Analysis of Workload Behavior in Scientific and Historical Long-Term Data Repositories

Technical Report UCSC-SSRC-11-01
March 2011

Ian F. Adams	Mark W. Storer	Ethan L. Miller
iadams@soe.ucsc.edu	mwstorer@netapp.com	elm@soe.ucsc.edu

Storage Systems Research Center
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
http://www.ssrc.ucsc.edu/

Analysis of Workload Behavior in Scientific and Historical Long-Term Data Repositories

Ian F. Adams
U. of California, Santa Cruz

Mark W. Storer
NetApp

Ethan L. Miller
U. of California, Santa Cruz

Abstract

The scope of archival systems is expanding beyond cheap tertiary storage: scientific and medical data is increasingly digital, and the public has a growing desire to digitally record their personal histories. Driven by the increased cost efficiency of hard drives compared to tape, and the rise of the Internet, content archives have become a means of providing the public with fast, cheap access to long-term data. Unfortunately, designers of purpose-built archival systems are either forced to rely on workload behavior obtained from a narrow, anachronistic view of archives as simply cheap tertiary storage, or extrapolate from marginally related enterprise workload data and traditional library access patterns.

To provide relevant input for the design of effective long-term data storage systems, we examined the workload behavior of several scientific and historical archives, covering a mixture of purposes, media types, and access models. Our findings show that, for scientific archival storage, files have become larger, but update rates have remained largely unchanged. However, in public content archives, we observed behavior that diverges from the traditional "write-once, read-maybe" behavior of tertiary storage. Our study shows that the majority of such data is modified relatively frequently, and that indexing services such as Google and internal data management processes may routinely access large portions of an archive, accounting for most of the accesses. Based on these observations, we identify areas for improving the efficiency and performance of archival storage systems.

1 Introduction

Archival storage has traditionally been viewed as inexpensive, tertiary storage [21, 25, 34, 35]; however, this anachronistic definition is too narrow to accurately describe the current gamut of long-term storage use cases. Changing storage media and access methods and increasingly digital workflows have radically affected how long-lived content is created, stored, and published. In the business arena, data preservation is often mandated by law, and data mining has proven to be a boon in shaping business strategy. For individuals, archival storage is being called upon to preserve sentimental and historical artifacts such as photos, movies, personal documents, and medical records [13, 19]. Finally, in one of the biggest disruptive shifts in recent history, the Internet has radically changed how users interact with data, catalyzing an explosion in the number of publicly-accessible, long-term content repositories [3, 14, 17, 26, 27, 41].

While the growing variety of archival use cases share a common characteristic of long data lifetimes, understanding of their respective workloads is out of date at best, and non-existent at worst. There have been no detailed studies of archival storage usage in nearly two decades; the most recent studies were of supercomputing environments in the early 1990s [21, 25]. As a result, current work in long-term storage relies upon questionable workload assumptions: observations of out-dated archives with different media types and purposes, marginally related studies of shorter-term enterprise workloads, or access patterns found in traditional library data stores [8, 37]. In contrast, a number of general purpose and high performance workload studies have been published in recent years, demonstrating the value of up to date, empirical data [2, 4, 22, 31].

To close this knowledge gap, we present a study of *long-term data repository* characteristics and workload behavior covering between one and three years of activity in several long-term data stores. The archives we chose allow us to compare and contrast a range of workloads found in tertiary storage and public content repositories, comprising prime examples of use cases aimed at preserving data with indefinite life times. One archive stores an approximately 1.3 PB scientific data set from Los Alamos National Laboratory (LANL), spread across disk and tape, with file metadata crawl summaries covering 1 year; this store is most similar to those from prior archival storage studies and typifies the tertiary storage system use case. Another store, representative of the expanded role of long-term storage, is the Washington State Digital Archives [41]; we analyzed 2.5 years of access logs and record metadata from this store. The third archive is a repository of water table reports from the California Department of Water Resources [11], with logs covering 3 years of activity. Other use cases often grouped in the archival space—such as backup [23, 46], and compliance [19, 32]—exhibit different behavior and thus warrant their own examination as well.

Our analysis of the LANL metadata summaries revealed that tertiary storage archives have changed in a number of ways. First, the current LANL data set is considerably larger than the NCAR system studied in 1993 [25], over 1 PB compared to 25 TB. Second, the ratio of disk to tape has changed from the NCAR system's ratio of 1:262, to the LANL system ratio of 1:3.3 at the end of our study. Interestingly however, despite the increased use of hard disks, overall update behavior is largely similar to previous studies. Third, the typical file size has grown considerably, although many of the files are quite sparse.

Our results with the public water and historical archives mark one of the first critical examinations of the emerging class of publicly accessible long-term data repositories, and reveal that their behavior deviates radically from conventional wisdom. First, the majority of data in both the Washington State and water datasets were updated at least once, and often several times over the course of our traces. This finding directly contradicts the widely held belief that archives are "write-once". Second, we found that large batch processes routinely touch vast amounts of repository data, dominating traffic to publicly accessibly archives. This behavior suggests that a separate batch interface for low-priority accesses could provide significant benefit. Third, we found that even though some items within the data repositories are moderately more popular, the distribution of accesses is extremely long tailed, reducing the effectiveness of LRU caching for reads. Fourth, we found that accesses exhibit strong content locality within user sessions, though a variety of content tends to be retrieved across user sessions, suggesting that grouping data based on semantic content could yield performance and efficiency gains.

The rest of the paper is arranged as follows. Section 2 further illustrates the current gap between archival systems, and our understanding of long-term workloads. Section 3 describes each of our datasets included in our study, as well as the traces collected over them. Next, in Section 4 we present our observations. Finally we discuss the implications of our results in Section 5, and conclude in Section 6.

2 Related Work

In this section we present a brief timeline of workload studies, illustrating that our understanding of long-term data behavior predates the ubiquity of the Internet, and the expanding role of archival storage. As a result, relevant systems utilize behavior assumptions that are outdated, and overly narrow in scope.

The first generation of studies date back to 1981 [34, 35]. In those studies, Smith studied the file system of the Stanford Linear Accelerator Center in the context of optimizing file migration algorithms, and defined the basic patterns of tertiary storage behavior.

Establishing the approximate ten-year frequency precedent, the next generation of studies occurred in the early nineties. Workloads investigated included workstation file systems [38], and mixed disk and tape tertiary storage [21, 25]. These studies are particularly interesting, as they appear at the cusp of the Internet, and the growth of public content repositories. In 1993, Miller and Katz examined storage use at the National Center for Atmospheric Research (NCAR) [25], which at the time consisted of approximately 100 GB of disk, and 25 TB of tape. In contrast, the LANL corpus we studied is expected to grow to 2 PB by 2011.

Finally, the recent past has seen the latest generation of workload studies [2, 4, 22, 31], albeit none that specifically examine long-term data behavior. Thus, while these examinations demonstrate the value of an up to date understanding of storage system behavior, it is difficult to generalize their findings to long-term data repositories. For example, Agrawal *et al.* examined long-term metadata trends, but for desktop PC workloads, and Anderson examined high performance workload traces from an animation company [4]. Similar studies done over other relevant datasets have proven to be a boon to the research community [2, 6, 7, 29, 33, 40].

As a result of the research community's lack of analysis of archival storage system behavior, a number of recent long-term preservation systems are based on assumptions that may not be valid, or only pertain to a narrow view of archival storage. For example, many long-term data systems explicitly assume that contents are immutable, or imply this by using WORM media [9, 24, 30, 36, 44, 45]. Others assume that data is rarely read. For example, Pergamum claims dramatic cost savings largely predicated on the ability to keep drives spun down due to low read rates [37]. Though these assumptions *might* hold true, we have no up to date knowledge with which to confirm them. It has been over 15 years since the last tertiary storage study, and to the best of our knowledge, there have been no studies of access behavior in modern public content archives.

