IBM Research

Tape's Not Dead

Lucas C. Villa Real
lucasvr@br.ibm.com

Tapes... seriously?

Yes!

51% of all archive data is on tape (24 exabytes this year)

Longevity is typically rated at 30 years

High capacity and data streaming rate

SATA Disk vs LTO-4

- Cost ratio/terabyte is ~23:1
- Energy cost ratio is ~290:1
- BER is at least one order of magnitude higher

So.. why aren't they used everywhere?

Inconveniences

No real standard format for data on tape

- TAR?

  A header preceding each file in the archive

  There's no centralized index

Tapes are not self-contained

- External databases

Not considered a good medium for data interchange

The good: industry standards

Linear Tape Open (LTO)

LTO Consortium

- Defines an industry standard for tape drives and media

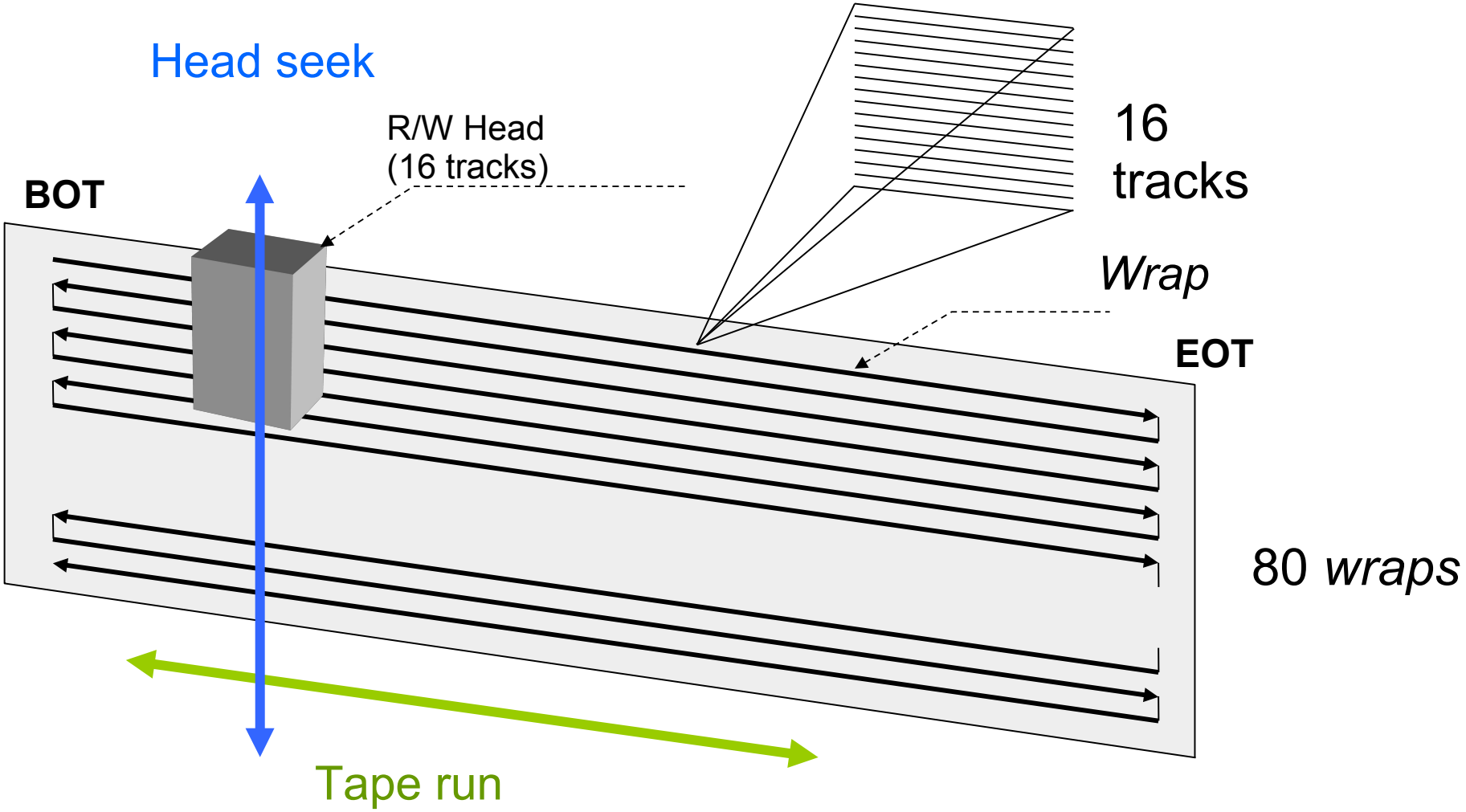
LTO Tapes

- Serpentine recording, shingled writing, block-addressable
- Essentially an append-only media

