ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Performance Analysis of Distributed Storage Cluster using Network Coding

Priyanka Ramesh Nanaware [1], Shreya Bokare[2], Zia Saquib[3]

P.G. Student, Department of Electronics Engineering, PIIT New Panvel, Mumbai, India[1]

Technical Officer, Department of CNIE, CDAC Mumbai, India[2]

Executive Director, CDAC Mumbai, India[3]

ABSTRACT: Tremendous growth in the data generation using applications like networks, file, and video sharing is demanding for the data availability and reliability. Distributed storage clusters performs an important role in this case by providing data redundancy over network storage node. This paper presents the performance analysis of ceph cluster: one of the distributed storage cluster freely available in the storage technology, by using different network coding techniques. Various network coding techniques and their significance in data distribution and regeneration during node failure are discussed in this paper. Here we used network codes such as ceph default crush algorithm, Erasure code, Regenerating code and Self-repairing code to evaluate the performance of ceph storage cluster. The experimental setup and procedure flow graph presents the insights of the testing carried over ceph cluster. Finally result analysis shows the comparison of different network codes.

KEYWORDS:Ceph cluster; Distributed Storage cluster; Network Coding; Performance Parameters

I. INTRODUCTION

With growth in variety of applications like networks, file and video sharing in recent years, the demand for large-scale data storage has increased significantly with the requirement of seamless secured storage access. Single node deployment may become unreliable in case of node failure. The redundancy in the storage node is required to improve the reliability. The simplest solution to improve the reliability is to use node replication by using RAID architecture [1]. However, RAID comes with its own set of issues, like increased in the amount of storage and bandwidth used, performance penalties. The industry has worked to overcome these issues by developing new techniques like distributed storage cluster using network coding [2]. For distributed storage, the idea of using network coding was introduced by, scenario.A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes" is mentioned in Proc. IEEE/ACM Int. Symposium on Information Processing in Sensor Networks (IPSN), April 2005. The distributed storage cluster couples data distribution with replication of the dataset, which improves the data regeneration rate. In distributed storage cluster the data is distributed and stored onto multiple disks in the form of blocks or objects. In case of block data storage, file data is split in the form of block i.e chunk of bytes of fixed size and stored on disk using SCSI or related protocol [3]. However object data storage, file is treated as an object i.e actually divided into data, metadata and unique identifier which have flexibility on block size during data storage [4].
Ceph cluster is one of the distributed cluster, which can store data in block or object format [5][6]. One can select type of the data storage (block or object) during configuration of Ceph cluster. By default Ceph cluster uses CrushMap algorithm [7] to distribute and store data across multiple disks. One can also use other network codes to distribute data across multiple disks attached to Ceph cluster [8]. In this paper we have compared various network coding algorithms to distribute and store data in Ceph cluster. We mainly focused on comparison of various parameters such as Bandwidth Consumption, Throughput, Latency and Recovery Bandwidth using multiple network coding algorithms on Ceph cluster. The paper is organized as follows. Section 2 provides details about working and internals of network coding algorithm. In section 3, we explained the working components of Ceph cluster, Section 4 explains the

experimental setup prepared in our lab for testing. The testing procedure is explained in section 5, testing results and analysis is provided in section 6, finally we conclude our paper in section 7.

II. NETWORK CODING ALGORITHMS

In the conventional routing, each intermediate node in the network simply stores and forwards the information received. Network coding is a generalization of the conventional routing (store-and-forwarding) method. The network coding allows the intermediate nodes to generate output data by encoding (i.e., computing certain functions) previously received input data. Fundamental idea of Network Coding is combining of packets instead of just routing which allows information to be “mixed” at intermediate nodes [9]. Thus network coding techniques can be used to improve a network's throughput, efficiency and scalability along with resilience to attacks and overhearing [10]. In general, network codes stores a file of size M bytes in the form of n fragments each of size M/k bytes; any k of which can be used to reconstruct the original file. There are different types of network coding namely Replication code, Erasure Code, Regenerating Code, Self-repairing code currently used in the storage networking etc.

