2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing

Understanding Data Characteristics and Access Patterns in a Cloud Storage System

Songbin Liu*†, Xiaomeng Huang*, Haohuan Fu*, Guangwen Yang*†
* Ministry of Education Key Laboratory for Earth System Modeling, and Center for Earth System Science
† Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
Email: liusb08@mails.tsinghua.edu.cn, {hxm,haohuan,ygw}@tsinghua.edu.cn

***Abstract*—Understanding the inherent system characteristics is crucial to the design and optimization of cloud storage system, and few studies have systematically investigated its data characteristics and access patterns. This paper presents an analysis of file system snapshot and five-month access trace of a campus cloud storage system that has been deployed on Tsinghua campus for three years. The system provides online storage and data sharing services for more than 19,000 students and 500 student groups. We report several data characteristics including file size and file type, as well as some access patterns, including read/write ratio, read-write dependency and daily traffic. We find that there are many differences between cloud storage system and traditional file systems: our cloud storage system has larger file sizes, lower read/write ratio, and smaller set of active files than those of a typical traditional file system. With a trace-driven simulation, we find that the cache efficiency can be improved by 5 times using the guidance from our observations.**

***Keywords*-Cloud Storage; File System; Data Characteristic, Access Pattern**

I. Introduction

Cloud storage is a new business paradigm that provides scalable, reliable and virtualized storage resources as a service. With a cloud storage service, users can exploit the power of a data center on demand without investing in large hardware, software and maintenance costs. Numerous of commercial cloud storage platforms, such as Amazon Simple Storage Service (S3), Google Drive, Microsoft Skydrive, Dropbox, Apple iCloud, Mozy, Cumulus, and so on, have been established and have had great commercial success around the world.

Based on the knowledge of the characteristics and access patterns of the data residing in the cloud storage system, cloud storage designers and administrators can make better design and optimization decisions, and provide cost-effective storage service. Many studies have been conducted to examine the characteristics and access patterns of various file systems. Several studies have been focused on the characteristics and usage of personal files systems [1]–[4]. In addition, numerous studies have examined the characteristics and access patterns of distributed file systems and parallel computing workloads [5]–[10]. The findings from these studies have motivated the design and optimization of many file systems and storage systems [11]–[14]. However, little work has been performed on the file characteristics and access patterns in cloud storage systems.

In this paper, we study how storage space is consumed and how files are accessed in a campus cloud storage system, called Corsair [15]. Corsair was deployed at Tsinghua University in 2008, and has become an important campus storage service for the students and teachers. By pooling many pieces of storage resource, the capacity of Corsair has been increased from 2 TB to 57 TB. At the point we began this work, there were already 19,892 registered users and 17.5 TB of data.

We present an analysis of two types of data in Corsair: a snapshot of the underlying file systems of the cloud storage system and an access trace to the files for a five-month period. Our work includes two contributions. First, we make the collected dataset, including both the snapshot of the file systems and the access trace, available to other researchers. To obtain this dataset, contact the authors.

Second, we analyze the dataset and make the following observations:

- File Size
  (1) More than 90% of the files in the system are smaller than 4 MB, and they occupy less than 10% of the consumed storage space. Over 90% of the accesses are to these small files.
  (2) The file sizes have a clear bimodal distribution.
- File Type
  (1) The ten most popular types of files account for 50% of the total number of the files, and another ten types of files consume 70% of the storage space.
  (2) Over 70% of the read files belong to 10 file types.
- Read/Write
  (1) The read/write ratio in this system is lower than that in traditional file systems.
  (2) Newly updated files tend to be read soon and files

978-0-7695-4996-5/13 $26.00 © 2013 IEEE
DOI 10.1109/CCGrid.2013.11

Table I
ACCOUNT INFORMATION OF CORSAIR

Type	Registered	Assigned	Capacity
Personal	19,892	3,791	2 TB
Group	513	513	18 TB

tend to be read-only once they have been read.

- Active files
  Most of the files are rarely accessed, while a small proportion of the active files consume a large share of the bandwidth.

Using these findings, we can optimize the underlying file systems of Corsair and improve the I/O performance to provide a better user experience with minimal cost. In section VI, we give an example of how to use the observations presented above to improve the cost-effectiveness of the cloud storage system's cache.

The remainder of this paper is organized as follows. The next section gives an overview of Corsair. Section III describes the methodology of collecting the snapshot data and the access trace, and discusses some of our general presentation and analysis techniques. Section IV presents our findings about the data characteristics, and Section V discusses the access patterns to the files. Section VI gives an example of how to use the observations to optimize the cache efficiency. Section VII reviews previous work on file systems about both snapshot and trace driven analysis, and Section VIII concludes the paper.

