IJETCR Open Access

ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 2, Issue 4, Page No. 19-24

A Novel Approach on Client Server Assignment Problem in Distributed System

Bharati Patil[1], Dr. S.B. Patil[2], Shital Salve[3]

[1&3]*Research Scholar, Computer Engineering Department, Pune University, JSPM's Imperial college of Engineering & Research, Wagholi, Pune, India*
[1]c2patil.s@gmail.com

[2] *Professor, Head & PG Coordinator, Computer Engineering Department, Pune University, JSPM's Imperial College of Engineering & Research, Wagholi, Pune, India*
[2]dr.psuryakant@gmail.com

ABSTRACT
Inter server communication within two or more servers in distributed system shows interactive performance of the system. While assigning client to server there is problem of latency and interaction time between clients. When we are deal with client assignment problem, two states are consider. In initial assignment phase the zone of virtual world are assigned to server then in next phase, a client is assigned to an appropriate (nearby) server to communicate with the server. By using many tools and algorithms distributed system try to balance the load of server and total communication cost should be less. In this paper based on the analysis of different techniques to assign the client to server, considering the parameters like Load Balancing and Total communication cost, we have proposed Pre-Emptive algorithm to balance the load and minimize the total communication cost of the network.

Key words: Client Assignment Problem (CAP), Load Balance, Total Communication Cost, Inter-latency.

I. INTRODUCTION

Distributed system has various client and server. Clients are communicate with each other with the help of nearby server. So interserver latency problem is there for assigning a client. In distributed virtual environment geographic concern solve interserver latency problem by using heuristic algorithm. Load balancing and total communication is NP hard. By using semi-definite programming algorithm we can find optimal solution for client assignment problem. Different clustering algorithm work in distributed system for server clustering. Cluster makes division of servers or groups of an object. Scheduling algorithms are used to server load balancing.The modern internet is a collection of interconnected networks of various systems that share resources. A superlative distributed system provides every node with equal responsibility, and nodes are similar in terms of resource and computational power. However in real world scenarios it is difficult to achieve overhead of coordinating nodes, which results in lower performance. Typical distributed system consists of servers and clients. Servers are more computational and provide powerful resources than clients. Examples of such systems are e-mail, instant messaging, e-commerce, etc. Communications between two nodes happen through intermediate servers. When node A sends mail to another node B, communication first flows from A to its email server. Email server is responsible for receiving and sending emails, from and for the clients assigned to it. Node A's email server sends data to node B's email server, which in turn is responsible for sending mail to node B. Email servers communicate with each other on behalf of their client nodes. Clients are assigned to a server based on various parameters like organizations, domains, etc. In our paper we solve client-server assignment problem based on total communication load and load balancing on servers. The problem of static load balancing in single class job distributed systems has been studied extensively. The previous studies have employed only the global and the no cooperative approach.

i) ***Global approach:***The focus is on minimizing the expected response time of the entire system over all jobs. Tantawi and Towsley formulated the load balancing problem as a nonlinear optimization problem and gave an algorithm for computing the allocation. Kim and Kameda derived a more efficient algorithm to compute the allocation. Li and Kameda proposed algorithms for static load balancing in star and tree networks. Tang and

Chanson proposed and studied several static load balancing schemes that take into account the job dispatching strategy. Also, there exist several studies on static load balancing in multi-class job systems.

ii) *Cooperative approach:* There are no known studies that involve the cooperative approach for the load balancing problem in distributed systems. This paper makes an attempt to study the cooperative approach.

