International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 10, October 2016
www.ijates.com
ijates
ISSN 2348 - 7550

CLOUD BASED MULTISERVICE LOAD BALANCING APPROACH FOR MULTIMEDIA SYSTEM

P.Mounika[1], D. Nethaji Babu[2], Prof. S.V. Achutha Rao[3]

[1]Pursuing M.Tech (CSE), [2]Assistant Professor, [3]Professor & Head, Department of CSE

[1,2,3]Vikas College of Engineering & Technology, Nunna, Vijayawada, AP, Affiliated to JNTUK, (India)

ABSTRACT

Present day's world, Internet is very popular technology. Each day about 400 billion users introduced with internet. With this network traffic is increasing simultaneously over the internet. So, it is mandatory to control the large traffic over internet. All the information of internet is been storing on cloud. Existing cloud is able to store only text type of data were as proposed cloud can store huge amount of various types of data. This Load balancing mechanism is achieved by using centralized hierarchical cloud-based multimedia system consists a resource manager, server cluster and cluster heads. Resource manager permits clients requests for multimedia service tasks and assigns server clusters based on the task features. Based on data size cluster head divides the task to the various servers within its server cluster. DES algorithm is used for load balancing that unroll the multimedia service task load on servers with the low cost for sending information among server clusters and clients without breaking the maximum load limit of each one of server cluster. This is effective load balancing system not only balancing the load among servers but also it assigns the data consistently and uniformly by checking various parameters. Here balancing the load from heavy loaded severs to low loaded server hence response time from the server will reduce and performance for system will improve.

I. INTRODUCTION

With development of technology, cloud-based media framework (CMS) shows up on account of a bigger number of clients' requests for diverse sight and sound registering and stockpiling administrations through the Internet in the meantime. It for the most part not corporate framework, stages, and programming to bolster a substantial number of customers all the while to store and process their sight and sound application information in a gives way and meet diverse interactive media QoS prerequisites through the Internet. Most sight and sound applications (e.g., sound/video gushing services, and so forth.) require impressive calculation, and are frequently performed on cell phones with obliged force, so that backing of distributed computing is emphatically required. All in all, cloud employments suppliers offer the utilities taking into account cloud offices to clients, with the goal that clients don't have to take much cost to demand media occupations and procedure sight and sound information and additionally their computation results. Thusly, mixed media applications are prepared on intense cloud servers, and the customers just need to pay for the used assets when. This paper considering an incorporated various leveled CMS made out of an asset server and various server groups, each of which is composed by a bunch head, and we expect the servers in distinctive server groups to give diverse servers. Such a CMS is worked as takes after. Each time when the CMS gets clients solicitations for mixed media servers errands, the asset administrator of the CMS allots those undertaking solicitations to distinctive server groups as indicated by the qualities of the asked for assignments. In this manner, the bunch leader of every server group disperses the relegated assignment to some server inside of the server group. It is not difficult to watch that the heap of every server bunch fundamentally influences the execution of the entire CMS. By and large, the asset director of the CMS ais in quest for decently disseminating the assignment load crosswise over server bunches, and consequently, it is of significance and enthusiasm to have the capacity to adapt to load adjusting in the CMS.

