Sharing On-line Storage with Various Flat Forms of Information Devices

Seok Il Song*
Dept. of Computer Engineering
Chungju National University, Chungbuk, Korea

Yoon-Sik Kwak
Dept. of Computer Engineering
Chungju National University, Chungbuk, Korea

ABSTRACT

In this paper, we propose a new storage system architecture called as U-Storage (ubiquitous storage system). U-Storage allows a user to access an on-line storage with any type of information devices that are able to connect to internet. The on-line storage is virtualized to the user's information devices as a local hard disk or a memory card by our U-Storage. With devices supporting U-Storage, users can read and write their data anytime and anywhere without downloading and uploading operation.

1. INTRODUCTION

With the rapid advances in wireless communications and ubiquitous computing technologies, a user owns several information devices besides personal computers (PC) such as laptops, ultra mobile PCs, cell phones, smart phones and so on. Users that use multi devices suffer from sharing data among their devices. For example, in order to store the photographs of a cell phone into a laptop or a PC, we should use synchronizing tools such as serial cable and sync S/W. Also, to transfer data of a PC to a laptop or another PC, we should use data transfer utilities (FTP Client/Server, internet disk and so on) or floppy disk, CD, USB memory and so on. Even though data can be shared by above annoying methods, we suffer from managing data consistently across various devices. We need to manage carefully replicated data on several devices and storage media by ourselves so as to be consistent.

Recently, internet disk have been developed and used widely [1, 7]. The internet disk allows users to access their data on-line with general purpose web browsers or native browsers provided by internet disk companies. Most of existing internet disk systems are based on upload and download services. Users upload their data to internet disk to store and download them from internet disk to read by using web browsers or native browsers. However, managing data consistency is still users' responsibility. Also, uploading and downloading data must be performed through special softwares.

In this paper, we propose a new on-line storage system called as ubiquitous storage (U-Storage). The U-Storage is a client/server architecture. The U-Storage server partitions its storage logically according to users' requests, and provides the partitioned storage to clients through U-Storage protocol and client software. The U-Storage client is implemented as a virtual device driver. The virtual device driver make the storage provided by the server become a local storage for users' information devices. Since the server's storage is recognized as a local storage such as hard disk or flash memory of information devices, users can access and share the storage without any special browser. Users do not need to upload or download their data so they do not need to maintain data consistency.

The U-Storage provides another benefit to mobile devices. Usually, mobile devices have storage constraints; the size of memory cards that are widely used for mobile devices is 4 Giga Bytes. However, U-Storage expands the size of mobile devices unlimitedly. The problem of U-Storage for mobile devices is the low communication speed. However, recently high speed wireless communication technologies such as HSDPA (High Speed Downlink Packet Access) and WiBro (Wireless Broadband Internet) have been emerged so U-Storage can be applicable for mobile devices.

This paper is organized as follows. In Section 2, we describe the overall architecture of U-Storage and design issues. Section 3 gives the description of U-Storage protocol. Then, Section 4 presents the implementation of U-Storage and finally we conclude in Section 5.

2. ARCHITECTURE OF U-STORAGE

Figure 1 shows the architecture of U-Storage. As mentioned above, the proposed U-Storage is client/server

* *Corresponding author. E-mail :* sisong@chungju.ac.kr
Manuscript received Sep. 11, 2007 ; accepted Sep. 8, 2007

architecture. The U-Storage server is composed of volume manager, session manager, file system converter and PDU encoder/decoder. The volume manager manages a logical volume created based on a disk array. It partitions the logical volume into multiple volumes, and assigns partitioned volumes to users. Also, it expands and shrinks an assigned volume according to users' requests.

