



# A FRAMEWORK FOR DYNAMIC MULTITASK LOAD EQUILIBRATING IN CLOUD BASED MULTIMEDIA SYSTEM

**K. Raja Rao<sup>1</sup>, Dr. M.V.Bramhananda Reddy<sup>2</sup>**

<sup>1</sup> Pursuing M.Tech (CSE), <sup>2</sup>Working as Professor & Head of the Department (CSE),

Nalanda Institute of Engineering & Technology (NIET), Kantepudi(V),

Sattenpalli(M), Guntur(D)-, Andhra Pradesh (India)

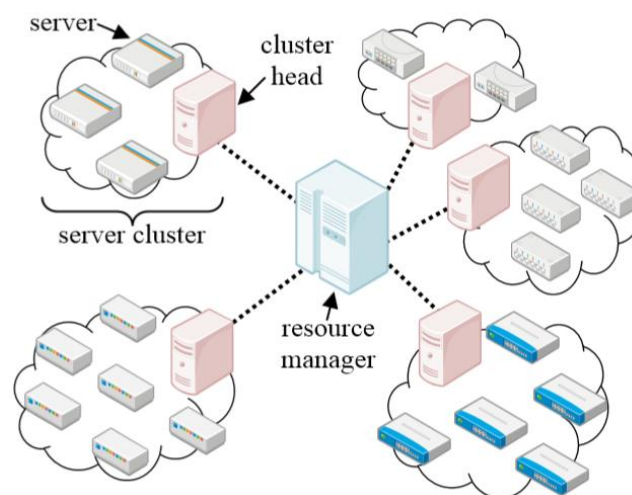
## ABSTRACT

*In this multimedia system we are using centralized hierarchical cloud. Centralized hierarchical cloud based multimedia system (CMS) contains resources manager, server clusters and cluster heads, from these resource manager which appoint clients' request for multimedia service task to server clusters as per the task characteristics, and each cluster head decides to distribute the allocated task to the servers with in its server cluster. Though it is complex CMS- cloud based multimedia system, it is a research provocation to construct an efficient load balancing algorithm which unroll the multimedia service task load to servers with the low cost for sending data between server cluster and client, without violating the maximum load limit of every server cluster. Apart from previous works, this paper takes into a more experimental dynamic multi service framework in which every server cluster only control a particular type of multimedia tasks, and every client request a various types of multimedia services at different instances. Such a scenario can be formed as an integer linear programming problem. This is computationally collaborative in general. This is been to be acceptable for dynamic problems. Simulation results demonstrate that the enhanced genetic algorithm can coherently subsist with dynamic multi-service load balancing in CMS.*

## I. INTRODUCTION

Cloud-based multimedia system (CMS) emerged because of a large number of users' request for different multimedia computing and storage services through the Internet at the same time, It generally integrate infrastructure, platforms, and software to assist a large number of clients concurrently to store and process their multimedia application information in a distributed manner and meet various multimedia QoS requirements through the Internet. Most multimedia applications (e.g. audio/ video streaming services, etc.) need significant computation, and are often performed on mobile devices with constrained power, so that the reinforcement of cloud computing is strongly required. In general, cloud service furnishes offer the benefits based on cloud facilities to clients, so that clients do not need to take high cost to request multimedia services and process multimedia information as well as their calculation results. Thus, multimedia applications are organized on powerful cloud servers, and the clients only need to pay in ordered get the useful resources by the time.

This enhanced paper considers a centralized hierarchical cloud-based multimedia system (CMS) as shown in below figure, formulate of a resource manager and a number of server clusters, each of which is coordinated by a cluster head, and we assume the servers in several server clusters to process different services. Such a CMS (Cloud-based multimedia service) is controlled as follows. Every time when the CMS receives clients' requests for multimedia service tasks, the resource manager of the CMS allocate those task requests to various server clusters according to the characteristics of the requested work. Eventually, the cluster head of every server cluster dispense the allocated task to few servers within the server cluster. It is not tough to distinguish that the load of every one server cluster eventually affects the performance of the entire CMS. In common, the resource manager of the CMS is in pursuit of fairly dispensing the task load over server clusters, and hence, it is of significance and interest to be able to manage with load balancing in the CMS. Load stabilizing for wireless networks has been deliberated extensively in the previous literature, e.g., multiple-factor load balancing, load balancing with strategy mechanism, load balancing based on game theory, load balancing in WLANs, multi-service load balancing and soft load balancing, and scheduling in heterogeneous wireless networks between others. Some previous works have also prevailed on load balancing for CMSs. Among them, the load balancing problem for CMSs in is concerned with spreading the multimedia service task load on servers with the low cost for transmitting multimedia information between server clusters and clients, while the high load limit of each server cluster is not violated. A simplified concern in their setting is to imagine that all the multimedia service tasks are of the similar type. In implementation, however, the CMS offers services of provoking, editing, organizing, and searching a diversity of multimedia information, e.g., hypertext, images, video, audio, graphics, and so on. Different multimedia services have various requirements for the functions provided by the CMS e.g., the QoS requirement of hypertext webpage services is unattached than that of video streaming services. moreover, the settings in the previous works did not contemplate that load balancing should adjust to the time change.



