On the Continuous Service of Distributed e-Learning System

Kazunari Meguro, Shinichi Motomura, Takao Kawamura, and Kazunori Sugahara

Abstract—In this paper, backup and recovery technique for Peer to Peer applications, such as a distributed asynchronous Web-Based Training system that we have previously proposed. In order to improve the scalability and robustness of this system, all contents and function are realized on mobile agents. These agents are distributed to computers, and they can obtain using a Peer to Peer network that modified Content-Addressable Network. In the proposed system, although entire services do not become impossible even if some computers break down, the problem that contents disappear occurs with an agent's disappearance. As a solution for this issue, backups of agents are distributed to computers. If a failure of a computer is detected, other computers will continue service using backups of the agents belonged to the computer.

Keywords—Distributed Multimedia Systems, e-Learning, P2P, Mobile Agent

I. INTRODUCTION

Nowadays, e-Learning systems, especially asynchronous Web-Based Training systems (hereafter we abbreviate as WBT) are very popular. A WBT allows a learner to complete the WBT on his own time and schedule, without live interaction with the instructor. Although a large number of studies have been made on asynchronous WBT [1] [2] [3], all of them are based on the client/server model. The features of the client/server model are that all are to execute management and to offer the exercise by the server machine. Although the client/server model have an advantage of easy construction and maintenance, however, the client/server systems generally lack scalability and robustness. There is Peer to Peer (hereafter we abbreviate as P2P) model to supplement the disadvantage of client/server model. The feature in the system is based on P2P model that each computer works as a client or a server. The feature can distribute the load to a node. The function of the entire system doesn't stop even if some nodes break down.

We have proposed and implemented a distributed e-Learning system based on P2P architecture [4] [5]. The proposed e-Learning system has two distinguishing features. Firstly, it is based on P2P architecture to improve the scalability and robustness of the system. In the proposed e-Learning system, every user's computer plays the role of a client and a server. While a user uses the system, his/her computer (hereafter we refer to such a computer as a node) is a part of the system. The node receives some number of contents from another node when it joins the system, and has responsibility to send appropriate contents to the requesting nodes. In addition to the above

K. Meguro is with Ph.D. student in Tottori University.
S. Motomura is Information Media Center Tottori University, T Kawamura and K Sugahara are with the Department of Information and Knowledge Engineering Tottori University.

advantages of using P2P architecture, the proposed e-Learning system can be constructed at low cost because the system need no server computers. Secondly, each exercise in the system is not only data but also an agent which has functions, such as scoring user's answers, telling the correct answers, and showing some related information without human instruction.

In the proposed system, exercises and functions are distributed among all node. If a node failure occurs, another node of the system will continue service. However, when a node failure increases, the exercises in the node are lost and cannot be studied by anyone afterward.

As a solution for this issue, we propose an e-Learning system that backups of agents are distributed to nodes. In this system, if a node failure occurs, another node will continue service using backup of the agent which belongs to the failure node.

This paper is organized in 5 sections. The proposed e-Learning system is described in Section 2. We describe the design overviews of the proposed system in Section 3 and the experimental result in Section 4. Finally, some concluding remarks are drawn in Section 5.

II. PROPOSED E-LEARNING SYSTEM

A. Overview

All exercises in the proposed system are classified into categories, such as "Math/Statistic", "English/Grammar", etc. A user can obtain exercises one after another through specified categories of the required exercises. While a user uses the proposed e-Learning system, his/her node is to be a part of the system. The node receives some number of categories and exercises from another node when it joins the system and has responsibility to send appropriate exercises to requesting nodes.

