JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# SECURE BIG DATA STORING AND SHARING IN CLOUD COMPUTING USING DATA CHUNKS

[1] **DODSI SRIKAR**, [2] **Ch Mounika**
[1] ASSOCIATE PROFESSOR, [2] STUDENT
[1,2] COMPUTER SCIENCE ENGINEERING,
[1,2] BHIMAVARAM INSTITUTE OF ENGINEERING AND TECHNOLOGY, BHIMAVARAM, INDIA

***Abstract***: In order to comply with the legal specifications of confidential personal data, we introduce CHARON, a close to POSIX cloud-sponsored garage gadget able to storing and sharing massive information with minimum control and no dedicated infrastructure. CHARON is cloud-based storage system capable of storing and exchanging big data in a secure, effective and scalable manner utilising various cloud services and storage repositories. The distinct features are introduced by CHARON: First feature is, it does not rely on a single entity for trust. Second feature is, it does not use a client-managed server and third feature is, it efficiently manages huge files across a network of geo distributed storage providers. In addition, to prevent write-write clashes between customers using shared libraries, we established a new Byzantine-resilient data-centric leasing protocol. Furthermore, CHARON is able to manage massive information in a stable manner through dividing documents into chunks, using encryption, erasure codes, and compression, and the use of pre fetching and historical past uploads. We use micro and application-based benchmarks to test CHARON, simulating representative workflows from a popular big data domain, bioinformatics. A Byzantine-resilient information-centric lease algorithm that exploits exceptional cloud offerings without requiring accept as true with on any of them individually. The findings indicate that our unique architecture is not only practical, but also provides up to 2:5 greater end-to end efficiency than other cloud-backed solutions.

***Index Terms*: Big data storage, security, privacy, Cloud storage, Byzantine fault tolerance.**

## I.INTRODUCTION

### 1.1 What is cloud computing?

Cloud storage is the use of information resources (such as hardware and software) distributed over a network as a service (in general, the Internet). The word comes from the normal use of system diagrams of a cloud-shaped symbol as an abstraction of the complicated architecture that it contains. Cloud storage confide remote services with a user's data, s/w and computation. Cloud storage includes tools available on the Internet, such as third-party vendors managed for hardware and apps. These features typically provide connections to advanced operating systems and high-end computer networks for servers.

1.2 Structure of cloud computing

**How Cloud Computing Works?**

Cloud computing is intended to use traditional numerical modeling or high-level computing tools, traditionally used by military labs and scientific labs, to store big data, to large power, immersive digital games in consumer-oriented applications for ex, financial portfolios, or To execute tens of thousands of trillions of calculations per second, to provide personalized information. Cloud storage uses networks of large computing clusters with specialized connections, usually running low-cost consumer PC hardware, to extend data processing functions across them. There are large pools of interconnected networks in this mutual IT infrastructure. In addition, to leverage the ability of cloud computing virtualization techniques are used.

fig1: cloud computing metaphor

**1.3 The Cloud Computing characteristics are as follows:**

**1.Rapid Elasticity**: This feature allows users to transparently increase or decrease resources according to their computing needs..

**2.Broad Network Access**: Services on the Internet can be accessed anytime, anywhere in the world via multiple devices such as mobile phones, tablets, laptops, etc.

**3.Measured service**: On a pay-for-what-you-use model Cloud Computing is based, where monitored the resource usage, measured and transparently reported, based on utilization.

**4.Resource Pooling**: The resources are pooled by multiple users like virtual server space, bandwidth, network connections etc., at the same time from any location, the other users without being interfered.

**5.On-Demand Self-Service**: Resources can shared by the multiple clients in cloud computing and applications simultaneously. These cloud resources and services can be used on-demand and purchased frequently on a subscription basis without human interaction with service providers.

