SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT

ON

"RELIABILITY AWARE DATA DEDUPLICATION FOR CEPH"

Submitted toward the partial fulfillment of the Requirement of

BACHELOR OF ENGINEERING
(COMPUTER ENGINEERING)

BY

YOGESH BOJJA	B150284217
PANKAJ JAGTAP	B150284247
ABHISHEK BEDARKAR	B150284210
MILIND KULKARNI	B150284264

UNDER THE GUIDANCE OF

PROF. AMAR MORE

MIT | Academy of Engineering
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

School of Computer Engineering and Technology

MIT Academy of Engineering

Alandi(D), PUNE - 412105

2018-2019

A PROJECT REPORT
ON

"RELIABILITY AWARE DATA DEDUPLICATION FOR CEPH"

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

Submitted toward the partial fulfillment of the Requirement of

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING

BY

YOGESH BOJJA	B150284217
PANKAJ JAGTAP	B150284247
ABHISHEK BEDARKAR	B150284210
MILIND KULKARNI	B150284264

UNDER THE GUIDANCE OF

PROF. AMAR MORE

MIT Academy of Engineering
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

School of Computer Engineering and Technology

MIT Academy of Engineering

Alandi(D), Pune - 412105

2018-2019

MIT Academy of Engineering
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

School of Computer Engineering and Technology
MIT Academy of Engineering
ALANDI(D), PIN CODE-412 105

CERTIFICATE

This is to certify that the project entitled

"RELIABILITY AWARE DATA DEDUPLICATION FOR CEPH"

submitted by

YOGESH BOJJA	**B150284217**
PANKAJ JAGTAP	**B150284247**
ABHISHEK BEDARKAR	**B150284210**
MILIND KULKARNI	**B150284264**

is a bonafide work carried out by students under the supervision of Prof. Amar More and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Date: / /

(Prof. Amar More)
Project Guide

(Prof. R. R. Badre)
Dean, SCET

(Dr. Yogesh Bhalerao)
Director

Name of Internal Examiner

External Examiner

MIT Academy of Engineering
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

PROJECT APPROVAL SHEET

Is successfully completed by

“RELIABILITY AWARE DATA DEDUPLICATION FOR CEPH“

submitted by

YOGESH BOJJA	**B150284217**
PANKAJ JAGTAP	**B150284247**
ABHISHEK BEDARKAR	**B150284210**
MILIND KULKARNI	**B150284264**

at

School of Computer Engineering and Technology

MIT Academy of Engineering, Alandi(D)

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2018-19

Prof. Amar More	Prof. R. R. Badre
Internal Guide	Dean SCET
Computer Engineering.	

Acknowledgements

We are profoundly grateful to **Prof. AMAR MORE** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Dr. Y. J. BHALERAO**, Director, MIT ACADEMY OF ENGINEERING, **Prof. R. R. BADRE**, Dean, School of Computer Engineering and Technology and **Prof. S. KHANDELWAL**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

YOGESH BOJJA
PANKAJ JAGTAP
ABHISHEK BEDARKAR
MILIND KULKARNI

ABSTRACT

To reduce the cloud storage space and bandwidth in cloud storage while upload, the technique of abstract-data is widely used which eliminates duplicate copies. Cloud-storage utilizes storage reduces reliability as only one copy of the file is maintained even though the file is owned by multiple users. Due to the outsourcing of data by users to cloud, privacy issues for sensitive data keep arising. Keeping in mind above security challenges we attempt to standardize the notion of distributed system which is also reliable, in this paper.

Multiple cloud servers distribute data chunks among themselves in our new proposed distributed system. By the introduction deterministic secret sharing scheme in distributed storage system, we achieved the security requirements of data being confidential and tag consistent. Security analysis and terms specified in the proposed security model shows that, our system is secure. We demonstrate that the experienced overhead is very limited in a environment that is realistic and we implemented the proposed system, as a proof of the concept.

Keywords: Cluster, Object Storage Devices, CEPH, File Systems.

Contents

List of Figures

List of Tables

Synopsis

Chapter 1

Synopsis

1.1 Project Title

Reliability Aware Data Deduplication For CEPH

1.2 Project Option

Internal Project

1.3 Internal Guide

Prof. Amar More

1.4 Sponsorship and External Guide

None

1.5 Technical Keywords (As per ACM Keywords)

1. Computer Systems Organization
   (a) COMPUTER-COMMUNICATION NETWORKS
      i. Distributed Systems
         A. Client/server
         B. Distributed applications
         C. Distributed databases
         D. Network operating systems

E. Distributed file systems
F. Security and reliability issues in distributed applications

1.6 Problem Statement

To reduce the storage space, I/O disk operations which are used to store and manage large volume of data. We aim to provide a environment which is highly reliable and provide active deduplication support for CEPH.

1.7 Abstract

- To reduce the cloud storage space and bandwidth in cloud storage while upload, the technique of abstract-data is widely used which eliminates duplicate copies. Cloud-storage utilizes storage reduces reliability as only one copy of the file is maintained even though the file is owned by multiple users. Due to the outsourcing of data by users to cloud, privacy issues for sensitive data keep arising. Keeping in mind above security challenges we attempt to standardize the notion of distributed system which is also reliable, in this paper. Multiple cloud servers distribute data chunks among themselves in our new proposed distributed system. By the introduction deterministic secret sharing scheme in distributed storage system, we achieved the security requirements of data being confidential and tag consistent. Security analysis and terms specified in the proposed security model shows that, our system is secure. We demonstrate that the experienced overhead is very limited in a environment that is realistic and we implemented the proposed system, as a proof of the concept.

1.8 Goals and Objectives

- To reduce the amount of storage needed for given set of files.
- To reduce costs of storage.
- To increase space efficiency in distinct storage environment.
- To reduce I/O disk operation.

1.9 Relevant mathematics associated with the Project

System Description:

- k: This is the number of chunks the original data is divided into, also known as data chunks.
- m: This is extra code added to original data chunks to provide a data protection, also known as coding chunks. For ease of understanding, you can consider it as reliability level.
- n: This is the total number of chunks created after erasure process. In continuation to erasure coding equation , there are couple of more terms which are :- n: This is the total number of chunks created after erasure process. In continuation to erasure coding equation , there are couple of more terms which are :-
- Recovery: To perform recovery operation we would require any k chunks out of n chunks and thus can tolerate failure of any m chunks
- Reliability Level: We can tolerate failure upto any m chunks.
- Encoding Rate (r): r = k / n, where r ¡ 1
- Storage Required: 1 / r
- Placement Group calculations Choosing the correct number of PGs for each pool is one of the most important decisions when planning a Ceph cluster. We can target the total number of placement groups for a pool using the following formula : Total PGs = (Total OSDs * PSPerOSD)/Replication factor Put another way the ratio can be calculated PGPerOSD = (Total PGs)/(Total OSDs/Replication factor)
- (Total OSDs * PGPerOSD)/Replication factor = Total PGs (50*200)/3 ¡= 3333

1.10 Names of Conferences / Journals where papers can be published

- IEEE 2019 International Conference for Convergence in Technology, March 2019.
- RF-384th International Conference on Science, Engineering and Technology (ICSET 2019).

- International Journal of Computer Engineering and Application.
- National Conference on Emerging Trends in Computer Engineering and Technology. (NCETCET19).

1.11 Review of Conference/Journal Papers supporting Project idea

- Sage A. Weil, Scott A. Brandt, L. Miller Ethan, Carlos altzahn, CRUSH: Controlled scalable decentralized placement of replicated data, Proceedings of the 2006 ACM/IEEE conference on Supercomputing, pp. 122, 2006.
- Prof. Carlos Maltzahn, Dr. Esteban Molina-Estolano , Ceph: A scalablehigh-performance distributed file system, Proceedings of the 7th symposium on Operating systems design and implementation, pp. 307-320, 2006.
- Amandeep Khurana, Dr. Alex Nelson, Ceph as a scalable alternative to the Hadoop Distributed File System, login: The USENIX Magazine, vol. 35, pp. 38-49, 2010
- Ceph Storage Red Hat, 2015, [online] Available: http://ceph.com/. .
- Ceph Benchmarks, Sebastien Han, [online] Available: http://www.sebastienhan.fr/blog/2012 benchmarks/. .
- Sage A. Weil, Scott A. Brandt, L. Miller Ethan and Carlos altzahn.. Blocklevel security for network-attached disks. In Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST), pages 159174, San Francisco, CA, 2003
- Benchmark Disk IO with DD and Bonnie++ James Cole, [online] Available: http://www.jamescoyle.net/how-to/599-benchmark-disk-io-with-dd-andbonnie.
- A. Azagury, R. Canetti, M. Factor, S. Halevi, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, and J. Satran. A two layered approach for securing an object store network. In IEEE Security in Storage Workshop, pages 1023, 2002.
- K. Fu, M. F. Kaashoek, and D. Mazieres. Fast and secure distributedread-only file system. In Proceedings of the 4th Symposium on Operating Systems Design and Implementation (OSDI), pages 181196, San Diego, CA, Oct. 2000.

- G. A. Gibson, D. F. Nagle, K. Amiri, J. Butler, F. W. Chang, H. Gobioff, C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka. A cost-effective, high-bandwidth storage architecture. In Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 92103, San Jose, CA, Oct. 1998

1.12 Plan of Project Execution

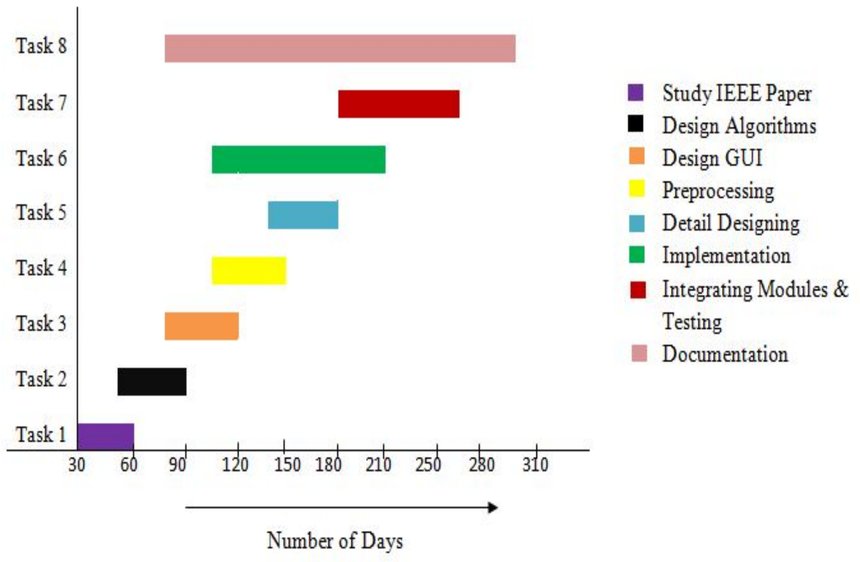


Figure 1.1: Plan of Project Execution

Chapter 2

Technical Keywords

2.1 Area of Project

Cloud Computing and Distributed Storage Systems

2.2 Technical Keywords

1. Computer Systems Organization
    (a) COMPUTER-COMMUNICATION NETWORKS
        i. Distributed Systems
            A. Client/server
            B. Distributed applications
            C. Distributed databases
            D. Network operating systems
            E. Distributed file systems
            F. Security and reliability issues in distributed applications

Chapter 3

Introduction

3.1 Project Idea

- Secure Distributed De-Duplication Systems With Improved Reliability

3.2 Motivation of the Project

- Production of data is expanding at an astonishing pace
- 60-70 percent of the data stored in the data server are duplicate data
- SLA say that data should be available 99.999
- Huge amount of data require more storage, processing and network bandwidth.

3.3 Literature Survey

- Sage A. Weil, Scott A. Brandt, L. Miller Ethan and Carlos Altzahn.[1] proposed an approach that deals with huge files and directories, and coordinates activities of thousands of disks, provides parallel access to metadata on a massive scale authenticate and encrypt at scale, to shrink and grow dynamically, system failures, and dynamic cluster expansions. As you cannot just shut down the system because of a disk failure or shortage of space. CEPH is an object-based parallel file system whose design is based on two key ideas. The first key idea is object-based storage, that splits the traditional file system architecture into a client component and storage component.
- Prof. Carlos Maltzahn, Dr. Esteban Molina-Estolano.[2] idealized that the storage component manages disk scheduling and layout locally, relieving clients

and servers from low-level per-disk details and increasing scalability. This design allows clients to communicate with storage nodes via a high level interface and manage data in terms of objects, which are chunks of data much larger than 512-byte blocks. The T10 standard of the SCSI Object Storage Device (OSD). Ceph uses and significantly extends the concept of OSDs. For all practical purposes, think of a Ceph OSD as a process that runs on a cluster node and uses a local file system to store data objects. MDS in CEPH manages the metadata in CEPH.

- Amandeep Khurana, Dr. Alex Nelson.[3] suggested that data management differs fundamentally from management of metadata: file data storage is trivially parallelizable and is limited primarily by the network infrastructure. Metadata management is much more complex, because hierarchical directory structures impose interdependencies (e.g. POSIX access permissions depend on parent directories) and the metadata server must maintain file system consistency. Metadata servers have to withstand heavy workloads: 3080 percent of all file system operations involve metadata, so there are lots of transactions on lots of small metadata items following a variety of usage patterns. Good metadata performance is therefore critical to overall system performance and thus is present in CEPH. Popular files and directories are common, and concurrent access can overwhelm many schemes

- Sage A. Weil, Scott A. Brandt, L. Miller Ethan and Carlos Altzahn.[4] said Brick and object-based storage architectures have emerged as a means of improving the scalability of storage clusters. However, existing systems continue to treat storage nodes as passive devices, despite their ability to exhibit significant intelligence and autonomy. They presented the design and im- plementation of RADOS, which is a reliable object storage service that can scales to many thousands of devices by leveraging the intelligence present in individual storage nodes. RADOS preserves consistent data access and strong safety seman- tics while allowing nodes to act semiautonomously to self- manage replication, failure detection, and failure recovery through the use of a small cluster map. Our implementa- tion offers excellent performance, reliability, and scalability while providing clients with the illusion of a single logical object store.

- Scott A. Brandt Ethan L. Miller.[5] developed CRUSH, Controlled Replication Under Shared Hashing which is a scalable pseudo-random data distribution function designed for distributed object-based storage systems that efficiently maps data objects to storage devices without relying on a central directory. Because large systems are inherently dynamic, CRUSH is designed to facilitate the addition and removal of storage while minimizing unnecessary data movement. The algorithm accommodates a wide variety of data replication and reliability mechanisms and distributes data in terms of user defined policies that enforce separation of replicas across failure domains. CEPH file system uses CRUSH Algorithm for Data Sharing amongs the Object-Storage nodes and manages the data making the file system Scalable as a whole.

- Carlos Maltzahn and Scott A. Brand.[6] They have developed and implemented RADOS, a Reliable, Autonomic Distributed Object Store that seeks to leverage device intelligence to distribute the complexity surrounding consistent data access, redundant storage, failure detection, and failure recovery in clusters consisting of many thousands of storage devices. Built as part of the Ceph distributed file system, RADOS facilitates an evolving, balanced distribution of data and workload across a dynamic and heterogeneous storage cluster while providing applications with the illusion of a single logical object store with well-defined safety semantics and strong consistency guarantees.

- Scott A. Brandt Ethan D. E. Long developed Ceph.[7] which is a distributed le system that provides excellent performance, reliability, and scalability. Ceph File System maximizes the separation between data and metadata management by replacing allocation tables with a pseudo-random data distribution function (CRUSH) designed for dynamic clusters of unreliable object storage devices(OSDs). Device intelligence by distributing data replication, failure detections and recovery to semi-autonomous OSDs running a specialized local object le system. A dynamic distributed metadata cluster provides hihghly efcient metadata management and seamlessly adapts to wide range of general purpose and scientic computing le system workloads.

- Myoungwon Oh, Sejin Park, Jungyeon Yoon, Sangjae Kim.[8] described how it is a challenge to store and manage very large sets of contents being generated by the explosion of data. One of the promising solutions to mitigate big data issues is data deduplication, which removes redundant data across many nodes of the storage system. To address these challenges, they proposed a new deduplication method, which is highly scalable and compatible with the existing scale-out storage. Specifically, our deduplication method employs a double hashing algorithm that leverages hashes used by the underlying scale-out storage, which addresses the limits of current fingerprint hashing. Also, their design integrates the meta-information or meta-data of file system and deduplication into a single object, and it controls the deduplication ratio at online by being aware of system demands based on post-processing.

- Kang-won Lee, Sage Weil, Heon Y. Yeom, Myoungsoo.[9] Jung said applying data deduplication on the existing distributed storage system considering redundancy scheme is non-trivial due to its data processing procedure and additional metadata management, the deduplication system compares each chunk with the existing data chunks, stored in the storage previously. A unique attribute (such as a fingerprint) index that stores the hash value of each chunk is employed by the deduplication system in order to easily find the existing chunks by comparing hash value rather than searching all contents that reside in the underlying storage and thus manages the distribution over OSDs.
- Christopher Olson and Ethan L. Miller.[10] said that Network-Attached Secure Disk (NASD) model has become a more widely used technique for constructing large-scale storage systems. However, the security system proposed for NASD assumes that each client will contact the server to get a capability to access one object on a server. While this approach works well in smaller-scale systems in which each file is composed of a few objects, it fails for largescale systems in which thousands of clients make accesses to a single file composed of thousands of objects spread across thousands of disks. The file system we are building, Ceph, distributes files across many objects and disks to distribute load and improve reliability.

Chapter 4

Problem Definition and scope

4.1 Problem Statement

In this project we aim to reduce the storage space, I/O disk operation which are used to store and manage large volume of data. We aim to provide a environment which is highly reliable and provide active deduplication support for CEPH.

4.1.1 Goals and objectives

- To reduce storage redundancy.
- To reduce network traffic
- To provide active deduplication support to CEPH.

4.1.2 Statement of scope

- Input type in layer is file which is spllited into 128 kb of block size.
- filename with arbitary file system is provided which is stored in osd's
- No restriction on file size any size is accepted.
- output of application is retrival of file from osd's

4.2 Major Constraints

- text file as some storing and retrieving problems.
- compressed file given as input file to application may create problem at file retrieval

4.3 Methodologies of Problem solving and efficiency issues

- Distributed file storage is used as method to solve redundancy problem.
- SHA(256) problems helps to hash blocks on metadata server and hash value is used to recognize redundant blocks.
- File level hashing method is eliminated and block level hashing style is implemented.
- encryption and decryption of file is omitted to reduce complex implementation.

4.4 Outcome

- A reliability aware ceph architecture which provides active deduplication support on all types of file storage.

4.5 Applications

- Cloud Storage services [Amazon S3,Amazon DB].

