**International Journal of Management, IT & Engineering**
Vol. 9 Issue 1, January 2019,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

# CORAL : A CLOUD-BACKED FRUGAL FILE SYSTEM

**Dr.M.I.Thariq Hussan** [*]

**K.Krupakar, T.Satwik, Y.Ranaveer** [**]

**Abstract**

Cloud storage is becoming booming field in the software technology. Cloud storage had made possible to store large file for each enterprises and individual users. However, ancient file systems with intensive optimizations for native disk-based storage backend can't absolutely exploit the inherent options of the cloud to get fascinating performance. During this paper, we present the look, implementation, and analysis of Coral, a cloud based frugal filing system that a balance between performance and financial value. At present dramatically increase in the business and internet applications, storage is becoming a major issue in cloud computing. Storage costs are increasing day- by-day. Cloud backed is backing up data that involve distribution copy of the data over a community network to an off-site server. Uncomplicated access interfaces and elastic billing models, cloud storage has become a gorgeous solution to make simpler the storage organization for both enterprises and individual users. This paper presents a survey on the different cloud backed frugal file system. This enables effective storage management, increase the performance and reduce the cost in the cloud. The cloud is used efficiently in this system and performance is improved.

**Keywords:** Cloud computing, Coral, Frugal file system, kd-tree, cloud backup

[*] **Professor & Head, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad**

[**] **B.Tech Scholars, Department of Information Technology, Guru Nanak Institutions Technical Campus, Hyderabad**

## I. Introduction

The Platform-as-a-Service cloud storage has become an infrastructure service of the Internet as a promising way to simplify storage management for enterprises 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.

Cloud computing is used for storing and managing the data in a remote location through internet rather than the local server or a private system. Cloud computing is shared pools of configurable computer system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility [1]. Cloud Backed is models that provide data backed up remotely, maintained and managed. Users access the data through the network. Users normally compensate for their data storage on cloud as per-usage or monthly rate. The cloud Storage providers provide a platform as a service, is one of the infrastructure service on cloud storage to shorten storage management for enterprises and personality users [2].

### Cloud Computing

Cloud computing is shared pools of configurable computer system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility.

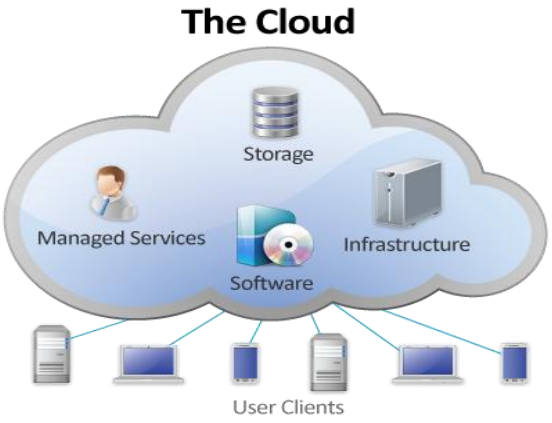


**Fig 1: Cloud Computing**

We start with the discussion on background and related work in Section II, followed by a problem statement on Section III. Section IV explains the proposed work while Section V presents the comparison and Section VI concludes the work.

## II. Background and Related Work

### Blue Sky

BlueSky is a network file system backed by cloud storage. It stores data persistently in a cloud storage provider such as Amazon S3 or Windows Azure, allowing users to take advantage of the reliability and large storage capacity of cloud providers and avoid the need for dedicated server hardware. Clients access the storage through a proxy running on-site,

which caches data to provide lower-latency responses and additional opportunities for optimization. We describe some of the optimizations which are necessary to achieve good performance and low cost, including a log-structured design and a secure in-cloud log cleaner. BlueSky supports multiple protocols both NFS and CIFS and is portable to different providers.

BlueSky provides service to clients in an enterprise using a transparent proxy-based architecture that stores data persistently on cloud storage providers shown in Fig 2. The enterprise setting we specifically consider consists of a single proxy cache located with enterprise clients, with a relatively high-latency yet high-bandwidth link to cloud storage, with typical office and engineering request workloads to files totaling tens of terabytes. This section discusses the role of the proxy and cloud provider components, as well as the security model supported by BlueSky [3].

