

Implementation of an academic system to connect foreign countries with quick performance over WWW.

Yutaka Katsuyama, Osanori Koyama, Chen Ying, and Chen Kee

Department of Electrical and Electronic Systems

Graduate School of Engineering, Osaka Prefecture University

katsu@neptune.ees.osakafu-u-u.ac.jp, koyama@neptune.ees.osakafu-u-u.ac.jp,

chen-y@neptune.ees.osakafu-u.ac.jp

Abstract

We have designed and implemented a file cache function in an academic system over WWW so as to connect different universities located in different countries. In the file cache scheme, access number counting and automatic file transfer procedures were adopted to obtain the quick performance. The system performance was evaluated, including the retrieving speed improvement for the students to download learning materials stored in the other local server connected with common ISDN-BR (64kbps). By this function, the academic system can be used internationally with the quick performance over WWW.

1. Introduction

Recent progress of information transmission and software technologies makes it possible to develop effective information systems to supply various types of information for different purposes. As for educational applications, many systems have been reported, where the information is transmitted by a satellite [1], or the internet [2]. We designed and implemented such an education system by a stored data approach that students can retrieve learning materials and exercises by a web browser over the world-wide-web (WWW) and can study by themselves [3]. We also designed and implemented remote measurement and calculation functions over WWW to support research activities in universities [4]. We added the functions to our education system to form an academic system, unifying all the functions for education and research supports. For example, teachers can author teaching materials by using the graphs obtained by the remote measurement function. This enables us to use the latest information obtained in research activities and provide it to students by this system.

It is effective for this system to be operated so as to connect different universities located in different countries. However, the international connections are low speed occasionally. In such a system, one of the most important functions is the quick transmission of required information. One typical example is the problem on the retrieving: it takes longer time for students to download learning materials, if they want the materials stored in the server in foreign countries. To solve this problem we designed a file cache scheme by access management and automatic file transfer approaches. The procedures were designed and implemented to operate the system with one central and 2 local servers. The system performance was evaluated, including the retrieving speed improvement for the students to download learning materials stored in the other local server connected with common ISDN-BR (64kbps). The design consideration and the system performance are described.

2. System design and configuration

In the existing system [3], the main function is the capability to author teaching materials, and to supply them over WWW for students to study either inside or outside university. We re-designed this system so as to add the file cache function for foreign students to download the teaching materials with quick performance. The teaching materials were designed to be a presentation form [3] consisting of digital slides, which were authored by inserting multimedia elements, such as figures, tables, photographs, and animations. Thus, a presentation is stored as files corresponding to the constituent digital slides and the data files for the administration. Teaching materials are assumed to be in such a presentation form throughout this paper.

2.1. System configuration

Figure 1 shows the system configuration. We designed this system so as to be used internationally under distributed server environment. The central server manages the whole system to be operated properly, and the local servers process the tasks requested by the local users. The central and local servers have a web server, central or local server application (AP) and database (DB), respectively. The APs are composed of some modules (M.) to process particular tasks. Each local server has a file server to store the files of the teaching materials. The central and local servers are connected with an intra-server network, which is either ISDN, a leased line, or a private network. When a web server receives a request in HTML format sent by a web browser in a client, the server AP and the web server convert the request to the database access instructions and send them to the database server. The database server processes the instructions and sends the result back. The result is converted to HTML format and is sent to the user.

The central and local APs were designed in such a way that the request from a client is processed in the central server or transferred to a local server via system and local interfaces (IFs), depending on the type of tasks and data. The location where the data should be stored is based on such a design consideration [3] as to minimize the system maintenance work. As a first step, all the logging-in requests and the authentication are executed in the central server.

We allocated one process client machine for the file cache processing in each university. After any users are authenticated by the server AP in the central server, the users can download the teaching materials stored in the local file server or other teaching materials stored in other local file servers. The access is detected in the server AP and the access data are analyzed and saved. Each process client contains a cache program, which manages and processes the access number evaluation and the file transfer, as described below.

2.2. File cache scheme

Figure 2 shows the file cache scheme, illustrating 2 local servers for universities A and B. Whenever a client (CL) gets access to a teaching material, the server AP detects the access. The server AP checks each access in the background and save the data, if the access should be counted. In the process client (process CL), a cache program is installed. At every midnight, the cache program is executed automatically and sums up the access data. In University A, the access number N_B should be paid a special attention. Here, N_B means the number for the University A file which was gotten access by the clients in University B. When N_B is bigger than a threshold number N_0 , the files are transferred to a

particular directory A of the local server in University B. At the same time the cache program informs this event to the server AP in University B, and the cache status is saved in the DB in University B. This cache status information includes the ID of the transferred presentation and the university ID to indicate the university where the files are stored originally. When a client in University B wants to download the presentation, the server AP checks the cache status. If the presentation is included in the status data, then the server AP download the file from the file server in University B.

