Swift and Media Storage at Wikimedia

Ben Hartshorne
Operations Engineer
<bhartshorne@wikimedia.org>
What is Media Storage?

- All images, sounds, and videos on all wikis
- All scaled versions of all those images
- It just keeps growing..

http://commons.wikimedia.org/wiki/Commons:

[MIME type statistics](http://commons.wikimedia.org/wiki/Commons:MIME_type_statistics)

Files on Commons by MIME type as of 2012-07-08 06:01:27 (UTC)

<table>
<thead>
<tr>
<th>MIME type</th>
<th>Media type</th>
<th>Files</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/ogg</td>
<td>AUDIO</td>
<td>159,119</td>
<td>137,740,897,523</td>
</tr>
<tr>
<td>application/ogg</td>
<td>VIDEO</td>
<td>15,135</td>
<td>416,045,941,908</td>
</tr>
<tr>
<td>application/pdf</td>
<td>OFFICE</td>
<td>23,970</td>
<td>120,378,314,149</td>
</tr>
<tr>
<td>audio/midi</td>
<td>AUDIO</td>
<td>2,451</td>
<td>13,298,263</td>
</tr>
<tr>
<td>image/gif</td>
<td>BITMAP</td>
<td>130,459</td>
<td>23,933,529,446</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>BITMAP</td>
<td>11,250,437</td>
<td>14,747,665,370,021</td>
</tr>
<tr>
<td>image/png</td>
<td>BITMAP</td>
<td>982,320</td>
<td>543,671,948,791</td>
</tr>
<tr>
<td>image/svg+xml</td>
<td>DRAWING</td>
<td>593,115</td>
<td>137,535,894,060</td>
</tr>
<tr>
<td>image/tiff</td>
<td>BITMAP</td>
<td>84,722</td>
<td>842,335,343,318</td>
</tr>
<tr>
<td>image/vnd.djvu</td>
<td>BITMAP</td>
<td>22,295</td>
<td>274,908,628,366</td>
</tr>
<tr>
<td>image/x-xcf</td>
<td>BITMAP</td>
<td>312</td>
<td>955,816,108</td>
</tr>
<tr>
<td>video/3gpp</td>
<td>UNKNOWN</td>
<td>1</td>
<td>1,015,808</td>
</tr>
<tr>
<td>video/mp4</td>
<td>MULTIMEDIA</td>
<td>1</td>
<td>11,900,528</td>
</tr>
</tbody>
</table>

Total: 13,264,337 | 17,245,197,898,289

http://ganglia.wikimedia.org/latest/?r=year&cs=&ce=&m=&c=Commons:Project_scope/Allowable_file_types
Kitten

From Wikipedia, the free encyclopedia

For other uses, see Kitten (disambiguation).

A kitten is a juvenile domesticated cat.[1] A feline litter usually consists of two to five kittens. To survive, kittens need the care of their mother for the first several weeks of their life. Kittens are highly social animals and spend most of their waking hours playing and interacting with available companions.

Contents [hide]
1 Etymology
2 Birth and development
3 Health
4 Orphaned kittens
5 See also
6 References
7 External links

Etymology

The word "kitten" derives from Middle English kitoun (ketoun, kyton etc.), which itself came from Old French chitoun, chaton: "kitten".[1]

The young of big cats are called cubs rather than kittens. Either term may be used for the young of smaller wild felids such as ocelots, caracals, and lynx, but "kitten" is usually more common for these species.

Birth and development

A feline litter usually consists of two to five kittens. The kits are born after a gestation that lasts between 64
What do we need from a Media Store?

- **Large Capacity**
  - currently 25TB; minimum for growth: 50-100TB

- **Fault Tolerance**
  - any component must be able to fail without impact

- **Medium Throughput**
  - rate of image requests, additions is about 100/s

- **Medium Latency**
  - most end-user actions are served from cache
What is Swift?

