International Journal of Computer Science and Business Informatics

IJCSBI.ORG

Load Balancing using Peers in an E-Learning Environment

Maria Dominic
Department of Computer Science,
Sacred Heart College, India

Sagayaraj Francis
Department of Computer Science and Engineering,
Pondicherry Engineering College, India

ABSTRACT
When an e-Learning System is installed on a server, numerous learners make use of it and they download various learning objects from the server. Most of the time the request is for same learning object and downloaded from the server which results in server performing the same repetitive task of locating the file and sending it across to the requestor or the client. This results in wasting the precious CPU usage of the server for the same task which has been performed already. This paper provides a novel structure and an algorithm which stores the details of the various clients who have already downloaded the learning objects in a dynamic hash table and look up that table when a new request comes in and sends the learning object from that client to the requestor thus saving the precious CPU time of the server by harnessing the computing power of the clients.

Keywords
Learning Objects, e-Learning, Load Distribution, Load Balancing, Data Structure, Peer – Peer Distribution.

1. INTRODUCTION

1.1 e-Learning
Education is defined as the conscious attempt to promote learning in others to acquire knowledge, skills and character [1]. To achieve this mission different pedagogies were used and later on with the advent of new information communication technology tools and popularity gained by internet were used to enhance the teaching learning process and gave way to the birth of e-learning [2]. This enabled the learner to learn by breaking the time, geographical barriers and it allowed them to have individualized learning paths [3]. The perception on e-Learning or electronic learning is that it is a combination of internet, electronic form and network to disseminate knowledge. The key factors of e-learning are reusing, sharing resources and interoperability [4]. At present there are various organizations

providing e-learning tools of multiple functionalities and one such is MOODLE (Modular Object Oriented Dynamic Learning Environment) [5] which is used in our campus. This in turn created difficulty in sharing the learning objects between heterogeneous sites and standards such as SCORM & SCORM LOM [6], IMS & IMS DRI [7], AICC [8] and likewise were proposed by different organizations. In Berner-Lee's famous architecture for Semantic Web, ontology's are used for sharing and interoperability which can be used to build better e-learning systems [9]. In order to define components for e-learning systems the methodology used is the principle of composibility in Service Oriented Architecture [10] since it enables us to define the inter-relations between the different e-learning components. The most popular model used nowadays in teaching learning process is Felder-Silverman learning style model [11]. The e-Learning components are based on key topics, topic types and associations and occurrences. VLE – Virtual Learning Environment is the software which handles all the activities of learning. Learning Objects are the learning materials which promotes a conscious attempt to promote visual, verbal, logical and musical intelligence [12] through presentations, tutorials, problem solving and projects. By the multimedia, gaming and simulation kin aesthetic intelligence are promoted. Interpersonal, intrapersonal and naturalistic intelligence are promoted by means of chat, SMS, e-mail, forum, video, audio conference, survey, voting and search. Finally assessment is used to test the knowledge acquired by the learner and the repository is the place which will hold all the learning materials.

This algorithm is useful when the learners access the learning objects which are stored in the repository. It reduces the server's response rate by the directing a client to respond to the requestor with the file it has already downloaded from the server.

1.2 Load Balancing

The emergence of large and faster networks with thousands of computers connected to it provided a challenge to provide effective sharing of resource around the computers in the network. Load balancing is a critical issue in peer to peer network [14]. The existing load balancing algorithms for heterogeneous, P2P networks are organized in a hierarchical fashion. Since P2P have gained popularity it became mandatory to manage huge volume of data to make sure that the response time is acceptable to the users. Due to the requirement for the data from multiple clients at the same instance may cause some of the peers to become bottleneck, and thereby creating severe load imbalance and the response time to the user. So to reduce the bottlenecks and the overhead of the server there was a need to harness the computing power of the peers [15]. Much work has been done on harnessing

the computing power of the computer in the network in high performance computing and scientific application, faster access to data and reducing the computing time is still to be explored. In a P2P network the data is de-clustered across the peers in the network. When there is requirement for a popular data from across the peer then there occurs a bottleneck and degrading the system response. So to handle this, a new strategy using a new structure and an algorithm are proposed in this paper.