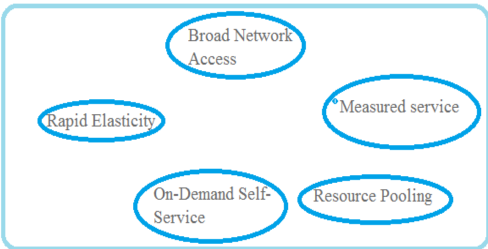

fig2: cloud computing characteristics

**1.4 Versions of Features and Services:**

Based on the definitions provided by the National Institute of Standards and Terminology (NIST), the cloud computing salient features are described below:

1.**Rapid elasticity**: Easily and elastically can provides the capacities and released quickly, in some circumstances automatically to scale in quickly. The capabilities required for provisioning are often infinitely visible to the customer and can be ordered anytime.

2. **On-demand self-service:** A user can provide single computer capacities instantly, such as network storage and processing time, if necessary, without the requirement for human interaction with each service provider.

3. **Wide network access** : Across the network capabilities are available and enabled by standard protocols that enable heterogeneous thick or thin user platforms to be used (e.g., mobiles, notebook, laptops, and PDAs).

4. **Measured service**: Cloud providers dynamically track and manage resource utilization by utilizing service category metering capabilities at each level of capture (e.g., processing, storage bandwidth, and active client accounts). To ensure that both the seller and the consumer of the service used are held accountable, the use of the services should be tracked, monitored and recorded.

5. **Capital pooling**: The supplier's computing facilities are pooled to support multiple clients through a multi-tenant model with distinct physical and virtual infrastructure efficiently distributed and reassigned according to customer demand. There is a sense of independence from the location in that the user usually has no control or knowledge of the actual location of the facilities delivered, yet may evaluate the location at a greater level of abstraction (e.g., country, state, or data center). All types of resources are storage, computing, memory, network bandwidth, and virtual machines.

### 1.5 Benefits of Cloud Computing:

1. **Efficiency and Cost Reduction** :You don't have to put money into in hardware, equipment, utilities, or building out a huge data center to grow your business. To manage your cloud data center operations, you do not even need large IT teams, as you can use the expertise of your cloud provider's staff.
2. **Scalability**: This scalability reduces the risk of internal operational problems and maintenance. With Professional Solutions and Zero Up-Front Investment, You Have High-Performance Resources. Scalability is perhaps the greatest advantage of the cloud.
3. **Back-up and Restore Data**: Storing data in the cloud without any capacity limitations also helps for backup and restore purposes. As end-user data changes over time and need to be tracked according to terms or conditions, older software versions will be stored in later stages, In some cases they may be needed for recovery or rollback.
4. **Data Loss Prevention:** Losing data is a major concern for all companies, along with data security. Storing your data in the cloud guarantees that data is always available, even if your equipment like phones or PCs, is damaged. Generally, Cloud services provide fast data recovery for all types of emergency scenarios.

Cloud infrastructure can also help you with damage mitigation. If you rely on traditional on-premises policy, all of your data is stored locally, on office computers. Despite your best efforts, computers can malfunction for various reasons from malware and viruses to age-related hardware degradation, to common user error.

But, if you upload your data to the cloud, whatever happens to your work computer can access any computer with an internet connection.

### 1.6 Advantages:

**1. Cost:** Purchasing hardware for storage can be expensive. Compared to using external drives, without the need for physical cloud storage is exceptionally cheaper per GB.

**2. Accessibility:** Cloud based storage gives you access to client files from anywhere that has an internet available.

**3. Recovery:** In the event of a hard drive failure or other hardware malfunction, you can access your files on the cloud. It acts as a backup solution for your local storage on physical drives.

**4. Syncing and Updating:** When you work with cloud storage, the file is synced and updated on all your devices that access the cloud.

**5. Security:** Cloud storage providers add extra security layers to their services. Because there are so many people whose files are stored in the cloud, these providers go to great lengths to ensure that your files are not accessed by an unwanted person.