Replication Code:
In replication code, as the name indicates it will stores the replicated copy of original data into another disk. In case of original disk failure, data recovery is possible with another disk which has the same copy of the original data. However due to the data replication, the bandwidth required is more. e.g bandwidth required to store Analysis.txt file using replication code is twice as of the normal case.

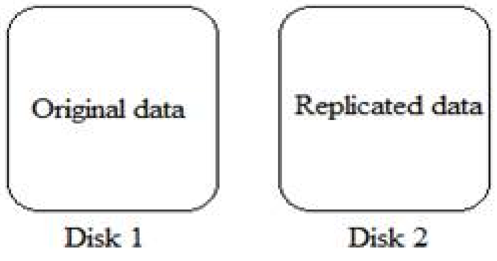


Fig. 1 Replication code strategy

Erasure Code:
Erasure code is a network code where original data k is divided into p fragments to be stored on n number of network nodes, where k+p=n. Each node stores p/k data fragments to n number of nodes. During any one of the node failure, all the remaining nodes are required to regenerate the data[11]. In case of node failure, erasure code needs to recreate first the original data and then it will restore the failure node data in new node. During data recovery needs to recreate the original data from all the remaining nodes, therefore repair bandwidth and latency required is more. Thus, Erasure code does not support the high density area (heavy load). Consider there are 6 nodes, node 1 to node 6 where the data is distributed using erasure code distribution.Node2 is failed whose data needs to be regenerated and restored on new spare node say node 7. In this case, node 2 data will be reconstructed from all the remaining nodes i.e. from node 1, node 3, node 4, node 5 and node 6. And then restoration of data on spare node 7 takes place. Erasure code is more storage efficient than replication code as it does store the entire data copy replicated onto another disk. However in erasure coding at the recovery needs to recreate original data by connecting with all nodes except failed node and also multiple concurrent node failure recovery is not possible using erasure code.

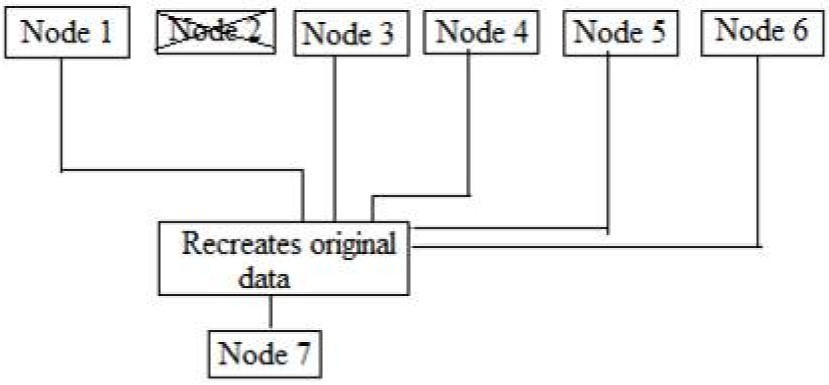


Fig. 2 Erasure code data recovery

Regenerating Code:
Regenerating code is the variation of erasure code. Original data k is divided into M fragments which are stored in n nodes. Each node stores M/k data fragments and less than n-1 nodes are required to recreate the original data [11][12]. The advantage of regenerating code over erasure code is there is no need to recreate the original data first by connecting with all remaining nodes. Therefor repair bandwidth and latency required is less than erasure code. Regenerating code is a priority based code thus all nodes are not treated equally. In case of data recovery the number of nodes required for recovery are depend on which specific node is missing. Consider 6 nodes, node 1 to node 6, and node 2 & node 4 are failed. One may require different number of nodes to recover the failed node data which eventually depends on the priority and data distribution pattern. Thus repair bandwidth and latency required to regenerate the data will not be same for all nodes. Regenerating code can handle concurrent multiple failures, but data regeneration takes place sequentially.

