AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

STG302

Dive deep on Amazon S3

Seth Markle

(he/him)
Senior Principal Engineer
Amazon S3

James Bornholt

(he/him)
Principal Engineer
Amazon S3

Using scale to our advantage … and yours

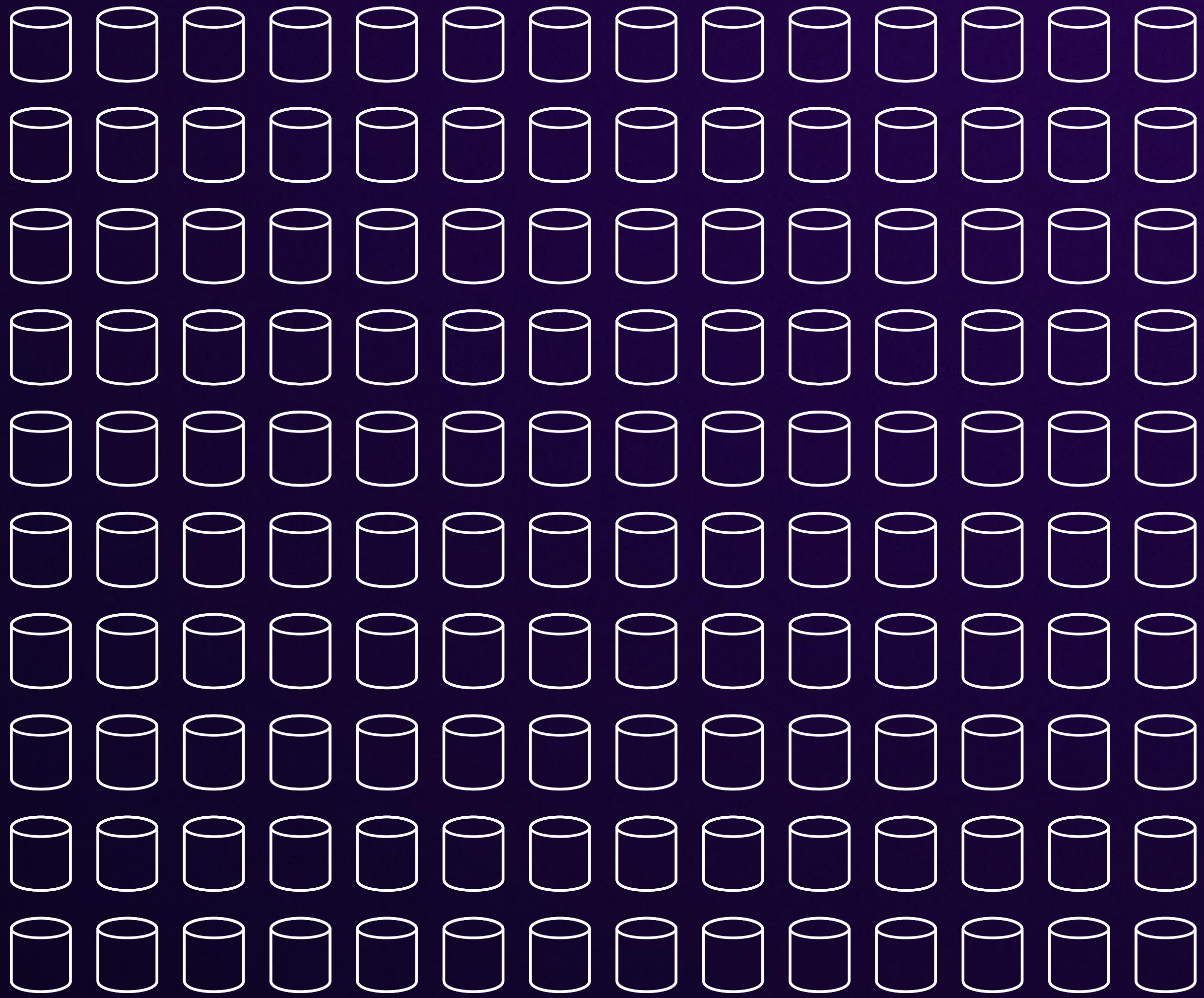
Amazon S3 at scale

Over 400 trillion objects

Over a quadrillion requests per year

Over 200 billion events daily

Over 1 PB/s transferred at peak



01 Physics of data

02 Designing decorrelated systems

