RESURRECTION: RETHINKING MAGNETIC TAPES FOR COST EFFICIENT DATA PRESERVATION

A Thesis
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

By
Varun Shimoga Prakash
December 2013

RESURRECTION: RETHINKING MAGNETIC TAPES FOR COST EFFICIENT DATA PRESERVATION

Varun Shimoga Prakash

APPROVED:

Weidong Shi, Chairman
Dept. of Computer Science

Nikolaos V. Tsekos
Dept. of Computer Science

Xiaojing Yuan
College of Technology
University of Houston

Dean, College of Natural Sciences and Mathematics

Acknowledgements

This thesis not only represents the academic contribution made at the University of Houston, it is also a milestone in more than a year of research work done in the I2C Lab in the Computer Science department. It gives me great pleasure in expressing my gratitude to all those people who have supported me and had their contributions in making this thesis possible. First and foremost, I would like to give my devout gratitude to the Almighty and my family for the endless support and strength I have received from them throughout my life.

I express my profound sense of reverence to my supervisor and promoter Dr. Weidong Shi, for his constant guidance, support, motivation, and untiring help during the course of my Masters at U of H. His in-depth knowledge of computer and software systems has been extremely beneficial for me. He gave me freedom and opportunity, the two most important aspects of advising which nurtured my imagination. My sincere thanks also to Dr. Omprakash Gnawali for his constant support and help during my research.

Some of the unforgettable memories of my days in the lab are incomplete without the occupants of the lab. My thanks to Yang, Xi, Tao, Coco, Dainis, and Nick for being there every day, all the time. A special thanks to Pranav and Madhuri for cooking food for me (almost every day), joining me for lunch and being my family away from home. I am also very grateful to Qiang and Donny for being great friends and for the many enjoyable walks to Starbucks.

RESURRECTION: RETHINKING MAGNETIC TAPES FOR COST EFFICIENT DATA PRESERVATION

An Abstract of a Thesis
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

By
Varun Shimoga Prakash
December 2013

Abstract

With the advent of Big Data technologies-the capacity to store and efficiently process large sets of data, doors of opportunities for developing business intelligence that was previously unknown, has opened. Each phase in the processing of this data requires specialized infrastructures. One such phase, the preservation and archiving of data, has proven its usefulness time and again. Data archives are processed using novel data mining methods to elicit vital data gathered over long periods of time and efficiently audit the growth of a business or an organization. Data preservation is also an important aspect of business processes which helps in avoiding loss of important information due to system failures, human errors and natural calamities.

This thesis investigates the need, discusses possibilities and presents a novel, highly cost-effective, unified, long- term storage solution for data. Some of the common processes followed in large-scale data warehousing systems are analyzed for overlooked, inordinate shortcomings and a profitably feasible solution is conceived for them. The gap between the general needs of 'efficient' long-term storage and common, current functionalities is analyzed. An attempt to bridge this gap is made through the use of a hybrid, hierarchical media based, performance enhancing middleware and a monolithic namespace filesystem in a new storage architecture, Tape Cloud.

The scope of studies carried out by us involves interpreting the effects of using heterogeneous storage media in terms of operational behavior, average latency of data transactions and power consumption. The results show the advantages of the new storage system by demonstrating the difference in operating costs, personnel costs and total cost of ownership from varied perspectives in a business model.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Explosion of Data

The world of Big Data is changing dramatically right before our eyes, from the amount of data being produced to the way in which it is structured and used. The modernization and technological advancement in many fields has contributed to the generation of large quantities of soft information which provides critical feedback on ways to make operating practices in the field better. This not only provides exceptional opportunities in improving businesses, but the sheer size of the data also presents enormous challenges in terms of efficient data maintenance and computing. Experts in the field point to an average 4300% increase in the data generated every year by 2020 [9]. This is no surprise to anyone in the information technology arena who has faced or is currently facing the burden of data proliferation. Growth rates are especially high for the digital archives of compliance data (both government and corporate mandated), video/multimedia files, digital images, fixed content, and social-network-related data[21].

Management of data includes the ability to efficiently compute and economically store raw data and results. The computation of data has been supported by processing systems which have undergone continuous improvement over time. This includes central processing units, high-performance processing techniques, specialized kernels, platforms and operating systems. We can imagine the overall management of data to be a pipelined process made up of three distinct stages. The first stage is the collection of raw data to be processed, the second stage is the processing of this raw data and the final stage is the storage of the newly generated results. The size of the data storage needed in the first and final stage has grown rapidly over time. This is due to the "more data, better results" postulation that is commonly revered in many fields[12][38]. A common requirement for any organization to maintain or improve the profitability of managing such data is to minimize the cost of storing these data. An unavoidable concern arises when the importance of the data stored cannot be judged and its long-term need is uncertain.

1.2 Data Archive and Backup: Correlations and Contrasts

At any time, data can be easily distinguished on the basis of its momentary importance. While much research and development work has been done to suggest effective ways of handling transiently valuable data by using faster caches and buffers, the trivial, usually much larger data are not the focus of active research any more. The fact of the matter is that the smaller, faster storage devices used to store the momentarily important data are much more expensive than the slower, much bigger storage devices. In the hierarchy of memory and storage, the less important data slowly trickles down until it is stored in media that are less potent in performance and speed. This process is at the heart of most data archiving methods followed today.

While it is commonplace to use the terms archiving and Backup interchangeably, distinct features differentiate the two processes. Archiving of data involves the transfer of inessential data to economical storage tiers in the storage stack at regular intervals. The storage stack can be represented as a pyramidal structure with the archiving tiers taking up the base of the pyramid while the apex comprises of expensive, agile and highly-responsive storage tiers that are closest to the data processing units. The access patterns, as discussed in the later parts of this document, show high rates of data transactions at the apex of the pyramid and a lower, nevertheless prominent data input and output rates at the base.en a unit of data stored at the foot of the pyramid needs to be accessed, it moves up the hierarchy to ensure quicker access in case it is needed again in the near future. The units of data that have resided unaccessed for long durations of time at the higher levels in the memory hierarchy are slowly pushed down the pyramid.

Backup, on the other hand, is the process of organizing and storing data for long-term preservation. The probability of access of backed up data is smaller than that of archived data. Backup of data servers taken at regular intervals can help in auditing, disaster recovery and data loss prevention. Backup servers are located on-site at the location of the organization generating the data or in remote locations and the transfer of data are usually done over the network or by manually shipping the media that contains the data to the location of the storage.

The correlations between archiving and data backup, however, can be found in the fact that the speed of data transactions at both the archiving and backup levels is much slower than regular, secondary storage. We attribute this to the use of certain technology where delay in data transactions is inherent in its operation. Moreover, the large size of data stored in data archives and backup systems is a common feature, leading to the investment of extra efforts and time to seek the required units of data within the storage infrastructure.

1.3 Current Traditions and Infrastructures

As mentioned before, archiving and backup of data can be performed either at the site of the production of data or at remote locations. While both have its advantages and disadvantages, either of the methods is chosen based on the need.-site archiving might help in faster recovery of data but off-site archiving might contribute to data protection against natural calamities. Organizations with sensitive and important data such as financial institutions prefer using both on-site as well as remote backup of data. Archiving systems are generally made up of redundant array of independent hard disks (RAID) [22] based storage systems that are normally attached to generators of data such as web servers over a network. This setup is commonly known as network attached storage or NAS[30]. Organizations usually combine the storage needs of more than one web server together to set up an exclusive network of data storage devices called as storage area network or SAN[31]. A dedicated group of servers within the NAS or SAN is used to store the archived data.

Data backup, on the other hand, conventionally uses the well-known magnetic tapes as a medium to store data. Magnetic tapes have demonstrated extended shelf life and easy transportability characteristics over many years. Magnetic tapes, which started off as a primary storage media decades ago, have been preferred for backup storage of information generated by organizations for a long time now. There has been a continuous development of quality, form factor, capacity and robustness of the storage cartridges[7]. Magnetic tapes are much cheaper than hard disks in storing unit data so it continues to be an economic type of storage medium. Periodic copies or snapshots of data on web servers are stored sequentially on tapes, which are then vaulted for long-term preservation.

Some archival data may be retained for decades, although rarely (if ever) accessed. And now, there is considerable growth of archived data due to the activities of billions of data-centric applications in addition to tens of thousands of traditional data-generating businesses. The situation demands effective archiving solutions that make data easy to find while lowering the cost of ownership for whatever organization is ultimately tasked with the responsibility for storing it. Today's archival storage environments need to be robust, inexpensive, secure, and scalable. Fortunately, significant enhancements to tape drives, tape media, and libraries have occurred in recent years, meaning that for a lot of use cases, tape technology is now an optimal choice for staying ahead of data growth. To be specific, the tape is now exceedingly appealing for use as cloud storage. In fact, some cloud storage service providers are reaching unprecedented levels of cost control, security, scalability, and availability by deploying tape in the cloud[33][21].

1.4 The Cloud Backup Model

Cloud storage is a model commonly found in networked enterprise systems where data are stored in virtually partitioned pools of storage which are generally hosted by designated service providers or hosts [36]. Hosting companies operate large data storage centers and individuals and organizations that require data to be hosted, buy or lease storage capacity. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers and multiple placements. The safety of the files depends upon the hosting company, and on the applications that leverage the cloud storage. Information stored with these providers is accessible via the internet or Wide Area Network (WAN). The economy of scale enables providers to supply data storage cheaper than the equivalent electronic data storage devices.

The introduction of cloud storage and the virtual-storage models have brought about notable changes in both the technical and business specifications of organizations requiring large and affordable data storage space[21]. The pay-per-requirement feature offered by the cloud service providers helps service users to choose the exact amount of resources needed to avoid wastage and exempts them from infrastructure and personnel costs. Cloud data storage has many advantages. From the service users' perspective, it is cheap, doesn't require installation of heavy software or hardware components, doesn't need replacing, has backup and recovery systems, has no on-site physical presence, requires no environmental conditions, requires no personnel and doesn't require energy for power or cooling.

1.5 Challenges

1.5.1 Cost of In-House Ownership

Economic factors have an immense influence on the storage initiatives and storage infrastructure that an IT organization can deploy. The main issue with having in-house NAS and SAN infrastructures for archiving and backup of data is the cost of ownership and operation. The infrastructure needed in the form of hardware, storage media, software and operating personnel is usually a large investment for any organization to make. High-capacity electronic data storage devices like file servers, Storage Area Networks (SAN) and Network Attached Storage (NAS) provide high-performance, high-availability data storage accessible via industry standard interfaces. However the electronic data storage devices have many pricey drawbacks, including the need for regular maintenance and have short operational lifetimes. To make the SAN system generally efficient for all the web servers using it, the commonly used storage medium is the hard disk which is not only expensive in nature, but also demands other support needs such as the software needed to establish RAID and adequate cooling systems in data centers. In this regard, magnetic tapes and tape-based infrastructure fare much better in terms of operating and media costs. Historically, magnetic tapes have been a more affordable medium compared to hard disks and the fact that the tapes do not require perpetually powered infrastructure for its working, decreases the power consumption and reduces the need for cooling systems in large data centers.

Despite the cost advantage, the growth of the use of magnetic tapes has not been as impressive as that of hard disks. This is due to the comparatively poorer performance of magnetic tapes when used in a SAN like storage in terms of average response time of data retrieval and data write as compared to hard disks. Also, the systems needed to operate magnetic tapes, which usually involves tape drives and tape libraries are much too expensive to highlight the economic benefits of the tape media alone. The large investment needed to host an on-site backup system with either of the storage media tends to be a considerable investment to make.

1.5.2 Monetary “Leak” and its Causes

The flexibility and potential cost savings of storage clouds hold great appeal for several reasons. Cloud storage provides elasticity, with capacity growing on-demand as the business requires and scaling back when capacity is not needed. You pay only for what you use. Typically, public cloud storage service providers charge for the capacity used monthly, the data-transfer bandwidth, and value-added services performed in the cloud (i.e., security, encryption, deduplication, or replication in other locations for redundancy).

At this stage, it is clear that in cloud-based storage services, there are usually more than one player involved, such as service users and service providers. From the service user's perspective, the motives for choice of storage would be reduced costs per unit data stored, efficient retrieval, options for specialized support and secure, long-term data storage. But, a service provider's considerations span operating cost efficiency, labor, scalability, support for different types of data, varied policies from multiple clients and managing workload uncertainty among others. A closer observation shows that the cost factor favors one or the other player but not both. Some factors such as the likelihood of recovery of data after they have been backed up, also firmly influence the offering by service providers.

A study by Joab Jackson [17] showed that over 90 percent of the data in networked servers sits untouched. In such situations, service users either end up unnecessarily paying for large and expensive storage space to store unimportant data, or the service providers offer cheap storage at the cost of operating and maintaining the servers which again leads to lower profits. Even with efficient auditing of the required storage space, both players still face tough tasks in dealing with unexpected workloads and unauthorized elastic expansion. We term this concept as "Monetary Leak", which results in inefficiencies in business and technology propositions. In order to cope with such anomalies, service providers must have at their disposal, a cheap, energy efficient and performance sufficient storage tier which can be used for storing data over long periods of time without having to invest much in its maintenance and a platform which can intelligently transport specific units of data to this tier when needed. Thus, a sensible inclusion in the storage tiers to archive low-read/write-only data would be a low cost, low maintenance yet durable media [17].

1.6 Architecture for a Solution

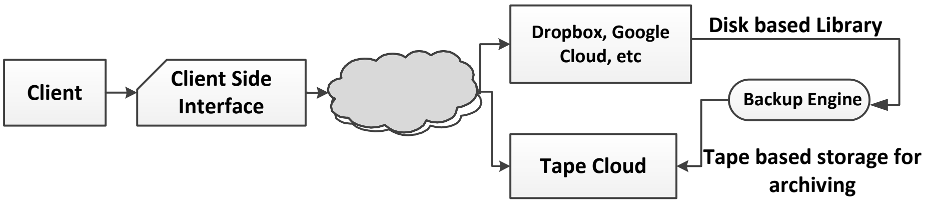


Figure 1.1: Representation of the Architecture of Tape Cloud.

In this thesis, we propose and evaluate a design prototype to overcome the challenge of monetary leak and provide cost beneficial storage options to both service providers and service users. Our design leverages the economic advantages of magnetic tapes and combines them with the cloud-based storage models. Technically, the proposed design is the result of elaborate exploration of specifications and configurations to enable the integration of magnetic tape libraries into cloud-based storage systems such that the resulting storage stack is a centrally accessible, unified backup and archiving infrastructure. The tape layer, in this case, serves by capturing and restraining the redundant, archaic and trivial valued data within its storage boundaries as an inline activity in the regular process of storing and not as an explicit activity that a user needs to initiate. It can also provide periodic backup at low costs with little operation and maintenance costs.

Tape Cloud (figure 1.1), as christened in this document, is a venture that seeks to find suitable solutions that are analogous to the ideals mentioned before. Tape Cloud is a cloud-based, nearline storage Infrastructure-as-a-Service which makes use of magnetic tapes as the main backend storage medium. The cloud model exempts users from the large initial investments needed for in-house backup infrastructure, external tiers for archiving legacy data and its maintenance. From the service provider's perspective, using tapes allows hassle-free scaling of systems and reduces the total cost of ownership due to its characteristic low power usage, durability and form factor per unit data.

In designing Tape Cloud, we defined certain goals that helped in conceiving the system as per the requirements. Our principal intention is to **1.** Reduce the average response times for read requests issued by applications - This is the ultimate requirement and a non-empirical measure to judge the performance of Tape Cloud. **2.** Conjointly, ensure efficient data writes to the tapes tier of storage - Productive write capability ensures the need for smaller buffers and prevents clogging of data when there are more than one sources competing to store its data; and **3.** Strengthen the infrastructure's support for a large and diverse client base - Given that Tape Cloud is offered as a cloud-based service, it is natural for the systems to expect a wide range of data with varying workloads. We cite the efficiency of Tape Cloud in terms of its support for and stability in dealing with various kinds of data[13]. However, overcoming the latency offered by tapes is a complex problem to be solved. Even with the latest in tape technology, high-performance in terms of reading and writing of data cannot be achieved as delays caused due to seeking and winding of tapes is still persistent. There is also a delay induced by the stock robots and other ambulatory mechanics within the tape library which physically handle and move the tape cartridges.

The main contributions of our work are as follows,

- We analyze some of the common practices in large-scale backup and archiving systems and their shortcomings. In seeking a suitable solution, we focus our research on the cost benefits and possibilities for enhancement that are available through the use of tapes and related technology.

