

Bachelor's Thesis (AMK)

Information Technology

2009

Chunhui Liu

# An Alternative Scalable Storage System



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

UNIVERSITY OF APPLIED SCIENCES

Degree programme | Specialisation : Information Technology

Date: 20.Nov.2009 | Total number of pages: 45

Instructor: Vesa Slotte

Author: Chunhui Liu

## An Alternative Scalable Storage System

With the development of computer processor, the Input/Output (I/O) gap between the Central Processing Unit (CPU) and the storage system widens. The storage system becomes the I/O bottleneck of the whole system. Solving this problem is a popular topic for many researchers. Redundant Array of Independent / Inexpensive Disks (RAID) is a widely used technique to handle this problem nowadays. Many RAID products are available on the market. However, for small companies, these products are too expensive. In this thesis, a design method will be introduced to set up a scalable Software RAID. Compared to the RAID products, Software RAID is cheaper and easier to realize. Customers can configure the Software RAID according to their own demands. Software RAID will be a suitable choice for small companies seeking storage solutions.

KEYWORDS: RAID iSCSI SAN NAS

## **CONTENTS**

<b>ABBREVIATIONS</b>	<b>5</b>
<b>1 INTRODUCTION</b> .....	<b>6</b>
<b>2 RAID BACKGROUND</b> .....	<b>7</b>
2.1 Introduction of RAID	7
2.2 RAID Classifications	8
2.2.1 Software RAID	8
2.2.2 Hardware RAID	8
2.3 RAID Levels	9
2.4 Network storage introduction	15
2.5 Introduction of iSCSI	17
2.5.1 Definition of iSCSI	17
2.5.2 How does iSCSI work?	17
2.5.3 iSCSI initiator	18
2.5.4 iSCSI target	18
2.6 Other terminologies	18
<b>3 THE DESIGN OF SYSTEM ARCHITECTURE</b> .....	<b>21</b>
3.1 Background of the design idea	21
3.2 The structure of the design	21
3.3 The benefit of this system compared with RAID products in the market	22
<b>4 EXPERIMENTS</b> .....	<b>24</b>
4.1 Setting up the experiment environment	24
4.2 Experiment 1	35
4.3 The result analysis	37
4.4 Experiment 2	38
4.5 The result analysis	41
<b>5 CONCLUSIONS</b> .....	<b>42</b>
<b>6 REFERENCES</b> .....	<b>43</b>

## **FIGURES**

FIG. 1 RAID 0	10
FIG. 2 DATA APPING IN RAID 0	11
FIG. 3 RAID 1	11
FIG. 4 RAID 4	13
FIG. 5 RAID 5	13

FIG. 6	RAID 5 WRITING	14
FIG. 7	RAID 6	14
FIG. 8	DAS, SAN AND NAS STRUCTURE	16
FIG. 9	ISCSI PACKET FRAME	18
FIG. 10	THE ARCHITECTURE OF SOFTWARE RAID	18
FIG. 11	FREENAS GUI	27
FIG. 12	THREE 120 GB DISKS ON LINE	28
FIG. 13	ISCSI TARGET STARTS	28
FIG. 14	RAID 5 ARRAY OF 228GB OUT OF THREE DRIVES	29
FIG. 15	RAID 5 PACK COMPLETE	29
FIG. 16	NEW DISK IS CONNECTED (DISK2-- 447,25GB)	33
FIG. 17	PARTITIONING OF NEW DISK (DISK 2)	33
FIG. 18	NEW VOLUME OF VIRTUAL DISK (I) CREATED	34
FIG. 19	TWO VIRTUAL DISKS OF (I) AND (J) ARE AVAILABLE TO USE	34
FIG. 20	QUICK BENCH TESTING	35
FIG. 21	THE READING SPEED OF DISK (I)	35
FIG. 22	THE READING SPEED OF DISK (J) OF SINGLE RAID 5 PACK	36
FIG. 23	SINGLE DISK (G) WITH NO RAID IS AVAILABLE.	36
FIG. 24	THE READING SPEED OF A SINGLE DISK (G) FROM BACK-END	37
FIG. 25	COPYING DATA	38
FIG. 26	THE DATA IS COPIED TO DISK (I)	38
FIG. 27	REMOVING ONE DISK FROM RAID 5	39
FIG. 28	RAID 5 IN DEGRADED MODE	39
FIG. 29	THE VOLUME (I) IS HEALTHY AFTER REBUILDING	40
FIG. 30	THE LOST DATA IS FOUND	40

## **ABBREVIATIONS**

ATA drive—AT Attachment drive

CPU—Central Processing Unit

CD—Compact Disc

DAS—Direct Attached Storage

DVD—Digital Versatile Disc or Digital Video Disc

ECC—Error Correcting Code

FC—Fibre Channel

GB—Giga byte,  $10^9$  byte

Gb—Gigabit,  $10^9$  bit

Gb/S— $10^9$  bit per second

GUI – Graphical User Interface

HP—Hewlett-Packard Company

IDE drive—Integrated Drive Electronics drives

IETF— Internet Engineering Task Force

I/O—Input /Output

iSCSI—Internet SCSI

JBOD—Just a Bunch of Disks

LAN—Local Area Network

MB—Mega byte,  $10^6$  bytes

Mb/S—Mega bits per second i.e.  $10^6$  per second

MD—Multi Device

MS—Microsoft Corporation

NAS—Network Attached Storage

OS—Operating System

PC—Personal Computer

RAID—Redundant Array of Independent / Inexpensive Disks

RPM—Round Per Minute

SAN—Storage Area Network

SAS— Serial Attached SCSI

SATA—Serial ATA

SCSI—Small Computer System Interface

TCP /IP—Transmission Control Protocol / Internet Protocol

WAN—Wide Area Network

XOR—the logical disjunction symbol of Exclusive Or, sometimes EOR

# 1 Introduction

Along with the electronic technique development, computers and the Internet have made people's work and everyday life easier at. People use them to do business, provide services, communicate with friends, watch videos, or "travel" to another country just by clicking the mouse. Fast access to the Internet is gradually becoming an important part of social life and as people increasingly rely on this virtual world, once a break occurs, people may feel that they are back to the prehistoric age.

