An Effective Genetic Algorithm for Minimizing Load Balancing In Cloud Multimedia System

Glory V. Umoh.

***Abstract*— Cloud Computing plays a role of resource sharing such as information, software, servers, storage, application and services in clouds-based multimedia system (CMS) incorporating a resource manager, cluster heads, and clusters nodes, in which the resource manager allocate client's' request for multimedia service tasks to server clusters in accordance to the cloud user's based on demand basis, and then each cluster head apportion the allocated task to the servers within its server cluster. In a complex CMS, it is a research challenge to design an effective load balancing algorithm that boost the multimedia service task load on servers with the minimum cost of distributing the dynamic workload across multimedia data between server clusters and clients, while the maximum load limit of each server cluster is not disrupted. Unlike previous work, this paper takes into account a more practical dynamic multiservice scenario in which each server cluster only handles one multimedia task, and each client requests a different type of multimedia service at a different time. Such a scenario can be modelled as load balancing problem because of limited user capacity in the server, which is computationally effective but in general it is insufficient. As a result, this paper emphasizes the difference between an existing work and the proposed work and solved the problem of the existing work using an effective genetic algorithm, which is an appropriate method to solve dynamic problems. The results provided in this work give an evidence that the proposed genetic algorithm can effectively use the cloud application**

***Index Terms*— Cloud computing, resource manager, cluster node, server clusters, centralized hierarchical cloud-based multimedia system (CMS), genetic algorithm, Decentralized multimedia system (DMS.**

I. INTRODUCTION

Cloud-based multimedia system (CMS) comprises of the centralized multimedia system and decentralized multimedia system. The CMS emerges because of vast number of user's demands for various multimedia services, storage services, processing and accessing multimedia document through the internet. Most multimedia application are images, sound/audio and video , it need considerable computation , which are represented digitally to output information and it is often used by devices such as mobile , memory card , tablet computer and Mp3 player.

Manuscript received Oct 06, 2015

Glory V. Umoh., Department Of Information technology. SRM University Kattankulathur-603203 Kancheepuram Dt, Tamil Nadu, India

To reduce the cost of storage and easy access cloud facilities is required, multimedia applications are processed on powerful cloud servers, and the users only pay for the utilization of the resources with time.

This paper considers a centralized CMS which composed of a resource manager , server clusters a number of clusters node, each of which is coordinated by a server cluster , and we assume the servers node provide different services. The operation is as follows. Each time the CMS receives clients' requests for multimedia service tasks, the resource manager of the CMS assigns those task requests to different clusters node according to the feature of the requested tasks. It can be clearly observed that the load of each server node significantly affects the performance of the CMS. The resource manager of the CMS is in quest of evenly distribute the task load across cluster node based on subscription list to minimised the number of resources used , thereby achieving load balancing in the CMS.

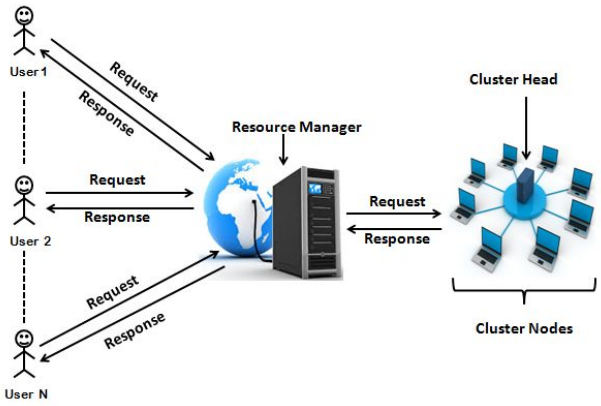


Fig 1 ARCHITECTURE OF THE CENTRALIZED MULTIMEDIA SYSTEM.

II. OBJECTIVES OF MY RESEARCH

The objective of the research is to minimize load balancing problem and to improve the performance of the CMS. The resource manager fetch the client's multimedia task type and to assign the task to the particular cluster node in the cloud. So that client can get data without any interruptions. In practice, however, the CMS offers services of generating, editing, processing, and searching a variety of multimedia data, Different multimedia services have various requirements for the functions provided by the CMS.

To implement single server multiple service concept, in this each cluster node will be in position of having all the multimedia services and each cluster node in the CMS were distributed with Server heads.