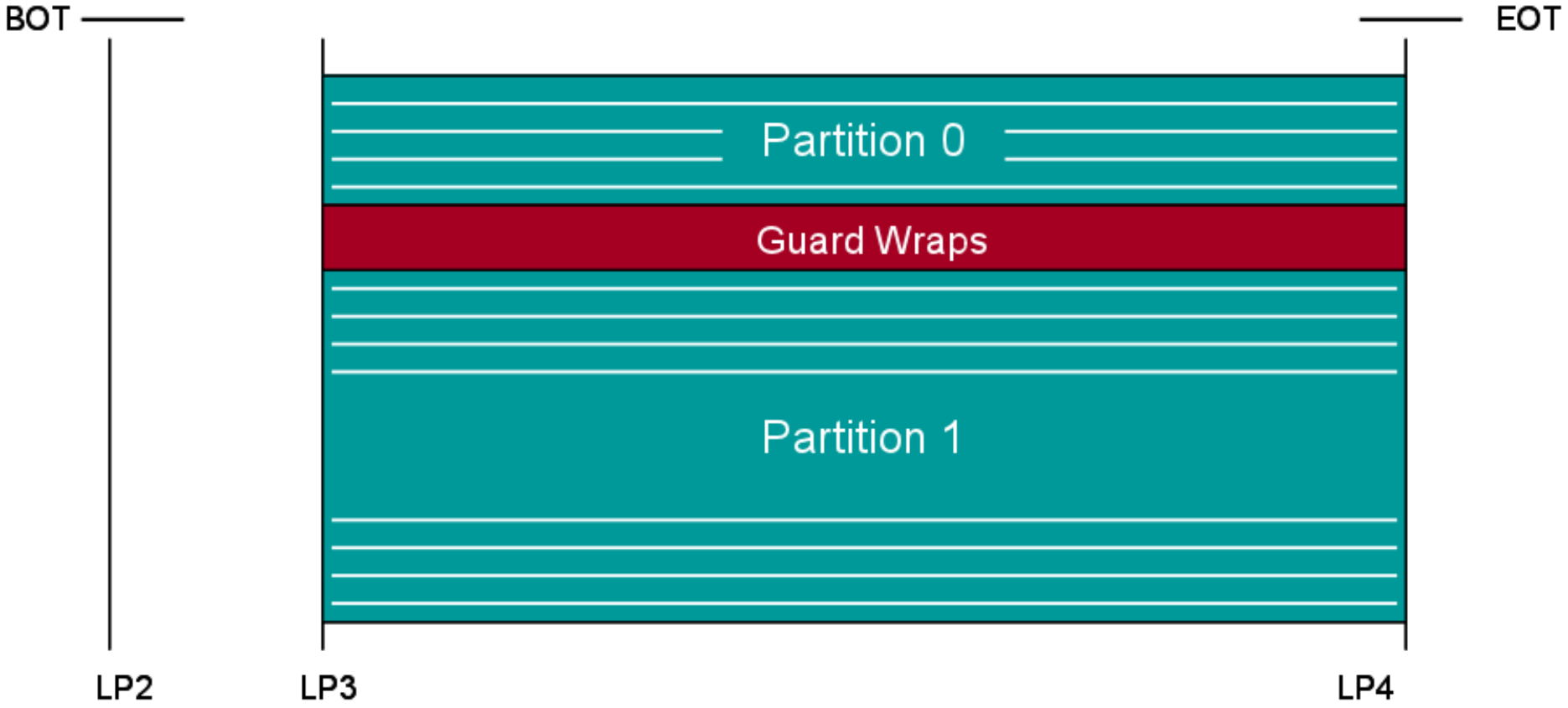
LTO Generation 5 (LTO-5)

- 1.5 TB per cartridge (uncompressed)
- 140 MB/sec streaming data rate
- Dual-partitioning capability

LTO Track Recording



Dual-Partition Tape – A Logical View



Introducing the Linear Tape File System

LTFS: Summary

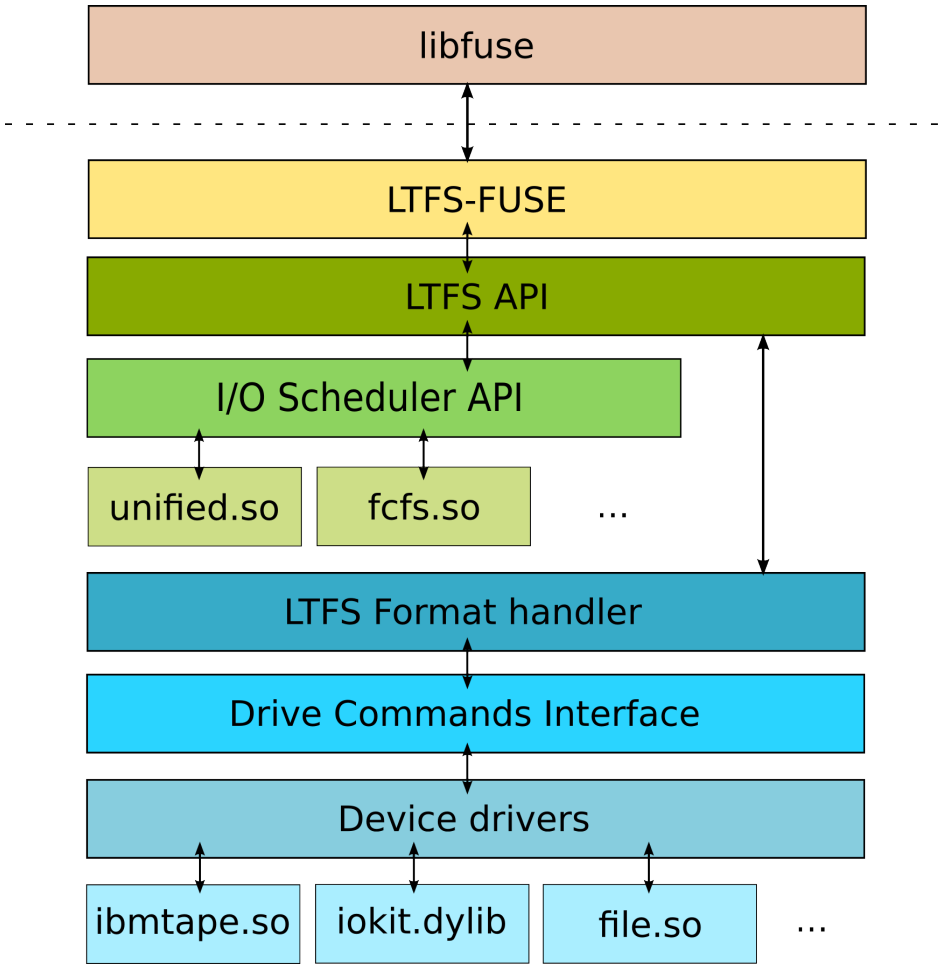
A filesystem for dual-partition linear tapes

