

HPTS 2026

Ten years of Magic Pocket

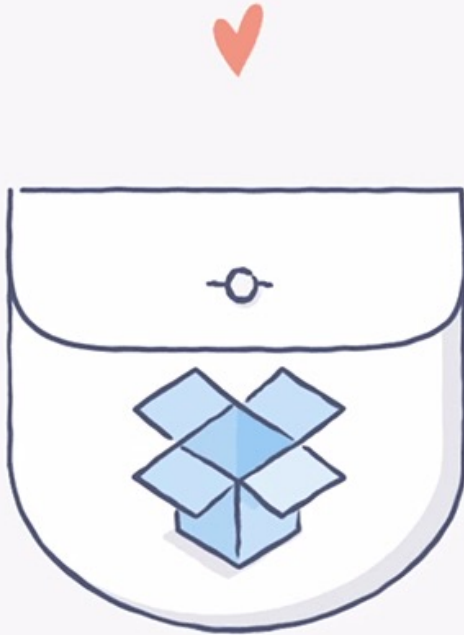
Lessons from scaling an on-prem exabyte-scale blob
store

Facundo Agriel / Dropbox



Dropbox's content store

A custom, on-prem, immutable blob store built for Dropbox user content



>10 EB

installed raw capacity

Trillions

of blobs

5 metros

independent operating regions

Billions

of deletes each day

From a few hundred petabytes to tens of exabytes of installed capacity

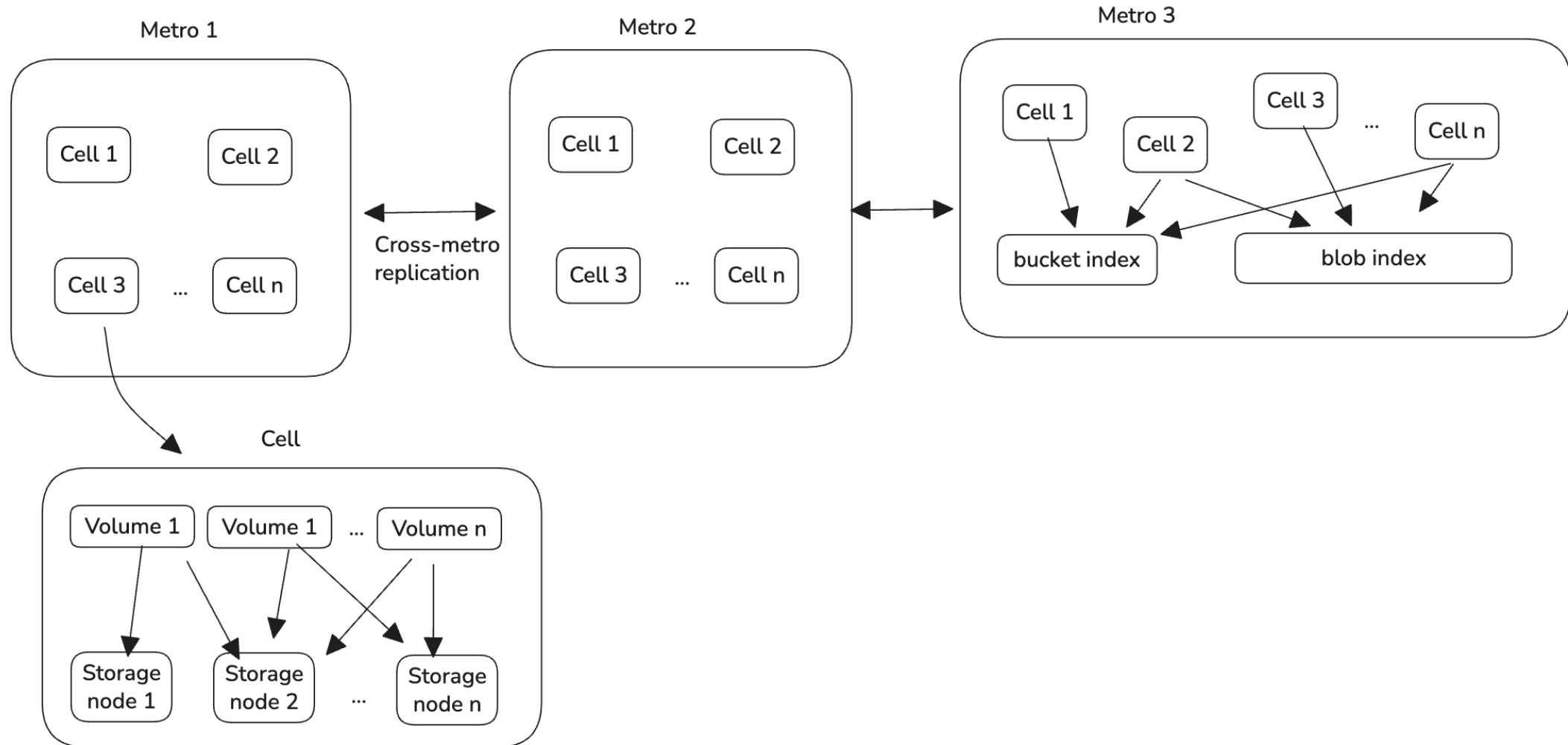
Lessons learned

01 **Maintaining durability under pressure**

02 **The importance of checksums and verifiers**

03 **The changing trade-offs of HDD storage**

High-level architecture



Isolation and failure domains

UNIT	ROLE	SCOPE AND BEHAVIOR
Blob	Content	Hash-addressed, immutable data
