A BETTER APPROACH TO INCREASE SECURITY AND PERFORMANCE IN MULTI-CLOUD ENVIRONMENT.

A Dissertation Submitted

By

Amandeep Tohan

To

Department of Computer Science

In partial fulfillment of Requirement for the

Award of the degree of

Master of Technology in Computer science

Under the guidance of

Ms. Monika Sehgal (17674)

Assistant Professor

(May 2015)

INDIA'S LARGEST UNIVERSITY

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science & Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Amandeep Tohan

Registration No: 11310091

Batch: 2013-14

Roll No. K2305B55

Session: 2014

Parent Section: K2305

Details of Supervisor:

Designation: AP

Name Monika Sehgal

Qualification: MTech CSE

U.ID 17674

Research Experience: 1 year

SPECIALIZATION AREA: Cloud Computing (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

✓ Security Implementation in Cloud Computing By Intrusion detection & filtration technique

2. Quality of service in MoBile cloud computing

3. Big Data Analysis and Hadoop.

Monika Sehgal

Signature of Supervisor

PAC Remarks:

first topic is approved, publication expected

30/9/14

APPROVAL OF PAC CHAIRPERSON: Signature: Date:

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

ABSTRACT

Cloud computing has a potential to provide economic and scalable computing facilities. Any IT related need can be provided online as a service. Security and effective performance are the two basic factors, but are the most important and key aspects for a robust and reliable cloud. Cloud facing many security issues like network security, application security, data security, confidentiality(C) of data, integrity(I) of data, authentication(A) of users, data access controls, secure communication etc. CIA components hold a major part of security issues and Multi-cloud system providing solution to these issues. Customers are increasing their interest in multi-cloud due to high service availability and lesser fear that malicious insiders to harm their data. More the number of clouds participating in a multi-cloud system better will be the security level. But this raises another issue of increase in storing and retrieval time of data from these multiple clouds which are logically associated to each other. Therefore, this leads to high computation input need but lower performance output. Now in this study, we are focusing on better data distribution techniques which reduce the transaction time and maintain the high security.

CERTIFICATE

This is to certify that **Amandeep Tohan** has completed M.Tech dissertation titled **A Better Approach to Increase Security and Performance in Multi-cloud Environment** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma.

The dissertation proposal is fit for the submission and the partial fulfillment of the conditions for the award of M.Tech Computer Science & Engg.

Date: ____________

Signature of Adviser

Name:

UID:

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Ms. Monika Sehgal (Dissertation Mentor) for her valuable guidance that I could take up initiative of such a good topic of thesis. I am also very thankful to Lovely Professional University for giving me opportunity to propose the dissertation.

I would like to convey thanks to all my friends who gave their full support and encourage me for this thesis work.

Amandeep Tohan

DECLARATION

I hereby declare that the dissertation entitled, **A Better Performance to Increase Security and Performance in Multi-cloud Environment** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: _________________

Investigator

Regn. No.__

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Theoretical Foundation

Cloud computing is collectively a collaboration of computing services. These services are maintained by number of servers. Customer need not to have expensive software install at its end or neither need to buy costlier and high performance hardware at their end. Cloud provides resources on your machine, customer just need an internet connection to avail these services through web-based tool at reasonable prizes. Users access their desktops or personal computing devices and utilize efficient resources on-demand.

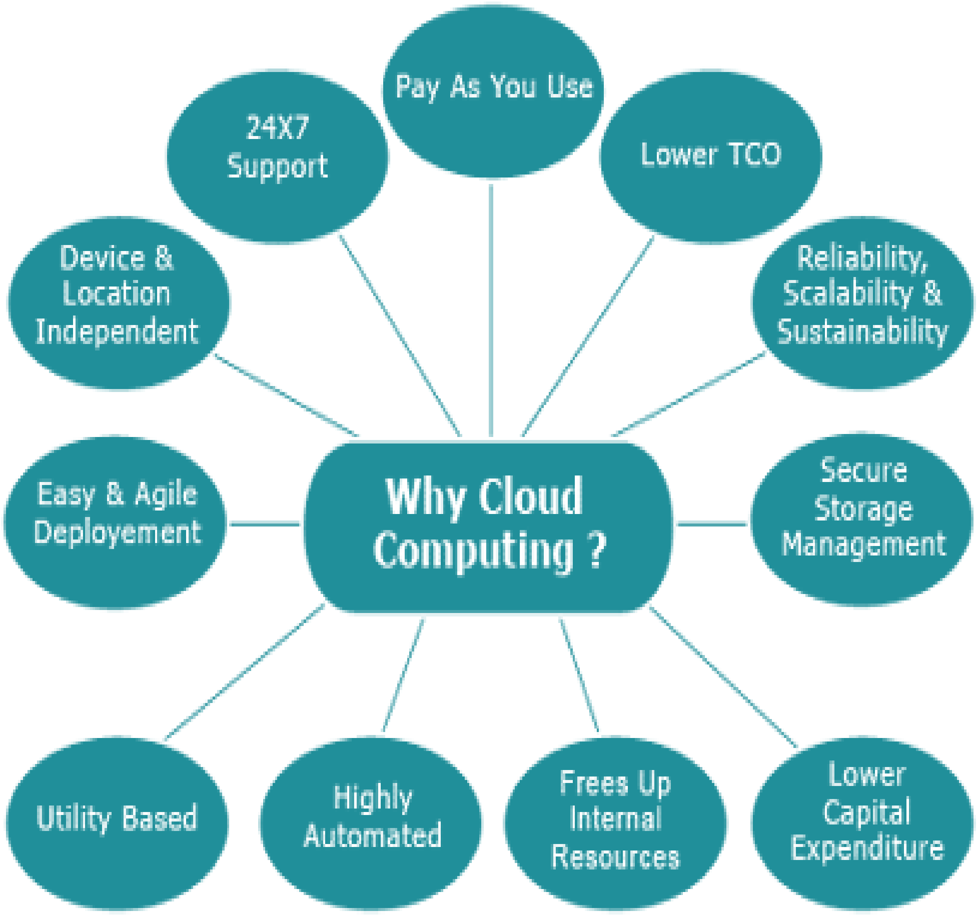

Pay As You Use
24X7 Support
Lower TCO
Device & Location Independent
Reliability, Scalability & Sustainability
Easy & Agile Deployement
Why Cloud Computing ?
Secure Storage Management
Utility Based
Highly Automated
Frees Up Internal Resources
Lower Capital Expenditure

Figure 1.1. Needs to have cloud computing environment

Cloud computing is an on demand service over network which includes applications or software or OSs, networks management, storage provision etc. and clients will have to pay only as per the use of these services to CSP. These services are flexible to use and can be released at any point of time with a little interaction with service provider.

The word cloud storage can be described as storing the data in online cloud datacenters environment. Cloud storage provides access of data to relevant users like customers, administrators, trusted software in local machines of cloud. Clients are increasing their interest in cloud storage due to facilities like greater flexibility of resources, easy accessibility, more reliability and safe backups being maintained at CSP end. Therefore, now companies or individual clients need not to worry about cost factor of buying high quality hardware at their local end and maintain them for use. However, despite of provisions of these facilities cloud computing does suffering from security issues. A cloud environment is actually a virtual environment, a high performance processor may be divided, similarly large hard disk space can be divided virtually and distributed among customers and it seems from customer end that customer is having a complete set of computing machine provided to him.

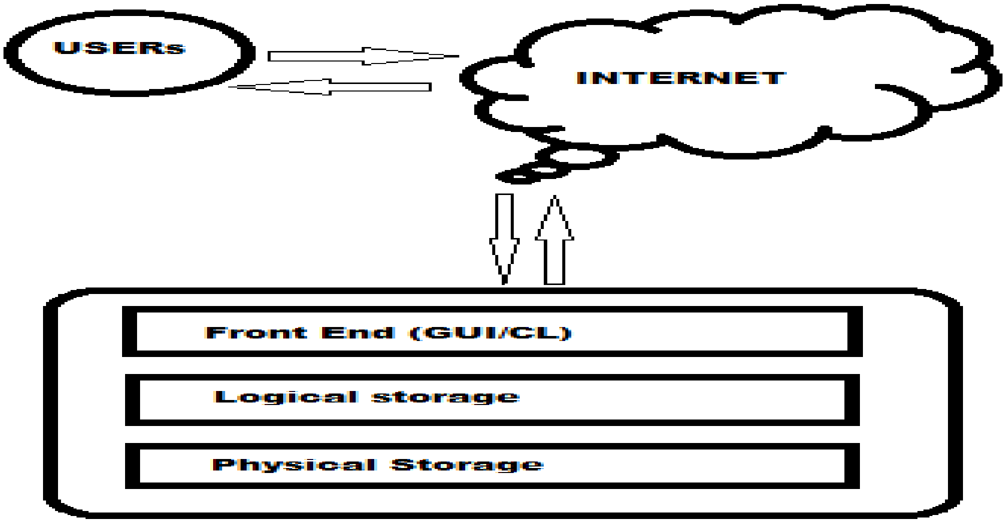


Figure 1.2: Cloud storage architecture

The above storage system is manageable with minimal resources. Some access methods are used to access the online storage. Performance of the system is measured in the form of bandwidth and latency. Large storage can be logically divided so that different users can access their allotted portions. The storage can be logically increase or decrease, known as scalability of the system. Various protocols like FTP, NFS, CIFS are used to access the files located in storage. Users access data via an interface account provided by the cloud storage provider. Data of a single account could be replicated to other datacenters for backup purposes or for faster access of the data at geographically far locations. Different storage providers may have different policies regarding storage of the type of data.

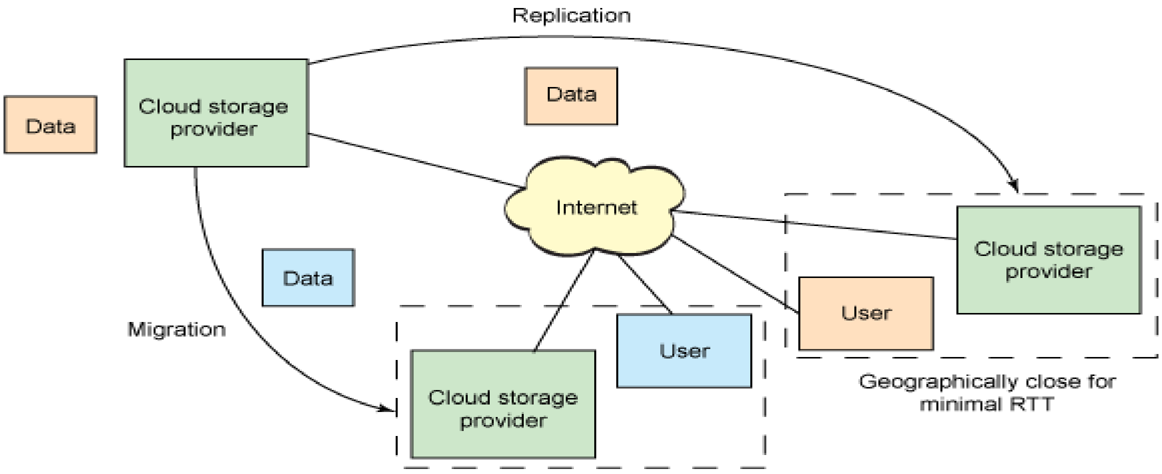


Figure 1.3: Data accessibility from cloud storage provider

Now, **Data security** is a big concern in cloud storage. Organizations, either small or large scale are using cloud services for hosting their sensitive data that is related to customers, employees, deals or something that is confidential. Availability of this data and security from intrusion is very important. There are many threats to cloud security, some of them are coated below:

- **The threat of malicious insider**

  There is no public example to explain at this time. In this case the service provider do not tells the customer that what permissions are granted to the employees inside the cloud to access customer's data. This can lead to harvest confidential data with little or no risk of detection.

  The remedy is to apply security breach notification mechanism. Need more transparency.

- **Shared resources**

  In the case, Infrastructure As A Service the resources are virtually divided and allotted to different customers. For example, in a single server CPU cores, Hard drive, RAM etc are logically divided among customers. This does not give a strong isolation from each other.

  Monitoring and strong policies should be adopted to be ensure that no user is interrupting or harming or users on the same physical machine. Strong authentication and access control is required. Periodic vulnerability and configuration audit should be conducted.

- **Data Loss**

  Data could be lost in many ways. Backup is always required. Data deletion and alteration are big issues. Data should not be access by unauthorized parties. Loss of encryption key, account passwords other sensitive details may lead to complete destruction of the sensitive data. Due architectural design and operational characteristics the cloud environment is more risky for such losses.

  Need to apply strong access control methods. Backups, strong key generation methods, encrypt and protection of data.

- **Account Hijacking**

  This is not new. Attacks like phishing, session hijacking, MITM attack etc. are used for exploitation and stealing of passwords and other credentials. In cloud hacking of a cloud can lead to data manipulation, transactions, redirection to malicious sites, eavesdropping etc.

  Account credentials should not be shared among each other. We should use two factor authentications wherever possible.

1.1.2 Characteristics of cloud

To more describe cloud computing let us discuss its characteristics.

- **First**, on demand self -service, which means a user can demand any computing resource service from service provider at any time 24*7 and resource will be available to him or her as soon as possible.
- **Second**, Broad network access, means client can access the resources anywhere, anytime and at any mobile device or computer machine.
- **Third**, Resource pooling, means resources for all individual clients are logically separated from resources of other clients in the same cloud.
- **Fourth**, Rapid elasticity, means users can have provision and can leave the facilities and resources by CSP at any time he/she wants.
- **Fifth**, Measured service, means usage of all the resources requested are recorded transparently.

1.1.3 Cloud models

Single cloud system contains two models, deployment model and service model.

Deployment model is selected on the bases of structure of the organization, it is further categorized into 3 types.

- First, **Public cloud**, it purely deals on internet. It is open for general public. It is basically a pay-per use model. However, public cloud may not be beneficial for those organizations having sensitive data because of its security reasons. Some known public clouds are Google AppEngine, Windows Azure, Amazon Elastic Compute Cloud (EC2), sun cloud and IBM's blue cloud.
- Second, **Private clouds** are those which have infrastructure and services owned by a single organization or company. No other unauthorized person can have access to these services, that means who does not belongs to this particular organization until and unless the access is given by the company itself.
- Third, **Hybrid clouds**, it uses features of both private and public cloud. A company may deploy private cloud for all its sensitive tasks and can avail the services of public

cloud where there the operations performed are not so sensitive. Hybrid cloud can be implemented in further 3 ways. First, company can contact two different service providers for both private and public cloud services. Second, a single service provider can give a complete package of both. Third, organization can manage their own private cloud and public cloud services can also be integrated into it.

- Others, **Community cloud**, several inter related communities or organizations share a single infrastructure with a common concern like security, information etc. **Distributed cloud**, includes number of machines geographically at different places but providing cloud services all together. **Intercloud,** it is a global cloud of clouds. More than one cloud can inter connect, if one cloud is not having those resources that other one can provide. **Multicloud**, multiple cloud services are combined to reduce the reliance on a single vendor.

1.1.4. Service model categorization: also known as SPI model is broadly classified as follows:

- **SAAS: Software as a service**

  It holds the biggest market in cloud. Generally, this service is directly provided through web browser with a small client side interface or in some cases it need some plugins or small client side application to avail it. This service replaces the old install and use of software on individual machines. Some examples are Google Gmail, Microsoft 365, Salesforce, Citrix GoToMeeting, Cisco WebEx

- **PAAS: Platform as a service**

  This service is use for operating systems, servers, virtualization, storage, networking or application deployment. Customer (developer) gets a framework platform to design or create new application or modify the applications. PAAS provide a fast, easy and cost-effective way to create, test and deploy applications. For example, Appendra is a PAAS provider for .NET and JAVA.

- **IAAS: Infrastructure as a service**

  Instead of buying costly hardware and later upgrade and maintain it, it is better to use someone else's infrastructure and pay for what you use. This model includes the use of bare hardware computer machines, virtualized hardware infrastructure, networking services like firewall and storage etc. Infrastructure as a service providers are Amazon Web Service (AWS), Microsoft Azure, Google Compute Engine (GCE).

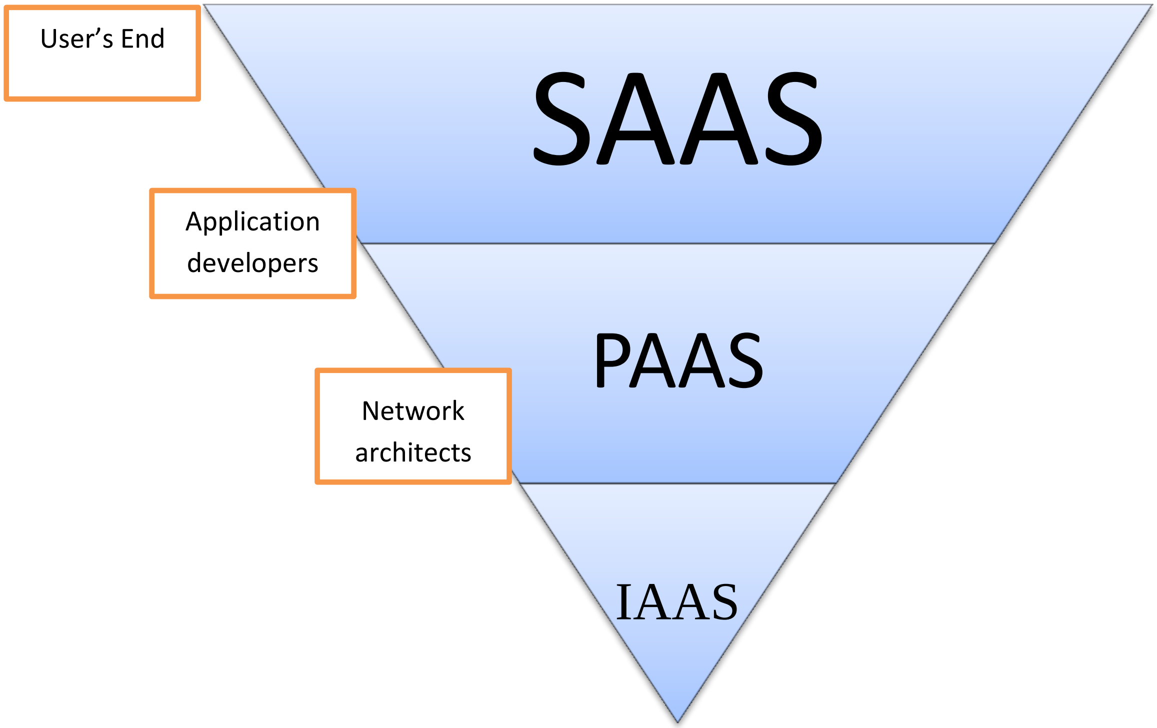


Figure 1.4: Service model (SPI)

1.1.5. Benefits of cloud computing:

- **Reduce capital cost**, no need to invest in buying hardware or software etc.
- **Improves accessibility**, anyone can access their belongings in cloud anytime anywhere.
- **Monitoring** your projects more efficiently and with latest ideas.
- **Less personal training** is needed to maintain the infrastructure or software.

- **Improve flexibility** in infrastructure scaling or moving to other service provider for better options.
- **Dedicated security team**
- **Low-Cost Disaster Recovery** and Data Storage Solutions.
- **On-Demand Security** Controls. [1]

1.1.6. Challenges in Cloud computing:

Despite of having so many benefits in cloud computing, still the list of challenges is very big. There are so many areas which needs improvements. Some of these areas are as follows:

- **Security and privacy**

  The most important concern is security and privacy, as the valuable and confidential data of organizations stay here. In shared environment of cloud hacking of even a single account may lead to a serious security attack. Various techniques like encryption, security applications, data loss software, backup mechanisms, secure hardware are being implemented to reduce this factor.

- **Reliability and Availability**

  Cloud environment should be reliable in providing the services, a 24 hour monitoring or services is requires by administrators with the use of specialized software which alerts you about the activities happening in cloud services. If the service is not available when needed, it leads to frustration.

- **Performance and Bandwidth Cost**

  We are getting better infrastructure at reasonable prizes but it need high network bandwidth to enjoy the performance of these quality services.

- **Other**, Service Delivery and Billing, Interoperability and Portability, QoS, Encryption needs for cloud computing, Data retention issues, network attacks etc.