iii) *No cooperative approach:* There exist only few studies on game theoretic models and algorithms for load balancing in distributed systems. All of them involve non cooperative games. Kameda *et al*. studied non cooperative games and derived load balancing algorithms for both single class and multi-class job distributed systems. For single class job systems they proposed an algorithm for computing the War drop equilibrium. Rough garden formulated the load balancing problem as a Stackel berg game. In this type of non cooperative game one player acts as a leader and the rest as followers. He showed that it is NP-hard to compute the optimal Stackelberg strategy and presents efficient algorithms to compute strategies inducing near optimal solutions. Routing traffic in networks is a closely related problem which was studied from a game theoretic perspective. Orda*et al*. studied a non cooperative game in a network of parallel links with convex cost functions. They studied the existence and uniqueness of the Nash equilibrium. Altman *et al*. investigated the same problem in a network of parallel links with linear cost functions. An important line of research was initiated by Koutsoupias and Papadimitriou , who considered a non cooperative routing game and proposed the *coordination ratio* (i.e. the ratio between the worst possible Nash equilibrium and the overall optimum)as a measure of effectiveness of the system. Mavronicolas and Spirakis derived tight bounds on coordination ratio in the case of fully mixed strategies where each user assigns its traffic with non-zero probability to every link. Rough garden andTardos showed that in a network in which the link cost functions are linear, the flow at Nash equilibrium has total latency at most 4/3 that of the overall optimal flow. They also showed that if the link cost functions are assumed to be only continuous and non decreasing the total latency may be arbitrarily larger than the minimum possible
total latency. Recently, applications of game theory to computer science have attracted a lot of interest and have become a major trend. It is worth mentioning the recent DIMACS Workshop on Computational Issues in Game Theory and Mechanism Design

II.CLUSTERING ALGORITHMS

K-means algorithm

- Objects are classified into one of 'K' group
- Initialize the code book vector of the K cluster
- For every new sample vector the distance between
- New vector and every clusters codebook vector.
- Time complexity O(nkl)
- Space complexity 0(k+n)

A. Hierarchical Clustering Algorithm

- Distance between each pair of patterns called as clusters.
- Find similar pair of patterns and merge them into one cluster.
- This algorithm is more versatile than K-means algorithm because it follows statically data analysis.

B. Self- Organization Map Algorithm

- It is used for Vector quantization and speech recognition.
- Nodes are consider as a number of clients and server for graph.
- Different kinds of distance are measured and joins are used to connect different graph this structure is called as big cluster.
- The stages of SOM algorithms are---

1. Initialization
2. Sampling
3. Matching
4. Updating
5. Continuation

C. *The Expectation Maximization Algorithm*

- It has strong Stastical basis.
- It is linear in database size.
- It accepts desired number of cluster as input.
- In statistics, an expectation–maximization algorithm is an iterative method for finding maximum likelihood or maximum a posteriori estimates of parameters in statistical models, where the model depends on unobserved latent variables

II Client Server Assignment for Distributed Virtual environment

- Distributed virtual environment interact with virtual world via DVE model and number of entities.
- DVE is applied on military simulation, E-learning, multiplayer and games.
- Due to geographical distribution of client server assignment , the virtual world is divided into distinct zones to distribute the load among the servers.

Heuristic Algorithm

- This algorithm finds an approximate solution when other methods to find any exact solution.It is used to improve efficiency of client server assignmentFor server

allocation in distributed system heuristic algorithm is used to solve the NP hard problems.Technical Specifications for this algorithm is---

1. Optimality
2. Completeness
3. Accuracy and Precision
4. Execution Time

The client assignment problem is NP-complete, there is no polynomial time to find the optimal client server assignment. Brute force approach, in which all possible assignment instances in the solution space are evaluated, lacks to find the optimal solution in reasonable time even for small number of clients and servers. For that reason, heuristic algorithms are preferred to find near optimal solutions to the client assignment problem. However, previous solutions fall into a major drawback, which is the dynamicity in real networks. All proposed solutions in the literature assume that clients and servers are always active in the system. On the contrary, in real networks and DIAs, it is highly likely for a client to leave the system for some time and then reconnect. Moreover, servers can fail and may not be able to respond the incoming client requests. Furthermore, performance of the existing solutions is evaluated using the delay measurements obtained by existing.

III. LOAD BALANCING ALGORITHM

1. Least Connection: server selection based on which server currently has fewest client connections.
2. Least response Time: server selection is based on which server has quick response time.

