Sahana *et al. Braz J Sci Technol (2016) 3:12*
DOI 10.1186/s40552-016-0024-5

Brazilian Journal of Science and Technology

RESEARCH Open Access

Harnessing RAID mechanism for enhancement of data storage and security on cloud

Sudipta Sahana[1*], Rajesh Bose[2] and Debabrata Sarddar[2]

*Correspondence:
ss.jisce@gmail.com
[1] Department of CSE, JIS College of Engineering, Kalyani, Nadia, West Bengal, India
Full list of author information is available at the end of the article

Abstract

As one of the most sought-after technologies in the world today, cloud computing finds a prominent place among researchers, service providers, hardware and software engineers, and consumers alike. Among the key services provided by cloud computing, the most widely used is storage on the cloud. To ensure a high degree of reliability and an almost constant state of availability, cloud storage systems rely on replication of data. Replicating data is not without its costs. In an attempt to mitigate costs of storage, designers have banked on erasure coding practices in lieu of data replication. This has resulted in development of Cloud RAID systems. An ideal Cloud RAID system strikes an optimum balance between performance, storage, and reliability. In addition to this, data security is of paramount interest. In this paper, we introduce an entirely new technique of storing data in cloud networks by file type classification. Combining RAID types with this, our paper proposes a model that is robust and can withstand instances of hard disk failures. Further, we discuss a methodology that is embedded in our proposed framework that supports user authentication for data storage and retrieval. Our Secure User Authenticated Cloud RAID model shields user data from prying eyes and even from unauthorized attempts by the cloud service provider our Secure User Authenticated Cloud RAID architecture.

Keywords: Cloud computing, Cloud security, RAID, Cloud storage security, Cloud RAID

Background

Introduction

Cloud computing offers a number of advantages not evident earlier in the annals of information technology. It is being touted by experts and researchers alike as the future of information architecture that would find prominence in large scale applications. What sets apart cloud computing in terms of enterprise computing and network is the ability to extend users to access services based on individual requirements with little or minimal intervention that has the immediate effect of mitigating procedural delays. Further, resources on the Cloud can be access from almost anywhere with network access. Cloud administrators are able to quickly deploy resources and take advantage of flexible location-agnostic resource federation. Cloud service providers are in a position to be able to adjust pricing to reach maximum operating profit levels without sacrificing on

quality and user satisfaction. As an evolving technology, cloud computing has opened up a whole new market that has made deep inroads in the way traditional data center architecture or models were generally thought of. What began as somewhat of a hesitant acceptance, quickly turned into a free acceptance of Cloud models commercially available, today. Most large enterprises and business have come to realize the importance of the OPEX approach cradled by cloud computing as opposed to the more cumbersome CAPEX model.

Cloud computing represents a major change in the manner in which data storage is perceived. From being centralized, data can now be spread across the cloud. The way in which data is spread, has several tangible benefits. These are realized in the form of flexible storage management, ability to access this data from any corner of the globe with suitable network access, and optimization of capital and human resource expenditures.

With the advantages that cloud computing brings along with it, so does the risks concerning data security rise along with the benefits gained. Ensuring data integrity and data verification with untrusted servers is a major hurdle concerning cloud data storage. To hide its failings from its consumers, a cloud storage provider may glean an undue advantage. However, in most cases, the seemingly Byzantine nature of the problems, a cloud storage service provider might simply want to push such errors out of prying eyes.

It, therefore, become a major challenge to ensure the accuracy of data sourced from the cloud without the benefit of having a local copy of the same data to check for consistency and validation. It could be argued that data can always be downloaded to conduct integrity checks. But then it would also be cost-prohibitive, not to mention, of course, being counter-productive at the same time. Downloading data in bulks, or even in small tranches, in order to verify would only serve to burden the associated networks. Most cloud users would find this to be quite a daunting task to inspect for data veracity on a regular basis. Further, data may already be lost or irrevocably damaged by the time it is checked for any anomalies or errors that may have crept in. To solve this problem, several models have been proposed through various research works.

To increase reliability of stored data, and to increase performance, RAID technology has been in use for a long time. RAID is an abbreviation for Redundant Array of Inexpensive Disks. In its simplest form, it involves two drives or several in parallel configuration. Traditionally, RAID has always been implemented using hard disks. However, as storage technology evolved and became available at affordable rates, solid state drives are often used in cases where RAID implementation is a necessity.

