International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | June -2017 www.irjet.net p-ISSN: 2395-0072

An Enhanced Cloud Backed Frugal File System

Devesh Kumar Das[1], MD. Shabaaz Hayat[2], Jerosh Joshi[3] , Mrs. Ashwini C[4]

[1]*Student Computer Science & Engineering @ Global Academy of Technology, Bengaluru*
[4]*Assistant Professor, Computer Science & Engineering @ Global Academy of Technology, Bengaluru*

Abstract - *with simple access interfaces and flexible billing models, cloud storage has become an attractive solution to simplify the storage management for both enterprises and individual users. However, traditional file systems with extensive optimizations for local disk-based storage backend cannot fully exploit the inherent features of the cloud to obtain desirable performance. In this paper, we present the design, implementation, and evaluation of a cloud based file system that strikes a balance between performance and monetary cost. Unlike previous studies that treat cloud storage as just a normal backend of existing networked file systems, this file system is designed to address several key issues in optimizing cloud-based file systems such as the data layout, block management, and billing model. With carefully designed data structures and algorithms, such as identifying semantically correlated data blocks, hash map based caching policy with self-adaptive thrashing prevention, effective data layout, and optimal garbage collection, this file system achieves good performance and cost savings under various workloads as demonstrated by extensive evaluations.*

Key Words*:* Cloud Computing, Frugal File, Cloud Backed

1. INTRODUCTION

A large body of work has advanced the state of art of cloud storage research, including but not limited to the topics mentioned above. In particular, a recent work proposed a cloud-based storage solution called Bluesky for the enterprise, which acts as a proxy to provide the illusion of a traditional file server and transfer the requests to the cloud via a simple HTTP-based interface. By intelligently organizing storage objects in a local cache, Bluesky serves write requests in batches using a log-structured data store and merges read requests using the range request feature in the HTTP protocol. In this way, many accesses to the remote storage can be absorbed by the local cache, avoiding undue resource consumption accordingly. As for cost optimization, FCFS proposed a frugal storage model optimized for scenarios concerning multiple cloud storage services. Similar to local hierarchical storage systems, FCFS integrates cloud services with very different price structures. We present the design and implementation of a cost-effective file system based on the PaaS cloud storage. In contrast to current research activities that view cloud service as a backend for network file systems and adopt classical caching strategies and data block management mechanisms, we argue that many inherent characteristics of cloud storage, especially the billing model, should be considered in order to improve resource utilization and minimize monetary cost. Specifically, we exploit the semantic correlation of data blocks, and propose a system design that takes advantage of a smart local cache with effective eviction policy and an efficient data layout approach. Evaluations performed on our proof-of-concept implementation demonstrate prominent savings in cost and gains in performance as compared to the state of the art. The main challenge in designing this file system is how to manage data blocks effectively. Comparing to the potentially unlimited storage capacity in the cloud, the relatively small cache on the client side may result in severe performance degradation due to the low cache hit rate and the high latency of the WAN.

Figure 1: Cloud Storage providers

2. RELATED WORK AND EXISTING SYSTEM

Cloud storage driven by the availability of commodity services from Amazon S3 and other providers has attracted wide interests in both academia and industry. As a new option of storage backend, it greatly eases storage management but brings new challenges as well. The work in this paper reflects our endeavours to address the interrelated issues such as caching strategy, object Organization, and monetary cost optimization in designing a cloud based file system.

Cloud based storage. For the service provider, Depot [1] considers safety and liveness guarantees based on distributed sets of untrusted servers. For safety and security reasons, SafeStore [2], DepSky [3] and SCFS [4] operate on

diverse storage service providers to prevent cloud vendor lock-in. Recently, Bluesky [5] presents a cloud based storage proxy for enterprise environments. With the log-structured [6] data layout, the proxy can absorb massive remote write requests to achieve high-throughput. However, the design proposed by Blueksy does not consider the billing model that is a major distinct feature compared to classical storage model. Exemplar open source projects like S3FS [7], S3QL [8], and SDFS [9] provide the file system interface for the cloud backend like Amazon S3. Similar to Bluesky, commercial systems such as Cirtas [10], Nasuni [11], and TwinStrata [12] act as cloud storage gateways for enterprises rather than personal users.

