RAIT and Other Software Technologies for Long-Term Massive Tape Archives

Harry Hulen
harryhulen@gmail.com

High Performance Storage System

20-year collaboration of 5 DOE labs and IBM

HPSS

Cluster-based hierarchical storage

	Site	PBs	Millions of Files
	ECMWF	41.49	111.4
	NOAA R&D	34.66	49.3
TOP500 SUPERCOMPUTER SITES #1	LLNL SCF	30.69	292
	LANL SCF	29.63	232.4
TOP500 SUPERCOMPUTER SITES #6	ORNL	27.62	39.1
	BNL	27.05	67.5
TOP500 SUPERCOMPUTER SITES #9	CEA Secure	21.81	4.7
	DKRZ	20.41	25.9
	LBNL User	18.51	134.4
	IN2P3	17.62	35.5
	NCAR	17.03	108
	LLNL OCF	15.16	260.5
	UKMO	14.07	78.4
	DWD	12.51	24.3
	SLAC	11.86	7.3
	LBNL Backup	10.82	15
	IU	9.16	48

	Site	PBs	Millions of Files
	KEK	5.44	13.6
TOP500 SUPERCOMPUTER SITES #3	ANL	4.62	289.8
	PNNL	3.88	46.9
	LaRC	3.68	13.4
	RZG	2.81	6.5
	CEA Open	2.73	1.4
	LANL OCF	2.7	41.7
TOP500 SUPERCOMPUTER SITES #2	Riken	2.6	11
	HLRS	2.38	3.5
	NCDC	2.13	75.1
	Purdue	2.1	17.6
	SciNet	1.61	63.5
	SNL Secure	1.13	2.7
	WUSTL	0.56	0.1
	LoC	0.49	5.2
	SNL Open	0.46	2.7
	JAXA	0.46	5.8

Uses of tape

- Backup and restore of disk-resident data
- Tape-only file systems typified by Linear Tape File System (LTFS)
- Space management of disks (also known as hierarchical storage management)
- Massive, long-term file repository or archive

This talk focuses on some technologies of interest to archivists and IT custodians of long term massive tape archives.

Hittite Clay Tablets

Our ancestors in data storage included the Hittites in modern-day Turkey who left us 25,000 clay tablets about 3500 years ago

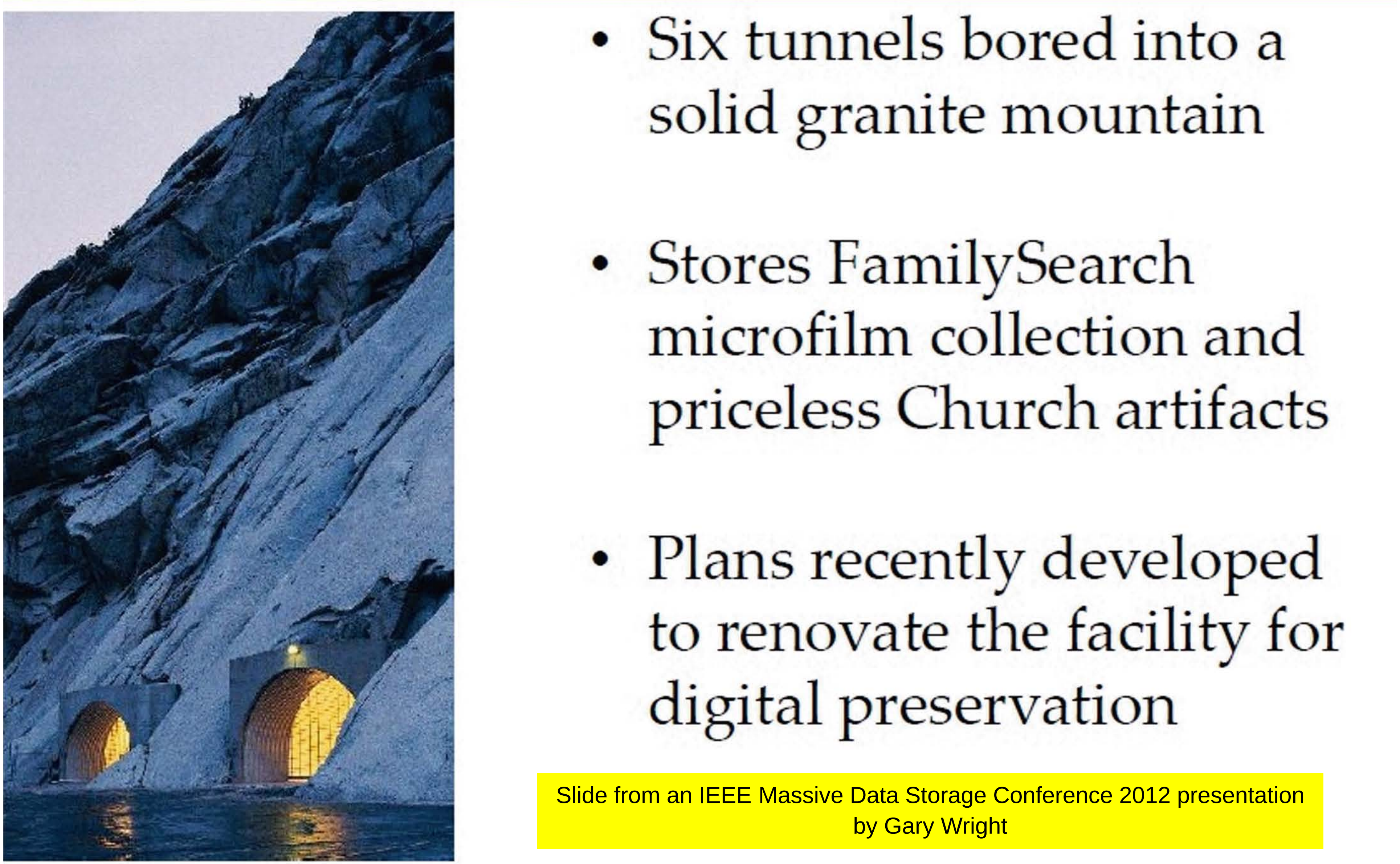
Definition of Archive

With apologies to professional archivists, I will define the "archive" or "long term file repository" as follows:

- A massive collection of digital information
- That the archivist is responsible for passing on to the next generation
- Without loss or degradation*
- That does not depend on proprietary middleware technology that could hinder migration to new middleware technology
- That is internally self-defining and that will be readable by an educated professional in a field where the data has meaning.