II. SYSTEM OVERVIEW

We developed a campus cloud storage system called Corsair, which was deployed on campus on December 1st, 2008. Corsair provides scalable, high performance and reliable storage service for both personal and group accounts on the campus. There are two types of accounts in Corsair: personal account and group account. Each registered user can apply for one personal account, which includes 2 GB of private storage by default. The group account is for student clubs, and members of the club can share files in their group account. A registered user can apply to create one or more group accounts, and the default quota for each group account is 100 GB. Registered users can apply to join multiple groups and share files in the groups. Each account can obtain more storage space if needed by submitting online applications.

As shown in Table I, there were about 19,892 registered users and more than 500 registered groups by September 1st, 2010. The number of user accounts is over 90% of the total number of students at Tsinghua University, and almost all of the campus student clubs have a group account. However, due to limited storage resources, we limited the storage assignments for registered users. Only 3,791 of the registered users obtained the 2 GB of private storage.

Table II
OVERVIEW OF THE DATASET

	Personal	Group
Accounts#	452	456
Directories#	33,621	270,726
Files#	309,189	3,201,915
Storage Consumption	415 GB	17,421 GB
Trace Length	five months(147 days)	
I/O Requests#	209,017	2,785,038
Accessed Files#	122,627	1,217,051
Touched File Size#	289 GB	7,942 GB
Bytes Transferred (GB)	524 GB	15,365 GB

III. DATA COLLECTIONS

To study the data characteristics of the files in the cloud storage, we took a snapshot of the files on September 1st, 2010. The snapshot contains most of the metadata of the files, including file name, size, timestamps and other attributes associated with each file or directory. Because the two types of accounts are used for different purposes, we collected the file metadata from the personal accounts and the group accounts separately. In many of our figures, we will use legend *personal* for personal accounts and *group* for group accounts.

To study the access patterns of the users, we record the access logs on the Corsair server side. Each time a user writes or reads a file, the server will write a record containing the account information, file path, file size, access time and operation type. In this paper we study the access trace of five months (147 days); from March to July 2010. This period was chosen for two reasons. The first is that in 2010, Corsair was popular on the campus and the workload was stable. The second reason is that during the summer holidays (in August) and the winter vacations (a period from January to February), few students are on the campus and there is little access to Corsair. The workload of Corsair is similar during each semester, so we studied a period of time during the semester.

We use the access trace to identify inactive accounts and remove their data and access log from our dataset. Our work is mainly based on this filtered dataset. Table II gives overviews of both the snapshot and the five-month access trace of the dataset. The dataset includes 452 active personal accounts and 456 active group accounts, approximately 33,621 directories for personal accounts, and 270,726 directories for group accounts. The 309,189 personal files and 3,201,915 group files consume approximately 17.5 TB of storage in total. About one-third of the files are accessed, and almost 16 TB of data are transferred during these five months.

Note that because all of the data in this study come from a campus storage system, there are the following limitations with the dataset:

- The users are mainly college students, whose usage

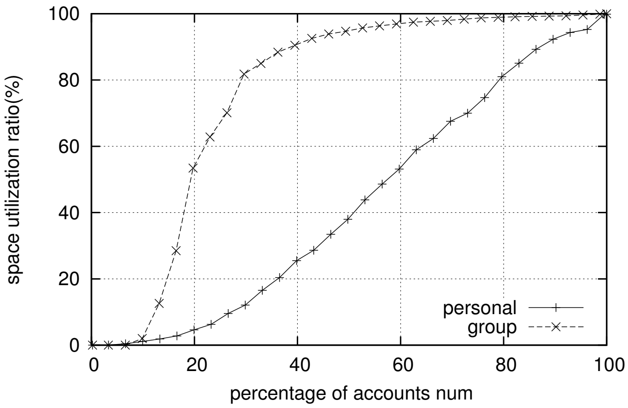

Figure 1. Space utilization of the accounts

patterns may be different from that of other people.
- The storage size limit is 2 GB for each personal account and 100 GB for each group. This default setting may affect the usage for these two types of accounts.

IV. Data Characteristics

In this section, we study the file size distribution and the popular file types in the system. Three interesting observations are found from the study: (1) The file sizes have a bimodal distribution. (2) Most of the files are smaller than 4 MB, and these files consume only 10% of the storage space. (3) Several popular file types account for half of the files and consume approximately 70% of the storage space.

A. Storage utilization

We investigate the storage utilization of each account and find that a significant portion of the accounts utilize a small amount of their storage quota. Figure 1 shows the distribution of account storage utilization. The space utilization for personal accounts is very random, and the mean utilization is approximately 50%. This indicates that there is a large profit margin for personal accounts. On the other hand, the utilization rate of group accounts is much higher. Approximately 70% of the group accounts used more than 80% of their assigned storage space.

