Tiered storage file systems

Challenges and solutions

Nils Haustein
Version 1.0

ESCC
EMEA Storage Competence Center

Agenda

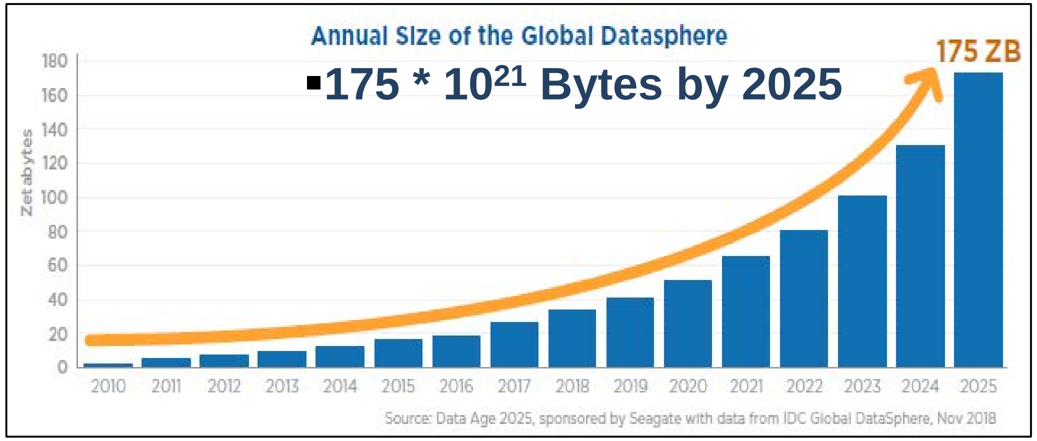
- ▶ Motivation

- Challenges with tiered storage systems

- Solutions

Global data volume predictions

- **By 2025** the global data volume is predicted to be **175 Zettabytes**
  - Includes data generated in data centers (on- and off prem), branch offices and servers, mobile devices and IoT devices



- **$175 * 10^{21}$ Bytes by 2025**

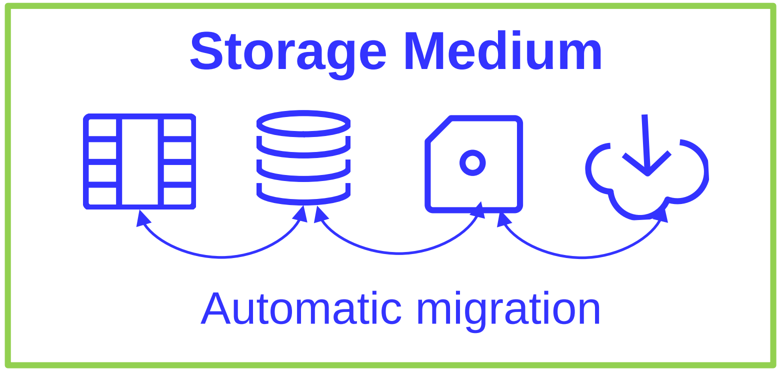
Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

- Where should all this data be stored?
  - Flash, Disk, Tape or Cloud?
  - The majority of this data is used seldom or not after it has been created and processed

Tiered storage

- Tiered storage systems are well suited to store huge amount of data over long period of time

- Tiered storage combines advantaged of different storage media
  - **Flash** for data requiring low latency and high IOPs for immediate processing
  - **Disk** for data that is still accessed and is stored over medium period of time
  - **Tapes** for data that is accessed seldom or never and is stored over long period of time



Why tapes?

- Tapes are cost-efficient storage media for huge volumes of data that are accessed seldom or never
  - Tape do not consume power when not in use
  - Tape storage capacity is extreme scalable
    - In 2017 IBM demonstrated a tape at 330 TB
  - Throughput can be scaled linearly with the number of tape drives
  - Tapes are specially suite for air-gap solutions
  - Standardized formats like LTO and LTFS promise guaranteed future

- However
  - ...Tapes are not suited for all use case because of the high data access latency

