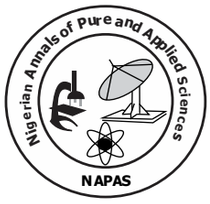

User-Centric Cyber Disaster Recovery as a Service

***Karim, U.[1], Inyiama, H.C.[2] and Karim, R.[3]**

[1]Department of Mathematics and Computer Science, Benue State University, Makurdi, Nigeria, kusman@bsum.edu.ng

[2]Department of Electronics and Computer Engineering, NnamdiAzikiwe University, Awka, Nigeria

[3]Software Development Unit, Kreative Information Technology Nigeria Limited

Abstract

In a world of interdependent economies and online transactions, a large volume of data hosted on the cyberspace a daily bases. Cyber threats and attacks are steadily increasing. Most time, these threats and attacks are targeted at service providers but service users are greatly affected by the attacks due to their vulnerability level. When disasters knockdown the infrastructures of a single service provider, it will have ripple effects on thousands of innocent service users. Therefore, service users need more than ever to prepare for major crises targeted at their service providers. To cope with this trends, every service user requires an independent business continuity plan (BCP) or disaster recovery plan (DRP) and data backup policy which falls within their cost constraints while achieving the target recovery requirements in terms of recovery time objective (RTO) and recovery point objective (RPO). The aim of this paper is to develop a model for a user-centric disaster recovery system to enable service users to independently develop their data backup policies that best suits their remote databases, and host same as a cloud service deployable on public cloud for users to subscribe to and be billed on pay-as-you-go billing model. The system developed is highly compatible with MYSQL, MSSQL and Oracle databases. A combination of Dynamic System Development Methodology (DSDM) and Object-Oriented Analysis and Design Methodology (OOADM) were used to design the system while Java Enterprise Edition (JEE) is used to develop the system. The encryption and compression mechanisms of the system were tested with various sizes of backup files ranging from 64 Kb to 20Mb and several performance metrics such as (1) Encryption time; (2) Compression size; (3) CPU clock cycles and battery power are compared and analysed with some well-known encryption and compression algorithms.

Keywords: BCP, DRP, RTO, RPO, Public Cloud, Cyberspace

Introduction

In the world of technology, change is the only constant factor. The change is so dynamic that what was considered in one era to be an obsolete technology can become the current trend in another era. Computing in the past three decades witnessed a tremendous change in the way and manner data were processed which resulted in the emergence of distributed systems against the pre-existing centralized systems. It is interesting to see that presently, computing is going back to some sort of centralization with a brand name called Cloud computing (Westerlund & Kratzke, 2018).

Prior to the emergence of cloud computing, supercomputers were used in specific areas like the military, government agencies, universities and research laboratories to handle enormous complex calculations. Cloud computing, therefore, aims at further diversifying the use of supercomputers by applying their power to solving problems that require complex computational resources (Buyya & Son, 2018). With the Internet connection, users are granted immediate access to a large number of the world's most sophisticated supercomputers together with their corresponding processing power, interconnected at diverse locations around the world.

The evolution of cloud services has enabled entities to do more with less few resources and better operating efficiency. This has many tangible benefits for business, however, there are inherent security risks that must be evaluated, addressed, and resolved before business owners will have confidence in completely outsourcing their IT requirements to service providers. IT companies take security, performance, data availability and difficulty in bringing back data in-house (i.e. Data Backup) as top challenging factors inhibiting them from adopting cloud services (Jangra & Bala, 2012). These hiccups explain why many business owners and some government agencies are yet to trust and utilize the immense benefits of cloud computing. Many enterprises which have planned to migrate to cloud prefer using the cloud for less sensitive data and store important data within enterprise boundary (Jangra & Bala, 2012).