From an IT engineering point of view, all of these activities depend on the database of the system, how large, how fast, and how stable it can be. In the past decades, the heart of computer, the CPU, has achieved speed of above 2,5Ghz<sup>[12]</sup> and the volume of RAM has also increased up to 12(3\*4) GB<sup>[13]</sup>.

However, the advance of the mechanical part of the data storage has not been realized so quickly. The hard disk's rotation speed has now been up to 15 000 RPM (round per minute) from 5 400 RPM ten years before. That means, even if the CPU can process a large amount of data, the hard disks cannot handle enough data in time for it. A lot of precious CPU time is wasted. The I/O gap between the CPU and the disk makes the storage system become the bottleneck of the whole computer system.

This thesis attempts to answer the following questions: How is it possible to provide middle and small companies with a relatively good service with the computers and hard drives that are already obsolete in companies? Is there a way to use these computers and drives better and enhance their performance? How can cost and time-efficient solutions for providing relatively larger data storage be realized?

This thesis focuses on setting up a system to provide a data storage service using ATA drives by RAID techniques. The first chapter gives a brief introduction, the second one is about the background of the RAID technique, the third one records the practice of the experiments and the last chapter concludes this thesis and discusses the advantages and disadvantages of the experiments.

## 2 RAID Background

Network connectivity plays a crucial role as it connects the outside world through servers to users. And beyond bandwidth for connecting with networks, I/O gap difficulties also exist in the bus level of computer systems we usually use.

From a user's perspective, for example, hospitals and banks need a high degree of safe and stable service, film producing manufacturers need large reading of high performance, as well as newspaper reading group.

Such demands require the data storage system be provided with high performance, good fault tolerance, easy operation, and scalability.

While researchers and engineers are devoted to find a way to increase the RPM of hard drives, they try to explore some methods to change the structure of the existing way to transfer data inside the computer. Thus, many solutions have been put forward.

One practical solution is to distribute the controller and bus loads to multiple and identical parts. The data access can be done in such a way that users just see the whole multiple packages of disks as a single simple normal disk. As the number of hard disks increases, the accumulated speed of all disks may alleviate the bottleneck problem in system. RAID <sup>[1]</sup> (Redundant of Independent Disks, which is several independent disks running in parallel, provides us such a kind of method to provide quick and simple data access.

Computer storage area provides clients/users with a faster connection with central database hardware management. ISCSI offers a way to connect in different locations with RAID technology.

### 2.1 Introduction of RAID

RAID is the abbreviation of Redundant Array of Inexpensive Disks. RAID technology is used to store data across a group of disks. Sometimes Independent is also used instead of Inexpensive That is to say, two or more disks work cooperatively to satisfy disk I/O requests. These cooperative groups of disks are called disk arrays. Each disk array is treated by the operating system as a single



logical volume. It was firstly put forward by David A. Patterson, Garth A. Gibson, and Randy H. Katz<sup>[1]</sup>. Nowadays, with technology evolution both small companies and consumers can afford large drives, and RAID provides an important solution for computer users whether there is a cost issue. RAID is a technology in computing storage area that focuses on deploying a method to use PC disk drives as a redundancy by array arrangement. Like many other computer techniques, RAID is divided into two categories as Hardware RAID and Software RAID and it has been developed to many levels<sup>[3, 4]</sup>.

## **2.2 RAID Classifications**

RAID configurations can be implemented by either software-based or hardware-based arrays. Considering that all RAID is actually based on specialized software, it really comes down to where that specialized software loads from. Software-based arrays are usually controlled through the host computer's CPU and hardware-based arrays are implemented as firmware to an on-board host-based adapter card or external array controller.

### **2.2.1 Software RAID**

Software RAID is a software implementation of RAID. It is based on operating systems. It is software that runs in the kernel (block device) code and does not need to install the expensive special controller card into the computer. The Linux software RAID function is enabled through the kernel and the "md" kernel module. The major advantage in software RAID is the low cost of implementing compared with some higher priced RAID hardware. In this lab design, software RAID is adopted.

### **2.2.2 Hardware RAID**

With hardware RAID, a dedicated hardware will be used to control the array instead of software. Hardware RAID is often a nice solution for Enterprise computing and I/O. One of its major advantages is that no software is required by the operating system to recognize the disk arrays. Unlike Software RAID,

Hardware RAID uses the co-processor to perform I/O not using cycles from CPU. Any calculation of parity data or duplication of disk writes happens at the co-processor and this frees the system CPU to handle its normal tasks.

In most array products, hardware RAID is implemented. They are especially designed for each product type. There are typically two classifications<sup>[7]</sup>:

1. Bus-based or Controller Card Hardware RAID
2. Intelligent, External Hardware controller

One of the disadvantages of hardware RAID is, regardless of which type of hardware RAID is chosen, the cost is much higher than a new host CPU.

## 2.3 RAID Levels

Inside a RAID, disks are formed into a massive capacity, high speed and high reliable storage system<sup>[4, 5]</sup>. By using striping, the system can process data in RAID in parallel, providing users with a faster I/O speed and shortening the disk I/O waiting time.

### Striping

Striping is not a true RAID level, but it is a useful method by which a block of data can be broken down to smaller pieces and written to multiple places in a round robin fashion simultaneously<sup>[3]</sup>. Reading the data is also substantially faster, especially in parallel disk access. RAID 0 uses this method and sometimes RAID 0 is called Striping. (But striping itself is a technique to deliver and spread data among the drives.)

### Linear mode

Linear mode is a non-RAID capability. Linux supports it when setting up arrays. This is a way to gather all the drives sequentially therefore forming a large drive. In linear mode, there is no redundancy and performance increase.