Agenda

Motivation

▶ Challenges with tiered storage systems

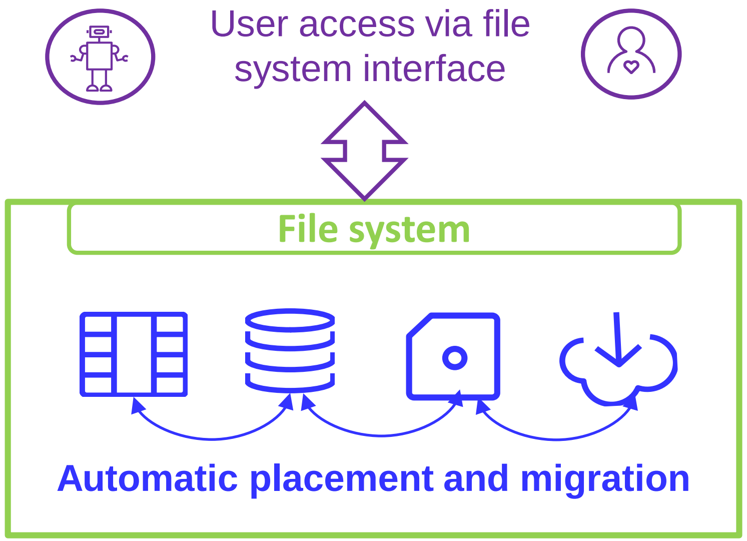
Solutions

Tiered storage file systems

- File system resides on different storage tiers
  - Placement rules define storage tier where files are stored
  - Migration rules define next storage tier and criteria
  - Placement an migration is fully automated

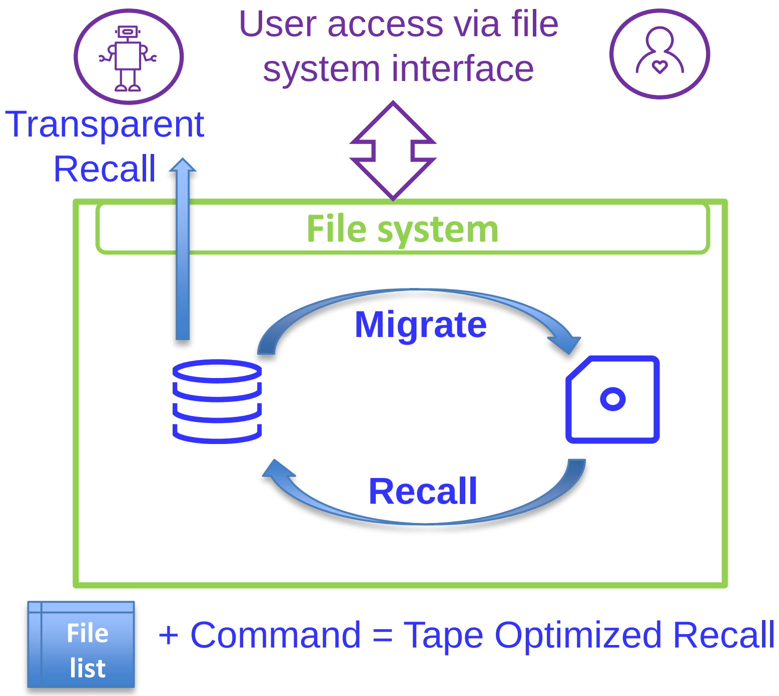
- User access via standard file system interfaces
  - Such as SMB, NFS and POSIX

- User can see and access all files transparently
  - Regardless if files are on disk or tape
  - Upon access files located on tape are copied to disk transparently


User access via file system interface
File system
Automatic placement and migration

Tape operations in tiered storage systems

- **Migration** copies the file to tape and leaves a stub on disk
  - Stub includes a reference to the tape-ID
  - All data blocks of the files are deleted from disk
- **Transparent recalls** copy files from tape to disk in the order of file access
  - Triggered by file access
  - No optimization of tape mount and locate operations
- **Tape optimized recall** copies multiple files from tape to disk in order they are stored on tapes
  - Triggered by administrative command
  - Sorts the files by tape-ID and position on tape and copies in parallel from multiple tapes
  - Much faster than transparent recalls, because it requires less tape mounts and works very well in parallel


Transparent Recall
User access via file system interface
File system
Migrate
Recall
File list
+ Command = Tape Optimized Recall