The session manager establishes connections with clients and monitors the status of each connection. The file system converter transform the block IO(Input and Output)s of FAT file system block I/O to those of EXT or UFS file system and vice versa. There may be heterogeneous platforms of information devices, i.e., information devices may have different operating systems and support different file systems. Assume that there are two platforms of clients. One is WIN CE PDA and the other is Linux PC. The WIN CE PDA supports FAT16 file system and the Linux PC supports EXT2 file system. Since the U-Storage provides the two clients a raw virtual hard disk and memory card, the WIN CE PDA needs to format the virtual memory card as FAT16 file system and the Linux PC needs to format the virtual hard disk as EXT2 file system. However, different file systems cannot be put on a volume. In this case, we formatted the volume as EXT2 file system. The Linux PC can recognize the volume as its local file system. However, the EXT2 volume cannot be provided to WIN CE PDA as a local memory card. The file system converter solves the problem by mapping EXT2 block IOs to FAT16 block IOs and vice versa. The read/write PDUs of WIN CE PDA is converted to PDUs for EXT2 file system by the file system converter. Also, the U-Storage server's response PDUs are converted to FAT16 PDUs.

The U-Storage clients are virtual device drivers. Different virtual device drivers should exist for various platforms such as WIN XP devices, WIN CE devices, cell phones and Linux or Unix devices as shown in Figure 1. Usually, IO requests of file systems are delivered to hard disks or memory cards by device drivers. However, the U-Storage clients (virtual device driver) deliver the IO requests to the U-Storage server.

In the next section, we will describe the protocol between the server and the clients.

3. U-STORAGE PROTOCOL

In this section, we describe the communication protocol between the U-Storage server and clients. There are eight PDUs such as LIREQ (login request), LIRES (login response), LOREQ (logout request), LORES (logout response), WREQ (write request), WRES (write response), RREQ (read request) and RRES (read response). A PDU consists of *Length*, *Type* and *Data*. *Length* denotes the length of *Data* of a PDU, and *Type* is a code number that indicates the type of PDUs as shown in Table 1.

