DISTRIBUTED E-LEARNING SYSTEM USING AN HYBRID P2P-BASED CONTENT ADDRESSABLE NETWORK ARCHITECTURE

Telesphore Tiendrebeogo[1] and Kabre Laciné[2]

[1]Department of Mathematics and Computer Engineering, Nazi Boni University, Bobo-Dioulasso, Burkina Faso
[2]Department of Mathematics and Computer Engineering, Professor Joseph Ki-Zerbo University, Ouagadougou, Burkina Faso

ABSTRACT

We have developed a distributed asynchronous web-based training system. In order to improve the scalability and robustness of this system, all the content and a function are realized on mobile agents. These agents are distributed to computers, and they can use a Peer to Peer network that modifies the DHT Content Addressable Network (CAN). In this system, all computers offer the function and exercise by themselves. However, the system that all computers do the same behavior is not realistic. In this paper, we present an e-learning system composed of computers with different types of participation as a solution to this problem. Activating computers of different types of participation will improve the convenience of the system.

KEYWORDS

Distributed Multimedia Systems, E-Learning, P2P, Distributed Mobile Multimedia Systems, Mobile Agent.

1. INTRODUCTION

Asynchronous Web-Based Training (WBT) systems are very popular in e-learning systems. WBT allows a learner to complete the WBT on their own time and on their own schedule, without live interaction with an instructor. A large number of studies have been conducted on asynchronous WBT [1] [2] [3], and all are based on the client/server model. The features of the client/server model are that all of them must run the management and offer the exercise through the server machine. While the client/server model has the advantage of easy construction and maintenance, client/server systems generally lack scalability and robustness. There is the Peer-to-Peer (P2P) model to complement the disadvantage of the client/server model. The function in the system is based on the P2P model that each computer functions as a client or server. The function can distribute the load to a node. The function in the whole system does not stop even if some nodes break.

We proposed and implemented a distributed e-learning system based on P2P architecture [4] [5]. The proposed e-learning system has two distinct features. First, it is based on the P2P architecture to improve the scalability and robustness of the system. In the proposed e-Learning system, each user computer plays the role of a client and a server. While a user is using the system, his computer (hereafter we refer to such a computer as a node) is part of the system. The node receives a certain amount of content from another node when it joins the system and is responsible for sending the appropriate content to the requesting nodes.

DOI:10.5121/ijdps.2020.11501

In addition to the above advantages of using P2P architecture, the proposed e-Learning system can be built at low cost because the system does not require server computers. Secondly, each exercise in the system is not only data, but also an agent, so it has functions, such as correcting the user's answers, formulating the correct answers and presenting some related information without human instructions. The proposed system also processes all nodes and distributes functions.

However, when the system is composed of nodes that have a low machine specification or use a narrow network bandwidth, the performance of the entire system is reduced. A node that only runs exercises cannot participate in the system.

In our work, as a solution to this problem, an e-learning system composed of computers with different types of participation is proposed. That is, a system is composed of nodes that have the same functions as a pure P2P model, another system is composed of nodes that may or may not have a function as a client/server model. One node only provides data and functions as a server, and another node only learns. By building the system that uses the nodes with only the server function, even if the problem occurs in one of the nodes that make up a server group, the system will continue the service. By building the system that uses the nodes to perform learning only, the performance of the whole system will be stable.

This work is divided into five sections. The proposed e-learning system is described in Section 2. Overviews of the design and implementation of the proposed system are described in Section 3 and the experimental result in Section 4, with some concluding remarks in Section 5.

2. PROPOSED E-LEARNING SYSTEM

2.1. Overview

All the exercises in the proposed system are classified into categories, such as "Math/Statistics", "English/Grammar", etc. The user can obtain exercises one after the other by specifying the categories of the required exercises. Although a user is using the proposed e-learning system, his or her node must be part of the system. The node receives a number of categories and exercises from another node when it joins the system and is responsible for sending the appropriate exercises to the node on demand.

The important point to note is that a node's categories are independent of the categories in which the node user is interested, as shown in Figure 1. Figure 1 illustrates that User A's request is first forwarded to the neighboring node, and the request is forwarded to the node that has the requested category.

