SDC 19

September 23-26, 2019
Santa Clara, CA

Developments in LTO Tape Hardware and Software

Takeshi Ishimoto – IBM, SNIA LTFS TWG co-chair

David Pease – LTFS Inventor and TWG Founder

Tape Is NOT Dead: Still Denser, Faster, and Cheaper

SDC 19

EARLY DAYS

IBM 726 Tape System in 1952

2 million digits
in single 8-inch reel

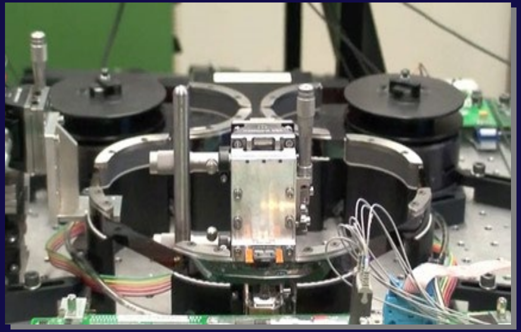
TODAY

LTO Generation 8 in 2017

12TB in palm of your hand
(30TB with 2.5:1 compression)

Sustained I/O Rate @ **360MB/s**

FUTURE



IBM Research Prototype in 2017

Equivalent to **330TB**
in single tape

Photos from IBM

Linear Tape File System (LTFS)

Linear Tape File System (LTFS)

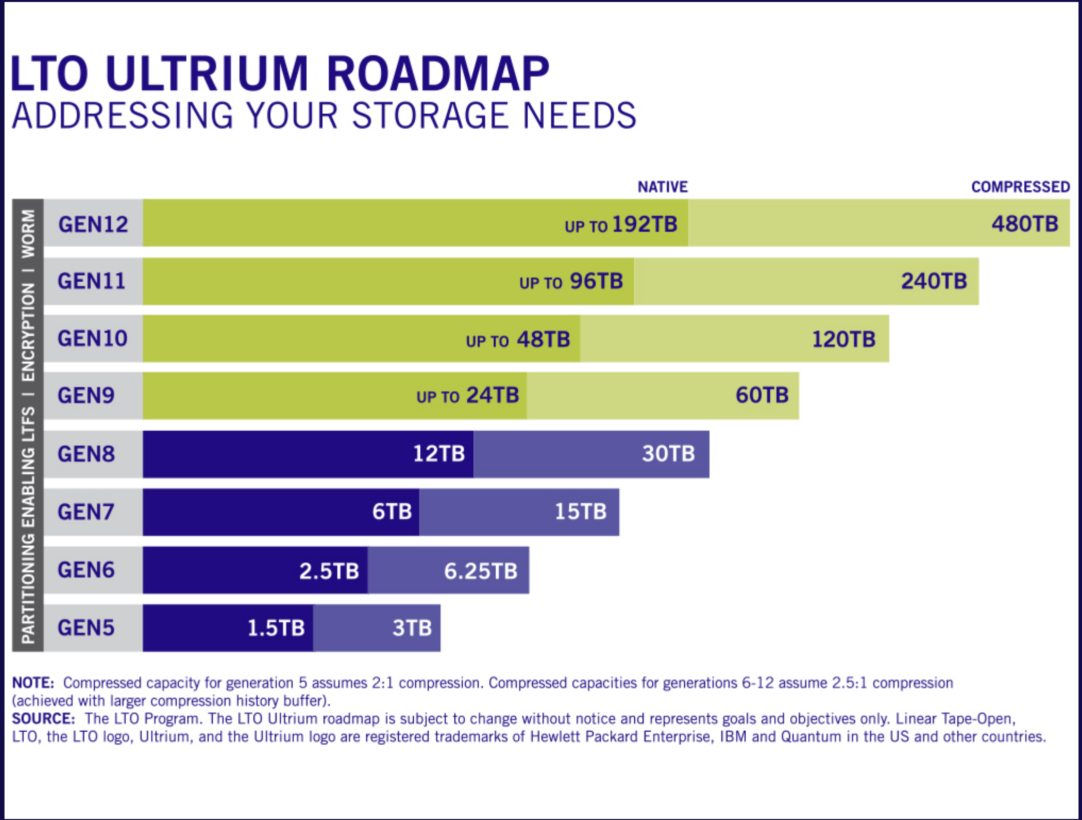
- File System designed for dual-partitioned computer tape storage media
  - Linear Tape-Open (LTO) Tapes
  - IBM TS11xx Enterprise Tapes
  - Oracle T10000 Tapes
- Available from 2010, and widely adopted by the hardware vendors and software companies
- Formed SNIA LTFS Technical Workgroup in 2012 for standardization
  - LTFS Format Specification
  - LTFS Bulk Transfer Specification (collaboration with SNIA Cloud Storage TWG)
- Explore the new use cases of tape with the media portability between locations, systems, and generations

