2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems

Evolutionary Trends in a Supercomputing Tertiary Storage Environment

Joel C. Frank, Ethan L. Miller, Ian F. Adams, Daniel C. Rosenthal
Storage Systems Research Center
University of California, Santa Cruz

***Abstract*—Tracking archival usage and data migration in a long term supercomputing system is critical to understanding not only how users' needs and habits have changed over time, but also how the archive itself evolves in response to these external factors. Yet this type of study has not previously been performed. To address this need, we conducted an in-depth comparison of user initiated file activity on the mass storage system (MSS) at the National Center for Atmospheric Research (NCAR) during two periods, one in the early 1990s, and another nearly twenty years later. In addition to confirming earlier findings, our analysis turned up three surprising results. First, the read:write ratio went from 2:1 in the earlier trace to 1:2 in the later trace, a reduction of a factor of four in reads relative to writes. Second, only 30% of the current archive was accessed during the three year period of the study, in stark contrast to the 80% seen in the 1992 trace analysis. Third, access latency to the first byte of data actually got slower despite much faster computers and storage devices. These findings indicate that archival behavior has shifted towards a write-heavy workload, and that future archives can be more optimized for write activity than previously believed. Furthermore it may be worth considering the value of data being archived when it is stored, since later retrieval is increasingly less likely.**

I. Introduction

Digital data storage is becoming increasingly important as the mechanism for transmitting scientific and cultural knowledge to future generations. As a result, organizations are collectively spending billions of dollars on systems that can preserve data for the long-term, yet there is surprisingly little known about how users actually *use* the data stored in archives—knowledge that could help system designers build better archives. Perhaps more importantly, there has been no research on how a long-term archive *evolves* over relatively long periods of time. The study done by Agrawal, *et al.* at Microsoft [1] is one of the longest storage system studies conducted, but it only covers only 5 years of desktop file systems, which is a relatively short time for archival data. Furthermore, desktop systems are quite different from archives, and thus we cannot generalize their findings. As a result, archival storage system designers must rely more on "common sense" than on actual data.

To address this problem, we analyzed trace data from the mass storage system (MSS) at the National Center for Atmospheric Research (NCAR) for the period from 2008 to 2010 to determine data access frequencies, overall archive usage, response latencies, and trends in file size and density. We then performed the same analyses on the corpus of data gathered on the MSS at NCAR by Miller and Katz in 1990–1992 [2] in an effort to determine not only modern archival behavior, but evolutionary trends as well. Observations about long-term evolution are possible because NCAR's primary mission—modeling the Earth's climate—has remained relatively unchanged over the twenty year period covered by the two traces. Thus, these traces afford a unique opportunity not only due to the rarity of access to long-term archival storage traces, but also because they are both from the same archival storage system separated by two decades. To our knowledge, this is the first time that the same system has been traced twice with nearly a two decade separation between the traces.

Our analyses conducted in this study yielded several surprising findings, in addition to confirming widely-held beliefs about scientific archival storage. Primary among these findings was a dramatic shift in the read/write ratio from a read-dominated workload in 1992 to a write-dominated workload in 2010, a shift in the read/write ratio by a factor of four. This shift has major implications for archival storage system design, *e. g.* designing the system to primarily handle the write workload. Another key finding was that the fraction of the archive that was accessed more than once dropped from 80% in 1992 to 30% in 2010; a system with a lower access density may need to favor low storage cost over the ability to access files quickly, and may need more frequent archive scrubbing to compensate for the lack of user-driven accesses that may catch "bit rot". The shift towards a write-dominated archive and decrease in the fraction of the archive actually accessed suggest that future archives may need to increasingly focus on preservation rather than providing high-speed access to archived data. In addition, these two trends highlight the need for effective archive data organization and search across millions to billions of files to identify the few files that are needed in response to a given query.

The rest of this paper is organized as follows. Section II discusses earlier file system studies, explaining how they influence our study. We then discuss the methodology of our study in Section III, comparing of the original NCAR system to the current one, and describing the trace data as well as the data scrubbing algorithm. Section IV details the findings of this study and the implications to the design of modern supercomputing archival storage systems. In Section V, we discuss future work to be completed in order to answer questions that are beyond the scope of this study, and we summarize our findings in Section VI.

1526-7539/12 $26.00 © 2012 IEEE
DOI 10.1109/MASCOTS.2012.53

II. Related Work

There are two factors that make this study stand apart from previous trace file studies. First, this study addresses long term storage, which has vastly different workloads [3] and design requirements than those attributed to enterprise systems. For example, enterprise systems are often much more performance oriented. Second, this study analyzes the same site using the same analysis techniques twenty years later. We are unaware of other cases of multiple studies being performed on a single site, particularly with a separation of nearly two decades.