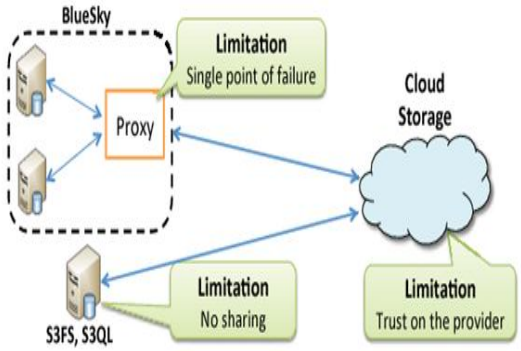


**Fig 2: Cloud-backed file system and their limitations**

**Distributed File System (DFS)**

Distributed File System (DFS) is a set of client and server services that allow an organization using Microsoft Windows servers to organize many distributed SMB file shares into a distributed file system. DFS has two components to its service: Location transparency (via the namespace component) and Redundancy (via the file replication component). Together, these components improve data availability in the case of failure or heavy load by allowing shares in multiple different locations to be logically grouped under one folder, the "DFS root" [4]. There is no requirement to use the two components of DFS together; it is perfectly possible to use the logical namespace component without using DFS file replication, and it is perfectly possible to use file replication between servers without combining them into one namespace.

A DFS root can only exist on a server version of Windows (from Windows NT 4.0 and up) and OpenSolaris (in kernel space) or a computer running Samba (in user space.) The Enterprise and Datacenter Editions of Windows Server can host multiple DFS roots on the same server. OpenSolaris intends on supporting multiple DFS roots in "a future project based on Active Directory (AD) domain-based DFS namespaces". DFS Root is DFS replicas through DFS link shown in Fig 3.

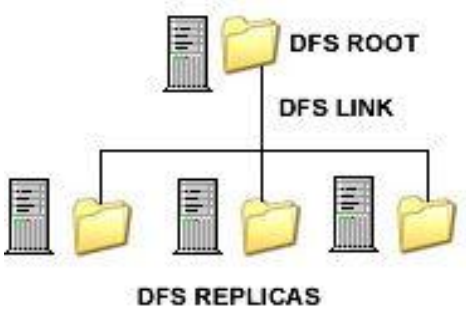


**Fig 3: Distributed File System**

**SCFS: A Shared Cloud-backed File System**

File backup, data archival and collaboration are among the top usages of the cloud in companies, and they are normally based on cloud storage services like the Amazon S3, Dropbox, Google Drive and Microsoft SkyDrive. These services are popular because of their ubiquitous accessibility, pay-as-you-go model, high scalability, and ease of use. A cloud storage service can be accessed in a convenient way with a client application that interfaces the local file system and the cloud. Such services can be broadly grouped in two classes: Personal file synchronization and cloud-backed file system [5].

File synchronization software is used to store copies of on-premises data to another device or to the cloud. The files are typically available to be accessed via a Web-based portal. Some examples of file sync software include Box and Dropbox. Personal File synchronization service is a service provided to keep files that are stored in several different physical locations up to date. Cloud and storage vendors often offer software that helps with this process. SCFS agent implements most of the functionalities of SCFS. The figure shows the backend cloud storage as cloud-of-clouds formed by Amazon S3, Google Storage, RackSpace Cloud Files and Windows Azure. To store data in the cloud, SCFS uses DepSky. More specifically, uses DepSky protocol once we want address the main cloud storage limitations. SCFS allows also a different backend formed by only one cloud as shown in the Fig 4 [6].

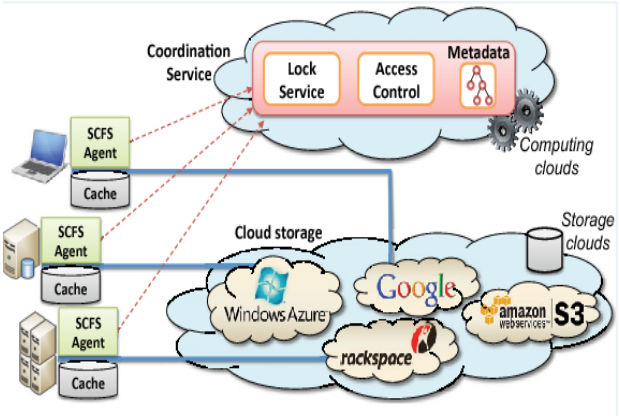


**Fig 4: SCFS Shared Cloud-backed File System**

**DEPSKY**

Dependable and Secure Storage in a cloud application extend their accessibility through the Internet by using large data centers and powerful servers that host web applications and services. Anyone with a suitable Internet connection and a standard internet browser can

access a cloud application. Rapid evolution of cloud computing technologies can easily blur its definition perceived by the public.

