Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Novel Method for Client Server Assignment in Distributed System Using Clustering Algorithm

Bharati Patil[1], Prof. V. S. Wadne[2]
[1] Research Scholar, [2] Assistant Professor
Computer Engineering Department, Pune University
JSPM's Imperial College of Engineering & Research, Wagholi, Pune, India

Abstract— Distributed system is a collection of heterogeneous network. All networks are communicated with each other via internet. Distributed system has centric approach towards client-server. In client server network there are two distinct nodes, one is client and second is server. All these distinct nodes are communicated and co-ordinate with each other via messages. There are so many servers and clients. Assignments of client to the server is based on priority, for priority there are so many algorithms which shows load balancing. Round robin algorithm is uses many policies for assignment and load balancing.Server affinity is used in combination with one of the standard load balancing methods: round-robin, weight-based, or random:

- ***round-robin-affinity— server affinity governs connections between external Java clients and server instances; round robin load balancing is used for connections between server instances.***
- ***weight-based-affinity—server affinity governs connections between external Java clients and server instances; weight-based load balancing is used for connections between server instances.***
- ***random-affinity—server affinity governs connections between external Java clients and server instances; random load balancing is used for connections between server instances.***

Key terms: Load Balance, client server assignment, round robin, Load distance balancing problem, Round robin, Clustering algorithm, shared nearest algorithm etc.

I. INTRODUCTION

A client can request an initial context from a particular server instance in the cluster, or from the cluster by specifying the cluster address in the URL. The connection process varies, depending on how the context is obtained:

- If the initial context is requested from a specific Managed Server, the context is obtained using a new connection to the specified server instance.
- If the initial context is requested from a the cluster, by default, context requests are load balanced on a round-robin basis among the clustered server instances.

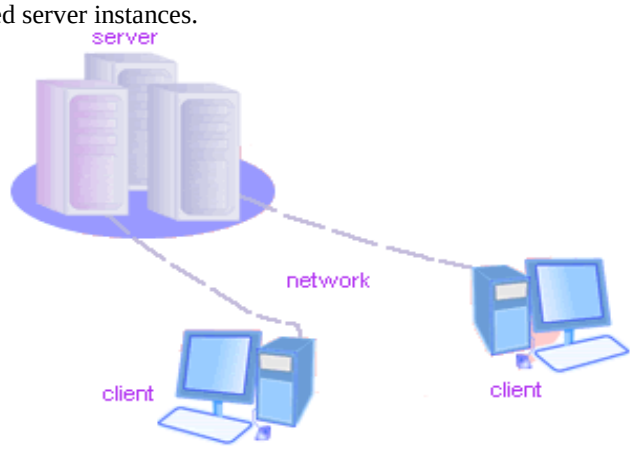


Fig. 1. Distributed System

II. RELATED WORK

Client server assignment can be designed based on the following observations;
1. If two clients are assigned to the same server, theserver will receive message from one client andforward to another one. If they are on differentservers, the sender client first sends to its server. Thesender's server will receive data and forward it to thereceiver's server. The receiver server will receive dataand forward it to receiver client. Thus the totalcommunication between two frequently interactingclients increases if they are assigned to two differentservers. Assigning these two clients to a single servermakes all information exchange locally. It is moreefficient to have all the clients assigned to fewservers to minimize total communication.

2. On the contrary having fewer servers, results in those
servers being heavily loaded while others are leftunderutilized. If a server is heavily loaded it results inlow performance due to excessive resource usage onthat server. Thus it is necessary to consider loadbalance on the server while assigning clients.

System Architecture

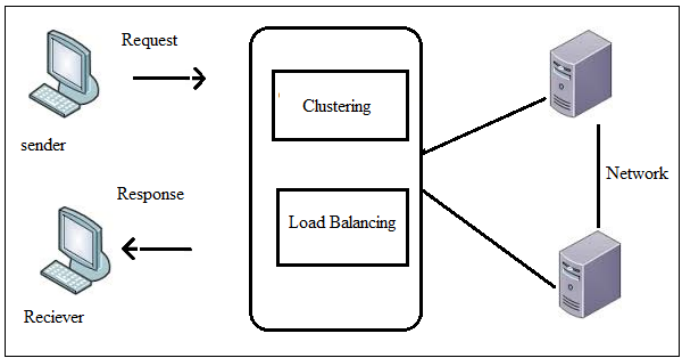
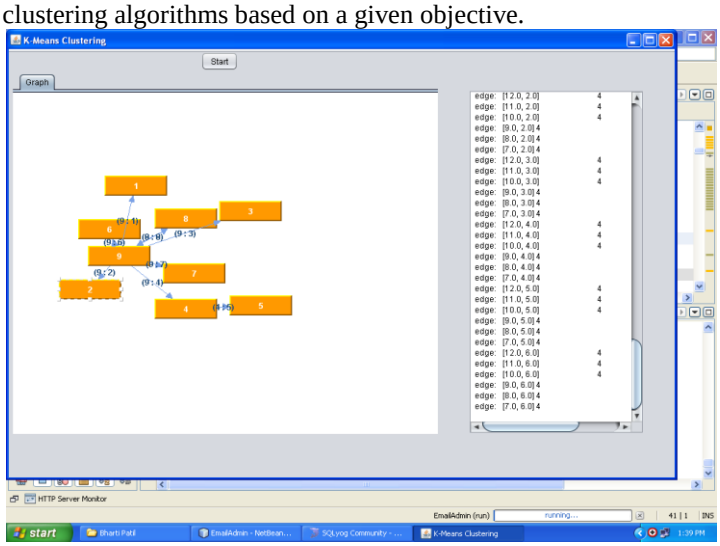