1.2. Evolution of subject and its Dimensions

The cloud storage environment we are talking yet is in a **Single cloud system,** which belongs to one CSP. For example: Amazon, Google, IBM, Microsoft etc. Let us discuss some of the main security issues in this single cloud system. First (integrity), Malicious insiders are the users of the data that work inside the CS and they have access permissions by which they can harm the valuable data of user. Second (availability), what if, an accident (hardware damage, hard disk failure) causes a long term down time of cloud system. Third (confidentiality), what if, the data stored in plain text format is being read by administrators or other users inside the cloud environment (malicious insiders). To avoid such issues, a concept of Multi-cloud system is a center of attraction in today's time. **Multi cloud system** consists of more than one single cloud systems jointly working to give away a better service to client. Different clouds may use for different services like one for security another for storage etc. In our case, multiple number of CSPs providing storage services to a single vendor, to store their data. Multi cloud system with some special and effective storage techniques and security measures, can resolve the above described security issues in single cloud system. Hence, minimize the risk of data loss and downtime due to localized components.

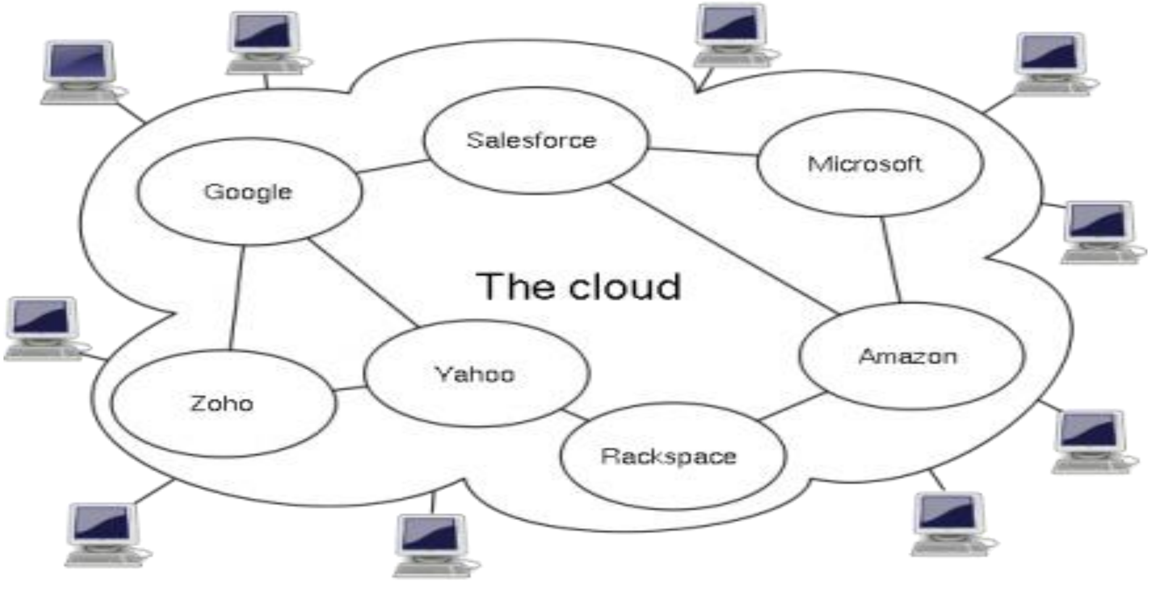


Figure 1.5. Multi-cloud environment

Now a days, while accessing a website, each one of us want a fast loading of home page or any page of that website. Because the fast is the loading the better interest would be shown by the visitors. Therefore, on one hand if we are using distributed environment of multi clouds, on other side we have to keep the concern of performance of data retrieval and storage time. Better performance need better storage techniques which should be secure and swift.

Intrusion can be explained as accessing the customer's personal data or tempering it. Customer should be aware about when is his data is being access, who is accessing it and if there is any harm to his/her data by the accessing party. So that it can be easier to take decision by the customer to continue the business in future.

1.2.1. Benefits of multi cloud:

- **Autonomy**: reduction in dependency on a single cloud vendor. Easy switching may offer a better facility package at lower cost.
- **Extended capability**: if a service is not available at one cloud one can have it on other cloud
- **Hybridity**: different applications running on different clouds, so as to provide better performance on the basis of geographical performance of customer and cloud.
- **Better service availability than single cloud**
- **Better security than single cloud**

1.2.2. Challenges in multi cloud:

- **Complexity**: It is a biggest challenge because two different clouds can have different policies, different technologies, different infrastructure, different terminology, also the difference can be there in services.
- **Management overhead**: multi cloud system increases management overhead for expertise.
- **Low performance**.

CHAPTER 2

REVIEW OF LITERATURE

Mohammed A. AlZain, Ben Soh, Eric (2012) “A New Model to Ensure Security in Cloud Computing Services”, PhD Candidate , Department of Computer Science and Computer Engineering, La Trobe University, Bundoora 3086, Australia, Associate Professor, Department of Computer Science and Computer Engineering, La Trobe University, Bundoora 3086, Australia. Ph.D, Department of Computer Science and Computer Engineering,La Trobe University, Bundoora 3086, Australia., respectively, SPRINGER, 2012. In this paper the purpose is to reduce the security and privacy risk being faced in single cloud system. Proposed model uses Shamir’s secret sharing algorithm to ensure security and privacy. Data should be divided such that out of n parts minimum k parts plus a secret key is required to retrieve the original secret value of data. In this technique whole data is encrypted and complete encoded sets of data is stored at different clouds in multi-cloud system, means all clouds need equal space to store this data. Author explains benefits of why prefer multi-cloud over single cloud system. The proposed model consists of five components namely end user, http server, servlet engine, data source and data storage. The model tends to resolve all the three CIA components of security with this approach. But facing a problem that the storing and retrieval time increases with increase in number of shares and security decreases with decrease in numbers of shares. They plan to compare more multi cloud models to be compare as future work.[2]

Pradnyesh Bhisikar and Prof. Amit Sahu (2013) “Security in Data Storage and Transmission in Cloud Computing”, G.H.Raisoni College of Engineering and Management, Amravati , IJARCSSE, 2013. In this paper the author proposed a method to secure data from unauthorized access and the level of security will be decided by the customer itself so that the data security is at high priority selected by the customer. Three network parties are explained User, CSP and Third party auditor (TPA). Author explained Week adversary as those that get the access to data of a single client on the shared server and they simply modify the data or add their own additional data in between and Strong adversary is taken as a worst case which can leads to compromise of complete server. The model proposed include a user interface through which user tries to connect. A security server before data access holds level of securities through which user have to pass to access the data centers. All communication takes place in encrypted form. In future work they plan to investigate the problem of fine-grained data error localization.[3]

Prof.V.N.Dhawas, Pranali Juikar, Neha Patekar, Neha Lendghar, Sushant Vartak "A Secured Cost Effective Multi-Cloud Storage in Cloud Computing", Department Of Computer Engineering ,Sinhgad Institute of Technology,Lonavla, IJSER, 2013. IN THIS PAPER proposed a secure cost effective multi-cloud storage (SCMCS) model which economically distribute the data among the availed cloud providers to ensure the security and availability of data. Author explained that existing sole cryptographic techniques are insufficient to maintain privacy of customer's data. He proposed that data should be cut into pieces and no single piece at any of the cloud will be sufficient to retrieve the meaningful data. Threat model in paper explains two threats, first is of data availability if a server holding the data is crashed and second is what if hired service providers patch up and reconstruct the data. Author proposed a linear programming model, which helps in quality of service level, which further will help in protecting data misuse by the cloud provider.[4]

B.AmarNadh Reddy and P.Raja Sekhar Reddy, "Effective Data Distribution Techniques for Multi-Cloud Storage in Cloud Computing" CSE, Anurag Group of Institutions, Hyderabad, A.P, India , IJERA, 2012. In this paper the author is explaining about the concept of distributing the data among two different clouds, so that a single cloud can't achieve the data without having access to both the clouds. Author is telling about the encryption and fragmentation process for secure data storage. Different type of fragmentation techniques are explained like horizontal, vertical, mixed or derived. Author suggests not to store those data on cloud servers which is not use by local processes. Two models, system and distribution are proposed. In system model all the data is stored on clouds for same time period. Each service provider may provide different quality of service as requested by the customer according to the level of security customer wants for specific data. Distribution model offers data division in such a way that no single cloud can have meaningful information.[5]

Sultan Ullah and Zheng Xuefeng, “TCLOUD: A Trusted Storage Architecture for Cloud Computing”, School of Computer and Communication Engineering, University of Science and Technology, Beijing People Republic of China, IJAST, 2014. In this paper the author proposed model consists of two modules, client side and cloud side. Client module further include access module, which provide client side Authentication using two passwords, it also include split/merge module which will divide the data when forwarding and also merge it when retrieving by algorithm and then encrypted. Server side module also include authentication for data access. Private section holds private data of user like secret keys. Public section holds encrypted data formed by different operations and all related users.[6]

Kalyani.M.Borse, Ankita.G.Deshpande, Ashlesha.A.Deshpande, Ms.Juily.S.Hardas, “Security in multi-cloud data storage with SIC architecture”, Student, Computer Engineering Department, GES R.H.S.COE, Maharashtra, India , IJRET, 2014. In this paper the author proposed a Secure Inter-Cloud 3 tier architecture. It is proposed here to have a central server with very high security level and any data access should be done through this server only. Hence, this central server act as a gate-way to the storage. They also make use of data encryption techniques to store data at cloud storage. Disadvantages of single cloud like high cost, data loss etc. are discussed and objective of multi-cloud like high security in case of data integrity, data intrusion, service availability, high performance and cost reduction are discussed.[7]

Sandip Agarwala, Ramani Routray, “Cluster aware storage provision in a Data Center”, IBM corporation, US, IEEE, 2010. In this paper the author is telling Clustering is being used to achieve high performance and availability. Clustering include multiple servers for load balancing. The overall behavior of the cluster is dependent on the end-to-end configurations of multiple components of the network and cluster. There is no end-to-end configuration management tool, which can help in managing the load across various nodes in cluster. In a high availability cluster, if both active and standby nodes are being access through a single network path, then in case of failure of this network path both the nodes will be inaccessible, hence, unavailable. Also, if the same network path is being used by many cluster then this path may lead to the bottleneck for the requests and responses. CASPER is a tool that automates the process of planning, optimization and deployment of shared network storage for clusters. Casper distributes storage load across different storage nodes and across different network paths to ensure load balancing and prevent single point of failure. Casper allocates storage and automatically configure the network path to make SAN storage accessible to the cluster nodes in a way that balances the overall load and avoid any single point of failure.[8]

Mr. Ajey Singh and Dr. Maneesh Shrivastava Department of Information Technology, LNCT- Bhopal- India, “Overview of Security issues in Cloud Computing “,Department of Information Technology, LNCT- Bhopal- India, IJACR. In this paper, author discuss security issues in cloud computing and some security tips are provided to deal with those issues. Author discussed the Types of Clouds, Public cloud: User can access the services through web browser, Private cloud: For a single organization, Hybrid cloud: Mixture of both Public and Private cloud, Community cloud: Cloud environment being share by specified number of organization. Also the Structure of cloud computing: Infrastructure as a service, Platform as a service, Software as a service. Some attacks like DOS (Denial Of Service), Cloud malware Injection Attack, Man in the middle attack. Security Tips: Understand and find out the loose structured holes in the cloud from where data is being transfer in and out. Cloud provider should provide basic security architecture and should provide regular security audit from independent third party. Check the way of data is entering the cloud is legal or not. Pay attention to the emerging cloud technologies and test if that can help you in rising your cloud security.[9]

Arun Arunachalam, Shree harsha Chandrashekar, “Cloud Computing Featuring Ontology”, Computer Engineering Department, VJTI , Matunga, Mumbai, India, IOSRJEN, 2012. In this paper the author proposed an SMS gateway working at server end to reply for the queries of the customers at their feature phones. This include, the customer will sms to server requesting for the detail of any organization, application or process running in server will search for the details relayed to the query in its database and send back an sms with details and some related extra details. The concept is to provide immediate information to those people who are unable to afford high end smart phones to get information from internet.[10]

Randeep Kaur Chhabra, Prof. Ashok Verma MTECH (CSE), “Strong authentication system along with virtual private network: A secure cloud solution for cloud computing”, ,Head of Department of Computer Science & Engineering Gyan Ganga Institute of Technology & Sciences Jabalpur, Madhya Pradesh, India IJECSE. In this paper the author discussed that Cloud computing is facing big risks in the area of integrity, privacy, network security and also greatly in users authentication. due to the insecure limitation of static password, which can be easily determined, because humans have a limited scope to remember so many good passwords and change them after a suitable time period. So, to overcome this, a technique (OTP) one time password is being used and mobile phones are the mediums of strong authentication by two factor authentication. OTP is valid for 3 minutes only and will be of no use after this time period. If there is no mobile network to receive the OTP, then mobile phones should be synchronized with the servers via some applications and that application must provide the OTP, as it was going to be provided by server on network. All communication must be done on secure socket layer (SSL) with virtual private network (VPN), hence, all the above techniques can provide a good security solution.[11]

Shaik.Aafreen Naaz, Pothireddygari.Ramya, P.Vishunu Vardhan Reddy, “cloud computing: use of multicloud” Department of IT, G.Pullaiah College Of Engineering and Technology. In this paper the author introduce a Rain cloud model. Before that it explains the architecture of the cloud computing. Author explains the different ways of data distribution and fragmentation types. Four type of fragmentation is explained horizontal, vertical, mixed, derived and no fragmentation. Author tells the need of fragmentation, it works with views rather than entire relation. It provides better efficiency. Author explains a cost effective multi-cloud storage. The architecture of the model includes number of clouds connected together. The main point about the model is works in single or private organization only. It uses service request messages, Service response messages, Service level agreement. Author explain that let say there is a client4 that trying to connect cloud1 via SRM, but cloud1 is also serving to client1, client2 and client3. But client4 is repeatedly sending SRM to cloud1, so at last cloud1 will get congested and at this situation the message SLA to connect to public cloud for further services.[12]

Vartika, J.Nithin Chowdary, Dr.N.Srinivasu, “Data Storage Security in Multiple Unrelated Cloud Architecture” Department of Computer Science and Engineering, K.L.University, Vaddeswaram, India. International Journal of Computer Trends and Technology, April 2013. In this paper the author proposed a data distribution technique under An unrelated cloud model. The architecture includes multiple numbers of cloud providers. Each cloud provider have sub clouds for services, i.e. each cloud is a cloud of clouds. The data of the customer is first divides among cloud providers then each cloud distribute the data among their sub clouds. Data storage is an important aspect as the cloud providers are different from each other, they are not related to each other. So, the security should be of high level. According to author, several security issues are solved by this architecture. Author in this paper have also compare multicloud and single cloud. Author said in this way the data is protected from malicious insiders and is protected from hackers attacking the cloud system.[13]

B.AmarNadh Reddy,P.Raja Sekhar Reddy, “Effective Data Distribution Techniques for Multi-Cloud Storage in Cloud Computing “,CSE, Anurag Group of Institutions, Hyderabad, A.P, India. In this paper, author tells about architecture of cloud computing first and then explain about multicloud model. Author highlights that data stored on two different clouds may be collude by the two service providers and generate the valuable information from it, which could be confidential. Author provides a data distribution such that no service providers can collude with each other and be able to retrieve the value. Author explains various types of fragmentations, with the use of these fragmentation techniques, users can have better data storage security and high quality of service.[14]

CHAPTER 3

PRESENT WORK

3.1 Problem Formulation

In a very fast growing era of Information technology, each second data in the form of bits, bytes, MB, GB, TB is being created over internet. Organizations have many type of data to be store, data could be of sensitive information about various deals by organization, employee's personal data, website contents etc. Now this data could be in the form of tables, records, music files, videos files, documentaries etc. Each file could be of different importance for different companies.

Cloud computing technology provides various services to help organization to achieve better business heights. It provides various business solutions and support by simple use of internet services. One can pay as per the use of service.

But, cloud computing brings new security threats to the system. The attacker just attacks the access control systems or impersonates, i.e. the identity theft of existing user. Data security, privacy, resource availability and monitoring are some of the basic needs that customer need from cloud providers.

Despite of having so many security techniques to ensure security of data, the data is insecure and this is because of the malicious employees at client side, malicious employees at cloud side or the cloud service provider himself can be malicious to intrude the sensitive information store inside his cloud. Similarly, outside the cloud the network attackers or intruders attack the sensitive information.

Now, to get rid of all these issues in the cloud system, a new proposal of multi-cloud is introduce. Multi-cloud computing includes more than one single cloud service providers. Multi- cloud computing solves the problem of depending on single service provider for resources and services. Customer can simultaneously contract with other service provide for different services or for those resources which are not available and earlier or present service provider.

Or user can avail the second service provider's service for better quality of service for some of its data or special requirements.

In the base paper, the model called Multi-cloud Database model, the author introduce a technique which ensures the data integrity, availability and security from intrusion.

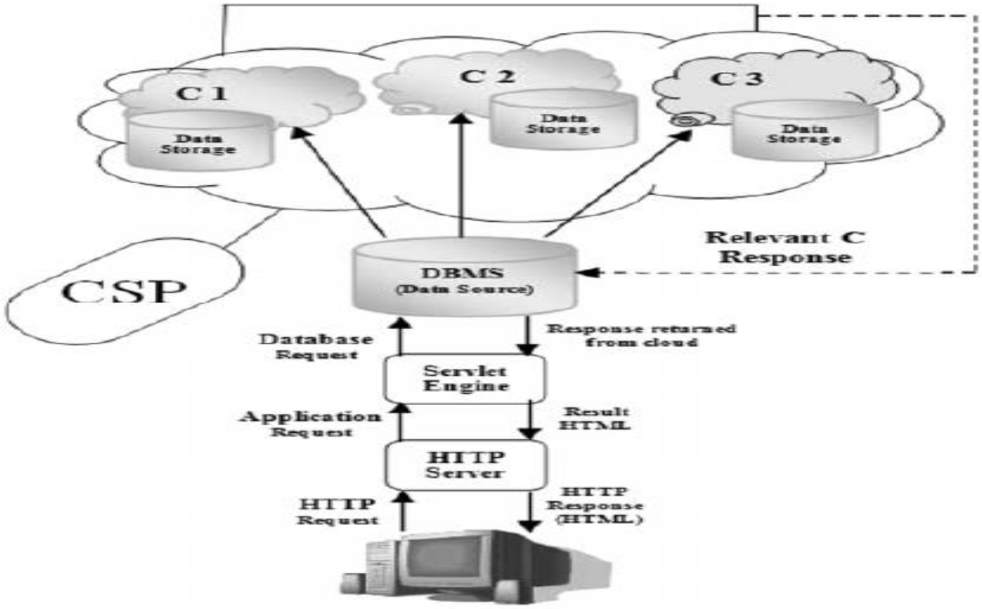
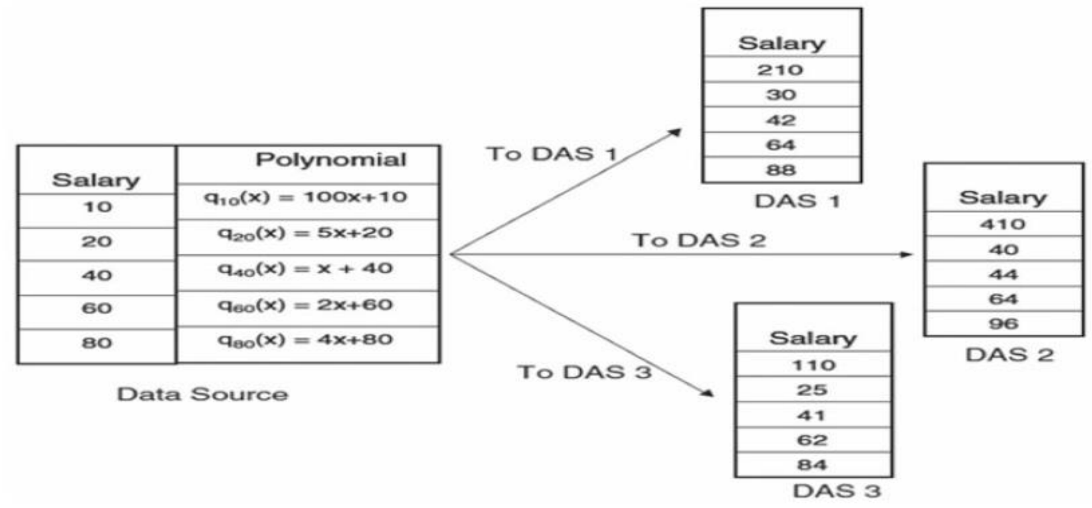


Figure: 3.1. General overview of multi-cloud

Given above is a multi-cloud system overview, which includes client system at the bottom for http queries. Then above is the http server and servlet engine to resolve the http queries. DBMS act as a data distributer and combiner at sending and receiving stage of the query, respectively.