- We observe and record the common delay incurred in the operation of a commercially available tape library. Some of the latencies of tape drives and tape filesystem are analyzed using typical benchmarking tools. These data, along with the delay are used to model the performance characteristics of unit hardware, which are later used to simulate data centers.

- We analyze backup and archiving application traces to obtain typical workload characteristics. We employ specific methods to trace the operations at different stages in the storage infrastructure and aggregate the traces into meaningful statistics. This not only provides information about backend storage medium activity, but also provides information about activities at the application and filesystem levels.

- We propose the design of an exciting new alternative in tape-based big data storage system, tape Cloud", which allows organizations to benefit from a large storage bandwidth without having to invest in the infrastructure itself. The crux of this approach lies in the optimization of the usage of the most affordable storage medium. Tape Cloud allows storage service providers to focus on storage technicalities such as using affordable medium, setting up of the required infrastructure and planning the essential man power while permitting client organizations to concentrate monetary expenditures away from investing on the fundamentals of data backup and storage.

- We evaluate the merits of a middleware that is designed to work with Tape Cloud. The functions of the middleware include the aggregation and batch processing of data, IO request management and efficient distribution of data over available resources. The middleware is independent of factors such as operating system, type of data, workload and deployment specifications.

- The middleware, which is constituted by a FUSE filesystem, an implementation of priority-based queuing of IO tasks and a latency preemptive, probabilistic data distribution scheme, acts between the backup application tier and proprietary filesystems that is commonly used with tapes.

- We synthesize artificial workloads which emphasize prominent features of backup applications and use them to evaluate the impact of using the proposed middleware in a simulated deployment of Tape Cloud. In keeping with the project's goals, we demonstrate and provide proof of improved data distribution ability, improved response time for read requests originating from each of the applications' workloads and regulation of write requests that the middleware provides through experimental results.

Chapter 2

State-of-the-Art and Research Direction

In this chapter, we present the results of some of the studies we performed to understand the positive and negative aspects of the state of the art in storage technologies. We conducted a comparative analysis of different storage media and storage configurations which helped us obtain a clear definition of the problem at hand and a direction for research to find a solution. We begin the chapter with a brief introduction to the data backup process, its importance and the different types of data backup that is commonly performed. A study about the types of storage systems and its internal architecture was performed, including a one-on-one comparison of the different aspects (cost of operation, maintenance, personnel, random IO, sequential IO etc.) of the three major storage media used for data backup namely hard disks, solid state drives and tapes. Collectively, the difference in costs arising due to these aspects can lead to the phenomenon of “Monetary Leak” introduced earlier. With a special inclination towards tapes, we note the benefits and constraints posed by tapes and tape libraries in the course of its regular operation. This information helps us in obtaining a clear understanding of expenses, monetary or otherwise, at various levels within the storage infrastructure and clarifies our research direction.

At the highest level, the goals of our research are deceptively simple. Throughout this document, all the assumptions and claims that are made and the experiments conducted are ultimately aimed at fulfilling ALL of these goals in the best possible and compatible ways.

- REDUCE the overall cost of storing data for clients who want to use the backup and archiving service.
- REDUCE the cost of owning, maintaining and operating the archiving and backup infrastructure for service providers.
- MAINTAIN a record of good performance in terms of fast data read and efficient data write into storage.
- Enable ADAPTATION of the archiving and backup infrastructure to ensure support for different types and workloads of data.

2.1 Types of Backup

Backing-up of data is a crucial process that many organizations perform in order to have a fail-safe reserve, for when the inevitable happens. The principle is to make copies of particular data in order to use those copies for restoring the information if a failure occurs (a data loss event due to deletion, corruption, theft, viruses etc.). Each program has its own approach in executing the backup, but there are four common backup types implemented and generally used in most of these programs: full backup, differential backup, incremental backup and mirror backup. The type of backup also influences the infrastructure deployed for the purpose. Thus, it is recommended that a clear understanding of the needs and circumstances be obtained before enabling a backup application to use a certain backup storage system.

2.1.1 Full backup

Full backup is the starting point for all other types of backup and contains all the data in the folders and files on personal computers and servers that are selected to be backed up. Because full backup stores all files and folders, frequent full backups result in faster and simpler restore operations.

2.1.2 Differential backup

Differential backup contains all files that have changed since the last full backup. The advantage of a differential backup is that it shortens restore time compared to a full backup or an incremental backup. However, performing a differential backup too many times might cause the size of the differential backup might grow to be larger than the baseline full backup.

2.1.3 Incremental backup

Incremental backup stores all files that have changed since the last full, differential OR incremental backup. The advantage of an incremental backup is that it takes the least time to complete. However, during a restore operation, each incremental backup must be processed, which could result in a lengthy restore job.

2.1.4 Mirror backup

Mirror backup is identical to a full backup, with the exception that the files are not compressed in zip files and they cannot be protected with a password. A mirror backup is most frequently used to create an exact copy of the source data. It has a benefit that the backup files can also be readily accessed using filesystem tools.

2.2 Comparing the Costs of Storage Media

Throughout modern history many digital storage systems have been researched, developed, manufactured, used and eventually shelved in an effort to address ever-increasing demand for high-speed, efficient and economical storage systems. This cycle of innovation has lead us to a new generation of NAND Flash memory-based solid state drives (SSDs) that represent the next evolutionary step in both enterprise and consumer storage applications. While many of the storage devices have seen trends of decreased costs of operation as its usage increased, the current era is still in its infancy for SSD-based systems to be widely and profitably deployed. The most popular storage media used in a large number of the systems consists primarily of hard disks of larger capacity and a comparatively smaller and faster random access memory tier made up of SSDs.

Magnetic tapes were in active usage over the last 60 years, a time period that is comparatively high for a storage technology in the growing IT age. This has been possible due to the continuous improvement in the quality, capacity, durability and areas of application of tapes. The dominant motivation to use tapes, however, arises from the fact that tapes are highly economical and ideal to store large amount of data which may or may not be useful to the organization in the near future. The Linear Tape-Open is a set of standards that directs development and manages licensing and certification of media and mechanism manufacturers. The standard form-factor of LTO technology goes by the name Ultrium, the original version of which was released in 2000 and could hold 100 GB of data in a cartridge.

Table 2.1: Comparison of Costs of Storing Unit Data On Different Storage Media

	Solid State		Hard Disk		Tapes	
	$/TB/yr	% total	$/TB/yr	% total	$/TB/yr	% total
Media	19456	96.0	220	25.1	23	5.4
Capital	197	0.009	163	18.6	155	36.4
Maintenance	152	0.007	176	20.1	88	20.7
Facilities	221	0.010	118	13.5	56	13.1
Personnel	234	0.01	197	22.5	103	22.4
Total	20260		874		425	

LTO version 6 released in 2012 can hold 2.5 TB in a cartridge of the same size as its predecessors. Another very important storage media that has been around for a long time is the Hard Disk Drive. The similarity between hard disk and tapes is only the principle of using magnetic material to store data. Hard disks rose to the scene as a better option to store data that needed faster retrieval. The cost of operating hard disks has been an attractive tradeoff to the cost of the medium itself. This also has proved to be the "tape killer" as the hardware required to operate tapes and the personnel required to maintain them cost unreasonably high. A comparative study performed let us see that the monetary expenses in the form of initial investment involved are too large for many small organizations to have backup solutions on tapes despite its competitive cost advantages, results of which have been shown in table 2.1.

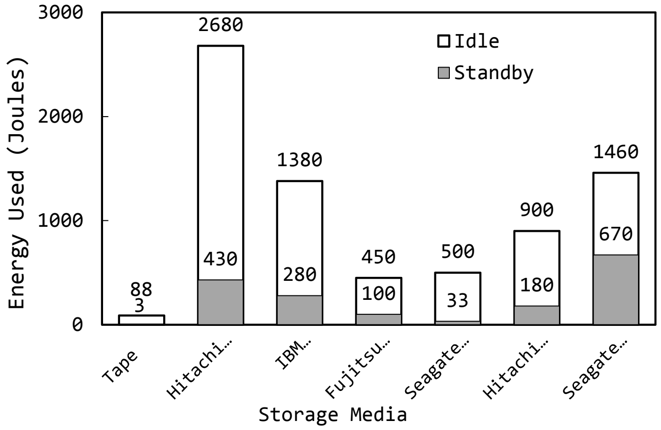


Figure 2.1: Idle State Power Consumption of Hard Disk vs Tapes.

Another contributor to costs is the overall energy consumption of storage media and the associated infrastructure. Power inefficiencies in storage systems can arise at two stages, when the system uses a lot of energy even in the idle state (in figure 2.1, the column title tape shows least power consumed) and when the operating temperatures (in figure 2.2, tape libraries show least increase in operating temperatures) of the system results in the need for external temperature conditioning systems. A study performed with a tape drive and six hard disk drives of different specification and manufacturer reveals that tape is much more power efficient than hard disk when used on a large- scale such as in data centers. The fact that hard disks need to be electrically powered for operation is a mammoth disadvantage over tapes which is a non-powered static storage and has very low power consumption per unit data [28].

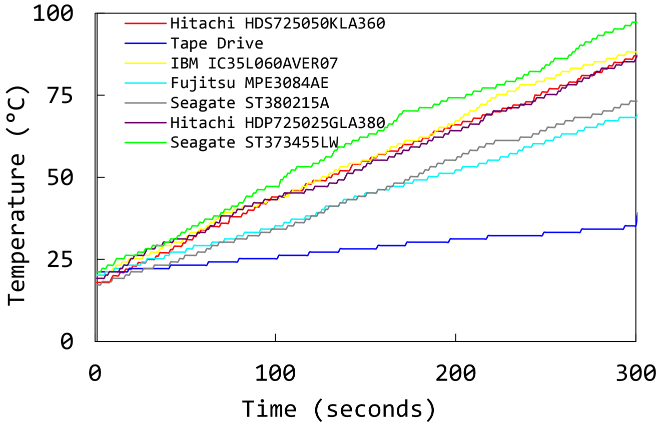


Figure 2.2: Operating Temperatures of Hard disks and Tapes-based systems.

Figure 2.3 shows the total cost of ownership of tape versus hard disk-based storage systems as a function of time. From the figure, the energy cost incurred over a five year period for disks is comparable with the acquisition cost of the tapes with equal storage capacities. From the perspective of the storage service provider, scalability becomes very expensive [16] [4] [3]. From the Clippers model [27][26], for long-term archiving of digital data, the average disk-based solution costs 15 times the average tape-based solution. The cost of energy alone for the average disk-based solution exceeds the entire TCO for the average tape-based solution.

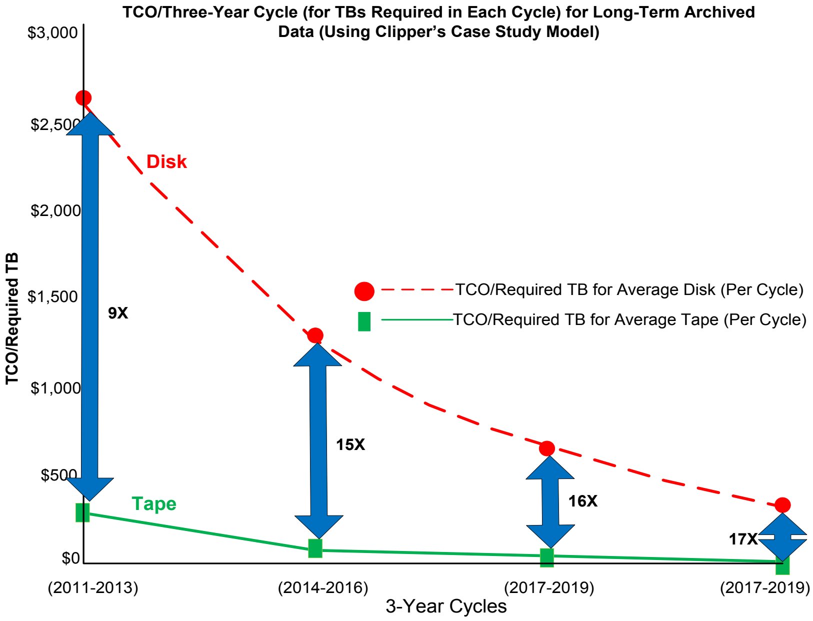


Figure 2.3: Tape and Disk: Acquisition and Energy Cost from INSIC Study (Clippers Model).

The overall cost of storing data on different types of storage system can directly depend on the above mentioned factors. Closer investigations show that individual benefits of a particular type of storage can vary significantly from one another. For instance, magnetic tapes require tape libraries to operate on enterprise scales and hard disks require the appropriate networked software and hardware infrastructure for its operation. While the medium itself is less expensive in the case of tapes, the overall cost of operation sets it on par with hard disks. This is caused mainly due to the extensive gap that exists between the costs related to medium and costs related to using the medium in a system. Figure 2.4 shows the difference in costs of the storage hardware and the complete storage system for varying amounts of data. Apparently, due to the offering by companies which include more features than the hardware that is available off the shelf, the total cost of ownership in both cases differ slightly.

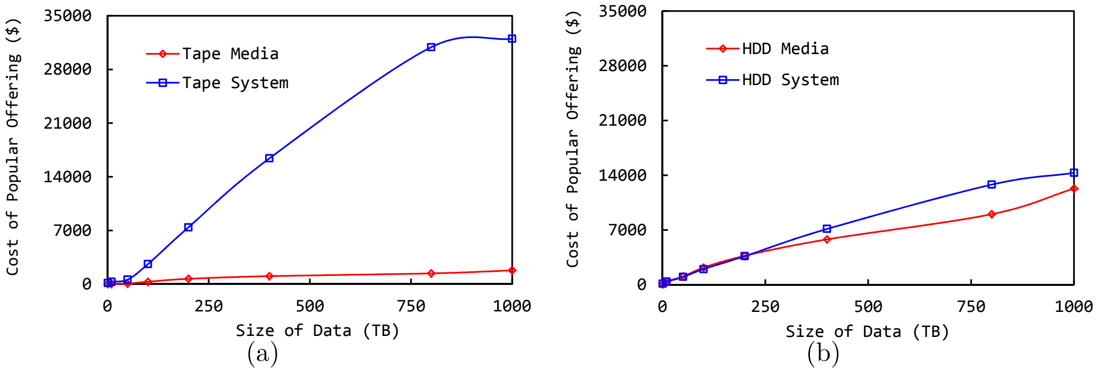


Figure 2.4: Comparative costs of storage media alone versus the total cost of storage system.

2.3 Contrasting Performance and Overhead

The infrastructure for using tapes which, in most cases, is the tape library, introduces a new kind of expenses in the form of delays which need to be incurred due to the use of robotics inside the device. The delay caused can create a slowdown of I/O processes, indirectly affecting business processes of the organization [7]. We monitor and study some of the device specific delays by conducting experiments over a commercial LTO-5 tape library (e.g., Tandberg tape library[1]).

Figure 2.5 shows the diagrammatic representation of the simple tape library (the number of tape slots is a parameter that varies among different tape library products). The numbered slots are tape cartridge holders. The robotic cart runs on the rail in front of the tape driver and helps load the tapes into the driver. To complete the analysis, we perform a multi trial recording of the delay of various operations performed within the library. The basic principles of operation can be well understood and a run-time profile can be created for these operations which help us in creating faster and more efficient hardware. Table 2.2 shows the delay incurred in moving tape cartridges in the numbered slots to the drive and back to the slots after performing the operation. The average time taken for the transport of cartridges in both cases is more than a minute. Once the tape is in place, it takes nearly 30 seconds for it to load and be ready to read or write. The tape transportation cart has an upward path time of 2.6 seconds and a total end to end path time of 7.4 seconds. The robot usually performs both together during the load or unload operation from a slot at the very end of the library so time can be saved. Based on the numbers, it is possible to get a clear time profile about the tape library operations. One of the design objectives is to reduce the time spent on the tasks such as movement of tapes from the slots to the drive.