Client server assignment in ismodeled on the fact that the delay incurred by a client is dependent on load of the server and the distance to the assigned server. Delay on the client side is the sum of network delay (proportional to distance to its server) and congestion delay at the server. Hence it focuses on distance between the client and the server and the load on the server in client-server assignment. The problem is defined as Load-distance balancing (LDB) problem. This paper targets 2 flavors of LDB. In the first flavor, the objective is to minimize the maximum incurred delay using an approximation algorithm and an optimal algorithm. In the second flavor, the objective is to minimize the average incurred delay, which the paper mentions it as NP-hard and provides a 2-approximation algorithm and exact algorithm.

However to the best of our knowledge there is no clustering algorithm that is designed as a Semi definite programming problem to achieve our goal: Load balancing on servers and minimizing the communication load between servers, for client-server assignment. Steps for server load balancing is-

(a) The client sends a request which is intercepted by the load balancer transparently.
(b) The load balancer collects the state of the servers.
(c) The load balancer selects an underloaded server.
(d) The load balancer redirects the client's request to the newly selected server

II.ROUND ROBIN ALGORITHMS

Round robinmethod uses a circular list and pointer to last entry for making dispatch decisions. Modern day clusters are being developed with heterogeneous computers, a number of interaction devices and variety of communication medium. Randomized load distribution schemes may not be sufficient. The load balancing system should be able to support a heterogeneous system of servers whose configuration may vary frequently. Configuration may change as a result of addition or removal of servers, server breakdown problem or link failure. Weighted round robin technique is used as a variation of round robinin which each server has integer weight in proportion to the it's capacity. Two new distributed protocols for fair and efficient bus arbitration are presented. The protocols implement round-robin (RR) and first-come first-serve (FCFS) scheduling, respectively. Both protocols use relatively few control lines on the bus, and their logic is simple. The round-robin protocol, which uses statically assigned arbitration numbers to resolve conflict during an arbitration, is more robust and simpler to implement than previous distributed RR protocols that are based on rotating agent priorities. The proposed FCFS protocol uses partly static arbitration numbers, and is the first practical proposal for a FCFS arbiter known to the authors. The proposed protocols thus have a better combination of efficiency, cost, and fairness characteristics than existing multiprocessor bus arbitration algorithms. Three implementations of our RR protocol, and two implementations of our FCFS protocol, are discussed. Simulation results are presented that address: 1) the practical potential for unfairness in the simpler implementation of the FCFS protocol, 2) the practical implications of the higher waiting time variance in the RR protocol, and 3) the allocation of bus bandwidth among agents with unequal request rates in each protocol. The simulation results indicate that there is very little practical difference in the performance of the two protocols.

There are some types of Round Robin Algorithm--------

A. Random Scheduling Algorithm
B. Round Robin Scheduling Algorithm
C. Weighted Round Robin

D. Least Connection Scheduling Algorithm
E. Least Loader Scheduler Algorithm

III. Semi definite Programming Algorithm

Semi definite Programming algorithm is used to solve client server assignment problem. We divide our procedure in some steps-------

1) Formulate the problem satisfying integer quadraticProgramming.
2) Relax the integer variables x1… xn to real variables
3) y1….yn. Change the formula to Vector Programming with variables v1….vn, and change the vector programming
satisfyingSemidefinite programming.
4) Solve the relaxation Semidefinite programming in polynomial time optimally. Obtain the vectors v1…vn throughCholesky Factorization.
5) Use rounding method to get the solution to integer quadratic programming. The rounding methods can be threshold rounding, randomized rounding, etc.

II. **Scheduling Algorithms Evaluation**

1.Deterministic Modeling: Takes a predetermined workload and compute the performance of each algorithm for that workload
2.Queuing Models:Mathematical Approach for handling stochastic workloads
3.Implementation / Simulation: Build system which allows actual algorithms to be run against actual data. Most flexible / general.