**Cloud Backed File System**

Cloud backup [7] also identified by online backup, is an approach for backing up data that involves a replica of the data over a public network to an off-site system. Cloud Backed is models that provide data backed up remotely, maintained and managed. Users access the data through the network. Users normally compensate for their data storage on cloud as per-usage or monthly rate. The cloud Implementing cloud data backup is able to help boost an organization data protection without raising the workload on information technology. Online backup systems are classically built a client software application that run on a program determined by the purchase stage of service. Cloud backups contain the software and hardware component to keep an organization's data, include applications Exchange and SQL Server. Online backup is used by small and medium sized businesses (SMBs) and larger enterprises to back up the data. For larger organization, cloud data backup as a complementary form of backup.

**Frugal Cloud File System**

Different embodiment grant a techniques and tools of providing a frugal cloud file system [9] that proficiently uses the blocks of different types of storage devices with special properties for various purposes. The various types of storage strategy reduce the storage and bandwidth transparency. Favorably, the storage and bandwidth reduction in the clouds achieved by the frugal cloud file system, reduce the cost-effective of managing the file system at the same time, maintain high performance. Frugal file system is a structure that optimizes on the whole storage cost between various Cloud storage services and different type of price. The Frugal file system's storage services like a twin storage system, one is low latency (e.g., Amazon ElastiCache) and the other one is high latency (e.g., Amazon S3). In low latency the data transfer cost is low and cost of storage per byte is high (i.e. cache) but the high latency, cost of storage per byte is low and data transfer cost is high (i.e. disk).

**CHARON**

CHARON is one of the cloud backed file system that able to store and share the large amount of data between various cloud providers and cloud storage system in secure, reliable manner. The two main feature of CHARON is server less design and efficient management of file system. CHARON supports three types of data locations as cloud of clouds, public cloud storage and private cloud storage. Cloud of clouds provides multi cloud availability, confidentiality. Single storage cloud is low cost compared to cloud of clouds but it requires confidence provider. Private cloud storage based on adopted method and solution, also provides the dependability level. CHARON data are separated by file data and Metadata. Metadata are stored in cloud of clouds. CHARON use data centric. Byzantine-resilient leasing algorithm ignores the concurrency conflicts. CHARON divides the files into constant size blocks. Files are stored in various data location based on the requirements. POSIX interface is provided by CHARON that allow the user interact with any file system.[10]

## III. Problem Statement

In traditional file system, extensive optimizations for local disk-based storage backend cannot fully exploit the inherent features of the cloud to obtain desirable performance by intelligently organizing storage objects in a local cache. It is based on HTTP-based interface. A server-side web API is a programmatic interface consisting of one or more publicly exposed endpoints to a defined request-response message system typically expressed via the web-most commonly by means of an HTTP-based web server. High cost file system based on the cloud storage.

## IV. Proposed Work

Here we present the design, implementation, and evaluation of Coral, a cloud based file system that strikes a balance between performance and monetary cost. Storage objects are not stored in a local cache. The technique proposed is FCFS (First Come First Served). 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 structures by dynamically adapting the storage volume sizes of each service. The design and implementation of a cost-effective file system based on the cloud storage. A frugal storage model optimized for scenarios concerning multiple cloud storage services.

### Basic Design of Proposed Work

### Splitting and Merging Algorithm

Step 1: We first discuss the procedure of evicting cached blocks to make room for new data. The metadata including the last access time of blocks/inodes, the size and access counts for each block in the cache space, is queried from the database.

Step 2: Commonly, when a block is selected as a candidate for eviction, it can be locked to prevent any further accesses. However, the eviction operation consists of several sub-steps such as metadata querying, data merging, and encryption/compression, each demanding different. The main steps performed by the caching subsystem.

