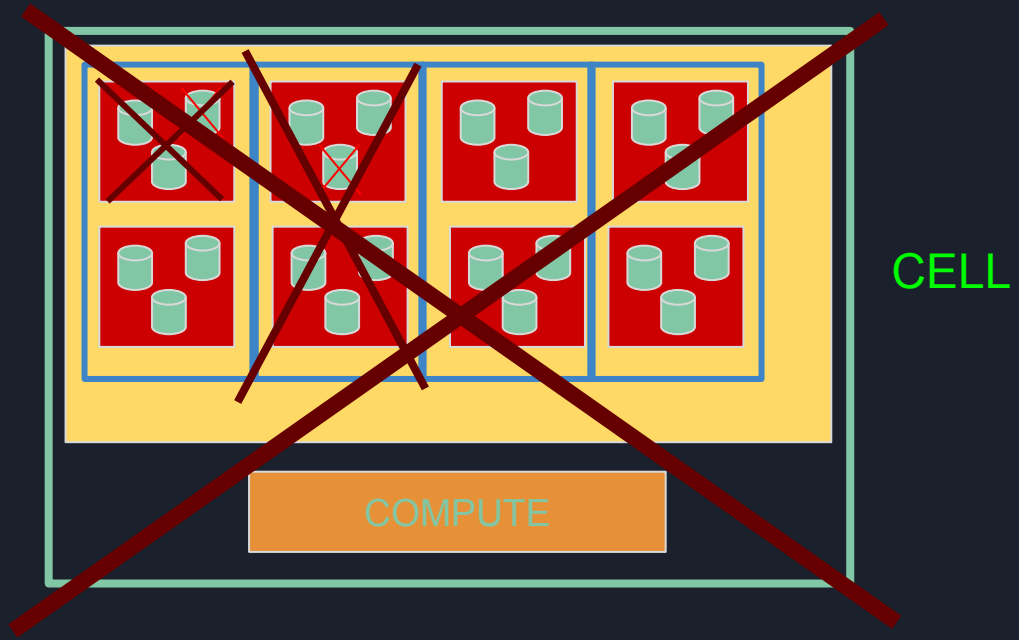
Data + Index



CELL

f4: Haystack on cells

Data + Index



Data splitting

10G Vol

Data splitting

10G Vol

Data splitting

10G Vol

Reed-Solomon
Encoding

Reed-Solomon
Encoding

Data splitting

10G Vol

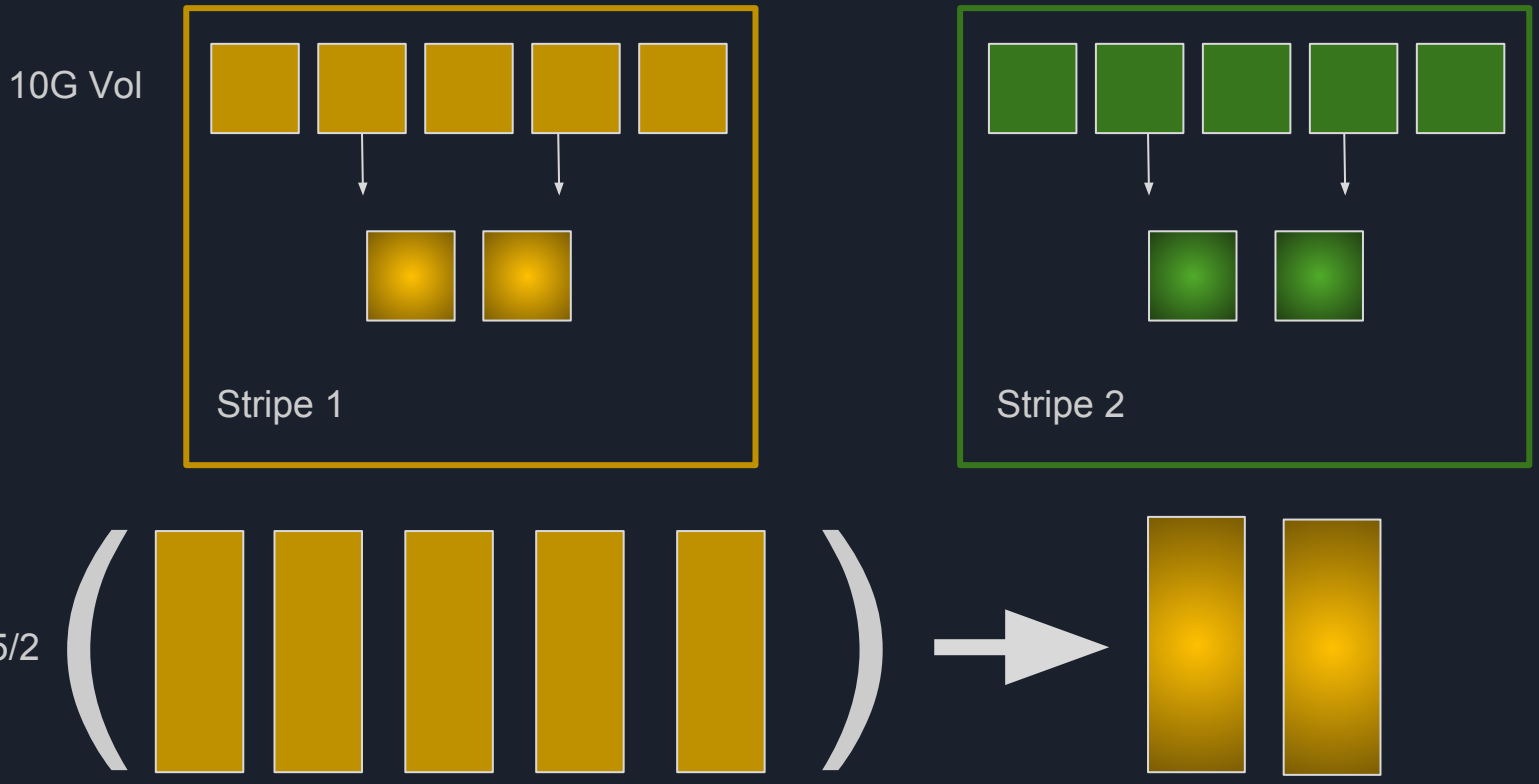
Reed-Solomon
Encoding

Reed-Solomon