II. SYSTEM ARCTECHTURE

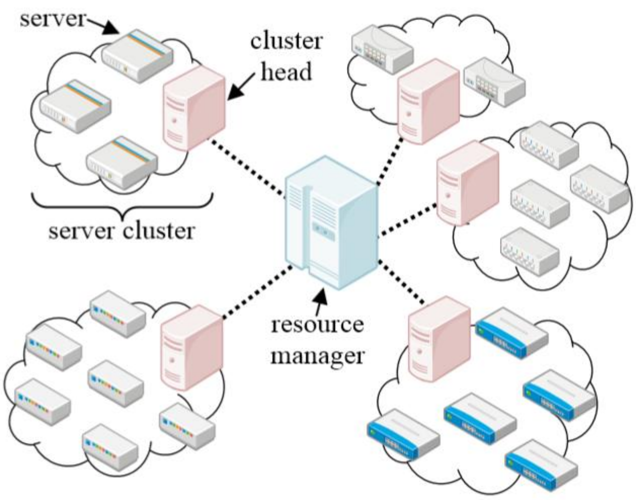


Fig: Illustration of hierarchical cloud-based multimedia system

III. RELATED WORKS

Submit Load adjusting for remote systems has been concentrated comprehensively in the before writing, e.g., Effective burden adjusting for cloud-based interactive media framework, Optimal asset portion for sight and sound cloud in view of lining model, Among them, the heap adjusting issue for CMSs in viable burden adjusting in cloud-based mixed media framework is exasperates with circulation the mixed media administration errand load on servers with the negligible expense for transmitting mixed media information between server groups and customers, while the maximal burden breaking point of every server bunch is not raided. A simple worry in their setting is to assume that all the interactive media administration errands are of the same sort. Practically speaking, then again, the CMS offer administrations of creating, altering, handling, and looking an assorted qualities of mixed media information, e.g., hypertext, pictures, video, sound, design, etc. Distinctive interactive media administrations have different prerequisites for the capacities gave by the CMS (stockpiling, focal preparing unit, and design handling unit bunches), e.g., the QoS necessity of hypertext site page administrations is looser than that of video spilling administrations. What's more, the settings in the past works [did not consider that heap adjusting ought to adjust to the time change. To react to the commonsense necessities said above, to accept that in the CMS, every server group can just switch a particular sort of sight and sound administration errands, and every customer demands an alternate kind of interactive media administrations at diverse time. At every particular time step, such an issue can be displayed as a whole number direct programming definition, which is computationally obstinate when all is said in done. Generally, obstinate issues are typically unraveled

ijates
ISSN 2348 - 7550

by met heuristic methodologies, e.g., recreated strengthening, hereditary calculation, molecule swarm enhancement and so forth. In CMSs, to propose a hereditary calculation (GA) for the concerned element

Problem Formulation to formulate the CMS that can adapt to time dynamics, we assume time to be divided into different time steps. At the t^{th} time step, the CMS can be modelled as a complete weighted bipartite graph Gt = (U, V, E, φ,ψ^t,q,r^t,w^t) in which

• U is the set of vertices that represent the server clusters of the CMS;

• V is the set of vertices that represent clients;

• E is the set of edges between U and V , in which each edge eij ∈ E represents the link between server cluster i ∈ U and client j ∈ V ;

• φ : U →Nis a function used to restrict that server cluster i can only cope with multimedia tasks of type φ^i;

• ψ^t : V → N is a function used to represent that client j requests the multimedia service of type ψ^t_j at the t-th time step;

• q : U∪V →N is a function used to represent that server cluster i can provide the multimedia service of QoS qi;

• r^t : U∪V →N is a function used to represent that client j requests the multimedia service of QoS requirement r^t_j at the t-th time step;

• w^t : E → R+ is the weight function associated with edges, in which wt ij denotes the wt value that represents the cost for transmitting multimedia data between server cluster i and client j at the t-th time step, which is defined as follows:

$$w_{ij}^t = \begin{cases} \infty, & \text{if } d_{ij}^t \to \infty \text{ or } \phi_i \neq \psi_j; \\ d_{ij}^t l_{ij}^t, & \text{otherwise.} \end{cases} \quad (1)$$

Where d^t_{ij} is the network proximity between server cluster i and client j; l^t_{ij} is the traffic load of the link betweenserver cluster i and client j that is defined as follows:

$$l_{ij}^t = \sum_{k \in K_i} u_{ikj}^t C_{ik} \quad (2)$$

Where Ki is the set of servers in server cluster i; utikj is the server utilization ratio of server k in server cluster i due to client j, and Cik is its capacity. Note that the proximity dtij between server cluster i and client j in Equation (1) is required to be measured at every time step due to dynamic change of network topology. This paper continues applying the setting of based upon the distributed binning scheme to calculate the proximity dtij. Like other previous works, we measure the proximity between the server cluster and the client as a distance between them. Take an example to explain how to calculate the proximity as follows. Here, we say that a node may be a server cluster or a client. First, we measure the distance of a node to a given set of landmark nodes in

ijates
ISSN 2348 - 7550

the network by the network link latency. Suppose that there are three landmarks in the network. The latencies from the concerned node to the three landmarks are 45, 10, and 25, respectively. Nodes are ranked according to the latency information: range 0 for latencies in [0, 15], range 1 for latencies in (15, 40], and range 2 for latencies higher than 40. Hence, the landmark order of the concerned node is "201". By using the landmark order, all the nodes can be classified into different bins, i.e., the nodes with the same landmark order fall into the same bin. By doing so, we only calculate the proximity between two nodes in the same bin, while the others in different bins mean that they are too far to communicate with each other, so their proximity is infinity. With the above notations, the mathematical model of our concerned problem at the t-th time step can be stated as the following integer linear programming formulation