RAID is classified into many different levels according to its redundant structure. Normally there are RAID0, RAID1, RAID2, RAID3, RAID4, and RAID 5<sup>[1]</sup>

## RAID 0

With RAID 0, the disks are striped and the data stored on them are evenly delivered among these disks. The greater the number of disks, the less reliability it can provide.

When an ideal RAID 0 is implemented, it separates I/O operations into identical size and delivers the data evenly to two or more designed disks therefore increasing the performance. This array level is useful if large amount of reading is required by the clients, but since the data is stored without data redundancy (only striping), if one disk fails, the whole array pack becomes useless.

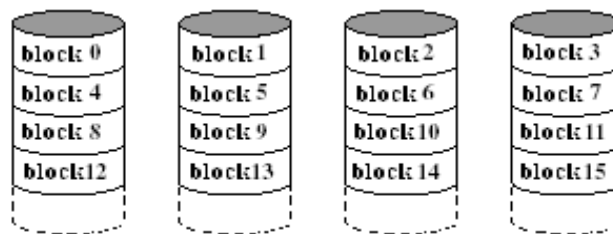


Fig. 1 RAID 0

As shown in Figure 1, reading and writing sectors of data are interleaved between multiple drives. The disadvantage of RAID 0 is that: even if only one drive fails, the entire array is affected. Figure 2 shows the mapping between physical space and logical space in RAID 0.

RAID 0 is adopted when the users need high performance quality and do not care much of data protection. It is recommended for vast reading implementations, such as video producing, editing, and pre-press applications.

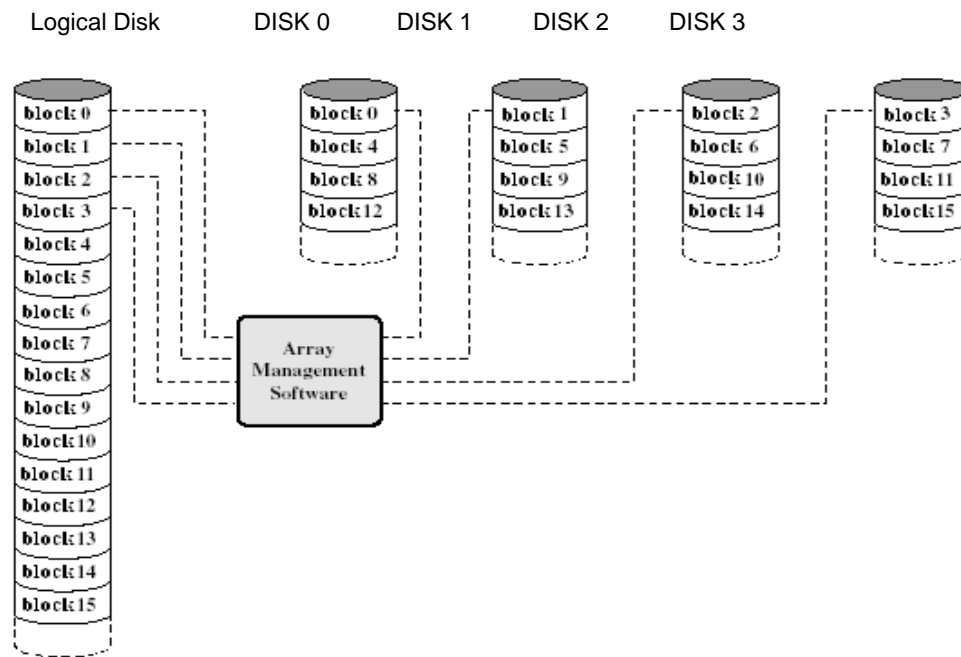


Fig. 2 Data mapping in RAID 0

### RAID 1

RAID 1 is also called mirroring. It uses a large space to provide 100% redundancy. In RAID 1, the same data is simultaneously written to the two devices, so, if one drive fails, all the data is available on another drive with no significant loss in read/write performance. This functions like a mirror, providing a quite stable and safe backup for the data. Each disk has its own data channel to transfer data in parallel. I/O requests will be answered by any disk which has a shorter response time.

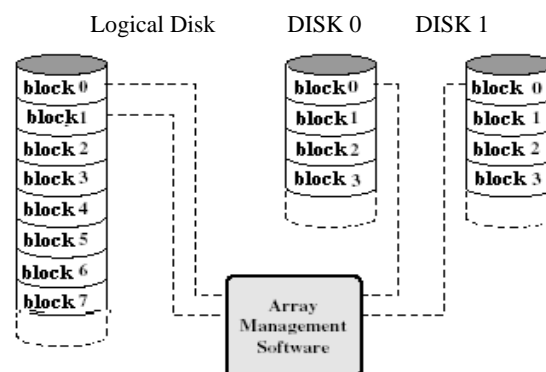


Fig. 3 RAID 1

RAID 1 provides full fault tolerance for a single disk failure. When one disk fails, all of the data is still accessible on the mirrored disk. It is not needed to rebuild data. There is no significant performance degradation when accessing the remaining<sup>[14]</sup>. RAID 1 is useful for read-intensive, fault tolerant required services such as in banks, hospitals when reading performance and reliability is more important than storage capacity. However it also consumes a large amount of time and disks apparently.

### **RAID 2**

RAID 2 uses the Hamming Error Correcting Code (ECC). In RAID 2, every data bit is striped across a disk array. The ECC code is kept in an ECC disk. When reading, the ECC code is used to correct single disk error. This level of RAID is seldom in use nowadays.

### **RAID 3**

RAID 3 is a byte-level striped array. It has a dedicated parity information disk. The data of all disks in the same stripe performs the XOR parity calculation and enters the result into the parity disk. In normal cases, i.e. , no disk fails in the system, the RAID 3 read performance is the same as the RAID 0 one, but it has data fault-tolerance. The RAID 3 write performance drops a lot since a write operation may produce a parity calculation which may bring several reading operations, parity calculation and a parity disk write operation. RAID 3 is not used often now in practice.