Encoding

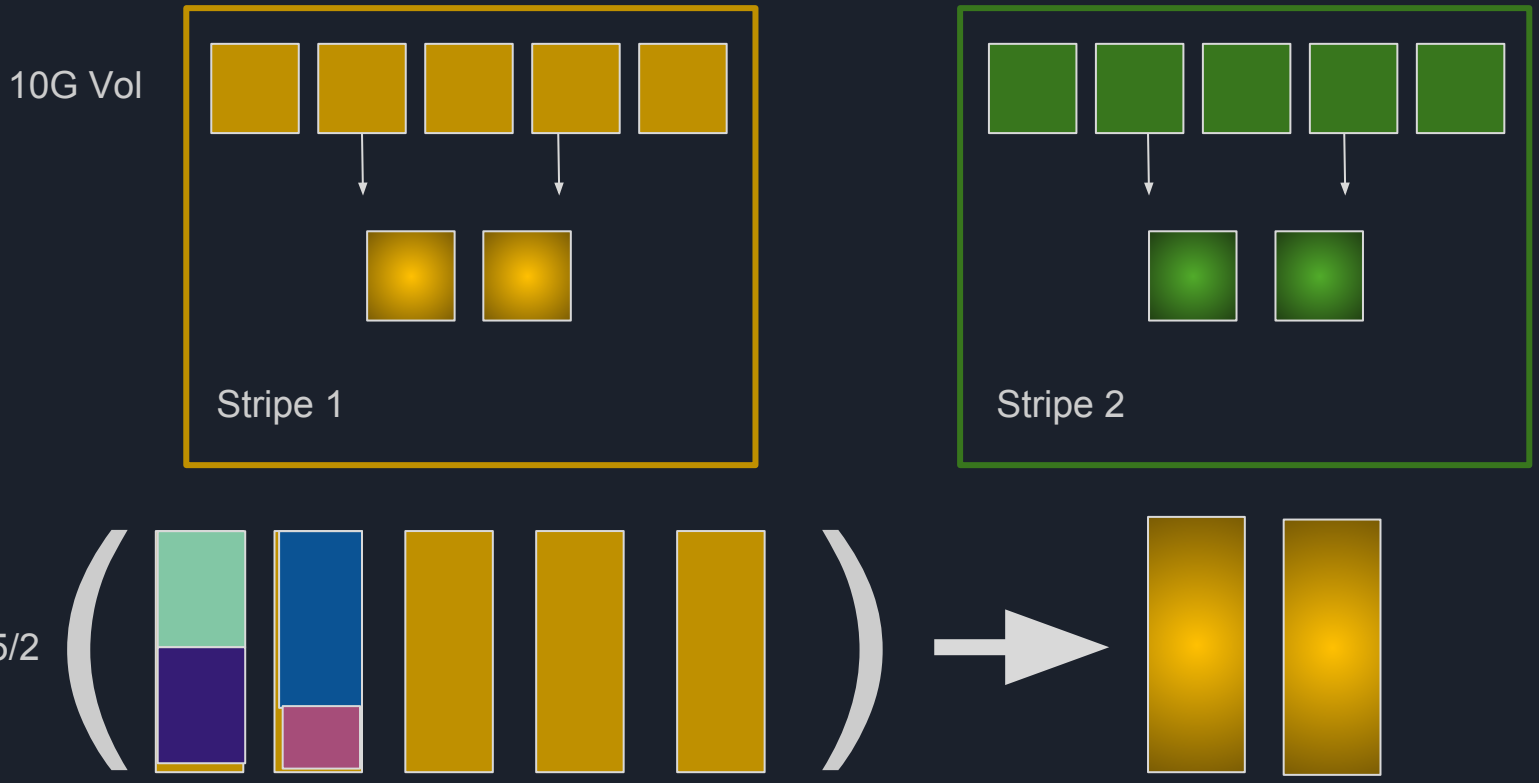
Data splitting

10G Vol

Data splitting



Data splitting



Data placement

10G Vol

Cell with
7 racks

Data placement

10G Vol

Cell with
7 racks

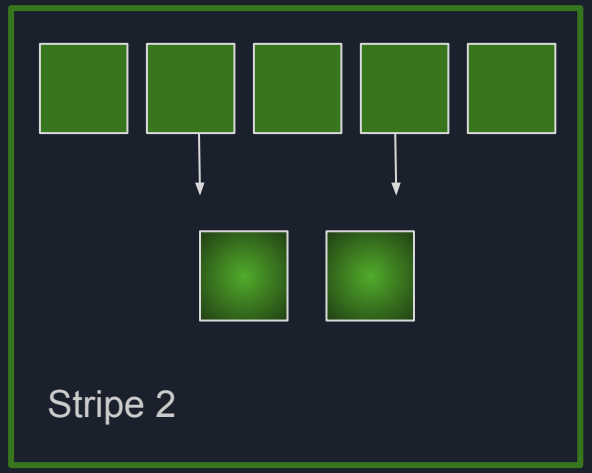
Data placement

10G Vol

Cell with
7 racks

Data placement

10G Vol

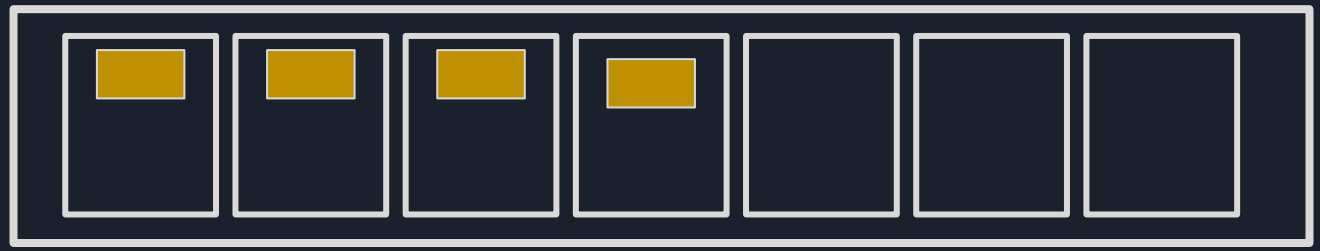


