ATLANTIS PRESS

Advances in Engineering Research (AER), volume 142

International Conference for Phoenixes on Emerging Current Trends in Engineering and Management (PECTEAM 2018)

Effective Remote Communication for Cloud Backup

[1]Sai Krishna K, [2]D. John Aravindhar, [3]M.N.Sushmitha
[1]Undergraduate Student, [2]Professor, [3]Assistant Professor
Department of Computer Science and Engineering,
Hindustan institute of technology and Science, Padur, Chennai, India kattamanchisaikrishna1996@gmail.com,
jaravindhar@hindustanuniv.ac.in, mnsushmitha@hindustanuniv.ac.in

Abstract:**Today we live in a world where tons of data is been generated, so in order to store all these data, we need many numbers of servers and networking which are more expensive in buying and maintaining them which creates more complexity. In order to resolve this problem we had implemented cloud computing i.e it provides IT resources as the service to the user and charge as per their usage. Generally public cloud lacks security and there is a loss of control over the data center so we implement a private cloud in an organization as it has proper security and also there is a complete control over the data center, this private cloud provides us the system and the services like SAAS, PAAS and IAAS limited to an organization. This paper tells us how to establish a private cloud in an organization. When we establish a private cloud there should be a proper backup i.e copying the data for limiting the loss in case of a corruption or failure, 90% of the companies who did not have a backup did not come back to business after 9/11 incident in the Twin Tower disaster in the USA. This is why we need to backup for protection and generally, we use the progressive incremental backup. Generally logs are been used to have a long-term storage and there is a need to have storage efficiency by using compression algorithms like log specific and tradition lossless compression algorithms, the main drawback is that it ignores the semantics of logs so there is no satisfactory compression ratio mainly in the well-structured composition and the similarity between the both. To make better compression ratio of the cloud backup system this paper explains about multi level log compression method. In order to prevent the data being lost due to natural calamities and also few network problem issues, we do remote-backup. In the present society many of the large data centers generally use remote backups for the improvement of system function, it includes heavy communication overhead and potential errors because of the more distance and costly network transmission. To solve this problem we invented a remote communication service called Neptune. It is a cost-effective time-saver method, it transfers huge data even to the far distance data centers. The time saver method in Neptune is deleting the mistakes making a group for similar files. To remove the space between them it uses chunk level duplication and delta compression, the chunk level duplication eliminate wrong files where as the delta compression makes a group of similar files to control the overheads and difficulty, it uses hashing for grouping same files and time-saver method is for quick remote recovery and examine the Neptune by making use of the present world applications linked HP, GOOGLE, MSN when in comparison with state of the art work.**

Index Terms**-Multilevel log compression, communication overhead, chunk level duplication.**

I.INTRODUCTION

In our day to day life lots and lots of data is been generated, so all this massive data need to be stored. Generally in an IT infrastructure we have servers like data base server, application server, print server, internet server and transaction server etc, for various needs we have various kinds of server and there is a networking between all these servers, access point, switches, routers and the PC's. So when there is lots of information that is been generated the number of servers for storage is being increasing there by the cost in purchase of all these is been increasing rapidly and there is a need for more space to place all these and there is more networking involved, all these makes complexity and it is very much difficult to manage. To resolve this problem we adopt to the concept of cloud computing (cloud computing- It provides IT resources as the service to the user and charge as per their usage).

It is basically classified into three types

Public cloud- It enables resources and services available to every one.

Private cloud- It enables system and services available limited to the organization.

Hybrid cloud- It is a mixture of private and public cloud where all the critical information will maintained by the private cloud and all the non-critical information will be maintained by the public cloud.

It provides us three kinds of services

IAAS: It provides all the infrastructural components as a service to the end user such as compute time such as memory and CPU, network bandwidth and storage.

PAAS: It provides the platform to end user to run and deploy the applications.

SAAS: Through internet the customers can make use of applications that are been hosted by the cloud service provider.

When we implement a public cloud there is lack of security and there is a loss of control over the data center so we implement private cloud in an organization in order to have a proper security and proper control over the data center[1][2]. This paper tells us how to establish a private cloud.

