ceph DreamHost

Ceph中国社区 IT大咖说 知识共享平台

CEPH

CEPHALOCON APAC 2018

THE FUTURE OF STORAGE

22-23 March 2018 | BEIJING

RGW S3: Features vs deep compatibility

What lurks beneath the API

ceph DreamHost

Ceph中国社区 | IT大咖说 知识共享平台

CEPH

CEPHALOCON APAC 2018

THE FUTURE OF STORAGE

22-23 March 2018 | BEIJING

RGW S3: Features vs deep compatibility

What lurks beneath the API

Background

- Wrote (most) of RGW static website hosting on contract for Dreamhost
  - Credit to Yehuda Saleda for early work
- Went to work full-time for Dreamhost in 2015
  - Ceph development (RGW) & operations
- Open Source
  - Gentoo Linux core developer (since 2003)
  - MogileFS (2007-2013): LiveJournal’s open-source distributed content store
  - phpMyAdmin (2001-2003)

ceph

Quick terminology

- S3: the protocol itself
- Specification: Public AWS S3 API document
- AWS-S3: shortened to AWS
- RGW-S3: shortened to RGW
- S3 API calls may include specific features in their requests
- S3 API calls may have only immediate or persistent impact

Specification

- Amazon publishes a single API specification as:
  - Amazon Simple Storage Service, API Reference, API Version 2006-03-01
- The version number has never been bumped
- Document history is a high-level summary only
- No public itemized list of changes known

S3 Feature dimensions

- Storage: configured per-object, persistent
  - ACL, Expiration, SSE, Storage Classes, Tagging, Versioning
- Access: specific to the upload/download process
  - Accelerate, Browser POST, CORS, Policy, requestPayment, STS, torrent, website
- Services: interact with objects some time later
  - Analytics, Inventory, Lifecycle, Logging, Metrics, Notification, Replication

Features: AWS vs RGW

- The “Features Support” of the main RGW document is high-level only
- The “RADOS Gateway S3 API Compliance” page is very out of date
- Protocol testing in the s3-tests repo “best” indicator of coverage

Features: AWS vs RGW (Jewel)

- Storage: configured per-object, persistent
  - ACL, ~~Expiration, SSE, Storage Classes**, Tagging,~~ Versioning
- Access: specific to the upload/download process
  - ~~Accelerate~~, Browser POST, CORS, ~~Policy, requestPayment, STS, torrent,~~ website
- Services: interact with objects some time later
  - ~~Analytics, Inventory, Lifecycle, Logging, Metrics, Notification, Replication~~

Features: AWS vs RGW (Luminous)

- Storage: configured per-object, persistent
  - ACL, Expiration, SSE, ~~Storage Classes**, Tagging,~~ Versioning
- Access: specific to the upload/download process
  - ~~Accelerate~~, Browser POST, CORS, ~~Policy,~~ requestPayment, ~~STS~~, torrent, website
- Services: interact with objects some time later
  - ~~Analytics, Inventory,~~ Lifecycle, ~~Logging, Metrics, Notification, Replication~~

Features: AWS vs RGW (Mimic)

- Storage: configured per-object, persistent
  - ACL, Expiration, SSE, ~~Storage Classes**,~~ Tagging, Versioning
- Access: specific to the upload/download process
  - ~~Accelerate~~, Browser POST, CORS, Policy, requestPayment, ~~STS~~, torrent, website
- Services: interact with objects some time later
  - ~~Analytics, Inventory,~~ Lifecycle, ~~Logging, Metrics, Notification, Replication~~

s3-tests

- Good for basic feature testing
- Slow! Takes 25+ minutes for a single complete run
- Testing in corner cases lags even further
- No explicit coverage for data written under OLD Ceph/RGW versions for upgrades

S3 API Usage

- Prioritizing S3 features by customer request & usage
  - Requests for SSE
- Need a way to measure existing feature usage
  - Spoiler: cool stuff doesn’t get used

S3 API Usage (what)

- Need request & headers to parse non-POST
- Need entire body as well for some POST requests
- RGW is already parsing it (but spread out all over codebase)

S3 API Usage (where)

