LTFS and CDMI - Tape for the Cloud

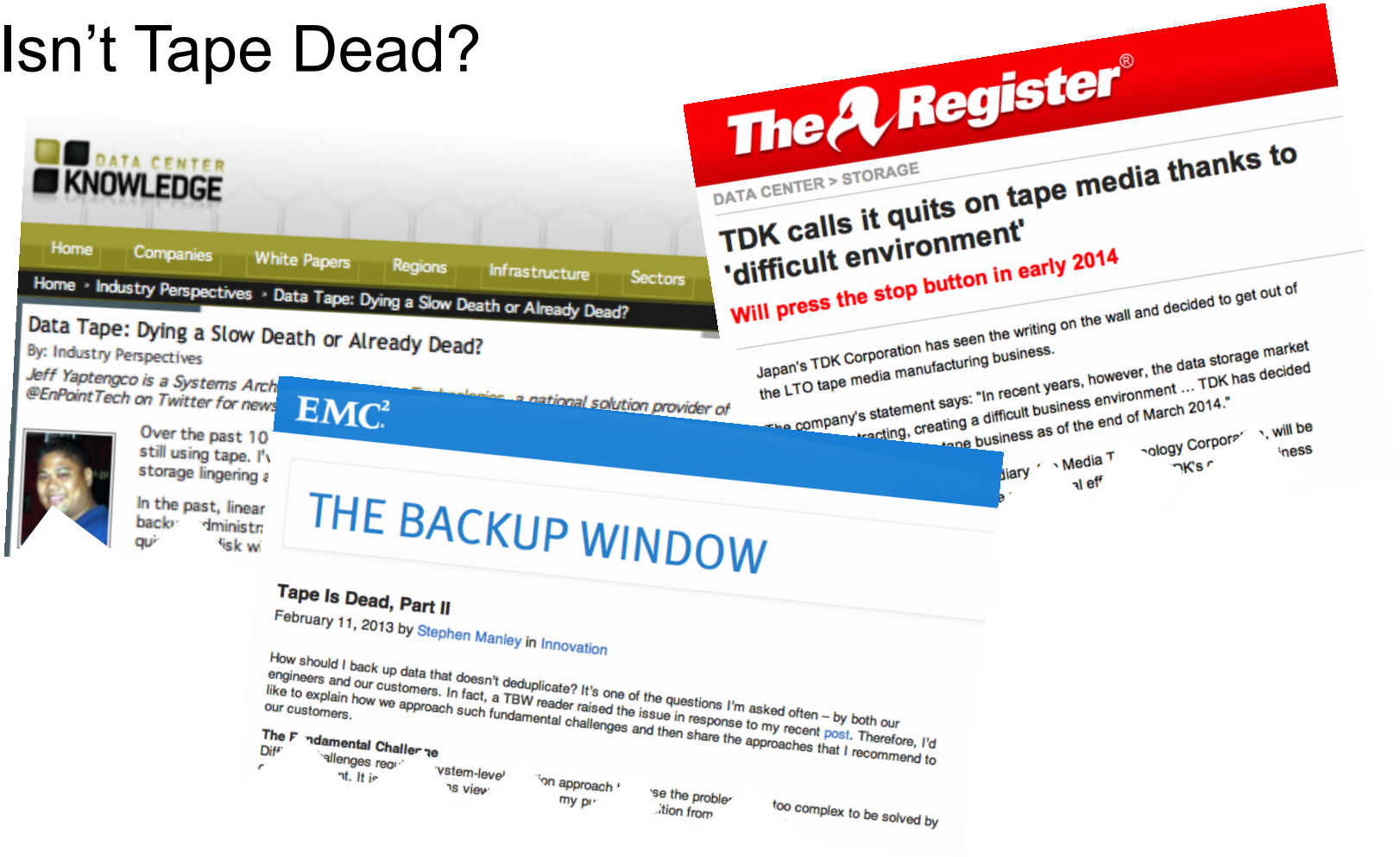
David Slik
NetApp, Inc.

Session Agenda

- Why Tape?
- Tape as a Cloud Protection Backing Store
- Tape as a Cloud Cold Storage Tier
- Tape for Bulk Transport
- Object Storage for LTFS
- Demo

Why Tape?

- Isn't Tape Dead?



Why Tape?

- Isn't Tape Dead?
  - **Not where it costs less then the alternatives!**
- For **PB Scale Archives**, tape has significant economic savings compared to disk:
  - Lower capital cost ($/GB),
  - Lower power & cooling costs, and a
  - Longer amortization period

Why Tape?

