Geographically distributed storage over multiple universities with NetApp StorageGRID

January 2017

Presenter: Chris Madden
NetApp
Object Stores on the market

Commercial and Open Source Solutions

**Erasure Coding or other fancy algorithms**
- Being efficient
  - Everybody got something
  - More or less good

**API Support**
- Enabling Access
  - Everybody got S3 and Swift
  - But the completeness of the implementations vary

**Multi-Site Support**
- Connect multiple DCs
  - Supported by most
  - Adding more sites is usually difficult

**Expand-as-you-grow**
- Adjust to demand
  - The air is getting thinner…
  - Often on paper, but a pain in a real deployment

Critical in Research Environments
StorageGRID Webscale

- Software-defined Object Storage
- Geo-redundant and Multi-Site
- Policy-based to simplify management at scale
- S3, Swift, NFS and CIFS access
- Durable, low-cost cloud storage for Active Archive
Global Scalability and Namespace

User ingests data in San Francisco

StorageGRID Site 1

StorageGRID Site 2

StorageGRID Site 3

StorageGRID Site 4

User reads data in Tokyo

Data Placement – Policy Driven

16 Sites

70 PB Capacity

100 B Objects

© 2017 NetApp, Inc. All rights reserved. — NETAPP CONFIDENTIAL —
Use case with File Sync'n'Share

- More and more research federations between universities

- Solution:
  - Local owncloud deployment + Owncloud Federation

- But what about data availability, durability and efficiency?

- Solution:
  - Federated data protection between universities through StorageGRID
With StorageGRID

- Capacity can be added dynamically
- New Sites (Universities) can be added dynamically
- Capacity can be moved between sites non-disruptively
Federated Data Protection Policies explained

- Can react to
  - All Objects or just from a specific university
  - Object Key, Metadata and Size
  - Bucket Name
  - Last Access Time
  - Or any combination of the mentioned parameters

- Can influence if data should be
  - Replicated to a set of universities
  - Erasure Coded between universities
  - Tiered to other S3 or Tape Target

- Dynamically changeable!
## Shared Datasets

**Reference Time**: Ingest Time

### Placements

<table>
<thead>
<tr>
<th>From day</th>
<th>store</th>
<th>for</th>
<th>120</th>
<th>days</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Create</th>
<th>copies as</th>
<th>in</th>
<th>or temporarily in</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>replicated</td>
<td>University A</td>
<td>-- Optional --</td>
</tr>
<tr>
<td>1</td>
<td>replicated</td>
<td>University B</td>
<td>-- Optional --</td>
</tr>
<tr>
<td>1</td>
<td>replicated</td>
<td>University C</td>
<td>-- Optional --</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From day</th>
<th>store</th>
<th>forever</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Create 1 copies as erasure coded in Universities A-B-C (EC-6-3)
Final Result Example

Shared Datasets

For Object Type: S3/Swift
Reference Time: Ingest Time
Filtering Criteria:

Matches all of the following metadata:

LOCATION equals EVERYWHERE

Retention Diagram:

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Day 0</th>
<th>Day 120</th>
</tr>
</thead>
<tbody>
<tr>
<td>University A</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
<tr>
<td>University B</td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td>University C</td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
</tr>
<tr>
<td>Universities A-B-C (EC-6-3)</td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Duration

120 days Forever
Keeping it secure!

- Encryption at rest
- Transport layer encryption
- Data Integrity checks
- Tunable Availability
  - 99,9999% (6x 9's) and more
- Tunable Durability
  - > 99,9999999999999% (15x 9's) and more
Three things to take home…

- StorageGRID Webscale = Distributed Object Store
- Focus: Policy-driven data storage over many sites
- Ideal to store large datasets over many years