CONSISTENCY AND PRIVACY BASED REPLICATION SYSTEM IN CLOUD SHARING FRAMEWORK

N. Santhi* & R. Selva Kumar**

* PG Scholar, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

** Assistant Professor, Department of Master of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu

Abstract:

Cloud computing is an emerging technology which provides a lot of opportunities for online distribution of resources or services. The most effective benefit of using cloud computing is higher availability of services with lower cost and easy scalability. Cloud provides different type of services and resources effectively but some challenges are still present in it. Out of which security concerns, synchronization, scalability, load balancing and replication are important issues. Data replication means maintaining multiple copies of same data on same server or on different servers. In connection with cloud computing data replication can be said as storing multiple copies of same data on different locations (servers), locally or at remote sites. If data is present at one site only, then it will be very difficult to handle the requests for accessing the data. Server will face a heavy load situation and system performance may degrade. So in this project implements Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS) approach to fragment data sets which are uploaded by data owner and implement graph based approach to calculate the distances using T-Coloring method to predict the data nodes for placing fragmented data. This proposed approach is very useful to data owner for protecting data from attackers. Then we extend our approach for checking consistency in cloud system at the time file updation. And propose a heuristic auditing strategy (HAS) which adds appropriate reads to reveal as many violations as possible. It can be done by using user operation table. Each user maintains a UOT for recording local operations. Each record in the UOT is described by three elements: operation, logical vector, and physical vector. Experimental results provide improved security and reduced retrieval time for accessing data from cloud system and implemented in real time cloud environments.

Key Words: Centrality, Cloud Security, Fragmentation, Replication & Performance

1. Introduction:

1.1 Project Description:

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on a very fundamental principal of „reusability of IT capabilities'. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries. Forrester defines cloud computing as: “A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption.” Cloud Computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. A simple example of cloud computing is Yahoo email, Gmail, or Hotmail etc.

1.2 Cloud Computing Services:

Cloud Providers offer services that can be grouped into three categories.

- ✓ **Software as a Service (SaaS):** In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Sales force, Microsoft, Zoho, etc.
- ✓ **Platform as a Service (Paas):** Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider‟s infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google‟s App Engine, Force.com, etc are some of the popular PaaS examples.
- ✓ **Infrastructure as a Service (Iaas):** IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, Go Grid, 3 Tera, etc.

1.3 Cloud Computing Models:

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can play a vital part in determining the right cloud path for each organization.

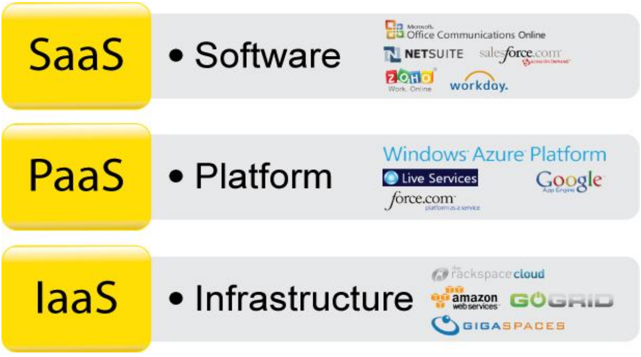

Figure 1: Types of Cloud Services

- ✓ **Public Cloud:** Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, "Pay-as-you-go" model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

- ✓ **Private Cloud:** Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. There are two variations to a private cloud:
- ✓ **On-Premise Private Cloud:** On-premise private clouds, also known as internal clouds are hosted within one's own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability. IT departments would also need to incur the capital and operational costs for the physical resources. This is best suited for applications which require complete control and configurability of the infrastructure and security.
- ✓ **Externally Hosted Private Cloud:** This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don't prefer a public cloud due to sharing of physical resources.
- ✓ **Hybrid Cloud:** Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload

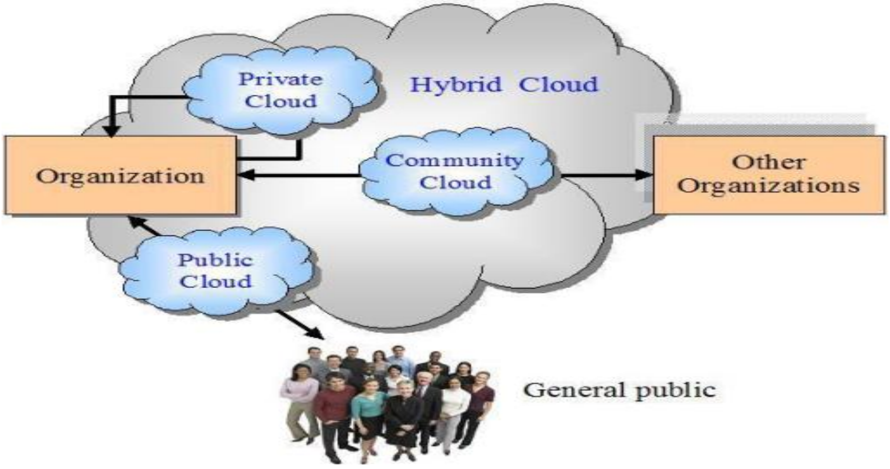

Figure 2: Cloud Computing Models

1.4 Existing System:

Data replication means maintaining multiple copies of same data on same server or on different servers. In connection with cloud computing data replication can be said as storing multiple copies of same data on different locations (servers), locally or at remote sites. If data is present at one site only, then it will be very difficult to handle the requests for accessing the data. Server will face a heavy load situation and system performance may degrade. But excessive replication can also adverse effects like high storage cost or degradation in systems overall performance due to excessive use of bandwidth. So DROPS framework is better to use because it can understand fragments of the data. T-coloring algorithm can provide better results in case when system is in ideal state In T-coloring, measure the distances of each data for placing data in cloud system. Distances are calculated using centrality measure. Centrality is measure of the relative importance of a node in the network. But in DROPS framework, data can be lost due to updation at the time of retrieving from cloud storage.

Disadvantages of the Existing System:

- ✓ There is no replication available at the time of data sharing
- ✓ Data can be corrupted due to update the document
- ✓ Computationally high cost and expensive
- ✓ Provide large time for retrieving the documents

1.5 Proposed System:

Cloud computing service providers require a system which can handle a large number of requests at a time. For processing the huge cloud of requests for data access, services need to be highly available. System keeps multiple copies of the blocks of data on different nodes by replication. A large number of replication strategies for management of replicas have been implemented in traditional system. As a result of replication, data replicas are stored on different data nodes for high reliability and availability. Replication factor for each data block and replica placement sites need to be decided at first. In existing framework data can be lost so in this paper propose improved DROPS framework that includes heuristic auditing strategy to protect the data from loss. It present efficient consistency as a service model, where a group of data owners that constitute service provider can verify whether the data cloud update the data or not and design user operation table to change status of fragmented files with different metrics

Advantages:

- ✓ Difficult to hack the files from cloud storage because fragmentation and replication placement approach.
- ✓ Improved scalability to store large data files.
- ✓ Proposed reduce response time for retrieving data from cloud.

2. Literature Survey:

Kashif Bilal, "Quantitative comparisons of the state-of-the-art data center Architectures": This paper provides a comparative study and analysis of major DCN architectures that are proposed in the recent years by implementing: (a) proposed network architectures, (b) customized addressing scheme, (c) customized routing schemes, and (d) different network traffic patterns. We have implemented the fat-tree based architecture, recursively defined architecture, and legacy three-tier DCN architecture to compare the performance under six different network traffic patterns. For the fat-tree DCN architecture, we implemented the n-pod based network interconnection design, customized network addressing scheme for servers and switches at different levels, and customized two-level routing algorithm. For the recursive-based DCell DCN architecture, we applied customizable n-level network architecture (up to four levels scalable for more than 3.6 million servers), a generic network addressing scheme, and the DCell routing algorithm. DCell routing algorithm returns a series of nodes as intermediate hops between source and destination. We formulated an algorithm to find the network address-based end-to-end path and implemented source-based routing in the ns-3 simulator. Moreover, the DCell routing algorithm pseudo code had some missing information for implementation and working. We formulated the missing information to complete the algorithm. For the legacy three-tier DCN architecture, we implemented customizable network architecture as reported. We used the Equal Cost Multi-Path (ECMP) routing to obtain realistic results for the three-tier DCN architecture. Presumably, it is the very first comparative study of DCN architectures employing implementation and simulation techniques.