It is important to note, however, that no matter how careful you are with your personal data, by subscribing to cloud services, you will be giving up control to an external source. This distance between you and the physical location of your data creates a barrier. It may also create more space for a third party to access your information without your knowledge or approval. With this, regular back up of your private data becomes very difficult. This Inadequate data backups and improper data syncing are what has made many businesses vulnerable to ransomware, a specific type of cloud security threat (Stergiou, Psannis, Kim, & Gupta, 2018).

Presently, most service users depend on the security and the backup policies provided for them by their service providers (Rachana & Guruprasad, 2014). In an event of major attach targeted at a single service provider, thousands of service users who have their enterprise applications hosted on the infrastructures of the service users will be greatly affected even when the attack is not originally targeted at them. To cope with this trends, every service user requires an independent business continuity plan (BCP) or disaster recovery plan (DRP) and data backup policy which falls within their cost constraints while achieving the target recovery requirements in terms of recovery time objective (RTO) and recovery point objective (RPO).In (Usman & Inyiama, 2019)auser-centric cyber disaster recovery system which permits service users to periodically creates back up of their remote databases and transfer such files to their premises was proposed. However, when these backup files are created, transferring them from service providers' premises to service users' premises can be very challenging. The backup file may be attack by malicious users thereby compromising the entire process or it might be too large to be easily downloaded which will lead to much consumption of network resources.

Most of the available disaster recovery systems are tilted toward service users. This explains the reason why service users most of the time have to depend on their service providers for disaster recovery processing. This in one of the gaps this work hope to address.

This paper is organized as follows: Section I Introduces the current ways of backup policies implementation by most service providers. Section II describes some of Disaster Recovery Solutions currently available in the cyber space. Most of these Disaster Recovery Solutions are completely implemented and

controlled within the premises of service providers. Section III proposes a user-centric cyber disaster recovery model. With this model, backup policies will be controlled by service users rather than service providers. Section IV describes the methodology with which the new model will be designed and implemented. Section V presents and analyses the results obtained from the prototype designed to simulate the model. Section VI listsome area where further research can be done in future. Section VII provides conclusions derived out of this study.

Related Works

Rajagopalan, S., *et al*. (Rajagopalan, Cully, O'Connor, & Warfield, 2012) present a disaster tolerance as a service called Second Site. This platform is intended to handle three challenges: Reducing RPO, Failure detection and Service restoration. Second Site increases ability to fast failure detection and also differentiate between network failures and host failures. Using DRDB, resynchronize storage can be done for recovering primary site without VMs interruption in the backup

Cully, B., *et al*. (Cully *et al*., 2008) present Remus as a high availability cloud service to tolerate disaster using storage replication combined with live VM migration. In this system, protected software is encapsulated in the virtual machines to asynchronously replicate whole-system checkpoints in a backup site with a high frequency. It is assumed that both replicas are in the same local area network (LAN). Remus is aimed at three main goals:

1. Providing low-level service to gain generality
2. Transparency
3. Seamless failure recovery.

Remus uses an active primary host and a passive backup host to replicate checkpoints. All writes have to be stored in backup RAM until a checkpoint completes. Migrated Virtual machines execute on the backup only if a failure is detected.

Romulus as a Disaster Recovery Solution is an extension of the Remus system (Caraman, Moraru, Dan, & Kristaly, 2009). It is based on the KVM hypervisor (Kivity, Kamay, Laor, Lublin, & Liguori, 2007). Romulus provides an accurate algorithm for disaster tolerant in seven detailed stages which are:

1. Disk replication and network protection
2. VM checkpoint
3. Checkpoint synchronization
4. Additional disk replication and network protection
5. VM replication
6. Replication synchronization
7. Failure detection and failover.

The flaw of Remus is that it uses one buffer to replicate writes between primary host and backup. If a failure occurs in this buffer before transferring checkpoint, it causes an inconsistency between the disk and VM state; and it can break fault tolerance of Remus. For this reason, Romulus uses a new buffer to replicate disk writes after any checkpoint. The second flaw is that network egress traffic cannot be released until completely transferring checkpoint to storage backup host which can decrease system performance.