OpenStack Storage (http://openstack.org/software/openstack-storage/)

<buzzword>scalable fault tolerant object store</buzzword>

- **Scalable**: Increase cluster size (and throughput) by adding additional hardware; latency shouldn't increase with cluster size
- **Fault Tolerant**: No single point of failure
- **Object Store**: Not a filesystem - stores whole objects
Swift Architecture

Four main server processes

Many ancillary processes for background jobs: synchronization, auditing, replication, etc.
Swift Architecture

Grouped onto two machine types
Machine Types

- **Frontend Proxy Server**
  - dual 6-core CPU
  - 16GB RAM
  - two 250GB SATA disks RAID1

- **Backend Storage Server**
  - dual 6-core CPU
  - 48GB RAM
  - two 160GB SSDs
  - twelve 2TB SATA disks (no RAID)
Swift Architecture

Grouped onto two machine types

- SSD
- SATA
Swift Architecture

- Four main server processes
- Many ancillary processes for background jobs: synchronization, auditing, replication, etc.

Grouped onto two machine types:
- SSD
- SATA

- Traffic
- Proxy Server
- Container Server
- Object Server
- Account Server

5 Proxies
12 Storage
How is Swift used in MW?

- Thumbnails
- Originals
  - Mediawiki FileBackend class has multiple modules; calls Swift using CloudFiles
Thanks!

Ben Hartshorne
Operations Engineer
<bhartshorne@wikimedia.org>

http://wikitech.wikimedia.org/view/Swift
end of presentation

some optional slides follow - they might be used if people ask specific questions
Rewrite middleware

- New thumbnails are scaled on demand
- 404 handler tries to scale images that don't exist
- swift-proxy is built for this
  - in /etc/swift/proxy-server.conf:
    ```
    [pipeline:main]
    pipeline = rewrite healthcheck cache swauth proxy-server
    ```
- rewrite does two things
  - call back to get the scaled version of the image
  - write that scaled version into swift
Query flow: client to scaled image
What about that 404 handler?

Perfect for middleware in the proxy pipeline

[pipeline: main]
pipeline = rewrite healthcheck cache swauth proxy-server

Rewrite does two things:

- Handle 404s
  - if the object doesn't exist in swift
  - call back to mediawiki to generate the image
  - optionally write the generated image into swift
What about that 404 handler?

Perfect for middleware in the proxy pipeline

[pipeline:main]
pipeline = rewrite healthcheck cache swauth proxy-server

Rewrite does two things:
- Change URL into Container / Object

http://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/Persian...
swift://wikipedia-commons-local-thumb.1b/1/1b/Persian_...
Integration with Mediawiki

- MW storage mechanisms abstracted to a FileBackend class with multiple subclasses
  - local filesystem, swift, azure, S3, etc.
- All interactions with the FileBackend implemented as appropriate for each backend storage module
- Swift storage implemented using CloudFiles
  - https://github.com/rackspace/php-cloudfiles
- More detail on this part: Aaron Schulz
Throughput and Latency Performance
Initial tests

- Tried to use apache bench
  - ab is restricted to one URL
  - abmulti can only handle 20k URLs
  - wound up writing my own
- geturls* showed we could get
  - 1300 reads per second
  - 120 writes per second
  - (full details at http://wikitech.wikimedia.org/view/Swift/Performance_Metrics#test_4)

* geturls code available at https://gerrit.wikimedia.org/r/gitweb?p=operations/software.git;a=tree;f=geturls;hb=HEAD
Effect of load on performance

- Under heavy read load
  - PUT and DELETE latency increases
  - GET latency decreases

![Graph showing effect of load on performance](image)
Effect of node failure

- One (out of 5) storage nodes crashing
  - 0.5s timer on connection failures - adjustable
  - 2x read latency (from 100ms to 200ms)
  - 3x write latency (250ms to 750ms)
  - 2.5x delete latency (200ms to 500ms)
- No data (yet) on proxy nodes crashing
Some problems encountered along the way

- Effect of one storage node crashing on performance is too large
  - solved by reducing the connection timeout from 0.5s to 0.1s
- Container listing latency is high
  - solved by moving container and account servers to SSD leaving objects on spinning media
- Consistency issues with rewrite middleware
  - ETags help
  - Still have issues sometimes (cleaner script)
  - solved by having mediawiki write to swift instead
- It's difficult diagnosing problems with rewrite
  - natural effect of asynchronous code (eventlet)
  - eg. stack trace in proxy logs