A Method Based on K-Center Problem for Efficient Client-Server Assignment in Internet Distributed Systems

Saheeda A, Shanavas K A, Sandra G

Abstract—The internet is a modern collection of interconnected networks of various systems that share resources. Such a system consists of various nodes communicating with each other directly or indirectly. The client server assignment takes part in optimizing the overall performance of these systems. The optimal client server assignment can be done on internet distributed system based on some pre-specified metric on communication cost and load balancing. When we use heuristic via relaxed convex optimization for finding the approximate solution to the client-server assignment problem, inter server communication become increases, while making an average traffic equal in all groups. So in this paper an algorithmic solution based on the k - center problem is introduced to solve the client-server assignment problem for optimizing the performance of a class of distributed systems over the Internet. This algorithm gives better results and done a performance comparison of this algorithm with heuristic via relaxed convex optimization.

Index Terms—Client-server systems, communication overhead,convex optimization, distributed systems, k-center problem, load balancing,optimization.

1 INTRODUCTION

The internet is a collection of several nodes interconnected to each other in such a way to share the resources and computation. In an ideal distributed system, every node has equal responsibility, no node is more computational or more resource powerful than others. But when we consider the real world scenario of a distributed system, implementation of such a system has low performance because of the overhead of coordinating those nodes in a purely distributed manner. In a typical distributed system (Fig.1) servers are more powerful than clients. Examples of such systems are email, instant messaging system (IMS), e- commerce. In intermediate servers handle the communication between two nodes. In this work client server assignment is based on communication load and load balancing.

Emerging application of client server assignment problem is in social network application such as Facebook and Twitter to online distributed auction systems such as eBay. In Facebook, a circle of friends share their messages and pictures. Friends are communicating with other more than non friends. So assigning group of friends to a single server reduces the inter server communication and overall communication load will decrease. This is same as the problem encountered in IMS. In distributed database systems, assigning the search keywords that are frequently queried together to the same server improves the performance by reducing the inter-server communication.

We can begin the client server assignment from the following observation:

1. If two clients who are frequently communicating with each other are assigned to two different servers, the sender (client) first sends messages to a server which has been previously assigned to handle all the messages to and from it. The server receives and forward the messages to the another server which has been previously assigned to handle the messages to and from the receiver (second client). The second server will receive the message and forward it to receiver client. Thus the total communication load increases. If they are assigned to a single server then all communication becomes local. Therefore, it is more efficient to assign all the clients to as few servers as possible.

2. Consider the case with more servers, assigning all clients to only few servers as possible, then some of the servers become heavily loaded and others are idle. So the performance is low with heavily loaded servers. Thus we must consider the load balancing on the server while assigning the clients.

From the above observations, it is clear that total communication load and load balancing are two contradicting features. Hence we need to maintain equilibrium between overall communication load and load balancing of the servers.

This paper solves the client-server assignment problem based on the k - center problem. First, find k servers and assign them to each partition, then clients are assigned to each partition based on their communication pattern. The result will be an approximately optimal client server assignment for a pre-specified tradeoff between load balancing and total communication.

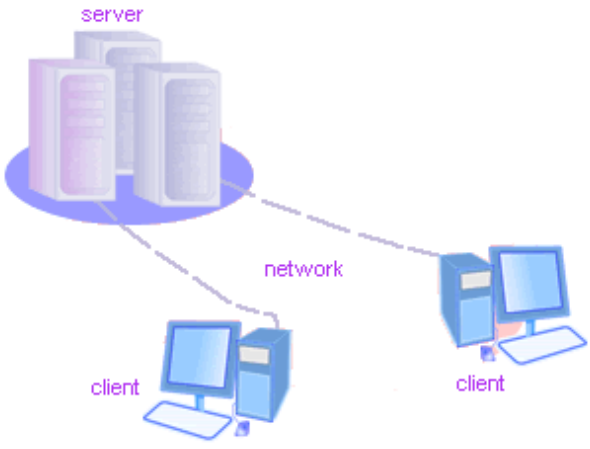

Fig. 1. Typical Distributed System

The literature [1] shows an optimal client server assignment for pre-specified requirements on total communication load and load balancing is NP-hard. A heuristic algorithm based on relaxed convex optimization [1] is used for finding the approximate solution to the client server assignment problem. In this paper the author says an approximation via relaxed convex optimization.

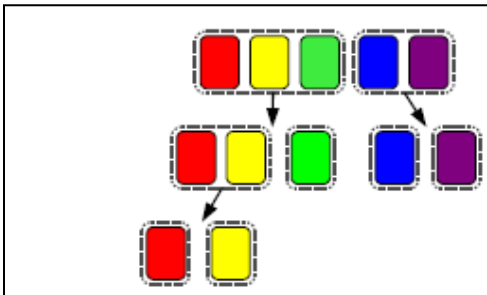

Fig. 2. Example of splitting servers

2 Related Works

There have been a lot of researches on the client server assignment problem based on various metrics.

