International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e): 2321-2063
Volume-5, Issue-12, Dec.-2017, http://iraj.in

A REVIEW ON UTILIZATION OF MEMORY MANAGEMENT MECHANISM OF CLOUD COMPUTING

[1]M. PRABU, [2]B. SAI PRANEETH, [3]B. SAI SUMANTH, [4]M. VENKATA VIJAYA RAMA RAJU, [5]S. TEJA VENKATA RAMA RAJU

[1]Assistant Professor, SRM University, Chennai
[2,3,4,5]UG Scholars, SRM University, Chennai
E-mail: [1]manavalanprabu@gmail.com, [2]saipraneethbrg19@gmail.com, [3]sai.sumanthbattula@gmail.com, [4]mandapativijay43@gmail.com, [5]stvramaraju@gmail.com

Abstract - The usage of cloud computing has rapidly increased during past ten years in many fields. Many cloud providers say Google Drive, One Drive, etc are attracting the users with free data storage. Though all the clouds provide storage, user-friendly environment and easy to access from anywhere, there are many differences in them when it comes to security features, storage size, energy efficiency, and ease of access. The project mainly focuses on accessing all the clouds through a single platform and providing the efficient cloud by analyzing the features of respective clouds. We are going to design a user-friendly website which enables the user to sign-in into the public clouds and upload a file such that result is obtained to the user stating "X" cloud is more efficient in the available clouds. So, one can use the analysis and upload the file in an efficient cloud. It also gives the user, the next efficient cloud among available clouds when the most efficient cloud is out of space.

Finally, this project provides the user a clear idea about the cloud services and storing the data in the efficient cloud more efficiently and managing the clouds.

Keywords - Memory Management Mechanism(MMM), Cloud Storage Analysis(CSA), Efficient Cloud Identification(ECI), Space Management, Personal Cloud, Quality of Experience(QoE).

I. INTRODUCTION

Cloud storage services (e.g., Google Drive, Dropbox) are a popular means for storing data and performing collaborative work. Personal cloud storage services are data-intensive applications on the Internet that allow the user to synchronize files with servers in the cloud and among different devices [1]. Cloud computing paradigm provides with end-users and on-demand access to a shared pool of resources [3].The high public interest has pushed dozens of providers to the cloud storage market. New cloud providers have to compete against established ones such as Google, Microsoft, Drop Box, Box, which offer large amount of storage space for cheaper prices, while the high competition for customers continues to decrease the cost per GB, and the other important aspect is if synchronization, performance, and Quality of Experience (QOE), are mostly unknown given the proprietary design of most services. In competing for the customers in the crowded market, it is to be expected that performance and QOE plays a major role along price to attract customers in future [1].

So, keeping the Quality of Experience (QoE) [1] in mind, we design a user-friendly website in which:

- We will manage all the public clouds at one place in which we have the account.
- Then according to the analysis that we have done by uploading 1MB and 5MB files in to the cloud storage under same network conditions, and also taking the properties like Chunking, Bundling, Compression, Delta Encoding and P2P synchronization [1] in to consideration, we have found the efficient clouds among which we have under analysis.
- When the user is going to upload file, a website will show the most efficient cloud among the cloud storage according to the data of the analysis.
- If the most efficient cloud storage was out of space, the website will show the next efficient cloud to store the file.

The methodologies used in public clouds are:

- **BUNDLING**: When a batch of files is transferred, files will be bundled, so, that transmission latency and control overhead are reduced. Our experiment to check how services handle batches of files consists of 3 file sets of 1MB: (i) 1 file of 1MB (ii) 10 files of 100KB (iii) 100 files of 10KB. [1]
- **CHUNKING**: Large files can either monolithically transmitted to the cloud or chunked into smaller pieces. Chunking is advantageous because it simplifies recovery in case of failures. [1]
- **COMPRESSION**: In general, reduce traffic and storage requirements at the expense of local processing time. We check the compression capability by contrasting the number of Bytes observed in the network with the original benchmark size when submitting highly

compressible text files sizes from 100KB to 1MB. [1]