Cell with
7 racks

Data placement

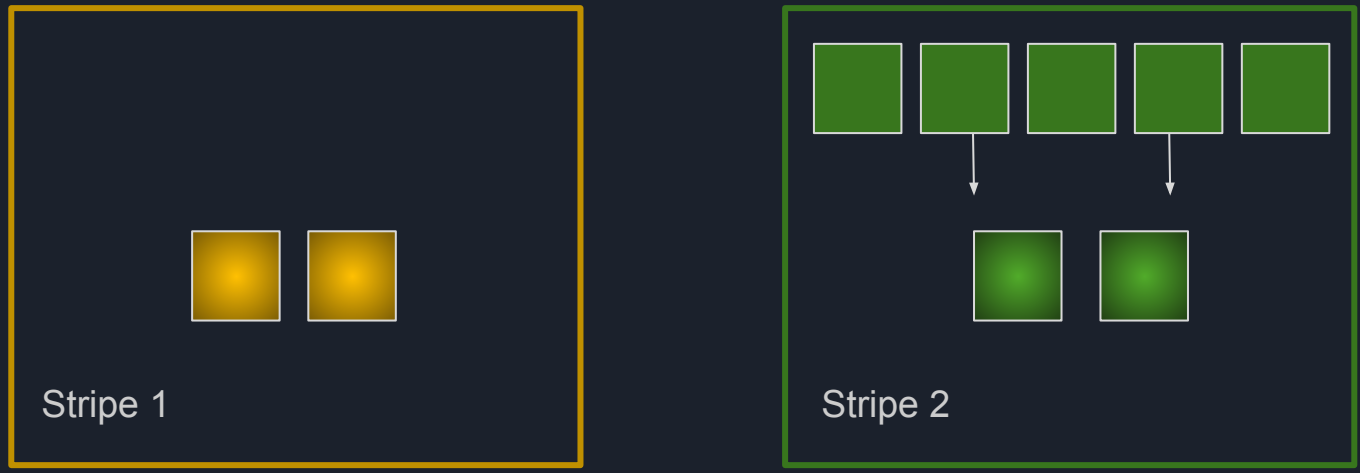
10G Vol

Cell with
7 racks

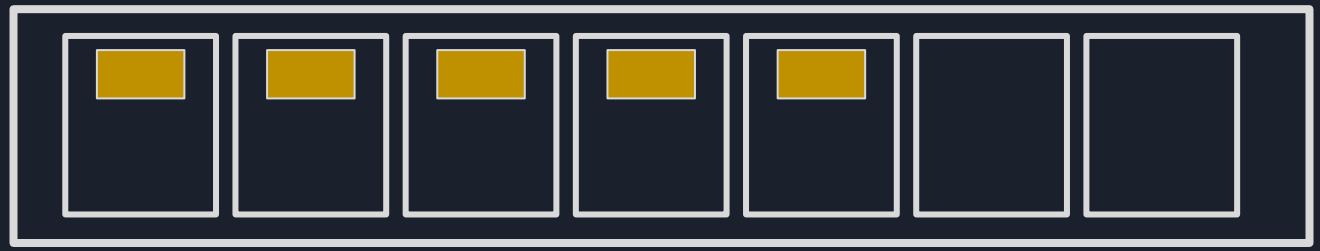


Data placement

10G Vol

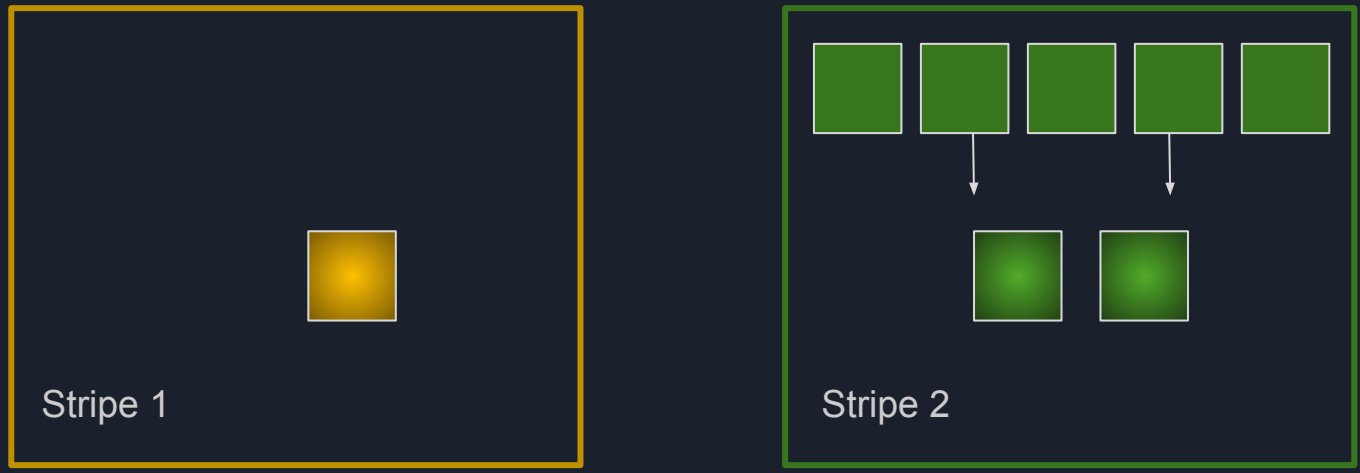


Cell with
7 racks

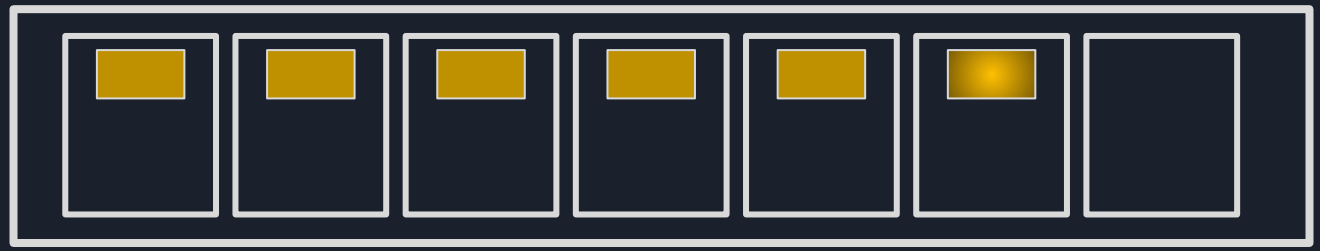


Data placement

10G Vol



