EECS 426

Multimedia Streaming with Caching on Pure P2P-based Distributed e-Learning System using Mobile Agent Technologies

Masayuki Higashino
Tottori University

Tadafumi Hayakawa
Tottori University

Kenichi Takahashi
Tottori University

Takao Kawamura
Tottori University

Kazunori Sugahara
Tottori University

Abstract

This paper presents a method for streaming multimedia contents with caching on a pure P2P-based distributed e-Learning system using mobile agent technologies. The e-Learning system contains a lot of mobile agents who have study contents (e.g., texts, sounds, movies, etc.) and functions (e.g., scoring, showing question, correct answers, etc.), and are mapped on DHT (Distributed Hash Table). However, when these systems handle multimedia data (e.g., sounds and movies), the user cannot start quickly learn with at the first parts of the contents; because multimedia data may be a huge size of data, and the agents contain these data, and the user cannot start learn until the mobile agent finished to migrate. In order to solve this problem, we divide multimedia data into fragments and prepare mobile agents which manages each fragments, and lookup location of agents before users request, and make the computers cache these agents when agents migrate.

Keyword: e-learning system, mobile agent, distributed system, P2P, multimedia streaming.

1. Introduction

E-Learning systems, especially asynchronous Web-Based Training systems (WBT) [1, 2] are very popular. A WBT does not require interactions with instructors and allows a learner to study on his/her own time and schedule. The mainstream e-Learning systems 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 contents by the server machine. Although the client/server model has an advantage of easy construction and maintenance, however, the client/server model requires expensive initial investments in order to achieve fault-tolerance. Furthermore, in cases where the systems is discontinued due to economic or social reasons, the systems infrastructures and communities will be lost in a moment. Such systems are not open learning infrastructures in that the system administrators would restrict the system user's publications or subscriptions. It is desirable that learners be able to get learning opportunities freely in a low-cost way.

In order to realize such e-Learning systems, P2P (Peer to Peer) technologies are well suited. In a pure P2P architecture, every user's computer (hereafter we refer to such a computer as a node) plays the role of a client or a server, thus users are able to publish or subscribe contents freely. While a user uses the system, user's node is a part of the system. The systems can work with not only a stand-alone node but also a cluster of multiple nodes that communicate through a network. The nodes maintain an equal relationship each other. Therefore, the systems have no a center server, and the systems are able to have fault-tolerance. Moreover, the systems are able to coordinate and manage its performance and required investment, in accordance with the number of user's nodes. However, in a pure P2P e-Learning system, given that all nodes belong its end-users, each and every node is not a high-performance machine such as a centralized server. Thus, the systems need (e.g., CPU time, memory usage, disk usage, network band wide, etc.) to balance load into each nodes. If not, teacher's nodes will go down because teacher's nodes that hold contents have a higher than student's nodes loads. For these reasons, a pure P2P e-Learning system is required high-flexibility and high-scalability.

In this paper, we propose basic functions of a pure P2P based e-Learning system, and focus on an efficient method of multimedia contents management in particular.

2. Pure P2P-based Distributed e-Learning System

2.1 Overview

All of learning contents (exercises) in the proposed system are classified into categories, such as "Math/Statistic", "English/Grammar", etc. A user can obtain exercise one after another through the category. While a user joins the e-Learning system as a client, his/her node must be a part of the system as a server. When it joins the system, the node receives some number of categories and exercises from other node. Then, it has responsibility to send appropriate exercises request to requesting nodes. The categories managed in each node are independent of categories in which the node's user is interested, as shown in Figure 1.

Figure 1 illustrates that the request from user A is forwarded first to neighbor nodes, then their neighbor nodes forward the request to next neighbor nodes, and finally, the request reaches to the node which manages a category requested. These exercise locations are managed by CAN (Content Addressable Network) [3] as DHT (Distributed Hash Table).

2.2 Components

We have to consider not only the distribution of learning contents but also the distribution of functions to provide above services. Our system consists of mobile agents and user interface program. In our system, functions are provided as the part of mobile agent. A mobile agent migrates from one node to another node with a function it provides. These mobile agents are implemented in the mobile agent framework called Maglog [4]. Figure 2 shows an interaction between agents.

Node Agent (NA): 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 (EA): 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 (CA): One category agent is installed in one learning category. A category agent manages exercise agents related on its category and dispatches the learning request to them.