## II. LITERATURE SURVEY

In the software development process the most crucial step is conducting a literature review. It is vital to determine the time factor, the economy, and the company's strength before building the tool. Once these requirements have been met, the next 10 steps are to choose which operating system and programming language will be utilised to construct the tool.

**BYZANTINE DISK PAXOS:** For BYZANTINE Mutual MEMORY Optimum Durability We present Byzantine Disk Paxos, an algorithm for asynchronous shared-memory consensus Used a set of n > 3t discs, t of which will fail by being unresponsive or corrupted arbitrarily. We give two constructions of this algorithm; that is, we create two separate building blocks of t- tolerant (i.e. tolerating up to t disc failures), both of which can be used to address agreement together with a leading oracle. A T-tolerant wait-free mutual stable registry is one building block. A t-tolerant standard register that meets a weaker termination (liveness) requirement than waiting independence is the second building block: its writing operations are wait-free, although its read operations are expected to return only in executions with a limited number of writes. We name this termination state finite writes (FW), and illustrate that with FW terminating registers and a leading oracle, wait-free agreement is solvable. From n > 3t base registers, we build each of these tolerant registers, t of which may be nonresponsive or Byzantine. In this model, all previous t-tolerant wait-free constructions used at least 4t + 1 fault-prone registers, and in this model, we are not acquainted with any previous FW terminating constructions. UNIDRIVE: Various CONSUMER CLOUD Computing Resources SYNERGIZE In order to store and synchronise files via apps installed on their smart phones, consumer cloud storage (CCS) services have become common among users. However, a single CCS has inherent limitations on networking efficiency, stability of the operation, and data protection. We are introducing Uni Drive, a CCS software that synergizes several CCSs (multi cloud) by utilising just a few basic public RESTful Site APIs to solve these limitations. A server-less, client- centered architecture follows Uni Drive, in which synchronisation logic is purely applied on client devices and all correspondence is transmitted through file upload and download operations. A quorum-based distributed mutual-exclusive lock function ensures good continuity of the metadata. By judiciously spreading erasure-coded files through many CCSs, Uni Drive enhances durability and security.

Uni Drive leverages all usable clouds to optimize concurrent transmission possibilities to improve networking efficiency, but the core insight behind is the notion of data block over-provisioning and dynamic scheduling. The diversified and changing network conditions of the underlying clouds are obscured by this suite of strategies, and the faster clouds are more exploited by a basic but successful in-channel probing scheme. Extensive preliminary findings from 272 users on the global Amazon EC2 network and a real-world evaluation verified Uni Drive's substantially superior and reliable sync efficiency over every single CCSS.

## III. SYSTEM ANALYSIS & DESIGN

### Existing System

Paxos Byzantine disk[26] is a consensus protocol developed on top of mutual discs that are not trusted. More recently, an optimised variant of this protocol has been released expressly for the usage of file synchronisation services (e.g., DropBox, Google Drive) instead of disks[21]. These algorithms may be used to apply reciprocal exclusion that satisfies deadlock-freedom (a stronger liveness guarantee than obstruction-freedom). However, a significantly greater amount of cloud access will be needed for these solutions. On the other side, our lease protocol needs just two or four cloud access to obtain a lease. There are only two fault-tolerant data-centric lease algorithms in the literature[15],[39], to the best of our comprehension. Compared to CHARON's BFT composite lease, the lease algorithm of Chockler and Malkhi [39] has two major variations. Second, it does not have an always-safe contract, since it recognises the presence of more than one legitimate leasing procedure. Second, it only tolerates failures, thereby requiring some trust in individual cloud providers. DepSky's BFT reciprocal exclusion algorithm[15] is a natural candidate in CHARON for controlling access contention. Our composite lease algorithm, however, is 4/10 faster than DepSky's (see x5.2), does not need synchronised clocks for users, and does not focus on weakly consistent operations such as the list of object storage. Systems such as Hybris[23], SCFS[24] and RockFS[70] use a hybrid approach that utilises unmodified cloud storage resources with few processing nodes to store metadata and coordinate access to info. The key drawback of these technologies is that they need cloud vendors to deploy servers, which means increased costs and difficulty of management. The same restriction occurs, if implemented in several clouds, to modern (single-provider) geo-replicated storage structures such as Spanner[71], SPANStore[25] and Pileus[72]. The integration of various file synchronisation systems (e.g. DropBox, Box, Google Drive) into a single stable service proposes a somewhat different form of work [20], [21], [22]. CYRUS[20] does not enforce any form of concurrent control, enabling different clients to build various copies of simultaneously accessed files Disadvantages in the new arrangement The framework is less powerful when working with mutual files in the current job. When dealing with data chunks, the device has fewer security.

