Modified Unified Virtual Storage Technique for Improvement in Network Storage Implemented on Local Area Network

K.N.Honwadkar
D.Y.Patil College of Engineering,
Sector 29, PCNTDA, Akurdi, Pune 411 044,
Maharashtra, India
Tel: +91-20-27653054 E-mail: knhonwadkar@yahoo.co.in

Dr.T.R.Sontakke
Principal, Siddhant College of Engineering,
Sudumbare, Pune – 412 109,
Maharashtra, India
Tel: 86-10-8888-7777 E-mail: trsontakke@yahoo.com

Abstract

This paper describes a way to efficiently utilize free disk space on Desktop machines connected over a network. In many Networks today, the local disks of a client node are only used sporadically. This is an attempt to manage the data storages in a network efficiently and to provide the software support for sharing of disk space on Desktop machines in LAN. In the current situation, storage expansion on conventional servers has constraints like, maximum expansion limitation, costly affair and in case of hardware replacement, up gradation; the manual relocation of Data becomes messy. UVS (Unified Virtual Storage) is an attempt to efficiently utilize freely available disk space on Desktop machines connected over a network. Its purpose is to reduce load of data traffic on network server, to efficiently utilize space on client nodes thereby avoiding wastage of space. It also eliminates Hardware restriction for Storage Expansion and provides Location transparency of data store. The main advantage of UVS is that it can be seamlessly integrated into the existing infrastructure (Local Area Network system). Virtual Storage is virtually infinite supporting scalable architecture. The performance of prototype implemented on a UVS Server connected by network and performance is better than the centralized system and that the overhead of the framework is moderate even during high load.

Keywords: Location Transparency, Centralized system, Distributed Storage, Scalable Architecture.

1. Introduction

This paper proposes Unified Virtual Storage system [11], which is suitable for backup data on network storage. Here, disk-virtualization concept is used by implementing client - server module, locking & file manager module. The performance of UVS is found superior as compared to LanStore and Windows share drive. 'UVS' is an attempt to efficiently utilize free disk space [2] on Desktop machines connected over a network. This is a client - server based model. The main purpose is to reduce load of data on centralized servers & efficiently utilize space on client machines. This will support Distributed Level Application in which a client will loan some space (say 1 GB) to server to make that space openly available to all the clients (especially to the client which has no space). There will be Locking mechanism by which the space loaned to Server will be locked for that client. On server side, there will be a file system or a mapping table to find who are the loaning clients & where the file is stored. Non loaning clients can even use this virtual storage transparently. This involves developing of a client -server based utility with intelligence to scan the network nodes (desktop machine) for unconsumed Secondary Storage (HDD) spaces and allocate/manage the same 'on the fly' between different users for resource sharing. The client can use the Unified Virtual Drive as a single point access for distributed storage across different nodes thereby eliminating an individual addressing of the nodes.

The paper is organized as follows.

The proposed architecture and its algorithms are described in section2 and 3, respectively. The experimental results derived after implementations are tabulated in section 4. Modification for the improvement of UVS implementation is proposed and discussed in section5. Related work is described in brief in section 6 and Conclusions in section 7.

2. Unified Virtual Storage Architecture

This System can prove itself very useful & convenient replacement when there is a need to upgrade the existing Server or to install high capacity HDD on server, Which involves manual relocation of the existing data, which is a tedious job for system administrator in industrial scale and also it does not utilize the data storage, which is available on a remote server on the same network. Finally this will eliminate the restriction to expand storage on only one specific server. Above all, it will act as one virtually centralized repository accessible to everybody. Imagine this being used as a distributed document repository. This would be a new revolutionary concept of distributed storage with a virtually centralized access. Client machine does not have to worry about the exact physical location of data. The main Objective of Unified Virtual Storage (A Distributed Solution) is to deliver a robust and economic Distributed Mass Storage System, which can be seamlessly integrated into the existing infrastructure at no additional cost. UVS is to be used in Local area network (LAN) by the network administrators so as to utilize the existing storage space available on server machines efficiently, which should provide a virtually centralized single point of access for the users. There are some primary reasons for UVS implementation:

- Ease of Storage Expansion – Scalable Architecture
- Efficient Use of free Space on desktop machines – avoid Wastage
- Eliminate Hardware restriction
- Provides Location Transparency
- Increased Performance
- Lower Costs

2.1 System configuration

UVS is a Client-Server based utility. Network Administrator handles the creation, utilization and maintenance of Virtual Disk. Users or clients of the network can utilize this virtual storage space. However, the donors of the storage space must do an agreement that they should be always ON before contributing their local disk space to UVS. The system best runs on the client-side workstation with at least 64 MB of RAM and at least Pentium class processor with 350MHz or higher frequency. Interface to the LAN through an Ethernet card preferably with 100 Mbps data rate.

The basic UVS system is depicted in Fig 1. The client nodes which can spare some of their storage space, for use on the network storage, declare the size of such space to the server. This is known as ‘Loaning’ of space. The information of such loaned space is registered in the server database under ‘Loaned space’.

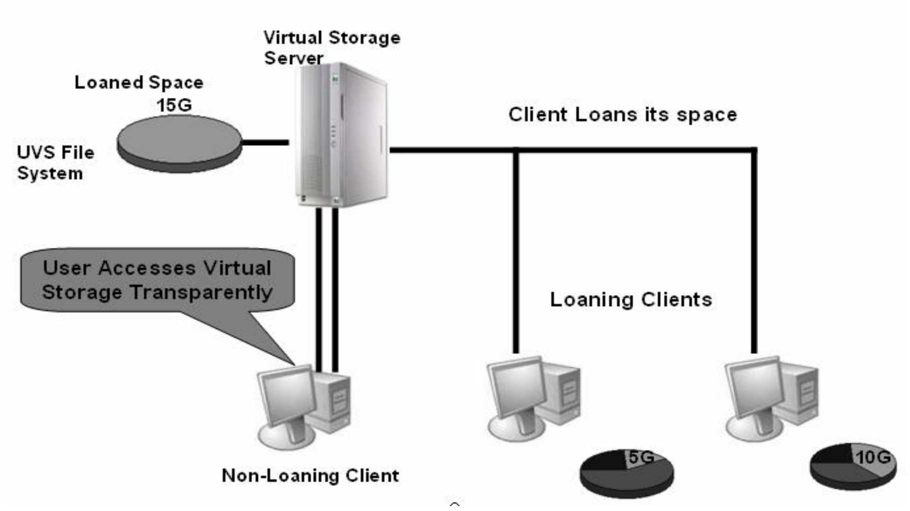

Fig 1 UVS System overview

Any other client, irrespective of whether it has loaned space earlier, if demands some space on the network storage, sends a ‘Demand’ request to the server; along with the size of the space required. The Virtual Storage

Server looks for suitable area in the loaned space and allocates the demanded space to the authenticated user. This information is also logged in the server database under 'Allotted Space' with all the details of the user to whom this space is allotted. The server informs the loaning client, the details of the allocation. The loaning client, in turn, creates a virtual drive on the spared space and allocates a drive name and informs the server about completion of the drive allocation process. The demanding client can upload files to this drive. The location of the drive is transparent to the user. The user can access the files stored on the virtual drive as if it is additional drive available on the local machine. The user can access these files from any member node of the network.

2.2 Software Architecture

The software architecture of the UVS system is shown in the Fig 2. It clearly underlines the roles of client and server in the system. At the client end there are two basic parts of the implementation viz. UVS Client and UVS Agent. Various software modules that make the UVS system are discussed below (refer to Fig 2).

Loaning & Locking Manager: This module is a part of UVS Agent. The UVS Administrator is given privileges to loan some amount of free space, which can be used by other machines. This space is spared from a drive which is in normal operating mode is used by the local user as a local drive.

Connection Manager: For every transaction on the network the user has to have well set communication facility. The module helps establish and maintains connection between two communicating entities.

File Transfer Manager: The module is meant to transfer of files between two UVS Clients through the upload & download functions. It also maintains data about where the file is now saved.

File Splitter: It is an operation invoked when there is no enough space on one machine to upload a file & that user does have enough space on other machine to save remaining part of the file.

Request Handler: This module is basically getting invoked when there is a request from UVS User to UVS Agent for displaying system information or to loan space.