Linear Tape-Open (LTO)

- Hardware technology jointly developed by 3 Technology Provider Companies (TPCs); HPE, IBM, Quantum (was Certance)
  - An “open” format, which allows for compatible offerings from multiple vendors
- Half-inch Wide Magnetic Tape
  - Keeps same physical form-factors
  - Automated tape handling with robotics in tape library
- High-density linear recording (not helical scan method as in the consumer video tapes)
  - Simple tape path with less contact to the surface and edges of tape
- Generation 1 (100GB) technology in 2000, and it has been updated every 2-3 years
  - Supports LTFS since Gen 5 (2010)

Linear Tape-Open: Achievements and Commitment

GEN8 in 2017
GEN1 in 2000



LTO ULTRIUM ROADMAP

ADDRESSING YOUR STORAGE NEEDS

NOTE: Compressed capacity for generation 5 assumes 2:1 compression. Compressed capacities for generations 6-12 assume 2.5:1 compression (achieved with larger compression history buffer).

SOURCE: The LTO Program. The LTO Ultrium roadmap is subject to change without notice and represents goals and objectives only. Linear Tape-Open, LTO, the LTO logo, Ultrium, and the Ultrium logo are registered trademarks of Hewlett Packard Enterprise, IBM and Quantum in the US and other countries.

Source: LTO Consortium LTO Roadmap Chart
https://www.lto.org/technology/what-is-lto-technology/

Linear Tape File System (LTFS)

- File System designed for dual-partitioned computer tape storage media
  - Linear Tape-Open (LTO) Tapes
  - IBM TS11xx Enterprise Tapes
  - Oracle T10000 Tapes
- Available from 2010, and widely adopted by the hardware vendors and software companies
- Formed SNIA LTFS Technical Workgroup in 2012 for standardization
  - LTFS Format Specification
  - LTFS Bulk Transfer Specification (collaboration with SNIA Cloud Storage TWG)
- Explore the new use cases of tape with the media portability between locations, systems, and generations

LTFS Solution Providers (Alphabetical Order, as of Sep 12th, 2019)

1Beyond

Archiware

Arkivum

Atempo

ATTO

Bright Technologies

Codex

Digital Bedrock

Estructure Media Systems

Etere

FOR-A

dternity FUJIFILM

Hewlett Packard Enterprise

IBM

Imagine Products, Inc.

Intellique

MT-C

MagStor

Masstech

mLogic

Odyssey Development

Oracle

Panasonic

proMAX

Qstar Technologies

Qualstar

Quantum

Retrospect

Spectra

StorageDNA

Strong Box Data Solutions

StorageHeaven

Tiger Technology

TOLIS Group, Inc.

TransMedia Dynamics

Unitex

XenData

YoYotta

Source: LTO Consortium
https://www.lto.org/technology/ltfs/ltfs-Implementers/

Linear Tape File System (LTFS)

- File System designed for dual-partitioned tape media
  - Linear Tape-Open (LTO) Tapes
  - IBM TS11xx Enterprise Tapes
  - Oracle T10000 Tapes
- Available from 2010, and widely adopted by the hardware vendors and software companies
- Formed SNIA LTFS Technical Workgroup in 2012 for standardization
  - LTFS Format Specification
  - LTFS Bulk Transfer Specification (collaboration with SNIA Cloud Storage TWG)
- Explore the new use cases of tape with the media portability between locations, systems, and generations