**Proposed System**: The framework proposes a CHARON that is a distributed file system that offers a near-POSIX interface for various cloud providers to reach an environment and enables data transfer between customers. Instead of utilising data objects, the option for a POSIX interface resorts to the reality that the intended consumers are likely to be non-experts, and current life sciences tools much of the time use files as their input. In specific, the device requires (1) to manage numerous storage locations effectively, (2) to accommodate relatively large files, and (3) to provide managed file sharing. Our priorities of excluding user-deployed servers and making no improvements to current cloud systems are exacerbated by these obstacles (for immediate deployability).

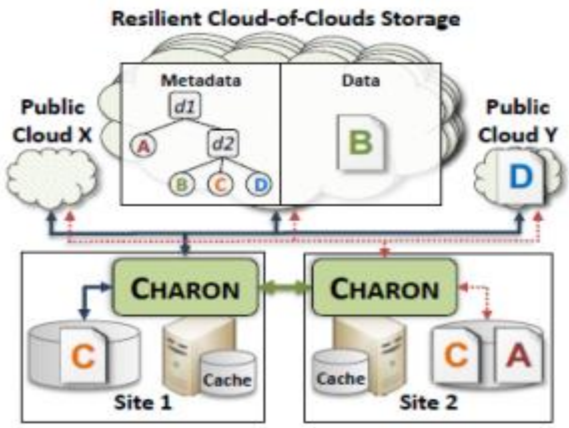


fig3: system architecture

Considering two major design choices, all the methods used in CHARON were merged. Next, the device absorbs files written on the local disc of the client and uploads them to their storage position in the background. Pre fetching and concurrent updates are similarly commonly used to speed up readings. This increases CHARON's accessibility as it requires significant time to migrate massive files to/from the clouds (see x5).

Second, the framework eliminates write-write disputes, leaving out any positive method that depends on dispute settlement applications/users. This judgement is justified by the projected scale of files and the expected users. More precisely, (1) it can be challenging and time-consuming to settle disputes manually in large files; (2) consumers are likely to be non-experts, typically unsure of how to resolve those conflicts; and (3) the expense of preserving duplicate copies of large files can be high. Collaborative repositories, such as Google Genomics[31], for example, need such control because they enable users to interpret, process and aggregate new information of usable samples by exchanging the resulting derived data in the bucket comprising the sample of interest. Advantages of the method suggested Mutual exclusion (safety): There are never two reasonable consumers for the same resource with a legal contract. Obstruction-freedom (liveness): A right consumer can prosper if he wants to lease are source without dispute. Term-boundedness (life): A right consumer who obtains a lease can hold it for a period of T time units until the lease has been renewed.

## IV. SYSTEM DESIGN

DATA FLOW DIAGRAM:

1.The DFD is often called a bubble map. It is a basic graphical formalism that can be used to describe the system in terms of input data to the system, the different processing carried out on that data, and the output data produced by that system.

2.The data flow diagram (DFD) is one of the most powerful methods for modelling. It is used to model the components of the device. This elements are the mechanism of the method, the details utilised by the process, the external party communicating with the system and the knowledge flows throughout the system.

3.DFD illustrates how the knowledge passes through the device and how it is changed through a sequence of transformations. It is a graphical strategy that represents the movement of information and the transformations that are implemented as data travels from input to output.