A. Enterprise Studies

There have been many trace-based studies conducted on enterprise and academic file systems over the past twenty years [4], [5], [6], [7], [8], none of which performed evolutionary trend analyses. However, taken as a whole, these analyses can suggest long-term trends in enterprise storage usage. Over time, file systems have grown dramatically in size, primarily by storing more files and a relatively small number of large files—individual file sizes in the traces have not grown as much. These trace studies also investigate user behavior at a relatively fine grain, since they can track individual users' read and write behaviors across all files in the file system, not just those deemed important enough to archive.

A recent study performed on enterprise systems is the work done by Leung, *et al.* [8]. Despite the different organization and workload (day-to-day enterprise file system versus archival scientific file system) many of their findings were similar. In particular, both systems were found to have a low read-write ratio as well as a tendency for files to be accessed very infrequently. These findings serve to highlight the fact that, although enterprise storage systems are neither intended nor designed to be archival in nature, they may gradually become an archive by accident [9].

In contrast, the study by Gibson [10] on long-term behavior in a UNIX file system and the trace analysis done by Agrawal, *et al.* on workstation file systems at Microsoft [1] are most similar in scope to the analyses conducted in this study. As expected, Agrawal, *et al.* found that, over the course of the study, many factors increased, including file sizes, file counts, file density (the number of files per directory), and others. Their study also showed that overall file age was *not* increasing, which is of particular interest because, if file age is not increasing and the rates for reading and writing stay constant, then the rate of deletion must increase or more storage must be added. For the system at NCAR, it is clear that the designers chose to increase overall storage; however, it is unclear whether workstation users do the same thing.

B. Long-Term Storage Systems

There has been relatively little study of usage patterns in long-term archival storage systems, perhaps because of the difficulty in gathering long-term traces. Many of the studies on archival storage systems for scientific computing were performed over twenty years ago [11], [12], [13], [14]. These studies had findings largely similar to the original study of the NCAR archival storage system [2], detailing usage patterns and user behavior. However, the largest archive studied in these systems was the 1992 NCAR archive, at 25 TB—the size of a workgroup disk array today. Given the advances in computing and storage technology over the intervening twenty years, the quantitative findings from these early studies are no longer relevant.

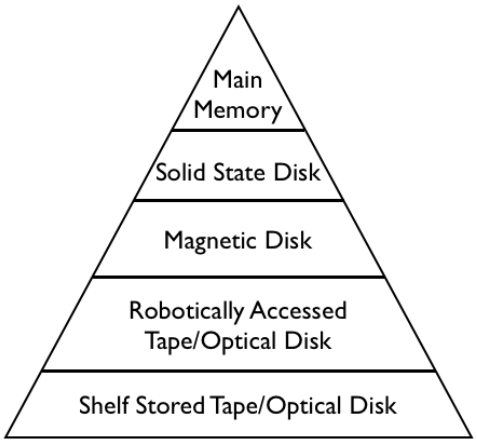


Fig. 1. Memory and storage hierarchy in large computer systems. This is also known as the storage pyramid.

More recently, there has been renewed interest in understanding usage behavior of archival storage systems. Adams, *et al.* [15] found that many modern archival storage users modified files in the archive, and that non-scientific archive usage was very bursty. These findings differ from the characteristics of scientific archival storage systems, as this paper demonstrates.

III. Methodology

The National Center for Atmospheric Research maintains a large supercomputer center whose primary responsibility is supporting climate researchers. These researchers use the archival storage system at NCAR to preserve both gathered data and the output of climate models over long periods of time, providing a historical record of the climate research over the lifetime of the center. This information is used for several purposes. Typically, data cannot be analyzed in real time; instead, it is stored in the archive for later analysis. In addition, older data is sometimes used for comparison with more recent climate models and, in some cases, verification of older model results against observed conditions.

A. Evolution of the NCAR Mass Storage System

The overall design of the archive has not changed over the past two decades: the system still has a disk cache in front of a large amount of tape storage, as suggested by the "storage pyramid" shown in Figure 1. However, the dimensions of the system have grown dramatically, from 25 TB in 1992 to over 30 PB capacity in 2010.

Field	Meaning
timestamp	event completion time in hours, minutes, and seconds
log record type	code word for the event type (e.g. read, write, or create)
host sequence number	ID tag for the event; when combined with the filename signify a unique event
data transfer time	time in seconds to complete the transfer
transaction time	time in seconds from start of event to completion
file size	file size in bytes
storage level	1 = disk cache, 2 = primary tape, 3 = second copy tape
filename	absolute path and filename

TABLE I
INFORMATION OF INTEREST CONTAINED IN A SINGLE TRACE RECORD.

NCAR's mass storage system has always consisted of three main levels. There is a controlling server that acts as the gatekeeper to the mass storage network. Behind the gate keeper, the first level of storage is the disk cache, whose capacity in 1992 was 100 GB. By comparison, today the disk cache is 1000 times larger and is comprised of 500–750 GB hard drives.

The next level of storage is the primary tape silo, which in today's system is a StorageTek SL8500 Tape Silo. In 1992, the primary tape silo was a StorageTek Automated Cartridge System 4400 with 6000 IBM 3480-style cartridges, each with a capacity of 200 MB [2].