* OK, we cannot guarantee perfection.
Here we will skip the math and focus on methods to get close.

A Latter Day Saints Church archive site



- Six tunnels bored into a solid granite mountain
- Stores FamilySearch microfilm collection and priceless Church artifacts
- Plans recently developed to renovate the facility for digital preservation

Slide from an IEEE Massive Data Storage Conference 2012 presentation by Gary Wright

Archive ←→Tape

- To re-state the obvious, we are here because we are convinced that the best medium for a massive archive is tape

  - Longest media life (well, not as long as clay tablets)
  - Lowest bit error rate
  - Greenest - Least energy consumption
  - Least physical space requirement
  - Lowest long-term cost of ownership (hard to beat clay tablets)
  - *And remarkably, most headroom for growth (see next slide)*

- This talk focuses on LTO, but concepts (if not specifics) also apply to IBM’s TS1140 and Oracle’s T10000C tape offerings

Areal Density Scenarios relative to 2014

- **HDD**
  - Conservative: 20% density increases achievable
  - Aggressive: 30% density increases are challenging
- **NAND Flash**
  - Conservative: 20% density increases are achievable given the lithography roadmap strategies project reducing feature size 10% annually
  - Aggressive: Sustained 30% density increases are difficult given the conventional understanding of lithography roadmaps and time driven optical processing tooling strategies. However, INTEL-MICRON has demonstrated a 40% areal density improvement from 2010 to 2011.
- **TAPE**
  - Conservative: 40% density increases achievable with anticipation of following the LTO Roadmap presently at Generation 5
  - Aggressive: 80% density increases are possible since the needed transducer technology presently exists in the HDD environment but “mechanical” issues related to positioning, wear, and tape stability must be addressed – not NANOSCALE issues

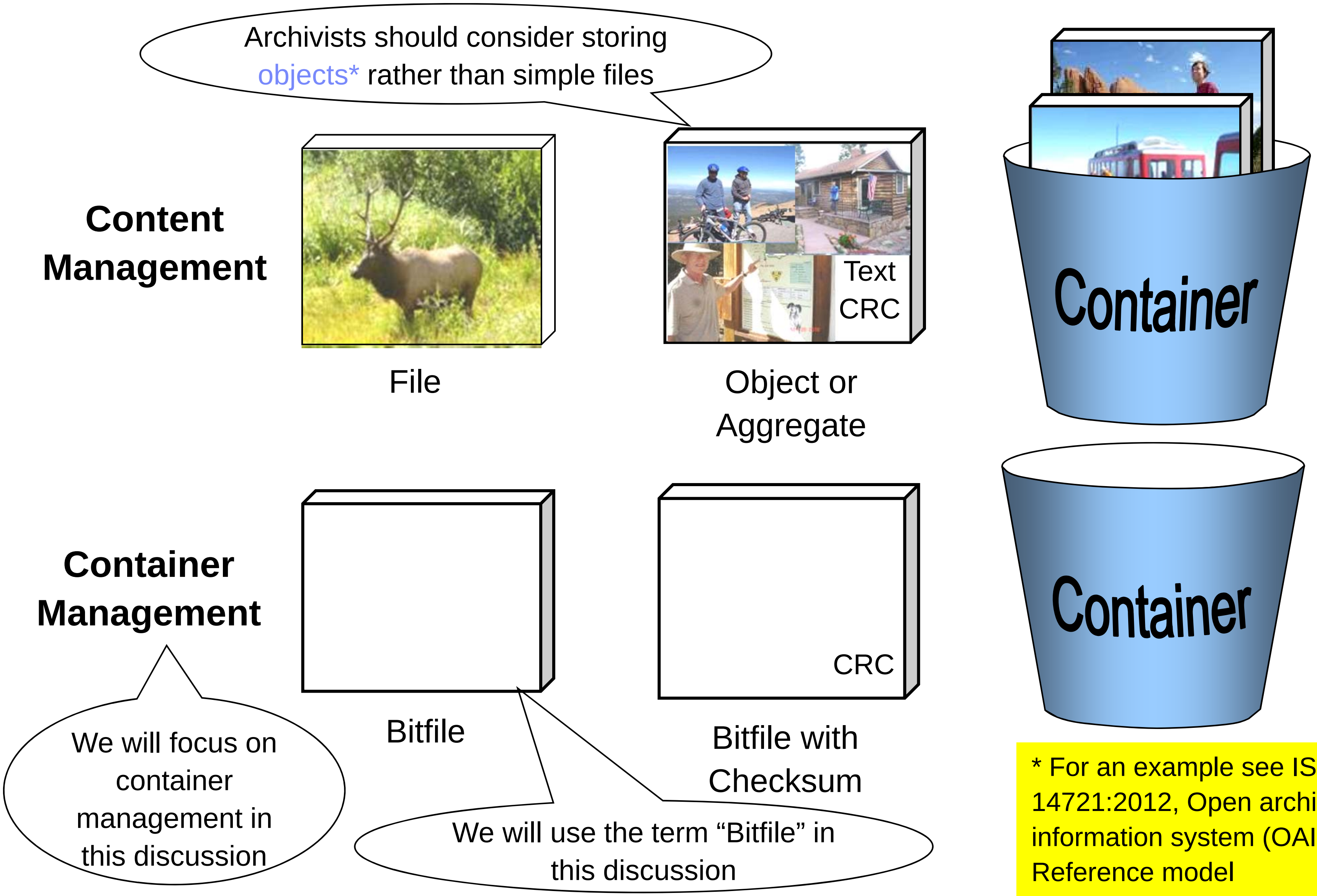
From a featured presentation at IEEE Conference on Massive Data Storage, April 2012, by Dr. Robert Fontana, 102 issued patents in thin film magnetic structures and past president of the IEEE Magnetics Society. http://storageconference.org/2012/Presentations/M09.Fontana.pdf

Error detection and correction codes: keys to archives

- Checksum or Code
  In common usage, any record included in or attached to a bitfile that enables errors to be detected. Can include any of the following:
- Cyclic redundancy check (CRC)
  In common usage, any code that can detect but not correct an error.
- Error correcting code (ECC)
  A code that can detect an error and can correct some errors.
  - Erasure code
    A high-function ECC that can detect and correct an “erasure” (loss) of multiple bits
    Example: 10100011101010110101XXXXXXXX11110010101010101010010110PPPPPPPP
  - Reed-Solomon code (R-S)
    The most common type of erasure code and a name often applied to any erasure code
  - Parity (P or Q)
    P is often used as shorthand for an erasure code. Q is for a second nested erasure code.