- **DEDUPLICATION**: server data de duplication eliminates replicas on the storage server. Client-side de duplication instead extends the benefits to clients and the network: incase a file is already present on the server; replicas in the client can be identified to save the upload capacity. This can be accomplished by calculating a file digest using the file content (e.g., SHA256 is used by Dropbox). The digest is sent to servers prior to submitting the complete file. Servers then check whether the digest is already stored in the system and skip the upload of repeated content. [2]
- **DELTA ENCODING**: Delta encoding calculates the difference between file revisions along the transmission of only the modified portions. Indeed, delta encoding provides similar benefits as the combination of chunking and de duplication, but with a finer granularity. It may have the positive impact on the performance when files are frequently changed-e.g. when people perform collaborative / literature work. On the other hand, the storage of static content is not affected by this feature. [1]
- **P2P SYNCHRONIZATION**: Devices hosting common files could be synchronized without retrieving every content from the cloud, thus saving both network and sever resources. Dropbox is known for implementing a LAN Sync protocol that allows devices, possibly from different users, to exchange content using P2P synchronization this manual inspection reveals that Dropbox offers such capability. [1]

In this way, we are going to find out the best cloud for storing the particular uploading file in an efficient way and making use of each and every cloud account.

II. LITERATURE SURVEY

A. Personal Cloud Storage Benchmarks and Comparison

Authors: Enrico Bocchi, Idilio Drago, Marco Mellia

A methodology has been developed to understand and benchmark personal cloud storage services. This methodology unveils the architecture and capability of a cloud. More over by means of repeatable and customizable tests, it allows the measurement of performance metrics under different workloads. The effectiveness of this methodology is shown in a case study in which 11 services are compared under the same conditions. Our case study reveals interesting differences in design choices. Their implications are assessed in a series of benchmarks. Results show no clear winner, with all services having the potential for improving performance. In some scenarios, the synchronization of the same files can take 10 times longer. In other cases, we observe wastage of twice as much network capacity, questioning the design of some services. This methodology and results are thus useful both as benchmarks and as guidelines for system design and our root of the project. [1]

B. Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage

Authors:Junbeom Hur, Dongyoung Koo, Youngjoo Shinz and Kyungtae Kangx

A two-phase space management framework for the large-scale storage system in this paper. For the collection phase, we proposed a new interval collection strategy, called the greedy bi-direction collection; and then for the allocation phase, we formulated it to a variant of bin packing problem and applied two heuristics to allocate the collected intervals to new added storage nodes. It is possible to extend the proposed algorithms to the distributed storage system, which can generate far less metadata of the space management efficiently serve the upper level requests. It is possible to make the large-scales torage system grow much larger. In future work, we plan to apply our proposed algorithms to build reliable cloud storage system like and evaluate its practical performance [2].

C. Security and Efficiency Trade-offs for Cloud Computing and Storage

Authors: Jian Li, Kai Zhou and Jian Ren

Cloud computing provides with end-users and on-demand access to shared pool of resources. Computational resource and storage are two of the most integral services of cloud computing. Instead of storing a file and its replication on multiple servers through minimum storage regeneration (MSR) [3] and minimum band width regeneration (MBR) [3], which can achieve optimal storage efficiency and storage capacity. To detect and correct malicious nodes with high error correction capability at much lower computation efficiency can be done by cost-aware server outsourcing (CASO) [3].

D. NewBalance: Efficient Data Space Management and Algorithmic Optimization for Large-Scale Storage Systems

Authors: XU Guangping, LIN Sheng, SHI Kai and ZHANG Hua

The demand for large scale storage system like data center is ever-growing data amount. As the infrastructure for various upper-level applications, the space management should meet efficient and scalable storage requirements [4].

Fragmentation usually occurs when data space of original storage nodes has to be reallocated to new added storage nodes during the scale out evolution of the large-scale storage system. An efficient space management framework, called New Balance, to reduce fragmentation with the minimum data movement while keeping the storage system load balance. The experimental results show that the amount intervals can be reduced by 25%-50% and the algorithmic optimization improves data lookup performance by at least 10% and the scale-out performance by two times [4].

E. Efficient Storage Management for Aged File Systems on Persistent Memory
Authors: Kaisheng Zeng, Youyou Lu, Hu Wan, and Jiwu Shu
Persistent memory (PM) [5] can be connected to CPU via direct memory access path, which offers advantage of fast and fine-grained access. To keep track of free space, a file system maintains a free space list. Frequently the free space list is implemented as a free list or in use list. Flexible bit map segment tree is developed to efficiently manage a region of file system, free space on persistent memory. Each block of data is represented in bits. If block is allocated bit is 1 else it is 0.

