ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

REARED PARSIMONIOUS FILE SYSTEM BY CLOUD

S.Revathi*[1], M.Sathya[2] & G.Simi Margarat[3]

*1&2B.Tech/ Department of Information Technology, RRASE College Of Engineering, Chennai, Tamil nadu, India

3Associate Professor, RRASE College of Engineering, Chennai, Tamil nadu, India

DOI: 10.5281/zenodo.1207830

ABSTRACT

With simple approach interfaces and flexible billing models, cloud storage has become an interesting solution to simplify the storage management for both business and individual users. However, old file systems with large advantage for local disk-based storage backend cannot fully accomplish the internal features of the cloud to obtain good performance. In this paper, we present the design, implementation, and decision of Coral, a cloud based file system that beat a balance between performance and financial cost. Unlike previous studies that treat cloud storage as just a normal backend of existing networked file systems, Coral is designed to address several key issues in improving cloud-based file systems such as the data layout, block management, and billing model. With carefully designed data structures and algorithms, such as analyzing semantically correspond data blocks, KD(k-Dimentional tree based caching policy with self-adaptive thrashing prevention, useful data layout, and excellent garbage collection, Coral achieves good performance and cost savings under various workloads as demonstrated by large evaluations.

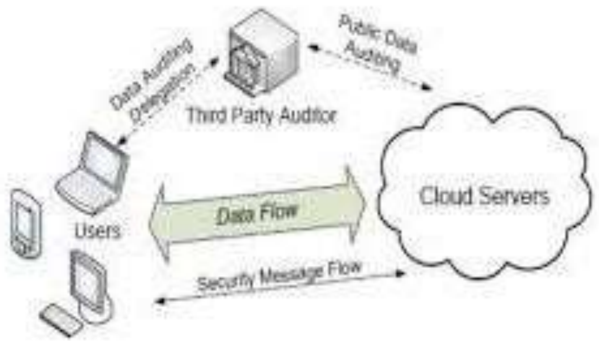
I. INTRODUCTION

The Platform-as-a-Service (PaaS) cloud storage has be- come an infrastructure service of the Internet as a promising way to simplify storage management for enter- prises and individual users. Coupled with the increasing demand for multi-device data synchronization and sharing, it is emerging as a new paradigm that helps migrate storage applications to the cloud. Due to its practical impact, significant research endeavors have been undertaken to address the problems in cloud storage based applications, such as the security of storage outsourcing, data consistency cost optimization.

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

[1] 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, avoid- ing undue resource consumption accordingly. As for cost optimization, FCFS
[2] 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. By dynamically adapting the storage volume sizes of each service, FCFS reduces the cost of operating a file system in the cloud. In this paper, 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 Coral 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 [3] and the high latency of the WAN. Thus, a better cache design is needed in order to save the operational cost for a cloud based file system.



To this end, we propose a cache eviction mechanism based on kd-tree [4] that is organized by considering the correlation metrics among data blocks. The proposed mechanism guarantees high performance in identifying and evicting cached content with flexible range selection of data blocks. Further- more, the data layout of concrete objects in the cloud also reflects the semantic correlation of data blocks, which has direct and important

Implications for several optimizations in our system, including data prefetching, cache thrashing minimization, and garbage collection. Integrated with the quantified model from analyzing the billing items of the cloud service, Coral orchestrates the interfaces provided by the cloud vendor and achieves improvement in both execution time and monetary cost as compared to existing systems

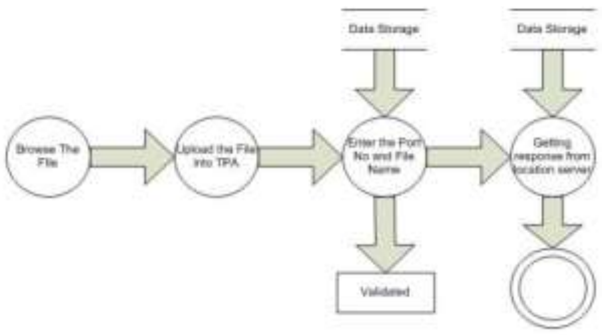
II. RELATED WORK AND EXISTING SYSTEM

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 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 previous studies that treat cloud storage as just a normal backend of existing networked file systems, Cache policy. Beyond typical LRU mechanism, SEER [5] improves cache performance using the sequence of file accesses for measuring the relationship among files. The follow- up study [6] discussed various semantic distances of files and designed an agglomerative algorithm in disconnected hoarding scenario. Also, [7] proposed a twofold 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 [7] evaluate cloud storage costs from economic perspective, which is at more abstract level and may not be suitable for complex usage scenarios. In [8], 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.

III. BLUE SKY FILE SYSTEM

Blue Sky is a file system of network based cloud storage. Cloud backup provide a persistent data storage, provided by Amazon s3 or windows Azure. Blue sky allows the consistency and large storage capacity and reduce use of hardware sever. Client access the cloud storage server with help of proxy running on –site. Cloud optimization is achieved by log-structured design and secure cloud log cleaner.



It uses multiple protocols as NFS and CIFS for various providers. Blue sky file system is maintained by object data structures format and log structured format [9] for cloud organization. Blue sky provides version ed store data for backups in the file system.

Object Types
Blue Sky uses four types of objects for representing data and metadata in its log-structured file system [7] format: data blocks, inodes, inode maps, and checkpoints. These objects are log segments for storage. Illustrates the irrelationship in the layout of the file system. On top of this physical layout BlueSky provides standard POSIX file system semantics, including atomic renames and hard links.