4.6 Hardware Resources Required

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	sufficient
2	RAM	8 GB	sufficient
3	Switch	5 Socket	Static Ip
4	Systems	5	Mon, Client, 4 OSDs

Table 4.1: Hardware Requirements

4.7 Software Resources Required

1. Operating System: Linux (CentOS 7)
2. IDE: Sublime Text, CodeBlocks, Eclipse Oxygen.
3. Programming Language: C++.

Chapter 5

Project Plan

5.1 Project Estimates

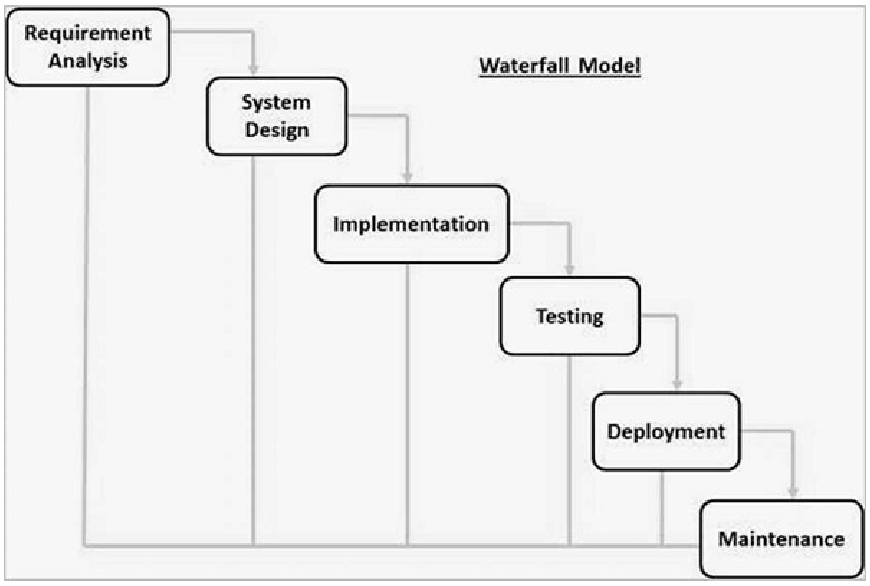


Figure 5.1: Waterfall Model

- **Requirement Gathering and analysis**: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design**: The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hard-ware and system requirements and also helps in defining overall system architecture.
- **Implementation**: With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit

is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing**: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system**: Once the functional and non functional testing is done,the product is deployed in the customer environment or released into the market.

5.1.1 Reconciled Estimates

Cost Estimate

As the software we have used is open source software. Our project is free of cost.

Time Estimates

The time required for the project is 10 months.

5.1.2 Project Resources

PC/Laptop with i5 or above Processor and 8 GB RAM, Any Linux disto, Ecipse IDE or Sublime Text Editor, LAN connection.

5.2 Risk Management w.r.t. NP Hard analysis

5.2.1 Risk Identification

Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents. A Risk Management Log will be generated and updated as needed and will be stored electronically in the project library located at file location.

5.2.2 Risk Analysis

The probability and impact of occurrence for each identified risk will be assessed by the project manager, with input from the project team using the following approach:

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Block size is not variable	High	Low	Low	Medium
2	Proper internet connection must be available	Medium	Low	High	High

Table 5.1: Risk Description

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.2: Risk Probability definitions

Probability :

- High Risk that has the potential to greatly impact project cost, project schedule or performance
- Medium Risk that has the potential to slightly impact project cost, project schedule or performance
- Low Risk that has relatively little impact on cost, schedule or performance

5.2.3 Quantitative Risk Analysis

Analysis of risk events that have been prioritized using the qualitative risk analysis process and their affect on project activities will be estimated, a numerical rating applied to each risk based on this analysis, and then documented in this section of the risk management plan.

5.2.4 Risk Response Planning

Each major risk (those falling in the Red Yellow zones) will be assigned to a project team member for monitoring purposes to ensure that the risk will not fall through the cracks. For each major risk, one of the following approaches will be selected to address it:

1. Avoid eliminate the threat by eliminating the cause.

2. Mitigate Identify ways to reduce the probability or the impact of the risk.
3. Accept Nothing will be done .
4. Transfer Make another party responsible for the risk (buy insurance, outsourcing.)

For each risk that will be mitigated, our project team will identify ways to prevent the risk from occurring or reduce its impact or probability of occurring. This may include prototyping, adding tasks to the project schedule, adding resources,etc. For each major risk that is to be mitigated or that is accepted, a course of action will be outlined for the event that the risk does materialize in order to minimize its impact.

5.2.5 Overview of Risk Mitigation, Monitoring, Management

The level of risk on a project will be tracked, monitored and reported throughout the project life cycle. A Top 10 Risk List will be maintained by the project team and will be reported as a component of the project status reporting process for this project. All project change requests will be analyzed for their possible impact to the project risks. Management will be notified of important changes to risk status as a component to the Executive Project Status Report. Following are the details for each risk.

Following are the details for each risk.

Risk ID	1
Risk Description	Block size is not variable
Category	Development Environment.
Source	Software requirement Specification document.
Probability	High
Impact	High
Response	Mitigate
Strategy	Variable block size will resolve this issue
Risk Status	Occurred

Risk ID	2
Risk Description	Proper internet connection must be available
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Connect to a good ISP.
Risk Status	Identified

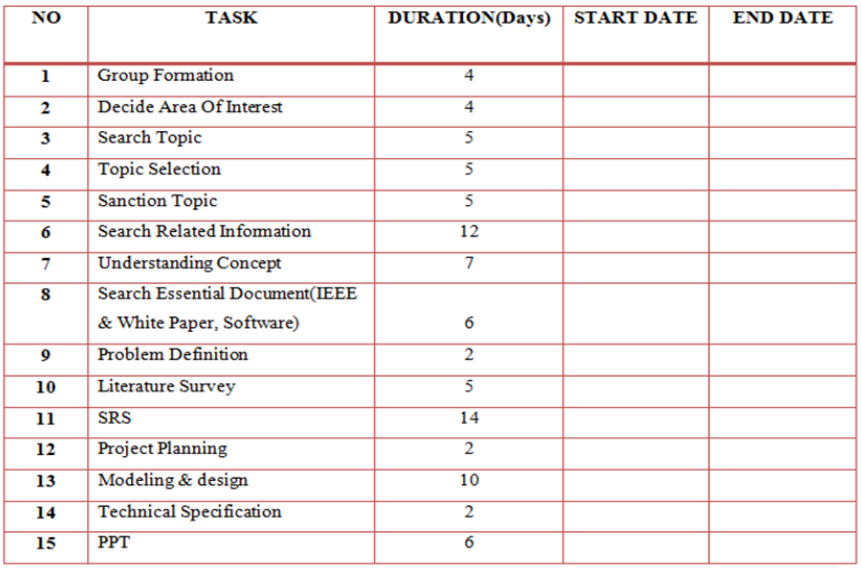
5.3 Project Schedule

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Install centos7 on all systems
- Task 2: Configure ceph architecture on nodes
- Task 3: Apply deduplication layer on default architecture
- Task 4: Pass file operations on deduplication layer
- Task 5: Analyze data stored in Object Storage Devices.

5.3.2 Task network



NO	TASK	DURATION(Days)	START DATE	END DATE
1	Group Formation	4		
2	Decide Area Of Interest	4		
3	Search Topic	5		
4	Topic Selection	5		
5	Sanction Topic	5		
6	Search Related Information	12		
7	Understanding Concept	7		
8	Search Essential Document(IEEE & White Paper, Software)	6		
9	Problem Definition	2		
10	Literature Survey	5		
11	SRS	14		
12	Project Planning	2		
13	Modeling & design	10		
14	Technical Specification	2		
15	PPT	6		

Figure 5.2: Task Network

5.3.3 Timeline Chart

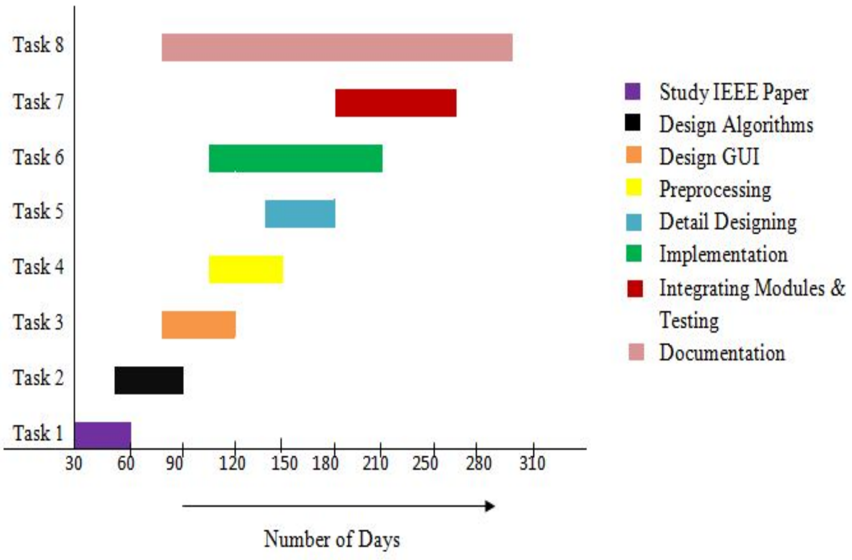


Figure 5.3: Timeline Chart

5.4 Team Organization

Our Internal Guide is Mr. Amar More. He always provides us helpful guidance, suggestions and immense support and also provided us with valuable suggestions.

5.4.1 Team structure

Our team consists of four members Abhishek Bedarkar, Yogesh Bojja, Milind Kulkarni and Pankaj Jagtap. All the four members contributed in each and every phase of the project, i.e. research work and survey and documentation. Even though the work was equally distributed we all cooperated with each other to get the job done faster. This has helped towards the development of project successfully.

5.4.2 Management reporting and communication

We all share a great bond with each other and try to help each other in the time of needs. We have a cordial relation with our guide. We keep them informed of our weekly progress and updates in the project. Their valuable advice are always helpful whenever we face any problem.

Software requirement specification

Chapter 6

Software requirement specification

6.1 Introduction

6.1.1 Purpose and Scope of Document

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to users. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project,outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the projects software requirements specification (SRS) document is specifically designed to utilize the distributed storage system approach to provide the facility of de-duplication in already existing storage system CEPH. Developer and end user interested in this documentation would include but not be limited to the system owners, the system users, project manager and the design team.

6.1.2 Overview of responsibilities of Developer

The key responsibilities of a developer are :-

- To understand the problem that the software is supposed to solve.
- To Design a solution.
- To develop and test it before releasing it to customers.

6.2 Usage Scenario

- Can do file opearation with data deduplication
- Provide active deduplication support
- Increase network bandwidth

6.2.1 User profiles

User can save all types and size of file in storage devices with active deduplication support. User is able to perform all type of file operations.

6.2.2 Use Case View

Use Case Diagram. Example is given below:

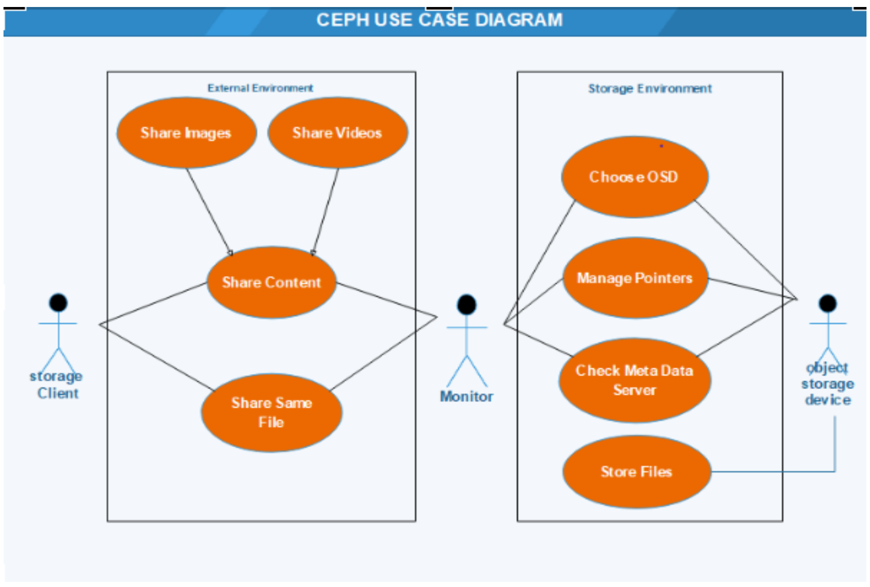


Figure 6.1: Use case diagram

6.3 Data Model and Description

6.3.1 Data Description

Data objects that will be managed/manipulated by the software are described in this section. The database entities or files or data structures required to be described. For data objects details can be given as below

6.3.2 Data objects and Relationships

Data objects and their major attributes and relationships among data objects are described using an ERD- like form.

6.4 Functional Model and Description

6.4.1 Activity Diagram:

- The Activity diagram represents the steps taken.

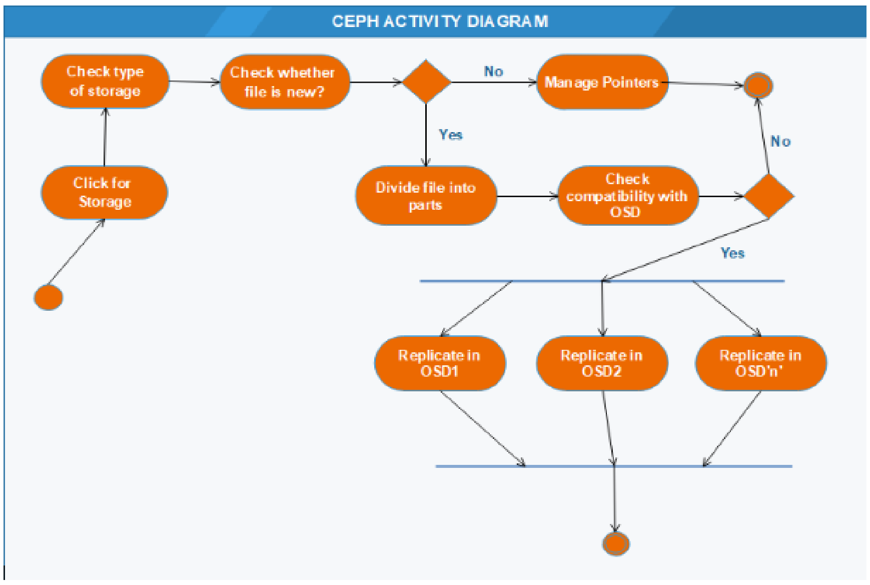


Figure 6.2: Activity diagram

6.4.2 Non Functional Requirements:

- Interface Requirements
- Performance Requirements
- Software quality attributes such as availability [related to Reliability], modifiability [includes portability, reusability, scalability] , performance, security, testability and usability[includes self adaptability and user adaptability]

6.4.3 State Diagram:

State Transition Diagram

Fig.6.3 example shows the state transition diagram of Cloud SDK. The states are represented in ovals and state of system gets changed when certain events occur. The transitions from one state to the other are represented by arrows. The Figure shows important states and events that occur while creating new project.

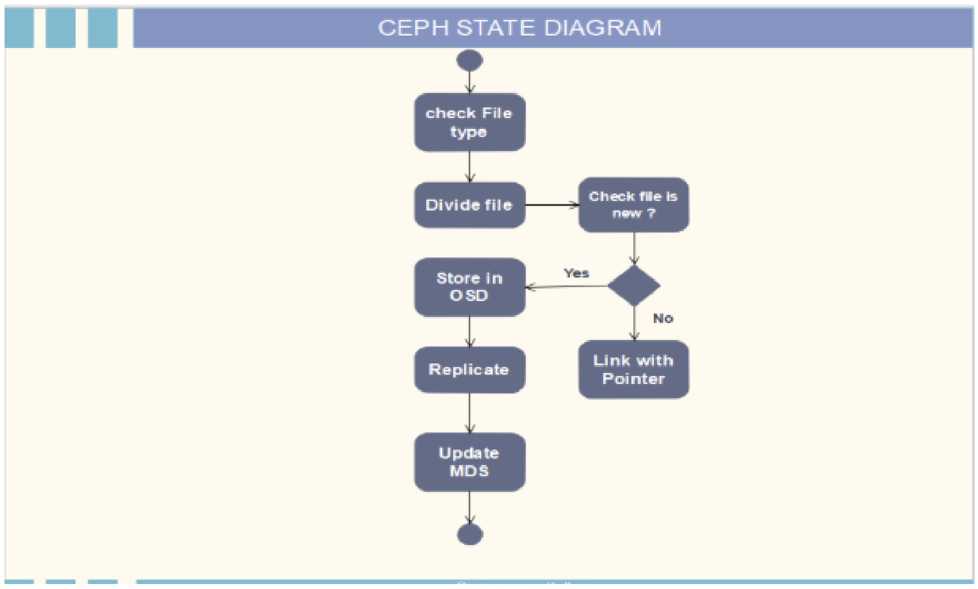


Figure 6.3: State transition diagram

6.4.4 Design Constraints

- Fixed block size.
- DHT is distributed but not in ring fashion.

6.4.5 Software Interface Description

The software interface(s)to the outside world is(are) described. The requirements for interfaces to other devices/systems/networks/human are stated.

Detailed Design Document using Appendix A and B

Chapter 7

Detailed Design Document using Appendix A and B

7.1 Introduction

The design implemented is Distributed in nature, it consists of multiple nodes that when combined together form a cluster to carry out the desired storage and retrieval process.

7.2 Architectural Design

Architechture consists of ADMIN, OSDs, MON, MDS and CLIENT systems. ADMIN administrates and manages the entire architechture, MON co-ordinates with OSDs to store the incoming data from CLIENT, MDS maintains the metadata for the same.

7.3 Data design (using Appendices A and B)

There are various data structures used, there are in-memory and permanent data structures used. Data structures like hash tables and files are the major ones used throughout the software.

7.3.1 Internal software data structure

Hash table is the Internal software data structure used, it is for storing hash values, duplication counter, name of the file blocks and the original file.

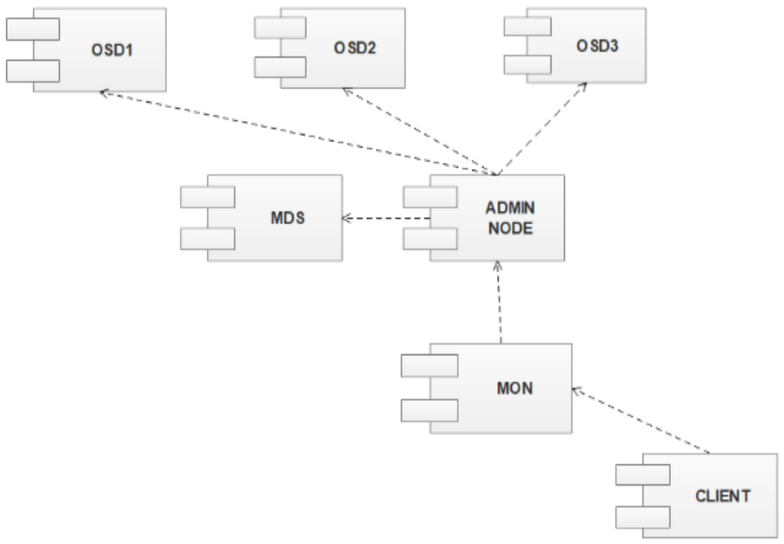


Figure 7.1: Architecture diagram

7.3.2 Global data structure

Files are used to store hash tables, ip addresses and the hash values of the file blocks.

7.3.3 Temporary data structure

Temporary files are used to store the IP addresses and hash values of file blocks.

7.3.4 Database description

No database was involved in this project.