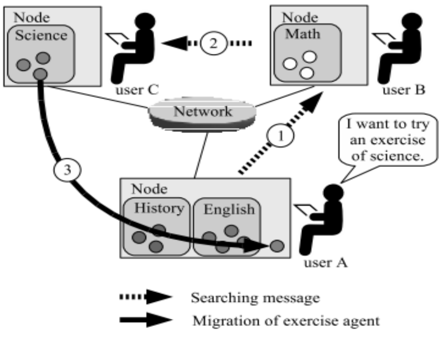

Figure 1. Migration of exercise agent.

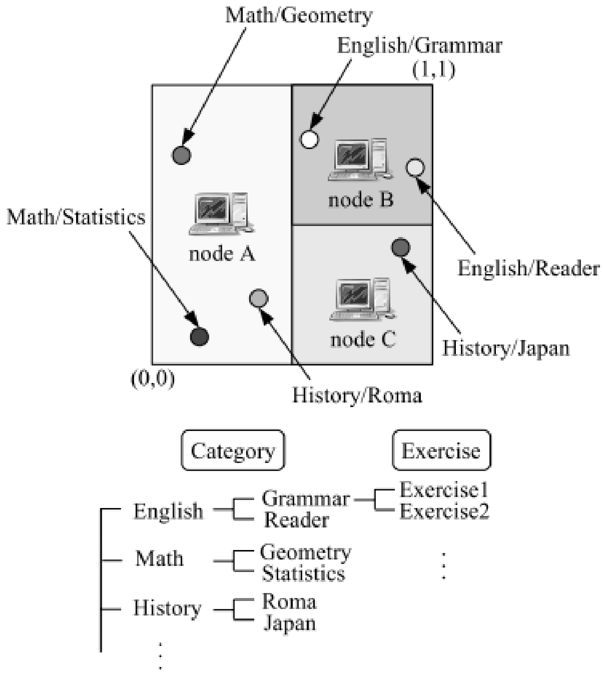

Figure 2. P2P network of the proposed e-learning system.

2.2. About P2P Network

When system bootstraps are proposed, an initial node has all categories in the system. When another node joins the system, it receives a number of categories from the initial node. The categories are distributed among all nodes in the system according to whether the nodes join the system or leave the system.

In existing P2P-based sharing systems, such as Napster [6], Gnutella [7], and Freenet [8], each shared file belongs to a particular node. In systems, files are originally distributed among all nodes. On the other hand, categories in the proposed system are initially concentrated. When a new node joins the system, not only the location information of a category, but also the category itself has to be transmitted to the new node. Whereas, the P2P network of the system can be built as a CAN [9].

Our P2P network is built with a two-dimensional [0.1] [0.1] coordinate space to store exercise categories, as shown in Figure 2. Figure 2 shows the situation where node C has just joined the system as a third node. Before node C joined, node A and node B shared half of the coordinate space. At that time, Node A managed the categories "Math/Geometry", "Math/Statistics", and "History/Rome" and Node B managed the categories "English/Grammar", "English/Reader", and "History/Japan", respectively. When Node C joins the system, we assume that Node C already knows the IP addresses of some nodes in the system and that Node C sends the join request to a node in the list. Then node C is mapped to a certain coordinate space based on a random number and takes the corresponding categories from another node. For example, in the case of Figure 2, Node C takes the category "History/Japan" from Node B, and then the exercises in the category go to Node C. After joining, Node C gets a list of the IP addresses of neighboring nodes in the coordinate space, such as Node A and Node B. Therefore, neighboring nodes can communicate with each other. When Node C joins the system, we assume that Node C already knows the IP addresses of some nodes in the system and that Node C sends the join request to a node in the list.

Then node C is mapped to a certain coordinate space based on a random number and takes the corresponding categories from another node. For example, in the case of Figure 2, Node C takes the category "History/Japan" from Node B, and then the exercises in the category go to Node C. After joining, Node C gets a list of the IP addresses of neighboring nodes in the coordinate space, such as Node A and Node B. Therefore, neighboring nodes can communicate with each other.

2.3. Specific System Component

For the proposed system to be considered a distributed WBT system, it is not enough that the exercises are distributed among all the nodes. The functions to provide the above services must also be distributed among all nodes. Mobile agent technology is adopted to achieve this goal. Here are the agents and user interface programs on each node. These agents have been integrated into the mobile agent framework we have developed [10].

