An Oracle White Paper
April 2014

How to Capture Wasted Cartridge Space in z/OS Implementations Using Tape Tiering Accelerator

Introduction

Tapes hold hundreds to thousands of gigabytes of data. Typically, the data is “stacked” on the media as different data sets, and each data set has a different expiration. When the expiration occurs, there is wasted space on the tape. Over time the wasted space becomes so large that customers must reclaim the tape, which can consume many hours. In order to save these customers time and money, Oracle created a method to capture the wasted space—the Tape Tiering Accelerator feature of Oracle's StorageTek T10000C tape drives. The tape drive creates a hard disk-like format so that more of the tape can be used. Tape tiering or partitioning (nonlinked) has existed for many years; however, many current tape drives don't implement them because developers didn't think it was useful. This implementation goes beyond tiers and creates automatically linked partitions so the data can span tiers or partitions and be noncontiguous on tape. Furthermore, the tape drive handles much of this formatting without host intervention.

The StorageTek T10000C tape drive Tape Tiering Accelerator specifications:

- Cartridge requirement: Long Fuji
- Number of partitions: 480
- Partition size: 9 GB

This paper describes how z/OS users can format a Tape Tiering Accelerator cartridge and access it, but it does not detail how to manage the partitions or the data in them. The StorageTek T10000C tape drive does not store read/write access status of partitions, but it does store the logical volume start and linkage status in the media information record (MIR).

When the term *automatically linked partition* (ALP) is mentioned in the paper, it is synonymous with Tape Tiering Accelerator.

.

Tape Tiering Accelerator Programming Methods for z/OS

The StorageTek T10000C tape drive introduces a new set of command codes and extensions to other command formats on z/OS. To implement these command codes, the use of EXCPVR is required; EXCPVR is documented in the following IBM publications:

- SC26-7400 DFSMSdfp Advanced Services
- SC26-7408 Z/OS Macro Instructions for Data Sets
- SA22-7605 MVS Programming: Assembler Services Guide

In Figure 1 the Tape Tiering Accelerator tape contains three files: A, B, and C. File A occupies a single partition (0). File B occupies three partitions that are linked consecutively (1, 2, 3). File C occupies eight nonconsecutively linked partitions (4, 13, 14, 15, 16, 17, 18, 19) indicating that partitions (5, 6, 7, 8, 9, 10, 11, 12) were not writable at the time file C was written to the tape.

Files	*A*	*B*			C	→								C						
TTAs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 1. Example Tape Tiering Accelerator partition allocation

The command procedures and their command formats are summarized below.

Set Host Type

Command X'77', 20 byte parameter data

Use the PSF set host to identify the drive as a StorageTek drive. The following parameter data byte must be set:

- Byte 0 = X'40'
- Byte 2 = X'08'

The set host type command must be issued before any other Tape Tiering Accelerator commands will be recognized.

Detect Current Tape Tiering Accelerator State

Command X'77', suborder X'18', 12 byte parameter data (prep for RSSD)

Command x'3E', 40 byte buffer to receive RSSD response, chained to prep for RSSD

The RSSD command is used to return the Tape Tiering Accelerator state of the StorageTek T10000C tape drive and loaded tape. The detect current TTA state command indicates the following about the tape drive: Tape Tiering Accelerator capable, Tape Tiering Accelerator enabled, Tape Tiering Accelerator tape loaded and current partition number at which the tape is positioned. To issue this command, the correct host type command must have been issued and the RSSD channel command word (CCW) must be chained to a PSF command with the prep RSSD parameter data specified. The PSF command must be the first CCW in the chain.

