ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Efficient Load Balancing and Disk Failure Avoidance Approach Using Restful Web Services

Neha Shiraz, Dr. Parikshit N. Mahalle

Persuing M.E, Department of Computer Engineering, Smt. Kashibai Navale College of Engg., Pune,

Savitribai Phule Pune University, Maharashtra, India.

PhD, Department of Computer Engineering, Smt. Kashibai Navale College of Engg., Pune,

Savitribai Phule Pune University, Maharashtra, India

ABSTRACT: RESTful interfaces are mainly used for implementation of web services and are based on the resource-oriented approach. REST proposes the use of uniform and predefined set of stateless operations to exchange heterogeneous resource representations. RESTful Web services provide interoperability between systems on the Internet. Stateless protocols and standard operations aids in improving system's performance and reliability. In this research work, Restful services are used for data storage and retrieval from Cloud system. Cloud storage generally provides different redundancy configuration to users in order to maintain the desired balance between performance and fault tolerance. Node failure is very common in cloud system which affects data availability. So, we have used Cauchy Coding technique for regeneration of codes as a recovery solution. It finds an optimal coding scheme within the capability of the current state of the art, for an arbitrary given redundancy configuration using restful web services. Also, this approach has reduced the amount of data used for recovery to almost half and it provides load balancing and also maintains a secure access control mechanism for authenticated user.

KEYWORDS: REST, HTTP, Web Services, load balance, XOR scheduling, Cauchy Coding.

I. INTRODUCTION

Cloud storage is built up of numerous inexpensive and unreliable components, which leads to a decrease in the overall mean time between failures. As storage systems grow in scale and are deployed over wider networks, component failures have been more common, and requirements for fault tolerance have been further increased. A cloud system consists of several elements such as clients, datacenter and distributed servers. It has characteristics such as fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Effective load balancing algorithm is required in order to cope up with these issues. There can be different types of loads such as CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various processors to improve resource utilization and the throughput time while also avoiding a situation where some nodes are heavily loaded while other nodes have very less load or are working scarcely. All the processors in the system or every node in the network does approximately equal amount of work at any point of time when load balancing is applied to the system. Load balancing is a pre-required service for increasing the performance and maximum utilization of the resources. Load balancing is the process of increasing system performance in the situations of heavy load. This process of removing the situation in which some of the nodes are overloaded while some others are under loaded. This phenomenon can drastically reduce the working efficiency.
This system performs recovery of data in case of disk failures using Cauchy matrix heuristics. First, it uses Cauchy matrix heuristics to produce a matrix set. Second, for each matrix in this set, it uses XOR schedule heuristics to generate a series of schedules. Finally, it selects the shortest one from all the produced schedules. In such a way, it has

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

the ability to identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration using restful web services
Some of the major goals of load balancing algorithms:
a) Cost effectiveness and low energy consumption with improvement in system performance at a reasonable cost.
b) The distributed system in which the algorithm is implemented may change in size or topology. Hence, scalable and flexible algorithms should be used to allow such changes to be handled easily. But load balancing is critical to serve requests cost effectively over cloud. So, to overcome limitations of balancing algorithm, we propose to implement re-balancing algorithm .Before serving requests over cloud, it will calculate performance and load on individual resources. Based on this calculated performance and current load, requests will be served by those specific resources over cloud.

II. LITERATURE SURVEY

Zhang et al. [1] proposed CaCo, an efficient Cauchy coding approach for data storage in the cloud. Firstly, CaCo uses Cauchy matrix heuristics in order to produce a matrix set. Second, for each matrix in this set, CaCo uses XOR schedule heuristics so that it can generate a series of schedules. In second phase, CaCo selects the shortest one from all the produced schedules. In such a way, CaCo can identify an optimal coding scheme, within the capability of the current state of the art, for an arbitrary given redundancy configuration. It also implements CaCo in the Hadoop distributed file system and evaluates its performance by comparing with "Hadoop 2.5". Finally, author proposed this system enhance the security in distributed file system with effective data storage scheme.
Trifonov [2] proposed approach is based on the cyclotomic fast Fourier transform algorithm and enables one to significantly reduce the number of expensive Galois field multiplications required. Appropriate storage techniques are required as huge amount of data is available in digital formats. In most of the cases, the amount of data to be stored is more than the capacity of a single disk drive. Also, for a particular application more than one reliable storage drives are required. These circumstances motivate the development of redundant arrays of independent disks (RAIDs). RAID designs use some kind of maximum distance separable (MDS) code. Reed-Solomon (RS) codes are an old and well studied class of such codes. However, the complexity of such algorithms is too high for practical usage.