B. File size

The file size distribution is important when designing and optimizing the file system, many studies have examined the file size distribution in various file systems [16], [17]. In this section, we will investigate the file size distribution in terms of file count, storage space consumption and the I/O traffic based on the snapshot and the access trace. We find that the file sizes have a clear bimodal distribution. Another finding is that most of the files are small files, and most of I/O requests are for those small files. However, the total volume of these small files is relatively small.

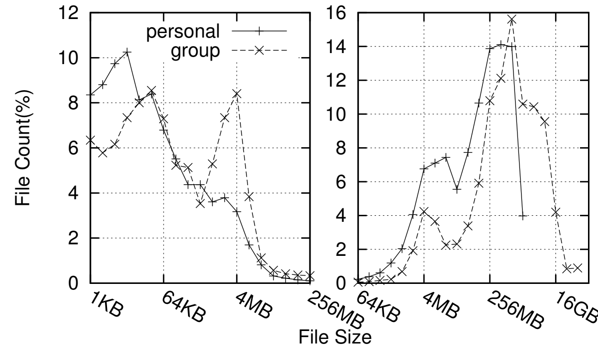

Figure 2. Bimodal file size distributions

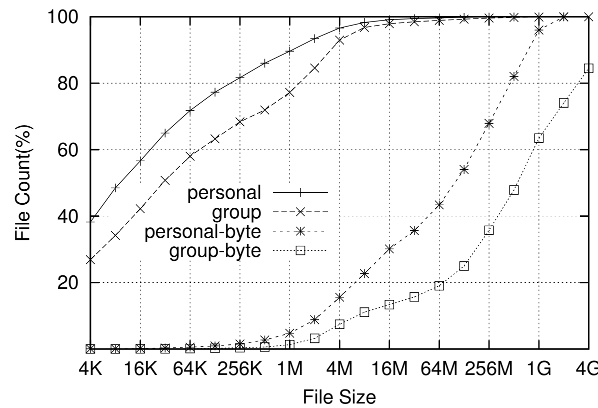

Figure 3. File size vs. file count and volume

Figure 2 shows the probability distributions of the file size both in terms of file count and the storage consumption, and Figure 3 is the cumulative distributions of the same dataset. Figure 2 shows that the file sizes have a clear bimodal distribution both in terms of file count and the used storage space. The bimodal distribution trend was first found in a previous study [2], but the file size distributions in this study display the bimodal shape more clearly. Two peaks are present in both distributions and especially for the files in the group data. Previous studies did not find a bimodal file size distribution in terms of file count and the only peak was approximately 1 KB [2]. In the left graph of Figure 2, however, the second peak for group files is approximately 4 MB. Another main difference from previous findings is that the second peak is much higher, indicating that there are more large files in our cloud storage system.

Figure 3 shows that more than 90% of the files are smaller than 4 MB and that they consume approximately 10% of the storage space. We also find that most of the I/O requests are for small files, and the volume of these small files is relatively small. Figure 4 shows the file size distribution in terms of both the I/O request count and the volume of these files according to the five-month period trace. The data