Volume / bucket	Durability and placement	Replicated or erasure coded type. Volumes are open or closed.
Cell	Logical isolation	Own coordination for erasure coding, repairs, optimizations and host allocation.
Data center / cluster	Physical failure domain	Site and network infrastructure shared by cells.
Zone	Storage isolation	Many cells in one zone.
Metro	Geographic failure domain	Physical location that can contain multiple zones.

LESSON 1

Maintaining durability under pressure

Designing a durability model

01 AFR

Annual failure rate.
Hardware driven.

02 Repair rate

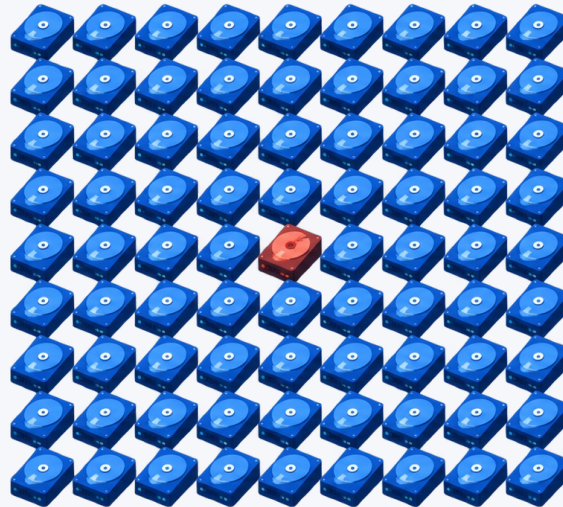
How quickly failed fragments are repaired.
Mostly limited by I/O.

03 Redundancy

Erasure coding or replication.
How many failures should we tolerate,
and at what cost?