Prep for RSSD parameter data supplied:

X'18000000000000300000000000'

The RSSD command returns the following Tape Tiering Accelerator-related status fields:

- Byte 14—Cartridge status, X'02' indicates and Tape Tiering Accelerator-formatted cartridge
- Byte 15—Drive status, X'02' indicates the drive is in Tape Tiering Accelerator mode
- Byte 16–17—2 byte current partition position
- Byte 24—Drive capability, X'02' indicates the drive is Tape Tiering Accelerator capable

Map Tape Tiering Accelerator Partitions

Command X'77', suborder X66', 132 byte parameter data to enable partitions for write

Perform subsystem function (PSF) suborder x'66' is used to define the partition map which will enable individual partitions for read-only or write authority. The map size is 128 bytes with a 4-byte header for a total parameter length of 132 bytes. The parameter data format is as follows:

- Bytes 0–3—X'66008000'
- Bytes 4–126—Partition authorization bits

The StorageTek T10000C tape drive supports 480 partitions represented as 0–479. Each bit position starting with the high order bit of byte 4 through the low order bit of byte 64 represents the corresponding partition number's state. A B'1' represents write authorization and a B'0' represents read-only authorization. The remaining bits are reserved for future use, setting those bits to B'1' will result in a command reject status, parameter exception. For example:

X'6600800080'—Partition 1 is enabled for write; the remaining bytes are set to zero.

X'66008000C0'—Partitions 1 and 2 are enabled for write; the remaining bytes are set to zero.

X'66008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF' – Every partition is enabled for write, the remaining bytes must be set to zero; otherwise, command reject status is returned.

X'66008000'—Partitions are enabled for read only; all partition bytes are set to zero. This is the default state when a Tape Tiering Accelerator tape is loaded.

When issuing the TTA partition map command the tape must be positioned at beginning of tape (BOT); otherwise; command reject status is returned.

Locate Tape Tiering Accelerator Partition

Command X'BF', 2 byte parameter data

The locate TTA command moves the tape into position on the target tape drive so that the application can write or read on a specific partition on tape. The tape drive positions the tape to the nearside of the first block of the partition.

This command requires a data length of 2 with the parameter data containing a 2-byte binary partition number (0–479). A Tape Tiering Accelerator-formatted tape must be loaded and ready; if any of these conditions are not true, a command reject (ERPA_code 27) will result. Initial status of channel end (CE) will be presented and device end (DE) will be presented upon the command's completion.

The 2-byte parameter data will be boundary checked and a command reject (CR) will be issued if the value is outside the boundary supported by the loaded tape.

Once the tape is located to a partition, then normal commands such as write, read, tape mark, sync, read RBID, and so on, are used to process the blocks within the partition. The RSSD command is used to return the current partition; the current partition is relative to blocks actually written on the physical tape. A SYNC command will flush the cache to tape in order to obtain an accurate Tape Tiering Accelerator partition number.

Report Tape Tiering Accelerator Linkage

Command X’B2’, return 2048 bytes of information

The report TTA linkage command returns the current Tape Tiering Accelerator mode tape linkage (mapping). The command returns 2,048 bytes of information. The information represents how the partitions are forward linked together. The information represents each of the partitions (2 bytes of data for each partition). The 2 bytes of data are the next partition number in the link or one of four special values. The four special values are:

X’FFFF’—Partition not linked or no more Tape Tiering Accelerator partitions linked to this partition.

X’FFFE’—Partition not used or the Tape Tiering Accelerator partition isn't used in the current tape format.

X’FFFD’—Partition link unknown or the drive doesn't know if the Tape Tiering Accelerator partition is linked to something else. This only happens when a tape is loaded and the drive is power cycled or receives an unexpected reset condition.

X’FFFC’—Partition blank or the Tape Tiering Accelerator partition hasn't been written since the last time the tape was converted to a Tape Tiering Accelerator tape.

Create Tape Tiering Accelerator Tape

This section describes the steps required to create a Tape Tiering Accelerator tape from a standard tape. After the required steps are completed, the tape is formatted as a Tape Tiering Accelerator tape with a logical volume starting at Tape Tiering Accelerator partition 0. The drive will recognize the tape as a Tape Tiering Accelerator tape when it is loaded.

A Tape Tiering Accelerator-formatted tape cannot be processed as a standard tape once formatted as a Tape Tiering Accelerator tape; z/OS will detect a permanent I/O error during open. Additionally, when a Tape Tiering Accelerator tape is created from a standard tape, it is necessary to bypass label processing by specifying BLP on the DD label parameter or by using other means.