LTFS Format Specification Chronology

- Version 1.0.1, 2.0, 2.1 – Maintained and published by IBM
- Version 2.2 (2013) – First Specification from SNIA. This became ISO/IEC 20919:2016
  - Vendor Name/Version in Extended Attributes (EAs and Cartridge Memory)
  - (Optional) File Spanning and File Permissions/ACL
- Version 2.3 (2015)
  - Character Encoding for Special Characters in File Name and EA Name (deprecated later)
  - Volume Advisory Lock EAs (Software Write Protect, and Media Recovery Indicator)
  - Media Pool Attribute (Tape Pool Name and Pool UUID)
  - (Optional) File Hash
- Version 2.4 (2017)
  - Character Encoding for Special Characters

New

- Version 2.5 (2019) – To be submitted to ISO
  - Incremental Index

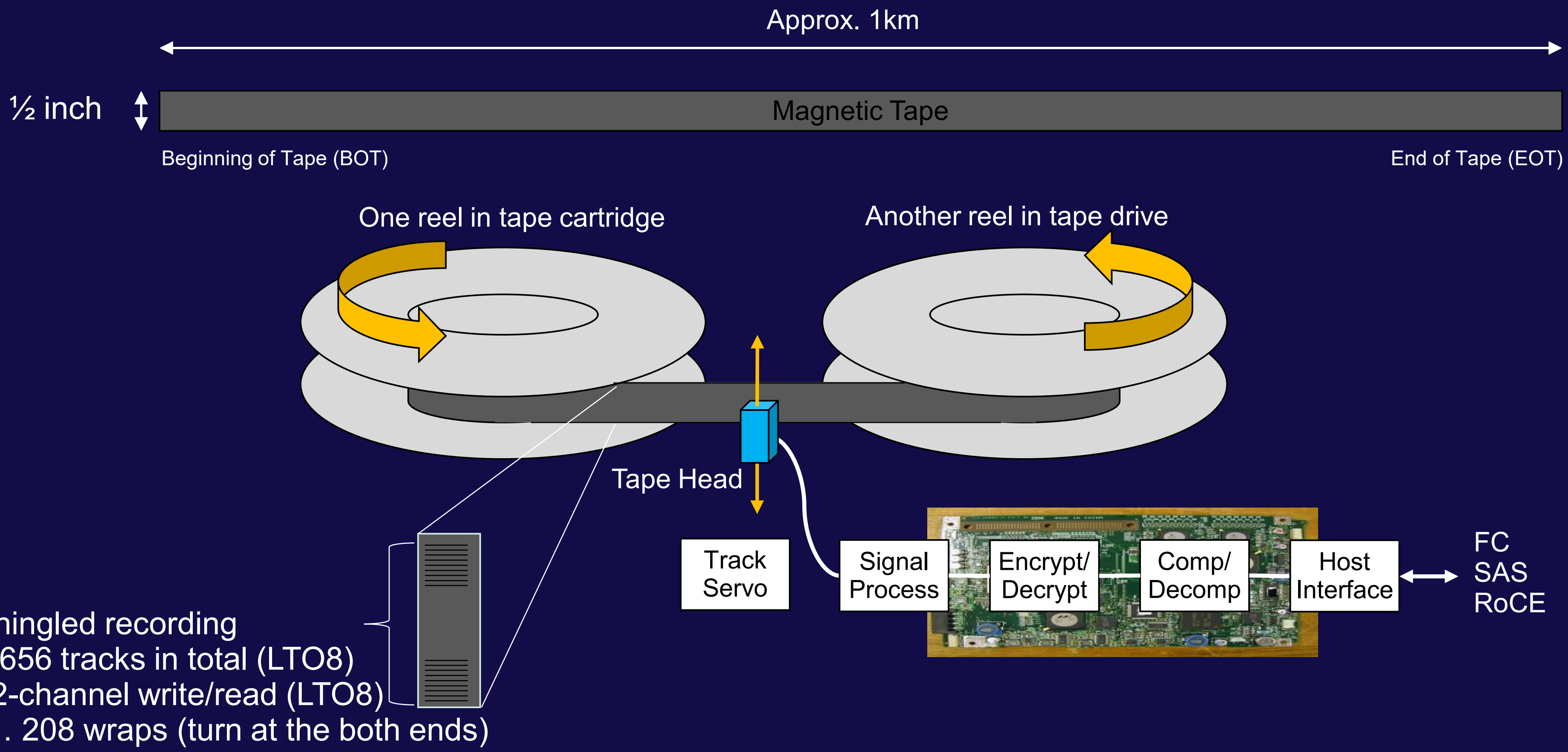
SNIA documents available from https://www.snia.org/tech_activities/standards/curr_standards/ltfs

Why Tape?