### **RAID 4**

RAID 4 uses block-level stripes. It is different from RAID 0 in that RAID 4 has a special disk as a redundant disk to keep the parity. As shown in Figure 4, the data in the same stripe group are calculated in parity and put in to the parity disk. For example,  $P(0-3) = \text{block } 0 \oplus \text{block } 1 \oplus \text{block } 2 \oplus \text{block } 3$ .

Although data disks can be processed in parallel, the parity disk must wait for all data disk finishing writing before it can be updated. The parity disk becomes the

bottleneck of RAID 4. This is also the reason why RAID 4 cannot contain more disks if users want high performance.

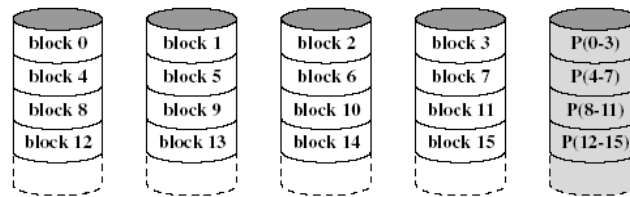


Fig. 4 RAID 4

## RAID 5

The difference between RAID 5 and RAID 4 is that the parity information in RAID 5 is distributed across in all member disks. There is no parity bottleneck any more. RAID 5 delivers both performance and data redundancy. Data is striped across two or more disks but parity is stored in another disk. If a disk problem happens, the array can be reconstructed using the parity information on the same stripe combined with the data on the remaining disks. That is to say, RAID 5 can tolerate a single disk failure and run in a degraded mode. Since some disk space is used for parity, some capacity is sacrificed for this redundancy and, therefore, the total useful logical size is reduced. The approximate formula for determining RAID 5 array capacity is as follows:

$$\text{Size of disk} * (\text{number of disks} - 1) = \text{total capacity} \quad (1)$$

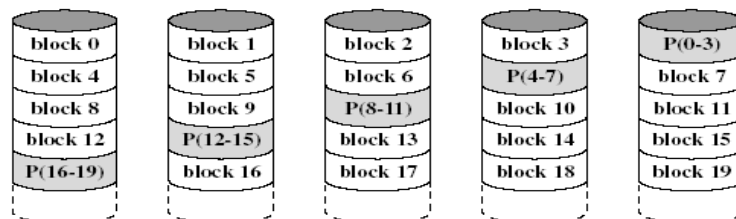


Fig. 5 RAID 5

RAID 5 requires a minimum of 3 disks. To get a balance among cost, performance and fault tolerance, RAID 5 is a compromised arrangement of RAID for data storage and it is widely adopted. However, the cost of calculating and storing parity is time consuming. For every application write request, RAID 5 must perform two reads, two writes and two XOR operations to complete the original write operation (See Figure 6). For example, if a user tries to write block 8,

1. The system first reads the old data block 8.

2. Then it writes the new data to replace block 8,
3. The new data and old data perform the XOR calculation and the value  $P_n$  is acquired. Now, the data update operation is over and the system will continue with the parity update operation.
4. The system reads the old parity  $P(8-11)$ .
5. The  $P_n$  value and the old parity perform the XOR calculation and the new parity is ready.
6. The system writes a new parity to replace  $P(8-11)$ .

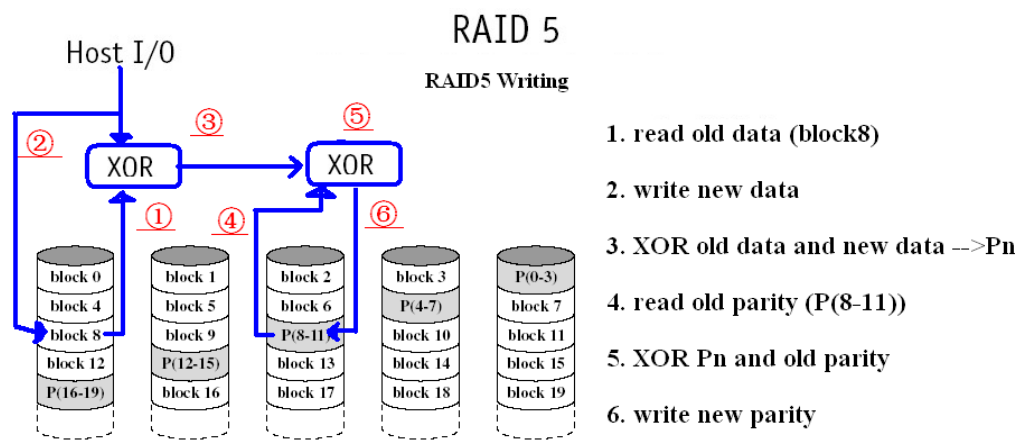


Fig. 6 RAID 5 writing

### RAID 6

RAID 6 is an advanced RAID 5 with the addition of another parity block. It can allow two disks fail at the same time in the array. When writing, at least three disks (data disk, two parity disks) should be operated twice. So the performance of RAID 6 is worse than RAID 5. But since this level allows two disks failure, it increases the system's fault tolerance ability.

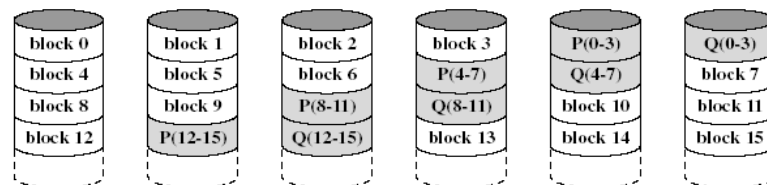


Fig. 7 RAID 6

### RAID 10(1+0)

This pattern is a hybrid one that combines RAID 1 with RAID 0 level functions. First, the disks are mirrored and then striped. RAID 10 has both the advantages of

RAID 1 and RAID 0, which means that the loss of a single disk will not affect the whole performance, as the array will provide a mirror and better performance such as in RAID 0 and RAID 1.

### **RAID 01(0+1)**

This type is similar to RAID 10 while the process is reverse.