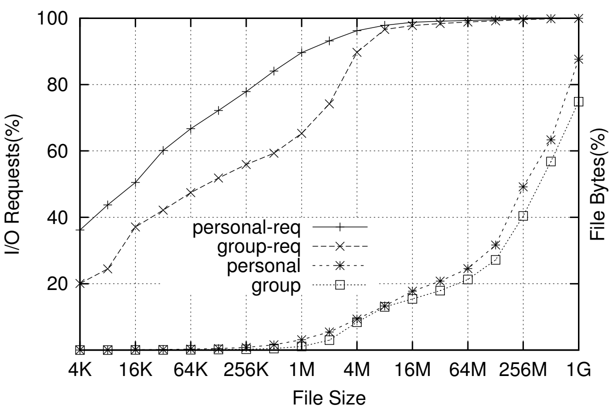

Figure 4. File size Vs. I/O requests and volume

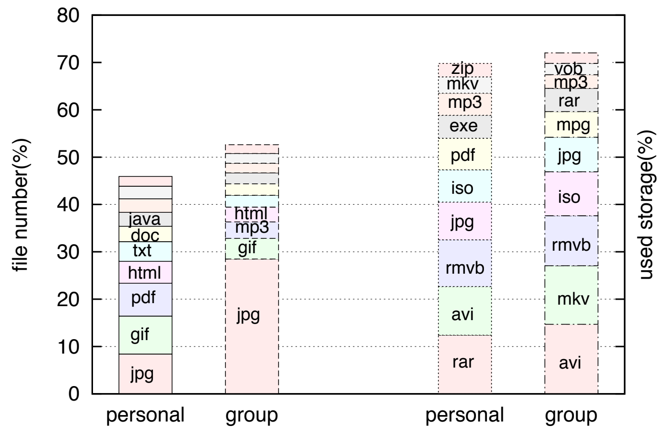

Figure 5. Fractions of files with popular extensions

shows that approximately 90% of the requests are for files smaller than 4 MB, and that almost 99% are for files smaller than 16 MB. On the other hand, the volume of these small files is only a small proportion of the total volume (10% for files smaller than 4 MB, and less than 20% for files smaller than 16 MB). This result is slightly different from the findings in a campus Peer-to-Peer file-sharing workload [18]; they found that approximately 91% of the requests were for files smaller than 10 MB, and that these requests consumed around 15% of the bandwidth. This is because most of the requests in their study were for audio clips, while most of the requests in this study were for photos and documents, which is discussed in the next section.

C. *Popular file types*

In this section, we study the file types that are popular in term of file count, storage consumption, and access count based on the snapshot and the five-month access trace.

Figure 5 shows the ten most popular file extensions according to the snapshot. The left two columns show the fractions in terms of file count, while the right two are in terms of consumed storage space. Figure 5 shows that the ten most popular file types account for approximately half of the total number of files. The file extensions *jpg, gif, html, pdf and doc*, are the most common file types in both types of accounts. Almost one third of the files in the group accounts are photos and other images files (*jpg, gif*). In fact, one of the most important usages of Corsair is for group photo sharing. As to the storage usage, another ten file types consume 70% of the storage space; most of them are video files (*avi, rmvb, mkv*), and some are archives files (*rar, zip, iso*). It should be noted that *jpg* files also make up a large share of the files. In a previous study [2], the ten most popular type files occupied less than 45% of the total storage. The most popular file types in this study are also different from those in the previous findings [2]: file extensions such as *dll, exe, and pdb* are most popular file types in the Windows file systems in their study.

Table III
FILE TYPES OF MOST READ FILES

Personal		Group	
File Type	Percent	File Type	Percent
doc	20.06	jpg	55.94
pdf	19.51	png	6.13
jpg	7.59	bin	5.82
docx	4.34	doc	4.72
w3g	4.16	html	4.48
opa	3.91	gif	3.67
html	3.79	txt	2.44
w3x	3.50	lua	2.29
txt	2.69	pdf	0.98
xls	2.49	null	0.73
Total	72.04	Total	87.20

Table III shows the file types of the ten most read files based on the five-month access trace. The file type list for personal accounts is different from that for group accounts. Documents (*doc, pdf, docx*) are at the top of the list for personal accounts. Although the group accounts included some of these document types, the largest share are for photos and images (*jpg, png*). This difference occurs because most personal accounts are used for personal workspace and for backing up documents, while the group accounts are often used for photo sharing. Furthermore, the files of these types are always small. The average size of each these file types is less than 4 MB (the largest are *pdf* files, which have an average size of 3.3 MB) according to our statistics.

Because files with the same extension tend to have similar properties and access patterns, benefits can be gained by developing storage management policies for particular extensions. For example, about half of the files and 70% of the bytes are files with a few popular extensions; developing such special-case treatment for only a few particular file extensions can optimize performance for a large fraction of the cloud storage system.

Table IV
READ/WRITE RATIO

	Personal	Group
Operations(k)		
Read	83	1,924
Write	126	861
R:W ratio	0.65	2.2
Bytes(GB)		
Read	254	10,242
Written	270	5,123
R:W ratio	0.94	2.0

V. ACCESS PATTERNS

The data characteristics described above are not the only factors that should be considered when designing and optimizing a cloud storage system. It is also important to gain knowledge of the access patterns to the cloud storage system. After studying the five-month access trace of Corsair, we identified the following access patterns: (1) The read/write ratio is very low (2:1); (2) Newly written or updated files tend to be read soon, but few files will be updated again once they have been read; (3) Only a small proportion (25%) of the files are accessed.

A. Read and write

According to the access trace, the read/write ratio is very low. Table IV presents the overall read/write operations and their associated I/O traffic for both personal and group accounts. Table IV shows that the ratio of read to write operations is 0.65 for personal accounts and 2.2 for group accounts. The ratios of read bytes to written bytes are 0.94 and 2.0 for personal and group accounts, respectively. The read/write ratios are very low when compared to other studies [5]–[7], [19]. In 2008, Leung et al. found a sharp decrease in the read/write ratios from 4:1 or higher to 2:1 from a CIFS workload [7]. We believe that the key reason for the decrease in the read/write ratios is that the Corsair GUI client absorbs most of the read requests. Another reason is that many users only use the Corsair system as a backup platform; they put their files into Corsair, and only read a small proportion of the files that are in the systems. Based on this low read/write ratio, we conclude that our cloud storage system is less read-centric, and that it is valuable to have a client end write-buffer, which can dramatically reduce the write-requests to the server.