Table 2.2: Tandgerg T24 Load and Unload Delays

Type	**From**	**To**	**Motion(sec)**	**Load(sec)**	**Type**	**From**	**To**	**Motion(sec)**	**Load(sec)**
LOAD	1	Drive	62.4	33.3	UNLOAD	Drive	1	61.6	30.1
LOAD	2	Drive	62.9	31.9	UNLOAD	Drive	2	62.3	30.6
LOAD	3	Drive	64.06	32.6	UNLOAD	Drive	3	62.26	30.3
LOAD	4	Drive	65.2	34.6	UNLOAD	Drive	4	64.0	30.3
LOAD	5	Drive	62.42	34.0	UNLOAD	Drive	5	61.3	30.9
LOAD	6	Drive	63.3	33.6	UNLOAD	Drive	6	61.76	31.01
LOAD	7	Drive	64.2	31.3	UNLOAD	Drive	7	62.22	30.1
LOAD	8	Drive	65.45	33.9	UNLOAD	Drive	8	63.8	29.62
LOAD	9	Drive	61.8	34.0	UNLOAD	Drive	9	60.7	30.3
LOAD	10	Drive	62.3	31.6	UNLOAD	Drive	10	61.4	30.34
LOAD	11	Drive	63.7	32.23	UNLOAD	Drive	11	61.97	33.9
LOAD	12	Drive	64.02	33.8	UNLOAD	Drive	12	63.6	32.59
Average	–	–	**63.64**	**33.1**	**Average**	–	–	**62.37**	**31.21**

Due to the delay induced by the robots in the tape library and the delay caused by the winding and seeking of tapes, random access of data on tapes perform poorly

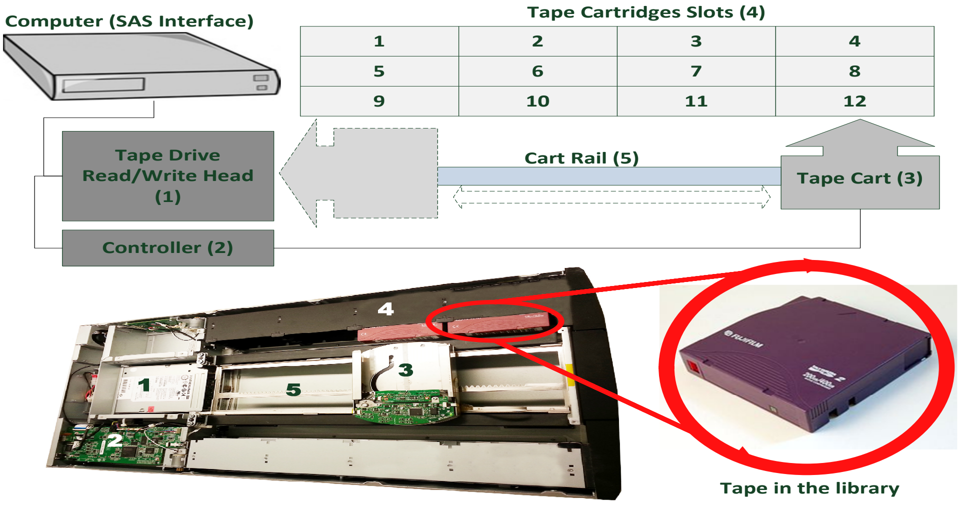
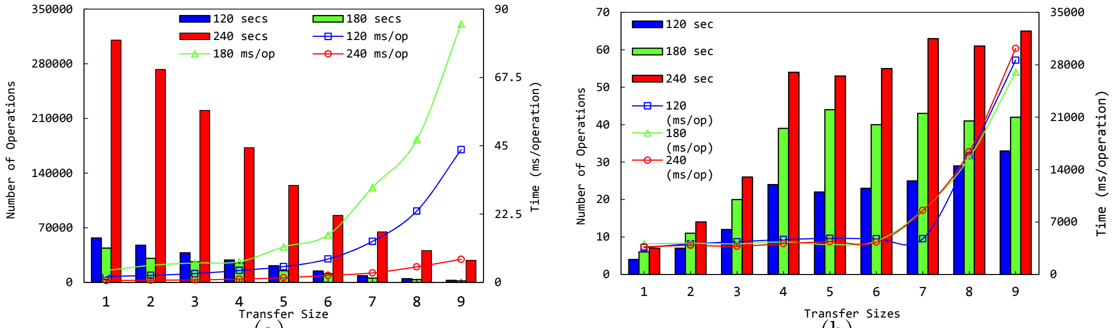


Figure 2.5: Representation of the Tandberg T24 Tape Library.

in comparison to hard disks. Although the rate of sequential read and write by tape drives is on par or better that that of most commercially available hard disks as shown in figure 2.8, the performance when judged based on the common parameters of input/output per second (IOPS) and milliseconds per operation (ms/op) is not in the same league as that of hard disks (and definitely not even worthy of comparing with SSDs considering the fact that SSDs have been designed for significantly high IOPS and much lower ms/op). Figure 2.6 and figure 2.7 gives a comparison between the IO performance of hard disks-based filesystem(a) and tape-based filesystem(b) in terms of IOPS and milliseconds per operation. The x-axis in all figures is a notation of the transfer size of data where 1 represents 8KB, 2 represents 16 KB and so on. As seen in the figure, the number of both random read and write operations on tapes is much lower than that of hard disks and takes a longer period for each operation to complete due to the periodic winding of tapes to the correct point that needs to be performed.



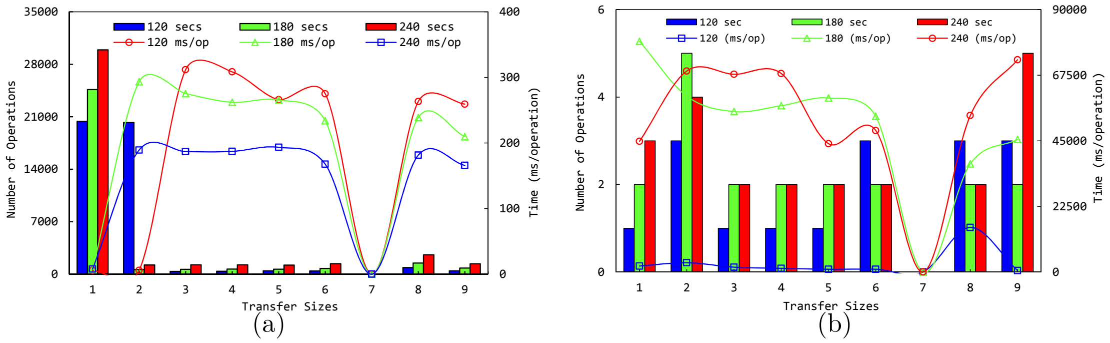
(a) (b)

Figure 2.6: Read IOPS of hard disk-based ext4 filesystem and tape-based LTFS filesystem.

2.4 Conclusion from Study

In this chapter, we focus on the ground truth and a side by side comparison of some of the popular storage media and associated systems that are available in the market today. Research in the area of tape technology has lost steam over the decades. Academia has given in to the marketing strategies of the industry and moved on to research in disk and SSD-based storage systems. The industry, although actively facing the problem of monetary leak, prevents the usage of tapes under the justification of advancement in technology as tapes are considered by most as obsolete. Few major players realize the importance of tapes and continue to market it, while most companies are taking a proactive role in assisting other organizations to overcome tape usage.

The cost of total ownership is a combined component of two major entities that is usually expressed for a fixed number of years, the initial investment needed and the



(a) (b)

Figure 2.7: Write IOPS of tape-based Linear Tape File System.

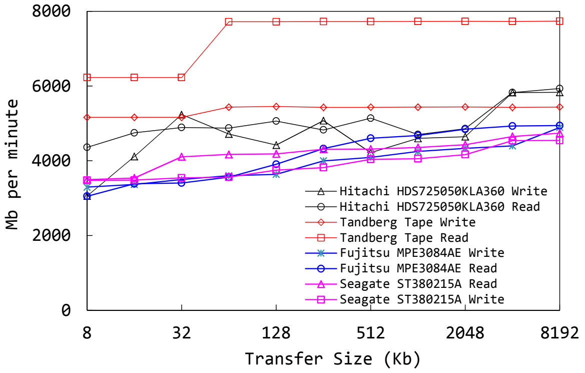


Figure 2.8: Sequential read and write performance of tapes versus hard disks-based filesystems.

cost incurred over the operational period of time. It is apparent from figure 2.3 that the difference in the total cost of ownership and operation per terabyte of data stored over a period of three years is much higher than that needed for tapes. This can be attributed to the low power consumption and reduced need of cooling systems which contributes to much of the savings. The total cost of ownership definitely benefits tape operators in the long run as it prevents monetary leak in both storage service users and providers. However, the initial investment needed in the form of tape libraries and infrastructure is much larger than the costs incurred for hard disk-based devices as shown in figure 2.4. The flat rate of growth of usage and popularity is another culprit for its rising costs. From the performance perspective as shown in figure 2.8, it is apparent that tapes have an impressive steaming capacity but not noteworthy numbers in terms of random read and write requests.

The numbers and a close observation of deficiencies of either storage system encourages finding a suitable solution which can benefit both service users and providers both commercially and technically. Some of the quick points that can be noted are:

- Creation of a service that can benefit storage service users by minimizing the cost incurred over a period of time to store massive amounts of data and help reduce the overall cost of operation and infrastructure ownership for service providers and effectively reduce monetary leak.
- Leverage the cloud business model and the best of what tapes and hard disks have to offer in a combination. The resultant system is a unified hybrid of the two storage types. This system will have the ability to migrate and manage data within the storage stack based on certain parameters such as storage time, the importance of data and the duration of data retrieval.
- Overcome the high cost of hard disk medium by reducing the number of hard disk units in the infrastructure, and substituting them with tapes. The quantity of hard disk that is expelled is a function of the duration of which a particular block of data with specific size is stored in the system. More the stale data stored in the system, the higher is the opportunity to eliminate expensive hard

disks and introduce cheaper tapes.

- Overcome the latency of tapes by reducing the delay associated with random access and library robotics. The options include redesigning tape libraries and have faster locomotion of tapes within it or managing the data write and read requests in a way that can improve performance and efficiency of tapes. To make the system compatible with current tape libraries, the management of delay needs to be performed above the hardware layer.

- Ensure minimum modifications to the current hardware and software to improve compatibility and ease of integration and deployment. Also induce the need for minimum alteration on current data backup and archive storage practices, including the logic followed by current backup and archiving applications.

Chapter 3

Operational Models and Solution Approach

In order to design an infrastructure around a particular storage medium, it is important to understand the characteristics, advantages and weaknesses that are associated with it. A clear understanding of the storage medium and devices can eventually lead to its large-scale deployment such as in data centers. These characteristics are expressed quantitatively using models which are conjured using some of the defining features of a particular storage medium and the algorithms used in the storage system. Using these models, it is possible to isolate the performance of system in dealing with individual units of data or blocks. Using these models, we can simulate the behavior of Tape Cloud. The models are mainly used to determine the delays caused by tape libraries and the consequence of particular hardware configurations. Therefore some of the main discussions in this chapter span the models that can be created by the general scenarios that are faced by the tape libraries and the hardware architectures.

3.1 Modelling Tape-Based Delays

Although many published works and projects focus on performance metrics and improvement of hard disk-based storage subsystems [20][39][15], little has been done in the direction of analyzing tape performance. Some of the works that have been done in [35] and [24] focus on how best magnetic tape technology can be used in archiving and backup systems. Since the advent of LTO-4 and LTO-5 magnetic tapes, there has not been an in-depth study of the performance of tapes. In this thesis, we study some of the quantitative characteristics of the hardware and the software that are required to design a suitable storage system. The study, unlike hard disk-based systems, involves the need for tape libraries which introduce new concerns for modelling tape storage performance. The resultant models are functions of the features of data such as block size, combined with the features of tape drives such as wind, seek and data transfer time and also with that of the tape library such as the speed of robotics. The results of a study performed on various block sizes shows that tape drives have a uniform data transfer rate compared to three other hard disks. However, a difference in performance can be seen when random reads and writes are performed. The time spent in changing tapes, loading and seeking to the correct point on the tape creates delays that are out of proportion when compared to the sequential performance of tapes. An important takeaway from the results is that one must assure that the tape drive and the infrastructure spends most of the time either writing or reading to tapes and less time performing seek operations. This helps us in deciding important parameters such as rate of batch processing of data. The following subsections present an introduction to important terminologies that are used often in the chapter.

3.1.1 Average Response Time

The time between the end of an inquiry or demand on a storage system and the beginning of a response by the storage system can be termed as the average response time. The intermediate time is used to locate and fetch the data from within the storage system. The average response time depends on a number of factors such as the rate of disk/tape IO, queue depth and heterogeneity of requests. Response time can be affected by changes to the processing time of the systems above the storage layer and by changes in latency, which occur due to changes in hardware resources or utilization.

3.1.2 Write throughput

The amount of data an application can write to stable storage on the server over a period of time is a measurement of the write throughput of a storage system. Write throughput is therefore an important aspect of performance. All storage systems must ensure that data is safely written to the destination file while at the same time minimizing the impact of server latency on write throughput.

3.2 Generic Models for Tape-Based Latency

Storage systems have a mode of operation that are similar in most of the cases. For example, they have read and write requests that are issued by more than one application. The requests are queued for completion by internal data IO queues or task queues. These queues store references to location in main memory where data temporarily resides and to locations that data needs to be written in the secondary storage tier. Some of the dynamic attributes of the queuing techniques include the time taken for tasks in each queue to be completed, queue depth and waiting times for the execution of the tasks. Some of the circumstances that is encountered based on these attributes can be used to model the overall behaviour of tape-based storage systems. Based on the facts obtained about the hardware and delays, an attempt to model the latency for generic cases[14] is made. For the models, the following are some of the constants that need to be considered.

- T_{search}(i) is the time taken by the robot to locate and move to the tape to execute the i^{th} request in the task queue.
- T_{load} is the time taken to load the tape into the drive.
- T_{unload} is the time taken to unload the tape from the drive.
- T_{seek}(average) is the time to wind the tape to seek to the position of the first byte to execute the new task. The average time for LTO5 tapes is considered in this case.

- γ_{read} is the data transfer rate for read operations of the tape. Similarly γ_{write} is the data transfer rate for write operations on the tape.
- The smallest unit of a data that is considered in this case is a block. A single read or write might involve transaction of a varying number of blocks. The representation of a unit block is made as BLK.

Keeping with the goals that were introduced earlier, the design of Tape Cloud seeks to reduce the average response time required to read data of a particular size from the tape tier in the storage. The system employs able techniques to reduce the average response time T_{read} for read tasks and furthermore, ensure that these read-friendly techniques, cause minimal distortion to the write throughput and total time T_{write} required to collect data and write it onto tapes. Thus, for a workload Θ,

$$T_{opt}(\Theta) = min(T_{read}(\Theta) + \Delta T_{write}(\Theta)) \tag{3.1}$$

Where $T_{opt}(\Theta)$ is the optimal time required to complete the execution of workload Θ. The analysis of some of the latencies and overhead incurred in achieving this goal is performed in different scenarios. These scenarios are commonly occurring cases in storage systems.

3.2.1 Scenario 1: Single Read/Write Task in Queue

When there is a single read task in the task queue, the total amount of time required to complete the task and obtain the block of data is given as the sum of times taken for a series of events. Thus $T_{singleRead} = T_{search} + T_{load} + T_{seek} + n(\frac{BLK}{\gamma_{read}})$ where n is the total number of unit blocks that need to be read. $\frac{BLK}{\gamma_{read}}$ is a constant, the total time required to read a single block and can be substituted by Γ_{read} to get

$$T_{singleRead} = T_{search} + T_{load} + T_{seek} + n\Gamma_{read} \tag{3.2}$$

Similarly, a single write operation in a queue undergoes similar delays as read operations, the only difference being the rate at which data is written to tapes. The delay for a single write operation is given by

$$T_{singleWrite} = T_{search} + T_{load} + T_{seek} + n\Gamma_{write} \tag{3.3}$$

3.2.2 Scenario 2: Write task(s) before Read task in Queue

In scenarios where there are one or more write tasks in the queue before a read task, the total time required to obtain the data will include the time required to complete the write task too. For a single write task before the read task, the total time required to complete the task will be equal to $T_{total} = T_{search} + T_{load} + T_{seek} + n\Gamma_{write} + T_{unload} + T_{search} + T_{load} + T_{seek} + n\Gamma_{read}$. This can simply be written as $T_{total} = T_{singleWrite} + T_{unload} + T_{singleRead}$. Generalizing this, when there are N write tasks before a read task, it is apparent that

$$T_{total} = N(T_{singleWrite}) + \xi(T_{unload}) + T_{singleRead} \tag{3.4}$$

where $0 \leq \xi \leq (N-1)$. ξ is called the tape switch rational which determines the probable number of tape changes that need to be made and is based on BLK and n. Thus BLK is an important value that influences the efficiency of write operations and helps in deciding the maximum size of data that can be written as a continuous process on to a single tape.