- Not in RGW itself at present :-(
- Choices!
  - Interception in HTTP reverse-proxy/load-balancer
  - Parse from logs: ops, or raw rgw/civetweb
- Control fields in Browser POST payload hard to capture that early

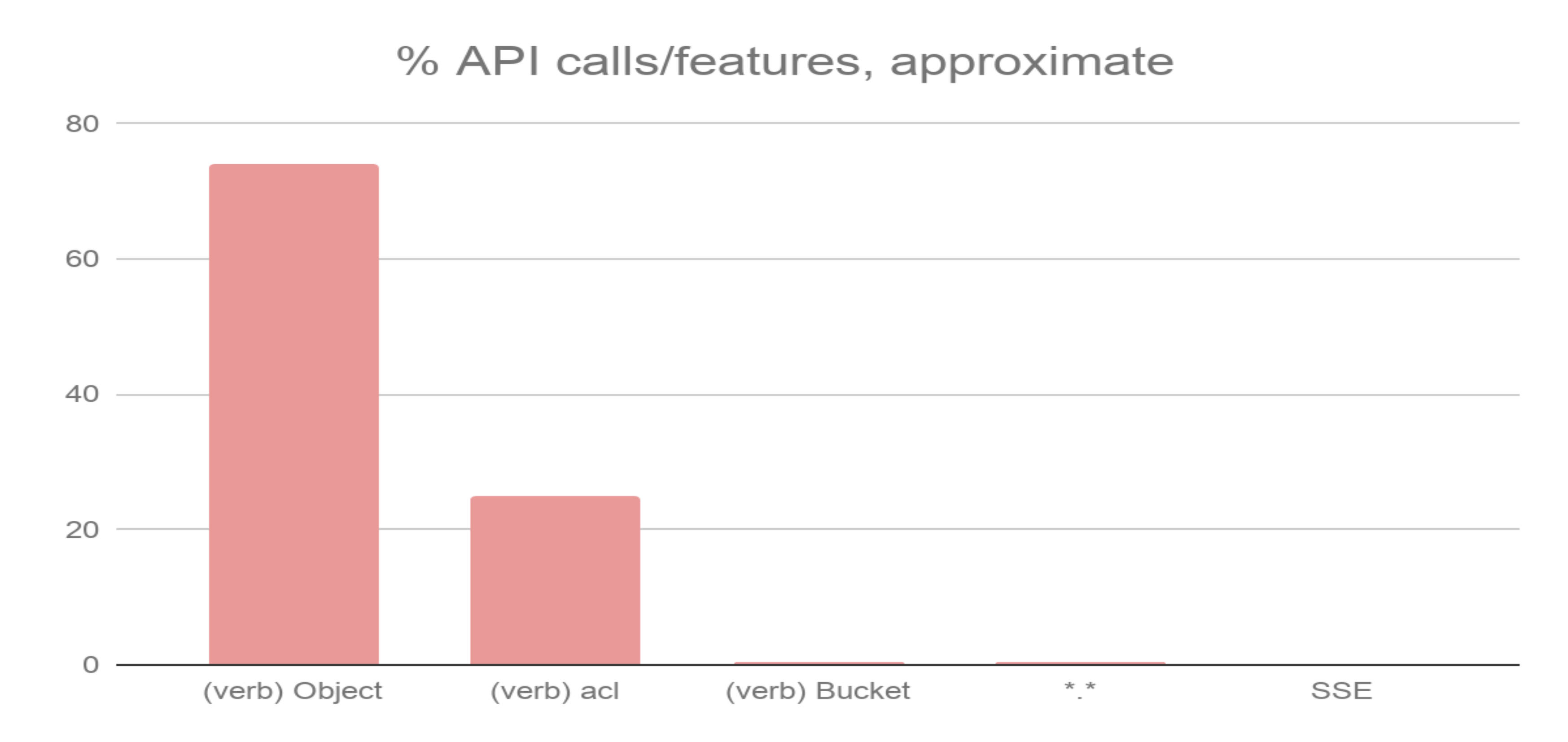
S3 API Usage (how)