4.DFD is often referred to as the bubble map. A DFD may be used to describe a system at any degree of abstraction. DFD can be categorised into levels that reflect an improvement in knowledge flow and functional detail.

## UML DIAGRAMS

UML is the Unified Simulation Language. UML is a structured general purpose modelling language in the area of object-oriented software engineering. The norm is managed and developed by the Object Management Community. The aim is to render UML a popular language for developing object-oriented computer software models. In its present shape, UML consists of two key components: a meta-model and a notation. A type of system or process can also be applied to or connected with UML in the future. Unified Modeling Language is a common language for defining, visualising, designing and recording the functionality of the software framework, as well as for market modelling and other non-software structures. UML is a set of best engineering practises that have proved effective in modelling broad and complex structures. UML is a very significant aspect of the development of object-oriented applications and software development methods. UML mostly utilises graphical notations to express the architecture of software projects.

## GOALS:

The key objectives of the UML specification are as follows:

1.Provide developers of ready-to-use, expressive visual modelling language so that they can create and share meaningful models.

2.Provide extensibility and specialisation structures for the broadening of key principles.

3.Be autonomous of existing programming languages and the creation phase.

4. Provide a systematic framework for interpreting the vocabulary of modeling

5.Encourage the development of the demand for OO tools.

6.Help higher-level architecture principles such as alliances, structures, trends and modules.

7.Integrate best practise.

**Architecture:**


fig4: architecture diagram

## IV. SYSYEM REQUIREMENTS :

- Processor: Intel Core i3.
- Hard drive: 500 GB;
- RAM: 2 GB.

**Software Requirements :**

- Windows 7 as the operating system
- Scripts: JavaScript
- Front End: HTML, Java, Jsp
- MYSQL database
- JDBC database connectivity

## IV. SCREENSHOTS:


## V. CONCLUSION :

CHARON is a cloud based file server. It is able to storing and exchanging massive files. CHARAON architecture is mainly focused on two essential principles. i.e, metadata files and data are maintained in various clouds, without needing individual confidence, and the framework is entirely data-centric. This architecture has contributed to the creation of a novel Byzantine resilient leasing protocol to prevent write-write disputes without any custom server. Our findings demonstrate that this design is practical and can be used in real-world organisations that need to store and exchange broad important datasets in a managed manner.

**REFERENCES:**

[1] Cloud Harmony, “Service Status,” https://cloudharmony.com/ status-of-storage-groupby-regions, 2019.

[2] Cloud Security Alliance, “Top Threats,” https://cloudsecurityalliance.org/ group/top-threats/, 2016.

[3] M. A. C. Dekker, “Critical Cloud Computing: A CIIP perspective on cloud computing services (v1.0),” European Network and Information Security Agency (ENISA), Tech. Rep., 2012.

[4] H. S. Gunawi et al., “Why does the cloud stop computing?: Lessons from hundreds of service outages,” in Proc. of the SoCC, 2016.

[5] European Commission, “Data protection,” https://ec.europa.eu/info/law/ law-topic/dataprotection en, 2018.

[6] G. Gaskell and M. W. Bauer, Genomics and Society: Legal, Ethical and Social Dimensions. Routledge, 2013.

[7] A. Bessani et al., “BiobankCloud: a platform for the secure storage, sharing, and processing of large biomedical data sets,” in DMAH, 2015.

[8] H. Gottweis et al., “Biobanks for Europe: A challenge for governance,” European Commission, Directorate-General for Research and Innovation, Tech. Rep., 2012.

[9] P. E. Verissimo and A. Bessani, “E-biobanking: What have you done to my cell samples?” IEEE Security Privacy, vol. 11, no. 6, pp. 62–65, 2013.

[10] P. R. Burton et al., “Size matters: just how big is big? Quantifying realistic sample size requirements for human genome epidemiology,” Int J Epidemiol, vol. 38, no.1,pp.263–273,2009.