Li et al [3] proposed generation of optimal Cauchy matrix which can lead to an efficient CRS coding scheme. Many applications such as distributed storage and fault tolerance system used one class of maximum distance separable (MDS) codes; Reed Solomon (RS) codes are proved to be very efficient. The best performing implementations of RS codes till date employ a variant called Cauchy Reed-Solomon codes (also known as Srivastava codes). In CRS coding schemes over some Galois field F =GF (2ω), ω ∈ Z+, the addition operation is bit-XOR and the multiplication operation is with respect to polynomials over GF (2), encoder complexity can be improved by transforming multiplication operation over F to bit-XOR Main problem here is the construction of a Cauchy matrix over the field F such that the corresponding CRS coding operation is as efficient as possible.

Gruner, S et al [4] proposed the idea of making OPC UA server restful by making few extensions to its protocol stack. Features of rest include Addressability, Statelessness, Connectedness, and Uniform Interface. Making small changes to OPC UA stack has several advantages like communication advantages, reduced communication overhead. This system also proposes introduction of caching layer for load balancing. Basically, this system used UDP protocol as a communication protocol in place of TCP to reduce communication overhead as there are no acknowledgements in UDP and also it is stateless.

Luo et al [5] this research work describes semantic web services as adding semantics to the descriptions of Web Services can automate the processes of discovery, selection and composition of services. Restful services uses unstructured HTML, therefore automation of such services is very difficult. Although many annotating models are proposed to support this automation, the adoption of these models is difficult due to the tedious manual annotation process. In order to solve this problem, this system proposed ASSARS, namely Automated Structural Semantic Annotation for RESTful Services, to automatically transform the unstructured RESTful service pages into structured RESTful service descriptions. In proposed method, there are two key steps: (a) The semantic block division of RESTful

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

service pages, and (b) the identification of the semantics of blocks which significantly describe the RESTful services from different aspects.

III. PROBLEM STATEMENT

Proposed system has analyzed the issues of data storage and retrieval in cloud system such as space complexity, time complexity, Node failure, CPU usage, unauthenticated data access in public cloud and has provided some optimal solutions for increasing the performance of the system and recovering data in case of loss of data.

IV. PROPOSED SYSTEM

In the proposed research work need to implement below objectives

- In this work system use (k,m,w) for matrix generation approach. Given a redundancy configuration (k,m,w) system goal is to find a Cauchy matrix, whose schedule is desired to be the shortest.
- In this work first find and analyze the drawbacks of CaCo system with strategic approach, after finding all the issue then design and implement a system that can eliminate such problem of existing system. The research contribution will provide better performance of system.
- The proposed system can maintain load balancing for data distribution and for data encryption used DES with MD5 encryption scheme. The system can generate caco matrix for each chunk using XOR operation and efficiently store the data into data node.