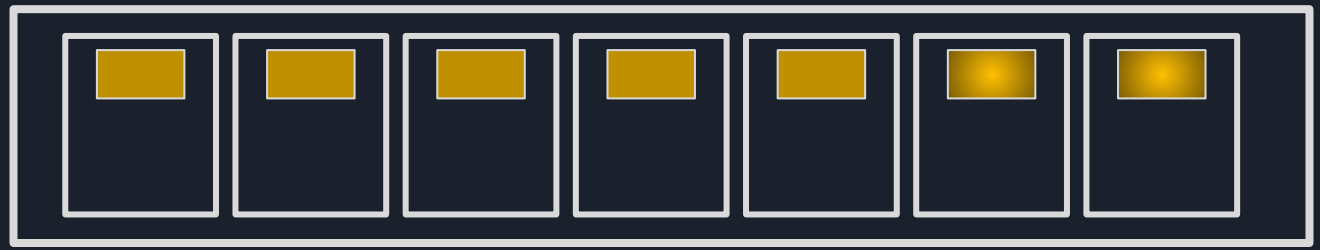
Cell with
7 racks



Data placement

10G Vol

Cell with
7 racks

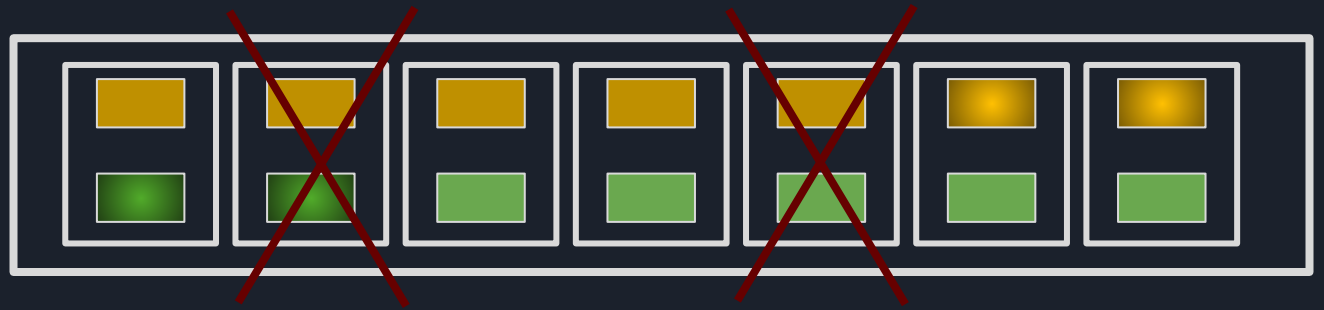


Data placement

Cell with
7 racks

Data placement

Cell with
7 racks

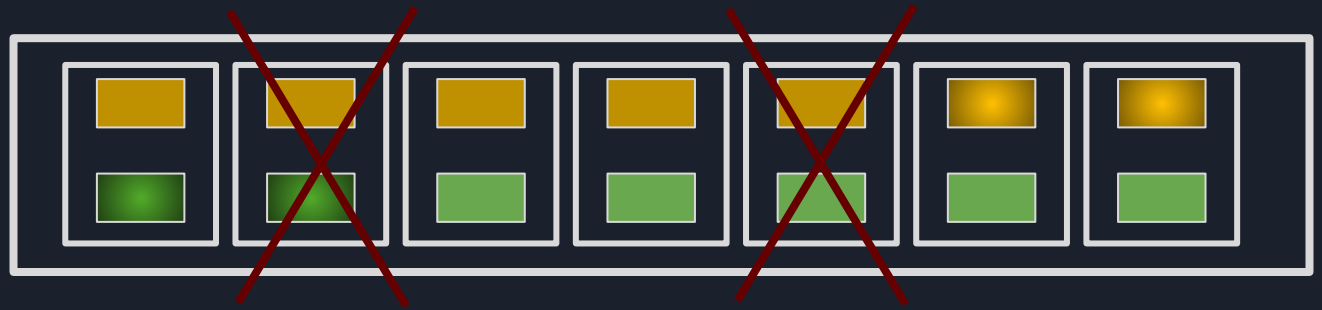


Data placement

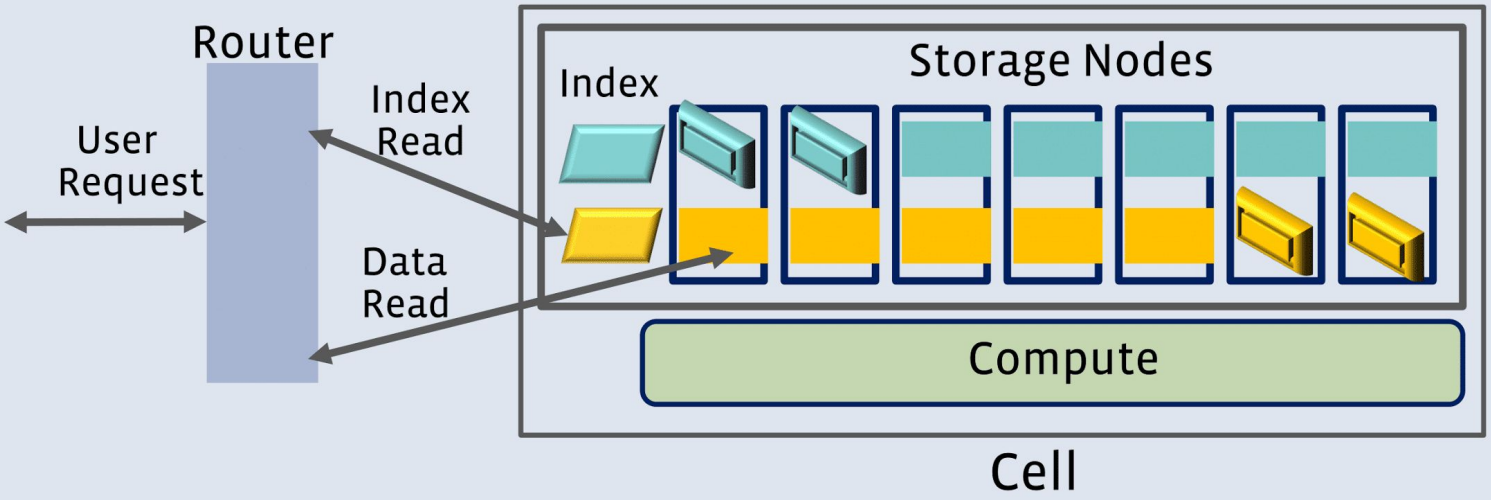
In practice:

- RS (10, 4) used - 1.4x replication factor
- Tolerates 4 disk/rack failures

Cell with
7 racks

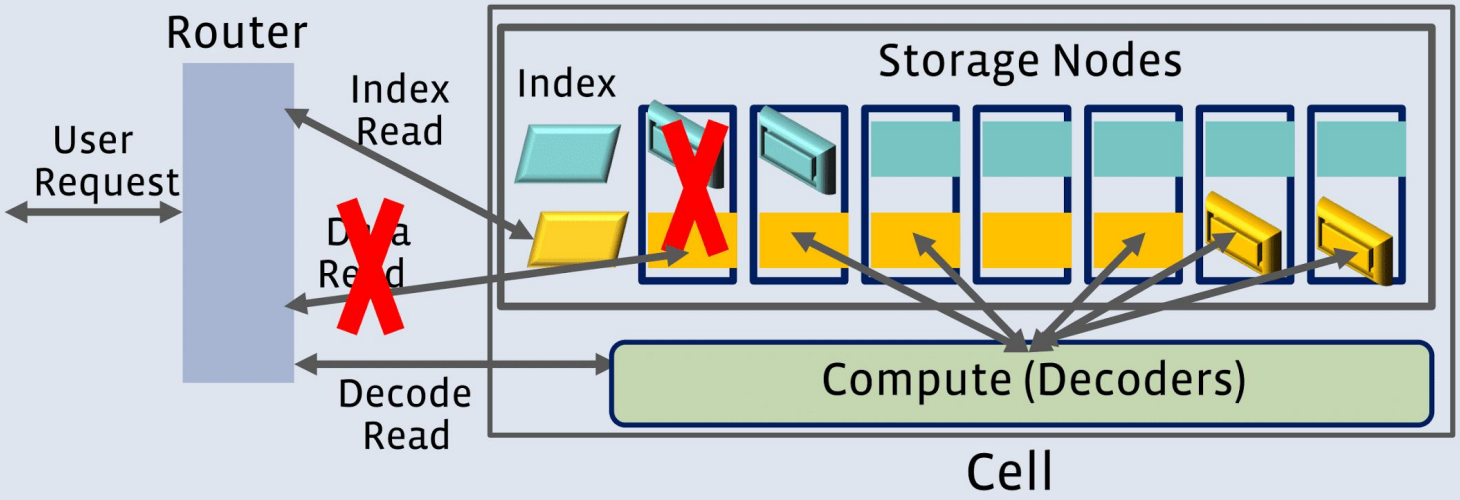


Reads



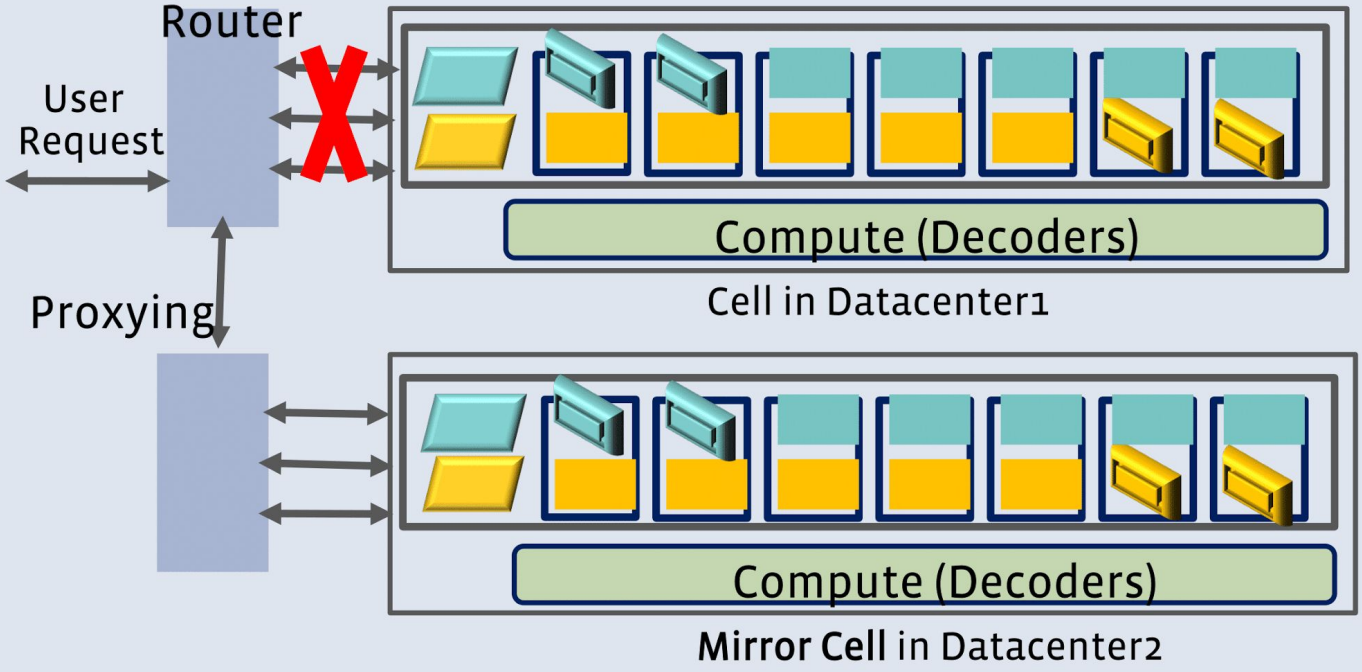
- 2-phase: Index read returns the exact physical location of the BLOB

Reads under cell-local failures



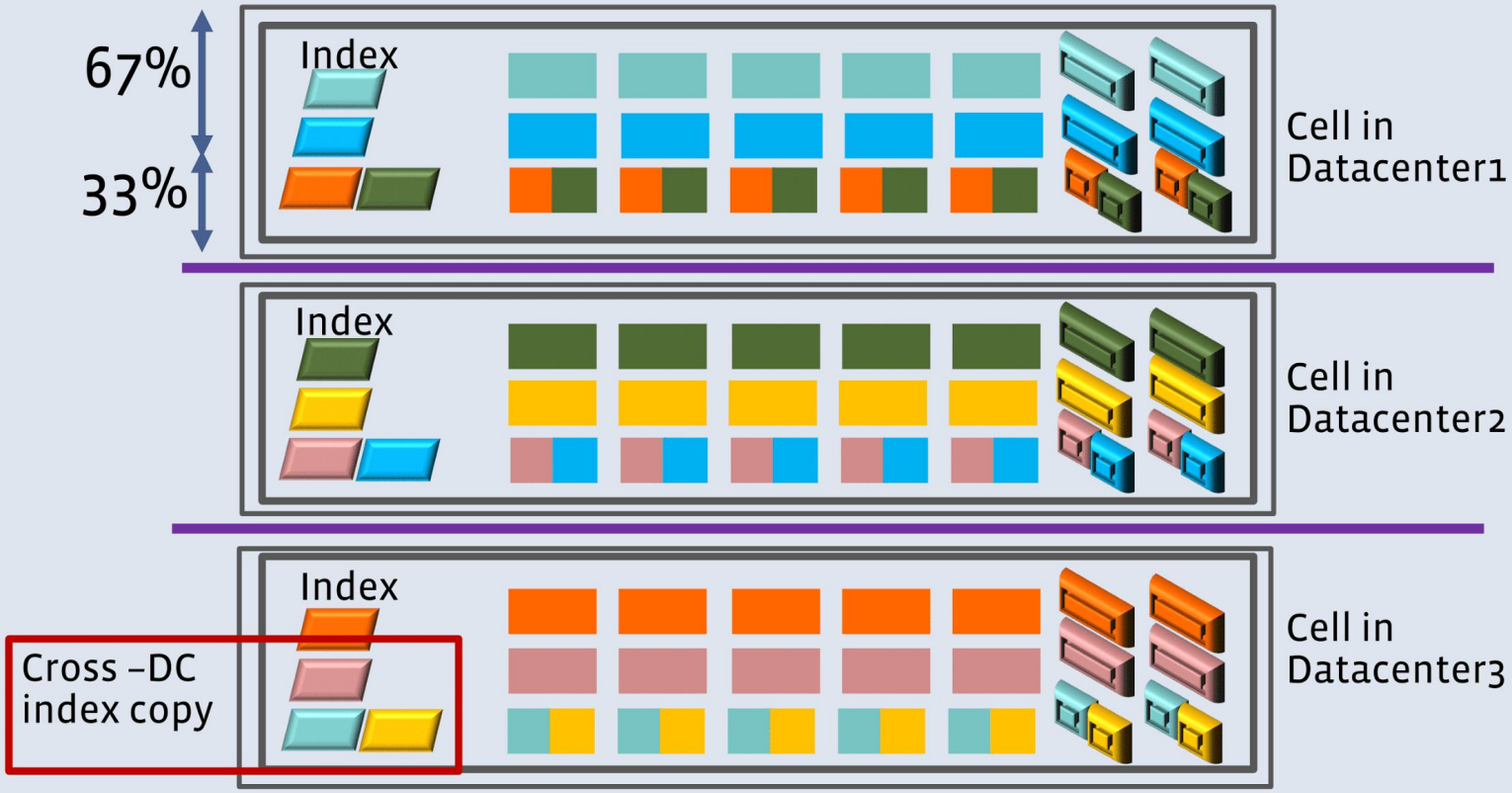
- Cell-Local failures (disks/hosts/racks) handled locally

