High Performance RAIT

James Hughes, Charles Milligan, Jacques Debiez
Storage Technology Corporation
1 Storage Tek Drive
Louisville CO 80028-2129 USA
james_hughes, charles_milligan, jacques_debiez@storagetek.com
Tel: +1-763-424-1676
FAX: +1-763-424-1776

Abstract

The ability to move 10s of TeraBytes of data in reasonable amounts of time are critical to many mass storage applications. This paper examines the issues of high performance, high reliability tape storage systems, and presents the results of a 2-year ASCI Path Forward program to be able to reliably move 1GB/s to an archive that can last 20 years.

This paper will cover the requirements, approach, hardware, application software, interface descriptions, performance, measured reliability and predicted reliability. This paper will also touch on future directions for this research.

The current research allows systems to sustain 80MB/s of incompressible data per Fibre Channel interface which is striped out to 8 or more drives. A RAIT system looks to the application as if it were a single tape drive from both mount and data transfer. Striping 12 RAIT systems together will provide nearly 1GB/s to tape.

The reliability is provided by a method of adding parity tapes to the data stripes. For example, adding 2 parity tapes to an 8-stripe group will allow any 2 of the 10 tapes to be lost or damaged without loss of information. An interesting result of this research is that the reliability of RAIT with 8 stripes and 2 parities exceeds that of mirrored tapes even though 8 mirrored tapes requires 16 actual tapes and 8 data tapes plus 2 parity tapes only requires 10 actual tapes.

Keywords: RAIT, High Performance, Archive

1 Introduction

This paper describes the RAIT system as designed for the US DoE as a part of the ASCI program. This system is designed to facilitate the long term archive of large quantities of information in the face of potential media failures.

The requirements of the project are three fold,

- Ensure the reliability of large archives

- Compatible with the existing applications
- Transfer the data at a high data rate

The reliability of tape varies from manufacturer to manufacturer. At STK, our high reliability 9840A tape devices have shown to have an average reliability of one permanent read error every 20TB of data read. While this is significantly better than some other vendors, this error probability is not zero, and can never be.

A probability of a read every every 20TB with a 20GB cartridge, means that a cartridge can be read 1000 times between errors. In general, this is not a significantly high number, but when combined with large multi-volume datasets (files that span and/or stripe out to many cartridges), the effects are multiplied.

For example, a 3 TB backup to 9840A with 1.5:1 compression will require 100 cartridges with a native capacity of 20 GB. Simply because of the number of cartridges involved, there is a 1 in 10 chance that there will be a permanent error in writing or reading the data. Since any error destroys the backup or restore operation, the results are catastrophic to the data.

2 Other Methods

For completeness, we mention other RAIT systems and documentation.

First, although somewhat dated, the Storage FAQ [1] discusses the general issues of RAIT and several vendor offerings. For commercial hardware offerings we find Ultera [10]. For software offering we find Computer Associates [9].

In addition, many backup companies offer striping solutions, these include IBM's HPSS, Veritas and Legato [8, 11, 12]. These striping solutions can provide the performance that a RAIT system provides, but does not add additional data protection. When using a striping solution care needs to be takes because striping multiplies the probability of problems. Our system focuses on solving the robustness problem of stripes tape making the result more reliable than a single tape drive.

This paper is focused on the data protection and transparency of a full virtualized RAIT system. We use the term "full virtualized RAIT" to mean a system that completely hides all aspects of the RAIT system from the application. The application only sees a single tape volume with a single volume serial number. The application issues a single ACSLS tape mount and transfers data to a single tape drive [4, 2, 3].

3 Approach

If we compare RAID to RAIT, they are very similar except that tape is a removable media. We have accomplished RAIT by adding parity (as in RAID), but we have extended this to virtualize the removable media and to provide additional redundancy beyond the single parity of RAID.

The approach that Storage Technology took for the ASCI RAIT project is to virtualize the entire tape operation. By "virtualize" we suggest that the application's view of the operation need not be in full agreement with the reality of the operation.