- PROPOSED SYSTEM ARCHITECTURE

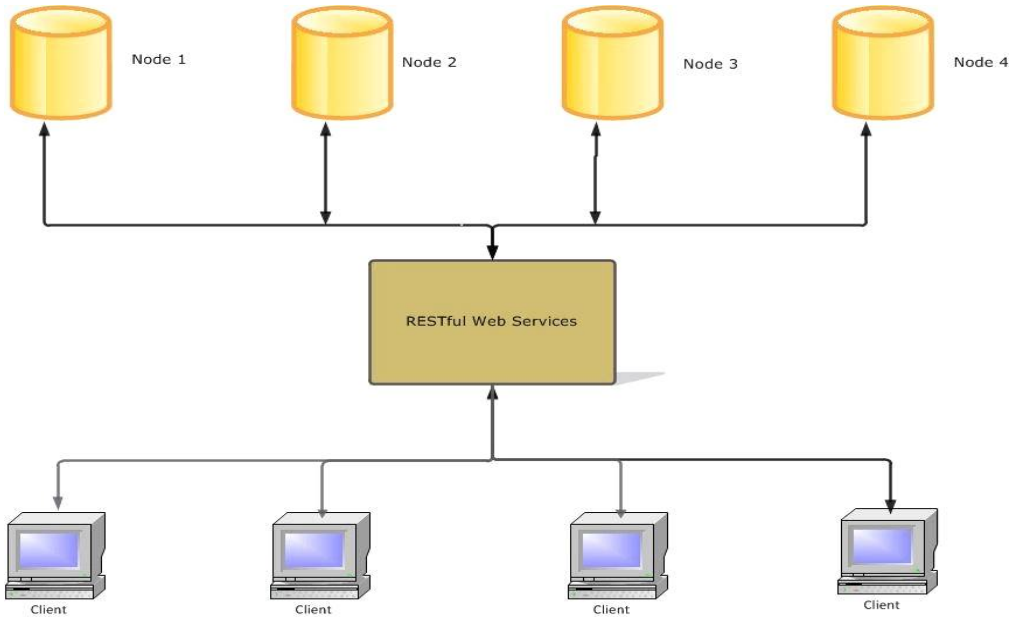


Figure 1: Proposed System Architecture

The CaCo model for cloud data hierarchy analysis and monitoring is shown in fig.1 and the system consist of a web service server which performs all the communication. All the requests to the databases are routed through web service server. Client requests are sent to the server which redirects them to the appropriate node. CaCo matrix is stored in data node and provides data in case of node failure. The system proposed in this architecture consists of a semi-automated

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

stack for data movements and address tracking. This system majorly focuses on data waiting time reduction and performance enhancement.

V. ALGORITHMS

Algorithm1: Load Balancer Algorithm
Input: Text files with data
Output: File operation with server load balancing
Step 1: Initialize server and its sub-servers
Step 2: Establish connection between sub-server and servers using the IP or Port number.
Step 3: Upload File to server that should be shared.
Step 4: Server encrypts data with MD5 Encryption.
Step 5: Split the file into multiple chunks
Step 6: Calculate each sub server memory
Step 7: Divide the total chunks value by total number of sub-servers
Step 8: Upload each chunk into sub servers based on its memory capacity
Step 9: If Capacity is less then transfer the excess chunks into next sub-servers
Step 10: Each chunk will be appended with an index value.
Step 11: When the client request for a file, that will be received from different sub-servers based on the index value.
Step 12: Client collects all the chunks then the file will be decrypted, then that will be viewed by client.

Algorithm2: Equally Load Re-Balancer Algorithm
Input: Each Node load base on current hitting
Output: Distributed data to each node.
Step 1: Initialize all data nodes which are connected to master node as n.
Step 2: for each (1 up to n)
Take each ith node server load.
A[i] => ith Node load degree or hitting load.
End for.
Step 3: get total length of A. create the data requested chunks.
K=A. length ();
Step 4: Generate k mappers for distribute a data.
Step 5: Assign each chunks to each mapper.
Step 6: Request to server for saving data.
Step 7: end procedure.