Tape Advantages

- Low Cost
  - Ultra-Fast Streaming Performance
  - High Capacity
  - Sustainable, Energy Conscious
  - Scalable
- Resilient
  - Dependable
  - Secure, Offline Storage
- Portable

Inside Tape Drive: How the data is recorded?



Approx. 1km

½ inch

Magnetic Tape

Beginning of Tape (BOT)

End of Tape (EOT)

One reel in tape cartridge

Another reel in tape drive

Tape Head

Track Servo

Signal Process

Encrypt/ Decrypt

Comp/ Decomp

Host Interface

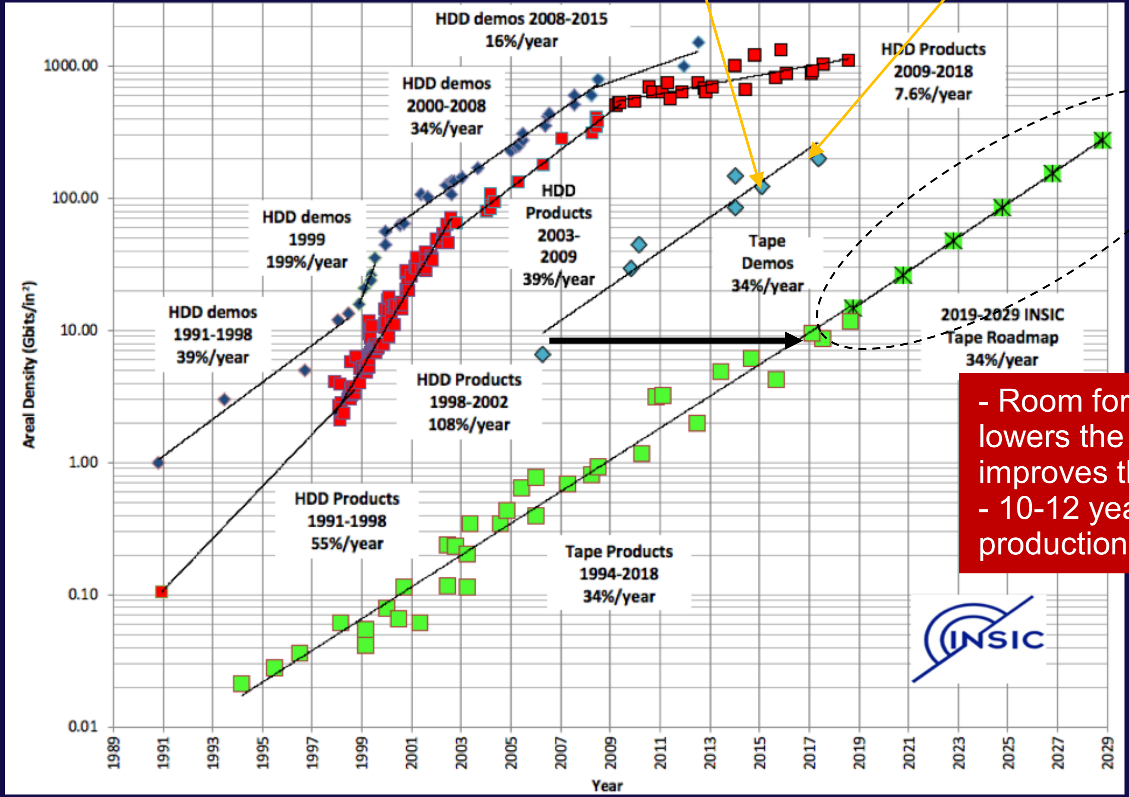
FC
SAS
RoCE

Shingled recording
6,656 tracks in total (LTO8)
32-channel write/read (LTO8)
… 208 wraps (turn at the both ends)