The last level is the manual tape drives that act as overflow and temporary storage for the primary tape silo. This layer is currently made up of 70 StorageTek T10000B drives fronting over 30 PB of tape storage, whereas in 1992 its capacity was only 25 TB of shelved tape.

Beyond the expansion in capacity, the biggest significant change as a result of innovations in storage technology is that, in 1992 the maximum file size was limited by the capacity of the tape cartridges to 200 MB. While there may be a similar limit today, it is less important because the tape cartridges have a capacity of approximately 1 TB, resulting in a limit for file size that is larger than most climate models produce. The impact of this change is discussed later in the paper.

B. Archive Storage Traces

The supercomputing center at NCAR maintains detailed trace records for their mass storage system; they use the traces both to assist in planning for upgrades to the storage system, to record the health of the system, and to serve as proof that a requested transaction took place. The traces only contain references to user-initiated activities, such as reads, writes, and migration between levels. However, they did not contain all of the records relating to data migration to a new storage system or to reads performed to check data integrity of stored data. While such reads may represent a significant load on an archival storage system [15], we were unable to include them in our analysis because of the lack of complete trace data.

The traces we obtained from NCAR were in ASCII format, and were designed to be easy to generate using standard logging software. Traces are maintained in ASCII for several reasons. First, ASCII is easily human-readable; this proved to be a boon for us because it allowed us to diagnose issues such as a format change that occurred during the tracing period. Second, ASCII traces require no trace-specific translation application to convert the trace logs into a usable format. While this approach may consume slightly more space for uncompressed traces, compression tools such as `gzip` are very fast, removing any additional storage overhead while preserving the advantages of ASCII traces. Table I shows the fields of interest in a single trace record.

Before analyzing the data sets, we first scrubbed the traces to remove any events not of interest, specifically any non-user based event. We then cleaned them up to address a naming convention change that occurred part way through the latter trace period. This operation was necessary to get an accurate measure of both unique events and files in the system. The information we obtained to deal with this change mid-trace was obtained from Gene Harano at NCAR; without his help, we might not have been able to run the analysis. This problem highlights an issue with monitoring and trace collection, particularly for long-term storage: the system *must* record not only activity but also changes to the log format itself. Failure to do so may render long-term traces far less useful.

Once we had the cleaned-up traces from the 2008–10 trace period, we converted the traces from the earlier 1990–92 trace period into the same format, allowing us to run identical analyses on both sets of traces. This approach removed the possibility of differences in the results being based on differing assumptions when processing the data, potentially yielding inaccurate evolutionary trend conclusions. Furthermore, by rerunning the original analyses, we were able to verify the accuracy of our new analysis algorithms and tools against the results in the original paper [2].

IV. RESULTS

At the time of the 2010 traces, the NCAR system contained approximately 69 million files. However, this study includes only those files that were actually accessed (read or written) by users during each trace period. In particular, read and write events involved in migrating data to newer hardware were not included. There are also atmospheric data files that are typically not analyzed by scientists until after three to five years, which means their access period would not fall within either trace period. The presence of these files does not invalidate the findings of this study however, since these types of files were also present in the original study. Therefore a comparison of the archive at these two periods of time is still valid.

An overview of the activity recorded in the traces is shown in Table II. Note that percentages shown are contributing percentages to the total events of that type. For example, read events to disk constituted 60% of the total number of

	Reads 1992	Reads 2010	Writes 1992	Writes 2010
References	800K (66%)	7,424K (34%)	411K (33%)	14,502K (66%)
Disk	488K (60%)	3,509K (22%)	324K (40%)	12,431K (78%)
Tape (Silo)	162K (66%)	3,913K (65%)	82K (33%)	2,070K (35%)
TB Transferred	21.8 (72%)	1,805K (39%)	8.0 (28%)	2,780K (61%)
Disk	1.60 (55%)	617K (24%)	1.30 (45%)	1,924K (61%)
Tape (Silo)	13.1 (66%)	1,187K (58%)	6.5 (34%)	856K (42%)
Avg File Size (MB)	61	730	44	575
Disk	7.6	528	8.9	464
Tape (Silo)	182	911	177	1240
Latency (sec)	130	379	92	182
Disk	30	8.8	22	17
Tape (Silo)	103	151	74	83

TABLE II
OVERALL TRACE STATISTICS, WITH ACTIVITY NORMALIZED TO AN ANNUAL BASIS.

disk references each year from 1990 to 1992, but only 22% each year from 2008 to 2010. However, summary values have been normalized to annual values to account for different trace durations for the 1992 and 2010 data sets. This means, for example, that there were about 7.4 million reads per year in the 2010 trace, and 21.8 TB of data read per year in the 1992 trace.

A. Read Density

The *read density* of a system is defined as the ratio of read events to write events for the system. In 1992, as Table II shows, the NCAR archive was read-dominated, with twice as many reads as writes. By 2010, however, writes dominated, with two times as many writes as reads. While this long-term trend is often assumed, our findings validate this assumption. This change has serious implications for archival system designers, who should consider optimizing the system for writes rather reads.