Owner	Name	Size	Records	Access	Media	Data Types
Los Alamos National Laboratory	Scientific	1.3 PB	60,000,000	Private	Disk, Tape	Multiple
Washington State Archive	Historical	Unknown	28,000,000	Public	Disk	Multiple
Calif. Dept. of Water Resources	Water	2.6 GB	57,000	Public	Disk	Single

Table 1: Overview of the corpora and archives covered by this study. We use *name* to identify the sketch throughout this paper. Data types are the number of different types of records in the corpus.

Corpus	Type	Length	Entry count
Scientific	Daily FSStats histograms	13 months	4716
Historical	User access logs	33 months	5.8 million
Historical	Record metadata	33 months	28.3 million
Water	Record update and metadata logs	51 months	900 thousand
Water	User access logs	33 months	100 thousand

Table 2: Overview of reports and logs in our sketches, including the duration and number of distinct entries in each log.

3 Datasets

In an effort to achieve consistency in our discussion, we begin by establishing a set of concise definitions. An individual element in a set of data is a *record*. A record may be a file, bitstream, or even a literal SQL record. We refer to a collection of records as a *corpus*, and a copy of that corpus as an *instance*. The hardware and software used to store an instance of the corpus is the *archive*; the long-data lifetimes and relatively short refresh cycle of modern hardware suggest that a corpus will reside on several archives over its lifetime. A *system* is a holistic view of the archives, corpus and potentially even users. Finally, we refer to the aggregate body of knowledge about a system as a *sketch*. A sketch includes trace logs, profiles, record metadata, and communication with system architects and administrators.

Tables 1 and 2 provide an overview of the sources we used to conduct this study, illustrating that the scope of this study is focused on tertiary storage and public content archives. Our first source, from Los Alamos National Labs (LANL), allows us to update our understanding of traditional tertiary storage systems. Our second source, illustrative of the shift the Internet and lowering storage costs and have brought, is a public repository of digitized historical documents, the Washington State Digital Archives [41]. The third source we examine is a publicly accessible repository of water table reports—such as ground water levels and salinity—from the California Department of Water resources [11]. This source is particularly interesting as it illustrates yet another new direction in the long-term data space; small, per-department specialized content repositories. Understanding the use of these small corpora is important, as many may be stored on a single physical archive where the aggregate behavior may be more important than individual behavior.

Histogram type	Description
Reported size	File length returned by `stat`
Allocated space	Number of bytes actually allocated
mtime	File modification times
mtime (KB)	File modification times, grouped by file size
Overhead	Difference between reported size and allocated space

Table 3: FSstats histogram reports collected over the scientific repository. One set of histograms covers the entire archive, and the other set is run once for each individual top-level directory, corresponding roughly to specific projects.

3.1 Los Alamos National Laboratory

The corpus from LANL contains files used in their supercomputing environment. We refer to this as the *scientific* corpus, and it most closely resembles the structure and intent of the classical view of long-term storage as tertiary storage. The corpus contains approximately 60 million files, totaling 1.3 PB spread across disk and tape. When a user is allocated compute time, he or she is allocated a top-level directory in the archive.

We have 13 months of two daily histogram reports collected over this corpus from a daily crawl of the system's inode metadata by FSstats [16]. One daily report covers the entire file system. The second covers each top-level directory corresponding roughly to summaries of individual projects. Table 3 describes the histograms we used. Note that atime (access time) tracking was explicitly disabled in the file system, so we could not effectively analyze retrieval patterns.

Field Name	Example	Null
Record ID	123555	No
Date 1	10-10-1910	Yes
Date 2		Yes
Type	Marriage Record	No
Ingest date	11-12-2008 12:25:06	No
Modify date	9-4-2009 12:52:00	Yes
Num. of objects	0	No

Table 4: Historical record metadata. A yes in the Null column indicates the value may be null. Number of objects is the number of digital objects associated with a record, possibly zero. The two date fields are used to hold record specific dates, such as birth and death times.

Field name	Example
Site	A00268
Site type	Surface Water
Parameter	Flow
Period of record	1997
File name	GW_DEPTH_POINT_DATA
File size	13050
File type	Plot

Table 5: Water record metadata, and representative values. Unique records are identified using a (Site, Period of Record, File Name) tuple.

3.2 Washington State Digital Archives

One public corpus we examined is a collection of digitized, historical artifacts—such as census information, military records, photographs, and land records—stored in an SQL database at the Washington State Digital Archives. We refer to this as the *historical* corpus. At the time of capture, it contained approximately 90 million records, 28 million of which are accessible via their public web interface; the rest must be accessed on-site. Records occasionally move between public and private status based on content or explicit request. In this study, we focus on the publicly available records, since this is the only portion of the corpus covered in the provided user access logs.

We obtained two logs for this corpus, spanning September 27, 2007 to June 17, 2010. The first log details per-record metadata, described in Table 4. The second is a user access log that records accesses to individual records. Each record is linked to zero or more digital objects—such as photographs and documents—but each digital object is only associated with one record. The trace does not note whether the digital objects linked to that record were retrieved. Further, while the access log provides information to group accesses from the same session, we cannot link different sessions to specific individuals or hosts.

It should be noted that our logs only reflect user retrieval of records within the corpus database and do not reflect access to any other content, *e.g.* HTML pages. Additionally our logs do not track the activity from data migration or integrity checking processes. As we discuss further in Sections 4 and 5, these administrative processes actually make up the dominant fraction of accesses.

3.3 California Dept. of Water Resources

The final corpus in our study, the *water* corpus, is a relatively small set of water table reports consisting of 57,000 records. We have two traces for the water corpus. The first is a set of update logs from approximately weekly and quarterly batch scripts. Each update log notes the records written to, the date, and record metadata, summarized in Table 5. The second is a set of access traces consisting of a per-user access log, where each entry notes the IP address that retrieved the record, as well as the site, period of record, and record retrieved. As with the historical corpus, the logs here do not reflect accesses to general web content, only downloads of the reports themselves. Similarly, if there are any internal indexing or integrity processes running, they are not reflected in our logs.

In the update trace, we identify a unique record using a tuple of site name, period of record, and file name. Complicating this, however, was an intermittent change in file naming conventions that made it difficult to map old names to new, introducing the danger of over-counting files and mapping updates to incorrect file names. To address this, we only count updates to files that map to names in existence on the last day of the update log. Though this discards approximately 50% of the 1.7 million updates, it ensures we have both a correct file count, and an accurate lower-bound on file update activity; more updates may have been required to keep the relevant files up to date, but no fewer.

4 Analysis

Our analysis is motivated by a hypothesis covering three primary areas. First, as media capacities and costs have changed, the tertiary storage use case has seen increased use of hard drives. Second, with the broadening variety of archival use cases, "write-once" does not accurately describe modification behavior in all long-term data stores. Third, it is similarly not accurate to characterize all long-term storage as "read-maybe".

We begin by comparing the scientific sketch of the tertiary storage archive at LANL to the archive Miller and Katz describe at NCAR, since theirs was the most recent study of a large—for the time—tertiary storage system. Following that, we examine update behavior in modern content preservation systems. Finally, we analyze ac-

	Disk (TB)	Tape (TB)	Total (TB)
1993	0.1	26.2	26.3
2010	300	1000	1300
Ratio	1:3000	1:38.2	1:49.4
CAGR	60.2%	23.9%	25.8%

Table 6: Tertiary storage comparison between the NCAR system in the 1993 Miller study [25], and the current LANL system, showing the ratio between the 1993 value and the 2010 value, and compound annual growth rate (CAGR).

