IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

A Review on Load Balancing Algorithm in Cloud Computing Using Restful Web Services

Tusha Agarwal[1*], Abhishek Saxena[2]

[1] Department of Computer Science, Noida International University, Noida, India
[2] School of Engg. & Tech., Noida International University, Noida, India

Available online at: www.ijcseonline.org

Accepted: 19/July/2018, Published: 31/July/2018

Abstract- As the IT industry is growing day by day, the need of computing and storage is increasing rapidly. The amount of data exchanged over the network is constantly increasing. These services return the required information in form of text, images, video etc. In Cloud computing, Load Balancing is required in such situations to avoid overload. A load balancer technique mediates client access requests to servers and intelligently decides which server is best placed to fulfil each request. Restful interfaces are mainly used for implementation of web services and are based on the resource-oriented approach. This paper discusses the some existing load balancing algorithms in cloud computing. In this review paper, Restful services are used for data storage and retrieval from Cloud system. Cloud is a storage mechanism in which one can store, process data on demand. Load balancing approach is used to distribute load dynamically among the all the nodes in the cloud. Load balancing avoids the occurrence of such a situation where some virtual machines are heavily loaded or in ideal situation**.** This approach has reduced the amount of data used for recovery to almost half and also maintains a secure access control mechanism for authenticated user.

Keywords: REST, HTTP, Load Balance, XOR scheduling,Web Services.

I. INTRODUCTION

Cloud computing is the combination of many pre-existing technologies that have matured at different rates and in different contexts. The goal of cloud computing is to allow users to take benefit from all these technologies. Many organizations are moving into cloud because it allows the users to store their data on clouds and can access at anytime from anywhere. Data breaching is possible in cloud environment, since data from various users and business organizations lie together in cloud.

Cloud storage is one of the primary use of cloud computing. We can define cloud storage as storage of the data online in the cloud. A cloud storage system is considered as a distributed data centres, which typically use cloud-computing technologies and offers some kind of interface for storing and accessing data. A cloud system consists of several elements such as clients, data-centre 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, and delay or network load [8].

Cloud computing is a distributed computing system that focuses on providing a wide range of users with distributed access to scalable, virtualized hardware and/or software infrastructure over the internet[10]. Cloud computing methodology completely changes the concept of parallel and distributed computing. It provide a very easy solution to all IT resources.. This is all suggests that cloud computing will change the way we interact with the resources via Internet. Cloud models use virtualization technology. It depends on the hardware configuration of the data centre or server in how may virtual machine they can be divided. Load balancing is the pre-requirements for increasing the cloud performance and for completely utilizing the resources [9].

Cloud computing architecture can be divided into two sections: front end and back end. They both are connected with each other through a network, usually the internet. The front end is what the user (client) sees whereas the back end is the cloud of the system. Front end has client's computer and the application required to access the cloud and the back end has the cloud computing services like various computers, servers and data storage. 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 underloaded. This phenomenon can drastically reduce the working efficiency [3].

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 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.

II. REST

REST (Representational State Transfer) is an architectural style for developing web services. REST is popular due to its simplicity and the fact that it builds upon existing systems and features of the internet's HTTP in order to achieve its objectives, as opposed to creating new standards, frameworks and technologies. REST compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. The term representational state transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. Fielding used REST to design HTTP 1.1 and Uniform Resource Identifiers (URI).

The principles of REST include:

1. Conceptual entities and functionalities are modelled as resources identified by universal resource identifiers (URIs).
2. Resources accessed and manipulated via standardized, well-known HTTP operations (GET, POST, PUT and DELETE).
3. Components of the system communicate via these standard interface operations and exchange the representations of these resources (one resource may have multiple representations).

In REST system, servers and clients typically travel through different states of resource representations by following the interlinks between resources.

By applying the principles of REST Web service (WS), development, Restful WSs are emerging as the choice for many of the leading Internet companies to expose their internal data and functionalities as URI identified resources. In contrast to the operation-centric perspective of WSDL/SOAP WSs, Restful WSs view the applications from a resource-centric perspective.

III. HTTP

HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. HTTP is the protocol to exchange or transfer hypertext. HTTP functions as a request response protocol in the client server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as

HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

IV. WEB SERVICES

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL). Other systems interact with the webservice in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards. In practice, the Web service typically provides an object-oriented Web-based interface to a database server, utilized for example by another Web server, or by a mobile application, that provides a user interface to the end user. Another common application offered to the end user may be a mash-up, where a Web server consumes several Web services at different machines, and compiles the content into one user interface.

V. LOAD BALANCE

Load balancers are used to increase capacity (concurrent users) and reliability of applications. They improve the overall performance of applications by decreasing the burden on servers associated with managing and maintaining application and network sessions, as well as by performing application-specific tasks.