When we establish a private cloud there should be a proper backup (backup- copying the data for limiting the loss in case of a corruption or failure). Most of the companies who did not have a backup did not come back to business after 9/11 incident in the Twin Tower disaster in the USA, this is why it is mandatory to have a proper backup, we use the progressive incremental backup. Generally logs are been used to have a long term storage [3]and there is a need to have storage

efficiency by using compression algorithms like log specific [4][5] and tradition lossless compression algorithms [6], the main drawback is that it ignores the semantics of logs so there is no satisfactory compression ratio mainly in the well structured composition and the similarity between the both. It performs by exposing the data redundancy among the log records and there by sorting according to the similarities and then all the similar ones are moved into the bucket, then these log records are been categorised based on the variation in the data compression and there by using traditional compression algorithm to improve the compression ratio[7]. This paper explains us how to increase compression ratio by using multi level log compression.

Normally all the data is been stored in the servers i.e in the data centers, when there is situation like toofan which took place in Chennai in the year 2015 where all the signals where gone, people in other places whose data center is in Chennai where not able to work there by all the work got delayed and in few cases where people prefer to have data centers near rivers and oceans, all these are disaster prone areas due to the occurrence of disaster, it may destroy the entire data center.In order to resolve this problem we prefer to have remote backup though it is costlier when we take into consideration backup time and network band width.To solve such a kind of problems we have to find the redundant data and then decrese the quality of data that is been transmitted there by there is an improvement in the load balance, availability and reliability and load balance. Latler enables effective backup, through put can be achieve but transmitting differences is given by data, this is known as delta. Regarding international data corporation study the data produced and the exact copy is more than 18 ZB in 2011, the digital data produced will exceed 40 zb in up coming few years and also the volume of is to increase more than 60%. According to some statistical reports two third of the digital world has a copy and only the rest is different. In order to prevent from redundant data that is being transferred we will do chunk level duplication and delta compression deduplication. All these separate files into multiple chunks and each chunks is identified independently by fingerprint called hash signature. It basically identifies each chunk by avoiding byte by byte compression and the delta compression compress the similar regions by differences if the both sender and receiver file similar to transmitted file [8]. So we transmit only the difference between 2 files like effective throughput which is used to measure the performance of long distance backup if 10 mb data is transfered in one sec then the throughput is 10 mb/s, if 19mb of them is redundant then the throughput is 1 mb/s, for remote backup chunks are delta compressed compared to the same chunks that already reside over the remote server, so we transfer those across the WAN to destination. [9].

A. Comprehensive filtration:

General this kind of filtration is done to the remote backup's source and the destination, Neptune deletes the wrong files and the grouped similar files from the source, it is mixed with application level similarity detection and system level deduplication, low level chunks cannot show their properties, Neptune delivers the best performance.

B. Cost effective remote backup:

Neptune reduces computation and space overhead, it uses semantic aware groups by Locality sensitive hashing .Its complexity is O (1). The top-S appropriate delta compression can find more chunks that to be reduced. This shortcut scheme of delta chains allows at least two files from version chain. It satisfies fast recovery[10].

C. Prototype implementation and real world evaluation:

It is used to calculate the fingerprints and there is a storage containers in which all features are stored. In this we use 7 KB and 4.3 MB containers to store chunks, fingerprints and features, we test this including ENDRE[11], CBD[12], SIDC[13].

DESIGN OF A PRIVATE CLOUD USING OPEN STACK

Open stack

- It is a platform to establish public cloud and private cloud by using software tools.
- Free and open- source software
- It provides IAAS
- Create and manage a cloud computing service similar to Amazon web services or Rackspace cloud.

Things that are required in advance

HARDWARE

- The machine should have 6 GB of RAM.
- The processor should have an extension of hardware virtualization.
- Their should be minimum one network adapter.

SOFTWARE

Their should be minimal installation of RHEL(Red Hat Enterprise Linux) or similar versions of the RHEL based Linux distributions like Scientific Linux, CentOS etc. At present only x86_64 is the architecture supported.

NETWORK

In order to establish an external access to the server and instances the network setting should be configured.
The network manager should be disabled and there should be static address IP address for the network card.