Fig : Two Server Architecture diagram

Clustering Algorithm

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.



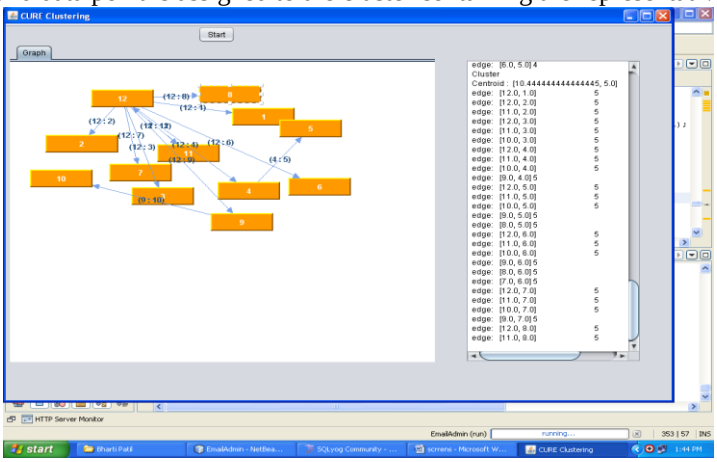
Cure Algorithm

CURE (Clustering Using Representatives) is an efficient data clustering algorithm for large databases that is more robust to outliers and identifies clusters having non-spherical shapes and wide variances in size.To avoid the problems with non-uniform sized or shaped clusters, CURE employs a novel hierarchical clustering algorithm that adopts a middle ground between the centroid based and all point extremes. In CURE, a constant number c of well scattered points of a cluster are chosen and they are shrunk towards the centroid of the cluster by a fraction α. The scattered points after shrinking are used as representatives of the cluster. The clusters with the closest pair of representatives are the clusters that are merged at each step of CURE's hierarchical clustering algorithm. This enables CURE to correctly identify the clusters and makes it less sensitive to outliers

The running time of the algorithm is $O(n^2 \log n)$ and space complexity is $O(n)$.The algorithm cannot be directly applied to large databases. So for this purpose we do the following enhancements

- Random sampling : To handle large data sets, we do random sampling and draw a sample data set. Generally the random sample fits in main memory. Also because of the random sampling there is a tradeoff between accuracy and efficiency.
- Partitioning for speed up : The basic idea is to partition the sample space into p partitions. Each partition contains n/p elements. Then in the first pass partially cluster each partition until the final number of clusters reduces to n/pq for some constant $q \geq 1$. Then run a second clustering pass on n/q partial clusters for all the partitions. For the second pass we only store the representative points since the merge procedure only requires representative points of previous clusters before computing the new representative points for the merged cluster. The advantage of partitioning the input is that we can reduce the execution times.

- Labeling data on disk : Since we only have representative points for *k* clusters, the remaining data points should also be assigned to the clusters. For this a fraction of randomly selected representative points for each of the *k* clusters is chosen and data point is assigned to the cluster containing the representative point closest to it.



Comparative Analysis of Algorithms

Table 1: Analysis of Algorithm

Parameter	Clustering Algorithm	Cure Algorithm
Start Time	155718 ms	155850 ms
End Time	155727 ms	155852 ms
Total Execution	9 ms	2 ms

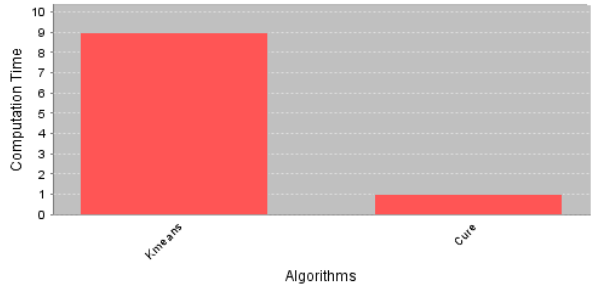
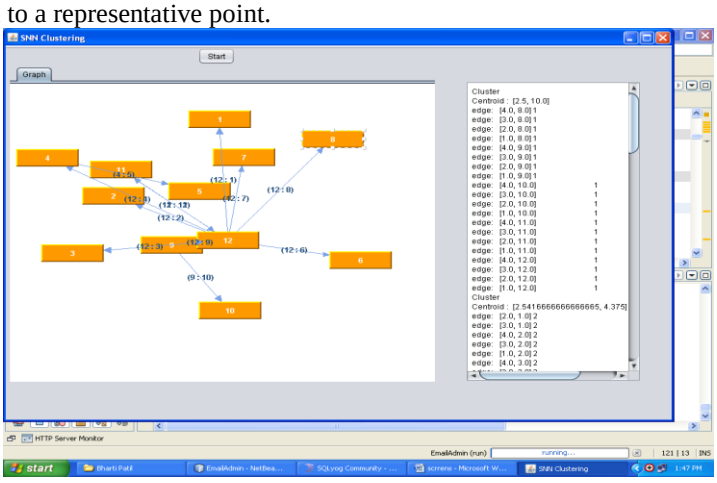