2. PROPOSED DATA STRUCTURE AND THE ALGORITHM

The objective of this architecture is to harness the computational power of the clients in the network. This architecture is with respect to the clients available in the e-learning network. The network comprises of Master degree students of Computer Applications accessing learning materials for their course. The degree programme is a three year programme. So the clients are categorized to three different clusters namely I MCA, II MCA, III MCA. We shall name it as class cluster. Every class cluster contains many clusters inside it, let us name it as file clusters, one cluster for one type of file since the learning objects can be made up of presentation, video, audio, picture, animation etc [13]. An address table, named file address table holds the address of each file cluster in the class cluster. When a request for a file is received the corresponding cluster is identified by reading the address from the address table. The following algorithm represents the working logic of the concept. The data structure is represented in Figure 1. Every file cluster holds a Dynamic Hash Table (DHT), Linked List and a Binary Tree. The dynamic hash table holds the address of the linked list which holds the file names that are already downloaded from the server. The hashing function used to identify an index in the DHT is as follows,

1. Represent every character in the filename with its position in the alphabet list and its position in the filename.
   Eg: File Name - abc.ppt = 112233, the value for *a* is got as 11 since the position of it in the alphabet list is 1 and its position in the file name is 1.

2. Sum all the digits calculated from step 1.
   Eg: 112233 = 12

3. Divide the sum by length of the file name, so 12/3 = 4, which becomes the index for the file in DHT. The above three steps are mathematical formulated in equation 1.

