ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 7, No. 1, 2012, *pp*. 019-024

A Fuzzy Intra-Clustering Approach for Load Balancing in Peer-to-Peer System

Rupali Bhardwaj[1], V.S.Dixit[2], Anil Upadhyay[3]

[1]Department of Master of Computer Applications, Ajay Kumar Garg Engineering College, Ghaziabad, UP, India

[2]Department of Computer Science, Atma Ram Sanatan Dharma College, Delhi University, New Delhi, India

[3]Department of Applied Mathematics & Science, Bipin Chnadra Tripathi Kumaon Engineering College, Dwarahat, Almora, India

(*Received October 23, 2011, accepted December 2, 2011*)

Abstract. Load balancing algorithms goal is to keep all nodes normally loaded through migration of modules from heavy loaded nodes to lightly loaded nodes. In addition, load balancing must involve low communication overhead and respond quickly to load imbalance in the system. In previous load balancing algorithms, classification of node as heavy, light or normal loaded node is done by using concept of threshold level, which is fixed and predefined. Now, today scenario is to change the status of nodes dynamically according to the state of system. So that, in this paper we proposed an algorithm for load balancing using fuzzy clustering; which improves the performance of the system without pre-defining the threshold values. The proposed algorithm is compared with other existing algorithms and is found to be fast and efficient in reducing load imbalance in Peer to Peer system.

Keywords: heavy weighted, light weighted, fuzzy system, clustering, load balancing

1. Introduction

A Peer to Peer (P2P) system in which every participating node acts both as a client and as a sever (servent) and share a part of their own hardware resources such as processing power, storage capacity or network bandwidth. A P2P system will have a number of peers (nodes) working independently with each other. Each node is classified either as heavy or light weighted node using fixed threshold level. The use of single-threshold value may lead to a useless load transfers and make the load balancing algorithm unstable because a node's status may be light weighted when it decides to accept a remote process, but it status may be heavy weighted whenever the remote process arrives. Therefore, a light weighted node becomes heavy weighted node and will again invoke load balancing algorithm to change its status. (Alonso and Cova, 1988) proposed a double-threshold levels algorithm to reduce the instability of the single-threshold level policy ; however those two threshold levels are fixed and predefined. One desired feature of load balancing algorithm is change the status of nodes dynamically. Some load balancing algorithms (A Rao & Stoica , 2003; G. Shivaratri & Singhal,1992; Rajeev Gupta & Prabha Gopinath ,1990; Sonesh Surana & I Stoica, 2004) use fixed threshold levels for load balancing. Load balancing using fuzzy system is a natural extension of double threshold level approach and it improves the performance without pre-defining the threshold values. The remainder of the paper is organized as follows: Section 2 briefly reviews fuzzy logic controller and clustering. The proposed algorithm using fuzzy clustering is presented in section 3. Section 4 & 5 describes existing load balancing algorithms such as Round Robin and BID algorithm and experimental study. Finally, we conclude the paper in section 6.

2. Background

2.1. Fuzzy Logic Controller

Fuzzy logic is a powerful mathematical tool in representing linguistic information and is very useful to solve problems that do not have a precise solution. The architecture of the fuzzy logic controller is shown in figure 1, it includes four components: Fuzzifier, Inference Engine, Fuzzy knowledge rule base and

Defuzzifier (A. Karimi & Saripan, 2009; Ally E. EI-Bad, 2002; M.C.Huang &Vairavan, 2003 *;* T. J. Ross, 1995)

- Fuzzification

It converts the crisp input value to a linguistic variable using the membership functions stored in the fuzzy knowledge base.

- Fuzzy knowledge rule base

It contains the knowledge on the application domain and the goals of control.

- The inference engine

It applies the inference mechanism to the set of rules in the fuzzy rule base to produce a fuzzy set output.

- Defuzzification

It converts the fuzzy output of the inference engine to crisp value using membership functions similar to the ones used by the fuzzifier.

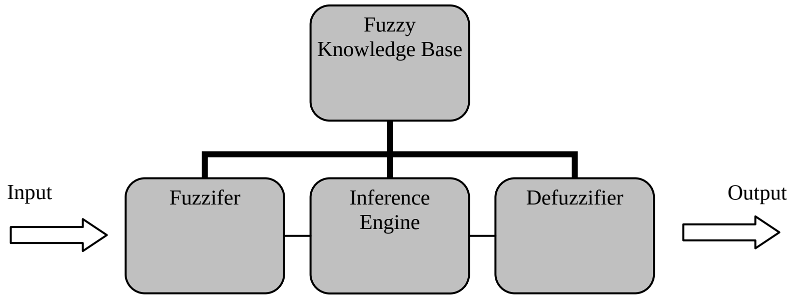

Figure 1: Architecture of fuzzy logic controller

2.2. Clustering

Clustering is a tool for data analysis, which solves classification problems. Its motive is to distribute cases (people, objects , events etc.) into groups, so that the degree of association to be strong between objects of the same cluster and weak between objects of different clusters (Clark F. Olson ,1996) .Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups. In brief, cluster analysis groups data objects into clusters such that objects belonging to the same cluster are similar, while those belonging to different clusters are dissimilar. A cluster is therefore a collection of objects which has “similarity” between them and has “dissimilarity” to the objects belonging to other clusters. We can show this with a simple graphical example (Osmar R. Zaïane,1999).