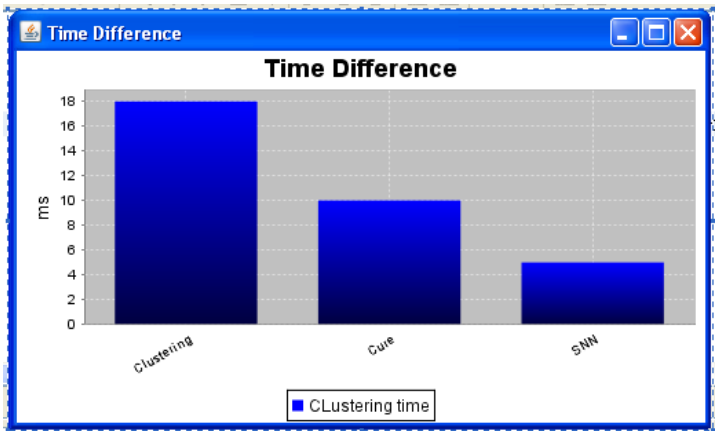
Fig.2 Comparative Study of Clustering and CURE Algorithm

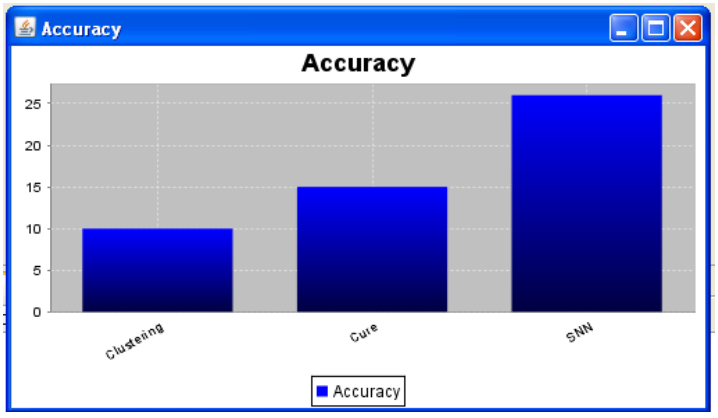
SNN Algorithm

1. Construct the similarity matrix.
2. Sparsify the similarity matrix using k-n sparsification.
3. Construct the shared nearest neighbor graph from k-nnsparsified similarity matrix.
4. For every point in the graph, calculate the total strength of links coming out of the point. (Steps 1-4 are identical to the Jarvis – Patrick scheme.)
5. Identify representative points by choosing the points that have high total link strength.
6. Identify noise points by choosing the points that have low total link strength and remove them.
7. Remove all links that have weight smaller than a threshold.
8. Take connected components of points to form clusters, where every point in a cluster is either a representative point or is connected to a representative point.



III. RESULT ANALYSIS





IV. CONCLUSION

In Distributed system clustering algorithm is used to make a partition of an object. By using clustering algorithm server balances the load. In this paper three clustering algorithms are used to determine the efficiency of an algorithm. As per result it is found that SNN algorithm has better execution time and accuracy as compare to k-means clustering algorithm and CURE algorithm.

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 ComponentAnalysis 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 MultilevelScheme for Partitioning Irregular Graphs," SIAM J. ScientificComputing, vol. 20, pp. 359-392, Dec. 1998.

[11] M.L. Huang and Q.V. Nguyen, "A Fast Algorithm for BalancedGraph 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 MinMaxCutGraph Clustering with Nonnegative Relaxation," Proc. EuropeanConf. Machine Learning and Knowledge Discovery in Databases: PartII, pp. 451-466, 2010.

[14] P. Chan, M. Schlag, and J. Zien, "Spectral K-Way Ratio-CutPartitioning and Clustering," IEEE Trans. Computer-Aided Design ofIntegrated 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-MaxCut 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 NetworkFlow Algorithms," IEEE Trans. Software Eng., vol. 3, no. 1, pp. 85-93, Jan. 1977.

[17] P. Sinha, Distributed Operating Systems: Concepts and Design. IEEEPress, 1997.

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