Kashif Bilal, “On the Characterization of the Structural Robustness of Data Center Networks”: In this paper, we evaluate various topological features and robustness of the state-of-the-art DCNs namely: 1) Three Tier, 2) Fat Tree, and 3) DCell. The major contributions include: modeling DCN topologies using multilayered graphs; developing a DCN graph topology generation tool; measuring several robustness metrics under various failure scenarios; comparative robustness analysis of the DCN topologies and indicating the inadequacy of the classical robustness metrics to evaluate DCNs; proposing new robustness metric for the DCN topologies. The robustness analysis of the DCN topologies unveiled noteworthy observations. The results revealed that the classical robustness metrics, such as average nodal degree, algebraic connectivity, and spectral radius are unable to evaluate DCNs appropriately. Most of the metrics only consider the largest connected component for robustness evaluation. Consequently, the metrics are unable to depict the factual measurements and robustness of the network. Moreover, none of the DCNs can be declared as more robust based on the measurements taken: 1) without failure and 2) under various failure scenarios. Therefore, we present a new metric named deterioration to quantify the DCN robustness. As the number of nodes (vertices) in each layer needs to be same, the proposal is inapplicable to DCNs. Moreover, the definition lacks the interconnection information between different layers in the proposal. Because none of the previously proposed graph models matches the DCN-based graph definition, present a formal definition for each of the DCN architectures. As the number of network elements in the Fat Tree is much higher than the Three Tier architecture, the number of failed nodes is around five times in the Fat Tree as compared to the Three Tier architecture.

Dejene Boru, “Energy-efficient data replication in cloud computing datacenters”: In this paper propose a data replication technique for cloud computing data centers which optimizes energy consumption, network bandwidth and communication delay both between geographically distributed data centers as well as inside each datacenter. Specifically, contributions can be summarized as follows. Modeling of energy consumption characteristics of data center IT infrastructures. Development of a data replication approach for joint optimization of energy consumption and bandwidth capacity of data centers. Optimization of communication delay to provide quality of user experience for cloud applications. Performance evaluation of the developed replication strategy through mathematical modeling and using a packet level cloud computing simulator, Green Cloud. Analysis of the tradeoff between performance, serviceability, reliability and energy consumption. Maintaining replicas at multiple sites clearly scales up the performance by reducing remote access delay and mitigating single point of failure. However, several infrastructures, such as storage devices and networking devices, are required to maintain data replicas. On top of that, new replicas need to be synchronized and any changes made at one of the sites need to be reflected at other locations. This involves an underlying communication costs both in terms of the energy and network bandwidth.

Wayne A. Jansen, “Looks: Security and Privacy Issues in Cloud Computing”: Cloud computing can be implemented entirely within an organizational computing environment as a private cloud. However, it should be clear from the service models described that a main thrust of cloud computing is to provide a means to outsource parts of that environment to an outside party. As with any outsourcing of information technology services, concerns exist about the implications for computer security and privacy, particularly with moving vital applications or data from the organization’s computing center to the computing center of another organization. While reducing cost is a primary motivation for moving towards a cloud provider, reducing responsibility for security or privacy should not be. Ultimately, the organization is accountable for the overall state of the outsourced service. Monitoring and addressing security and privacy issues remain in the purview of the organization, just as other important issues, such as performance, availability, and recovery. This paper looks at the main security and privacy issues pertinent to cloud computing, as they relate to outsourcing portions of the organizational computing environment. It points out areas of concern with public clouds that require special attention and provides the necessary background to make informed security decisions. The issues are organized into several general categories: trust, architecture, identity management, software isolation, data protection, and availability. Because cloud computing has grown out of an amalgamation of technologies, including service oriented architecture, virtualization, Web 2.0, and utility computing, many of the security issues involved can be viewed as known problems cast in a new setting. Nevertheless, it represents a thought-provoking paradigm shift that goes beyond conventional norms in de-perimeterizing the organizational infrastructure.