VI. CLIENT SERVER COMMUNICATION FOR EMAIL SERVER

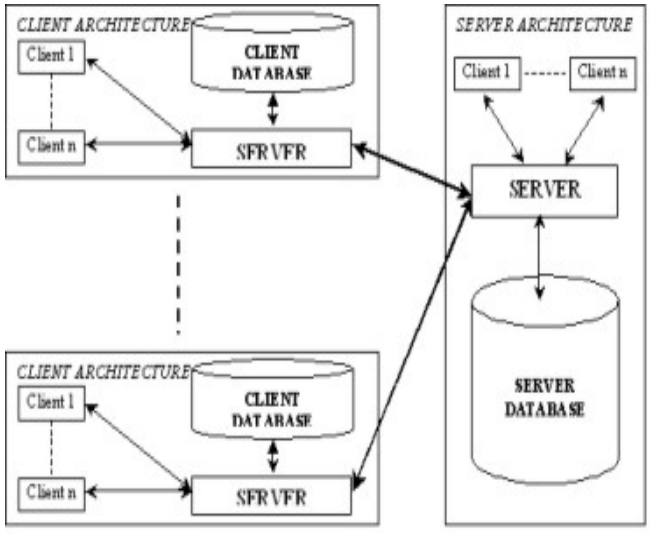

Figure 1: Email server architecture

A typical distributed system consists of a mix of servers and clients. The servers are more computational and resource powerful than the clients. A classic example of such systems is e-mail. When a client A sends an e-mail to another client B, A does not send the e-mail directly to B. Instead, A sends its message to its e-mail server which has been previously assigned to handle all the e-mails to and from A. This server relays A's e-mail to another server which has been previously assigned to handle e-mails for B. B then reads A's e-mail by downloading the e-mail from its server. Importantly, the e-mail servers communicate with each other on behalf of their clients. The main advantage of this architecture is specialization, in the sense that the powerful dedicated e-mail servers release their clients from the responsibility associated with many tasks including processing and storing e-mails, and thus making e-mail applications more scalable. A more interesting scenario is the Instant Messaging System (IMS). An IMS allows real time text-based communication between two or more participants over the Internet. Each IMS client is associated with an IMS server which handles all the instant messages for its clients. Similar to e-mail servers, IMS servers relay instant messages to each other on behalf on their clients. In an IMS that uses the XMPP (Jabber) protocol such as Google Talk, clients can be assigned to servers independent of their organizations.

II. General Solution for Client server assignment problem

The basic idea of our relaxed convex optimization approach for the two-server scenario. We can achieve an approximately optimal client-server assignment for M servers by splitting M servers into two groups and recursively splitting within each group as shown in Fig.

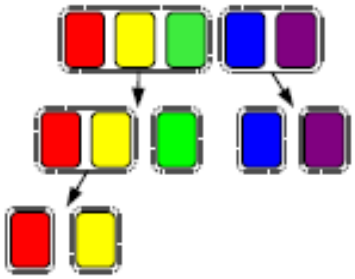

Figure 2: Example of splitting servers binarily

How to split M servers is the central question. If M is even, then it makes sense to split M servers into two equal groups with M=2 servers in each group. In the ideal case, optimizing the load balance between these two groups will result in individual servers in these two groups

having identical communication loads. On the other hand, when making the number of servers in these groups is not same, optimizing the objectives in will result in two groups having total identical communication loads. However, since the two groups have different number of servers, a server within a group with fewer servers will likely to have a higher load than a server in the group with more servers. This reduces the load balance. Therefore, when M is not even, it is necessary to modify the objective at each step, depending on how splitting is done, so as to maintain the similar load at individual servers. Intuitively, the modified objective should reflect the number of servers in each group.

III. Proposed Pre-emptive Algorithm

In distributed interactive applications, each client is connected to one of the servers and pushes/retrieves updates in the system through their connected servers. Thus, any interaction between two clients consists of both clients to server latency and inter-server latency which is called an interaction path. Our objective is to minimize the maximum of these interaction paths between any of the client pairs in the system and Load Balancing.