$$\text{Minimize} \quad \lambda \frac{\sum_{i\in U}\sum_{j\in V} x_{ij}^t w_{ij}^t}{\sum_{j\in V} w_{\max}} + (1-\lambda)\left(1-\frac{\sum_{j\in V}\sum_{i\in U} x_{ij}^t}{|V|}\right) \quad (3)$$

$$\text{subject to} \quad \sum_{i\in U} x_{ij}^t \le 1, \forall j \in V, \quad (4)$$

$$\sum_{j\in V} x_{ij}^t l_{ij}^t \le \sum_{k\in K_i} C_{ik}, \forall i \in U \quad (5)$$

$$x_{ij}^t \phi_i = x_{ij}^t \psi_j^t, \forall i \in U, j \in V \quad (6)$$

$$x_{ij}^t q_i \ge x_{ij}^t r_j^t, \forall i \in U, j \in V \quad (7)$$

$$x_{ij}^t \in \{0,1\}, \forall i \in U, j \in V \quad (8)$$

Where x^t_{ij} is an indicator variable defined as follows:

$$x_{ij}^t = \begin{cases} 1, & \text{if client } j \text{ is assigned to server cluster } i \\ & \text{at the } t\text{-th time step;} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In the above model, indicator variable x^t_{ij} (see Equation (8)) is used to determine whether to assign the link e_{ij} between server cluster i and client j in the complete bipartite graph U×V .The objective (3) of the model is a weighted sum of two terms: the first is to minimize the total weighted values of the

Bipartite graph, i.e., to minimize the total cost of transmitting multimedia data at the t-th time step, while the second is to maximize the number of link assignments. Note that we let wmax be the maximal possible weight (less than infinity), and hence, the denominators of the two terms of the objective are used for normalizing them to the range [0,1], and $\lambda \in [0,1]$ is used to adjust the weights of the two terms, so that the objective value always falls into the range [0,1]. Constraint (4) guarantees that each client only allows at most one link to be assigned. For each client j in V, the constraint enforces that x^t_{ij} of at most one server cluster i is 1. Constraint (5) enforces that the utilized capacity of each server cluster cannot exceed its capacity at the t-th time step. Constraint(6)enforces that the multimedia service type requested by each client j is consistent with that provided by server cluster i. Constraint (7) enforces that each client j requests the multimedia server of the QoS no more than that offered by server cluster i. As our model is rooted from the work in [3], the differences of our model from theirs are explained as follows.

• Different from the work in [3], we additionally consider four functions φ, ψ^t, q and rt.

• About the link assignment to each client, the model in [3] constrains each client to be assigned to exactly one link, while ours allows each client to be assigned to one or zero link (Constraint (4)). That is, the previous model guarantees to serve each client, but ours does not, because our concerned problem is more complicated.

• With the above constraint, our objective additionally considers to maximize the number of link assignments, i.e., the number of served clients (see the second term in Objective (3)).

• Our model additionally considers Constraints (6) and (7) for multiple service types and QoS requirements, respectively.

• The load balancing algorithm in [3] is not adaptive, but ours is robust with time change, as the time change can be seen via the superscript t in the model.

• We allow mobility of clients, i.e., clients can change their locations at different time steps. Note that the problems that consider mobility of nodes have received much attention recently, e.g., see the survey in.

As a result, our concerned problem can be stated as follows:

IV. DYNAMIC MULTI-SERVICE LOAD BALANCING IN CMS (CMS-DYNMLB)

Given a CMS with m server clusters and n clients, for t = 1,2,···, the bipartite graph $G_t = (U, V, E, \varphi, \psi^t, q, r^t, w^t)$ underlies the CMS at the t-th time step (as described above) in which clients have mobility, while the link between clients and server clusters need be assigned. The objective of the problem is to assign multimedia service load so that total cost of transmitting multimedia data is minimized and the number of served clients is maximized.