Tape Storage Predicts:

IBM-FujiFilm Demo in 2015
123Gb/in^2

IBM-Sony Demo in 2017
201Gb/in^2



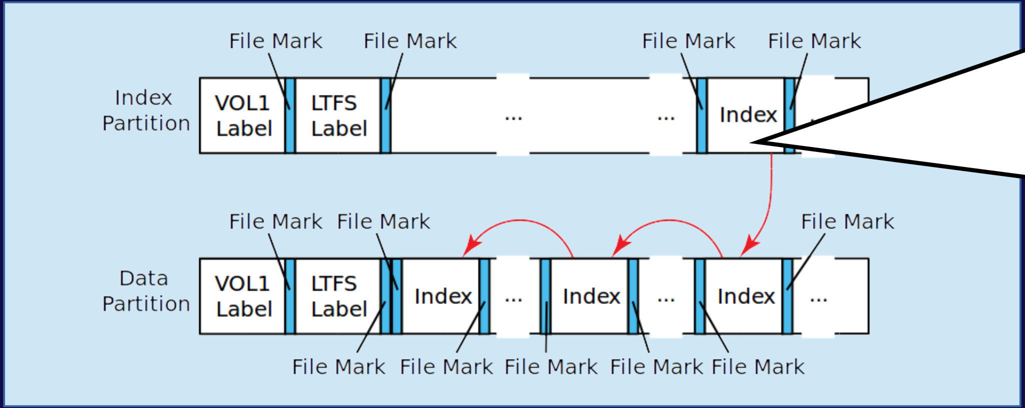
- Room for higher density which lowers the storage cost and improves the performance
- 10-12 years from demo to production

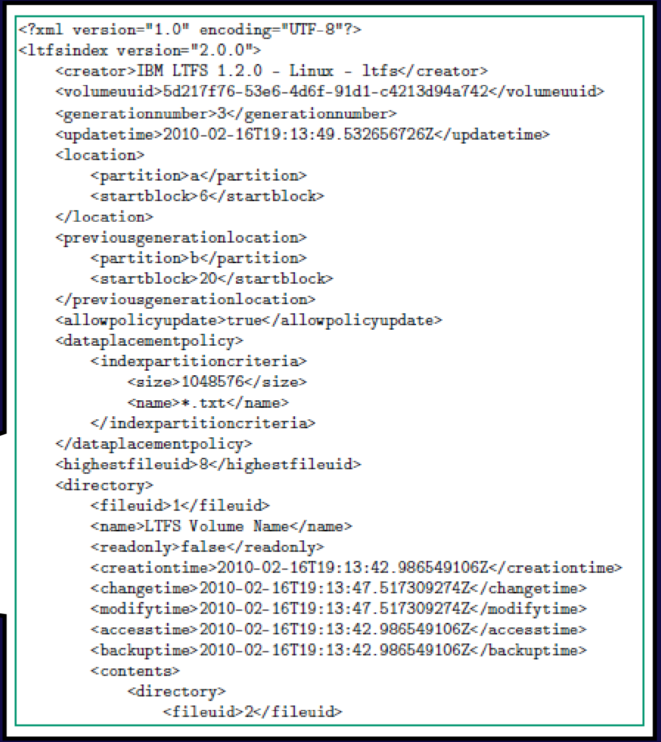
Source: Information Storage Industry Consortium (INSIC) Tape Roadmap Chart
http://www.insic.org/areal-density-chart/

LTFS Tape Format

LTFS Format - Index

- Metadata in XML format on tape
  - Records the file metadata, such as name, timestamps, EAs
  - Associates the file metadata and the data extents on tape
  - Enclosed by a pair of File Marks
  - Not immediately written to the tape at every metadata update
  - Contains the metadata of <u>all</u> the active files, to shorten the mount time and for easier consistency recovery