Reads under datacenter failures (2.8X)

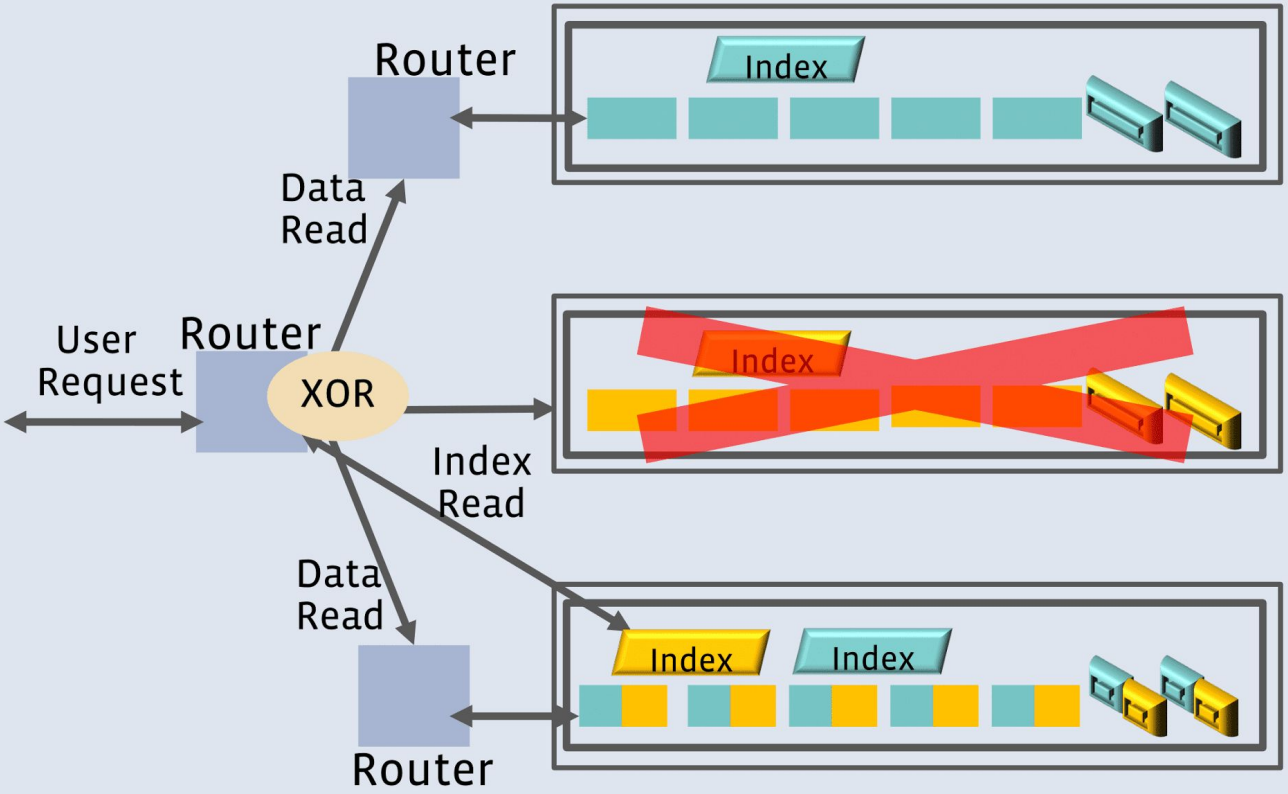


$$2 * 1.4X = 2.8X$$

Cross datacenter XOR (1.5 * 1.4 = 2.1X)



Reads with datacenter failures (2.1X)



Haystack v/s f4 2.8 v/s f4 2.1

	Haystack with 3 copies	f4 2.8	f4 2.1
Replication	3.6X	*2.8X*	*2.1X*
Irrecoverable Disk Failures	9	*10*	*10*
Irrecoverable Host Failures	3	*10*	*10*
Irrecoverable Rack failures	3	*10*	*10*
Irrecoverable Datacenter failures	*3*	2	2
Load split	*3X*	2X	1X