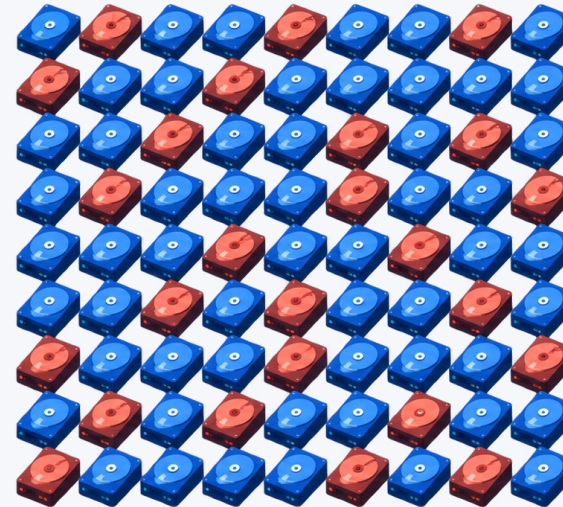
Annual failure rate

VERY LOW AFR



Failures rarely overlap

VERY HIGH AFR

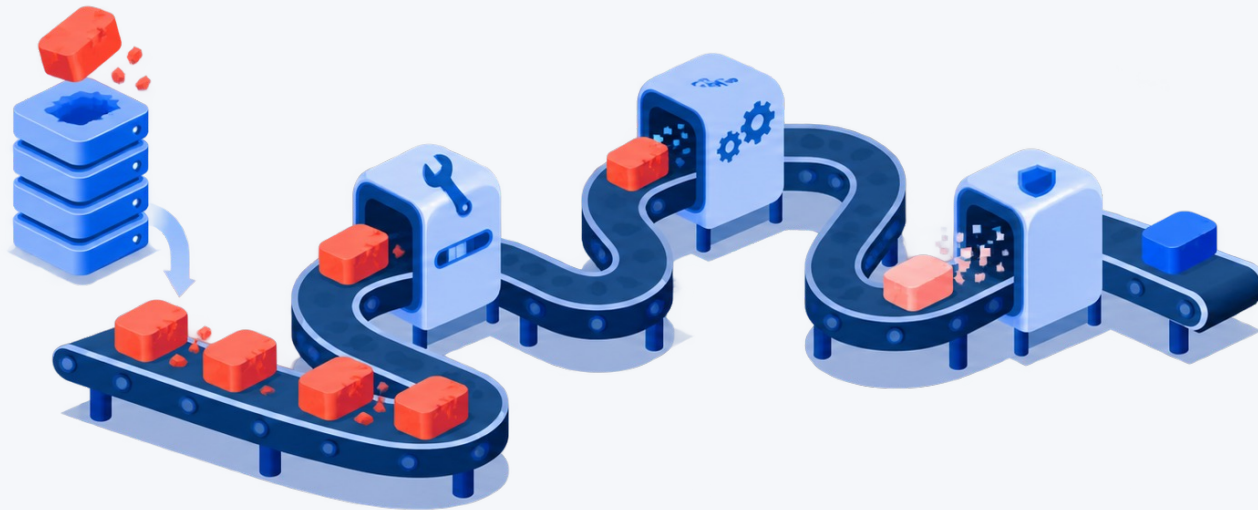


More failures overlap

Higher AFR increases the chance of overlapping failures.

Repair rate

SLOW REPAIR



A longer exposure window

FAST REPAIR

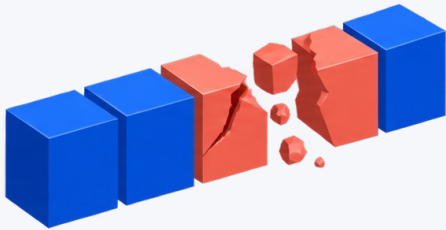


A shorter exposure window

Fast repair shrinks the window for another failure.

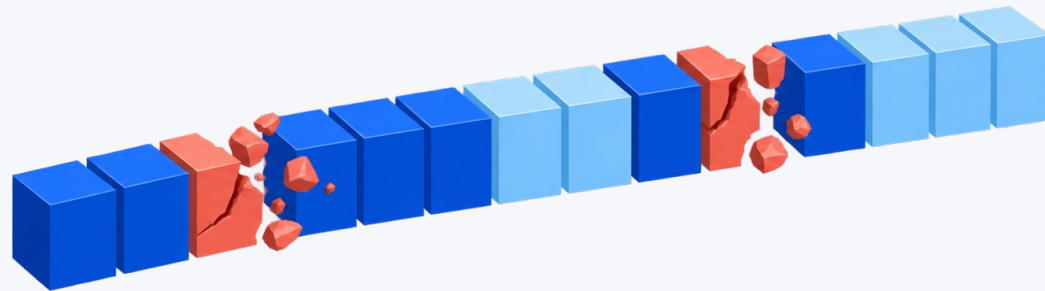
Redundancy

LOW REDUNDANCY



Few failures tolerated

HIGH REDUNDANCY



Many failures tolerated

More redundancy comes at a cost!

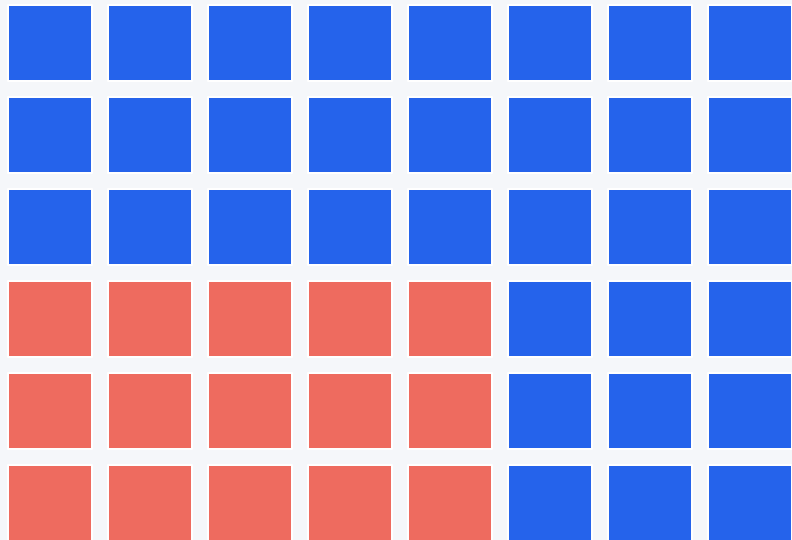
Designing a durability model

- | | | |
|----|--------------------|--------------------------------------|
| 01 | AFR | Annual failure rate |
| 02 | Repair rate | Fast is better |
| 03 | Redundancy | Less is cheaper, but more is better! |

Anything wrong with this?

