ISSN 2395-1621

A Step Towards Load Balancing in Client-Server Communication

Rucha R. Goundadkar [#1], Roshani K. Nakhate [#2], Sonali K. Koratkar [#3]

[1]ruchagoundadkar13@gmail.com

[2]nakhate.roshani@gmail.com

[3]koratkarsonali@gmail.com

[#1234] Department of Computer Engineering

Bhivarabai Sawant College of Engineering and Research (BSCOER)

Pune, Maharashtra, India.

ABSTRACT

Performance optimization of distributed system in internet is of immense interest in research community. As the number of internet users in distributed system increase the communication between client and server. The server actually performs the communication, so in the distributed system server which is having highest load, its time is considered as the total time of performance of that system. This measures the time of the slowest server which makes complicated Client-server assignment protocol. To reduce this our proposed system provides a way to optimize the request coming towards the server from client end by performing inter server communication and then load will be calculated to distribute among all the servers to achieve equal time of performance by all servers. This will increase the performance of the distributed internet system.

ARTICLE INFO

Article History

Received 26th January, 2015

Received in revised form

2nd February, 2015

Accepted: 4th February, 2015

Published online :

11th February 2015

Keywords— ***Distributed Internet System(DIS), Load balancing, performance optimization.***

I. INTRODUCTION

Numbers of internet users in distributed system are increasing day by day which results in increased communication load on the server in DIS. The clients and servers form a distributed system. The proposed system tends to assign almost equal load on each server in DIS so as to optimize the performance, this will enhance the user's experience in accessing the internet. If any single server is having the maximum load on it as compared to other servers, it degrades the performance of whole system, because time of slowest system is considered as the total processing tome of that DIS. Our proposed system performs the optimization by considering the incoming requests to the servers from clients distributed internet system consists of large number of clients. The addition of all such requests will be done, then equal number of requests will be assigned to each server in the DIS by performing interserver communication, it doesn't matter how many number of requests it was having before load balancing. This increases the performance of the DIS by assigning equal amount load on each server in the system after every specific period of time. A denial of service (DOS) system stops responding to the clients if number of requests exceed than the threshold point. Hence we are going to implement interserver communication in order to achieve the optimized client server assignment.

There are various existing systems which are used to optimize the performance of the client- server system, an optimization is achieved by focusing in message length[2] and the memory required to process the respective request. For doing this they check the bandwidth of the message and I/O buses[1] that are used in the system our system does not focus on these things as it tends to complexity in optimization problem. Another approach used is 'proxy server', when the communication overhead increased on particular server in system which may cause failure of that system, in such a situation another server is added and it will work as proxy server. Hence adding proxy server unnecessarily increases the cost for adding new hardware.

Optimization is also done in virtual environments[7] where the cost of virtual environment is a big issue. Achieving optimized performance of such a system is very critical job in virtual environments. Any distributed system or virtual system is represented in the form of graphs and partitioned into two or more groups for making the request distribution i.e. optimization easier. But it is important to grouping of such a system evenly based on the communication load on the servers, not by focusing on group size[9][10], which is also plays an important role in optimization.

Our proposed system divides the number of servers into two groups and then does the balancing using interserver

communication technique. We are dividing almost equal number of requests to each server in the system by calculating the total number of requests that are came to each server before the optimization. Unlike the existing systems, our proposed system simplifies the mathematical computations easy. This optimization will be performed after each interval of time. The data recovery in distributed system is a challenging process in case if loss of data. Hence to avoid such a situation our proposed system provides DOS (Denial Of Service) facility, which will prevent the access to the client when request on the servers reach to maximum value request, for specific period of time it will prevent the system from failure.

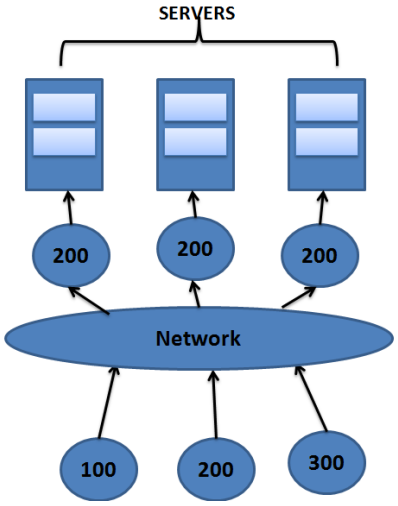

Fig. 1.

Above diagram shows how the load from different clients will be summed and distributed almost equally to each server in the DIS. So the performance degradation due to the overloaded and slowest server in the system can be avoided. Next segment of this paper contains the Literature survey of the existing systems related to our problem.