Once the drive is loaded and ready, and the tape is at beginning of tape (BOT), the following sequence of commands can be performed:

1. Set host type described in section 2.1 to allow the Tape Tiering Accelerator commands to be recognized.
2. Detect current TTA state described in section 2.2 to verify that the drive is Tape Tiering Accelerator capable and a standard (not Tape Tiering Accelerator) is loaded and the TTA mode bit should be zero.
3. Map TTA partitions described in section 2.3; partition 0 must be set to write.
4. Write at least one block and nonbuffered sync or one nonbuffered tape mark. Any number of blocks can be written (such as labels) followed by a nonbuffered tape mark.
5. Optionally, detect current TTA state may be performed and the TTA mode bit will be set, but the Tape Tiering Accelerator cartridge bit will not be set until the tape is reloaded.

At this point the tape is a Tape Tiering Accelerator-formatted tape and the map TTA partitions and locate TTA partition can be used to map and position the tape to the desired partition. Normal I/O command processing is available to process the tape. In some cases, general utilities may or may not work once a Tape Tiering Accelerator-formatted tape is loaded and positioned to a partition greater than 0.

Start New Logical Volume

To start a new logical volume, a Tape Tiering Accelerator-formatted tape must be loaded or created. Each partition potentially can be a standalone 9.68 GB logical volume or multiple partitions contiguous or noncontiguously linked to form a logical volume.

1. Map TTA partitions described in section 2.3; partition 0 must be set to write. This step may have been performed already; it only needs to be done if the map needs to be changed.
2. Locate TTA partition described in section 2.4 to position the tape at the start of the new logical volume.
3. Write at least one block and nonbuffered sync or one nonbuffered tape mark. Any number of blocks can be written (such as labels) followed by a nonbuffered tape mark.

Convert a Tape Tiering Accelerator Tape to Standard Tape

Command X’77’,suborder X’67’, 4 byte parameter data to convert to standard tape

This command sequence will convert a Tape Tiering Accelerator-formatted tape back to a standard-formatted tape; however, this process is not recommended except in situations where a standard tape was accidently formatted as a Tape Tiering Accelerator tape and never used. The process of positioning and processing Tape Tiering Accelerator-formatted tapes may cause wear and debris to form around partition start and could degrade the tape's performance as a standard-format tape.

Perform subsystem function (PSF) suborder x'67' is used to remove the partitions and revert to standard format. The parameter data format is as follows:

1. PSF command with parm bytes 0-3 equal to X'67000000' reverts to standard format.
2. Write at least one block and nonbuffered sync or one nonbuffered tape mark. Any number of blocks can be written (such as labels) followed by a nonbuffered tape mark.

Tape Tiering Accelerator Examples

These examples demonstrate typical uses for Tape Tiering Acclerator to save space and improve access time.

Save Space

To format a Tape Tiering Accelerator tape, the application issues a set TTA mode command. The tape should be empty or it should be a scratch tape since initializing it may prevent access to data previously written on the tape. Formatting a Tape Tiering Accelerator tape divides the tape into a fixed number of empty partitions with a guard band allocated between each partition. Next, the application sets a write mask by issuing a map TTA partitions command; this command tells the tape drive which instances of partitions to link together during subsequent write operations. In Figure 2, the first 20 instances of partitions on tape have been mapped for writing. Each block is an empty partition that can be used by the application.

<table>
<tr><td>Files</td><td colspan="20">Empty – No data has been written to the tape</td></tr>
<tr><td>ALPs</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr>
</table>

Figure 2. An empty Tape Tiering Accelerator tape

The application now sends a locate TTA command to partition number 1 and a start new logical volume command sequence. The application now can begin writing to this logical volume on tape using standard tape write commands. This process may occur over multiple mount and dismount cycles and across several drives.

In Figure 3 below, the application writes nine files. Those files span all 20 instances of partition. The data shown in the top row are application files, and the bottom row indicates the physical relationship of the 20 instances of partitions to those files. The files can span one or more instances of partitions or be contained within a single partition.

Files	***A***		***B***			***C***	***D***		***E***	***F***			***G***			***H***		***I***		
ALPs	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**

Figure 3. Host files and Tape Tiering Accelerator allocation

Over time the files that are written to tape expire and become obsolete or invalid. As the expired files become obsolete, the tape begins to resemble "Swiss cheese," with holes created throughout the tape. In Figure 4, host files B and F expire and become obsolete, which frees partition numbers 2, 3, 4, 9, and 10 for space reclamation.