This change in user behavior likely results from one of two influences. First, users may simply be storing more data that they care less about, bolstered by the long-term decrease in storage cost. Because storage is much less expensive now, users can afford to be less selective in the data they choose to store; the process of making the decision is more expensive than at least the initial storage cost. The second possible explanation is that the rate at which users are *accessing* the data is staying the same relative to computing power, but the rate at which users are storing data has increased greatly. As described later, this explanation is supported both by the drastic change in the read to write ratio as well as the fact that

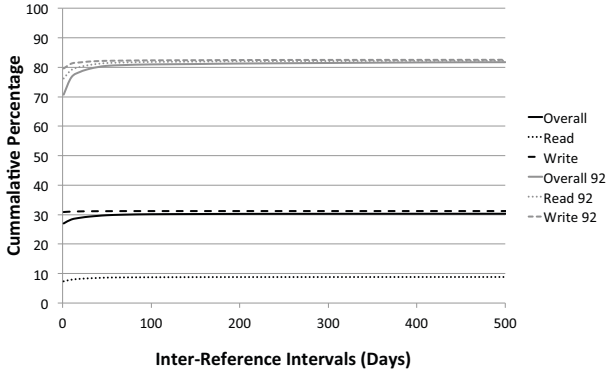


Fig. 2. **Comparison of read, write, and overall inter-reference intervals for the 1992 and 2010 trace data**. It is evident that only 30% of the 2010 archive is being accessed as compared to 80% in 1992. (Note: In this figure, 100% is the total number of files in the system, 69 million, not the total number of files seen in the traces.)

users only accessed 30% of the current archive over the three year period of this study.

This change has far-reaching implications for system designers, particularly those involved in domains in which funding for a file's maintenance is generated on a per-access basis, including advertising-supported sites such as video-sharing and photo-sharing sites that derive revenue from advertising displayed alongside access media. As the read-to-write ratio declines, the read density of the archive declines, providing less income to maintain the archive. Compounding the problem, the data in the archive must be migrated to newer media and devices, both to deal with aging devices and to leverage improved device performance. In addition, the decreased number of file reads also means that the system (rather than users) must initiate more read accesses to maintain data integrity and avoid bit decay [16]—infrequently accessed data must be explicitly checked by the system rather than implicitly checked during user accesses.

B. Inter-Reference Intervals

While the archive at NCAR is increasing in size, many of the files in the archive remain unaccessed for long periods of time, as Figure 2 shows. We can measure the amount of archive activity as well as the amount of reuse using the *inter-reference interval*: the time between successive accesses to a particular file. Files that are only accessed once do not have an inter-reference interval; such files are excluded from this graph.

The read and write events for a given file were broken down by event type and sorted by their timestamp. The inter-reference interval was then determined for each event type. Figure 2 shows a comparison of the inter-reference intervals for each event type as well as the overall inter-reference interval for both the 1992 system as well as during the 2010 traces. The most drastic change between the 1992 archive and the current one is that in the past, 80% of the system was

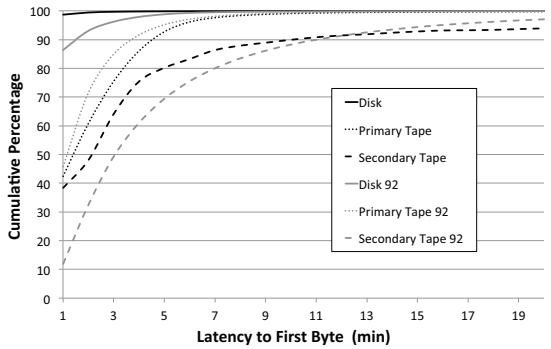


Fig. 3. Comparison of latencies to first byte. All latencies have improved with the exception of primary tape, whose latency *worsened* by a factor of 1.5

accessed more than once, whereas only 30% of the current system is being written to, and only 8% of it is ever being read. It also illustrates the dramatic shift in read density between the two trace periods.

This shift over the past twenty years allows designers to implement policies for files that haven't been accessed within a certain amount of time forcing them to be offloaded to a slower and less expensive storage medium since the probability of accessing that file in the future is almost zero. These findings are further supported by those found during the read density analysis in section IV-A.

C. Latency to First Byte

While most computer hardware has gotten orders of magnitude faster over the past twenty years, storage hardware has not seen similar performance increases. In particular, tape hardware—both drives and robots—have much higher bandwidth, but positioning delays (seek time, robotic load time) are not much lower twenty years later.

The average delay between when an event is requested and when the first byte of data is sent or received is the *latency to first byte*. With regard to the NCAR traces, the latency to first byte was calculated by subtracting the data transfer time from the total transaction time. Figure 3 shows a comparison of these latencies between the various types of storage devices for both the 1992 system and the current one. As the figure shows, latency for most types of storage devices decreased between the two traces, though only by a small factor. This is not unexpected, since storage devices have not dramatically reduced positioning delays over the past two decades.

