Analyzing User Behavior: A Trace Analysis of the NCAR Archival Storage System

Technical Report UCSC-SSRC-12-02
March 2012

Brian A. Madden
madden@soe.ucsc.edu

Ian F. Adams
iadams@soe.ucsc.edu

Joel C. Frank
jcfrank@soe.ucsc.edu

Ethan L. Miller
elm@cs.ucsc.edu

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

Analyzing User Behavior: A Trace Analysis of the NCAR Archival Storage System

Brian A. Madden Ian F. Adams Joel C. Frank Ethan L. Miller
University of California, Santa Cruz

Abstract

In this work we examine the user behavior of the National Center for Atmospheric Research's (NCAR) archive. In particular we characterized user sessions and examined namespace locality of accesses within those sessions. Preliminary results suggest within a single session, repeat file accesses are uncommon, and that accessed files are localized within a small set of directories. This work provides a baseline and preliminary results for a more detailed future study.

1 Introduction

As the need for archival storage systems grows it becomes increasingly more important that we examine and understand how existing systems operate and are used under real-world conditions. Without such research, the systems we build and the cases we optimize for may well be based on incorrect assumptions. The focus of this paper is to examine user access behavior, and the relationship of those accesses to the file system name space within the National Center for Atmospheric Research's (NCAR) large-scale digital archive.

We began our analysis by artificially breaking user actions into sessions, or groups of actions bounded by a time window. Once sessions were established a number of statistics were collected in such as the number of actions users would take in a session, the number of actions taken in a directory, the number of unique files seen in a session, and the number of unique files seen in per directory.

Using these data a number of conclusions were drawn in regards to average user behavior. First, the average user session consists of one hundred actions or fewer. Second, the average user only accesses any given file once (on average) per session. Third, most actions in a session are spread among a small number of directories. Observations such as this could have real impact on policy decisions such as physical data layout and placement within the archive. In addition to providing useful insight to base policy decisions on this work provides initial results that lay the ground work for future study and analysis.

The remainder of the paper is structured as follows. Section 2 provides background. Section 3 describes experimental setup and results. Section 4 provides a discussion of future work. Section 5 examines related works. Finally, section 6 concludes.

2 Background

Archival storage is a loaded term in computer science. To most, the term probably conjures ideas of huge tape silos with robotic arms shuffling tapes around, and endless amounts of never-read old data. In fact, archival storage is a much broader term applying to more than just tape backups; it is an entire sub field of computer science focused on the preservation of both bits as well as their semantic and logical meanings.

While our science is pretty good when it comes to bit preservation there is still much work to be done if we hope to store digital objects for any significant period of time. The unfortunate truth of the matter is that although our technologies for bit preservation are good, they are focused on reliability in the now while long term preservation is often overlooked. The growing amount of digital data coupled with the fact that we are quickly moving to all digital forms of production means that the archival storage is a real problem that must be dealt with sooner rather than later. This work is a first step in understanding current archive and user behavior that will help shape future research and work in the field.

The National Center for Atmospheric Research (NCAR) is a federally funded research and development center that is focused on the research and education of the Earth's atmosphere and related physical, biological, and social systems. As part of their ongoing research NCAR runs large scale simulations resulting in large quantities of output data. Much of this data must be stored for the long term as it may one day be vital to understanding an event, or re-processed by new algorithms to validate new methods.

The data produced by NCAR is stored on their in-house mass storage system (MSS). This storage system is a traditional tertiary storage archive consisting of a number of tape systems as well as a large array of traditional hard disks. The trace spans a three year period from 2008 through 2010. At the start of the trace the archive consisted of approximately 4 petabytes (PB) of data which grew to approximately 11.7PB over the three year span. The corpus is about 80% simulation output, about 15% is observational data used to seed or validate simulations, and less than 5% is system backup or scientific database backup.

3 Analysis Methods and Results

For our analysis we ingested all of the trace data into a relational database. A relational database was chosen in the hopes that it would provide an easy way to perform range queries over timed data. Unfortunately the database proved to be too slow for the general analysis so instead, it provided a substrate for selecting particular subsets of the data for further processing.

After the initial subsets of data were selected further processing was performed by a number of Python scripts. Further details and a discussion of results comprise the remainder of this section.

3.1 Creating User Sessions

Once the data was ingested, a series of queries was run to break user actions into *sessions*, or logical groupings of user actions that are likely to have occurred together. Creating these sessions was an important first step in understanding user behavior as it provides temporal bounds to groups of user actions, and discretizes periods of activity. Having these sessions gives us the opportunity to examine a user's workflow and based on that make intelligent policy decisions to improve performance, energy usage, or reliability.