### **JBOD**

JBOD is not a RAID level but it is useful in practice when creating disk arrays. The full name of it is 'Just a Bunch of Disks'. In the JBOD model, data is appended to the disks sequentially and the system is not affected by how much the size of each drive is. So, a user can create a large logical drive in the JBOD model using lots of different disks.

### **Other levels**

There are also some non-standard RAID levels promoted by individual vendors for example RAID 7, RAID 5E etc. which are not in the scope of this thesis.

## **2.4 Network storage introduction**

### **Direct Attached Storage (DAS)**

Direct Attached Storage refers to the kind of computer storage architecture where the data storage devices are directly connected to the server or workstation using cables, so there is no network between devices and the server. Usually, the cables are SCSI wires or FC cables. DAS devices depend on the host OS to handle I/O. All DAS devices share the same cable. With the numbers of DAS devices constantly increasing, the cable between DAS and host may be very busy. Thus, the system performance is affected.

### **Storage Area Network (SAN)**

A Storage Area Network (SAN) is a high speed architecture that interconnects remote storage devices with data servers. These storage devices are just like normal



local attached devices, that is, users can do reads and writes operations with them. In SAN, the storage system is separated from the host. The storage devices form a local network and communicate with outside networks via a switch. Inside the SAN, users can easily add more storage devices without affecting the host. Any device failures may not harm the whole system.

A SAN consists of servers, back-end storage devices and SAN connectors. There are two types of SAN, FC-SAN and IP-SAN, according to the connecting media. Compared with DAS, a SAN can consist of more devices. Devices in a SAN are connected using FC or Gigabit Ethernet which is faster than the SCSI cable in DAS.

### Network Attached Storage (NAS)

NAS consists of storage equipment such as hardware controller arrays, CD or DVD driver, tape driver or mobile data storage media and dedicated server. A NAS device has its own file system, usually a simplified UNIX/Linux, or a special Windows kernel. It optimizes the file system management and accessing, and provides file-level visiting and data sharing for remote transfer. NAS is plug-and-play. It can directly connect with the network

The difference between SAN and NAS is where the file system lies. In SAN, the file system is located in each server while NAS has its own file system. Figure 8 shows the architecture of DAS, SAN, and NAS.

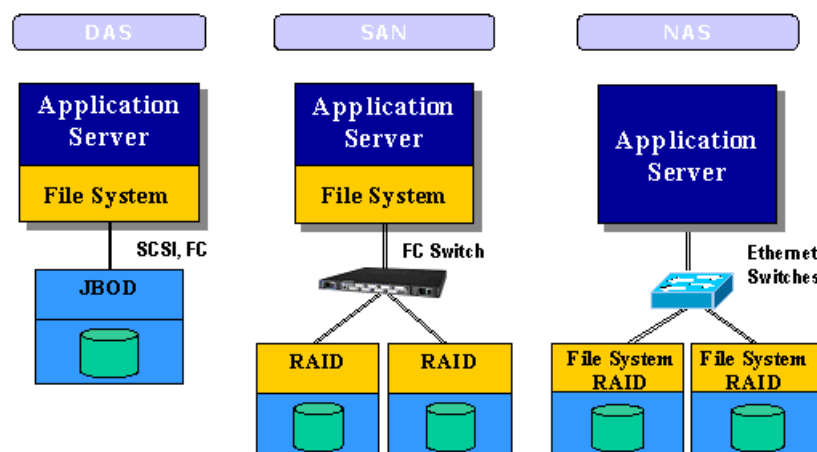


Fig. 8 DAS, SAN and NAS structure

(Redrawn from [http://en.wikipedia.org/wiki/Storage\\_area\\_network](http://en.wikipedia.org/wiki/Storage_area_network))

## 2.5 Introduction of iSCSI

iSCSI is a standards-based transport protocol for connecting storage devices over TCP/ IP (Transfer Control Protocol/Internet Protocol) network. When the operating system or user applications try to use the storage, an iSCSI device appears to be a local SCSI disk. The iSCSI protocol defines the method that SCSI commands can be put into TCP/IP packets and transmitted over Ethernet network.

### 2.5.1 Definition of iSCSI

iSCSI, which stands for Internet SCSI (Small Computer System Interface), is one of the main methods to connect data storage facilities based on TCP/IP network. It was put forward by CISCO and IBM together. iSCSI is a standard developed by the Internet Engineering Task Force (IETF). iSCSI incorporates the two traditional techniques: the SCSI and TCP/IP protocol. It transfers SCSI commands over IP networks; so on the distant side, users can execute the command and handle local storage. Therefore, not only intranet, local area network (LAN) data transmission, but also wide area network (WAN) can provide a good optional choice for storage if the iSCSI technique is adopted. This popular SAN protocol enables companies to provide data in centric storage arrays from remote servers while hosts just have the illusion that the disks are locally-attached. Compared with other network storage protocols, iSCSI solves the open, capacity, transfer speed, compatibility and security problems. The widely used internet is the solid base for iSCSI. 10 Gb Ethernet also provides larger bandwidth. .

### 2.5.2 How does iSCSI work?

When end users or clients (called Initiators) send a request to remote storage device servers (targets), SCSI commands are also generated with the data request, they are encapsulated into an IP packet and sent via the Internet. At the other end, the server will receive the IP packet and unpack it, then execute the SCSI command and process the data inside a storage device. iSCSI is a bi-directional protocol which can be used to return data in response to the original request. By choosing Hardware Initiator or Software Initiator, a user will get different features. iSCSI-

based products have been produced by many vendors, such as Cisco, IBM etc. Figure 8 shows detailed information of an iSCSI packet frame.