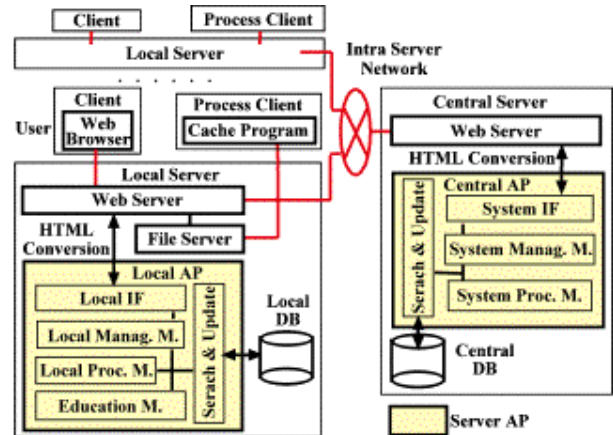


Fig.1 System configuration

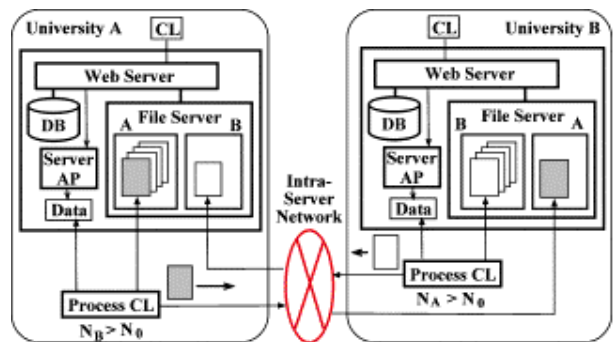


Fig.2 File cache scheme

Table 1 Conditions for access management

Item	Condition
Cache Unit	Presentation
Access Counting Unit Period	Counting/Day
Access Counting Period	2 Days
Data Holding Time	No Limitations
Access Counting	First Access/Day = 1 Access Additional Access by the Same Person - Within 2 Hours = No Counting - After 2 Hours = 1 Access
Data to be saved	Presentation ID, University ID, Account ID, Access Date & Time
Threshold Access Number N_0	10 Accesses / Presentation
Timing	Transferred at Midnight

Thus, the cache program together with the server AP manage the access counting and the file transfer for the quick performance.

Table 1 lists the conditions specified for the file cache management. The cache unit was determined to be a presentation. This means that the access counting and the file transfer are made for each presentation. The access number was counted and saved per a day, and the data are saved for 2 days. As for the counting, the first access in a day is counted to be one. The additional access by the same person is counted to be one, if the additional access is made after 2 hours. The access by the same person is not counted, if it is made within 2 hours. To count the access number according to the above conditions, the data should contain presentation ID, university ID, account ID, the access date, and access time. The threshold access number N0 is specified to be 10.

Figure 3 shows a flow showing how the access is counted and saved. We designed the flow such that the access data are saved in 2 files for each presentation. One is a present file that contains all the data obtained today. If a client gets access to the presentation, the data are added to this present file. Another one is a previous file that contains all the data saved yesterday. The access data are checked and saved as follows. (1) A user is authenticated by the system after the account and password are input. At this moment, the system holds the account ID. After a presentation is selected by the user, (2) it is checked whether the 2 files exist or not. (3) If no files exist, they are created and the data are saved in the present file. This is the case that the presentation is selected for the first time. All the data shown in Table 1 except the access date are saved in one line in the file. The access date is registered in the DB, because the date is common to all the access. (4) If the 2 files exist, the access date of the present file is retrieved from the DB and it is checked when the access data were saved. (5) If the date is yesterday, all the data in the present file are transferred to and saved in the previous file. The access data are overwritten in the present file to delete all the old data and save the new data. This is the case when this presentation is gotten access for the first time today. Today's date is registered in the DB for the present file. (6) If the date of the present file is today, it is checked that the access by the same account has been gotten. (7) If it has been gotten access within 2 hours, the data are discarded and nothing is saved. (8) In other case, the data are added to the present file. (9) If no access has gotten by the same account, the access data are

added in the present file. Thus, we can save the access data, keeping all the data for 2 days.