The general method of constructing sessions was to select all actions from a particular user and sort them by the date and time they occured. Once sorted, each action was placed into a session based on a sliding time window. For any given action, it's time was checked to see if it fell within a time window calculated by taking the previous action and adding a *window length* to it. Three window lengths, five minutes, ten minutes, and fifteen minutes were chosen for the first analysis of the data. Sessions consisting of a single action, or singleton sessions, were filtered out in all of the experiments.

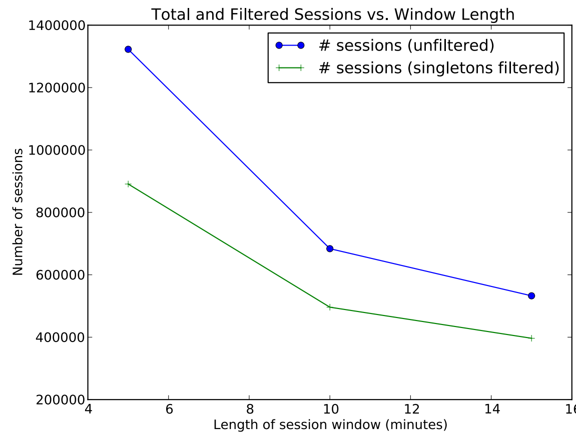

Figure 1: The total number of sessions seen, and total number of sessions used for stat calculation vs. session window length.

The results provided by the three window lengths indicate that more analysis with longer window lengths is necessary. As seen in figure 1 the rate of change of the total observed sessions decreases drastically with the ten minute session window and continues to flatten suggesting that a local optima may be nearby. This optima would be the ideal window length for session creation as it strikes a balance between including too many actions and too few actions in a session.

3.2 Actions per Session

An *action*, in the context of this analysis, is either a read, a write, or a create performed by a user. Action related statistics such as actions per session and actions per directory were captured and used to derive further statistics such as actions taken per directory per session.

Figure 2 shows the total number of observed actions as the window length increases. The *total* number of actions seen increases with window length rather than remaining constant. This is a somewhat un-intuitive result at first glance, but the increase is due to the number of singleton sessions being included that would have otherwise been filtered out. Recall, a singleton session is a session consisting of only a single action. This is a promising result and validates earlier claims that a longer session window is appropriate, as ultimately we would like to group a period of activity followed by single action together.

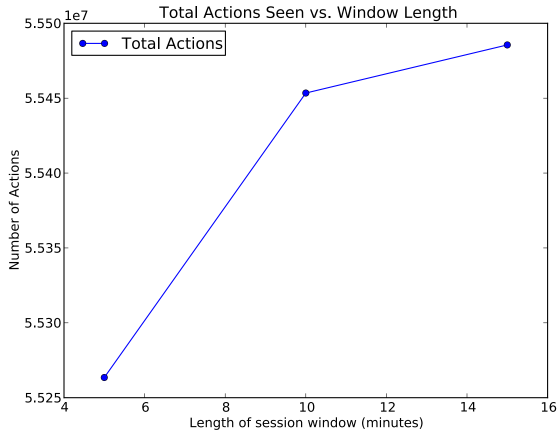

Figure 2: The total observed actions vs. window length.

Figure 3 shows the mean, standard deviation, and median number of actions per session vs. window length. The mean and median number of actions per session shows that typical user sessions consist of about one hundred or fewer actions, though the distance between the two lines shows us an almost two order of magnitude difference. Similarly, the distance between the median and the standard deviation is quite great showing a large variance in the number of actions per session. This variance warrants a deeper inspection and probably suggests a number of automated processes such as a web crawler at work. In future work these large outliers will be filtered out to get a more accurate mean and median session length.

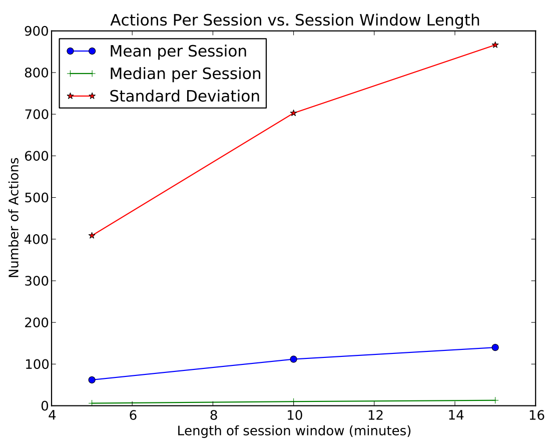

Figure 3: Mean, median, and standard deviation of actions per session.