Challenges with tapes in tiered storage file systems

- Tapes as storage tier in tiered storage file systems are blessing and curse
  - Files on tape visible in file system name space 🙂
  - Access to files on tape cause transparent recall that takes some time 😐

- It gets worse if many files are accessed and recalled transparently
  - Files will be recalled from tape individually, without sorting
  - Causes many tape mounts and start-stop tape motion
  - Takes even longer to get files back

- Unfortunately the user does not easily see in the file system if the file is on tape

Links: Blog Article

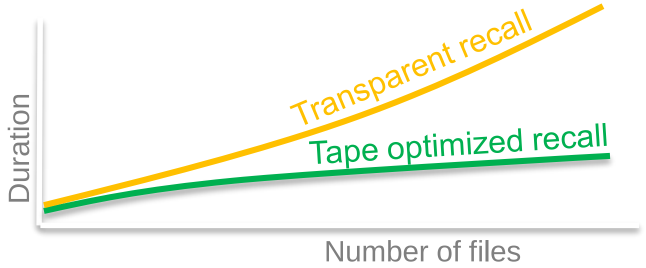
Conclusions

- Not all files are suited to be stored on tapes, for example
  - Files that are frequently accessed – cause many recalls
  - Files that have a short lifecycle – not worth to be migrated to tape
  - Many small files – make tapes performing slow

- Standard filesystems are tape agnostic
  - User has no simple way to see if a file is on disk or tape
  - Files are recalled from tape in the order of access and not in the storage order
  - After accessing files located on tape the user does not see the progress of the recall
  - File system interfaces are widely used and hard to change

Recommendations

- Buffer files on disk, do not directly store on tape because it is slow
- Prevent transparent recalls, use tape optimized recalls instead
  - Tape optimized recalls sort files by tape-ID and position
    - All files on one tape are recalled together



- Provide user a way to determine the location of a file (disk or tape)
- Provide mechanism to allow the user to request the recall of his files
  - Multiple recall requests are queued and performed as tape optimized recall
  - Establish service levels for the time period of recalls (e.g. 2 hours)
- Avoid tape tier in production file system as these have contradicting requirements

Links: Blog Article

Agenda

Motivation

Challenges with tiered storage systems

▶ Solutions

- ▶ OpenStack Swift High Latency Middleware
  - Tape Archive REST API
  - Integration with iRODS – a data management software
  - Amazon S3 Glacier

OpenStack Swift High Latency Media

- Swift HLM is an extension of the OpenStack Swift Object API
  - Support storage media with high latency

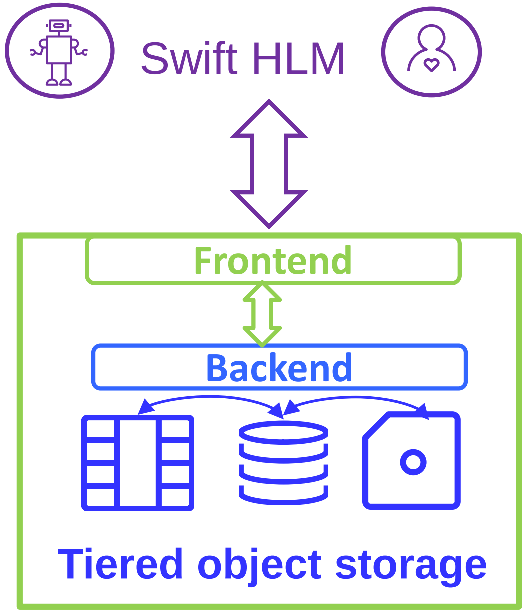
- Swift HLM provides 4 additional calls:
  - Migrate: initiate object or container migration (POST call)
  - Recall: initiate object or container recall (POST call)
  - Request: show running processes (Recall and Migrate, GET call)
  - Status: display migration state of object or container (GET call)

- Example for recall with Swift HLM

```
# curl -X POST -H "X-Auth-Token: $token"
"http://swift-IP/hlm/v1/recall/MyAccount/MyContainer/MyObject"
```