7.4 Component Design

7.4.1 Class Diagram

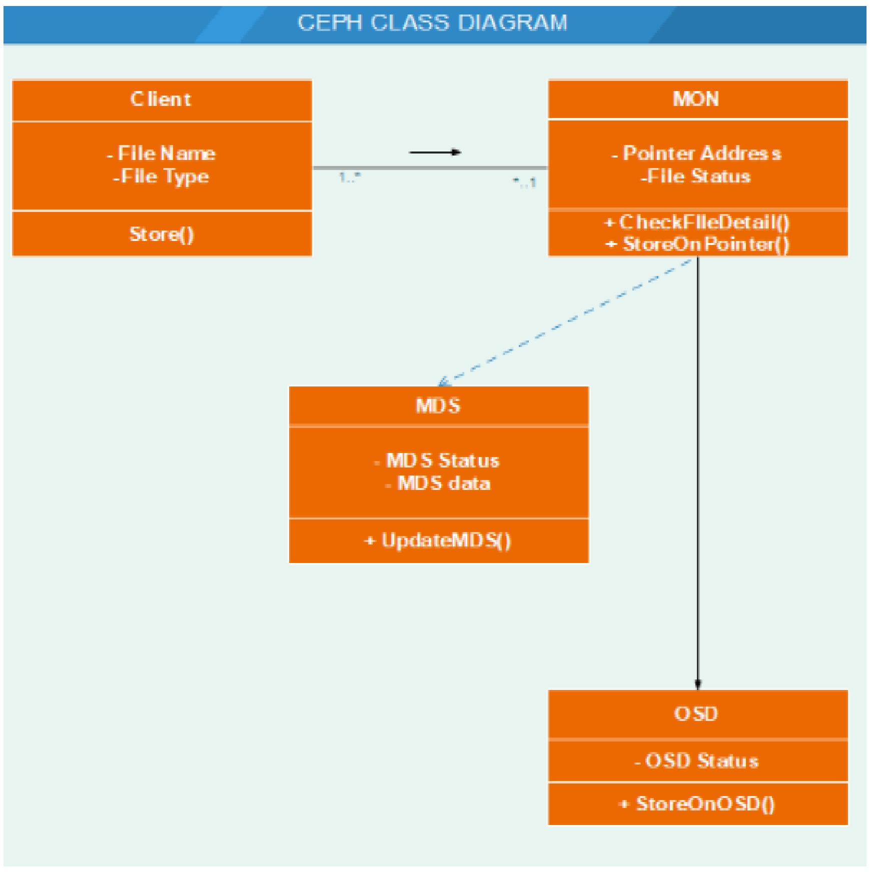


Figure 7.2: Class Diagram

Project Implementation

Chapter 8

Project Implementation

8.1 Introduction

To reduce the cloud storage space and bandwidth in cloud storage while upload, the technique of abstract-data is widely used which eliminates duplicate copies. Cloud-storage utilizes storage reduces reliability as only one copy of the file is maintained even though the file is owned by multiple users. Due to the outsourcing of data by users to cloud, privacy issues for sensitive data keep arising. Keeping in mind above security challenges we attempt to standardize the notion of distributed system which is also reliable.

8.2 Tools and Technologies Used

- CEPH (open source file system)
- CentOS 7 (Linux)
- C++ (File Handling, Bit Manipulation, Hashing)
- Data Structures
- IDEs and Text Editors (Sublime, Eclipse)
- Switch
- Ethernet Cables

8.3 Methodologies/Algorithm Details

8.3.1 Algorithm 1/Pseudo Code

store(FileHashTable **ft)

- Initialize File Pointer in WRITE mode i.e fout, fout1, fout2, fout3, fout4
- Open IP Files 192.168.6.10, 192.168.6.12, 192.168.6.13, 192.168.6.14
- Input fileName from User and add fileName to FileHashTable.
- Calculate Hash Value of the fileName with getHash() and store it into hash variable
- REPEAT
  - Read 128KB of Block and Calculate 28 Bit decimal value of it from SHA256, store this value in newHashValue variable.
  - Create file block with SHA256 value as its name.
  - Add the SHA256 value of the file block to its fileName in FileHashTable.
  - Fine the destination Host with findDestOSD() and WRITE the content(28 bit decimal + SHA256 Hash) in the respective IP File.
- REPEAT UNTIL Block size less than 128KB or Filepointer is NULL.
- Send this IP file from MON to respective IPs(OSDs).
- Recieve File OSD'X' from IPs(OSDs) where 'X' is 1/2/3/4 and which contains hash of non-duplicate blocks.
- Read Hash value from this file and store all of them using "Rados -p rbdpool put" command into CEPH.

8.3.2 Algorithm 2/Pseudo Code

retrieve(FileHashTable **ft)

- Input fileName from the User and calculate its Hash Value with getHash and store it into Hash.
- Retrieve the hashvalue of the Blocks using Hash in FileHashTable.
- Retrieve all the blocks from CEPH using "Rados -g rbdpool get" command.
- Open the file with Name fileName(given by user) and write content of all the block to it sequentially.

8.4 Verification and Validation for Acceptance

The testing process is a part of broader subject referring to verification and validation. We have to acknowledge the system specifications and try to meet the customer's requirements and for this sole purpose, we have to verify and validate the product to make sure everything is in place. Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are properly met or not by the end product. Verification of the project was carried out to ensure that the project met all the Requirement and specification of our project. We made sure that our project is up to the standard as we planned at the beginning of our project development.

Software Testing

Chapter 9

Software Testing

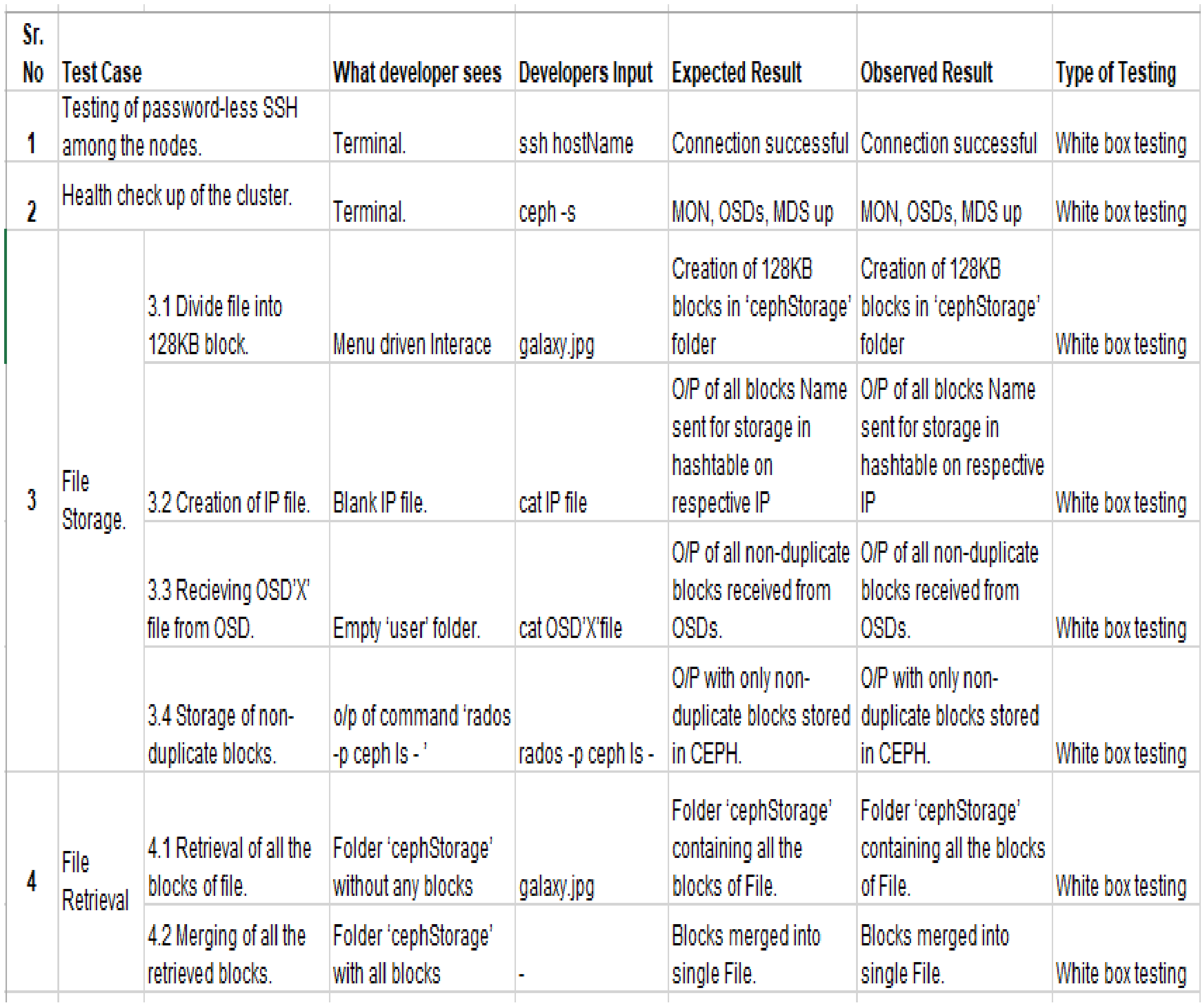
9.1 Type of Testing Used

- **Functionality testing:**
  Functionality testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Using Functionality testing we check whether the flow of Execution is accurate or not.
  - Applying for storage.
  - Checking file type.
  - Dividing the file in parts.
  - replication of files
- **Implementation testing:**
  - **Black-box testing:**
    It is carried out to test functionality of the program and also called Behavioral testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested ok, and problematic otherwise
  - **White box testing:**
    It is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.
- **Unit Testing:**
  Unit testing focuses on the smallest unit of software design i.e. the smallest component or module. Important control paths are tested to uncover errors

within the boundary of the module. It focuses on the internal processing logic and data structures within the boundaries of a component.This type of testing can be conducted in parallel for multiple components.

- **Integration Testing:**
  Integration testing is a systematic technique for constructing the software architecture while at the same time conducting tests to uncover errors associated with interfacing. The different modules in our project were interfaced and tested in small increments, thus making the errors easy to isolate and correct. This is known as incremental integration.

9.2 Test Cases and Test Results



Sr. No	Test Case		What developer sees	Developers Input	Expected Result	Observed Result	Type of Testing
1	Testing of password-less SSH among the nodes.		Terminal.	ssh hostName	Connection successful	Connection successful	White box testing
2	Health check up of the cluster.		Terminal.	ceph -s	MON, OSDs, MDS up	MON, OSDs, MDS up	White box testing
3	File Storage.	3.1 Divide file into 128KB block.	Menu driven Interace	galaxy.jpg	Creation of 128KB blocks in 'cephStorage' folder	Creation of 128KB blocks in 'cephStorage' folder	White box testing
		3.2 Creation of IP file.	Blank IP file.	cat IP file	O/P of all blocks Name sent for storage in hashtable on respective IP	O/P of all blocks Name sent for storage in hashtable on respective IP	White box testing
		3.3 Recieving OSD'X' file from OSD.	Empty 'user' folder.	cat OSD'X'file	O/P of all non-duplicate blocks received from OSDs.	O/P of all non-duplicate blocks received from OSDs.	White box testing
		3.4 Storage of non-duplicate blocks.	o/p of command 'rados -p ceph ls - '	rados -p ceph ls -	O/P with only non-duplicate blocks stored in CEPH.	O/P with only non-duplicate blocks stored in CEPH.	White box testing
4	File Retrieval	4.1 Retrieval of all the blocks of file.	Folder 'cephStorage' without any blocks	galaxy.jpg	Folder 'cephStorage' containing all the blocks of File.	Folder 'cephStorage' containing all the blocks of File.	White box testing
		4.2 Merging of all the retrieved blocks.	Folder 'cephStorage' with all blocks	-	Blocks merged into single File.	Blocks merged into single File.	White box testing

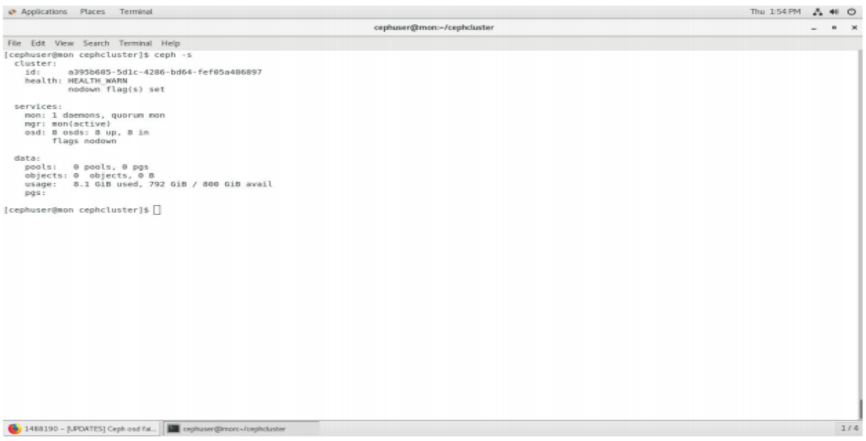
Figure 9.1: Testing Table

Result

Chapter 10

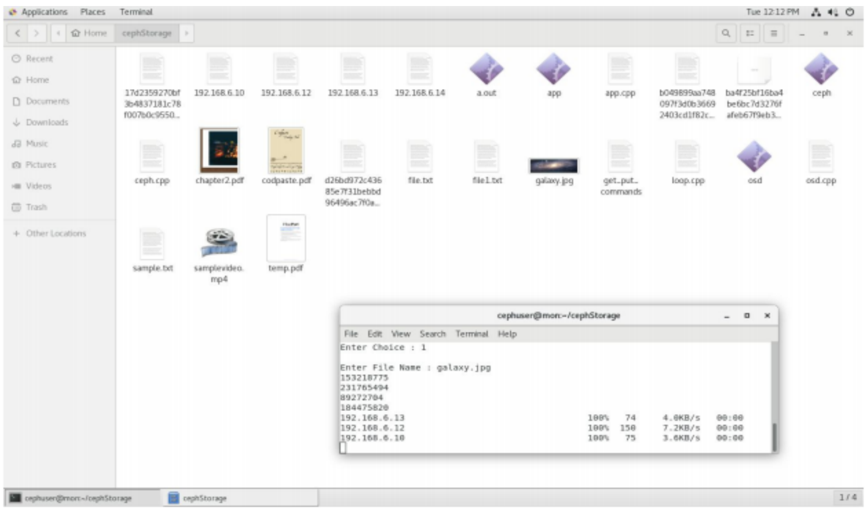
Results

10.1 Screenshots



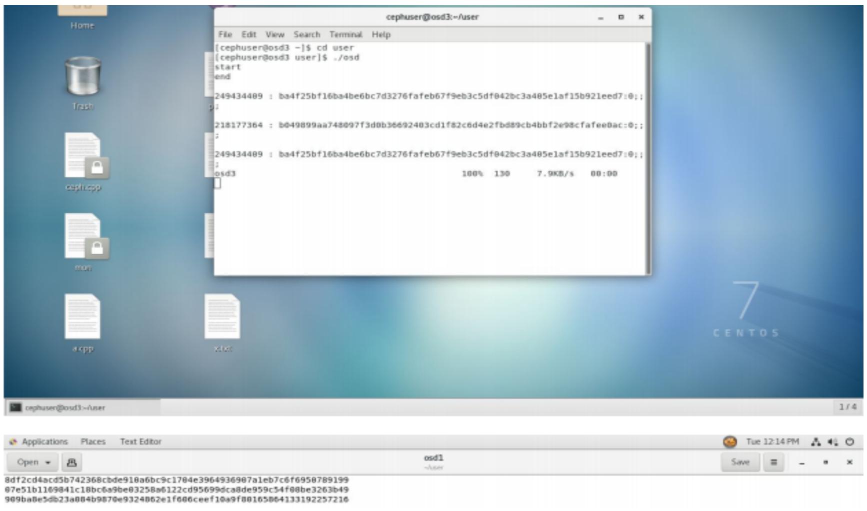
```
[cephuser@mon cephcluster]$ ceph -s
  cluster:
    id:     a395b685-5d1c-4286-bd64-fef05a486897
    health: HEALTH_WARN
            nodown flag(s) set

  services:
    mon: 1 daemons, quorum mon
    mgr: mon(active)
    osd: 8 osds: 8 up, 8 in
         flags nodown

  data:
    pools:   0 pools, 0 pgs
    objects: 0  objects, 0 B
    usage:   8.1 GiB used, 792 GiB / 800 GiB avail
    pgs:

[cephuser@mon cephcluster]$
```

Figure 10.1: Cluster status



```
Enter Choice : 1

Enter File Name : galaxy.jpg
153218775
231765494
89272704
184475820
192.168.6.13                                   100%   74     4.0KB/s   00:00
192.168.6.12                                   100%  150     7.2KB/s   00:00
192.168.6.10                                   100%   75     3.6KB/s   00:00
```

Figure 10.2: Transfer status and File blocks



```
[cephuser@osd3 ~]$ cd user
[cephuser@osd3 user]$ ./osd
start
end

249434409 : ba4f25bf16ba4be6bc7d3276fafeb67f9eb3c5df042bc3a405e1af15b921eed7:0;;
;

218177364 : b049899aa748097f3d0b36692403cd1f82c6d4e2fbd89cb4bbf2e98cfafee0ac:0;;
;

249434409 : ba4f25bf16ba4be6bc7d3276fafeb67f9eb3c5df042bc3a405e1af15b921eed7:0;;
;
osd3                                         100%  130     7.9KB/s   00:00
```

```
8df2cd4acd5b742368cbde910a6bc9c1704e3964936907a1eb7c6f6950789199
07e51b1169841c18bc6a9be03258a6122cd95699dca8de959c54f08be3263b49
909ba8e5db23a884b9870e9324862e1f606ceef10a9f80165864133192257216
```

Figure 10.3: IP-files and File Block Table

10.2 Outputs

- 128 kb sized blocks of input file
- 256 bit Hash values of file blocks
- Increment in duplication counter
- Distribution of blocks on OSDs
- Successful de-duplication of file
- Successful upload and download of file

Deployment and Maintenance

Chapter 11

Deployment and Maintenance

11.1 Installation and un-installation

1. Block Device Quick Start
2. Install Ceph
3. Create a Block Device Pool
4. Configure a Block Device
5. Filesystem Quick Start
6. Create a Filesystem
7. Create a Secret File
8. Kernel Driver
9. Filesystem in User Space (FUSE)
10. Additional Information
11. Object Storage Quick Start
12. Installing Ceph Object Gateway
13. Creating the Ceph Object Gateway Instance
14. Configuring the Ceph Object Gateway Instance

11.2 User help

- The user can monitor all services from the homepage of web application.
- A user guide will be provided while deploying the project to user.

Conclusion and future scope

Chapter 12

Conclusion and future scope

Conclusion:

- Successfully created a IN-MEMORY data structure which is able to detect duplicate data and store only one instance of the duplicate data hence improving the resources like storage space, disk I/O operation .
- Successfully provided an environment which is highly reliable and have active de-duplication support.
- Successfully managed the trade off between replication and de-duplication.

Future Scope:

- Ceph file system modification.
- To provide API for data de-duplication.
- More replication of important blocks.

Appendices

References

References

[1] Sage A. Weil, Scott A. Brandt, L. Miller Ethan, Carlos altzahn, CRUSH: Controlled scalable decentralized placement of replicated data, Proceedings of the 2006 ACM/IEEE conference on Supercomputing, pp. 122, 2006.

[2] Prof. Carlos Maltzahn, Dr. Esteban Molina-Estolano , Ceph: A scalablehigh-performance distributed file system, Proceedings of the 7th symposium on Operating systems design and implementation, pp. 307-320, 2006.