- In 2013, 2 PB crossover: Disk CapEx > Tape

Cost Analysis	Disk		Tape
Equipment Cost	$198,000 (1)		$123,200 (2)
Media Cost	$0		$73,300 (3)
Replacement Costs per year	$66,000 (4)		$19,650 (5)
Power and Cooling per year	$7,154 (6)		$1,208 (7)
Floor Space Cost per year	$13,650 (8)		$3,900 (9)
Maintenance Costs per year	$39,600 (10)		$24,640 (10)
Total Capital Costs	$198,000	=	$196,500
Per year Costs	$126,404	>	$49,398

http://snia.org/sites/default/files/CloudTapeUseCases_V1.0.pdf

- Tape OpEx is almost 1/3 that of Disk

Why Tape?

- Tape has flourished in several vertical market niches:
  - Oil and Gas: Tape is environmentally robust and easy to safely transport in harsh environments
  - Media & Entertainment: Tape simplifies workflows and data exchange
  - Archiving & Preservation: Tape provides low bit error rates and long shelf life for large scale archives

Why Tape?

- So, to summarize:
  - If you have many PB's of archival data, tape makes sense
  - If you need to physically move data around, tape makes sense
  - If you need a long shelf life, tape makes sense

Tape and cloud

- If you are a cloud provider, you have several challenges:

1. How do you protect against data loss?
2. How do you provide lower cost offerings?
3. How do you bulk transfer data between/into/out of the cloud?

Tape can help!

Cloud Protection Tier

- Scenario 1 – Reducing cost for data protection
- Assume you are a cloud provider and you want to reduce the probability of customer data loss.
  - Your options are:
    1. Deploy more disks and mirror
    2. Deploy tape and archive
- Google chose #2, and it saved more than money

For more details, search for: “gmail outage 2011 tape”

Cloud Protection Tier

- As tape has higher latency, a cloud provider must still have disk storage
- To handle common failures and maintenance activities, at least two disk locations are required
- However, two disk locations are insufficient to provide sufficient survivability and fault isolation
  - Tape reduces the cost of additional copies
  - Tape reduces the probability of cascading failures that corrupt/destroy all copies

Cloud Archival SLO

- Scenario 2 – Reduced Cost Storage
- If data is stored directly to tape (or through a small staging area), savings can be passed on to the customer
- This allows a cloud service provider to offer a lower-cost differentiated service, similar to what Amazon has done with their Glacier offering

Cloud Protection Tier

- A couple of important restrictions:
  - This only works for infrequently accessed data. If data is randomly access at a frequent enough rate, tape wear will increase costs due to media replacement rates
  - This only works for data where high latencies can be tolerated by the customer
  - This requires different software interfaces in older to handle the higher latency, typically involving notifications of data availability

Cloud Protection Tier

- LTFS standardization reduces complexity, simplifies development, and enables new service offerings:
  - For example, if customer data is stored on standard tapes, in a standard format, that opens the option for a customer to request that the tapes (or copies of the tapes) be sent to them

Which leads us into our third and final scenario…

Bulk Cloud Transfer

- Scenario 3 – Bulk Cloud transfer
- Q. How do you get large amounts of data in and out of the cloud?
- A. Slowly and expensively!
- This is a significant problem for organizations that generate more data then they have bandwidth to send, and when they need to retrieve large amounts of data quickly.

Bulk Cloud Transfer

- Transferring 2 PB over an OC-12 Link

Cost Analysis	Network	Tape
Provisioning Cost	$0	$0
Provisioning Time	0 Days	0 Days
On-Site Data Preparation Cost	$0	$73,300 for tape cartridges (1)
On-Site Data Preparation Time	0 Days	10.3 Days (2)
Transfer Cost	$61,700 (3)	$6,000 (4)
Transfer Time	370 Days	2 Days
Cloud Storage Cost	$0	$9,900 (5)
Cloud Storage Time	0 Days	10.3 Days (2)
Total Cost	$61,700	$89,200
Total Time	370 Days	23 Days

http://snia.org/sites/default/files/CloudTapeUseCases_v1.0.pdf

- Save time!

Bulk Cloud Transfer

- Transferring 10 TB over an 10 Mbit/sec Link

Cost Analysis	Network	Tape
Provisioning Cost	$0	$0
Provisioning Time	0 Days	0 Days
On-Site Data Preparation Cost	$0	$366 for tape cartridges (1)
On-Site Data Preparation Time	0 Days	1 Day (2)
Transfer Cost	$1,852 (3)	$52 (4)
Transfer Time	92.6 Days	2 Days
Cloud Storage Cost	$0	$50 (5)
Cloud Storage Time	0 Days	1 Day (2)
Total Cost	$1,852	$468
Total Time	92.6 Days	4 Days

http://snia.org/sites/default/files/CloudTapeUseCases_v1.0.pdf

- Save time AND money!