2.1. Clustering Algorithms

The client server problem can be considered as an instance of clustering problem. Here, the clients and the communication pattern between them can be represented as a graph, with vertices representing the clients and the edges between two vertices representing the communication between the respective clients. Communication frequency between two clients can be represented by the weight of the edge between the corresponding vertices. Fixed number of clusters of clients can be formed from clustering algorithms based on a given objective.

The objective of the clustering algorithm is to minimize the amount of inter group communication and balance the sum of weights of all edges in the group. Most related clustering algorithms is Normalized cuts (NC) [3] that partitions an undirected graph into two disjoint partitions such that F_{ncut} is minimized. NC uses Eigenvalues of the adjacency matrix for solving F_{ncut} efficiently. But NC has a tendency to cause unbalance in the total weight of associated edges of the groups by isolating the vertices that do not have strong connection to others.

In the second literature balanced clustering algorithms for power law graphs [2] are examined. When trying to partition some power law graph in real world situation at yahoo cut quality varies inversely with cut balance. The author investigates about how can find a cut with good balance. The conclusion of this paper is that the most effective approach is to solve a semi definite program and combining multiple trials of randomized flow-based rounding methods which gives effective results.

An approximately optimal client server assignment can be obtained by splitting M servers into two groups and recursively splitting within each group as shown in Fig.2. The time complexity of our algorithms is at least O (N^2logM), which is considerably expensive. Let F_c denote a metric for the sum of the weights of the intergroup edges and F_l denote a metric for the balance of the sums of the weights of the associated edges in the groups. We need to minimize $F_c + F_l$.

2.2. Load Balancing

Here the amounts of tasks and the task assignment in the classical task assignment problem can be viewed as the total load and the client-server assignment, respectively. With our architectural clients communicate with each other indirectly via their servers. So the total load dynamically varies according to the client-server assignment. Similar issues can be found in the client-server assignment for distributed virtual environment (DVE) systems - balancing the workload and reducing the communication between the servers. The DVE systems allow multiple users working on different client computers to interact in a shared virtual world. Efficient client-server assignment for DVE systems can be examined in [4], [5], [6], [7]. However, the load of communication is not seriously considered in DVE systems since the load is primarily generated by processing 3D images. Therefore, unlike our problem, their overall workload is assumed to be constant regardless of the client-server assignment, and uses the amount of communication as a constraint for their optimization problem.

3 Optimal Client Server Assignment

This paper considers the heuristic approach based on relaxed convex optimization [1].Split the number of servers m into two groups with m/2 and m/2 servers in each group. Repeat this

step for each of these two groups until the number of servers in each group equal to 1. Consider as an example, there are 6 say n1, n2, n3, n4, n6 nodes, we need to assign these nodes to 3 servers. According to the heuristic approach of Hiroshi Nishida first splitting these 6 nodes into two groups. So there are 6C2 combinations are there. Then find the pairs which have minimum average traffic (F_c+F_I) minimum. For finding this pair, we need to check 6C2 combinations. After finding those pairs again splitting is done until we get three groups to assign to 3 servers. When we consider a large system with n nodes and m servers, time is more for checking nCm combinations. This makes this algorithm slower and the time complexity will be more This is the first problem encountered in this approach. The second one considers two servers S1 and S2. A, B, C are the clients who are communicating frequently through the server S1. To make the average traffic minimum, we need to assign the node C to S2, then inter server communication increases when those nodes communicating with each other. So we cannot make an optimal solution with this approach. Our paper introduces a new algorithm based on a k - center problem which solves the above disadvantages. And it takes less time.

4 A Method Based On K-Center Problem

Suppose there are k servers and n clients. This method works as follows:
This method first finds k-center nodes, then create k partitions for each of these nodes and assign each center node to each partition. The n-k nodes are assigned to each partition based on their maximum interaction with the center nodes in the partition. Here, center nodes can be obtained by first selecting a node that has maximum traffic with all the other
(n-1) nodes. Next, select a node with minimum traffic with the first node. Again select a node that has minimum traffic with the nodes that are already selected as center nodes. Repeat this until we get k centers. Then assign each center node to each partition. The n-k nodes are then assigned to each partition based on their maximum interaction which each node has at the center nodes belonging to the respective partition. Finally, these k partitions can be assigned to k servers.

As an example considers 100 clients and 10 servers, we need to make an optimal assignment of these 100 clients to 10 servers. So, first we have to find 10 center nodes from 100 clients. First select a node that has maximum traffic with all the other nodes. Next step is to create 10 partitions for each of these nodes and assign each center node to each partition. All the other nodes are assigned to the partition on the basis of the maximum interaction which each node has at the center nodes belonging to the respective partition. Finally, each partition can be assigned to each server.