Chang, F.W., *et al*. [10] propose a new approach for achieving disaster tolerance in large, geographically-distributed storage systems. The system, called *Myriad*, can achieve the same level of disaster tolerance as a typical single mirrored solution, but uses considerably fewer physical resources, by employing cross-site checksums (via erasure codes) instead of direct replication (Chang *et al*., 2002). It provides a disaster tolerant service with respect to resource allocation issue which is a challenge in DT services. Host and backup clusters are monitored by high availability controllers. Each cluster has three different controllers:

1. Storage controller: To control and manage the cluster storage.
2. Cluster controller: To manage IPs, centralized memory and CPU availability.
3. Node controller: To load, start and stop the VM.

Different nodes and also different clusters can communicate with each other for better resource allocation. For this purpose, the backup cluster controller allocates a VM to a node. Then, node controller loads and starts the

VM and allocates it to the primary host. Finally, primary node controller loads and starts the VM.

Kemari is a cluster system which tries to keep VMs transparently running in the event of hardware failures (Tamura, Sato, Kihara, & Moriai, 2008). Kemari uses the primary-backup approach so that any storage or network event that changes the state of the primary VM must be synchronized in backup VM. This system has gained the benefits of Lock stepping and the Check pointing (Bressoud & Schneider, 1996). Two main approaches for synchronizing VM state include:

1. Less complexity compared to lock stepping approach.
2. It does not need any external buffering mechanisms which can affect the output latency.

RUBiS, according to (Wood et al., 2010) is a cloud architecture aims at both Disaster Recovery and also minimizing costs with respect to the Service Level Agreement. In ordinary operation, a primary data centre including some servers and a database accomplish normal traffics. A cloud is in charge of disaster recovery with two types of resources;

1. Replication mode resources for getting backup files before a disaster which is active
2. Failover mode resources that will be activated only after a disaster

It is notable that service providers can rent inactive resources to other customers for revenue maximization. In the case of a disaster, leased resources must be released and allocated to the failover procedure.

MTN Backup as a Service (BaaS) is a prepaid hosting service that offers MTN subscribers the ability to back up their SIM and Phone securely onto backup servers in the MTN datacentre via the internet contacts (Paschal, 2016). The MTN BaaS is a service that provides MTN subscribers with the ability to initiate and manage backups and restores on their phones virtually from anywhere in the world via a secure connection. The service requires subscribers to buy a new MTN 128k SIM Card with MTN Backup enabled, in order to back up the contacts stored on their SIM. The service is not compatible with other networks or MTN SIM that is not MTN Backup enabled. It is ideal for small-size data backup from mobile devices. It does not use the pay-as-you-go billing system.

Chen &Zheng (2008) designed Cross Platform Backup System. The work was an XML-based GUI for cross platform backup system (Chen & Zheng, 2008).The interface definitions are based on XML storage format. The implementation of the work achieves the plug-in management for centrally backup/restore processing of heterogeneous systems. The system is compatible with most DBMS. It also provides friendly user interfaces. However, it lacks system security since it does not encrypt data. Data leakages are also very possible in multi tenant's environment.

Dhane & Joshi (2015) proposes Auto-Data Recovery System on Cloud Scheme. This scheme offers data storage and sharing services to users (Dhane & Joshi, 2015). The system uses a Trusted Third Party Auditor (TTPA) to publicly audit the integrity of shared data in the cloud for users. In a group, there is one original user and a number of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is further divided into a number of blocks. A user can modify a block in shared data by performing an insert, delete or update operation on the block. However, the system lack adequate data security and prevention from Man in the Middle (MITM) attack.