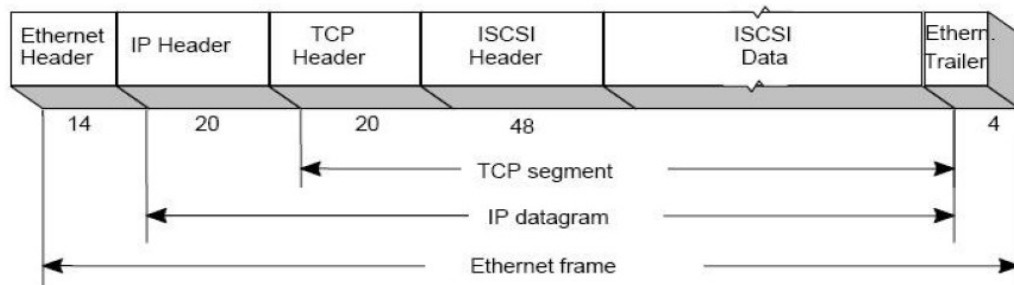


Fig. 9 iSCSI packet frame<sup>[15]</sup>

### 2.5.3 iSCSI initiator

To set up an iSCSI dialogue, there should be two parts connected, the iSCSI initiator and a target. The iSCSI initiator is the part that sends the iSCSI command. Hardware initiators use dedicated hardware to implement the firmware which is professional solutions for large storage needs. Software initiators use code to realize iSCSI commands and this is a common way in most of operating systems.

### 2.5.4 iSCSI target

The iSCSI target is the target drive that is designed to receive the data and information. These targets can be from remote storage backup places, appearing locally as virtual volumes. Users can use them just as local disks to do the routine work conveniently.

## 2.6 Other terminologies

### SCSI

SCSI which stands for Small Computer System Interface was first brought out in 1979. It is a processor-independent standard for system level interfacing between a

computer and intelligent devices that include hard disks, CD-ROM, printers, scanners and more. SCSI can connect up to seven devices to a single controller on the system bus. It hides the complexity of physical format, provides faster data transmission rates than standard serial and parallel ports. All devices are attached to the SCSI bus in a similar manner.

SCSI has developed from SCSI-1 to SCSI-2 and to SCSI-3 now. SCSI-1 supports seven 8-bit peripheral devices. The maximum transfer speed of SCSI-1 is 5MB/s, 20MB/s for SCSI-2 and 320MB/s for SCSI-3.

## **FC**

The Fibre Channel protocol for SCSI is a technology standard that defines a way to transmit the SCSI command and data between a SCSI initiator and a SCSI target across the fibre. FC is a point-to-point serial data channel in the logical structure connection. Its architecture is designed to be implemented with high performance hardware. The data transmission speed is quite fast up to gigabit per second between computer and computer devices. FC does not have its own I/O command set, it just lets other protocols add their command sets onto itself and then transports. FC is a layered protocol; it consists of 5 levels, which are listed below<sup>[16]</sup>:

FC-0 the physical layer

FC-1 the data link layer, which implements 8b/10b signal encoding and decoding.

FC-2 the network layer, which consists of the FC core and the main protocols.

FC-3 the common services layer

FC-4 the protocol mapping layer, in which other protocols are encapsulated into a unit for delivery to FC2

## **SAS**

SAS stands for Serial Attached SCSI. SAS is a point-to-point serial protocol. It combines the advantages of FC, SCSI and SATA; it can provide much higher data transfer speed than SCSI can. So this technology is more used in enterprise computing environment.

SAS is compatible with SATA. SAS controllers support SATA tunneling protocol (STP), they can communicate with SATA devices. Its full-duplex signal transmission reaches 3.0 Gb/s and now 6.0Gb/s. SAS devices can be hot-plugged.

## **SATA**

SATA, the Serial ATA, is a high speed serial link protocol that replaces the desktop parallel ATA. It uses the same low level commands but in a serial way to communicate between devices. It offers several advantages over ATA interface, SATA transfer rates at the lowest of 150MBps, while the fastest speed of ATA is 100MB/s<sup>[17]</sup>. The difference between SATA and SAS lies in that SATA is designed for low cost. SAS is designed for critical data environments in which performance and reliability are important.

## **3 The Design of System Architecture**

### **3.1 Background of the design idea**

As the demand for storage increases, improving the system storage quality becomes vital to designers. It is not uncommon to find out from Internet that the price of ready-made disk arrays is quite expensive and the maintenance work and cost also are not a small amount. For example, a 12 TB disk array from HP costs more than 7000 pounds <sup>[18]</sup>, and a lot of the similar products are more expensive than 10,000 dollars or euros depending on the capacity volume, the type of the arrays and other features they can provide.

At the same time, companies usually have some computers and hard drives that are idle or no longer used so often. Can we use these hard drives to build up a large capacity, readily available and cheap disk array so as to satisfy the companies' demands for high performance massive storage system? This is the background from which the design idea of the lab in this thesis comes.

### **3.2 The structure of the design**

As shown in Figure 10, there are two parts in this system: the front-end and the back-end, connected by an Ethernet switch.

The back-end consists of the computers with several hard disks used to perform the processing data task. The computers are in the same level providing the service. The front-end station will do the maintenance work of the whole system on which both target and initiator will be configured. The back-end PCs will be used to install hard disks to provide the storage service. Both the client PC customers and end users can use the service. After the configuration of the initiator for Windows to provide connection between the two PCs has been performed, the client PC and the front-end PC, and the end user can save his/her documents on these virtual disks in the same way as normal local disks from the client PC.