3.2.3 Scenario 3: Other Read Task(s) before Read Task in Queue

The total time required for a particular read task to complete when there are one or more read task ahead of it differs from the previous scenarios in that, read requests are usually not localized to a single tape mostly due to replication and data striping. Continuous read requests mean more search, load and seek operations, thus increasing the overall time taken. In the worst case, the total time taken can be given by

$$T_{total} = (N+1)(T_{singleRead}) + (N)(T_{unload}) \tag{3.5}$$

where there are N read requests ahead of the read task in question. This not only causes excessive delays in retrieving data but also leads to the pile up of write tasks at the queue in situations where there is an equal ratio of read to write requests.

3.3 Multiple Reader Model

When we imagine a large-scale storage system, it is hard to narrow down on the exact specification of the storage hardware needed. As far as Tape Cloud is concerned, a certainty that can be assumed is the synchronized operation of a large number of tape drives or tape readers and writers. If there are more than one readers/writers available, the data IO request scheduling problem is NP-Hard. Since it is impossible to find the optimal solution, a partitioned solution for multiple readers is examined as shown in Alg.2. In this design, each reader is responsible for a specific set of tapes within the tape library unit. Each reader only operates on the tapes assigned to it. For example, if there are two readers and 100 tapes, the first reader may work on the first 50 tapes while the second reader is in charge of the rest. Thus we can also postulate that the tapes belonging to a particular set are physically close to each other.

However, this design may result in a “hot spot” situation where one reader may be busy all the time while others are idle. To solve this, the system allows the idle readers help to process the requests, but have to be back to their own duties if requests to their assigned tapes arrive.

Algorithm 1 Partitioned Task Scheduling Algorithm for Multiple Tape Readers

```
Require: $m$ tapes: $\Gamma_t = \{tape_1, tape_2, \cdot, tape_m\}$
    $n$ tape readers: $\Gamma_{tr} = \{reader_1, reader_2, \cdots, reader_n\}$
Ensure: Online Schedule for each coming requests
Assign the $m$ tapes to $n$ readers. Each reader will take care at most $\lceil m/n \rceil$ tapes which
    are close to each other.
Store the assignments to the global schedule manager.
while TRUE do
    Wait for the next request, $req$.
    Get the meta-information of the $req$ from the database and find the $tape_{id}$ for $req$.
    Forward $req$ to the reader, which is in charge of the requests for $tape_{id}$.
    Schedule $req$ at the reader locally using elevator schedule algorithm.
end while
```

3.4 Proposed System's Approach to Overcome Latency

The models present an elaborate and quantitative depiction of the expected average response time in regular working scenarios of a data storage system. As mentioned in the opening paragraph of this chapter, the main purpose of these models is to simulate a large deployment of the storage infrastructure with a clear understanding of how the infrastructure's smallest units behave under particular workloads. With this understanding, it is possible to interpret the details about the delay statistics of tape storage systems.

In this thesis, we consider some of the constraints of the models and propose methods to overcome excessive delays by designing novel solution approaches to the problem. The following sections present the ideas that are at the core of our efforts to achieving the goals of Tape Cloud.

3.5 Prioritizing Read Tasks over Write Tasks

3.5.1 The Concept of Priority Queues

A priority queue is an abstract data type which is similar to a regular queue or stack data structure, but where additionally, each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according

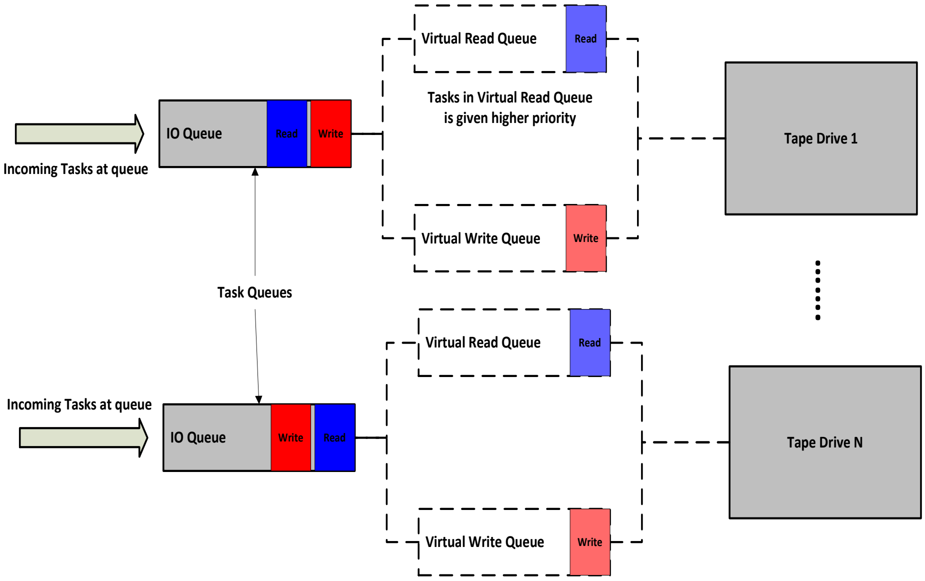


Figure 3.1: A pictorial representation of the split queuing mechanism.

to their order in the queue. A priority queue must at least support the following operations **1.** Insert with priority: Add an element to the queue with an associated priority and **2.** Pull highest priority element: Remove the element from the queue that has the highest priority, and return it.

For Tape Cloud, a Split Queue Priority Queuing technique (figure 3.1) is implemented where the read and write requests are assigned different priorities and placed in two virtually split queues. We have to handle read requests appropriately in order to reduce the average response time of data retrieval. Therefore, a higher priority is given to read requests compared to write requests. In a multi tape reader model discussed in the previous section, each tape drive is assigned a queue made up of a pair of virtual queues.

3.5.2 Split Queue Priority Queueing

From equation 3.4, we can see that a major share of the delay occurs due to tasks that are ahead of the read task in the queue. In order to reduce the overall time taken for retrieving data, an approach involving biasing between read and write tasks is used. In our case, the read tasks are given a higher preference over write tasks. We place read requests in the designated, high-priority virtual queue. When there is at least one read request in the queue, the subsequent write requests are blocked and the tape drive immediately caters to the read request after the completion of the current task. Thus the resulting time taken can be given by

$$T(Pri)_{total} = T_{total} - (N(T_{singleWrite}) + \xi(T_{unload})) + (T_{unload} + T_{singleRead}) \tag{3.6}$$

$$T(Pri)_{total} = \rho + T_{unload} + T_{singleRead} \tag{3.7}$$

where $T(Pri)_{total}$ is the total time taken when priority queueing is applied. ρ is the time spent for completion of current task and $\lceil \rho \rceil = T_{singleWrite}$ and $0 \leq \xi \leq (N-1)$. By implementing priority queuing, read tasks can be accelerated to be completed much faster.

3.6 Read Probability-Based Data Distribution

Under the circumstances of scenario 3, applying priority queuing would not significantly reduce the total response time as multiple read requests that need to be executed by the same tape drive still induce delay associated with the search and seeking processes. This thesis proposes a method to overcome this by considering the Balls into Bins problem [25][23][8].

3.6.1 Balls Into Bins Problem

We associate the issue faced in scenario 3 with a well-known computer science problem called Balls into Bins. This association helps us in refining our options for an optimal solution. The problem involves the sequential assignment of m balls into n bins by placing each ball into a bin chosen independently and uniformly at random. The questions that arise at this point are related to the maximum number of balls in any bin and the difference in the number of balls between the bin with the highest number of balls and the bin with the lowest number of balls. The gap between the weights of the most loaded bin and the average is given by

$$W_{gap} = \sqrt{\frac{m \log n}{n}} \tag{3.8}$$

where m is the number of balls and n is the number of bins.

One of the important applications of this method includes online load balancing. With the rapid growth of parallel and distributed computing, the load balancing problem has gained considerable attention during the last decade. The problem is to assign the specific requests to servers in such a way that all servers handle (about) the same number of requests. By introducing a central dispatcher one can easily achieve uniform load on the servers. Randomized strategies have been applied very successfully for the development of good and efficient load balancing algorithms. In their simplest version, each request is assigned to a server chosen independently and uniformly at random. If all requests are about the same size, the maximum load of a server then corresponds exactly to the maximum number of 'balls in a bin' in the balls-into-bins model introduced above. The problem assumes all requests to be of a similar kind in terms of efforts needed to complete them or equal priorities. In the case of Tape Cloud, we consider a similar analogy with data write requests but with a difference. Write requests can be classified based on a numeric value that is assigned to them based on its inherent property. These values are the probabilities of the data being read again once it is written to storage, a novel proposal introduced in this thesis. The varying probabilities, or in this case, weights, creates the motivation to further augment the Balls into Bins and consider the "Weighted" Balls into Bins case[23].

3.6.2 Read Probability Weight (RPW)-Based Data Distribution

Every block of data that needs to be written to tapes has a certain probability of being read again. This probability or "weight" is based on the type of application and its historic transactions with the storage system. Intuitively, it is possible to note that blocks of data with a higher weight causes a higher delay when written to tapes by the same tape drive as compared to data with lower weight (because the read requests that come in eventually still have to be queued at the same tape drive). So the motive to reduce this delay has to be to distribute the data blocks of higher weight equally among the available tape drives such that a single tape drive need not take the entire burden of heavy weighted objects. This is similar to Balls into Bins problem except that in the case of Tape Cloud, balls are of different weights. Assume that there are n types of data blocks, where $W_n = \{P_r^1, P_r^2 ... P_r^n\}$ are its respective weights. Given m tape drives $T_{drive} = \{t_1, t_2 ... t_m\}$, the RPW data distribution makes sure that

$$\forall t \epsilon (T_{drive}), (\sum_{i=0}^{k} P_r^i)/k \approxeq \varphi(W_n) \tag{3.9}$$

where $\varphi(W_n)$ is the arithmetic mean of all the elements in the set (W_n) and

$$\sum_{j=0}^{p/2}((\sum_{i=0}^{k} P_r^i)/k) - \sum_{j=p/2}^{m}((\sum_{i=0}^{k} P_r^i)/k) \approxeq 0 \tag{3.10}$$

Where k is the total number of write tasks in a particular queue t. If data originating from an application q is assigned a weight P_q, then each queue will have a weight S_q equivalent to $P_q / \sum_{n=1}^{N} P_n$ of data pertaining to application q where N is the total number of weighted tasks in the queue. No single application can have all its data written to a single location. RPW-based data distribution coupled with priority queuing not only improves average response time efficiency, but also contributes towards maintaining write throughput as it reduces the overall delay caused due to continuous blocking of write tasks by a series of read tasks.

The following excerpt shows the algorithm to assign tasks to queues using the RPW method.

The process of assigning tasks to queues are offline and are performed in batches. At any given time, the total number of tape drives available is very large considering the scale of the deployment of Tape Cloud. In that case the findTapeDrive function needs to loop through the entire set of tape drives to obtain a suitable tape drive

Algorithm 2 Read Probability Weight-Based Task Distribution

```
Require: $m$ tape drives $\Gamma_{td} = \{td_1, td_2, \cdot, td_m\}$
    $n$ write tasks: $\Gamma_{tr} = \{tk_1, tk_2, \cdots, tk_n\}$ where $n$ is many times larger than $m$
    $n$ weights of tasks: $\Gamma_w = \{w_1, w_2, \cdots, w_n\}$
Ensure: $m$ packages, each for tape drive, containing encapsulated tasks.
Assign the first task $tk_1$ to first tape drive $td_1$
for $i = 2$ to $n$ do
    $idx = placeInIdealTapeDrive(tk_i, w_i)$
    $td_{idx} = assignTask(tk_i)$
    increment i
end for
```

to insert the task into. To make the process of searching for the ideal tape drive to insert into more efficient, we introduce a method **complementary hashing** which uses hash maps of the instantaneous average RPW of tape drives to obtain the next ideal task queue to assign the task. The algorithm titled placeInIdealTapeDrive gives an idea of how the complementary hashing technique works.

3.7 The Combined Strategy

In this thesis, we use a combined strategy of Split Queue Priority Queuing and RPW-based data distribution. These strategies form the core of the functionality of the middleware that is used in Tape Cloud. The use of this strategy is compared with other commonly followed methods of data distribution in SAN and NAS-based storage systems. The elaborate result set in the evaluation section shows the benefits of the methods proposed, the use of the middleware and ultimately, the feasibility of Tape Cloud.

Algorithm 3 Function placeInIdealTapeDrive to place a task in the ideal task queue

```
Require: $m$ tape drives $\Gamma_{td} = \{td_1, td_2, \cdot, td_m\}$
    Singular Hashtable $h < ReadProbability, LinkedList < \Gamma_{td} >>$ where $ReadProbability = [0.1 \cdots 0.9]$
    Parameter: task $task$ and weight of task $w_{task}$ $W_{highest}$ the highest value of the RPW among all the queues.
    $W_{lowest}$ the lowest value the RPW among all the queues.
    $W_{average}$ the average of the interim $W_{highest}$ and $W_{lowest}$ values.
Initialize $W_{highest}$ and $W_{lowest}$ to 0.
for $i = 0.1$ to $0.9$ do
    Initialize empty linked list $linkedlist$
    Put in $h$ key $(i)$ and value $(linkedlist)$
end for
Get $W_{average} = \frac{W_{highest} + W_{lowest}}{2}$
if $w_{task} < W_{average}$ then
    Get tape drive $td_x$, head of linked list at $W_{highest}$
    Assign task $task$ to $td_x$
    Recalculate RPW at $td_x$
    Reassign $td_x$ in the $h$ to appropriate key based on RPW
    Recalculate $W_{highest}$ and $W_{lowest}$ values.
else
    Get tape drive $td_x$, head of linked list at $W_{lowest}$
    Assign task $task$ to $td_x$
    Recalculate RPW at $td_x$
    Reassign $td_x$ in the $h$ to appropriate key based on RPW
    Recalculate $W_highest$ and $W_{lowest}$ values.
end if
```

Chapter 4

System Design

Figure 4.1 is a bird's eye view of the Tape Cloud architecture. The arrows represent the direction of flow of data. We propose a hybrid middleware that performs efficient hard disk caching, data block management, data distribution and IO task scheduling. The middleware functions as an agent, arbitrating the various components in order to reduce the overhead caused by using the slower backend medium. Figure 4.2 provides the logical representation of the middleware and some of its functionalities. The solid lines show the path taken by control statements and metadata while dotted lines show the path of the actual data blocks that need to be stored on tapes. The data that needs to be written to tapes is collected and channeled suitably before it reaches its destination. IO requests are processed in batches. This helps in easy read and write of data to and from collection servers and for efficient distribution. In order to ensure efficient data transactions in the Tape Cloud on the whole, buffers constituted by faster storage media such as hard disks are used. The buffer storage tiers Figure 4.1 are named collection servers, where data that is obtained from clients is first collected before it is written into tapes. The rate at which data are removed from these buffers and written into tapes is user defined. It is based on the rate of data read requests generated by the application.

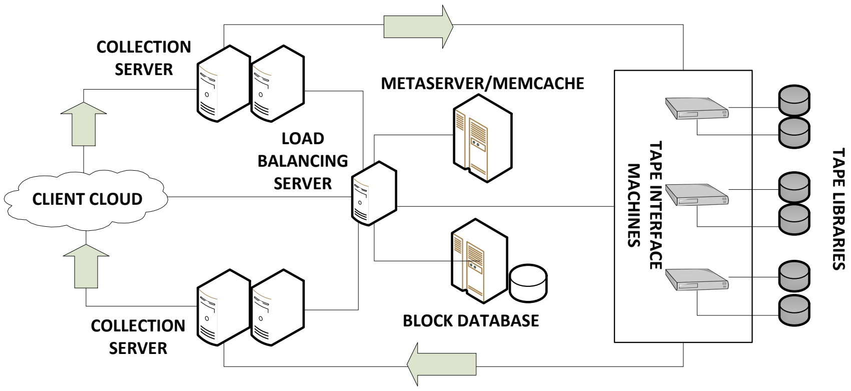


Figure 4.1: Implementation Architecture of Tape Cloud.

4.1 An Abstraction for Tape Library Hardware

The massive scale of the focus of Tape Cloud encompasses multiple drives, high speed robotics that seek, grab and load tapes to drivers at high speeds and hazard resistant tape storage space. To make data retrieval and deposition more efficient, a design proposal for the conventional driver to be split into two parts is considered. One part rewinds the subsequent scheduled tapes to the correct position and the other part is involved in reading and writing into tapes (the read and write head also has a rewinding functionality to perform small and quick seeks so it could be the addition of a rewinder in the system). This way, it is possible to pipeline the rewinding process and isolate the rewind latency from the overall read operation. The introduction of multiple synchronized tape drives also contributes in efficient reading and writing of striped data.