Mayur &Vani (2016) developed Server Virtualization Data Storage and Backup System integrated server using private cloud model for backup and data storage(Mayur & Vani, 2016). They developed an application for the implementation in such a way that automatically synchronizes all information backed up or stored by the user in the virtual folder to the cloud. They used Parallel Virtual File System (PVFS) for data storage in order to increase the performance of applications that requires high I/O data demands. PVFS is an open source file system. The system was able to reduce access time to data by allowing both input and output

operations to be carried out simultaneously. On the client-side, an application is developed that allows data to be transferred much faster. The advantages of this implementation are that it can reuse existing infrastructure (servers, cluster, and other devices) that reduces the cost and increases the throughput. However, backup files are not encrypted and this makes it very vulnerable for Man in the Middle (MITM) Attack.

The Proposed Model

In this work, we present a model for user-centric cyber disaster recovery system. With this system, service users have the liberty to independently define and implement their private backup plans and disaster recovery policies. Platforms are created for service users to configure their remote databases by selecting the entities to be backed up and for each selected entity, the backup frequency is also selected. The system creates backup files from remote databases in accordance with their configuration settings. The backup files are encrypted to prevent its contents from Man in the Middle attacks (MITM). The encrypted backup files are also compressed to enhance its transmission across networks primary reason for encrypting the backup file is to prevent it from Man in the Middle (MITM) attack. In the event of any disaster or service interruption, users will then retrieve their encrypted backup files, decrypt and restore their services with another service provider and their businesses with continue to run with very little or no delay. With the new system, service users can now have their own private organized disaster recovery plan instead of depending on service providers aimlessly and helplessly for their disaster recovery processes.The new User Centric Cyber Disaster Recovery System will automatically generate data backup files at the pre-set interval of time (T). A checksum algorithm is used to compare the backup file (Fn) generated at time (Tn) with the previous backup (Fn-1) generated at time (Tn-1). This algorithm is used to calculate the modular difference between the two files. If the checksum calculator shows any difference in the files then the new backup file will be generated and stored on the system else it will be discarded. Once backup files are generated for storage at any given time, the system automatically encrypts and place it on a private secured compartment created for the users. The system maintains privately secured compartments for every registered user to enhance data segregation in a Multi-tenancy Architecture. The users log onto the system and download encrypted backup files to their local system. The primary reason for encrypting the backup file is to prevent it from Man in the Middle (MITM) attack. In the event of any disaster or service interruption, users will then retrieve their encrypted backup files, decrypt and restore their services with another service provider and their businesses with continue to run with very little or no delay. With the new system, service users can now have their own private organized disaster recovery plan instead of depending on service providers aimlessly and helplessly for their disaster recovery processes.

Methodology

A combination of the iterative method of Dynamic System Development Methodology (DSDM) and Object-Oriented Analysis and Design Methodology (OOADM) is used in this work. DSDM assumes that all previous steps may be revisited as part of its iterative approach. Therefore, the current step need be completed only enough to move to the next step, since it can be finished in a later iteration. The premise is that the business requirements may probably change as understanding increases, such that any further work would probably be a waste. According to this approach, the time is taken as a constraint i.e. the time is fixed; resources are fixed while the requirements are allowed to change. This does not follow the fundamental assumption of making a perfect system the first time but provides a usable and useful 80% of the desired system in 20% of the total development time. This approach has proved to be very useful under time constraints and varying requirements. On the other hand, the OOADM methodology is used to identify the objects needed in the system and their interrelationships. Adequate and relevant UML diagrams such as class diagram, use case diagram, activity diagram and deployment diagrams were generated which makes the coding process quite easy and

straightforward. Most qualities of object-oriented programming such as polymorphism, inheritance, encapsulation and code reusability were employed in the development of the new user-centric disaster recovery system which was able to identify, authenticate and assign functionalities to different categories of users based on their login detail.

