Object Storage 101
Understanding the What, How and Why behind Object Storage Technologies
Today’s Presenters

Alex McDonald, SNIA –ESF
File Protocol SIG Chair - NetApp

Duncan Moore
Director, Object Storage
NetApp, Inc.

Paul S. Levy
System’s Engineer & Architect
Intel Storage Division
SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.
Topics

- The Emerging Object Storage Market (Market Sizing and Growth)
- Contrasting approaches: Objects, Files & Blocks
- Object Storage Use-Cases
- Components of an Object Storage Solution
- Object Durability Approaches
- Design/Selection Considerations
The World Has Gone Digital

Data processed by Google* every day in 2011

4 billion
Pieces of content shared on Facebook* every day by July 2011

5.5 million
Legitimate emails sent every second in 2011

7 Exabytes
Data traffic by mobile users worldwide in 2011

24 Petabytes
Data processed by Google* every day in 2011

Managing petabytes is commonplace
Block, File & Object

**Block**
- Specific location on disks / memory
  - Tracks
  - Sectors

**File**
- Specific folder in fixed logical order
  - File path
  - File name
  - Date

**Object**
- Flexible container size
  - Data and Metadata
  - Unique ID
Challenges driving the adoption of Object

- Scalability – Accommodate boundless growth
- Durability – Tolerate hard drive, system, and datacenter failures
- Manageability – Accommodate seamless expansion and migration
Workload will guide the choice...

**File Systems**
- Structured datasets
- Lots of readers and writers
- Location/path aware
- Needs volume management

**Object Stores**
- Unstructured data
- Embedded metadata
- Write-once (immutable)
- Location unknown
- No volume management

---

**Fast Data**
- Transactional

**Massive Data**
- Occasional

---

[SNIA Logo]
A Sample Object Use-Case

Media Asset Management use-cases often push the boundaries of traditional storage approaches.

<table>
<thead>
<tr>
<th>Business Requirement</th>
<th>Technical Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deliver an “Entertainment as a Service” offering. Manage a billion+ media across a</td>
<td>• Ability to provide SLO based storage and access</td>
</tr>
<tr>
<td>huge range of sizes (MB’s (\rightarrow) TB’s).</td>
<td>• Erasure Encoding for cost optimization</td>
</tr>
<tr>
<td></td>
<td>• Single Copy support</td>
</tr>
<tr>
<td></td>
<td>• Simplified Install/Expansion/Operations</td>
</tr>
<tr>
<td></td>
<td>• Half million jobs a day</td>
</tr>
</tbody>
</table>
Network Attached File System

- **Scalability**
  - Strict volume and file limits
  - File locking
  - Fixed attributes (metadata)

- **Durability**
  - Volume and/or RAID replication
  - Snapshot
  - Low level solutions (RAID)

- **Manageability**
  - POSIX interface

- **Consistency**
  - Read after write consistent
Object Store

- **Scalability**
  - Flat namespace
  - No volume semantics
  - No Locking/Attributes
  - Contains metadata

- **Durability**
  - Replication or Erasure code

- **Manageability**
  - REST API
  - Low overhead

- **Consistency**
  - Eventually consistent
At what anchor point do you consider a protection policy?

- **Block**
  - LUN’s – Logical container that is referenced.

- **File**
  - Directory – Smallest reasonable unit to secure.
  - Volume/Mount Point – Covenant reference point

- **Object**
  - Object Property – Part of the declaration
  - Namespace - Covenant reference point
What is Erasure coding

- A method of Forward Error Correction which produces set of fragments by which only a subset is needed to re-hydrate.
- Erasure coding policy defines the number fragments that are created with the number of fragments needed to re-hydrate.
- RAID5 & RAID6 are examples of Erasure Code.
  - An 8 element RAID5 would be (7 of 8)
  - An 8 element RAID6 would be (6 of 8)
Erasure Code Encoding

Object

EC Encode Policy (6 of 9)

EC Encoder

Fragment 1 crc
Fragment 2 crc
Fragment 3 crc
Fragment 4 crc
Fragment 5 crc
Fragment 6 crc
Fragment 7 crc
Fragment 8 crc
Fragment 9 crc

Table Assign and Storage write (Metadata)

F1+crc
F2+crc
F3+crc
F4+crc
F5+crc
F6+crc
F7+crc
F8+crc
F9+crc
Erasure Code Decoding

EC Policy = First 6 good Fragments
Example: 6 of 9 Erasure Coding
Example of advanced object level replication

1. Upon ingest make a local replica and 2 remote copies
Example of advanced object level replication

1. Upon ingest make a local replica and 2 remote copies
2. After 90 days remove local copies
Example of advanced object level replication

1. Upon ingest make a local replica and 2 remote copies
2. After 90 days remove local copies
3. After 6 years remove all copies
Design Considerations: Performance

- Throughput and Latency
  - Object Size and Storage Policy
  - PUT/GET Mix
  - Scale of System

- Fault Insertion/Failure
  - Network & CPU impact
  - Application latency
Learn what’s available. RESTful APIs are the norm, but there are many flavors.

- **CDMI (Cloud Data Management Interface):** SNIA standard currently in v1.02
- **SWIFT:** SWIFT is the object storage system component of the OpenStack cloud software project.
- **Amazon S3:** S3 (Simple Storage Service) is Amazon’s cloud storage offering.
- **Proprietary RESTful:** Many Enterprise vendors include support for their own proprietary RESTful API along with one or more “standard” API’s. (ex. NetApp StorageGRID™ SGAPI)

Factors influencing API selection

- What API’s do my desired ISV’s use?
- Do I have needs beyond simple CRUD?
- What expertise is available to me?
Design Considerations: Form Factor

Software Only

+ Can integrate into existing IT infrastructure
+ May enable adoption of commodity h/w
- IT becomes the integrator
- May result finger-pointing support situations

Appliance

+ Simple installation and provisioning
+ Single vendor support
- May have limited performance/scale options
The Emerging Object Storage Market (Market Sizing and Growth)
Contrasting approaches: Objects, Files & Blocks
Object Storage Use-Cases
Components of an Object Storage Solution
Object Durability Approaches
Design/Selection Considerations
After This Webcast

- This webcast will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand
  - [http://www.snia.org/forums/esf/knowledge/webcasts](http://www.snia.org/forums/esf/knowledge/webcasts)

- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog
  - [http://sniaesfblog.org/](http://sniaesfblog.org/)

- Follow and contribute to the SNIA-ESF blog thread on many storage-over-Ethernet topics, both hardware and protocols
  - [http://sniaesfblog.org/](http://sniaesfblog.org/)
Conclusion

Thank You