To complement the graph of mean and median actions per session figure 4 shows an empirical CDF of the number of actions taken per session. This graph shows that approximately 99% of all user sessions have less than 1000 actions. Interestingly, the tails of each line extends into the hundreds of thousands range, validating the need to perform deeper analysis of user sessions to better understand the heavy action outliers.

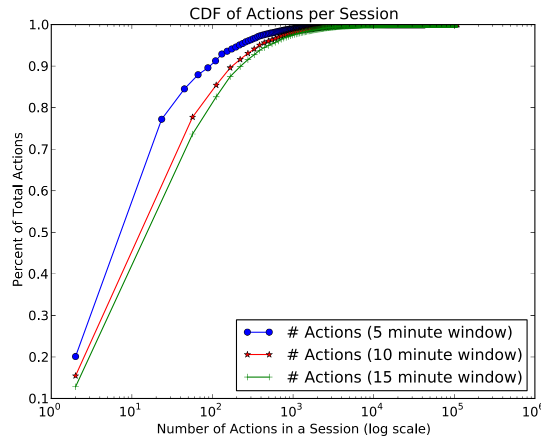

Figure 4: Empirical CDF of actions taken per session.

In summary, the analysis of user actions provided us with a number of useful results. First, we have observed that the average user session created with a fifteen minute window consists of about one hundred actions. Second, we've observed that the average user spreads their actions over a small set of directories. Third, we've determine that further analysis would be prudent to determine whether the spread of these actions is depth oriented or breadth oriented.

3.3 Files and Directories

In addition to statistics regarding user actions the number of observed unique directories and unique files were collected. The major findings in regards to files and directories are twofold. First, the average user session consists of actions spread among a small number of directories. Second, the average user tends to act on any given file only once per session.

Figure 6 shows the total number of unique directories observed during the trace and figure 5 shows the number of total unique files observed during the trace. These were collected in a rather naive method where each full file name (including path) was hashed. This means that if a file were moved to a new directory it would be counted as many times as it was observed in a different directory. In the future we would like to make more accurate counts by accounting for file moves and migrations. These two results are included in the interest of completeness and to bring context to other results that we will discuss shortly.

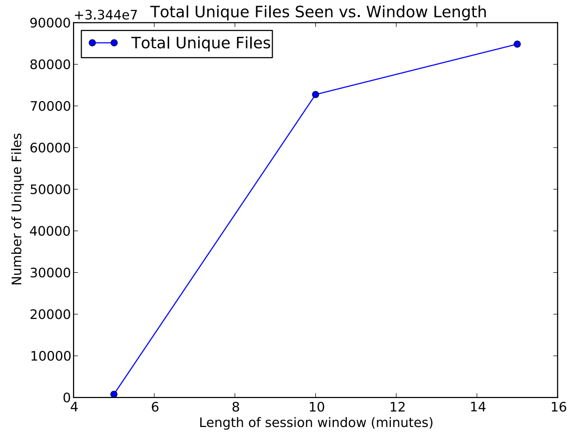

Figure 5: Total number of unique files seen vs. window length.

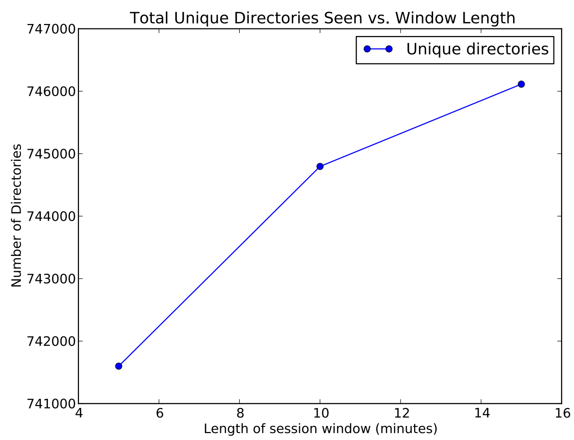

Figure 6: Total number of unique directories seen vs. window length.

In addition to counting the number of unique directories and files that were seen, we collected the number of unique directories touched per session. Figure 7 shows the mean, median, and standard deviation number of unique directories touched per session. We can see that the number of unique directories touched in an average session tends to be small compared to the total number of actions. This fact, when coupled with the number of actions per unique directory (illustrated in figure 8) shows that the actions taken during a session tend to be spread among a small number of directories. This result warrants more in depth study to determine if the spread of actions tends to be depth first, growing into sub directories of some common parent, or breadth first.