First example: Drives hate the heat

HUNDREDS OF PB ACROSS THE FLEET



EXPECTED RANGE

HIGH-TEMPERATURE COHORT

- 1 High temperatures affected 30% of specific drives
- 2 AFR shot up for this revision, but only for these drives
- 3 Quickly built system to drain those drives early
- 4 Capacity runway and accelerated spares covered the moves

Second example: SSDs wear out together



Preparing for failure

Failures don't care about your durability models.

Prepare for the worst.

Identify single points of failure.
Communicate the risks,
including the unknowns.

SEE

Per-drive health and
environmental statistics

TARGET

Cohort-level drain controls
instead of fleet-wide action

MOVE

Capacity runway and repair bandwidth
reserved for surprises

SUBSTITUTE

Qualified heterogeneous drives
in the same storage platform and spread them around

LESSON 2

The importance of checksums and verifiers

Verifiers and checksums

VERIFIERS

**Is my data where
it's supposed to be?**

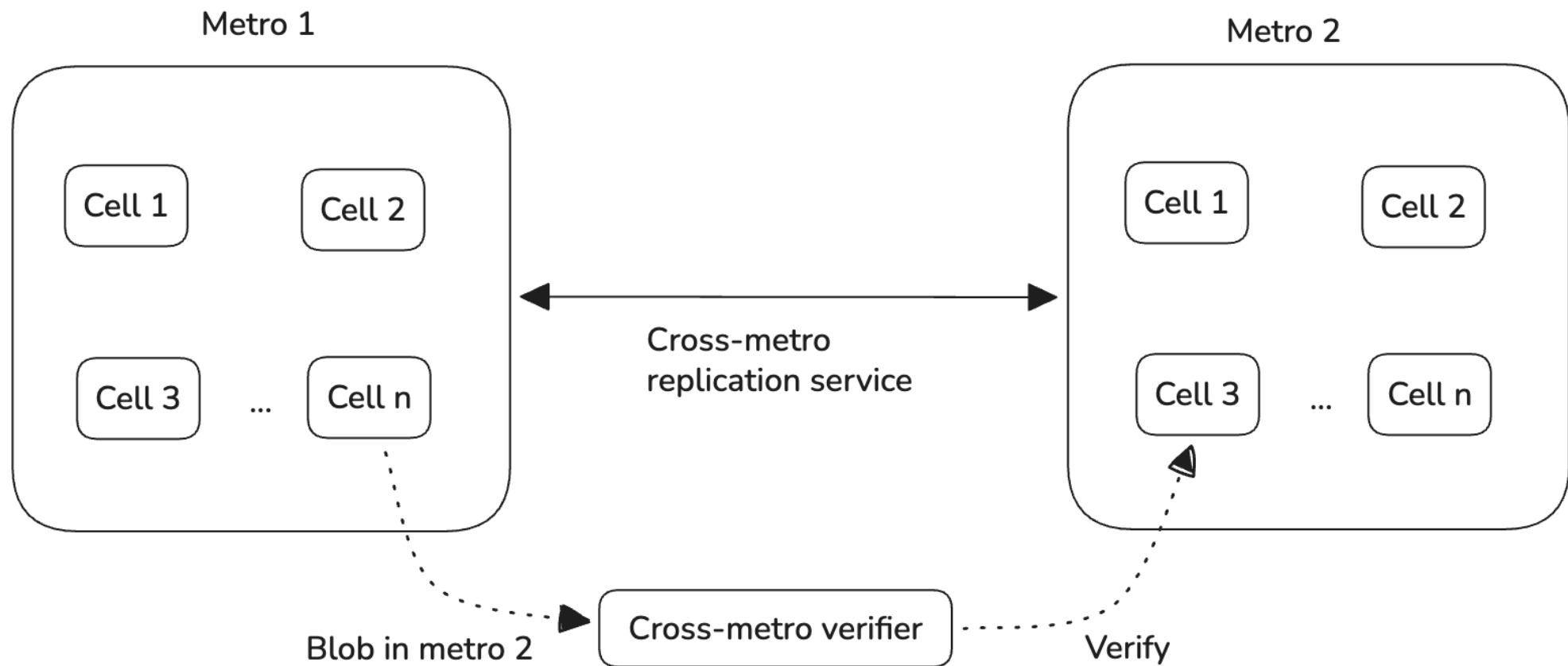
Confirm and validate
the presence of state.