In [13], the authors reported metadata characteristics over 60 K PC file systems in a large corporation during five-year period. The changes in file size, file type, file age, storage capacity and consumption etc. motivated the database design for metadata management in our work. Also, the study [14] analyzed the whole-file versus block level elimination of redundancy, which reflects the necessity of block-level deduplication. A recent work [15] showed the behaviours of a popular personal cloud storage service Dropbox [16]. One of the main findings is that the performance of Dropbox is driven by the distance between clients and storage data centres. That corresponds to our design principle on bottleneck shifting to the network. Storage management. To close the widening semantic gap between computer system and storage system, [17] proposed an classification architecture for disk I/O. The same class of objects (e.g., large files, metadata, and small files) are combined according to the caching policy, which boosts end to- end performance. Similarly, BORG [18] performs automatic block reorganization based on observed I/O workloads, including non-uniform access frequency distribution, temporal locality, and partial determinism in non-sequential accesses. With reducing the monetary cost as one of the major concerns, we exploit the semantic information among data blocks in all aspects when designing Coral which is the existing file system.

Hystor [19] describes a high-performance hybrid storage system with solid state drive (SSD) based on active monitoring of IO access patterns at runtime. HRO [20] treats SSD as a by passable cache to hard disks. By estimating the performance benefits based on history access patterns, the system can maximize the utilization of SSD.

Cache policy. Beyond typical LRU mechanism, SEER [21] improves cache performance using the sequence of file accesses for measuring the relationship among files. The follow- up study [22] discussed various semantic distances of files and designed an agglomerative algorithm in disconnected hoarding scenario. Also, [23] proposed a two-fold mining method of block correlation mainly based on frequent sequences. In contrast, Coral describes additional eviction basis (besides hotness of data) with temporal-based metric of the file and block, which is simple and easy to measure in file system at runtime with cloud backend rather than offline mining. Few research has specifically studied the issue of monetary cost optimization for the cloud. Chen [23] evaluate cloud storage costs from economic perspective, which is at more abstract level and may not be suitable for complex usage scenarios. In [24], the authors present a system called FCFS with the main focus on the monetary cost reduction for cloud based file systems. However, this work focuses on the optimization for scenarios integrating multiple cloud storage services with distinct cost and performance characteristics. In fact, Coral is complementary to FCFS since optimizing the cost for a single cloud storage service can also benefit systems like FCFS.

Cloud Storage has many benefits and great advantages. Some of the advantages are to provide better accessibility; to be able to easily access their data from anywhere by using Internet. As cloud computing is maturing gradually, cloud computing applications are more and more, as the important aspect, cloud storage received wide attention of industry experts. Based on the deep research of the cloud storage technology, in the existing system we present the design and implementation of a cost-effective file system based on the PaaS cloud storage. In contrast to current research activities that view cloud service as a backend for network file systems and adopt classical caching strategies and data block management mechanisms, we argue that many inherent characteristics of cloud storage, especially the billing model, should be considered in order to improve resource utilization and minimize monetary cost. Specifically, we exploit the semantic correlation of data blocks, and propose a system design that takes advantage of a smart local cache with effective eviction policy and an efficient data layout approach.

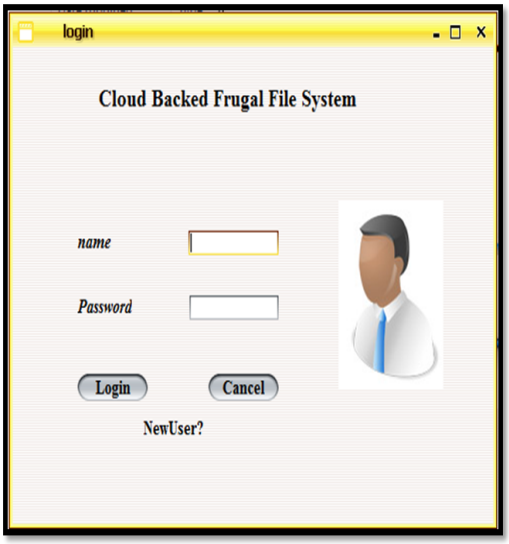
3. RELATED WORK AND EXISTING SYSTEM

In this paper, the service provider, Depot considers safety and liveness guarantees based on distributed sets of untrusted servers. For safety and security reasons, SafeStore, DepSky and SCFS operate on inverse storage service providers to prevent cloud vendor lock-in. Recently, Bluesky presents a cloud based storage proxy for enterprise environments. With the log-structured data layout, the proxy can absorb massive remote write requests to achieve high-throughput. However, the design proposed by Blueksy does not consider the billing model that is a major distinct feature compared to classical storage model. Exemplar open source projects like S3FS, S3QL, and SDFS provide the file system interface for the cloud backend like Amazon S3. Similar to Bluesky, commercial systems such as Cirtas, Nasuni, and TwinStrata act as cloud storage gateways for enterprises rather than personal users. Metadata. In, the authors reported metadata characteristics over 60 K PC file systems in a large corporation during five-year period. The changes in file size, file type, file age, storage capacity and consumption etc. motivated the database design for