RAID may have different forms or levels. Each such level is suitable for a specific task. These are, however, not regulated by any committee formed by manufacturers from the industry. Each RAID level is associated with a number. As such, it is not uncommon to find companies designing their own RAID level and numbering to meet their requirements. The software required to manage RAID features is either packaged and deployed as a driver, or embedded on hardware—usually known as RAID controller card.

RAID can be interfaced with IDE, SCSI, SATA, or FC. RAID can also be implemented in situations where systems have internal SATA disks, but are interfaced using SCSI or FireWire. RAID implementations today can take several forms each of which is unique and geared to provide the best optimization given the needs and demands of the data storage and performance.

Simply having a number of disks running in a storage system is not a reflection of RAID implementation. Such disks act independent of each other, and are usually used to host meant for spooling data or swap files. These disks are not managed by any RAID level, and are defined as "Just a Bunch of Disks" or JBOD.

The rest of the paper is subdivided into four sections consisting of "Related work", Methods, "Results and discussion" and "Conclusions".

Related works

The concept of multi-tenancy is founded in cloud computing itself. Multi-tenancy offers flexibility, robustness, and instant provisioning of services on demand. However, multi-tenancy has roots in the fundamental concept that each application would run in its own secure virtual environment. Each such application is known as a tenant. Multi-tenancy in a cloud computing environment is achieved through virtual machine multiplexing (Ristenpart et al. 2009). In other words, virtual machines serving different users are hosted on the same server hardware.

To leverage the best of both worlds, extensive research has been conducted to combine private cloud and cloud storage services (Deng et al. 2010). It has been proposed by Wu et al. (2012) to harness the infrastructure of cloud storage and shield users from the complexity of managing IT services. Both these research works are based on cloud storage, and focus on custom specifications, scalability, reliability and high performance.

Research methodologies in the field of cloud storage applications have been proposed in Feel and Khafagy (2011), Srinivasan et al. (2011), He et al. (2010). There have been a significant number of proposed methods to tackle the issue of data security in the context of cloud computing. Maintaining data security in public cloud is costly in terms of building necessary software frameworks that programmatically control secure access to the data, or hardware resources that act as secure gateways. Proposing an architecture that can allow users to access data in a secure manner, Koletka and Hutchison (2011) also formulated a framework that allows for searching through encrypted user data.

A somewhat common approach is observed in Hao and Han (2011). Zhang, etc. has examined traditional storage model to that of private cloud storage (Zhang et al. 2011). The authors examine the efficacy and benefits of private cloud storage technology, and present methods based on Hadoop that offer mass data storage and storage extensions which are adaptable. The storage system proposed by the authors demonstrates the suitability of private cloud medium as a storage platform for highly intense business processes and applications. Service Level Agreement (SLA) has been used as a common standard to ensure cloud storage data securing between cloud service provider and user (Zhang et al. 2011).

In the course of their research, Liu (2012), investigates security problems associated with cloud computing. The authors highlight the challenges faced in terms of security and the benefits that can be achieved from data storage on private cloud.

Private cloud storage is not without its inherent risks. Organizations would need to plan well in advance to factor in critical issues such as drop or total failure of networks. In such scenarios, requests for data access should be directed seamlessly to copies present in other data centers. This usually involves data replication to maintain redundancy, and offer a consistently high level of data protection. In situations such as these, data can

be imported from clouds as and when required. It is obvious that cloud storage is faced by a challenge that demands the assurance of high availability. A framework modeled on component-based availability has been proposed (Machida et al. 2011) to ascertain systems availability. This has been formulated keeping in mind a comprehensive architecture for further enabling cloud services.

Addis et al. (2010) have focused on automatic management in cloud computing to maximize efficiency of deploying resources. Elasticity has been the primary motivation in this work. By examining availability of resources and characteristics of application, Widjajarto et al. (2012) have proposed designing a reference model to optimize services to be enabled on-demand. Reducing instances of idle resources in cloud has been the focus of attention of the researchers. Further, in proposing a novel method of placing virtual machines, Jayasinghe et al. (2011) have shown how cloud services can be improved in terms of availability and performance through a structured and constraint-sensitive framework. However, descriptive models illustrating how resources may be borrowed from public cloud to achieve high availability in a private cloud setup, are conspicuous by their absence in works of these researchers.