User Authentication Manager: The module authenticates all types of users on the UVS

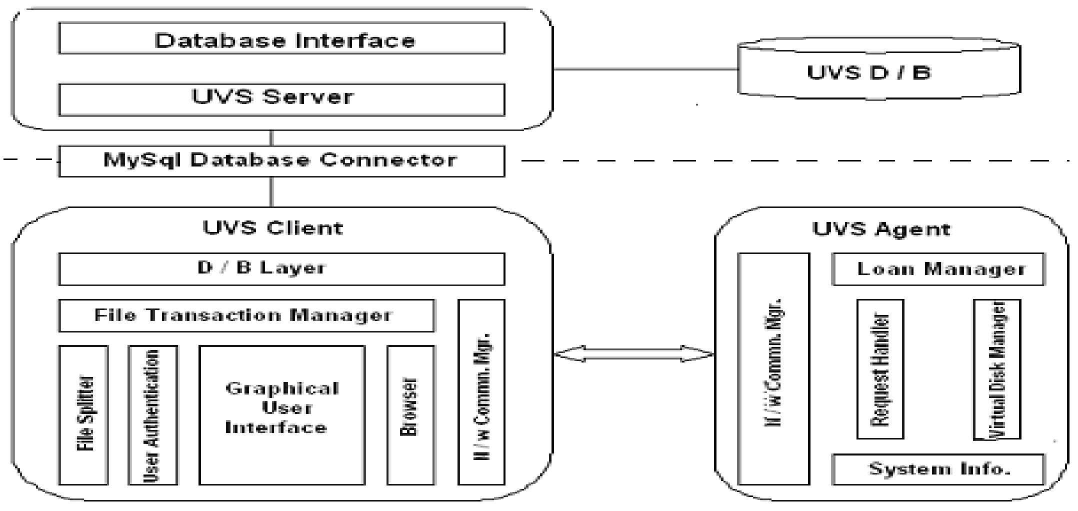

Fig 2 Architectural Diagram of UVS

This is done only when there is demand for UVS space. Once some space is allocated to any user and drive letter is assigned to the virtual drive created on the UVS space for the demanded size, that drive is locked for further use by the same user to whom the drive is allocated. The space can be used by the local user, till it is allotted and locked. This makes the UVS loaning and allocation very flexible.

3 Algorithms

The algorithms are divided in four basic modules. These modules are...

1. UVS Server
2. UVS Client
3. UVS Agent and
4. Communication Manager

There are sub-modules within each of the modules stated above. These modules are interrelated. This is shown in Fig 3 below.