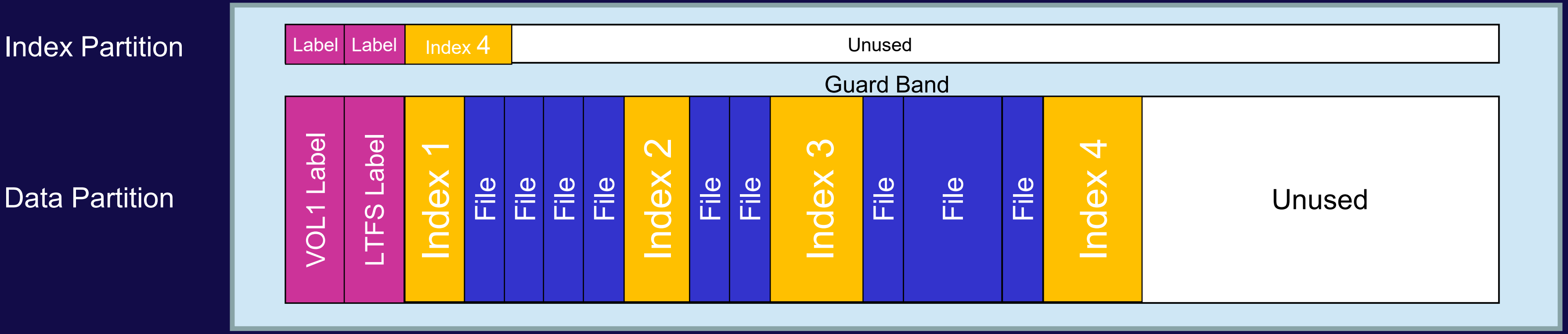




```xml
<?xml version="1.0" encoding="UTF-8"?>
<ltfsindex version="2.0.0">
    <creator>IBM LTFS 1.2.0 - Linux - ltfs</creator>
    <volumeuuid>5d217f76-53e6-4d6f-91d1-c4213d94a742</volumeuuid>
    <generationnumber>3</generationnumber>
    <updatetime>2010-02-16T19:13:49.532656726Z</updatetime>
    <location>
        <partition>a</partition>
        <startblock>6</startblock>
    </location>
    <previousgenerationlocation>
        <partition>b</partition>
        <startblock>20</startblock>
    </previousgenerationlocation>
    <allowpolicyupdate>true</allowpolicyupdate>
    <dataplacementpolicy>
        <indexpartitioncriteria>
            <size>1048576</size>
            <name>*.txt</name>
        </indexpartitioncriteria>
    </dataplacementpolicy>
    <highestfileuid>8</highestfileuid>
    <directory>
        <fileuid>1</fileuid>
        <name>LTFS Volume Name</name>
        <readonly>false</readonly>
        <creationtime>2010-02-16T19:13:42.986549106Z</creationtime>
        <changetime>2010-02-16T19:13:47.517309274Z</changetime>
        <modifytime>2010-02-16T19:13:47.517309274Z</modifytime>
        <accesstime>2010-02-16T19:13:42.986549106Z</accesstime>
        <backuptime>2010-02-16T19:13:42.986549106Z</backuptime>
        <contents>
            <directory>
                <fileuid>2</fileuid>
```

LTFS Format – Tape Partitioning

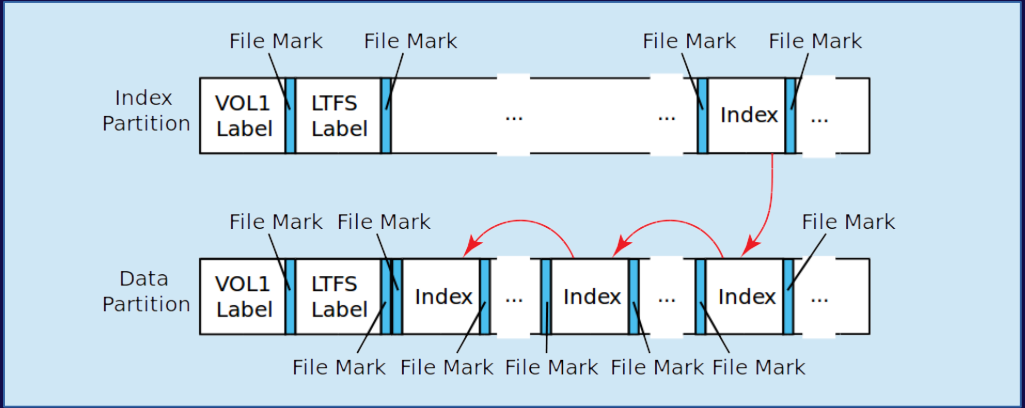
- Asymmetric Dual Partitions on Tape
  - Index Partition (IP): Overwrite Mode
    - Stores the latest file system metadata at the beginning of tape
    - Roughly 3% of total capacity (including the guard wrap between the partitions)
  - Data Partition (DP): Append-only Mode
    - Stores the user data and the historical indices
    - Does not reuse the space occupied by deleted files (until reformat of tape)