4.2 Data Source or Clients

The focus of Tape Cloud is consistent with most cloud-based services. It provides an efficient storage service for a variety of data. Clients who wish to archive data on Tape Cloud, run a service to deliver data to the storage collection servers (see figure 4.1). One of the features of Tape Cloud is that it allows clients to deliver data in more than one way as discussed in the subsequent section. Large data sets (which is an unavoidable attribute of archive data) can also be delivered by mailing the media itself. From the storage system's perspective, each client is tagged and labelled based on the physical attributes of the data, relative storage activity over time, space requirements and the frequency of requests for data IO that is derived from the clients. This information serves as policies which is used by the middleware to make decisions on the location of data, level of security, and distribution of data blocks and also provides the recipe to cook the read probability weight (RPW) information of data pertaining to particular clients. The data manager, with access to the central block database, updates and maintains maps of blocks of data to its physical location on tapes, in libraries and section of the data center.

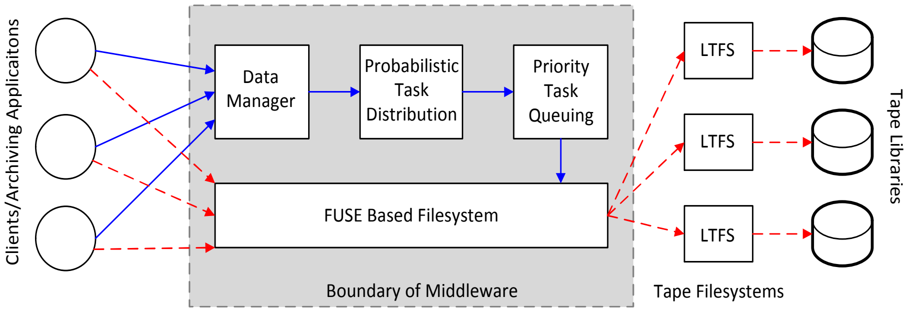


Figure 4.2: The placement and interfacing of the functional blocks of the Middleware.

4.3 Data and Resource Manager

The Data manager is the point of interaction between the clients and the storage infrastructure. More importantly, it is the interaction point between the client application and the middleware as no data is directly written to tapes without the data manager's consent. The data manager module runs on the load balancing server and manages the other parts of the middleware such as the filesystem, task queues and data distribution modules. To perform efficient management, the data manager relies on informative references to the actual client data. These references or metadata contains details about the blocks of data such as its location in the file system, size, type and RPW along with other client specific information. The metadata is used as representation of data blocks in the queuing and the distribution modules of the middleware. This prevents the overhead of moving around large amounts of data within the system.

An important task the data manager undertakes is the grouping of data stored in the middleware's filesystem to be processed in batches. The data manager employs a specific technique to pick metadata pertaining to blocks of data which are most

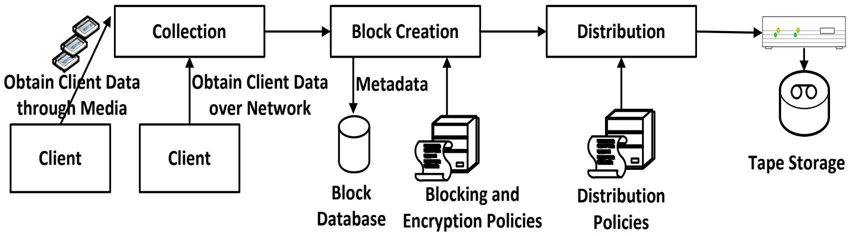


Figure 4.3: Stages and functions of each stage of the filesystem for Tape Cloud.

probable to be retrieved as a single unit from the filesystem, packs them and passes them to the data distribution module. Other responsibilities include the attestation of data deposition requests from and client and allocating suitable resources.

4.4 Multi Tier File System

FUSE [34] is a framework that helps develop the customized file system. FUSE module has been officially merged into the Linux kernel tree since kernel version 2.6.14. FUSE provides 35 interfaces to fully comply with POSIX file operations. A design a file system using FUSE is drawn up to be operated in the middleware of the architecture. The implementation presents a monolithic image of the filesystem, but internal divisions exist based on functionalities. Figure 4.3 shows the pathway taken by data to be written to tapes and the various actions taken upon it. Although distributed by functionality, the filesystem is monolithic across the storage system. The filesystem depends on external databases to maintain records of the locations of blocks of data. In order to prevent loss of data due to server failure, the filesystem performs a replication of similar data in multiple location similar to HDFS [32].

The filesystem acts as a temporary data holding space while the data is being audited. The mount point of the filesystem can be remotely created on the client side or using the dedicated interface on the server side. This provides users with more than one way to upload data. Some of the common ways of shipping data to an archiving or backup site and how the filesystem developed for Tape Cloud can support it is discussed below.

- **Subscription for HDE (Hard Disk Exchange:)** The HDE continues to be a popular method of data backup for many organizations set up in areas with inefficient data networks. The HDE model includes specialized backup service providers who ship usable hard disks to clients and collect hard disks with data that need to be archived. This rotation of hard disks, although poor on security, ensures that either the client or the backup service providers maintain a copy of the data at any point of time to avoid loss due to system failures. The Tape Cloud file system allows docking stations for hard disks and treats them as new file system mounts. So Tape Cloud supports and is completely compatible with the HDE model.

- **Data transfer over Inter-Network:** An efficient and secure wide area network (WAN) can allow the convenient transfer of data over the internet. Although an expensive option which demands extremely large bandwidth, many organizations have remote back up locations where data is periodically transferred over the internet. Tape Cloud offers a client system mountable networked file system (NFS) interface. Backup and archiving applications need only read or write data to a particular path in centralized servers or every individual

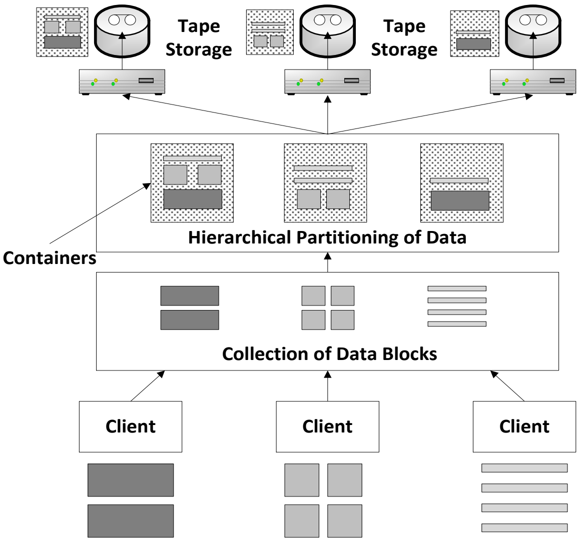


Figure 4.4: Pictorial representation of the stages and the data at each stage of the filesystem for Tape Cloud.

computer. The data are transferred to Tape Cloud for archiving or backup in the background.

- **Shipment of Tapes:** One of the other famous methods of ensuring backup of data is by shipping magnetic tapes to remote locations for preservation. This method is mainly preferred to avoid loss of data through natural calamities. Tape Cloud maintains the compatibility by operating conventional tape libraries which allow inclusion and accounting of standalone tapes into the system.

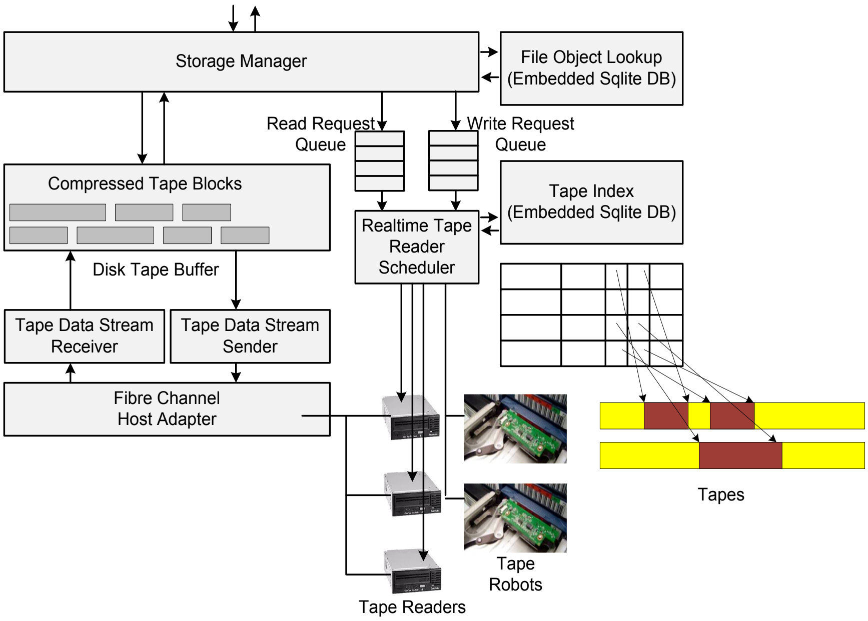

Storage Manager
File Object Lookup (Embedded Sqlite DB)
Read Request Queue
Write Request Queue
Compressed Tape Blocks
Tape Index (Embedded Sqlite DB)
Realtime Tape Reader Scheduler
Disk Tape Buffer
Tape Data Stream Receiver
Tape Data Stream Sender
Fibre Channel Host Adapter
Tapes
Tape Robots
Tape Readers

Figure 4.5: Data flow representation within the functional units of Tape Cloud.

4.5 Data Handling

Figure 4.5 shows the possible paths that data takes within the Tape Cloud infrastructures. Data, in this case includes the metadata of the actual data blocks which move to and from the block databases and the blocks of data that need to be archived or backed up.

4.5.1 Data Acquisition

The previous section elaborated some of the possible methods and ways in which data that needs to be backed up on tapes are delivered to Tape Cloud. These methods can be aptly selected based on the size and the frequency of the backup that needs to be performed. The client side has an application that requests a data upload session from the load balancing server which then allocates the required amount of space in the collection server. The client directly uploads its data to the specified location. Another data acquisition protocol is designed for very large data sets where data is received in the form of storage media units itself. The collection servers, in this case, provide a docking interface functionality where the media can be mounted directly on the servers. The extracted data is treated in the same way as uploaded data.

4.5.2 Data distribution to Tape Interface Machines (TIM)

Once the data has been segmented at the collection servers, it is audited, tagged for identity, recorded in the block database and partitioned to be written to tape. The next set of data blocks to be written to tapes are then placed in virtual containers. Each of these containers are said to be RPW weight normalized so that the difference in weight between each of these containers is said to be minimal at the time of creation of the containers. A software daemon that processes data in batches is triggered, which collects the particular blocks of data and then encapsulates them into packages, each of which is then delivered to a tape drive to be written to tape. Another consideration that is made when data needs to be delivered to a particular tape drive is its current IO queue depth.

4.5.3 Distributed Tape Write/Read

Like any other cloud service, the architecture is complex with a large number of tape drives. In order to increase the IO bandwidth, parallel reading and writing of data into multiple tapes can be envisioned. This consideration, however, needs to be analyzed from more than one perspective. The larger the number of subdivisions, the larger the latency induced by seeking for the data. Yet, a huge single read or write to tape can cause an increase in the waiting time of other requests. Many task scheduling algorithms have been designed[29] to improve performance of IO in tapes and better utilize drive resources, but little has been done with the latest LTO5 tape drives. Based on the latency test results of the tape device, it can be noted that the seek time of the robotics takes a considerable longer time than reading or writing of data to the disk. In order to improve time efficiency, the I/O requests are scheduled such that each request spends the least amount of time performing seek operations and longer periods of read or write operations.

4.6 Read Probability Weight-Based Data Distribution

The analysis of latencies that is performed leads to induction of a technique where some of the delays are preempted before data is written to tape. As discussed in section 3.4, this is to ensure that a small group of task queues does not take the burden of a large number of discontinuous read tasks. The probabilistic data distribution module is an important part of the middleware that distributes blocks of data to the tape interface machines based on a particular weight associated with the data. The weight or the read probability weight (RPW) is the probability of the block of data being read once written onto tape. The probabilistic data distribution module is designed to obtain the RPW by two ways. It can be enclosed in the metadata that is handed down by the data manager. The other avenue that can be taken to deduce the RPW is over time, when the middleware notices that there are some blocks of data that have undergone access in a manner inconsistent with its knowledge about the RPW. In this scenario, the middleware updates the RPW of data incoming from the client and adapts to the workloads of different clients over time. After the references have been assigned specific tapes or drives, the references of data blocks are handed over to the task queuing module of the middleware.

4.7 Task Queueing

The large-scale operation of the storage system involves the use of multiple tape drives. The entire tape storage facility is divided into sections, each of which can be serviced by a tape drive. Each of these tape drives has an exclusive queue assigned to it which holds the IO task to be performed on tapes which are in its logical vicinity. These task queues are maintained and used by the middleware and should not be confused with the ones that are used by the storage media or drivers. One of the approaches to decrease the delay in retrieving data is to prioritize between the read and write requests as discussed in section 3.4. The task queuing module caters to this need by assigning each tape drive with two virtual queues, one each for write and read requests. Read requests having higher priority over write requests are granted resources immediately after the completion of the current task regardless of the depth of the write queue. After completion of the read task, the system continues with the execution of other tasks in the write queues. Assuming an efficient distribution of data, the task queuing module ensures that read tasks are performed under strict time constraints while maintaining acceptable standards of throughput for write tasks. The task queues provide periodic feedbacks to the data manager about the overall time taken in performing tasks associated with a specific batch. This feedback is used by the data manager to assess the overall performance of the data distribution module and the distribution parameters in the system.

4.8 Security in Tape Cloud

Many prospective cloud storage users agree that security concerns in general and a lack of appropriate security tools in particular are among the top factors preventing their organizations from using cloud storage more pervasively[21]. Those concerns are legitimate, and CIOs and IT managers must address them when deciding whether to move to cloud-based storage. Public clouds are where, by definition, users must be willing to have their data reside side-by-side with data belonging to possible competitors and other completely unknown parties. Additionally, disk arrays in the cloud use RAID algorithms to break up and spread data at the block level across multiple disks for resiliency. In multi-tenant cloud environments, the potential result is that pieces of data from multiple cloud customers could commingle on the same drive. It also means that a drive failure could affect a large number of users. Additional software and possibly encryption are required to ensure that a particular customer's data is isolated secure. Conversely, in a tape environment, each tape cartridge is a separate object. The customer or cloud provider has control over which files go onto which tape. In addition, the encryption and WORM capabilities of tape provide security for data at rest, as they are fundamental to delivering a secure cloud archive strategy. Tape is also easily movable/removable. The removable/vaulting capability of tape as it relates to security centers on the fact that it can be literally locked down to prevent unauthorized access. That kind of security is not generally available with disk storage, which is one reason why storage service providers such as Iron Mountain use tape technology as the foundation for many of their offerings.

Chapter 5

Characterizing Archive Workload from Traces

Testing storage systems is a widely performed activity by the IT department in every organization. They evaluate their current storage infrastructures to judge its abilities to cope with the organization's needs. Infrastructure administrators perform periodic tests to report the abilities and shortcomings of the current systems so that investments in newer and advanced technologies can be made. There are many ways of evaluating a storage system stack out of which two methods are very popular.

- Large amounts of data pertaining to the operation of storage servers are collected and analyzed. The method highlights anomalies and unusual behaviors in the system but is also a very tedious process which involves the efforts of a dedicated team of highly skilled professionals and tools.

- Storage systems are run on artificially generated workloads or benchmarks. These benchmarks can be used for comprehensive testing of system performance or validation of individual units. A globally accepted verity of benchmarks involves the acceptance of results obtained by running benchmark workloads on an as-is basis. Benchmarks serve as a common set of parameters used to judge different systems or methods so that an ideal choice from the available options can be made.