However, note that primary tape latencies are *slower* in 2010 than in 1992 by a factor of approximately 1.5. 85% of tape silo requests completed within 3 minutes in 1992, as compared to only 70% of similar requests in 2010. This disparity can be explained in several ways. First, the fact that tape requests are so small in the first place indicates that many requests to tape are for files on a currently-mounted tape; fetching a tape takes far longer than 3 minutes. As tapes become larger, however, even files on a single tape may be separated by a large seek, increasing latency. Second, the larger number of files in 2010 make it less likely that the file being requested next is actually sequential on tape, again necessitating a (slow) tape seek.

It is safe to assume that overall system performance has not decreased over time due to this increased latency; otherwise, users of the system would likely demand that this problem be addressed. One explanation for this increasing latency being masked is the increased performance of the disk cache, which is much larger and faster than it was in the original system. As a result, the system still appears responsive to the typical user, even though accesses to the primary tape silo have actually gotten slower. This allowed NCAR to focus their monetary investment into ensuring that the disk cache meets the needs of their users while allowing the tape silo to be slower without impacting user access times.

This increase in tape silo latency has other implications as well. It is possible that performance could be increased by decreasing the seek times for tape. Both hard drives and tape silos have initial start up costs [17], but the seek times for tape drives are much longer.

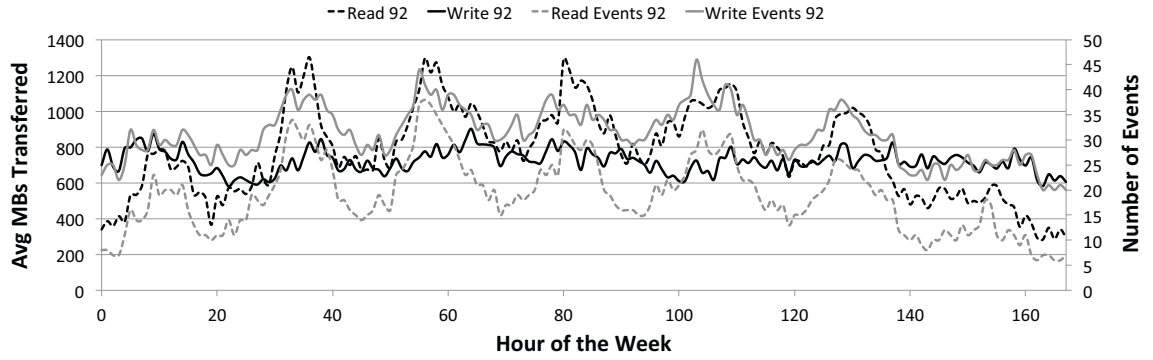
D. Hourly Usage Patterns

Figure 4 shows that it may be better to optimize modern archival systems for a write-intensive workload rather than fully optimized for read. This is a large shift from the workload of twenty years ago that was primarily read-oriented. Write events can simply be cached during peak read times, and sent out to the archive during off hours. This approach allow the system to remain responsive to users during working hours, and then catch up on write activity when the read workload drops back to baseline.

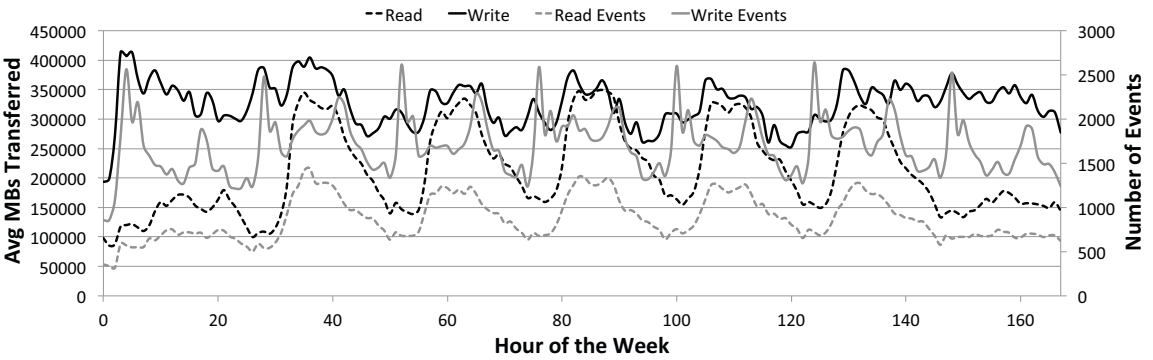
In both the 1992 and 2010 traces, overall system load is highest during working hours: 8 AM to 5 PM, with the gradual decline in system load later in the day most likely due to a percentage of the staff working past normal working hours. This shift is primarily driven by read events and read bandwidth. Given the highly cyclical pattern of reads, and the relatively low level of read events and read bandwidth on nights and weekends, it seems clear that reads are primarily driven by interactive users rather than batch processing.