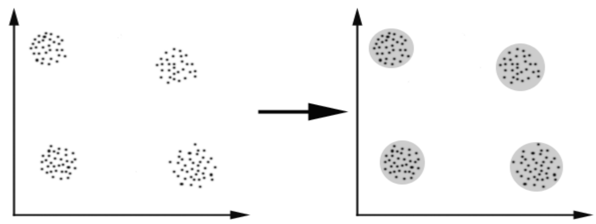

Figure 2: organizing objects into groups whose members are similar in some way

Clusters are formed by two methods-“distance –based clustering”, two or more objects belong to a given distance (in this case geometrical distance). Another kind of method is “conceptual clustering”, two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts.

3. Fuzzy-Clustering Load Balancing Approach

Assuming the poll of dynamic cluster heads in hierarchical method, based on dynamic cluster node; dynamic cluster heads are constructed by using the terms in node and cluster heads will change when different cluster nodes are merged. The degree of similarity between node clusters is calculated based on these dynamic node cluster heads. Dynamic clusters are formed by basically two things-

(1) Similarity between two data method.

(2) Distance from data to cluster head.

Weight w_{ik} of term t_i in node n_k is calculated by using the following formula given by (Salton G., 1971).

$$w_{ik} = \frac{tf_{ik}}{\max_j tf_{ik}} \times INF_j \quad (1)$$

where $INF_i = \log_{10} \frac{N}{n_i}$, N is total number of nodes, INF_i denotes the inverse node frequency of term t_i, n_i denotes the number of nodes which contain term t_i and tf_{ik} denotes the frequency of term t_i appearing in node n_k.

Calculate fuzzy degree of node with help of equation 2, v_{ki} of term t_i with respect to node n_k based on inverse node frequency factor $INF_{k,i}$ of term t_i, where $v_{ki} \in [0, 1]$. $INF_{k,avg}$ denotes the mean value of the terms in node n_k and $INF_{k,\max}$ denotes the maximum INF values of the terms in node n_k.

$$v_{ki} = \frac{INF_{k,\max} - INF_{i,\min}}{INF_{k,\max} - INF_{k,avg}} \quad (2)$$

Calculate the 'relative time to live' with respect to every term t_i in node n_k, where

$$I_{ki} = \left(\frac{v_{ki}}{v_{k,\max}} \right) - \left(\frac{v_{ki}}{v_{k,\min}} \right) \quad (3)$$

where $I_{ki} \in [0, 1]$, v_{ki} denotes the degree of effect of term t_i in node n_k and $v_{k,\max}$ denotes the maximum degree of effect of the terms in node n_k. Calculate the degree of similarity between two node clusters c_i and c_j, s denotes the number of terms appearing in both node clusters c_i and c_j.

$$s(c_i, c_j) = \frac{\sum_{k=1,2,3,....s}^{N} Min(v_{ik}, v_{jk})}{\sum_{k=1,2,3,...,s}^{N} Max(v_{ik}, v_{jk})} \quad (4)$$

where $s(c_i, c_j) \in [0, 1]$

Now, clusters are organized as dynamic hierarchical clusters (Witold,2005) and cluster loads are categorized as very light , light, heavy and very heavy cluster.

Table 1: Membership Function

Number of node	μ_{VLN}	μ_{LN}	μ_{HN}	μ_{VHN}
0	1.0	1.0	0.0	0.0
1	1.0	1.0	0.0	.1
2	.8	.9	.2	.2
3	.7	.8	.3	.3
4	.6	.8	.4	.4
5	.5	.7	.5	.5
6	.4	.6	.6	.6
7	.3	.5	.7	.7
8	.2	.4	.8	.8
9	.1	.3	.9	.9
10	.1	.2	1.0	1.0

Apply the inference rules according to Table 1-

$$\mu(receiver)_{\max} = (VLN) \wedge (LN) \; \mu(sender)_{\min} = (HN) \vee (VHN)$$

Load transfer table is presented in table2, assuming sender initiated load balancing algorithm, and proposed rule base is as follows-

Rule 1: If (cluster_load is very_light) then (cluster is reciever)

Rule 2: If (cluster_load is light) then (cluster is reciever)