[3] Amandeep Khurana, Dr. Alex Nelson, Ceph as a scalable alternative to the Hadoop Distributed File System, login: The USENIX Magazine, vol. 35, pp. 38-49, 2010

[4] Ceph Storage Red Hat, 2015, [online] Available: http://ceph.com/. .

[5] Ceph Benchmarks, Sebastien Han, [online] Available: http://www.sebastienhan.fr/blog/2012/08/26/ceph-benchmarks/. .

[6] Sage A. Weil, Scott A. Brandt, L. Miller Ethan and Carlos altzahn.. Blocklevel security for network-attached disks. In Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST), pages 159174, San Francisco, CA, 2003

[7] Benchmark Disk IO with DD and Bonnie++ James Cole, [online] Available: http://www.jamescoyle.net/how-to/599-benchmark-disk-io-with-dd-andbonnie.

[8] A. Azagury, R. Canetti, M. Factor, S. Halevi, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, and J. Satran. A two layered approach for securing an object store

network. In IEEE Security in Storage Workshop, pages 1023, 2002.

[9] K. Fu, M. F. Kaashoek, and D. Mazieres. Fast and secure distributedread-only file system. In Proceedings of the 4th Symposium on Operating Systems Design and Implementation (OSDI), pages 181196, San Diego, CA, Oct. 2000.

[10] G. A. Gibson, D. F. Nagle, K. Amiri, J. Butler, F. W. Chang, H. Gobioff, C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka. A cost-effective, high-bandwidth storage architecture. In Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 92103, San Jose, CA, Oct. 1998

Laboratory assignments on Project Analysis of Algorithmic Design

Appendix A

Laboratory assignments on Project Analysis of Algorithmic Design

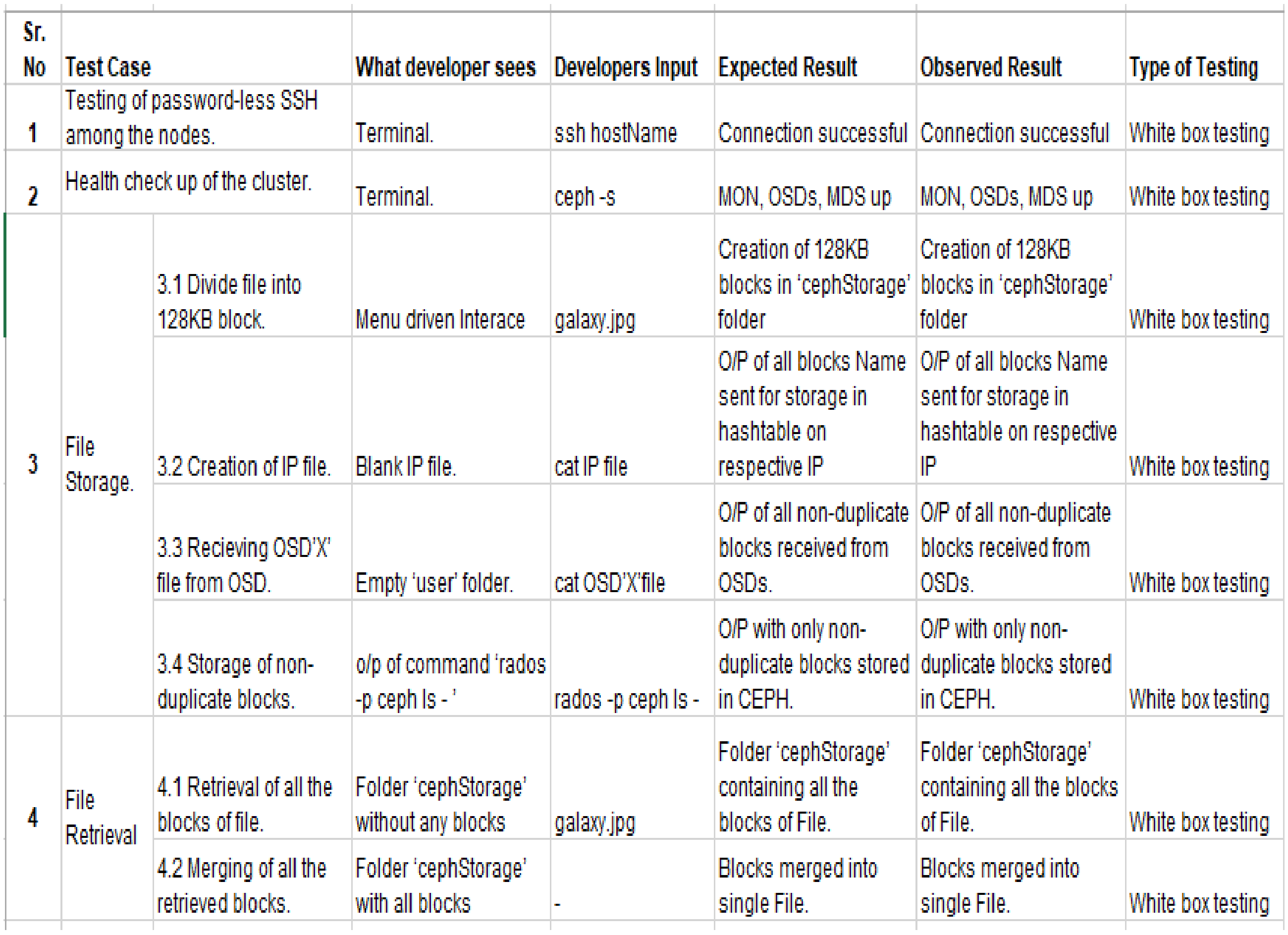
Relevant mathematics associated with the Project System Description:

- Input: Any type of file (image, video, text, spreadsheet etc).
- Output: De-deplicated storage using in-memory data structures.
- Data Structure: Hash Tables, Linked Lists, Files, Arrays.
- Functions: Data Storage, Data Retrieval, Data De-duplication.
- Success Conditions: Seamless storage and retrieval of input file.
- Failure Conditions: Failure in storage or retrieval, corrupt output file.

Laboratory assignments on Project Quality and Reliability Testing of Project Design

Appendix B

Laboratory assignments on Project Quality and Reliability Testing of Project Design



Sr. No	Test Case		What developer sees	Developers Input	Expected Result	Observed Result	Type of Testing
1	Testing of password-less SSH among the nodes.		Terminal.	ssh hostName	Connection successful	Connection successful	White box testing
2	Health check up of the cluster.		Terminal.	ceph -s	MON, OSDs, MDS up	MON, OSDs, MDS up	White box testing
3	File Storage.	3.1 Divide file into 128KB block.	Menu driven Interace	galaxy.jpg	Creation of 128KB blocks in 'cephStorage' folder	Creation of 128KB blocks in 'cephStorage' folder	White box testing
		3.2 Creation of IP file.	Blank IP file.	cat IP file	O/P of all blocks Name sent for storage in hashtable on respective IP	O/P of all blocks Name sent for storage in hashtable on respective IP	White box testing
		3.3 Recieving OSD'X' file from OSD.	Empty 'user' folder.	cat OSD'X'file	O/P of all non-duplicate blocks received from OSDs.	O/P of all non-duplicate blocks received from OSDs.	White box testing
		3.4 Storage of non-duplicate blocks.	o/p of command 'rados -p ceph ls - '	rados -p ceph ls -	O/P with only non-duplicate blocks stored in CEPH.	O/P with only non-duplicate blocks stored in CEPH.	White box testing
4	File Retrieval	4.1 Retrieval of all the blocks of file.	Folder 'cephStorage' without any blocks	galaxy.jpg	Folder 'cephStorage' containing all the blocks of File.	Folder 'cephStorage' containing all the blocks of File.	White box testing
		4.2 Merging of all the retrieved blocks.	Folder 'cephStorage' with all blocks	-	Blocks merged into single File.	Blocks merged into single File.	White box testing

Figure B.1: Testing Table

Project Planner

Appendix C

Project Planner

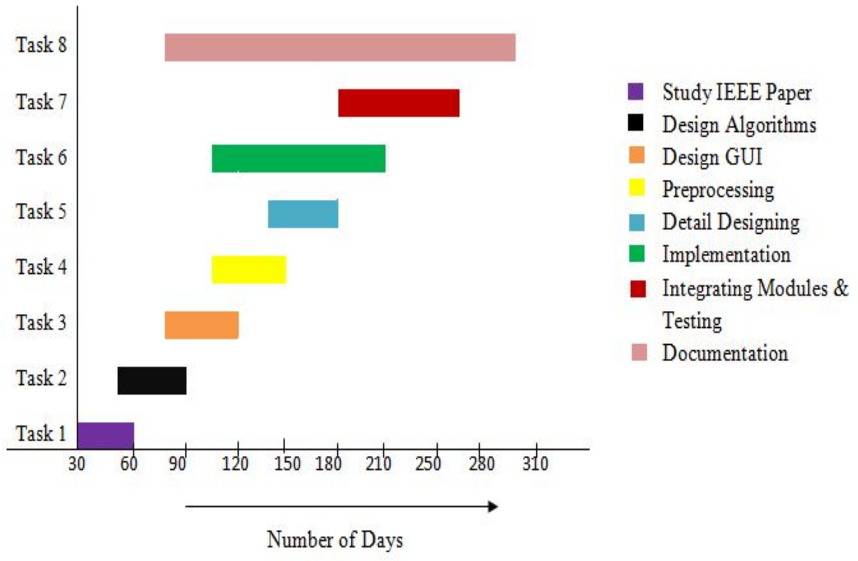


Figure C.1: Plan of Project Execution

Reviewers Comments of Paper Submitted

Appendix D

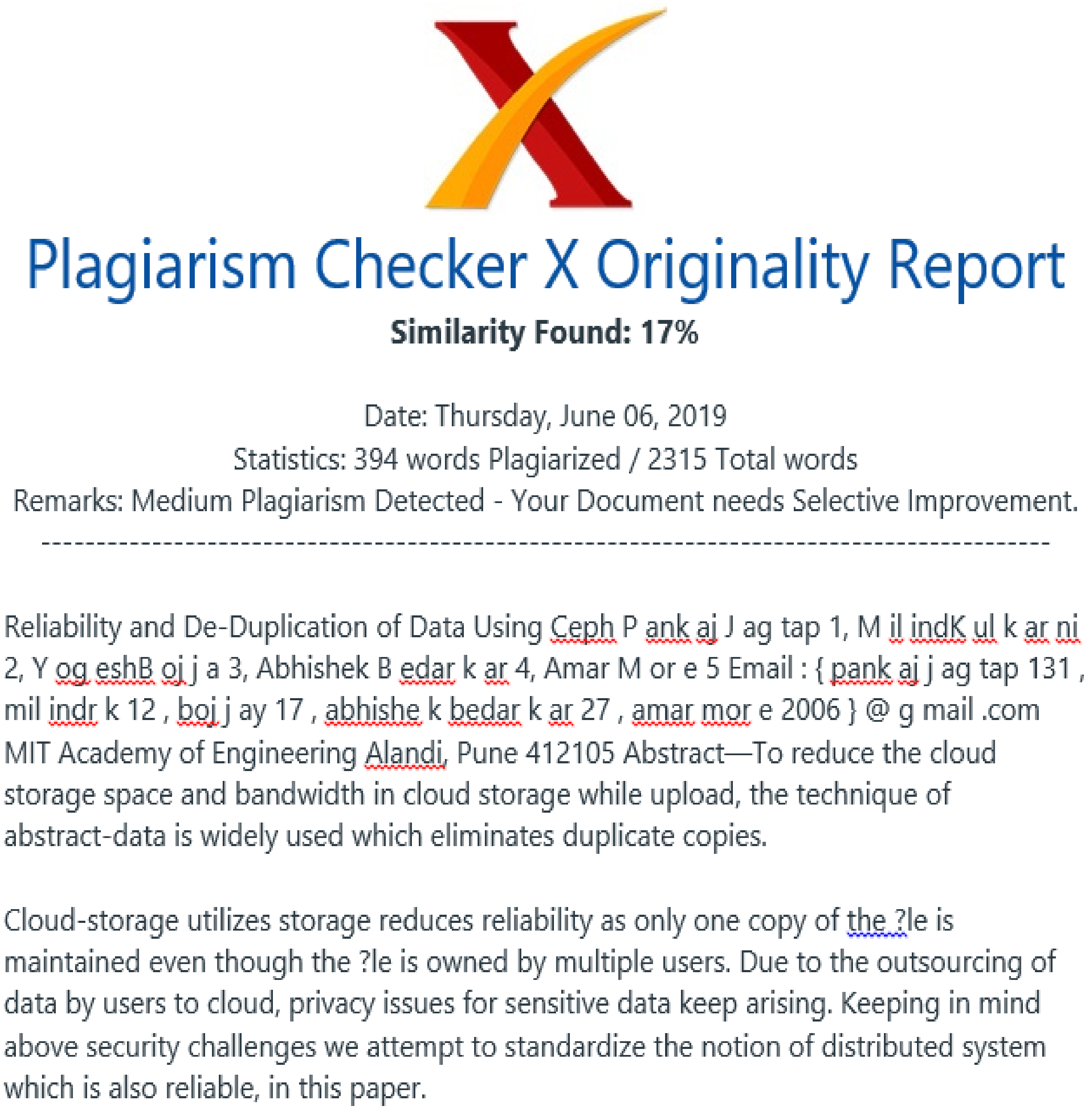
Reviewers Comments of Paper Submitted

1. Paper Title: Reliability aware data deduplication for CEPH.
2. Name of the Conference/Journal where paper submitted : NCETCET-Research Journal Of Engineering Technology And Management.
3. Paper accepted/rejected : Accepted
4. Review comments by reviewer : Good Presentation, Identify more services.
5. Corrective actions if any : Paper Was Reviewed And Submitted.

Plagiarism Report

Appendix E

Plagiarism Report



Plagiarism Checker X Originality Report

Similarity Found: 17%

Date: Thursday, June 06, 2019

Statistics: 394 words Plagiarized / 2315 Total words

Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.

Reliability and De-Duplication of Data Using Ceph P ank aj J ag tap 1, M il indK ul k ar ni 2, Y og eshB oj j a 3, Abhishek B edar k ar 4, Amar M or e 5 Email : { pank aj j ag tap 131 , mil indr k 12 , boj j ay 17 , abhishe k bedar k ar 27 , amar mor e 2006 } @ g mail .com MIT Academy of Engineering Alandi, Pune 412105 Abstract—To reduce the cloud storage space and bandwidth in cloud storage while upload, the technique of abstract-data is widely used which eliminates duplicate copies.

Cloud-storage utilizes storage reduces reliability as only one copy of the ?le is maintained even though the ?le is owned by multiple users. Due to the outsourcing of data by users to cloud, privacy issues for sensitive data keep arising. Keeping in mind above security challenges we attempt to standardize the notion of distributed system which is also reliable, in this paper.

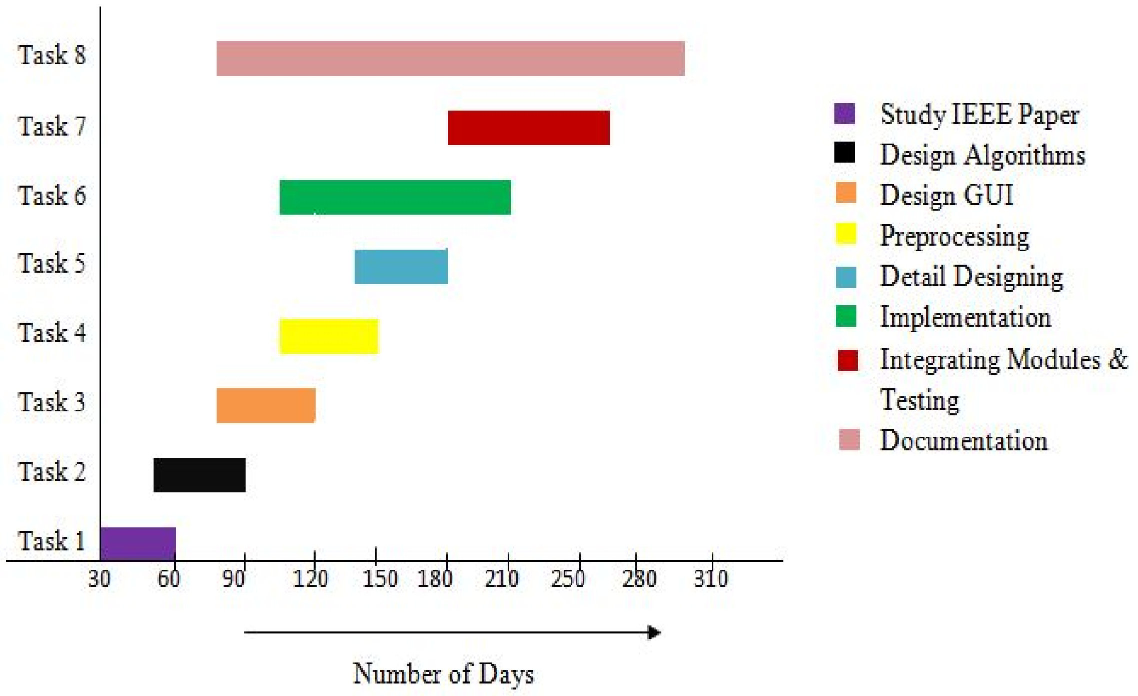
Term-II Project Laboratory Assignments

Appendix F

Term-II Project Laboratory Assignments

ASSIGNMENT NO. 1

- **Title:** Review of proposed design and necessary corrective action is taking to consider and submit publication/presentation details with review report.
- Paper selection in Research Journal Of Engineering Technology And Management.