03 Engineering for failure is engineering for velocity

01 Physics of data

02 Designing decorrelated systems

03 Engineering for failure is engineering for velocity

01 Physics of data

02 Designing decorrelated systems

03 Engineering for failure is engineering for velocity

aws

The hardware, HDDs

Rotational latency
Seek time

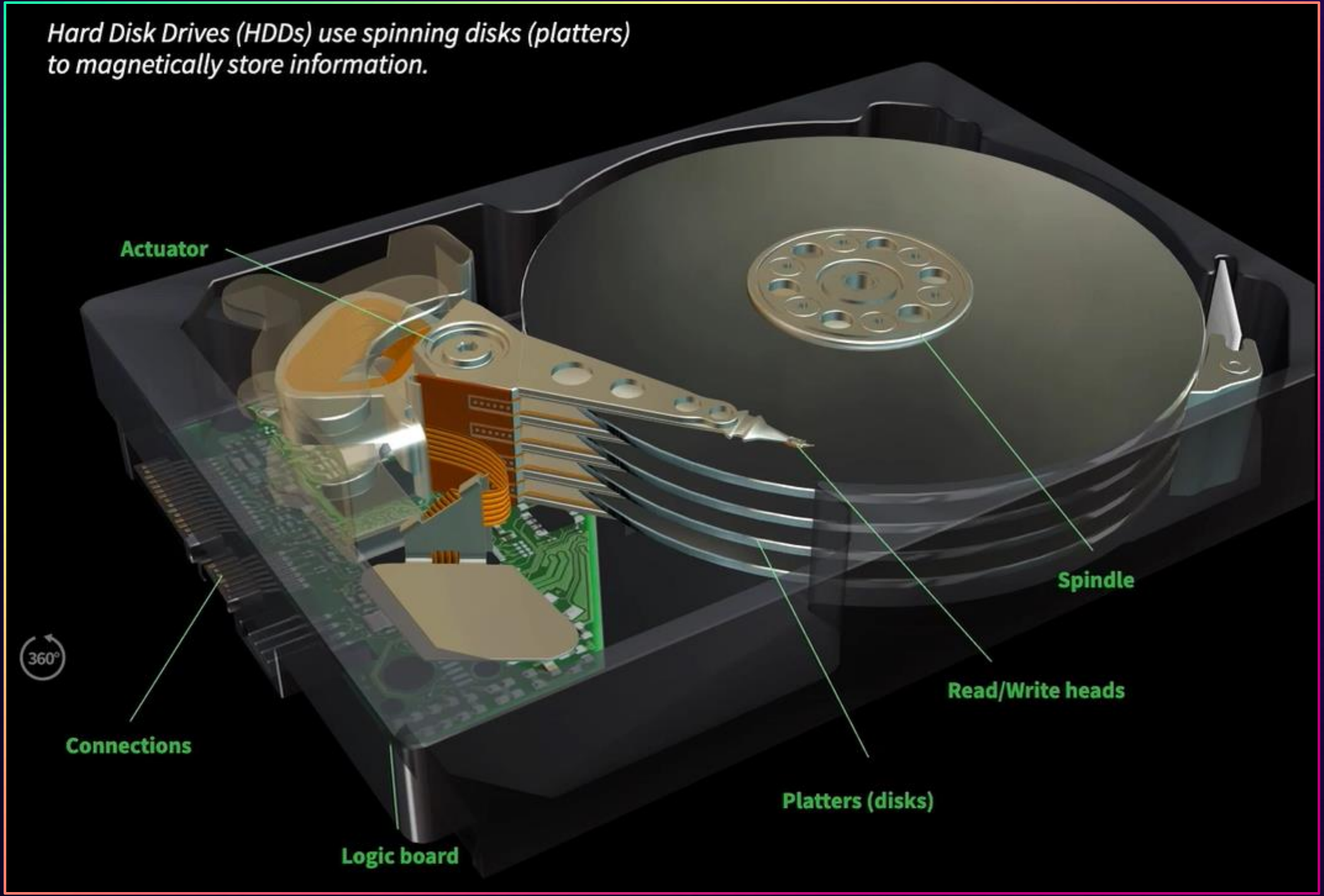
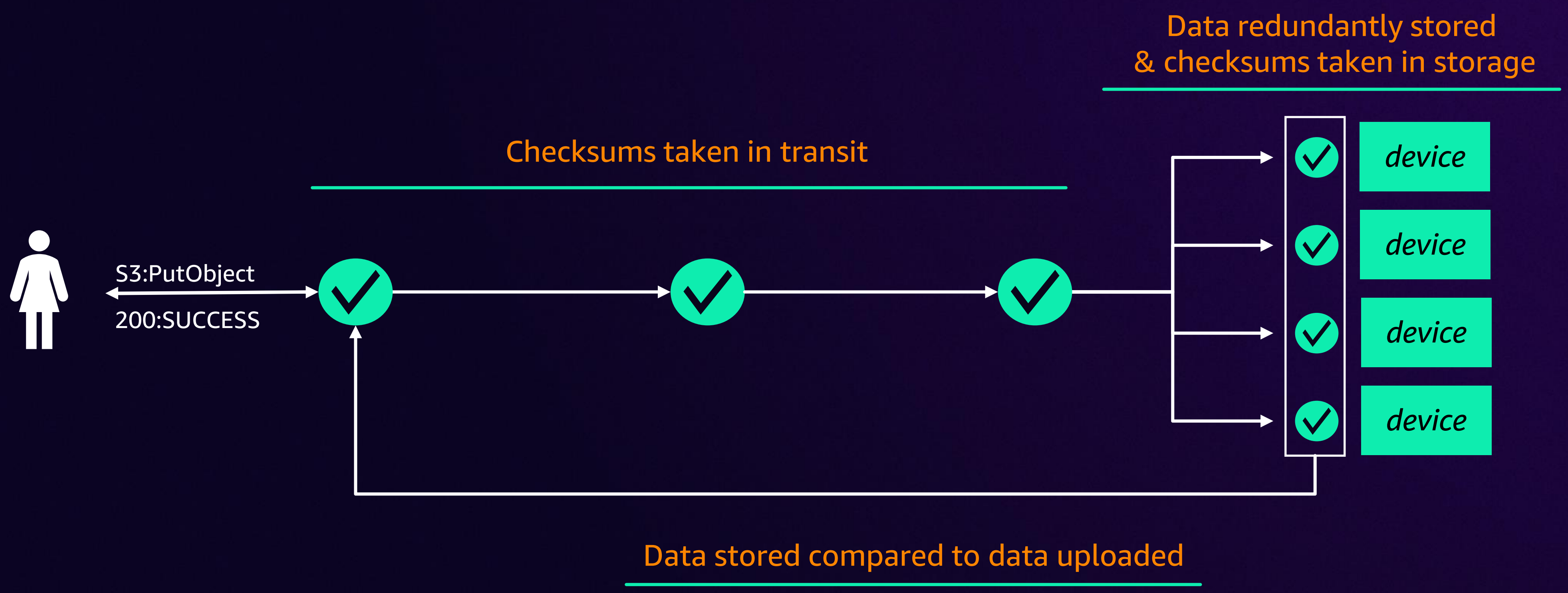
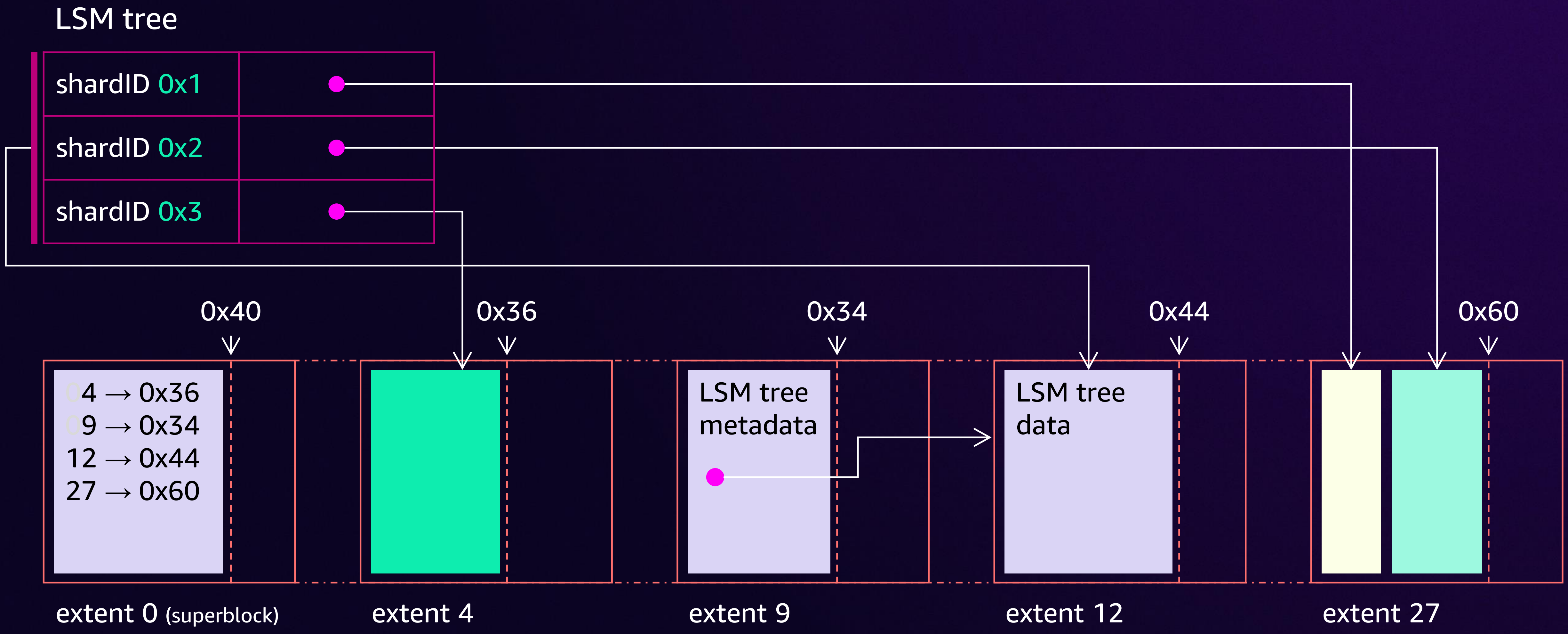