Files	A		Expired			C	D		E	Expired		G		H		I				
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 4. Host files becoming invalid

The application now can reuse the instances of partitions containing the expired files by starting a second logical volume and overwriting them. Initially, the application should execute a new set writeable TTAs command sequence that maps partitions 2, 3, 4, 9, and 10 as writeable. Then the application positions to the first partition that was freed when record B expired. In this case it is partition 2. To locate to partition 3, the locate TTA command is used. The application now can write files M and N. As these files are written to tape, the drive automatically positions to the next available partition. This process may occur over multiple mount and dismount cycles and across several drives. Although read access is still permitted to all instances of partitions, it is possible to write to partitions that form the current writable instances of partitions (2, 3, 4, 9, and 10) shown as green blocks below. File N links from partition 4 to partition 9 and then continues into partition 11. The space between file A and M (end of partition 1), N and C (beginning of partition 5), and E and N (end of partition 8), are wasted space.

This space is not empty; it contains the residual data from the expired files B and F. If the application attempts to read the old file B data, the beginning of that file will be read. Then the drive will report an end of data (EOD) error at the end of ALP 1. As a result, the application will need to stop reading at the end of file A. This process is discussed in detail below.

Files	A		M		N	C	D		E	N		G		H		I				
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 5. New files written to freed partitions

The application has to retain a record of the start of each record (host block ID and ALP number) to navigate through logical volumes where files have expired and partitions are reallocated to a new logical volume. To enable the first logical volume (yellow) in Figure 5 to be read, the the following steps must be taken:

1. Locate to TTA#0

2. Locate to the start block ID for record A

3. Read record A

4. Locate to TTA#5

5. Locate to the start block ID for record C

6. Read records C, D, and E

7. Locate to TTA#11

8. Locate to the start block ID for record G

9. Read records G, H, and I

Users should note that it is necessary for the application to track which set writeable TTA command map corresponds to each logical volume. There is a different write map for the first (yellow) and second (green) logical volumes.

Now the assumption is that files D and H shown in Figure 5 expire. Following the same process as described above, the application now can start a third logical volume (purple). Again, users start by sending a new set writeable TTAs command sequence that maps partitions 6, 7, 14, and 15, a locate TTA command to Tape Tiering Accelerator number 6. As shown in Figure 6, the host then can write this third logical volume with files S and T.

Files	*A*		*M*		*N*	*C*	*S*	*T*	*E*	*N*		*G*			*T*		*I*			
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 6. New files written for third logical volume

To read the second logical volume (green) in Figure 6:

1. Locate to TTA#2

2. Read files M and N

To read the third logical volume (purple) in Figure 6:

1. Locate to TTA#6

2. Read files S and T

It is worth noting that the application only needs to track the partition number and host block ID when space for expired files is reclaimed and used by another logical volume. In logical volumes that have not had space reclaimed, like the green and purple examples in Figure 6, the Tape Tiering Accelerator feature of the tape drive will move from partition to partition as if the logical volume was written sequentially in a single tape partition.

Improve Access

This example describes features that are only supported in the StorageTek T10000C tape drive. The following representation of a Tape Tiering Accelerator-formatted tape has five sections with five instances of partitions in each section, for a total of 25 ALPs. This format does not actually exist, and it is for illustration purposes only. The actual StorageTek T10000C tape drive ALP format has 480 instances of partitions and is too large to show in a small drawing. In this illustration, Tape Tiering Accelerator 0 is located at the beginning of tape (BOT) at the bottom left side and the ALPs serpentine through the tape in a linear fashion from BOT to end of tape (EOT) and back again.