ASSIGNMENT NO. 2

- **Title:** Project Workstation selection, Installations details with setup and Installation procedure.
- **Objectives:**
  Select the proper workstation.
- **Problem Statement:** Project Workstation selection, Installations details with setup and Installation procedure.
- **Theory:**
  The following software and platform used for implement bug triage project.We Configure our project by using below software.
  - CentOS 7
  - Ceph File System
- **Client Side:**
  CentOS 7:
  - Download The ISO Image
    To get a copy of CentOS 7 download from its source mirror. CentOS 7 is now shipping for 64 bit platforms, and currently there is no 32 bit ISO image. This is primarily due to the fact that most servers in production are 64 bit.
  - Make A bootable Drive command:
    root: dd if=/iso/CentOS-7-x86-64-DVD-1602-99.iso of=/dev/sdb

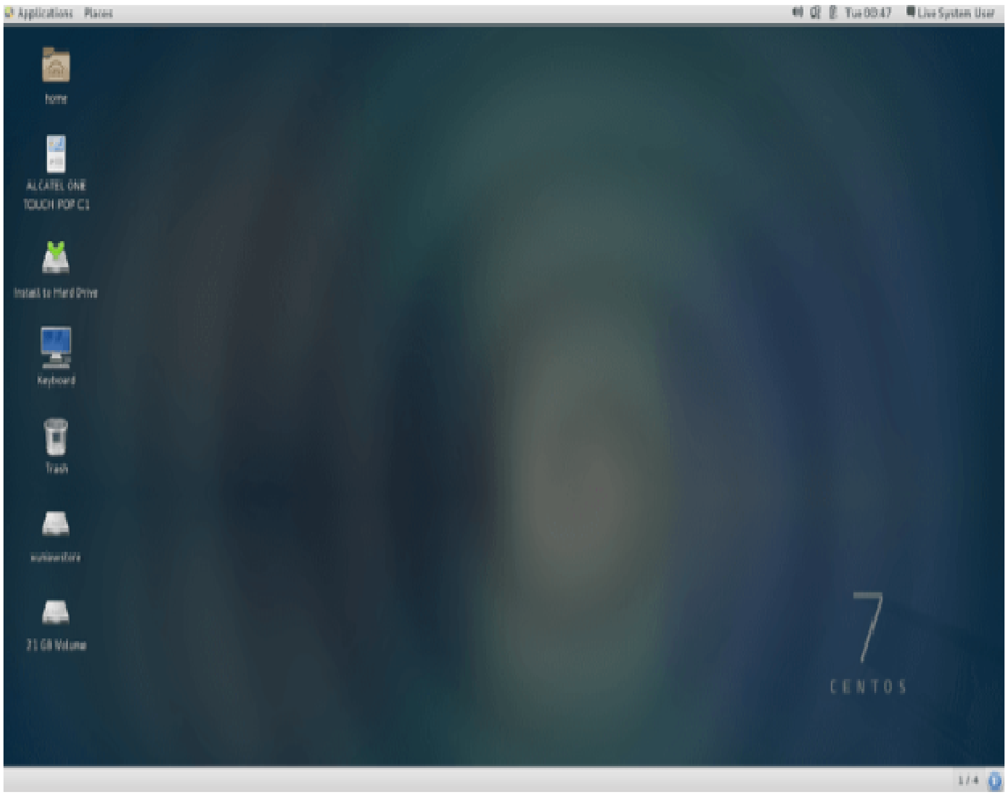


Figure F.1: 3. Begin Installation

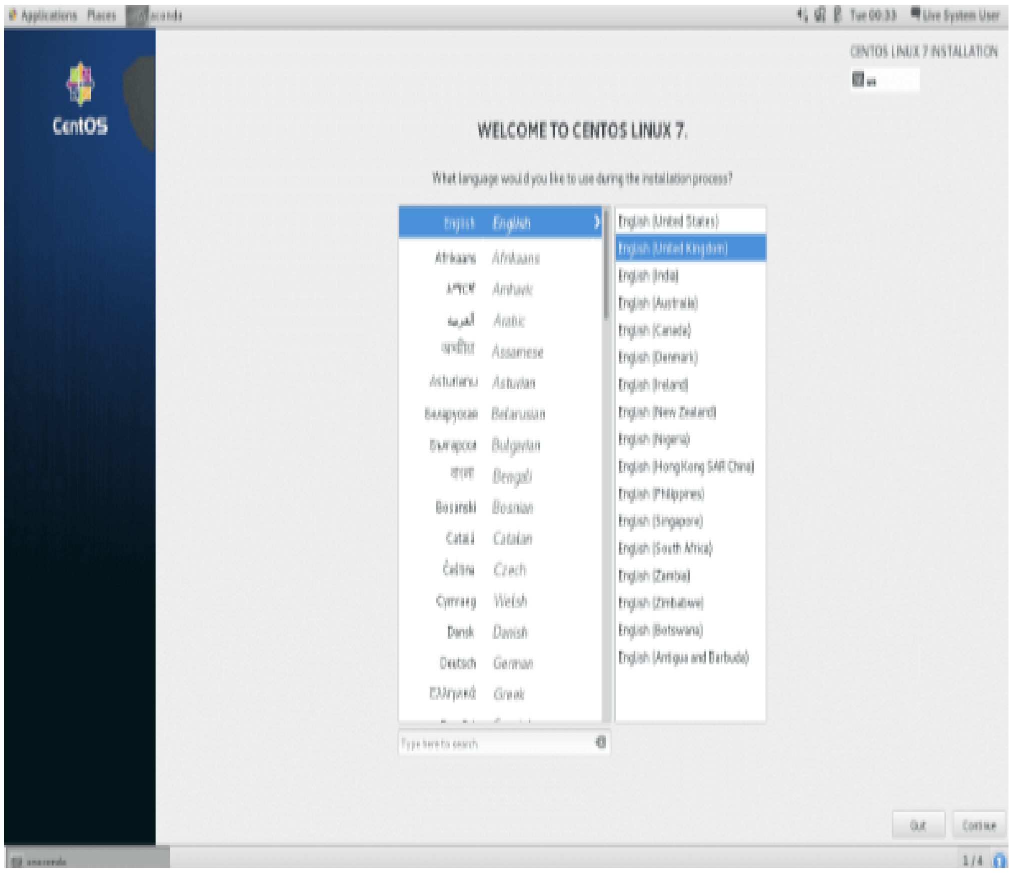
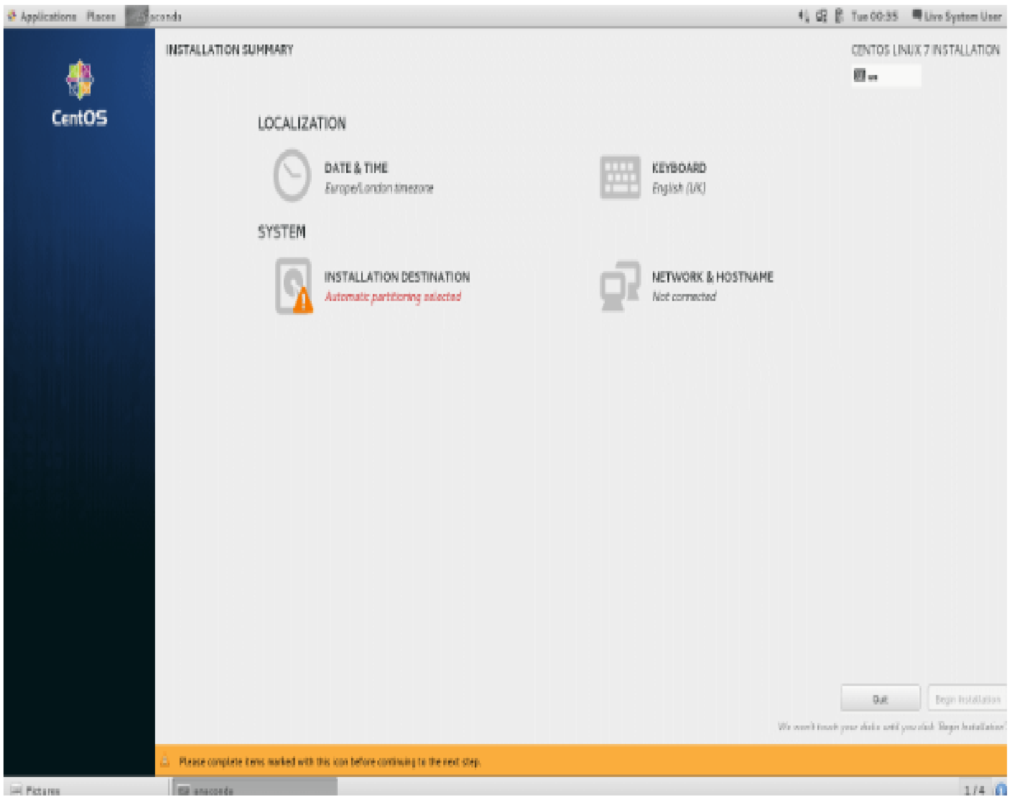


Figure F.2: 4. Select Language And Keyboard.


INSTALLATION SUMMARY

CENTOS LINUX 7 INSTALLATION

LOCALIZATION

DATE & TIME
Europe/London timezone

KEYBOARD
English (UK)

SYSTEM

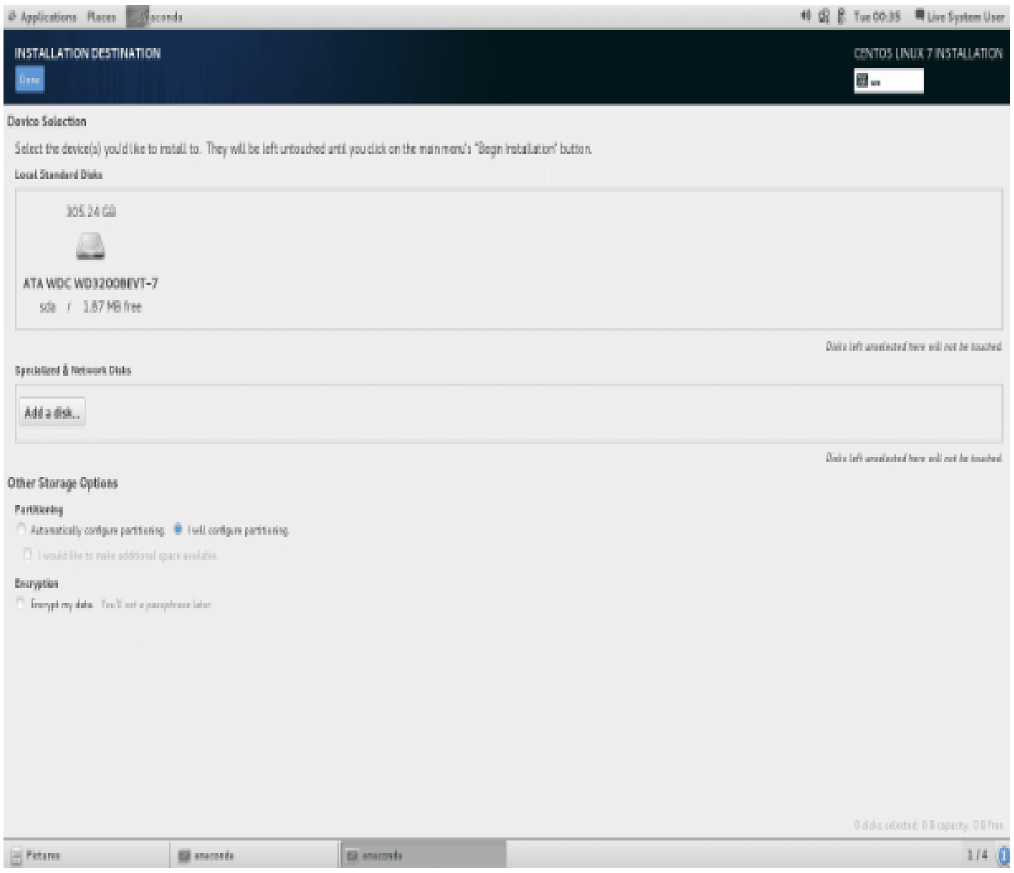
INSTALLATION DESTINATION
Automatic partitioning selected

NETWORK & HOSTNAME
Not connected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

Figure F.3: 5. Change The Installation Destination.


INSTALLATION DESTINATION

CENTOS LINUX 7 INSTALLATION

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

305.24 GB

ATA WDC WD3200BEVT-7
sda / 1.87 MB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Other Storage Options

Partitioning

Automatically configure partitioning. [✓ I will configure partitioning.]

I would like to make additional space available.

Encryption

Encrypt my data. You'll set a passphrase later.

0 disks selected; 0 B capacity; 0 B free

Figure F.4: 6. Select The Partitioning Scheme.

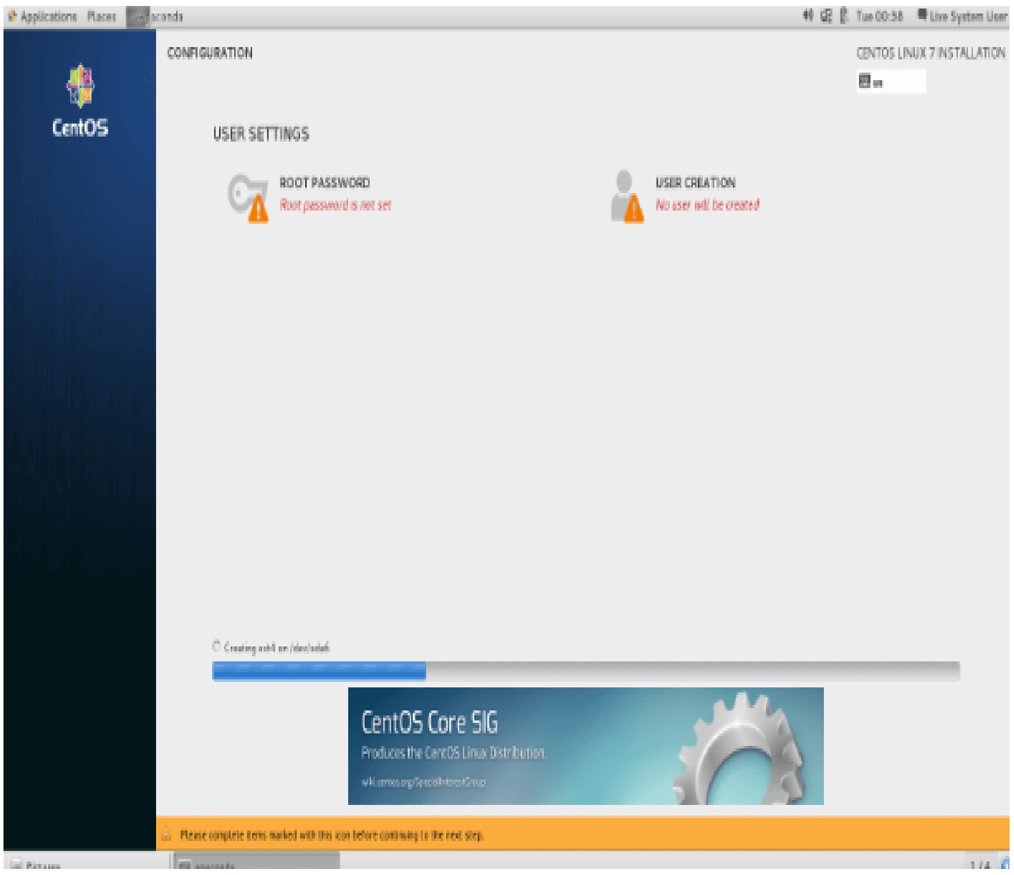

Applications Places
CentOS
CONFIGURATION
CENTOS LINUX 7 INSTALLATION
USER SETTINGS
ROOT PASSWORD
Root password is not set
USER CREATION
No user will be created
CentOS Core SIG
Produces the CentOS Linux Distribution.
Please complete items marked with this icon before continuing to the next step.

Figure F.5: 7. Finish Installation

- **CEPH Installation Steps:**
  - Configure All Nodes
    - ∗ Create a Ceph User:
      useradd -d /home/cephuser -m cephuser
    - ∗ Install and Configure NTP
      yum install -y ntp ntpdate ntp-doc
      ntpdate 0.us.pool.ntp.org
      hwclock –systohc
      systemctl enable ntpd.service
      systemctl start ntpd.service
    - ∗ Install Open-vm-tools
    - ∗ Disable SELinux
      sed -i 's/SELINUX=enforcing/SELINUX=disabled/g'
      /etc/selinux/config
    - ∗ Configure Hosts File
  - Configure the SSH Server
    ssh-keygen

- Configure Firewalld
- Configure the Ceph OSD Nodes
  sudo parted -s /dev/sdb mklabel gpt mkpart primary xfs 0
  sudo mkfs.xfs /dev/sdb -f
- Build the Ceph Cluster
  - Install ceph-deploy on the ceph-admin node
    sudo rpm -Uhv http://download.ceph.com/rpm-jewel/el7/noarch/ceph-release-1-1.el7.noarch.rpm
    sudo yum update -y
    sudo yum install ceph-deploy -y
  - Create New Cluster Config
  - Install Ceph on All Nodes
  - Adding OSDs to the Cluster
- Testing the Ceph setup

- **Conclusion:**
  CentOS 7 has greatly improved from version 6.5 and now is easier to adopt it as a Desktop OS compared to its predecessor. For those that probably cannot keep up with Fedora releases every 6 months, CentOS 7 is a good consideration.

ASSIGNMENT 3

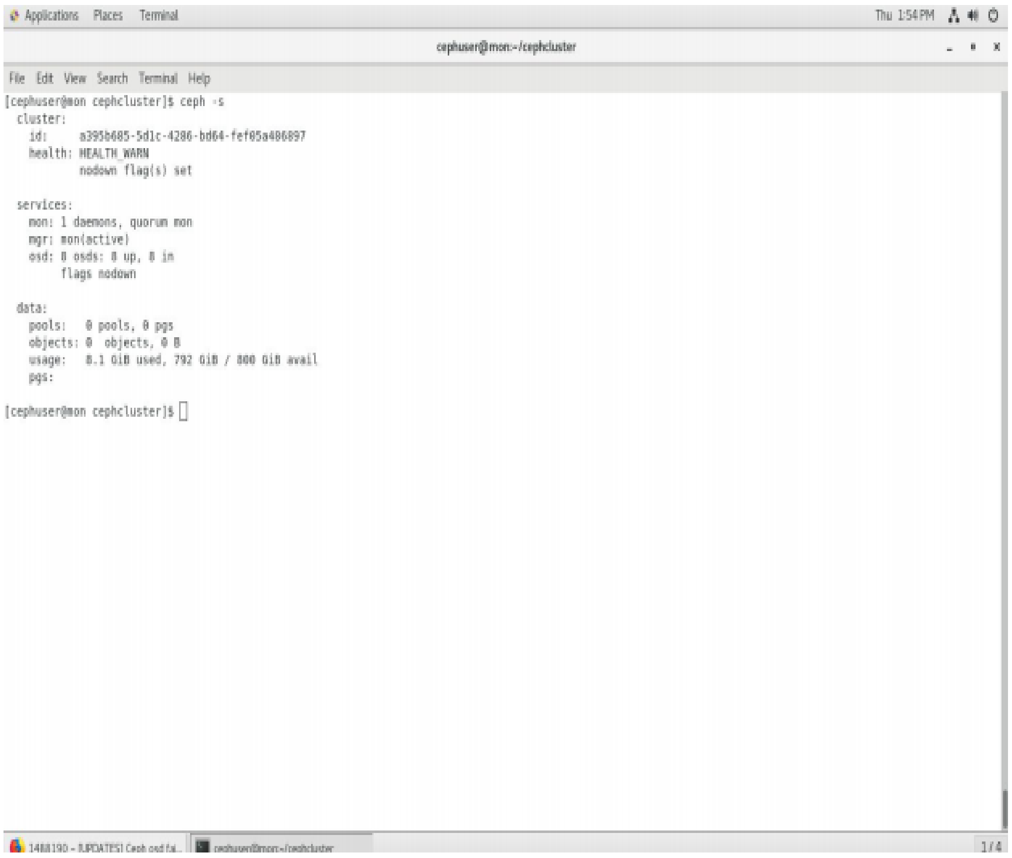
- **Title:** Programming of the project functions as per UML diagrams mentioned in earlier submission.
- **Objectives:**
  Study the different function used in project
  Enlist the function used for the interface and UI design.
- **Problem Statement:**
  Programming of the project functions as per UML diagrams as per 1 st Term term-work submission using corrective actions recommended in Term-I assessment of Term-work.
- **Theory:**
  The following are the various functions as well as agents in our Project as per previously submitted UML diagrams.
  - Client:
    This is the user whose data is uploaded on our distributed storage systems. The data can be of any type and supports all types of extensions e.g. Images(jpg, jpeg), pdfs, documents(txt, docx, pptx) and audio/video files(mp3, mp4 etc).
  - Monitor (MON):
    This is a server, all the data goes through here before it can be distributed across the cluster for storage.
  - Object Storage Devices (OSD):
    These are servers that actually store the data. The data stored here is in the form of blocks. These blocks are replicated and distributed across the cluster to ensure reliability.
  - Meta Data Server (MDS):
    This server stores the Meta Data of the data that is stored on the cluster by the client. This metadata includes name of files, their size, location in cluster, permissions etc.
  - CRUSH:
    This is the algorithm that is used implicitly by CEPH for distributing the data over the cluster also the data is distributed that one of the OSDs failure wont affect any loss in data.