Ensuring users can utilize applications from any place and at any time is the aim and objective of setting up an infrastructure for cloud system availability. In terms of mission-critical systems and data safety, making certain that users can access cloud services is a key mandate. Concerns relating to availability are also inter-related with the requirement to move from one service provider to the next. Assurance of uptime and long-term viability of a cloud service provider (Chow et al. 2009) are also significant.

Just as an internet service provider is best served by not depending on any one individual network provider, so, too, should a user not rely on a single cloud service provider alone (Abu-Libdeh et al. 2010). Although, failure rates are perceived to be low, reliance on just one cloud service provider in itself gives rise to a single point of failure that can have unpredictable or even catastrophic consequences. Employing the services of more than once cloud service provider is, therefore, advisable (Abu-Libdeh et al. 2010).

Sahana and Sarddar (2015) have introduced a smooth search optimization technique with an enhanced storage management scheme that helps client in storing their data according to their desire place based on their accessing policy and finding data quickly in cloud data center. Based on access policy the storage has been categorized in three segments for quick access of data searching for. The technique consumes less Energy as only the possible segment is accessed instead of the entire storage, taking a special care on less carbon emission generated by the cloud data center.

Methods

The aim of this paper is to present a secure storage management that should be able to handle assorted file types using several RAID implementations. To set up several types of RAID architecture, we have linked different types of storage modules to each server. Our proposed model consists of four servers each associated with four different RAID implementations. Servers 1 through 4 are associated to RAID 0, RAID 1, RAID 5 and RAID 10, respectively. At the gateway of the data center, a supervisor is installed. The function of this supervisor is to filter files based on applicable file formats. The supervisor also communicates with the servers involved in the process. The four different RAID

implementations have been introduced to handle files based on the associated file type. In our model (Fig. 1), Servers 1 and 2 are configured with RAID 0 and RAID 1. These servers are used to handle media types. Data would be stored in Server 2 when a customer chooses the privilege mode. The cost involved is almost doubled as soon as RAID 1 is selected. It has been generally observed that large volumes of data that traverse the internet mostly consist of entertainment content. These files typically consume a significant portion of available bandwidth. RAID level 1 has been used to regularize read/write operations and optimize bandwidth consumption. To enhance data security, cryptography technique has been applied.

Cryptography technique has also been used for document files. Based upon the choice of a user, documents are stored in servers associated with RAID 5 or RAID 10 levels. The supervisor offers a choice to the client for selecting either a normal or secure mode to store documents. In this case, a single bit flag is used to segregate between a plain document and a cipher one. For the encrypted version of document, the flag is set to 1. Using this method, security can be enforced not only at the point where the data enters the data center but also throughout the transmission of the media in the cloud.

We present two tables: Tables 1 and 2, which together demonstrate the basic RAID framework of our proposed model and the manner in which the Supervisor handles assorted file types in terms of the four servers selected.