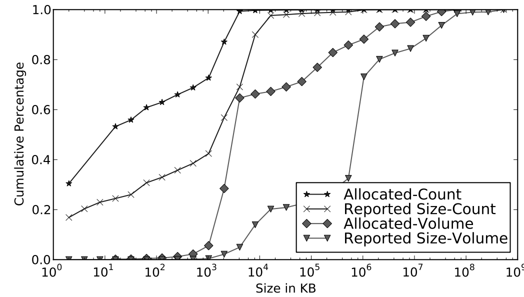


Figure 1: CDF of reported file sizes and allocated space at the end of the scientific trace. Volume refers to the aggregate amount of storage consumed (in KB), count to the number of files. Sparse files have larger reported file size than allocated space, causing the difference between the curves.

cesses to complete our investigation into the validity of the "write-once, read-maybe" assumption [15, 37, 43].

4.1 Tertiary Storage Evolution

Compound Annual Growth Rates *Compared to the previous study, total corpus size exhibited a compound annual growth rate (CAGR) of 25.8%. The CAGR for tape was only 23.9% compared to 60.2% for disk.*

As summarized in Table 6, the scientific corpus from LANL contained about 1.3 PB at the end the report period, and is hosted on 1000 TB of tape, and 285 TB of hard drives. Note however that the LANL corpus is continuing to grow, and is estimated to grow to over 2 PB by the end of 2010. While the LANL administrators have designed the tape library to expand to partially accommodate this growth, overall system growth will still be dominated by disk. Compared to the scientific corpus of the NCAR study, the total corpus exhibited a CAGR of 25.8%. However, most of the growth in capacity occurred in hard drive storage. Compared to the earlier study, our data shows a hard drive CAGR of 60.2%. Note, we restricted our hard drive comparison to a holistic high level view, as the NCAR archive did not use commodity hard drives, relying instead on proprietary storage modules. Interestingly, that hardware was fairly old when it was studied in 1993; the IBM 3380 systems [20] in the NCAR archive were introduced in 1980, with the final revisions released in 1987.

Similarly, the NCAR archive used IBM 3480 tapes, with a capacity of 200 MB per tape. This format was introduced in 1984, making it 9 years old at the time of the study; by 1992 IBM was producing the fourth generation 3490E IDRC tapes, with a capacity improvement of 12 times that of the 3480. By comparison, the current LANL archive uses the relatively recent LTO-4 format tapes, with 1 TB of capacity, 5000 times the storage of the IBM 3480 tapes. Even with the potentially exaggerated gap in tape capacity, we still see a CAGR of 23.9%, which lags slightly behind the total storage CAGR of 25.8%. The impact of this disruptive shift towards more disk-centric tertiary storage on file usage and migration patterns is of keen interest; however, the LANL sketch lacks access time and user behavior information so we must relegate its investigation to future work.

File Sizes *Files between 1 and 2 GB consumed 40% of reported storage, but many large files were sparse.*

Moving to the record level, we next examined file sizes within the corpus. Figure 1 shows a CDF of file sizes calculated from the last day's histogram in our dataset. Nearly 50% of the data written in the NCAR study consisted of files between 10 and 100 MB; in contrast, we found that 40% of the total reported usage in the LANL corpus consisted of files between 1 and 2 GB.

Interestingly, however, when comparing the reported file sizes to the amount of storage space actually allocated to files, we see that 60% of allocated space is consumed by files sizes between 2 and 8 MB. Thus, while the bulk of storage is consumed by files that are considerably larger than the previous study, those files tend to be sparse; over a petabyte is accounted for when looking at file sizes, but only around 100 TB is actually allocated. This behavior may be partially attributable to scientific super-computing's use of shared checkpoint files [10].

4.2 Data Modifications

Tertiary Storage Updates *Despite a growing reliance on hard drives, traditional tertiary corpora continue to be fairly static; 60% of the LANL content we observed was not modified in nearly a year.*

We begin by examining the private scientific dataset, the corpus most similar to the previous study of tertiary storage [25]. Figure 2 is a heat-map showing the fraction of individual records (files) in the corpus that fall into various age ranges over time. The y-axis corresponds to histogram bucket ranges, and the x-axis the day of the trace. The heat-scale on the right maps shade to the total fraction of archive contents. Thus, a corpus that exhibited a high degree of content modification across many files would be warmest along the base of the y-axis; many records would have a recent modification time.

When records are ingested into the archive, they tend

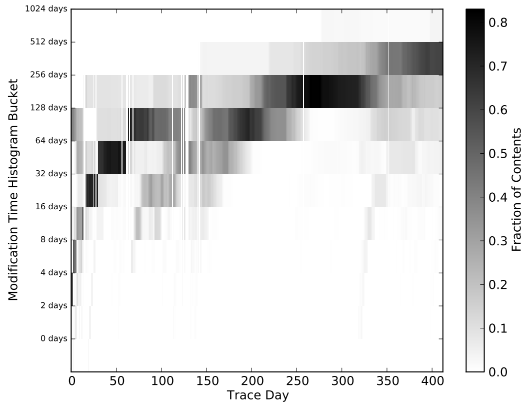


Figure 2: Heatmap of the scientific corpus's daily record modification histograms over 400 days. The color indicates the fraction of the archives contents, x-axis the day in the trace, and y-axis the modification time histogram bucket. For example, on day 275 of the trace, 80% of archive contents received their most recent modification 128–256 days ago. Note y-axis is log scaled (to match the histogram report), and we truncate it after 1024 days, as most contents are below that age.

to be ingested in batches, and they maintain their existing modification time, explaining why the temperature warms in areas other than the histogram bucket for 0–2 days. At the start of the trace, the archive ingested a batch of files with recent modification times. In Figure 2 this appears as a warm area near trace day 0, for histogram bucket with files 2–4 days old. As the trace proceeds, those records remain static, and age steadily. This is seen on the heat map by the high temperature region of the histogram moving from the 2–4 day bucket to the 64–128 day bucket as the trace proceeds from day 0 to day 100. Other ingests follow the same behavior, as seen near days 60, 100, 150 and finally 310.

Despite the growing use of hard drives, our results show that aggregate modification behavior in traditional tertiary storage is still much the same as it was at NCAR over 15 years ago. That study showed that 65% of files referenced in the 24 month trace were only written to a single time, and over 20% were read but never written to. Similarly, at the end of our dataset's duration, despite only having a 13-month trace, we see that approximately 60% of corpus records had modification dates more than 256 days in the past.

Content Storage Mutability *Long-term content corpora are highly dynamic: 50% of records in the water corpus received 5 or more updates, often stemming from automatic data management processes. Similarly, 75% of the historical corpus saw at least one update during the trace.*

To compare tertiary storage update times with those of long-term content repositories, we examine data updates within the publicly accessible water corpus. One complication to note is that we can only deduce record creation in the water sketch by noting a record's first appearance in an update log. In our analysis, we associate each record with a list of updates generated from the update logs. As discussed in Section 3, we filtered the updates such that only updates mapped to files present on the last day were included in our analysis. Though this introduces the danger of under-counting updates, it ensures that our results remain conservative and removes potentially misleading update counts caused by record renaming.

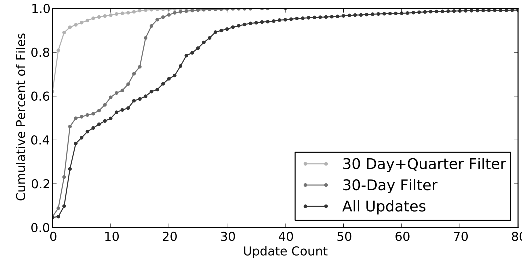


Figure 3: CDF of records, showing the number of updates per record over the duration of the water sketch's update trace.