- **Node Agent:** Each node has a node agent. It manages the zone information of a CAN and transmits messages to the category agents in the node.

- **Exercise Agent:** Each exercise agent has questions and functions to evaluate the user's answers, to give the correct answers, and to show related exercise information

.

- **Category Agent:** Each Category Agent represents a unit of a particular subject. It manages the exercise agents per se and sends them to the requesting node.

- **Interface Agent:** There is an interface agent for each user interface, such as a student interface and an exercise manager interface on each node. It acts as an interface between the interface program and other agents, and between agents and applications.

- **User Agent:** Each user has its own User Agent. A User Agent manages the information of its user, which includes username, password, IP address of the user's computer, online/offline status, and study log or a list of created exercises.

- **Student Interface:** A student interface is located on each node where a user logs in as a student. It is a user interface program for studying.

3. NODE' INVOLVEMENT TYPE

The following nodes are used in the proposed system.

- **Server node:** only data and functions are offered.

- **Common node:** Data and functions are offered, there are the requirements of the agent's system and he works directly as a learner and teacher.

- **Client Node:** Neither data nor function is offered, they have the environment that the agent uses, and they function directly as learner and teacher.

- **User Interface Node:** Neither data nor function is offered, the agent's work environment is unsuitable, and they function directly as learner and teacher.

3.1. Client Node Design and Implementation

In the proposed system, the distributed hash table (DHT) that maps over the two-dimensional coordinate space is used. The DHT area is given to the node participating in the system. As a result, the node composes a P2P network, and the contents are managed. However, the client node does not manage the content and the P2P network function is used. To realize the type of participation that uses only the P2P network function, when the client node participates in the P2P network, it should not receive the DHT zone. Instead of not receiving the DHT zone, the client node registers the address of the node that participates in the P2P network as the connection destination. The registered node is named Proxy node.

The proxy node is selected according to the following procedures.

- When the client node participates in a system for the first time, the client node connects to the node it has already known. The node is named the known node.
- The known node generates a coordinate.
- A node with the coordinate is chosen as a proxy in the domain to manage its own.

When the proxy node quits, the query of the known node is executed again, and the node that includes the coordinate in the area is retrieved and then it registers as a new proxy node. When the zone of the proxy node changes and the zone does not include the coordinate that is generated by the known node, it recovers by searching for the known node, and it registers as a new proxy node. The client node can use all functions except content delivery. There is an advantage that there is no influence on the P2P network even if the problem occurs in the proxy node, because the client node unilaterally executes the registration of the proxy node.

3.2. User Interface Node Design and Implementation

The interface agent acts as an intermediary between an agent and a user interface that the learner uses. If the procedure can be called remotely, which is one of the functions of the Agent Interface, users can practice using the Student Interface. In other words, for the User Interface (UI) node to participate as a learner, simply prepare the Agent in the connected node. Therefore, the interface agent for the UI node to exercise is prepared.

In the proposed system, the Agent ID is essential to communicate with the agent. The agent ID is necessary to avoid confusion in the message, the same applies to the communication between the user interface and the interface agent. In communication, since the Node Agent is only one in each server, it does not need the Agent ID. The user interface node asks the Node Agent in the known node to generate the interface agent. Then the Node Agent generates the Interface Agent and returns the ID to the User Interface Node. The User Interface Node can exercise through this mechanism.

However, there is a problem to think about when designing the UI node. This is the case when the node that generates the Agent Interface leaves. The migration method to the node that integrates the DHT zones of the leaving node cannot be performed for two reasons. One reason is that the Interface Agent exists only through one or less in one node. The other is that the UI cannot find the embedded node even though two or more UI Agent may exist in one node.

In the proposed system, communication with the agent can be established from the user interface, reverse communication cannot be established. Not being able to establish UI communication from the agent, this means that the UI agent cannot tell the UI that is migrating to the embedded node, as a result, the UI cannot find the UI agent.