III. EXISTING SYSTEM

In the previous dynamic multiservice scenario in which each server cluster only handles a specific type of multimedia task, and each client requests a different type of multimedia service at a different time. Such a scenario can be modelled as an integer linear programming problem, which is computationally intractable in general. The load balancing problem for the CMS is based on all the multimedia service tasks are of the same type, and did not consider the dynamic scenario where load balancing should adapt to the time change. It is not hard to observe that the load of each server cluster significantly affects the performance of the whole CMS. In general, the resource manager of the CMS is in pursuit of fairly distributing the task load across server clusters, and hence, it is of importance and interest to be able to cope with load balancing in the CMS. The decentralized technique is only suitable for smaller systems, it is still easier to implement.

A. Existing System Disadvantages

- Decentralized framework method is suitable for only small collection of system
- Server cluster can handle only one type request at a time so that load balancing problem occurs.

IV. PROPOSED SYSTEM

In the proposed system we implement a centralized CMS and we proposed a genetic algorithm (GA) for the concerned dynamic load balancing problem in CMS. The resource manager of the centralized CMS stores the global service task load information collected from server clusters, and decides the amount of client's requests assigned to each server cluster so that the load of each server cluster is distributed as balanced as possible in terms of the cost of transmitting multimedia data between server clusters and clients. Each time the CMS receives client's requests for multimedia service tasks, the resource manager of the CMS assigns those task requests to different server clusters according to the characteristics of the requested tasks. Subsequently, the cluster head of each server cluster distributes the assigned task to some server within the server cluster.

A. Proposed System Advantages

- The centralized CMS framework is scalable
- CMS framework is suitable for larger collection of systems
- Client's requests assigned to each server cluster so that the load of each server cluster is distributed as balanced as possible.

V. METHODOLOGY

This covers both the systematic and theoretical analysis of the methods applied to my work to accomplish a quintessential result. In other to analyse this project, the methodology is based on Genetic algorithm for an even distribution of resources among the server node.

VI. TECHNIQUE USED

A. Genetic Algorithm (GA)

A solution generated by genetic algorithm is called a chromosome, while collection of chromosome is referred as a population. A chromosome is composed from genes and its value can be either numerical, binary, symbols or characters depending on the problem want to be solved. These chromosomes will undergo a process called fitness function to measure the suitability of solution generated by GA with problem. Some chromosomes in population will mate through process called crossover thus producing new chromosomes named offspring which its genes composition are the combination of their parent.

B. The Algorithm is as follows:

- Step 1: Determine the number of chromosomes: - The Resource Manager checks the number of cluster node that is available in the server
- Step 2. Find the number of Generation: - finding the number of users.
- Step 3. Calculate the mutation rate from the two step above: - calculating the full strength request of the cluster node.
- Step 4. Find out the crossover node from the Chromosomes: - finding out the number of request that is handled by the cluster node.
- Step 5. Generate chromosome to chromosome number of the population, and the initialization Value of the genes chromosome to chromosome with a random value: - this depends upon the user's registration cloud owner can increase the cluster node.
- Step 6: Evaluation of fitness value of chromosomes by calculating objective Function: - At present how many requests are handled by the particular Cluster node.
- Step 7: Chromosomes selection
- Step 8: Crossover
- Step 9: Repeat steps 5-8 until the number of generations is met
- Step 10: Mutation
- Step 11: New Chromosomes (Offspring):- Design a new Cluster node
- Step 12: Solution (Best Chromosomes):- Which node always handle the request and response.

VII. SYSTEM REQUIREMENT

A. Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

PROCESSOR	:	PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
RAM	:	2 GB DD RAM
MONITOR	:	15" COLOR
HARD DISK	:	40 GB
CDDRIVE	:	LG 52X
KEYBOARD	:	STANDARD 102 KEYS
MOUSE	:	3 BUTTONS

B. Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

Operating system	:	Windows 07/ 08
IDE	:	Visual Studio 2010
Front End	:	C#.NET, Windows Form
Database	:	SQL Server 2008

VIII. MODULES

A. Cloud Owner

1. Authentication.
2. File Uploading.
3. View File Description.
4. View User Activities.

B. User

1. Authentication.
2. View Multimedia Files.
3. Service Request.
4. Retrieve Files.

IX. MODULE DESCRIPTION & DIAGRAMS

A. Admin

1. Authentication
   The user has to provide exact username and password which was provided at the time of registration, if login success means it will take up to main page else it will remain in the login page itself.

2. File Uploading
   In this scheme data owner upload the multimedia files in the cloud server. Each service has different set of files. Data owner collect several file from the local path and stored in the Cloud Server. This cloud server has collection of server cluster which uniquely connected with the cloud server.

3. View File Descriptions
   The admin will view uploaded file details which includes file upload time and size.

4. View User Activities
   The admin will view the user requested service and downloaded file details which includes the file retrieve time and keys.