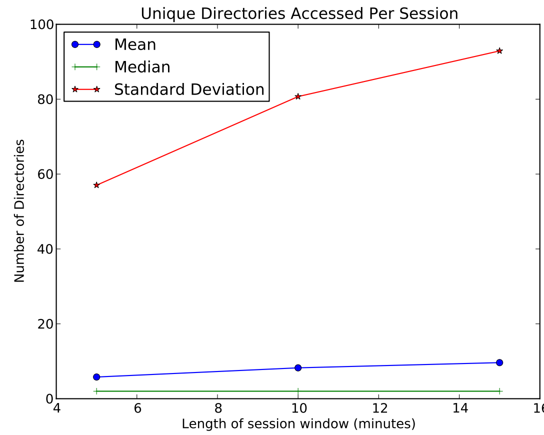

Figure 7: Mean, median, and standard deviation of unique directories seen per session vs. window length.

Figure 8 shows the mean, median, and standard deviation number of actions taken per directory. The mean and median lines show that there is a rather low number of actions taken per directory when compared to the number of actions taken per session. This validates our earlier claim that the actions tend to be spread over a small number of directories. Much like the number of actions per session there is a large variance here that warrants further study. Again, this variance is likely due to some automated activity rather than a marathon user session.

Figure 9 shows the mean number of unique files touched per session graphed next to the mean number of actions per session. These two lines closely follow the same trend suggesting that most actions taken tend to be taken on unique files rather than there being many actions taken on a single file or set of files.

3.4 Completeness of Results

It should be noted that after performing the analysis it was discovered that the data set used was not the complete three year trace as was expected. One of the log files containing the final six months of the log had been missing from the storage device where the trace was kept. This missing data fortunately was not lost, but unfortunately there was not enough time to re-run all of the code to and an analysis on the full dataset before the time of this writing. We do not expect that the current results are

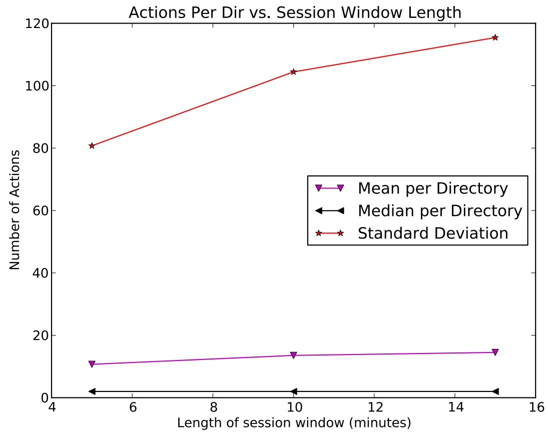

Figure 8: Mean, median, and standard deviation actions per directory per session.

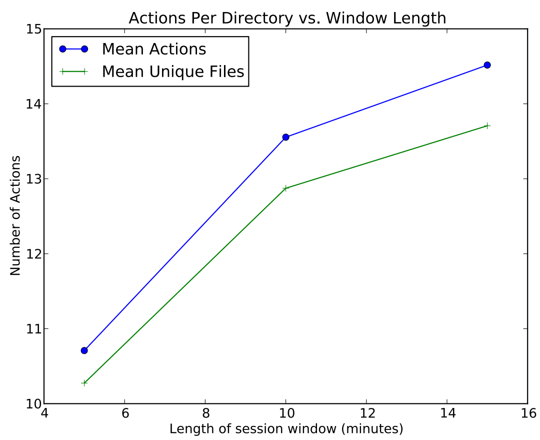

Figure 9: Actions per directory per session and unique files seen per directory per session vs. window length.

in any way invalid, and do not expect the additional data to significantly change the results. Currently, a re-run of the analysis is planned with the complete data set and a more exhaustive set of tests.

4 Future Work

While trace analysis is certainly a straight forward concept, in practice it is difficult. One of these difficulties lies in not really knowing what information is useful to collect a priori. While there are certainly some data that seem useful before analysis, the interesting questions don't seem to become apparent until after the first pass of analysis is done. For this reason, we have a number of additional tests that we would like to perform after having a cursory understanding of the NCAR system.

The first work that is planned is to re-run the entire initial analysis on the full data set. We assume that the addition of the extra data will not change much of the initial analysis, but we intend to perform it for completeness sake. We will also improve on our methods for counting items such as unique files and unique directories to account for file migrations and movement.

The second piece of future work intended is to analyze how the actions relate to the file system's namespace. Our initial work focused simply on looking at the number of actions taken in a session, and per directory. These numbers, while useful, provide us little insight into the way that the data a user is accessing is grouped. To further our understanding we will examine whether the actions taken tend to be focused in directories organized in a depth first manner or in a breadth first manner.