- Hash Tables:
  These are created on both the Monitor and OSD side, to main the metadata and duplication/replication counter respectively. The hashtables on OSD are distributed in nature to support high values of the hash values generated and compressed by our algorithm.
- STORE function:
  This programming function is used for dividing an input file from the user into block of 128 Kbs and then calculating its 256 bit hash values using SHA 256 algorithm. These blocks are then sent to the OSDs where hashtables are maintained for avoiding duplicacy and thus resulting in deduplication of data i.e. storing only those blocks that are unique.
- RETREIVE function:
  This programming function is used for generating the output file requested by the client, the file must be previously present on the cluster to be merged. This file hashtable on the monitor is traversed which stores the metadata of the file and retrieves the blocks from CEPH to merge them later on to get the original file uploaded by the client.

• **Conclusion:**
Thus we studied the UML diagrams and implied them programmatically in the Project.

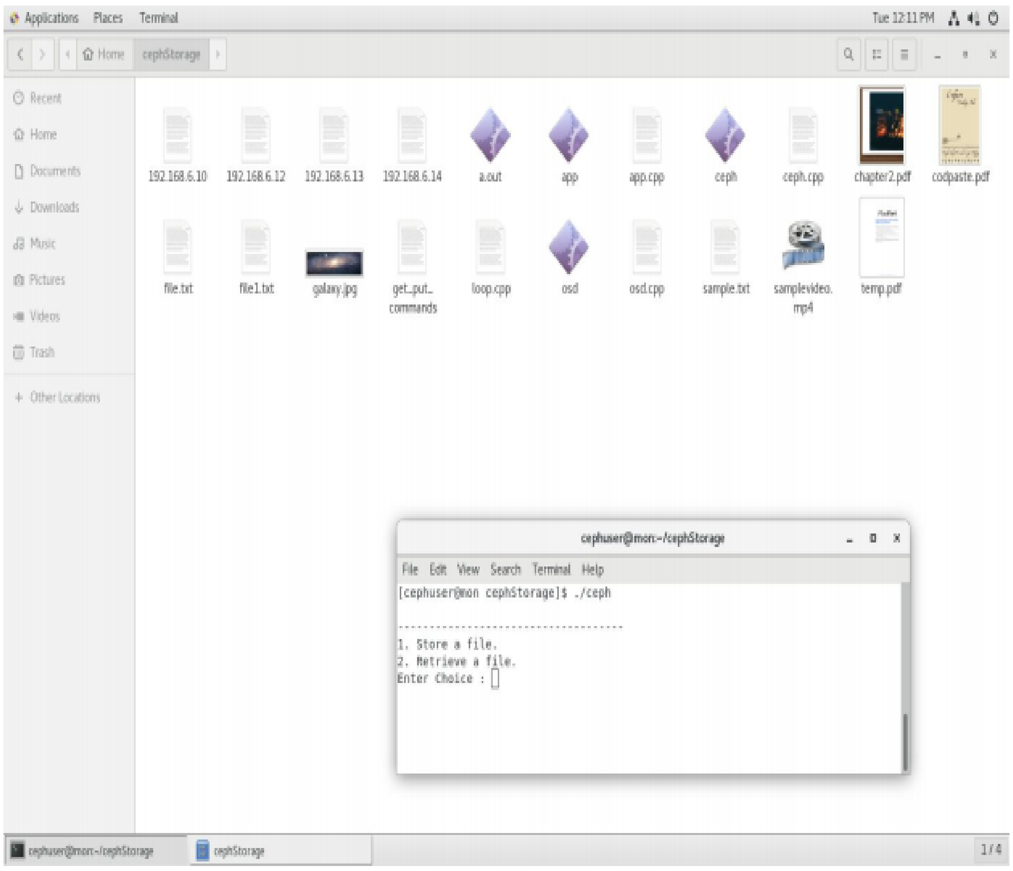
ASSIGNMENT 4

- **Title:**
  Interface and GUI project, after necessary corrective action taking as per submission.
- **Objective:**
  To study GUI of project.
- **Problem Statement:**
  Interface and GUI of project, after necessary corrective action taking as per submission.


Applications Places Terminal
Thu 1:54 PM
cephuser@mon:~/cephcluster
File Edit View Search Terminal Help
1488190 – [UPDATES] Ceph osd fai...
cephuser@mon:~/cephcluster
1/4

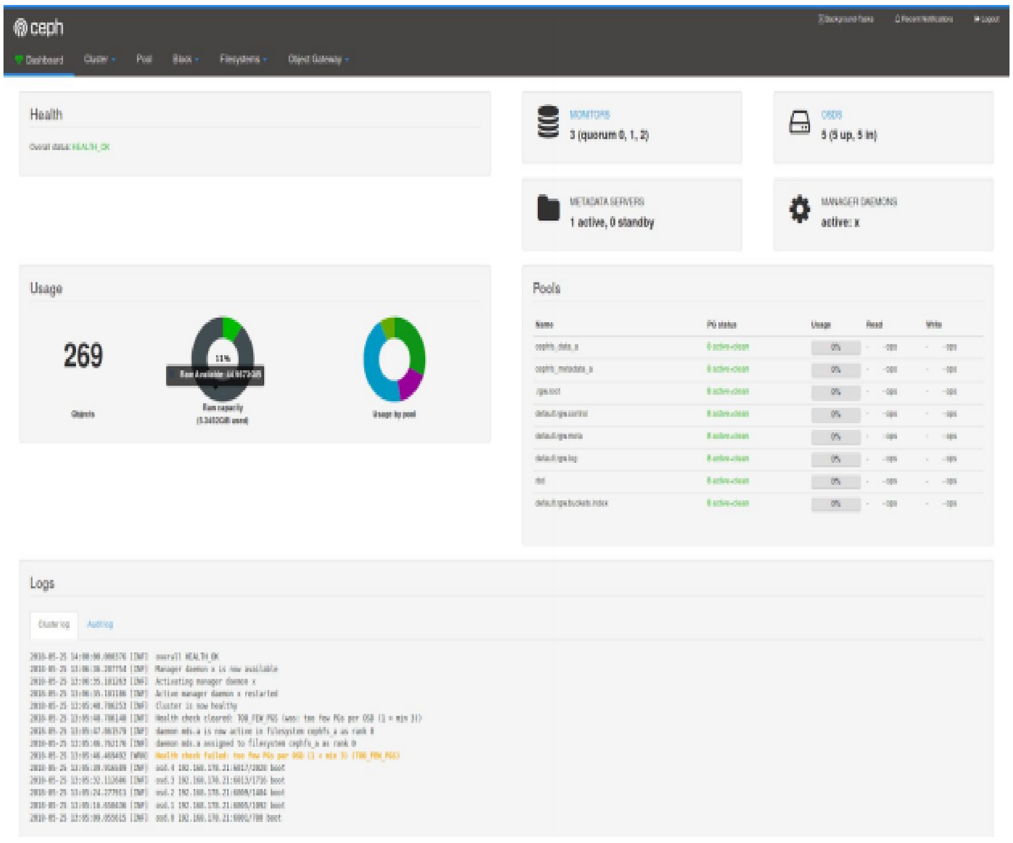
```
[cephuser@mon cephcluster]$ ceph -s
  cluster:
    id:     a395b685-5d1c-4286-bd64-fef05a486897
    health: HEALTH_WARN
            nodown flag(s) set

  services:
    mon: 1 daemons, quorum mon
    mgr: mon(active)
    osd: 8 osds: 8 up, 8 in
         flags nodown

  data:
    pools:   0 pools, 0 pgs
    objects: 0  objects, 0 B
    usage:   8.1 GiB used, 792 GiB / 800 GiB avail
    pgs:

[cephuser@mon cephcluster]$ 
```

Figure F.6: 1. Terminal view



```
[cephuser@mon cephStorage]$ ./ceph

------------------------------------
1. Store a file.
2. Retrieve a file.
Enter Choice :
```

Figure F.7: 2. Input options



Overall status: HEALTH_OK

MONITORS
3 (quorum 0, 1, 2)

OSDS
5 (5 up, 5 in)

METADATA SERVERS
1 active, 0 standby

MANAGER DAEMONS
active: x

Name	PG status	Usage	Read	Write
cephfs_data_a	8 active+clean	0%	- ops	- ops
cephfs_metadata_a	8 active+clean	0%	- ops	- ops
.rgw.root	8 active+clean	0%	- ops	- ops
default.rgw.control	8 active+clean	0%	- ops	- ops
default.rgw.meta	8 active+clean	0%	- ops	- ops
default.rgw.log	8 active+clean	0%	- ops	- ops
rbd	8 active+clean	0%	- ops	- ops
default.rgw.buckets.index	8 active+clean	0%	- ops	- ops

```
2018-05-25 14:00:00.000376 [INF]  overall HEALTH_OK
2018-05-25 13:06:38.207758 [INF]  Manager daemon x is now available
2018-05-25 13:06:35.181263 [INF]  Activating manager daemon x
2018-05-25 13:06:35.181186 [INF]  Active manager daemon x restarted
2018-05-25 13:05:48.786253 [INF]  Cluster is now healthy
2018-05-25 13:05:48.786148 [INF]  Health check cleared: TOO_FEW_PGS (was: too few PGs per OSD (1 < min 3))
2018-05-25 13:05:47.863579 [INF]  daemon mds.a is now active in filesystem cephfs_a as rank 0
2018-05-25 13:05:46.792176 [INF]  daemon mds.a assigned to filesystem cephfs_a as rank 0
2018-05-25 13:05:46.469492 [WRN]  Health check failed: too few PGs per OSD (1 < min 3) (TOO_FEW_PGS)
2018-05-25 13:05:39.958589 [INF]  osd.4 192.168.170.21:6817/2828 boot
2018-05-25 13:05:32.112646 [INF]  osd.3 192.168.170.21:6813/1716 boot
2018-05-25 13:05:24.277913 [INF]  osd.2 192.168.170.21:6809/1484 boot
2018-05-25 13:05:16.650436 [INF]  osd.1 192.168.170.21:6805/1092 boot
2018-05-25 13:05:09.055615 [INF]  osd.0 192.168.170.21:6801/700 boot
```

Figure F.8: 3. Dashboard

- **Conclusion:**
  Thus we studied the GUI of project successfully.

Assignment 5

- **Title:**
  Selection of tools and various cases for project perform.
- **Objective:**
  To study different testing methods for project implementation.
- **Problem Statement:**
  Selection of tools and various cases for project perform.
- **TYPE OF TESTING USED**

  1. **Unit Testing**
     Unit testing concentrates verification on the smallest element of the program the module. Using the detailed design description important control paths are tested to establish errors within the bounds of the module.In this system each sub module is tested individually as per the unit testing such as campaign, lead, contact etc are tested individually. Their input field validations are tested.
  2. **Integration testing**
     Once all the individual units have been tested there is a need to test how they were put together to ensure no data is lost across interface, one module does not have an adverse impact on another and a function is not performed correctly. After unit testing each and every sub module is tested with integrating each other.

- **System testing for the current system:**
  In this level of testing we are testing the system as a whole after integrating all the main modules of the project. We are testing whether system is giving correct output or not. All the modules were integrated and the flow of information among different modules was checked. It was also checked that whether the flow of data is as per the requirements or not. It was also checked that whether any particular module is non-functioning or not i.e. once the integration is over each and every module is functioning in its entirety or not.

  In this level of testing we tested the following: -

  – Whether all the forms are properly working or not.

  - Whether all the forms are properly linked or not.
  - Whether all the images are properly displayed or not.
  - Whether data retrieval is proper or not.

- **Test Cases and Test Cases Result:**
  Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

  - **Test Cases:**
    file read write command : rados -p foo -s mysnap get myobject blah.txt.old

    1. file name write
    2. correct file name read
    3. Wrong file name read
    4. file name write

- **Test Design Techniques:**
  Typical black-box test design techniques include:

  - Decision table testing
  - All-pairs testing
  - State transition Analysis
  - Equivalence partitioning
  - Boundary value analysis
  - Causeeffect graph
  - Error guessing

- **Advantages:**

  - Efficient when used on large systems.
  - Since the tester and developer are independent of each other, testing is balanced and unprejudiced.
  - Tester can be non-technical.
  - There is no need for the tester to have detailed functional knowledge of system.

  - Tests will be done from an end user's point of view, because the end user should accept the system. (This testing technique is sometimes also called Acceptance testing.)
  - Testing helps to identify vagueness and contradictions in functional specifications.
  - Test cases can be designed as soon as the functional specifications are complete.

- **Disadvantages:**
  - Test cases are challenging to design without having clear functional specifications.
  - It is difficult to identify tricky inputs if the test cases are not developed based on specifications.
  - It is difficult to identify all possible inputs in limited testing time. As a result, writing test cases may be slow and difficult.
  - There are chances of having unidentified paths during the testing process.
  - There is a high probability of repeating tests already performed by the programmer.

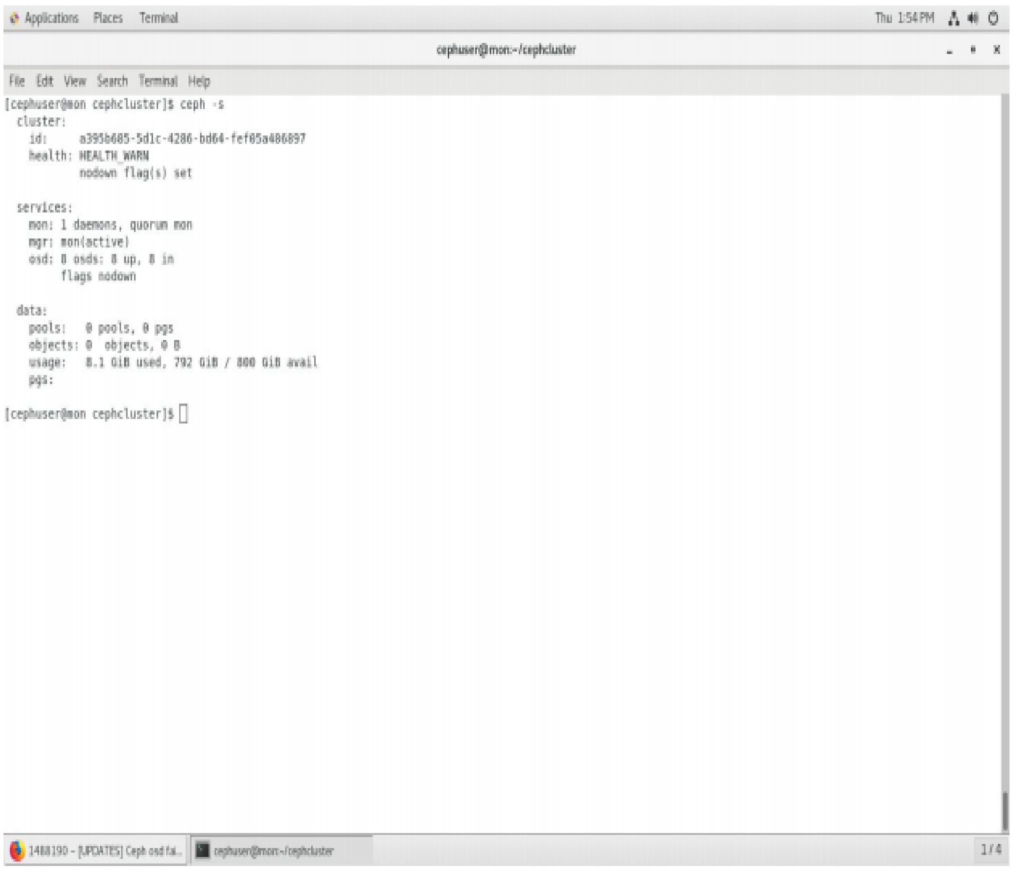
- **Conclusion:**
  We have successfully implemented different test cases.

Assignment 6

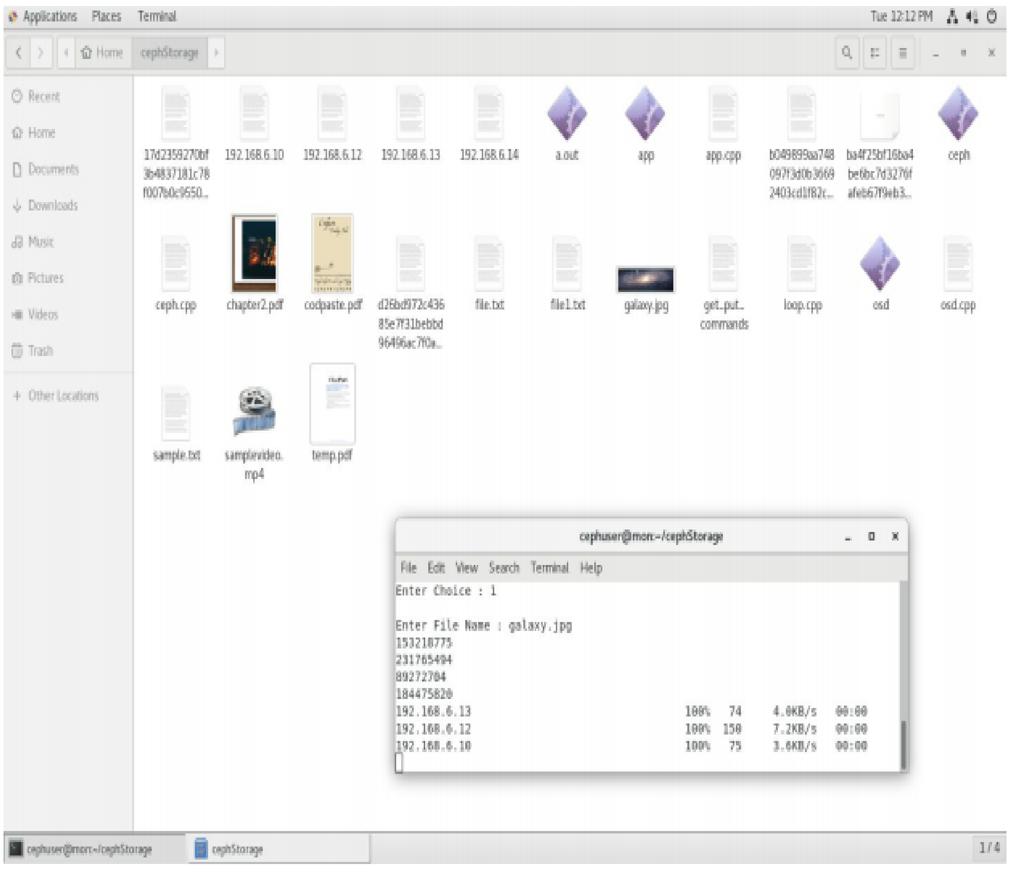
- **Title:**
  Result as per various test case perform for project work and submit result charts, table of result and graph including reliability testing.

- **Objective:**
  To analyze results of different test case . To measure performance of cloud controller of ceph cloud.