3. System implementation and the performance

The access management program in the server AP and the cache program in the process client have been implemented to connect universities located in foreign countries over WWW in such a way to have the quick performance. The server AP was implemented by server side scripts in Microsoft Windows 2000[®] Server, and Microsoft SQL server[®] 7.0 was used as the database. The cache program in the process client was coded by using Microsoft visual C++[®]. The system was constructed under distributed server environments of one

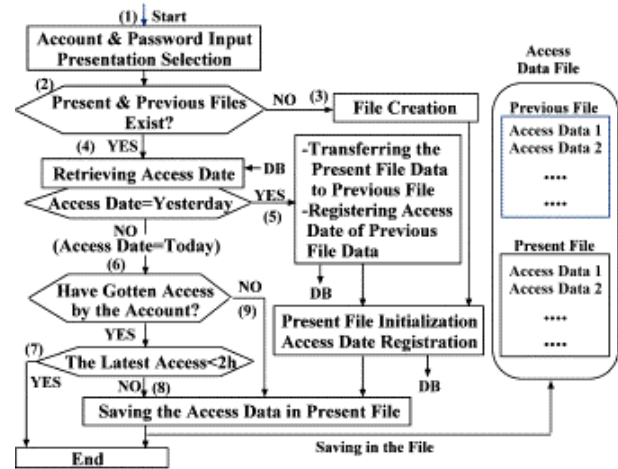


Fig. 3 Access counting flow

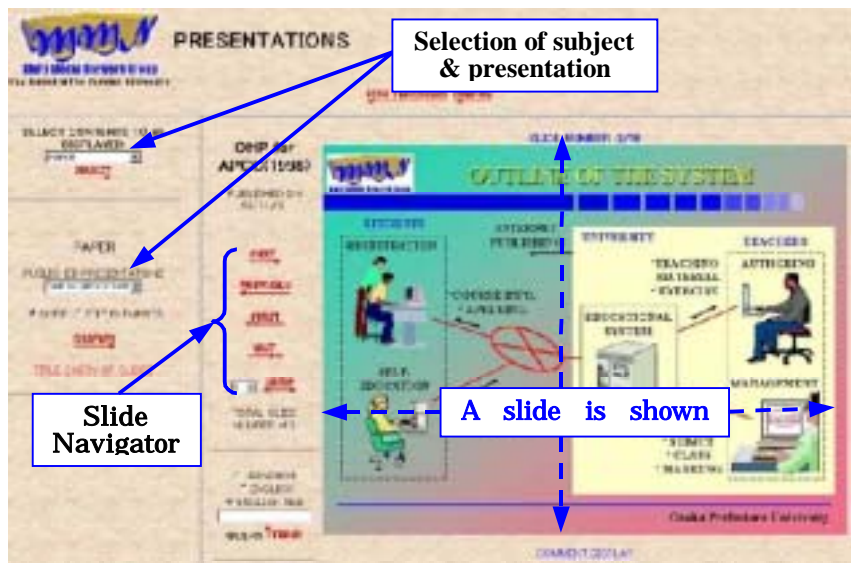


Fig. 4 Display of retrieved presentation

central and 2 local servers. Some client machines with an operating system of Windows-NT[®] or Windows 2000[®] are connected to the local servers, and the performance was examined.

3.1. Access number counting and file cache performance

Figure 4 shows an example of a presentation display on a web browser. To obtain this display, the following steps are taken. A student logs in the system by inputting its account and a password by a web browser in a client machine. If the input data agree with the registered values, it is verified in the server that the student is an authenticated user. Then, the system allows the student to start downloading for the self-education, and the student sends a downloading request. As shown in Fig.4, the first slide is shown inside lower-right area, after a subject and a presentation are specified by selecting the options on the left. The slide to be displayed in the presentation can be selected by the slide navigator by which we can specify the next or previous one, or jump to the first or the last one. Thus, any slide can be displayed. In the process of the retrieving a presentation, the access is checked. After the presentation is selected and before the first slide is displayed, this access is analyzed and the access number is determined in the background to be counted or not to be counted, according to the flow shown in Fig. 3.

Table 2 summarizes the experimental results of the access counting. The experiment was conducted for a system with one central and 2 local servers of Universities A and B. Clients who belong to university A got access to a presentation stored in Universities A and B, respectively. The numbers in Table 2 indicate the actually counted number by the system over the access number tried by the client. When different clients got access, the same number as the trial times was counted by the system. Four accesses by the same person within 2 hours was counted as just one, showing the right processing. In the same way the clients who belong to university B got access to a presentation in Universities A and B. The files stored in University A was gotten access by the clients in University B more frequently than the threshold value of 10. It was confirmed that the presentation files were transferred automatically to the local file server of University B.

After the automatic file transfer was completed, the presentation downloading experiment was conducted. When a client of University B sent a downloading request for the presentation, it was confirmed that the files were downloaded from the file server in University B. This result clarified that the cache status information was sent

and saved properly in the local server in University B, and the cache effect could be obtained.