	Section 0	Section 1	Section 2	Section 3	Section 4	
	← ALP 39	← ALP 38	← ALP 37	← ALP 36	← ALP 35	
	ALP 30 →	ALP 31 →	ALP 32 →	ALP 33 →	ALP 34 →	
	← ALP 29	← ALP 28	← ALP 27	← ALP 26	← ALP 25	
BOT	ALP 20 →	ALP 21 →	ALP 22 →	ALP 23 →	ALP 24 →	EOT
	← ALP 19	← ALP 18	← ALP 17	← ALP 16	← ALP 15	
	ALP 10 →	ALP 11 →	ALP 12 →	ALP 13 →	ALP 14 →	
	← ALP 9	← ALP 8	← ALP 7	← ALP 6	← ALP 5	
	ALP 0 →	ALP 1 →	ALP 2 →	ALP 3 →	ALP 4 →	

Figure 7. Representation of a Tape Tiering Accelerator-formatted tape

Section 0 in Figure 7 contains partition numbers 0, 9, 10, 19, 20, 29, 30, and 39. Since all of these are in a single section, the average access time to locate to any partition in section 0 is relatively short. A StorageTek T10000C tape drive has an average access time of approximately 10 seconds for any partition in the same section.

If the application requires fast tape access, this can be achieved with the construction of logical volumes within the same section. To build a logical volume in section 0, the following commands should be sent to the drive:

1. Load a StorageTek T10000C tape drive ALP-formatted tape.
2. At BOT issue a set writeable TTAs command sequence that maps partition numbers 0, 9, 10, 19, 20, 29, 30, and 39.
3. Issue a start new logical wolume command sequence.
4. Begin writing this logical volume.

As files are written to this logical volume, the drive will automatically link the mapped partition instances starting with the lowest numbered ALP. If application files are written to the first three partition instances in the logical volume to the partition, instances shaded in yellow will contain the logical volume as shown below. A wrap turn will be performed at the end of partition 0 and partition 9 will be written in the opposite direction. At the end of partition 9, another warp turn will be performed, and direction reverses again.

	Section 0	Section 1	Section 2	Section 3	Section 4	
	← ALP 39	← ALP 38	← ALP 37	← ALP 36	← ALP 35	
	ALP 30 →	ALP 31 →	ALP 32 →	ALP 33 →	ALP 34 →	
	← ALP 29	← ALP 28	← ALP 27	← ALP 26	← ALP 25	
BOT	ALP 20 →	ALP 21 →	ALP 22 →	ALP 23 →	ALP 24 →	EOT
	← ALP 19	← ALP 18	← ALP 17	← ALP 16	← ALP 15	
	ALP 10 →	ALP 11 →	ALP 12 →	ALP 13 →	ALP 14 →	
	← ALP 9	← ALP 8	← ALP 7	← ALP 6	← ALP 5	
	ALP 0 →	ALP 1 →	ALP 2 →	ALP 3 →	ALP 4 →	

Figure 8. partition numbers 0, 9, and 10 are written in section 0

If users desire to set a write mask for an entire section, the mask table below can be referenced for the binary values. For any section, the mask values are a repeating binary sequence indicated by the values show in the table below. For example: To mask the entirety of section 0 as writable, a writable mask containing 60 bytes with the hexadecimal values of 80,60,18,06,01,80,60,18,06,01……should be written to the drive.

Drive Operation

The following picture is taken from the Tape Tiering Accelerator example. For the discussion of drive operations, users can refer to Figure 9. It also may be helpful for users to refer to the Tape Tiering Accelerator example to see how the tape got into this condition.

Files	*A*		*M*		*N*	*C*	*S*	*T*	*E*	*N*			*G*		*T*		*I*			
ALPs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 9. Three different logical volumes with ALP

Logical Volume

A logical volume is defined as a Tape Tiering Accelerator group in which instances are linked together that have both a block 0 and a physical EOD. In the above figure there are two complete logical volumes. The first logical volume maps partition numbers 2, 3, 4, 9, and 10 (files M and N), and the second logical volume maps partition numbers 7, 8, 15, and 16 (files S and T).

Partial Logical Volume

A partial logical volume is a Tape Tiering Accelerator group in which instances are linked together with either no block 0 or no physical EOD, or neither a block 0 or an EOD. In Figure 9 there are five partial logical volumes. Partitions 0 and 1 are a partial logical volume with a block 0 but no physical EOD. Partitions numbers 16, 17, 18, and 19 make up a partial volume with no block 0 but it does contain an EOD. Partition 5, partition 8, and partition numbers 11, 12, and 13 are partial volumes without a block 0 or a physical EOD. These partial logical volumes contain the files that were not expired for the originally written logical volume shown in Figure 3. When combined, they still represent that logical volume.