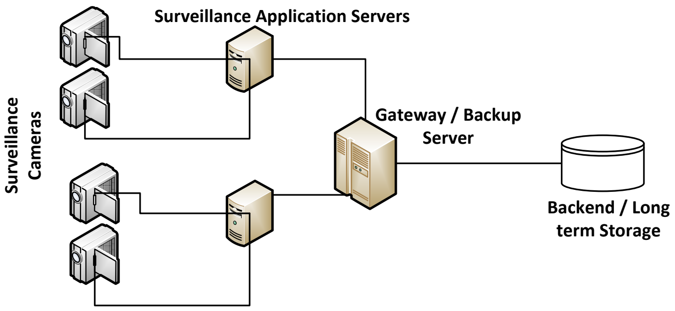
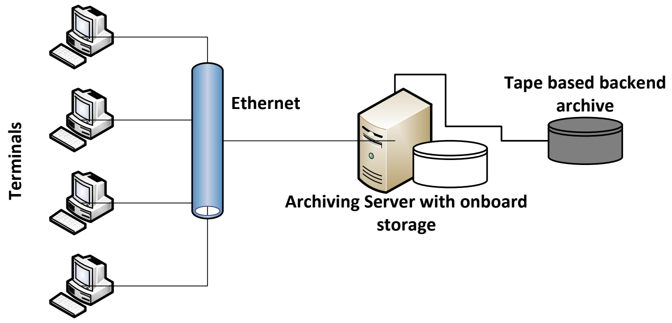
When we look for effective benchmarks to evaluate archiving and backup-based storage systems, we realized that most of the benchmarks that are available are either outdated or considers parameters for the modern hard disk-based backup and archiving backend. The lack of availability of benchmarks leads to the need for a novel approach for evaluation. A combination of the above mentioned approaches is considered to create a set of custom workload traces that are congruent with the workloads generated in conventional backup and archiving systems. The process is initiated by collecting workload traces from backup and archiving systems at various stages in the software stack. The traces are analyzed and then recreated using workload generating tools for testing Tape Cloud. The new results are compared with the results that are obtained when the same workload has been tested on other backup systems to note the difference that Tape Cloud brings to the table.

5.1 Characterizing Archive Workload from Traces

While a number of articles provide benchmarks and suggest methods to evaluate various aspects of storage such as the medium, queues, IO characterization[6] and filesystem [5][11], there has been a comparatively limited literature on evaluating performance of archival storage systems. Kavalanekar et. al.[18] provide elaborate results on storage workloads from production windows servers. But the variation in workload type between non archival and archival storage varies as suggested by Lee et. al. in [19], who make an attempt to create benchmarks. But their work is limited to providing a better understanding of the type of files and its sizes than provide a complete set of results. Another important contribution has been provided by Wallace et. al.[37] for EMC production servers. Although a large number of aspects have been covered, the impact of different types of archiving and application level transactions with the storage have not been projected.

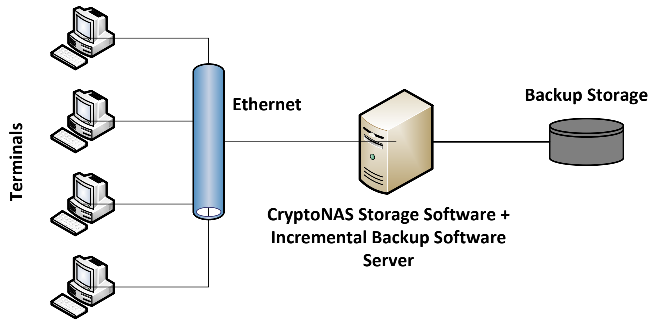
In order to perform a bias free evaluation of the middleware, it is subjected to a workload that has been characterized by traces obtained from live archiving applications. The traces are collected from the archiving infrastructure of IVigil, a company that provides video surveillance services to a local client base. The backed up data usually includes surveillance videos, security related data, virtual disks and documents that are wielded by the company on a daily basis. Aspects which are important to the working of the middleware such as the rate of requests with respect to time, inter arrival time of requests and a comparison of the rate of read to write request are recorded and analyzed. Some of the characteristics of Table 5.1 and

Table 5.1: Applications Contributing Workload Traces for Evaluation of Middleware

<table>
<tr><th>Sl. No.</th><th>Archiving Type</th><th>Description</th></tr>
<tr><td>1</td><td>Periodic Full Backup</td><td>10 disk array on 3 networked attached storage (NAS) servers archiving surveillance video and security data. Videos and related information is collected from local systems once every 24 hours through a customized asynchronous pull server-based system. High churn rate.</td></tr>
<tr><td colspan="3">
</td></tr>
<tr><td>2</td><td>Periodic Full Backup + LRU Archiving</td><td>Application archived least recently used support files on larger disk-based backend storage with smaller churn rate. Periodic archiving of data on individual terminals are performed by servers with small internal fast access storage and a larger remote storage (usually tape-based system).</td></tr>
<tr><td colspan="3">
Terminals
Ethernet
Tape based backend archive
Archiving Server with onboard storage

</td></tr>
</table>

Table 5.1 (continued)

<table>
<tr><th>Sl. No.</th><th>Archiving Type</th><th>Description</th></tr>
<tr><td>3</td><td>Incremental+Full Backup</td><td>Incremental backup of hard disks and virtual disks at the end of every login session and periodic full backup of 22 computers on hard disk-based NAS storage running CryptoNAS software. Files are of comparatively smaller size with high data transaction rates even at remote tape storage.</td></tr>
<tr><td colspan="3">
Terminals
Ethernet
Backup Storage
CryptoNAS Storage Software + Incremental Backup Software Server

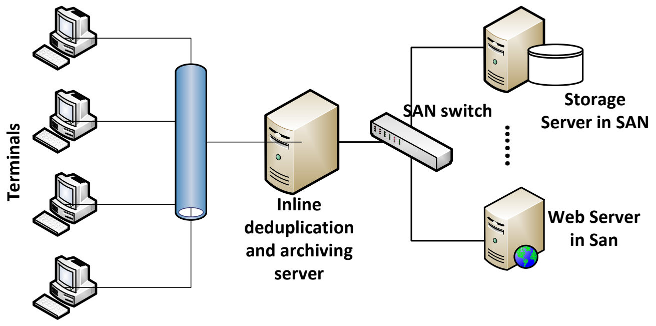
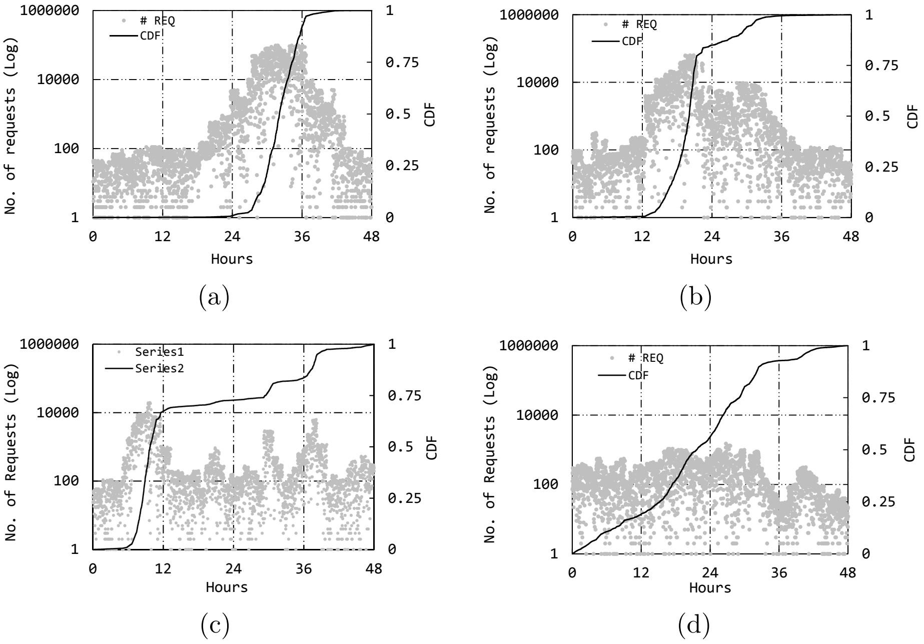
</td></tr>
<tr><td>4</td><td>Non Periodic Mirroring Backup</td><td>Shared documents backup on request on dedicated LUN in SAN infrastructure with a duplication client running on an inline server. Number of overall requests are low considering backup is triggered on request.</td></tr>
<tr><td colspan="3">
</td></tr>
</table>

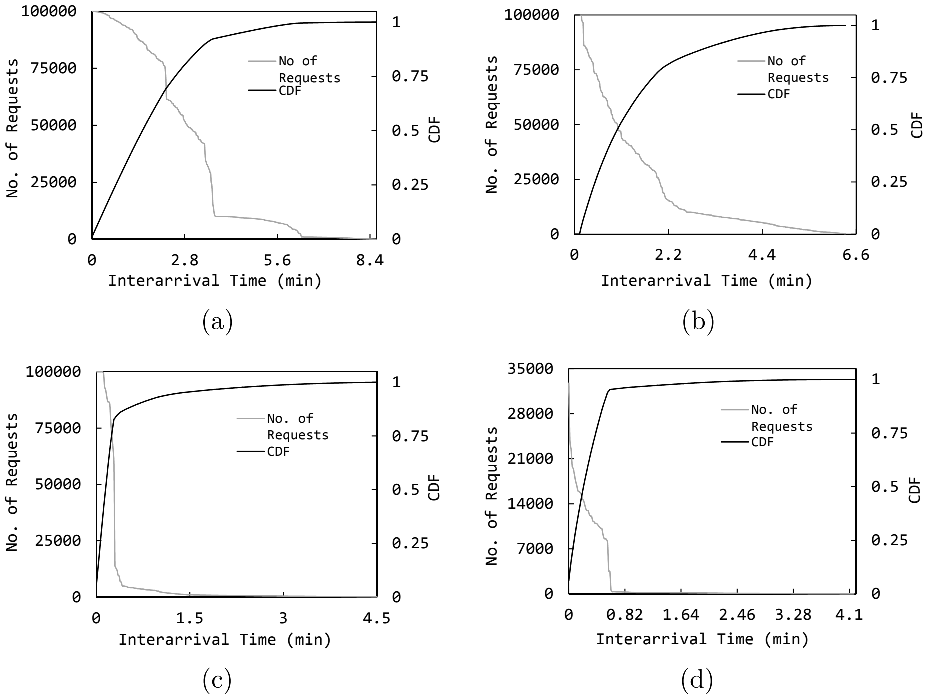
Table 5.1 provides some information about the characteristics of the infrastructure and applications.



(a) (b) (c) (d)

Figure 5.1: The total number of requests generated by archiving applications.

The applications show characteristics that prove common beliefs about archival data wrong[19]. The application level traces help in understanding the frequency with which IO requests are generated. This serves as a clear indicator of how backup types differ from each other. The filesystem level traces provide a defined understanding of what each IO request generated by the application demands. Each of the applications varies in infrastructure so it is important to co-relate traces obtained to reflect a common operation at each stage. The following are the results of the characteristics extraction from the traces.



(a) (b) (c) (d)

Figure 5.2: Interarrival(IA) time of requests generated by archiving applications.

Figure 5.1 shows the total number of IO requests generated by the archiving applications 1(a), 2(b), 3(c) and 4(d). The record for the number of requests are collected and generated at the application level for discrete read or write requests to the underlying filesystems. Interarrival(IA) time of requests generated by archiving applications 1(a), 2(b), 3(c) and 4(d) is shown in figure 5.2. Application, type of data, file sizes and temporal locality are some of the factors influencing inter-arrival time. The asynchronous nature of some applications and storage system software also affect IA time.

As discussed earlier, random IO is responsible for the major share of the delay in a

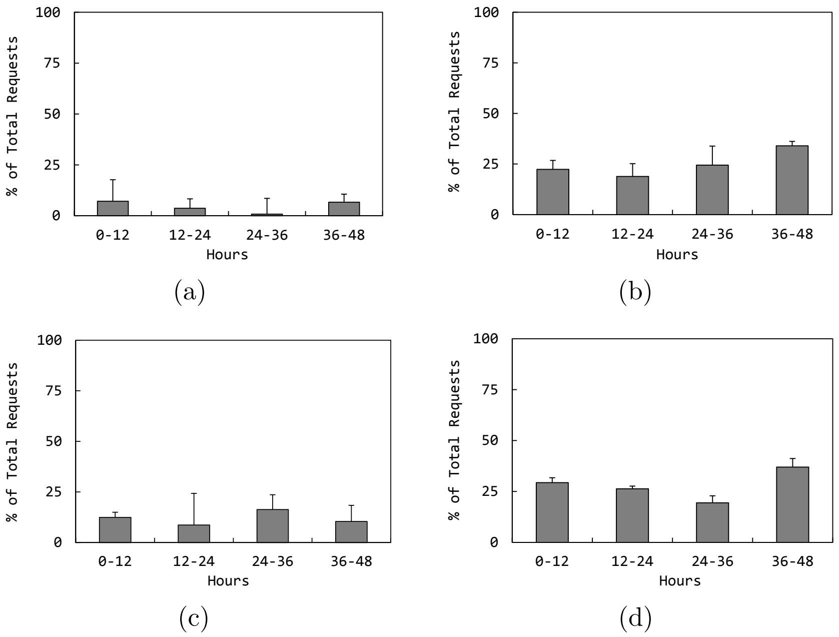
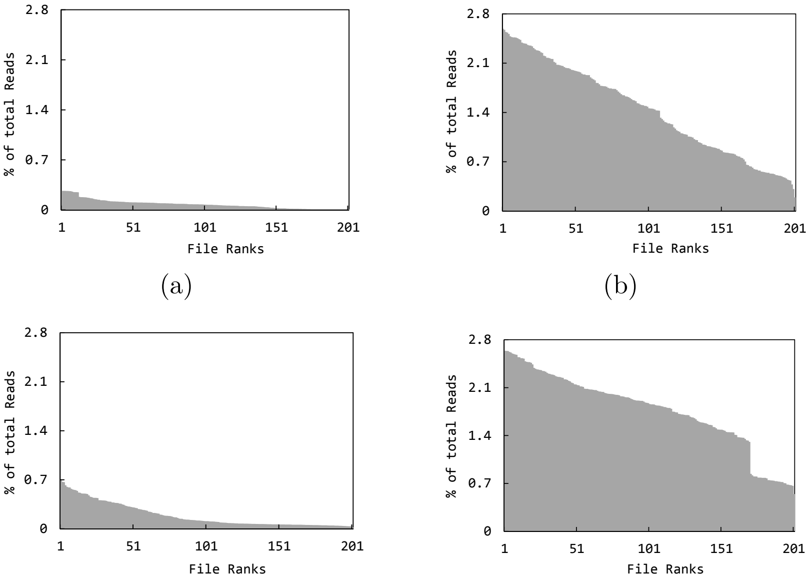


Figure 5.3: Average number of read requests as a percentage of the total IO requests

tape infrastructure. Figure 5.3 is the average number of read requests as a percentage of the total IO requests in 12 hour buckets by archiving applications 1(a), 2(b), 3(c) and 4(d). The whiskers show the maximum percent of read requests received during the particular 12 hour interval. Figure 5.4 shows the total number of read requests for the 200 most frequently accessed files as a percentage of the total read requests received by archiving applications 1(a), 2(b), 3(c) and 4(d). Figure 5.3 and Figure 5.4 provide a better understanding of the number of read requests obtained as a ratio of write requests, or in other words, the relation between the number of read and write requests which is a critical quantity in the modelling process. It also shows how frequently 200 individual "hot" files are accessed within the storage system as a function of all reads that is generated by an application.



(a)

(b)

(c)

(d)

Figure 5.4: Total number of read requests for the 200 most frequently accessed files.

5.2 Workload Modeling and Generation

Existing workload generation tools are not flexible enough to generate workloads that vary in three key dimensions - variations in the amount of load, variations in the mix of operations performed by clients (e.g., changes in read vs. write or customer usage-patterns), and variations in the popularity of the data accessed, i.e., data hotspots. These dimensions are needed for making resource allocation decisions for storage systems. There have been many projects in developing synthetic workload to test storage systems such as in [10][14] which depend on models created by Markov chains of states and virtualized environments. The commendable results focus on workloads that vary from archiving workloads. A workload using Vdbench[2] in order to test the middleware's and Tape Cloud's performance is synthesized.

5.2.1 VDBench: The Workload Generator

Vdbench is a disk I/O workload generator that is used for testing and benchmarking of storage products. The objective of Vdbench is to generate a wide variety of storage I/O workloads, allowing control over workload parameters such as I/O rate, LUN or file sizes, transfer sizes, thread count, volume count, volume skew, read/write ratios, read and write cache hit percentages, and random or sequential workloads. This applies to both raw blocks on the medium and filesystem files and is integrated with a detailed performance reporting mechanism eliminating the need for the Solaris command iostat or equivalent performance reporting tools. Raw I/O workload parameters describe the storage configuration to be used and the workload to be generated. The parameters include General, Host Definition (HD), Replay Group (RG), Storage Definition (SD), Workload Definition (WD) and Run Definition (RD) and must always be entered in the order in which they are listed here. A Run is the execution of one workload requested by a Run Definition. Multiple Runs can be requested within one Run Definition. The Parameter files serve as the input for generating the workload. The parameter files entered will be read in the order specified. All parameters have a required order as defined here: General, HD, RG, SD, WD and RD, or for file system testing: General, HD, FSD, FWD and RD.