F. OmniBox: Efficient Cloud Dropbox and Box
Authors: Huu Dinh, Alexander Dworkin, Christopher O'Neill, Scott Savage, Jimmy Leak, Mohammad Aazam, Marc St-Hilaire
The invention of cloud computing has completely changed the way to store and manage data. In the paradigms like Internet of Things (IoT), there will be a lot of data to be stored and managed in the cloud. As it was important to know which cloud was best for the particular data management tasks.The first goal of Omnibox[6] is to provide an evaluation of different cloud providers.It will provides statistics such as upload throughput, download throughput, jitter, single-key and multi-key user accounts, and concurrent download time. It is going to provide a unified data management service where users can manage files from different cloud providers within a single interface. It provides a feature called smart upload, where cloud providers are evaluated and suitable service is selected for uploading a file based on Quality of Service (QoS)[6], file-type, file-size, upload throughput, download throughput, available space, jitter, and latency. [6]

G. Transmission Time Estimator for Social and Cloud Applications in Smartphones
Authors: Amit Panghal, Kannan Govindan and Karthikeyan Subramaniam
Present day users tend to use Smartphone for heavy load multimedia file sharing, and require a lot of upload bandwidth. However, the bandwidth is an expensive affair and upload could take a lot of time in a limited bandwidth scenario. A solution has been developed to estimate time to share or upload multimedia content from a Smartphone to social networks or cloud servers. Time to transmit information prior to transmission can help system/user manage uploads better. The problem of estimating time to upload prior to start and this paper [7] proposed a machine learning approach to solve the problem. Appropriate machine learning model for the problem, feature collection, training and evaluation has been handled effectively. Features of different clouds have been taken into consideration and time for uploading files has been implemented using machine learning and efficient cloud is determined [7].

III. ANALYSIS

Analysis is done based on the time taken and the properties of respective clouds. The efficiency depends on the time taken to upload/download a file. In this analysis, we are going to take some n clouds and calculate the time taken to upload a file in various network speeds. We are going to analyze the time taken for uploading different formats of files and sizes. By this analysis we can prove the efficient cloud to store the data without wasting time. Based on the analysis in our working model we provide the list of efficient cloud for the selected file to upload based on time. The list is given in the order of efficient clouds and their properties.

For measuring download speed, the formula will be
AverageSpeed = Smoothing_factor * LastSpeed + (1-Smoothing_factor) * Average_Speed;

Smoothing_factor:-
Smoothing Factor is a number between 0 and 1. If the files become older they will be discarded. As you can see in the formula, when Smoothing_factor is 1 you are simply using the value of your last observation. When smoothing_factor is 0 AverageSpeed never changes. So, you want something in between, and usually a low value to get decent smoothing. I've found that 0.005 provides a pretty good smoothing value for an average download speed.

LastSpeed:-
LastSpeed is the last measured download speed. You can get this value by running a timer every second. And also you can calculate how many bytes have been downloaded since the last time you ran it.

AverageSpeed:-
AverageSpeed is used to calculate the estimated time remaining to upload/download. Initialize this to the first LastSpeed measurement you get while calculating.

A. Architecture diagram
The architecture diagram depicts the situation of a user interacting with the system. User uploads the file and waits for the result to be analyzed. User now gets the result of efficient cloud analyzed by the system,

then user can upload files directly to the efficient cloud.