Writing

When writing, the drive only allows writes on partition instances identified with the Set Writable TTAs command? using the Map TTA Partition command sequence. On load, the tape defaults to no partition instances being writable. The host should give the Set Writable TTAs command? by using the Map TTA Partitions command mask after load, and this must be done at BOT. If the Set Writable TTAs command? isn't issued, the tape will operate in a read-only mode.

The drive always chooses the lowest partition after the current partition when linking multiple instances of partitions. The drive does not allow the next partition to wrap around the writable list. For example, partition 255 will not be forward linked to partition 20.

The drive will report LEOV on the last free partition in a manner similar to a normal tape. The amount of space between LEOV and PEOV has not changed. When starting a write, the drive uses the current block as the append point, just like in normal tape operation, with one exception—if the write command is proceeded by a start new logical volume, the drive writes at block 0 at the start of the current partition. The host should be located at the start of the partition before the Start New Logical Volume command is issued. Regardless of where the host is logically positioned, the write will take place at the start of the partition.

When the application is finished writing a file, the application should issue a Detect Current TTA State command sequence, and the partition number where the last block was written will be in bytes 16 and 17.

Reading

When reading the tape, the drive automatically moves to partition instances that are linked together. For example, if the host is positioned to read partition 11, the drive transitions from partoition 4 to 5, 10, and 11 with no host intervention.

It is assumed that the host does not read old host files as a matter of course. But if the host requests to read the start of host file B in partition 2, for example, the data is returned until the end of partition 2, at which point the drive returns EOD.

Spacing and Locating

The drive operates on (partial) logical volumes when processing space block/file and locates commands. When spacing/locating prior to first block in a partial volume, the drive reports back that BOT has been crashed. For example, if record G in partition 11 starts at block 10,000, then any operation that ends up before 10,000 reports a unit check (UC) BOT.

Similarly, when spacing/locating beyond the last block in a partial volume, EOD is reported. For example, if record E in partition 8 ends at block 1,500, then any operation that ends beyond 1,500 reports a UC EOD.

Figure 3 and Figure 5 serve as examples. In Figure 3, file A starts in partition 0, and for this example, the starting block ID for partition 0 is zero. File B spans partition 2, and for this example, the starting block ID in partition 2 is 20,000. Now, file B expires. In Figure 5 the user decided to write file M starting in partition 2. The result of putting file M into ALP 2 caused an EOD to be put down at the end file A.

Starting block	**0**	**10,000**	**20,000**
Files	**A**	**B**	**C**
TTAs	**0**	**1**	**2**

Figure 10. Three files written into one logical volume

EOD

Starting block	**0**		**20,000**
Files	**A**		**C**
TTAs	**0**	**1**	**2**

Figure 11. File B expires but nothing has reclaimed that space yet

EOD

Starting block	**0**	**0**	**20,000**
Files	**A**	**M**	**C**
TTAs	**0**	**1**	**2**

EOD **EOD** **EOD**

Figure 12. File M reclaims the space that file B used to occupy

Normal Spacing and Locating

In figure 10, if the user can locate to any blocks in the range of 0 through 29,999. It is no problem because the partition instances are linked. If the user locates to 10,000, then the tape will end up in partition 1.

UC into EOD Example

Figure 10 shows files A, B, and C. Partition numbers 0, 1, 2 are all linked together into one logical volume. As shown, the EOD is at the end of partition 2. In figure 11, file B expires at the host. In Figure 12, the host reclaims partition 1. When that reclaim occurs, the links are broken and each partition has its own EOD. If the user now tries to locate to block 10,000, then the tape drive will generate a crash into EOD because partition 0 is no longer forward linked to partition 1.