- $ sudo systemctl disable firewalld
- $ sudo systemctl stop firewalld
- $sudo systemctl disable NetworkManager
- $ sudo systemctl stop NetworkManager
- $ sudo systemctl enable network
- $ sudo systemctl start network

STEPS INVOLVED IN ESTABLISHING PRIVATE CLOUD

- Install the server with Cent OS with minimal installation.
- Login as the root after the installation is successfully completed.
- Check the status of selinux and the firewall settings.
  # cat /etc/sysconfig/selinux
- If they are active then disable selinux and the firewall settings.
  # cat /etc/selinux/config

- In Cent OS the extras repository provides the Red Hat Package Manager, it enables the Open Stack repository. So by installing the RPM we can set up the Open Stack repository.
  $ sudo yum install –y centos-release-openstack-newton
- Update the packages
  $sudo yum update –y
- In this step, install Pack stack Installer
  $ sudo yum install –y openstack-packtrack
- Pack stack enables single node open stack deployment by using the command
  $ sudo packstack –allinone
- If we had run the Pack stack in advance in the home directory there will be files in.txt format and if want to use that files again we have to choose --answer-file option.
- All the host nodes that are installing on the network should be given a root password and to enable the remote configuration of the host puppet is used as it can remotely configure each node.

BACKUP

We generally do progressive incremental backup, in this method after a full backup there is also a subsequent incremental backup are all collapsed into a full version. In this activity, data is not used from the host as the "full backup" is created from the available incremental copies.
This method is optimized by efficiency using the data movement and avoiding a full copy. It is suitable for backing up file system.

We generally do backup by
- Network based backup
- Network free backup

NETWORK BASED BACKUP

- In this kind of backup the movement of the data is by TCP/IP.
- This process basically involves backup from the internal disk drive of a server to a tape drive connected to a backup server.
- All the servers are connected and the host reads all the data and all these data is been sent to the backup server through a backup server via network.
- The backup server writes the data to the data connected to the tape connected to it.

NETWORK FREE (SAN) BACKUP

- In this we do not use any kind of network.
- In the first step the initiator initiates read and write using fiber channel host bus adopter (HBA).
- Switch acts as an interface between the disk or tape and the initiator that provides the ability to switch and connect various devices.
- If there are three servers namely 1 2 3 then the server 3 has the external disk, server 1 and server 2 has SAN.
- If the data of server 3 needs to be backed up then the data would be read from the internal drive and send to the tape connected to the SAN through filter channel HBA.
- If the data of server 1 and server 2 need to be backed up then the data that would be read from the SAN drive and sent to the tape connected on the SAN through HBA that connect the disk drive.

During backup there are few associated topics which are very much useful such as compression, data deduplication and archive.

COMPRESSION

It is a process of reducing the space occupied by a set of data and Lempel-Ziv algorithm is used in data protection application and hardware compression in tape and disk drives. The data type decides the ratio of compression.

DATA DEDUPLICATION

It is a process used to delete the repeating data and it is very much important in reducing the size while sending.

ARCHIVE

It is the long-term storage of data.

Multi level log compression
It involves 3 steps
- Bucketing preprocessor module
- Delta compression
- Decompression

BUCKETING PREPROCESSOR MODULE

It is a process in which all the similarities in the log entries are to be taken into buckets. This process is done by similarity indicator i.e Jaccard distance.
It involves basically 3 steps
- Content defined chunking
- Jaccard distance
- Load balancing

CONTEST DEFINED CHUNKING

In this process it divides the data into separate variables size chunk of approximate length. It generally happens in the process deduplication.
The actual string is broken into a segment set (f6d138, 9d112xy3, 7fg3313k, 4f9d3f86).

f6	d1	38	9d	11	2x	y3	7f	g3	31	3k	4f	9d	3f	86

JACCARD DISTANCE

It is a similarity indicator.
If A and B are the two sets then
Jaccard index= $|x \cap y| / |x \cup y|$ then
Jaccard distance=1-Jaccard index.

LOAD BALANCING

It pushes the most similar entries into the bucket and if the bucket is full then it pushes into another bucket.
Steps involved in this process
Step1: Check weather the number of entries is equal to 0 or not.
Step2: If is not equal to 0 then push all the similar entries into the bucket.

