Analyses of the usefulness of Software Defined Storage Solutions for Web-based Digital Preservation Applications

Peter Tiernan
Systems and Storage Engineer
Digital Repository of Ireland
TCD
Outline

- Storage Requirements
- Storage solutions we tested
- Why we made our choice
- DRI Infrastructure
- DRI bit preservation
The Digital Repository Of Ireland (DRI) is an interactive, national trusted digital repository for contemporary and historical, social and cultural data held by Irish institutions.

The DRI follows the Open Archival Information System (OAIS) ISO reference model and The Trusted Repository Audit Checklist (TRAC)
OAIS Model:

[Diagram of OAIS Model with steps: Producer, Ingest, Archival Storage, Administration, Data Management, Preservation Planning, Access, Consumer]

Source: www.digital-preservation.com
DRI Storage Requirements:

OAIS/TRAC requires the following from storage:

- Minimal conditions for performing long-term preservation of digital assets
- Long Term Preservation of digital assets, even if the OAIS (repository) itself is not permanent or present.
DRI Storage Requirements:

- Open Source/Open Standards
- Independence
- High Availability
- Dynamically Configurable
- Ease of Interoperability (Interfaces, APIs)
- Data Security/Placement (Replication, Erasure coding, Placement, Tiering, Federation)
- Self Contained
- Commodity Hardware
Software Defined Storage vs SAN:

- Lower Cost (Open Source, Commodity hardware)
- No Vendor Lock-In
- Utilise old or existing servers/infrastructure
- Flexibility (IOPS or Space or Bandwidth)
- Incremental hardware upgrade path
Storage Solutions We Tested:

- Hadoop
- iRODS
- IBM GPFS
- Ceph
HDFS:
Why we didn't choose HDFS:

- Only provides RESTful API interface. No posix or RBD.
- Performance geared towards large data sets. I/O of many small files is poor.
- Single point of failure and bottleneck at its Namenode.
- Doesn’t provide any federation
iRODS:
Why we didn't choose iRODS:

- Default Interfaces limited. No Restful, RBD.
- Single point of failure at its iCAT metadata server
- Overlapping functionality with Fedora Commons
GPFS:
Why we didn't choose GPFS:

- Default Interfaces limited. No Restful, RBD.
- Data Replica limit of 2.
- Closed source
CEPH:
Why we chose Ceph:

- We like its distributed, clustered architecture
- Provides complete high availability on install
- Scales out horizontally to massive levels
- Data Security/Placement: Distributed, Replicated
- Many interface options
- Rich, documented, multi-level APIs
- Dynamically configurable
- Good Performance for general use (many small file I/O)
- Solid release schedule, new features
### Findings:

<table>
<thead>
<tr>
<th>Feature</th>
<th>HDFS</th>
<th>iRODS</th>
<th>Ceph</th>
<th>GPFS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>API</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Fedora 3.6.x Driver</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Interface: Posix</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Interface: RBD</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Interface: RESTful</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Dynamic Configuration</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>High Availability: Data</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>High Availability: Service</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Max Raw Storage (PetaByte)</strong></td>
<td>&gt;100</td>
<td>N/A</td>
<td>&gt;100</td>
<td>4 - 10^14</td>
</tr>
<tr>
<td><strong>On-Read Data Checking</strong></td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Max Replicas</strong></td>
<td>512</td>
<td>&gt;2</td>
<td>~2.1 Billion</td>
<td>2</td>
</tr>
<tr>
<td><strong>Federation</strong></td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
DRI Infrastructure
DRI Bit Preservation

- Asset
- Ingest
- Archive Package in Bagit Format
- External integrity checks
- Replicated, Cold Storage, Geo-Replicated
- Scrubbing, Checksumming Erasure Coding
- Scrubbing, Checksumming

DRI Bit Preservation Diagram:
- Asset
- Metadata Checksums
- Replicated Archive Pool
- OSDs
- BTRFS
- Disks
New Ceph Features:

- Asynchronous Geo-Replication
- Erasure Coding
- Tiering
Questions?

DRI:  www.dri.ie
Trinity HPC:  www.tchpc.tcd.ie
Trinity College Dublin:  www.tcd.ie
Links:

Ceph: www.ceph.com
HDFS: hadoop.apache.org
IRODS: www.irods.org

GPFS: www.ibm.com/systems/software/gpfs/

Project Hydra: projecthydra.org
Fedora Commons: www.fedora-commons.org
Apache SOLR: lucene.apache.org/solr/

HAProxy: haproxy.1wt.eu
Performance

Poor performance with low number of OSDs (6) and replication.
Performance

Adding OSDs (26) improves replicated performance