Storage

COSC349—Cloud Computing Architecture

David Eyers

Learning objectives

- Can contrast **object storage** with **filesystem storage**
- Indicate why object storage scales-out well
- Define Amazon S3 **buckets**, **objects** and **keys**
- Contrast Internet speeds against couriered hard-disks
- Explain how S3's use of REST can allow it to serve **resources for websites** effectively

Cloud storage is a multifaceted topic

- Software engineering side: **architectures for storage**
  - Transitions from previous ways of managing storage
    - e.g., files and folders on OS filesystems
  - New solutions that are specifically cloud relevant
    - e.g., **object storage** such as S3
    - usually depends on scalability of cloud

- Also issues of **data transfer rates** and **costs**
  - Transfer of data through the Internet is actually quite slow…

Internet bandwidth often beaten by couriers

- Data transfer involves **data volume** and **time delay**
  - Internet pipes are quite responsive, but transfer slowly
    - 100 megabit/s dedicated Internet; 50 terabytes of data …
    - 50 TB × 1000 × 1000 × 8 ÷ 100 = 4,000,000s … approximately **50 days**!
  - Courier over an 50 TB hard disk? Likely to take **a few days**…
- Amazon Snowball makes **hard-disk shipping** a service
  - Hardened storage appliances are shipped from Amazon
  - Client transfers data on/off
  - Prepaid courier service returns the device to Amazon

Amazon S3 (2006)—terminology

- Amazon Simple Storage Service (S3) is **object-based**
  - **Not a traditional filesystem** with files and folders
  - Actual Amazon storage infrastructure is not specified
- **Buckets** represent a storage collection
- **Objects** are stored in buckets
- Objects are referred to by developer-selected **keys**
- Each object can contain up to 5TB of **data**
- Requests can be **authenticated** or **anonymous**

S3 buckets and objects

- **S3 objects** are referred to using **URIs**, e.g.:
  - `http://mybucketname.s3.amazonaws.com/mydata/file.jpg`
- Buckets serve a few main purposes:
  - Bucket names are the top-level of S3 **namespace**
  - **Charges** for S3 service use accrue at the bucket level
  - Also **usage reporting** is accumulated at the bucket level
  - Overall **access control** configuration for objects in the bucket
  - Buckets can be placed in a given **AWS Region**
  - Optional **versioning** can be enabled
    - Amazon assigns unique version ID to all objects added to a bucket

S3 objects compared to files

- Both S3 objects and files contain **data** and **metadata**
  - In filesystems: modification time, file size, access control, …
  - In S3 metadata is a set of key/value pairs in two groups
    - **System-defined**: time updated, also HTTP headers like Content-Type
    - **User-defined**: key/value data useful to tenants' applications

- Can read/write **parts of files** but S3 has **atomic access**
- Files' metadata can be updated dynamically
  - S3 fixes metadata at the time an object is stored

S3 keys identify objects within buckets

- S3 keys are the **names for objects** within a bucket
  - Previous example URL had key `mydata/file.jpg`
- Delimiters in keys can be used to infer structure
  - Amazon tools support this, but really **keys are a flat structure**
  - In S3, objects in a bucket are treated as a single collection
  - This is unlike filesystems, that really scope files within directories
- Key names can use any UTF-8 character…
  - … but there is a safe set likely to work across all applications

Amazon's many S3 storage classes

- **S3 Standard Storage**—high durability, multi-zone, fast
- **S3 Standard-Infrequent Access**—slower access
- **S3 One Zone-Infrequent Access**—lower resilience
- **S3 Intelligent Tiering Frequent / Infrequent**
  - Monitors access patterns and auto-migrates
- **S3 Glacier Storage**—retrievals take minutes to hours
  - Also S3 Glacier Deep Archive Storage—12 hour retrieval
    - e.g., for organisations with annual audits: retrieve 1 or 2 times a year
- S3 lifecycle management can **automate class change**

Payment for S3

- Prices are based on the location of bucket (its region)
- Two broad cost classes: storage and data transfer
- **Storage costs** depend on storage class (set per object)
- **Data transfer costs** are asymmetric:
  - Transfer in from internet to S3 is free
  - Transfer out to internet is tiered:
    - First 1 GB / month is free
    - Next 10 TB / month is around $0.09 / GB
  - Transfer to other Amazon regions is around $0.02 / GB

Representational State Transfer (REST)

- REST is a notion retrofitted to **HTTP's "object model"**
- **Resources** have a standardised, universal form (URIs)
- Predefined set of **generic operations** are used on URIs
  - **Operations are stateless** on the server's side
- Consider how the web works:
  - URIs are addresses such as `https://www.google.com/`
  - HTTP methods include `GET`, `HEAD`, `POST`, `PUT`, `DELETE`, …
  - First request for webpage from web browser uses `GET` method
  - A form submission might then later use a `POST` method

Amazon S3 REST API

- Interacting with S3 can be done using REST but…
  - Amazon suggest using SDK and/or CLI to ease cert. generation
- Buckets:
  - `HEAD` method indicates **whether bucket exists** and accessible
  - `GET` method **lists objects** within the bucket
  - `PUT` and `DELETE` **create and destroy buckets**, respectively
- Objects—`GET`, `PUT`, `DELETE` do expected operations
  - Also supports `POST` method from web browser HTML forms
- Amazon's REST API documentation is a ~787 page PDF

Web functionality cross-over

- S3 REST suited to **direct use from web browsers**
- GET request for an image on S3 just as from web server
  - S3 is frequently used to **store other web resources**, like video
- S3 wasn't quite a static web hosting service though:
  - **1**—Accessing bucket root produced a **list of objects in bucket**
  - **2**—Errors in accessing objects produced **S3 error messages**
- In 2011 bucket configurations added fix for these issues
- S3 versus GitHub Pages? **Pay for QoS**; S3 is configurable
  - e.g., can control region(s) where buckets are hosted