Step3: Check weather the bucket is full or not

- If the bucket is full then push the similar buckets into another bucket.
- If the bucket is not full then push the similar entries into the same bucket.

DELTA COMPRESSION

f6	d1	38	9d	11	2x	y3	7f	g3	31	3k	4f	9d	3f	86

f6	d1	38	9d	11	2x	y3	6d	g3	31	3k	4f	9d	3f	86

It compares the both and finds the segments that are same and the segments that are different, in the above example A(0, 6)B(7f)A(7, 7) and A(0, 6)B(6d)A(7, 7) there is only change in 7f to 6d rest all are same, it copies 6 bytes from position 0 and copies 7 bytes from position 7.

DECOMPRESSION

It is opposite to compression and it involves 2 steps such as general decompression and delta compression. It firsts decoded by using decompression algorithms and the delta decompression decompresses the decoding.

DESIGN OF NEPTUNE

A file is transmitted from sender to receiver, both sender and receiver consists of reference and similar files. The difference between these two files are transmitted as if to reduce the overhead which is communicated between them through the network service we can reduce the link loads and the capacity is increased, therefore it increases the number of bandwidth intensive applications. To enterprise cloud the existing single point redundancy elimination solution is difficult to use, Neptune can reduce the network wide redundancy elimination.

SIMILAR MEASURES

It checks the approximation of two files like modifications, error corrections etc.it is resemblance between 0 and 1,when the value is one. Both the files are alike. More similarity represents the duplication.

Definition: The resemblance r(P,Q) of two files P and Q is characterize as r(P,Q)= $| X_P \cap X_Q | / | X_P \cup X_Q |$ are finger prints sets of P and Q, hashing function is chosen uniformly and randomly.

We use fixed files sketch for each file to represent similarity, Rabin fingerprints are used to figure out finger print in the sketch. It is based on polynomial arithmatics and constructed in any length. The low collision we choose length of fingerprint, 64 bit Rabin fingerprints are suitable for all applications.

SIMILARITY DETECTION

In delta compress chunks if any similar chunks are found, in which has already transmitted the similarities can identify the features of chunk. By using rolling hash function match the small portions of the data to calculate features in an efficient way. Maximal hash value is used as a feature multiple hash function.

To produce independent multiple features we compare fingerprints against a mask by using Rabin fingerprints over rolling windows of chunk. Super feature is build together with multiple features underlying feature values are represented by super feature. If two super feature has same value consecutive feature points the beginning and ending positions quality of matches is increased by increasing the super feature features, super feature is used as a query request.

PRACTICAL OPERATION AND WORKFLOW

By the data filtration with lower bandwidth source send the information to the target in Neptune remote back up, data filtration has deduplication and approximate data compression. To deliver high performance to the destination it reduces the space overhead and core related files into groups. Source is to encode and destination to decode. Base chunk is the main components in the data filtration for each chunk calculate each hash value to represent the fingerprint, in index if fingerprint does not found then it is considered as a new and it is stored. The metadata maintains the reference to the before chunk. If chunk requests from the starting point of transmission its fingerprint is related with the cache at the last point. Blue bloom filter is used to check whether the fingerprint is used on-this index. Corresponding container fingerprints are loaded into cache, if eviction occurs on the basis of LRU policy, all the fingerprints from the container are expelled. The resemblance search find the approximate top-S fingerprints that are same to the query types. For encoding we use Xdelta to optimize the reduction of same data portions, initialize encoding.

DATA PENETRATE

It contains chunk level deduplication as well as appropriate data compression, deduplication is the main key point or component in backup and archive systems. It breaks the data strain to filed chunks and also changes or replaces the duplicate chunks with pointers into a previously stored copy, it identify chunk by fingerprint. Neptune aggregate chunks into containers for backup and it transfers only the differences between two files. It access target and reference files.

DELTA CHUNK:

Transmitting the files to and from a server is directly proportional to the amount of data to be sent for a backup and restore systems in the time overhead.

It has two types

i. Linear delta chains

ii. Short cut delta chains

Linear delta chains

A linear sequence of versions construct the whole matter of modification to a files, this line of arrangement is called as version chain. It can frequently access to the previous i.e old data and it may occur more transmission delay, linear version chain is used by conventional scheme.