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

B. P2P Network

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

World Academy of Science, Engineering and Technology
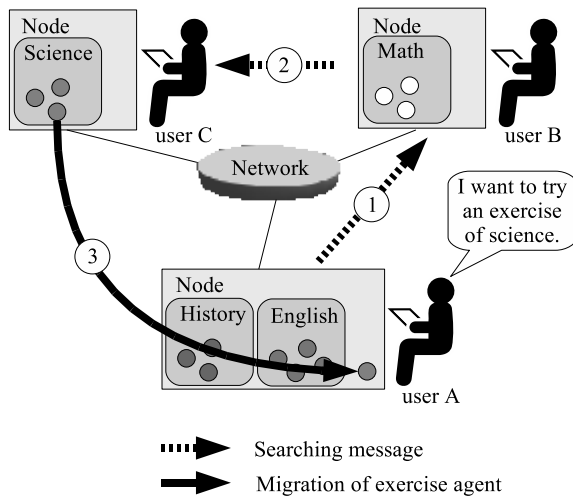International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

Fig. 1. Migration of exercise agent.

In existing P2P-based file sharing systems, such as Napster [6], Gnutella [7], and Freenet [8], each shared file is owned by a particular node. In the systems, files are originally distributed among all nodes. On the other hand, the categories in the proposed system are originally concentrated. When a new node joins the system, not only location information of a category but the category itself must be handed to the new node. Considering that, the P2P network of the proposed system can be constructed as a CAN [9].

Our P2P network is constructed with 2-dimensional coordinate space $[0,1] \times [0,1]$ to store exercise categories, as shown in Figure 2. The figure shows the situation that node C has just joined the system as the third node. Before node C joins, node A and node B shared the whole coordinate space half and half. At that moment, node A managed "Math/Geometry", "Math/Statistics", and "History/Rome" categories and node B managed "English/Grammar", "English/Reader" and "History/Japan" categories, respectively. When node C joins the system, we assume that node C already knows IP addresses of some nodes in the system and node C sends the join request to some node in the list. Then node C is mapped on a certain coordinate space according to a random number and takes on corresponding categories from another node. For example, in the case of Figure 2, node C takes on the "History/Japan" category from node B, then exercises of the category move to node C. After joining, node C gets a list of IP addresses of neighbor nodes in the coordinate space, such as node A and node B. Therefore, neighbor nodes can communicate with each other.

C. Components of System

In order for the proposed system to be considered as a distributed WBT system, it is not enough that only exercises are distributed among all nodes. Functions to provide the above services must be also distributed among all nodes. Mobile agent technology to achieve this goal is adopted.

There are following agents and user interface programs on
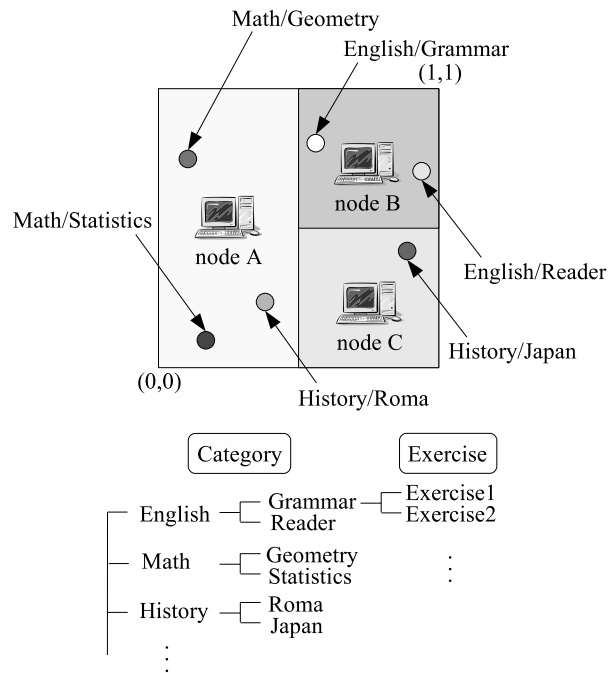

Fig. 2. P2P network of the proposed e-Learning system.

each node. These agents have implemented in the mobile agent framework that we have developed [10].