Lastly we will perform all of the aforementioned techniques in this paper on a filtered set of data that excludes excessively large sessions. These sessions will be set aside and examined separately to determine why they are so large, and why they are so vastly different than the average.

5 Related Work

Adams et. al. perform a similar style archival workload study [1]. While similar in nature, the Adams study covers a larger number of datasets as well as being a more generalized. Like our own study, the Adams study suggests that most user sessions consist of a relatively small number of accesses, while a smaller number of outliers perform vastly more. Our study despite sharing some similarities focuses only on a single dataset and looks at more specific user behaviors, and approaches them from the context of the file system's namespace.

A study performed by Shohbit Dayal looks at a number of high performance computer and traditional workstation file systems [2]. This study focuses primarily on file statistics such as filesize, file age, and file overheads. This study did not include a trace analysis and was simply a static file system snapshot analysis.

In Analysis of Long Term File Reference Patterns for Application to File Migration Algorithms [7] Smith examines static filesystem snapshots to better develop file migration algorithms from workstations to shared file server. This study, unlike our own, focuses on file system snapshots rather than traces, and is concerned with primary storage rather than archival or tertiary storage.

Miller and Katz [6] perform a similar study of the same

NCAR system, but almost two decades prior. Their study focused on a wider breadth of analysis and looked at items such as file and directory sizes, file reference patterns, and usage patterns over the course of a 24 hour period.

Leung et. al. [4] examine two large-scale file system workloads by measuring CIFS traffic for two enterprise-class file servers in the NetApp data center. These studies examined traces of a three month period, and are notable as the first major scale analysis of the CIFS protocol. This study however does not examine the usage of a large scale archive as ours does, and also covers a wider breadth of statistics such as looking at specific patterns for reads, or writes.

Stephen Strange [8] performs a study of file access patterns to assist in designing file migration algorithms for traditional file systems. Their study differs from ours in a number of ways. First, they examine traditional file server workloads rather than HPC archival workloads; second, they use their results in simulation to validate proposed file migration algorithms. Our study on the other hand examines a more recent dataset, is explicitly focused on user behavior within hthe archival context, and is not concerned with file migration.

Leung et. al. [5] develop a file metadata search system for large-scale storage systems. In this work they make use of real file system traces, but do not perform any explicit analysis of the traces. Instead the traces are used as workloads to validate their Spyglass system. In contrast, our study focuses on the analysis of traces and does not use them to validate any new algorithms or systems.

Ganger et. al. [3] design a system that includes two new techniques, embedded i-nodes and explicit grouping to improve the performance of traditional disk based systems. This work uses collected file system statistics to motivate and verify their design decisions, but ultimately is not focused on trace analysis. In contrast our study is focused on understanding user *behavior* whereas their analysis is concerned with properties of files. In addition, we do not use our analysis to validate any new methods or algorithms.

6 Conclusion

In conclusion, a study of a three year trace of the NCAR archival storage system was performed. Our analysis focused on user behaviors and how they related to the file system's namespace. We collected statistics on user actions per session, and how they were concentrated within the name space. Our major findings show that the average user takes 100 or fewer actions per session, but that a number of outliers perform orders of magnitude more actions. In addition we found that the average user's actions are spread among a small number of unique directories, and that the majority of actions in a session tend to be on unique files.

This work provides a basis and motivation for future work. In our future work we will filter outliers and focus on the average user session to better understand how user actions relate to the file system's namespace. In particular we will examine whether the user's actions tend to be depth oriented versus breadth oriented within the directory hierarchy.

7 Acknowledgements

We would like to acknowledge and thank Gene Harano and NCAR for their data and willingness to answer questions.

References

[1] I. Adams, E. L. Miller, and M. W. Storer. Analysis of workload behavior in scientific and historical long-term data repositories. Technical Report UCSC-SSRC-11-01, University of California, Santa Cruz, Mar. 2011.

[2] S. Dayal. Characterizing hec storage systems at rest. Technical Report CMU-PDL-08-109, Carnegie Mellon University, July 2008.

[3] G. Ganger and M. F. Kaashoek. Embedded inodes and explicit grouping: Exploiting disk bandwidth for small files. In *Proceedings of the 1997 USENIX Annual Technical Conference*, pages 1–17, 1997.

[4] 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*, June 2008.

[5] A. W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. L. Miller. Spyglass: fast, scalable metadata search for large-scale storage systems. In *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FAST)*, Feb. 2009.

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

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

[8] S. Strange. Analysis of long-term UNIX file access patterns for application to automatic file migration strategies. Technical Report CSD-92-700, University of California, Berkeley, Aug. 1992.