User Agent (UA): Each user has its own User Agent. A User Agent manages user's learning information that includes login name, password, IP address of the user's computer, online/offline status, and log of learning and/or a list of created exercises.

Group Agent (GA): Each group agent manages user agents.

User Interface: One user interface is on each node which a user logs in as a student. It provides a user interface program for learning.

Interface Agent (IA): There is one interface agent for each user interface, such as a student interface and an exercise manager interface on each node. It intermediates between the interface program and agents, and between agents and applications.

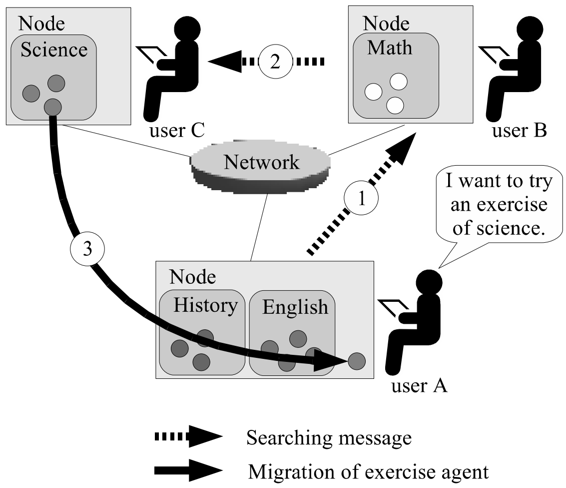

Figure 1. System overview.

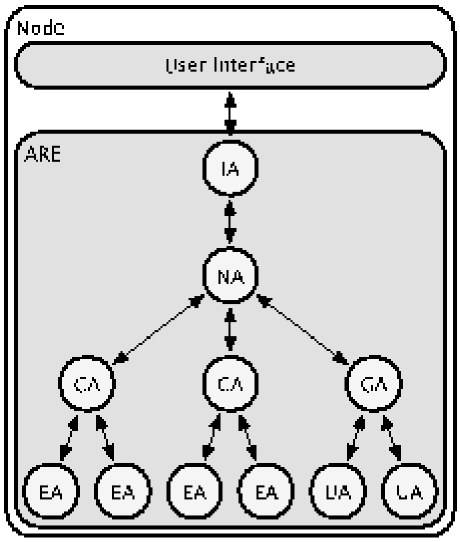

Figure 2. Interactions between Agents.

3. Management of Streaming Multimedia Data

3.4 Content Distribution

In the Pure P2P-based distributed e-Learning system, when considering a location where content is stored, as one example, there is a way that a teacher stores accountably contents into own node. However, in this way, teacher's node will be SPOF (single point of failure). For example, if a teacher provides highly popular contents then the concentrated load occurs at the teacher's node. Additionally, when a failure occurs at the teacher's nodes, or the teacher's nodes go offline, students cannot access the contents. Therefore, the learning contents must be distributed in a network of the e-Learning system.

3.5 Multimedia Content Distribution

If a learning content is a multimedia such as a video, a size of a multimedia will be reached MiB or GiB. There really is difference of data size between a text content and a video content. Thus, simply distributing a learning content as a unit of a file or a group, the inequality of resource usage (e.g., CPU time, memory usage, disk usage, network band wide, etc.) between nodes occurs. Additionally, in the case of a transfer of a text content, a transfer from a content requested node to a content request node is finish at short time. However, in the case of a video content, it takes a long time. This means that the concentrated load, a concurrent video streaming processing, occurs at a node holding a video contents.

Therefore, if a learning content contains a multimedia data, then the multimedia data is divided to small fragments, and this learning content has a description of a multimedia and references of locations of these fragments. Thus, the possibility that a long time and multiple load occurs at a node which requested a multimedia content is low, because this node transfers only of an description and references.

3.6 Distributed Multimedia Content Streaming

Each fragment is mapped on DHT based on its keys. Since a multimedia content is composed of some fragments, we have to know the location of these fragments on DHT. Therefore, each fragment has the key of a next fragment. Then, each fragment can find next fragment according to its key. However, it needs a lot of message to find a node which manages next fragment. Because our system uses 2-dimensional CAN, ($O(\sqrt{n})$ messages are required to find next fragment. Here, n is the total number of nodes. This may impede smooth playing of the multimedia data. To solve this issue, before the multimedia content is required from learners, each fragment finds a node which manages next fragment and records its location. Thus, every fragment is linked in the order of time series of the multimedia data. Consequently, we can reduce messages to find the location of fragments except first fragment's search.