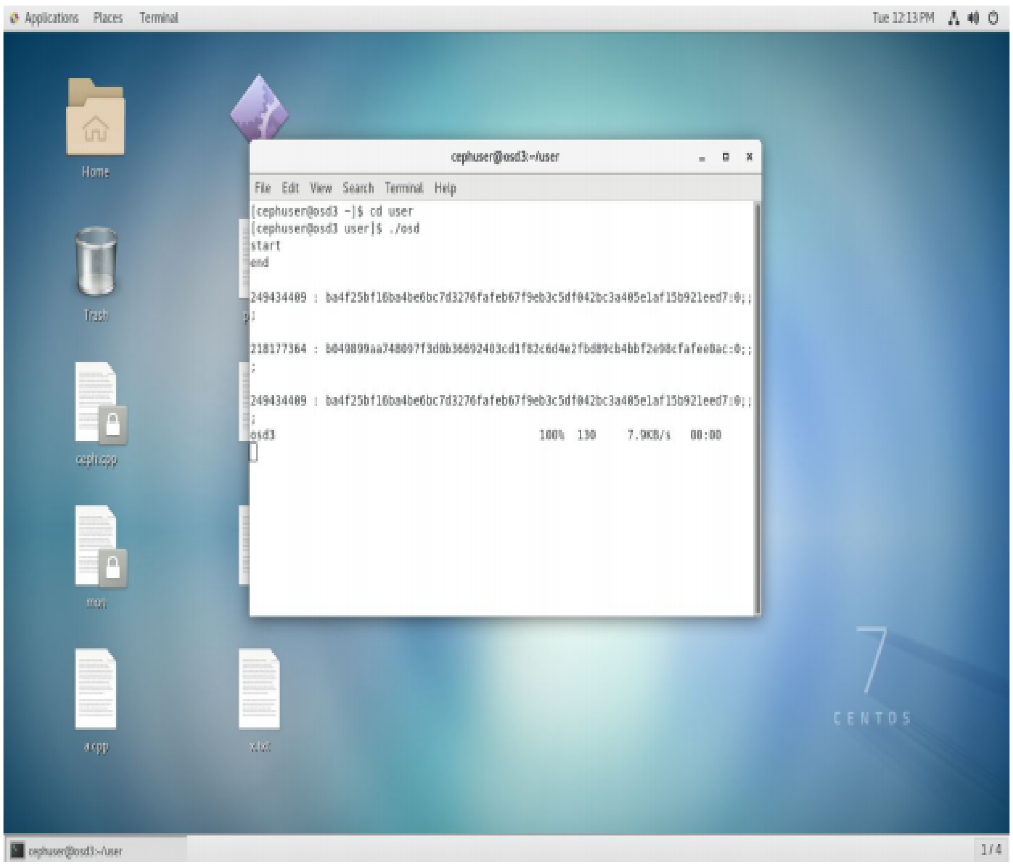
- **Problem Statement:**
  Result as per various test case perform for project work and submit result charts ,table of result and graph including reliability testing.


Applications Places Terminal
Thu 1:54 PM
cephuser@mon:~/cephcluster
File Edit View Search Terminal Help
1488190 – [UPDATES] Ceph osd fai...
cephuser@mon:~/cephcluster
1 / 4

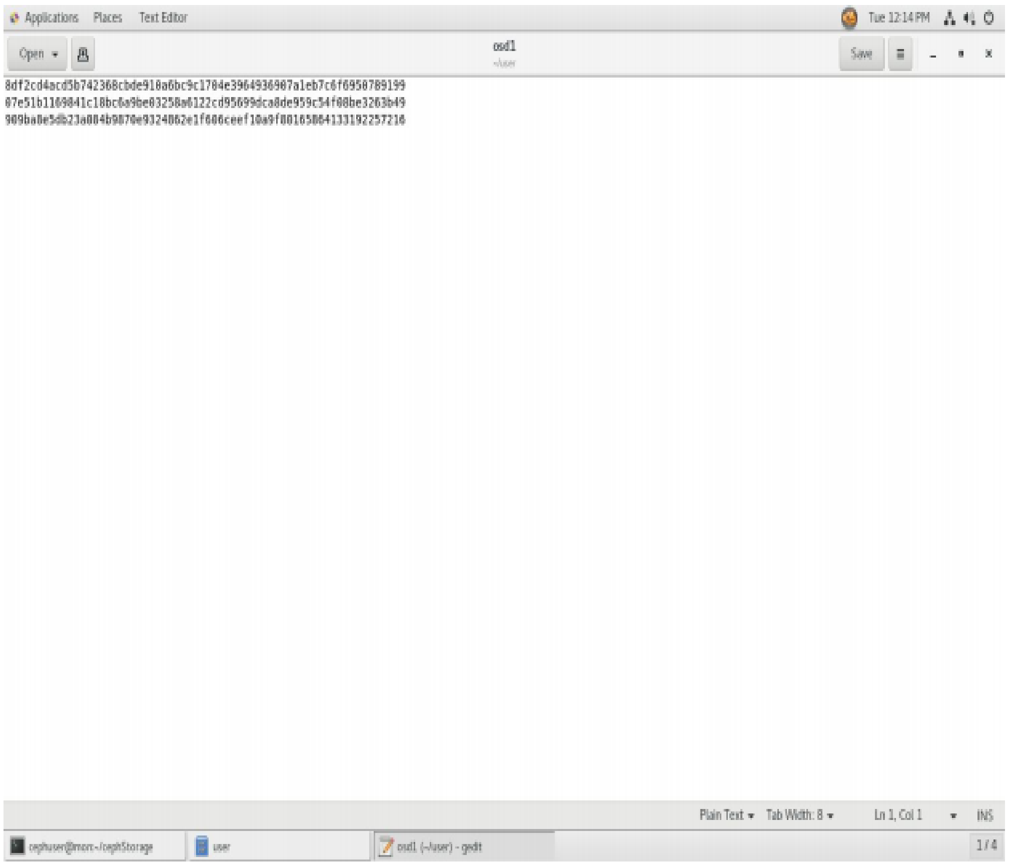
```
[cephuser@mon cephcluster]$ ceph -s
  cluster:
    id:     a395b685-5d1c-4286-bd64-fef05a486897
    health: HEALTH_WARN
            nodown flag(s) set

  services:
    mon: 1 daemons, quorum mon
    mgr: mon(active)
    osd: 8 osds: 8 up, 8 in
         flags nodown

  data:
    pools:   0 pools, 0 pgs
    objects: 0  objects, 0 B
    usage:   8.1 GiB used, 792 GiB / 800 GiB avail
    pgs:

[cephuser@mon cephcluster]$ 
```



```
Enter Choice : 1

Enter File Name : galaxy.jpg
153210775
231765494
89272704
184475820
192.168.6.13                                   100%   74     4.0KB/s   00:00
192.168.6.12                                   100%  150     7.2KB/s   00:00
192.168.6.10                                   100%   75     3.6KB/s   00:00
```



```
[cephuser@osd3 ~]$ cd user
[cephuser@osd3 user]$ ./osd
start
end

249434409 : ba4f25bf16ba4be6bc7d3276fafeb67f9eb3c5df042bc3a405e1af15b921eed7:0;;
;

218177364 : b049899aa748097f3d0b36692403cd1f82c6d4e2fbd89cb4bbf2e98cfafee0ac:0;;
;

249434409 : ba4f25bf16ba4be6bc7d3276fafeb67f9eb3c5df042bc3a405e1af15b921eed7:0;;
;
osd3                                     100%  130     7.9KB/s   00:00
```

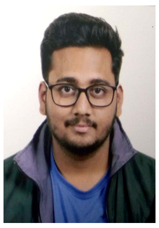


```
8df2cd4acd5b742368cbde910a6bc9c1704e3964936907a1eb7c6f6950789199
07e51b1109841c10bc6a9be03258a6122cd95699dca0de959c54f08be3263b49
909ba0e5db23a004b9870e9324062e1f606ceef10a9f001650041331922572l0
```

- **Conclusion:**
  We have evaluated result as per various test case perform for project work and submitted result charts ,table of result and graph including reliability testing.

Appendix G

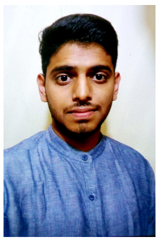
Information of Project Group Members



1. Name : Yogesh Bojja
2. Date of Birth : 13/11/1996
3. Gender : Male
4. Permanent Address : Vadgaon Sheri, Pune-411014.
5. E-Mail : bojjay17@gmail.com
6. Mobile/Contact No. : 9922422497
7. Placement Details : Higher Studies
8. Paper Published : Paper selection in Research Journal Of Engineering Technology And Management.

1. Name : Pankaj Jagtap
2. Date of Birth : 16/10/1997
3. Gender : Male
4. Permanent Address : Vadgaon Sheri, Pune-411014.
5. E-Mail : pankajjagtap131@gmail.com
6. Mobile/Contact No. : 8446307427
7. Placement Details : Placed
8. Paper Published : Paper selection in Research Journal Of Engineering Technology And Management.

1. Name : Abhishek Bedarkar
2. Date of Birth : 27/05/1997
3. Gender : Male
4. Permanent Address : Akola-444101
5. E-Mail : abhishekbedarkar27@gmail.com
6. Mobile/Contact No. : 9850933049
7. Placement Details : Placed
8. Paper Published : Paper selection in Research Journal Of Engineering Technology And Management.



1. Name : Milind Kulkarni
2. Date of Birth : 12/12/1997
3. Gender : Male
4. Permanent Address : Dhanori, Pune-411015.
5. E-Mail : milindrk12@gmail.com
6. Mobile/Contact No. : 7709743895
7. Placement Details : Placed
8. Paper Published : Paper selection in Research Journal Of Engineering Technology And Management.

Appendix H

Sponsorship Details

Project is not sponsored.

Appendix I

Publication Details

Reliability and De-Duplication of Data Using Ceph

$PankajJagtap^1$, $MilindKulkarni^2$, $YogeshBojja^3$, $AbhishekBedarkar^4$, $AmarMore^5$
Email :{pankajjagtap131, milindrk12, bojjay17, abhishekbedarkar27, amarmore2006} @gmail.com
MIT Academy of Engineering
Alandi, Pune 412105

***Abstract*—To reduce the cloud storage space and bandwidth in cloud storage while upload, the technique of abstract-data is widely used which eliminates duplicate copies. Cloud-storage utilizes storage reduces reliability as only one copy of the file is maintained even though the file is owned by multiple users. Due to the outsourcing of data by users to cloud, privacy issues for sensitive data keep arising. Keeping in mind above security challenges we attempt to standardize the notion of distributed system which is also reliable, in this paper. Multiple cloud servers distribute data chunks among themselves in our new proposed distributed system. By the introduction deterministic secret sharing scheme in distributed storage system, we achieved the security requirements of data being confidential and tag consistent. Security analysis and terms specified in the proposed security model shows that, our system is secure. We demonstrate that the experienced overhead is very limited in a environment that is realistic and we implemented the proposed system, as a proof of the concept.**

I. Introduction

Data is most valuable thing in world as it stores the past, present and the future of the entire universe. It is the most crucial element. Day by day the amount data that is collected is abundant. We need to handle this data efficiently and with proper care. Data which is in large enterprises is growing at a rate of 40 to 60 percent every year.Generally companies are doubling their data footprint each year. IDC analysts estimated that there was 54.4 exabytes of total digital data worldwide in the year 2000. By 2007 this reached 295 exabytes and by 2014 it reached 8,591 exabytes. For the growing data explosion of this planet Ceph is one of the solution.

Ceph is one of the solution to handle the storage in more reliable manner. The unstructured data is stored with help of Ceph it is an open source project which provides unified storage solutions and is software defined. It is massively scalable and high performing without any single point of failure as it is a distributed storage system. Ceph is open, scalable and has distributed nature, due to this it is getting most of the buzz in storage industry. Ceph provides a cloud storage solution. Scalability, reliability are the one Ceph's key features. Objects are building blocks of Ceph. Clusters in Ceph store any format of data in the form of objects, may it be a block, object, or file.

II. Background

Distributed scale-out storage systems can be categorized according to how they share information: Centralized or de-centralized also called shared nothing. In Ceph a decentralized storage approach is accepted.The Ceph file system has three main components: Client, each instance of which exposes a near-POSIX filesystem interface to a host or process; a cluster of Object Storage Devices also called OSD's, which collectively stores all data and metadata; and a metadata server cluster, which manages the namespace. MDS stores connecting information between a data and a storage and in the decentralized storage, a CRUSH algorithm determines the placement of a data. All the client requests a file which are translated by an MDS and CRUSH algorithm. Ceph directly addresses the issue of scalability while simultaneously achieving high performance, reliability and availability through three fundamental design features: decoupled data and meta-data, dynamic distributed metadata management, and re-liable autonomic distributed object storage. In this paper, we aimed at a distributed reliable file system with active deduplication support.

III. Deduplication Range

Applying deduplication to already existing distributed storage systems is complex because we need to deduplicate complete data while keeping the rules of the existing systems. One of the most straight forward ways that can deduplicate without any violation against the policies of underlying storage system is to individually apply the deduplication to each single node by a leveraging block level deduplication solution.Data deduplication can be operated at the file or block level. File deduplication eliminates duplicate files, but is not an efficient means of deduplication.File-level data deduplication compares a file to be backed up or archived with copies that are already stored. This is done by checking its attributes against an index. If the file is unique, it is stored and the index is updated; if not, only a pointer to the existing file is stored. The result is that only one instance of the file is saved, and subsequent copies are replaced with a stub that points to the original file. Block-level deduplication looks within a file and saves unique iterations of each block. All the blocks are broken into chunks with the same fixed length which is 128KB. Each chunk of data is processed using a hash algorithm, here SHA-256.Hash collisions are a potential problem with deduplication. When a piece of data receives a hash number, that number is then compared with the index of other existing hash numbers. If that hash number is already in the index, the piece of data is considered a duplicate and does not need to be stored again. Otherwise, the new hash number is added to the index and the new data is stored. In rare cases, the hash algorithm may produce the same hash number for two different chunks of data. When a hash collision occurs, the system won't store the new data because it sees that its hash number already exists in the index. This is called a false positive, and it can result in data loss. Some vendors combine hash algorithms to reduce the possibility of a hash collision. Some vendors are also examining metadata to identify data and prevent collisions.We compare SHA-256 values to deal with collision and then apply chaining.

IV. Problem and Key Idea

Though Ceph is great for Object Storage when it comes to huge amount of data, storing it and replicating it among other nodes for backup takes up huge memory.By default ceph replicates thrice however it can be deliberatly modified. Though this may seem quite insignificant at first while considering applications like WhatsApp, Facebook and the data generated by them on day to day is lot more repetitive in nature and thus replicating it makes it even worse situation. De-duplicating this data helps a lot on such situations.
De-duplication in Ceph is basically managing the MetaData Server so that the data entries are managed without actually storing the data itself.Ceph has no active support for data deduplication.Following are practical problems to apply deduplication on distributed file system Ceph.

1) Detection of duplicate block in file.
2) Compatibility between the newly applied deduplication metadata and exiting metadata.
3) Effectively handling CRUD operations of file.

A. Detection of duplicate block in file:

A simple and straightforward approach is to directly apply file level comparison however Block level will be an efficient option.The most popular approach for determining duplicates is to assign an identifier to a chunk of data, using a hash algorithm, for example, that generates a unique ID or "fingerprint" for that block. The unique ID is then compared with a central index. If the ID exists, then the data segment has been processed and stored before. Therefore, only a pointer to the previously stored data needs to be saved. If the ID is new, then the block is unique. The unique ID is added to the index and the unique chunk is stored. However this may lead to collision on hash table.

B. Compatibility between the newly applied deduplication metadata and exiting metadata:

Ceph works on core algorithm called CRUSH which is resposibile for file or block storage.The CRUSH mechanism works in such a way that the metadata computation workload is distributed and performed only when needed i.e crush lookup.For a read-and-write operation to Ceph clusters, clients first contact a Ceph monitor and retrieve a copy of the cluster map. The cluster map helps clients know the state and configuration of the Ceph cluster. The data is converted to objects with object and pool names/IDs. The object is then hashed with the number of placement groups to generate a final placement group within the required Ceph pool. The calculated placement group then goes through a CRUSH lookup to determine the primary OSD location to store or retrieve data. After computing the exact OSD ID, the client contacts this OSD directly and stores the data.However we have proposed a change in its processing method by intrupting the flow of block storage.This is acheived by adding a additional layer on top of CRUSH to eliminate possibiliy of deduplication.Before client connecting to monitor, it is make sure that deduplication is performed on file.

C. Effectively handling CRUD operations of file:

In this proposed method we have chosen block size deduplication, hence CRUD operation are complex to handle reliably.Shared files are provided through reference instead of copy this make process prone to errors.User sharing file across multiple clients may change file content which are not supposed to change client file.Deletion of file by one client should not affect others.

V. Implementation Notes

We implemented the proposed deduplication method ontop of Ceph:

A. Duplication detection:

Whenever client sends file to ceph, Top layer splits file into chunk size of 64KB.For each chunk of file SHA(256) value is calculated. Chunks are considered as files for ceph which are renamed to sha(256) values.In proposed method each osd maintains a hashtable which is shown in following figure. SHA(256) bits are converted into 33bit decimal value for indexing on osd. Nodes in hashtable maintain chunks of file with sha(256) value,number of users sharing block and pointer to next node. Redundant block is detected by comparing sha(256) value.If two blocks are having same sha(256), reference to block is increased.Designed top layer wont allow redundant blocks to go ceph osds.

B. Object metadata:

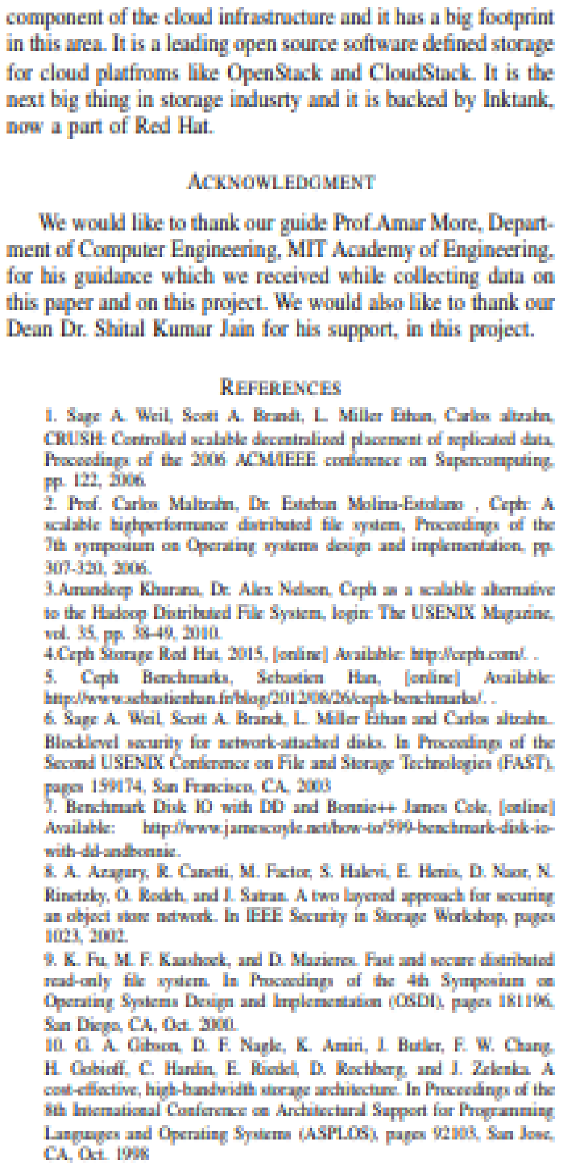
A external in memory data structure need to be maintained for compatibility between existing metadata and new metadata.Object file is stored at ceph-monitor containing orignal file name along with its chunk's sha(256).Object file plays an important role while file retrival.