B. User

1. Authentication
   If you are the new user going to login into the application then you have to register first by providing necessary details. After successful completion of sign up process, the user has to login into the application by providing username and exact password.

2. Login
   The user has to provide exact username and password which was provided at the time of registration, if login success means it will take up to main page else it will remain in the login page itself.

3. View Multimedia Files
   The user after the successful login goes to view the category of multimedia services. In that category contains different type of multimedia files such as audio, video and picture.

4. Requesting Service
   In this phase the authenticated users view the multimedia services. The user wants to see the particular category of files then they have to access the category and they can generate a request. Once the Request is generated the Resource managers assign the task to the cloud server.

5. Retrieve file
   In this module the user will get a response file for the corresponding request. The request is initially generated by the user now the cloud server responds that request.

X. FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs. Our system requires minimum three systems to achieve this concept.

A. Non-functional Requirements

1. Efficiency
   Our application efficiently characterizes the server and the cluster requests and response.

2. Throughput
   Our application provides higher throughput for cloud file accessing.

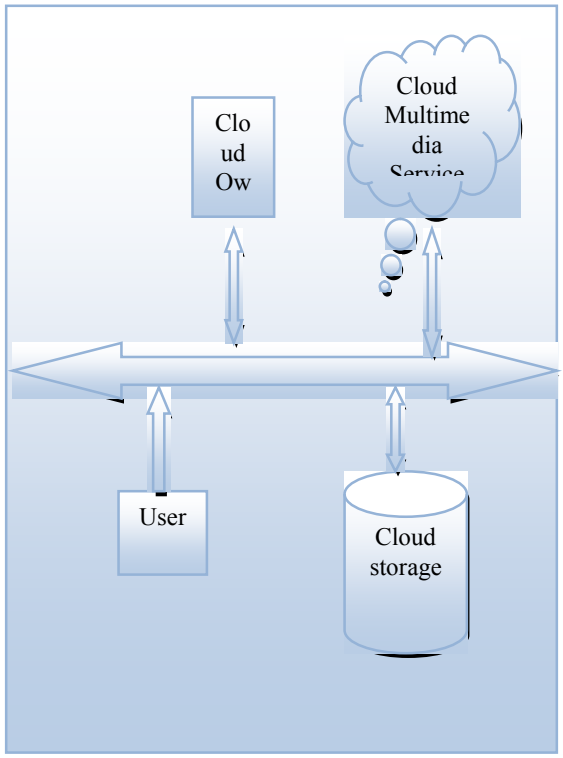
XI. SYSTEM DESIGN

Design deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

A. System Architecture

The users or nodes involved in our projects are Sender, Intermediate and Receiver. In order to send file, the sender

has to find out the list of nodes which are connected with the sender. From that available list he can choose receiver. Then the sender has to analyse the performance of each and every node which is connected with the sender. The performance analysis list will return the priority based result so that sender can choose the intermediate to send the file. The Intermediate will receive the file from sender then it will analyse the performance so that it can send data to another intermediate or receiver. In the receiver side, the receiver has to select the file path to receive the file from sender or intermediate. Then the receiver can view the file received file.



B. OBJECT DIAGRAM

An **object diagram** in the Unified Modelling Language (UML) is a diagram that shows a complete or partial view of the structure of a modelled system at a specific time. An Object diagram focuses on some particular set of object instances and attributes, and the links between the instances. A correlated set of object diagrams provides insight into how an arbitrary view of a system is expected to evolve over time.

Object diagrams are more concrete than class diagrams, and are often used to provide examples, or act as test cases for the class diagrams. Only those aspects of a model that are of current interest need be shown on an object diagram.

C. State Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. There are many forms of state diagrams, which differ slightly and have different semantics.

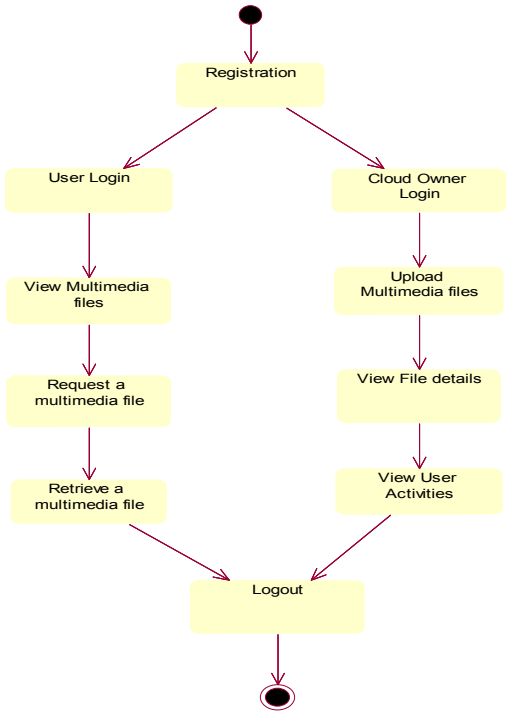


Figure 3. **STATE DIAGRAM**

XII. RESULTS AND DISCUSSION

A. Implementation

This describes the implementation of my project. An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system though computer programming and deployment.