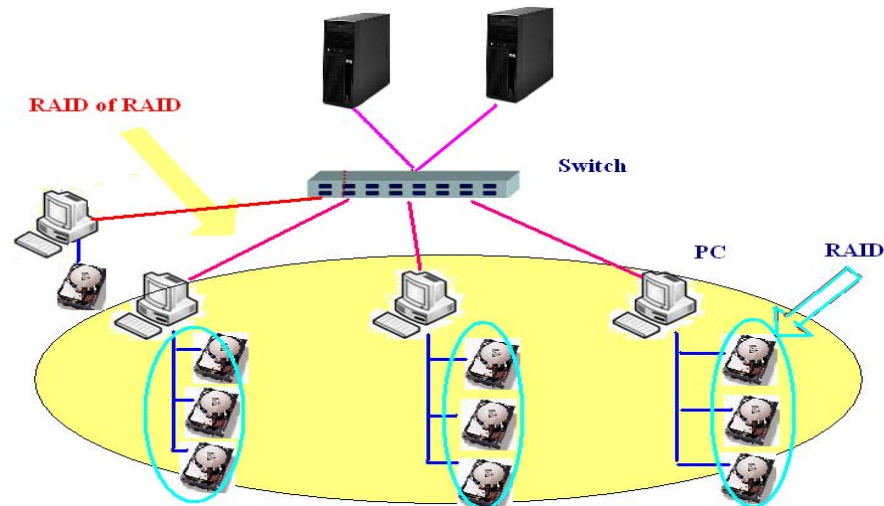


Fig. 10 The Architecture of Software RAID

Each back-end PC has several disks. The disks can be configured as RAID 0, RAID1, RAID3, RAID5, JBOD, and RAID 10 respectively according to the users' demand. We can also treat the group of disks inside the same PC as a virtual disk, and set up another RAID-based on them, i.e., RAID of RAID.

End users generate an I/O operation and send the request. The I/O command (SCSI command) will be contained inside an iSCSI packet, and be delivered to the Ethernet switch via the network. The switch is a bridge connecting the front-end and back-end PCs. After it receives the packet, it will distribute to the PC where the packet aims to go. When the packet arrives, it will be unpacked first, and then the SCSI command will be delivered to the designated disk.

### 3.3 The benefit of this system compared with RAID products in the market

Companies can use the existing machines and ready TCP/IP connections to easily realize this system. Compared with the RAID products in the current market, customers may receive the following benefits from our software RAID system:

- 1) Eliminate the need to buy the expensive new RAID machines and other hardware.
- 2) Regenerate the old idle hard disks and PCs.
- 3) Save the money which will be used to buy new services of software and Internet connection designing.

- 4) Easily manage the RAID system.
- 5) Eliminate the need for new technical training for this service.
- 6) Easily extend the system. Users can add more disks in the back-end PCs to provide large storage space.
- 7) Accumulate the multiple disks processing speed to achieve high I/O performance.



## 4 Experiments

This chapter describes the two experiments carried out with Software RAID. First, detailed information is given about how to set up a RAID 5 using the three disks inside the same PC, and then a RAID 0 will be built on the package of two RAID 5 arrays, treating each RAID 5 array as a single virtual disk. The first experiment is to compare RAID 0 with a single disk to verify that the speed of RAID 0 is faster than the single one. The second experiment is processed with RAID 5. During the processing, one of the disks inside RAID 5 will be pulled out, letting RAID 5 work in rebuild mode. This thesis tries to prove that RAID 5 can provide the lost data again even one disk inside the system fails.

### 4.1 Setting up the experiment environment

**Step 1** I collect all the information of the hardware and software that are needed for the design.

**Step 2** Prepare all the software and equipments to be ready according to the structure of the system design.

A. the physical structure of the system

Hardware conditions:

--Back-end:

4 computers;

10 IDE 120 GB hard drives;

1 Ethernet switch with two gigabit ports;

--Front-end:

1 computer

Some cables

--Client-end

1 personal computer

B. Software versions:

--Back-end: FreeNAS Version 0.69

--Front-end: Debian (Lenny)

--Client-end: Windows XP

Note: The system consists of four back-end PCs, one switch with two gigabit Ethernet ports and two front-end PCs. Three back-end PCs will be installed with 3 identical 120GB IDE disks on each, the fourth one will be installed with a single 120GB IDE hard drive. One front-end, the PC will be installed with Debian operating system, and the client PCs with Windows XP installed on. Hard disks are IDE disks as the existing PCs are IDE-supported. They are all 120GB hard disks.

### **Step 3** Back-end PCs' configurations.

The main consideration of the back-end PCs is that, the back-end PCs can be a lot in numbers. According to the existing situation in company, as many storage PCs can be added to connect with the switch as the ports number allows. Therefore it can provide large volumes for single/different usage purpose. For example, news providing server, video or audio servers can be arranged with different RAID levels.

In this experiment, the back end includes 4 individual computers, three of them for each three 120GB hard disks which are connected to serve as dedicated data storage disks, except that there is a hard disk for computer Operating System usage. The fourth one is also installed with FreeNAS system, the only difference is that there is only one 120GB hard disk assembled for dedicated storage usage. It is used to test what the speed of a single disk is in this environment.

- A. There are only two channels for the four ATA disks. To each channel at most two IDE disks can be connected. In addition, a master drive and a slave drive have to be assigned; otherwise, the system does not work.
- B. Install FreeNAS which is based on Free BSD as the operating system and can be downloaded from the Internet. Instructions for setting up the FreeNAS system can also be obtained from the Internet. When the OS of each PC is configured and the hard disks are ready, the network addresses are needed to connect with a switch.
- C, the network addresses for the four PCs are:

172.26.3.3

172.26.3.4

172.26.3.5

172.26.3.7 (With single disk)

FreeNAS documents describe how to set up software RAID in detail. The three disks needed be formatted, mounted to a mount point and then built in RAID 5 level and after that the array of RAID 5 can be put as an iSCSI target. It has defined a file system also for the formatted disks. **In the experiment I found out that the mount point step should be ignored otherwise there will be a problem connecting the target and initiator.** For the PC with single disk, I did not set a RAID level for it under FreeNAS. This PC with FreeNAS OS is directly connected to the front-end PC through the switch.

FreeNAS can provide JBOD, RAID 0, RAID1, RAID 5 level arrangements with software Raid<sup>[19]</sup>.

#### **Step 4** Front-end PC configurations

The front-end PC in this design serves as a management platform. It establishes the connection between the data storage back-ends and the customer users by an Ethernet switch. With variant customer groups, the different data storage disk packages can be mounted to the customer PCs so they can have own functions on each PC. Here only one client or customer PC is connected.