Salary	Polynomial
10	$q_{10}(x) = 100x+10$
20	$q_{20}(x) = 5x+20$
40	$q_{40}(x) = x + 40$
60	$q_{60}(x) = 2x+60$
80	$q_{80}(x) = 4x+80$

Salary
210
30
42
64
88

Salary
410
40
44
64
96

Salary
110
25
41
62
84

Figure: 3.2 Data distibution in the clouds.

In this, the data is being store in the multiple number of clouds, in this case 3. As the above picture showing the data is being consider is a salary column of the employees to be store at different clouds(c1,c2 nd C3). A number of different polynomials with degree one are used to conver the actual data to a data to be store at clouds. A secret key value(x=2,4,1) is used for each cypher value to be store at respective cloud storage.

Let say there are n number of clouds, then atleast k number of cloud data is required to generate the actual value. Above technique provide a very good solution for security in case of availability, integrity and intrusion prevension. But the problem author mentioned in this technique is that the whole data is in need to be store at all the clouds. So, the storing and retrieval time is high. Also this time increases with increase of number of cloud availed. The graphical representation of the problem is shown below.

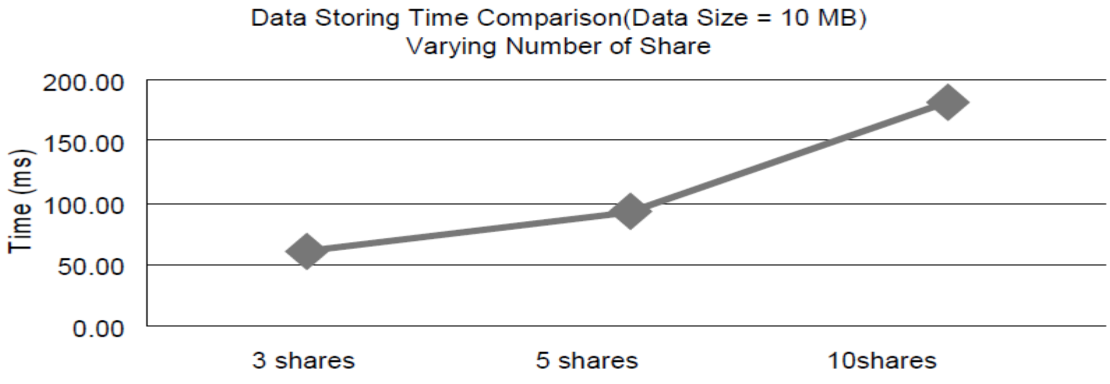


Figure: 3.3. Data storage time increases with increase of cloud shares

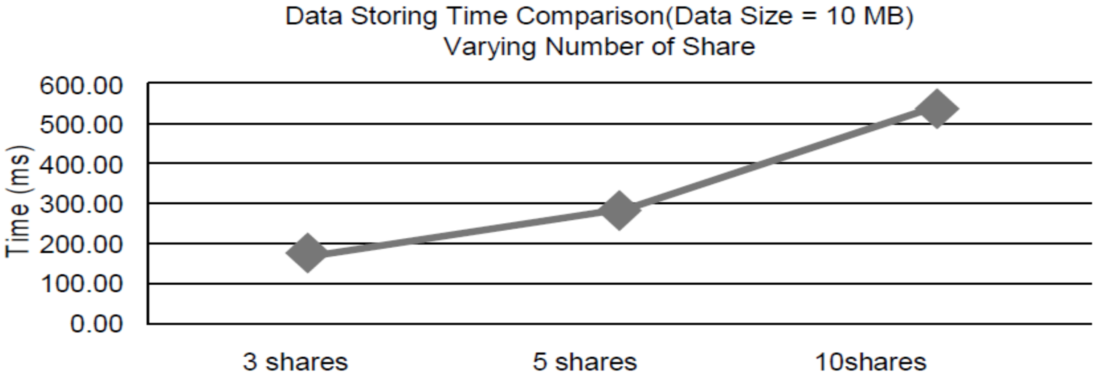


Figure: 3.4. Data Retrieval time increases with increase of cloud shares

Now, we know that speed is a very important factor when we are using resources onine. We can reduce this storage and retrieval time while maintaining the similar type of security level by propper distribution of the data into the cloud shares used.

3.2 Objectives

The main objective of the problem is to maintain the security and simultaneously reduce the fetching and storing time of data.

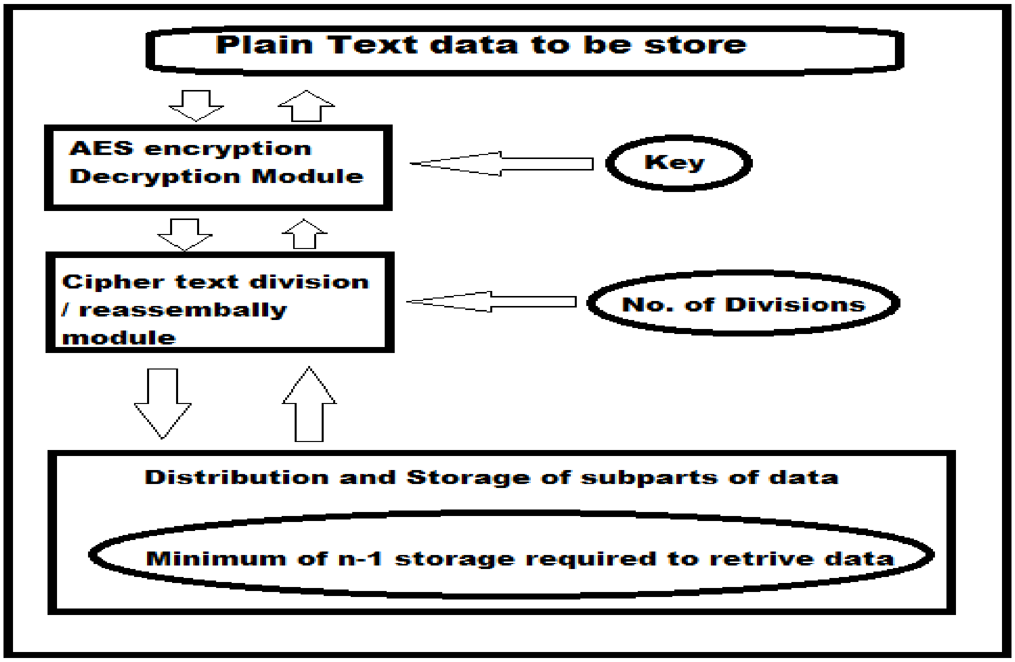
Websites access data from data centers, needs a faster response, multiple numbers of clouds used may reduce this response but a better data distribution can help in reducing the response time performance.

We are also trying to maintain the security provided by the model proposed in the base paper.

3.3 Methodology

Work is devided into following modules.

- Data encription/decription
- Data division/reassembaly
- Distrubution and storing of data



Plain Text data to be store
AES encryption Decryption Module
Key
Cipher text division / reassembally module
No. of Divisions
Distribution and Storage of subparts of data
Minimum of n-1 storage required to retrive data

Figure: 3.5 Reasearch Methodology Flow chart.

3.3.1 Data encription/decription

AES encryption technique is used to change the plain text contents of the file to a cypher text. Encrypted data is saved into a buffer and is later saved into the number of new files equal to the number of cloud shares.

Similarly, at the time of decryption these two file will be required to generate the original data. Otherwise,the actual data will not be possible toretrieve.

3.3.2 Data division/reassembaly

Encrypted data is equally divided and saved into new files. Reassembally of the data need these shares of the cypher text to retrieve the original text. Absence of any part will leed to errer in decryption phase.

3.3.3 Distribution and storing of data

Data disribution to the cloud is such that no single cloud contain the complete information. So that any intruder either malicious insider or hacker from outside can not harm or retreive the information by any hook or crook from single cloud.

Multi-cloud means more that one cloud, so in case there are two clouds C1 and C2. Let say the complete data is devided in two parts A and B and stored in C1 and C2 respectivly. But this scheme has less security, as data loss at any cloud will lead to loss of availabily of complete data. Data intrusion at any cloud is possible. Also integrity maintainance is very difficult.

In the case when there are three clouds C1, C2, C3 and the data should be divided into 3 parts (A,B,C).

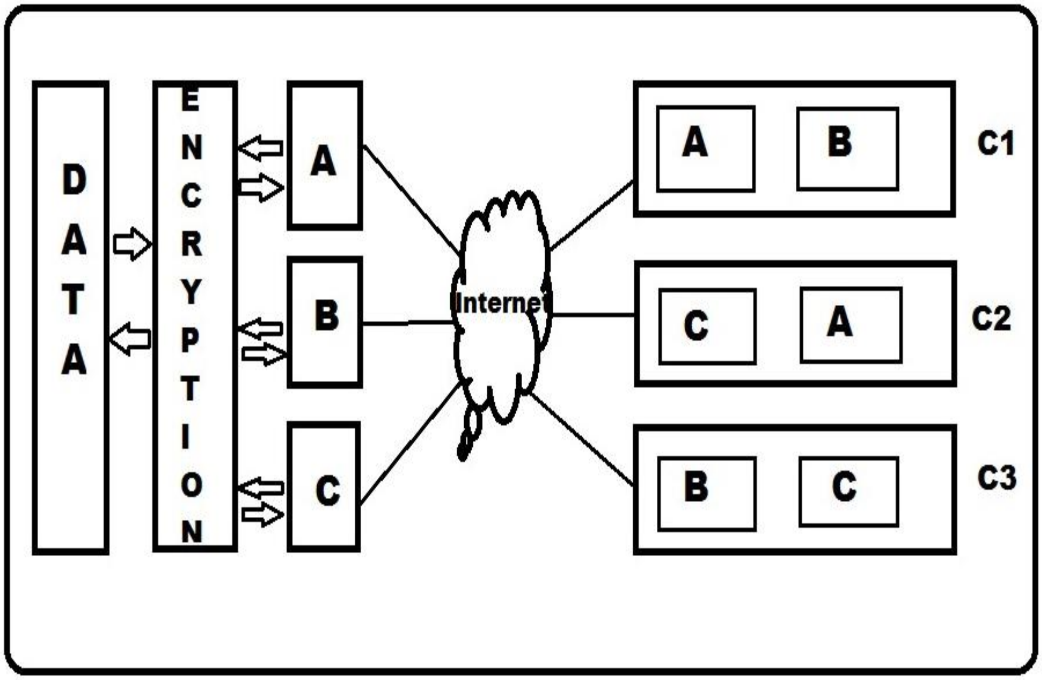


Figure. 3.6: Case - Three clouds.

Cloud C1 contain two parts A and B, Cloud C2 contains C and A, similarly in cloud C3 the data parts are B and C. The thing to be notice here is no single data contains the complete set of information. So any malicious user inside the cloud can never get the useful information from available parts of data. Authorized user can retrieve the data from any two clouds. This case maintains integrity, availability and intrusion prevention. If by any mode data is getting changed in cloud, user can get the same data he had stored from other clouds. At the time of down time at one cloud or DOS attack at any single cloud will not disturb the availability of data from other two clouds. Similarly, if any intruder is able to do intrusion at any single cloud it won’t cause any damage to the data at other two clouds.

The only lacking in this case is, if in order the malicious users of any two clouds try to combine the data. No doubt the data they will get will be in encrypted form but only an encrypted data is not completely safe.

Above lacking can be resolve in the case when there are 4 clouds, i.e. more the number of clouds more will be the security. Four clouds (C1,C2,C3,C4) contains data parts A,B,C,D in the form of distribution shown in following figure.

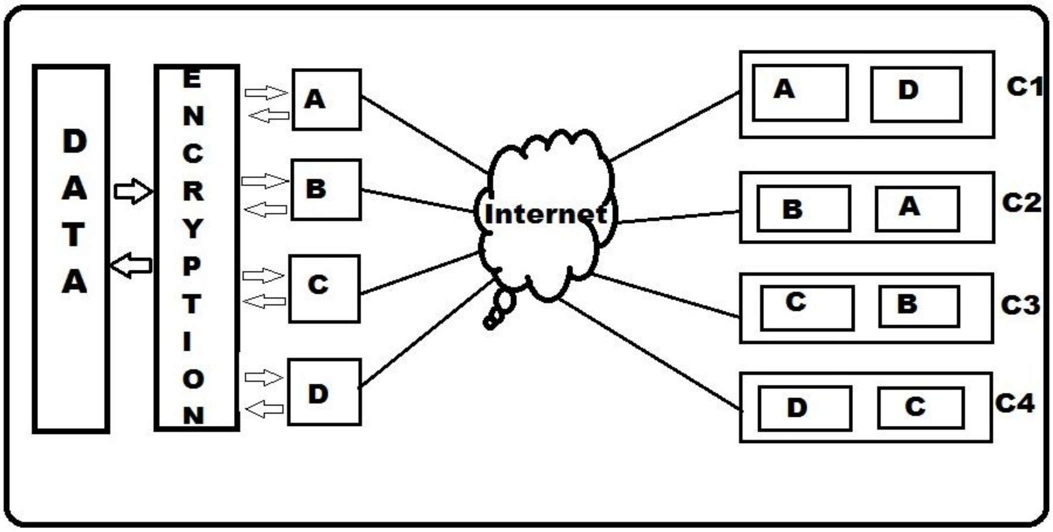


Figure: 3.7. Case – Four clouds.

This case fulfill all the cases explained in the previous case i.e. availability, integrity and intrusion prevention. Also it solves the problem that two clouds can combine and get encrypted data because to create complete encrypted data the malicious user need to get information from at least three clouds. There for we can increase the security by increasing the number of clouds.

CHAPTER 4

RESULT AND DISCUSSION

4.1. Introduction to the Simulator Used.

The proposed methodology is implemented and compared with the help of a simulator called cloud-sim. As utilization of real time cloud environment could be less friendly for experiments and result regenerations. One will have to use the environment according to the restrictions or availability of resources. Also the access to real cloud system will lead to exchange of cost in currency. So an alternative approach is used by researchers for emerging simulations, models and experiments related to clouds. Starters and developers in industries can design and mould the system that they want with the help of utilities provided in cloud-sim simulator. It supports modelling of large scale data-centres. One can design virtual machines and create policies of access control. Researchers can work upon network topology and energy-aware calculations. Cloud-sim with source code, examples, jars are available at https://github.com/Cloudslab/cloudsim/releases.

One can refer to tutorial of cloud-sim given at youtube for basic understanding about the environment. The latest version of cloud-sim is 3.0.3 and it contains the following directories and files.

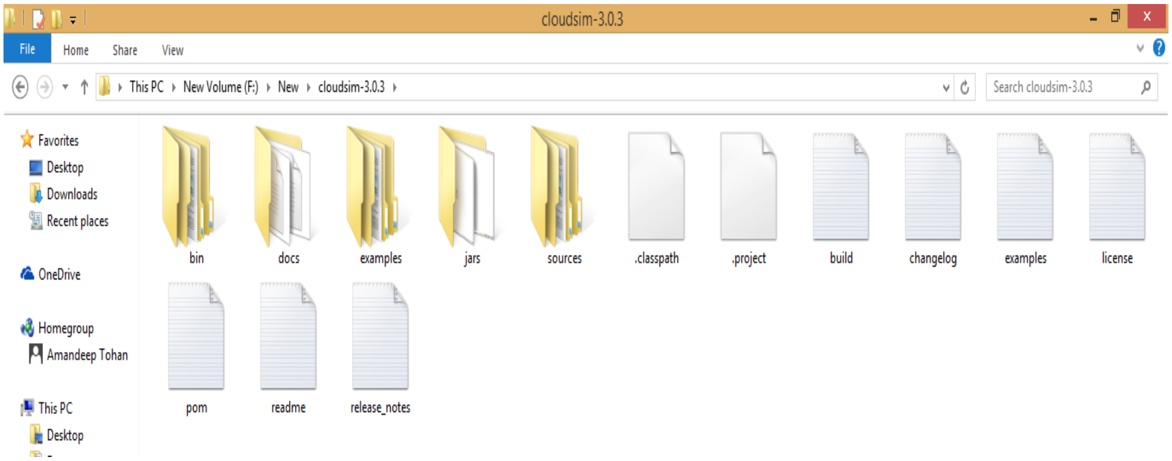


Figure: 4.1. Extracted files and directories of cloud-sim 3.0.3

To start working with cloud-sim we need to import it in eclipse. Following are the steps:

1. Install java compiler: Example: jdk-8-windows—586
2. Download and install Eclipse

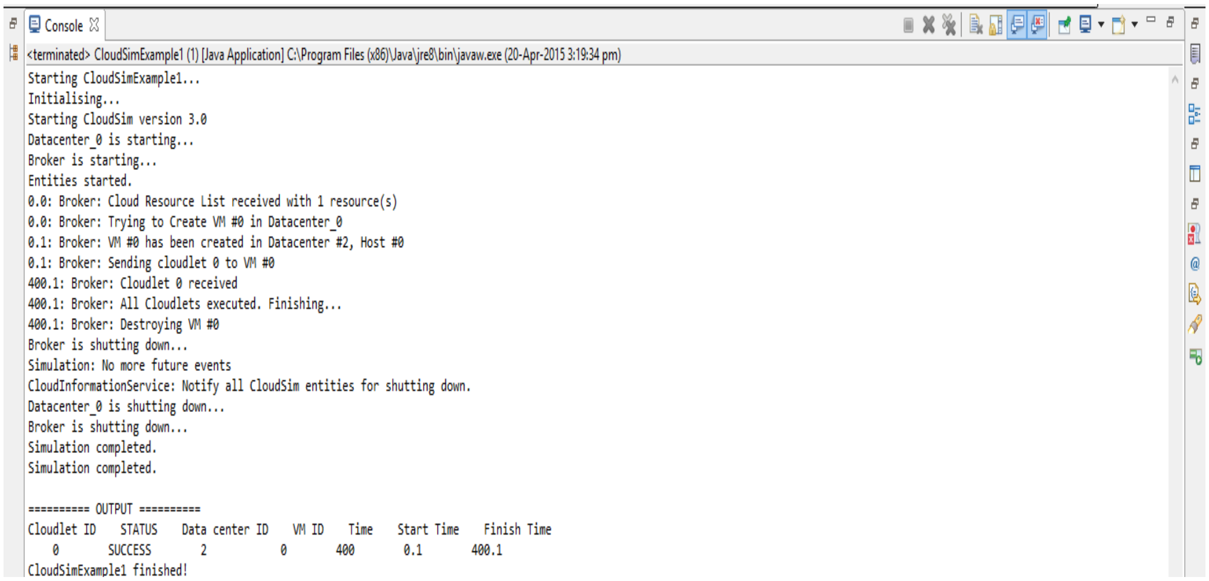
   Just double click on eclipse icon and eclipse will start and ask you to give the name of workspace (for beginners) you want to work in.
3. One can create new project by selecting File>New>Project>java project and Next. Now, give the name of project and the project will be added into a current workspace or one can change the location. Initially new project folder will just contains SRC directory and JRE system library. Later one can import files and folders required.
4. Adding libraries of cloud-sim

   Locate the cloudsim-3.0.3.jar file in jars directory of already downloaded and extracted cloud-sim folder. And now paste the same in the project created. Example folder in cloud-sim includes examples related to cloud environment, network, network datacentres, power, power planetlab, power random. We need the first one i.e. org.cloudbus.clooudsim.examples. It contains 8 examples.