II. Related Work

There are various existing techniques in the area of performance optimization in distributed internet system.

Krishna Kant [1] proposed one method for planning server's capacity for minimizing workload on a server in a network. He tries to minimize the memory required using various I/O buses and with the help of proxy servers, this increases the cost as well as complexity in minimization of work load of the server. D. Saritha [2] proposed a dynamic method , in this method they check the load on the server, message length and the number of available servers in the system. To do this they require message bandwidth, which is not feasible with our problem scenario.

P. Morillo [3] suggests a method for improving the performance of the distributed virtual machine environment, he partitions the virtual distributed environment system in very low cost for achieving the optimization in client server system but its result shows absence of correlation and the average response remains practically invariable until the system reaches its threshold point. The system that Jeffery Dean [5] has proposed runs on large cluster of commodity systems and they are scalable. This system is easy to use even though the user is not at all having the deep knowledge of it. This system restricts programming model to make it easy and to parallelize and distribute computations to make the system fault tolerant.

Jianbo Shi [6] proposed Normalized Cuts algorithm (NC) to partition the data into two or more groups. It represents the client server system in the form of graphs, where nodes represent the servers and load on the server is represented by the edges with weight. Partition is done according to value of Fncut which is roughly represented as Fc*Fl, it is the multiplication if communication load and its cost. This is mainly suitable for image segmentation and bioinformatics.

For optimizing performance optimization of distributed virtual environment two phase approach is used as suggested by Duong Nguyen [7] , these two phases are initial assignment phase and refined assignment phase. This approach works effectively with small size DVEs when the size of DVE increases then it is unable show good performance optimization. Kernel Kmeans is an algorithm proposed by Indrejit S. Dhillon [8] and it does spectral clustering along with the NC cut algorithm. The objective of this algorithm is recast as a trace maximization and minimization; they developed an EM-system which can be used to solve Kernel Kmeans problem.

Indrejit S. Dhillon also proposed another algorithm called as Graclus [9], this algorithm minimizes the time consumption for dividing the graph into two separate clusters, hence we can to the optimization within minimum time. Graclus works faster than NC algorithm. This algorithm does the partitioning of graph according to the size of the group, in our problem our main focus is on the load on the particular server not on the size of group of servers, hence this algorithm can't be direcly applied to our problem.

Kevin Lang [10] has proposed an algorithm which performs the clustering of group of servers. This algorithm randomizes the flow based rounding method and hence gives better results. But similar to Graclus it does the clustering according to size of the cluster and this is not the case with our problem, that's why it can't be direcly applied to our problem even if it is showing better performance in clustering.

There is an algorithm proposed by Zhenyu Wu [11] which partitions the large system into groups by the values associated with the edges in the graph, unlike Graclus, those values on edges in the graph represent the load on that server. This algorithm also tries to include edges in the calculation those who do not have strong connection link between them which causes the imbalance of load among the groups. Inderjit S. Dhillon [12] proposes an algorithm which is mathematically equivalent to popular existing algorithms such as NC algorithm and Kernel K-means algorithm. By using this equivalency, he proposed a system which does clustering of weighted graphs efficiently and faster without using eigenvector calculations. So it reduces computation complexities. But the refinement phase of Inderjit S. Dhillon's algorithm contains the implementation of kernel k-means algorithm.

Hiroshi Nishida [4] proposes an algorithm which effectively does the client server optimization which gives better performance than various existing techniques also including the popular algorithm which is mainly used for optimization i.e. NC algorithm. It calculates the total load on each server in the system and does the optimization by distributing almost equal load on the each server in the system. This algorithm includes very large computational calculations for optimization which are complex and may consume time.

III. PROPOSED METHODOLOGY

CDD (Collect divide and distribute) Algorithm :-

The CDD algorithm collects the requests on each server in the system, divide them almost equally and then we distribute them to the servers for achieving the optimized performance of the distributed client server system. We know that clients requests may unequally assigned to the servers, hence speed of that system is considered as the speed of slowest server in that system. Hence for optimization our proposed algorithm includes following steps:

1) One of the servers in the system will act as admin server which will communicate with all other servers. The admin server will calculate the total number of requests to all servers.
2) The admin server gets the other servers status, if there is a communication between them then it is represented by 1 otherwise it is represented by 0.
3) It makes status vector according to the status of other servers in the system. This status vector is achieved by interserver communication only. The purpose of status vector is to identify which request has came from which specific client. So that the response can be forwarded to correct client.
4) The requests are divided by total number of servers in the system, suppose the answer is shown by the variable R.
5) R number of requests are distributed to all servers in the system to attain equilibrium. Hence the balance is distributed equally to the servers.
6) In such a case where number of requests exceed on the server more than its capacity, hence to avoid the failure of server and loss of data.
7) Get the number of source IP and store it as Ns. Set a value Ts as threshold point. If number Ns>Ts, then DOS attack detected, stop responding to client.