- Custom HAProxy 1.7 Lua plugin
  - Initially written to fairly rate-limit AccessKey
- Parses request line & headers BEFORE RGW
- Does not have access to request body
- Improvements:
  - “Standardized” operations names in the logs (all of them)?
  - How to track feature usage in API calls? SSE? Metadata? Tagging?
  - Has to parse the RGW response as well for logging

S3 API Usage (numbers)

- Caveat: these are statistics based on Dreamhost’s public cloud offering, which targets low-skill users & existing clients
- Clients may consume S3 as a product (and use features by design)
- SSE: CloudBerry Backup, Duplicati, QNAP

S3 API Usage (numbers)



Specification vs AWS vs RGW

- Subtle differences in behavior
- AWS is more lenient than the Specification
- AWS behavior differs slightly between regions
- RGW is based mostly on the Specification
  - Plus observed AWS behavior
  - Plus special RGW-only logic

Spec/AWS/RGW: CreateBucket

- CreateBucket, of an already existing bucket, owned by you
- `us-east-1: 200 OK`
- Other AWS regions: `409 BucketAlreadyOwnedByYou`
- RGW: `200 OK`
    - Some clients mishandle BOTH potential responses
- This detail is in the specification, but you need to read carefully

Spec/AWS/RGW: Content-Length (1)

- Should every HTTP PUT request include a Content-Length header?

Spec/AWS/RGW: Content-Length (2)

- Should every HTTP PUT request include a Content-Length header?
- Specification: yes**
- RGW: Jewel & earlier: mostly
- RGW: Luminous: yes
- S3: Only if length non-zero!

Spec/AWS/RGW: Content-Length (3)

- Object `PUT ?acl` operation has a case where there is no body, because everything is in the HTTP headers.
- RGW started to require more Content-Length because it made code easier
- Old Amazon-official S3 clients did NOT include Content-Length header unless there was a body
- Patched in load-balancer, not yet RGW

Spec/AWS/RGW: Regions vs Signatures

- How many user reports have you seen of new S3 clients that don’t work quite right?
- Some clients have hard-coded logic that depends on the exact name of the region
  - us-east-1 gets special treatment again
- AWS4 signature includes the region
- AWS signature calculation bugs
  - Multipart & POST
  - Adjcent spaces stripping

RGW strictness

- How strict should RGW S3 really be?
- Should RGW follow the Robustness Principle (Postel's Law)?
- The Content-Length change broke clients
  - Possibly for the better
  - But was unexpected behavior in upgrade
- Need tests to replicate old client behavior
  - Without the HTTP library interfering!
- Some HTTP/1.0 behaviors still exist in AWS
  - Depends on region
  - Path-encoded hostname without Host header

Impact of a missing feature

- Will the lack of a feature cause problems for later RGW versions?
- Protocol design:
    - No immediate feedback mechanisms to confirm some features were used!
    - Can re-query most to verify
    - Eventual consistency may interfere

Brief SSE case study (1)

- Jewel & earlier
    - What happens if you set SSE headers?

Brief SSE case study (2)

- Jewel & earlier
  - Data stored unencrypted
  - Client may have associated key stored externally

Brief SSE case study (3)

- Jewel & earlier
  - Data stored unencrypted
  - Client may have associated key stored externally
- Luminous
  - New SSE uploads will be encrypted correctly
  - Fetches of old data break if SSE headers set!

TODO Client choices...

- TODO
- Will those differences negatively impact S3 client implementations, and are they intentional?
- What happens when customers use unexpected clients & features?
- Old & undermaintained clients might not get new feature support
- BUT
- Bugs do arise in new Ceph releases as well as new client releases
- Multipart uploads have lots of nuanced corners
- CyberDuck 6.2 (TODO: verify number) broke client

Internal compatibility (1)

- What’s the oldest RGW data you have in production?
- Have you verified you can read it back?
- End-to-end?

Internal compatibility (2)

- Intact, complete?
- Head/tail bugs in multipart
- Truncation at boundaries
- Checking the correct pool!
- #23232: RGWCopyObj silently corrupts the object that was multipart-uploaded in SSE-C
- Previous silent write discards have also happened