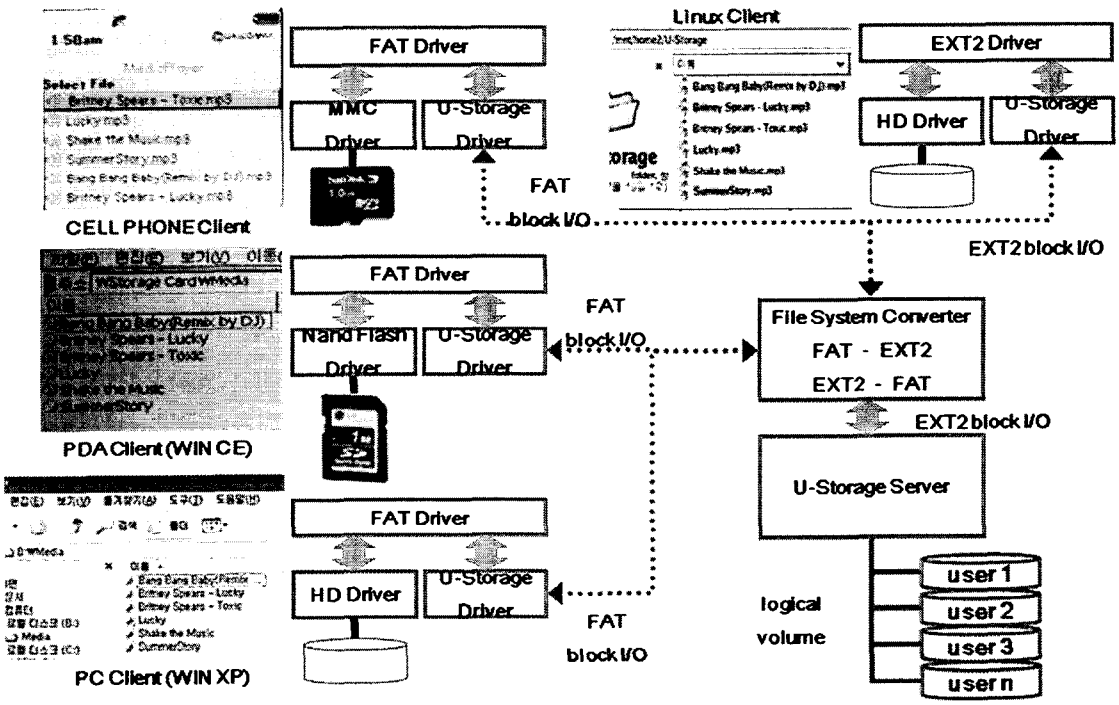


Figure 1. Architecture of U-Storage

Table 1. PDU Types

PDU Type No.	PDUs
001 / 002	LIREQ / LIRES
003 / 004	LOREQ / LORES
005 / 006	RREQ / RRES
007 / 008	WREQ /WRES

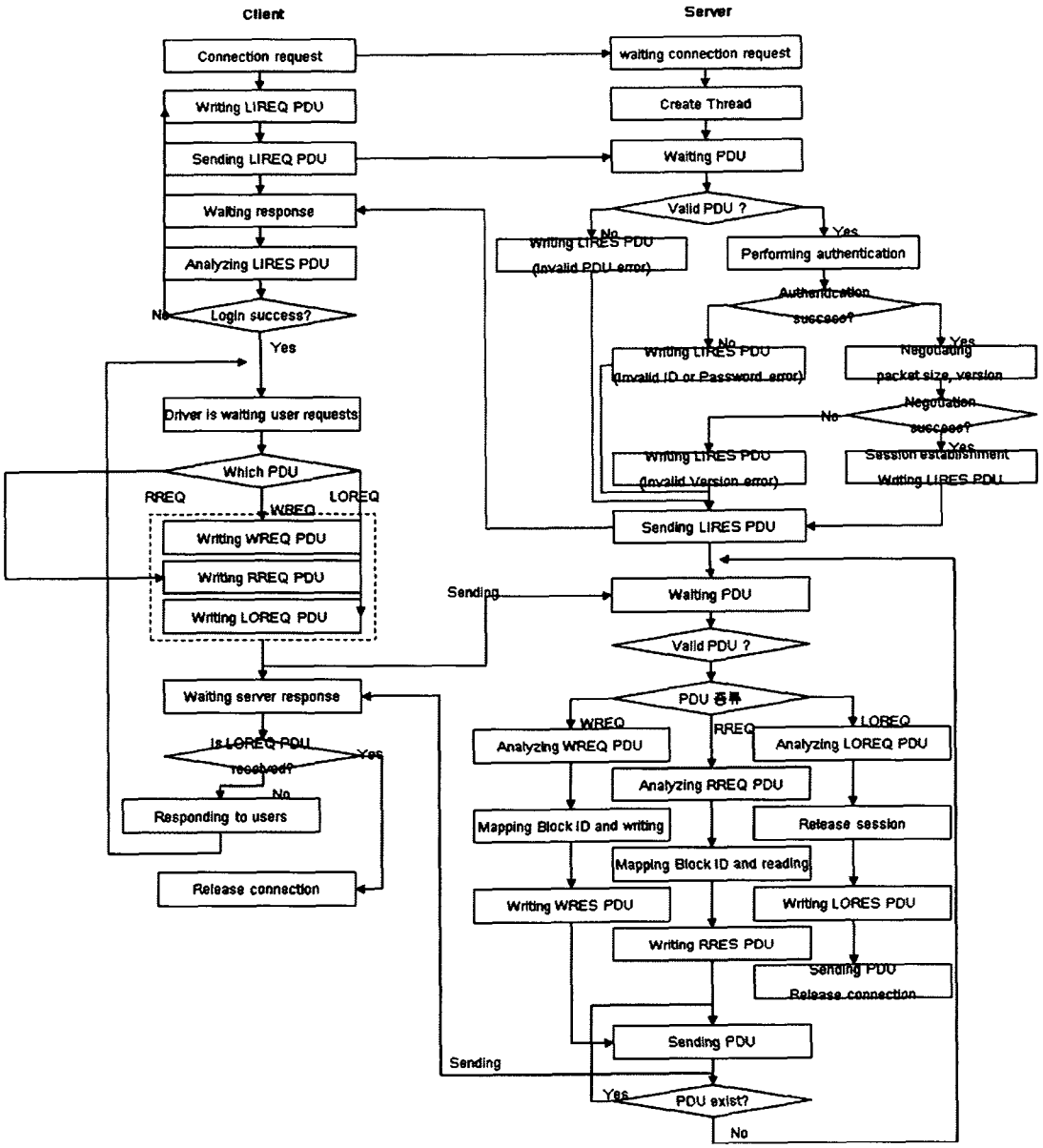


Figure 2. U-Storage Protocol

Data denotes the contents to be delivered. The contents of *Data* vary according to the *Type* of PDUs. For example, LIREQ contains user ID and password, WREQ contains output data, RRES contains input data, LIRES and WRES contain the error codes and so on.

Figure 2 shows the work flow of U-Storage protocol. In the beginning, server assigns a volume for a user and the user register with user ID and password. Then the user set user ID, password and the address of U-Storage server. If a client sends the connection request to U-Storage server after configuring process, the server that is waiting connection requests starts the connection establishment process.

After that the client sends a LOREQ PDU. The LOREQ contains user ID and password. The server that is waiting PDUs receives the LOREQ PDU, and analyzes the PDU to find the *Type* and the contents of *Data* of the PDU. Then the server performs authentication process with the user ID and password stored in *Data* of the LOREQ. If the authentication succeeds, a session between the server and client is established.

When the session is established successfully, the client (virtual device driver) is waiting IO requests from file system. All of file system IOs are transformed to RREQ and WREQ PDUs by PDU encoder/decoder and sent to the server. The server receives the RREQ and WREQ PDUs and process IOs on the volume assigned to the user. If needed, the server performs file system converting.

4. DESIGN AND IMPLEMENTATION OF U-STORAGE