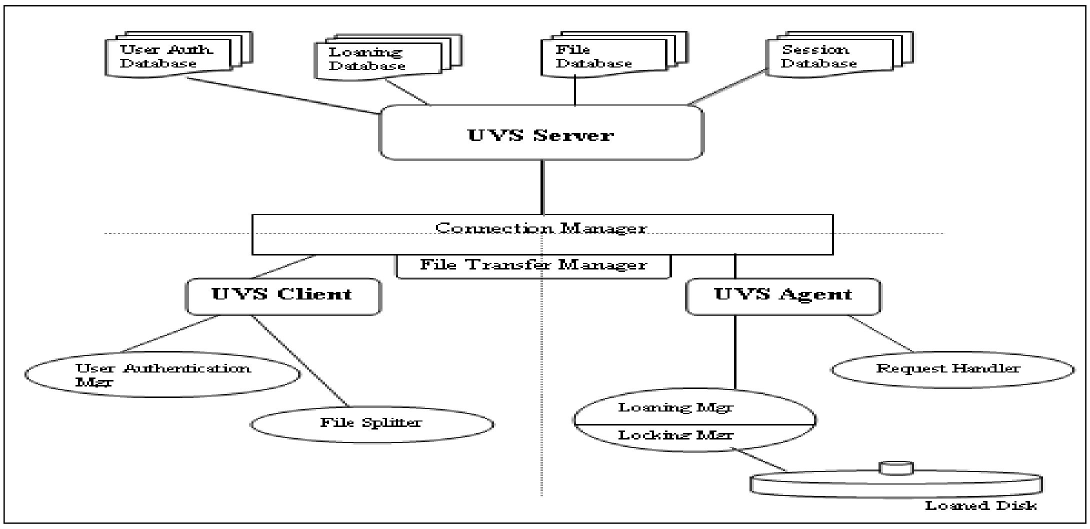

Fig.3 Module Interaction diagram of UVS

UVS Server:

```
        Create Socket
        Listen to the request from client
        Request received
        Check the request
                1. User authentication
                        Get the User Name and Password
                        Verify with entry in the User Authentication Database
                                Found 'Correct' –    Allow the user to access UVS by transferring the details of
                                                     Files stored by the user on UVS space, if any
                                      'Incorrect' –  Prompt the user about failure
                2. Loan request
                        Get the size of space a client node wants to Loan
                        Create record of the loaned space, against the node, in the Loaning Database
                3. UVS Service
                        Demand of Space request
                                Get the size of space required by the user
                                Check whether contiguous space loaned by node is available –
                                        'Yes' – Allocate the space to the user by creating a virtual drive of the
                                                demanded size and make necessary entry in the database
```

```
                    'No'  – Compute the demanded size of space from more than one node.
                    Allocate space to the user and make entries in the database
        File Upload
            When user uploads a file, to the allocated Space; make record entry in file database
        File Delete
            When the user wants to delete a file from UVS space allotted; delete the record
            corresponding to the file from the file database
After completion of any of the task return to Listen mode
```

UVS Client:

```
When the user wants to use UVS for File Upload or Download
    Send user name and password to the server for authentication on the server socket.
    Get permission granted from the server
    If server informs about contiguous space available on the UVS space; 'Upload' the file to this space.
    If sufficient space not available on single node; get the sizes of spaces available on different nodes.
    Split the file in fragments of appropriate sizes
    Send the file fragments to respective nodes
While 'Download'
    Get the information about the location/s of the file
    Read the file from the source/s; If from more than one node – merge the fragments received from
    different nodes in proper order
```

UVS Agent:

When space is 'Loaned' to UVS space, the loaned space is spared from use locally. The server informs about allocation of space and then a virtual drive of the reported size is created with proper drive letter assigned.

```
  File Write Request
    Get request from the authenticated user for storing a user file or fragment of a file
    Create a file (or directory) by the name specified by the remote UVS user
    Receive contents of the specified file or fragment and store the same in appropriate place in the drive
    Make necessary entry to the local file system
    File Read Request
        Get the file name with path, if any from the authenticated user
        Locate the file in the UVS space
        Open the file in 'read' mode
        Send the file as a response to the request on the client socket
    File Delete Request
        Get the file name with path, if any from the authenticated user
        Locate the file in the UVS space
        Delete the file from the UVS space
        Update the local file system
After completion of every task report to the server to append necessary changes to the UVS system database/s
```

```
Communication Manager:
    Create a Socket
    Listen for any request from any user from any node of the network
    Request received
    Check whether this is a 'write' or 'read' or 'delete' request
                If 'write' request
                        Get the user name
                        Find the virtual drive from the mapping function
                        Receive file from the user; store it on the specific drive
                        Connect to the server socket; transfer of file information
                If 'read' request
                Get the file name with path
                        Keep the client socket active till the complete file is read
                If 'delete' request
                        Get the file name from the remote user
                        Find the file on the UVS space and delete the file
                        Delete metadata record of the file from the local drive
                        Connect to server socket; inform the server the deletion
```

Careful study of the algorithms given above, it is very clear that, except for the actual data storage and maintenance, major tasks are server centric. The beauty of the technique lies in the fact that, once the space is allocated by the server, further file handling and handling requests from users whose files are stored on the loaned space, is looked after by the two or more nodes involved in the file transaction. This task is carried out by the Communication manager and file transfer manager. The server is spared from this burden.

4. Performance Measurement

To test the performance, five PC's with slightly different configurations are taken. Two of them were having P4-3GHz processors, 256MB of RAM. Another two were Atom processor, 1.6GHz, 2GB of RAM & Pentium processor, 1.2 GHz, 512MB of RAM; the fifth machine was having configuration as Pentium processor, 1.2GHz, 512MB of RAM.. These machines were running on Microsoft Windows XP Operating system platform. The performance is taken after running Upload & Download function of UVS Client. The observations are noted by considering different constraints. The details are as follows.