- **Node Agent:** Each node has one node agent. It manages the zone information of a CAN and forwards messages to the Category Agents in the node.
- **Exercise Agent:** Each Exercise Agent has questions and functions to score user's answers, to tell the correct answers, and to show some related information about the exercise.
- **Category Agent:** Each Category Agent stands for a unit of a particular subject. It manages Exercise Agents in itself and sends them to the requesting node.
- **Interface Agent:** There is one interface agent for each user interface, such as a student interface and an exercise manager interface on each node. It plays a role of interfaces 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 its user's information that includes login name, password, IP address of the user's computer, online/offline status, and log of studying or a list of created exercises.
- **Student Interface:** One student interface is on each node of which a user logs in as a student. It is a user interface program for studying.

III. OUR RECOVERY APPROACH

In the proposed system, the Distributed Hash Table(hereafter we abbreviate as DHT) that does the mapping on the coordinate space two dimensions is used, the area of DHT is given to the node which participates in the system. As a result, the node compose a P2P network, where the contents are managed.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

However, when a node failure occurs, the exercises in the node cannot be studied by anyone and the reference by the DHT doesn't work correctly because of the disappearance of contents and the lack of the DHT. In order to resolve this issue, the method of backup by a node and the method of backup by a category are designed.

A. Backup by Node

1) Management of Backup: In the method of backup by a node, one of the neighbor nodes of each node takes its own backup. If node n takes a backup of node m, node n is called a backup node for node m, while node m is called an original node for node n. The backup is managed by the following procedures.

- **Making of Backup** When a node participates in the network, the backup is made.
    1) The number of contents that all neighbor nodes manage is investigated.
    2) A neighbor node that has the minimum number of agents is selected for a backup node to balance the load of the system
    3) The participated node commissions the management of backup to the neighbor node.
- **Update of Backup** When the area that a node manages changes, the backup is updated.
    1) The backup node is requested to delete an old backup.
    2) The backup of node that the area was changed is made again.
- **Delete of Backup** When the node leaves the network, the deletion of the backup is requested.

2) Update of Network Information: When a node failure occurs, a network information is used to restore an original node. To use this method, the network information should be kept in the latest state. By the update request of the routing table that an original node sent, if the routing table is changed, the routing table of the network information in the backup is updated.

3) Recovery from Node Failure: A node failure can be detected, because an original node and a backup node watch it each other. Both of the nodes receive the update request of the routing table from the other node periodically, if the update request isn't performed during the fixed time, it is judged whether or not a node failure occurs. If a backup node detects the failure of its original node, the following steps are performed.

1) The backup node generates a temporary node from the backup of the original node.
2) The temporary node executes the leaving procedure. Consequently, the managed zone of original node is formally handed over to one of the neighbor nodes.

If an original node detects the failure of its backup node in the contrast, it simply makes a new backup of itself. In this method, it is impossible for the failures that occurs in both an original node and its backup node at the same time to be recovered. The method can be recovered from the other types of combinations of failures.

B. Backup by Category

1) Management of Backup: In the method of backup by a category, an original and a backup of each category are prepared, and they are mapped on the DHT. However, if the original and the backup categories are managed in the same node, it is not possible to recover when the failure occurs. In order to resolve this issue, each pair of the original and the backup categories are mapped symmetric to the point[0.5, 0.5] on the DHT, as shown in Figure 3.
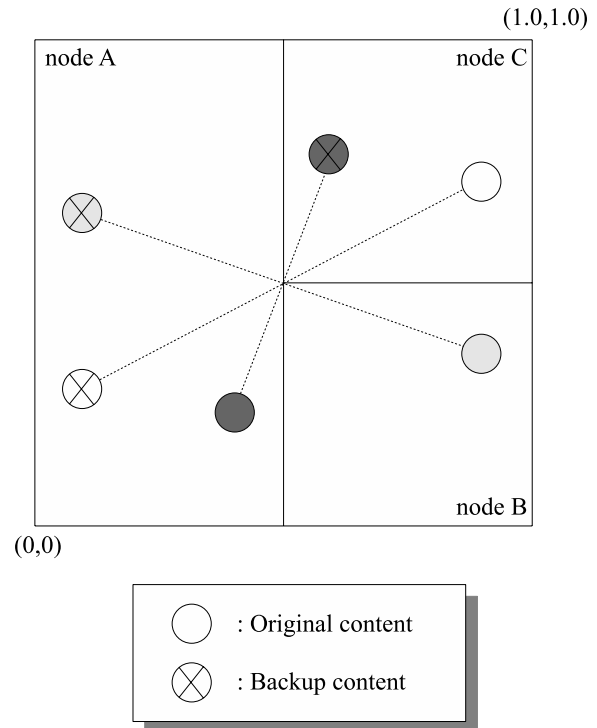