CHECKSUMS

Is my data corrupted?

Check data *integrity*.

Example: Cross-metro verifier



Verifying disk integrity

1–2 GiB

**Checksum granularity
for all data**

Every disk goes through
a scrubbing cycle.

A very expensive process!

On write

Compute a checksum for each
physical volume on disk.

During scrubbing

Continuously read,
compute checksum and verify.

Checksums at every handoff

BLOB DATA

Every blob has a checksum.

Verify at every handoff from disk to memory and between services.

BLOB METADATA

Metadata on disk gets its own checksum.

Includes lists of metadata.

All state on disk will experience integrity issues.

LESSON 3

The changing trade-offs of HDD storage

HDD costs, density and I/O

01

Increased
costs

02

More
density

03

Similar
I/O

HDD costs, density and I/O

01

Increased
costs

02

More
density

03

Similar
I/O

So what?

Storage hardware generations over time

2013

Big Show

144 TB

4 TB per drive

36 drives per server



2014

Andre

192–240 TB

4 TB per drive

48–60 drives per server



2016

Atlas

270 TB

6–8 TB per drive

35–45 drives per server



2017

Axel

624–720 TB

8 TB per drive

78–90 drives per server



144 TB to 720 TB per server

Drive and chassis density are pushing servers > 3PB

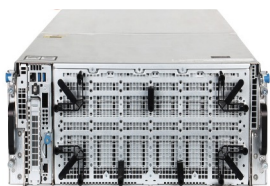
2018

Stimpy

1.4–1.6 PB

14–16 TB per drive

~100 drives per server



2021

Scooby

1.8–2.6 PB

18–26 TB per drive

~100 drives per server



2024

Sonic

2.7 PB+

26 TB+ per drive

96 drives per server



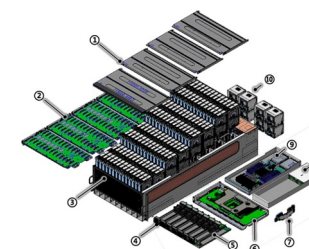
2026

Shadow

>3.6 PB

36 TB+ per drive

112 drives per server



14–16 TB to 36 TB+ drives

For a decade, density reduced unit cost

~10x

drive capacity growth

over the last 13 years

10–25%

unit-cost step

with a new drive generation,
typically, every few years

The cost model expected cheaper capacity every few years.