Swift HLM Architecture

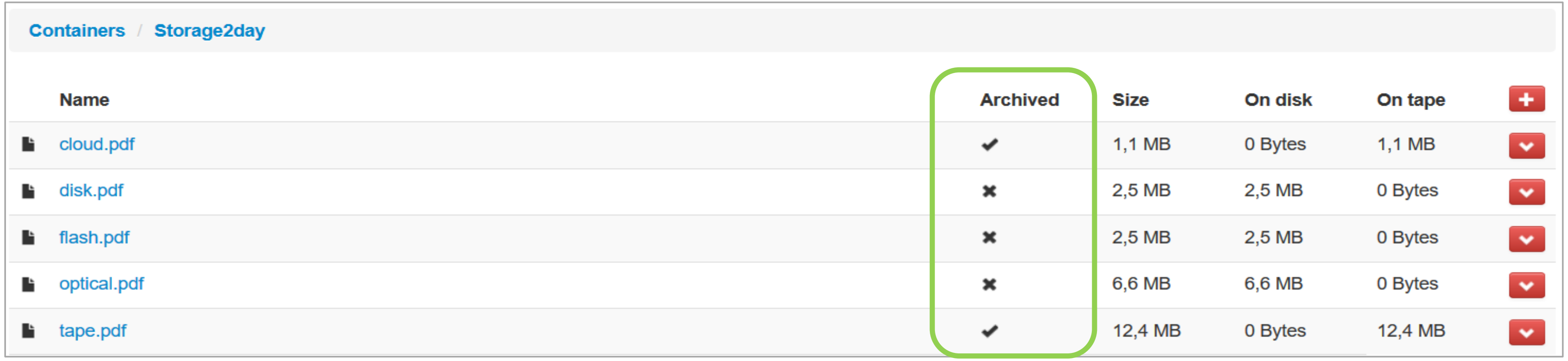
- Swift HLM is based on tiered object storage
  - Including fast and slow storage media
- There is a frontend and backend
  - Frontend: receives and validates the calls and provides it to the backend
  - Backend: executes the requested operation (Migrate, Recall, ...)
- Backend can aggregate migrate and recall requests
  - And execute tape optimizes recalls and migrations in accordance with service levels
- User can easily determine migration state of objects
  - GET operation for migrated object fails, requires recall first
- Sample implementation available with IBM Spectrum Scale and IBM Spectrum Archive EE or IBM Spectrum Protect for Space Management



Links: Swift HLM redpiece | Blog Article

Swift HLM in action

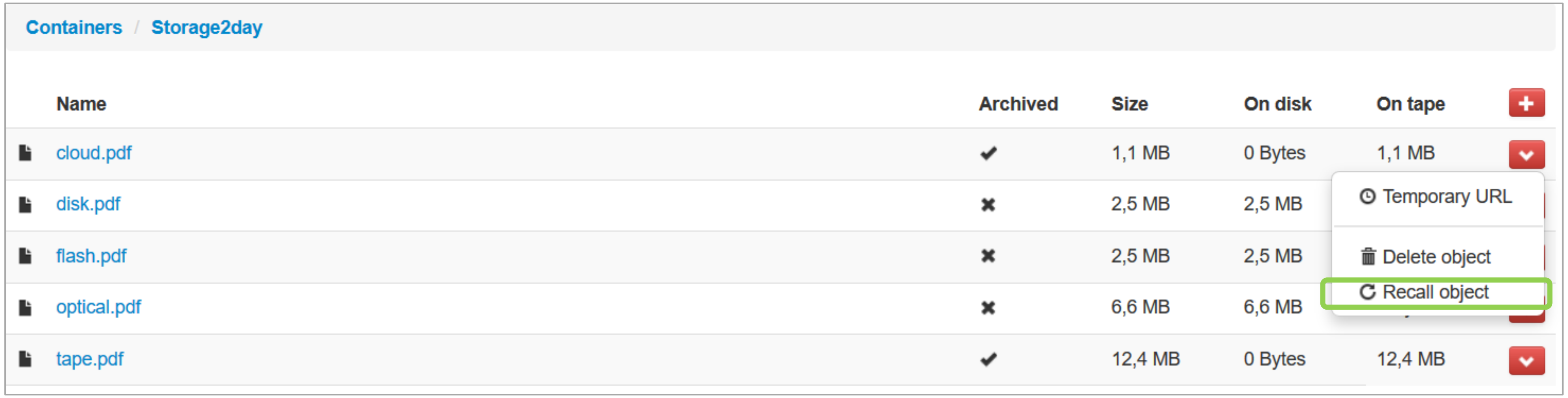
- User can see the storage tier where objects are stored