UC into BOT Example

Figure 10 shows files A, B, and C. Partition numbers 0, 1, 2 are all linked together into one logical volume. As shown, the EOD is at the end of partition 2. In Figure 11, file B expires at the host. In Figure 12, the host reclaims partition 1. When that reclaim occurs, the links are broken and each partition has its own EOD. If the user locates to partition 2 with the locate TTA command, then the tape drive will be positioned in partition 2. Now, if the user tries to locate to block 10,000, then the tape drive will generate a UC into BOT because partition 1 is no longer backward linked to partition 2.

Tape Tiering Accelerator Layout

A StorageTek T10000C tape drive formatted for Tape Tiering Accelerator is represented in the StorageTek T10000C tape drive format specification. It has five sections of 96 partition instances for a total of 480 partition instances. The partition instances are organized in a linear serpentine pattern starting at ALP 1 at the beginning of tape (BOT) at the bottom left side, increasing sequentially down the length of tape to end of tape (EOT) on the right side, and then returning to BOT. Figure 8 shows this pattern.

Tape Tiering Accelerator Capacity

For the StorageTek T10000C tape drive, each partition is approximately 9 GB (480 * 9 GB is not equal to the full 5,000 GB that can fit on a StorageTek T10000C tape drive). This loss of full capacity is needed to allow the partition instances to be written in any order.

Potential Tape Tiering Accelerator Benefits and Effects

Increased Access Times

If an application identifies data for frequent access, that data can be located in partition instances at the beginning of the tape for fast access. The host can control where data is located on tape by setting writable partition instances.

Like data sets can be located together, so that access to multiple files of like data can be performed more efficiently.

Efficient Use of Tape Capacity

Depending on the application, much of the data on a particular tape is old, not needed data. This data now can be reclaimed without having to rewrite the valid data on the tape. With tape capacities ever increasing, the effective tape capacity will continue to worsen.

Theory of Operation

While there are many valid ways to use the set of commands to perform an individual application's requirements, this section shows how Oracle envisioned the use of the commands.

Initializing a Tape Tiering Accelerator Tape

To initialize a Tape Tiering Accelerator tape, the only requirement is to issue the create TTA tape command sequence on a tape that is in the free pool (it is important to know that the activate process is destructive). Best practices also set the writable partitions mask for byte 0, bit 7 to a logical 0x1 and recommend writing some sort of label or tape mark at partition 0. Partition 0 must be initialized before volumes that utilize remaining partition instances may be created or accessed.

Loading a Tape Tiering Accelerator Tape

After loading a Tape Tiering Accelerator tape, a map TTA partitions command should be issued so the tape can be written. The writable mask is kept by the host application based on which partition instances no longer hold valid data.

Positioning to a Host File

The host is responsible for keeping track of where each host file is on tape in the form of partitions and block ID (RFID). Positioning to a host file is as easy as issuing a locate TTA command to the proper partition and then issuing a standard tape locate command to get to the start of the host file.

Tape Tiering Accelerator tape read position byte-level references are relative to a start of volume position for a particular partition. If, for example, a second or third partition is created at partition 10 by issuing a start volume at that partition 10, then once a file is written beginning at tape partition 10, any subsequent read position information (RFID) for that partition will be relative to the volume beginning at partition 10. The same is true regardless of how many partitions are created. Issuing a start volume at a particular partition does not immediately set the beginning RFID to RFID 0; a write to that volume must occur first.

Reading a Host File

After positioning to a host file, reading is a simple as issuing standard tape read commands; the drive will automatically transition to next partition instances as needed.

Appending a Host File

When users append a new host file to an existing one, the host will first position the drive to the end of the existing host file. Next, the host should issue a detect current TTA state command sequence and note the value at bytes 16 to 17 in the host's file location structure. Next, the host should issue a standard read position command to get the RFID; this also should be noted in the host's file location structure. These values are used later when the new host file is accessed or for determining when a partition is free to be reused. Next, the host writes the new file. Another detect current TTA state command sequence is issued to determine the current partition. Based on the starting partition and current partition, the host can determine in which partition instances the file resides. This information is kept in the host file location structure and later used to determine when a partition is free to reuse.