The files with average size 68.9 MB were, simultaneously, uploaded on different machines, with n/w speed as 54Mbps, was taking on an average 136 Sec. The bandwidth was getting divided as the Network traffic.

The time taken for download of the same file took considerably less time, as the location of the file on UVS space was known. The download time recorded was 70 sec. Throughput of UVS is compared with other systems. The results are summed up in Table1 and Table2 for upload and download of files, on UVS space.

Table 1 Comparison of Performance for File Upload

System / Parameter	Windows Share Drive	LanStore	Disk Clustering	UVS
Average delay for 1MB (Sec)	0.86	51.2	22.28	1.97
Throughput MB/Sec	1.16	0.02	0.04	0.51

Table 2 Comparison of Performance for File Download

System / Parameter	Windows Share Drive	LanStore	Disk Clustering	UVS
Average delay for 1MB (Sec)	1.5	12.1	2.5	1.02
Throughput MB/Sec	0.66	0.08	0.4	0.98

Observations of Table1 and Table 2 result into a very important fact that UVS gives better performance than Windows share drive utility. Both the utilities have similar way of functioning. UVS has advantage over Windows share drive utility; UVS mounts the drive where user files are stored on UVS space when the user logs into the system and in case of Windows share drive; the user has to find the drive where the files are loaded and mount the drive whenever needed.

5. Modified 'UVS'

The algorithm at the server end if modified, slightly, the resultant algorithm will give better flexibility for the storage space allocation. The implemented algorithm for UVS server is repeated below for ready reference of the reader. The modifications in the algorithm are shown in bold script

```
UVS Server:
    Create Socket
        Listen to the request from client
        Request received
        Check the request
            1. User authentication
                Get the User Name and Password
                Verify with entry in the User Authentication Database
                    Found 'Correct' –   Allow the user to access UVS by transferring the details of
                                        Files stored by the user on UVS space, if any
                          'Incorrect' – Prompt the user about failure
            2. Loan request
                Get the size of space a client node wants to Loan
                **Create a virtual drive of the size 'Loaned' on the same node and drive letter to this node**
                Create record of the loaned space, against the node, in the Loaning Database
            3. UVS Service
                Demand of Space request
                    Get the size of space required by the user
                    Check whether contiguous space loaned by node is available –
                        'Yes' – Allocate the space to the user by creating a **Folder by the name
                                of the demanding user** and make necessary entry in the database
                        'No' – **Find a node where space is available; create a folder by the**
```

```
                                        name of the demanding user on this node also and make
                                        necessary entries in the database
                    File Upload
                         When user uploads a file to the allocated Space; make record entry in file database
                    File Delete
                         When the user wants to delete a file from UVS space allotted; delete the record
                         corresponding to the file from the file database
After completion of any of the task return to Listen mode
```

The effect of the modifications can be described as follows…

Proposed modification is related to the way the system responds to the 'Loan' request. The space loaned on any of the nodes used to be spared for network storage under UVS and recorded on the loaning database; in the implemented algorithm. When any user demanded for the space on UVS, part of the loaned space was allotted to the user by creating a virtual drive on the loaned space and assigning a drive name. This arrangement was creating a UVS space of fixed size, per demanding user. If the user is interested in 'uploading' a file which could not be accommodated in the remaining space allocated to the user, the user was prompted about non availability of sufficient space and was expected to create space by removing some files on the UVS space. At the same time there could be user, who had been allotted space on the same node, had not used the space allotted on the UVS drive. The modified algorithm rectifies this shortcoming by providing UVS space of flexible size to the users on fixed size UVS space loaned by individual nodes in the system.

Since space allocated to the users by creating folders by their names on available UVS space on virtual drive on any of the member nodes, every demanding user will have space of variable size on the UVS space. The fragmentation of files can be minimized, thereby, improving the system performance.

