Phobos: an open-source object store implementing tape library support

Patrice LUCAS, patrice.lucas@cea.fr
Phobos: an open-source object store implementing tape library support

Patrice LUCAS, patrice.lucas@cea.fr
## Context

- Exaflopic supercomputers in the 2020's
- Huge amounts of data to ingest: petabytes per day
- Huge amounts of data to store: exabytes

### Next scale of mass storage

<table>
<thead>
<tr>
<th></th>
<th>Today</th>
<th>Tomorrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily production</td>
<td>Hundreds of TB</td>
<td>Petabytes</td>
</tr>
<tr>
<td>Storage system capacity</td>
<td>Hundreds of PB</td>
<td>Exabytes</td>
</tr>
</tbody>
</table>
Benefits of object-based storage

Suppressing POSIX filesystem's bottlenecks

- Object stores have proved their scalability
- Widely adopted for Internet services, Cloud computing, social networks...
Needs for extremely scalable storage systems at a reasonable price

- Object store: horizontal scalability
- Tape library: safe long term storage at low cost
Proposition

Phobos: Parallel Heterogeneous Object Store

- Manages a distributed set of file systems on different storage media technologies
  - hard drives or magnetic tapes
Phobos: optimizes I/O access depending on the storage technology

- minimizing data syncs for magnetic tapes
Phobos: directly manages a tape library (SCSI)

- Supports all common tape drives model
Design guidelines

• Scalability and fault-tolerance
• Based on open formats, open protocols, interoperable
  – E.g. LTFS as tape filesystem (ISO/IEC 20919:2016)
• Simple and common interfaces (REST, object stores API)
• Simple administration (intuitive, admin-friendly CLI)
• Light, easy to deploy, easy to maintain
  – As of today: 14k lines of C, 2.5k lines of Python
History of the project

• 2013: high level design
• 2014-2015: development of the initial version
Scope:
  - Storage on tape, or in a filesystem
  - SCSI-controlled tape library and LTO drives
  - Single server
• 2016: Phobos v1.0 in production
  - Multi-Petabyte storage of genomics data
  - IBM TS3500 library, LTO5/6 drives
• 2019: Phobos made available on github as open-source (LGPL v2.1)
• 2020: Implementation of an S3 front-end (collab. with ICHEC and DDN) + required features in the Phobos core
• Next steps:
  - Advanced IO scheduling
  - Parallelization across multiple servers
Phobos overview, today

Phobos CLI and API (put/get)

Object Store layer
Layouts
- simple
- Mirroring

Resource scheduler

I/O Adapters
- POSIX
- LFS

Metadata key-value distributed DB

Media Tags

prepare media

(m)put/get

access data

Data

Resource scheduler

I/O Adapters

Metadata key-value distributed DB

Media Tags

Phobos overview, today
• IO adapters: multiple storage technologies (Posix, LTFS)
• IO adapters: multiple storage technologies (Posix, LTFS)
• Layout plugins: performance and fault-tolerance
Architecture of Phobos

- IO adapters: multiple storage technologies (Posix, LTFS)
- Layout plugins: performance and fault-tolerance
- Tags: storage partitioning
Architecture of Phobos

- IO adapters: multiple storage technologies (Posix, LTFS)
- Layout plugins: performance and fault-tolerance
- Tags: storage partitioning
- Resource scheduling: optimizes tape fill rate, minimizes tapes mounts
Architecture of Phobos

- IO adapters: multiple storage technologies (Posix, LTFS)
- Layout plugins: performance and fault-tolerance
- Tags: storage partitioning
- Resource scheduling: optimizes tape fill rate, minimizes tapes mounts
- Key-value metadata schema:
  - Distributed NoSQL Database
  - Saved within objects on media (recovery, tape import)
Perspectives

Phobos CLI and API (put/get)

Object Store layer

(m)put/get

Resource scheduler

QOS queues

prepare media

access data

Phobos CLI and API (put/get)

S3

Swift

NFS

Metadata key-value distributed DB

Media Tags

Mirroring

Striping

Erasure coding

Layouts

I/O Adapters

POSIX

LTFS

NVMe

OBJECT

S3

Swift

NFS

Mirroring

Striping

Erasure coding

Layouts

I/O Adapters

POSIX

LTFS

NVMe

OBJECT

prepare media

access data
• Providing S3, Swift and NFS connectors
• Providing S3, Swift and NFS connectors
• Adding new layouts: striping, erasure-coding
- Providing S3, Swift and NFS connectors
- Adding new layouts: striping, erasure-coding
- New IO adapters: NVMe, Object

Perspectives
• Providing S3, Swift and NFS connectors
• Adding new layouts: striping, erasure-coding
• New IO adapters: NVMe, Object
• Media life cycle management: automatic migrations between storage technologies
Perspectives

- Providing S3, Swift and NFS connectors
- Adding new layouts: striping, erasure-coding
- New IO adapters: NVMe, Object
- Media life cycle management: automatic migrations between storage technologies
- Optimizing resource scheduler policies: prioritizing and grouping I/O
Synchronization in the distributed mode:
- Through the distributed key-value DB (object location, resource reservation...)
- Redirection of client requests to the preferred I/O node (max 1 hop)
How easy is it?

Setting up a tape storage in a couple of commands

phobos drive add --unlock /dev/st1
phobos tape add -t lto6 [073200-073222]L6
phobos tape format --unlock [073200-073222]L6

That's done! Your system is ready for I/Os.
Example of use-cases
In production use-case

DNA sequencers

- Multi-petabyte genomics datasets
- In production since 2016

Phobos
- IBM TS3500 tape library (SCSI)
- LTO6 and LTO8 drives

HPC data clusters
Upcoming use case

Object store with an S3 interface

- S3 interface exposed to end-users
- Phobos: high-performance, scalable storage
  - Can manage a wide variety of capacitive storage, including tape libraries
  - Provides an easy/uniform management of these storages
Upcoming use case

Phobos as a Lustre/HSM backend

- Lustre: filesystem user front-end
- Phobos as capacitive and scalable backend (hierarchical storage)
Summary

• Tape object storage at scale (and more)
• Phobos is open-source, available on github:
  https://github.com/cea-hpc/phobos
• Contributions are welcome, as well as testers!
Thank you for your attention!
Patrice LUCAS, patrice.lucas@cea.fr

https://github.com/cea-hpc/phobos