Results and Analyses

For our experiment, the researchers use a laptop with the following processor configuration: Intel(R) Core (TM) i5-2540M CPU @ 2.60GHz, RAM of 6GB and Hard disk of 500GB. The Operating System is Windows 10 64-bit. Performance data is collected. In the experiments, various sizes of remote databases are used and the backup files were encrypted and compressed. The sizes of the backup files range from 64Kb to 20Mb. Several performance metrics are collected: 1) Encryption time; 2) Compression size; 3) CPU clock cycles and battery power. The encryption time is considered the time that an encryption algorithm takes to produce a cypher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time.The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time is used in the encryption process, the higher is the load of the CPU. The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

The following comparison analysis tasks performed are as shown below;

1. A comparison is conducted between the results of the selected different encryption schemes in terms of the encryption time of five different encryption algorithm with ten different sizes of backup files in .sql format. The result is as shown in Table 1 & Figure 1

A comparison is also conducted between the results of the selected different compression schemes in terms of the sizes of the compressed files of four different compression algorithm with ten different sizes of backup files. The result is as shown in Table 2 and Figure 2

Table 1: Comparison between DES, 3DES, BF, AES, and the Hybrid Encryption Time(s)

Input File Size (Kb)	DES (s)	3DES (s)	BF (s)	AES (s)	M-AES(s)
64.00	0.01	0.02	0.01	0.01	0.01
128.00	0.02	0.05	0.01	0.02	0.03
512.00	0.06	0.19	0.05	0.10	0.10
1,024.00	0.13	0.38	0.10	0.19	0.20
5,120.00	0.64	1.91	0.51	0.96	1.00
8,192.00	2.02	3.06	1.82	1.53	1.60
10,240.00	3.27	3.83	2.03	1.91	2.01
15,360.00	3.91	5.74	3.54	2.87	3.01
18,432.00	4.29	6.89	3.85	3.44	3.61
20,480.00	4.55	7.66	4.05	3.83	4.01

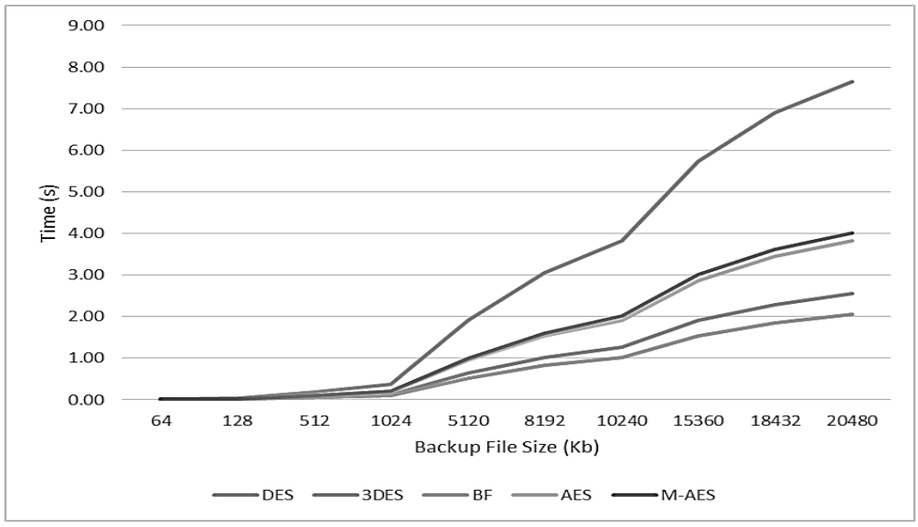

Figure 1: Graph of Input File Size versus Encryption Time for some encryption algorithms

The results are as expected. The Modified AES (M-AES) requires more processing time than the Advanced Encryption Standard (AES) because of its key-chaining nature. The results are shown in Table 1 and Figure 1 indicates also that the extra time added is not significant for many applications, knowing that the new system is an enhancement over AES and that it better in terms of file protection from Man in the Middle (MITM) attacks.