Step 3: Cache replacement logic in Coral. The data unit downloaded from the cloud is a segment that contains data blocks with correlated relationship, and its structure. Segment based data transfer also implies that blocks swapped in can act as pre-fetched data to improve the hit rate of future requests. In addition, we use the feature of HTTP range request (indicated by the parameters from and to) to precisely control the amount of data remotely fetched, which is marked by green squares for the cloud storage layer. Fig 5 describes the file system, cache and cloud storage services.

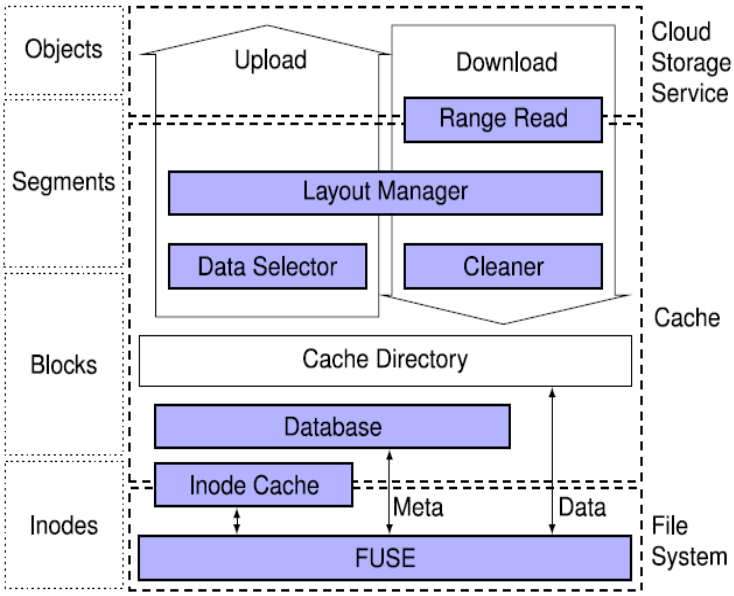


**Fig 5: Proposed network architecture**

## V. Comparison

Existing System in Traditional file system extensive optimizations for local disk-based storage backend cannot fully exploit the inherent features of the cloud to obtain desirable performance. High cost file system based on the cloud storage. Local disk based storage is implemented by intelligently organizing storage objects in a local cache. Most existing systems uses HTTP-based interface. A server-side web API is a programmatic interface consisting of one or more publicly exposed endpoints to a defined request–response message system, typically expressed via the web most commonly by means of an HTTP-based web server. High cost file system based on the cloud storage. Local disk-based storage Cloud storage file recovery is faster and effectiveness. In cloud storage the recovered data are located, streamed through the WAN. It minimizes the time and eliminating the infrastructure of local tape. It is a consistent and efficient technology.

## VI. 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. In the future, we plan to investigate new ways 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. N. Agarwal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch (2007), A five year study of file-system metadata, Proceedings of 7th Conference in File Storage Technology, pp. 31-45.
2. J. L. Bentley, Multidimensional binary search trees used for associative searching, ACM, vol. 18, no. 9, pp. 509-517.
3. https://en.wikipedia.org/wiki/Open_Shortest_Path_First.
4. https://en.wikipedia.org/wiki/Distributed_File_System_Microsoft.
5. Alysso Bessani, Ricardo Mendes, Tiago Oliveira, Nuno Neves, Miguel Correia, Marcelo Pasin, Paulo Verissimo (2014), SCFS: A Shared Cloud-backed File System, USENIX Annual Technical Conference Philadelphia.pp.169-180
6. http://cloud-of-clouds.github.io/SCFS.
7. Shivam Raj, Puneet Gupta, SanjnaSinha, Priyansh Bendre (2015), International Journal of Innovative Science, Engineering & Technology, pp. 648-654.
8. N. Bessani, M. P. Correia, B. Quaresma, F. Andre and P. Sousa (2011), DepSky: Dependable and decure storage in a cloud-of-clouds, Proceedings of 6th Conference in Computer Systems, pp. 31-46.
9. N. Bessani, R. Mendes, T. Oliveira, N. F. Neves, M. Correia, M. Pasin, and P. Verssimo (2014), SCFS: A shared cloud-backed file system, USENIX Annual Technical Conference, pp. 169-180.
10. P. Rohini , K. Palraj , A Survey on Coral - A Cloud-Backed Frugal File System (2016), International Journal of Advanced Research in Computer and Communication Engineering, pp. 459-464.