For a decade, density reduced unit cost

~10x

drive capacity growth

over the last 13 years

10–25%

unit-cost step

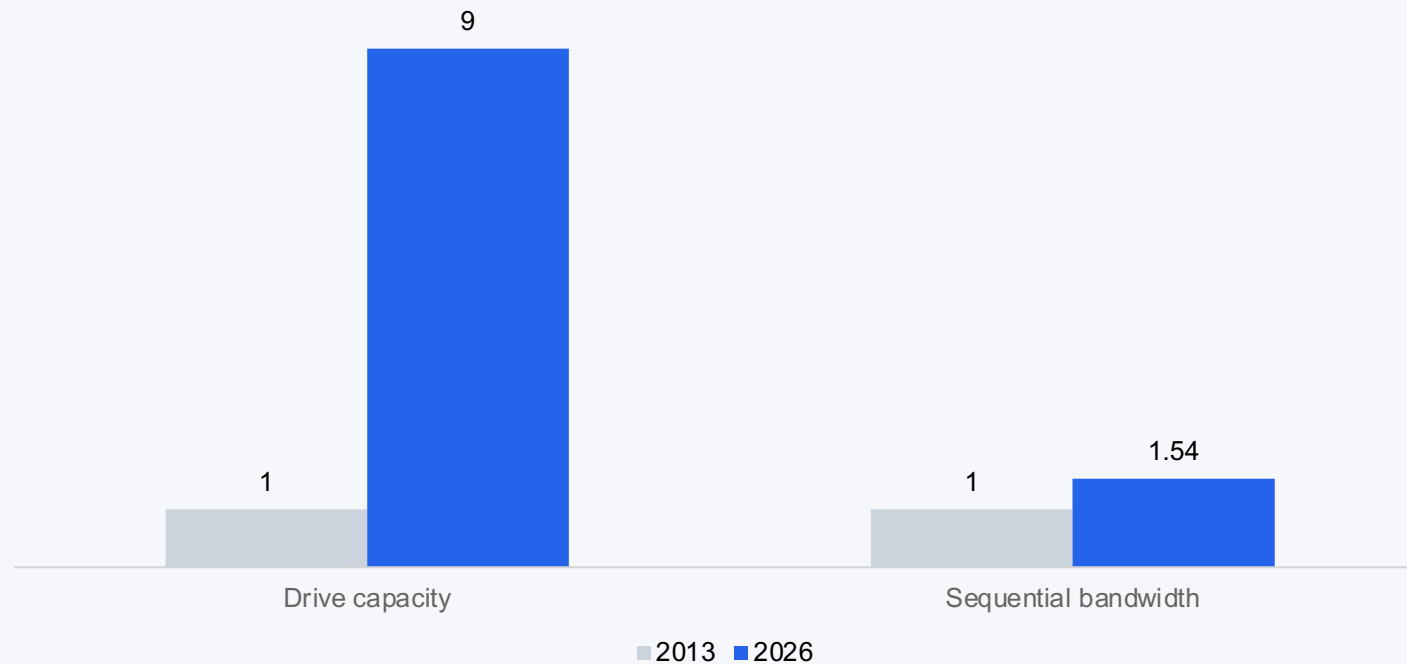
with a new drive generation,
typically, every few years

The cost model learned to expect cheaper capacity every few years.

Life is good!

Drive capacity grew 9x while bandwidth grew 1.5x

NORMALIZED TO 2013



-83%

sequential bandwidth per TB

2013 example
4 TB / 175 MB/s / 7,200 RPM

2026 example
36 TB / 270 MB/s / 7,200 RPM

Denser drives, same I/O → tighter I/O budgets

5.8x

Longer ideal full-drive
sequential pass

4 TB at 175 MB/s	6.35 h
36 TB at 270 MB/s	37.04 h

LOWER STORAGE OVERHEAD

Requires efficient compaction
to keep volumes dense.

HIGHER DRIVE DENSITY

Gives each stored TB
less media bandwidth.

FAST REPAIR

Competes for the same disks,
network and destinations.

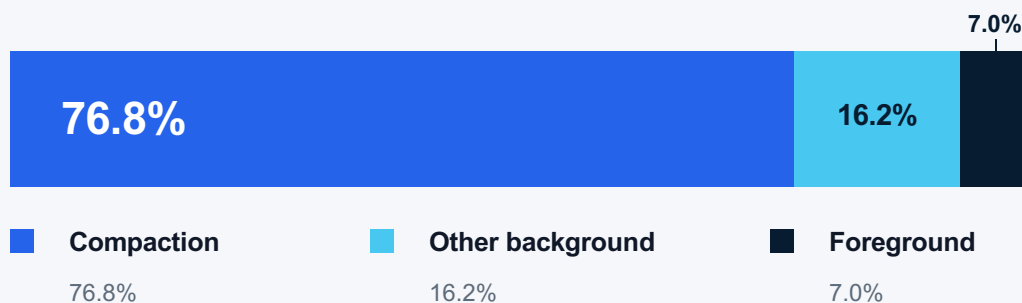
Keeping effective replication low and adopting denser drives
pull on the same background I/O budget.

Doing more with less I/O

SOURCE-ATTRIBUTED DISK BYTES

93% came from background work

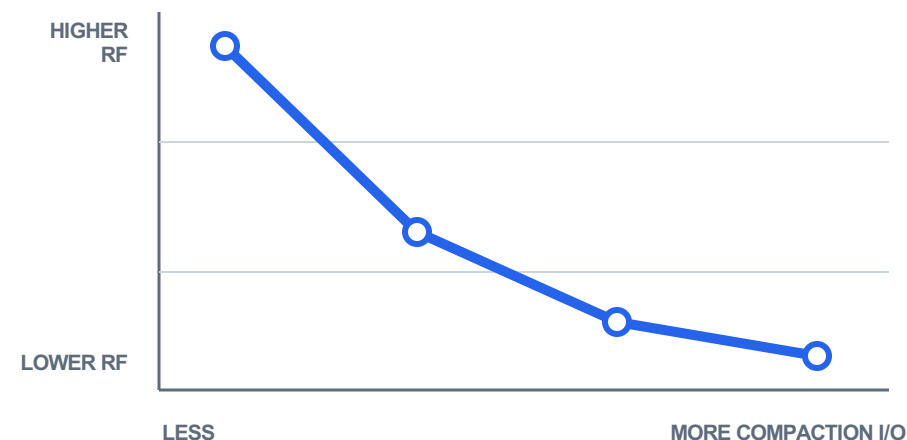
Production block pockets, seven completed days ending 6 Sep 2026 UTC