When a node joins or leaves (churn), fragments are transferred to other nodes. If a fragment is transferred, the link between fragments becomes useless anymore. To keep the link, when a node joins or leaves, each fragment records not only the location of a next fragment but also a previous fragment. When a fragment is transferred, the fragment notifies its new location both to a next and previous fragment. Thus, the link between fragments is kept continuously even when a node joins or leaves.

3.7 Distributed Multimedia Content Caching

When a user learns using a multimedia content, fragments of the multimedia content temporarily gather in the learner's node. Therefore, this node can be used as a cache node. When a user learns using a multimedia content, the user's node stores the caches of the fragments into own node; and a node that has an original fragment, stores references of cache stored nodes. In this case, the node with original fragments collects references of caches of fragments. Thus, this node can balance a self-load like a DNS round robin shown as Figure 3.

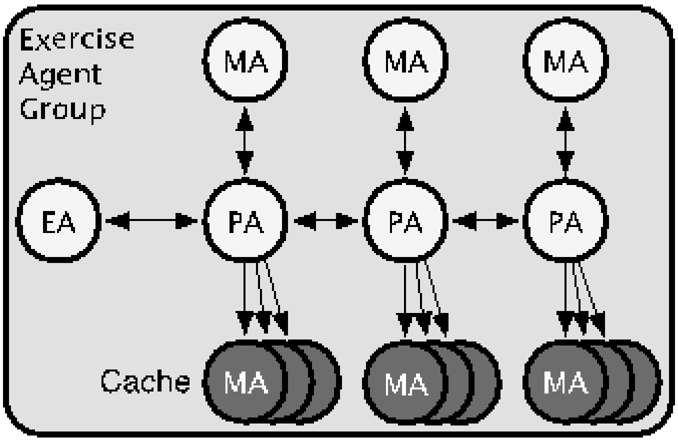

Figure 3. Interaction between agents.

4. Experiment

We have investigated the effectiveness of distributed multimedia data cache. This experiment has done with proposed e-Learning system that runs on computers (Intel Core i5 Processor 3.2 GHz, 4 GiB RAM, JRE 1.6, Debian GNU/Linux 5.0.5) connected through Ethernet of 100 BASE-T. In this experiment, we have defined a 5.6 MiB sized video file, and its file is divided into three fragments. These fragments are distributed to 3 nodes.

Figure 4 shows the result of this experiment. When the number of simultaneous requests to get a fragment from distributed nodes is 1 or 2, the response time of *with cache* is longer than *without cache*. Because, in spite of the requested node has no large load, the requested node routes these requests to other nodes which have cache of fragment of video file. However, when the number is greater than or equal to 3, the response improving. Thus, greater the number of requests, the cache is more effective.

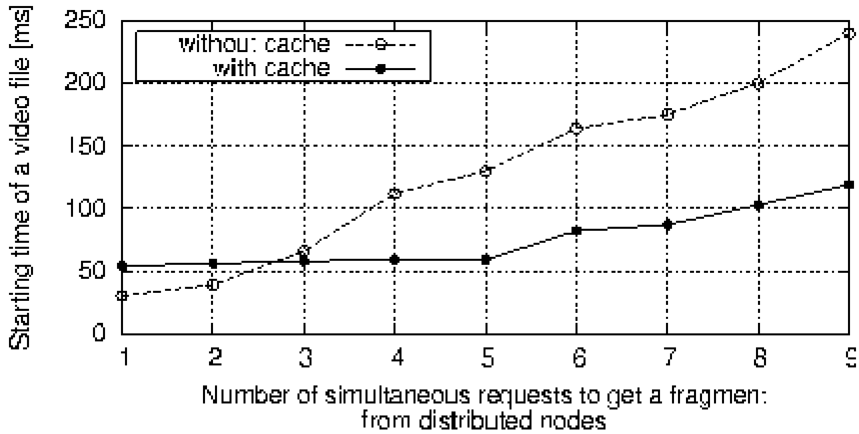

Figure 4. Comparison of response time between with cache and without cache.