Rule 3: If (cluster_load is moderate) then (cluster is normal)

Rule 4: If (cluster_load is heavy) then (cluster is sender)

Rule 5: If (cluster_load is very_heavy) then (cluster is sender)

Table 2: Load Transfer Table

Operation	Notation	Expression
Moderate=NOT(ML)	$\mu_m(x)$	$\{1-\mu_m(x)\}$
Sender=HN OR VHN	$\mu_{HN \vee VHN}(x)$	Max$\{\mu_{HN}(x), \mu_{VHN}(x)\}$
Receiver=VLN AND LN	$\mu_{VLN \wedge LN}(x)$	Min$\{\mu_{VLN}(x), \mu_{LN}(x)\}$

4. Existing Reference Algorithms

We have used two algorithms, which are relevant to our context, as reference algorithms to compare the result of our algorithm.

4.1. Round Robin Algorithm

In the round robin algorithm (Pradeep K. & Sinha, 1996), processes are equally divided among all processors. Each new process is assigned to a new processor in a round robin fashion. Whenever number of processes larger than number of processors, round robin algorithm works well. No need of inter-process communication in case of round robin algorithm.

4.2. BID Algorithm

In the bidding algorithm (Z. Xu & Haang, 2008) , when a node is heavy loaded, it multicasts a request for bids to the other nodes in the system. After collection of all bids by heavy loaded node, the best bid is chosen as light weighted node. If none of the node is found in the group for load transfer, the bidding procedure starts over again.

5. Experimental Study

We use MATLAB to evaluate our load balancing algorithm. We show

- Compare the performance of our proposed algorithm with the other algorithms discussed in section 4.
- Compare the performance of sender node vs. receiver node using fuzzy expert system.
- The effect of node failure, concluding that nodes transmission is more stable with fuzzy expert system as compare to non fuzzy system.
- The effect of throughput, concluding that throughput is better in proposed algorithm as compare to BID algorithm.

5.1. Performance Comparison

Figure 3 captures the tradeoff between number of nodes and total execution time. Each point on the lower line corresponds to the effects of our proposed algorithm. This observation indicates that performance of our algorithm is far better than existing algorithms in section 4.

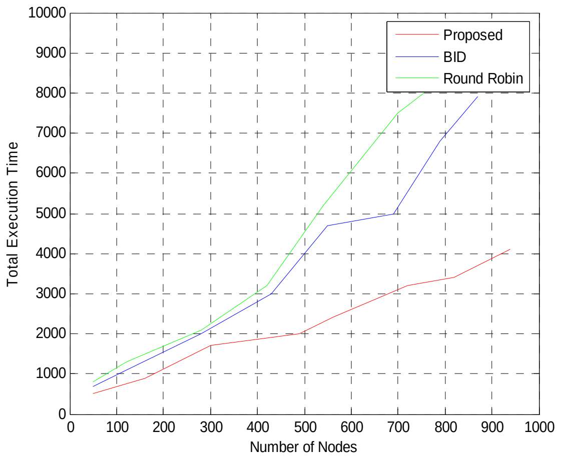

Figure 3: Total execution time vs. number of nodes

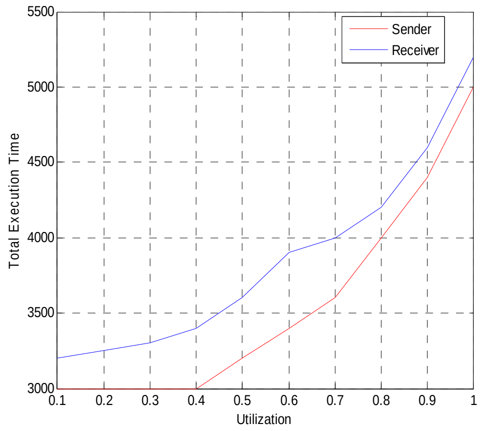

Figure 4: Total execution time vs. utilization

5.2. Performance comparison of sender vs. receiver node

In this section, we will evaluate the performance of sender node with receiver node using fuzzy expert system. This observation (figure 4) indicates that sender node execution time is less as compare to receiver node, means performance of sender initiated load balancing algorithm using fuzzy expert system is better than receiver initiated load balancing algorithm.

5.3. Effect of Node Failure

As we have already mentioned that in case of node failure, nodes transmission is more stable using fuzzy expert system as shown in figure 5. The various Parameter values used for simulation are shown in Table 3.