The block diagram of U-Storage Server is shown in Figure 3. The U-Storage server is implemented based on Linux machine (kernel ver. 2.6). As we mentioned in Section 2, the volume manager create a logical volume and partition the volume into a number of sub-volumes according to users' requests. It is implemented based on LVM (logical volume manager) provided by kernel. Logical volume management provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions. This gives the system administrator much more flexibility in allocating storage to applications and users [8].

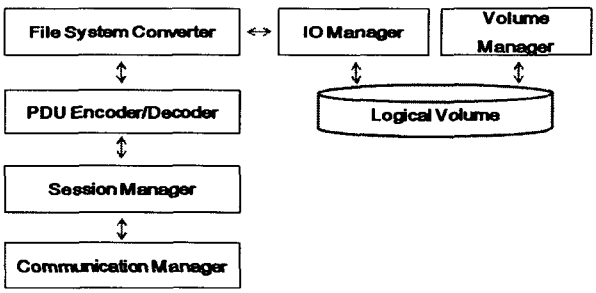


Figure 3. Block Diagram of U-Storage Server

The communication manager receives a PDU from a client. Then, it is passed to the session manager. The session manager finds the session ID of the PDU. PDU encoder/decoder analyzes the PDU. If the PDU is RREQ or WREQ, the file system converter transforms the IO of PDU to EXT2 I/O. Finally, IO manager read or write the transformed EXT2 I/O on the sub-volume corresponding to the session ID.

Figure 4 shows the block diagram of U-Storage clients. U-Storage clients are implemented as device drivers at WIN XP, WIN CE, cell phone and Linux. In this paper, we cannot implement cell phone client fully because the OS of commercial cell phone is not opened. Therefore, the cell phone client is implemented with educational development kit. The communication manager monitors the connection with the server and sends/receives PDUs.

The session manager establish sessions with the server. When the session is disconnected by accidents, it stores the states of the session and keeps up the previous connection after the connection is reestablished. The cache manager stores meta-blocks of the server's volume so the client does not need to communicate with the server to navigate directory.

