Control Theory and Informatics
ISSN 2224-5774 (print) ISSN 2225-0492 (online)
Vol 2, No.1, 2012

Distributed Virtual Disk Storage System

Muhammad Sharif*, Nasir Mehmod Butt, Mudassar Raza, Muhammad Arshad
Department of Computer Sciences
COMSATS Institute of Information Technology, Wah Cantt, Punjab-Pakistan
E-mail: nasir_meh2001@yahoo.com, mudassarkazmi@yahoo.com, marshadmcs@yahoo.com
* E-mail of the corresponding author: muhammadsharifmalik@yahoo.com

Abstract
Distributed Virtual Disk Storage System (DVDSS) is a reliable and fully decentralized storage system which is based on the concept of Distributed Storage and Virtual Disk. It is Client / Server architecture and utilizes all free space of desktop machines of a LAN for storage purposes. DVDSS converts all storage of computers on the LAN into a large disk, create a virtual disk and provide this disk to all users of the system. Users can store their data files and folders on this disk and they can retrieve their data whenever they need. DVDSS provides a transparent and reliable access to data. It also provides data consistency and data security. The recovery of data when a node fails is also provided by DVDSS. The objective of system is to combine the all storage space of a LAN and utilize it for storage of user's data in transparent and reliable manner.

Keywords: Distributed Storage, Virtual Disk, Space Utilization, data recovery, data consistency.

1. Introduction

Information plays a vital role in today's world. The scientific and logistic data's importance is increasing sharply. The storage capacity can be enhanced up to hundred of Giga Bytes as the storage devices are very cheap. But the overall cost of backup is very high. Nowadays most of industries and universities have several servers providing storage services. When storage demand increases they need to purchase new storage devices. Most of the storage is performed on the server's storage devices. In a standard lab more than 50 PC's were selected, each having more than 100 Giga Bytes of storage capacity. Most of this storage capacity remains unutilized. To utilize this storage space, a Distributed Virtual Disk Storage System is proposed to utilize the idle storage. The main goal of this system is to combine all idle storage of desktops on a LAN and make an efficient and reliable storage system.

The idea is to collect the hard disk space of all client PC's over network and utilize that space for distributed storage. The idea of virtual disk is also applied. In next section some of the similar existing systems and their drawbacks are discussed. After this, the proposed system and its overall architecture is described. At the end the different modules of system, features provided by system, and implementation are discussed.

2. Related Work

Distributed storage is not a new idea. Many systems are built for distributed storage. The oldest system is RAID (Redundant Array of Independent Disks) P. M. Chen *et. al.* (1994), Kai Hwang *et. al.* (2001) which is used for computers with many hard disks inside. This system distributes data on different disks of the same computer. Petal Edward K. Lee *et. al.* (1996) uses the idea of virtual disks and distributes data between servers not client PC's. Virtual Mass Storage System Nayyer Masood *et. al.* (2005), Nayyer Masood *et. al.* (2004) also uses the idea of creating virtual disk from spare resources shared by donor. It is a grid based storage system. It does not truly utilize all the space available by Desktops on a LAN. It defines some of the machines as donors which share their resources. Not all Client Machine's available space is used in virtual disk creation. Another drawback is in Resource Allocation for storage. It starts allocation from minimum available resource which may divide every file with very less size which degrades the performance in both storage and retrieving. Only single file storage facility is provided. It uses the certification mechanism for security but does not provide any recovery method. If one of the donors, who shares space and stores files, crashes there is no way to recover data. For one LAN there is one Donor Manager who provides information of donor and if only DM crashes the information of all those assigned

donors by that DM will be lost.

3. Proposed System

The idea is to collect the information of hard disk from each PC on network and make a large virtual drive which is available to each and every node on network so every node has a storage capacity equal to aggregate amount of storage capacity of all nodes combined. The user only drags files from local disk to virtual disk for storing data for backup and click on files in virtual drive to access files from backup storage. It will be the responsibility of software to manage back up storage and restore information and also the recovery from failure. The overall architecture of this idea is shown in the figure 1.

The figure shows the hard disks of all PC's over network are combined to make a virtual disk which is accessible to each and every node on the network. In this solution the idea of distributed storage and virtual disk is combined to build a new solution for backup storage. For distributed storage instead of the disks of server, the hard disks of all PC's on the network are used. The users only need to use virtual drive GUI for storing and accessing data. The complete architecture and design is discussed in next section.

3.1. Architecture and Design

The system is based on client server architecture. There are two subsystems in it. The layered architecture of the system is shown in figure 2.