LTFS V2.5 Incremental Indexes

LTFS Indexes

- Indexes are periodically written to Data Partition
  - Recovery point (sync point)
  - Rollback point
    - Unbroken chain of backpointers



Problem

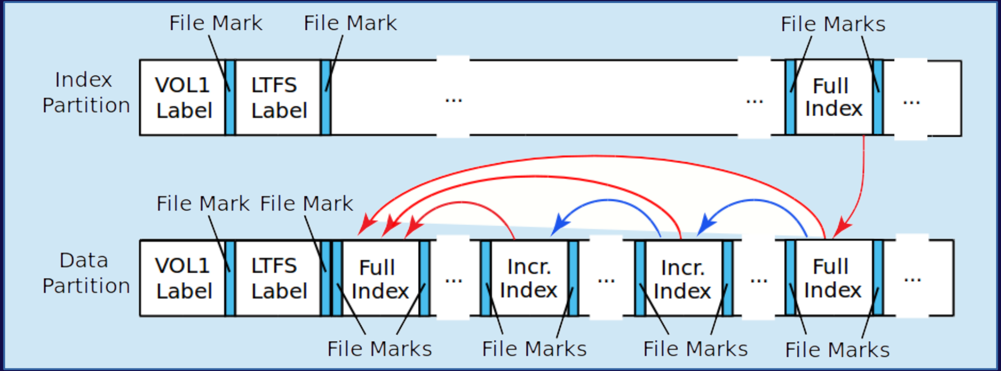
- Tape capacities are growing
  - Increase in number of files, thus in index size
  - Corresponding increase in overhead of recording indexes in Data Partition
    - Time
      - To build index
      - To write index
    - Space
      - Used on tape

Solution: Incremental Indexes

- Record only changes to file system since prior index (full or incremental) was written
- Incremental Index details:
  - Can only appear in Data Partition
    - Index Partition must only contain Full Index(es)
  - Interspersed with Full Indexes in Data Partition
    - Full Index written periodically (every 5-10 indexes?)
    - Full Index must be written at unmount

Backpointers

- We now have two types of index backpointers
  - Traditional full index chain (in red)
    - required for backwards compatibility
  - Incremental chain (in blue)
    - for V2.5 recovery or comprehensive rollback



Backwards Compatibility

- Major concern in design of Incremental Indexes
- Consistent volumes always end with Full Indexes
  - Can be mounted by earlier version
  - However, rollback will miss Incrementals
- Inconsistent volumes are recovered using ltfsck
  - Older versions of ltfsck utility will fail if Incremental Index is encountered during recovery
  - Unlikely occurrence
    - but, can use newer ltfsck version with older LTFS implementation to perform recovery

Thank you! Questions?

Backup: The Gory Details

Gory Details 1

- New XML tags:
  - <ltfsincrementalindex>
  - <previousincrementallocation>
  - <deleted>
    - indicates deleted file or truncated dir tree
- Changed objects require only <name>, <fileuid>, and any changed attributes
- Deleted objects can only have <name>, <deleted> elements

Gory Details 2

- Moved objects are represented as a deleted object plus a new object in new location
- All objects must have parent directory tree entries to indicate their location
  - Only <name> and <contents> tags required
- Only one object with a given name can appear in a directory
  - e.g., rm foo, touch foo
  - only new foo object (with new fileuid) appears in index

Gory Details 3

- Simple and quick to apply Incremental changes to a base index
    - A full simulation was developed to validate design