Writing a Host File to a Free Tape Tiering Accelerator

When users start a new logical volume, the first command to issue is a locate TTA command to the first free partition that is kept by the host file location structure. Next, the host file location structure is updated with the first free partition and a host block ID of 0. Next, the host initiates a start new logical volume command sequence and writes the new file. A detect current TTA state command sequence is issued to determine the current partition. Based on the starting partition and current partition, the host can determine in which ALPs the file resides. This information is kept in the host file location structure and later used to determine when a partition is free to reuse.

Tape Tiering Accelerator Data Definitions

RSSD data definition:

```
MACRO
RSSDMAP,
*******************************************************************
*  Copyright (c) 2011, Oracle and/or its affiliates.             *
*  All rights reserved.                                           *
*                                                                 *
*  Dsect map for RSSD information returned for Generic Host       *
*                                                                 *
*  Revised 6-01-2011                                              *
*******************************************************************
*
*        RSSD Map
*
RSSDMAP  DSECT
RSSDLEN  DC    XL2'0'              Length of this RSSD Message
RSSDMID  DC    XL1'0'              Message ID
RSSDNMSG EQU   X'00'               .. No Message
RSSDIDST EQU   X'02'               .. ID Status
RSSDSTK  EQU   X'E0'               .. STK Message ID
RSSDMCD  DC    XL1'0'              Message Code
RSSDMCD1 EQU   X'01'               Dependent Format
RSSDMSGI DC    XL4'0'              Message ID
RSSDFLG1 DC    XL1'0'              Flag Byte
```

```
RSSDNNS  EQU   X'01'                .. Notify Non-supported
RSSDSOLM EQU   X'02'                .. Solicited Message
RSSDRSV2 DC    XL5'0'               Dependent on format and msg code
RSSDCART DC    XL1'0'               Cartridge Status
RSSDTTAC EQU   X'02'                .. TTA Cartridge is loaded
RSSDVOLS EQU   X'80'                .. Volsafe Cartridge is loaded
RSSDFET1 DC    XL1'0'               Drive Enabled Features
RSSDRB1E EQU   X'01'                .. Reserved
RSSDTTAE EQU   X'02'                .. TTA Drive Mode is Enabled
RSSDRB4E EQU   X'04'                .. Reserved
RSSDBFTM EQU   X'08'                .. Buffered Tape Marks
RSSDSTRA EQU   X'10'                .. Stop read Ahead
RSSD32BI EQU   X'20'                .. 3490 32bit BlockID Emulation
RSSD3590 EQU   X'40'                .. 3590 Emulation
RSSDVLSE EQU   X'80'                .. Volsafe Enabled
RSSDPART DC    XL2'0'               TTA Partition Number
RSSDRSV3 DC    XL6'0'               Reserved
RSSDCAP  DC    XL1'0'               Drive Capibilities
RSSDRS1F EQU   X'01'                .. Reserved
RSSDTTAF EQU   X'02'                .. TTA Capable
RSSDRS4F EQU   X'04'                .. Reserved
RSSDBTMF EQU   X'08'                .. Write Buffered TM
RSSDSRAF EQU   X'10'                .. Stop Read Ahead
RSSD349F EQU   X'20'                .. 3490 Capable
RSSD359F EQU   X'40'                .. 3590 Capable
RSSDVLSF EQU   X'80'                .. Volsafe Capable
RSSDRSV4 DC    XL8'0'               Reserved
RSSDHOST DC    XL1'0'               Host Type ID
RDDSRSV4 DC    XL10'0'              Reserved
RSSDSIZE EQU   *-RSSDMAP            Size of area
```

```
MEND
```

Conclusion

The Tape Tiering Accelerator feature of Oracle's StorageTek T10000C tape drive helps users capture wasted space that is caused by data set expiration. Tape Tiering Accelerator is preferred to reclaiming tapes because it takes far less time, thus reducing labor cost, increasing access times, and enabling more efficient use of tape capacity. Additionally, the tape drive handles much of the formatting without host intervention.

This white paper instructs users in z/OS environments about the use of Tape Tiering Accelerator, which creates automatically linked partitions so the data can span tiers or partitions and be noncontiguous on tape.

ORACLE®

How to Capture Wasted Cartridge Space in z/OS Implementations Using Tape Tiering Accelerator
April 2014

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together