   Let us quickly discuss the first example. The example contains:

   1. cloudlets -- which holds the task to be perform
   2. cloudletlist – the list of all the cloudlets in the given example
   3. createBrocker() : DatacentreBrocker
   4. createDatacentres(String) : Datacentre
   5. main(String[]) :void
   6. printCloudletList(List<Cloudlet>) : void

This example shows how to create one datacentre and with one host on it and run one cloud let on it. Following is the console output of the first example execution, which shows the initialization of datacentre, start of broker, entities will be executed, shut down of broker, datacentre shutting down and completion of simulation.



```
<terminated> CloudSimExample1 (1) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (20-Apr-2015 3:19:34 pm)
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0        SUCCESS        2            0       400       0.1       400.1
CloudSimExample1 finished!
```

Figure: 4.2. Console output of first example in cloudsim

4.2.Discussion about implementation of methodology

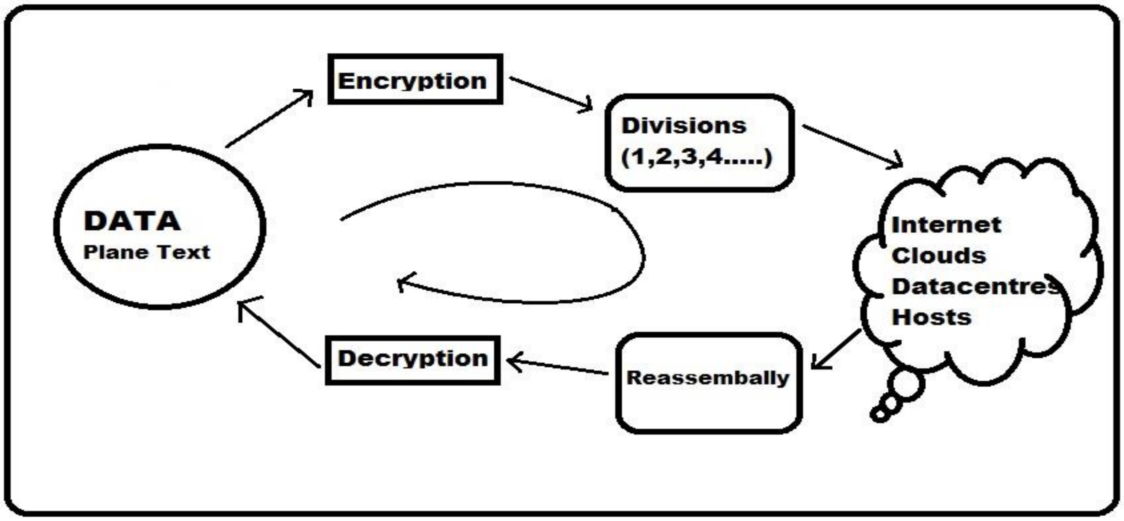
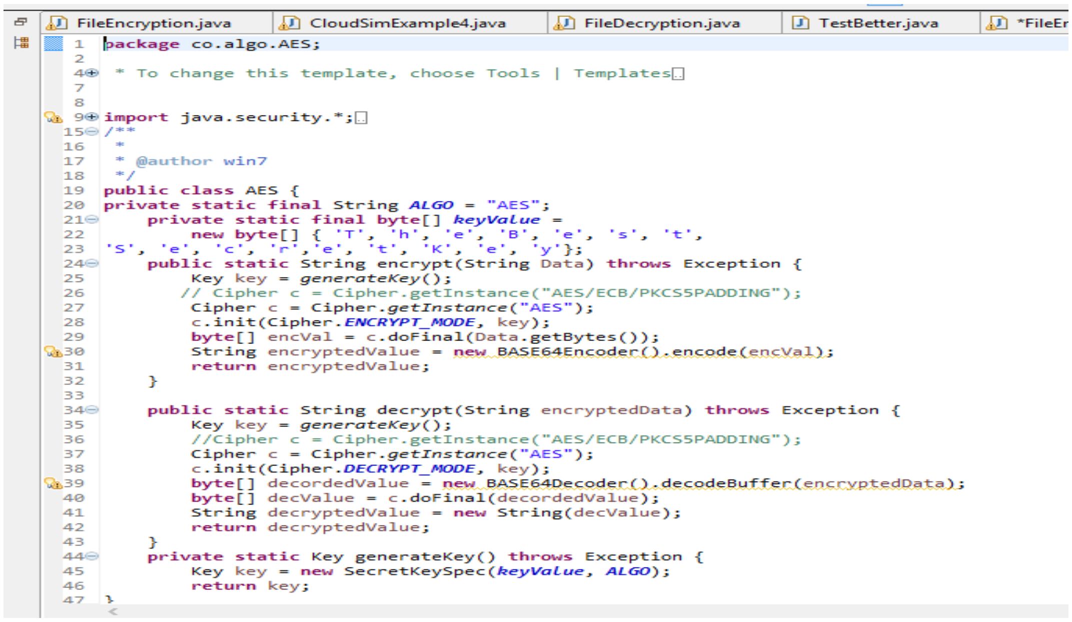


Figure: 4.3. Stepwise execution

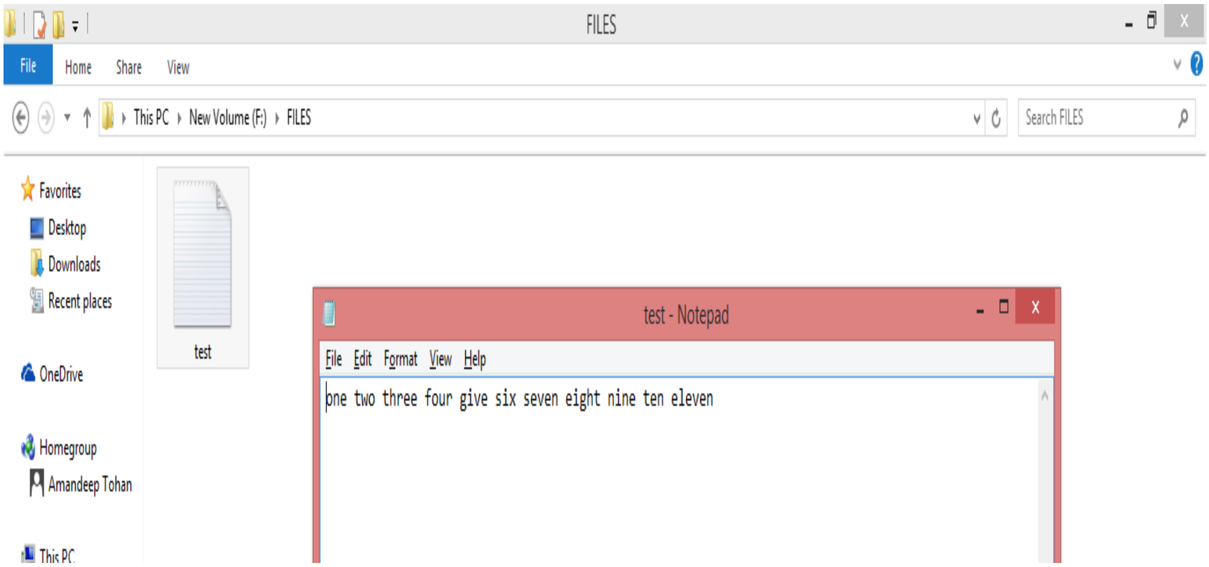
The first phase of methodology is Data encription/decription. AES the Advance Encryption Standerd algorithm is being used for encryption and decryption of the original data.



```java
package co.algo.AES;

 * To change this template, choose Tools | Templates


import java.security.*;
/**
 *
 * @author win7
 */
public class AES {
private static final String ALGO = "AES";
    private static final byte[] keyValue =
        new byte[] { 'T', 'h', 'e', 'B', 'e', 's', 't',
'S', 'e', 'c', 'r','e', 't', 'K', 'e', 'y'};
    public static String encrypt(String Data) throws Exception {
        Key key = generateKey();
       // Cipher c = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        Cipher c = Cipher.getInstance("AES");
        c.init(Cipher.ENCRYPT_MODE, key);
        byte[] encVal = c.doFinal(Data.getBytes());
        String encryptedValue = new BASE64Encoder().encode(encVal);
        return encryptedValue;
    }

    public static String decrypt(String encryptedData) throws Exception {
        Key key = generateKey();
        //Cipher c = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        Cipher c = Cipher.getInstance("AES");
        c.init(Cipher.DECRYPT_MODE, key);
        byte[] decordedValue = new BASE64Decoder().decodeBuffer(encryptedData);
        byte[] decValue = c.doFinal(decordedValue);
        String decryptedValue = new String(decValue);
        return decryptedValue;
    }
    private static Key generateKey() throws Exception {
        Key key = new SecretKeySpec(keyValue, ALGO);
        return key;
}
```

Figure: 4.4. Advance Encryption Standared

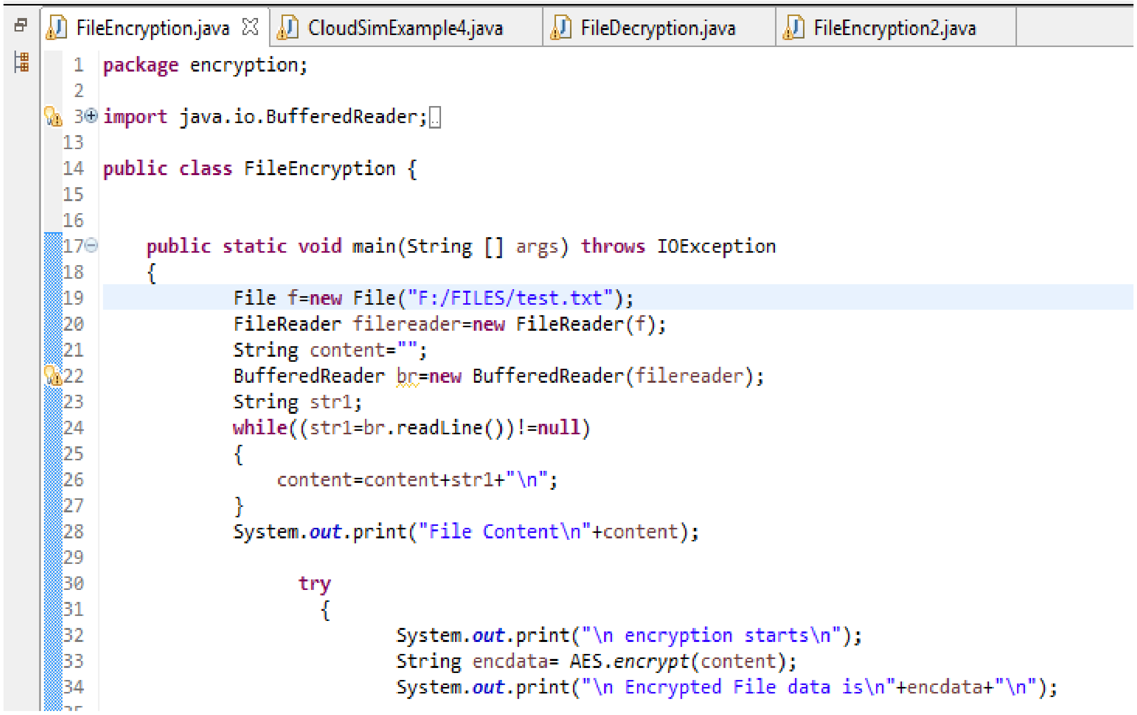
Let say I am the cloud user and want to save my data on cloud. The data will be first converted into ciphertext. Following screen short display a text file with name “test” located at F:/FILES/ and it containa text as “ one two three four give six seven eight nine ten eleven “



one two three four give six seven eight nine ten eleven

Figure: 4.5. File containing plain text

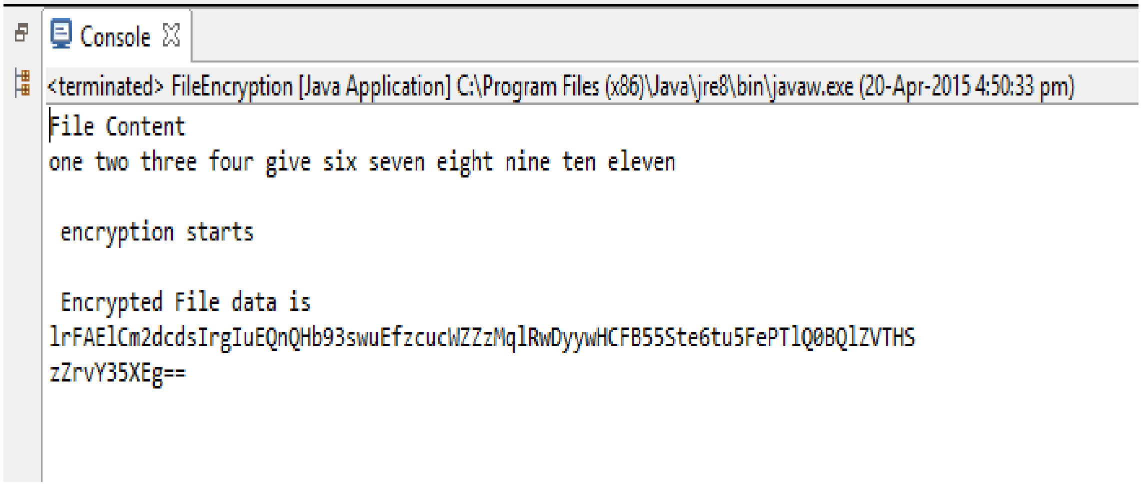
Following program is used to convert the above mentioned content to be in encrypted form.



```
package encryption;

import java.io.BufferedReader;

public class FileEncryption {


    public static void main(String [] args) throws IOException
    {
            File f=new File("F:/FILES/test.txt");
            FileReader filereader=new FileReader(f);
            String content="";
            BufferedReader br=new BufferedReader(filereader);
            String str1;
            while((str1=br.readLine())!=null)
            {
                content=content+str1+"\n";
            }
            System.out.print("File Content\n"+content);

                try
                  {
                        System.out.print("\n encryption starts\n");
                        String encdata= AES.encrypt(content);
                        System.out.print("\n Encrypted File data is\n"+encdata+"\n");
```

Figure: 4.6. Encryption

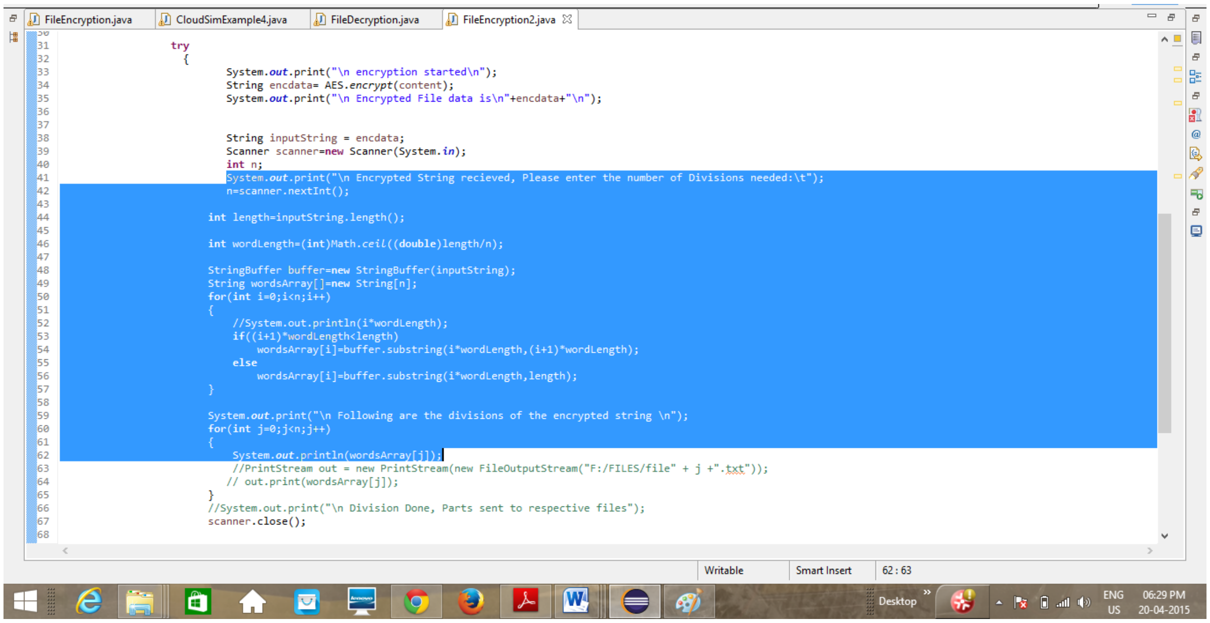
Output: the encrypted data is a continuous string of characters and digits.



```
<terminated> FileEncryption [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (20-Apr-2015 4:50:33 pm)
File Content
one two three four give six seven eight nine ten eleven

 encryption starts

 Encrypted File data is
lrFAElCm2dcdsIrgIuEQnQHb93swuEfzcucWZZzMqlRwDyywHCFB55Ste6tu5FePTlQ0BQlZVTHS
zZrvY35XEg==
```

Figure: 4.7. Cipher Text after encryption

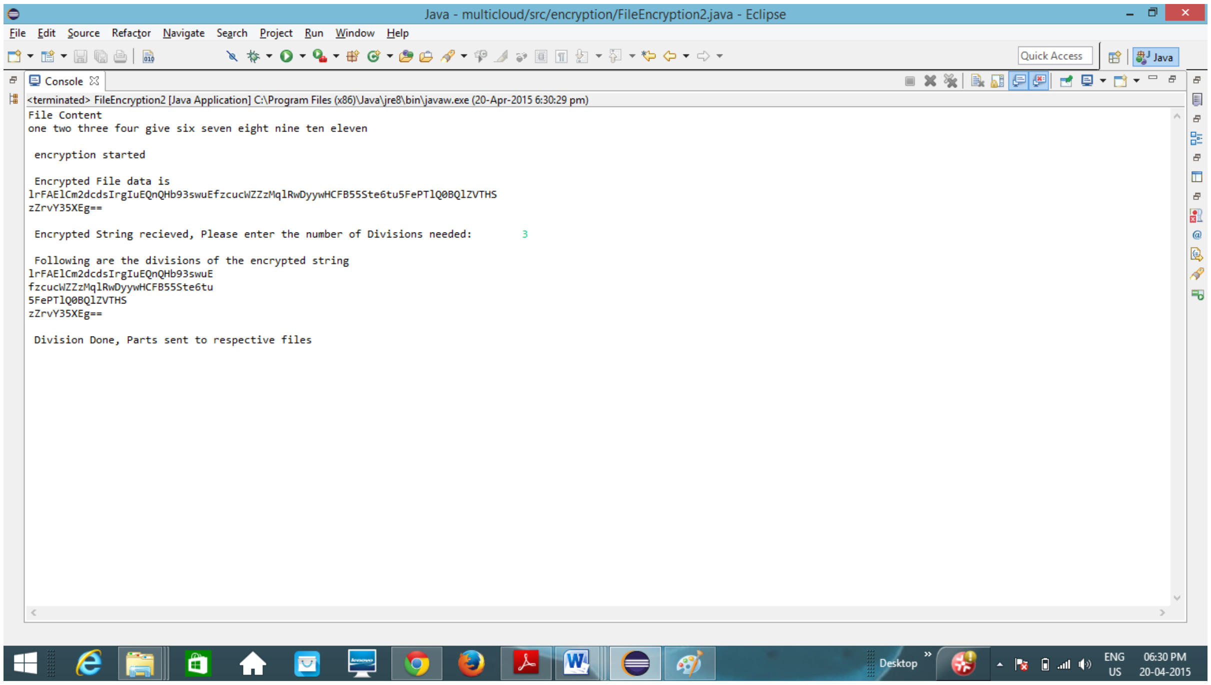
Now the next step is to divide this continuous string into substrings, equal in numbers to the number of clouds available. The substrings will be store into new files automatically created. Let us discuss case when we have 3 clouds.



```
                    try
                     {
                            System.out.print("\n encryption started\n");
                            String encdata= AES.encrypt(content);
                            System.out.print("\n Encrypted File data is\n"+encdata+"\n");


                            String inputString = encdata;
                            Scanner scanner=new Scanner(System.in);
                            int n;
                            System.out.print("\n Encrypted String recieved, Please enter the number of Divisions needed:\t");
                            n=scanner.nextInt();

                         int length=inputString.length();

                         int wordLength=(int)Math.ceil((double)length/n);

                         StringBuffer buffer=new StringBuffer(inputString);
                         String wordsArray[]=new String[n];
                         for(int i=0;i<n;i++)
                         {
                             //System.out.println(i*wordLength);
                             if((i+1)*wordLength<length)
                                 wordsArray[i]=buffer.substring(i*wordLength,(i+1)*wordLength);
                             else
                                 wordsArray[i]=buffer.substring(i*wordLength,length);
                         }

                         System.out.print("\n Following are the divisions of the encrypted string \n");
                         for(int j=0;j<n;j++)
                         {
                             System.out.println(wordsArray[j]);
                             //PrintStream out = new PrintStream(new FileOutputStream("F:/FILES/file" + j +".txt"));
                            // out.print(wordsArray[j]);
                         }
                         //System.out.print("\n Division Done, Parts sent to respective files");
                         scanner.close();
```

Figure: 4.8. Code to divide string

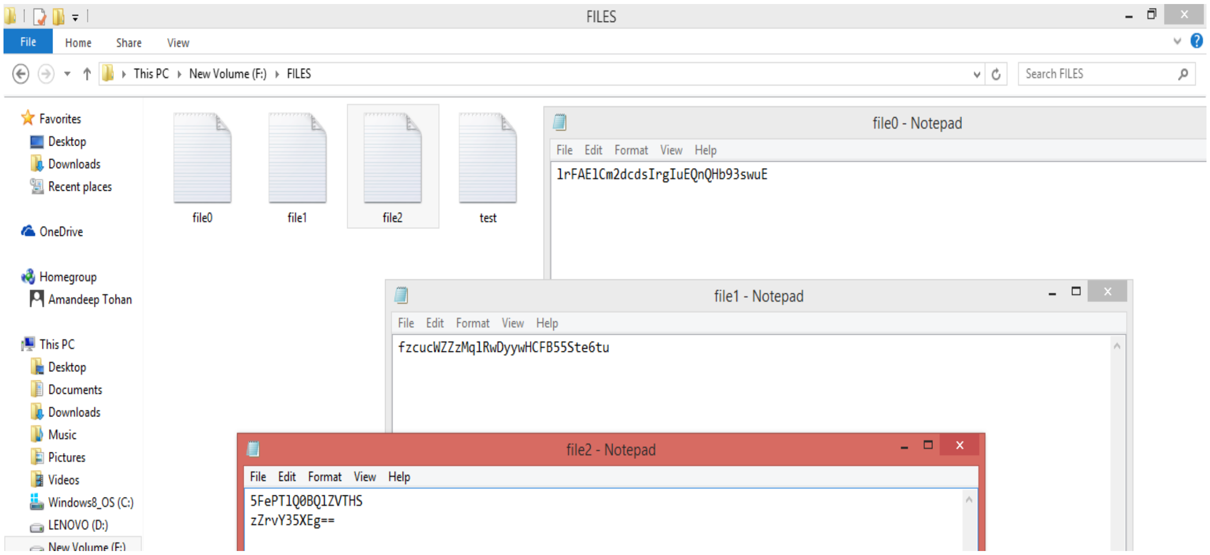
Output of the above code:



```
<terminated> FileEncryption2 [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (20-Apr-2015 6:30:29 pm)
File Content
one two three four give six seven eight nine ten eleven

 encryption started

 Encrypted File data is
lrFAElCm2dcdsIrgIuEQnQHb93swuEfzcucWZZzMqlRwDyywHCFB55Ste6tu5FePTlQ0BQlZVTHS
zZrvY35XEg==

 Encrypted String recieved, Please enter the number of Divisions needed:	3

 Following are the divisions of the encrypted string
lrFAElCm2dcdsIrgIuEQnQHb93swuE
fzcucWZZzMqlRwDyywHCFB55Ste6tu
5FePTlQ0BQlZVTHS
zZrvY35XEg==

 Division Done, Parts sent to respective files
```

Figure:4.9. String divided into 3 parts.