Image by Animagraffs
https://animagraffs.com/hard-disk-drive/

The software, Replication



The software, ShardStore



Individual workloads

- Individual workloads are bursty
- Provisioning for peak

Individual workloads

- Individual workloads are bursty
- Provisioning for peak
- Customer Example: FINRA

The physics of data

- Example:
  - 1 PB of data
  - 1 MB per object
  - Accessed in a single hour

The physics of data

- Rotation = 4 ms, average
- Seek = 4 ms, average
- 0.5 MB Transfer = 2 ms, average
- Total: 10 ms per read (100 reads/second)
- 50 MB per second at 0.5 MB/read

The physics of data

- 1 PB of data
- 275 GB per second

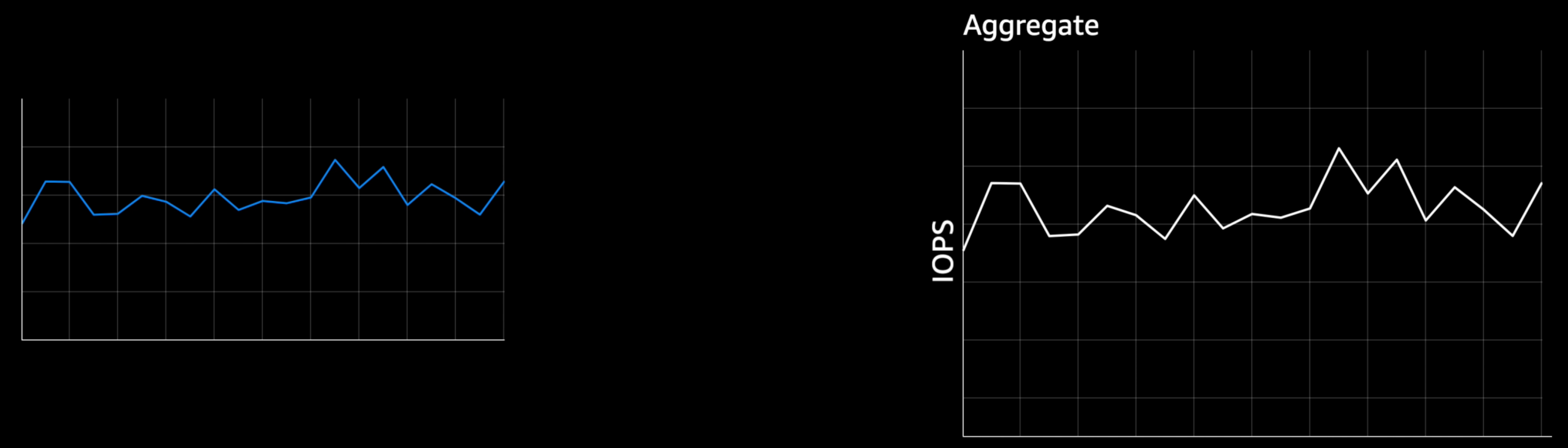
The physics of data

- 1 PB of data
  - 50 drives at 20 TB per drive
- 275 GB per second
  - 5500 drives at 50 MB per second

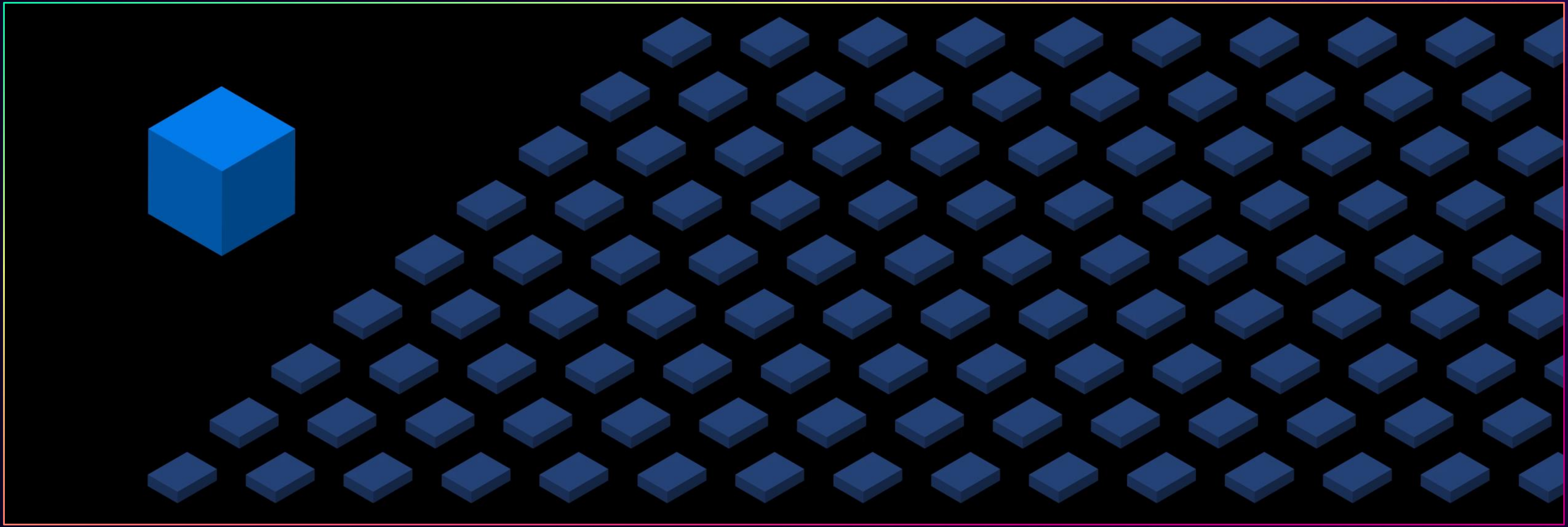
The physics of data

- 1 PB of data
  - 50 drives at 20 TB per drive
- 275 GB per second
  - 5500 drives at 50 MB per second
- 100x difference to support bursts!

Effect of aggregating decorrelated workloads on net system load



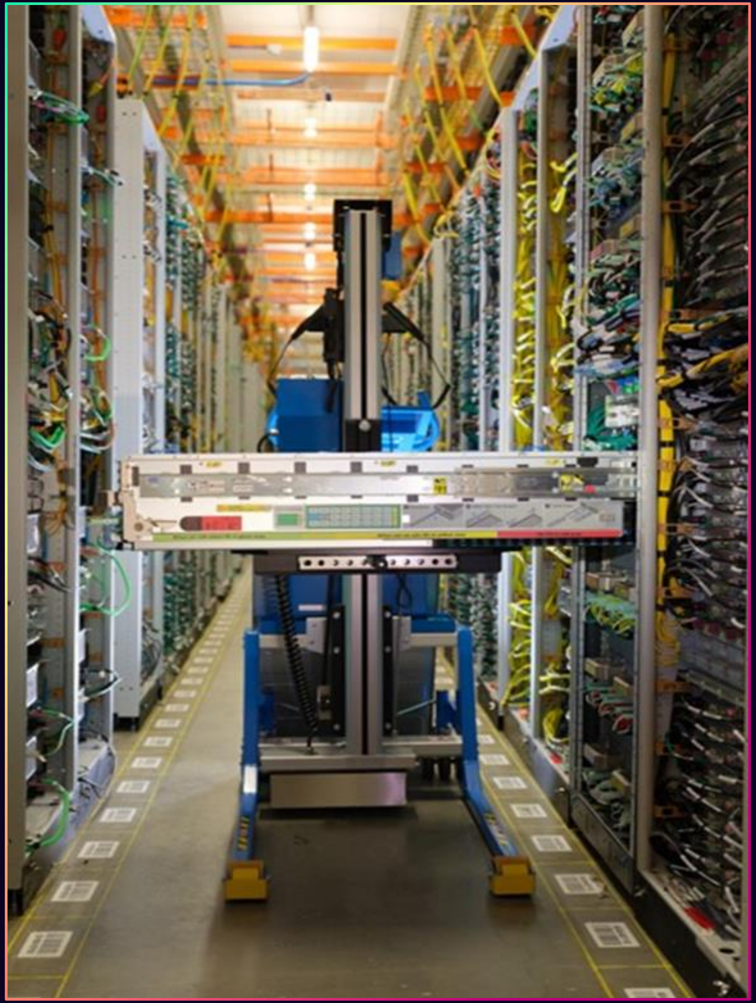
Spread shards across a large number of diverse disks