Examining the logs in the water sketch reveals a surprisingly high number of updates caused by corpus management: automatic policy rules frequently overwrote generated reports, whether or not they had actually received updated data. Two scripts in particular generated a large volume of data updates. The first ran approximately weekly, and modified any report that had data updated within 30 days. The second ran on an irregular, but roughly quarterly schedule, and overwrote all reports in the corpus regardless of the last update they received.

To identify the source of updates, we break our analysis into three sets. The first contains all the updates seen by the corpus. To isolate the results of the weekly script, the second set only considers updates that occur to a file after 30 days have passed. We call this the *30-day filter*. The last set takes the results of the 30-day filter, and removes all mass updates that touch over 10,000 records. We call this the *quarter filter*. Using this approach, we can identify a lower bound on the number of necessary updates; more may have been required to keep the relevant reports up to date, but no fewer.

The results, shown in Figure 3, demonstrate behavior that deviates dramatically from the "write-once" assumption of traditional tertiary storage. When no filters are applied, we see that only 40% of the records receive 5 or fewer updates, and those that receive 20 or fewer updates still only account for around 65% of all records. Applying the 30-day filter, we see that a significant fraction of the corpus still receives 5 or more updates. We observe a

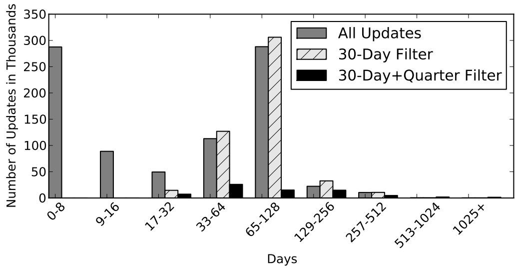


Figure 4: Histogram of inter-update arrival times for all records in the water corpus.

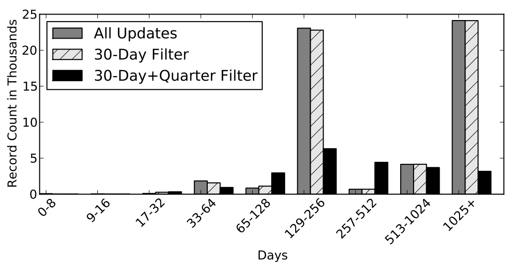


Figure 5: Histogram of time between a record's creation, and its last update within the water corpus. Note records that receive no updates are not counted.

shift, however, when we filter out the quarterly updates, as 60% of the records receive zero updates. This is still far from the "write-once" scenario; 10% of the records receive 3 or more updates, and 20% receive one or more. Many of these are complete "Period of Record" reports that are running summaries of all prior data for a site.

Content Storage Activity *The water and historical records were both highly active long after their ingest times. In the water corpus, 50% of records received updates more than 256 days after their creation. In the historical corpus 85% of modification times were more than 256 days past the record's creation date.*

When we examine the inter-arrival time of updates, the time between any two consecutive updates to a record, illustrated in Figure 4, we see surprisingly large numbers of records with long inter-update periods. 35% of the approximately 900,000 observed updates occurred after a period of over 64 days. When we apply our 30-day and quarterly filters, we still see 70% and 50%, respectively, of updates occur with an inter-arrival time of more than 64 days, though the total volume of updates drops significantly.

To further investigate update behavior within the water corpus, we examine the range of time over which records were receiving updates. Figure 5 shows a histogram of the time between a record's creation and the last update it received. There are, however, two important points to note with this histogram. First, it does not include

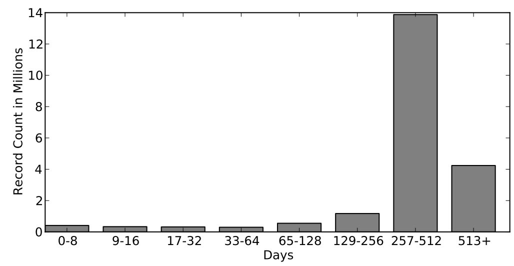


Figure 6: Histogram for the historical sketch, showing the range of time between a record's ingest date and its last modification time. Note that records not updated are not counted.

records that were never updated after creation as they would not contribute to the update count. Second, the record's ingest time relative to the start and end of the trace period impacts the update range we observe; for example, a record ingested two days before the end of trace would have at most a two day range. Nonetheless, this is still a valid method of demonstrating that records continue to be modified long after their ingest time. Using this approach, we see that over 50% of records that receive updates do so over a range of over 256 days. When we apply the 30 day and quarter filters we see the proportions remain roughly the same; approximately 50% of records that received updates were modified over 256 days after record creation. However, the *number* of records receiving updates drops due to the filtering.

In our historical sketch we do not have the same level of access granularity we did in the water sketch; rather, we can only see a record's last modification time. This time reflects the most recent time that any of a record's fields or associated digital objects are modified. This level of detail is still sufficient to show that of the approximately 28 million publicly accessible records, over 21 million had a non-null modification date, meaning that approximately 75% of the corpus content was updated at least once. This is significantly more than that shown in both Agrawal's desktop trace study [2], where over 80% of files remained unwritten each year for over 5 years; and the modern tertiary storage behavior illustrated in Figure 2, where approximately 80% of the corpus remained unmodified.

Update time ranges were also similar between the water and historical sketches. When we look at the time between a historical record's ingest and its last recorded modification time, shown in Figure 6, we see that 85% of the modification times show a difference of 256 or more days from the record's creation.

The surprising amount of update activity we see across both the water and historical corpora is made possible by the use of cheap random access media. The use of tape or optical media based in the face of so many modifications would be problematic, as they require significant extra hardware. Additionally, the long access times of such media are a barrier to frequent modification of data.

4.3 Accesses

In our water and historical logs, record modifications appear as session-less, system-generated operations. In contrast, record accesses are associated with a distinct session. Thus, our access analysis looks at both aggregate and session-oriented access behavior. As mentioned previously, the LANL archive disabled access time updates, and histogram reports were generated from metadata mirrors. Thus, we are unable to analyze user accesses in the scientific corpus.

Large-Scale Retrievals *Accesses are dominated by a few, often machine generated large-scale retrievals, such as a Google crawl or integrity checking process.*

In the historical sketch, we observe approximately 5.88 million distinct accesses between September 27, 2007 and June 16, 2010. The accesses are across 1.05 million user sessions, accessing 2.28 million distinct records. From discussions with the repository administrators, we also know that *all* records are integrity checked monthly. Though only 8% of the 28 million publicly available records were accessed by users over 3 years, 100% of the records were read via the integrity checking process each month. If we consider integrity checking to be equivalent to record retrieval, then less than 1% of reads come from end-users. Even assuming that files are checked for integrity once per year, only 10% of read traffic would come from users. This finding has significant implications on archive design. Effective, low-latency end-user retrievals are critical to the perception of a useful system, but only make up a small fraction of the actual workload. On the other hand, administrative processes, which make up the bulk of accesses and are critical to the integrity of the system, are typically less latency-sensitive. Thus, as we discuss further in this section as well as in Section 5, a separate batch interface for bulk accesses could provide significant benefit to future systems.

In the water sketch, we see roughly 98,000 distinct retrievals between August 28, 2007, and July 1, 2010. By artificially grouping accesses originating from the same IP address that arrive within 10 minutes of one another, we identify approximately 8500 user sessions. We choose 10 minutes as the threshold based on our observation that the number of sessions created by our grouping method taper off after approximately 10 minutes. We exclude approximately 1200 retrievals that had a null value for their files, accounting for approximately 1% of all retrieval requests.

We find that approximately 70,500 of the 98,000 total accesses in the water sketch originated from Google, and 27,000 from other users. Since there were 57,000 records in the last quarterly update, and non-Google users made 27,000 requests, we observe that no more than 50% of the archive's contents could be retrieved by non-Google users. On the other hand, Google likely requested nearly all of the reports given the methodical nature of their crawls, though we cannot conclusively state this given the file renaming issues we noted earlier.