The substrings are sent to new files at specified location as follows:



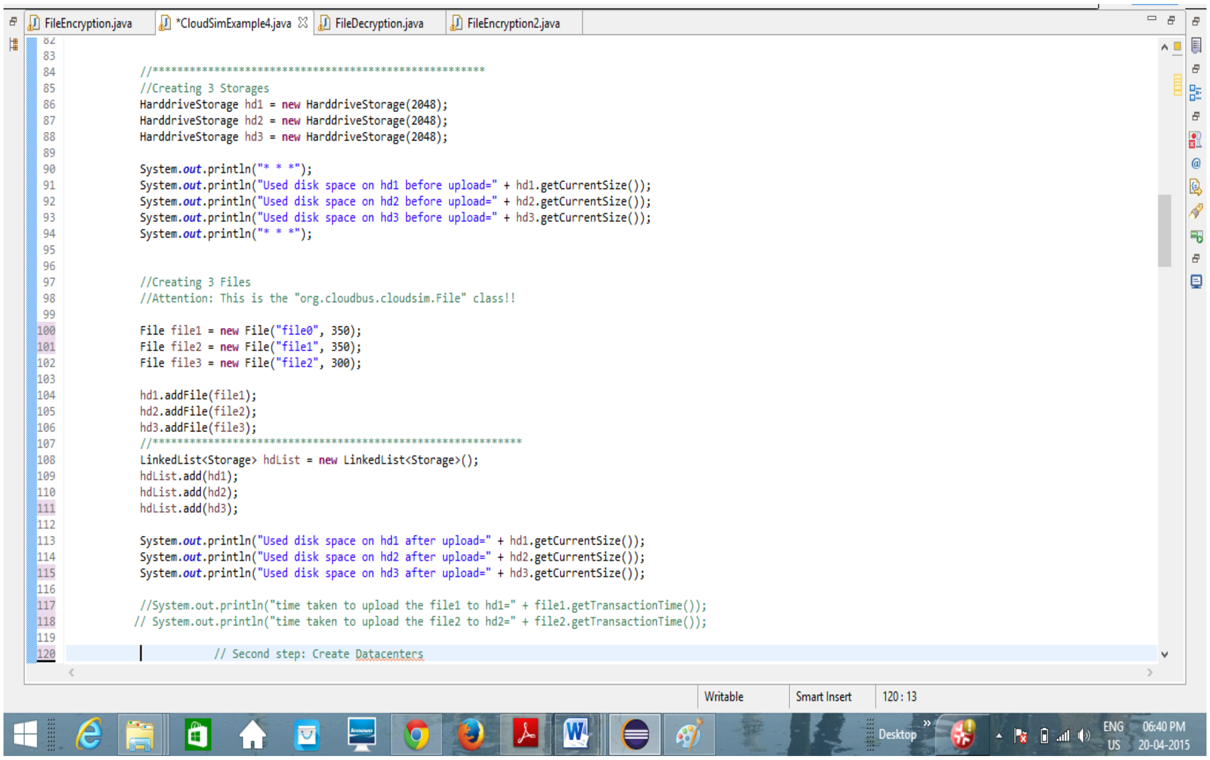
```
1rFAE1Cm2dcdsIrgIuEQnQHb93swuE
```

```
fzcucWZZzMq1RwDyywHCFB55Ste6tu
```

```
5FePT1Q0BQ1ZVTHS
zZrvY35XEg==
```

Figure:4.10. Three substrings received in three different files

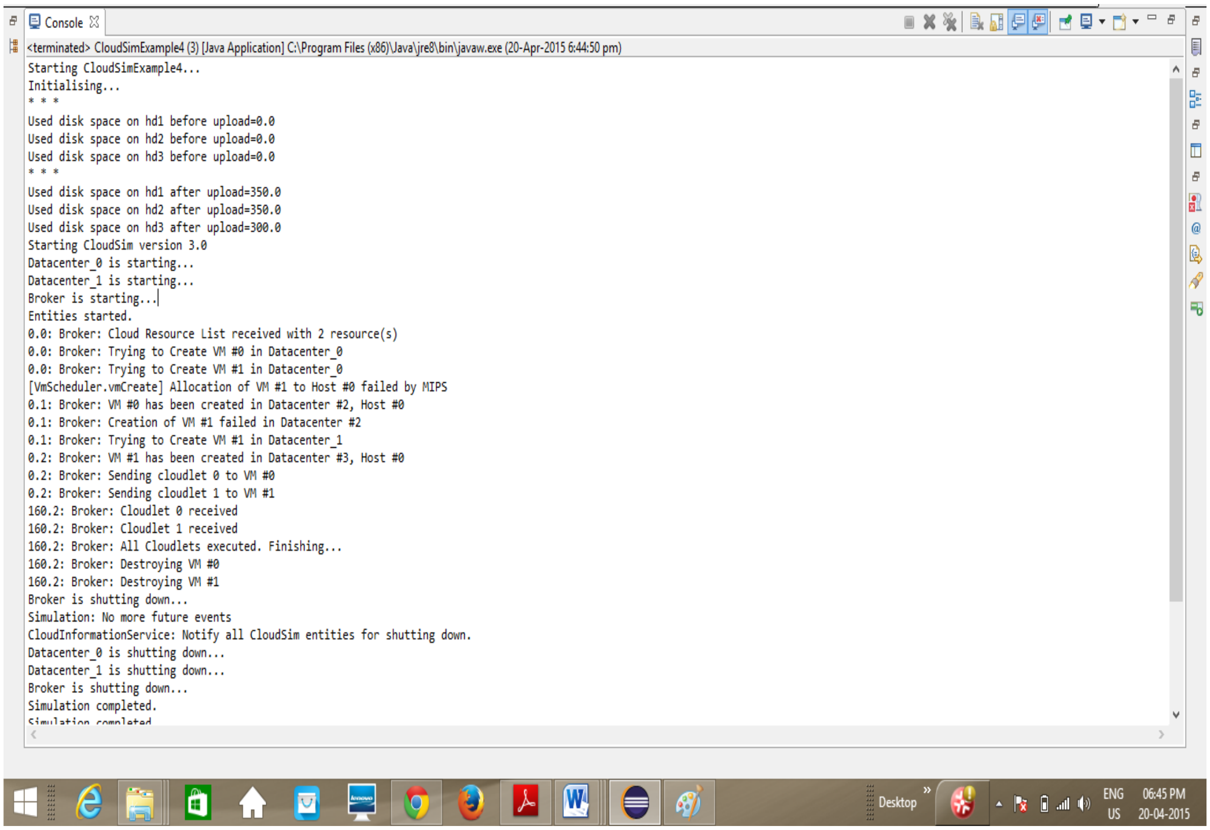
Encryption and division is done, now the next step is to store these files.



```
        //******************************************************
        //Creating 3 Storages
        HarddriveStorage hd1 = new HarddriveStorage(2048);
        HarddriveStorage hd2 = new HarddriveStorage(2048);
        HarddriveStorage hd3 = new HarddriveStorage(2048);

        System.out.println("* * *");
        System.out.println("Used disk space on hd1 before upload=" + hd1.getCurrentSize());
        System.out.println("Used disk space on hd2 before upload=" + hd2.getCurrentSize());
        System.out.println("Used disk space on hd3 before upload=" + hd3.getCurrentSize());
        System.out.println("* * *");


        //Creating 3 Files
        //Attention: This is the "org.cloudbus.cloudsim.File" class!!

        File file1 = new File("file0", 350);
        File file2 = new File("file1", 350);
        File file3 = new File("file2", 300);

        hd1.addFile(file1);
        hd2.addFile(file2);
        hd3.addFile(file3);
        //************************************************************
        LinkedList<Storage> hdList = new LinkedList<Storage>();
        hdList.add(hd1);
        hdList.add(hd2);
        hdList.add(hd3);

        System.out.println("Used disk space on hd1 after upload=" + hd1.getCurrentSize());
        System.out.println("Used disk space on hd2 after upload=" + hd2.getCurrentSize());
        System.out.println("Used disk space on hd3 after upload=" + hd3.getCurrentSize());

        //System.out.println("time taken to upload the file1 to hd1=" + file1.getTransactionTime());
       // System.out.println("time taken to upload the file2 to hd2=" + file2.getTransactionTime());

        |           // Second step: Create Datacenters
```

Figure:4.11. Storing 3 files in 3 different storages

Three virtual files file0,file1,file2 of sizes 350,350,300 are stored into three virtul storage hd1,hd2,hd3. For testing purpous we have checked the used storage before and after the file stored in the storage. Which can be display in following output of the code.

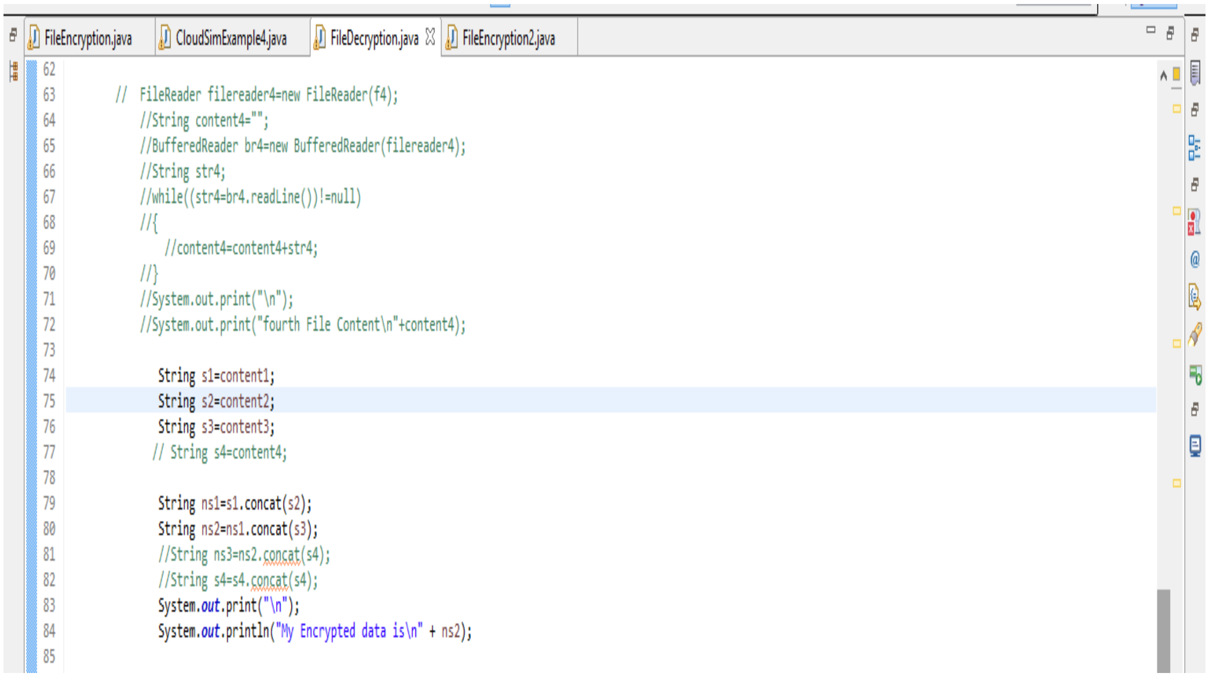


```
<terminated> CloudSimExample4 (3) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (20-Apr-2015 6:44:50 pm)
Starting CloudSimExample4...
Initialising...
* * *
Used disk space on hd1 before upload=0.0
Used disk space on hd2 before upload=0.0
Used disk space on hd3 before upload=0.0
* * *
Used disk space on hd1 after upload=350.0
Used disk space on hd2 after upload=350.0
Used disk space on hd3 after upload=300.0
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed
```

Figure:4.12. Storage before and after file transfer

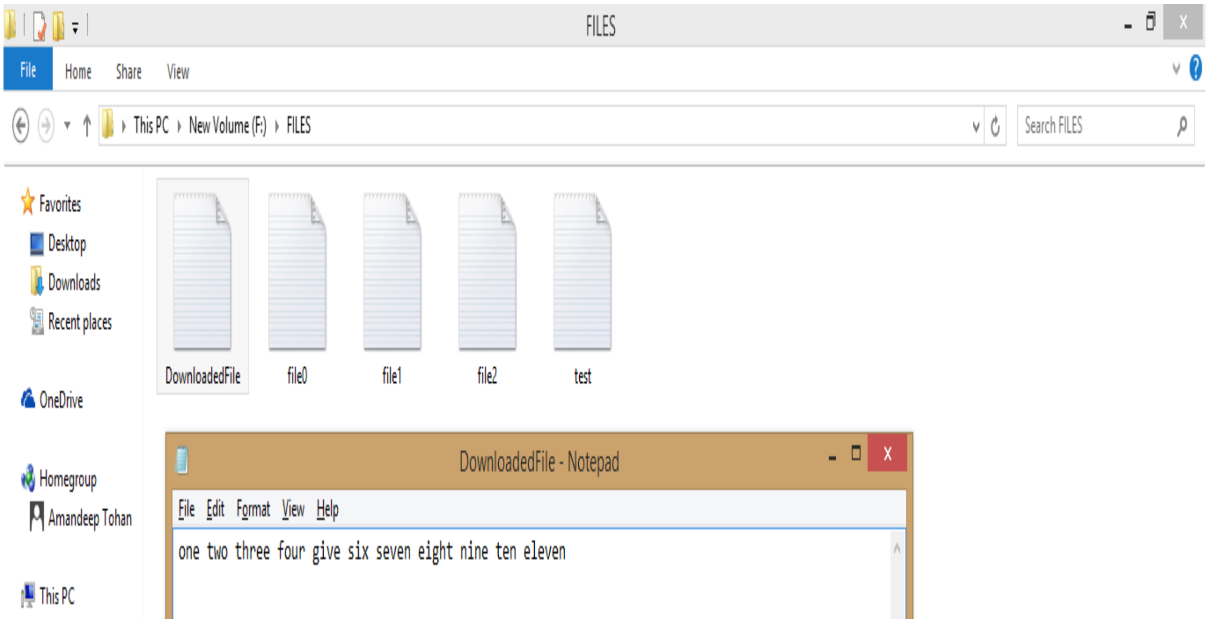
Distribution part of methodology is done uptill here. Now the next we will discuss how the original file will regenerated from the encrypted part files downloaded from the clouds. That means the reassembally and decryption phase.

The contents of the downloaded file at specified location will be saved into a buffer and these substring will be concatinated to generate the original encrypted string.



```
   //  FileReader filereader4=new FileReader(f4);
       //String content4="";
       //BufferedReader br4=new BufferedReader(filereader4);
       //String str4;
       //while((str4=br4.readLine())!=null)
       //{
           //content4=content4+str4;
       //}
       //System.out.print("\n");
       //System.out.print("fourth File Content\n"+content4);

          String s1=content1;
          String s2=content2;
          String s3=content3;
        // String s4=content4;

          String ns1=s1.concat(s2);
          String ns2=ns1.concat(s3);
          //String ns3=ns2.concat(s4);
          //String s4=s4.concat(s4);
          System.out.print("\n");
          System.out.println("My Encrypted data is\n" + ns2);
```

Figure:4.13. Concatinating the substrings



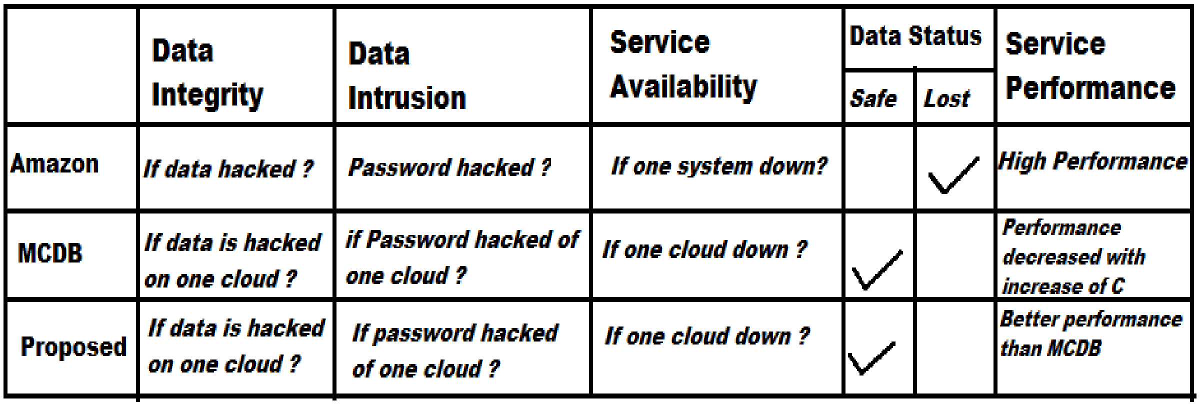
```
one two three four give six seven eight nine ten eleven
```

Figure:4.14. Original plain text genrated after concatination and decryption.

We can see that the original text of “test” file is received in the new file “DownloadedFile”.

4.3. Comparisons and improvement

Let us compare the Single cloud system, the base paper MCDB model and proposed model.