Fig. 3. Each pair of categories are point symmetric to the point [0.5, 0.5].

2) Recovery from Node Failure: The recovery method of a category is the same as the method of a node excluding following points.

- The node responsible to recover a failure node back to normal is one of a neighbor nodes of a failure node.
- Every category which was the pair of lost category must be doubled on the temporary node.

In this method, as well as the method of a node, it is possible to recover from failures unless the failures occur in both an original node and its backup node at the same time.

IV. Experiment

This section presents the experimental results for comparison of performance in the distributed backup method, and comparison of performance of the failure recovery according to the distributed backup method. Table I shows the machine specification.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:7, 2009

TABLE I
THE MACHINE SPECIFICATION.

CPU	Intel Pentium4 3.0GHz
Memory	1GB
Network	1000Base-T
OS	TurboLinux 10 Desktop

A. Performance Comparison of Participant

We investigate how the total amount of bytes of transferred agents changes in participation of agents, with or without the backup scheme. Table II shows the experimental conditions. The experiment is done according to the following procedures.

1) An initial node is started.
2) A second node is participated in the system. At this time, the traffic generated among all nodes that participate is measured from the start of the request for participation to the end of the participation processing.
3) Until the 30th node participants, the measurement is performed as above.
4) The method of backup by a node, the method of backup by a category and the method without backup, processing from the above 1 to 3 is done by each method, and the ratio of the traffic is calculated based on the result of the measurement.

Figure 4 shows the ratio of the total amount of transferred agents with and without the backup scheme. The horizon expresses the average of the ratio of the traffic when it used the method with and without backup, and the solid line expresses the average by the method of backup by a node, and the dotted expresses the average by the method of backup by a category. As a result, in comparison with the method without backup, the traffic by the method of backup by a node is 2.4 times, and the traffic by the method of backup by a category is 1.8 times.

TABLE II
THE EXPERIMENTAL CONDITIONS.

Number of node	30
Number of category	50
Number of exercise	150

B. Performance Comparison of Recovery

We investigate how the total amount of bytes of transferred agents changes in failure recovery, with or without the backup scheme. The experimental conditions is similar to section 4.1 except that the number of the nodes participate in the system become 7. The experiment is done according to the following procedures.

1) An initial node is started.
2) Nodes participate in the system to the seventh.
3) A failure is caused at the second node.
4) The required time and the ratio of traffic is measured from the start of the recovery to the end of it.
5) The method of backup by a node and the method of backup by a category, are processed as the above 1 to 4 respectively.
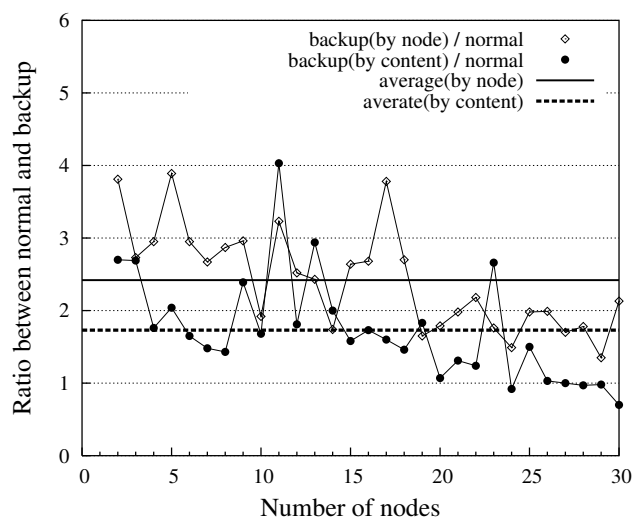

Fig. 4.   Ratio of the total amount of transferred agents.