LRU Caching *LRU caching is moderately effective at absorbing per-session record re-retrievals and flash traffic. As a whole it is ineffective at absorbing day-to-day traffic due to limited record popularity.*

One peculiar behavior we notice in both the water and historical sketches is significant numbers of user-sessions re-retrieving the same record in the same session, often within a few seconds. Communication with system administrators and architects yielded no explanation for this odd behavior. We note that these re-retrievals accounted for 3% of the retrievals in the water-sensor log, and nearly 35% (2.04 million) of the record retrievals for the historical archive. These re-retrievals have a noticeable impact on our results and implications for archival system design.

From the daily access counts in the historical sketch shown in Figure 7(a), we observe that the number of accesses on any given day is relatively stable, and exhibits a slow growth trend. We do, however, observe a number of moderate spikes, and one large spike around day 900.

A microanalysis of the large spike finds that it is comprised almost entirely of sessions that only retrieved a single record, and that the records retrieved were predominantly (over 90%) photographs. Further, the upper quartile of distinct records retrieved in the spike received 5 or more accesses, as opposed to the usual 1 or 2 on most prior days we examined. Consultation with the system and corpus administrators yielded no clear explanation for behavior seen in the spike. Further, we confirmed that external indexers, such as Google, only have access to around 6000 records, ruling out a possible explanation.

To explore the potential effectiveness of caching on daily traffic and spike mitigation, we ran our daily access count analysis with two different sizes of a simple LRU caching filter: 0.01% and 0.1% of the total number of retrievals, corresponding to 500 and 5000 records. When we include re-retrievals during the same session in the count, even a small cache is shown to be moderately effective at absorbing accesses, with overall hit ratios of 37% and 38% for a 500 and 5000 record cache, respectively. When we remove the re-retrievals the cache effectiveness plummets, exhibiting an overall hit ratio of less than 7% for even the 5000 record LRU cache.

Interestingly, as Figure 7(c) shows, the cache is effective at reducing the magnitude of several of the access spikes. Even the small cache absorbed nearly 50% of the traffic during the day 900 spike. Thus, while their overall impact is low, read caches in long-term content stores may be useful for handling flash traffic and record re-retrievals.

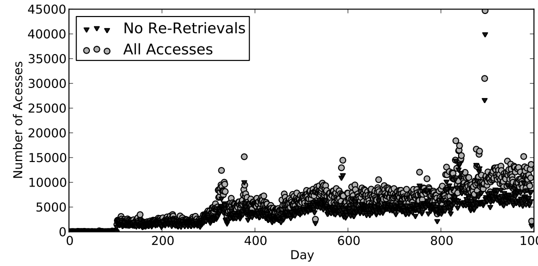
Next, we examine daily access counts and cache effectiveness for the water repository, illustrated in Figure 8. One of the first things we note is an extended access spike, approximately between days 700 and 750. Using a reverse IP look up we confirmed this was Google slowly crawling the repository contents. In total, Google accounted for over 70% of *all* record retrievals. For our subsequent analysis of the water accesses, we filtered the large, external index crawl from the dataset. Note that while other user sessions did occasionally exhibit bot-like behavior (fast inter-retrieval times, and mass retrievals) we could not conclusively identify them as such, and left them in the trace.

In the water sketch, as with the historical sketch, we see a moderate number of re-retrievals within user sessions, and examine the impact of caching with and without these re-retrievals, shown in Figure 9. As with our previous observations, with re-retrievals included, we see low to moderate cache effectiveness with hit rates of 12% for a cache size of 10 records, and 17% for 100 records. When we eliminate session level re-retrievals the hit ratios drop to 2% and 8% respectively. In the water sketch, however, caching remains largely ineffective even on days with significantly increased traffic, as figure 9(b) illustrates.

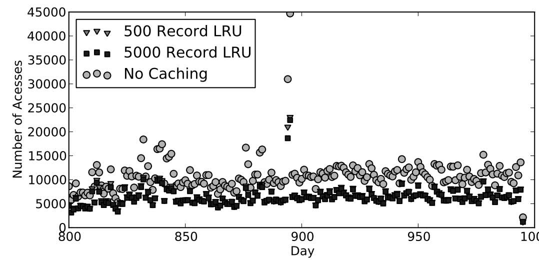
Per-Session Behavior *50% of users' sessions only retrieve a single record, though this accounts for fewer than 10% of the total retrievals.*

In Figure 10, we illustrate the number of retrievals per session with and without re-retrievals. Interestingly, for both the historical and water datasets, we see that over 50% of sessions only retrieve a single record. Further we observe that the distribution quickly flattens out, with approximately 90% of sessions retrieving 15 or fewer records. Since many sessions are coming from humans interacting via a web interface, the time between user retrievals is relatively long, often seconds to minutes.

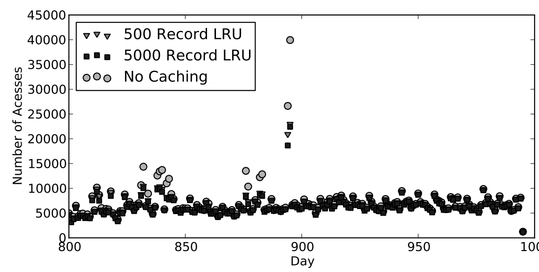
While 50% of sessions—with re-retrievals—only retrieve a single record, those sessions in the historical trace account for fewer than 10% of the total retrievals, and fewer than 5% for the water sketch, as shown in Figure 11. The vast majority of data was accessed from larger sessions. In the historical corpus, 40% of all accesses come from sessions of more than 20 retrievals, and nearly 80% in the water sketch are made during similarly large sessions. In the water sketch, this skew is due to a Google index crawl of the corpus that occurred over several large sessions, each retrieving hundreds to thousands of records. The prevalence of these large mass retrievals, much like the wholesale integrity checking, suggests the utility of a batch interface, as we discuss further in Sec-



(a) Complete historical corpus daily access rates with and without re-retrievals



(b) Historical corpus cache impact with re-retrievals for days 800-1000.



(c) Historical corpus cache impact without re-retrievals for days 800-1000.

Figure 7: Daily access counts to the historical corpus with and without re-retrievals and the associated LRU cache impacts. If a retrieval was absorbed by a cache hit it was not counted.

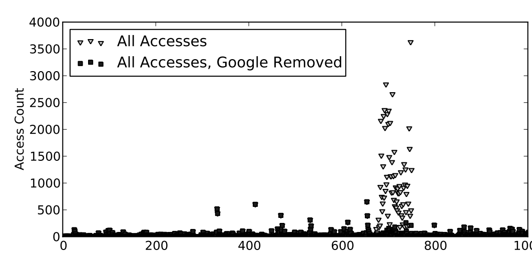
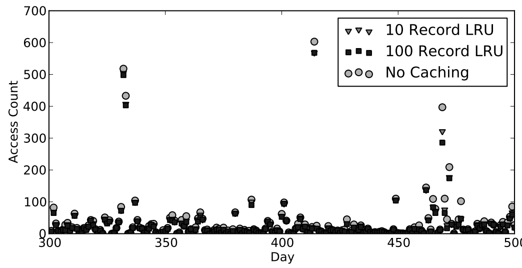
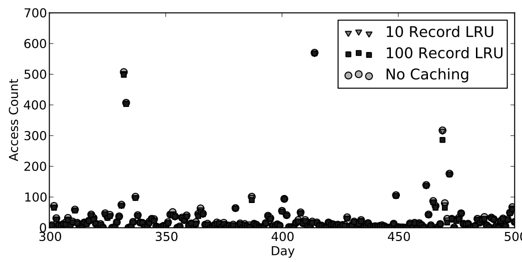


Figure 8: Daily access counts to the water corpus with and without the retrievals by Google



(a) Water corpus daily access counts with re-retrievals, days 300-500.