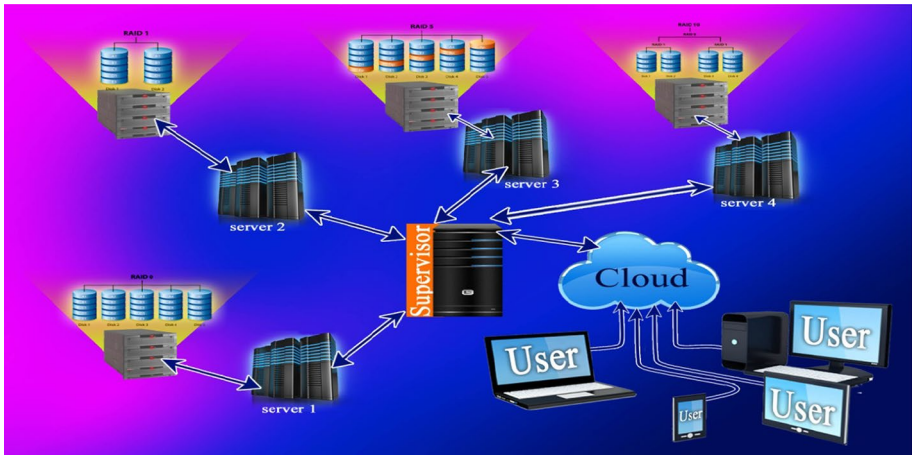


Fig. 1 Proposed diagram

Table 1 Table showing RAID levels and the respective configurations

Type	RAID 0	RAID 1	RAID 5	RAID 10
Performance	Maximum	Slow	Moderate	Maximum
Capacity	(500 × 4)GB	(500 × 4)/2 GB	(4 − 1) × 500 GB	(500 × 4)/2 GB
Security	Minimum in case of disk failure	Maximum in case of disk failure	Moderate	Maximum in case of disk failure
Costing	Minimum	Maximum	More than RAID 0	Maximum

Table 2 Storage handling table(SHT) by supervisor

Type of file	File size	Responsible server
Editable document file along with web file	Irrespective of size	Server 4
Non editable document file	Irrespective of size	Server 3
Executable file	Irrespective of size	Server 3
AV file	Irrespective of size	Server 1
	Irrespective of size (privilege mode)	Server 2
Image file	Irrespective of size	Server 1
	Irrespective of size (privilege mode)	Server 2
Zip file	File size greater than 100 MB	Server 1
	File size greater than 100 MB (privilege mode)	Server 2
	If file size is $\leq$100 MB and contains non editable file mostly in count	Server 3
	If file size is $\leq$100 MB and contains editable file mostly in count	Server 4

Algorithm for data storing

Step 1: Client approaches Data Center to store a file

Step 2: Supervisor responds requesting information on the file type and size

Step 3: In response to the query made, client provides the required information

Step 4: Supervisor matches the values received with the parameters tabulated in the Storage Handling Table (SHT). A synchronization message is sent to the corresponding server

Step 5: In case of the file being of document type, the Supervisor offers client the option of selecting either the normal or the secure mode

Step 6: If client selects the secure mode, then the following actions are performed:

- **Step I:** A secure web page containing crypt code is displayed by Supervisor
- **Step II:** Client selects file for uploading using the web page
- **Step III:** The underlying crypt code works to secure the document during transmission

Supervisor receives the encrypted file and matches the file type and size with the parameters initially supplied by the client.

- **Step IV:** Subject to the file type determined, the Supervisor forwards the file to either Server 2 or Server 3.

Step 7: In case of normal mode selection, client sends the file to the Supervisor without any intervening encryption process

Step 8: The Supervisor checks the file received against values available in the SHT. The file is verified and is sent to the appropriate server.

Encryption technique

To enable secure mode during upload of file, a sequence of steps has been proposed. Since, the complete process is executed at the client side, channel security has also been included in the process. The whole operation is based on encrypting characters. The

following algorithm has been designed for characters with ASCII values in the range of 0 to 255. In case a particular character falls out of the range, it is left unchanged.

Algorithm for encryption

Step 1: Calculate the length of each word that is accepted in plain text format
Step 2: A key is made with the word "EARTH". This may be any user-defined string
Step 3: The plain text format word is converted to ASCII equivalent for each character in the word. The string used as key is also converted to its ASCII representative values
Step 4: Both the ASCII sets of the word provided as input, and the key, are converted to binary equivalents
Step 5: A bit-wise XOR operation is carried out with the characters of the input text and those of the key. Spaces, non-breaking spaces, and tabs are excluded. The key elements are repeated in case where the number of characters of the input string exceeds that of the key
Step 6: Segregate the resultant binary values into two groups each with four bits. Interchange the position of the two groups
Step 7: Join the two groups to form a combined value that is 8-bits long
Step 8: Create four groups of two bits each. Number each of the groups beginning from 1 to 4
Step 9: Change the places of the groups. For example, groups in places 1 to 4 should now occupy places 1, 4, 2 and 3
Step 10: Combine the groups in the interchanged locations to form a single binary value. Compute the value resulting from one's complement on every bit in the odd position of the single binary value. The odd positions would be 1, 3, 5, and 7
Step 11: After obtaining the result following computation of one's complement on each bit in an odd position number, reverse the entire result such that bits in position 0 to 7 are now arranged in position 7 to 0
Step 12: Following rearrangement of the bits, convert the resultant binary data to a corresponding decimal value and, thus, the related ASCII character. This produces the final cipher text
Step 13: The value of the flag for the document is set to 1 to signify that the encryption process has been completed successfully.

