A Hybrid of Improved Weighted Round Robin and Least Load Algorithm To Optimize The Load Balancing In Cloud Computing Environment

Ms. Tanvi Gupta, M. V. Ramana Murthy, S.S Handa

Abstract: Day by Day there is an increase in internet users which leads to increase in traffic in the network, this requires the balancing of the load in the network on the servers by different Load balancing techniques. If we review on the load balancing techniques , there are several possibilities to enhance and improvised the techniques. Few techniques that are already defined are round robin algorithm (static load balancing),Weighted Round Robin algorithm and Least Load algorithm(Dynamic Load Balancing). A researcher named D.Chitra Devi .et .al has given the concept of improved weighted round robin algorithm(IWRR) which gives much better response as compared to simple round robin algorithm. Another researcher Rashmi saini.et.al suggested the hybrid of round robin algorithm and least Load Algorithm. So, my proposed work is to hybrid the IWRR algorithm with Least Load Algorithm which gives much more improvised results as compared to both of their work as IWRR algorithm also consider the length of task on the priority basis.

Keywords: Load Balancing, weighted round robin, improved weighted round robin algorithm, least load algorithm

I. INTRODUCTION

Figure 1 shows the concept of Load Balancing and scheduling design. Figure shows the job request is given by the user through the interface and passed to the task manager for dependency and independent task analysis .Then the job is given to scheduler. The scheduler has the logic to find out the most suitable server and assign the task to the servers based on the algorithms.Other than the scheduler ,there is a load balancer ,the function of it is to decide the migration of the task from a heavily loaded server to an ideal server or least loaded server.

Also ,the Resource monitor communicates with all the machine's resource prober and collects the machine capabilities, current load on each machine and number of jobs in execution /waiting queue in each machine .

Manuscript received July 08, 2016

Ms. Tanvi Gupta, CSE dept, Manav Rachna International University Faridabad, India

M. V. Ramana Murthy, Dept. Of Mathematics& Computer Science. Osmania University, Hyderabad, India

S.S Handa, CSE dept, Manav Rachna International University,Faridabad,India.

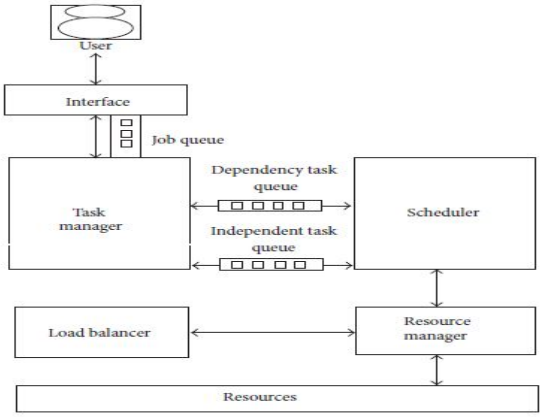

Fig 1: Scheduling and load balancing design[9]

Next thing to be considered is the algorithms for scheduling and load balancing.

1)Dynamic Load Balancing

In dynamic load balancing , work load is distributed among the processors at runtime. The master assigns new processes to the slaves based on the new information collected.

Least Load Algorithm[23]:

Algorithm:

```
BEGIN PROCEDURE LEAST_LOAD_ALGO
SET array ERVERS={s1,s2,s3……sn};
SET array SERVER_LOAD={l1,l2,l3….ln};
WHILE(request) DO
    Integer POS=FIND_MIN(SERVER_LOAD);
   GOTOSERVER[pos];
END WHILE
END PROCEDURE
BEGIN PROCEDURE FIND_MIN(SERVER_LOAD[1 to n])
Set Integer POS=0;
For I=1 ti n-1 do
IFSERVER_LOAD[POS]>SERVER_LOAD[I]THEN
   POS=I;
END IF
RETURN POS;
END PROCEDURE
```

The above algorithm uses the policy of Shortest Remaining Processing Time which is the optimal algorithm for minimizing mean response time. The job that has the least remaining process time will be served.