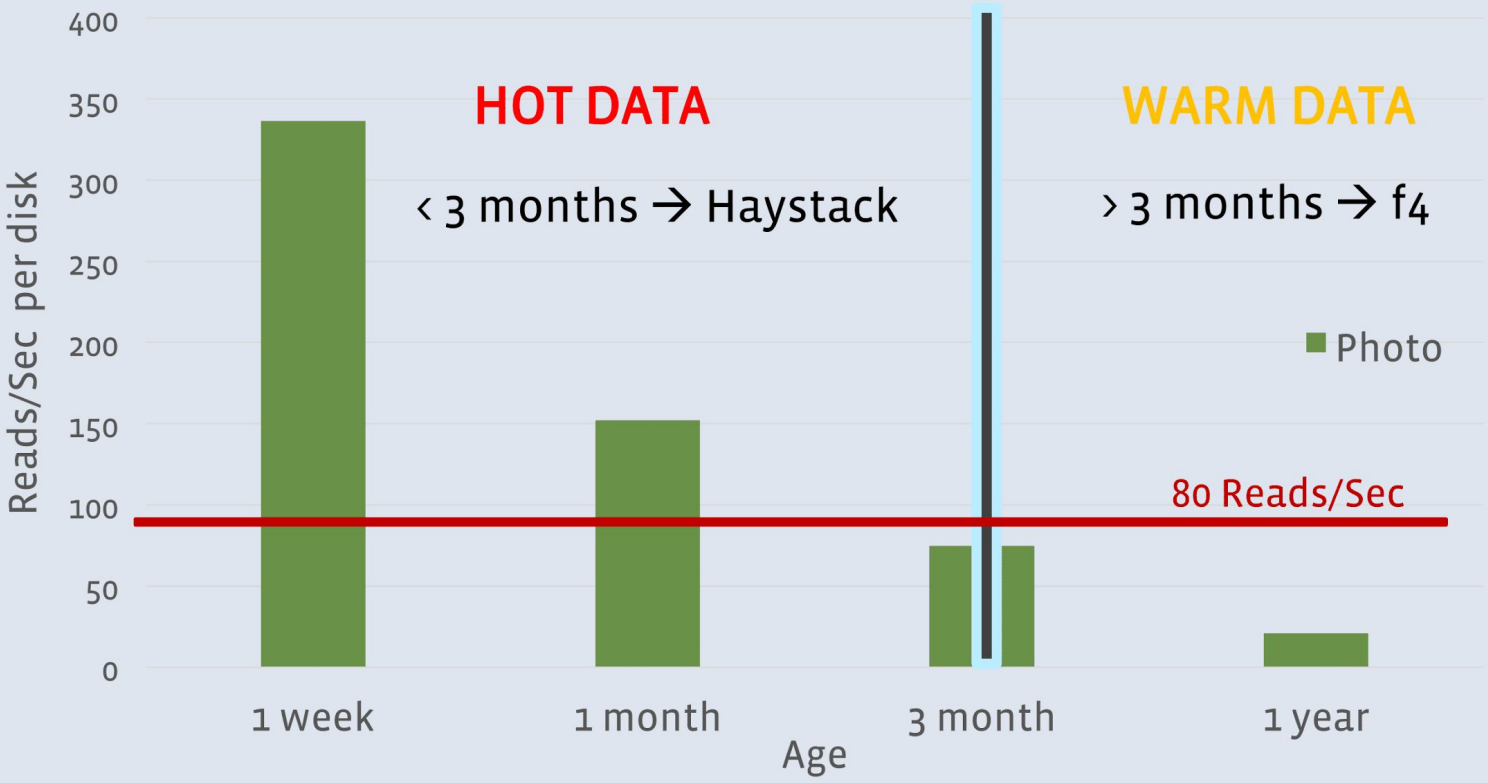
Evaluation

- What and how much data is “warm”?
- Can f4 satisfy throughput and latency requirements?

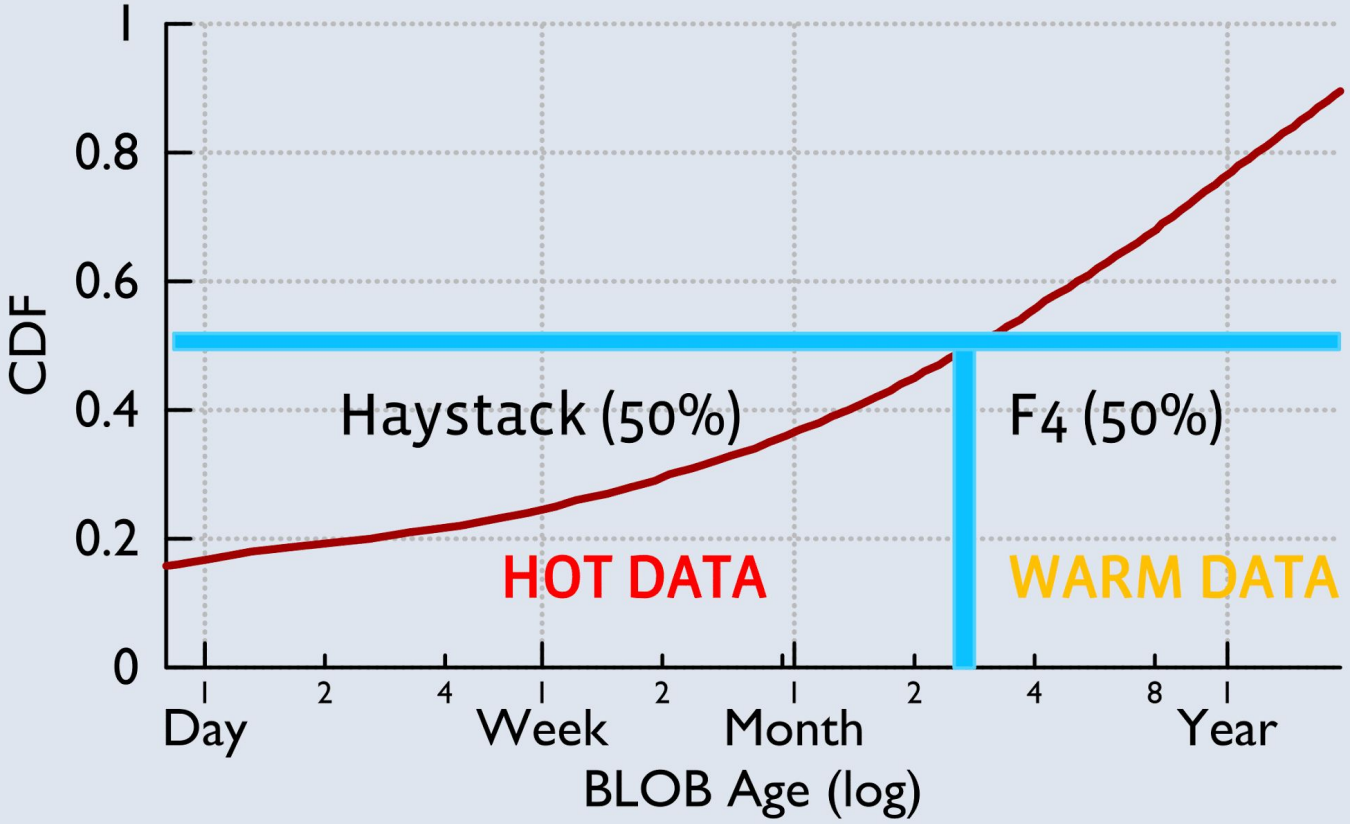
Methodology

- CDN data: 1 day, 0.5% sampling
- BLOB store data: 2 week, 0.1%
- Random distribution of BLOBs assumed
- The worst case rates reported