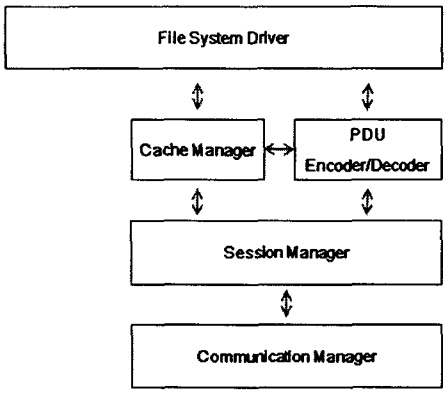


Figure 4. Block Diagram of U-Storage Client

Through virtual device drivers, remote volume can be a local hard disk or memory card for clients. Consequently, users access the remote volume naturally in the same fashion to the local storage. Figure 5 and 6 show the U-Storage on WIN CE PDA. In the PDA, U-Storage is shown as a folder like other memory cards. In WIN XP PCs, U-Storage is shown as a hard disk drive.

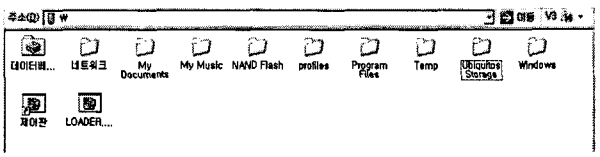


Figure 5. U-Storage on WIN CE PDA

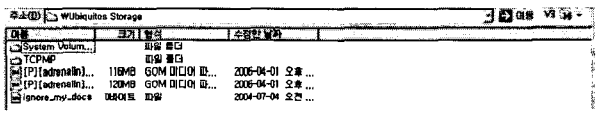


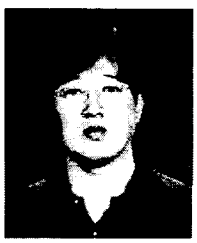
Figure 6. U-Storage on WIN CE PDA

5. CONCLUSION

In this paper, we propose and implement U-Storage which is a new on-line storage system. The U-Storage allows users to access remote storage as their local hard disks or memory cards. Since upload/download operations are not needed, they are not required to manage replicated files. We implement the U-Storage on various platforms such as Linux, WIN CE, WIN XP and cell phone. Therefore, users can share their data with any platform of device in the same manner to access local storage. Also, the U-Storage expands the size of storage of mobile devices such as WIN CE devices and cell phones.

REFERENCES

[1] http://www.webhard.co.kr/pda
[2] Tom Clark, "Designing Storage Area Networks : a Practical Reference for Implementing Fibre Channel and IP SANs 2nd Edition", *ADDISON-WESLEY*, 2003
[3] Thomas Clark, "IP SANs An Introduction to iSCSI,FCP and FCIP Protocols for Storage Area Networks", *ADDISON-WESLEY*, 2000
[4] Alan Johnston, "SIP: Understanding the Session Initiation Protocol 2nd Edition ", *Artech House Publishers*, 2003
[5] Douglas Boling, "Programming Microsoft Windows CE. NET - Third Edition", *Microsoft*, 2004
[6] James Y. Wilson and Aspi Havewala, "Building Powerful Platforms with Windows CE", *ADDISON-WESLEY*, 2001.
[7] http://www.internetdisk.com/
[8] http://tldp.org/HOWTO/LVM-HOWTO/
[9] Tae-Ho Noh, Cho Kyung-Won, Seok Il Song and Yoon-Sik Kwak, "Implementation of Internet Disk for Mobile Device - Client/Server Drivers -", *KIMICS*, vol.9, no.2, pp.253-261, 2005



Seok Il Song
He received the B.S., M.S. and Ph.D degrees in Computer and Communication Engineering in 1998, 2000 and 2003 from Chungbuk National University, Cheongju, South Korea. He is now a professor in the department of Computer Engineering, Chungju National University, Chungju, South Korea. His main research interests are the database system, index structure, distributed computing and storage management system.

Yoon-sik Kwak
He received his B.S. degree in Electrical Engineering from the University of Cheongju in 1984, his M.S.E.E. degree from the University of Kyunghee in 1986 and his Ph.D. degree from the University of Kyunghee in 1994. He worked at Chungju National University in the Department of Computer Engineering and rose to the level of Full Professor. His research interests are in the areas of signal processing and image processing, Internet communication, microcomputer system , applications of these methods to mobile system.