- Cryptographic hash function
  A code that can provide strong assurances about data integrity, regardless of whether changes are accidental or maliciously introduced. Not a focus of this discussion.

Define “Bitfile” (from IEEE Mass Storage Reference Model)



Archivists should consider storing objects* rather than simple files

Content Management

File

Text
CRC

Object or Aggregate

Container

Container Management

Bitfile

CRC

Bitfile with Checksum

Container

We will focus on container management in this discussion

We will use the term “Bitfile” in this discussion

* For an example see ISO 14721:2012, Open archival information system (OAIS) Reference model

A Fundamental Tape Archive Question

How do I know that the bitfile I just wrote to tape was written correctly?

Mother of All Checksums – the one you create

- Uncle Harry’s first rule for archival:

- Create a checksum for every important bitfile at the earliest opportunity,
- Ideally at the time and place that that you have determined the bitfile to be “correct”, usually in computer memory.
- The checksum must be bound to the bitfile, ideally inside the bitfile but possibly in an accompanying “attribute” of the bitfile,
- And must remain with it forever

- This “Mother of All Checksums”
  - Is not touched by the OS or middleware and is therefore the one that protects you when others fail
  - And it ***Transcends migrations to new hardware and middleware***

- In this presentation we will discuss checksums (codes) that are applied by middleware and hardware

A Standards Committee is Helping Us!

- The standards committee that governs tape and disk I/O protocols is the **ANSI T10 Technical Committee.**

- Of particular interest to us is a **T10 SCSI Stream Commands (SSC-4)** standard, which applies to tape

- SSC-4 includes a **Logical Block Protection (LBP)** Feature for tape, which makes it much easier to verify that data on your tapes is valid

- We will also briefly mention a **T10 SCSI Block Commands (SBC-3)** verification standard for disks

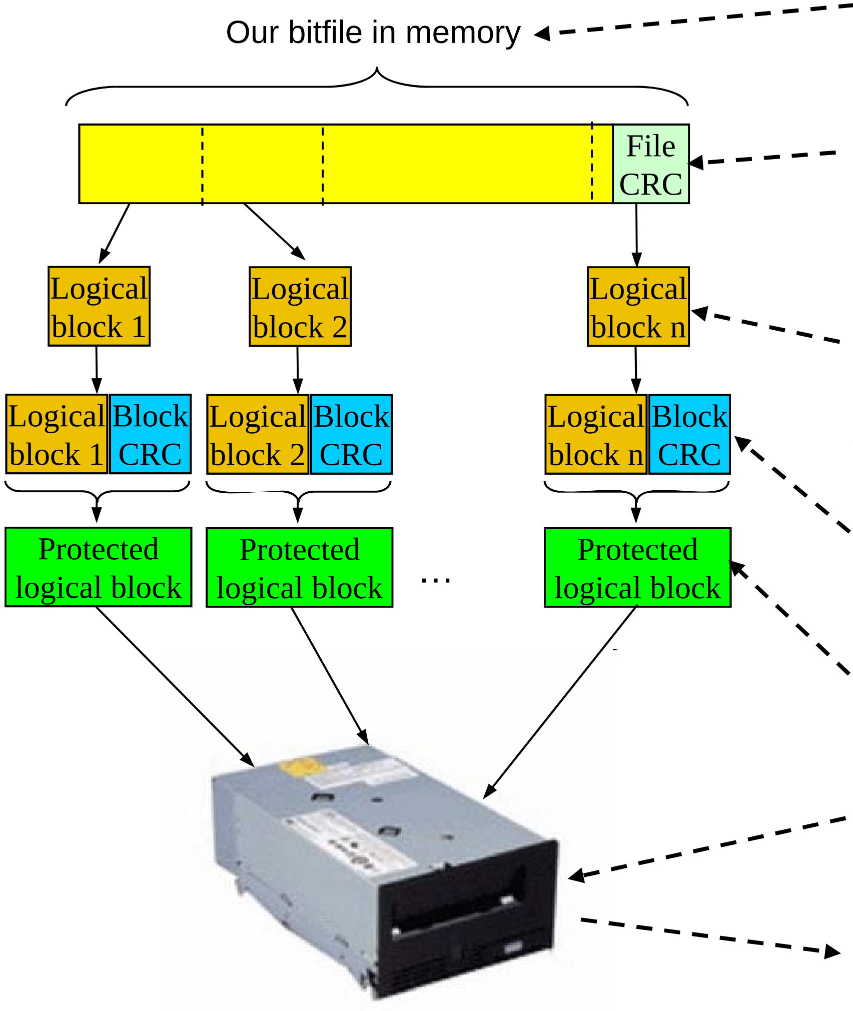
- Why SCSI?
  Tapes and disks use SCSI commands over FC and Serial SCSI.
  T10 owns SCSI.

Tape Logical Block Protection starts with LTO-5

- With LTO-4 and earlier, the only way to be sure a bitfile was written to tape correctly was for software to read it back and compare, which is very slow.
- LTO-4 drives created and used a checksum to verify that the data on tape was exactly what the tape drive intended…
- …BUT, the data could have been corrupted in transit from the computer over a network to the tape drive.
- NOW, because of the T10 SSC-4 Logic Block Protection standard, that checksum can be created in the software that writes the logical block, usually in what I am calling “Middleware”.
- The LTO-5 tape drive can use this received checksum to verify that the tape actually received the logical block that the computer software sent,
- Thus, the Logical Block Protection standard eliminates* uncertainty about whether your data was transferred correctly and written accurately to tape.

- * Tape HW people remind us that bad SW can always defeat the best HW

How a bitfile becomes a set of one or more “logical blocks”



User presents a bitfile to tape middleware.

The bitfile may have a file-level “Mother of All Checksums”, but the tape middleware is not aware of this checksum and treats it as part of the file.