But the problem in this policy is that , the jobs with the larger size may be waiting for a while and dispatcher or load balancer must know the size of job s beforehand.

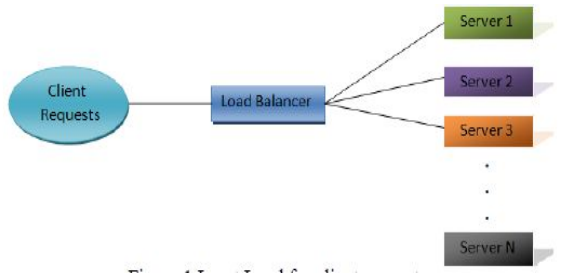

Figure2 : Least Load for client request.[23]

2) Static Load Balancing

In static Load balancing, the processor's performance is determined at the beginning of the execution. Then only the work load is distributed to the processors at the start level according to the performance determined. At the initial stage , the processors are considered to be unloaded. It sends message to all the remote processors regarding new the load state, if the load state of the processor exceeds a load level limit. If it is not overloaded than the process is allocated locally.

Round Robin[23]: Round robin is a simple continuous looping fashion technique , in which the user's content access request is responded to by the load balance in the rotational manner, the first access grant access to the first available content server giving its IP Address and second to the second server IP Address and so on. Whenever a server IP Address is given, instantly its IP Address is moved back to the list of available IP Addresses and gradually it moves back to the top of the list and becomes available again.

```
ALGORITHM:
BEGIN PROCEDURE ROUND_ROBIN_ALGO
SET Static Integer COUNT = 0;
SET integer QUANTUM = q;
SET array SERVERS = {s1,s2,s3.... ....sn};
Integer SQ = q*n;
WHILE ( request) DO
        Integer RANGE = COUNT % SQ;
        IF RANGE > 0 AND RANGE < = 1*Q THEN
                GOTO SERVER[s1]
        ENDIF
        IF RANGE > 1*Q AND RANGE < = 2*Q THEN
                GOTO SERVER [s2]
        ENDIF
        :
        IF RANGE > (n-1)*Q AND RANGE < = n*Q THEN
                GOTO SERVER [sn]
        ENDIF
END WHILE
END PROCEDURE
```

Weighted Round Robin Algorithm[9]:

In this algorithm, the resource capabilities of the machines are considered and the machines having the higher capacity assigns the higher number of tasks which is based on the weightage given to each machines.

But the problem is that it does not consider the length of the tasks to select the appropriate machine.

Improved Weighted Round Robin Algorithm[9]:

D.Chitra Devi .et.al states that the IWRR algorithm is the most optimal Algorithm and it allocates the job to the most suitable machine based on the machine's information like its processing capacity , load on the machine and the length of the arrived tasks with its priority. The static scheduling of this algorithm uses the processing capacity of the machine, the number of incoming tasks, and the length of each task to decide the allocation on the appropriate machine.

The dynamic scheduling (at run time) of this algorithm additionally uses the load on each of the VMs along with the information mentioned above to decide the allocation of the task to the appropriate machine .There is a probability at run time that, in some of the cases, the task may take longer execution time than the initial calculation due to the execution of more number of cycles (like a loop) on the same instructions based on the complicated run time data.

In such situations, the load balancer rescues the scheduling controller and rearranges the jobs according to the idle slot available in the other unutilized/underutilized machines by moving a waiting job from the heavily loaded machines.

Nature of Load Balancing[23]

Co-operative: In the cooperative situation all processors have the accountability to carry out its own portion of the scheduling task, but all processors work together to achieve a goal of better efficiency. In the non-cooperative individual processors act as independent entities and arrive at decisions about the use of their resources without any effect of their decision on the rest of the system.

Process Migration: Process migration parameter provides when a system decides to export a process. It decides whether to create it locally or create it on a remote processing element. The algorithm is capable to decide that it should make changes of load distribution during execution of process or not.

Resource Utilization: Resource utilization include automatic load balancing A distributed system may have unexpected number of processes that demand more processing power. If the algorithm is capable to utilize resources, they can be moved to under loaded processors more efficiently.