(b) Water corpus daily access counts without re-retrievals, days 300-500.

Figure 9: Daily access counts for the water corpus with and without re-retrievals, and associated LRU cache impacts of size 10 and 100 records. If a retrieval was absorbed by a cache hit it was not counted. Google accesses have been filtered out. Note cache impact is nearly eliminated when re-retrievals are removed.

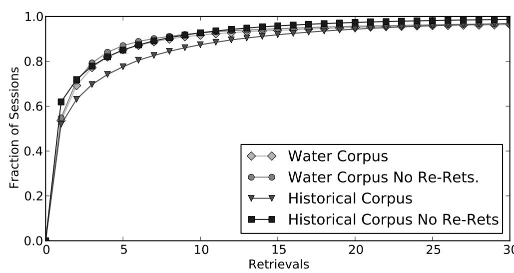


Figure 10: CDF of accesses per session showing the number of records retrieved per session in the water and historical corpus, with and without per session re-retrievals. We truncate at 30 accesses, the few large sessions would distort the plot.

tion 5.

Intra-Session Access Locality *Users' sessions tend to show strong content correlation, retrieving a limited number of content types. Inter-session (system-wide) content popularity is extremely long tailed.*

Next, we look at content popularity, independent of sessions, to see if we can identify a subset of records or content types that account for a disproportionate fraction of accesses. Figure 12 shows that all of the distributions exhibit a long tail, with the exception of the types-based popularity for the historical archive. For example, the

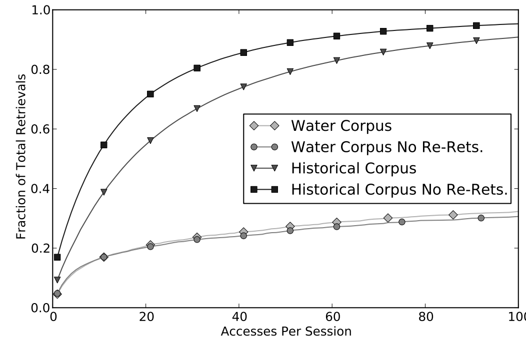


Figure 11: CDF showing what fraction of total retrievals were contributed per session size with and without re-retrievals for the water and historical corpora. Note we truncate the x-axis at 100 to maintain readability as retrievals to the water corpus dominated by a few large sessions; sessions with over 300 retrievals account for 60% of the total retrievals.

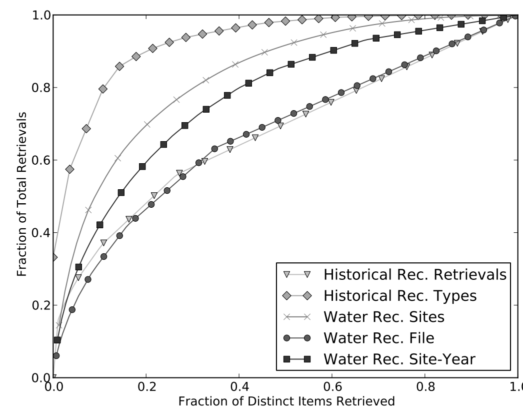


Figure 12: CDF of record popularity by individual record, and distinct content type. Note that x-axis values are ordered by popularity, that is, the most popular items are plotted first. With the exception of the historical corpus record types all CDFs are subsampled.

sites—*i.e.* water well location—in the water set exhibit the second strongest popularity affinity with 20% of sites accounting for 60% of accesses, but the next 20% of sites only account for another 20% of accesses. At file level granularity, this trend becomes even more pronounced. We note, however, that the file naming issues within the water-sensor archive may mask some amount of file popularity. The content popularity distribution corroborates our early findings showing LRU read-caching to be largely ineffective; while certain categories of data are more popular, individual records do not appear to be particularly more popular.

We next examine access locality within sessions to see if individual user sessions tended to access a single or few types of content. In our analysis, we first

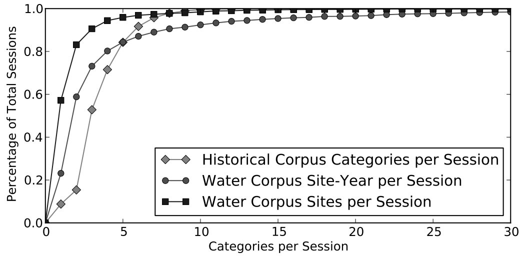


Figure 13: CDF showing the number of different content categories retrieved per session. In the historical corpus, records have an associated category. In the water corpus, we examine how many distinct sites as well as site year combinations are accessed per user session. We truncate at 30 types to avoid a distorted plot.

remove re-retrievals from all sessions; we then exclude sessions in which only a single record is retrieved. Additionally we eliminate records from the historical sketch that are found to be missing metadata (less than 0.5% of retrievals), as well as those with the category listed as "Restricted Type"; these records were originally public and subsequently moved into the private archive, so we cannot determine their category. The restricted type retrievals account for 12% of accesses after removing re-retrievals and excluding singleton sessions.

Figure 13 shows that, across both the historical and water traces, individual user sessions tend to retrieve strongly related content. We see that nearly 50% of sessions in the historical trace retrieve three or fewer record types, and similarly 50% of sessions in the water trace retrieve data pertaining to only a single site. When we include the year, 25% of sessions retrieve records within a single site-year combination, but still exhibit strong per-session content locality.

Across sessions, however, a wide variety of content and individual records are retrieved. This is evident in the poor cache performance, as shown earlier in Figures 7 and 9. The strong individual session locality does suggest that grouping data based on content along with prefetching may be effective [42], provided the content type has a sufficiently small number of records. For example, this would likely be more effective for the water corpus where any given site-year combination rarely has more than 15 or so reports at a few tens of kilobytes apiece, than for the historical corpus where there may be many millions of records connected by an associated category. In contrast the lack of individually popular records we noted earlier impacts systems that aim to conserve energy by duplicating or migrating commonly used data, as they require relatively fine grained, predictable system-wide accesses in order to be effective [12, 28]. This is because while we can make strong statements about individual session behavior, aggregate system wide activity is largely unpredictable in regards to the popularity of records.

5 Lessons Learned

Our investigation into the behavior of tertiary storage and long-term public content corpora revealed a number of high-level lessons. In this section we interpret our observations, and discuss their implication for long-term storage system design and future research directions.

5.1 Read-Write Behavior

While the scientific corpus exhibited the classical tertiary storage behavior of fairly static content, the water and historical corpora showed a surprisingly high degree of change; data modifications were frequent, widespread, and over a much longer duration than we expected. However, even in the scientific corpus we observed a storage medium shift from tape to disk, suggesting that file immutability should be an enforceable policy independent of the media type. This flexibility is even handy for explicitly immutable data–such as compliance stores–as such datasets often have a specific expiration date, after which the owners would like the data to be immediately deletion.

With respect to reads, contrary to assumptions established by tertiary storage, we found that both the water and historical corpora were quite active. While user requested reads were relatively rare, data management tasks, indexing requests, and the inevitable migration of long-term data, make the "read-maybe" pattern patently false; *all* content is eventually read, and it is often read *en masse*. This could severely impact the effectiveness of system designs that rely on low read-rates. For example, systems that rely on spun down disks for power savings may overestimate the cost savings they can deliver [28, 37], unless accesses can be tightly controlled and scheduled.

Further, our results suggest a potential danger in optimizing for the wrong operations. In the water trace, the vast majority of total accesses were from a few large-scale requests—such as Google crawls—with the remainder originating from user accesses that often only retrieve a single record. We argue, however, that these small numbers of user requests are latency sensitive, and critical to the users' perception of effective, long-term storage.