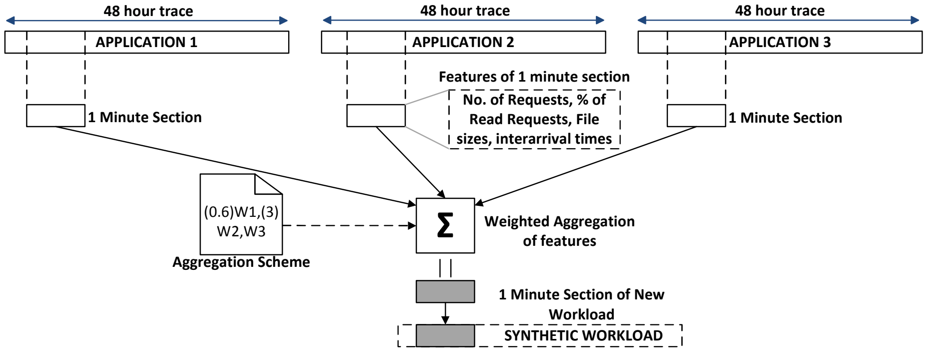


Figure 5.5: The process of synthesizing a workload based on previously analysed application traces.

5.2.2 Generating Archival and Backup Workloads

The workload generator is carefully designed by performing a sectional analysis of the results obtained in the real archive workload traces. The real time workloads are spliced on the basis of a user defined time interval and the features of each division such as the number of requests, types of requests, file sizes and inter-arrival times are extracted. The newly created workload is essentially a time-based, weighted aggregation hybrid of the workloads. Since Tape Cloud is a cloud-based storage infrastructure, it needs to support multiple types of workloads from more than one application. Although the applications considered for synthesizing workloads are singletons in their specific domains and are not related to each other, the experiments consider a simulation where all applications are made to use single backend storage for archiving and backing up of data. This way, experiments conducted in this section are an aggregated synthetic product of all the workloads. The weighted aggregation provides the flexibility to produce workloads in any combination of amounts of the given traces. It depends on a workload aggregation scheme provided by the user

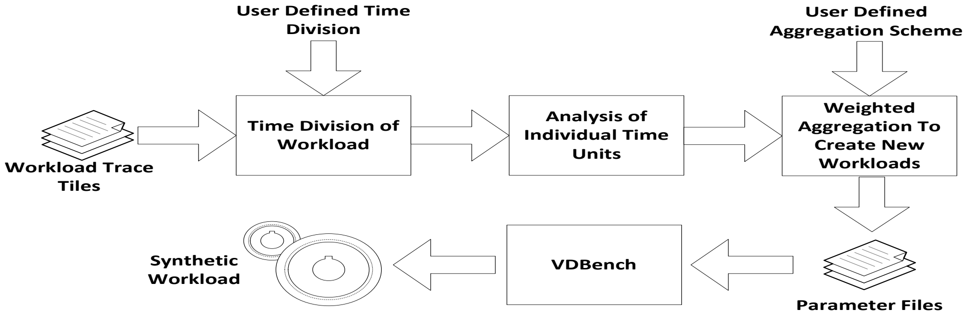

User Defined Time Division
User Defined Aggregation Scheme
Time Division of Workload
Analysis of Individual Time Units
Weighted Aggregation To Create New Workloads
Workload Trace Tiles
Synthetic Workload
VDBench
Parameter Files

Figure 5.6: The work flow of the synthesis of the workload through weighted aggregation process.

which generates a Vdbench script based on the input. For example, an aggregation scheme (W1,W2,W3,W4) would produce a workload from the 4 participating workloads in equal proportion, ((2)W1,(0.5)W2,W3) would produce twice the amount of workload 1, half the amount of workload 2 with no change to workload 3 and no trace of workload 4. This type of modelling has proven to provide a wide range of options for generating workloads. The focus of this paper being the evaluation of the middleware, an equal proportion workload to record the difference in performance is used. Figure 5.5 gives an informative sketch of the process that is followed in generating the synthetic workload. The traces are divided based on a user defined time interval, features extracted and an aggregation performed to create a block of the new artificial workload.

5.2.3 Workload Generation Flow

Figure 5.6 shows the workflow of the generator. The original workload trace files are parsed and the information is assembled as multiple units of time. This means that a workload that is collected for 48 hours is divided into 2880 values, each with the time stamp during which it was generated. Each of these units is fed into a workload parameter module which creates parameter files based on the aggregated values obtained from the workload divisions. The parameter module then uses the user defined aggregation scheme to multiply or create different configuration of the original workload patterns. The new workload configurations are written into different parameter files which are used by the VDBench tool to create the synthetic workload. The user has control over the aggregation scheme and the time units.

Chapter 6

Performance Evaluation

This section presents the experimental results on the benefits of using Tape Cloud. The evaluation of Tape Cloud is analogous with the goals that have been previously discussed. The results of the evaluation exhibit the benefits of Tape Cloud in terms of monetary savings and also the overall performance of the technology used. In chapter 2, some of the individual perks of using components such as magnetic tapes have been highlighted along with some of its drawbacks. The implementation of Tape Cloud focuses on preserving some of the advantages while attempting to overcome the drawbacks to the best possible extent at minimum costs. This chapter is divided into two parts. The first section focuses on the technical improvements of the methods in comparison with the conventional practices followed in most storage infrastructures. The conventional methods serve as cases where the improvements, in this case the middleware, is absent. Effectively, the evaluation of Tape Cloud is the evaluation of the performance of the middleware. In the second section, some of the savings

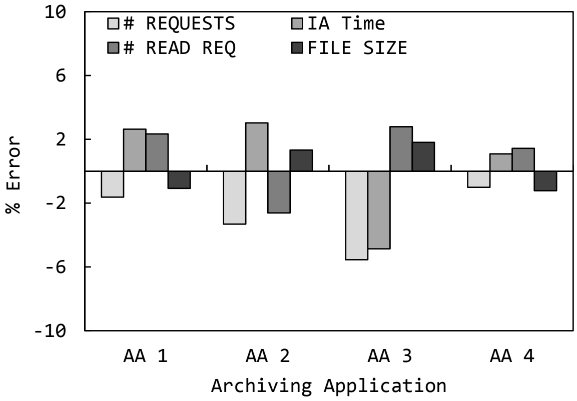


Figure 6.1: The difference or error % between the actual and synthetic workloads used in the experiments.

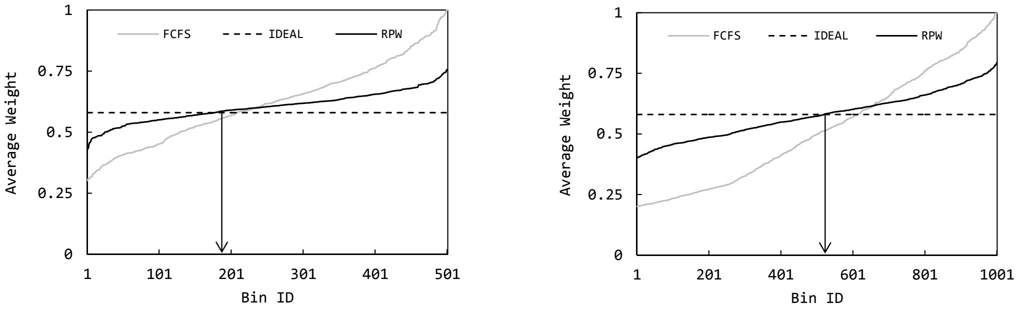
achieved by both the storage service users and storage service providers are discussed. The assessment includes cost reduction of data storage, ownership and operation of Tape Cloud.

6.1 Synthetic Workload Generator

An evaluation using the models and synthetic workload created on the basis of the actual archiving workloads is performed. The performance of the middleware and its contribution in achieving the goals to minimize average response time and efficient data distribution, are assessed in the absence and in the presence of the middleware. In the former case, a commonly preferred way of task and data distribution such as First Come First Serve (FCFS) with Round Robin and Application specific task queuing techniques are made use of, which are among the commonly opted methods in large-scale storage systems. As mentioned in chapter 3, section 3.5, the priority queuing technique has a few drawbacks which are overcome with the RPW Data distribution method. First of all, it is important to check for inconsistencies in the synthetic workloads as compared to the real time workloads obtained from traces. Figure 6.1 gives the error percentage of the synthetic workloads.

6.2 Read Probability Weight-based Data Distribution

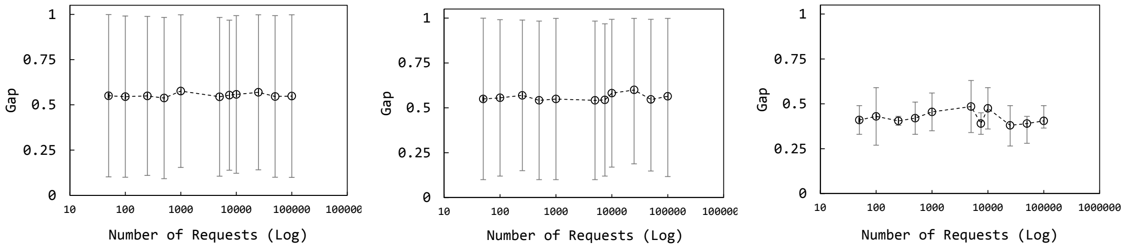
The novel idea of preempting delay caused due to the large number of read requests especially in a system like Tape Cloud calls for preliminary evaluation of the technique. RPW considers the probability of a block of data being read once written to tapes and distributes blocks based on this probability. To verify the correctness of the assumption and theories presented in this document, 10000 random weighted objects are considered and distributed into bins. Two tests are performed, where each has 500 and 1000 bins. This emulates blocks with different probabilities that need to be assigned to different tape drives. Figure 6.2 shows that RPW offers a distribution that is closer to the ideal case than other approaches like FCFS in both cases. As shown in the figure, compared to FCFS, RPW offers a higher convergence in the ideal case. Here Fig(a) is with 500 bins and Fig(b) is with 1000 bins. The arrow points to the queue ID which serves as the point of distribution balance. The technique used for RPW ensures that a balance of read probability of all the tasks assigned to a particular task queue. The result of this is the minimum difference in the average of the RPW of all the task queues.



(a) (b)

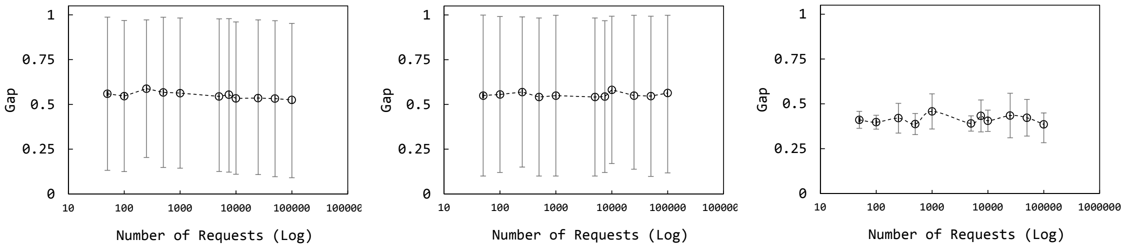
Figure 6.2: Verifying the correctness of the RPW approach.

In evaluating the RPW using the synthetic workload, two cases are considered where there are 500 tape drives (figure 6.3) and 1000 tape drives (figure 6.4). A comparison of RPW with FCFS and Application Specific Queuing which distributes data blocks generated by specific applications to specific queues is made. The application specific approach has clear boundaries between queues for each application in the system. When the number of total requests generated by the synthetic workload are varied, a more efficient distribution is provided by RPW where the gap between the queue with the largest average weight and the queue with smallest average weight is much lesser than that of the other approaches. The whiskers show the largest and smallest average weights of queues. The gap between the average weights of the heaviest and lightest queues for different number of requests for 500 queues. FCFS (a) and Application Specific Queuing (b) show inefficient weight distribution as compared to RPW (c).



(a) (b) (c)

Figure 6.3: The gap between the average weights of the heaviest and lightest queues for different number of requests for 500 queues.



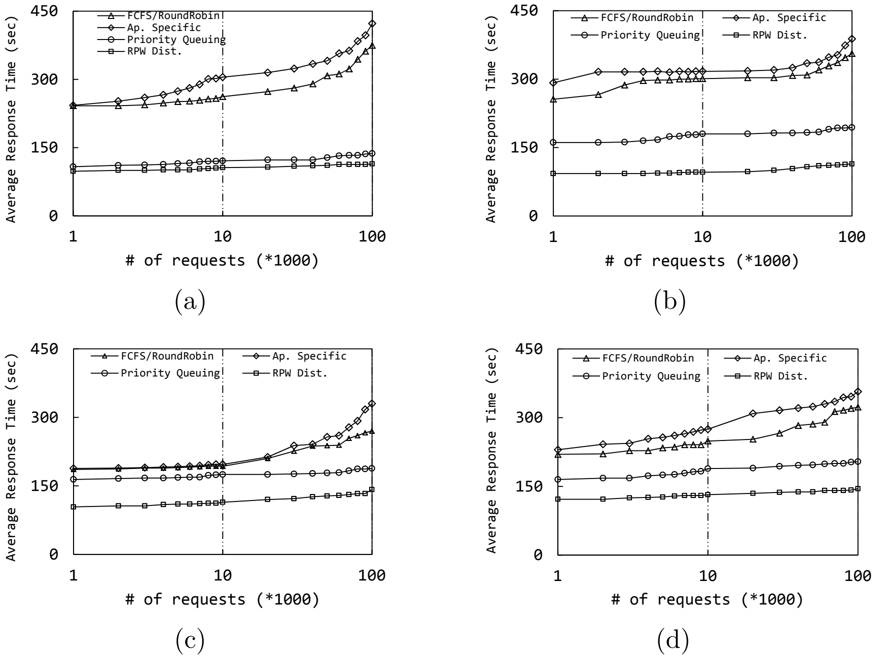
(a) (b) (c)

Figure 6.4: The gap between the average weights of the heaviest and lightest queues for different number of requests for 1000 queues.

6.3 Average Response Time for Read Requests

The use of RPW-based data distribution helps in avoiding long stretches of read operations that is localized to a small set of task queues. As shown in section 3.2.3 of chapter 3, concentrating a large number of read requests at a particular task queue can induce large amounts of delays through the load, unload or the seek operation as not all read operations are localized to a single tape of a tape drive. By using the RPW-based distribution, blocks of data with varying probabilities of being read are distributed evenly over the available resources. This reduces the aggregation of read requests at a particular task queue. This in turn reduces the average delay caused at each of the queues. When Tape Cloud is tested with the synthetic workload, the absence of the middleware leads to the use of conventional data distribution and queuing techniques such as FCFS, Round Robin and application specific queuing of tasks. But with the middleware and enhanced task management, there is an overall reduction in the response time for read tasks generated by every application as shown in figure 6.5. The average response time of read requests under the synthetic workload is shown for application 1 (a), application 2 (b), application 3 (c) and application 4 (d). Note the clear difference and the reduction of the average response time for each of the applications. Also, RPW-based data distribution offers a very small rate of increase of response time even over larger variations of the number of requests. The graphs have Log values in X axis which show the rate of change of average response time when number of requests are varied and the RPW have negligible rate of change of response time even for a large number of requests.

One of the notable differences that can be seen in the traces of the four applications is the variation in the number of requests over time. Theoretically, the induction of RPW-based data distribution along with priority queuing must make the average response time immune to the number of total number of requests. It is apparent from figure 6.6 that, along with having the smallest response time, the combination of priority queuing and RPW distribution provides a nearly constant response time over the entire period of the test, making it independent of other requests. Applications 1 and 2 are considered because application 1 has the highest write requests and application 2 has highest read requests. Compared to the other methods such as FCFS and Application specific Queuing, RPW-based data distribution maintains



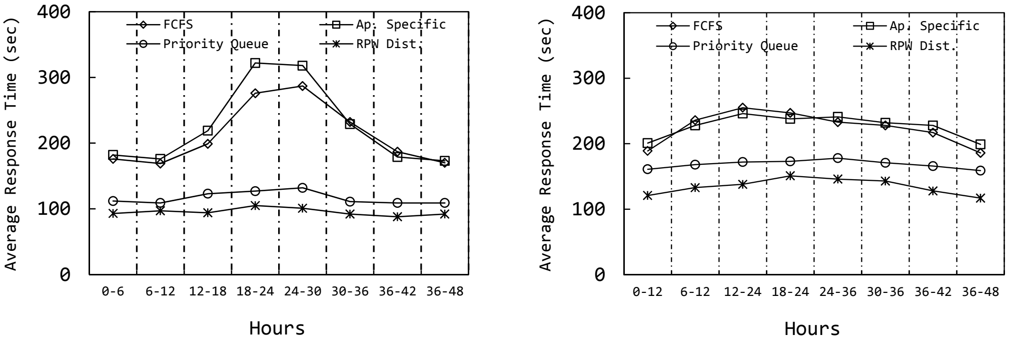
(a)

(b)

(c)

(d)

Figure 6.5: The average response time of read requests under the synthetic workload (Based on number of requests).