Denote an understanding compress ith version of the file by Vi,the main differences between two versions A¡ and Aj is delta (Ai,Aj),Aj is again restored by the inverse differencing method of operation Ai and delta.(Ai,Aj) is the differencing

operation which is represented as (Ai,Aj)->delta (Ai,Aj)inverse differentiating operation as (Δ (Ai,Aj) , Ai)->Aj,versions Ai, Aj are nearby if j-i=1, construct Ai via Aj - 1.
The linear sequence of versions A1,A2,...,Ai−1,Ai,Ai+1,.....
It uses series of delta to store there versions if two adjacent versions Ai & Ai +1 preserves the difference between two files Δ (Ai), Ai +1. It leads to a delta chain as v1, Δ (A1, A2), _ _ , Δ (Ai-1, Aj), Δ (Ai, Ai+1)_ _ _ .Execute inverse differencing algorithm on the intermediate function.

*Shortcut Delta Chains*It is to develop and give back the fulfillment and decreases operational delay, it is used for multi version delta chain, it has changed forward delta and simultaneously chooses files.It access mostly two files from version chain when user computes the delta compression, the files that has to be sent to the server to be preserved $|\Delta(A_i,A_j)|$ denotes the size of $\Delta(A_i, A_j)$ if j-1 increases then the size increases and there is a decrease of potential of the compression quality.Two nearby versions A_i andA_i +1 have modified fraction $\gamma|A_i|$ between them. Alfa represent composition quality between versions. Neptune uses differencing algorithm to construct delta file, $\Delta(A_i, A_{i+1})$ and the version compression is A_i=1-$|\Delta(A_i, A_{i+1})| / |A_{i+1}|$.The result demonstrate the relative compressibility of all new versions that have similar size deltas, for worst case compression the size of delta file is γ|vi| and the new version has variable size from |vi| to (1+ γ)$|A_i|$. The $\gamma|A_i|$ replaces the existing fractions in A_i i.e $|A_i|=|A_{i+1}|$ between versions A_{i+1} and A_{i+2} we obtain $\gamma|A_{i+1}|$ modified fractions. Hence between A_i andA_{i+2} the union bound on number of modified fractions we obtain $2\gamma|A_i|$ changed fractions.

Delta compression and recovery
This method is utilizedin networking and storage systems so that it can increase the efficiency of data transition and it can slow down or reduce the space requirements and the last data encoding can control the data redundancy, it has two inputs which is target file which is to be compressed and an reference file. The encoding deltas compresses difference between target and source files, source file generates exact copy of target file, in this we consist of two files B_{new} , B_{old} fold, the server and the client are connected by a communication link. The client has the duplication of B_{new} and server has a duplication of fold, the server can reconstruct B_{new} from fold and BΔ. By computing the hash value at rolling window position we can process the target chunk from a file. The hash value into the index can find the pair to the base chunk, target bytes are kept in the output buffer if the bytes failed to pair then the duplicate files are removed by the Neptune and the deleted files can be backup by using this process which as substantial resources for cost effective and efficient recovery solution. The base file has well recognized clustering algorithms for delta decoding in delta recovery.

We represent the implementation details including co-relation - aware grouping, in this co-relation aware grouping we have LSH i.e locality sensitive hashing to find same files in the same hash bucket to narrow the scope of processing data and the mistakes with more probability we define S as domain of the files we consider H as family of hash functions. We consider getting from H normally, let us consider two points a and b the their probability is g (a)=g (b).
Definition: Definition : LSH function family , is called (X,cX,Y_1,Y_2) sensitive for distance function ‖ # ‖ if for any a,b ϵ S
• If ||a,b|| ≤ X then Y_H f(a) = f(b)] ≥ Y_1,
• If ||a,b|| > cX then Y_H [f(a) = f(b)] ≤ Y_2.
For the similarity check we use c > 1 and Y1 > Y2. By the process of multiple hash function we have to provide more space between p1 and p2, each hash function Ga,b(v): Rd → Z to connect a dimensionality vector with the set of integers is defined as Ga,b(v) = [(a+v+b)/ω].

II.EVALUATING PERFORMANCE

Neptune based on multiple performance metrics by experimental results.