The last step of above algorithm is used for implementing DOS system by checking Ns which is a threshold value that has been set. All the above computations will be done matrix from where we will be maintaining three matrices. One matrix will represent the client to client communication. The numbers of messages that are exchanged between particular clients are represented in matrix form. Another matrix will contain client to server communication details as mentioned in second step of above algorithm. Third matrix will be calculated using first two matrices. Hence we can calculate the total load on each server before optimization by taking total of column values of third matrix.

IV. CONCLUSION

We propose a system which is capable of optimizing the client server assignment, the system is able to balance the communication cost and communication load on the servers in the distributed environment. DOS attack detection is done by checking the threshold value this improves the server's performance significantly which is an important task in our project. The CDD algorithm provides the effective way to distribute the client requests almost equally among all the servers in the system..

V. SIMULATION RESULT

By referring the various existing terminologies we came to the results that NC algorithm is an effective and most popular method which is used in optimizing the performance in client server communication. Unlike NC, Graclus algorithm does not try to isolate the nodes which do not have stronger connection between them, graclus concentrates on group size instead of communication cost. There are many more technologies which are used to do the clustering of weighted graphs in distributed system, but most of them are dealing with igenvector computations, hence this computation leads to the complexity in optimization.

Hence we came to the result that our proposed CDD algorithm is capable of doing the optimization in better and efficient way as compared to the various existing techniques like NC and Graclus algorithms, also there is no need to add new server for optimization.

VI. FUTURE WORK

There are various emerging applications where the CDD algorithm can be applied especially in social networking sites such as Facebook, Twitter or online commercial sites such as Flipcart, Ebay etc. where the number of clients will be very large and this may cause the large communication overhead on the application, hence this may cause the server failure or performance degradation for providing services to the clients. Hence the CDD algorithm is highly applicable for such systems. CDD algorithm can be extended to use in large systems such as Facebook application, it will provide higher efficiency than the existing techniques.

This algorithm can be further extended to apply to various chatting applications where large amount of message exchanging is done between buddies or friends such as What's App, Hike, We chat etc. The communication load due to large message exchanges will be reduced and load fairness i.e. load balancing can be achieved without adding new or proxy server. Hence there number of applications where this algorithm can be applied and we can get significantly better results.

REFRENCES

[1] Krishna Kant, Youjip Won, member IEEE "Server Capacity palnning for web traffic workload", IEEE transaction on knowledge and data engineering, Sept-Oct 1999

[2] D.Saritha, Ch. Satyananda Reddy Andhra Pradesh, India "Optimal Dynamic Load Balance in Distributed Systems for Client Server Assignment" , Vol. 5 (6) , 2014, 8051-8054.

[3] P. Morillo, J. M. Ordu˜na, M. Fern´andez, and J. Duato, Member, IEEE, "Improving the Performance of Distributed Virtual Environment Systems", Jully 2004.

[4] Hiroshi Nishida, Member, IEEE, and Thinh Nguyen, Member, IEEE "Optimal Client-Server Assignment for Internet Distributed Systems", March 2013.

[5] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simpli ed Data Processing on Large Clusters", 2004.

[6] Jianbo Shi and Jitendra Malik, Member, IEEE, "Normalized Cuts and Image Segmentation", August 2000.

[7] Duong Nguyen Binh Ta1, Suiping Zhou1, "Efficient Client-to-Server Assignments for Distributed Virtual Environments", IEEE 2006.

[8] Inderjit S. Dhillon, Yuqiang Guan, Brian Kulis, "Kernel kmeans, Spectral Clustering and Normalized Cuts".

[9] Inderjit S. Dhillon, Member, IEEE, Yuqiang Guan, and Brian Kulis, "Weighted Graph Cuts Without Eigenvectors a Multilevel Approach", November 2007.

[10] Kevin Lang, "Finding Good Nearly Balanced Cuts in Power Law Graphs", 15 November 2004.

[11] Zhenyu Wu, Richard Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation", November 1993.

[12] Inderjit S. Dhillon, Member, IEEE, Yuqiang Guan, and Brian Kulis, "Weighted Graph Cuts without Eigenvectors: A Multilevel Approach" , Nevember 2007.