Tape middleware breaks bitfile into “logical blocks” of a suitable size (~256KB to 1GB) to write to tape drive.

Middleware, generates and appends a checksum (CRC) to each logical block.

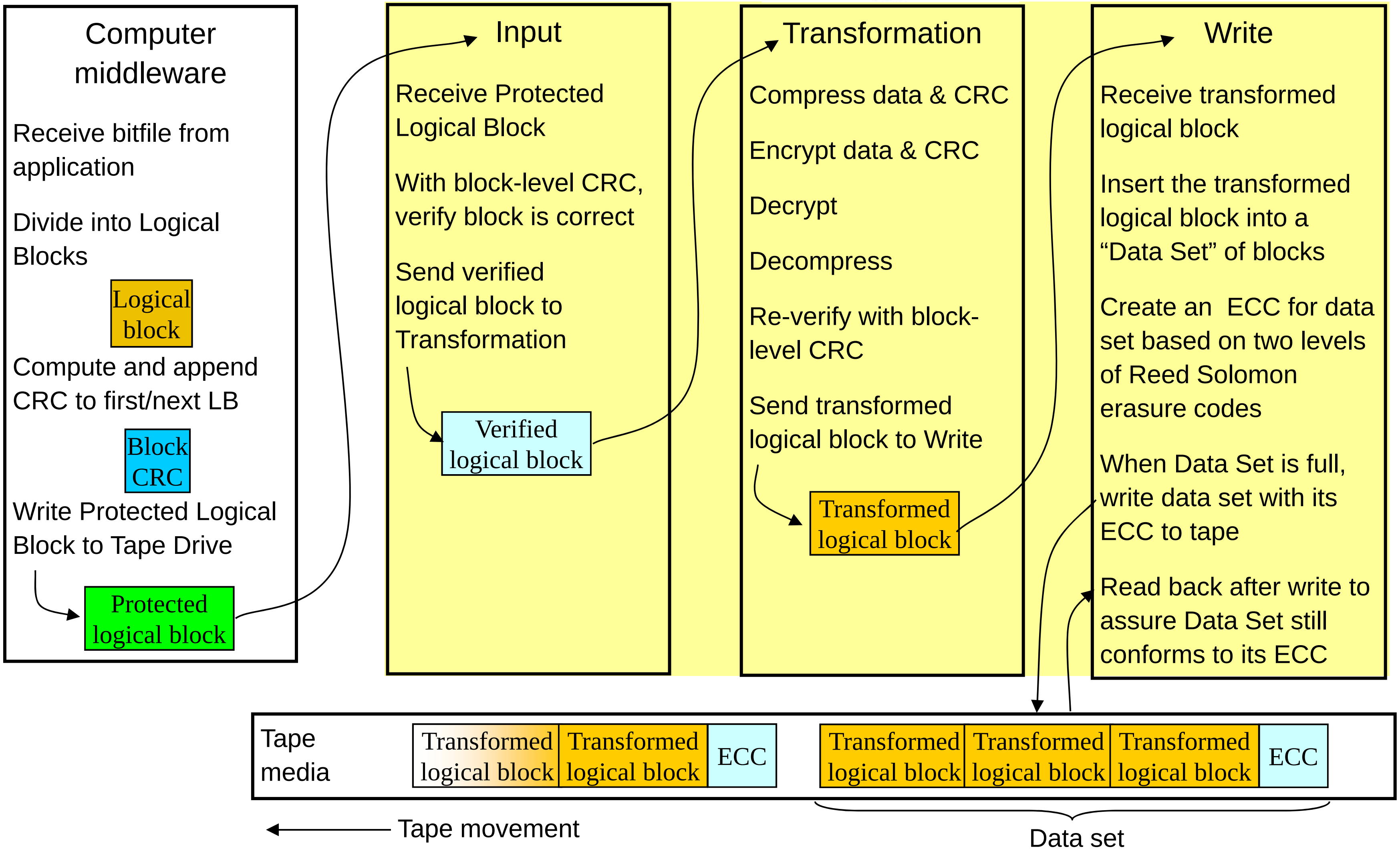
The result is a string of protected logical blocks (data+CRC).

And these protected logical blocks are written to a tape drive tape.

(Process continues on next slide)

A look inside the LTO-5 drive

(From previous slide)