C. Collision management:

Redundant data blocks are detected by comparing sha(256) values but while indexing it is reduced to 33bit which makes hashtable prone to collisions. 256 bits cannot be used for indexing because of lack of memory space on ceph-monitor.Problem of collision is solved by chaining on condition of sha(256) comparision, so whenever collision is detected sha(256) values of blocks are checked if both are same then redundant blocks is detected and reference is incremented else chaining is done.

VI. Conclusion

Ceph is more complete and economic as compared to other storage system. It has its entire new way of storing data. For all the storage needs it provides a enterprise grade open source technology for the storage needs and gives the power to break the knots of expensive vendor lock-in solutions. Ceph has gained maturity over the period of a decade which makes it stand out in the crowd. It is the technology of today and future, for all the requirements for data storage unified Ceph storage system has a solution for it. Ceph is the most essential



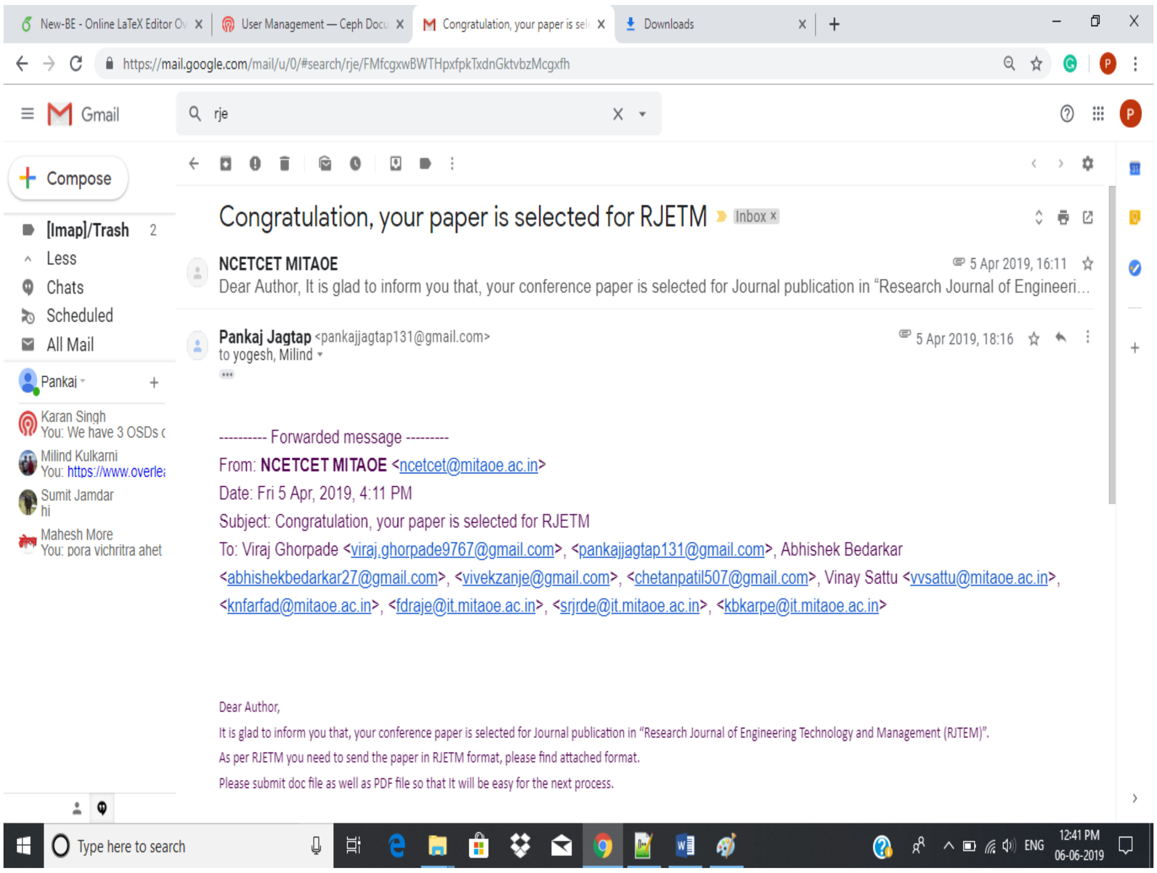
component of the cloud infrastructure and it has a big footprint in this area. It is a leading open source software defined storage for cloud platfroms like OpenStack and CloudStack. It is the next big thing in storage indusrty and it is backed by Inktank, now a part of Red Hat.

Acknowledgment

We would like to thank our guide Prof.Amar More, Department of Computer Engineering, MIT Academy of Engineering, for his guidance which we received while collecting data on this paper and on this project. We would also like to thank our Dean Dr. Shital Kumar Jain for his support, in this project.

References

1. Sage A. Weil, Scott A. Brandt, L. Miller Ethan, Carlos altzahn, CRUSH: Controlled scalable decentralized placement of replicated data, Proceedings of the 2006 ACM/IEEE conference on Supercomputing, pp. 122, 2006.
2. Prof. Carlos Maltzahn, Dr. Esteban Molina-Estolano , Ceph: A scalable highperformance distributed file system, Proceedings of the 7th symposium on Operating systems design and implementation, pp. 307-320, 2006.
3.Amandeep Khurana, Dr. Alex Nelson, Ceph as a scalable alternative to the Hadoop Distributed File System, login: The USENIX Magazine, vol. 35, pp. 38-49, 2010.
4.Ceph Storage Red Hat, 2015, [online] Available: http://ceph.com/. .
5. Ceph Benchmarks, Sebastien Han, [online] Available: http://www.sebastienhan.fr/blog/2012/08/26/ceph-benchmarks/. .
6. Sage A. Weil, Scott A. Brandt, L. Miller Ethan and Carlos altzahn.. Blocklevel security for network-attached disks. In Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST), pages 159174, San Francisco, CA, 2003
7. Benchmark Disk IO with DD and Bonnie++ James Cole, [online] Available: http://www.jamescoyle.net/how-to/599-benchmark-disk-io-with-dd-andbonnie.
8. A. Azagury, R. Canetti, M. Factor, S. Halevi, E. Henis, D. Naor, N. Rinetzky, O. Rodeh, and J. Satran. A two layered approach for securing an object store network. In IEEE Security in Storage Workshop, pages 1023, 2002.
9. K. Fu, M. F. Kaashoek, and D. Mazieres. Fast and secure distributed read-only file system. In Proceedings of the 4th Symposium on Operating Systems Design and Implementation (OSDI), pages 181196, San Diego, CA, Oct. 2000.
10. G. A. Gibson, D. F. Nagle, K. Amiri, J. Butler, F. W. Chang, H. Gobioff, C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka. A cost-effective, high-bandwidth storage architecture. In Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 92103, San Jose, CA, Oct. 1998


Congratulation, your paper is selected for RJETM

NCETCET MITAOE 5 Apr 2019, 16:11
Dear Author, It is glad to inform you that, your conference paper is selected for Journal publication in "Research Journal of Engineeri...

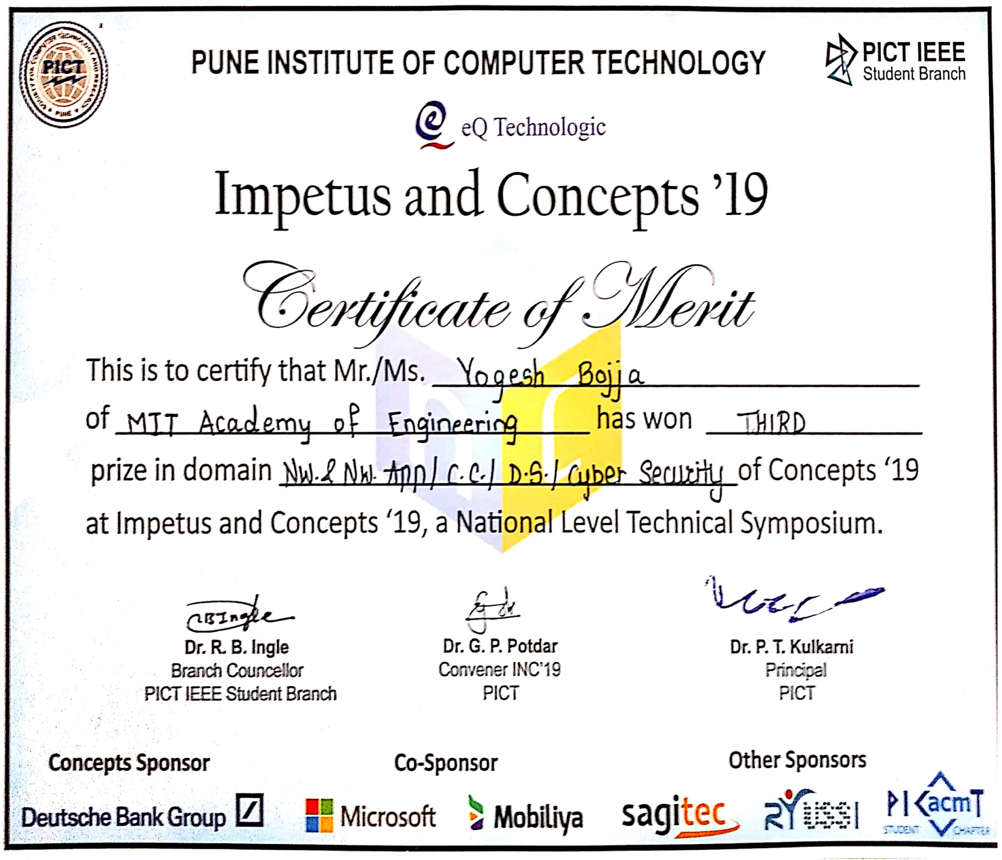
Pankaj Jagtap <pankajjagtap131@gmail.com> 5 Apr 2019, 18:16
to yogesh, Milind

---------- Forwarded message ---------
From: NCETCET MITAOE <ncetcet@mitaoe.ac.in>
Date: Fri 5 Apr, 2019, 4:11 PM
Subject: Congratulation, your paper is selected for RJETM
To: Viraj Ghorpade <viraj.ghorpade9767@gmail.com>, <pankajjagtap131@gmail.com>, Abhishek Bedarkar <abhishekbedarkar27@gmail.com>, <vivekzanje@gmail.com>, <chetanpatil507@gmail.com>, Vinay Sattu <vvsattu@mitaoe.ac.in>, <knfarfad@mitaoe.ac.in>, <fdraje@it.mitaoe.ac.in>, <srjrde@it.mitaoe.ac.in>, <kbkarpe@it.mitaoe.ac.in>

Dear Author,
It is glad to inform you that, your conference paper is selected for Journal publication in "Research Journal of Engineering Technology and Management (RJTEM)".
As per RJETM you need to send the paper in RJETM format, please find attached format.
Please submit doc file as well as PDF file so that It will be easy for the next process.

New-BE - Online LaTeX Editor
User Management — Ceph Docu
Congratulation, your paper is sel
Downloads
https://mail.google.com/mail/u/0/#search/rje/FMfcgxwBWTHpxfpkTxdnGktvbzMcgxfh
Gmail
rje
Compose
[Imap]/Trash 2
Less
Chats
Scheduled
All Mail
Pankaj
Inbox
Type here to search
ENG
12:41 PM
06-06-2019

Figure I.1: Publication Details


PUNE INSTITUTE OF COMPUTER TECHNOLOGY

PICT IEEE Student Branch

eQ Technologic

Impetus and Concepts '19

Certificate of Merit

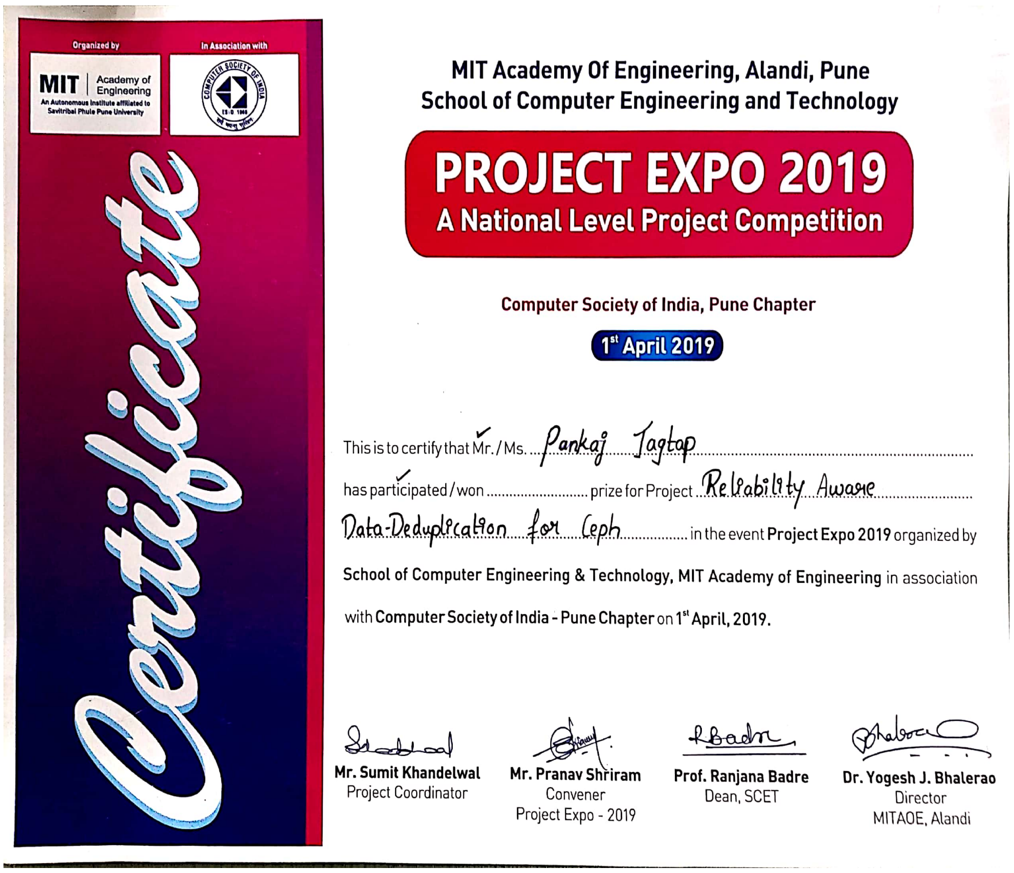
This is to certify that Mr./Ms. Yogesh Bojja of MIT Academy of Engineering has won THIRD prize in domain NW & NW Appl / C.C / D.S / Cyber Security of Concepts '19 at Impetus and Concepts '19, a National Level Technical Symposium.

Dr. R. B. Ingle	Dr. G. P. Potdar	Dr. P. T. Kulkarni
Branch Councellor	Convener INC'19	Principal
PICT IEEE Student Branch	PICT	PICT

Concepts Sponsor: Deutsche Bank Group

Co-Sponsor: Microsoft, Mobiliya

Other Sponsors: sagitec



Organized by

MIT | Academy of Engineering
An Autonomous Institute affiliated to Savitribai Phule Pune University

In Association with

Certificate

MIT Academy Of Engineering, Alandi, Pune
School of Computer Engineering and Technology

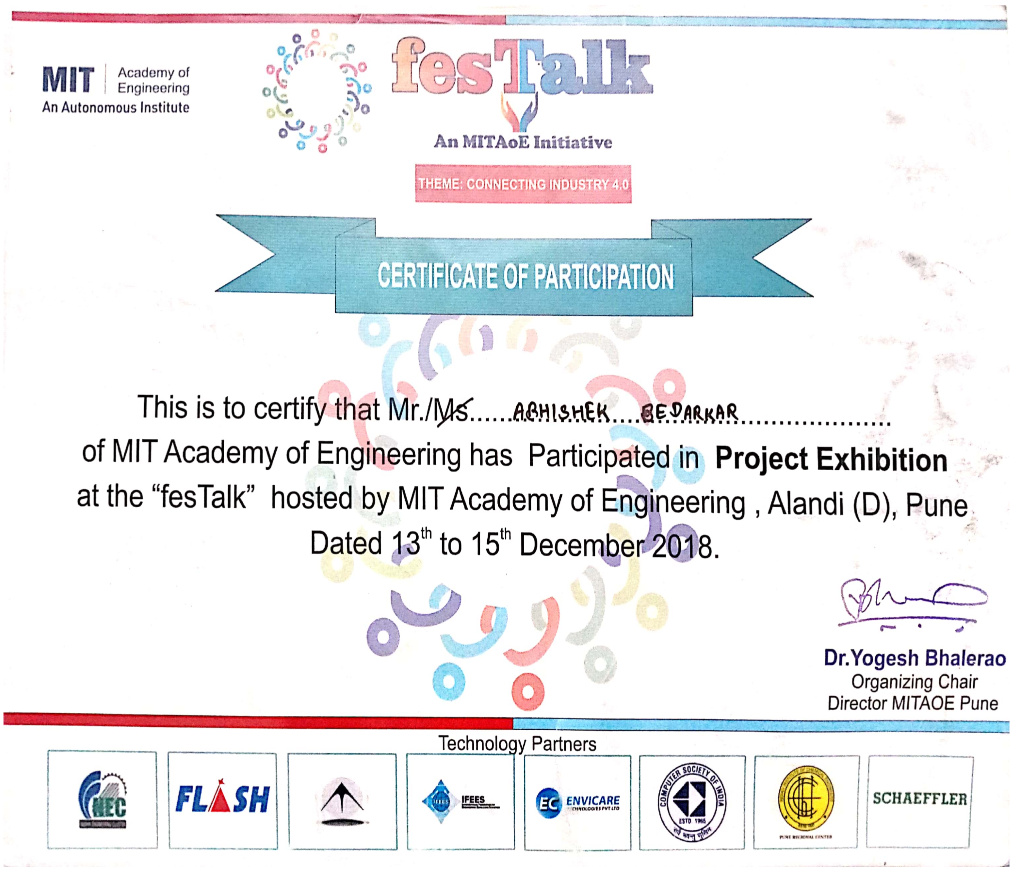
PROJECT EXPO 2019
A National Level Project Competition

Computer Society of India, Pune Chapter

1st April 2019

This is to certify that Mr. / Ms. Pankaj Jagtap has participated / won ..... prize for Project Reliability Aware Data-Deduplication for Ceph in the event **Project Expo 2019** organized by **School of Computer Engineering & Technology, MIT Academy of Engineering** in association with **Computer Society of India - Pune Chapter** on 1st April, 2019.

[illegible]	[illegible]	[illegible]	[illegible]
Mr. Sumit Khandelwal	**Mr. Pranav Shriram**	**Prof. Ranjana Badre**	**Dr. Yogesh J. Bhalerao**
Project Coordinator	Convener	Dean, SCET	Director
	Project Expo - 2019		MITAOE, Alandi



MIT | Academy of Engineering
An Autonomous Institute

fesTalk
An MITAoE Initiative

THEME: CONNECTING INDUSTRY 4.0

CERTIFICATE OF PARTICIPATION

This is to certify that Mr./~~Ms~~......ABHISHEK....BEDARKAR.................... of MIT Academy of Engineering has Participated in **Project Exhibition** at the "fesTalk" hosted by MIT Academy of Engineering , Alandi (D), Pune Dated 13th to 15th December 2018.

[illegible]

Dr.Yogesh Bhalerao
Organizing Chair
Director MITAOE Pune

Technology Partners

NEC FLASH IFEES ENVICARE SCHAEFFLER