6. Related Work

1) Study of used space on desktop machines On an average, 62% of a machine's raw disk capacity is unused, which is consistent with results of other disk usage studies, which have reported storage utilization rates of about 50% [2] . It shows the minimum space available on a machine during the study period – providing a more realistic assessment of how much space a mass storage system can safely use on a machine. The results are nearly the same – the aggregate of the minimum available disk space is 32.8 TB with an average of 45 GB per machine.

2) Existing Systems

There are some existing systems, which can work as a performance measurement tool for UVS. One of these is LanStore [3]. The main design goal was to gather the empty storage capacity into a virtual storage unit. To utilize in an equal way the storage capacity of the member client nodes, the files are divided into equal fragments. Every storage node has the same number of stored data fragments Storage@desk[2] (SD) utilizes excess desktop disk capacity within an organization to create large virtual storage volumes that meet target QoS and security goals and provide access to a large number of client nodes using standard interfaces. The distributed RAID approach used in xFS [7] and Petal [6] was applied to build a virtual disk with block interface. On top the virtual block devices, higher level file systems and distributed file systems were built, like the metadata manager in xFS and Frangipani.

7 Conclusion

Success of proposed UVS (Unified Virtual Storage) System would grossly depend on its ability to extend storage capacity using existing infrastructure, tapping unused storage space on network nodes & effectively managing/allocating the loaned space. The performance of UVS is dependent on the LAN speeds and Traffic conditions. The performance is quite nice with 100 Mbps networks, but fairly good with 54Mbps wireless connectivity. Mobility can also be provided to the users of UVS.

Identification of the mobile nodes calls for a little tedious algorithm. One more advantage of UVS over other systems is, UVS allows the user to have hierarchical directory structure for storing and categorizing the files stored on UVS space, as against flat file system used for some other implementations. This adds to the metadata stored on the server, as the directory structure also has to be maintained. This is a bottle neck while stripping files before they are uploaded. UVS technique is implemented on local area networks with different hardware configurations but homogeneous operating environments. The system is not, really, tested for heterogeneous environments. It is very much possible to implement this over the Internet which, if done, will be possible to support web-based clients.

In the implementation of UVS the 'Loaning' and allocation of space from the loaned space is done by the administrator. This can be made software driven so that the loaning and reclaiming of spare space is easier for any client.

References

[1] Sun Microsystems Inc., NFS: Network File System

[2] H. Howie Huang, John F. Karpovich, and Andrew S. Grimshaw "A Feasibility Study of a Virtual Storage System for Large Organizations", 2nd International Workshop on Virtualization Technology in Distributed Computing (VTDC 2006).

[3] Vilmos Bilicki, LanStore: A highly Distributed Reliable File Storage .System; .Net Technologies Conference Proceedings, Unioin Agency,-Science Press, Plzen, Czech Republic2005

[4] Eunsung Kim Hyeong S. Kim Heon Y. Yeom. "GiSK: Making Secure, Reliable and Scalable VO Repository Virtualizing Generic Disks in the Grid", in Eighth IEEE International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA'05). Protocol Specification, RFC 1094. March 1989

[5] Paul massiqlia and frank bunn, "Virtual Storage redefined: technology and applications for storage virtualization", 2nd ed., VERITAS, 2000

[6] E. Lee, and C. Thekkath, "Petal: Distributed virtual disks," in Proceedings of the ACM 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPWS), 1996.

[7] T. Anderson, M. Dahlini, et al.. "Serverless Network File Systems," ACM Transactions on Computer Systems (TOSC), Feb. 1995.

[8] Edward K. Lee and Chandrohan A. Thekkah. "Petal: Distributed virtual discs". SIGPLAN Notices, 31(9):84-92,1-5 October 1996.

[9] E. Riedel, "Storage systems - not just a bunch of disks anymore," QUEUE, pp. 32–41,2003.

[10] Vazhkudai, S, et al. "Freeloader: Scavenging Desktop Storage sources for Scientific Data". In Supercomputing 2005 (SC'05): Int'l Conference on High Performance Computing, Networking and Storage. 2005.

[11] Ms. S.V.Patil, K.N.Honwadkar, "Unified Virtual Storage: Virtualization of Distributed Storage in a Network"; Digital Library URI: http://www.ijcaonline.org/archives/number22/447-681, published in IJCA Journal on Feb 25, 2010.