6) The traffic in the normal state is measured.
7) The ratio of the traffic is calculated based on the result of the measurement.

Table III shows the experimental result. The required time and the ratio of traffic is almost the same in each method of backup. When the failure node is restored, the method of backup by a node performs all processing, the method of backup by a category is processed with more than two nodes. As a result, the influence that the method of backup by a category will be a little in comparison with the method of backup by a node.

TABLE III
THE REQUIRED TIME AND THE RATIO OF TRAFFIC.

	By a Node	By a content
The Required Time	143 sec	131 sec
The Ratio of Traffic	3.3 times	3.2 times

V. CONCLUSION

Two backup and recovery methods for our distributed e-Learning system have developed to prevent agent's disappearance. Each agent's backup has distributed to the node besides the node that manages the agent. Other nodes have managed the backup of the routing table of each node.

As a result, the agent with the node doesn't disappeared even though the node shuts down suddenly. In comparison with the method of backup by a node and the method of by a category, the method of backup by a category takes somewhat shorter time for making backup and recovering than that of a node. Because of the method of backup by a category processed by more than two nodes when the failure node is restored, the load given to the system is distributed. Therefore, the method of backup by a category is adopted in our e-Learning system.

In future work, we consider that when an agent disappears during transfer and how the number of backup is prepared in a failure recovery.

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).

[10] Motomura, S., Kawamura, T. and Sugahara, K.: Logic-Based Mobile Agent Framework with a Concept of Field , *IPSJ Journal*, Vol. 47, No. 4, pp. 1230-1238 (2006).

Kazunari Meguro was born in 1977. He received his B.Eng. and M.Eng. degrees from Hiroshima city University, Japan, in 2002 and 2004, respectively. He is currently a Ph.D. student in Tottori University. His research interests multi-agent systems and distributed systems.

Shinichi Motomura was born in 1973. He received his B.Eng. and M.Eng. degrees in Computer Engineering from Toyohashi University of Technology, Japan, in 1995, 1997, respectively. From 2004 to 2008, he joined Tottori University as a research associate. In 2008, he received the D.Eng. degree from Tottori University, Japan. Since 2008 he had been in Tottori University as an associate professor in the Information Media Center. His research interests include multi-agent systems and distributed systems.

Takao Kawamura was born in 1965. He obtained his B.Eng., M.Eng. and Ph.D. degrees in Computer Engineering from Kobe University, Japan, in 1988, 1990 and 2002, respectively. Since 1994 he had been in Tottori University as a research associate and has been in the same University as an associate professor in the Faculty of Engineering since 2003. His current research interests include mobile-agent systems and distributed systems. Dr. Kawamura is a member of IPSJ and IEICE.

Kazunori Sugahara received the B.Eng. degree from Yamanashi University, Japan, in 1979 and M.Eng. degree from Tokyo Institute of Technology, Japan, in 1981. In 1989, he received the D.Eng. degree from Kobe University, Japan. From 1981 to 1994, he was on the staff of the Department of Electronic Engineering, Kobe City College of Technology. In 1994, he joined Tottori University as an associate professor of the Department of Electrical and Electronic Engineering and he is a professor of the department of Information and Knowledge Engineering. His current interest lies in the fields of computer architectures and hardware realizations of image processing algorithms. Prof. Sugahara is a member of IEEE, IPSJ and IEICE.