A. Experiment
We can implement Neptune by using many servers, each server with same configuration, all the code is been written in C and in a Linux environment, we generate patterns according to the datasets.
There are 4 types of datasets

- LANL

It affords numerous datasets, it consists of unique ID, file size, block size, creation time, modification time and the path of the file[14].

- HP file system trace

Read, write, lookup, open and close operation on the accessed files with file names and device numbers are performed by datasets[15].

- MSN trace

All the metadata is been divided into periods and the datasets is divided in the form of intervals and all these provides read and write operation with specific ID and size[16].

- GOOGLE clusters

Log data from the clusters are performed by google.it contains set of tasks[17].

We allocate the data segment using Round Robin and the client uses the Neptune design is for deduplication and approximate delta compression To exchange messages and data via TCP/IP multiple threads are used by both clients and servers. It supports top-S approximate compression and deliver high system performance. The number of transmitted fingerprints from destination to source servers is identified by S to find same fingerprints., the more base fingerprints can be found as the value of S is larger, which unfortunately incurs the longer latency due to computation-intensive indexing. The used metric is the normalized rate against different k values count the band width and compute their normalized values between the minimum and maximum values. In same way, we calculate the normalized latency values of executing top-S indexing.

A. Analysis
1. Deduplication performance
a. Deduplication Throughput
The Neptune improves the amount of data that is been transmitted in compared with the EndRE and CBD.

Neptune can transfer based on the grouping which is limited to the scope of processing and there is an increase in the overhead due which ultimately there is an increase in the deduplication throughput.

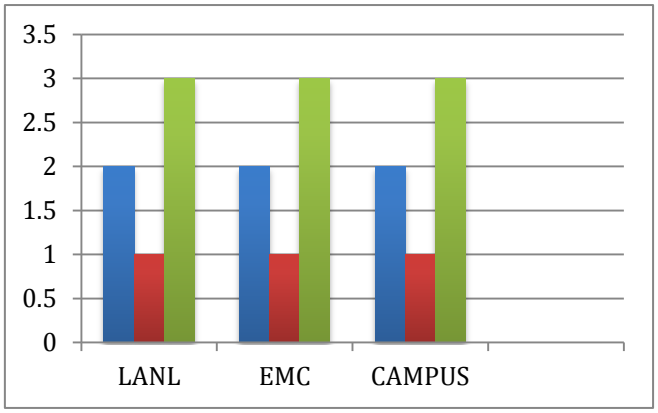

Figure 1 - Deduplication throughput

b. Time Overhead
Neptune enables us to obtain exactly similar duplicates in a very small amount of time.

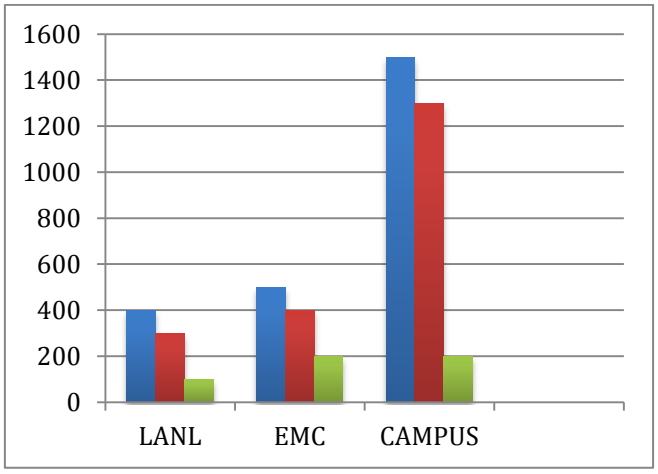

Figure 2 - Time overhead

c. Space Overhead:
Neptune helps us in saving space occupied by in the RAM in compared to the rest.

	EndRE	CBD	Neptune
MSN	1	0.65	0.17
LANCL	1	0.80	0.15
GOOGLE	1	0.50	0.10
HP	1	0.85	0.13

Table 1

From the table 1 we can easily conclude that Neptune occupies less space.

2. Delta Compression In Remote Backups:
a. Effective Network Throughput
Neptune transfers small amount of data over the network by using specific algorithms such as delta compression algorithms.

b. Multi-level Delta Compression
The amount of data that is been sent is reduced by the Neptune.