Data access technique

To access a stored file from a server, request from the client is received by the Supervisor which then analyzes the request in terms of type of file being sought. Depending on the file type, the Supervisor enables a connection between the client and the server designated to handle such file type. Whenever a client makes a search request for a file, the corresponding flag bit of the document is set to 1. In case the flat bit is not set to 0, the decryption algorithm is enabled at the client end for downloading of the file requested.

Decryption algorithm

Step 1: The cipher text is converted to its binary equivalent by converting each character to its 8-bit binary value

Step 2: The entire 8 bit number is now reversed. Bits in positions 0 to 7 are transposed at locations 7 to 0

Step 3: Obtain the result following computation of one's complement on each bit in an odd position numbers of 1, 3, 5 and 7

Step 4: Create four groups of two bits each. Number each of the groups beginning from 1 to 4

Step 5: Change the places of the groups. For example, groups in places 1 to 4 should now occupy places 1, 4, 2 and 3. Combine the arranged groups into a single binary value

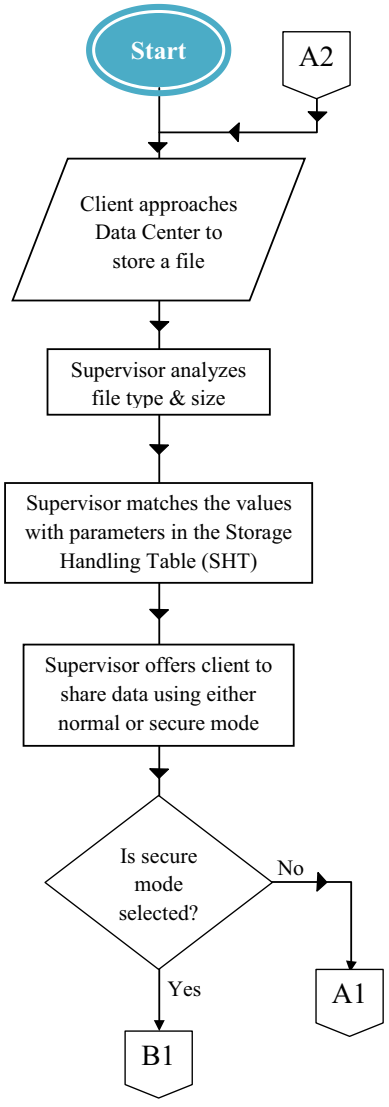
Step 6: Form two four bit binary values by dividing the single binary number

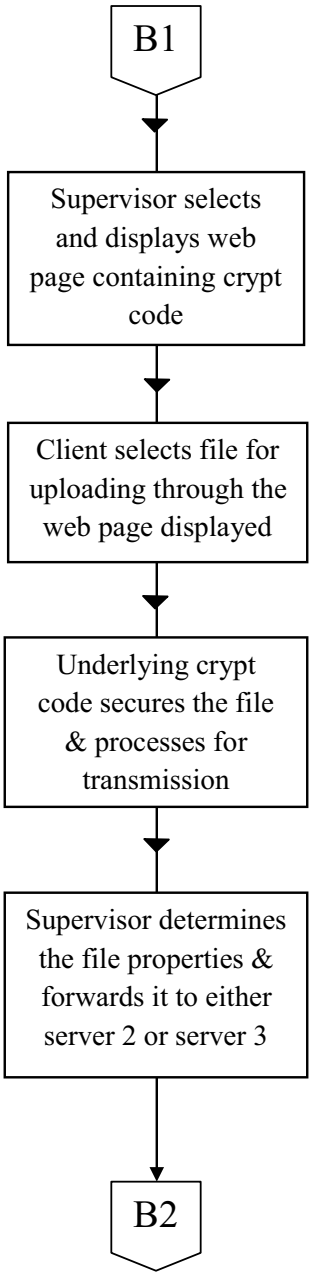
Step 7: Interchange the positions such that the position of the group on the left is switched with that of the right