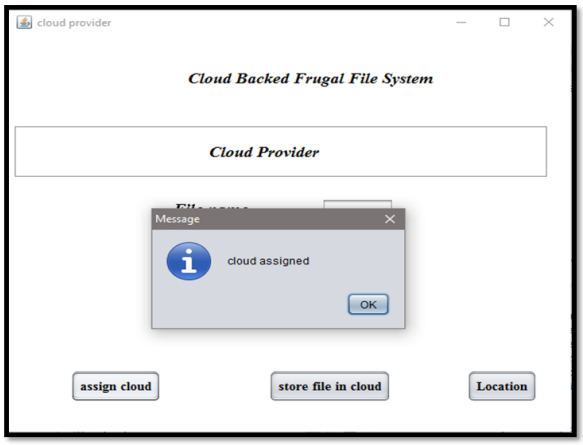
metadata management in our work. Also, the study analyzed the whole-file versus block level elimination of redundancy, which reflects the necessity of block-level deduplication. A recent work showed the behaviors of a popular personal cloud storage service Dropbox. One of the main findings is that the performance of Dropbox is driven by the distance between clients and storage data centers. That corresponds to our design principle on bottleneck shifting to the network. Storage management. To close the widening semantic gap between computer system and storage system, proposed an classification architecture for disk I/O. The same class of objects (e.g., large files, metedata, and small files) are combined according to the caching policy, which boosts end to-end performance.

4. MODULES USED

i. **User: -** In this module the user may be existing or new users. Then the users may register to the cloud process to upload or download process in the certain public or private cloud.



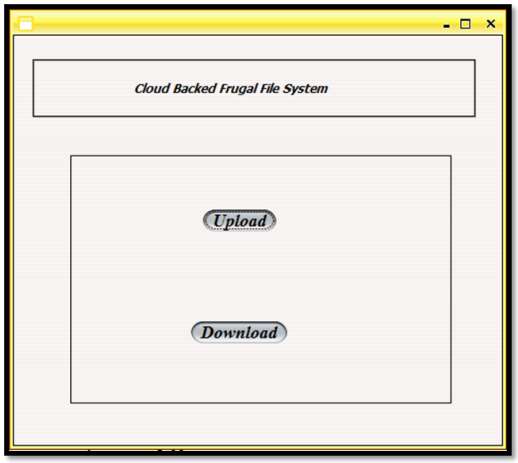
ii. **Cloud process**: - In this module the two can be further maintain are time slot and the process of the cloud process. In the timeslot the cloud can be assigned related to cloud usages. In the validity the data can be maintained and then the related users data can be processed for further requests.



iii. **Process selection**: - In this module the process can be selected related to the users choice. The users can be choose either upload or download of file.

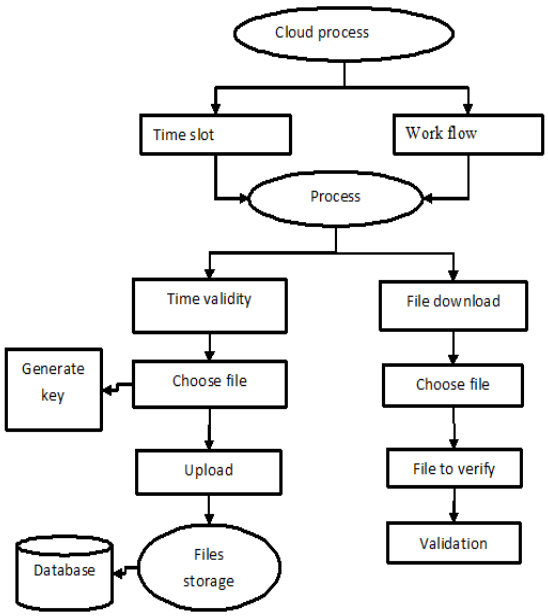
iv. **Upload: -** In this module the data can be uploaded related to the users condition and the key can be generated related to users file selection to upload. The data can be uploaded related to the certain choice of cloud providers to the users.

v. **Download: -** In this module the data can be chosen and the related key can be given to verify the users validity of the data and make the secure transmission. The data can be maintained in the cloud with the security methodology.



5. ARCHITECTURAL DETAILS

A system architecture is a conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.



6. RESULT

The graph shows the performance comparison between the old file systems with the proposed system. The comparison for cost has been shown in blue for the proposed system and the red line shows the old file system cost metrics. The graph shows the comparison of cost along y-axis versus pages along x-axis.