The components of the system are discussed below:

3.1.1. Server Request Handler (SRH)

This module is the backbone of this system. The module is used for managing and monitoring all processes and operations of the system. SRH controls the connection establishment between clients and server. Server (SRH) manages all of the user requests, storage and retrieval mechanisms. The main operation of SRH is to collect the hard disk information from all computers. For this purpose it listens to the client and when any client agent sends hard disk information it saves this information; creates a large disk from all available storage and sends the drive information to all connected users. In this way it provides the large storage space to all users.

All users' requests are also monitored by SRH. It obtains request from user, passes it and hands it over to specific request handler for processing.

On request of data storage; it obtains data from user, divides it into equal chunks and stores them on computers connected to it. The selection of computers to store chunks is decided by Best Fit Algorithm applied on size of chunk and availability of computer. Here it also generates error correcting codes for data and stores it on server for recovery purposes. It also generates unique file name to remove file name clashes that are stored by different users with same name.

On data retrieval request SRH searches the location of data; retrieves data from that computer and sends it to its user then deletes it from drive. It also provides recovery of data if one of the chunks of file is destroyed. It rebuilds data from exiting data and recovery codes.

On data deletion request, it simply searches location of data (i.e. search computer on which data is stored) and deletes data from that location.

While overwriting data, it also assures that all data chunks and error correcting code for that data are updated. Therefore, data must be consistent.

For recovery of data, the system uses the error correction codes. While storing the data on different computers, before sending data to specific computes, error correction codes are generated for each file of user. The codes are stored at server side. So if a part of file cannot be retrieved then the file can be recovered from exiting portion and the code is generated for that file. This saves system from replication of data and provides reliability and security.

3.1.2. Server Administrator (VDAdmin)

The module is developed to provide administrative facilities in system. This is a GUI based application which provides an interface to administrator for different operations. Using this interface the administrator

can check information about server i.e. status (Running or stopped), total number of files stored on system by users (uploads), size of total uploads, total number of files retrieved by users (downloads), total size of down loads and total uploads and their size. Administrator can also check information of connected computers. The information about users; like user name, its status (logged on or logged off), ip of computer on which users logged on, login time, its total uploads, total downloads, maximum file size, total space usage and its stored files etc. can also be checked by administrator. Administrator can also check information of stored files. Using Administrator interface the administrator can delete any type of data of any user at any time. If the files are in use (i.e. the owner of files is logged on) then the files cannot be deleted. This provides the administrator control to clean unwanted data from system to secure system from damages. The administrator can also delete and add users in system. For performing its operation it coordinates with VDClient.

3.1.3. Client Agent (VDClient)

This component is the basic part of client module. It lies on all computers of LAN. The main purpose of this component is to provide the local disk information to the SRH when computer is available on LAN for storage. This service is installed on computer and when it switches on in LAN, it gets information and status of local disk and sends it to SRH with its IP Address. It then gets the information of Virtual Disk from SRH and stores it. It also performs many other tasks assigned by server. As the disks of these computers are used for data storage the SRH sends data to VDClient, it receives data from SRH and stores it on local disk where space is available. It also sets access control of data. When the request of data retrieved is sent from SRH it searches data from local disk and sends it back to SRH. The data deletion operation is also performed by VDClient. For each request there is specific handler. To perform all these operations it communicates with SRH. It also assigns some tasks from VDAdmin. It provides the Virtual disk information to VDInterface for user of system.

3.1.4. Client Interface (VDInterface)

This component is also part of client module and plays a very vital role in the system. This is a GUI based application provides the interface of virtual disk to user of the system. It is the component where the users truly use the storage of system. It is windows like interface and provides its user a visualization of actual disk interface. This component provides an easy to use interface for different user files and folder operations. It also shows the total space available to the users. When a user wants to perform any operation on virtual disk it runs this application, the virtual drive is opened. If user wants to store a file it selects file from local disk, file is shown in virtual disk as it were actually stored on it but it is sent to SRH for storage transparent to user. All data stored by user on virtual disk is visible in it. If user wants to retrieve or delete data from virtual disk or wants to open a file, it selects that file or folder, the data is retrieved or deleted from virtual drive. All requests of these operations are sent to SRH by VDClient transparent to the user. To perform these tasks SRH, VDClient and VDInterface cooperate with each other.

3.1.5. VDWatcher

This component is the part of Client Interface. It is a background process run when VDInterface is run by user and stops when VDInterface is stopped. It is used to handle the file Save AS operation. When it runs it creates a drive in My Computer folder with same name as that of virtual disk. When user wants to save a file directly on the virtual disk it selects the newly created drive and saves file in it. This file is sent to SRH for storage and is visible in Virtual disk Interface. All these operations are performed by VDWatcher transparent to the user. The user only knows that he/she saves file in virtual disk.