Bulk Cloud Transfer

- The LTFS TWG is working on a standard way to transfer collections of data:
  - An XML manifest that describes:
    - Which tapes are used to store the data
    - Which files, directories and objects are being transferred
    - Fixity and integrity verification information
    - Instructions on how to merge data into an existing namespace
  - A standard workflow for bulk data transfer

Demonstration

Ruby Cloud -> LTFS Transfer Demonstration

Object Storage for LTFS

- LTFS provides:
  - Standardized POSIX-style directory and files
  - Standard file metadata and ACL storage
  - Standard tape spanning for large files
- This reduces the complexity of using tape as a backing store, and simplifies development
- The LTFS TWG has begun an effort to standardize how objects are stored on tape

Object Storage for LTFS

- Storing objects on LTFS adds:
  - Support for rich metadata
  - ID-based namespaces for object access
  - Support for composite objects (Queues, etc)
  - Support for object versioning

- This allows objects from object storage systems using Azure, CDMI, S3, and Swift to be stored on LTFS and accessed in a standard way

Object Storage for LTFS

- CDMI <-> LTFS Mapping Examples:
  - CDMI Named Data Object “LTFS.pdf”
  - CDMI Unnamed Data Object “00007ED90…”
  - CDMI Container “SDC 2013”
  - CDMI Queue “Messages”
- S3 & Swift mappings in the works
  - Standard Header Metadata mapping

Object Storage for LTFS

- CDMI Named Data Object “LTFS.pdf”
  - Metadata “Author” : “LTFS TWG”
- LTFS Layout:

/	LTFS Root
/LTFS.pdf	LTFS file with object name as file name
ltfs.vendor.cdmi.objectid	“00007ED90010F0E4FA063BCEB659D6ED”
ltfs.vendor.cdmi.mimetype	“application/pdf”
ltfs.vendor.cdmi.metadata	{“Author” : “LTFS TWG”}
ltfs.vendor.cdmi.valuetransferencoding	“Base64”
/cdmi_objectid/	Object ID Container
/cdmi_objectid/00007ED90010F0E4FA0…	Symlink to /LTFS.pdf

Object Storage for LTFS

- CDMI Unnamed Data Object “00007ED90…”
  - Metadata “Conference” : “SDC”
- LTFS Layout:

/	LTFS Root
/cdmi_objectid/	Object ID Container
/cdmi_objectid/00007ED90010A49F2A0…	LTFS file with object ID as file name
ltfs.vendor.cdmi.objectid	“00007ED90010F0E4FA063BCEB659D6ED”
ltfs.vendor.cdmi.mimetype	“application/pdf”
ltfs.vendor.cdmi.metadata	{“Conference” : “SDC”}
ltfs.vendor.cdmi.valuetransferencoding	“Base64”

Object Storage for LTFS

- CDMI Container “SDC2013”
  - Metadata “cdmi_latency” : “1000000”
- LTFS Layout:

/	LTFS Root
/SDC2013/	LTFS directory
ltfs.vendor.cdmi.objectid	“00007ED900105E38846F7EAA6C061CA7”
ltfs.vendor.cdmi.metadata	{“cdmi_latency” : “1000000”}
/cdmi_objectid/	Object ID Container
/cdmi_objectid/00007ED900105E38846F…	Symlink to /SDC2013/

Object Storage for LTFS

- CDMI Queue “Messages”
- LTFS Layout:

/	LTFS Root
/Messages	LTFS file with queue name as file name
ltfs.vendor.cdmi.objectid	“00007ED90010A49F2A0F1F996095A626”
/Messages.cdmi_queue/	LTFS directory for queue values
/Messages.cdmi_queue/0	LTFS file corresponding to first queue value
ltfs.vendor.cdmi.mimetype	“text/plain”
ltfs.vendor.cdmi.valuetransferencoding	“UTF8”
/Messages.cdmi_queue/1	LTFS file corresponding to next queue value
ltfs.vendor.cdmi.mimetype	“text/plain”
ltfs.vendor.cdmi.valuetransferencoding	“UTF8”
/cdmi_objectid/	Object ID Container
/cdmi_objectid/00007ED90010A49F2A0…	Symlink to /Messages

Next Steps

- Read the SNIA Cloud Tape Use Cases document:
  - http://snia.org/sites/default/files/CloudTapeUseCases_v1.0.pdf
- Join the SNIA Joint Cloud/LTFS Technical Working Group
- Active projects include:
  - Cloud Data Transfer Workflow & XML
  - Object storage for LTFS Tape

Thank You!

Questions and Answers

Contact Info:

dslik@netapp.com