3.2. Evaluation of quick performance

The quick performance was evaluated. A file transfer time measurement tool [5] was developed for this purpose. With this tool, we measured the file transfer time through 10Base LAN, and 100Base LAN. Figure 5 shows file size dependence of the transfer time. We also measured the transfer time, when a client and the sever is connected with ISDN basic rate. On the other hand, we calculated the transfer time by an equation

$$t(F, n) = \frac{F}{u} \left(t_u + \frac{n-1}{2} x \right),$$

which was derived theoretically [5]. Here t is the transfer time to transmit a file with a size of F . The symbols n , u , t_u , and x denote client number, unit packet size, transfer time of the unit packet u , and a unit waiting time to establish the link, respectively. The unit packet size u is equal to 1460 bytes in CSMA/CD. The parameter values of t_u and x should be determined by experiment. The determination process and the values are shown in

Table 2 Access and transfer experiment results

Univ. Where Clients Belong	Access Experiment		Univ. Where Files are Stored	
			Univ. A	Univ. B
Univ. A	Different Client		10/10	5/5
	Same Client	Within 2 Hours	1/4	1/4
		After 2 Hours	2/2	2/2
	Total		13/16	8/11
Univ. B	Different Client		10/10	10/10
	Same Client	Within 2 Hours	1/4	1/4
		After 2 Hours	2/2	2/2
	Total		13/16	13/16

The numbers indicate counted number/access trial number

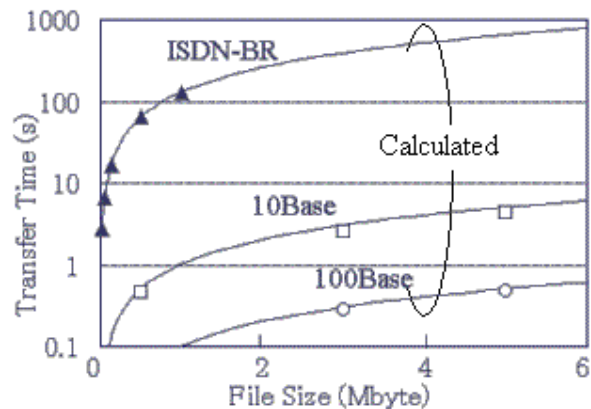


Fig.5 Evaluated quick performance

Ref.[5]. The calculated values of the transfer time are also shown in Fig.5. It is found that the measured and calculated values are in good agreement. The results show that it takes long time, if the file is transferred through ISDN basic rate (BR). This is the case for a student to download the file from the other local server via ISDN. On the other hand, if the file is stored in the local server of the university by the file cache, the student can download the file much faster via 10Base or 100Base LANs than via ISDN-BR. Thus, the quick performance is obtained.

4. Conclusion

We have designed and implemented a file cache function in an academic system over WWW so as to connect different universities located in different countries. In the file cache scheme, access counting and automatic file transfer procedures were adopted to obtain the quick performance. The procedures were implemented to operate a system with one central and 2 local servers. One client machine was allocated in each local server as a process client where the cache program was installed for the automatic file transfer processing, whereas the access counting program was implemented in the server AP in the local servers.

The system performance was examined. It was confirmed that the access counting and the file cache were processed properly. The quick performance was also evaluated, including the retrieving speed improvement for the students to download learning materials stored in other local server connected with common ISDN-BR (64kbps). By the cache scheme, the academic system can be used internationally with the quick performance over WWW.

Acknowledgment

The authors would like to thank the students of the multimedia network laboratory in Osaka Prefecture University for assisting the examination of the system performance.

This research is partly supported by the telecommunications advancement foundation.

References

- [1] Y. Nemoto et al., "Construction and utilization experiment of multimedia education system using satellite ETS-V and internet", IEICE Trans. Inf. & Sys., Vol.E80-D, pp.162-169, 1997. (in Japanese).
- [2] K. Okawa et al., "Virtual university", <http://www.sfc.wide.ad.jp/soi/library/>, 1997.
- [3] Y. Katsuyama, "Implementation of education system under distributed databases over WWW for international use", in *Proceedings of the 1999 Asia-Pacific Conference on Communications, Beijing, China, pp.872-875, 1999.*

- [4] O. Koyama and Y. Katsuyama, "Design and implementation of remote measurement and calculation functions in a stored-data education system over WWW", in *Proceedings of the 2000 Int. Conf. Computers and Advanced Technology in Education, Cancun, Mexico, 311-067, pp.166-169, 2000.*
- [5] S. Fujimoto, K. Tsujimura, S. Hayashi, J. Wang and Y. Katsuyama, "Limiting Performance Evaluation of Client/Server Local Area Networks, Based on Transfer Time under Traffic Congestion", accepted for publication in *Proceedings of the 2001 Asia-Pacific Conference on Communications, Tokyo, Japan, Sept., 2001.*