Algorithm3: Caco Matrix Generation
Input: Data block d
Output: Cauchy matrix as X
Step 1: Constructing the matrix ONES. First, CaCo constructs a matrix named ONES, whose element (i,j) is defined as the number of ones contained in the binary matrix M(1/i+j).
Step 2: Choosing the minimal element. Second, CaCo chooses the minimal element from the matrix ONES. Supposing the element is (x1,y1), we initialize X to be {x1} and Y to be Y1.
Step 3: Determining the set Y. Besides the element (x1,y1) CaCo chooses the top k-1 minimums from row x1. Then, CaCo adds the corresponding k-1 column numbers to Y , and we have Y={y1,y2,y3....yn}.
Step 4: Finally generate matrix as X for each row each X instance is Cauchy matrix of given data block.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

- MATHEMATICAL MODEL

Inputs:
The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth.
Process:-
1. Define a load parameter set: F= {F1,F2…Fm} with each Fire presents the total number of the parameters.
2.Compute the load degree as
Load Degree(N)= Σ αiFi
Where i= 1…m
3. Average cloud partition degree from the node load degree statistics as:
Load degree avg= $\Sigma^{n}_{i=1}$ Load Degree(Ni)
4. Three level node status are defined
Load degree(N)=0 for ***Idle***
0<Load Degree(N)<Load Degree(N)high for ***Normal***
Load_Degree(N)high <= Load_Degree(N) for ***Overloaded***
Generate matrix for each data block
M[]=Cauchy(D[i])
Output:-
Data store into multiple data node with Cauchy matrix generation.
=Failures and Success conditions.
Failures:

- Huge database can lead to more time consumption to get the information.
- Hardware failure.
- Software failure.

Success:

- Data node connection failure.
- User gets result very fast according to their needs.

- IMPLEMENTATION DETAILS

1. **System interfaces**: Windows Operating System
2. **User interfaces**: User interface using Jsp and Servlet
3. **Hardware interfaces:**
   Processor: Intel Pentium 4 or above
   Memory: 512 MB or above
   Hard Disk: 40GB
4. **Software interfaces**:
Front End:Jdk 1.7.0 and Eclipse Luna
Back-End: Visual Studio
5. **Database:** MySQL 5.0