Giorgos Kappes, "Dike: Virtualization-aware Access Control for Multitenant File systems": In this design we require that each client directly mounts the file system instead of having the file system mounted by an intermediate proxy. The file system natively manages the access control metadata of each tenant, and ensures that each tenant can only access its own namespace. Controlled file sharing is relatively straightforward as a result of the file based access to a common file system with file-granularity access control. We provide prototype implementation of the above approach in the Ceph production-grade, distributed File system. With micro benchmarks and application-level experiments we quantitatively demonstrate the limited performance overhead of our design. A software-as-a-service provider supports business customers with disjoint end users. The file system treats each business customer as a tenant with separate application files in writable mode (e.g., databases), but possibly shared system files in read-only mode (e.g., configuration scripts). A public cloud provides a shared software repository that different groups of developers can fork into separate branches. The members of a group obtain writable access to their own branch and read-only access to the branches of other groups. A simpler scheme without branches could also be used for sharing scientific datasets. The file system protects the confidentiality and integrity of stored data and metadata by restricting accesses to authorized principals. We assume that the provider has no malicious intent to compromise the system security. However, there may be other reasons (e.g., poor practices) for which the provider is not trusted for some applications. In that case the tenant may externally apply techniques of encryption, hashing and auditing to achieve end-to-end confidentiality, integrity and freshness

3. Implementation:

- ✓ Cloud Framework
- ✓ Data Chunking
- ✓ T-Coloring Method
- ✓ Data Access
- ✓ Security Performance

Cloud Framework: Cloud service is any service offered to its users by cloud. As cloud computing comes in service there are some drawbacks such as privacy of user's data, security of user data is very important aspects. Cloud computing is demand on shared computing resources. With the continuous development of cloud computing technology, its appliance is more and more widely. However, for data owners who are becoming increasingly concerned about their privacy of the data which contains some personal information about individuals. In this module, cloud data storage service three different entities such as the cloud owner, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources; cloud user, who access data from cloud storage.

Data Chunking: In this module all data records are split into data chunks. A chunk is a fragment of information which is used in many formats, such as .txt, .doc and so on. Each chunk contains a header which indicates some parameters e.g. the type of chunk, comments, size etc. In the middle there is a variable area containing data which are decoded by the program from the parameters in the header. Chunks may also be fragments of information which are downloaded or managed by distributed programs. In distributed computing, a chunk is a set of data which are sent to a processor or one of the parts of a computer for processing.

T-Coloring Method: In this module, fragments are placed in various providers using T-Coloring method. This is used for channel assignment assigns channels to the nodes, such that the channels are separated by a distance to avoid interference. T-coloring prohibits storing the fragment in neighborhood of a node storing a fragment, resulting in the elimination of a number of nodes to be used for storage. In such a case, only for the remaining fragments, the nodes that are not holding any fragment are selected for storage randomly.

Data Access: In this module, user accesses the data from cloud through providers. The users those who are having matching secret key defined in the cipher text can retrieve the entire data content. It aims to allow the users with eligible key to decrypt the entire data stored in the cloud server. However it cannot limit the users from accessing the data's which are not accessible to them. That is it cannot limit the data access control to the authorized users.

Security Performance: In this module we can evaluate the performance of the system using the performance metrics such as storage overhead, communication cost and computation efficiency. The storage overhead is one of the most significant issues of the access control scheme in cloud storage systems. In our scheme, besides the storage of attributes, each sub server also needs to store a public key and a secret key for each user in the system. Thus, the storage overhead on each server in our scheme is also linear to the number of in the system. The communication cost of the normal access control is almost the same. The communication cost of attribute revocation is linear to the number of cipher texts which contain the fragments. We compare the computation efficiency of both encryption and decryption in two criteria: the number of sub server and the number of fragments per server.

