From ACID to CAP and back again: Making S3 reliable

Fred Simon - JFrog
Fred Simon, JFrog co-founder and Chief Architect
What Frog?
What Frog?
What Frog?
What Frog?
What Frog?
What Frog?
What Frog?
WHAT IF I TOLD YOU

GIT DIDN'T INVENT
CHECKSUM STORAGE
Because the filesystem is too stupid

METADATA-RICH STORAGE
THIS IS OUR FILESYSTEM

WHEN IT COMES TO METADATA AND CONCURRENCY
That’s what filesystem gives us:

- Path
- Name
- type
- Owner
- Modified
- permissions
The Age of Binaries

Explosion of Binaries

- Docker
- Microservices
- DevOps
- C D
- C I
- Agile
QUANTITY HAS A QUALITY OF ITS OWN
Artifactory Service

60TB
Repository
52 Million Artifacts
Up to 9GB+ Binary Size
200+ Repositories

26.5M
One Day In SLC
20M (47.8TB) Downloads
150K (334GB) Publishes
2900+ SQL/Sec
13000+ IOPS

128GB
Node
<1% CPU for G1 GC
CPU Peak ~ 10%

5
Global Datacenters
43K Artifacts Replicated/Day
Metadata in the Age of Binaries

- Finding the needle in haystack of binaries
- What is deployed where?
- Tracing binaries to sources
- Pick the right one to deploy
What we need:
What we need:

- Build name
What we need:

- Build name
- Build number
What we need:

- Build name
- Build number
- Vcs commit hash
What we need:

- Build name
- Build number
- Vcs commit hash
- License information
What we need:

- Build name
- Build number
- Vcs commit hash
- License information
- Quality information
What we need:

- Build name and number
- Vcs commit hash
- License information
- Quality information
- Platform, arch information
- ...
Metadata-rich Storage
WAIT A SECOND...

THAT WASN'T AN ORDINARY FILESYSTEM!
Metadata-rich Storage

Artifact on disk

Record in DB
WAIT A SECOND!

HOW DO YOU RENAME?!
Append Only, GC-ed
Append-Only -> No Locks -> Speed
LET'S TALK ABOUT ACID
What do you even mean, transaction?!
What do you even mean, transaction?! 

- File starts to stream to file renamed to checksum and record added?
What do you even mean, transaction?! 

- File starts to stream to file renamed to checksum and record added? 
- Part of this as enclosed transaction?
What do you even mean, transaction?! 

- File starts to stream to file renamed to checksum and record added?
- Part of this as enclosed transaction?
- Only database insertion as transaction?
Full DB -> FS -> NFS -> S3 -> HDFS

SCALING THE STORAGE
Full DB

Metadata

Blobs
Disk Cache?

WAIT FOR IT
Local Filesystem

Metadata

Blobs
Network File Storage
Let’s talk about cache (part 1)
Let’s talk about cache (part 1)

- Directory on a local ssd drive
Let’s talk about cache (part 1)

- Directory on a local ssd drive
- cache for all the non-local storages
Let’s talk about cache (part 1)

- Directory on a local ssd drive
- cache for all the non-local storages
- Improves performance dramatically
Let’s talk about cache (part 1)

- Directory on a local ssd drive
- cache for all the non-local storages
- Improves performance dramatically
- Lru
Let’s talk about cache (part 1)

- Directory on a local ssd drive
- cache for all the non-local storages
- Improves performance dramatically
- Lru
- Customizable size
Elasticity Matters
Once upon a time...
DESTROY THEIR BASE?

WE'LL JUST USE SOME STRONG ACID
Remember this?

**LET'S TALK ABOUT CACHE (PART 1)**

- Directory on a local SSD drive
- Cache for all the non-local storages
- Improves performance dramatically
- LRU
- Customizable size
Artifactory HA
ONE DOES NOT SIMPLY SYNCHRONIZE CLUSTER ON S3
Let’s talk about cache (part @)

- S3 uploads are slooooodow
- So, not only we need download cache to fight eventually consistent storage
- We need upload cache to release the clients ASAP
- Write-behind
Now let's combine them!

Looks Familiar? Servlet Filters!
A.k.a. “Decorator” Design Pattern

Head First Design Patterns by Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson
public interface BinaryProvider {
    boolean exists(String sha1);
    @NonNull
    InputStream getStream(String sha1) throws BinaryNotFoundException;
    BinaryInfo addStream(BinaryStream binaryStream) throws IOException;
    boolean delete(String sha1);
    @NonNull
    StorageInfo getStorageInfo();
}
We call it “Binary Providers Chain”
WAIT A SECOND...

S3 IS NOT EVENTUALLY CONSISTENT ANYMORE!
WHAT’S THE TRADE-OFF FOR EVENTUAL CONSISTENCY?
What's the trade-off for eventual consistency in AWS S3?

- A: Multi-zone redundancy is gone
- B: The uploads are much slower
- C: The uploads are failing more
- D: No trade-off, just magic
What's the trade-off for eventual consistency in AWS S3?

- A: Multi-zone redundancy is gone
- B: The uploads are much slower
- C: The uploads are failing more
- D: No trade-off, just magic
The retry provider
The retry provider

- Another binary provider
The retry provider

- Another binary provider
- Does what the name implies
The retry provider

- Another binary provider
- Does what the name implies
- Might be useful stand-alone
The retry provider

- Another binary provider
- Does what the name implies
- Might be useful stand-alone
- For example for hdfs
Remember?
Y U NO USE S3
IN ARTIFACTORY SAAS?
Why not s3?
Why not s3?

- Remember why elasticity matters?
Why not s3?

- Remember why elasticity matters?
- But we don’t do cleanups in client’s Artifactory servers!
Why not s3?

- Remember why elasticity matters?
- But we don’t do cleanups in client’s Artifactory servers!
- Zfs on EBS gives much better performance
Why not s3?

- Remember why elasticity matters?
- But we don’t do cleanups in client’s Artifactory servers!
- Zfs on EBS gives much better performance
- No need to fight eventual or retry
BUT EBS IS NOT REPLICATE
ACROSS AVAILABILITY ZONES?!
Another Binary Provider FTW!
One More Thing!