	Data Integrity	Data Intrusion	Service Availability	Data Status		Service Performance
				Safe	Lost	
Amazon	*If data hacked ?*	*Password hacked ?*	*If one system down?*		✓	*High Performance*
MCDB	*If data is hacked on one cloud ?*	*if Password hacked of one cloud ?*	*If one cloud down ?*	✓		*Performance decreased with increase of C*
Proposed	*If data is hacked on one cloud ?*	*If password hacked of one cloud ?*	*If one cloud down ?*	✓		*Better performance than MCDB*

Table:4.1. Comparing single, MCDB and Proposed cloud model

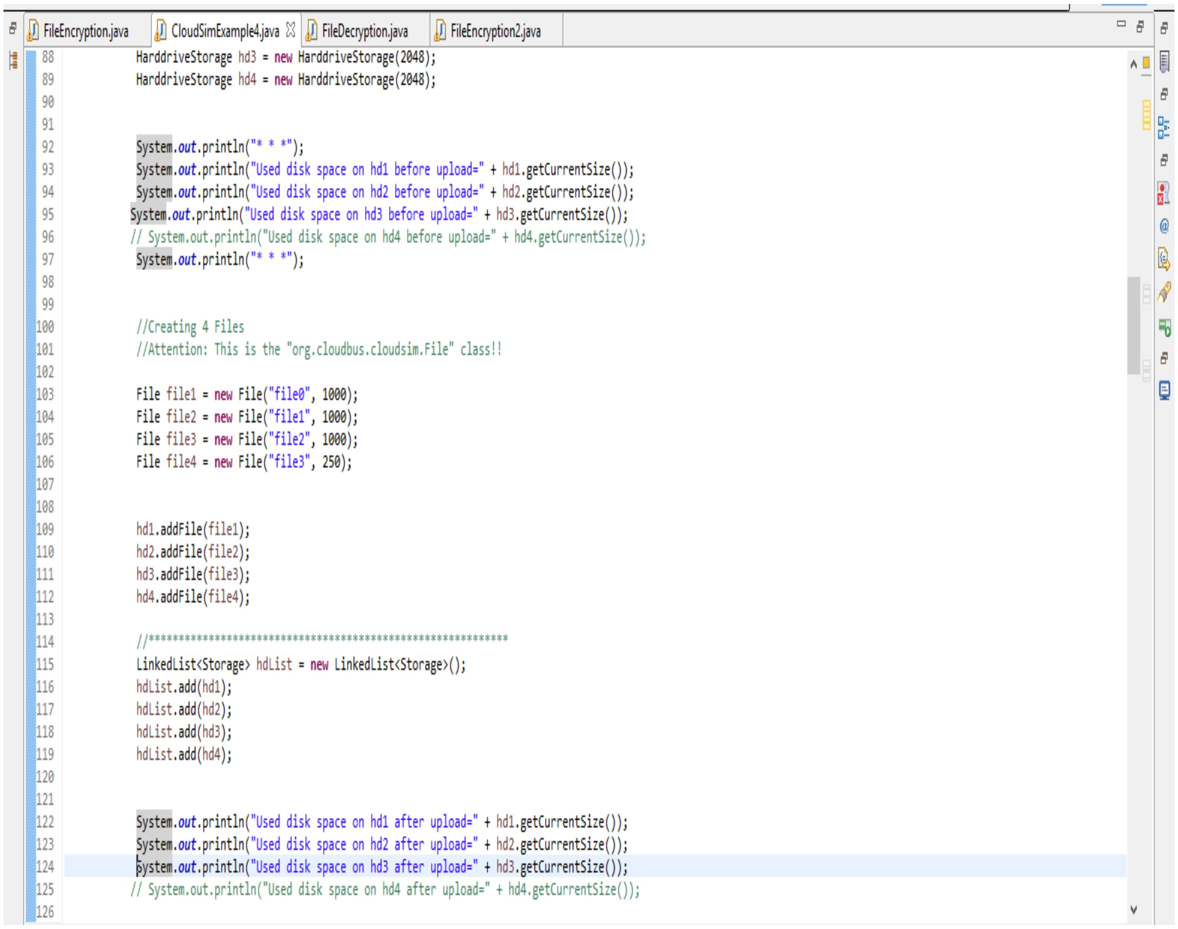
Data Integrity, i.e. authorized used should get the same value of data that he has stored at cloud. At single cloud system, example Amazon, if the data is hacked, user will lose the data. In MCDB and Proposed approach the data is safe, if one cloud data is hacked the data can be retrieve from other clouds.

Data Intrusion, i.e. confidential data should not be available to any unauthorized user, not only to write buy even for reading. In single cloud, if the owner of the account lost his account password or the password is hacked, steeled by hacked then there is a possibility that intrude mat read the data without the knowledge of data owner. But in MCDB and Proposed approach the data available to intruder will be some part of the data and that too could not be converted into readable content without availability of other parts of the same data.

Data Availability, i.e. availability of data should hold. In single cloud system if a single host or system that is either partially responsible for the availability of data is down then the data will not be available to user. The reason of system down could be any, DOS attack, high load, system failure etc. But in MCDB and proposed approach the availability is high and the data will be available if in case the complete single cloud gone down. Data is safe.

Performance measures in the units of time taken to store the data into the cloud(s) or units of time taken to retrieve the data from the cloud(s). In a single cloud the performance is high, because one needs to store and retrieve the same amount of data and too from only a single location. The performance decreased in MCDB as the number of clouds increase, i.e. more the number of clouds more security will be there but the time taken to store and retrieve increases with increase of the number of clouds. Now, in proposed approach we store the data in multiple clouds in such a way that the high security of MCDB should remain the same and the performance should be comparatively increased. Therefore the proposed model is better than the base paper's MCDB and single cloud model.

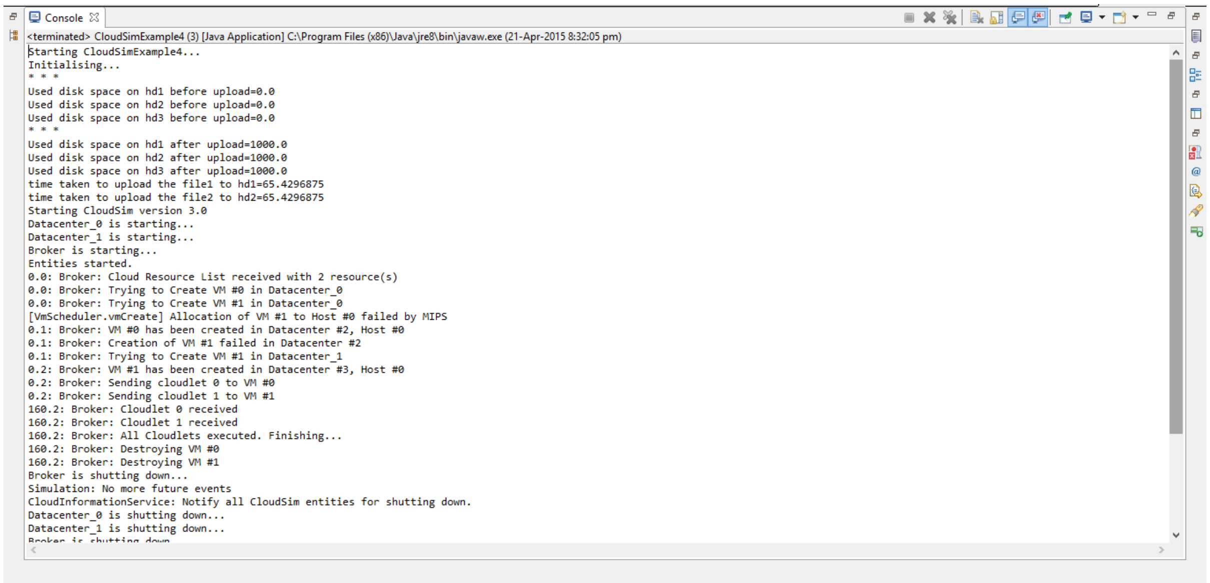
Now, let us discuss how this concept holds in implementation.



```
            HarddriveStorage hd3 = new HarddriveStorage(2048);
            HarddriveStorage hd4 = new HarddriveStorage(2048);


            System.out.println("* * *");
            System.out.println("Used disk space on hd1 before upload=" + hd1.getCurrentSize());
            System.out.println("Used disk space on hd2 before upload=" + hd2.getCurrentSize());
           System.out.println("Used disk space on hd3 before upload=" + hd3.getCurrentSize());
           // System.out.println("Used disk space on hd4 before upload=" + hd4.getCurrentSize());
            System.out.println("* * *");


            //Creating 4 Files
            //Attention: This is the "org.cloudbus.cloudsim.File" class!!

            File file1 = new File("file0", 1000);
            File file2 = new File("file1", 1000);
            File file3 = new File("file2", 1000);
            File file4 = new File("file3", 250);


            hd1.addFile(file1);
            hd2.addFile(file2);
            hd3.addFile(file3);
            hd4.addFile(file4);

            //**********************************************************
            LinkedList<Storage> hdList = new LinkedList<Storage>();
            hdList.add(hd1);
            hdList.add(hd2);
            hdList.add(hd3);
            hdList.add(hd4);


            System.out.println("Used disk space on hd1 after upload=" + hd1.getCurrentSize());
            System.out.println("Used disk space on hd2 after upload=" + hd2.getCurrentSize());
            System.out.println("Used disk space on hd3 after upload=" + hd3.getCurrentSize());
           // System.out.println("Used disk space on hd4 after upload=" + hd4.getCurrentSize());

```

Figure:4.15. Code to store 1000 units files

Total 3 clouds, MCDB approach, equal amount of data at all clouds.



```
<terminated> CloudSimExample4 (3) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (21-Apr-2015 8:32:05 pm)
Starting CloudSimExample4...
Initialising...
* * *
Used disk space on hd1 before upload=0.0
Used disk space on hd2 before upload=0.0
Used disk space on hd3 before upload=0.0
* * *
Used disk space on hd1 after upload=1000.0
Used disk space on hd2 after upload=1000.0
Used disk space on hd3 after upload=1000.0
time taken to upload the file1 to hd1=65.4296875
time taken to upload the file2 to hd2=65.4296875
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down
```

Figure: 4.16. Console output showing transaction time

When equal amount of data is being stored to all clouds, the amount of time taken should also be equal. In the above experiment when 1000 units of data is being store to each cloud, the amount of time taken is 65.42 if in parallel case otherwise 65.42*3 units approximately.

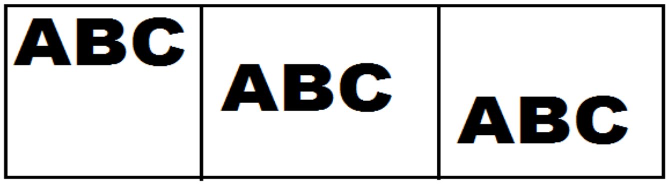


Figure: 4.17. Storage pattern in MCDB

Similarly, in case of retrieval, the time taken to get whole data will be equal to the time taken to get data from one cloud in case data is fetched in parallel form.

Now let us compare the output if the data is divided into 3 equal sub parts (1000/3=333.33, approximately 333).

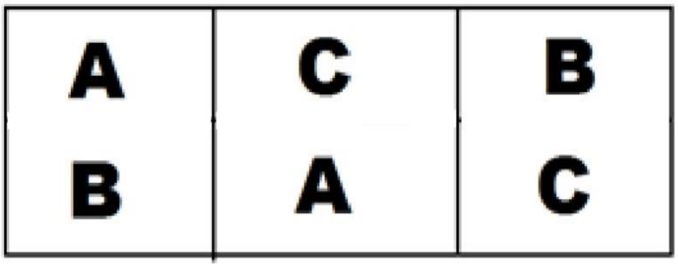
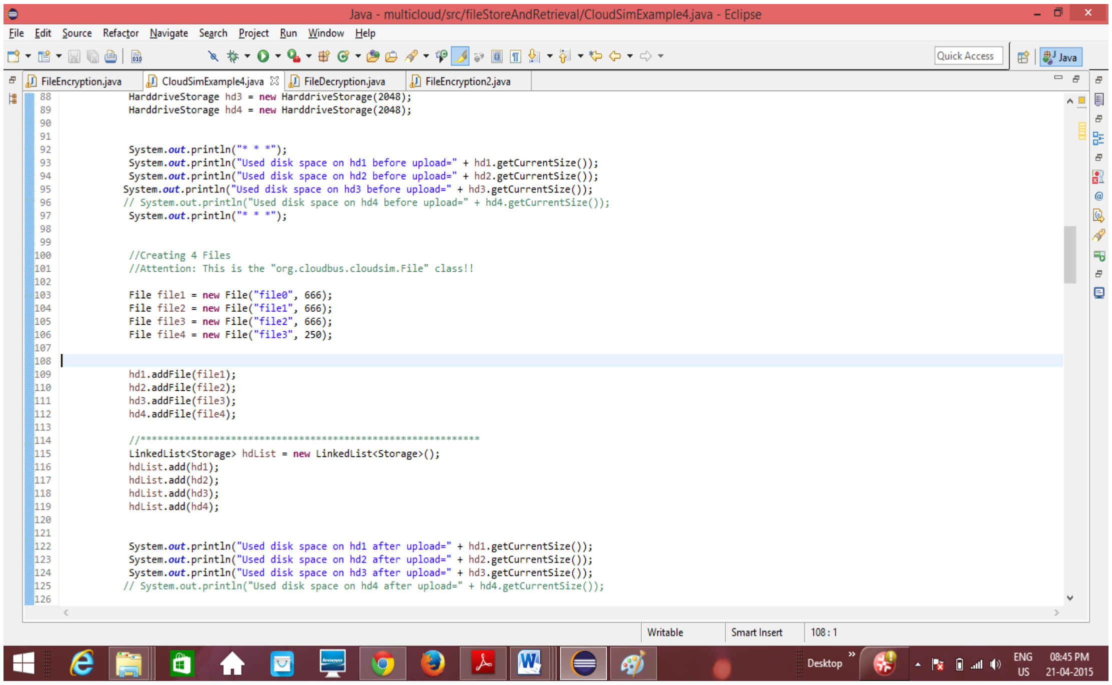


Figure: 4.18. Storage pattern in proposed model

In case of 3 clouds, each cloud holds 2 parts of data.



```
            HarddriveStorage hd3 = new HarddriveStorage(2048);
            HarddriveStorage hd4 = new HarddriveStorage(2048);


            System.out.println("* * *");
            System.out.println("Used disk space on hd1 before upload=" + hd1.getCurrentSize());
            System.out.println("Used disk space on hd2 before upload=" + hd2.getCurrentSize());
           System.out.println("Used disk space on hd3 before upload=" + hd3.getCurrentSize());
           // System.out.println("Used disk space on hd4 before upload=" + hd4.getCurrentSize());
            System.out.println("* * *");


            //Creating 4 Files
            //Attention: This is the "org.cloudbus.cloudsim.File" class!!

            File file1 = new File("file0", 666);
            File file2 = new File("file1", 666);
            File file3 = new File("file2", 666);
            File file4 = new File("file3", 250);


            hd1.addFile(file1);
            hd2.addFile(file2);
            hd3.addFile(file3);
            hd4.addFile(file4);

            //***********************************************************
            LinkedList<Storage> hdList = new LinkedList<Storage>();
            hdList.add(hd1);
            hdList.add(hd2);
            hdList.add(hd3);
            hdList.add(hd4);


            System.out.println("Used disk space on hd1 after upload=" + hd1.getCurrentSize());
            System.out.println("Used disk space on hd2 after upload=" + hd2.getCurrentSize());
            System.out.println("Used disk space on hd3 after upload=" + hd3.getCurrentSize());
           // System.out.println("Used disk space on hd4 after upload=" + hd4.getCurrentSize());
```

Figure: 4.19. Storage containing 2 of the 3 parts (666 units)

Now, first thing is that to upload the data of 2 parts in each cloud, instead of all 3 parts at each cloud will be less. But User will be able to fetch 2 parts from any one cloud and 1 part from any other cloud.

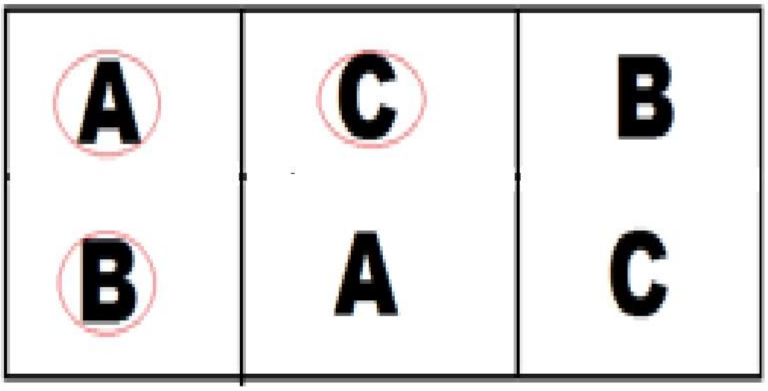
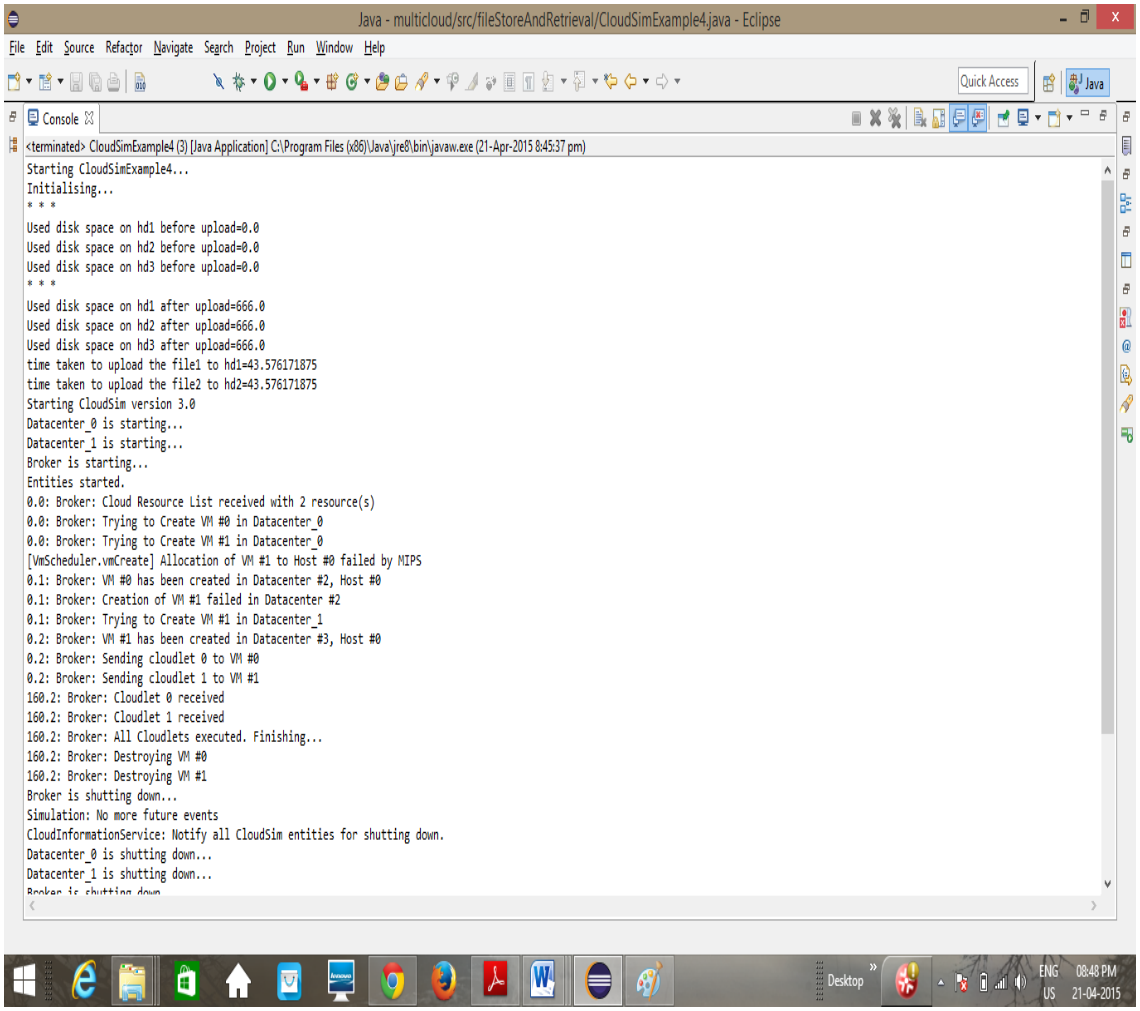


Figure: 4.20. All 3 parts from any 2 clouds

Therefore the maximum time taken to retrieve the data will be equal to the time taken to retrieve those 2 parts from single cloud (21.78*2=43.56 units). Because the third part will be retrieve in parallel from other cloud. On the other hand if data is being fetch serially, then the total amount of time taken will equal to the time taken to fetch those sub parts only, not the replicas of the original data.


Java - multicloud/src/fileStoreAndRetrieval/CloudSimExample4.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Console
Desktop
ENG US
08:48 PM
21-04-2015

```
<terminated> CloudSimExample4 (3) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (21-Apr-2015 8:45:37 pm)
Starting CloudSimExample4...
Initialising...
* * *
Used disk space on hd1 before upload=0.0
Used disk space on hd2 before upload=0.0
Used disk space on hd3 before upload=0.0
* * *
Used disk space on hd1 after upload=666.0
Used disk space on hd2 after upload=666.0
Used disk space on hd3 after upload=666.0
time taken to upload the file1 to hd1=43.576171875
time taken to upload the file2 to hd2=43.576171875
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down
```

Figure: 4.21. Console output showing transaction time for two files.