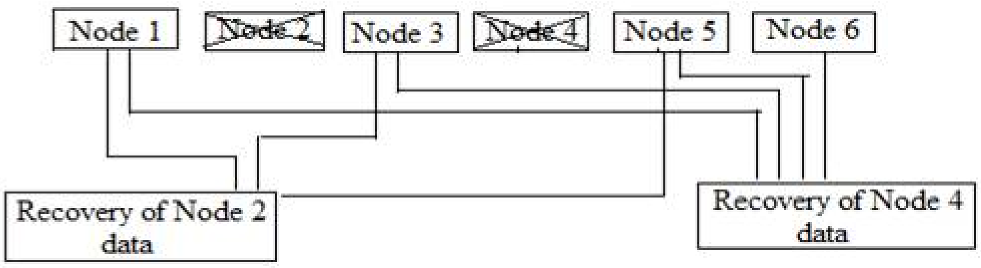


Fig. 3 Regenerating code data recovery

Self-repairing Code:
Self-repairing code is the variation of erasure code and regenerating code. It is not a priority based code therefore all nodes are treated equally. Here the number of nodes required for recovery are fixed number and does not depend on which specific node which is missing. Therefor the repair bandwidth and latency required is very less than erasure code and regenerating code. As shown in fig. 4, consider there are 6 nodes, node 1 to node 6, and node2 & node 4 are failed which needs to be recovered. Now there is need to recover the failed node data. In self-repairing code to recover the node 2 and node 4 data it requires 4 nodes. As mentioned previously it is not a priority based therefor all nodes treated equally and the number of nodes required for recovery is depend on how many nodes are failed and not depend on which specific node is missing [12].The main advantage of self-repairing code is the data recovery of multiple nodes can be done in parallel.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

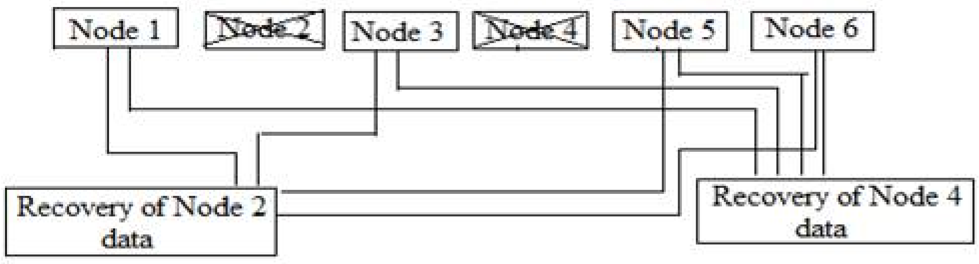


Fig. 4 Self-repairing code data recovery

Table 1 Parameter Comparison of network codes

Parameters	**Erasure code**	**Regenerating code**	**Self-repairing code**
Restore node	Brick node	Provider node	New comer node
Recovery	At the time of recovery needs to recreate original data	No need to recreate original data	No need to recreate original data
Recovery nodes	Independent on how many nodes are failed and which specific node is failed	Dependent on which specific node is failed	Dependent on how many nodes are failed
Number of nodes for recovery	All excepting failed node	Not fixed number of nodes	Fixed number of nodes
Density Area	Supports only low density area. Does not support high density area	Supports high density area	Supports high density area
Packet delivery Ratio	Very low. Because does not support high density area	Moderate.	High.
Latency	Higher. Because at the time of recovery it needs to recreate the original data first	Low	Very low
Bandwidth consumption	High	Low	Very low
Recovery Speed	Low	Moderate	High
Throughput	Low	Moderate	High

III. DISTRIBUTED STORAGE CLUSTER USING CEPH

In distributed storage computing, Ceph is a free software storage platform that stores data on a single distributed computer cluster, and provides interfaces for object, block and file level storage [13][14][15]. In this paper, we have used Ceph distributed storage cluster and carried out performance evaluation of different network codes on it. Distributed storage cluster maintains data redundancy in such a way that even if one or more disks get failed one may be able to recover the lost data using data redundancy on other remaining disks. For our experimental analysis data is distributed in the form of objects in ceph storage cluster. Fig. 5 shows the important working components of ceph cluster.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

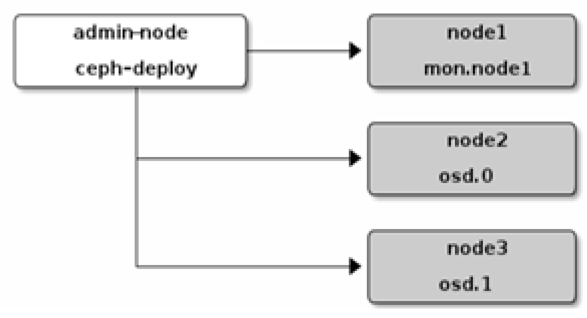


Fig. 5 Working components of Ceph cluster

There are three main components of ceph cluster:

1. Admin node: Original data is stored or put on an admin node. Admin node is responsible for the way how data should be distributed and also it is responsible for manual process of Ceph installation and Ceph deployment on remaining nodes.
2. Monitor node: It is responsible to keep track of active and failed cluster nodes. Mainly monitor node is responsible for detecting the failure node.
3. Object Storage Devices (OSD): This store the distributed content of files. Ideally, OSDs stores the data which is distributed by the admin node of the ceph cluster [16][17][18].

By default ceph cluster uses crush map algorithm for data distribution and recovery. The details of crush map algorithm are mentioned here:

Crush map Algorithm:

In ceph cluster the CRUSH (Controlled, Scalable and Decentralized Placement of Replicated Data) algorithm determines how to store and retrieve data on/from data storage locations by performing computation on it. CRUSH allows Ceph admin to communicate with OSDs. CRUSH requires a map of cluster, and uses the CRUSH map to store and retrieve data in OSDs with a distribution of data across the cluster. CRUSH maps algorithm uses rules to determine placement of data for a pool, where pools are the logical groups for storing data objects in ceph cluster. CRUSH rules defines placement and replication strategies or distribution policies that allow to specify exactly how CRUSH places object replicas.

IV. EXPERIMENTAL SETUP

As mentioned in section 3, in this paper we used ceph cluster for carrying out network codes experiments on it. We have used 4 virtual machines and deployed ceph components on it to form a ceph storage cluster. The configuration of the virtual machines are mentioned in table 2.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

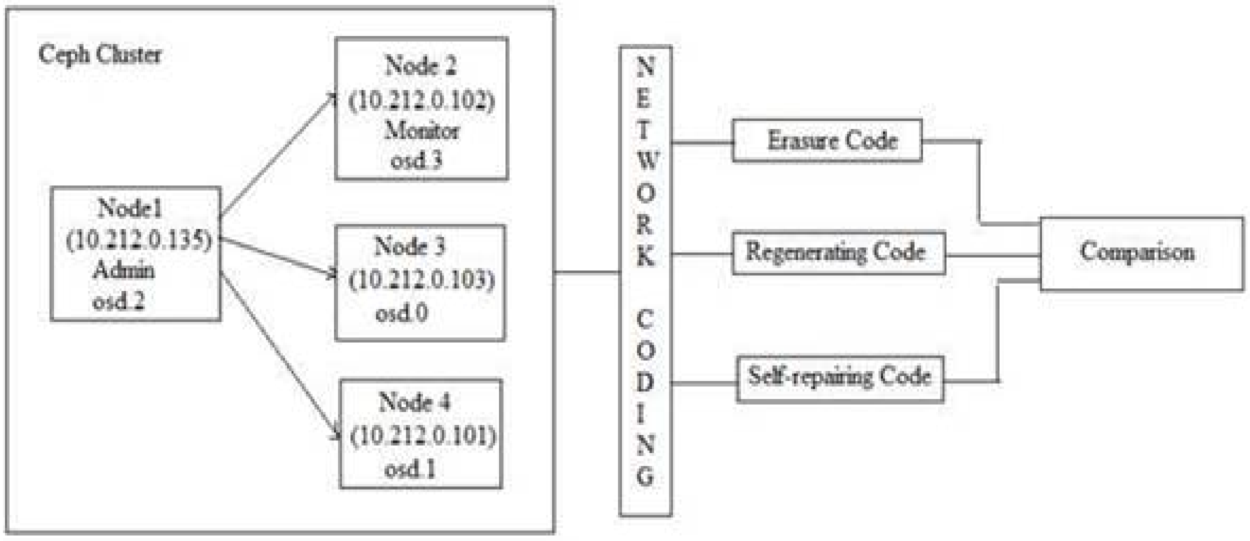


Fig. 6 Experimental Setup

Table 2. Ceph cluster component details

Node Number	IP Address	Ceph Component	OS	RAM	storage volume
Node 1	10.X.X.135	Admin node and Object storage device (osd.2)	Centos 6.5	2 GB	40 GB
Node 2	10.X.X.102	Monitor node and object storage device (osd.3)	Centos 6.5	2 GB	40 GB
Node 3	10.X.X.103	Object storage device (osd.0)	Centos 6.5	2 GB	40 GB
Node 4	10.X.X.101	Object storage device (osd.1)	Centos 6.5	2 GB	40 GB