(a)

(b)

Figure 6.6: Time-based average response time for application 1 (a) and application 2 (b).

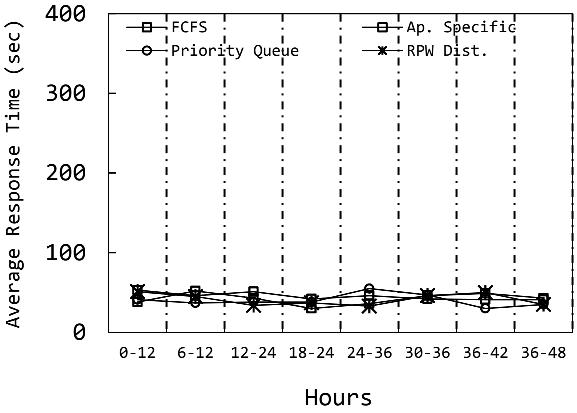
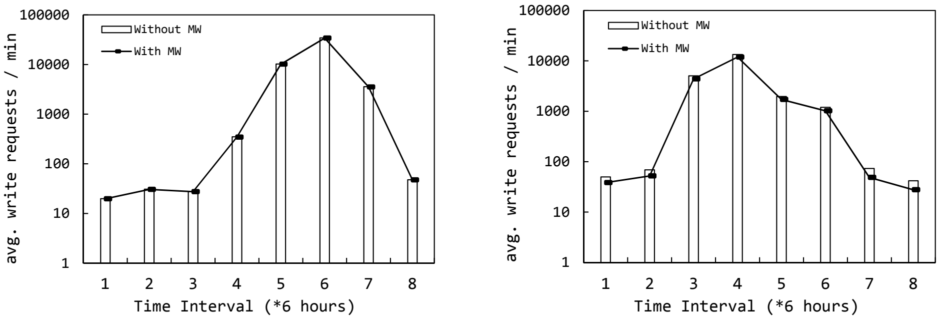


Figure 6.7: Time-based average response time for application 3.

a stable average response time regardless of the density of the workload. A similar analysis for application 3 is performed. In figure 5.2, it is apparent that requests generated by the application 3 have small inter arrival time. Tape Cloud uses collection servers (figure 4.1) which collect data from clients and stores them until it is processed to be written into tapes. The read requests issued by application 3 arrives before the data can be written to tape so data is fetched from the collection servers which is mainly hard disk-based temporary storage. Thus the fetching time of data is much lesser equal in all cases as shown in figure 6.3.

6.4 Preserving Rate of Write Task Execution

In keeping with our goals, the middleware is tested for its notoriety to bring about a negative impact on the write task completion rate of the workload. Figure 6.8 provides a comparison of the write performance before and after the deployment of



(a) (b)

Figure 6.8: The difference in write throughput with and without the middleware.

the middleware. The difference in write throughput with and without the middleware for application 1 (a) and application 2 (b). Although small differences exist, the middleware successfully provides a nearly equal write rate to all applications. The test includes cases that present extreme scenarios such as application 1 which has the highest write requests and application 2 which has the highest read requests for the aggregation scheme in use and it is very clear that, along with dutifully improving data retrieval efficiency, the middleware also maintains that similar justice be done to write tasks as well. There is only a negligible reduction in the number of write tasks performed per minute in both cases proving the abilities of the middleware.

6.5 Improved Cost Efficiency

In keeping consistent with the primary motives and the goals of the project, an evaluation of the improvement in terms of monetary savings needs to be performed.

The evaluation performed in this respect involves the calculation and recognition

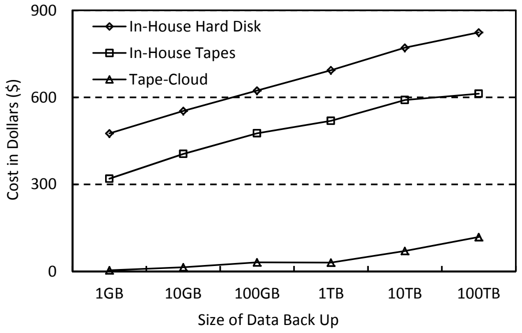


Figure 6.9: Comparison of cost for service users for different sizes of data.

of per aspect costs incurred by the service user and the service provider. The parameters such as cost of energy needed and the cost of media are consistent with the current, local commercial values. Thus, on the service users' side, some of the results obtained include the cost of using Tape Cloud for specific sizes of data backed up and the length of time of the data backup. Similarly, the results obtained on the service providers' end include, most importantly, the improved cost of operation.

6.5.1 Tape Cloud Costs for Service Users

Figure 6.5.1 shows a comparison of costs of backing up different sizes of data on Tape Cloud against having an in-house tape-based backup infrastructure and an in-house hard disk-based infrastructure. The in house tape-based infrastructure has a constant component of cost of ownership that is added to the cost of storing data. This external component is applied to all in house infrastructure deployments. Figure 6.5.1 shows similar results when projected for the time period for which the data has been backed up for the three types of infrastructures.

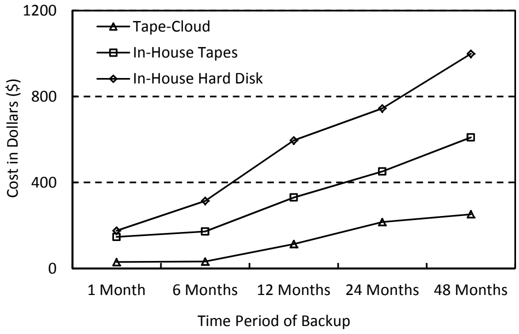


Figure 6.10: Comparison of cost for service users for different periods of 1TB data backup.

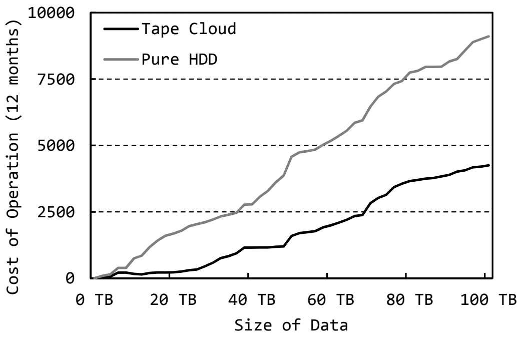


Figure 6.11: Cost of operation for service provider over a period of 12 months.

6.5.2 Tape Cloud Costs for Service Providers

Figure 6.5.2 shows the cost of operation for storing various sizes of data on Tape Cloud as compared to other forms of backup and archival storage mainly constituted by hard disk for a period of 12 months. It is clear from the diagram that Tape Cloud offers much lower operating cost which benefits not only the customers in the way of low cost online data archiving and backup solutions, but also increases profitability and scalability for service providers.

Chapter 7

Conclusion

This thesis studies some of the current practices that are followed for archiving and backing up large amounts of data. After serving as the preferred media for data backup, tapes have now improved its position as the preferred choice for long-term data retention in the cloud. A particular area of interest is the costs incurred in the long term preservation of data. Some of the common system characteristics of data storage systems such as storage media, data distribution techniques and deployment details is analyzed. The shortcomings that entail the current systems in terms of high infrastructure costs, high operating costs and cost of ownership is noted and the focus of the research is shifted to finding feasible and economical solutions to these problems. The solutions that are considered include the examination of the cloud storage models and exploiting the economical nature of magnetic tapes. The bottom line is that tape in the cloud offers a significantly lower total cost of ownership (primarily by leveraging low $/GB and significantly lower energy costs), better reliability, longer life, and as good or even faster streaming performance than today's disk alternatives[21].

As a result, the thesis presents and evaluates Tape Cloud, a design for a cost efficient, hybrid, cloud based storage which mainly makes use of magnetic tapes as backend storage media. Tape Cloud is a large-scale unified data backup and archiving solution with a number of synchronized tape drives, each associated with a task queue. Tape Cloud is designed to be compatible with current backup infrastructure and can support various types of data acquisitions as discussed in chapter 4. Tape Cloud is focused on providing the much needed cost efficiency in long term storage of data, especially when its importance in the long run is uncertain. The proposed tape cloud framework also points to a new direction in creating service oriented, cost effective, massive scale infrastructure to meet the growing storage challenge in the coming era of big data enabled industries and research.

Although tapes have been widely categorized as a slow and unpopular storage medium, it outperforms magnetic disks in total cost of ownership and energy consumption (tapes don't consume power when stored in a tape library), which makes tape technology an ideal choice for cloud based archiving services. The research explores the benefits of the state of the art in tape storage technology. The need for a managerial middleware, which is a combination of algorithms and data distribution policies, that contributes in overcoming the latency offered by tapes in order to improve performance of IO processes is proposed and evaluated. The middleware serves its purpose and by improving data distribution efficiency and decreasing the overall response time for read requests. The test cases have been generated using the extensive analysis of live archiving workloads and modelling techniques.

7.1 Efficient Data Distribution

At the core of the approach discussed in the thesis, is the Read Probability Weight based data distribution which is a novel method to distribute blocks of data internally within Tape Cloud to decrease the time needed to retrieve it from tapes. The idea is based on ensuring even distribution of data blocks based on the probability of it being read again. Some of the delay models that are created based on the commercially available tape libraries suggest that accumulation of read requests at task queues can increase the average response time needed to retrieve data. The end result of RPW based data distribution is the improved average response time of the Tape Cloud infrastructure.

7.2 Improved Cost Efficiency

Tape Cloud is a hybrid infrastructure made of hard disk and magnetic tapes. The hard disks are used as data buffers for collecting and temporarily storing data while it is processed before being written into tapes. The major storage is constituted by magnetic tapes. This reduces the comparatively high costs of hard disk based storage media and that of operating hard disk based systems. Tape Cloud being primarily a cloud based offering, the decreased overall cost benefits both the service users and service providers.

7.3 Future Work

7.3.1 New Economics of Legacy Storage

One of the most exciting aspects of our work is the doors of opportunity it opens for new research. Understanding the economics of revisiting a legacy system to solve the data explosion problems of today requires an overhaul of nearly every piece of technology associated with the storage system. The impact of Tape Cloud in live deployment scenarios will shed light on the performance of the middleware and related components and their ability in handling diverse workloads.

7.3.2 Adaptability to Versatile RPW

Currently the implementation of RPW based distribution is based on the pre-assigned weights to data blocks generated by specific applications. Although this is logically correct and serves in evaluating the performance of Tape Cloud, there could be instances where clients provide incorrect information about the importance of data being backed up on Tape Cloud. This leads to incorrect assignment of RPW to data blocks. To avoid this, one of the future enhancements to Tape Cloud include the automated and adaptive learning-based assignment of RPW to data blocks. Future plans of the project also include the improvement of the middleware and the filesystem to support message passing-enabled, adaptive data weight management and IO parallelization. Another area of focus is the elaboration of operation of Tape Cloud for a variety of data types, application and magnitude of serviceability.

Bibliography

[1] Tandberg storagelibrary t24.

[2] Vdbench.

[3] Two thirds of disk-only users look to add tape into storage infrastructure. *Storage Newsletter*, March 2008.

[4] International magnetic tape storage roadmap. *INFORMATION STORAGE INDUSTRY CONSORTIUM*, nov 2011.

[5] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch. A five-year study of file-system metadata. *Trans. Storage.*

[6] I. Ahmad. Easy and efficient disk i/o workload characterization in vmware esx server. In *Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization*, IISWC '07, Washington, DC, USA. IEEE Computer Society.

[7] A. J. Argumedo, D. Berman, R. G. Biskeborn, G. Cherubini, R. D. Cideciyan, E. Eleftheriou, W. Häberle, D. J. Hellman, R. Hutchins, W. Imaino, J. Jelitto, K. Judd, P.-O. Jubert, M. A. Lantz, G. M. McClelland, T. Mittelholzer, C. Narayan, S. Ölçer, and P. J. Seger. Scaling tape-recording areal densities to 100 gb/in 2. *IBM J. Res. Dev.*, 52(4):513–527, July 2008.

[8] P. Berenbrink, T. Friedetzky, Z. Hu, and R. Martin. On weighted balls-into-bins games. *Theor. Comput. Sci.*

[9] C. Corporation. Big data just beginning to explode. 2013.

[10] C. Delimitrou, S. Sankar, K. Vaid, and C. Kozyrakis. Decoupling datacenter studies from access to large-scale applications: A modeling approach for storage workloads. In *Workload Characterization (IISWC), 2011 IEEE International Symposium on.*

[11] J. R. Douceur and W. J. Bolosky. A large-scale study of file-system contents. In *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '99, New York, NY, USA. ACM.

[12] M. Douglas. Supply chain gain: Better data, better results. 2011.

[13] I. Giurgiu, C. Castillo, A. Tantawi, and M. Steinder. Enabling efficient placement of virtual infrastructures in the cloud. In *Proceedings of the 13th International Middleware Conference*, New York, NY, USA. Springer-Verlag New York, Inc.

[14] A. Gulati, C. Kumar, and I. Ahmad. Modeling workloads and devices for io load balancing in virtualized environments. *SIGMETRICS Perform. Eval. Rev.*

[15] T. Hara and M. Tomizuka. Performance enhancement of multi-rate controller for hard disk drives. *Magnetics, IEEE Transactions on*, 35(2):898–903, 1999.

[16] IDC. Worldwide storage in the cloud 2010-2014 forecast: Growth in public cloud storage services continues as firms decapitalize it.

[17] J. Jackson. Most network data sits untouched. *Government Computer News*, July 2008.

[18] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda. Characterization of storage workload traces from production windows servers. In *Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*, pages 119–128, 2008.

[19] D. Lee, M. O'Sullivan, and C. Walker. Benchmarking and modeling disk-based storage tiers for practical storage design. *SIGMETRICS Perform. Eval. Rev.*, 40(2):113–118, Oct. 2012.

[20] Y. Lu and D.-C. Du. Performance study of iscsi-based storage subsystems. *Communications Magazine, IEEE*, 41(8):76–82, 2003.

[21] M. K. Mark Peters and J. Buffington. Cloud storage: the next frontier for tape. In *Oracle White Paper*, 2013.

[22] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). In *Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, SIGMOD '88, pages 109–116, New York, NY, USA, 1988. ACM.

[23] Y. Peres, K. Talwar, and U. Wieder. The (1 + SS)-choice process and weighted balls-into-bins.

[24] S. Quinlan and S. Dorward. Venti: A new approach to archival storage, 2002.

[25] M. Raab and A. Steger. "balls into bins" - a simple and tight analysis. In *Proceedings of the Second International Workshop on Randomization and Approximation Techniques in Computer Science*, RANDOM '98, London, UK, UK. Springer-Verlag.

[26] D. Reine. In search of the long-term archiving solution Ű tape delivers significant tco advantage over disk. *The Clipper Group*, December 23 2010.

[27] D. Reine and M. Kahn. In search of the long-term archiving solution. 2007.

[28] D. S. Rosenthal, D. Rosenthal, E. L. Miller, I. Adams, M. W. Storer, and E. Zadok. The economics of long-term digital storage. In *The Memory of the World in the Digital Age: Digitization and Preservation*, Sept. 2012.

[29] O. Sandsta and R. Midtstraum. Improving the access time performance of serpentine tape drives. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 542–551, 1999.

[30] W. C. S. Series. Network-attached storage. 2012.

[31] W. C. S. Series. Storage area network. 2012.

[32] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.

[33] O. StorageTek. Getting more value from enterprise tape libraries with storagetek tape analytics. 2012.

[34] S. Tewari and L. Kleinrock. Proportional replication in peer-to-peer network, 2006.

[35] D. Thompson and J. Best. The future of magnetic data storage techology. *IBM Journal of Research and Development*, 44(3):311–322, 2000.

[36] Unknown. Cloud storage. 2012.

[37] G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu. Characteristics of backup workloads in production systems.

[38] G. Wu. Why more data and simple algorithms beat complex analytics models. 2013.

[39] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling hard-disk power consumption. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, FAST '03, pages 217–230, Berkeley, CA, USA, 2003. USENIX Association.