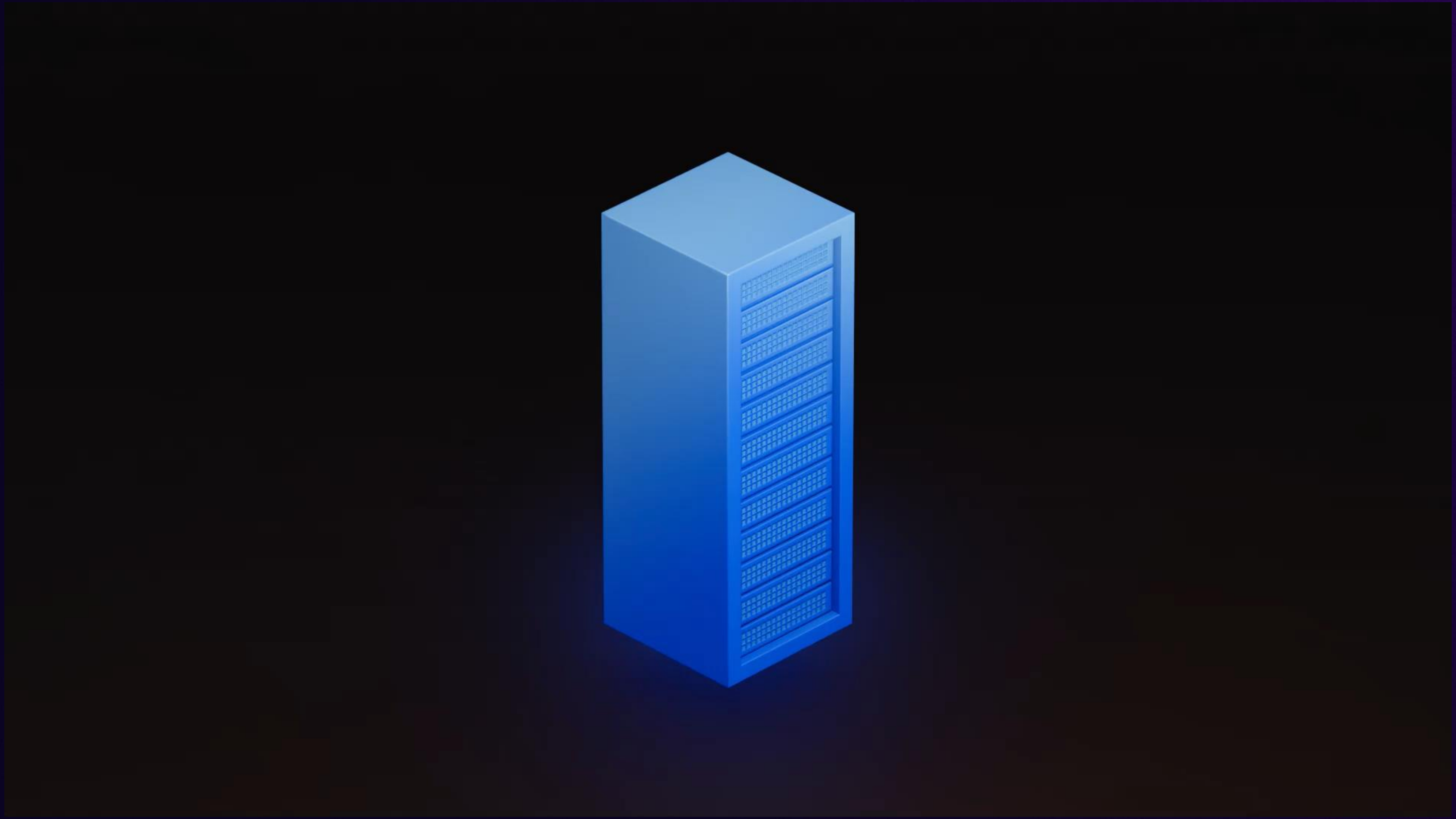
Thermodynamics: Balancing the aggregates