As shown in fig. 6, in the experimental setup we used four object storage devices (OSD) to store the distributed data. To form a ceph storage cluster, the deployment is initiated from node 1 having ip address 10.X.X.135 on which admin service is deployed. On node 2 Monitor component is deployed from admin node i.e node 1. On each node, the OSD directory structure is created at a particular location to use these nodes as data storage device. In this way ceph cluster is ready with admin, monitor and four osd's for distributed data storage.

V. TESTING PROCEDURE

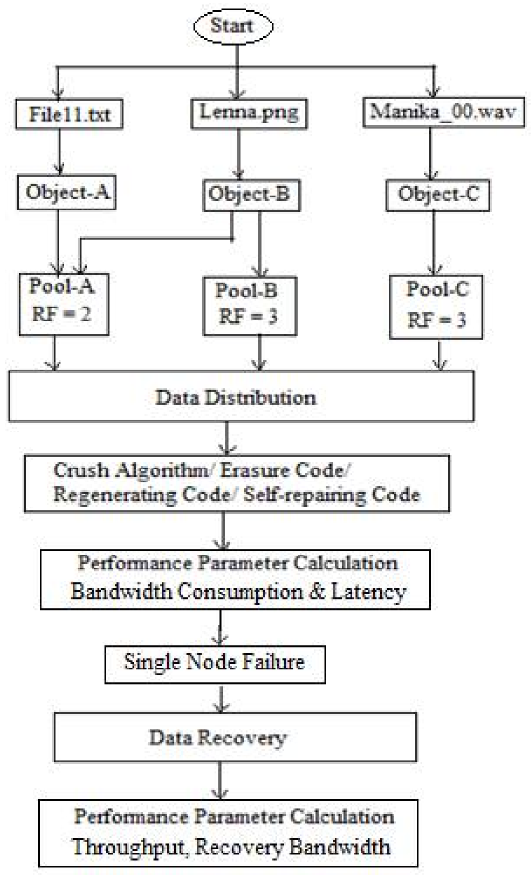
We consider three type of data format for distribution in ceph cluster viz. text (File11.txt), image (Lenna.png) and audio (Manika_00.wav) of sizes 12k, 464k, 144k respectively. We want to distribute data in object format in ceph cluster, therefore we created 3 objects of the data namely Object-A, Object-B and Object-C. File11.txt, Lenna.png, Manika_00.wav is placed in Object-A, Object-B and Object-C respectively. One need to create logical data groups, called as "pools" to distribute data in ceph cluster. In this case we created 3 pools namely Pool-A, Pool-B and Pool-C with replication factor 2, 3, 3 respectively. Pool-A contains Object-A and Object-B, Pool-B contains Object-B and Pool-C contains Object-C. The performance parameters (Bandwidth consumption, Latency, Throughput and Recovery bandwidth) of distributed storage cluster are evaluated by carrying out data distribution and recovery using crush algorithm, Erasure code, Regenerating code, Self-repairing code. The above mentioned codes are available in open source python format. These format files are used from admin node to initiate data distribution in respective OSD's and data recovery after node failure. The process flow is explained in fig. 7

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016


Start
File11.txt
Lenna.png
Manika_00.wav
Object-A
Object-B
Object-C
Pool-A RF = 2
Pool-B RF = 3
Pool-C RF = 3
Data Distribution
Crush Algorithm/ Erasure Code/ Regenerating Code/ Self-repairing Code
Performance Parameter Calculation Bandwidth Consumption & Latency
Single Node Failure
Data Recovery
Performance Parameter Calculation Throughput, Recovery Bandwidth
Fig. 7 Test procedure of data distribution and recovery

VI. RESULTS AND ANALYSIS

In this paper, we listed four performance parameters of distributed storage cluster and evaluated them using above discussed network coding techniques. The detailed structure of object data distribution pattern is presented in table 3. Table 4 provides the details of four parameters evaluated using four different network codes. The analysis of the performance parameters evaluated using four different network codes are depicted in the form of bar graph fig 8-11.