Or in other case if data is being fetched in parallel from all the 3 clouds then the time taken to retrieve the whole data will be equal to the time taken to get a single part from each cloud (21.78 units only). Also, in serial case the total time taken will again be equal to the time taken to fetch only the parts not the replicas.

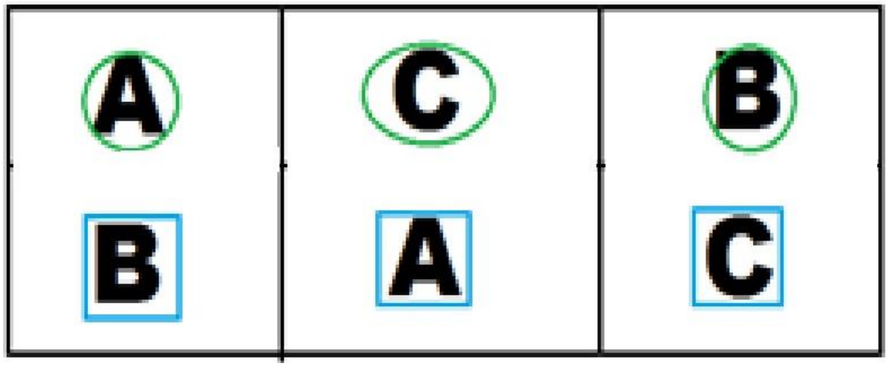
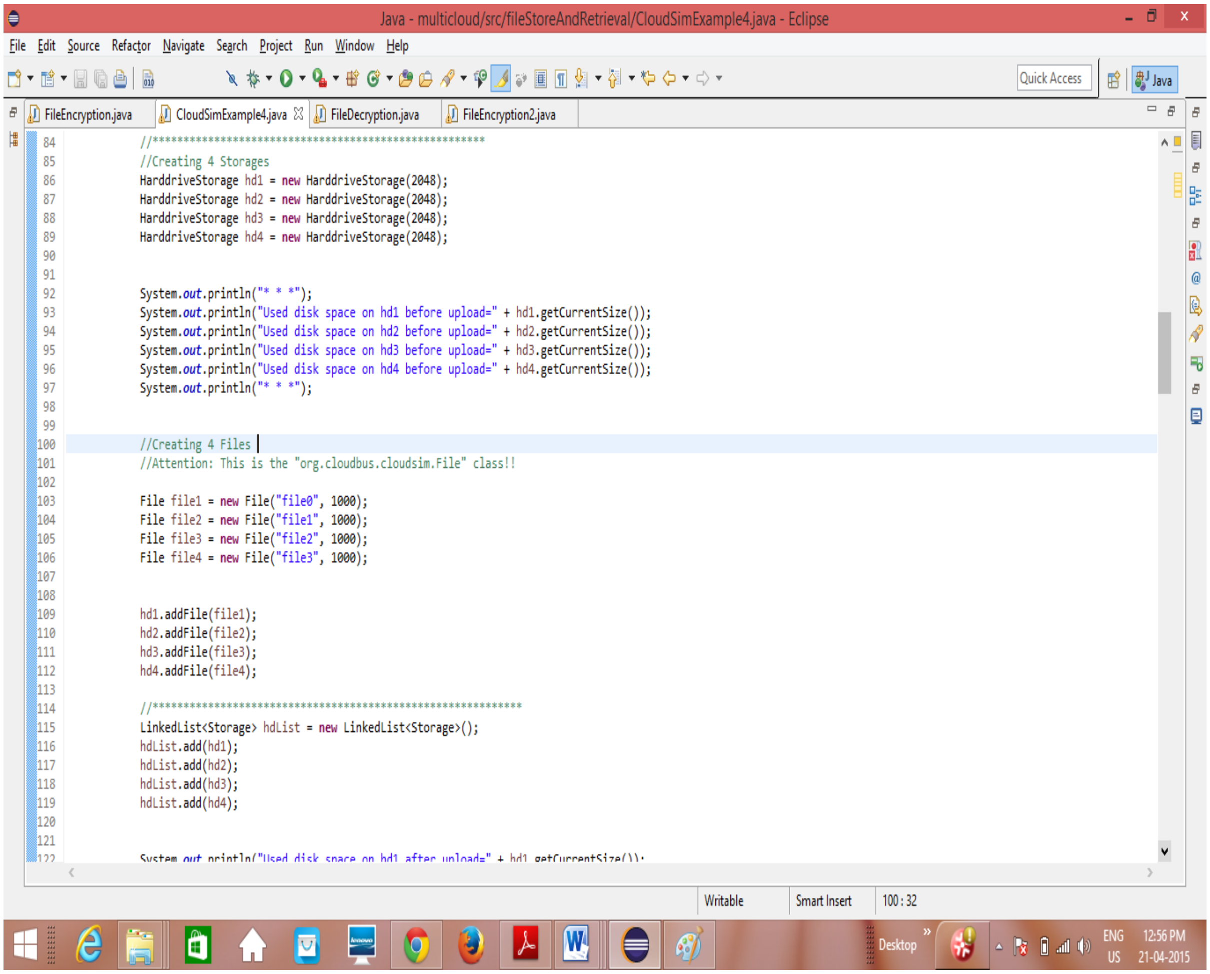


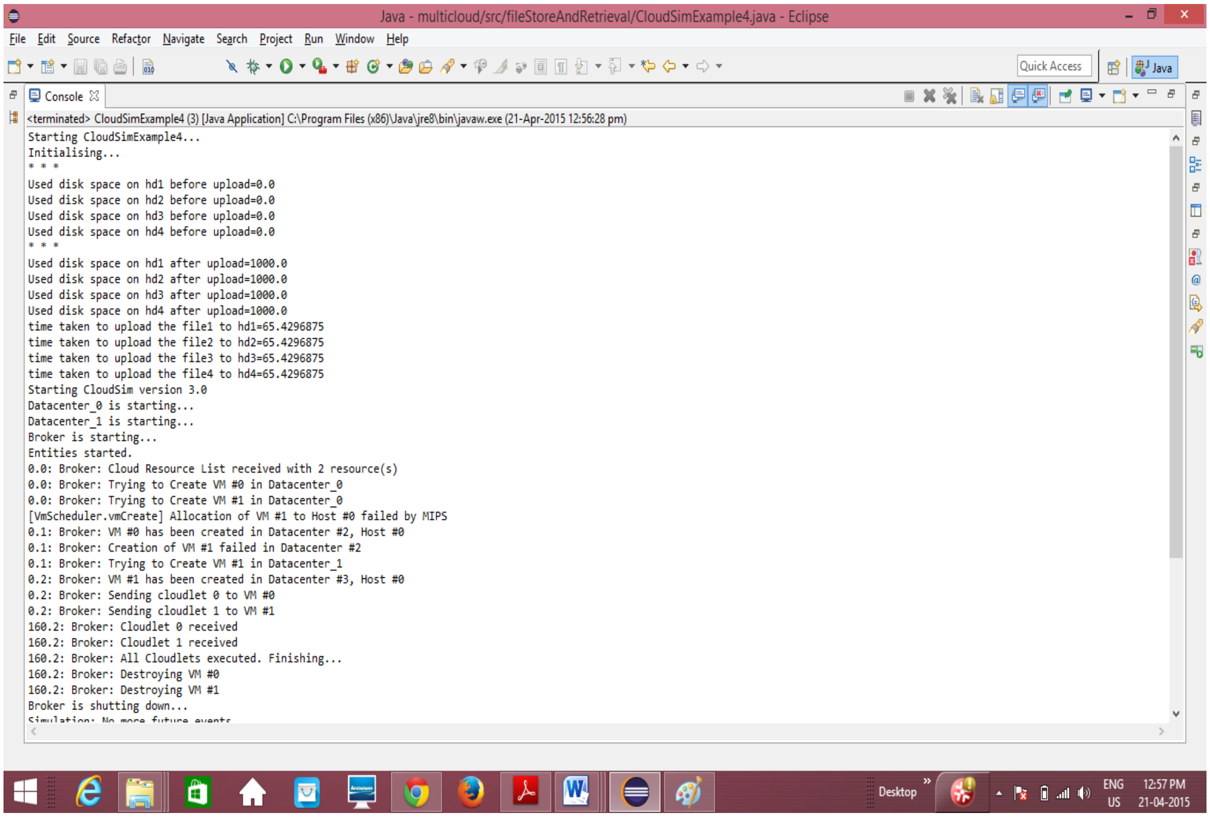
Figure: 4.22. All 3 parts received from all 3 clouds.

Similarly, in case of 4 clouds the security and performance will be even better.



```
        //******************************************************
        //Creating 4 Storages
        HarddriveStorage hd1 = new HarddriveStorage(2048);
        HarddriveStorage hd2 = new HarddriveStorage(2048);
        HarddriveStorage hd3 = new HarddriveStorage(2048);
        HarddriveStorage hd4 = new HarddriveStorage(2048);


        System.out.println("* * *");
        System.out.println("Used disk space on hd1 before upload=" + hd1.getCurrentSize());
        System.out.println("Used disk space on hd2 before upload=" + hd2.getCurrentSize());
        System.out.println("Used disk space on hd3 before upload=" + hd3.getCurrentSize());
        System.out.println("Used disk space on hd4 before upload=" + hd4.getCurrentSize());
        System.out.println("* * *");


        //Creating 4 Files
        //Attention: This is the "org.cloudbus.cloudsim.File" class!!

        File file1 = new File("file0", 1000);
        File file2 = new File("file1", 1000);
        File file3 = new File("file2", 1000);
        File file4 = new File("file3", 1000);


        hd1.addFile(file1);
        hd2.addFile(file2);
        hd3.addFile(file3);
        hd4.addFile(file4);

        //************************************************************
        LinkedList<Storage> hdList = new LinkedList<Storage>();
        hdList.add(hd1);
        hdList.add(hd2);
        hdList.add(hd3);
        hdList.add(hd4);


        System.out.println("Used disk space on hd1 after upload=" + hd1.getCurrentSize());
```

Figure: 4.23. Complete 1000 unit data at all 4 clouds


Java - multicloud/src/fileStoreAndRetrieval/CloudSimExample4.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Console
Desktop
ENG
US
12:57 PM
21-04-2015

```
<terminated> CloudSimExample4 (3) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (21-Apr-2015 12:56:28 pm)
Starting CloudSimExample4...
Initialising...
* * *
Used disk space on hd1 before upload=0.0
Used disk space on hd2 before upload=0.0
Used disk space on hd3 before upload=0.0
Used disk space on hd4 before upload=0.0
* * *
Used disk space on hd1 after upload=1000.0
Used disk space on hd2 after upload=1000.0
Used disk space on hd3 after upload=1000.0
Used disk space on hd4 after upload=1000.0
time taken to upload the file1 to hd1=65.4296875
time taken to upload the file2 to hd2=65.4296875
time taken to upload the file3 to hd3=65.4296875
time taken to upload the file4 to hd4=65.4296875
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
```

Figure: 4.24. Console output showing transaction time of complete 1000 units.

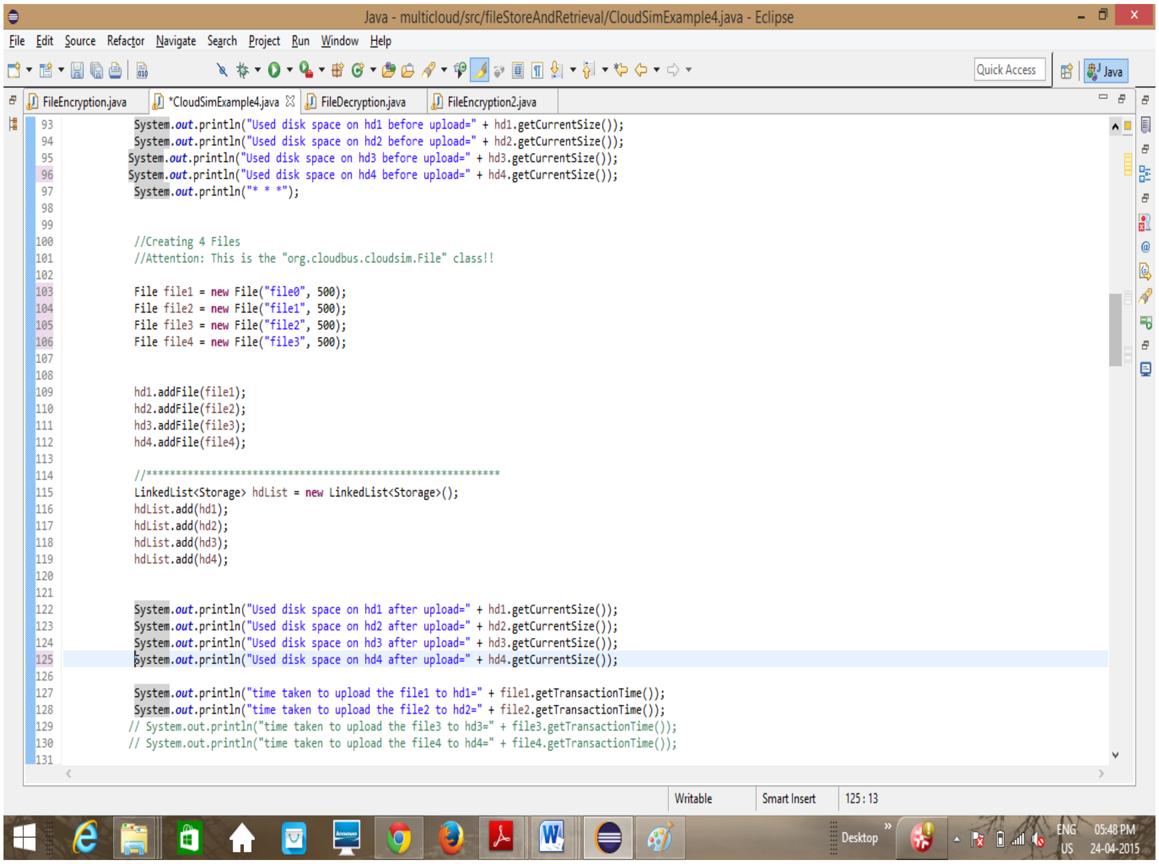
1000 units of 4 files are sent to 4 clouds. The time taken is 65.42.

A	A	A	A
B	B	B	B
C	C	C	C
D	D	D	D

Figure: 4.25. Storage pattern in MCDB

At the time of retrieval the time taken to get the whole data will be equal to data retrieve from at least 3 clouds. Therefore the time taken to get data in parallel from all cloud will be equal to the time taken to get the data from a single cloud to.

Now, let us discuss the case of 4 clouds under proposed approach.



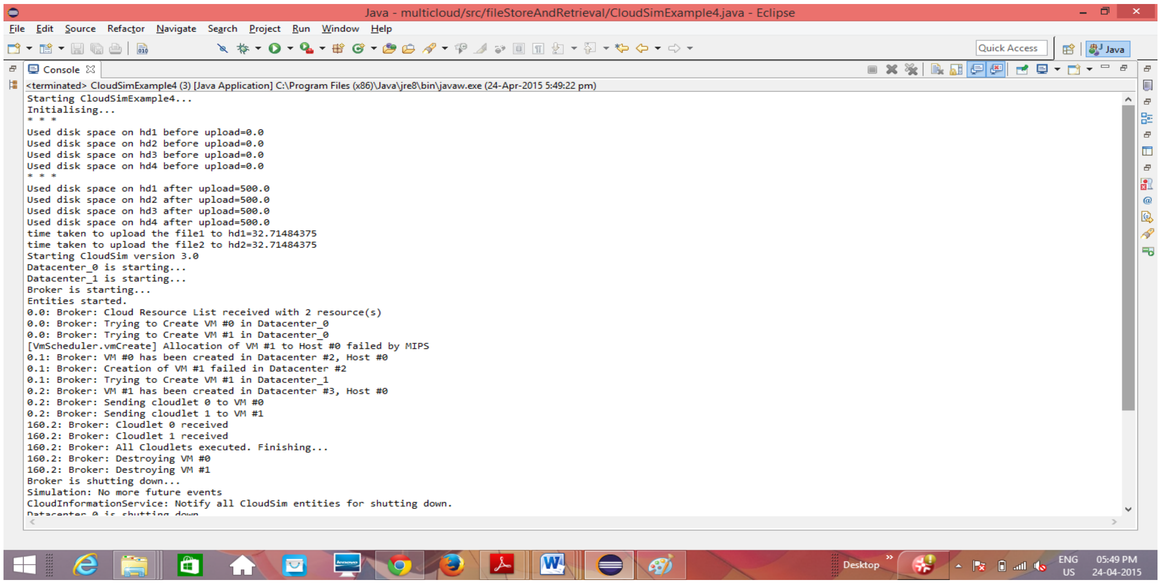
```
           System.out.println("Used disk space on hd1 before upload=" + hd1.getCurrentSize());
           System.out.println("Used disk space on hd2 before upload=" + hd2.getCurrentSize());
          System.out.println("Used disk space on hd3 before upload=" + hd3.getCurrentSize());
          System.out.println("Used disk space on hd4 before upload=" + hd4.getCurrentSize());
           System.out.println("* * *");


           //Creating 4 Files
           //Attention: This is the "org.cloudbus.cloudsim.File" class!!

           File file1 = new File("file0", 500);
           File file2 = new File("file1", 500);
           File file3 = new File("file2", 500);
           File file4 = new File("file3", 500);


           hd1.addFile(file1);
           hd2.addFile(file2);
           hd3.addFile(file3);
           hd4.addFile(file4);

           //************************************************************
           LinkedList<Storage> hdList = new LinkedList<Storage>();
           hdList.add(hd1);
           hdList.add(hd2);
           hdList.add(hd3);
           hdList.add(hd4);


           System.out.println("Used disk space on hd1 after upload=" + hd1.getCurrentSize());
           System.out.println("Used disk space on hd2 after upload=" + hd2.getCurrentSize());
           System.out.println("Used disk space on hd3 after upload=" + hd3.getCurrentSize());
           System.out.println("Used disk space on hd4 after upload=" + hd4.getCurrentSize());

           System.out.println("time taken to upload the file1 to hd1=" + file1.getTransactionTime());
           System.out.println("time taken to upload the file2 to hd2=" + file2.getTransactionTime());
          // System.out.println("time taken to upload the file3 to hd3=" + file3.getTransactionTime());
          // System.out.println("time taken to upload the file4 to hd4=" + file4.getTransactionTime());
```

Figure: 4.26. Storage receiving 500 units of data

A	B	C	D
D	A	B	C

Figure: 4.27. Storage pattern in proposed approach

While uploading files to clouds we are having 4 parts of data and data should be placed in such a way that only two parts are there at each cloud and at the time of retrieval there will be a need of at least 3 clouds to generate a complete set of information.



Java - multicloud/src/fileStoreAndRetrieval/CloudSimExample4.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Console
Desktop
ENG US
05:49 PM
24-04-2015

```
<terminated> CloudSimExample4 (3) [Java Application] C:\Program Files (x86)\Java\jre8\bin\javaw.exe (24-Apr-2015 5:49:22 pm)
Starting CloudSimExample4...
Initialising...
* * *
Used disk space on hd1 before upload=0.0
Used disk space on hd2 before upload=0.0
Used disk space on hd3 before upload=0.0
Used disk space on hd4 before upload=0.0
* * *
Used disk space on hd1 after upload=500.0
Used disk space on hd2 after upload=500.0
Used disk space on hd3 after upload=500.0
Used disk space on hd4 after upload=500.0
time taken to upload the file1 to hd1=32.71484375
time taken to upload the file2 to hd2=32.71484375
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down
```

Figure: 4.26. Console output showing transaction time at single storage

We can notice in above screen shorts that a single storage took just 32.71 units of time. Now similarly there are two cases at the time of fetching the data parts.

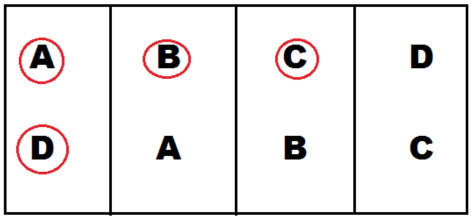


Figure: 4.27 All 4 parts selected from 3 clouds.

First, when all 4 parts are retrieved from any 3 clouds. In this case, the 2 parts will be retrieve from one cloud and other two parts from any other 2 of the remaining 3 clouds. Hence, the time taken to retrieve the data will be equal to the time taken to retrieve those 2 parts from one cloud only (parallel case).