On the other hand, write workload stays high through the week in both data sets, even though the number of write events is much lower during the weekend. This implies that batch processing is a key driver behind large write events, as would be expected: supercomputing applications produce large files that are subsequently written to archive. However, in 2010, write event rates are somewhat decoupled from write bandwidths, indicating that different types of writers may have different behaviors with respect to file size. More specifically, batch writes appear to put twice as much bandwidth load on the system as user-initiated ones.

Another major difference between the two trace periods is that, in the original system, read events were comparable to write events in the amount of data transferred, whereas in the current system write events make up a much larger percentage of the overall data transferred. This supports the conclusion



(a) 1992 workload. Note that the write workload stays relatively constant throughout the week, even though the number of write events increases by 40% during the workday. Read workload follows the number of read events across the entire week.



(b) 2010 workload. The write workload tracks the number of write events throughout the week, but not during the weekend, where the number of write events drops off, while the bytes transferred remain constant. Read workload follows the number of read events throughout the week, but the amount of data transferred per event during the workweek is more than double than during the weekend.

Fig. 4. Comparison of the average amount of data transferred per hour to the number of read and write events per hour. 0 is Sunday at midnight.

that the system should be designed for the relatively constant write workload, with concessions taken to handle the spurious read traffic.

The approach of sizing for a constant write bandwidth and buffering writes during periods of high read is bolstered by the observation that, in 2010, batch reads typically put little load on the system, as shown by the low read rates on weekends. However, read rates spike during weekdays, likely due to users retrieving large archived data files for analysis. Thus, postponing writes during the day will not require excessive amounts of disk for buffering, making it more feasible to use this approach to reduce the required support level for concurrency in the system.

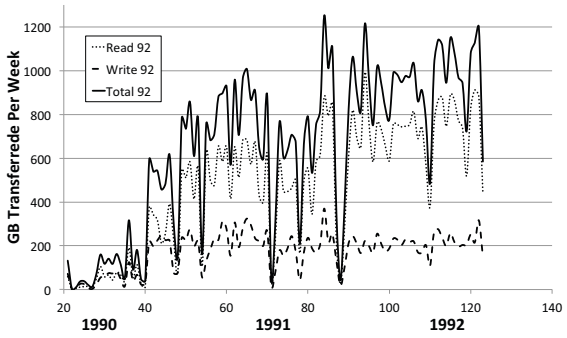
E. Weekly Usage Patterns

The amount of data transferred each week in the original archive and the current archive is shown in Figure 5. These graphs further support that conclusion that the system should be designed for the write workload: in 2010, the amount of read data is well below write data, in contrast to the 1992 workload, where reads dominate writes.

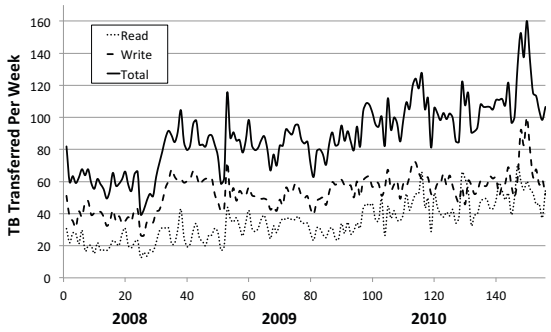
During the two year period in the original archive, a ramp up of workload can be seen; this is expected, since the archive came online shortly before the trace period began. One thing to note, however, is that over the two year period the write rate does not increase with the read rate. One explanation for this was that the computing center was already running at full capacity, so reads were primarily due to user-oriented tasks such as visualization [2].

In contrast, in the current archive, there is no ramp up, but there is a steady linear increase in workload. Furthermore, the write rate is considerably higher than the read rate, whereas in the original archive the converse is true. Researchers are storing much more information than they are accessing to analyze, so the system must be designed to handle this type of workload.

Another observation is that the drop in system load over holiday periods is much more severe in the 1992 trace data as compared to the current one. This is most likely due to heavier use of batch processing, which would keep the system loaded without direct user interaction. Recall that Figure 4(b) illustrated the decreased contribution that read events have in in the overall system workload for the week.



(a) 1992 Workload



(b) 2010 Workload

Fig. 5. **Data transferred per week in the two archives.** In 1992, overall workload increased over the trace period, but write rate did not. In 2010, there was a relatively constant workload across the trace period, including holidays, which implies an increase in the use of batch processing.

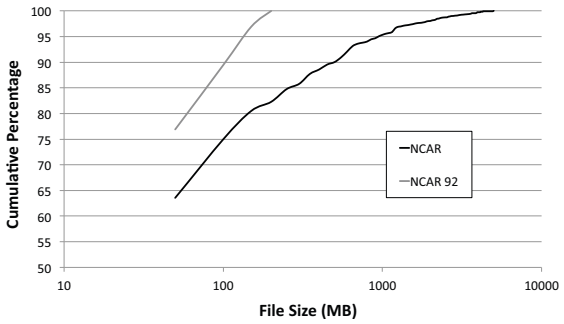
F. File Sizes

When file sizes increase, keeping the rest of the system constant, it follows that overall performance will decrease due to a degradation in the overall parallelism of the system. Since the number of files accessed is not growing linearly with the average file size, it follows that the number of spun-up drives per event is also not increasing linearly. Therefore, since more data is being read or written per event per spun-up drive, simply due to the increase in file sizes the system will become more serialized. This increase in serialization will have a direct negative impact on system responsiveness.