**Fig: Illustration of Hierarchical Cloud-Based Multimedia System**

To respond to the practical requirements mentioned above, let us assume that in the CMS, each server cluster can only control a significant type of multimedia service tasks, and every one client requests a different type of multimedia services at different instant. At each specific time step, such a problem can be figured as an integer linear programming emergence, which is computationally unmanageable in general. Conventionally,



unmanageable problems are normally unfolded by Meta heuristic approaches, e.g., replicated annealing, genetic algorithm, particle swarm optimization, etc. In this paper, we propose a genetic algorithm (GA) for the concerned dynamic load stabilizing problem for CMSs. GA has already initiated applications in a variety of areas in computer science and engineering, such as fast covariance matching, aircraft ground service scheduling problem, optimal electric network design, among others. In our setting of GA, best immigrants and random immigrants are appending to new population, because they are acceptable for solving the problems in dynamic environments. The fact-finding results show that to a certain extent, our approach is competent of dynamically expanding the multimedia task load evenly. Note that some previous works on other issues of cloud computing or dispense computing have also existed, e.g., cost-optimal scheduling on clouds, load balancing for distributed multi agent computing , communication-aware load balancing for parallel applications on clusters, among others. Also note that GA has been appeal to dynamic load balancing in, but their GA was designed for distributed systems, not particular to the cloud multimedia systems. In addition, they did have any multi-service concern.

**PROBLEM DESCRIPTION:** Our load balancing problem for the CMS is based upon, which, however, only considered that all the multimedia service tasks are of the similar type, and did not consider the dynamic scenario where load balancing should adapt to the time vary. By enhancing their model with these concerns, this section first gives the system overview of the CMS, and then prepares our apprehensive problem.

CMSs can be split into two categories centralized and decentralized. This paper considers a centralized CMS, which contains a resource manager and a no. of server clusters each of which is coordinated by a cluster head. Vary from the decentralized cloud monitoring system, every time when receiving clients' requests for multimedia service tasks, the resource manager of the CMS stores the global service task load information collected from server clusters, and declare the amount of client's requests allocated to each server cluster so that the load of each server cluster is dispensed as balanced as possible, in term of the cost of distributing multimedia data between server clusters and clients. The resolution of function is depend upon the characteristics of different service requests and the information composed from server clusters. In contrast to decentralized framework, the centralized framework is scalable as fewer overheads are urged on the system, thus, a lot of applications have existed, e.g., see. However, the centralized framework has lower accuracy since the load balancing algorithms may be flawed due to the failure of the resource manager. Although a decentralized framework is suitable to lower systems, it is still easier to implement.

**Problem Formulation** to formulate the CMS that can adapt to time dynamics, we assume time to be divided into different time steps. At the  $t^{\text{th}}$  time step, the CMS can be modelled as a complete weighted bipartite graph  $G_t = (U, V, E, \varphi, \psi^t, q, r^t, w^t)$  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  $e_{ij} \in E$  represents the link between server cluster  $i \in U$  and client  $j \in V$  ;
- $\varphi : U \rightarrow N$  is a function used to restrict that server cluster  $i$  can only cope with multimedia tasks of type  $\varphi^i$ ;
- $\psi^t : V \rightarrow N$  is a function used to represent that client  $j$  requests the multimedia service of type  $\psi_j^t$  at the  $t$ -th time step;



- $q : U \cup V \rightarrow N$  is a function used to represent that server cluster  $i$  can provide the multimedia service of QoS  $q_i$ ;
- $r^t : U \cup V \rightarrow N$  is a function used to represent that client  $j$  requests the multimedia service of QoS requirement  $r_j^t$  at the  $t$ -th time step;
- $w^t : E \rightarrow R^+$  is the weight function associated with edges, in which  $w_{ij}^t$  denotes the  $w_{ij}^t$  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:

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

Where  $d_{ij}^t$  is the network proximity between server cluster  $i$  and client  $j$ ;  $l_{ij}^t$  is the traffic load of the link between server 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  $K_i$  is the set of servers in server cluster  $i$ ;  $u_{ikj}^t$  is the server utilization ratio of server  $k$  in server cluster  $i$  due to client  $j$ , and  $C_{ik}$  is its capacity. Note that the proximity  $d_{ij}^t$  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  $d_{ij}^t$ . 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 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

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



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

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

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

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

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

where  $x_{ij}^t$  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  $x_{ij}^t$  (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  $U \times V$ . 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  $w_{max}$  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  $x_{ij}^t$  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  $\phi$ ,  $\psi^t$ ,  $q$  and  $r^t$ .
- 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:

**DYNAMIC MULTI-SERVICE LOAD BALANCING IN CMS (CMS-DYNMLB):** Given a CMS with  $m$  server clusters and  $n$  clients, for  $t = 1, 2, \dots$ , the bipartite graph  $G_t = (U, V, E, \phi, \psi^t, q, r^t, w^t)$  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-dyn MLB 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.

#### Algorithm 1: DYNAMIC LOAD BALANCING ALGORITHM

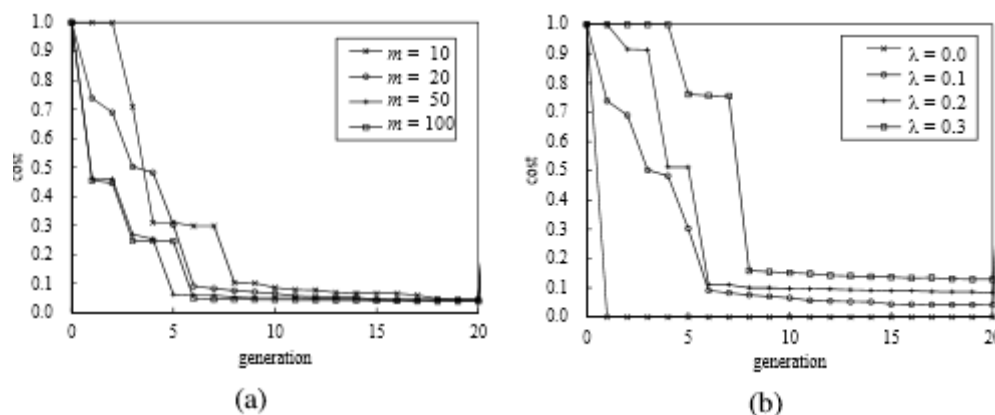
- 1: for  $t = 1, 2, \dots$  do
- 2: consider complete weighted bipartite graph  $G_t$
- 3: remove the links in  $G_t$  violating Constraints (6) and (7)
- 4: calculate  $\{l_{ij}^t\}$  and  $\{w_{ij}^t\}$  by calling Algorithm 2
- 5: assign  $\{x_{tj}\}$  by calling Algorithm 3
- 6: end for

#### 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.

## II. EXPERIMENTAL RESULTS



## III. CONCLUSION

A new approach for optimizing the dynamic multi-service load balancing in cloud-based multimedia system (CMS-dynMLB) has been enhanced and implemented. The main difference of our model from previous models is that we consider a practical multi-service dynamic scenario in which at different time steps, clients can change their locations, and each server cluster only controls a specific type of multimedia tasks, so that two performance objectives are optimized at the same time. The main features of this paper include not only the proposal of a mathematical formulation of the CMS-dynMLB problem but also a conceptual analysis for the algorithm convergence. Detailed simulation has also been conducted to show the performance of our DES approach.



## 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 (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 pricing for bandwidth sharing," 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 (ChinaCom 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 Vehicular 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. Kirkpatrick, 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.
- [17] 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.

**Author Details:**

	<p><b>K. Raja Rao</b> pursuing M.Tech (CSE) from Nalanda Institute Of Engineering &amp; Technology (NIET), Kantepudi(V), Sattenpalli(M), Guntur (D)-522438, Andhra Pradesh.</p>
	<p><b>Dr. M.V.Bramhananda Reddy</b> working as Professor &amp; Head of the Department (CSE) from Nalanda Institute Of Engineering &amp; Technology (NIET), Kantepudi(V), Sattenpalli(M), Guntur (D)-522438, Andhra Pradesh.</p>