To solve the above problem, the generated interface agent is mapped into DHT. When the Interface Agent is mapped into DHT, the username is treated as a key. Then the Interface Agent automatically migrates the node according to the change in the zone. Since the user interface can use the Node Agent function as long as the Known Node is identified, it becomes possible to find the Interface Agent that makes the username a key and uses the Agent Node search function.
When the UI node is exercising, a user's user agent and the exercising exercise agent are sent to the node where the UI agent exists. If the interface agent manages both the user agent and the exercise agent, these agents can be migrated together. This method allows to continue exercising, since the Agent Interface can be found even if the user interface loses sight of the Agent Interface.

4. EXPERIMENTATION

This section presents the experimental results of the e-Learning system where the node with different types of participation exists together. The following experiments are examined.

Table 1 shows the specification of the machine.

- Stopping the system's operation
- Comparison of response speeds by number of participation nodes

4.1. Stopping System Operation

The system is built "N1: Server Node", "N2: Common Node" and "N3: User Interface Node". Table I aims to show the experimental conditions. The experiment we are conducting aims to achieve the following:

When the node with a different function in itself exists, the category and exercise can be done and can be acquired.

- When the UI node participates in the system, the UI Agent corresponding to it can be done.
- Even if the proxy node exits, the UI node can be connected with a new proxy node.

Table 1: Experimental Conditions

Nodes number	N1:49, N2:1, N3:5
Groups number	50
User account number	Students: 50, Teachers: 3
Categories number	20
Exercises number	60

Figure 3 shows the appearance of the divided DHT area with the participating node, where the category agent and the interface agent are made corresponding to a learner. The P2P network is built according to the following procedure:

1. First, nine nodes of N1 and one node of N2 participate in the system, then forty nodes of N1 participate. Each node is named node 1, node 2, ..., and node 50 in order of participation respectively.

2. On node 10 which is node N2, the system is connected as a teacher using the user interface for the teacher.

3. The teacher makes four subjects of English, Mathematics, Biology and Physics, and establishes five categories for each subject, and makes three exercises for each category.

4. The teacher logs out of the system once after performing these operations and logs back into the system as a teacher.

As a result, categories and exercises are added to this system normally. When the teacher logs back into the system, the categories and exercises are in the system again.

Then, the P2P network is built according to the following procedures.

1. Five N3 nodes participate in the system, and then learners log in to each node through the student interface. Each node is named node 51, node 52, ..., and node 55 in order of participation respectively.

2. After logging in to the system, each learner asks the teacher who has already logged in to send the exercise.

3. Node 7 in which the interface agent corresponding to nodes 51 and 53 exists, after leaving the system, these learners request the exercises.

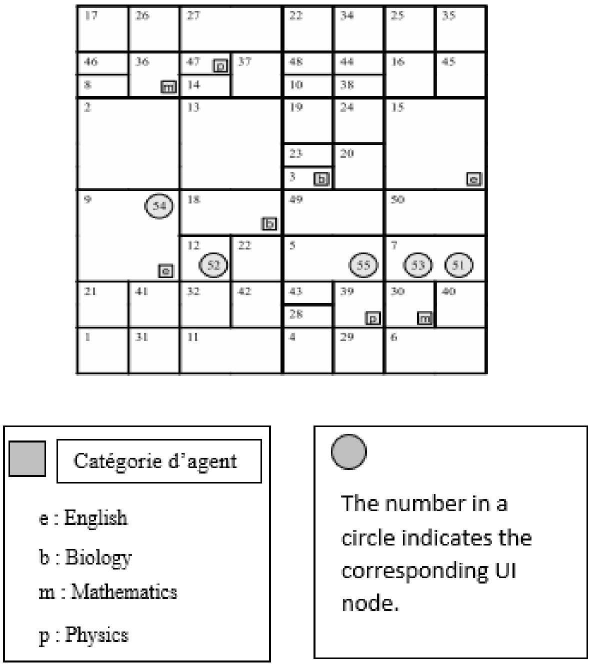

Figure 3: Management area division and distributed agents.

4.2. Response time corresponding to the server node number comparison

The system is built "N1: Server Node" and "N3: User Interface Node". It aims at examining the number of search nodes in relation to the response time. The response time in this experiment means the time it takes to display the exercise on the student interface after the learner has requested it.

N1 nodes number is 1, 2, 3, 6, 12 and 20. The number of knots N2 is 5 and the average response time is measured by repeating five times to acquire the exercises of all the knots at the same time. The average times are summarized in Figure 4.