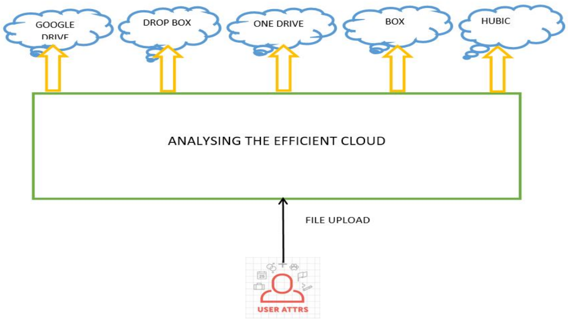

Fig 3.1 Architecture diagram

B. Analysis of clouds bassed on speed

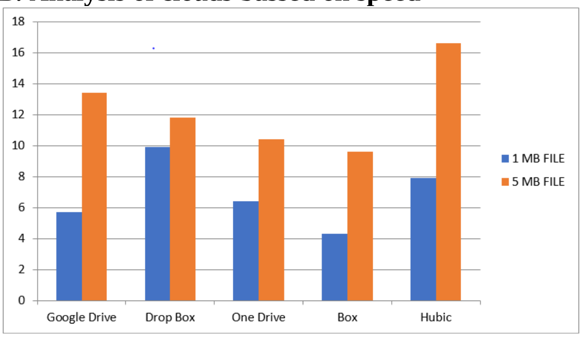

Fig 3.2 Analysis of clouds at 1mbps speed

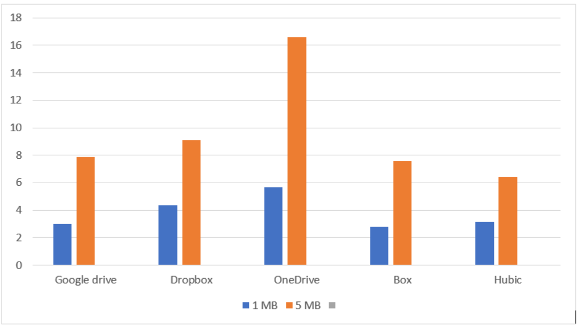

Fig 3.3 Analysis of clouds at 35 mbps speed

Above graphs represent the time taken to upload 1MB,5MB files by different clouds at same external conditions, we have drawn the graph assuming time variable on vertical axis and different clouds on horizontal axis. It provides us a clear idea about the variation of efficiencies of selected clouds from which we can conclude that all the drives are not equally efficient which is root for this project.

IV. PROJECT OUTCOME

The project currently integrates n number of clouds in a single page which allows user to access the clouds from a single platform so that all details of the respective clouds such as storage, efficiency are shown. Analysis is done for every file that is being uploaded and the efficient cloud is shown for storing the file in a practical way. The above graph shows the details of the analysis of a particular file with different network speeds. The uploading of a file depends on the network speed also so that all the parameters that are involved in the process are taken into consideration and analysis is done which helps the user to store his data. We are going to develop a web page which consists of all the clouds and if user logins into the clouds with the credentials, from then data can be stored in the efficient cloud at that time which depends on the analysis.

CONCLUSION

In our project, we present on the management of different public clouds according to their efficiency order, which was an important research topic for public clouds management. Here we have done analysis over different public clouds with respect to the time under the same network conditions with the file sizes of 1MB and 5MB, through which we found out the hierarchy of efficient clouds. By using the analysis and design we are going to create a webpage, where we will manage all the public clouds in one place. So whenever we will upload a file it will be uploaded to the most efficient cloud and if the most efficient cloud is out of space it will choose the next efficient cloud in the hierarchy order of efficiency by which we can use each and every public cloud present in our webpage

REFERENCES

[1] Enrico Bocchi, Idilio Drago, Marco Mellia for "Personal Cloud Storage Benchmarks and Comparison "(2015).
[2] Junbeom Hur, Dongyoung Koo, Youngjoo Shinz and Kyungtae Kangx for "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage "(2017).
[3] Jian Li, Kai Zhou and Jian Ren for "Security and Efficiency Trade-offs for Cloud Computing and Storage" (2017).
[4] XU Guangping, LIN Sheng, SHI Kai and ZHANG Hua for "NewBalance: Efficient Data Space Management and Algorithmic Optimization for Large-Scale Storage Systems"(2017).
[5] Kaisheng Zeng, Youyou Lu, Hu Wan, and Jiwu Shu for "Efficient Storage Management for Aged File Systems on Persistent Memory"(2017).
[6] Huu Dinh, Alexander Dworkin, Christopher O'Neill, Scott Savage, JimmyLeak, Mohammad Aazam, Marc St-Hilaire for "OmniBox: Efficient Cloud Dropbox and Box"
[7] Amit Panghal, Kannan Govindan and Karthikeyan Subramaniam for"TransmissionTime Estimatorfor Social and Cloud Applications in Smartphones"
[8] https://stackoverflow.com/questions/2779600/how-to-estimate-download-time-remaining-accurately