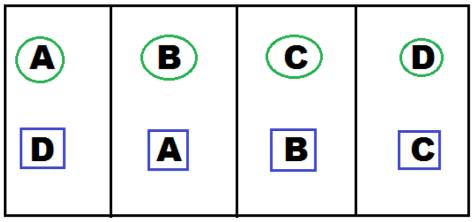


Figure: 4.28. All 4 parts selected from 4 clouds.

Second, when all the 4 parts are being retrieve from all the different clouds. In this case, the time taken to retrieve all 4 parts will be equal to the time taken to retrieve only a single part from a single cloud, if the fetching is being done in parallel way.

4.4. Results and Analysis.

Keeping user's data constant 1000MB, following values are found for transaction time. The transaction is considered in serial and parallel case. Further at the time of retrieval of data there are two cases of retrieving data from n or n-1 clouds.

4.4.1. Serial storage

No. of shares	MCDB	Proposed
2	130.84	65.42
3	196.26	130.71
4	261.68	130.71
5	327.1	130.71

Table: 4.2. Transaction time in serial storage

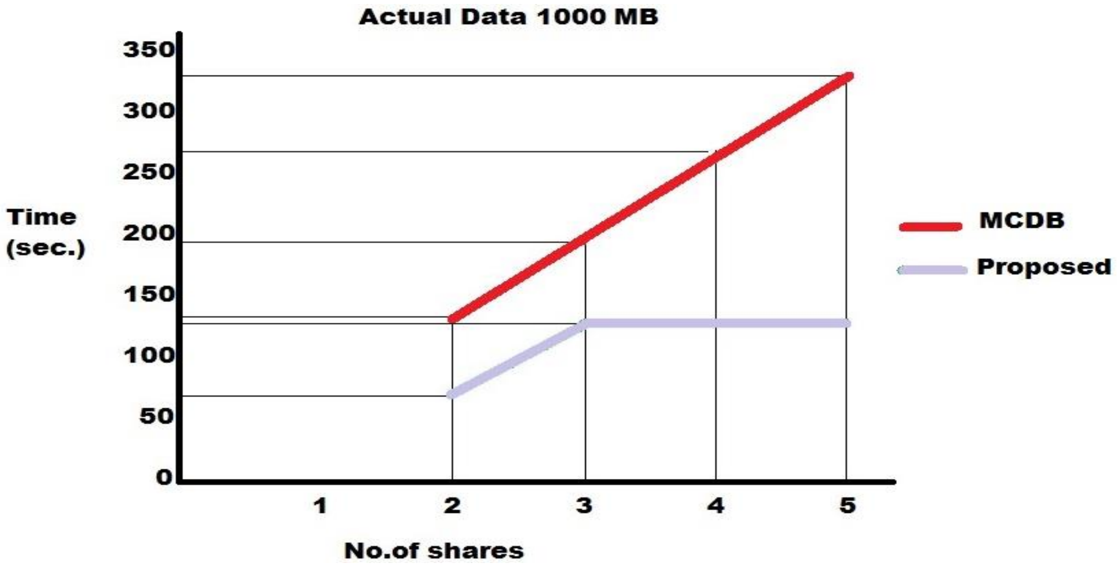


Figure: 4.29. Graphical comparison Serial storage of data

The above graph is for the case when data is being store in storage in a serial manner. In MCDB the time delay increases with increase in number of clouds because user will have to store complete data into each share. In proposed approach the delay remains almost constant after more than 2 clouds, user need to store only a double amount of total data, no matters what is the count of storages.

4.4.2. Parallel Storage

No. of shares	MCDB	Proposed
2	65.42	32.17
3	65.42	43.57
4	65.42	32.71
5	65.42	26.17

Table: 4.3. Transaction time in parallel storge

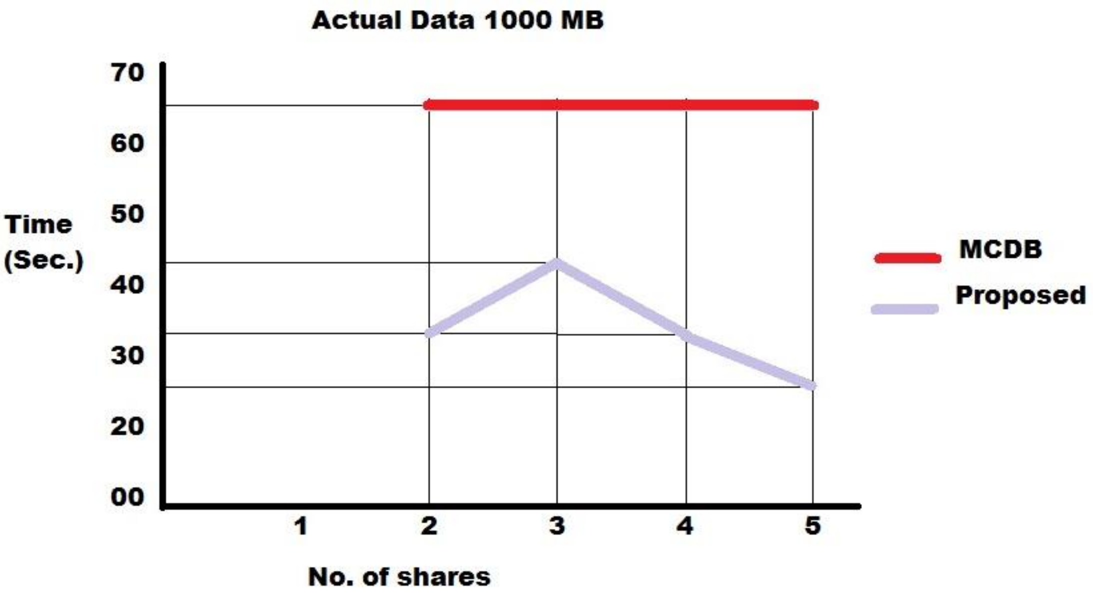


Figure: 4.30. Parallel storage of data

In parallel storage the data is being store simultaneously to all shares. In MCDB the time will remain same as it takes to store in one share. In proposed the storage time rises from 2nd to 3rd share and then reduces.

4.4.3. Serial retrieval.

No. of shares	MCDB	Proposed
2	65.42	65.42
3	130.84	65.42
4	196.26	65.42
5	261.68	65.42

Table: 4.4. Transaction time of serial retrieval

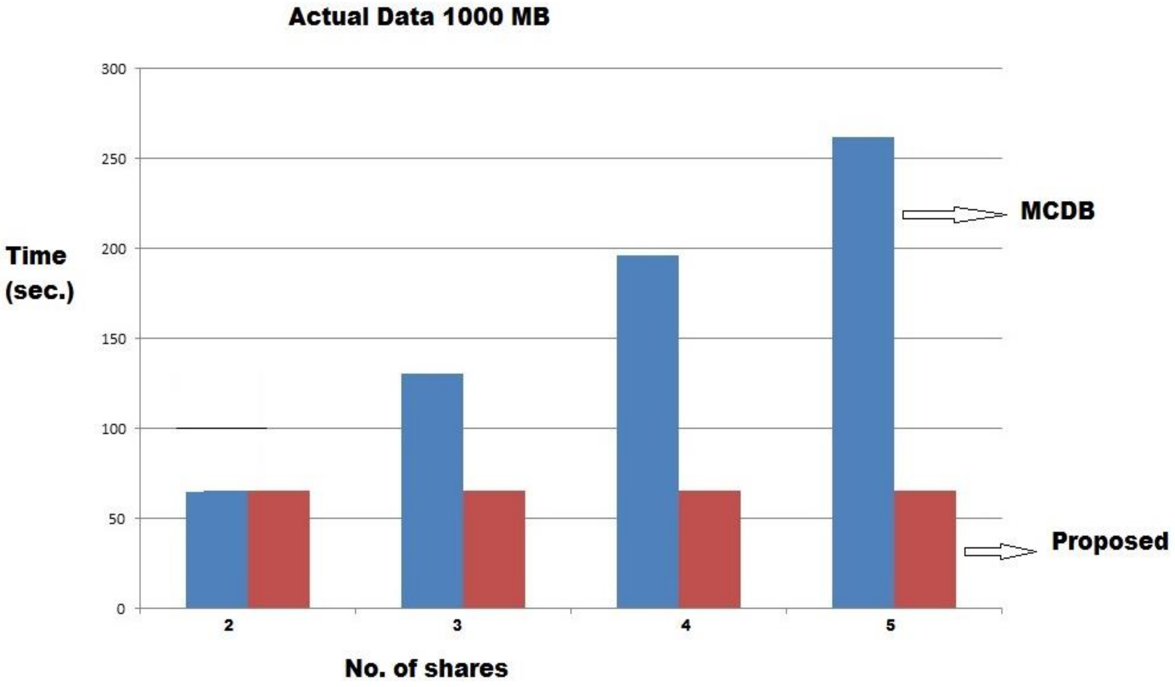


Figure: 4.31. Serial Retrieval of data

At the time of serial retrieval of the data, in MCDB the data should be retrieved from minimum (n-1) shares. Here n is the total number of shares. But in proposed approach the time taken to get the data is just equal to the time taken to get only the parts ones.

4.4.4. Parallel retrieval

In this case, MCDB will have constant retrieval time equal to the time taken to get a complete set of data from one share. But in proposed approach there are further 2 cases, that is when data is retrieve from n clouds and another is when data e retrieve from n-1 clouds.

No. of shares	MCDB	Proposed
2	65.42	32.71
3	65.42	43.57
4	65.42	32.71
5	65.42	26.56

Table: 4.5. Transaction time parallel retrieval, n-1 clouds

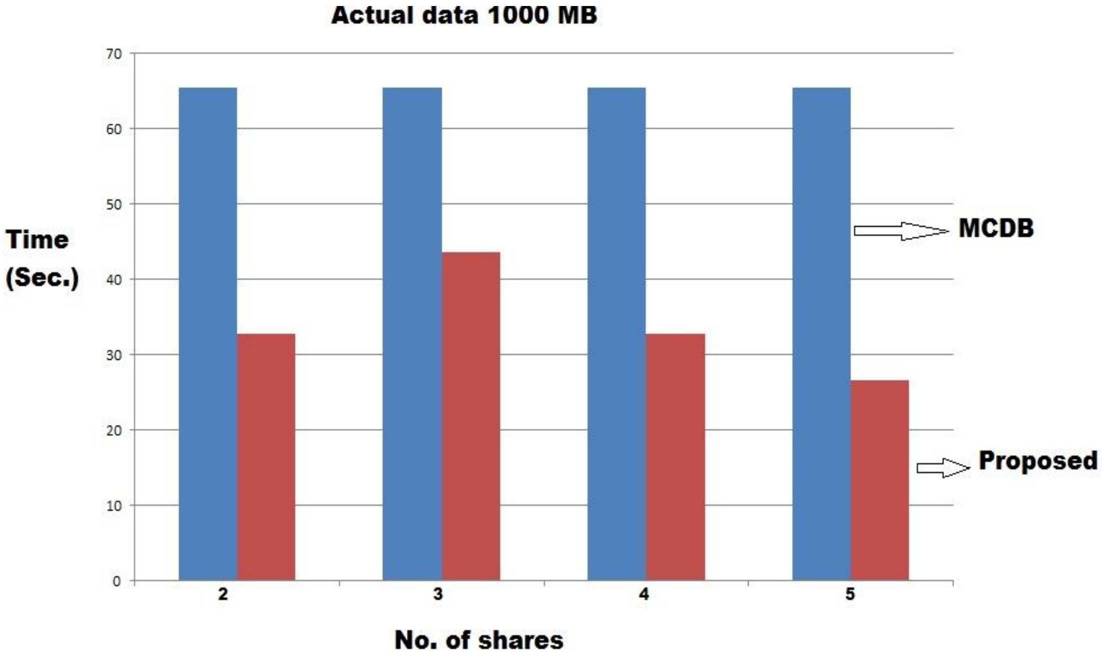


Figure: 4.32. Parallel retrieval of data from (n-1) clouds

In parallel retrieval, even better performance can be achieved if data is received from all the clouds. Minimum of n-1 is the restriction to retrieve the data, to remove the effect of internal malicious user, but to better utilize the resources, in proposed approach all the clouds are used to retrieve the data.

No. of shares	MCDB	Proposed
2	65.42	32.71
3	65.42	21.78
4	65.42	16.35
5	65.42	13.28

Table: 4.6. Transaction time of parallel retrieval, case n clouds

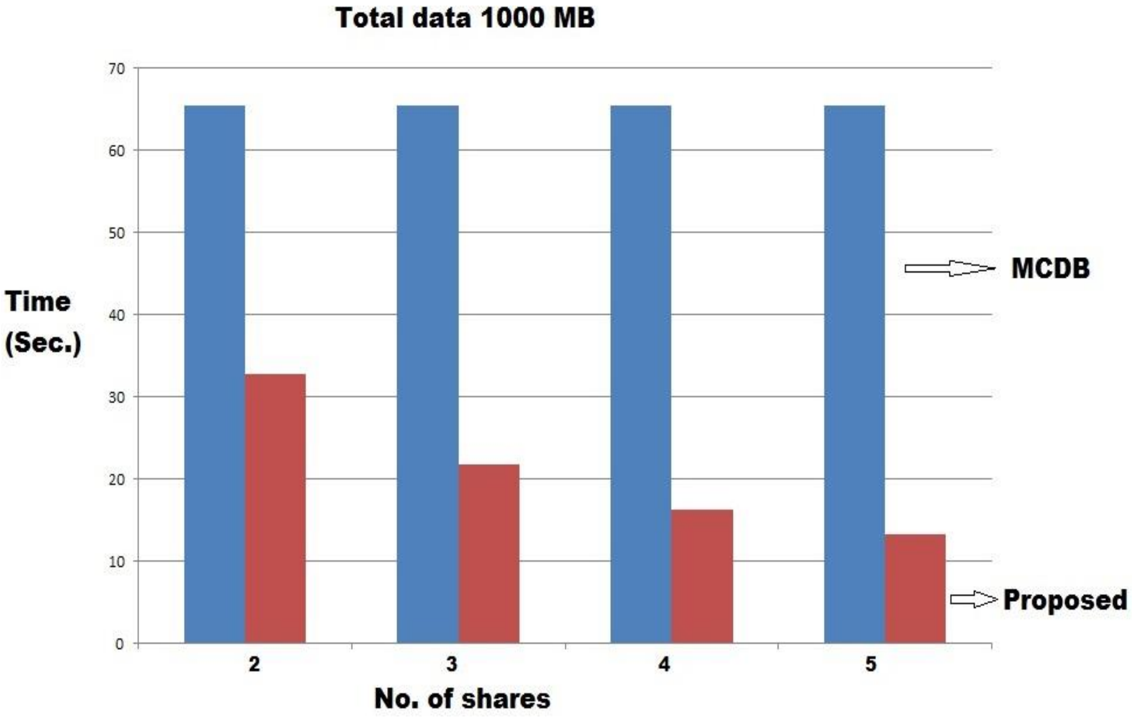


Figure: 4.33. Parallel Retrieval of data from (n) clouds

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

From the above study, we have drawn a conclusion that with multi-cloud system, we can achieve data security. We can maintain Integrity, Availability and can protect data from internal intrusion, i.e. the intrusion done by administrators or employees working inside the cloud environment. In the base paper, the problem of high transaction time of data was discussed only in the serial transfer manner. The problem is solved in proposed approach to very much extend and is discussed in both serial and parallel manner.

5.2 Future Scope

Multi-cloud computing is a fast growing technique in organizations. Those companies which are using private clouds to ensure confidentiality of their sensitive data can use this technique to store data securely on multiple public clouds for better facilities. Proposed system can be used in the present cloud scenario where customer wants their data to be confidentially secure, secure from intrusion, maintenance of integrity and with 24 hour availability.

Companies that need large number of resources but a single cloud is unable to provide all resources can make use of multi-cloud system to have better outcomes in their business.

Organizations which are having customers globally spread can make use of multiple clouds simultaneously on the basis of geographical locations according to density of customers and cloud location.

Future work to be done in this study is to further reduce the transaction time by inventing new techniques of data distribution or by making changed in the current scheme, while maintaining the security.

CHAPTER 6

REFERENCES

6.1. Book Reference

[1]Somasundaram Gnanasundaram, Alok Shrivastava, *Information Storage and Management, second Edition,* John Wiley and Sons, Chapter 13.

6.2. Research Papers

[2] Mohammed A. AlZain, Ben Soh, Eric (2012) “A New Model to Ensure Security in Cloud Computing Services”, PhD Candidate , Department of Computer Science and Computer Engineering, La Trobe University, Bundoora 3086, Australia, Associate Professor, Department of Computer Science and Computer Engineering, La Trobe University, Bundoora 3086, Australia. Ph.D, Department of Computer Science and Computer Engineering,La Trobe University, Bundoora 3086, Australia., respectively, SPRINGER, 2012.

[3] Pradnyesh Bhisikar and Prof. Amit Sahu (2013) “Security in Data Storage and Transmission in Cloud Computing”, G.H.Raisoni College of Engineering and Management, Amravati , IJARCSSE, 2013.

[4] Prof.V.N.Dhawas, Pranali Juikar, Neha Patekar, Neha Lendghar, Sushant Vartak (2013) “A Secured Cost Effective Multi-Cloud Storage in Cloud Computing”, Department Of Computer Engineering ,Sinhgad Institute of Technology,Lonavla, IJSER.

[5] B.AmarNadh Reddy and P.Raja Sekhar Reddy,(2012)“Effective Data Distribution Techniques for Multi-Cloud Storage in Cloud Computing” CSE, Anurag Group of Institutions, Hyderabad, A.P, India , IJERA.

[6] Sultan Ullah and Zheng Xuefeng,(2014) “TCLOUD: A Trusted Storage Architecture for Cloud Computing”, School of Computer and Communication Engineering, University of Science and Technology, Beijing People Republic of China, IJAST.

[7] Kalyani.M.Borse, Ankita.G.Deshpande, Ashlesha.A.Deshpande, Ms.Juily.S.Hardas, (2014) “Security in multi-cloud data storage with SIC architecture”, Student, Computer Engineering Department, GES R.H.S.COE, Maharashtra, India , IJRET.

[8] Sandip Agarwala, Ramani Routray,(2010) “Cluster aware storage provision in a Data Center”, IBM corporation, US, IEEE.

[9] Mr. Ajey Singh and Dr. Maneesh Shrivastava Department of Information Technology, LNCT- Bhopal- India, (2012) “Overview of Security issues in Cloud Computing” Department of Information Technology, LNCT- Bhopal- India, IJACR.

[10] Arun Arunachalam, Shree harsha Chandrashekar, (2012)“Cloud Computing Featuring Ontology”, Computer Engineering Department, VJTI , Matunga, Mumbai, India, IOSRJEN.

[11] Randeep Kaur Chhabra, Prof. Ashok Verma MTECH (CSE), “Strong authentication system along with virtual private network: A secure cloud solution for cloud computing”, ,Head of Department of Computer Science & Engineering Gyan Ganga Institute of Technology & Sciences Jabalpur, Madhya Pradesh, India IJECSE.

[12] Shaik.Aafreen Naaz, Pothireddygari.Ramya, P.Vishunu Vardhan Reddy, “cloud computing: use of multicloud” Department of IT, G.Pullaiah College Of Engineering and Technology.

[13] Vartika, J.Nithin Chowdary, Dr.N.Srinivasu, (April 2013) “Data Storage Security in Multiple Unrelated Cloud Architecture” Department of Computer Science and Engineering,

K.L.University, Vaddeswaram, India. International Journal of Computer Trends and Technology.

[14] B.AmarNadh Reddy,P.Raja Sekhar Reddy, “Effective Data Distribution Techniques for Multi-Cloud Storage in Cloud Computing “,CSE, Anurag Group of Institutions, Hyderabad, A.P, India.

CHAPTER 7

APPENDIX

7.1 Abbreviations

C	Cloud
CS	Cloud Share
CSP	Cloud Service Provider
SAAS	Software As A Service
PAAS	Platform As A Service
IAAS	Infrastructure As A Service
AEC2	Amazon Elastic Compute Cloud
OTP	One Time Password
VPN	Virtual Private Network
SSL	Secure Socket Later
DOS	Denial Of Service
QOS	Quality Of Service
OS	Operating System
CIA	Confidentiality Integrity Availability
GUI/CLI	Graphical User Interface/ Command Line Interface
CPU	Central Processing Unit
RAM	Random Access Memory
DAS	Database service provider