- IBM implementation released as LGPL
- Sits on top of FUSE
- Exposed API lets apps use it as a library too
- LTFS specs are open and made publicly available

LTFS makes tape look and work like removable media

http://www-03.ibm.com/systems/storage/tape/ltfs/index.html

LTFS: Software Stack



LTFS: Technical Aspects

Uses LTO-5 dual-partitioning system

- Index partition: 2 wraps, 37.5 GB
- Data partition: remainder

An on-tape structure tracks tape contents

- File names (UTF-8), timestamps, extents, xattrs, etc
- XML Index Schema

Why did you choose XML?

LTFS: XML Index Schema

To keep it simple and multi-platform

To keep it readable by humans

To make it very easy to extend it

To easily export the tape contents to external software

Moreover, tape drives have data compression

- The XML text data is compressed quite nicely

LTFS: Sample XML Index Schema

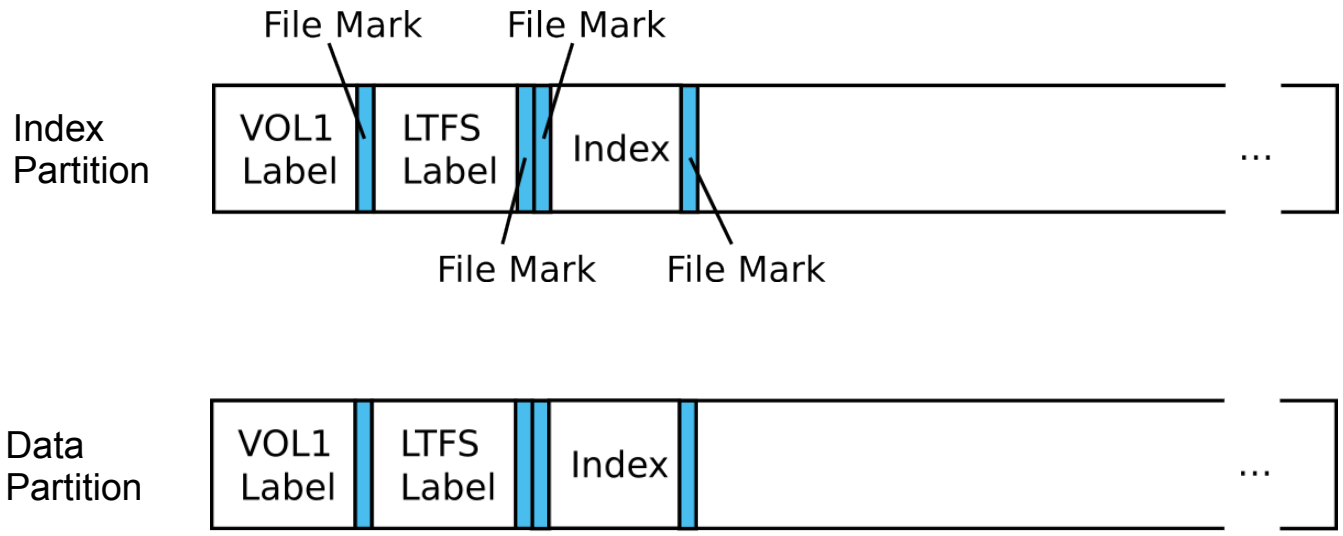
```
<?xml version="1.0" encoding="UTF-8"?>
<ltfsindex version="2.0.0">
    <creator>IBM LTFS 0.20 - Linux - ltfs</creator>
    <volumeuuid>9710d610-5598-442a-8129-48d87824584b</volumeuuid>
    <generationnumber>3</generationnumber>
    <directory>
        <name>LTFS Volume Name</name>
        <creationtime>2010-01-28 19:39:50.715656751 UTC</creationtime>
        <modifytime>2010-01-28 19:39:55.231540960 UTC</modifytime>
        <accesstime>2010-01-28 19:39:50.715656751 UTC</accesstime>
        <contents>
        ...
        </contents>
    </directory>
</ltfsindex>
```

LTFS: Sample XML Index Schema

```
<file>
    <name>binary_file.bin</name>
    <length>10485760</length>
    <extentinfo>
        <extent>
            <partition>b</partition>
            <startblock>8</startblock>
            <byteoffset>0</byteoffset>
            <bytecount>720000</bytecount>
            <fileoffset>0</fileoffset>
        </extent>
    </extentinfo>
    <extendedattributes>
        <xattr>
            <key>ltfs</key>
            <value>rocks!</value>
        </xattr>
    </extendedattributes>
</file>
```