We Propose a Pre-emptive algorithm which makes partitioning better than round robin algorithm and clustering algorithms with respect to following points

1. To assign a client to server with less time throughput
2. Performance
3. To reduce Interlatency between servers.
4. To balance the load.

IV. CONCLUSION

To make a partition of an object for distributed system clustering algorithms are used to make an partition. But it fails in throughput. Round robin algorithm does not show current status of the system. But pre-emptive algorithm work on scheduling of process of client with respective to time and performance.

V. ACKNOWLEDGMENT

I wish to express my sincere thanks and deep gratitude towards Dr. S.V. Admane [Principal ICOER] for his valuable suggestions and constant encouragement in all phases.

VI. 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,” YahooResearch Labs, Tech. Rep., 2004.
3. Nishida, H.; Thinh Nguyen; , "Optimal Client-Server Assignment forInternet Distributed Systems," Computer Communications andNetworks (ICCCN), 2011 Proceedings of 20th International Conferenceon , vol., no., pp.1-6, July 31 2011-Aug. 4 2011
4. Lu Zhang; Xueyan Tang; , "Client assignment for improvinginteractivity in distributed interactive applications," INFOCOM, 2011Proceedings IEEE , vol., no., pp.3227-3235, 10-15 April 2011doi: 10.1109/INFCOM.2011.5935173
5. Zhang, L.; Tang, X.; , "Optimizing Client Assignment for EnhancingInteractivity in Distributed Interactive Applications," Networking,IEEE/ACM Transactions on , vol.PP, no.99, pp.1, 0doi: 10.1109/TNET.2012.2187674
6. Bortnikov, E., Khuller, S., Li, J., Mansour, Y. and Naor, J. S. (2012),The load-distance balancing problem. Networks, 59: 22–29. doi:10.1002/net.20477
7. Stephen Boyd, Laurent El Ghaoui, Eric Feron, and VenkataramananBalakrishnan. Linear Matrix Inequalities in System and Control Theory.SIAM, 1994.
8. U. Feige and M. Langberg. The rpr2 rounding technique forSemidefinite programs. J. Algorithms, 60(1):1–23, 2006.
9. B. Scho¨lkopf, A. Smola, and K.-R. Mu¨ ller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” Neural Computation, vol. 10, pp. 1299-1319, July 1998.
10. G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” SIAM J. Scientific Computing, vol. 20, pp. 359-392, Dec. 1998.
11. M.L. Huang and Q.V. Nguyen, “A Fast Algorithm for Balanced Graph Clustering,” Proc. 11th Int’l Conf. Information Visualization, pp. 46-52, 2007.
12. K. Andreev and H. Ra¨cke, “Balanced Graph Partitioning,” Proc. 16th Ann. ACM Symp. Parallelism in Algorithms and Architectures, pp. 120-124, 2004.
13. F. Nie, C. Ding, D. Luo, and H. Huang, “Improved MinMax Cut Graph Clustering with Nonnegative Relaxation,” Proc. European Conf. Machine Learning and Knowledge Discovery in Databases: Part II, pp. 451-466, 2010.
14. P. Chan, M. Schlag, and J. Zien, “Spectral K-Way Ratio-Cut Partitioning and Clustering,” IEEE Trans.

Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 9, pp. 1088-1096, Sept. 1994.

15. C.H.Q. Ding, X. He, H. Zha, M. Gu, and H.D. Simon, "A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering," Proc. Int'l Conf. Data Mining (ICDM '01), pp. 107-114, 2001.

16. H.S. Stone, "Multiprocessor Scheduling with the Aid of Network Flow Algorithms," IEEE Trans. Software Eng., vol. 3, no. 1, pp. 85- 93, Jan. 1977.

17. P. Sinha, Distributed Operating Systems: Concepts and Design. IEEE Press, 1997.

18. 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.