Compaction moved about 11x as many disk bytes as foreground serving. Need to improve yield per compaction cycle

EFFECTIVE RF VS. COMPACTION I/O

More compaction can reduce effective RF, while consuming disk and network bandwidth.



I/O mix is measured in production; the trade-off curve is conceptual.

Three strategies for different fill levels

L1 Top off an already dense volume

L2 Combine several under-filled volumes

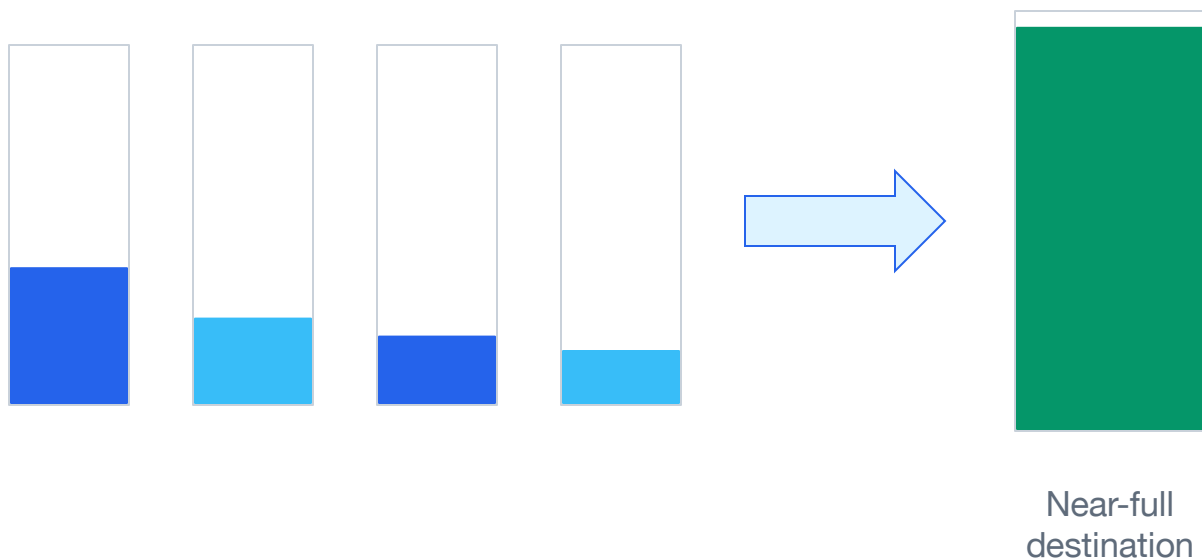
L3 Stream the sparsest tail



Live-data fill of source volumes

L2 reclaims faster by packing sparse volumes together

One bounded packing decision



2–3x

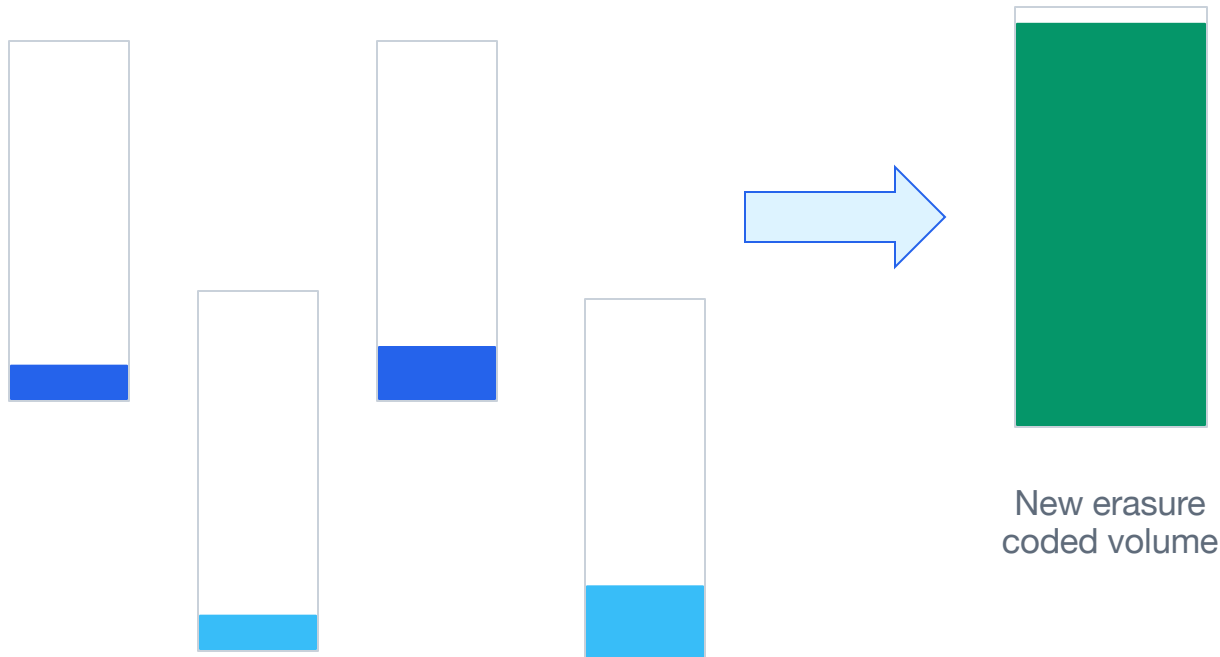
faster overhead reduction than L1

30–50%