We also find a dependency between the read and write to a file: (1) newly written files tend to read soon after being written; and (2) files are rarely updated (written) after they are read from the cloud. We borrow the data dependency definitions to help us to investigate the relationship between read and write. The read-write dependency can classify the access to a file into four categories: *Read After Write (RAW)*, *Write After Write (WAW)*, *Write After Read (WAR)* and *Read After Read (RAR)*. A *RAW* is said to exist between two operations if the first operation writes a file that is later read by the other operation within a certain time window. *WAW*, *WAR* and *RAR* are similarly defined.

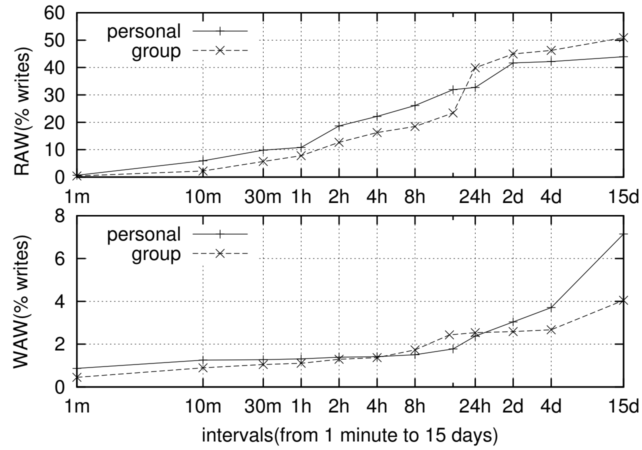

Figure 6. RAW and WAW

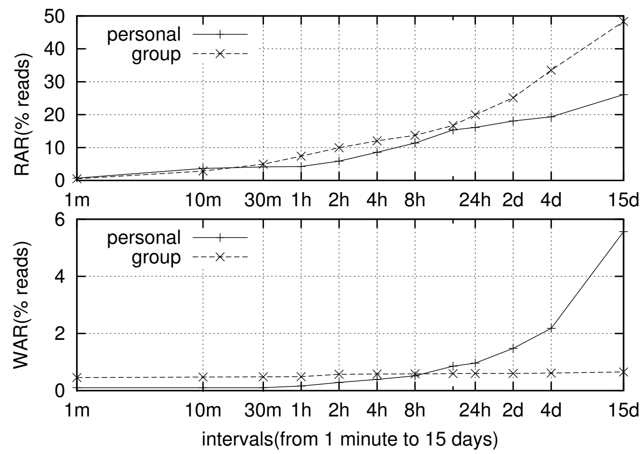

Figure 7. WAR and RAR

Figure 6 plots the percentage of writes for which there is a read (or write) that constitutes a *RAW* (or *WAW*) within τ units of time. We refer to τ as the window size. The *RAW* distribution shows that more than 40% of the newly written files are read within two days. This means when a file is uploaded to the system, users will generally access it within a few days. This reflects one of the most common usage scenarios: after a group member updates a file, the other group members will access the updated file in a few days. For *WAW*, much fewer files (less than 8% within a 30-day window) are further updated after they are put into the storage system. In a NFS file server, updated blocks are more likely to be updated again than to be read [20], which is in contrast to our findings. This difference is caused by the different ways the two storage systems are used. A NFS file system is mounted into a local file system, and the NFS server has to handle almost all of the updates. In contrast,

