

Technical Note

Backup and Recovery Scheme for Distributed e-Learning System

Kazunari MEGURO ,[†1] Shinichi MOTOMURA ,[†2]
Toshihiko SASAMA ,[†3] Takao KAWAMURA [†3]
and Kazunori SUGAHARA[†3]

In this paper, we present a backup 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 functions are realized on mobile agents. These agents are distributed to computers, and using a Peer to Peer network that modified Content-Addressable Network they can obtain. 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 problem, backups of agents are distributed to computers. If failures of computers are detected, other computers will continue service using backups of the agents belonging to the computer. The developed algorithms are examined by experiments.

1. 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 in his own time and schedule, without live interaction with an 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. In the client/server model, a server machine offers all functions and manages all data. The client/server model has advantages of easy construction and maintenance, however, the client/server systems generally lack scalability and robustness.

There is a Peer to Peer (hereafter we abbreviate as P2P) model to supplement the disadvantages of the client/server model. There is a feature in the system based on the P2P model that each computer works as a client and a server. The feature can distribute the load to a node. Moreover, 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 of a server. That is to say, 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. When a node joins in the system, the part of contents is given from a joined node, and has the responsibility of sending appropriate contents to requesting nodes. In addition to the above advantages of using P2P architecture, we can construct the proposed e-Learning system at low cost because the system needs 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, giving the correct answers, and showing some related information without human instruction.

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

In this paper, as a solution for this problem, we propose an e-Learning system where backups of agents are distributed to nodes. In this system, if a node failure occurs, another node will continue service using the backup of the agent belong 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 con-

†1 The Graduate School of Engineering, Tottori University
†2 Information Media Center, Tottori University
†3 Faculty of Engineering, Tottori University

cluding remarks are made in Section 5.

2. Proposed e-Learning System

2.1 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. That is to say, when a node joins in the system, the part of categories and exercises is given from a joined node and and has the responsibility of sending appropriate exercises to requesting nodes.

The important point to note is that the categories a node has are independent of the categories in which a 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.

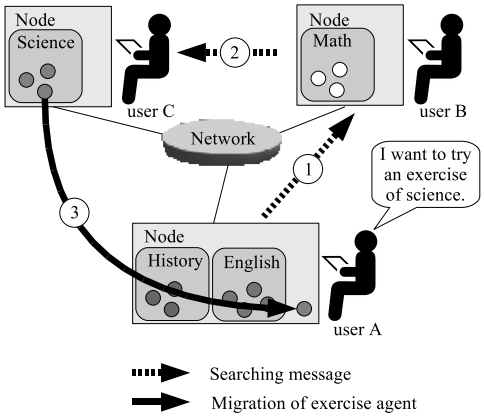


Fig. 1 Migration of exercise agent.

2.2 P2P Network

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

We would like to emphasize that 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. Accordingly, files are originally distributed among all nodes. On the other hand, the categories in the proposed system are originally concentrated. Consequently, when a new node joins in the system, not only location information of a category but the category itself must be handed to the new node. That being the case, 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 categories, as shown in Figure 2. The figure shows the situation that node C has just joined in 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 time, 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 in the system, we assume that node C already knows the IP addresses of some nodes in the system and node C sends a 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 in that 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.

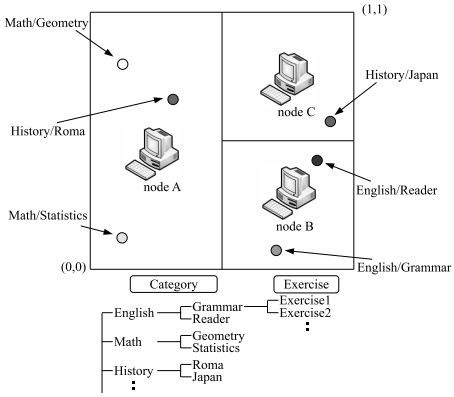


Fig. 2 P2P network of proposed e-Learning system.

2.3 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 also be distributed among all nodes. We adopt mobile agent technology to achieve this goal.

There are following agents and user interface programs on 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 Content-Addressable Network (hereafter we abbreviate as 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, give the correct answers, and 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 the 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 to which a user logs in as a student. It is a user interface program for studying.

3. Our Recovery Approach

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

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 the contents and the lack of the DHT. In order to solve this problem, the method of backup by a node and the method of backup by a category are designed.

3.1 Backup by Node

3.1.1 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 joins 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 joined 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 the node where the area was changed is made again.
- **Delete of Backup** When the node leaves the network, the deletion of the backup is requested.

3.1.2 Update of Network Information

When a node failure occurs, 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, when there is a change in routing table, the routing table of the network information in the backup is updated.

3.1.3 Recovery from Node Failure

A node failure can be detected, because an original node and a backup node both observe it. That is, 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 the 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.

Naturally, it is impossible for a failure that occurs in both an original node and its backup node at the same time to be recovered. Our system can be recovered from the other types of failures.

3.2 Backup by Category

3.2.1 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 impossible to recover when the node failure occurs. Accordingly, each pair of the original and the backup categories are mapped symmetric to a point [0.5, 0.5] on the DHT, as shown in Figure 3.

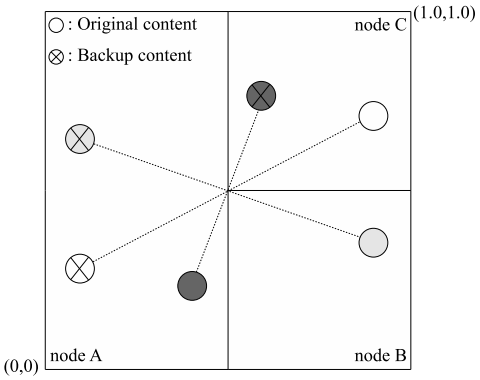


Fig. 3 Each pair of categories is pointed symmetric to point [0.5, 0.5].