4. Algorithm:

Design / Algorithm / Pseudo code (whichever is applicable)

Heuristic Auditing Strategy:

From the auditing process in the CaaS model, we observe that only reads can reveal violations by their values. Therefore, the basic idea of our heuristic auditing strategy (HAS) is to add appropriate reads for revealing as many violations as possible. We call these additional reads auditing reads. HAS divides physical time into L time slices, where l time slices constitute an interval. Each time slice is associated with a state, which can be marked with either normal or abnormal. A normal state means that there is no consistency violation, and an abnormal state means that there is one violation in this time slice. HAS determines the number of auditing reads in the (i+1)-th interval, based on the number of abnormal states in the i-th interval. Let ni denote the number of auditing reads in interval i. HAS determines ni+1, which is the number of auditing reads in the next interval with following equation.

$$\begin{cases} n_{i+1} = \min(l, kXn_i), n_{i\geq\alpha} \\ n_{i+1} = \max\left(1, \frac{1}{k}Xn_i\right), n_i < \alpha \end{cases}$$

where k is a parameter that is used to adjust the value of n_{i+1}, l is the number of time slices in an interval, and α is a threshold value that is used to determine whether the number of auditing reads in the next round should be increased by k times or be reduced to 1/k, compared to the number of auditing reads in the current round. Specifically, given a threshold value α, if a user issues n_i auditing reads and reveals more than α violations in interval i, in interval i + 1, the user will issue ni+1 = min(l, k ∗ n_i) auditing reads; that is, each time slice will be issued, at most, one auditing read, and the maximal number of auditing reads will not exceed l. Otherwise, the user will issue ni+1 = max(1, $\frac{1}{k}Xn_i$)auditing reads, that is, each interval will be issued at least one auditing read. Since the number of auditing reads should be an integer, $\frac{1}{k}Xn_i$ is actually the abbreviation of $\frac{1}{k}Xn_i$

Mathematical Model:

- ✓ Initial User Operation Table (UOT) with Ø
- ✓ While issue an operation op do
- ✓ If op = W (a) then
- ✓ Record W (a) in UOT
- ✓ If op = r (a) then
- ✓ W (b) ϵ UOT is the last write
- ✓ If W (a) → W (b) then
- ✓ Read-your-write consistency is violated
- ✓ R(c) ϵ UOT is the last read
- ✓ If W (a) → W(c) then
- ✓ Monotonic-read consistency is violated
- ✓ Record r (a) in UOT Where,

W (a) - Write operation R (a) - Read operation

Global Auditing Algorithm:

- ✓ For every operation in the global trace is represent by a vertex
- ✓ For operation op1 and op2 do
- ✓ If op1->op2 then time edge is added between op1 and op2
- ✓ If op1=w (a),op2=r (a) op1 and op2 comes from different user then data edge is inserted between op1 and op2
- ✓ If op1=w(a) and op=(b) and op1 and op2 comes from different users and w(a)->w(b)->r(b) then causal edge is inserted between op1 and op2
- ✓ Verify whether the graph is directed acyclic graph by topological sorting method

It is very is to prove effectiveness of local auditing. For monotonic-read consistency user is required to read a latest value or same value. If the dictating write of a latest read before the dictating write of the last read, we realized that monotonic- read consistency get unsuccessful result. Read your write consistency, user required to read his latest writes. Hence if the dictating writes of a new read before his last write, we realize that read-your-write consistency get unsuccessful result. For causal consistency we can prove that 1. if the Constructed graph is not directed acyclic graph then there must be unsuccessful result 2. if constructed graph is directed acyclic graph then there must be successful result from proposition 1 we can conclude that if graph contain a cycle then there exists an operation that is committed before itself, which is not possible to prove so we use method of contradiction to prove that let us consider if there is not get successful result when given graph is directed acyclic graph. Unsuccessful result which indicate that two write contain W (a) and W (b) that have causal relationship according our contradiction we have two read R (b) implies R (a) which means time edge is form between W (a) to W (b) and data edge form between W (a) and R (a) then w (a) to w (b) contain cycle hence contradicts our assumption.