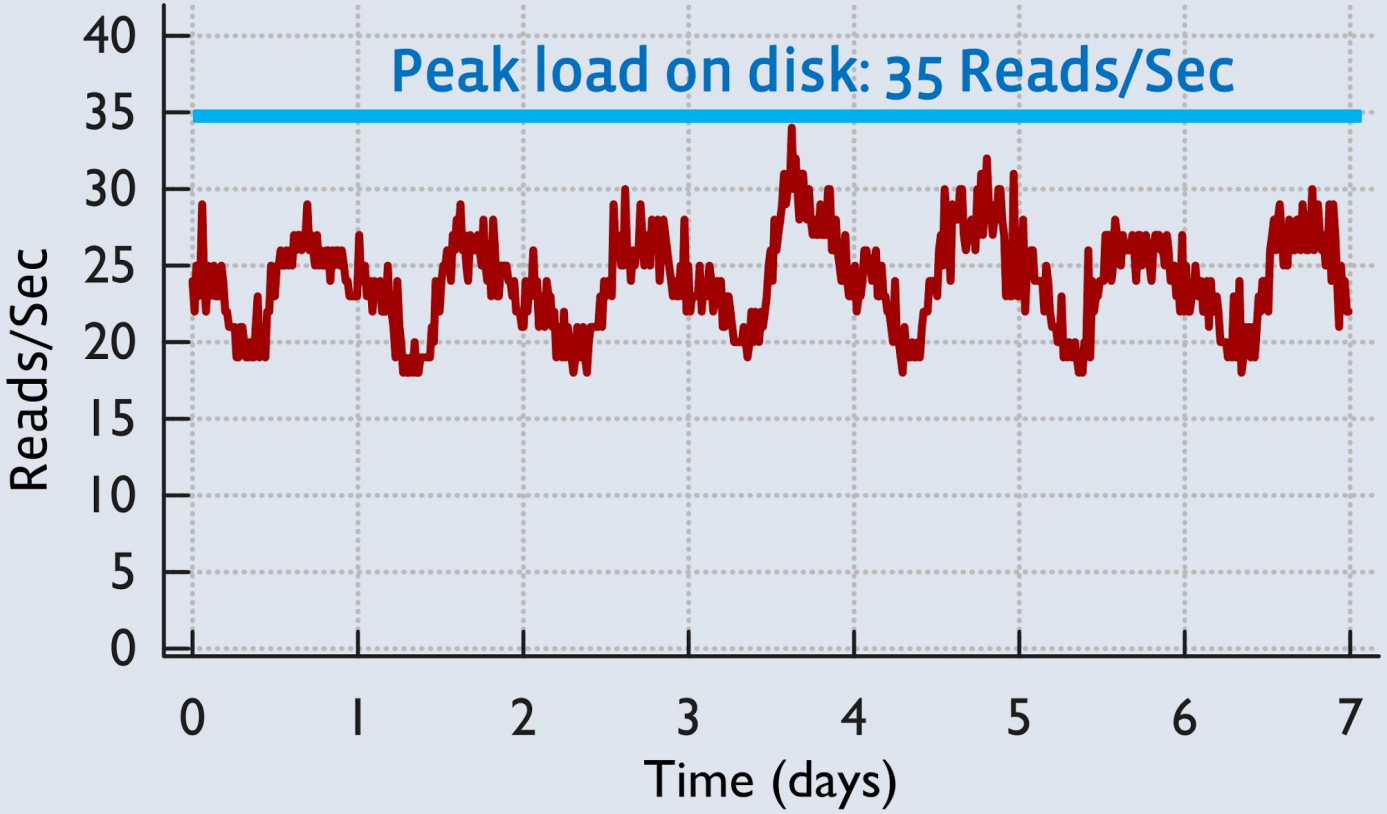
Hot and warm divide



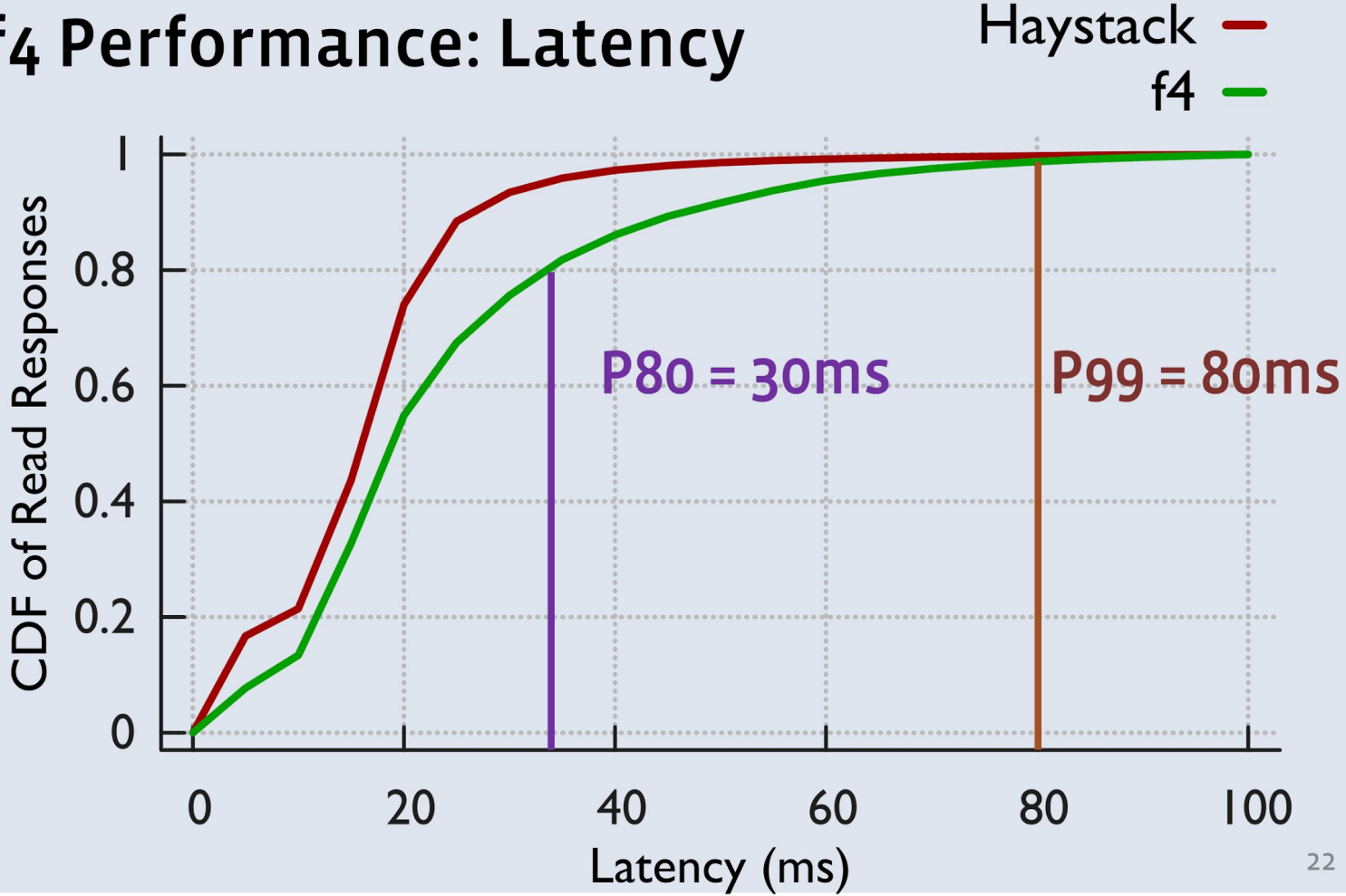
It is warm, not cold



f4 Performance: Most loaded disk in cluster



f4 Performance: Latency



Concluding Remarks

- Facebook's BLOB storage is big and growing
- BLOBs cool down with age
  - ~100X drop in read requests in 60 days
- Haystack's 3.6X replication over provisioning for old, warm data.
- f4 encodes data to lower replication to 2.1X