3.1.6. Recovery and Reliability

The most famous mechanism of data recovery and high availability is the data mirroring. But as the importance to storage space is given therefore it is not good for our system. The recovery of data using error correcting codes is provided. For each file stored on virtual drive the file is divided into chunks and an error correcting code is generated for each file. The code is generated by taking a XOR of all the chunks of file. This code is stored on server. The storage mechanism is shown in fig. 3.

Now if one of the chunks is not available or destroyed the data can be recovered from remaining chunks and the error correcting codes. Server retrieves available chunks and reads error code and rebuilds the destroyed chunk. The retrieval and recovery mechanism is shown in figure 4.

This mechanism provides high reliability with minimum wastage of storage.

3.2. Features of DVDSS

3.2.1. Desktop space utilization and virtual disk

The system will utilize all available space of all desktop on LAN and will use that space as virtual disk. The space is taped into virtual disk automatically and does not share explicitly.

3.2.2. Data Security

As files are stored on disk of client computer used by different users which may cause security problem i.e. they can delete files stored on that system by DVDSS; system ensures that the data of each user is secure and is not damaged accidentally by local users of the computer. To protect against such problems system implements and sets access permissions on data so that local users cannot delete files.

Another security issue is user authentication. Only those users can use system who are registered in system. And each user can view its own data and operate on its own data.

3.2.3. Data Consistency

The files are divided into equal sized chunks and are stored at different locations. The system ensures that if a file is updated then its all chunks must also be updated.

3.2.4. File Type and Unique File Names

The DVDSS uses its own file convention to store files. DVDSS has its own file types. It uses a unique file type for all stored files. Another type is used for error codes known as ecode. It also creates its own unique file name for each file. So no two same name files of two different users have same DVDSS file name. This convention protects from overwriting of different user files with same name. Names are created at runtime by software.

3.2.5. Administration Facilities

DVDSS provides administrative features to system administrator. It provides an interface to monitor different operations of the system.

3.2.6. File Operations

The system provides different file operations e.g. File Storage (Users can store their files on virtual drive), File Overwrite (if user stores a file which already exists then on user's choice it will be overwritten), File Retrieving (Users can retrieve their files from virtual drive to local drive), File Deletion (Users can delete their files from virtual drive), Save As (Users can save their files directly to the virtual drive) and File Open (Users can open their files and can make changes to them and save them). All working of these operations is handled by system automatically and is transparent to users.

3.2.7. Folder Operations

The system also provides some folder operations like Folder Storage (to store a complete folder on virtual disk), Folder View (to show the contents of folder stored on virtual drive), Folder Download (to download the complete folder from virtual drive to local disk) and Folder Deletion (to delete the folder stored on virtual drive).

3.2.8. Virtual Drive Interface

The system provides an easy to use graphical user interface of virtual drive. All of the above operation can be done using this interface.

4. Results And Conclusion

The ideas of distribution and virtual disk and the component used in this solution are not new but such a system which uses the desktop on LAN for Distribution storage integrated with virtual disk is absent that can be a good solution for LAN storage. Below is analysis of the proposed system. The first table (Table 1) shows the recovery percentage of data against different number of codes generated for data.

If data is divided into 'm' no. of blocks and 'n' codes generated for those blocks then data recovery %age can be calculated as:

Data Recovery % = (n/m) *100

The extra space required to store codes can be calculated as:

Space Wastage = (n/m)*DataSize

Where 'n' is no. of codes and 'm' is no. of blocks and DataSize is data sent by user for storage.

The table (Table 2) below shows the response time to a user while accessing data when no recovery involves. The time is shown in milliseconds.

The last table (Table 3) shows response time when recovery of one chunk i.e. (1/m*100)% is involved, where 'm' is no. of blocks. The time is again in milliseconds.

References

[1] Kai Hwang, Hai Jin, Roy S.C. Ho. Orthogonal Striping and Mirroring in Distributed RAID for I/O-Centric Cluster Computing. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 2001.

[2] J.H.Hartman and J. K. Ousterhout. The zebra striped network file system. Operating Systems Review – 14th ACM Symposiums on Operating System Principles, December 1993.

[3] P. M. Chen, E. K. Lee, G.A. Gibson, R. H. Katz and D. A. Patterson. RAID: High-performance, reliable secondary storage. ACM Computing Surveys, 1994.