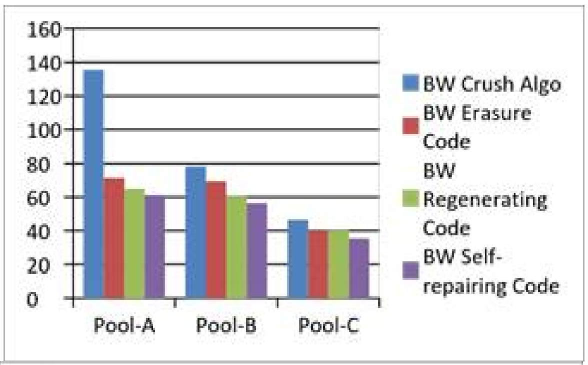
Table 3 Data Distribution Pattern in Ceph Cluster

Parameters		**Data Distribution Pattern**		
Name of pool		Pool-A	Pool-B	Pool-C
Replication Factor		2	3	3
No of object		2	1	1
Name of object		Object-A Object-B	Object-B	Object-C
Data type		Text(.txt)	Image(.png)	Audio(.wav)
File(data) name		File11.txt	Lenna.png	Manika_00.wav
Size of file(data)		12k	464k	144k
Distributed data(respective osd number)	Object-A	1,0	Not Applicable	Not Applicable
	Object-B	3,2	1,0,2	Not Applicable
	Object-C	Not Applicable	Not Applicable	0,2,3

No of placement group		128	128	128
Pool id		8	9	10
Placement group id	Object-A	8.68	Not Applicable	Not Applicable
	Object-B	8.7b	9.7b	Not Applicable
	Object-C	Not Applicable	Not Applicable	10.7f
Failed osd number		Osd.0 (Node 3)	Osd.0 (Node 3)	Osd.0 (Node 3)
Osd number used for recovery		Osd.2 (Node 1)	Osd.3 (Node 2)	Osd.1 (Node 4)

Table 4 Parameter Evaluation

Parameter	**Bandwidth Consumption**			**Latency**			**Recovery Bandwidth**			**Throughput**		
	Pool-A	Pool-B	Pool-C	Pool-A	Pool-B	Pool-C	Pool-A	Pool-B	Pool-C	Pool-A	Pool-B	Pool-C
Crush Algo	135.625 MB/sec	78.178 MB/sec	46.493 MB/sec	0.15 sec	0.16 sec	0.45 sec	48020 KB/s	102322 KB/s	95547 KB/s	37.4 GB/s	36.2 GB/s	38.2 GB/s
Erasure Code	64.957 MB/sec	69.487 MB/sec	40.387 MB/sec	0.11 sec	0.10 sec	0.28 sec	47613 KB/s	77272 KB/s	80900 KB/s	40.0 GB/s	37.2 GB/s	39.8 GB/s
Regenerating Code	71.467 MB/sec	60.411 MB/sec	40.349 MB/sec	0.09 sec	0.09 sec	0.21 sec	32858 KB/s	69140 KB/s	72198 KB/s	40.4 GB/s	39.2 GB/s	40.0 GB/s
Self-repairing Code	61.347 MB/sec	56.411 MB/sec	35.421 MB/sec	0.07 sec	0.07 sec	0.16 sec	28697 KB/s	63208 KB/s	49048 KB/s	41.8 GB/s	39.2 GB/s	40.3 GB/s