Thermodynamics: Balancing the aggregates

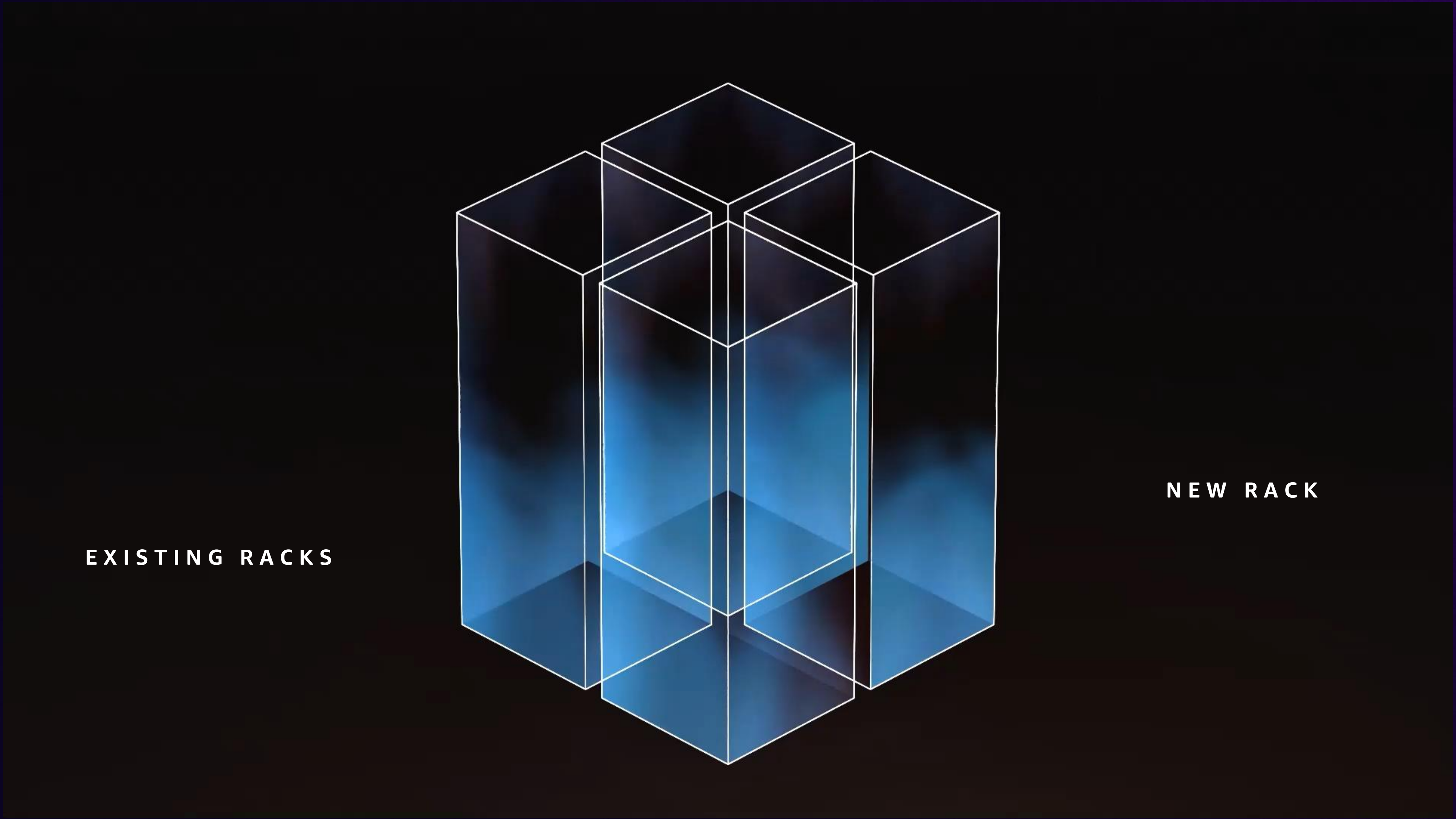
In-practice: Expanding capacity



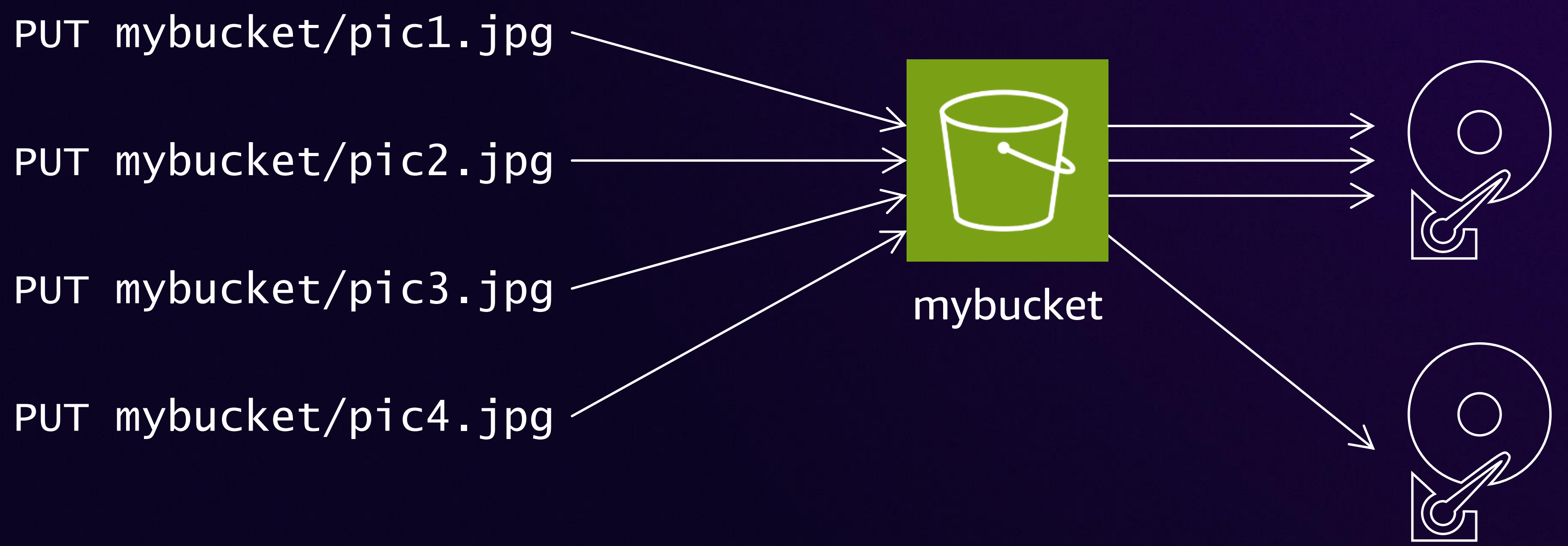
In-practice: Expanding capacity



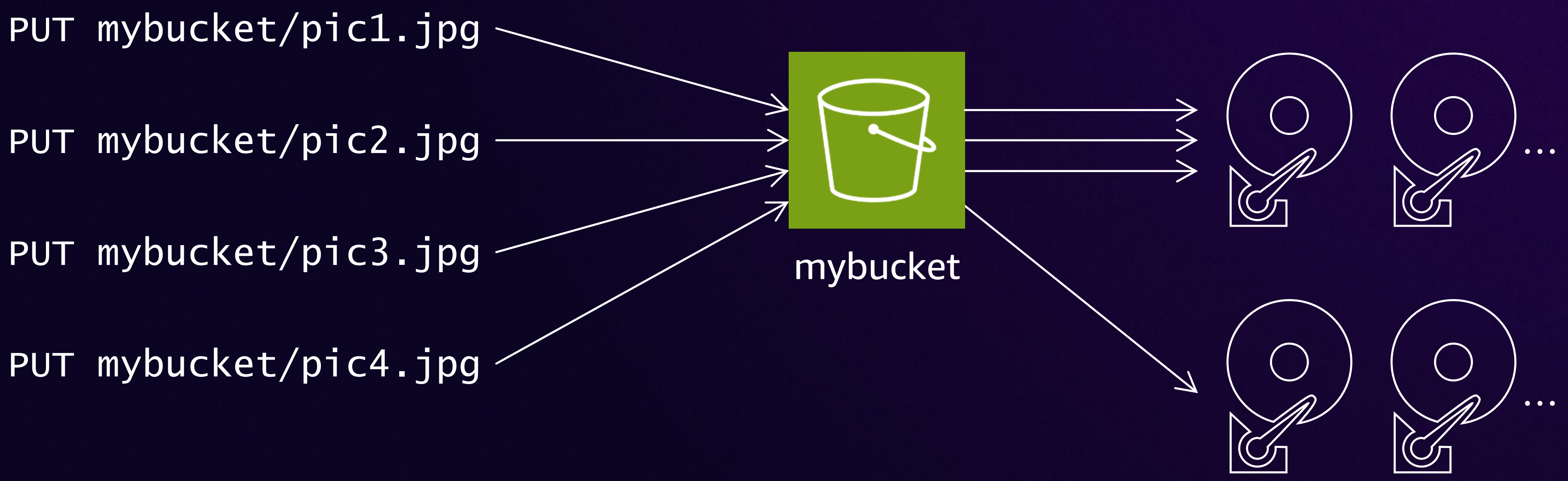
In-practice: Expanding capacity



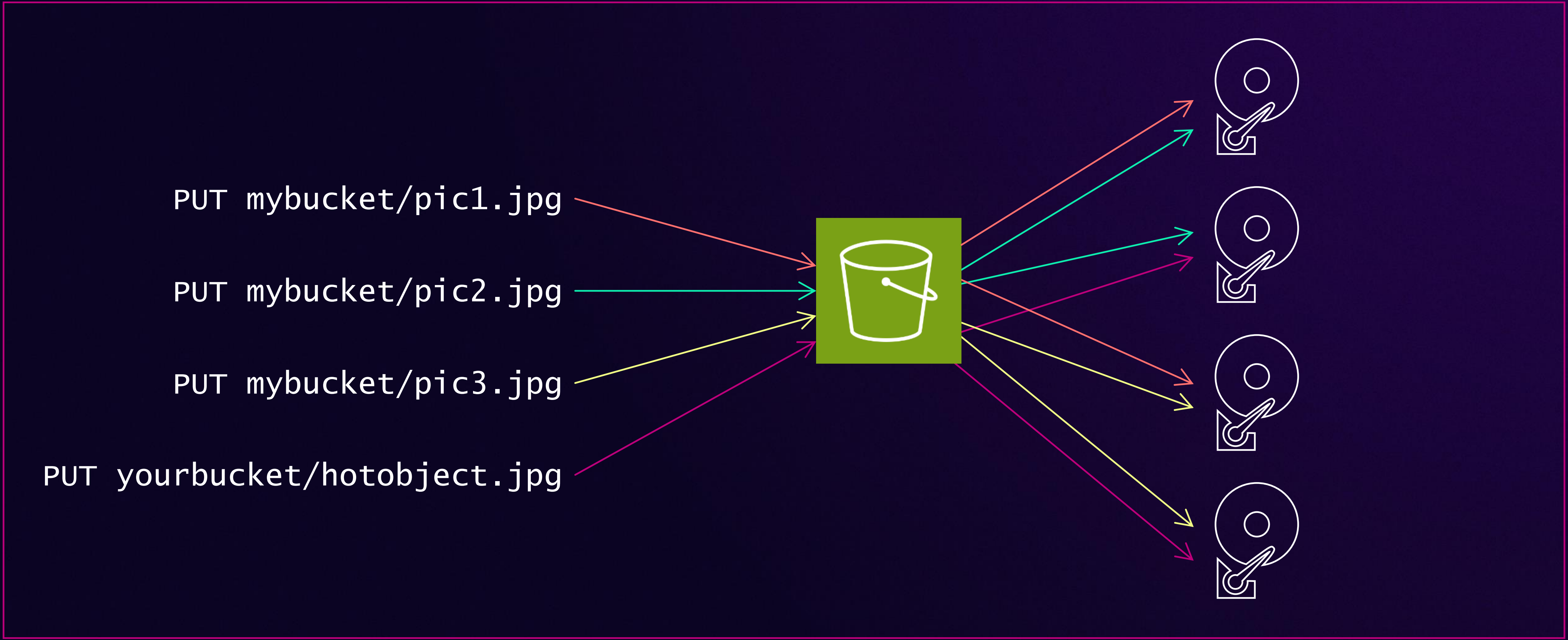
Assigning buckets to their storage



Assigning buckets to their storage

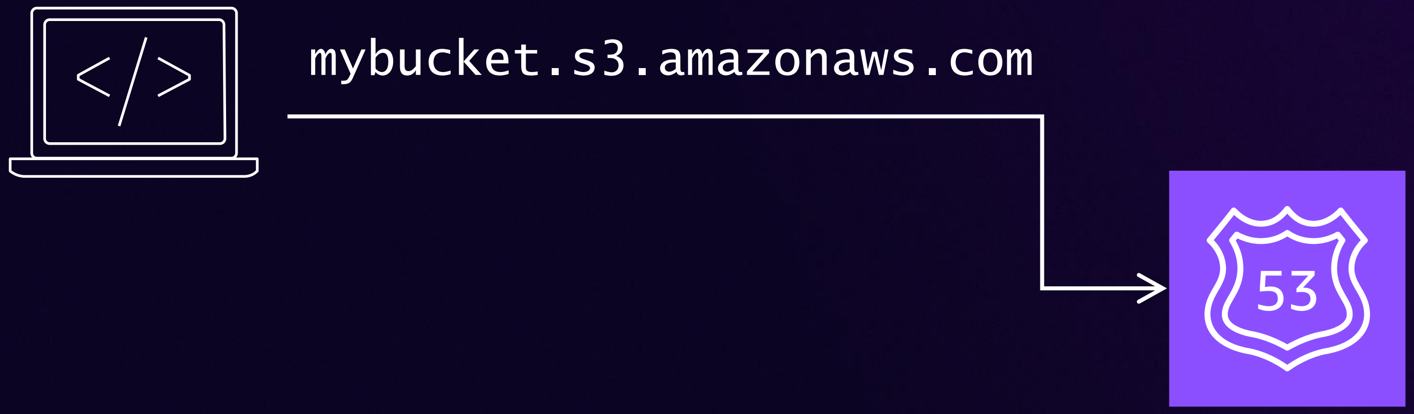


Shuffle sharding