Since the CMS-dynMLB problem at each fixed time step can be modelled as an integer linear programming problem as mentioned above, it is computationally intractable in general, i.e., there does not exist any efficient deterministic polynomial time algorithm for the problem. Hence, this paper proposes a genetic algorithm (GA) with immigrant scheme for solving the problem. The GA is a stochastic global search method that has proved to be successful for many kinds of optimization problems. GA is categorized as a global search heuristic. It works with a population of candidate solutions and tries to optimize the answer by using three basic principles, including selection, crossover, and mutation. For more details on GA, readers are referred to.

ijates
ISSN 2348 - 7550

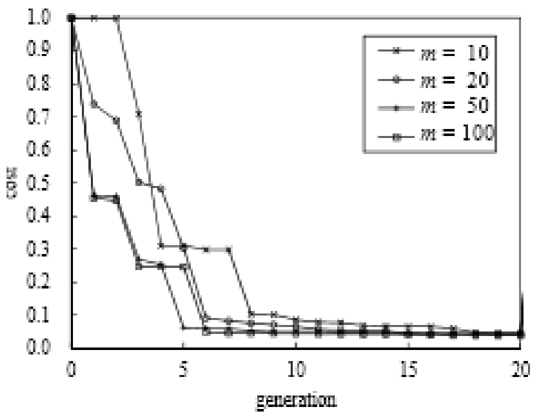
Algorithm 1 Dynamic Load Balancing Algorithm
for t = 1, 2, · · · do
create complete weighted bipartite graph G_t
remove the edges in G_t violating (6) and (7)
calculate $\{l_{ij}\}$ load and weight $\{w^t_{ij}\}$ on server cluster i by client j by calling Algorithm2
assign $\{x^t_{ij}\}$ by calling Algorithm 3
end for

4.1 DES Algorithm

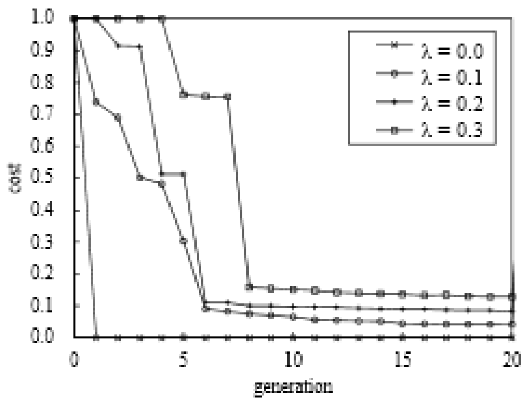
The main parts of the algorithm are as follows:

- Fractioning of the text into 64-bit (8 octet) blocks;
- Initial permutation of blocks;
- Breakdown of the blocks into two parts: left and right, named *L* and *R*;
- Permutation and substitution steps repeated 16 times (called **rounds**);
- Re-joining of the left and right parts then inverse initial permutation.

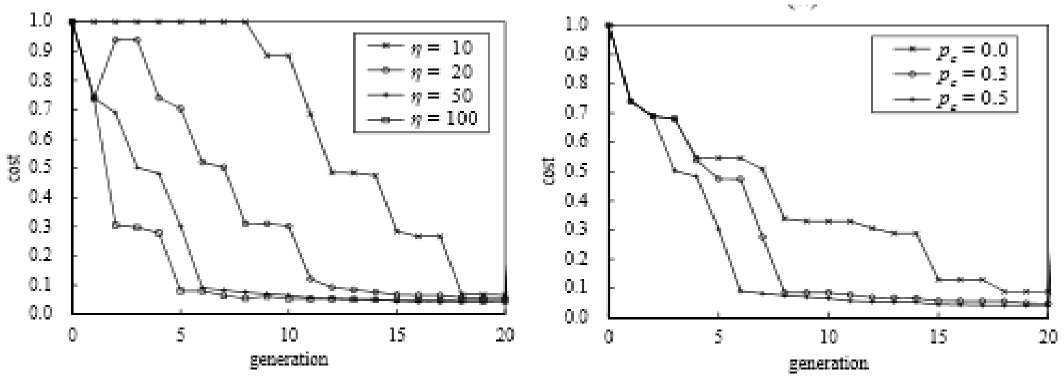
4.2 Experimental Results



(a)



(b)



(c) (d)

V. CONCLUSION

A dynamic encryption standard for optimizing the cloud based multimedia dynamic multiservice load balancing is proposed and implemented. Major difference between this model and previous model is , existing model will not allow to store all type of data except text docs where as proposed model allows multiple types of data to store on server and major part we implemented on storing data load balancing with multiple servers.