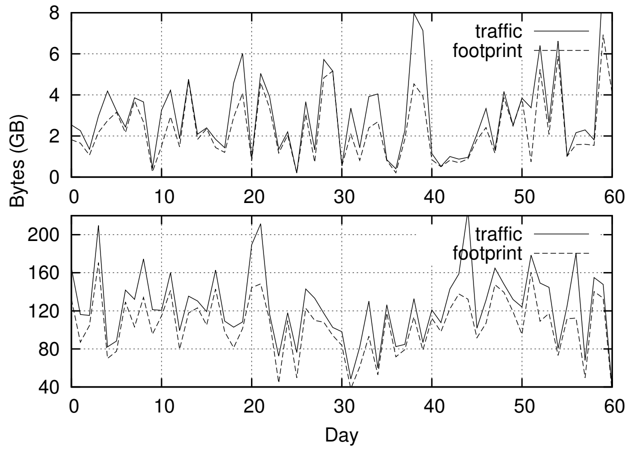

Figure 8. Daily I/O traffic and footprint

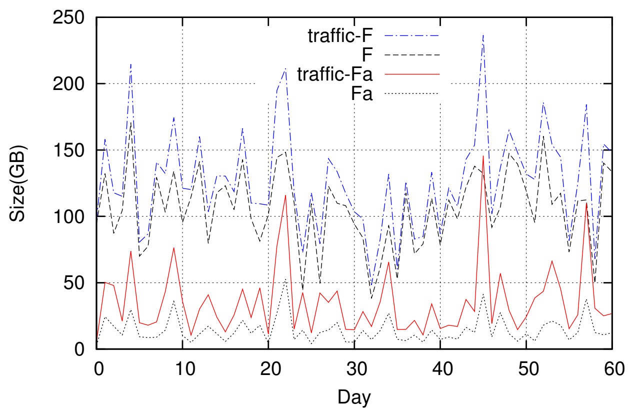

Figure 9. Footprint and active footprint

the Corsair GUI client absorbs most of the updates, so the Corsair server is not aware of most of them. Figure 7 plots similar distributions for *RAR* and *WAR*. There are even fewer *WAR*s than *WAW*s; this indicates that once a file is read, it is stable (immutable) and few updates are needed to the file. In group accounts, the file is usually not updated further after it has been read. With window sizes of less than 24 hours, both account types have similar distributions. However, with a larger window size, the group accounts tend to be read more, and personal accounts tend to have more updates.

B. Small data footprint

To better understand the daily I/O traffic, we borrow the definition of *Footprint* to denote the unique files and data accessed on each day. The footprint F(t, τ) is defined as the set of files accessed within a time window τ. This definition is similar to that of Windsor [20]. The main difference is that Windsor's definition is block based, while ours is file based. We redefine the footprint as:

$$F(t, \tau) = \{f : Count(f, t, t + \tau) \geq 1\} \qquad (1)$$

where Count(f, t , t+τ) denotes the number of times file f is accessed between t and t + τ.

Figure 8 shows the daily footprint size and the associated I/O traffic for the first 60 days of both accounts. The top graph is for personal accounts and the bottom one is for group accounts. Note that the *footprint F* of day x is defined as F(t=midnight of day x, τ=1 day). The daily footprint size is around 2 GB for personal accounts, and 120 GB for group accounts. When compared to the total storage (450 GB of personal, 17 TB of group), these daily footprint sizes are only a small proportion of total files. On average, the daily footprint for group accounts is merely 0.68% of the storage used, and the daily footprint for personal accounts is just 0.96%. The maximum footprints are several times larger than the average, but the maximum is only 1.6% of the total storage for the group accounts. These footprints are much lower than the findings in a previous study [20], which ranged from 4% to 7%.

In Figure 8 the curve of the footprint is smoother than that of the traffic and there is a gap between the two curves. The gap is caused by the files that are accessed more than once. We define *Active Footprint* to denote the set of files that are accessed more than once: $F_{activie}(t, \tau) = \{ f\!: Count(f,t,t{+}\tau) \geq 2 \}$. Figure 9 shows the daily footprint and active footprint for the group accounts. The size of *Active Footprint* is only approximately 1/6 of the overall footprint. In fact, the active footprint is very small; according to the five-month access trace, fewer than 12% of the files are active files, but these active files consume almost 70% of the read bandwidth. If these active files can be identified, it will be cost-effective to optimize the access to them, which would provide a better overall user experience.

The gap between the footprint and the traffic is larger for group accounts than for personal accounts. This indicates that there are more accesses to the same files in the group accounts, and conforms to the different usages of the personal accounts and group accounts: groups use the system for file sharing, while personal accounts are used for backup.

VI. IMPLICATIONS

The observations in previous sections can provide some implications for the design and optimization of the cloud storage system. In this section, we show how to apply these implications to improve the cache efficiency.

Caching is the key technology for bridging the performance gap across memory hierarchies via temporal or spatial localities, and is especially important for storage systems. In section V-B, we found that a small amount files are accessed more than once, and that they account for as much as 70% of the read requests. So if these files can be cached, the overall read access latency will be reduced. One challenge is how to identify them to make better use of the cache.