Also in this work, the encrypted backup files are always compressed to enhanced easy transmission of the files over network facilities. A modified version of Huffman Coding compression algorithm is used to compress backup files. The researchers also conducted a comparative analysis of compression module of the work with some other well-known file compression algorithms with various sizes of backup files ranging from 64 Kb to 20Mb and the result is as presented in Table 2 below;

Table 4: Comparison between LZW, Huffman Coding, Shannon Coding and the M-***Huffman***

Input File Size (Kb)	**LZW (Kb)**	**Huffman Coding (Kb)**	**Shannon-Fan Coding (Kb)**	**M-Huffman Coding (Kb)**
64.00	36.57	18.29	18.82	16.00
128.00	73.14	36.57	37.65	32.00
512.00	292.57	146.29	150.59	128.00
1,024.00	585.14	292.57	301.18	256.00
5,120.00	2,925.71	1,462.86	1,505.88	1,280.00
8,192.00	4,681.14	2,340.57	2,409.41	2,048.00
10,240.00	5,851.43	2,925.71	3,011.76	2,560.00
15,360.00	8,777.14	4,388.57	4,517.65	2,800.00
18,432.00	10,532.57	5,266.29	5,421.18	2,800.00
20,480.00	11,702.86	5,851.43	6,023.53	2,800.00

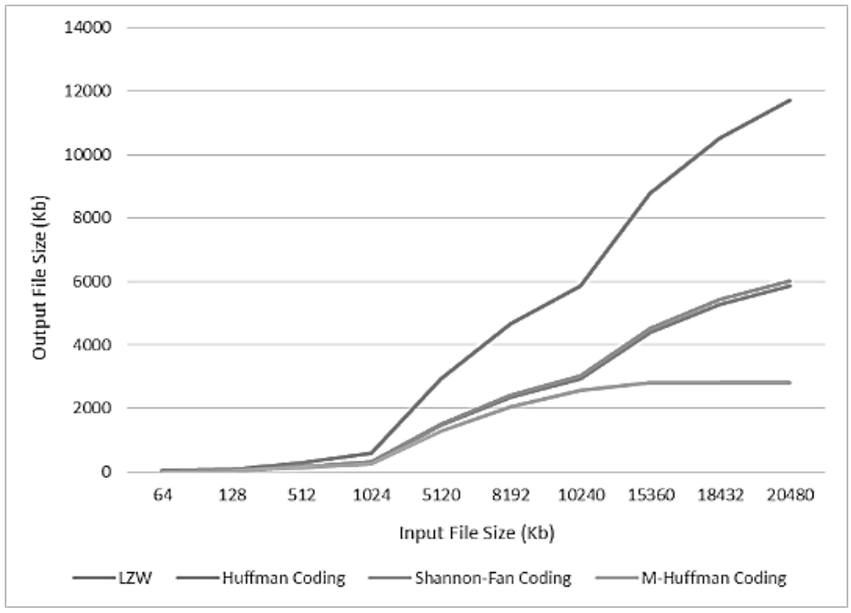

Figure 4.2: *Graph of Input File Size versus Compressed Output File Size*

The results from Table 2 and Figure 2 show the superiority of the hybrid compression algorithm used in the work over other algorithms. Figure 2 also shows that as the size of input file increases, the output file size of the hybrid algorithm converges.

Further Research

This paper has addressed the issues of incompatibility of storage services used by various Service Providers and the inaccessibility of Backup files stored in the premises of Service Providers by Service Users. However, the contribution to knowledge is not completely exhaustive.

The model can be enhanced to be compatible with all Relational Database Management Systems (RDBMS). Also, further research work should be done to investigate the possibility of coordinating all Service Providers in a view to deploying a common gateway for their storage management policy.

Conclusion

The research is very useful in many governmental and non-government institutions around the globe where application hosting on cyberspace has become eminent. In Nigeria for instance, it will be useful for Military and all the Para-Military to deploy this solution to enable them to bring their operation data which are currently hosted by Service Providers back to their premises. All Universities, Polytechnics and Colleges of Education in Nigeria have one portal or the other with which the institutions are being managed. Currently, all of these portals have their operational data in the custody of Service Providers. The findings of this research are useful to such institutions as it will help them to have access to their operational data and have a base to run back to in the event of a disaster. The findings of this research are also very useful to all the commercial banks, examination bodies and small & medium scale businesses all over the globe.