Containers / Storage2day

Name	Archived	Size	On disk	On tape
cloud.pdf	✓	1,1 MB	0 Bytes	1,1 MB
disk.pdf	✖	2,5 MB	2,5 MB	0 Bytes
flash.pdf	✖	2,5 MB	2,5 MB	0 Bytes
optical.pdf	✖	6,6 MB	6,6 MB	0 Bytes
tape.pdf	✓	12,4 MB	0 Bytes	12,4 MB

- User can initiate the recall



Containers / Storage2day

Name	Archived	Size	On disk	On tape
cloud.pdf	✓	1,1 MB	0 Bytes	1,1 MB
disk.pdf	✖	2,5 MB	2,5 MB	
flash.pdf	✖	2,5 MB	2,5 MB	
optical.pdf	✖	6,6 MB	6,6 MB	
tape.pdf	✓	12,4 MB	0 Bytes	12,4 MB

Agenda

Motivation

Challenges with tiered storage systems

▶ Solutions

OpenStack Swift High Latency Middleware

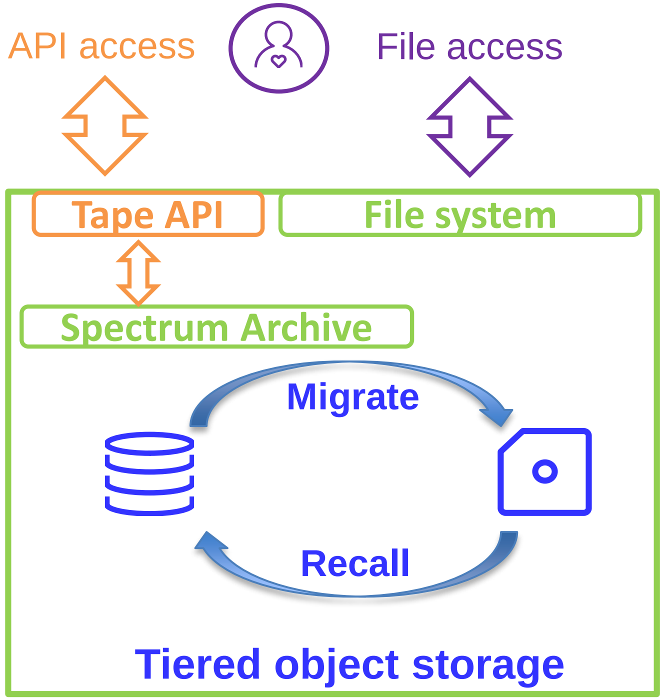
▶ Tape Archive REST API

Integration with iRODS – a data management software

Amazon S3 Glacier

Tape archive REST API

- Tape Archive REST API is an Open Source project allowing users to manage files on tape in a tiered storage file system
  - User can determine the file state and trigger migration and recall of files
  - Requires to disable transparent recalls
- User can access the tiered storage file system via standardized protocols (NFS, SMB, Posix)
  - User can see all files and open files that are not migrated
  - User cannot open migrated files since transparent recalls are disabled
  - User can determine file state using this API
  - User can request the recall of migrated files using the API
  - File recall requests are aggregated and recalled tape optimized in accordance to service levels
- Sample Implementation based on IBM Spectrum Scale and Spectrum Archive
  - IBM Spectrum Archive EE allows to disable transparent recalls



Links: Github Project | Blog Article

Tape archive REST API in action

- Determine file state:

```
# curl -X GET http://host:port/filestate/<path-and-filename>
Response:
  Name: <path-and-filename>
  State: migrated
  Tape 1: VTAP00L5@pool1@lib1
```

- Recall files provided in a list in the request body:

```
# curl -X PUT http://host:port/recall -d "<filelist>"
Response: Recall request queued!
```

- Migrate files provided in a list in the request body:

```
# curl -X PUT http://host:port/migrate?pool1=pool1@lib1 -d "<filelist>“
Response: Migrate request queued!
```