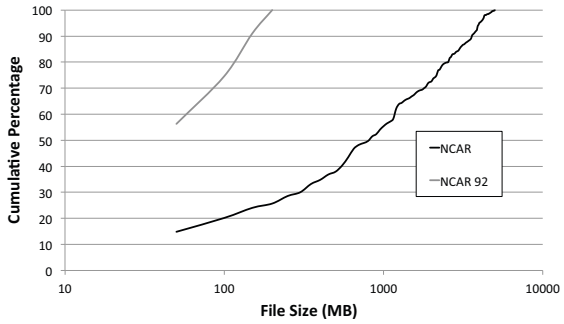
Figure 6(a) shows a comparison between the sizes of files in the system in 1992 and 2010. One thing to note is that, beyond the expected result of file sizes getting larger and there being more files per directory [18], the reason the largest file size in the original system was 200 MB is due to the physical limit of the storage media at the time. The largest tape in the original archive was 200 MB. As a result, there are many files of up to 4 GB and larger in the 2010 system; 5% of files are larger than 1.3 GB.

G. Directory Density

Increases in directory density can cause performance issues due to the non-trivial task of searching a given directory. However, even though according to Figure 7 the number



(a) Comparison of file sizes between the 1992 and 2010 data sets.



(b) Comparison of archive space consumed by file size.

Fig. 6. **File size comparisons in the two archives.** The max size in the 1992 archive was 200 MB due to the physical limit of the storage media at the time. Most of the space in 2010 is consumed by large files, even though most of the files are small.

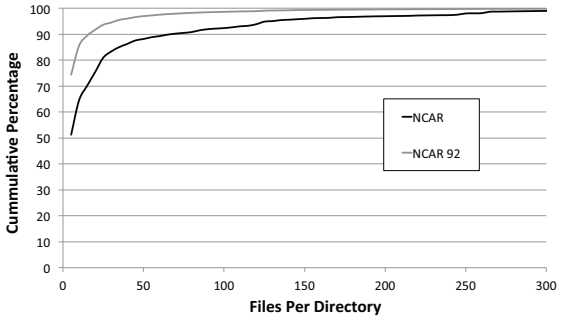


Fig. 7. **Comparison of directory density between the 1992 and 2010 data sets.** The number of files per directory has roughly doubled over the 20 year period.

of files has roughly doubled between the 1992 and 2010 studies, the number of files per directory in the NCAR system does not approach the issues associated with immense data structures [19]. Whether the small number of files per directory is due to system limitations influencing users or an intrinsic property of the workload is a subject for future investigation.

V. Future Work

There are several areas of research pertaining to evolutionary trends that were not covered by this study. First, we do not consider events due to system-directed operations, such

as migration between system events [11]. Events of this type move information from one layer of the storage system to another (*e. g.*, moving files from the disk cache to the primary tape silo). If these events are scheduled to occur without regard to user activity, *i. e.*, uniformly across the work day or work week, then they will not tend to alter the shape of the overall workload curve, as Figure 4(b) shows. However, if these migration events are scheduled to occur during periods of expected low user activity, as is logical, then they will definitely mask the impact that user activity has on the system by further flattening out the overall workload on the system. In other words, the impact of users' peak read times will be reduced, or even negated.

Locality of namespace accesses is another of interest that is beyond the scope of this analysis. How are access statistics different when viewed in a namespace centric manner? For example, during a user's access session, which files are accessed? Are they similar? Are they within a certain directory radius of each other?

Answers to these questions would better prepare system engineers to design the system to better meet the users' needs. For example, if it is found that users most often access files within the same or similar directories at the same time, then grouping files on media by directory or by user [14] and perhaps even prefetching them could dramatically improve read performance at relatively little cost, since the majority of the read cost is paid on access to the first byte.

VI. Conclusions

High-performance computing systems and storage systems have seen tremendous advances in computing power and storage capacity over the past two decades. Archival storage is increasingly important in storing the results of research in such environments, yet no study has done an "apples-to-apples" comparison of a single environment over such a long period of time.

This paper compared trace data for the NCAR center from 1992 to trace data taken from the current archive to determine evolutionary changes over the previous twenty years. The study produced several key findings that are relevant for designers building archival storage systems.

First, writes have become four times more frequent relative to reads over the past twenty years. This, combined with the reduction in the fraction of the archive that is actually accessed over three years, indicates that archives are becoming increasingly "write-only", with attendant implications for system design.