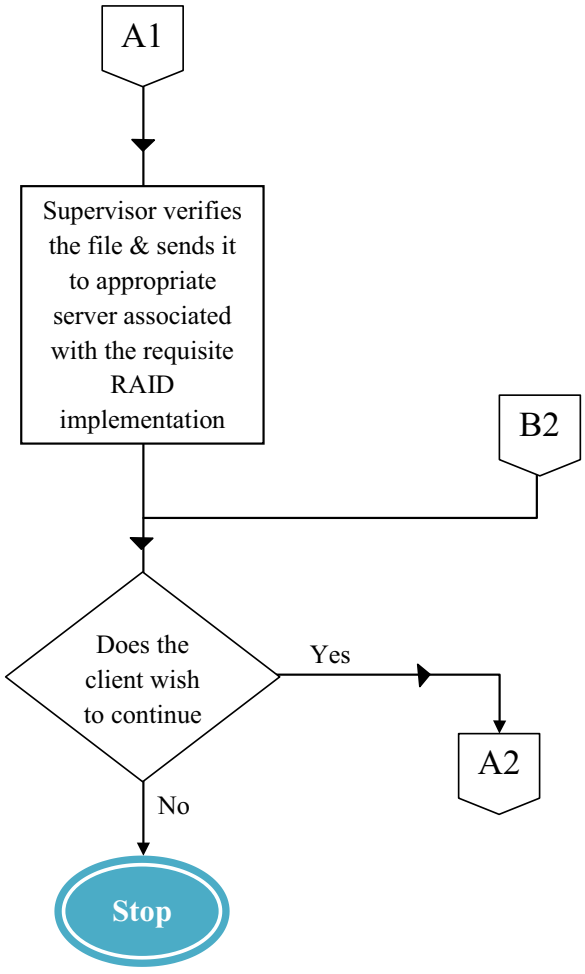
Step 8: Combine the groups to make it into a single binary value

Step 9: A bit-wise XOR operation is to be carried out with the characters of the input text and those of the key. Spaces, non-breaking spaces, and tabs are excluded. The key elements are repeated in case where the number of characters of the input string exceeds that of the key

Flowchart




B1
Supervisor selects and displays web page containing crypt code
Client selects file for uploading through the web page displayed
Underlying crypt code secures the file & processes for transmission
Supervisor determines the file properties & forwards it to either server 2 or server 3
B2



Results and discussion

The paper presents an idea for an effective cloud storage management designed to provide security for several types of file formats. Our literature survey reveals that documents are highly vulnerable to threats through network sources. The approach that we have proposed not only secures documents at a cloud data center but also throughout the transmission of the file. Decentralization of resource translates to ease of data availability. This makes for a significantly efficient method in comparison to traditional processes. Storage at data center is optimized by implementing RAID levels. Data is kept safe from disk failure. Optimized space, time and cost management takes place as a result of RAID level implementation. Analysis of results produced by our proposed model reveals that throughput is maximized while keeping delays regulated. Figures 2 and 3 shown below depicts a graphical performance comparison between proposed and traditional approaches. According to result analysis (Fig. 2) we have seen overall network performance in terms of throughput is much more in our proposed approach while it

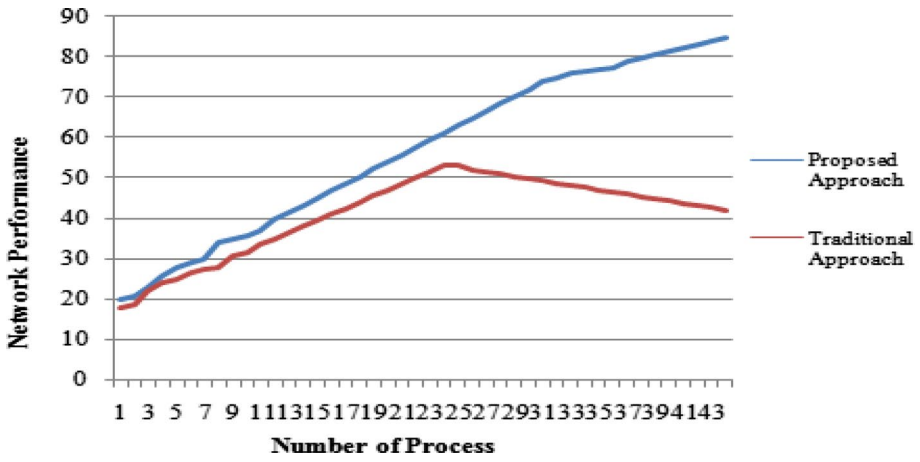


Fig. 2 Proposed approach versus traditional approach in terms of network performance