The steps are as follows:
N-Client nodes, K-Server nodes
1. Set Leader Set L=Φ (N=Nodes).
2. Select a node X with maximum traffic, add it to L.
3. Repeat K-1 times
   Find a node X that has minimum traffic with L, add it to L.
4. Create P_k partitions.
5. For each node Nk in L, add N_k to P_k.
6. For each node in N-L.
7. Find partition P_k to which it has maximum interaction, add X to P_k.
8. The nodes in each partition can be assigned to K servers.

In this method, the time complexity is less than the existing approach.Inter and intra server communication become less here because each partition contain the nodes which have maximum interaction with each other. So communications between servers become less. Also the time taken.

5 Experimental Results

We considered 15 LAN connections with 4 servers and 50 clients in the experimental set up for implementing FTP server. We are actually implementing an automatic request generator in the client. The files are transferred between different clients. The scenarios which we have taken are the client to client communication and our algorithm is applied. Then the message routing server disconnect and reconnect the clients to servers based on the current traffic to reduce the inter server and intra server communication load. This works better than the existing heuristic approach. Then the mean deviation server load is calculated and performance graph is plotted for existing algorithm and k-center method.

The graph in Fig. 3 shows that the execution time and the server load are less about k-center method than the heuristic approach based on relaxed convex optimization.

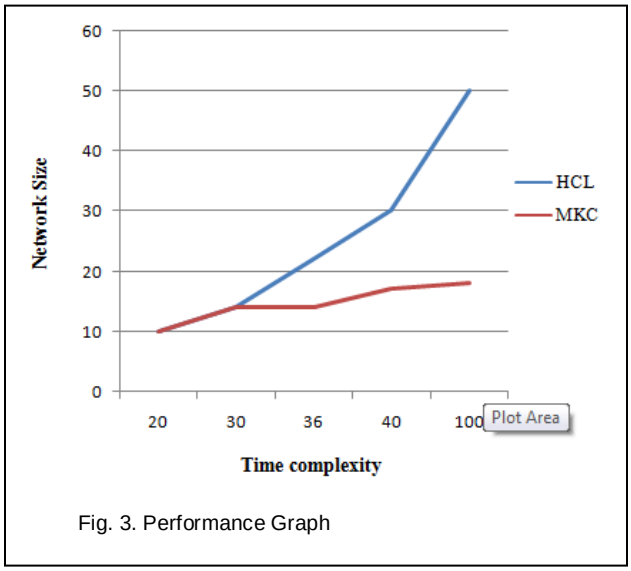

Fig. 3. Performance Graph

The heuristic approach consists of solving n convex optimization problems, each problem corresponding to quantization of a client.All n clients need to be searched for each quantization.

If the convex optimization routine takes the time f (n), then the general algorithm takes O(n (n +f (n)) log m) for m servers which is expensive. The proposed algorithm takes logarithmic time O (log n). Logarithmic running time O (log n) essentially means that the running time grows in proportion to the logarithm of the input size – i.e. time goes up linearly while the n goes up exponentially As an example, if network size is 10, it takes at most some amount of time x, and 100 takes at most, say, 2x, and 10,000 items take at most 4x then it's looking like an O(log n) time complexity.

6 Conclusion

In this paper, we developed a new algorithm based on the k - center problem to the optimal client server assignment problem, which optimizes the performance of distributed systems over the internet. This algorithm provides an optimal client server assignment which reduces the inter and intra server communication load and have less time complexity than the existing heuristic approach based on relaxed convex optimization. Our experimental results show that this algorithm works far better than the existing method.

References

[1] Hiroshi Nishida, Member, IEEE, and Thinh Nguyen, Member, IEEE"Optimal Client-Server Assignment for Internet Distributed Systems, " IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 3, March 2013.

[2] "Finding good nearly balanced cuts in power law graphs," Yahoo Research Labs, Tech. Rep., 2004.

[3] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 228, no. 8, pp. 888-905, Aug. 2000.

[4] J.C.S. Lui and M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems," IEEE Trans.
Parallel and Distributed Systems, vol. 13, no. 3, pp. 193-211, Mar. 2002.

[5] P. Morillo, J.M. Orduna, M. Fernandez, and J. Duato, "Improving the Performance of Distributed Virtual Environment Systems,"IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 7, pp. 637-
649, July 2005.

[6] Y. Deng and R.W.H. Lau, "Heat Diffusion Based Dynamic Load Balancing for Distributed Virtual environments," Proc. 17th ACM Symp. Virtual Reality Software and Technology (VRST '10), pp. 203- 210, 2010.

[7] D.N.B. Ta and S. Zhou, "Efficient Client-To-Server Assignments for Distributed Virtual Environments," Proc. 20th Int'l Conf. Parallel and Distributed Processing (IPDPS '06), 2006.

[8] Pankaj K Agarwal, Cecilia M Procopiuc "Exact and Approximation Algorithm for Clustering" Algorithmica Volume 33, pp 201-226, June 2002,