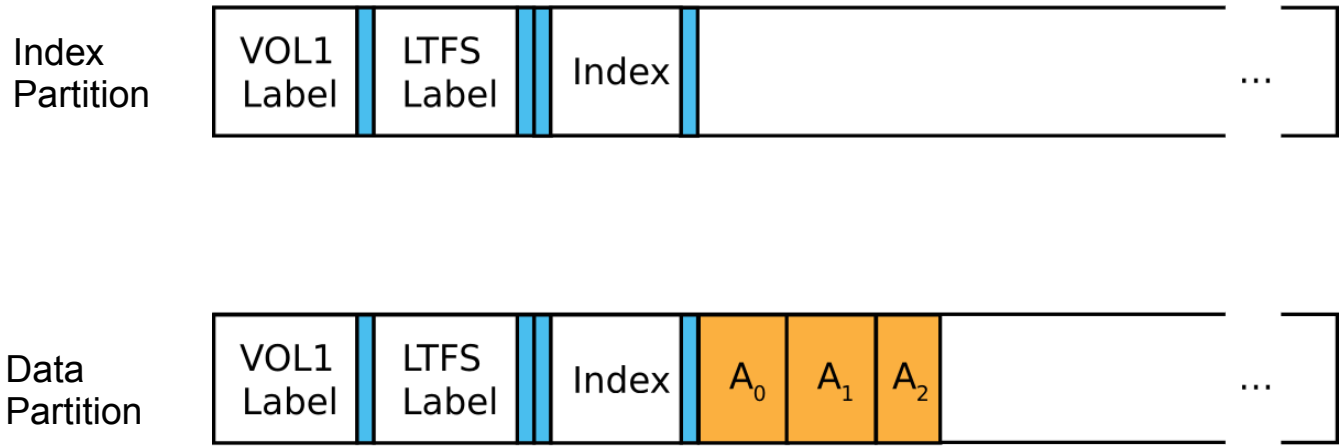
LTFS: Index Arrangement on Tape

Formatted LTFS tape



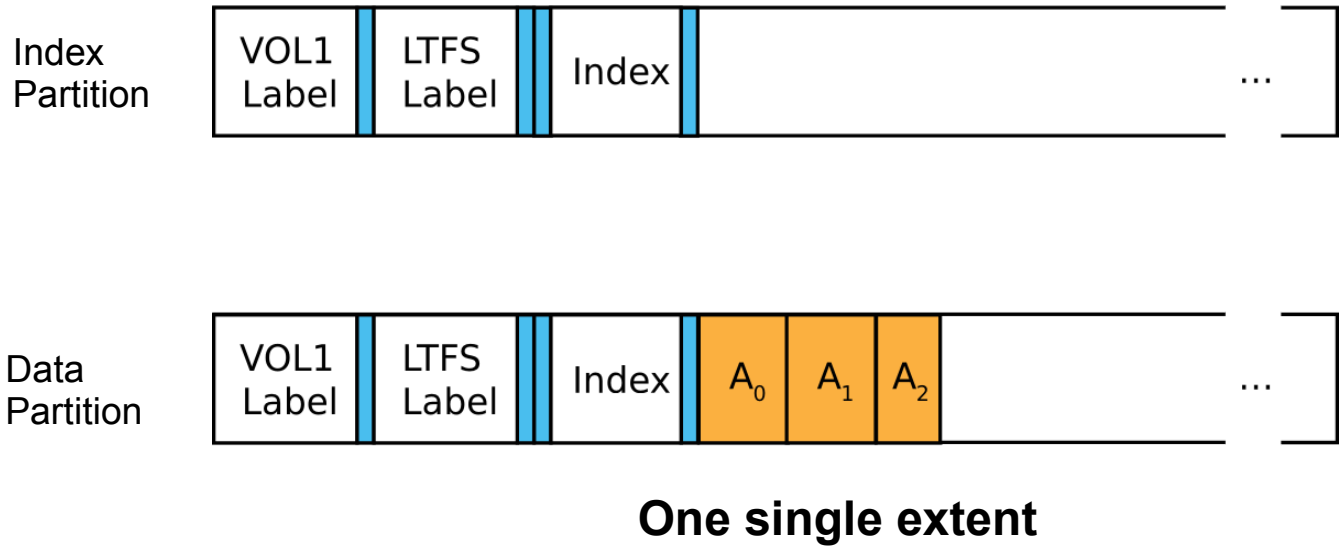
LTFS: Index Arrangement on Tape

Writing file 'A' to Tape



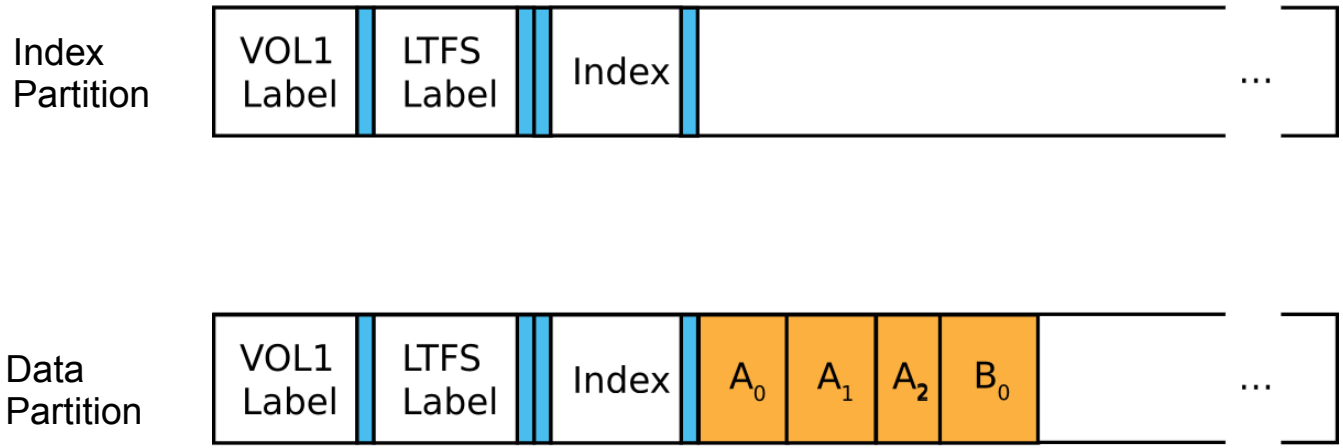
LTFS: Index Arrangement on Tape

Writing file 'A' to Tape



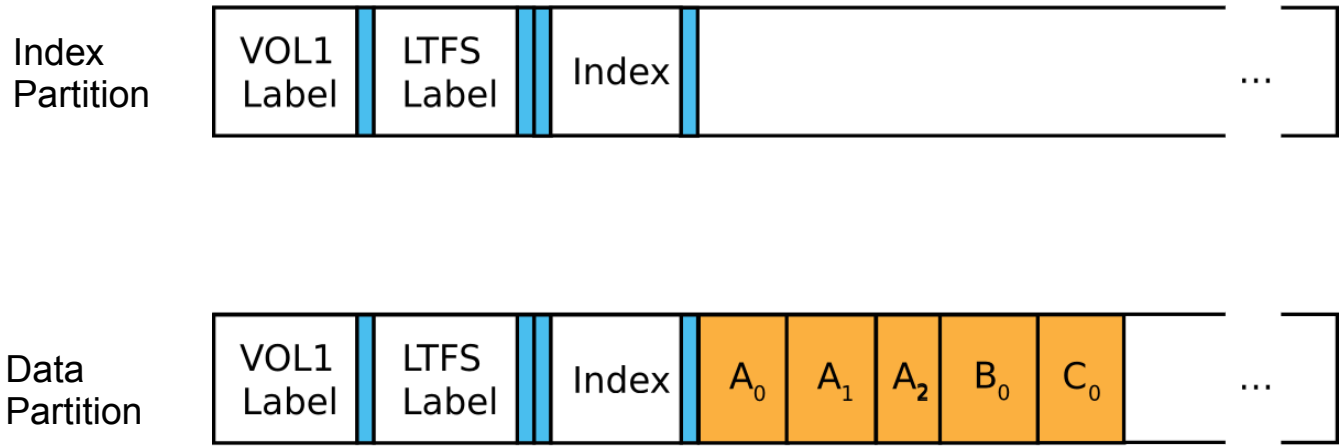
LTFS: Index Arrangement on Tape

Writing file 'B' to Tape



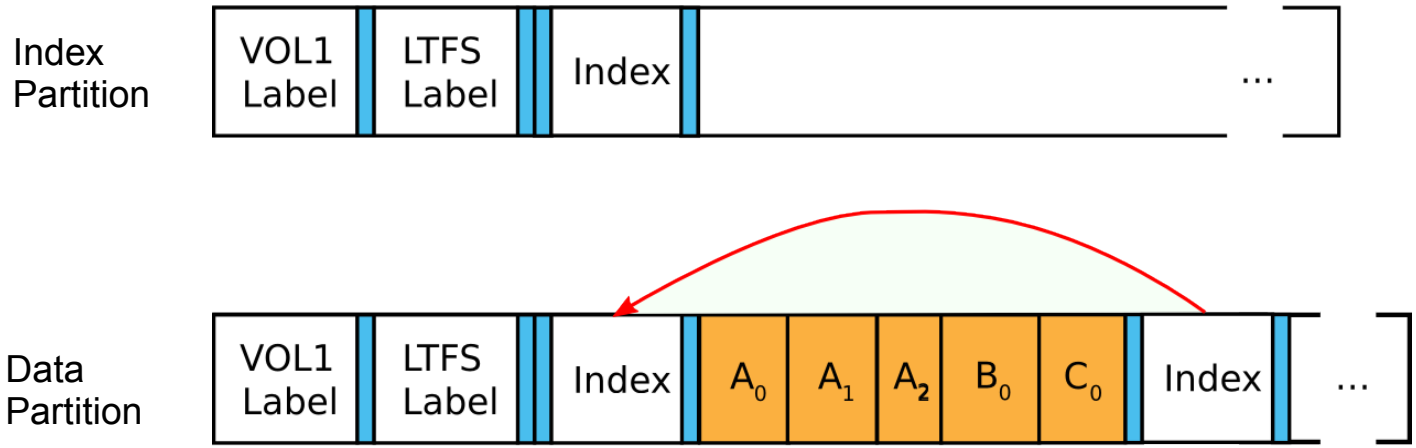
LTFS: Index Arrangement on Tape

Writing file 'C' to Tape



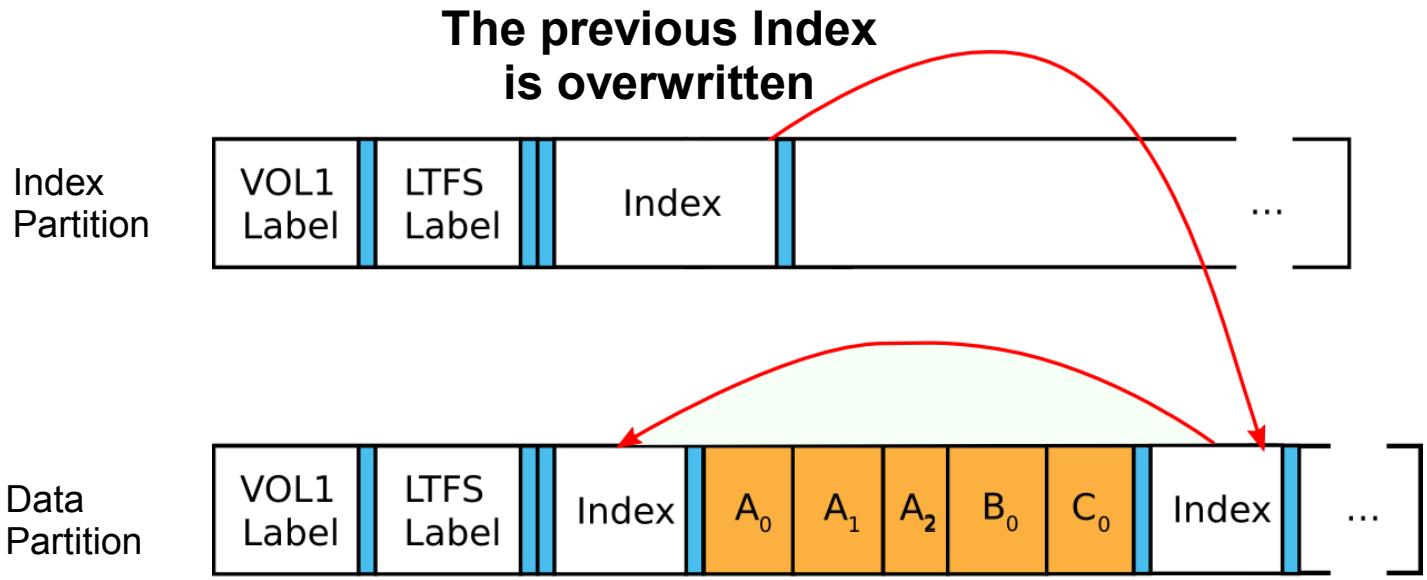
LTFS: Index Arrangement on Tape

Data synchronization

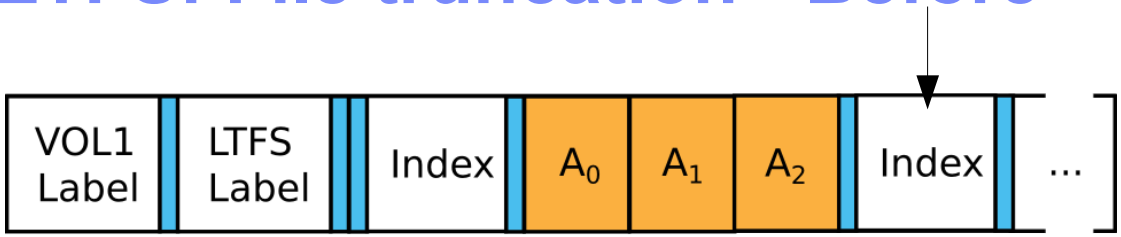


LTFS: Index Arrangement on Tape