In this system, the application thinks that it is mounting, writing or reading a single volume from a library. In reality, the “virtual volume” does not exist and another group of real cartridges contain the actual information and additional redundancy which contains what the customer actually gets when the data is read.

The reality is that the single virtual tape mount may have resulted in up to 10 or more real tape mounts, and the data that was transfered to the single virtual drive will have parity added and spread to all the real drives.

In RAID-5, there is a single parity drive. Many customers have experienced multiple failures on a single RAID stripe. In RAIT, multiple parities are created so that if there are any multiple failures (up to the number of parity tapes) the data will be intact.

Just like RAID where the application sees no difference between a RAIT controller and a non-RAID controller, the application that is requesting the RAIT tape mount and writing or reading the data has no knowledge of the reality. In general this will allow any application that reads or writes tapes to be able to write RAIT.

3.1 Hardware

All hardware RAIT Systems are designed to interpose a device between the host and the physical tape library and drives. The device presents a virtual image of the virtual tape and the other deals with “reality”. The STK device has 2 basic parts; a mount proxy and a parity generation/checking data path.

3.1.1 RAIT Proxy

The virtual-to-real tape mount operations are accomplished by the RAIT proxy. This is the device that understands the mapping between the virtual volumes and the real volumes. It also manages the creating of virtual tape pools, reconstruction pools, and control of data path.

The RAIT proxy has a database that contains the persistent information necessary to associate the mapping of virtual volumes to the real physical cartridges. This is critical so that applications that use the virtual volumes are hidden from this fact. This database is mirrored to multiple locations and backed up. In addition, to aid in the transportation and introduction of RAIT groups into other RAIT systems, and to act as a final fall-back to ensure that this information can not be lost, this information is also written onto the tapes as meta-data which is hidden from the user.

From the application point of view, it requests a mount from what looks like a standard STK ACSLS mount service. This single volume that the application requested is translated by the RAIT proxy to real volumes, and these real volumes are mounted.

This is a valuable feature in that it allows applications that only know about ”normal” tape volumes to take advantage of RAIT. This completely parallels RAID where the client hosts have no knowledge that the device is RAID.

Once the mounts are complete, the RAIT proxy initializes the data paths with information on where the tapes are mounted, the number of data stripes and the number parity stripes.

3.1.2 Data Path

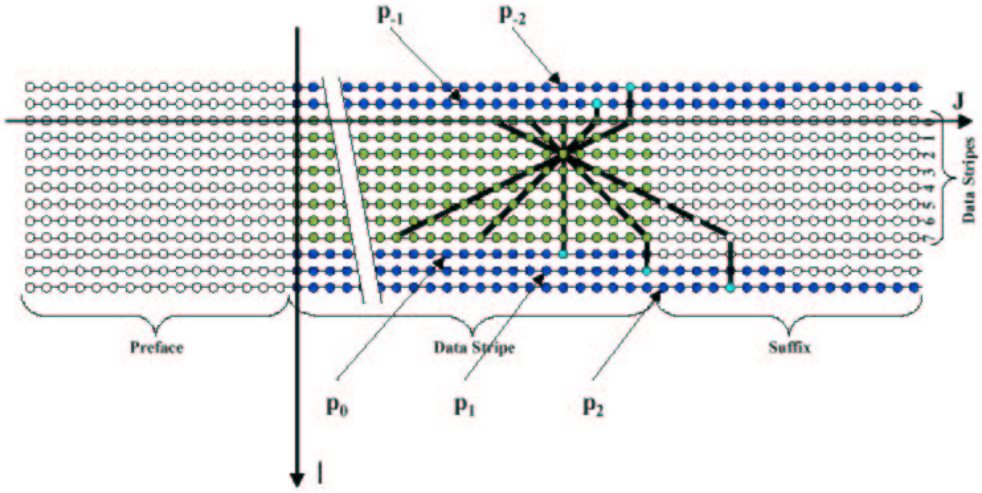

Figure 1: Striping data

The data path stripes the data and creates or checks the multiple parity stripes. Figure 1 graphically shows the representation of how the data is striped and parity calculated.