CLOUD OWNER

Cloud Owner Activities	GIVEN INPUT	EXPECTED OUTPUT
Authentication	Provide username and password to get permission for access.	Became authenticated person to request and process the request.
File Uploading	Insert any Multimedia file to the server.	It will be stored in the Cloud Server.
View Multimedia Files	Cloud owner will view the uploaded file details	It will display all the upload files with description
View User Activities	Cloud owner will view user requested files and user downloaded files.	It will display all the downloaded details

USER

User Activities	GIVEN INPUT	EXPECTED OUTPUT
Authentication	Provide username and password to get permission for access.	Became authenticated person to request and process the request.
Requesting Service	User request a multimedia file to the Cloud Server	It will send a request to the Cloud Server
View Multimedia Files	Select any multimedia Service from the cloud	It will show the multimedia file which has been uploaded by the cloud user
Retrieve file	User request a multimedia file to the Cloud Server and the server response a file	It will response a requested multimedia file.

XIII. SUMMARY

The study investigates dynamic load balancing algorithm for Centralized distributed systems using genetic algorithm, the modules involved are cloud owner and user and each have a role to play. We also have the resource manager, the users, the cluster head, the cluster node, which all this help in the evenly distribution of resources in the CMS.

The resource manager fetch the client's multimedia task type and to assign that task to the particular server cluster node in the cloud. So that client can get a data without any interruptions the performance of the resource manager has been verified under scalability. Some simulation results are presented to show the effectiveness of genetic algorithms for dynamic load balancing. Distributed Centralized computing is being widely applied to a variety of large size computational problems. These computational environments are consists of multiple heterogeneous computing modules, these modules interact with each other to solve the problem. In a Centralized distributed computing system (CDCS), processing loads arrive from many users at random time instants. A proper scheduling policy attempts to assign these loads to available clusters nodes so as to complete the processing of all loads in the shortest possible time.

CONCLUSION

In this work a genetic algorithm approach for optimizing the Cloud Multimedia System - dynamic Multimedia Load Balancing (CMS-dynMLB) was proposed and implemented. Load balancing is a major issue in cloud and Genetic Algorithm which I have implemented have a good potential of resolving this problem, Cloud computing has reached a maturity that lead it into a productive phase .this means that most of the main issues with cloud computing have been addressees to a degree that clouds have become interesting for full commercial exploitation.

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 Process. Mag.*, vol. 28, no. 3, pp. 59–69, May 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 Syst. J.*, vol. 4, no. 2, pp. 262–270, Jun. 2010.

[3] W. Hui, H. Zhao, C. Lin, and Y. Yang, "Effective load balancing for cloud-based multimedia system," in *Proc. Int. Conf. Electron. Mech. Eng. Inform. Technol.*, 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," *J. Internet Technol.*, 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 *Proc. 24th IEEE Int. Conf. Advanced Information Networking Appl. Workshops*, Mar. 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 Commun. Mag.*, vol. 45, no. 5, pp.140–146, May 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," *J. Internet Technol.*, 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 *Proc. 6th Int. ICST Conf. Commun. Networking China (ChinaCom)*, 2011, pp. 703–707.H. Son, S. Lee, S.-C. Kim, and Y.-S. Shin, "Soft load balancing over heterogeneous wireless etworks," *IEEE Trans. Vehic. Technol.*, vol. 57, no. 4, pp. 2632–2638, Jul. 2008.

[10] L. Zhou, H.-C. Chao, and A. V. Vasilakos, "Joint forensics-scheduling strategy for delay-sensitive multimedia applications over heterogeneous networks," *IEEE J. Selected Areas Commun.*, vol. 29, no. 7, pp. 1358–1367, Aug. 2011.

[11] [12] Fekri M. Abduljalil and Shrikant K. Bodhe." A Survey of Integrating IP Mobility Protocols and Mobile Ad Hoc Networks" IEEE Communications Surveys & Tutorials, vol. 9, NO. 1, pp. 1553-877X. April 2007.