7. CONCLUSION AND FUTURE WORK

This Paper presents the design, implementation and valuation of an enhanced cloud backed frugal file system specifically designed for cloud environments in which improving performance and monetary cost are both principally important for end users. With the efficient data structures and algorithmic designs, it achieves our goals of high performance and cost-effective. In the future, new ways can be investigated to further reduce the storage cost. For example, using byte-addressable compression algorithms, we can precisely control how much data the client needs to download instead of fetching a complete segment each time.

REFERENCES

[1] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud storage with minimal trust," ACM Trans. Comput. Syst., vol. 29, no. 4, pp. 1–38, Dec. 2011.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] R. Kotla, L. Alvisi, and M. Dahlin, "SafeStore: A durable and practical storage system," in Proc. USENIX Annu. Tech. Conf., 2007, pp. 129–142 .

[3] A. N. Bessani, M. P. Correia, B. Quaresma, F. Andr e and P. Sousa, "DepSky: Dependable and decure storage in a cloud-of-clouds," in Proc. 6th Conf. Comput. Syst., 2011, pp. 31–46. http://www.cirtas.com/,2013.

[4] A. N. Bessani, R. Mendes, T. Oliveira, N. F. Neves, M. Correia, M. Pasin, and P. Verıssimo, "SCFS: A shared cloud-backed file system," in Proc. USENIX Annu. Tech. Conf., 2014, pp. 169–180.

[5] M. Vrable, S. Savage, and G. M. Voelker, "BlueSky: A cloudbacked file system for the enterprise," in Proc. 7th Conf. File Storage Technol., 2012, p. 19.

[6] M. Rosenblum and J. K. Ousterhout, "The design and implementation of a log-structured file system," ACM Trans. Comput. Syst., vol. 10, no. 1, pp. 26–52, 1992.

[7] S3fS [Online]. Available: https://code.google.com/p/s3fs/, 2014.Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2014.

[8] S3QL [Online]. Available: ttps://code.google.com/p/s3ql/, 2013.

[9] Open Dudep[Online]. Available: http://www.opendedup.org/, 2015.

[10] Cirtas Bluejet Cloud Storage Controllers [Online]. Available: http://www.cirtas.com/, 2013.

[11] Nasuni: The Gateway to Cloud Storage [Online]. Available: http://www.nasuni.com/, 2015.

[12] TwinStrata [Online]. Available: http://www.twinstrata.com/, 2013.

[13] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch, "A fiveyear study of file-system metadata," in Proc. 7th Conf. File Storage Technol., 2007, pp. 31–45.

[14] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," ACM Trans. Storage, vol. 7, no. 4, pp. 14–20, 2012.

[15] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A.Pras, "Inside dropbox: Understanding ersonal cloud storage services," in Proc. Internet Meas. Conf., 2012, pp. 481–494.

[16] Dropbox [Online]. Available: http://www.dropbox.com/, 2015.

[17] M. Mesnier, F. Chen, T. Luo, and J. B. Akers, Differentiated storage services," in Proc. 23rd ACM Symp. Oper. Syst. Principles, 2011,pp. 57–70.

[18] M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami, and V. Hristidis, "BORG: Block-reorganization for self-optimizing storage systems," in Proc. 7th Conf. File Storage Technol., 2009, pp. 183–196.

[19] F. Chen, D. A. Koufaty, and X. Z. 0001, "Hystor: Making the best use of solid state drives in high performance storage systems," in Proc. Int. Conf. Supercomput., 2011, pp. 22–32.

[20] L. Lin, Y. Zhu, J. Yue, Z. Cai, and B. Segee, "Hot random off-loading: A hybrid storage system with dynamic data migration," in Proc. Simulation Comput. Telecommun. Syst., pp. 318–325.

[21] G. H. Kuenning, "The design of the SEER predictive caching system," in Proc. Workshop Mobile Comput. Syst. Appl., 1994, pp. 37–43.

[22] G. H. Kuenning and G. J. Popek, "Automated hoarding for mobile computers," ACM SIGOPS Oper. Syst. Rev., vol. 31, no. 5, pp. 264–275, Dec. 1997.

[23] Z. Li, Z. Chen, and Y. Zhou, "Mining block correlations to improve storage performance," ACM Trans. Storage, vol. 1, no. 2, pp. 213–245, May 2005.

[24] K. P. N. Puttaswamy, T. Nandagopal, and M. S. Kodialam, "Frugal storage for cloud file systems," in Proc. 7th ACM Eur. Conf. Comput. Syst., 2012, pp. 71–84.