Limitations of Round Robin and Least Load Algorithm

Both the Load Balancing Algorithms of Round Robin and Least Load Algorithms works good in particular criteria but also have certain limitations. The main disadvantage of Round Robin is that it is simple and runs fast but when it comes to load balancing of various sizes and complexity of load or request , there is a lack in precision ,where as in Least Load Algorithm , the load at particular server is considered before distributing it to the server , which results in decreasing efficiency of the system when only a set of servers gets load in case of sparse load condition thereby leaving some of the systems idol.

Properties	Round Robin	Least load
Dynamic/Static	Static	Dynamic
Stability	High	Medium
Cooperative	No	Yes
Resource Utilization	Low	Medium

Table 1:Comparison of Load Balancing Algorithms [23]

Properties	Round Robin	Weighted Round Robin	Improved Weighted Round Robin	Least Load
Dynamic/Static	Static	Static	Static/ Dynamic	Dynamic
Stability	High	High	Higher	Medium
Cooperative	No	No	No	Yes
Resource Utilization	Low	Medium	High	Medium

Table 2:Comparison among Round Robin ,Weighted Round Robin,IWRR ,Least Load

Hybrid of IWRR algorithm and Least Load Algorithm

Following features can be considered in the hybrid of two:

1)Before sending the request to the corresponding server ,it considers the server load.

2)Oversee the previously selected server

3)In the next server load decision this helps the selected server not to participate.

4)The sparse requests can be easily distributed among the n servers, more evenly .

5)Also , IWRR is both static and dynamic as well as it will consider the length of the tasks with its priority, processor capacity

```
BEGIN PROCEDURE HYBRID ALGO
SET static integer PRIORITY=0
SET static integer TASK_LENGTH=0
SET array SERVERS={S1,S2,S3,S4,…SN}
SET array SERVER_LOAD={L1,L2,L3….,Ln}
WHILE (request) DO
IntegerPOS={FIND(FIND_MIN-H(SERVER_LOAD,TASK_LENGTH))(FIND_MAX-H(PRIORITY))}
TASK_LENGTH+PRIORITY=POS;
GOTOSERVER[POS];
END WHILE
END PROCEDURE
BEGIN PROCEDURE FIND(FIND_MIN-H(SERVER_LOAD(1 TO N),TASK_LENGTH))(FIND_MAX-H(PRIORITY))
SET integer POS =0;
For I =1 to n-1 do
If I=PRIORITY+TASK_LENGTH THEN
CONTINUE;
END IF
ELSE IF SERVER_LOAD [POS]>SERVER_LOAD [I] THEN
POS=I;
END ELSE IF
RETURN POS;
END PROCEDURE
```

RESULT AND ANALYSIS

	Total Data (MB)	Request/sec	Bandwidth (Bytes/sec)	Busy Threads
Main server	168	5.78	480.23	8
Server 1	66	2.89	151.678	2
Server 2	78	3.032	167.35	4
Server 3	99	2.732	154.897	3

TABLE 3 ROUND ROBIN[23]

	Total Data (MB)	Request/sec	Bandwidth (Bytes/sec)	Busy Threads
Main server	189	6.078	512.899	7
Server 1	57	3.189	168	3
Server 2	65	3.032	176.789	4
Server 3	57	3.94	164.234	3

TABLE4 LEAST LOAD[23]

	Total Data (MB)	Request/sec	Bandwidth (Bytes/sec)	Busy Threads
Main server	180	8.67	497.567	7
Server 1	57	4.78	170	4
Server 2	63	3.986	168.095	4
Server 3	69	4.013	164.234	3

TABLE 5 HYBRID OF ROUND ROBIN AND LEAST LOAD[23]

	Total Data(MB)	Request/ sec	Bandwidth(Byte/sec)	Busy threads
Main Server	189	9.6	498.7	9
Server1	57	5.1	175	3
Server2	63	4.0	168	3
Server3	69	4.01	164.2	3