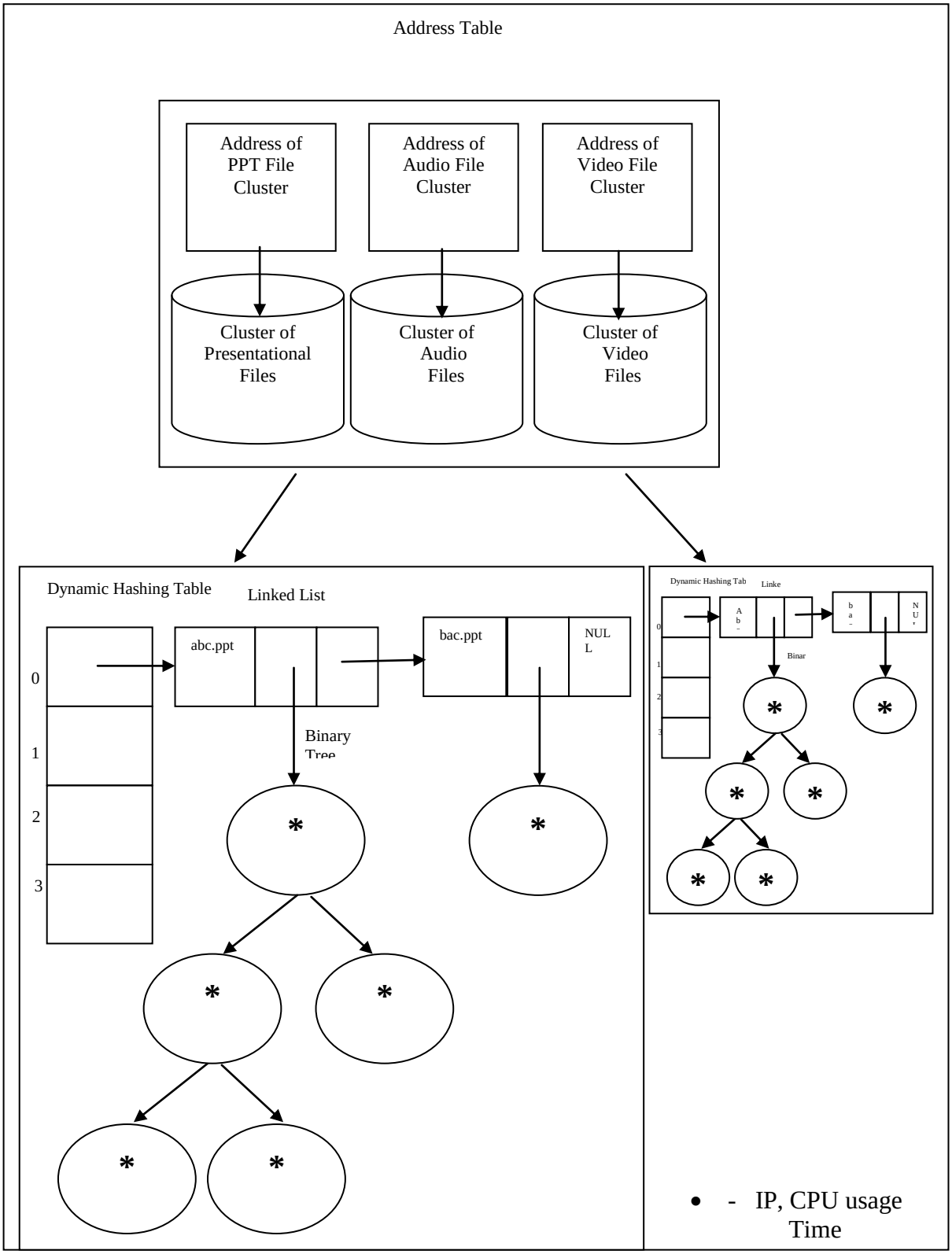

Figure 1. Proposed Data Structure

Every index of DHT holds the starting address of a linked list in which every node stores the file names that are already downloaded. The linked list structure is used to avoid index collision between filenames generating the same index in DHT. The index collision is avoided by creating a new node in the linked list for the new file name. As shown in Figure 1, every node in the linked list holds three values namely file name, address of the binary tree and the last part holds the address of the next node in the list. The nodes of the binary tree holds the active clients IP's and its current CPU processing status. The binary tree is used to identify the leased CPU used client to transfer the file to the requestor. This activity will harness the computing power of the least used CPU. The binary tree structure was used to reduce the search time for the least used client. If the file has not been downloaded by any clients i.e. when the last node of the linked list is reached, then the file is transferred from the server.

Algorithm 1 ***SEND ( )*** *{*

```
Request directed to the File Cluster
Address of the File Cluster taken from the Address Table
Index Location of the File = HASHED (File Name)
If the index is out of bound
  {
      The file has not been downloaded by any client
      It is sent from the server to the client
  }
Else
 {
     While (not end of Linked List OR  the node is not  found)
    {
           If (Node. data == File Name)
          {
             Node found = true
             While (end of binary tree}
            {
             Least usage time CPU IP = LEASTUSEDCPU()
            }
          }
If (Node found == true)
  Send the requested file from the IP to the Requestor
Else
  The file has not been downloaded by any  client
  It is sent from the server to the client
```

}
*End of the **SEND ( )***

Algorithm 2 ***int LEASTUSEDCPU ( )*** {

While (not end of Binary Tree)
{
Compare the CPU Usage Time of the first node with the CPU Usage Time of every other Node

If CPU Usage Time of the new node is Lesser
Leastusedcpu = IP
}
return (Leastusedcpu)
*End of **LEASTUSEDCPU ( )***

Algorithm 3 ***int HASHED (String Filename) {***

Len = StringLength(Filename)
While (End of String)
{
IndexString += (Position of the Character in the Filename + Position of the Character in the alphabet list)
}
IndexInt = ConverttoInteger(IndexString)
return (IndexInt = IndexInt/Length of the Array)
*End of **HASHED ( )***