“Elastic” means any S3 customer should be able to use every drive in our fleet on demand – so long as they don't interfere with each other

And it’s not just drives that shuffle shard



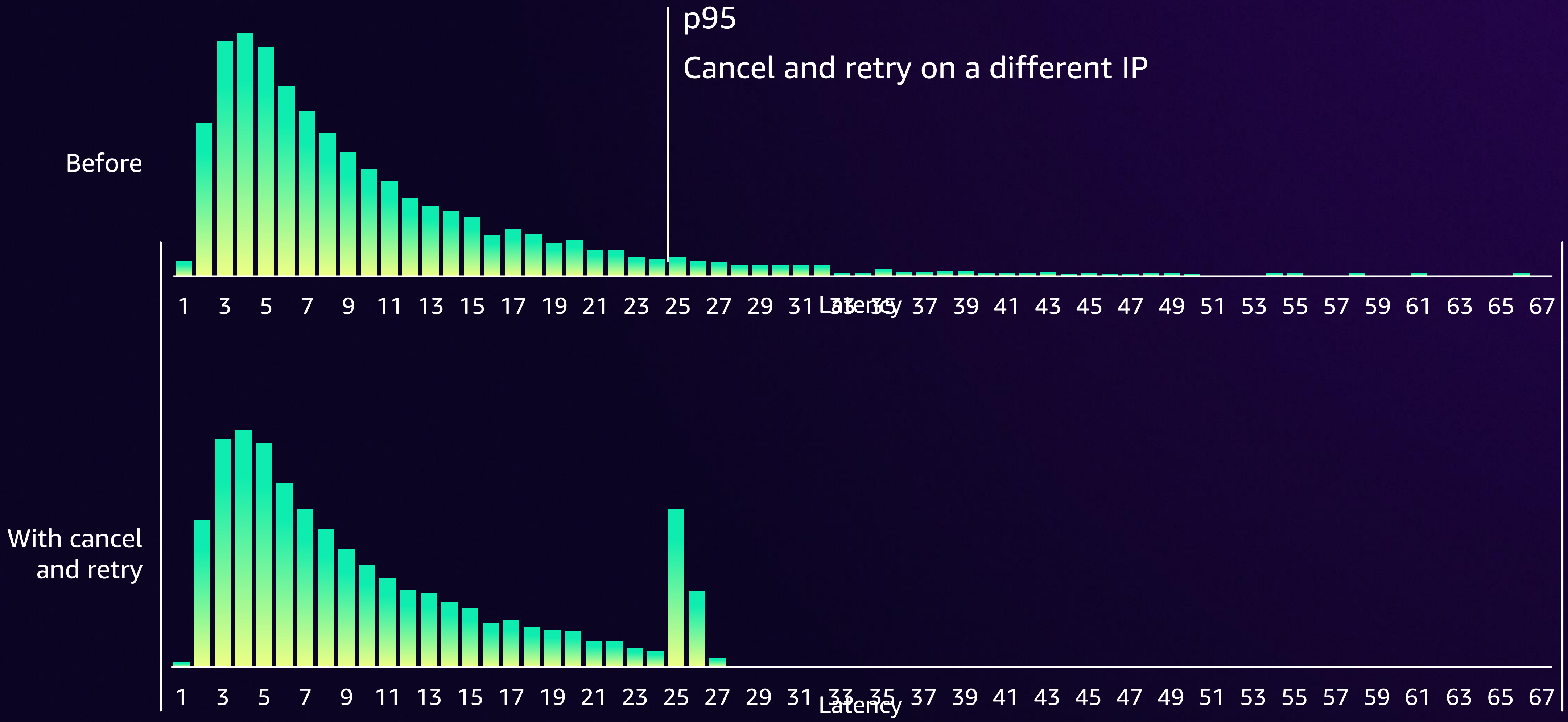

```
Non-authoritative answer:
mybucket.s3.amazonaws.com
canonical name = s3-us-west-2-
w.amazonaws.com.

Name:s3-us-west-2-w.amazonaws.com
Address:52.92.195.9
Name:s3-us-west-2-w.amazonaws.com
Address:52.92.131.57
Name:s3-us-west-2-w.amazonaws.com
Address:52.218.236.243
...
```