Data blocks store file data. Files are broken apart into fixed- size blocks (except the last block may be short). BlueSky uses 32 KB blocks instead of typical disk file system sizes like 4 KB to reduce overhead: block pointers as well as extra header information impose a higher per-block overhead in BlueSky than in an on-disk file system. In the evaluations in Section we show the cost and performance tradeoffs of this decision.

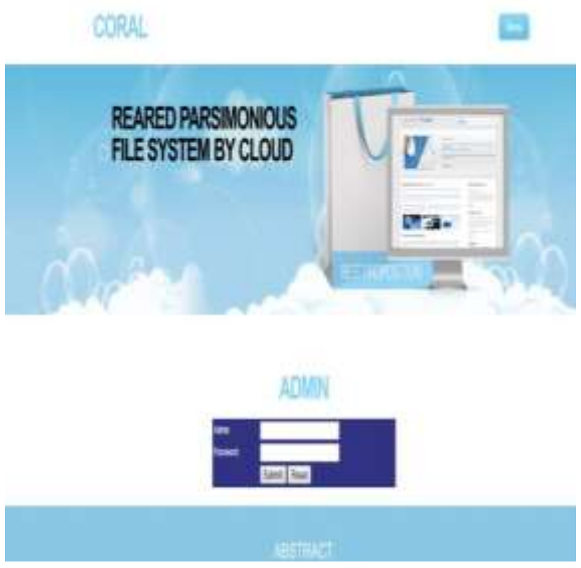
Clouds Log
For each file system, BlueSky maintains a separate log for each writer to the file system. Typically there are two: the proxy managing the file system on behalf of clients and a cleaner that garbage collects overwritten data.

IV. MODULES

Authentication module
In this module we are registering our details and we will get the unique id. So we can use cloud at any time with this id. It can be prevent our data safely.

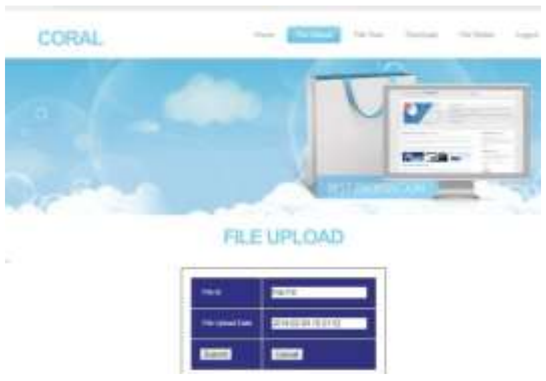
Creating cloud environment

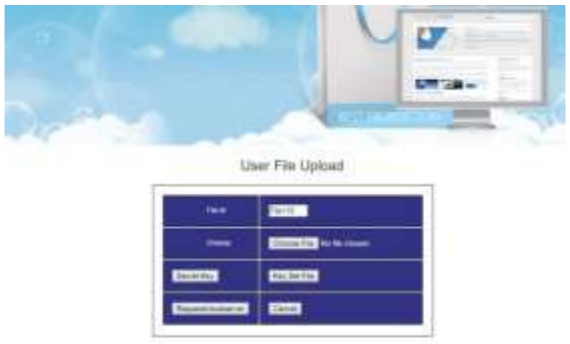
The Platform-as-a-Service (PaaS) cloud storage has become an infrastructure service of the Internet as a promising way to simplify storage management for enterprises and individual users. In this we are creating a cloud environment



Uploading and downloading

In this module we are performing the operations we need file upload or download. If you pay amount and we will get desire space of cloud.





Data layout module

In this, we are developing a layout that is Customized a data to separate a block, easily fetch a data effective and cost saving. we are developing with garbag collector**.**

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

FILE ID	FILE NAME	DATE
[illegible]	[illegible]	[illegible]
[illegible]	d.txt	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	encrypt.txt	[illegible]
[illegible]	[illegible]	[illegible]

V. CONCLUSION

This paper presents the design, implementation and evaluation of Coral, a cloud based 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, Coral achieves our goals of high performance and cost-effective

VI. REFERENCES

[1] M. Vrable, S. Savage, and G. M. Voelker, "BlueSky: a cloud-backed file system for the enterprise." FAST, p. 19, 2012.
[2] K. P. N. Puttaswamy, T. Nandagopal, and M. S. Kodialam, "Frugal rage for cloud file systems." EuroSys, pp. 71–84, 2012.
[3] S. Jiang, F. Chen, and X. Zhang, "CLOCK-Pro: an effective improvement of the CLOCK replacement," pp. 35–35, Apr. 2005.
[4] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching." Commun. ACM (), vol. 18, no. 9, pp. 509– 517, 1975.
[5] A. N. Bessani, R. Mendes, T. Oliveira, N. F. Neves, M. Correia, M. Pasin, and P. Ver´ıssimo, "SCFS: A Shared Cloud-backed File System." USENIX Annual Technical Conference, pp. 169–180, 2014.
[6] M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami,andV.Hristidis,"BORG:Block-
[7] reORGanizationforSelfoptimizing Storage Systems." FAST, pp. 183–196, 2009.
[8] F. Chen, D. A. Koufaty, and X. Z. 0001, "Hystor: making the best use of solid state drives in high performance storage systems." ICS, pp. 22–32, 2011.
[9] Y. Chen and R. Sion, "To cloud or not to cloud?: musings on costs and viability." SoCC, pp. 29–7, 2011.
[10] Google AWS . [Online]. Available: https://cloud.google. com/products/cloud-storage.
[11] Google Cloud Storage. [Online] Available: https://cloud.google.com/slideshare.net

CITE AN ARTICLE

Revathi, S., Sathya, M., & Margarat, G. S. (n.d.). REARED PARSIMONIOUS FILE SYSTEM BY CLOUD. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, *7*(3), 827-831.