Based on the observations in previous sections, we can determine which files should be cached and which files

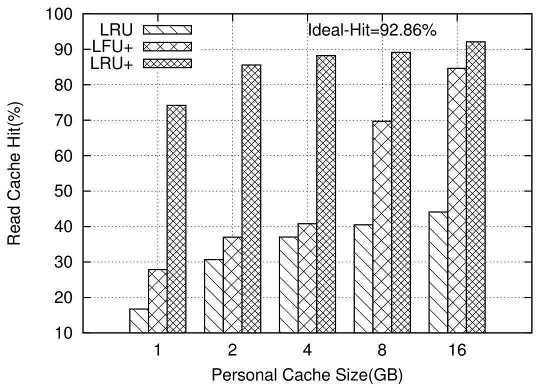

Figure 10. Cache hit improvement (personal workloads)

should be evicted when cache replacement is needed. In Section V-A, we know that there are many *RAR*s and *RAW*s and few *WAR*s or *WAW*s, so it is reasonable to cache all of the recently read and written files. Based on Section IV-B, almost 99% of the accessed files are smaller than 16 MB. Thus small files should have more of a chance to stay in cache. According to Section IV-C, most of the accessed files belong to a small set of file types, and hence these types of files should have more weight over other files. Using these guidelines, we try to optimize the cache efficiency by redesigning the cache eviction policy that takes file type and file size into account. We use the heuristic method to add these information to the original *Least Recently Used (LRU)* and *Least Frequently Used (LFU)* cache replacement policies and obtain the *LRU+* and *LFU+* algorithms. A trace-driven simulator is designed to evaluate the efficiency of the *LRU+* and *LFU+* algorithms over the five-month access trace.

Figure 10 shows the read cache hit rates of the different cache algorithms for the personal accounts. From this figure, we can see that with the guidance of our observations, the *LRU+* and *LFU+* algorithms are better than the original LRU algorithm. The *LRU+* is significantly better than the other two algorithms: with a 1 GB cache, its cache hit is approximately 5 times better than the original *LRU*, and with a 2 GB cache, its cache hit can achieve approximately 86%, while the ideal cache hit is 92.86%. It should be noted that the file type information does not have a large effect in our experiments. One reason may be that we do not know how to correctly use file type information in the algorithms. Another reason may be that most of the files with the popular file types are smaller than 4 MB, so the file type information cannot make much of a contribution because the file size information is already used.

VII. Related works

Prior to our work, there are few studies of data characters and usage patterns in the cloud storage. One recent work about Dropbox focuses on the analysis of its Internet traffic [21]. However, the usage characteristics of many types of file systems have been studied in detail; these studies have significantly impacted the design of file systems [11], [12] and storage systems [13], [14].

There are two methods to determine file system usage characteristics. One method is static data based analysis, by which file system metadata are collected at one or several instants in time and these snapshots are examined to study the attributes commonly stored in the metadata. These studies focus on the properties of the file system, such as the file size distributions [3], popular file types and directory depths [1], [2], [8]. The other method is analyzing the dynamic file-system access trace, which contains more information about the file system's real behavior. Baker et al. [5] conducted a set of analysis on four sets of two-day traces of four Sprite file servers. Roselli et al. [19] compared four file system workloads and showed that the block lifetime had increased; he also explored the effect on caching strategies. Ellard et al. [6] analyzed traces from both research and email workloads of NFS file servers, and found that path-names can be used to predict file properties. Leung et al. presented the CIFS file servers traffic during a three-month period [7]. There are many studies in analyzing the I/O characteristics of high performance computing systems [9], [10].

The findings from these studies have motivated the design and optimization of many file systems and storage systems [11]–[14]. The design of the cache system of Sprite file system [11] and the log-structured file system [12] was inspired by the analysis of a Unix file system trace [22]. Based on the observation that most of the accessed files are small files, Mullender et al. proposed an efficient disk organization that stores the first part of the file in the index (inode block). Zhang et al. presented a design based on this idea to optimize the access performance to small files [23]. Mesnier et al. assigned different storage policy to files according to file size and type [14]. Vairavanathan et al. suggested to study the data access patterns of workflows and implemented a workflow aware storage system [13].

Compared with the studies mentioned above, we study both the data characteristics and access patterns in a cloud storage system instead of traditional file systems. There are more users, files and storage resources in a single cloud storage system. Thus the data characteristics and access patterns of cloud storage system are different from those of a traditional file system.

VIII. Conclusions

In this paper, we studied the data characteristics and access patterns of a cloud storage system. Based on a snapshot of the underlying file systems and an access trace for five-month period, we found that in a cloud storage system the file sizes have a bimodal distribution and that most access requests are for small files. We also found that a small set of file types is very popular in terms of file count, storage consumption and access rate. We found the read/write ratio is very low, and newly written files are tend to be read soon after being written. Although the daily traffic is large, only a small portion of the files are accessed. With a trace-driven simulation, we found that the cache efficiency can be improved by five times using the findings in this paper. We believe that cloud storage designers can make cost-effective improvements in cloud storage systems using the guidance from our results.