Unmounting the tape

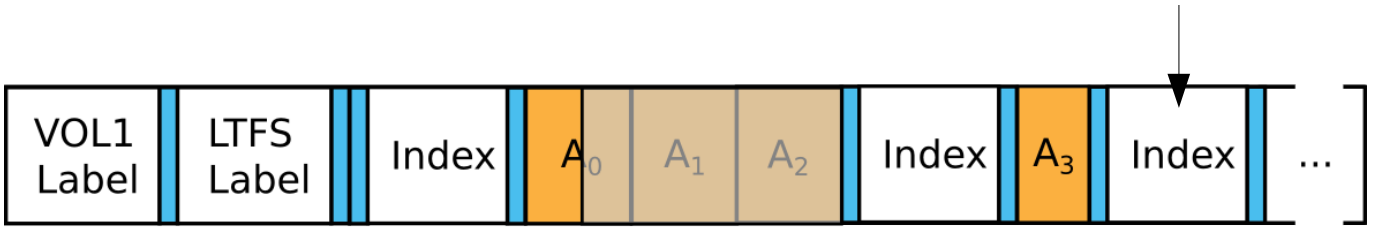


LTFS: File truncation - Before



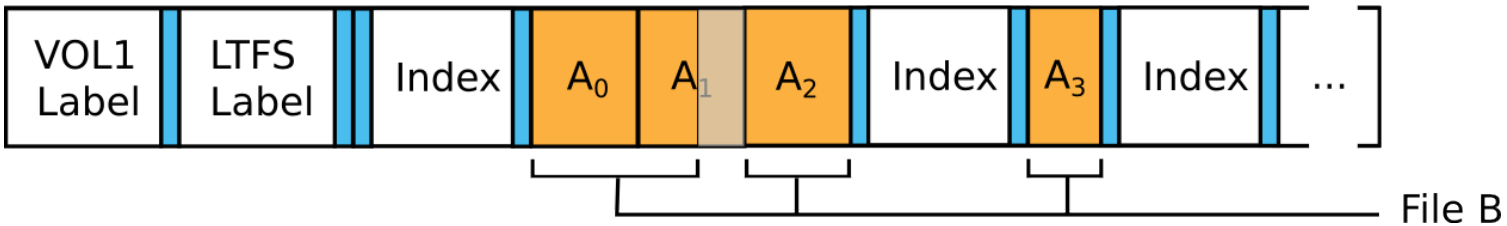

```
<file>
    <name>A</name>
    <length>1572864</length>
    <extent>
        <partition>b</partition>
        <startblock>18</startblock>
        <byteoffset>0</byteoffset>
        <bytecount>1572864</bytecount>
        <fileoffset>0</fileoffset>
    </extent>
    ...
```

LTFS: File truncation - After




```
<file>
    <name>A</name>
    <length>263175</length>
    <extent>
        <partition>b</partition>
        <startblock>18</startblock>
        <byteoffset>0</byteoffset>
        <bytecount>262144</bytecount>
        <fileoffset>0</fileoffset>