In this figure, the data is described as a block of **J** words and is striped into **I** horizontal groups. An additional prefix and suffix are added before and after and then the parities are added above and below. In this case there are 5 parities, they are -2, -1, 0, 1 and 2. There are no inherent limit to the number of parities. These parities are described by their row/column slope.

The prefix and suffix contain zeros so that the end-cases of parities which extend beyond the start or end of the user data will have deterministic results. These zeros do not really exist and are not stored on the media, but are included here to illustrate the parity construction.

P_0 calculation is simply the vertical line through the horizontal data stripes. This XOR of the data is stored in the third from the bottom parity stripe. The other parities go through the data and then are stored in their respective stripes.

You will also notice that the parities (other than P_0) are longer than the data. This data is necessary to "bootstrap" the correction function, to keep the blocks independent. This additional data is stored on the tapes. This does not represent a significant lengthening of the parity data. P_x has an additional length of $|x|I$ words. If the blocksize of the tape is 1 MB using 32 bit words on an 8 way stripe, this will result in a 0.003% increase in the length of the parity blocks over the data blocks. Since tape is a variable length block device, this is not a significant factor.

An alternative (and a formal) description; a block of application data that is sent from the host is divided into **I** stripes. Those stripes are sent to the parity hardware to create

multiple linearly independent parity stripes.

The parity generation is accomplished by the creation of vertical and various cross parities. Each of the parities P_k are taken in order from a list $k \varepsilon < 0, 1, -1, 2, -2, 3, -3... >$. The data is described as a word $D_{i,j}$ where i is the stripe number and j is the offset.

$$P_{k,j} = \bigoplus_{i=1}^{I} D_{i,j-ik}$$

Where if $D_{x,y}$ is out of bounds, it is assumed to be 0.

3.1.3 Variable Configuration

The configuration of the RAIT volume is selectable. The volumes can be simply defined as N+P where N is the number of data stripes and P is the number of parities.

An optional, and more exact description can be N+(P1,P2) where N is the number of data stripes and P1 and P2 describe the number of parities when failures during write operations are allowed. In this case, P1 is the total number of parities desired and P2 is the minimum number of parities that must be present for the write to be successful.

A simple example: 6+(2,1) will mean that a write of a volume starts out with 8 devices and that during the creation, one can fail and the overall write will be signaled to the host as being good. P2=1 specifies that if there is not at least 1 parities written at all times, then the host will be notified that the write operation has failed.

3.1.4 End of Tape Operations

As data is written, the first real tape device that reaches end-of-tape signals an end-of-tape to the application that this virtual volume is now full. Many years ago, hosts needed to know how many bytes can be written on a tape device. Modern host software assumes that data sensitive compression is occurring on the tape device and no longer needs to know how long a tape is anymore. Programs today simply write data until the tape says “enough”.

Before the data is written out to the drives, it needs to be rotated across the drives because the parities are not as compressible as the user data. Parities are less compressible because when two compressible pieces of data are XORed together, the result is less compressible.

Care must also be taken to ensure that the parity stripes (less compressible) are not written to separate tapes from the (more compressible) user data. Failure to do so will result in the parity tapes always getting to end-of-tape first thus wasting the compression on the data tapes. This is solved by rotating the data and parity stripes over the complete N+P group.

3.1.5 Reconstruction

To eliminate as many errors as possible, we leave all the drive’s error recovery on at all times. This means that all the data integrity features of the device are left enabled. On 9840 this means that Read After Write and full ECC are enabled.

If data can not be read or written correctly the drive notifies the RAIT controller. All retries are used to try to make sure that this is not a transient error.