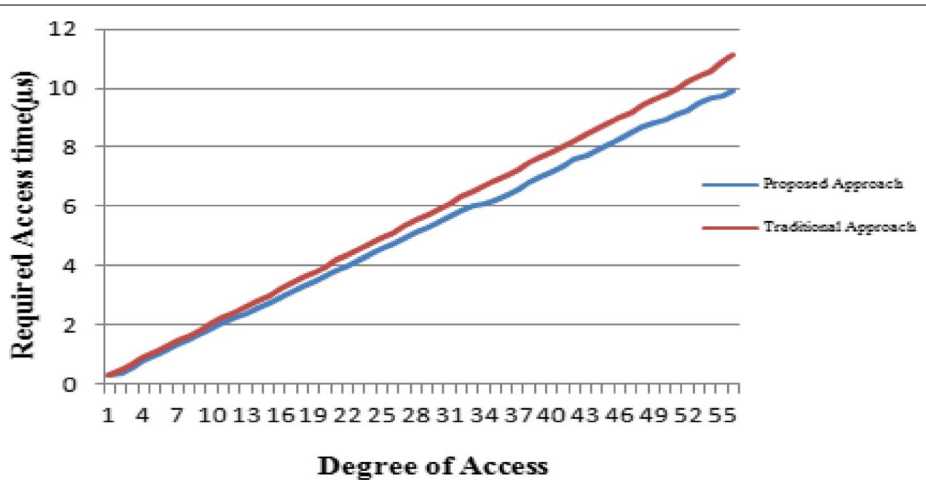


Fig. 3 Proposed Approach versus traditional approach in terms of required access time

deals with multiple numbers of processes. Category wise process handling by different server makes the entire work very efficient and less time consuming (Fig. 3).

Conclusions

Cloud storage is rapidly moving in the direction where tolerance for data storage and security are diminishing progressively. In that context, our proposed approach offers a solution that can effectively decentralize storage of data without compromising data security. Combining encryption technologies with encryption techniques, our model extends a radically new approach that promises protection not only in terms of storage, but also insofar as providing a cloak of encrypted layer to protect from unauthorized access.

Abbreviations
RAID: redundant array of inexpensive disks; SHT: storage handling table.

Authors' contributions
SS, RB and DS contributed to this work in the manuscript preparation. SS was instrumental in preparing the Algorithm and designing the proposed system. He also prepared the architecture and flowchart which details the process flow of the proposed design along with results and discussion. RB has done background, related work and finally concludes this paper. DS provided invaluable assistance in the course of preparing this work. All authors read and approved the final manuscript.

Authors' informations
Debabrata Sarddar Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, INDIA. He has done PhD at Jadavpur University. He completed his M. Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E. in Computer Science and Engineering from NIT, Durgapur in 2001. He has published more than 75 research papers in different journals and conferences. His research interest includes wireless and mobile system and WSN, cloud computing. Sudipta Sahana is an assistant professor of a renowned engineering college of west Bengal. More than 4 years he has worked in this region. He has passed his M.Tech. degree in Software Engineering and B.Tech. Degree in Information Technology from west Bengal university of technology with a great CGPA/DGPA on 2010 and 2012 respectively. He is recently working in Ph.D. on the domain of "cloud computing". He is a member of the Computer Science Teachers Association (CSTA), and also a member of International Association of Computer Science and Information Technology (IACSIT). Rajesh Bose is currently pursuing Ph.D. from Kalyani University. He is an IT professional employed as Senior Project Engineer with Simplex Infrastructures Limited, Data Center, Kolkata. He received his degree in M.Tech. in Mobile Communication and Networking from WBUT in 2007. He received his degree in B.E. in Computer Science and Engineering from BPUT in 2004. He has also several global certifications under his belt. These are CCNA, CCNP-BCRAN, and CCA(Citrix Certified Administrator for Citrix Access Gateway 9 Enterprise Edition), CCA (Citrix Certified Administrator for Citrix Xen App 5 for Windows Server 2008). His research interests include cloud computing, wireless communication and networking.