References

Bressoud, T. C., & Schneider, F. B. (1996). Hypervisor-based fault tolerance. *ACM Transactions on Computer Systems (TOCS), 14*(1), 80-107.

Buyya, R., & Son, J. (2018). Software-Defined Multi-Cloud Computing: A Vision, Architectural Elements, and Future Directions. *arXiv preprint arXiv:1805.10780*.

Caraman, M. C., Moraru, S. A., Dan, S., & Kristaly, D. M. (2009). ROMULUS: DISASTER TOLERANT SYSTEM BASED ON KERNEL VIRTUAL MACHINES. *Annals of DAAAM & Proceedings*.

Chang, F. W., Ji, M., Leung, S.-T., MacCormick, J., Perl, S. E., & Zhang, L. (2002).

Myriad: Cost-Effective Disaster Tolerance. Paper presented at the Conference on File and Storage Technologies (FAST), California, USA.

Chen, H., & Zheng, Z. (2008). *Design and implementation of XML-based GUI for cross platform backup system.* Paper presented at the 2008 International Symposium on Computer Science and Computational Technology.

Cully, B., Lefebvre, G., Meyer, D., Feeley, M., Hutchinson, N., & Warfield, A. (2008). *Remus: High availability via asynchronous virtual machine replication.* Paper presented at the Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation.

Dhane, S. V., & Joshi, P. (2015). Public Auditing System with Auto-Data Recovery System on Cloud Scheme. *International Journal of Science and Research (IJSR), 4*(11).

Jangra, A., & Bala, R. (2012). A Survey on various possible vulnerabilities and attacks in cloud computing environment. *International Journal of Computing and Business Research, 3*(1), 1-13.

Kivity, A., Kamay, Y., Laor, D., Lublin, U., & Liguori, A. (2007). *kvm: the Linux virtual machine monitor*. Paper presented at the Proceedings of the Linux symposium.

Mayur, S. A., & Vani, N. (2016). Server Virtualization using Cloud Environment for Data Storage &Backup *International Journal of Science and Research (IJSR), 5*(6).

Paschal, O. (2016). Backup as a Service from MTN – Cloud Storage Packages. Retrievedfrom https://www.naijatechguide.com/2015/08/backup-as-service-from-mtn-cloud.html

Rachana, S., & Guruprasad, H. (2014). Emerging Security Issues and challenges in cloud computing. *International Journal of Engineering Science and Innovative Technology, 3*(2), 485-490.

Rajagopalan, S., Cully, B., O'Connor, R., & Warfield, A. (2012). SecondSite: disaster tolerance as a service. *Acm Sigplan Notices, 47*(7), 97-108.

Stergiou, C., Psannis, K. E., Kim, B.-G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems, 78*, 964-975.

Tamura, Y., Sato, K., Kihara, S., & Moriai, S. (2008). *Kemari: Virtual machine synchronization for fault tolerance.*Paper presented at the Proc. USENIX Annu. Tech. Conf.(Poster Session).

Usman, K., & Inyiama, H. C. (2019). Development of User-Centric Cyber Disaster Recovery System. *Journal of Scientific and Engineering Research, 6*(2), 137-143.

Westerlund, M., & Kratzke, N. (2018). *Towards Distributed Clouds: A Review About the Evolution of Centralized Cloud Computing, Distributed Ledger Technologies, and A Foresight on Unifying Opportunities and Security Implications.* Paper presented at the 2018 International Conference on High Performance Computing & Simulation (HPCS).

Wood, T., Cecchet, E., Ramakrishnan, K. K., Shenoy, P. J., van der Merwe, J. E., & Venkataramani, A. (2010). Disaster Recovery as a Cloud Service: Economic Benefits & Deployment Challenges. *HotCloud, 10*, 8-15