lower compaction overhead over
one week in enabled cells

L3 feeds live data into a live erasure coding service

Combine writes from many different sources outside of a cell – live erasure coding, saves write amplification



More metadata overhead than L1/L2, but drain the tail faster than L2

The supply chain

2026

“pretty much sold out”

2.6–3.2x

**Memory contract-price
forecasts**

2028

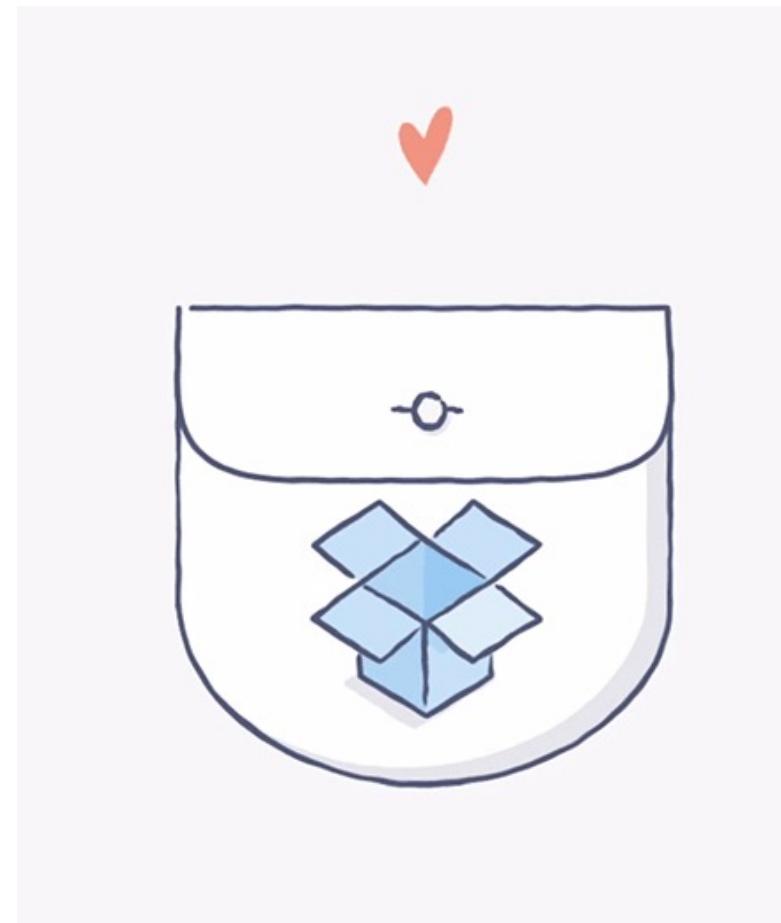
Nearline exabytes allocated

HPTS 2026

Thank you!

Lessons from scaling an on-prem exabyte-scale blob store

Facundo Agriel / Dropbox



Q&A Suggestions

- What are hard drive manufacturers doing about this?
- What are other ways to reduce IO usage?
- Where can AI help optimizations further?