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

VI. RESULTS AND DISCUSSION

Below table 1 shows storing data in chunks on different servers

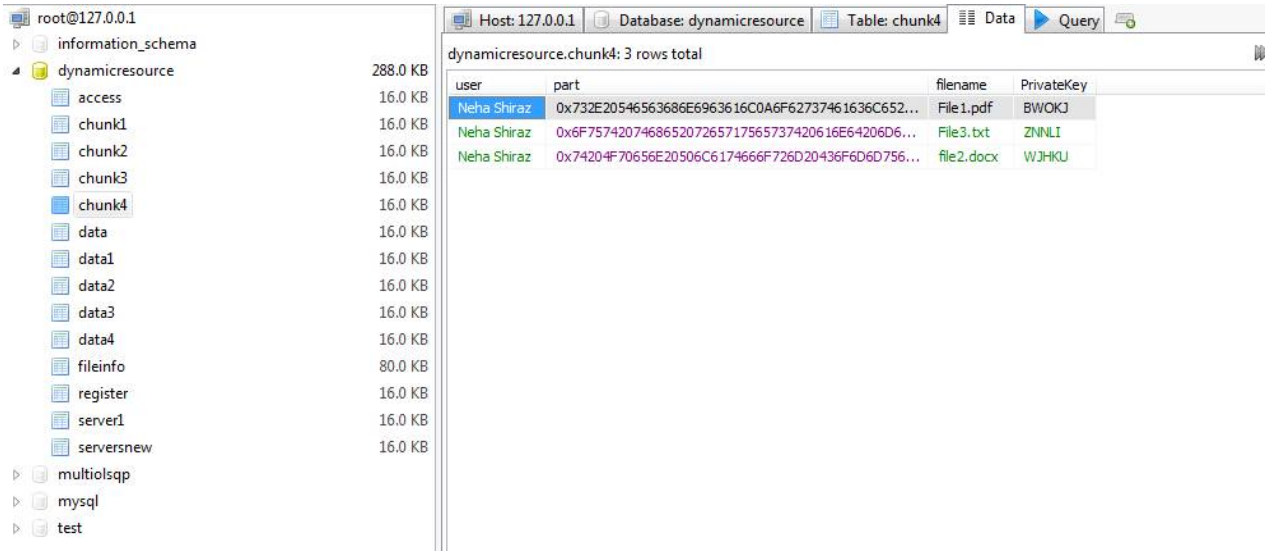
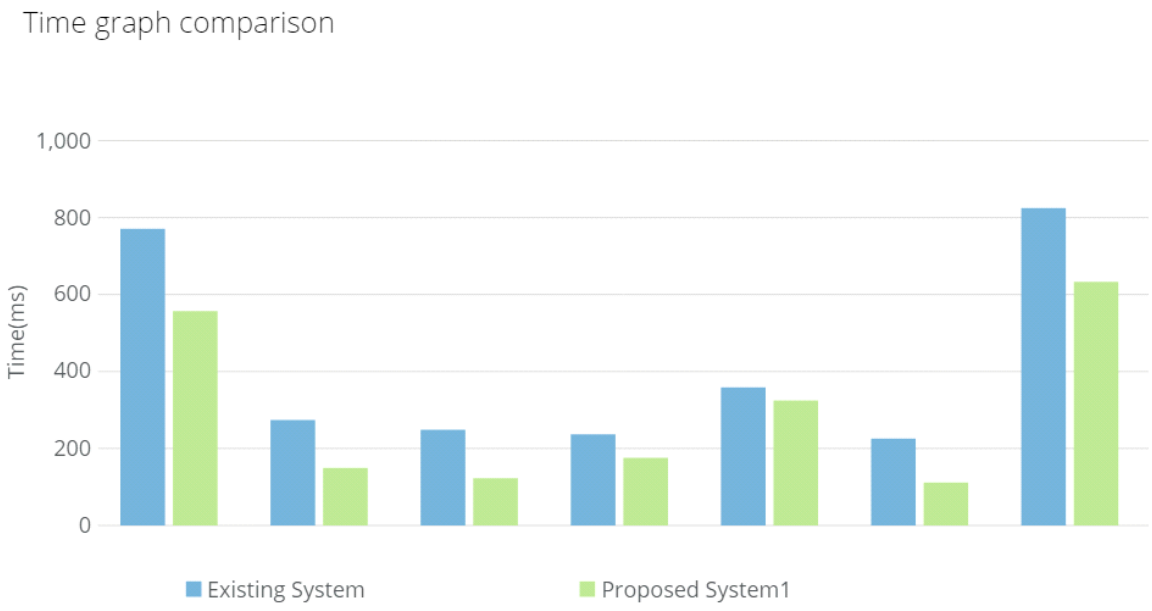


Table 1

Below Graph 1 shows improvement in data transfer time while file upload.



Graph 1

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

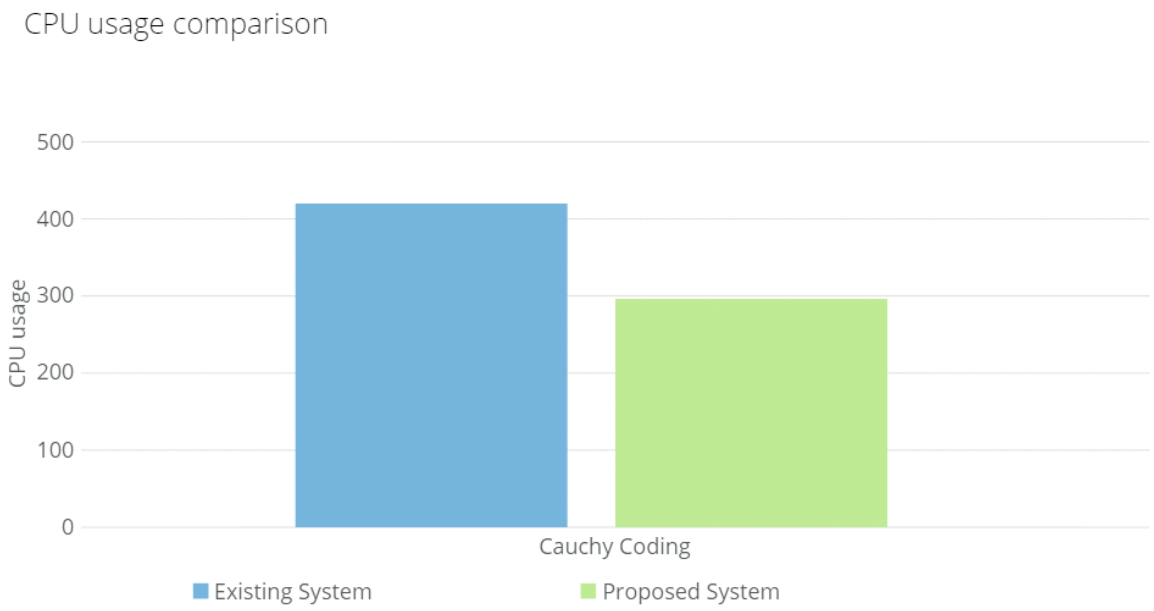
Below table 2 and graph 2 show improvement in CPU usage.