Computer middleware

Receive bitfile from application

Divide into Logical Blocks

Logical block

Compute and append CRC to first/next LB

Block CRC

Write Protected Logical Block to Tape Drive

Protected logical block

Input

Receive Protected Logical Block

With block-level CRC, verify block is correct

Send verified logical block to Transformation

Verified logical block

Transformation

Compress data & CRC

Encrypt data & CRC

Decrypt

Decompress

Re-verify with block-level CRC

Send transformed logical block to Write

Transformed logical block

Write

Receive transformed logical block

Insert the transformed logical block into a “Data Set” of blocks

Create an ECC for data set based on two levels of Reed Solomon erasure codes

When Data Set is full, write data set with its ECC to tape

Read back after write to assure Data Set still conforms to its ECC

Tape media

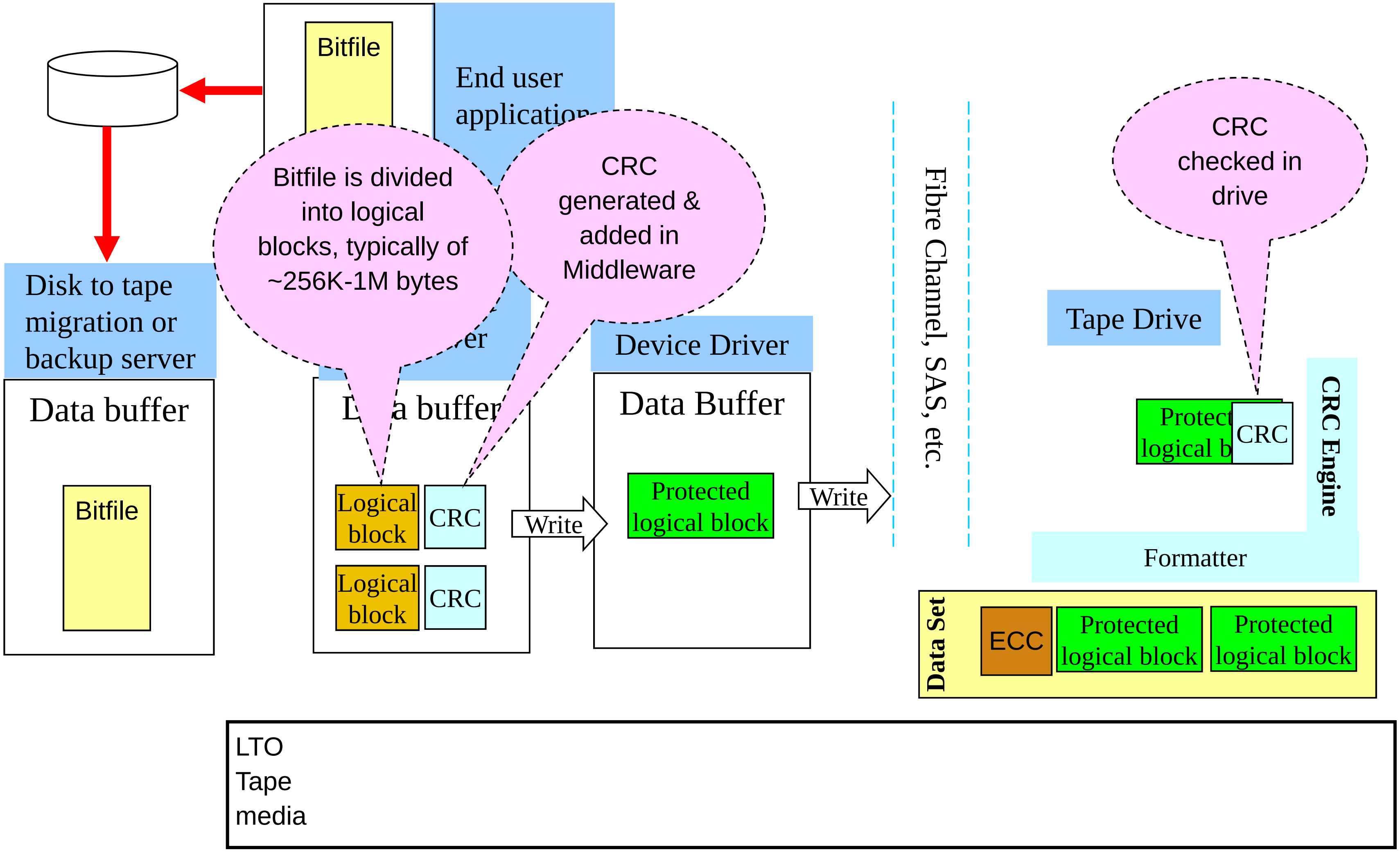
Transformed logical block | Transformed logical block | ECC | Transformed logical block | Transformed logical block | Transformed logical block | ECC

Tape movement

Data set

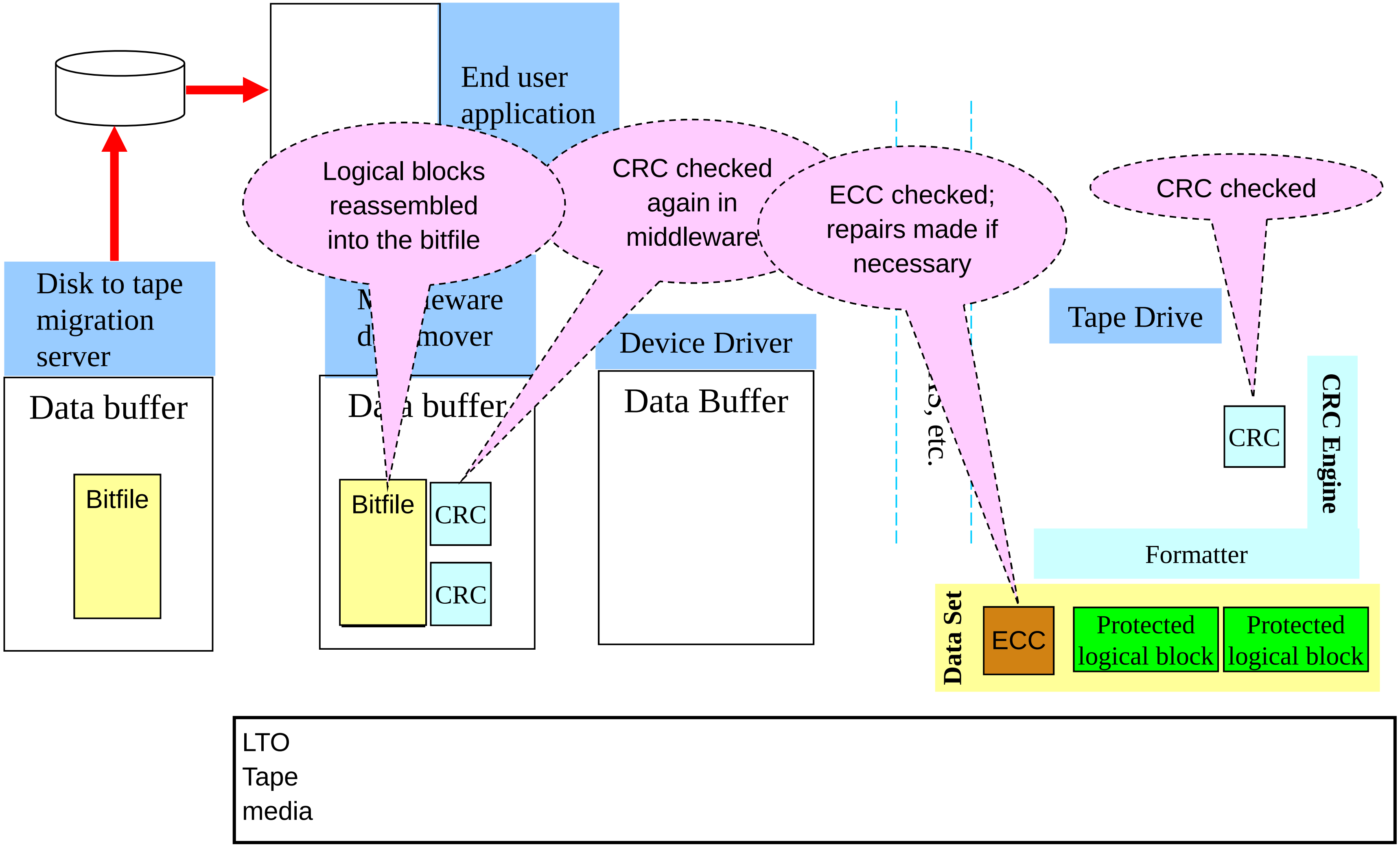
Writing a file to tape using logical block protection