Links: Github Project | Blog Article

Agenda

Motivation

Challenges with tired storage systems

- ▶ Solutions
  - OpenStack Swift High Latency Middleware
  - Tape Archive REST API
  - ▶ Integration with iRODS – a data management software
  - Amazon S3 Glacier

iRODS

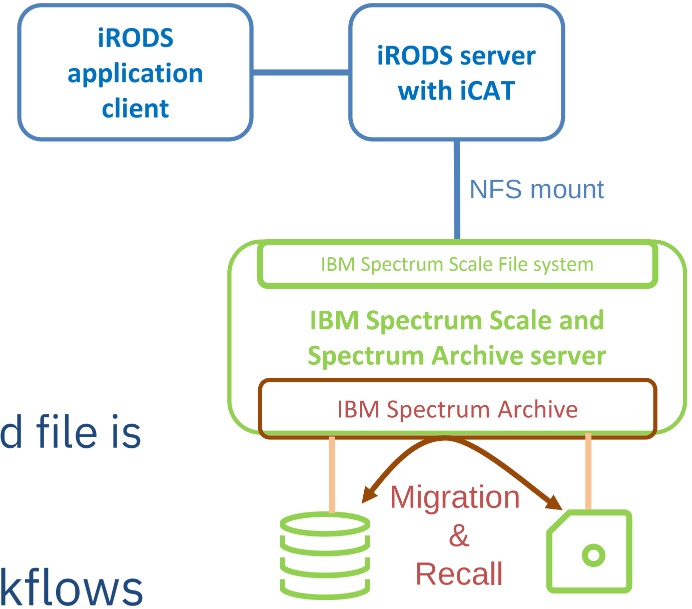
- The iRODS software is a data management layer
  - Sits above the storage and below domain-specific applications
  - Virtualizes data stored in heterogenous storage systems in single name space
  - Maintained by iRODS consortium as open-source software

- iRODS enables users to:
  - Access, manage, and share data across any type or number of storage systems through iRODS APIs (iCommands, REST, WebDAV, Python, C++, Java)
  - Automate workflows through powerful rules and microservices
  - Search and find data through descriptive metadata and query tools

Links: iRODS

Integration of iRODS with IBM Spectrum Archive

- iRODS supports many kind storage systems, like
  - Local and remote file systems, object storage and many vendor solutions
- iRODS can integrate with IBM Spectrum Scale and IBM Spectrum Archive
  - Spectrum Scale file system is exported via NFS and defined as iRODS storage resource
  - Files stored in Spectrum Scale file systems are migrated to tape by Spectrum Archive
  - When migrated files are accessed via iRODS then access request is failed and file is queued for tape optimized recall in accordance to service levels
- iRODS provides powerful rules intercept data operations are execute workflows
  - For example a file open can be intercepted and processed in a custom workflow



iRODS integration in action

- Prevent transparent recalls and queue requested file for tape optimized recall
  - Use iRODS rule to intercept open request, check file state and if file is migrated then add file to queue and present a message to the user
  - Files in queue are recalled in accordance to service levels by Spectrum Archive

```
$ iget -f file1
file /archive/home/mia/col1/file1 is still on tape, and queued for recall.
```

- Display file migration state
  - New user command that executes an iRODS rule to determine file state

```
$ ifilestate /archive/home/mia/col1/file1
Level 0: file /archive/home/mia/col1/file1 is MIGRATED
```

Links: Github Project | Blog Article

Agenda

Motivation

Challenges with tiered storage systems

▶ Solutions

OpenStack Swift High Latency Middleware

Tape Archive REST API

Integration with iRODS – a data management software

▶ Amazon S3 Glacier

Amazon S3 Glacier

- Amazon S3 Glacier is a cheap object storage offering for long term archiving
  - Uploads work normal, retrieves are asynchronous and take longer time
  - S3 Glacier has an own RESTful API - some consider it as an extension to the AWS S3 API
  - Integrates with S3 lifecycle management
- Jobs are used to control retrieves and queries in a S3 Glacier vault
  - Jobs are initiated using the S3 Glacier API and executed asynchronously
  - Jobs can be monitored
- Notifications can be configured to get informed when jobs have completed
  - Uses Amazon Simple Notification Service (Amazon SNS)
- User can use the S3 Glacier API to upload objects, initiate and monitor jobs
  - An existing object cannot be changed or overwritten