IX. ACKNOWLEDGEMENTS

This research was sponsored by the National High Technology Development Program of China (2010AA012401, 2011AA01A203).

REFERENCES

[1] J. R. Douceur and W. J. Bolosky, "A large-scale study of file-system contents," in *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 1999, pp. 59–70.

[2] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch, "A five-year study of file-system metadata," *ACM Transaction on Storage*, vol. 3, no. 3, p. 9, 2007.

[3] A. S. Tanenbaum, J. N. Herder, and H. Bos, "File size distribution on unix systems: then and now," *ACM SIGOPS Operating Systems Review*, vol. 40, pp. 100–104, 2006.

[4] T. Harter, C. Dragga, M. Vaughn, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "A file is not a file: understanding the i/o behavior of apple desktop applications," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*. New York, NY, USA: ACM, 2011, pp. 71–83.

[5] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. Shirriff, and J. K. Ousterhout, "Measurements of a distributed file system," in *Proceedings of the thirteenth ACM Symposium on Operating Systems Principles*. New York, NY, USA: ACM, 1991, pp. 198–212.

[6] D. Ellard, J. Ledlie, P. Malkani, and M. Seltzer, "Passive nfs tracing of email and research workloads," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, ser. FAST '03, Mar 2003.

[7] A. Leung, S. Pasupathy, G. Goodson, and E. Miller, "Measurement and analysis of large-scale network file system workloads," in *Proceedings of the 2008 USENIX Annual Technical Conference*, 2008, pp. 213–226.

[8] S. Dayal, "Characterizing hec storage system at rest," Carnegie Mellon University, Tech. Rep. CMU-PDL-08-109, Jul. 2008.

[9] A. Iamnitchi, S. Doraimani, and G. Garzoglio, "Filecules in high-energy physics: Characteristics and impact on resource management," in *15th IEEE International Symposium on High Performance Distributed Computing*, 2006, pp. 69–79.

[10] P. Carns, R. Latham, R. Ross, K. Iskra, S. Lang, and K. Riley, "24/7 characterization of petascale i/o workloads," in *IEEE International Conference on Cluster Computing and Workshops*. IEEE, 2009, pp. 1–10.

[11] M. N. Nelson, B. B. Welch, and J. K. Ousterhout, "Caching in the sprite network file system," *ACM Transactions on Computer Systems*, vol. 6, pp. 134–154, February 1988.

[12] M. Rosenblum and J. K. Ousterhout, "The design and implementation of a log-structured file system," in *Proceedings of the 13th ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 1991, pp. 1–15.

[13] E. Vairavanathan and et al., "A workflow-aware storage system: An opportunity study," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, ser. CCGRID '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 326–334.

[14] M. Mesnier, F. Chen, T. Luo, and J. B. Akers, "Differentiated storage services," in *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*. New York, NY, USA: ACM, 2011, pp. 57–70.

[15] P. Xu and et al., "Enabling cloud storage to support traditional applications," in *Proceddings of the 5th Annual ChinaGrid Conference*. IEEE, 2010, pp. 167–172.

[16] G. Irlam. (1994) Unix file size survey. [Online]. Available: http://www.base.com/gordoni/ufs93.html

[17] K. M. Evans and G. H. Kuenning, "A study of irregularities in file-size distributions," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, ser. SPECTS, 2002.

[18] K. P. Gummadi and et al., "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*. New York, NY, USA: ACM, 2003, pp. 314–329.

[19] D. Roselli, J. Lorch, and T. Anderson, "A comparison of file system workloads," in *Proceedings of the 2000 USENIX Annual Technical Conference*, San Diego, CA. Berkeley, CA, USA: USENIX Association, 2000, p. 58.

[20] W. Hsu and A. Smith, "Characteristics of i/o traffic in personal computer and server workloads," *IBM Systems Journal*, vol. 42, no. 2, pp. 347–372, 2003.

[21] I. Drago and et. al, "Inside dropbox: Understanding personal cloud storage services," in *Proceedings of the Internet Measurement Conference*, Boston, MA, USA, 2012.

[22] J. K. Ousterhout, H. Da Costa, D. Harrison, J. A. Kunze, M. Kupfer, and J. G. Thompson, "A trace-driven analysis of the unix 4.2 bsd file system," in *Proceedings of the tenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 1985, pp. 15–24.

[23] Z. Zhang and K. Ghose, "hfs: a hybrid file system prototype for improving small file and metadata performance," in *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, ser. EuroSys '07. New York, NY, USA: ACM, 2007, pp. 175–187.