5. Result & Discussion:

It is very is to prove effectiveness of local auditing. For monotonic-read consistency user is required to read a latest value or same value. If the dictating write of a latest read before the dictating write of the last read, we realized that monotonic- read consistency get unsuccessful result. Read your write consistency, user required to read his latest writes. Hence if the dictating writes of a new read before his last write, we realize that read-your-write consistency get unsuccessful result. For causal consistency we can prove that 1. if the Constructed graph is not directed acyclic graph then there must be unsuccessful result 2. if constructed graph is directed acyclic graph then there must be successful result from proposition 1 we can conclude that if graph contain a cycle then there exists an operation that is committed before itself, which is not possible to prove so we use method of contradiction to prove that let us consider if there is not get successful result when given graph is directed acyclic graph. Unsuccessful result which indicate that two write contain W (a) and W (b) that have causal relationship according our contradiction we have two read R (b) implies R (a) which means time edge is form between W (a) to W (b) and data edge form between W (a) and R (a) then w (a) to w (b) contain cycle hence contradicts our assumption.

6. Conclusion:

The proposed system provide main contribution is support for consistency-based SLAs that allow developers to declaratively specify their needs using a choice of consistency guarantees coupled with latency targets. Get operations access data that is partitioned and replicated among servers in all parts of the world while conforming to such SLAs. Consistency-based SLAs allow applications that were written to tolerate eventual consistency, as are many cloud applications today, to benefit from increased consistency when the performance cost is not excessive. When conditions are favorable, such as when the application is running in the same datacenter as up-to-date replicas, and it is able to deliver ideal consistency and latency to the application, and when conditions are less favorable, such as when nodes fail or become overloaded or clients are far from their frequently accessed data, the application's SLA indicates how best to adapt. In this project, we present enabling data integrity proof and consistency services over multi cloud system using Heuristic auditing strategy which helps in revealing violations as much as possible. The cloud consistency model and local auditing, global auditing that helps users to verify the cloud service provider (CSP) provides the promised consistency or not and quantify the severity of the violations. Therefore system monitors consistency service model as well as level of data upload which helps the user to get the data in updated version. User can understand various sub servers in cloud service provider. It is a strategic to provided automatic update mechanism to identify fragments easily and provide the data to users after updating only.

7. Future Work:

In future work, we can extend the project to implement various replica management systems and also implement in mobile cloud environment. And also reduce the bandwidth and throughput at the time of data sharing.

8. References:

1. K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, 2013, pp. 1771-1783.
2. K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, Vol. 1, No. 1, 2013, pp. 64-77.
3. D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," In IEEE Globe com Workshops, 2013, pp. 446-451.
4. W. A. Jansen, "Cloud hooks: Security and privacy issues in cloud computing," In 44th Hawaii IEEE International Conference on System Sciences (HICSS), 2011, pp. 1-10.
5. G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, "Dike: Virtualization-aware Access Control for Multitenant Filesystems," University of Ioannina, Greece, Technical Report No. DCS2013-1, 2013
6. W. K. Hale, "Frequency assignment: Theory and applications," Proceedings of the IEEE, Vol. 68, No. 12, 1980, pp. 1497-1514.
7. K. Hashizume, D. G. Rosado, E. Fernndez-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," Journal of Internet Services and Applications, Vol. 4, No. 1, 2013, pp. 1-13.
8. M. Hogan, F. Liu, A. Sokol, and J. Tong, "NIST cloud computing standards roadmap," NIST Special Publication, July 2011.
9. W. A. Jansen, "Cloud hooks: Security and privacy issues in cloud computing," In 44th Hawaii IEEE International Conference on System Sciences (HICSS), 2011, pp. 1-10.
10. A. Juels and A. Opera, "New approaches to security and availability for cloud data," Communications of the ACM, Vol. 56, No. 2, 2013, pp. 64-73.