5. Related Works

Several researches have proposed P2P-based e-Learning system. EDUTELLA [5] is a one of the earliest a P2P-based e-Learning system. This is based on JXTA [6]. JXTA is a P2P application framework. There are JXTA-based e-Learning systems such as PLANT [7] and HYDRA [8]. In addition, several researches have proposed a new overlay network for a P2P-based e-Learning system such as PROSA [9] and PeerLearning [10]. These systems are different from our proposed system in that multimedia contents are divided into fragments and managed by DHT. [11] has proposed an index caching mechanism for CAN. In the other hand, our proposed system caches not only indexes but also data of the learning contents. [12] has proposed an agent-based collaborative virtual environment architecture using grid technologies, however, this is not based on a pure P2P model. On the other hand, our proposed system is based on a pure P2P model.

In Nearcast [13], a locality-aware P2P live streaming method has proposed. However, in a live streaming, a source node will be a single point of failure. If the source node forced outage, the streaming will stop. On the other hand, in our proposed system, a multimedia content is preliminarily divided and is preliminarily distributed into many nodes before all node is notified of the multimedia content existence. Therefore, the multimedia content streaming will not be forced outage by economic or social reasons.

This characteristic is similar to P2P-based file sharing systems; but these systems do not expect to the multimedia streaming and cannot stream multimedia because the fragments are transferred in random order.

6. Conclusion

In this paper, a method to play multimedia data smoothly on a pure P2P-based distributed e-Learning system is proposed and its implementation is described. In our system, multimedia data is divided into multiple fragments by time series, each media agent manages each fragment, and their media agents are linked bidirectional. These agents are cached at nodes when users request learning contents. If division size of multimedia data is huge, multimedia data cannot be played smoothly because a lot of agents have to migrate to a requesting node simultaneously. Therefore, we devised the timing of sending a requesting message to next agent.

7. References

[1] Driscoll M. Web-Based Training: Creating e-Learning Experiences. 2nd Edition, John Wiley & Sons, New York, USA, 2002.

[2] Uskov V. Web-based Education. ACTA Press, Calgary, Canada, 2010.

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

[4] Motomura S, Kawamura T, Sugahara K. Logic-based mobile agent framework with a concept of “field”. *IPSJ Journal*, 2006, 47(4): pp.1230-1238.

[5] Nejdl W, Wolf B, Qu C, Decker S, Sintek M, Naeve A, Nilsson M, Palmer M, Risch T. EDUTELLA: a P2P Networking Infrastructure Based on RDF. Proceedings of the 11th International Conference on World Wide Web}, 2002, pp.604-615.

[6] Verstrynge J. Practical JXTA II. Lulu Enterprises Inc., 2010.

[7] Li M, Zhu H, Zhu Y. PLANT: a distributed architecture for personalized E-Learning. Proceedings of the 6th International Conference on Advances in Web Based Learning, 2007, pp.20-30.

[8] Zualkernan I A. HYDRA: A Light-Weight, SCORM-Based P2P e-Learning Architecture. Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies, 2005, pp.484-486.

[9] Carchiolo V, Longheu A, Mangioni G, Nicosia V. Adaptive E--Learning: An Architecture Based on PROSA P2P Network. Proceedings of the 21st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: New Frontiers in Applied Artificial Intelligence}, 2008, pp.777-786.

[10] Wang G, Yuan Y, Sun Y, Xin J, Zhang Y. PeerLearning: A Content-Based e-Learning Material Sharing System Based on P2P Network, World Wide Web, 2010, Vol.13, No.3, pp.275-305.

[11] Tagashira S, Shirakawa S, Fujita S. Proxy-Based Index Caching for

Content-Addressable Networks. IEICE - Transactions on Information and Systems,}, 2006, E89-D(2): pp.555-562.
[12] Huang C, Xu F, Xu X, Zheng X. Towards an Agent-Based Robust Collaborative Virtual Environment for e-learning in the Service Grid. Proceedings of the 9th Pacific Rim International Conference on Agent Computing and Multi-Agent Systems, 2006, pp.702-707.
[13] Tu X, Jin H, Liao X, Cao J. Nearcast: A Locality-Aware P2P Live Streaming Approach for distance education. ACM Transactions on Internet Technology (TOIT), Vol.8, No.2, Article 7, 2008, pp.7:1-7:23.