Table 3 Parameter Values

Parameter	Value
Number of nodes	50
Surface	$20m \times 20m$
Transmission range	15m
Data transmission rate	15 node/sec
Failure model	Random
Size of node	128 bytes

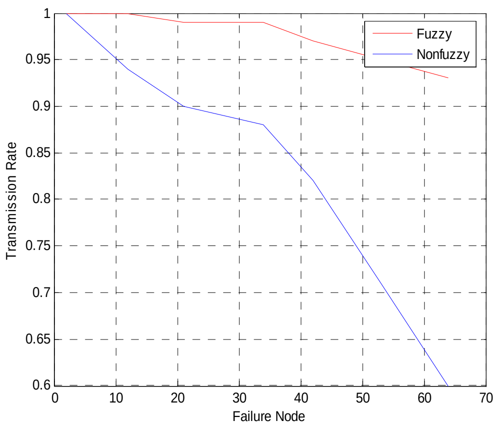

Figure 5: Transmission rate vs. failure node

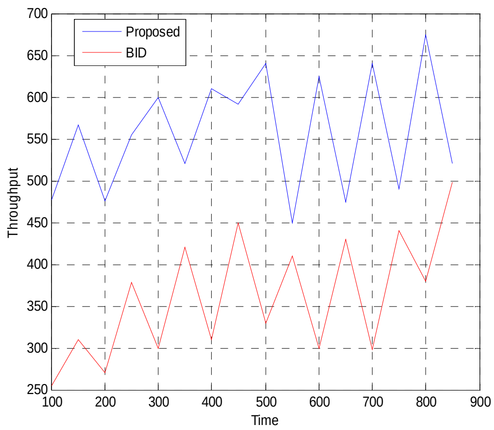

Figure 6: Throughput

5.4. Throughput

This observation (figure 6) indicates that throughput is more in case of proposed algorithm as compare to

BID algorithm, means throughput of load balancing algorithm using fuzzy expert system is better than BID algorithm.

6. Conclusion

A novel approach, to solve the problem of dynamic load balancing in P2P system using fuzzy clustering, is developed. Nodes are divided into clusters based on their membership function values. We have compared three load balancing algorithms: BID, round-robin and the proposed one. The simulation results show the proposed algorithm is more efficient and flexible than existing algorithms in terms of execution time and throughput for various number of nodes. The future work will be improving the performance of proposed algorithm by cluster partitioning.

7. References

[1] A.. Karimi, F.Zarafshan, A.b.Jantan, A.Ramil, M.Iqbal b. Saripan. A New Fuzzy Approach for Dynamic Load Balancing algorithm. *International Journal of Computer Science and Information Security*. 2009, **6**(1).

[2] Ally E. EI-Bad. Load Balancing in Distributed Computing Systems Using Fuzzy Expert Systems.*TCSET'2002*. Lviv-Slavsko, Ukraine, 2002.

[3] Alonso, R. and Cova, L. L. Sharing Jobs Among Independently Owned Processors. *In Proceeding of the 8th International Conference on Distributed Computing Systems*. NewYork: IEEE. 1988, pp. 282-288.

[4] Ananth Rao, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp and Ion Stoica. Load Balancing in Structured P2P systems. *In 2nd International Workshop on Peer to Peer Systems (IPTS)*, 2003.

[5] Clark F. Olson. Parallel algorithms for hierarchical clustering. *Technical report.* University of California at Berkeley, 1996.

[6] M.C.Huang, S.H.Hosseini, K.Vairavan. A Receiver Initiated Load Balancing Method In Computer Networks Using Fuzzy Logic Control. GLOBECOM 2003, 0-7803-7974-8/03. (2003)

[7] Iranian G. Shivaratri, Philip Krueger and Mukesh Singhal. Load Distributing for locally distributed System. *IEEE 1992*. 1992, pp.33-44

[8] Niranjan G. Shivaratri, Philip Krueger. Two adaptive location policies for global scheduling algorithms. *IEEE 1990*. 1990, pp. 502-509.

[9] Osmar R. Zaïane. *Principles of Knowledge Discovery in Database*.

[10] Pedrycz Witold. Knowledge-Based Clustering. *ISBN 0-471-46966-1*. John Wiley & Sons, Inc . 2005.

[11] Pradeep K. and Sinha. Distributed Operating Systems Concepts and Design. *IEEE 1996.*

[12] Rajeev Gupta, Prabha Gopinath. *A Hierarchical approach to Load Balancing in Distributed Systems*. *IEEE 1990*. 1990, pp. 1000-1005.

[13] Salton G. *The Smart Retrieval System-Experiments in Automatic Document Processing.* Englewood Cliffs, NJ, USA: Prentice Hall, 1 971.

[14] Sonesh Surana, B. Godfrey. K. Lakshminaraynan, R. Karp, I Stoica. Load Balancing in dynamic structured Peer to Peer system. *IEEE 2004.* 2004, pp. 2253-2262

[15] Timothy J. Ross. Fuzzy Logic with Engineering Applications. McGraw Hill, 1995.

[16] Z. Xu, R. Haang. *Performance Study of Load Balancing Algorithms in Distributed Web Server System*. 2008.