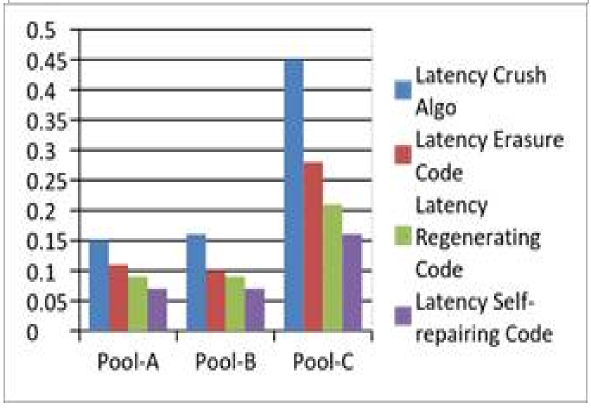


Fig. 8 Comparison of Bandwidth Consumption of Crush Algorithm, Erasure Code, Regenerative code and Self-Repairing code
Fig. 9 Comparison of Latency of Crush Algorithm, Erasure Code, Regenerative code and Self-Repairing code

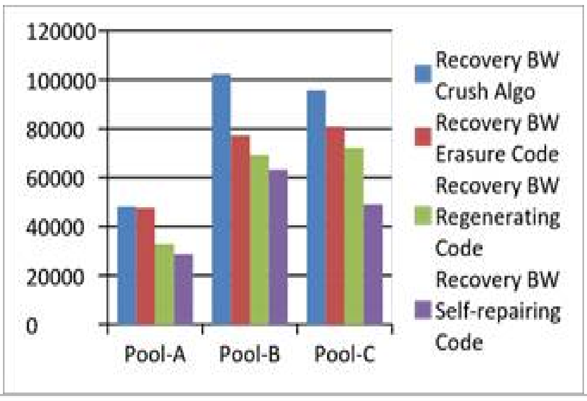
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Fig. 8 and Fig. 9 shows that, Self-repairing code require less bandwidth and less time (latency) for data distribution as compared with other network codes.



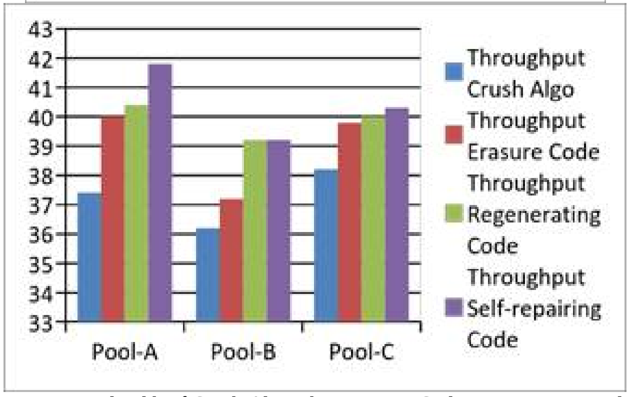


Fig. 10 Comparison of Recovery Bandwidth of Crush Algorithm, Erasure Code, Regenerative code and Self-Repairing code
Fig. 11 Comparison of Throughput of Crush Algorithm, Erasure Code, Regenerative code and Self-Repairing code

Fig. 10 and Fig. 11 shows that, after node failure the repair bandwidth required for node reconstruction using self-repairing code is the least one. We calculated the throughput values for all the network codes, it is observed that self-repairing code has the highest throughput values.

VII. CONCLUSION

In this paper, the performance of distributed storage cluster is evaluated using network coding techniques. We considered four performance parameters during data distribution and data restore to/from the object storage device. Crush Algorithm, erasure code, regenerating code, self-repairing code are the four different coding techniques used for data distribution and data recovery on ceph storage cluster. It was observed that the Self-repairing code is an optimized code for data distribution and data recovery(during node failure) as it provides less distribution bandwidth, low latency, fast recovery and better throughput than the other network codes. The experimental results shows that network coding techniques used instead of default distribution algorithm(crush algorithm), can improve the storage cluster performance.

REFERENCES

1. Sneha M, “Performance Analysis of RAIDs in Storage Area Network”, *International Journal of Computer Applications* (0975 –8887) Volume 126 –No.13, September 2015.
2. Alexandros G. Dimakis, P. Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and KannanRamchandran, “Network Coding for Distributed Storage Systems”, IEEE Transactions on Information Theory, Vol. 56, No. 9, Page no. 4539-4551, September 2010.
3. Rickard Broman,“A practical study of Network Coding in Distributed Storage Systems”, Second cycle Stockholm, Sweden 2013.