3.2.2 Recovery from Node Failure

The recovery method by a category is the same as the method by excluding following points.

- The node responsible for recovering categories is one of the neighbor nodes of a failure node.
- Every category which included the pair of the lost category must be doubled on a temporary node.

With this method, as well as the method by 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.

4. Experiment

This section presents experimental results for comparing the performance of the distributed backup method, and also the performance of the failure recovery according to the distributed backup method. Table 1 shows the machine specification.

Table 1 Machine specification.

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

4.1 Performance Comparison of Participant

We investigate how the total number of bytes of the transferred agents changes in the joining of agents, with or without the backup scheme. Table 2 shows experimental conditions. The experiment is done according to the following procedures.

- An initial node is started.
- A second node is joined in the system. At this time, the traffic generated among all nodes is measured from the start of a request for joining to the end of the joining process.
- Until the 30th node joins, the measurement is performed as above.
- 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 traffic is calculated based on the result of the measurement.

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

4.2 Performance Comparison of Recovery

We investigate how the total number of bytes

Table 2 Experimental conditions.

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

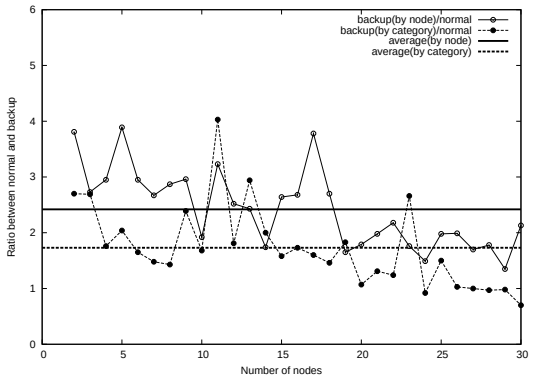


Fig. 4 Ratio of total amount of transferred agents.

of transferred agents changes in failure recovery, with or without the backup scheme. Experimental conditions are similar to section 4.1 except that the number of join nodes in the system becomes 7. The experiment is done according to the following procedures.

- An initial node is started.
- Nodes join in the system to the seventh.
- A failure is caused at the second node.
- The required time and the ratio of the traffic is measured from the start of a recovery to the end of it.
- In the backup method by a node and the backup method by a category, the process from the above 1 to 4 is done by each method.
- The traffic in a normal state is measured.
- The ratio of the traffic is calculated based on the result of the measurement.

Table 3 shows the experimental result. The required time and the ratio of traffic is almost the same in each backup method. When the failure node is restored, the backup method by a node performs all processing, the backup method by a category is processed with more than two nodes. As a result of the experiment, the influence of the backup method by a category will be small in comparison with the backup method by a node.

5. Conclusion

We have developed two backup and recovery methods for our distributed e-Learning system to prevent agent's disappearance. Each agent's backup is distributed to nodes which don't manage the original agent. Moreover, other nodes manage the backup of the routing table of each node. As a result of this, contents that nodes manage don't disappear even though the node shuts down suddenly.

Table 3 Required time and ratio of traffic.

	By Node	By Category
Required Time	143 sec	131 sec
Ratio of Traffic	3.3 times	3.2 times

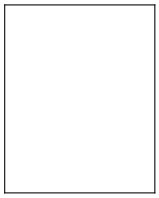
As a result of the experiments, the backup method by a category takes a somewhat shorter time for making backup and recovering compared with the backup method by a node. Moreover, when a content is restored, the backup method by a category is processed with more than two nodes, and the load given to the system is distributed. Therefore, the backup method by a category is adopted in our e-Learning system.

References

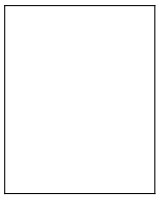
1) Nishita, T. and et al,: Development of a Web Based Training system and Courseware for Advanced Computer Graphics Courses Enhanced by Interactive Java Applets, *Proceedings of International Conference on Geometry and Graphics*, Vol. 2, pp. 123-128 (2002).
2) Homma, H. and Aoki, Y.: Creation of WBT 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).

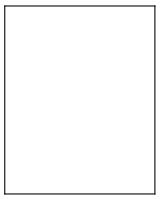
(Received May 25, 2009)
(Accepted August 7, 2009)



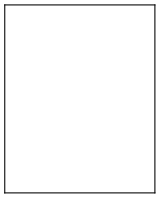
Kazunari Meguro was born in 1977. He received his B.Eng. and M.Eng. degrees from Hiroshima city University, Japan, in 2002 and 2004, respectively. He is currently a Ph.D. student in Tottori University. His research interests include multi-agent systems and distributed systems.



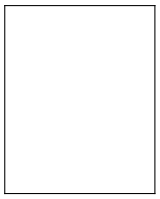
Shinichi Motomura was born in 1973. He received his B.Eng. and M.Eng. degrees in Computer Engineering from Toyohashi University of Technology, Japan, in 1995, 1997, respectively. From 2004 to 2008, he joined Tottori University as a research associate. In 2008, he received the D.Eng. degree from Tottori 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.



Toshihiko Sasama was born in 1972. He received his Ph.D. degree from Osaka University, Japan, in 2001. He is now an assistant professor in the Department of Information and Knowledge Engineering of Tottori University. His current research interests include mobile ad-hoc networks. Dr. Sasama is a member of IPSJ and IEICE.



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 a professor in the Faculty of Engineering since 2008. 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.