Links: Amazon S3 Glacier Documentation

Amazon S3 Glacier API in action

- Upload - simple POST request

```
POST /AccountId/vaults/VaultName/archives
Host: glacier.Region.amazonaws.com
... Glacier header ...
<Request body including the data>
```

  - returns a unique object ID (x-amz-archive-id) used to address the object

- Retrieve - requires a job and subsequent download of the object

```
POST /-/vaults/examplevault/jobs HTTP/1.1
... Header ...
{
  "Type": "archive-retrieval",
  "ArchiveId": "EXAMPLEArchiveId"
  "Description": "My archive description",
  "SNSTopic": "Glacier-ArchiveRetrieval-topic-Example",
}
```

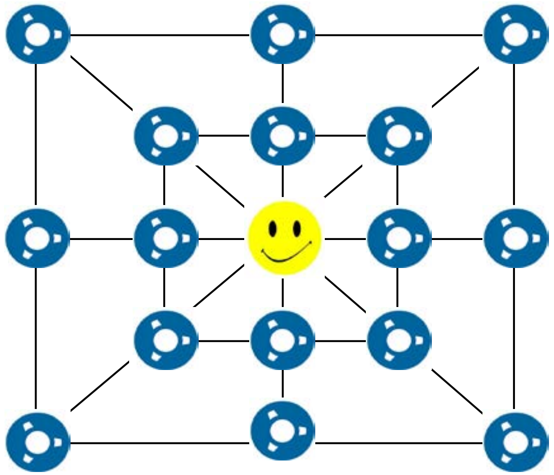
  - Returns a job-ID (x-amz-job-id), after job completion the object can be retrieved using GET:

```
GET /AccountId/vaults/VaultName/jobs/JobID/output HTTP/1.1
Host: glacier.Region.amazonaws.com
... Header ...
```

Links: [S3 Glacier API]()

Summary

- Tapes are beneficial for storing huge volumes of data over long period of time
- Tiered storage systems with tape hide some challenges
  - High latency on data access
  - Standard file system interfaces are not tape aware and cannot be changed easily
- There are solutions that address these challenges
  - For file system and object storage
  - Requires to put in place processes and SLA for file access on tape



Questions?

Disclaimer

This information is for **IBM Spectrum Scale user days 2020**, publication beyond this scope is forbidden

This information is provided on an "AS IS" basis without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Some jurisdictions do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information is provided for information purposes only as a high level overview of possible future products. PRODUCT SPECIFICATIONS, ANNOUNCE DATES, AND OTHER INOFORMATION CONTAINED HEREIN ARE SUBJECT TO CHANGE AND WITHDRAWAL WITHOUT NOTICE.

USE OF THIS DOCUMENT IS LIMITED TO SELECT IBM PERSONNEL THIS DOCUMENT SHOULD NOT BE GIVEN TO A CUSTOMER EITHER IN HARDCOPY OR ELECTRONIC FORMAT.

Important notes:

IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

IBM makes no warranties, express or implied, regarding non-IBM products and services, including but not limited to Year 2000 readiness and any implied warranties of merchantability and fitness for a particular purpose. IBM makes no representations or warranties with respect to non-IBM products. Warranty, service and support for non-IBM products is provided directly to you by the third party, not IBM.

All part numbers referenced in this publication are product part numbers and not service part numbers. Other part numbers in addition to those listed in this document may be required to support a specific device or function.

MHz / GHz only measures microprocessor internal clock speed; many factors may affect application performance. When referring to storage capacity, GB stands for one billion bytes; accessible capacity may be less. Maximum internal hard disk drive capacities assume the replacement of any standard hard disk drives and the population of all hard disk drive bays with the largest currently supported drives available from IBM.

IBM Information and Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both: the e-business logo, IBM, IBM Spectrum Scale, IBM Spectrum Archive, IBM Spectrum Protect

The SNIA logo is a registered trademark of Storage Networking and Industry Association.

The ISO logo is registered trademark of International Organization for Standardization

Microsoft Windows is a trademark or registered trademark of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.

IBM

ESCC

EMEA Storage Competence Center

https://escc.w3ibm.mybluemix.net/#