4. Data Direct Networks, “A Beginner’s Guide To Next Generation Object Storage”, White paper,2015.
5. Sage A. Weil Scott A. Brandt Ethan L. Miller Darrell D. E. Long, Carlos Maltzahn “Ceph: A Scalable, High-Performance Distributed File System”, OSDI ’06: 7th USENIX Symposium on Operating Systems Design and Implementation, Page no. 307-320.
6. Intel storage solutions reference architecture – fujitsu and ceph published in jan 2015.
7. Sage A. Weil Scott A. Brandt Ethan L. Miller Carlos Maltzahn, “CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data”, IEEE Tampa, Florida, USA, November 2006.
8. Benjamin Depardon, Gael Le Mahec, Cyril Seguin, “*Analysis of Six Distributed File Systems*”, hal-00789086, 15 Feb 2013.
9. Alexandros G. Dimakis, KannanRamchandran,YunnanWu,andChanghoSuh, “A Survey on Network Codes for Distributed Storage”, Proceedings of the IEEE Vol.99,No.3,March2011.
10. Frederique Oggier, AnwitamanDatta, “Self-repairing Homomorphic Codes for Distributed Storage Systems” arXiv:1008.0064v1[cs.DC] 31 Jul 2010.
11. Ming-Zhu Deng, Zhi-Guang Chen, Yi-Mo Du, Nong Xiao, and Fang Liu, “Erasure Codes in Big Data Era”, The 2014 International Conference on Control, Automation and Information Sciences (ICCAIS 2014) Gwangju, South Korea, Page No. 218-223, December 2-5, 2014.
12. K. V. Rashmi, Nihar B. Shah, KannanRamchandran and P. Vijay Kumar “Regenerating Codes for Errors and Erasures in Distributed Storage”, IEEE International Symposium on Information Theory Proceedings, 2012.
13. Christian Huebner, PawelStefanski, KostiantynDanilov, Igor Fedotov, “Ceph Best Practices Manual”, Version 1.0, 29-Feb-2016.
14. EMC white paper “EMC Unified Storage System Fundamentals for Performance and Availability”, August 30, 2011.
15. Michael Ratner, “Better Object Storage With Hitachi Content Platform” The Fundamentals of Hitachi Content Platform, white paper, November 2014.
16. Fujitsu Limited, Technical White Paper, FUJITSU Hyperscale Storage System ETERNUS CD10000, 2014.
17. Roman Talyansky, Bernd Scheuermann, Bj¨ornKolbeck, Jan Stender, “Trace-Driven Performance Evaluation of Distributed File Systems under Enterprise Application Workload”, arXiv:1001.2931v3 [cs.DC], 25 nov 2010.
18. EMC Corporation, White paper, “The Case For Tiered Storage In The Enterprise”, Feb 2016.

BIOGRAPHY

Priyanka Ramesh Nanaware is a P.G. Student in the Department of Electronics Engineering,PIITNew Panvel, Mumbai. She received Diploma in Electronics and Telecommunication in 2010 and Bachelor’s Degree in Electronics and Telecommunication Engineering in 2013 in Pune University. Her research interests are in Distributed Storage Area Network, network coding.

Shreya Bokare is Technical Officer in Computer Networks & Internet Engineering Research group at C-DAC Mumbai. She is working in the area of IPSAN and disaster recovery past 9 years and working as a team lead for projects in the area of IPSAN and Disaster Recovery.
ShreyaBokare is B.E and M.Tech in Electronics Engineering from Nagpur University and pursuing PhD in Electronics and Telecommunication Engineering from SNDT University. Her research interests are Storage area network, Disaster Recovery, I/O optimization, Network Coding.

Zia Saquib is Executive Director of C-DAC Mumbai and heads the Computer Networks & Internet Engineering Research group. He is the Chief Investigator of the project 'Cyber-security of SCADA Systems' and is the Architect of e-Pramaan Authentication System for the National e-Governance Plan of Govt. of India.
Zia Saquib received a Bachelor’s degree from Regional Engineering College Rourkela, M.S. Degree from Florida Institute of Technology, USA and PhD from National Institute of Technology Surat, all in Electrical Engineering. Dr Zia Saquib is a Fellow of Institution of Engineering & Technology (UK) and is Senior Member of IEEE.