One important side effect is that, if the data shows up and the drive says there was no error, it can be assumed to be correct. Conversely, if there is an error on read, we can just assume that the data will never be readable and treat that block as missing. Since we know which block is missing, then any one of the parities can be used to correct that stripe.

For instance, if a data stripe is missing and P0 is available, the simple parity of the valid data stripes and the parity *is* the missing data stripe.

Multiple parities are more complicated. For example, if there are 3 missing data stripes and three parities (P0, P1 and P-1) we perform the recovery as follows. Starting with the "correction line" as the first word of each stripe we notice that the top missing stripe can be corrected with the parity stripe going from left bottom to right top. This is because all the words to the left are good (because the prefix is known to be zero) and all words above the top missing stripe are (by definition) not missing. When we are at this case, we can correct the first word of the top missing stripe. We then correct the bottom missing word in the same manner with the other diagonal parity. At this time, there is one remaining missing word and one remaining parity (P0). We can simply use P0 to correct the remaining word. We can them move the correction line to the right by one word.

Subsequent words within this block can be corrected the same way because as we iterate this from left to right, all words to the left of the correction line have been corrected. This simple scheme can be enhanced to any number of errors as long as there are enough parities. When there are more than 3 errors, then the correction line is no longer straight.

These errors correction techniques are discussed in [6] as a "burst erasure channel". A burst erasure is defined as an event where, if there is an error detected, an entire burst (block in our case) is erased (in our case simply not returned from the device). To recover from an error, we simply use the parity to recover the known bad data stripe. IBM introduced the concept of "Crossed Parity"[7], and patents for further extensions to this have been proposed by the authors.

3.1.6 Reconstruction performance

Since this is a burst erasure channel, if all bursts (stripes) arrive without the drive saying there is an error, then we can assume there are no errors and simply reconstruct the user data block without employing the parity hardware at all.

In the case where a single stripe is missing, the parity of everything but the missing stripe *is* the missing stripe. We can employ the parity hardware to create the syndrome in the same time as we took to create the parity in the first place. This allows us to correct a single missing stripe with no performance penalty.

When multiple errors occur, we can use the parity hardware to create partial syndrome for each word and then do the word by word iteration in software.

3.1.7 Additional Data Integrity checks

Provided that all the parity is not needed to correct missing data stripes, the controller can do additional data integrity checks of the data.

3.2 Application software

In general, the application software does not need to understand the operation of the virtual tape devices. The initial customer uses HPSS and the testing of HPSS is accomplished without change to HPSS. Other software such as Veritas or Legato Backup software operates the same whether the tape device is RAIT (virtual) or real.

The one exception is in the area of job scheduling. If the job scheduling system manages the tape drive allocation to ensure that there are adequate resources, this needs to take into consideration that certain tape mounts will not require a single drive, but may require multiple drives. This has been added as a feature to HPSS.

3.3 Performance

The performance of the RAIT system is limited by the speed of the data channel, parity hardware and tape devices themselves. At this time, a 100MBytes/s Fibre Channel is used to connect to the host devices. Fibre Channel can be reliably utilized at 80% of capacity or 80MBytes/s. The parity generator hardware operates at more than 100MBytes/s so that it is not a bottleneck. The devices used are STK 9940 tape devices that have a raw speed of 9MB/s. This number is increased by the compression factor. If the user data is compressible 2:1, then the performance of the tape device will be 18MBytes/s.

A 5+2 RAIT system with 9940s operating with 1.8:1 compression can sustain 80MBytes/s as a single virtual tape device being striped out to 7 physical tape drives.

3.4 Reliability

It has been shown that STK 9840s have a read error about every 20TB of data with a cartridge size of 20GB. Since this is 1000 reads of a single cartridge, a very simple model is to assume a probability of error of $e = 10^{-3}$ per tape operation and unrelated failures [5].

If we assume that a tape operation is a mount and unmount of a tape regardless of the amount of data transfer, then this will be a conservative estimate and the actual reliability should be significantly better than this.