3. MATHEMATICAL FORMULATION

Mathematical formulation of the above dealt problem is as follows,

$$\text{index} = \left(\sum_{i=0}^{l} \left(a_{(i+1)+j} \right) / l \right) \quad (1)$$

$$\sum_{K=1}^{n} \text{index}\left(Z_K (f3) \right) = X(f2) \text{ goto 4, 5} \quad (2)$$

$$\sum_{K=1}^{n} \text{index}\,(Z_K(f3)) \neq X(f2) \quad \text{goto 6} \qquad (3)$$

$$\text{Min} \sum_{m=1}^{n} Y = C(m, m+1) \qquad (4)$$

$$f2 = X^{-1} Y(f1) \qquad (5)$$

$$f2 = X^{-1} S(f1) \qquad (6)$$

where,

index is the index in the Dynamic Hash Table
l is the length of the file name
i is the character position in the file name
j = { 1,2,3,4,5….26 }
k is the number of nodes in the linked list
Z is the node in the linked list
f3 is the file name in the node in the linked list
f2 is the targeted file
m is the nodes in the Binary Tree
Y is the node in the Binary Tree with the minimum C
C is the CPU usage time of the specified IP
S is the Server

4. CONCLUSION

The main advantage from this architecture is that the server time is saved by harnessing the computational power of the clients who have already downloaded the file to send across it to the requestor. Another advantage of the architecture is the file search, which has been fastened due to the Dynamic Hashing table and Binary tree structures. This algorithm is been currently implemented using PHP and the results of it will be published in the further publications. Initial results indicate that there is substantial reduction of the server's CPU processing time when this algorithm is executed on the server.

REFERENCES

[1] Lavanya Rajendran, Ramachandran Veilumuthu., 2011. A Cost Effective Cloud Service for E-Learning Video on Demand, *European Journal of Scientific Research*, pp.569-579.

[2] Maria Dominic, Sagyaraj Francis, Philomenraj., 2013. A Study On Users On Moodle Through Sarasin Model, *International Journal Of Computer Engineering And Technology*, Volume 4, Issue 1, pp 71-79.

[3] Maria Dominic, Sagyaraj Francis., 2013. Assessment Of Popular E-Learning Systems Via Felder-Silverman Model And A Comphrehensive E-Learning System, *International Journal Of Modern Education And Computer Science*, Hong Kong, Volume 5, Issue 11, pp 1-10.

[4] Zhang Guoli, Liu Wanjun, 2010. *The Applied Research of Cloud Computing.* Platform Architecture in the E-Learning Area, IEEE.

[5] www.moodle.org

[6] SCORM(Sharable Courseware Object Reference Model), http://www.adlnet.org

[7] IMS Global Learning Consortium, Inc., "Instructional Management System (IMS)", http://www.imsglobal.org.

[8] http://www.aicc.org

[9] Uschold, Gruninger., 1996. Ontologies, Principles , Methods and Applications, *Knowledge Engineering Review*, Volume 11, Issue 2.

[10] Papazoglou, Heuvel., 2007. Service Oriented Architectures: Approaches, Technologies, and research issues, *The VLDB Journal*, , Volume 16, Issue 3, pp. 389-415.

[11] Graf, Viola, Kinshuk., 2006. Representative Characterestics of Felder-Silverman Learning Styles: an Empirical Model, *IADIS*, pp. 235-242.

[12] Lorna Uden, Ernesto Damiani., 2007. The Future of E-Learning: E-Learning ecosystem, *Proceeding of IEEE Conference on Digital ecosystems and Techniques*, Australia, pp. 113-117.

[13] Maria Dominic, Sagyaraj Francis., 2012. Mapping E-Learning System To Cloud Computing, *International Journal Of Engineering Research And Technology*, India, Volume1, Issue 6.

[14] Chyouhwa Chen, Kun-Cheng Tsai., 2008. The Server Reassignment Problem forload Balancing In Structured P2P Systems, *IEEE Transactions On Parallel And Distributed Systems*, Volume 19, Issue 2.

[15] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica., 2006. Load Balancing in Structured P2P Systems. *Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS '03).*

This paper may be cited as:

Dominic, M. and Francis, S. 2014. Load Balancing using Peers in an E-Learning Environment. *International Journal of Computer Science and Business Informatics, Vol. 14, No. 1., pp. 22 -29.*