Within these critical user sessions, we found that there were a few favored content types, and the same data was often requested multiple times within a single session. Across sessions, however, it was much more difficult to identify popular content. Thus, aside from assisting with the re-retrieval problem, strategies that rely on migrating popular data may be ineffective at best, and harmful at worst [28, 47]; in a disk spin-down scenario, such movement could incur additional energy penalties for little or no benefit. Depending on the scale, this may also make the use of tape-based architectures, and those based upon immutable media types significantly less efficient.

It is important, however, not to downplay the importance of bulk accesses, since they dominate an archive's workload. Recall that we observed integrity checking accounting for 99% of read accesses in our analysis of the historical sketch. The disparity in large and small access properties suggests that current archive interfaces are insufficient. Since our data suggests that these large-scale accesses are often latency-insensitive administrative processes, we propose an asynchronous batch interface for large requests as a complement to the traditional single record interface. The benefit to the system is that such a request would provide full *a priori* knowledge of the records in the requests, allowing the archive to optimize its resource scheduling most effectively.

Such an interface would allow a client to specify the set of records desired, a schedule of when it needs the requests fulfilled by, and a means to alert the client when the request is complete. For writing, this could be useful for data management functions: we observed that public content archives provide anonymous read access, but writes came from the system itself. In the case of external indexing services, such an interface could help shift the large-scale requests from appearing parasitic at an energy cost and workload spike standpoint, to a more symbiotic relationship; the indexing service receives the data, and the archive can efficiently provide the means for users to find it. To prevent pathological use of the traditional, single-record access interface, archives could utilize strategies such as throttling or retrieval caps.

5.2 Understanding Corpus Behavior

While the results presented in this paper have helped to expand our understanding of long-term corpus behavior, a number of trends motivate the need to understand the aggregate behavior of multiple corpora hosted by a single archive. First, the growth of cloud storage marks a shift towards centralized data centers. Second, increasingly digital workflows have spurred the proliferation of small and mid-sized corpora. This leads to potential problems in optimization, as superficially similar corpora could be hosted on the same archive, despite the fact that they may benefit from different configurations. For example, while both the historical and water corpora showed strong content locality within user sessions, their record granularity is vastly different; a single record type in the water corpus may only contain 20 or 30 records, while in the historical corpus one record type may have millions of records. An optimal retrieval technique for one may be pathological for the other.

Even within the scope of a single corpus there is more work to be done with our current sketches. First, we would like to investigate both temporal and content locality in record ingests and updates. Second, we plan to more closely examine short-term behavior within the traces, to see if short-term behavior in accesses and updates is similar to that of the long-term behavior we observed. Finally, as an aid in the development of effective long-term repository systems, we hope to publish a series of workloads based on these traces. This would help in the evaluation of archival designs, and in the reproducibility of published results [1].

Finally, as we have noted, there is a wide gamut of systems that fall within the category of long-term data repositories. To that end we hope to locate and examine additional traces across all areas of archival storage in order to identify and quantify their common, and divergent characteristics. For example, within the tertiary storage area, more granularity and study is needed; the sketch we obtained from LANL had disabled recording of access times, impeding our ability to analyze read behavior.

5.3 Tracing Difficulties

Acquiring high-quality trace data for this study proved to be a vexing challenge. The worst examples we encountered were logs with no field descriptions or supporting documentation, making them effectively unusable. However, with immeasurable assistance from the archive owners, we were able to obtain and refine several relevant and useful traces. To this end, we see a strong need to continue the development of tools that enable organizations to easily collect and efficiently store descriptive and relevant long-term access data [5]. This is particularly true for archival storage, since, in contrast to storage systems such as those used in enterprise and the desktop, traces *must* be gathered over years. If data are not gathered properly, reacquiring a trace can take years, perhaps preventing the trace from driving an analysis to guide the design of the archival storage system's successor. In addition, there is a need for consistent tracing standards to ensure ease of use and readability far into the future.

Good tracing tools would also be a boon to archive operation, as our observations revealed counter-productive behavior of which the system administrators and architects were unaware. For example, in the historical sketch, our analysis revealed frequent record re-retrievals within the same session; the administrators did not know of this behavior, and were unable to explain it. Additionally, the water corpus exhibited many needless overwrites coming from data management processes. These irregularities highlight the need for good analysis tools to help administrators identify pathological behavior within the system.

Finally, *data-centric* (corpus-centric) tools for long-term tracing would be useful for several reasons. First, seemingly trivial actions can make analysis of longer-term trends extremely difficult. For example, in the water trace, files were identified by path names, but file renames did not capture the information needed to link old path names to new path names. Second, as systems become increasingly distributed, there may be multiple instances of the same corpus in a single system, motivating the need for tracing tools that can provide a holistic view across archives [39]. Third, given the intended long lifetime of many corpora, data will live on many systems over its life. In order to understand how data behavior evolves, a long-term trace must extend beyond the lifetime of any single system.

Even with useful tools and traces, the importance of good communication with system architects and administrators cannot be overstated. They can provide information that is not captured by trace reports, significantly altering conclusions. For example, in the historical sketch, communication with the administrators was instrumental in understanding the scale of user request traffic and system-generated integrity checking traffic, and in the water sketch the administrators explained the nature of their periodic batch processes.

6 Conclusions

As ever-growing quantities of our society's data are stored in long-term digital archives, it is increasingly important to understand how these archival storage systems are used, and how they behave. To address this question, we presented a detailed analysis of behavior in three archival storage systems, including both scientific and public data. Our study provides the first examination of a large tertiary storage system in over 15 years, and the first ever analysis of the behavior of public content archives. Based on our findings, we have made concrete suggestions for both archival storage system implementers and administrators.

By analyzing the LANL sketch, we were able to see how tertiary storage archives have evolved in the last seventeen years. Our analysis reveals that, compared to the NCAR system studied in 1993, the LANL corpus exhibits a CAGR of 25.8%. Further, hard drives play an increasingly important role in the archive; the NCAR system had a disk to tape ratio of 1:262, in sharp contrast to the LANL archive's ratio of 1:3.3. Despite this shift, the update patterns between then and now are largely unchanged. Additional access time information is needed to fully understand how the random access performance of hard drives is being utilized.

The public water and historical sketches demonstrated how long-term storage now covers a wide range of behavior. We found that the contents were both accessed and modified frequently; 75% of the historical corpus saw at least one update over the trace period, and 50% of the water corpus saw 5 or more writes. Access traffic was dominated by a few large-scale requests such as data management scripts and Google crawls. This behavior, along with the latency sensitivity of small requests suggest that two different interfaces are called for: one for the small, but critically important, user requests; and another for the large-scale, but latency-insensitive bulk requests. These results of our workload study and the guidelines developed from the results will help archival system designers in the construction and maintenance of archives that can efficiently and effectively preserve society's digital legacy for future generations.

Acknowledgments

Thanks to Gary Grider of LANL, Greg Smith of the Department of Water Resrouces, and June Timmons and Brian Smith of the Washington State Digital Archives. This research was supported in part by the National Science Foundation under grants CNS-0917396 (part of the American Recovery and Reinvestment Act of 2009 [Public Law 111-5]) and IIP-0934401, and by the industrial sponsors of the Storage Systems Research Center and the Center for Research in Intelligent Storage.

References

[1] AGRAWAL, N., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. Generating realistic *impressions* for file-system benchmarking. In *Proc. of FAST '09* (Feb. 2009), pp. 125–138.

[2] AGRAWAL, N., BOLOSKY, W. J., DOUCEUR, J. R., AND LORCH, J. R. A five-year study of file-system metadata. In *Proc. FAST '07* (Feb. 2007), pp. 31–45.

[3] Alaska's digital archives. `vilda.alaska.edu`, 2010.

[4] ANDERSON, E. Capture, conversion, and analysis of an intense NFS workload. In *Proc. of USENIX '09* (2009).