    </extent>
    <extent>
        <partition>b</partition>
        <startblock>22</startblock>
        <byteoffset>0</byteoffset>
        <bytecount>1031</bytecount>
        <fileoffset>262144</fileoffset>
    </extent>
    ...
```

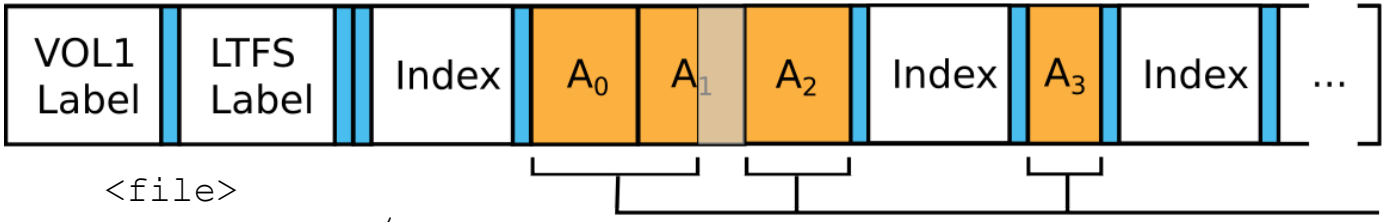
LTFS: Shared blocks



Blocks from other files are reused

- No need for reference counts (append-only)

LTFS: Shared blocks




```
<file>
    <name>B</name>
    <length>1311751</length>
    <extent>
        <partition>b</partition>
        <startblock>18</startblock><byteoffset>0</byteoffset>
        <bytecount>786432</bytecount>
        <fileoffset>0</fileoffset>
    </extent>
    <extent>