A load balancer sits between the client and the server farm accepting incoming network and application traffic and distributing the traffic across multiple back end servers using various methods. By balancing application requests across multiple servers, a load balancer reduces individual server load and prevents any one application server from becoming a single point of failure, thus improving overall application availability and responsiveness.

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.

Different load balancing algorithms provide different benefits,the choice of load balancing method depends on your needs:

- Round Robin–Requests are distributed across the group of servers sequentially.
- Least Connections–A new request is sent to the server with the fewest current connections to clients. The relative computing capacity of each server is factored into determining which one has the least connections.
- IP Hash–The IP address of the client is used to determine which server receives the request.

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 a 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.

VI. XOR SCHEDULING

XOR-based erasure codes to optimize their use of cache memory. We call the technique XOR-scheduling and demonstrate how it applies to a wide variety of existing erasure codes. We conduct a performance evaluation of scheduling these codes on a variety of processors and show that XOR-scheduling significantly improves upon the traditional approach. Hence, we believe that XOR-scheduling has great potential to have wide impact in practical storage systems.

The traditional XOR-scheduling algorithm follows the intuitive idea that coding words should be produced one by one. Instead, we can reorder the schedule so that it consumes data words one by one. Our new XOR-scheduling algorithm is based on this idea, and its characteristics are as follows:

1.The order of XOR operations is guided by the order of data words instead of coding words.

2. Each data word is used for all of its coding calculations before moving onto the next data word in the same packet.

Caco Approach for cloud data :-

The CaCo model is used for cloud data hierarchy analysis and monitoring. 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
stack for data movements and address tracking. This system majorly focuses on data waiting time reduction and performance enhancement.

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.

VII. CONCLUSION

Cloud computing is now becoming a business standard. It simplies the users accessibility Cloud computing is an emerging field of information technology (IT). It enables a wide range of users to access distributed, scalable, virtualized, hardware and/or software infrastructure over the Internet.Load balancing is one of the leading issue of
cloud computing. So there is need for a well ordered load balancing algorithm for efficient utilization of resources.

In this review paper, 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.

This paper is an earnest effort to unveil the concept of load balancing and its type's, especially dynamic load balancing. But, still there are miles to go. As cloud computing is yet in its infancy and there are many more open issues like resource utilization,security etc.

REFERENCES

[1].Digvijaysinh M. Rathod; M S Dahiya; Satyen M. Parikh, (2015) Towards composition of RESTful web services, 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp 1 6.

[2]Gaj, P., Jasperneite, J. and Felser, M. (2013). Computer Communication Within Industrial Distributed Environmental Survey. IEEE Transactions on Industrial Informatics, 9(1), pp.182-189.

[3]DU, Y., GUO, T. and CHEN, J. (2013). Fleet elastic load balancing mechanism in cloud environment. Journal of Computer Applications, 33(3), pp.830-833.

[4]XIE, H. and CHEN, X. (2013). Cloud storage-oriented unstructured data storage. Journal of Computer Applications, 32(6), pp.1924-1928.

[5]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.

[6]Trifonov, P. (2015)." Low-Complexity Implementation of RAID Based on Reed-Solomon Codes." *ACM Transactions on Storage*, 11(1), pp.1-25.

[7]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.

[8]Gruner, S., Pfrommer, J. and Palm, F. (2016). RESTful Industrial Communication With OPC UA. *IEEE Transactions on Industrial Informatics*, 12(5), pp.1832-1841.

[9]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.

[9] A Review of Load Balancing in cloud Computing by Gunpriya Makkar and Pankaj Deep Kaur.,*IJARCSSE volume 5 Issue 4, ISSN:2277128X*

[10] Load balancing in Cloud Computing : A Review by shiny, I*OSR journal of Computer Engineering e-ISSN:22780661.*

[12]Microsoft Windows Azure: Developing Applications for Highly Avail-able Storage of Cloud Service. (2015). International Journal of Science and Research (IJSR), 4(12), pp.662-665.

[13]Jabbar, S., Khan, M., Silva, B. and Han, K. (2016). A REST-based industrial web of things framework for smart warehousing. The Journal of Supercomputing, pp., 2016, ISSN 0920-8542.

[14]Henen, R. and Schleipen, M. (2014). Interoperability between OPC UA and AutomationML. Procedia CIRP, 25, pp.297-304.

[15]L. M. S. De Souza, P. Spiess, D. Guinard, M. Kohler, S. Karnouskos, and D. Savio, (2013). Socrades: A web service based shop oor integration infrastructure, in The internet of things, Spriger, pp. 5067.