The case where the N1 nodes is three is excluded, because the number of nodes increases, the response time is shortened. When the number of N1 nodes becomes 6 or more, the width of a decrease at the time of response becomes small.

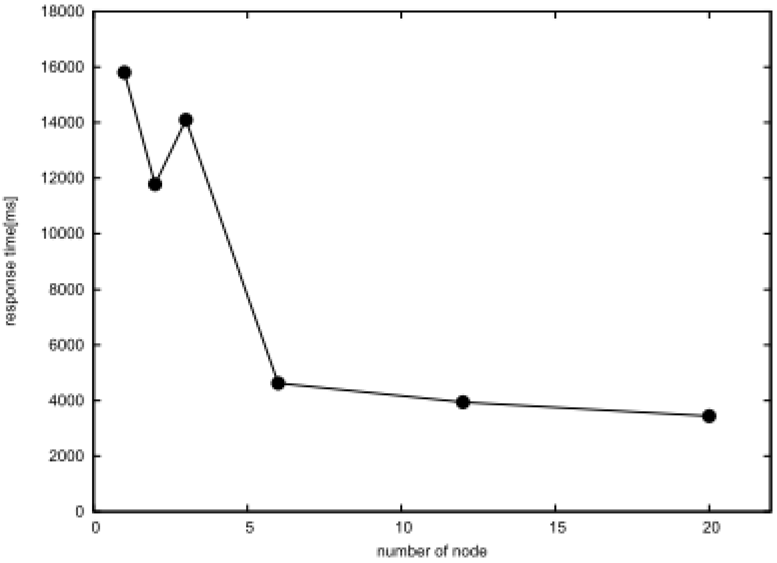

Figure 4: Average Response Time

5. CONCLUSION

A distributed e-learning system in which different types of users can participate has been proposed. The type of nodes that are implemented in the proposed system were "only participate in the P2P network as a server", "participate in the P2P network and participate as a learner", "participate only as a learner with agent system requirements" and "use only as a learner".

By increasing the nodes number participating as a server, it became not only easy to maintain the node that makes up the server group, but also distributed the load to each node that makes up the P2P network, and the response of the entire P2P network improved.

However, in the node type "use only as learner", even if the node where the interface agent has been mapped does not quit, the interface agent migrates with the change of DHT zone, then the interface agent is recovered again. Therefore, in the e-Learning system where the DHT zone changes frequently, the node cannot exercise effectively. In future work, we should check the usability with a larger scale system.

REFERENCES

[1] Nishita, T. and et el : Development of a Web Based Training system and Courseware for Advanced Computer Graphics Courses Enhanced by Interactive Java Applets, Proceedings of International Conference on Geometry and Graphics, Vol. 2, pp. 123-128 (2002).

[2] Homma, H. and Aoki, Y.: Creation of WBT Server on Digital Signal Processing, Proceedings of 4th International Conference on Information Technology Based Higher Education and Training, (2003). Marrakech, Morocco.

[3] Helic, D., Krottmaier, H., Maurer, H. and Scerbakov, N.: Enabling Project-Based Learning in WBT Systems, International Journal on E-Learning, Vol. 4, No. 4, pp. 445-461 (2005). Norfolk, VA.

[4] Kawamura, T. and Sugahara, K.: A Mobile Agent-Based P2P e-Learning System, IPSJ Journal, Vol. 46, No. 1, pp. 222-225 (2005).

[5] Motomura, S., Nakatani, R., Kawamura, T. and Sugahara, K.: Distributed e-Learning System Using P2P Technology, Proceedings of the 2nd International Conference on Web Information Systems and Technologies, pp.250-255 (2006). Setubal, Portugal.

[6] Napster, http://www.napster.com (1999).

[7] Gnutella, http://welcome.to/gnutella/ (2000).

[8] Clarke, I., Sandberg, O,. Wiley, B. and Hong, T. W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System, http://freenetproject.org/papers/freenet.pdf (2000).

[9] Ratnasamy S., Francis, P., Handley, M., Karp, R. and Schenker, S.: A scalable content-addressable network, Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications, ACM Press, pp. 161-172 (2001).