And it's not just drives that shuffle shard

Shuffle sharding for fault tolerance

Shuffle sharding in AWS Common Runtime (CRT)

RETRIES CAN ACTUALLY *IMPROVE* PERFORMANCE

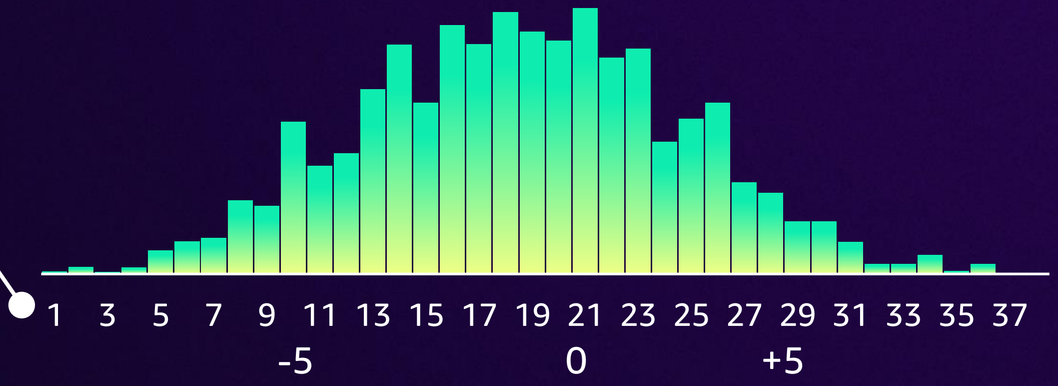


Placing data for shuffle sharding

PUT mybucket/pic1.jpg

Placing data for shuffle sharding

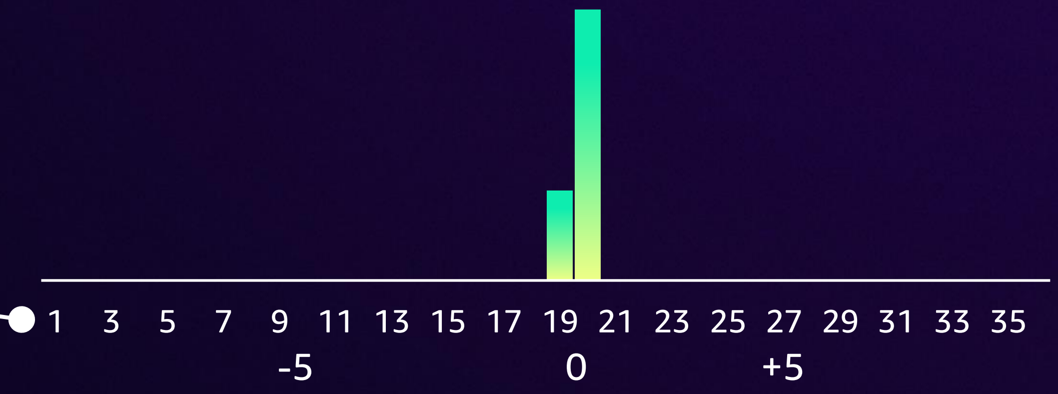
Choose a random drive



PUT mybucket/pic1.jpg

mybucket

"The power of two random choices"
Look at just two random drives and choose the better one



Engineering for decorrelation

- **Shuffle sharding** decorrelates customers and workloads
- It also enables **scale** and **fault tolerance**
- Use **two random choices** to balance shards

Workload isolation using shuffle sharding
Amazon Builder's Library

01 Physics of data

02 Designing decorrelated systems

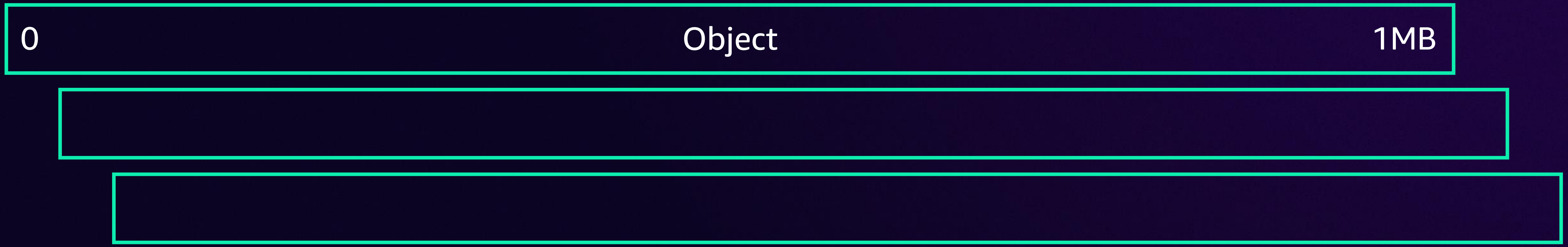
03 Engineering for failure is engineering for velocity

At scale, failures are a fact of life

At scale, failures are a fact of life

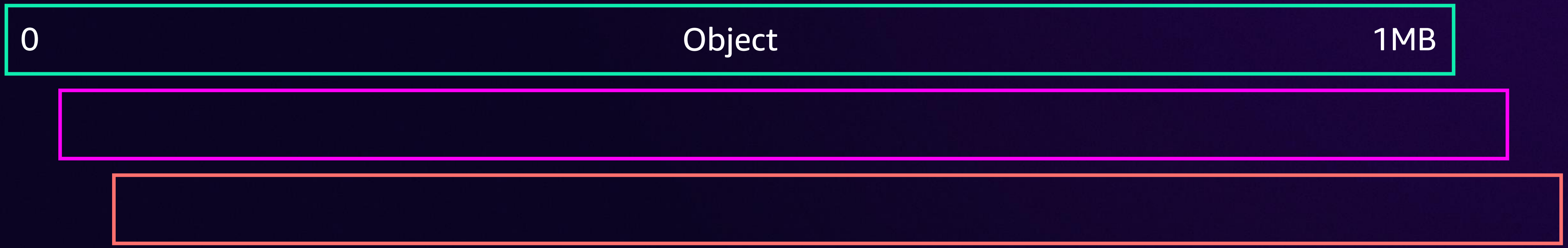
Erasure coding for fault tolerance

Replication is a simple way to tolerate faults – both individual drives and entire Availability Zones – but comes with high overhead