REFERENCES

[1] W. Zhu, C. Luo, J. Wang, and S. Li, “Multimedia cloud computing: An emerging technology for providing multimedia services and applications,” IEEE Signal Processing Magazine, vol. 28, no. 3, pp. 59–69, 2011.

[2] C.-F. Lai, Y .-M. Huang, and H.-C. Chao, “DLNA-based multimedia sharing system over OSGI framework with extension to P2P network,” IEEE Systems Journal, vol. 4, no. 2, pp. 262–270, 2010.

[3] W. Hui, H. Zhao, C. Lin, and Y. Yang, “Effective load balancing for cloud-based multimedia system,” in Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology. IEEE Press, 2011, pp. 165–168.

[4] C.-Y. Chen, H.- C.Chao, S.- Y.Kuo, and K.-D. Chang, “Rule-based intrusion detection mechanism for IP multimedia subsystem,” Journal of Internet Technology, vol. 9, no. 5, pp. 329–336, 2008.

[5] L. J. Wu, A. E. AL Sabbagh, K. Sandrasegaran, M. Elkashlan, and C. C. Lin, “Performance evaluation on common radio resource management algorithms,” in Proceedings of 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2010). IEEE Press, 2010, pp. 491–495.

[6] R. Yavatkar, D. Pendarakis, and R. Guerin, “A framework for policy based admission control,” Internet Requests for Comments, RFC Editor, RFC 2753, 2000.

[7] D. Niyato and E. Hossain, “Integration of WiMAX and WiFi: Optimal pricingforbandwidthsharing,” IEEE Communication Magazine,vol.45, no. 5, pp. 140–146, 2007.

[8] C.-Y. Chang, T.-Y.Wu, C.-C. Huang, A. J.-W. Whang, and H.-C. Chao, “Robust header compression with load balance and dynamic bandwidth aggregation capabilities in WLAN,” Journal of Internet Technology, vol. 8, no. 3, pp. 365–372, 2007.

[9] J. Sun, X. Wu, and X. Sha, "Load balancing algorithm with multiservice in heterogeneous wireless networks," in Proceedings of 6th International ICST Conference on Communications and Networking in China (China Com 2011). IEEE Press, 2011, pp. 703–707.

[10] H. Son, S. Lee, S.- C.Kim, and Y.-S. Shin, "Soft load balancing over heterogeneous wireless networks," IEEE Transactions on Vehichular Technology, vol. 57, no. 4, pp. 2632–2638, 2008.

[11] L. Zhou, H.-C. Chao, and A. V. Vasilakos, "Joint forensics-scheduling strategy for delay-sensitive multimedia applications over heterogeneous networks," IEEE Journal on Selected Areas of Communications, vol. 29, no. 7, pp. 1358–1367, 2011.

[12] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in Proceedings of 2011 IEEE 13th International Workshop on Multimedia Signal Processing (MMSP 2011). IEEE Press, 2011, pp. 1–6.

[13] M. Garey and D. Johnson, Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979.

[14] S. Kirkpatrik, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671–680, 1983.

[15] J. H. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of IEEE International Conference on Neural Networks. IEEE Press, 1995, p. 1942V1948.

[16] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Proceedings of IEEE International Conference on Evolutionary Computation. IEEE Press, 1998, pp. 69–73.

[17] X. Zhang, S. Hu, D. Chen, and X. Li, "Fast covariance matching with fuzzy genetic algorithm," IEEE Transactions on Industrial Engineering, vol. 8, no. 1, pp. 148–157, 2012

Author Details

	**P. Mounika** pursuing M.Tech(CSE)(13NQ1D5814)(2013-2015)from Vikas College of Engineering & Technology, Nunna, Vijayawada, Krishna (D)-521212, Andhra Pradesh, Affiliated to JNTUK, India.
	**Dakavarapu Nethaji Babu** M. Tech (CSE),(Ph.D) at Acharya Nagarjuna University. Is working as Assistant Professor, Department of (CSE) from Vikas College Of Engineering & Technology, Nunna, Vijayawada, Krishna (D)-521212, Andhra Pradesh, Affiliated to JNTUK, India.
	**Prof S.V. Achutha Rao**, is working as a HOD of CSE at Vikas College of Engineering and Technology, Nunna, Vijayawada, Affiliated to JNTU, Kakinada, A.P., India