        <partition>b</partition>
        <startblock>20</startblock><byteoffset>0</byteoffset>
        <bytecount>524288</bytecount>
        <fileoffset>786432</fileoffset>
    </extent>
    <extent>
        <partition>b</partition>
        <startblock>22</startblock><byteoffset>0</byteoffset>
        <bytecount>1031</bytecount>
        <fileoffset>1310720</fileoffset>
    </extent>
```

LTFS: Sparse Files

Useful to “allocate space” for files

- E.g.: virtual machine disk images

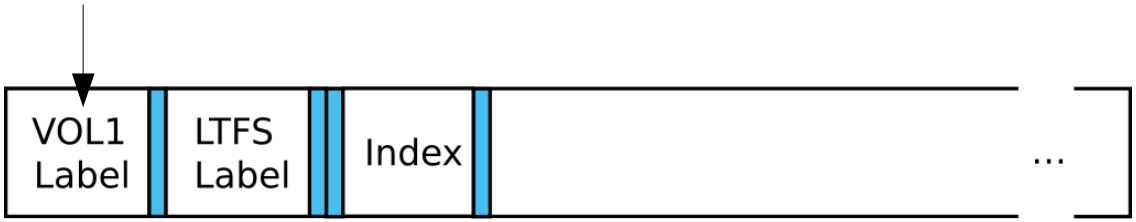
Allows files to have non-zero size but no extent list

Quite simple

- Bytes not encoded in the extent list are treated as zero
- Can appear at the begin or end of an extent