TABLE 6: HYBRID OF IMPROVED WEIGHTED ROUND ROBIN AND LEAST LOAD[ASSUMED]

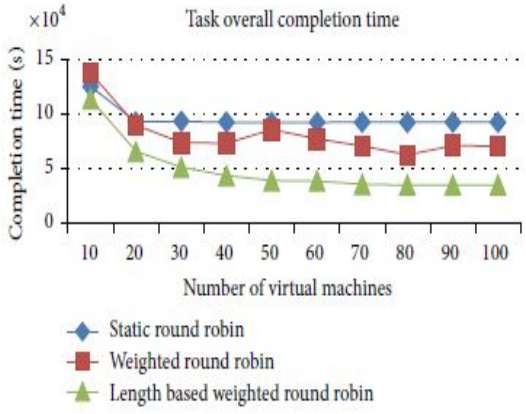

Figure 3[9]: Execution Completion time

Above tables shows the comparisons of all algorithms whether single or hybrid. The first table data represents that the main server has 8 busy threads where as server 1 has only 2 threads that means the threads are not evenly distributed. Table 2 represents the least load algorithm , which shows that as compared to round robin ,least load algorithm is much better. Table 3 is the hybrid of the two which gives the better result as proved by Rashmi Saini.et al. Table 4 and figure3 shows the efficiency of IWRR algorithm. Table 4 represents that using IWRR algorithm and Least Load algorithm as a hybrid will give better results.

CONCLUSION:

From the above tables and graphs , and discussed theory of IWRR algorithm, it is concluded that a hybrid of IWRR and Least Load Algorithm will give better results as compared to a hybrid of RR and Least Load Algorithm or simple RR or Least Load Algorithm as in IWRR algorithm both Length of task and priority is also considered, which leads to better results.

REFERENCES:

[1] Barazandeh and I.Mortazavi. S.*S " Dynamic Load Balancing algorithms in Distributed Systems"* Proceedings of 2009 second International Conference on Computer and Electrical engineering pp 516-521

[2] B.Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," *Procedia Technology*, vol. 4, pp. 783–789, 2012

[3] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software*, vol. 42, no. 7, pp. 419–425, 2011.

[4] Cardellini, Colajanni, and Philip S. Yu, ―*Dynamic Load balancing on web-server systems,* published in IEEE internet Computing, vol. 3, no. 3, pp 28-39, 1999.

[5] C. Lin and S. Lu, "Scheduling scientific workflows elastically for cloud computing," in *Proceedings of the IEEE 4th International Conference on Cloud Computing*, Washington, DC, USA, July 2011.

[6] C. Lin, S. Lu, X. Fei et al., "A reference architecture for scientific workflow management systems and the VIEW SOA solution," *IEEE Transactions on Services Computing*, vol. 2, no. 1, pp. 79– 92, 2009.

[7] Daniel A.Menascé, George Mason University *"Trade-offs in Designing Web Clusters"* IEEE Internet Computing1089- 7801/ 02 ©2002 IEEE.

[8] O.K.Tonguz and E.Yanmaz, ―*On the Theory of Dynamic Load Balancing*, in Proc. IEEE Global Telecom. Conf. (GLOBECOM'03), vol. 7, pp. 3626-3630, Dec. 2003.

[9] D.Chitra Devi and V.Rhymend Uthariaraj," Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks", Hindawi Publishing Corporation ,The Scientific World Journal ,Volume 2016, Article ID 3896065, 14 pages.

[10] D. Grosu, A. T. Chronopoulos and M. Y. Leung, "Cooperative Load Balancing in Distributed Systems", *Journal of Parallel and Distributed Computing*, Submitted 2002.

[11] F. Xhafa and A. Abraham, *Mata-Heuristics for Grid Scheduling Problems*, Springer, Berlin, Germany, 2008*2*

[12] G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, "Scheduling of scientific workflows using a chaosgenetic algorithm," *Procedia Computer Science*, vol. 1, no. 1, pp. 1445–1454, 2010, International Conference on Computational Science, ICCS 2010.