Host: 127.0.0.1 | Database: dynamicresource | Table: serversnew | Data | Query

dynamicresource.serversnew: 4 rows total | Next

IpAdd	ServerName	CpuUsage
192.168.0.1	Server1	45.10
192.168.0.2	Server2	41.10
192.168.0.3	Server3	39.20
192.168.0.4	Server4	57.48

Table 2



Graph 2

- Future Work

For the future enhancement we can consider as load rebalancing in hybrid cloud environment with hadoop. The thermal management and energy saving approach with resource virtualization is another interesting area for such concepts. Cauchy's rule can be incubated with bandwidth of performance with respect to time for series computation of delay in network node retiring. The delay time reduction under narrow network is still a bottle neck situation. Either of this can be improvised with technical reduction of data sets and its indexing.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

VII. CONCLUSION

In the proposed work, restful web services are used for communication and system focus on CaCo, an approach that incorporates all existing matrix and schedule heuristics, and thus is able to identify an optimal coding scheme within the capability of the current state of the art for a given redundancy configuration. The selection process of CaCo has an acceptable complexity and can be accelerated by parallel computing. It should also be noticed that the selection process is once for all. The experimental results demonstrate that use of restful web services for all communications have increased the performance by reducing the data transfer time, load balancing has made the efficient use of processors, matrix generated using Cauchy coding has helped to recover data in case of node failure.

REFERENCES

[1] Zhang, G., Wu, G., Wang, S., Shu, J., Zheng, W. and Li, K. (2016). CaCo: An Efficient Cauchy Coding Approach for Cloud Storage Systems. *IEEE Transactions on Computers*, 65(2), pp.435-447.
[2] Trifonov, P. (2015). Low-Complexity Implementation of RAID Based on Reed-Solomon Codes. *ACM Transactions on Storage*, 11(1), pp.1-25.
[3] Li, X., Zheng, Q., Qian, H., Zheng, D. and Li, J. (2009). Toward optimizing cauchy matrix for cauchy reed-solomon code. *IEEE Communications Letters*, 13(8), pp.603-605.
[4] Gruner, S., Pfrommer, J. and Palm, F. (2016). RESTful Industrial Communication With OPC UA. *IEEE Transactions on Industrial Informatics*, 12(5), pp.1832-1841.
[5] Luo, C., Zheng, Z., Wu, X., Yang, F. and Zhao, Y. (2016). Automated structural semantic annotation for RESTful services. *International Journal of Web and Grid Services*, 12(1), p.26.
[6] Zhao, X., Liu, E., Yu, H. and Clapworthy, G. (2015). A linear logic approach to the composition of RESTful web services. *International Journal of Web Engineering and Technology*, 10(3), p.245.
[7] Load Balancing In Cloud Computing. (2017). *International Journal of Recent Trends in Engineering and Research*, 3(3), pp.260-267.
[8] Microsoft Windows Azure: Developing Applications for Highly Available Storage of Cloud Service. (2015). *International Journal of Science and Research (IJSR)*, 4(12), pp.662-665.