(This was an animated slide in the live presentation)



- Protected Logical Block with CRC is from the T10 standard
- Data Set with ECC is from an LTO collaboration standard

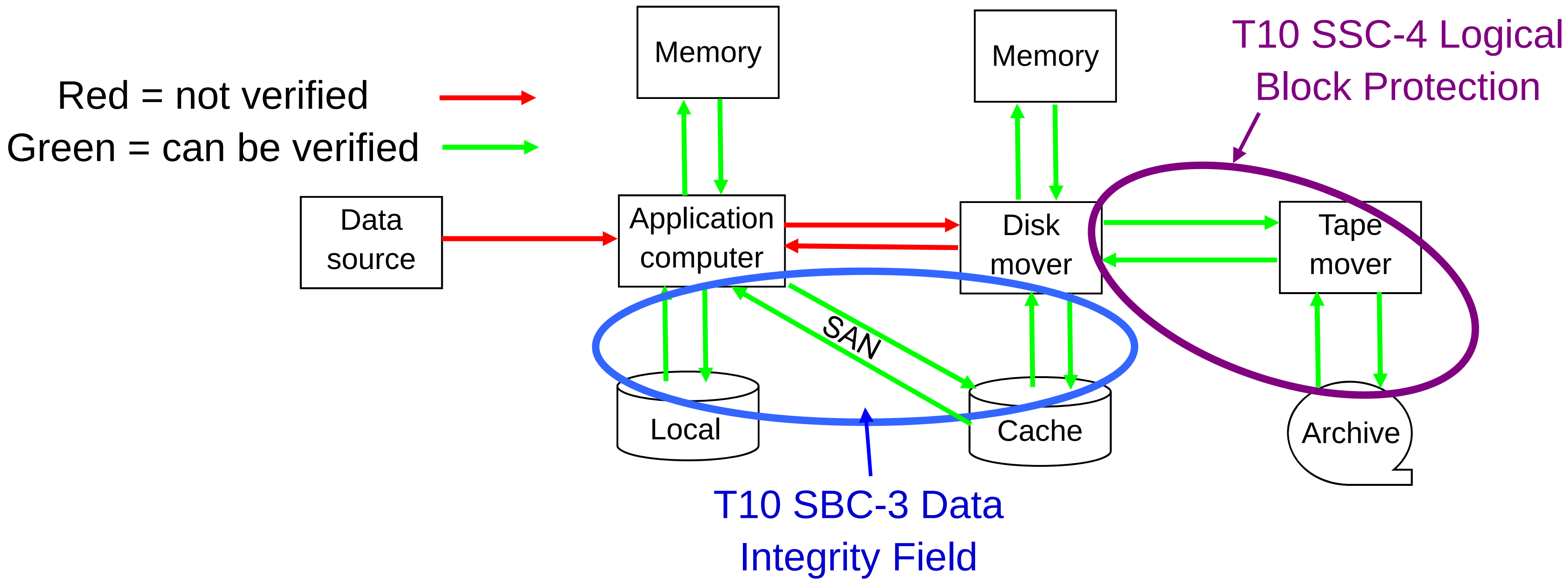
Reading a file from tape using logical block protection
(This was an animated slide in the live presentation)



- Data Sets are read into tape drive memory, validated and if necessary repaired using ECC
- Protected logical blocks are then read from tape drive memory, validated using CRC

T10 SSC-4 addresses tape verification – what about disks?

- SBC-3 Protection Information (PI) is a T10 standard for disk, also called Data Integrity Field (DIF)
- SBC-3 is conceptually similar to the methods used in the tape drive SSC-4 Logical Block Protection to ensure that the data both makes it to and leaves the disk system with integrity and can be properly checked/verified downstream.
- **Unlike for tape, Industry implementation of SBC-3 PI is slow**; for example, IBM has just recently offered its first compliant disk array product.