Author details
[1] Department of CSE, JIS College of Engineering, Kalyani, Nadia, West Bengal, India. [2] Department of CSE, University of Kalyani, Kalyani, West Bengal, India.

Acknowledgements
Authors gratefully acknowledge to CSE Department of University of Kalyani and JIS College of Engineering, for providing lab and related facilities for do the research.

Competing interests
The author(s) declare that they have no competing interests.

Received: 16 January 2016 Accepted: 14 March 2016
Published online: 22 March 2016

References
Abu-Libdeh H, Princehouse L, Weatherspoon H (2010) RACS: a case for cloud storage diversity. In: Proceedings of the 1st ACM symposium on cloud computing, SoCC '10. ACM, New York, pp 229–240
Addis B, Ardagna D, Panicucci B, Zhang L (2010) Autonomic management of cloud service centers with availability guarantees. In: Proceedings of the 3rd international conference on cloud computing (CLOUD 10). IEEE Press, pp 220–227
Chow R et al (2009) Controlling data in the cloud: outsourcing computation without outsourcing control. In: ACM workshop on cloud computing security
Deng J, Hu J, Liu ACM, Wu J (2010) Research and application of cloud storage. In: Proceedings of the 2nd international workshop on intelligent systems and applications (ISA 10). IEEE Press, pp 1–5
Feel HTA, Khafagy MH (2011) OCSS: ontology cloud storage system. In: Proceedings of the first international symposium on network cloud computing and applications (NCCA 11). IEEE Press, pp 9–13
Hao L, Han D (2011) The study and design on secure-cloud storage system. In: Proceedings of the international conference on electrical and control engineering (ICECE 11). IEEE Press, pp 5126–5129
He Q, Li Z, Zhang X (2010) Study on cloud storage system based on distributed storage systems. In: Proceedings of the international conference on computational and information sciences (ICCIS 11). IEEE Press, pp 1332–1335
Jayasinghe D, Pu C, Eilam T, Steinder M, Whalley I, Snible E (2011) Improving performance and availability of services hosted on IaaS clouds with structural constraint-aware virtual machine placement. In: Proceedings of the international conference on services computing (SCC 11). IEEE Press, pp 72–79
Koletka R, Hutchison A (2011) An architecture for secure searchable cloud storage. In: Proceedings of the international conference on information security South Africa (ISSA 11). IEEE Press, pp 1–7
Liu W (2012) Research on cloud computing security problem and strategy. In: Proceedings of the 2nd international conference on consumer electronics, communications and networks (CECNet 12). IEE Press, pp 1216–1219
Machida F, Andrade E, Kim DS, Trivedi KS (2011) Candy: component-based availability modeling framework for cloud service management using SysML. In: Proceedings of the 30th IEEE symposium on reliable distributed systems (SRDS 11). IEEE Press, pp 209–218

Ristenpart T, Tromer E, Shacham H, Savage S (2009) Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of ACM conference on computer and communications security (CCS 2009), pp 199–212

Sahana RBS, Sarddar D (2015) An enhanced storage management scheme with search optimization for cloud data center. Int J Appl Eng Res. 10(12):32141–32150. ISSN: 0973-4562

Srinivasan J, Wei W, Ma X, Yu T (2011) EMFS: email-based personal cloud storage. In: Proceeding of the 6th international conference on networking, architecture and storage (NAS 11). IEEE Press, pp 248–257

Widjajarto A, Supangkat SH, Gondokaryono YS, Prihatmanto AS (2012) Cloud computing reference model: the modelling of service availability based on application profile and resource allocation. In: Proceedings of the international conference on cloud computing and social networking (ICCCSN 12). IEEE Press, pp 1–4

Wu J, Ping L, Ge X, Wang Y, Fu J (2012) Cloud storage as the infrastructure of cloud computing. In: Proceedings of the international conference on intelligent computing and cognitive informatics (ICICCI 10). IEEE Press, pp 380–383

Zhang D, Sun F, Cheng X, Liu C (2011) Research on hadoop-based enterprise file cloud storage system. In: Proceedings of the 3rd international conference on awareness science and technology (iCAST 11). IEEE Press, pp 434–437

Zhang X, Du H, Chen J, Lin Y, Zeng L (2011) Ensure data security in cloud storage. In: Proceedings of the international conference on network computing and information security (NCIS 11). IEEE Press, pp 284–287