Erasure coding for fault tolerance

Replication is a simple way to tolerate faults – both individual drives and entire Availability Zones – but comes with high overhead



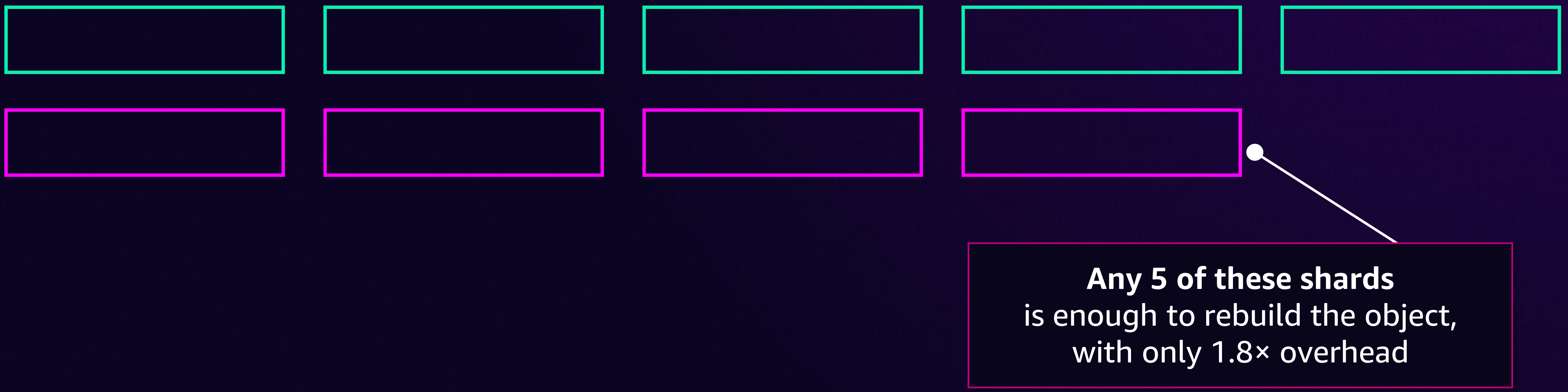
Erasure coding for fault tolerance

Erasure coding uses math to split the object into **shards** and create extra **parity** shards; the object can be rebuilt from **any K of the shards**

0	Object	1MB

Erasure coding for fault tolerance

Erasure coding uses math to split the object into **shards** and create extra **parity** shards; the object can be rebuilt from **any K of the shards**

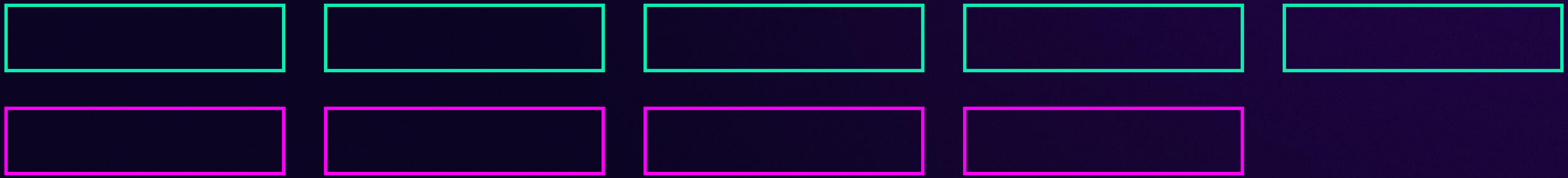


Erasure coding for fault tolerance

Erasure coding uses math to split the object into **shards** and create extra **parity** shards; the object can be rebuilt from **any K of the shards**

Fault tolerance also gives velocity

Erasure coding allows us to deploy new software or hardware safely by **exposing only a few shards** to the changes



Fault tolerance also gives velocity

Erasure coding allows us to deploy new software or hardware safely by **exposing only a few shards** to the changes

Fault tolerance and shuffle sharding

The **birthday paradox** tells us that shuffle sharding will expose **some** object to too much risk; we need to layer in constraints

Fault tolerance and shuffle sharding

The b ne
objec

Using Lightweight Formal Methods to Validate a Key-Value Storage Node in Amazon S3

James Bornholt
Amazon Web Services
& The University of Texas at Austin

Rajeev Joshi
Amazon Web Services

Vytautas Astrauskas
ETH Zurich

Brendan Cully
Amazon Web Services

Bernhard Kragl
Amazon Web Services

Seth Markle
Amazon Web Services

Kyle Sauri
Amazon Web Services

Drew Schleit
Amazon Web Services

Grant Slatton
Amazon Web Services

Serdar Tasiran
Amazon Web Services

Jacob Van Geffen
University of Washington

Andrew Warfield
Amazon Web Services

Abstract

This paper reports our experience applying lightweight formal methods to validate the correctness of ShardStore, a new key-value storage node implementation for the Amazon S3 cloud object storage service. By "lightweight formal methods" we mean a pragmatic approach to verifying the correctness

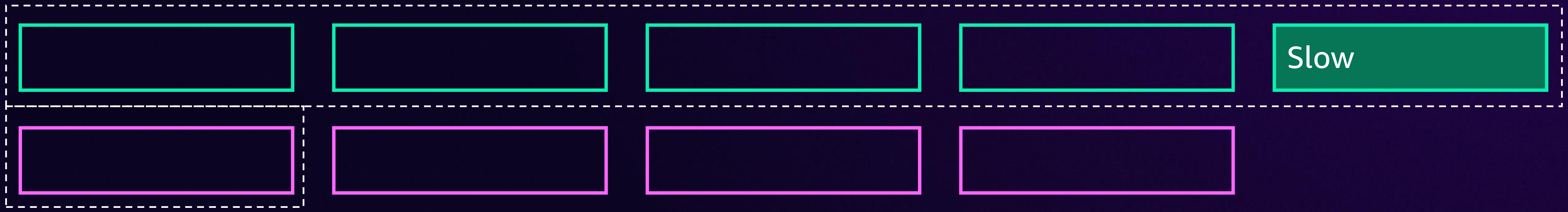
Using Lightweight Formal Methods to Validate a Key-Value Storage Node in Amazon S3. In *ACM SIGOPS 28th Symposium on Operating Systems Principles (SOSP '21), October 26–28, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3477132.3483540

1 Introduction

its
s

Improved performance through fault tolerance

Additional shards and shuffle sharding allow us to hedge against tail latency by **overreading**



Fault tolerance and velocity

- Failures are common at scale, and **fault tolerance is more than design**
- Fault tolerance, used wisely, also **improves developer velocity**

ShardStore

Physics of data

Engineering for decorrelation

Fault tolerance and velocity

Thank you!

Please complete the session survey in the mobile app

Seth Markle
smarkle@amazon.com

James Bornholt
bornholt@amazon.com