[4] Edward K. Lee and Chandrohan A. Thekkah. Petal: distributed virtual discs. SIGPLAN Notices, October 1996.

[5] Dr Nayyer Masood, Malik Muhammad Junaid, Hafiz Muhammad Farooq Khan. An Architectural Survey of Online Data Stores. OIC Member States 2005 Islamabad.

Dr. Nayyer Masood, Malik M Junaid, Hafiz M Farooq Khan. Virtual Mass Storage System. Frontiers of Information Technology 2004, Islamabad.

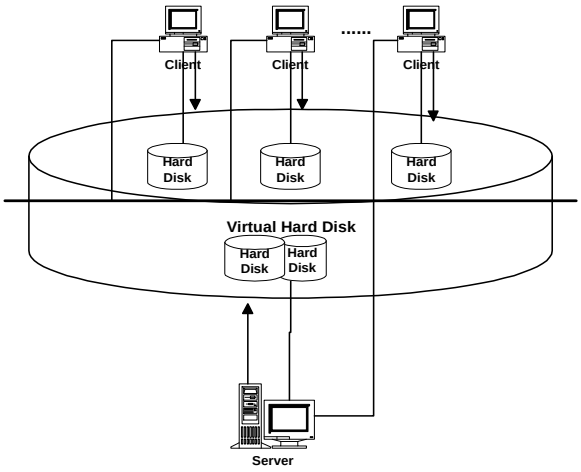

Fig.1. basic idea of DVDSS

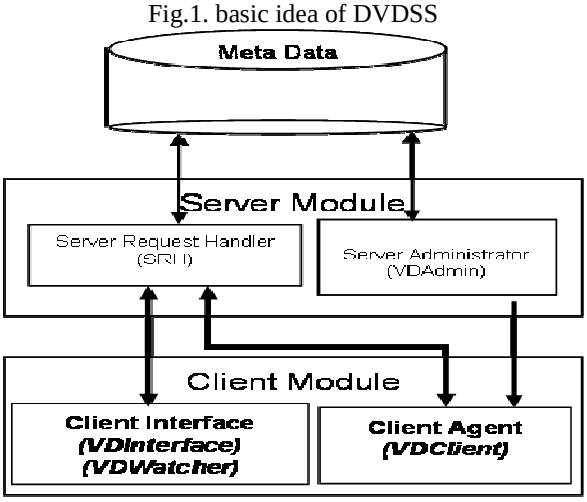

Fig.2. Architecture of DVDSS

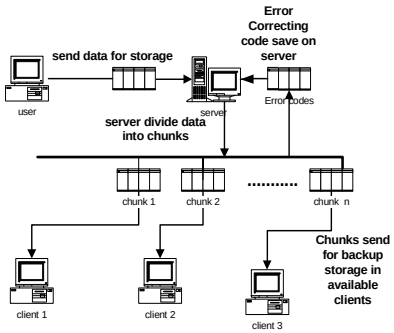

Fig. 3. Storage Mechanism of DVDSS

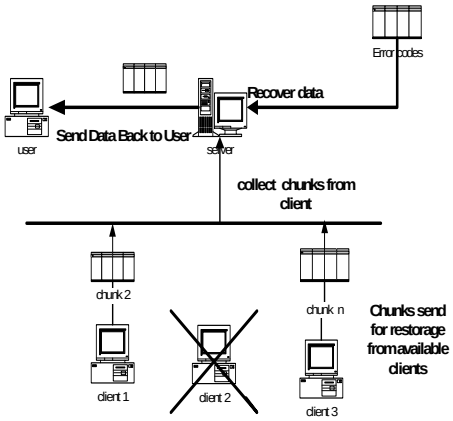

Fig. 4. Retrieval and Recovery Mechanism

Table 1. Recovery % of Data

No. of parts	No. of codes				
	1	2	3	4	5
2	50%				
4	25%	50%			
6	16.6%	33.3%	50%		
8	12.5%	25%	37.5%	50%	
10	10%	20%	30%	40%	50%

Table 2. Response time for different data size divided into different chunks with no recovery

Data Size	No. of computers				
	2	4	6	8	10
5KB	281	292	305	318	328
10KB	312	356	362	379	388
50KB	360	371	382	388	399
100KB	392	412	421	439	451
1MB	562	593	601	613	628

Table 3. Response time for different data size divided into different chunks with recovery

Data Size	No. of computers				
	2	4	6	8	10
5KB	290	298	308	323	331
10KB	318	335	342	354	386
50KB	351	362	370	382	401
100KB	360	375	389	397	425
1MB	625	650	671	686	699