[13] Harikesh Singh, Dr. Shishir Kumar *"Dispatcher Based Dynamic Load Balancing on Web Server System*, International Journal of Grid and Distributed Computing, Vol. 4, No. 3, September, 2011.

[14] H. M. Fard and H. Deldari, "An economic approach for scheduling dependent tasks in grid computing," in *Proceedings of the 11th IEEE International Conference on Computational Science and Engineering (CSEWorkshops '08)*, pp. 71–76, IEEE, San Paulo, Brazil, July 2008.

[15] Hua-fengdeng, yun - sheng liu, ying-yuanxiao. *" A novel algorithm for load balancing in distributed systems"* proceedings of the 2007 eight ACIS International Conference on software engineering, Artificial Intelligence, Networking and parallel/distributed computing, pp 15-19.

[16] H.W. D. Chang andW. J. B. Oldham. Dynamic task allocation models for large distributed computing systems. *IEEE Trans. Parallel and Distributed Syst.*, 6(12):1301–1315, December 1995.

[17] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 45–58, 2014.

[18] L. D. Dhinesh Babu and P.Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2292–2303, 2013.

[19] L.-T. Lee, C.-W. Chen, H.-Y. Chang, C.-C. Tang, and K.-C. Pan, "A non-critical path earliest-finish algorithm for interdependent tasks in heterogeneous computing environments," in *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC '09)*, pp. 603–608, Seoul, Republic of Korea, June 2009.

.[20] M. Rahman, R. Hassan, R. Ranjan, and R. Buyya, "Adaptive workflow scheduling for dynamic grid and cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 13, pp. 1816–1842, 2013.

[21] M.Xu, L. Cui, H. Wang, and Y. Bi, "AmultipleQoS constrained scheduling strategy of multiple workflows for cloud computing," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '09)*, pp. 629–634, IEEE, Chengdu, China, August 2009.

[22] Neeraj Nehra, R.B. Patel, V.K .*Bhat "A multi-agent system for distributed dynamic load balancing on cluster."* Proceedings of the International Conference on Advanced Computing and communications, 2006, pp 135-138.

[23]Rashmi Saini and Ashish Bisht, "A Hybrid Algorithm for Load Balancing" *International Journal of Advanced Research in Computer Science and Software Engineering 5(7), July- 2015, pp. 1-6*

[24] R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1787–1796, 2014.

[25] R. Basker, V. Rhymend Uthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," *International Journal of Recent Advance in Engineering & Technology*, vol. 2, no. 3, pp. 81–86, 2014.

[26] S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," in *Proceedings of the International Conference on Advances Science and Contemporary Engineering*, pp. 778–785, October 2012.

[27] S. Ijaz, E. U.Munir,W. Anwar, andW. Nasir, "Efficient scheduling strategy for task graphs in heterogeneous computing environment," *The International Arab Journal of Information Technology*, vol. 10, no. 5, 2013.

[28] Vijindra and S. Shenai, "Survey on scheduling issues in cloud computing," *Procedia Engineering*, vol. 38, pp. 2881–2888, 2012,

Proceedings of the International Conference on Modelling Optimization and Computing

[29] W. Kadri, B. Yagoubi, and M. Meddeber, "Efficient dependent tasks assignment algorithm for grid computing environment," in *Proceedings of the 2nd International Symposium onModelling and Implementation of Complex Systems (MISC '12)*, Constantine, Algeria, May 2012.

.[30] Y. C. Chow and W. H. Kohler. Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Trans. Comput.*, C-28(5):354–361, May 1979.

[31] Y.Xu,K. Li, L.He, and T. K. Truong, "ADAGscheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization," *Journal of Parallel andDistributed Computing*, vol. 73,no. 9, pp. 1306–1322, 2013.

[32] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107–1117, 2013.

[33] Z. Yu, F. Menng, and H. Chen, "An efficient list scheduling algorithm of dependent task in grid," in *Proceedings of the 3rd IEEE International Conference on Computer Science and*
Information Technology (ICCSIT '10), IEEE, Chengdu, China, July 2010.