Red = not verified 🟥
Green = can be verified 🟩

T10 SSC-4 Logical Block Protection

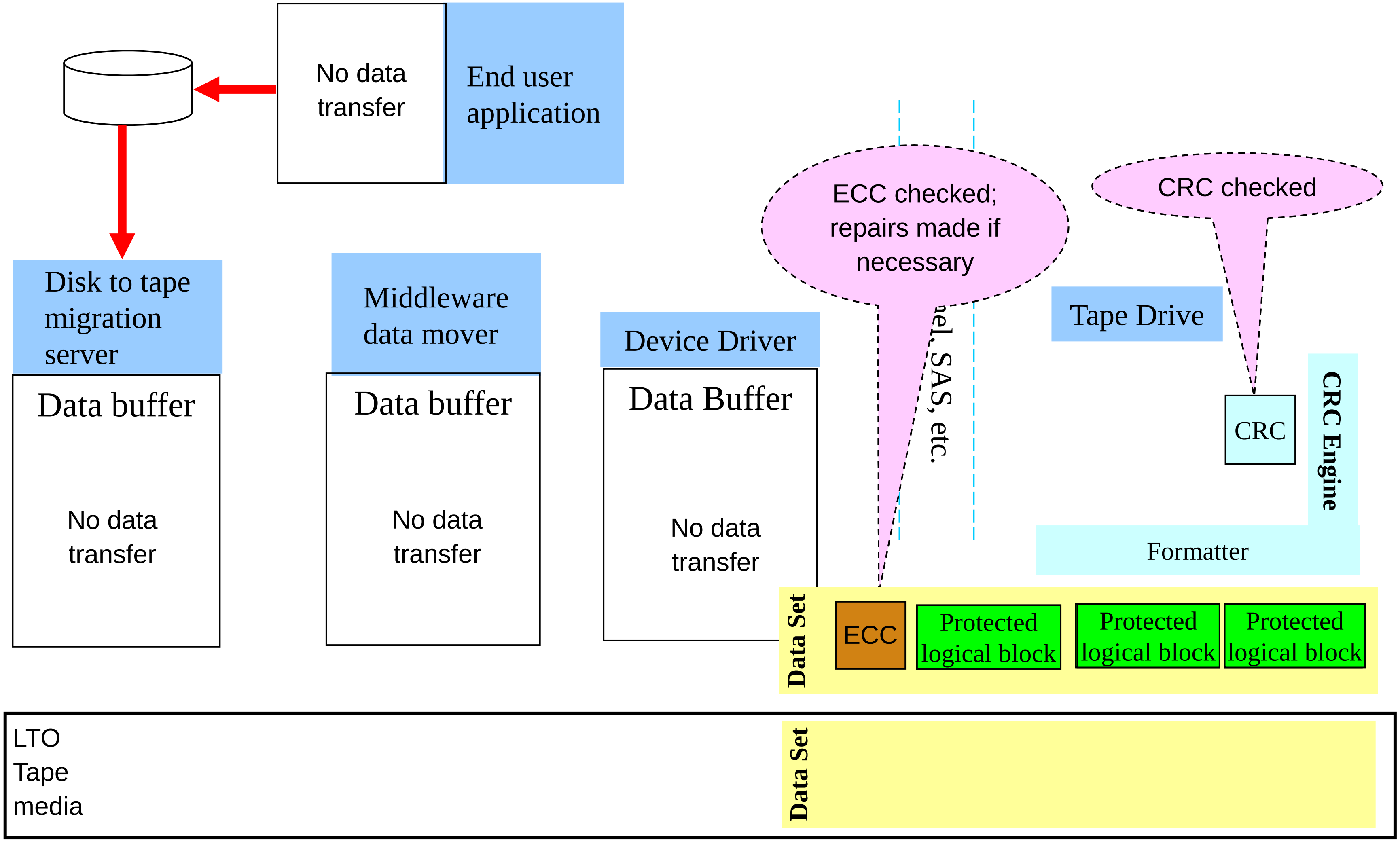
T10 SBC-3 Data Integrity Field

Another Fundamental Tape Archive Questions

How do I know that the bitfile I wrote months or years ago hasn’t rotted and is still correct on tape?

Let the tape drive do the verification of data on tape

(This was an animated slide in the live presentation)



- Validate is similar to Read, but no data is transferred out of the tape drive
- Entire tape can be validated at streaming speed with one function call

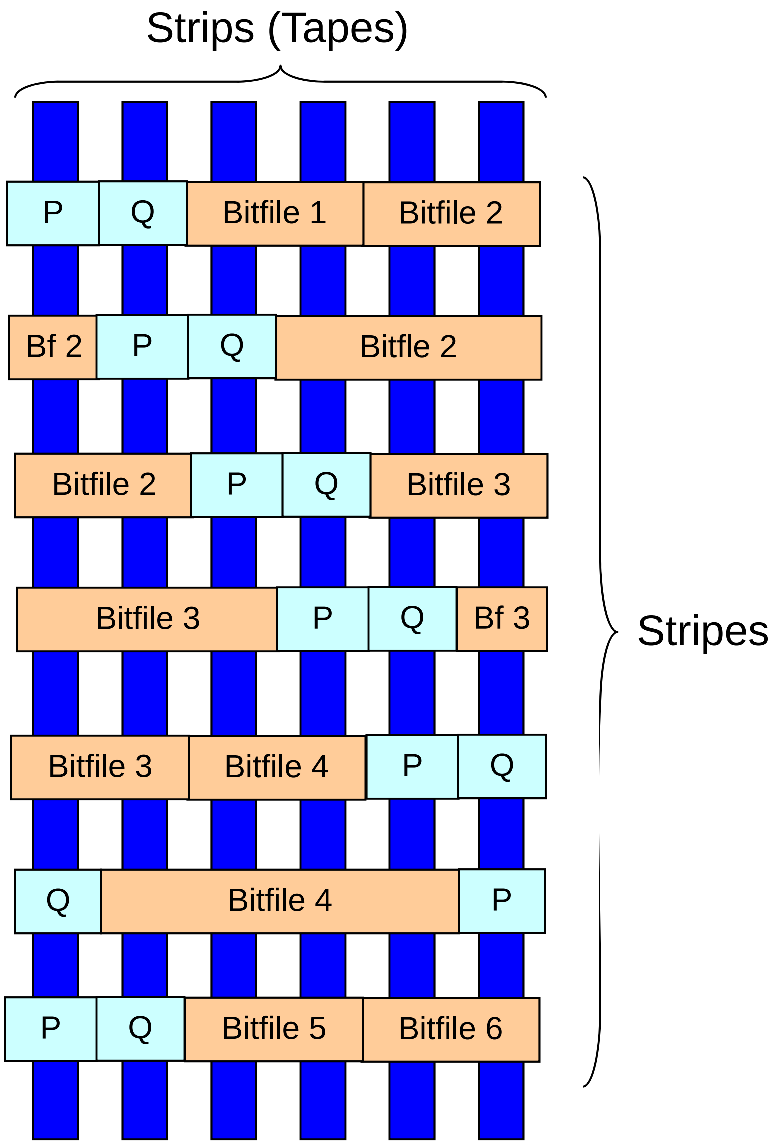
RAIT

- In this last segment we will take a peek into the future of RAIT
  - The Redundant Array of Independent Tapes
  - Analogous to RAID but with significant differences due to the differences in tape and disk architecture
- A technology with outstanding potential for massive long-term archives
  - RAIT can provide higher availability with fewer tape cartridges, as compared to mirroring
- Status:
  - Concept proved in a part-hardware, part-software solution in 2000 - 2001 by (then) StorageTek Inc. under a grant from the U. S. Department of Energy
  - A software RAIT solution for HPSS has been jointly developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign and the High Performance Storage System (HPSS) Collaboration and is now in final test at NCSA

RAIT information based on a paper *Operational concepts and methods for using RAIT in high availability tape archives* by Harry Hulen, Consultant, and Glen Jaquette, IBM Tucson Development, recorded in the on-line proceedings of the IEEE Massive Data Storage Conference 2011

RAIT: Strips and Stripes

- NCSA/HPSS RAIT-6 is shown
- P and Q are the two ECC “Parities”
- Parities are rotated to even out the data on each tape
- Any stripe can be reconstructed if one or two tapes in the stripe report errors
- The entire redundant array can be reconstructed if two entire tapes are bad
- It would take twice as many tapes (12 in this case) to achieve the same level of availability with only mirroring and no RAIT
- To read even one file it is necessary to mount all six tapes;
- Therefore, a large number of tape drives are needed for read performance