[5] ANDERSON, E., ARLITT, M., CHARLES B. MORREY, I., AND VEITCH, A. DataSeries: an efficient, flexible data format for structured serial data. *ACM SIGOPS Operating Systems Review (OSR) 43*, 1 (Jan. 2009), 70–75.

[6] BAIRAVASUNDARAM, L. N., GOODSON, G. R., PASUPATHY, S., AND SCHINDLER, J. An analysis of latent sector errors in disk drives. In *Proc. SIGMETRICS '07* (June 2007).

[7] BAIRAVASUNDARAM, L. N., GOODSON, G. R., SCHROEDER, B., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. An analysis of data corruption in the storage stack. In *Proc. FAST '08)* (Feb. 2008), pp. 223–238.

[8] BAKER, M., KEETON, K., AND MARTIN, S. Why traditional systems don't help us save stuff forever. In *Proc. of Hot Topics in System Dependendability '05* (June 2005).

[9] BAKER, M., SHAH, M., ROSENTHAL, D. S. H., ROUSSOPOULOS, M., MANIATIS, P., GIULI, T., AND BUNGALE, P. A fresh look at the reliability of long-term digital storage. In *Proc. of EuroSys '06* (Apr. 2006), pp. 221–234.

[10] BENT, J. *et al.* PLFS: a checkpoint filesystem for parallel applications. In *Proc. of the Conference on High Performance Computing Networking, Storage and Analysis* (Portland, OR, USA, Nov. 2009).

[11] California Department of Water Resources water reports. http://www.water.ca.gov/waterdatalibrary/docs/Hydstra/index.cfm.

[12] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Proceedings Supercomputing '02 (SC '02)*, Nov. 2002.

[13] Chronicles of life: Save your memories forever. http://www.chronicleoflife.com/.

[14] Cornell University Library arXiv. http://arxiv.org/, 2010.

[15] DAMOULAKIS, J. Opinion: Tape backup is WORN (write once, read never). http://www.computerworld.com/s/article/9026619/Opinion_Tape_backup_is_WORN_write_once_read_never_, July 2007.

[16] DAYAL, S. Characterizing HEC Storage Systems at Rest. Tech. Rep. CMU-PDL-08-109, Carnegie Mellon University, 2008.

[17] Distributed Active Archive Center. http://daac.ornl.gov/.

[18] GREENFIELDBOYCE, N. Houston, we erased the Apollo 11 tapes. *NPR Morning Edition* (July 2009).

[19] Health Information Portability and Accountability Act, Oct. 1996.

[20] IBM 3380 direct access storage device. http://www-03.ibm.com/ibm/history/exhibits/storage/storage_3380e.html.

[21] JENSEN, D. W., AND REED, D. A. File archive activity in a supercomputing environment. In *Proc. of Super Computing '93* (Tokyo, Japan, July 1993), pp. 387–396.

[22] LEUNG, A. W., PASUPATHY, S., GOODSON, G., AND MILLER, E. L. Measurement and analysis of large-scale network file system workloads. In *Proc. of USENIX '08* (June 2008).

[23] LILLIBRIDGE, M., ELNIKETY, S., BIRRELL, A., BURROWS, M., AND ISARD, M. A cooperative Internet backup scheme. In *Proc. of 2003 USENIX '03* (San Antonio, TX, 2003), pp. 29–42.

[24] MANIATIS, P., ROUSSOPOULOS, M., GIULI, T. J., ROSENTHAL, D. S. H., AND BAKER, M. The LOCKSS peer-to-peer digital preservation system. *ACM ToCS 23*, 1 (2005), 2–50.

[25] MILLER, E., AND KATZ, R. An analysis of file migration in a Unix supercomputing environment. In *Proc. of USENIX '93* (Jan. 1993), pp. 421–433.

[26] National Climatic Data Center. http://www.ncdc.noaa.gov/oa/ncdc.html, 2010.

[27] New York State Digital Archives. http://www.archives.nysed.gov/aindex.shtml, 2010.

[28] PINHEIRO, E., AND BIANCHINI, R. Energy conservation techniques for disk array-based servers. In *Proc. of the 18th International Conference on Supercomputing* (June 2004).

[29] PINHEIRO, E., WEBER, W.-D., AND BARROSO, L. A. Failure trends in a large disk drive population. In *Proc. FAST '07* (Feb. 2007).

[30] QUINLAN, S., AND DORWARD, S. Venti: A new approach to archival storage. In *Proc. of FAST '02* (Monterey, California, USA, 2002), USENIX, pp. 89–101.

[31] ROSELLI, D., LORCH, J., AND ANDERSON, T. A comparison of file system workloads. In *Proc. of the USENIX '00* (San Diego, CA, June 2000), USENIX Association, pp. 41–54.

[32] Sarbanes-Oxley act 2002. www.soxlaw.com.

[33] SCHROEDER, B., AND GIBSON, G. A. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *Proc. FAST '07* (Feb. 2007), pp. 1–16.

[34] SMITH, A. J. Analysis of long term file reference patterns for application to file migration algorithms. *IEEE Transactions on Software Engineering 7*, 4 (July 1981), 403–417.

[35] SMITH, A. J. Long term file migration: Development and evaluation of algorithms. *Communications of the ACM 24*, 8 (Aug. 1981), 521–532.

[36] STORER, M. W., GREENAN, K. M., MILLER, E. L., AND VORUGANTI, K. POTSHARDS: secure long-term storage without encryption. In *Proc. of the USENIX 2007* (June 2007), pp. 143–156.

[37] STORER, M. W., GREENAN, K. M., MILLER, E. L., AND VORUGANTI, K. Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage. In *Proc. FAST 2008* (Feb. 2008).

[38] STRANGE, S. Analysis of long-term UNIX file access patterns for application to automatic file migration strategies. Tech. Rep. UCB/CSD 92/700, University of California, Berkeley, Aug. 1992.

[39] THERESKA, E., SALMON, B., STRUNK, J., WACHS, M., ABD-EL-MALEK, M., LOPEZ, J., AND GRANGER, G. R. Stardust: Tracking activity in a distributed storage system. In *Proc. SIGMETRICS 2006* (Saint-Malo, France, June 2006).

[40] TRAEGER, A., ZADOK, E., JOUKOV, N., AND WRIGHT, C. P. A nine year study of file system and storage benchmarking. *ACM Transactions on Storage 4*, 2 (May 2008).

[41] Washington State Digital Archives. http://www.digitalarchives.wa.gov/, 2010.

[42] WILDANI, A., AND MILLER, E. L. Semantic data placement for power management in archival storage. In *Proc. PDSW '10* (2010).

[43] WILDANI, A., SCHWARZ, T., MILLER, E. L., AND LONG, D. D. E. Protecting against rare event failures in archival systems. In *Proc. MASCOTS '09* (2009).

[44] YOU, L. L., POLLACK, K. T., AND LONG, D. D. E. Deep Store: An archival storage system architecture. In *Proc. of ICDE '05* (Tokyo, Japan, Apr. 2005).

[45] ZHANG, Z., LIAN, Q., LIN, S., CHEN, W., CHEN, Y., AND JIN, C. BitVault: A highly reliable distributed data retention platform. *ACM SIGOPS OSR 41*, 2 (Apr. 2007), 27–36.

[46] ZHU, B., LI, K., AND PATTERSON, H. Avoiding the disk bottleneck in the Data Domain deduplication file system. In *Proc. of FAST '08* (Feb. 2008).

[47] ZHU, Q., CHEN, Z., TAN, L., ZHOU, Y., KEETON, K., AND WILKES, J. Hibernator: Helping disk arrays sleep through the winter. In *Proc. of SOSP '05* (Brighton, UK, Oct. 2005), ACM.