Partition Labels

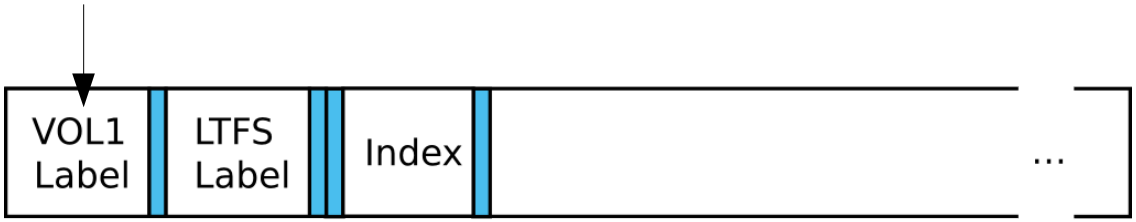
LTFS: Labels



Volume Label

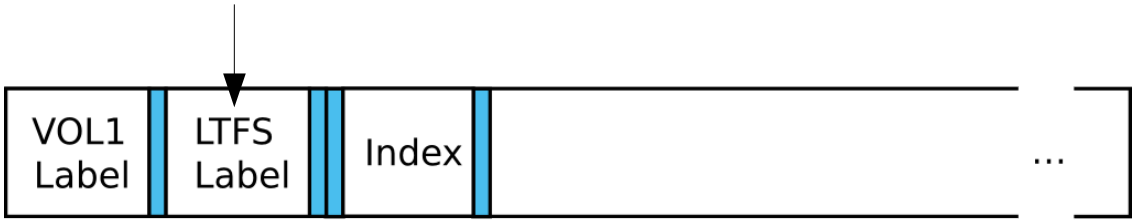
- Stored according to the ANSI Standard X 3.27
- 80 bytes long
- Kept around for backwards compatibility with legacy apps

LTFS: Labels



Offset	Length	Name	Value	Notes
0	3	label identifier	'VOL'	
3	1	label number	'1'	
4	6	volume identifier	<volume serial number>	Typically matches the physical cartridge label.
10	1	volume accessibility	'L'	Accessibility limited to conformance to LTFS standard.
11	13	reserved	all spaces	
24	13	implementation identifier	'LTFS'	Value is left-aligned and padded with spaces to length.
37	14	owner identifier	right pad with spaces	Any printable characters. Typically reflects some user specified content oriented identification.
51	28	reserved	all spaces	
79	1	label standard version	'4'	

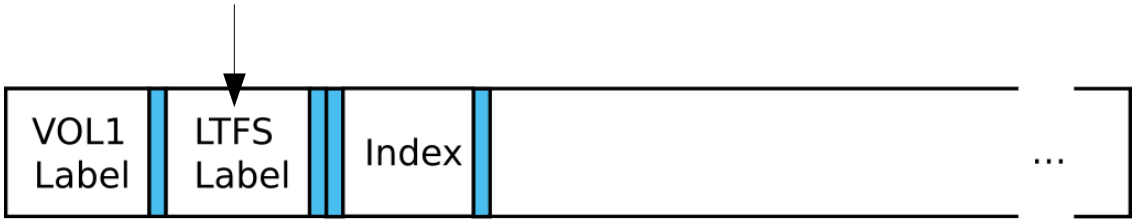
LTFS: Labels



LTFS Label

- Describes the LTFS Volume and the Partition on which the label is recorded
- Recorded as XML

LTFS: Labels




```
<?xml version="1.0" encoding="UTF-8"?>
<ltfslabel version="2.0.0">
    <creator>IBM LTFS 1.2.0 - Linux - mkltfs</creator>
    <formattime>2012-07-01T18:35:47.866846222Z</formattime>
    <volumeuuid>30a91a08-daae-48d1-ae75-69804e61d2ea</volumeuuid>
    <partitions>
        <index>a</index>
        <data>b</data>
    </partitions>
    <location>
        <partition>b</partition>
    </location>
    <blocksize>524288</blocksize>
    <compression>true</compression>
</ltfslabel>
```

LTFS: Data Placement Policies

Allow data extents to be saved on the Index Partition

- Quick access to important files

```
<?xml version="1.0" encoding="UTF-8"?>
<ltfsindex version="2.0.0">
    ...
    <dataplacementpolicy>
        <indexpartitioncriteria>
            <size>1048576</size>
            <name>*.txt</name>
            <name>*.bin</name>
        </indexpartitioncriteria>
    </dataplacementpolicy>
    <allowpolicyupdate>yes</allowpolicyupdate>
    ...
</ltfsindex>
```

LTFS: Extended Attributes

First-class objects

- Stored in the XML Index for quick access
- Not limited in size

Binary data encoded as base64

Reserved xattrs live in the “ltfs.” namespace

- Not returned by listxattr()

LTFS: Virtual Extended Attributes

Object metadata

- ltfs.{access,backup,change,modify,create}Time
- ltfs.{fileUID,partition,startBlock}

Volume metadata

- ltfs.index{Creator,Generation,Location,Previous,Time}
- ltfs.volume{Blocksize,Compression,FormatTime,Name,Serial,UUID}
- ltfs.partitionMap
- ltfs.policy{AllowUpdate,Exists,MaxFileSize}
- ltfs.{commitMessage,sync}

LTFS: Medium Auxiliary Memory

May store a Volume Coherency Information attribute

- Index generation number for the current Index
- On-media location of the current Index
- Volume Change Reference (VCR)

Used to:

- Determine whether a partition is complete
- Verify volume consistency without having to read the Index from both partitions

And the most important thing...

LTFS is the first filesystem to ever receive an Emmy Award!

Questions? :-)

IBM Research

Tape's Not Dead

Lucas C. Villa Real
lucasvr@br.ibm.com

LTFS Performance

File Size	Write (MB/s)	Read (MB/s)	Seek (secs)
1 GB	132.9	132.2	37.2
1 MB	98.2	133.0	37.6

Raw tape throughput is of ~133MB/sec

System:

- LTO-5 Full-Height drive connected to a 4 GB/s FC
- 2 quad-core Intel Xeon Core 2 @ 2.66 GHz
- CentOS 5.4 (Linux 2.6.18-164.el5, lin_tape-1.65.0, FUSE 2.7.4-8)