The probability of at least one error for any group of tapes (stripes or just long volumes) is the number of volumes (v) times the error ev. A 100-tape volume has an error probability of 10^{-1}.

A single RAIT virtual tape volume (r) with one parity per 6 volumes (6+1) will only fail if 2 tapes fail. A simple model of this is the probability of 1 of 6 failing and then 1 of 5 remaining fail or $r_{6+1} = 6e5e$ or $r_{6+1} = 3 \times 10^{-5}$.

A (6+2) system will only fail if 3 tapes fail. This is the probability of 1 or 7, 1 of 6 and then 1 of 5 remaining fail or $r_{6+1} = 7e6e5e$ or $r_{6+2} = 2.1 \times 10^{-7}$.

3.4.1 Striped RAIT Systems

Since it is still possible for the application to stripe the data, the application can be used to stripe the data over multiple independent RAIT systems. For instance, 4 RAIT groups at 80MB/s will sustain 320MB/s or more than 1TB/hour.

A striped RAIT system S-wide will have an overall reliability of Sr. In the case of 13 (6+2) RAIT units wide the probability of failure is Sr or 2.7×10^{-6}. This shows that the reliability of a 1GigaByte/s striped RAIT has an error probability of less than two in a million probability of data loss due to unrelated failures.

3.4.2 Other Failure Modes

The analysis in this paper focuses on unrelated drive, and media failures. The performance of the system in the face of related failures at the controller level is not considered. Generally, failures at the controller level do not effect the stored data, which can be read or written once the controller is repaired.

3.5 Future directions

Storage Technology Corporation is in the process of creating a Commercial Off The Shelf device for worldwide availability. STK is also creating a ”mirroring” capability so that tapes can be created simultaneously at multiple locations with the same kind of single virtual device image as RAIT. The performance of the system is also expected to increase as customer systems and tape devices become faster.

3.6 Conclusion

This paper has discussed the method of creating RAIT. The primary goal of reliability is accomplished by adding parity information to the virtual volumes. Performance is increased by striping the data. Further performance can be achieved by striping RAIT systems. In the future this capability will be commercially available.

References

[1] R. Van Meter. *comp.arch.storage FAQ*
http://alumni.caltech.edu/~rdv/comp-arch-storage/FAQ-1.html”

[2] M. Fisher. *Redundant Array of Independent Tape: RAIT*, THIC, October, 2000, Bethesda MD.
http://www.thic.org/Agenda_1000.html

[3] G. Sobol, *SAN Enabled RAIT/RAIL*, Computing in High Energy Physics, CHEP’00 Padova Italy, Feburary, 2000.
http://chep2000.pd.infn.it/abs/abs_c016.htm

[4] J Hughes, C. Milligan, J. Debiez. *High Performance RAIT*, Computing in High Energy Physics, CHEP’01 Beijing, China, September, 2001.
http://www.ihep.ac.cn/~chep01/paper/4-004.pdf

[5] R. Defouw, C. Milligan, and T. Noland, *The Level of Data Protection in Redundant Tape Arrays*, Storage Technology Internal Correspondence, May , 2000

[6] W. W. Peterson and E. J. Weldon, *Error Correcting Codes*, 1961, John Wiley & Sons Publishers.

[7] A. M. Patel, *Adaptive cross parity code for a high density magnetic tape subsystem*, IBM J. Res. Develop., vol. 29, pp.546–562, 1985.

[8] R. W. Watson and R. A. Coyne, *The parallel I/O architecture of the high-performance storage system (HPSS)*, Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems, IEEE Computer Society Press, September 1995, pp. 27–44.

[9] Computer Associates,
http://www.cai.com/products/san/saniti_strategy.htm

[10] Ultera Corporation,
http://www.ultera.com

[11] Veritas Corporation,
http://www.veritas.com

[12] Legato Corporation,
http://www.legato.com