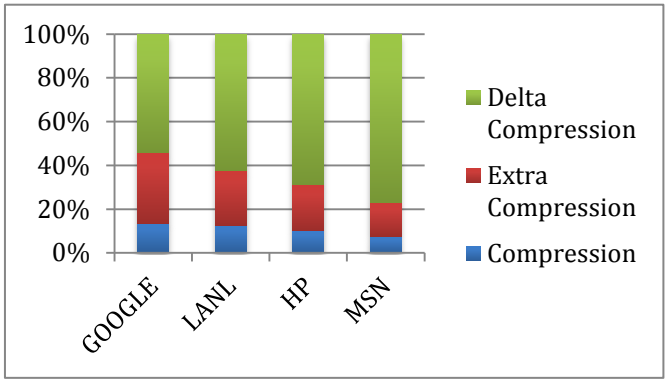

Figure 3 - Multi-level Delta Compression

From the above graph we can clearly tell that there is compression of data by using Neptune.

c. Delta Computation Overheads:
By using delta compression Neptune increases the throughput which produces disk input/output and extra computation overheads.

d. Recovery Overheads:
Neptune enables to reduce the system overhead in compared with the full backup.

III. CONCLUSION

This paper studied an efficient way to establish a private cloud by using open stack by the reference of, to improve the compression ratio we implement the multi level log compression and to establish a cost efficient remote backup.

REFERENCES

[1] www.rdoproject.org
[2] www.openstack.redhat.com
[3] Adam Oliner and Jon Stearley, *"What supercomputers say: A study of five system logs"*. In 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks 2007, pages 575–584.
[4] KimmoHatonen, Jean Francois Boulicaut, Mika Klemettinen, Markus Miettinen, and Cyrille Masson. *"Comprehensive log compression with frequent patterns InData Warehousing and Knowledge Discovery"*, pages 360–370. Springer, 2003.
[5] Hao Lin, Jingyu Zhou, Bin Yao, MinyiGuo, and Jie LiCowic, *"A column-wise independent compression for log stream analysis"*. In 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pages 21–30.
[6] Jon Stearley, *"informatic analysis of syslogs"*, In 2004 IEEE International Conference on Cluster Computing, pages 309–318.
[7] Bo Feng, Chentao Wu, Jie Li, *"MLC: An Efficient Multi-level Log Compression Method for Cloud Backup Systems"*, 2016 IEEE Trustcom/BigDataSE/ISPA.
[8] Shilane, Phlip, Mark Huang, Grant Wallace, and Windsor Hsu, *"WAN-optimized replication of backup datasets using stream-informed delta compression"*, ACM Transactions on Storage, 2012.
[9] Xia, Wen, Hong Jiang, Dan Feng, and Lei Tian, *"DARE: A Deduplication-Aware Detection and Elimination Scheme for Data Reduction with Low Overheads"*, IEEE Transactions on Computers, 2015.
[10] P. Indyk and R. Motwani, *"Approximate nearest neighbors: towards removing the curse of dimensionality"*, Proc. STOC, pp. 604–613, 1998.
[11] Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese, *"EndRE: an endsystem redundancy elimination service for enterprises"*, Proc. NSDI, 2010.
[12] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, *"Tradeoffs in scalable data routing for deduplication clusters"*, Proc. USENIX FAST, 2011.

[13] P. Shilane, M. Huang, G. Wallace, and W. Hsu, *"WAN Optimized Replication of Backup Datasets Using Stream-Informed Delta Compression",* Proc. USENIX FAST, 2012.
[14] Los Alamos National Lab (LANL) File System Data, http://institute.lanl.gov/data/archive-data/.
[15] E. Riedel, M. Kallahalla, and R. Swaminathan, *"A framework for evaluating storage system security",*Proc. FAST, pp. 15–30, 2002.
[16] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda, *"Characterization of storage workload traces from production Windows servers",*Proc. IEEE International Symposium on Workload Characterization (IISWC), 2008.
[17] J. L. Hellerstein, *"Google Cluster Data",*http://googleresearch.blogspot.com/2010/01/google-cluster-data.html, Jan.2010.