In addition, the high level of writes suggests that systems should be designed to handle the high write load, with writes postponed during periods of high read activity. Since reads are primarily user-driven, these periods are highly predictable, and can allow system designers to save money by reducing the maximum level of concurrency the system must support.

Finally, as is well-documented, access latencies are not declining very fast. Given the bursty nature of reads, it may be useful to design systems to group or prefetch data to reduce perceived latency, even if doing so means reading data from archive to disk cache that may never be used. It is also necessary to use a relatively large disk cache to hide this latency from users, perhaps even permanently caching small files to reduce access latency for them. Fortunately, this approach is cost-effective given the relatively low cost of disk.

By studying the same storage system being used for the same purpose at two different periods separated by nearly two decades, we have provided valuable insight into long-term archival storage system behavior. We have also provided a detailed look at current user behavior for archival storage systems. In doing so, we hope to enable archival storage system designers to build long-term storage for the next twenty years and beyond.

Acknowledgments

This research was supported in part by the National Science Foundation under awards CNS-0917396 (part of the American Recovery and Reinvestment Act of 2009), CCF-0937938, and IIP-0934401, and by the sponsors of the Storage Systems Research Center, including EMC, Hewlett Packard, IBM, NetApp, Northrop Grumman, and Samsung.

References

[1] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch, "A five-year study of file-system metadata," in *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)*, Feb. 2007, pp. 31–45.

[2] E. Miller and R. Katz, "An analysis of file migration in a Unix supercomputing environment," in *Proceedings of the Winter 1993 USENIX Technical Conference*, Jan. 1993, pp. 421–433.

[3] D. L. Lee, M. O'Sullivan, C. Walker, and M. MacKenzie, "Robust benchmarking for archival storage tiers," in *PDSW '11 Proceedings of the sixth workshop on Parallel Data Storage*, 2011.

[4] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. Shirriff, and J. K. Ousterhout, "Measurements of a distributed file system," in *Proceedings of the 13th ACM Symposium on Operating Systems Principles (SOSP '91)*, Oct. 1991, pp. 198–212.

[5] D. Ellard, J. Ledlie, P. Malkani, and M. Seltzer, "Passive nfs tracing of email and research workloads," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST '03)*. San Francisco, CA: USENIX, Mar. 2003, pp. 203–216.

[6] D. Roselli, J. Lorch, and T. Anderson, "A comparison of file system workloads," in *Proceedings of the 2000 USENIX Annual Technical Conference*. San Diego, CA: USENIX Association, Jun. 2000, pp. 41–54.

[7] W. Vogels, "File system usage in Windows NT 4.0," in *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP '99)*, Dec. 1999, pp. 93–109.

[8] A. W. Leung, S. Pasupathy, G. Goodson, and E. L. Miller, "Measurement and analysis of large-scale network file system workloads," in *Proceedings of the 2008 USENIX Annual Technical Conference*, Jun. 2008.

[9] A. Wildani and E. L. Miller, "Semantic data placement for power management in archival storage," in *Petascale Data Storage Workshop (PDSW), 2010 5th*, 2010.

[10] T. J. Gibson, "Long-term UNIX file system activity and the efficacy of automatic file migration," Ph.D. dissertation, University of Maryland, Baltimore County, May 1998.

[11] A. J. Smith, "Analysis of long term file reference patterns for application to file migration algorithms," *IEEE Transactions on Software Engineering*, vol. 7, no. 4, pp. 403–417, Jul. 1981.

[12] ——, "Long term file migration: Development and evaluation of algorithms," *Communications of the ACM*, vol. 24, no. 8, pp. 521–532, August 1981.

[13] E. Thanhardt and G. Harano, "File migration in the NCAR mass storage system," in *Digest of Papers, 9th IEEE Symposium on Mass Storage Systems*. IEEE, Oct. 1988, pp. 114–121.

[14] R. L. Henderson and A. Poston, "MSS-II and RASH: A mainframe UNIX based mass storage system with a rapid access storage hierarchy file management system," in *Proceedings of the Winter 1989 USENIX Technical Conference*, 1989, pp. 65–84.

[15] I. F. Adams, E. L. Miller, and M. W. Storer, "Analysis of workload behavior in scientific and historical long-term data repositories," *To appear in ACM Transactions on Storage*, vol. 8, no. 2, 2012.

[16] H. M. Gladney and R. A. Lorie, "Trustworthy 100-year digital objects: Durable encoding for when it's too late to ask," *ACM Transactions on Information Systems*, vol. 23, no. 3, pp. 299–324, Jul. 2005.

[17] 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)*, San Francisco, CA, Mar. 2003, pp. 217–230.

[18] J. D. Crabtree and D. Sheaves, "Evolution of a data archive," in *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries (Work-In-Progress Report/Poster Presentation)*, June 2007, pp. 478–478.

[19] S. Patil and G. Gibson, "Scale and concurrency of giga+: File system directories with millions of files," in *FAST'11 Proceedings of the 9th USENIX conference on File and stroage*, 2011.