Strips (Tapes)

P Q Bitfile 1 Bitfile 2

Bf 2 P Q Bitfile 2

Bitfile 2 P Q Bitfile 3

Bitfile 3 P Q Bf 3

Bitfile 3 Bitfile 4 P Q

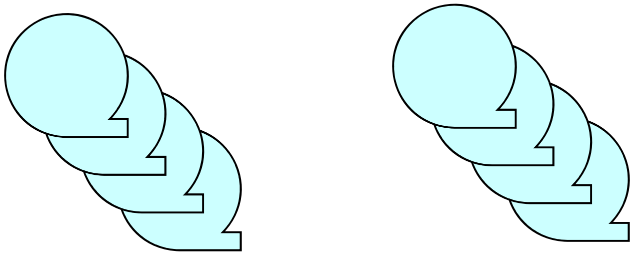
Q Bitfile 4 P

P Q Bitfile 5 Bitfile 6

Stripes

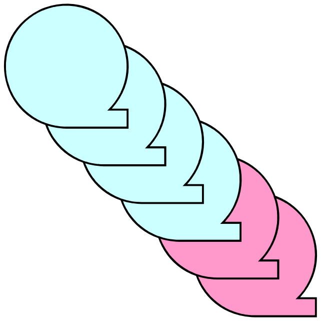
Mirrored vs. RAIT 6 at one location

Mirrored files at one location



- 8 tapes (2x) in this example
- Fast write
- Fast single file read
- Losing two tapes could lose data
- Hidden (that is, unreported) errors not found
- 8 tapes to verify with T10 SSC-4 Verify command

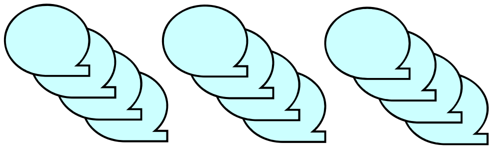
RAIT 6 redundant array at one location



6 tapes (1.5x) in this example

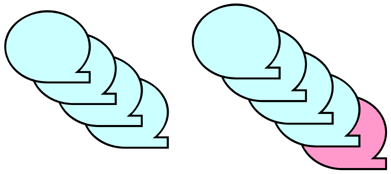
- Fast write
- Slower single file read (or need more tape drives)
- Losing two tapes will not lose data
- Hidden errors can be found and most can be corrected
- 6 tapes to verify with T10 SSC-4 Verify command

Protection against worst-case loss of two tapes



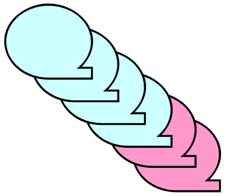
Three single, separate copies

- 3x tapes (3x4=12 in our example)



One single copy and one RAIT copy with one parity (“RAIT-5”)

- 2.25x tapes (4+5=9 in our example)
- Most reads will be from the single copy
- A robust choice where RAIT is available

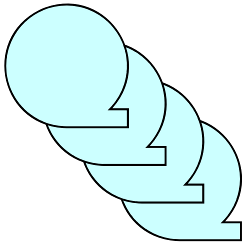


One RAIT copy with two parities (“RAIT-6”)

- 1.5x tapes (6 in our example)
- Can survive two errors in one stripe or loss of two tapes
- Can find and fix one hidden error
- Useful for situations with very low % of reads, or large number of tape drives

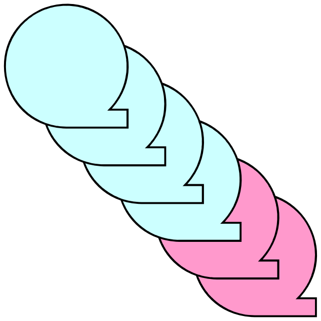
Remote Asymmetric Mirroring

Primary site



- 4 tapes in our example
- No RAIT
- Mount 1 tape to read 1 file
- Optimum for general access

Remote, protected site



- For every 4 tapes at primary site, remote site has 6
- RAIT 6 with two parities
- Mount 6 tapes to read 1 file
- Access by staff only

Remote Asymmetric Mirroring provides 4x availability with 2.5x tapes

Moving to new media – or not

- Media deteriorates over time, however…
  - With a well-maintained environment, tape is good for 20 or more years
  - T10 SSC-4 Verify command enables practical time-interval monitoring (random or 100%)
  - RAIT (when available) will add one or two more levels of RS-type erasure codes and spreads data across multiple tape volumes, thus compensating for even more physical deterioration
- Tape drives become obsolete before media does, however…
  - Fujifilm web site still offers new LTO-1 media, first sold in 2000
  - LTO web site commits to each generation reading tapes two generations back, so LTO-3 drives can read LTO-1 media, LTO-4 can read LTO-2 media, etc.
  - HP web site still offers new LTO-2, LTO-3, and LTO-4 tape drives (LTO-1 drives not needed)
  - IBM has no “sunset” announced for supporting LTO-1 drives (not an official statement)
  - LTO web site says 4 million LTO drives have been sold, meaning recycled parts and refurbished drives will be available years after new drives are no longer available
- New LTO generations hold more data
  - Use less physical space and fewer tape library slots
  - LTO-1 = 100GB, LTO-3 = 400GB, LTO-5 = 1500GB (native)
  - **This is usually the most compelling reason to move to new media**

- Opinion: Moving to new generations of media is usually an economic decision and not a stewardship decision

Summary of strategies for permanent archives

- Always apply a checksum to a valuable bitfile before turning the file over to storage middleware of any kind
- Middleware and hardware should further protect data with logical block-level checksums, using T10 SSC-4 capabilities
- Tape drives like LTO-5 that have T10 SSC-4 Verify greatly reduce cost of periodic tape scans, enabling monitoring of tape media life
- RAIT can provide better protection at less cost than mirroring can provide
- RAIT-6 can both detect and correct a hidden error
- RAIT is fast for writing but unless you have many tape drives, slow for reading
- Remote asymmetric mirroring should be considered for permanent preservation archives
- Economic considerations and not fear of obsolescence should be the driver for migration to new media in a well-maintained environment