The front-end computer is installed with Debian OS. This computer is dedicated to do the RAID of RAID arrangement and management.

##### **A.** Install the Debian system.

The Debian Linux Operating system can serve as a Linux server. There are already many versions freely available from the Internet and they are easy maintainable and reliable. FreeNAS was first tried as this server's OS but the result shows it is not as stable as Debian.

##### **B.** Select some packages that RAID will need to use.

In this experiment, “md” and FTP are chosen.

Debian has a GUI mode from where some packages can be chosen, as well.

### C. Assign a network address.

During the installation of the Debian system, a network address needs to be assigned to each PC. In this experiment, the two interfaces are assigned the following addresses:

IP1: a public Internet address

IP2: 172.6.3.2          subnet mask: 255.255.255.0

### Step 5 Connect all the stations.

An Ethernet switch was used to connect all the back-end computers and front-end. The front-end is also connected with another individual computer to test the result. The switch has two gigabit ports, one connecting to the backbone of the lab and another one connecting to the front-end PC. Other ports connecting with the back-end PCs and client PC have a speed of 100Mb/s.

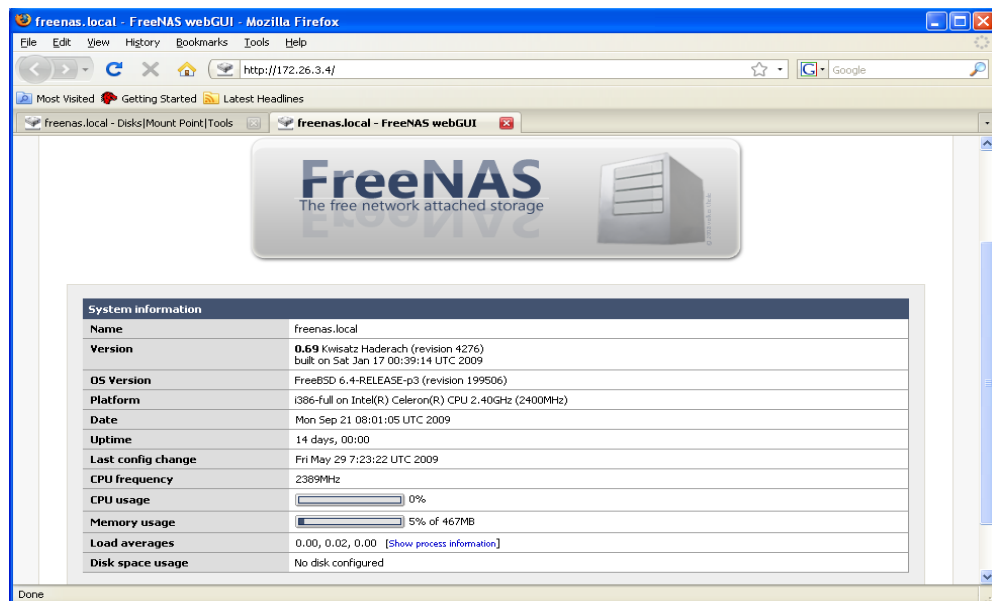


Fig. 11 FreeNAS GUI

And the network address has to be assigned. If the web network interfaces are set correctly, from the guest PC, the GUI of FreeNAS can be accessed as shown in Figure 11.

### Step 6 Create RAID 5 arrays in FreeNAS PCs.

As mentioned before, RAID 5 can be tolerant with one disk failure. With 3 hard drives a RAID 5 array can be formed. This array will appear as a single disk to the remote front end PC which is installed with Debian system.

In order to make a homogenous experiment environment, the same process is carried out for the three FreeNAS PCs. Three 120GB drives combine into a RAID 5 and form a single disk.

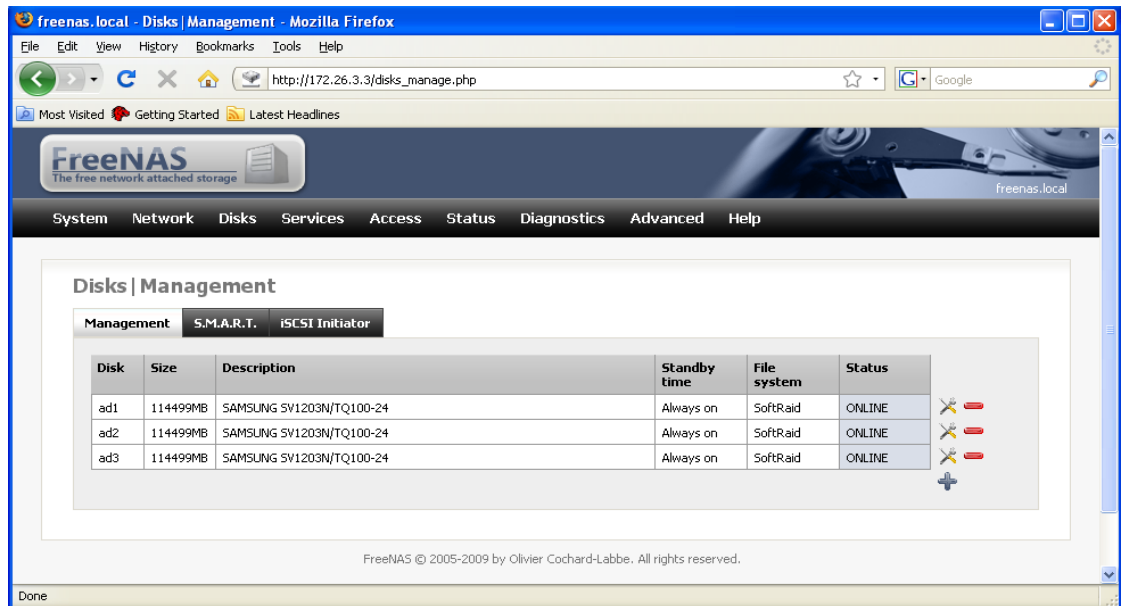


Fig. 12 Three 120 GB disks on line

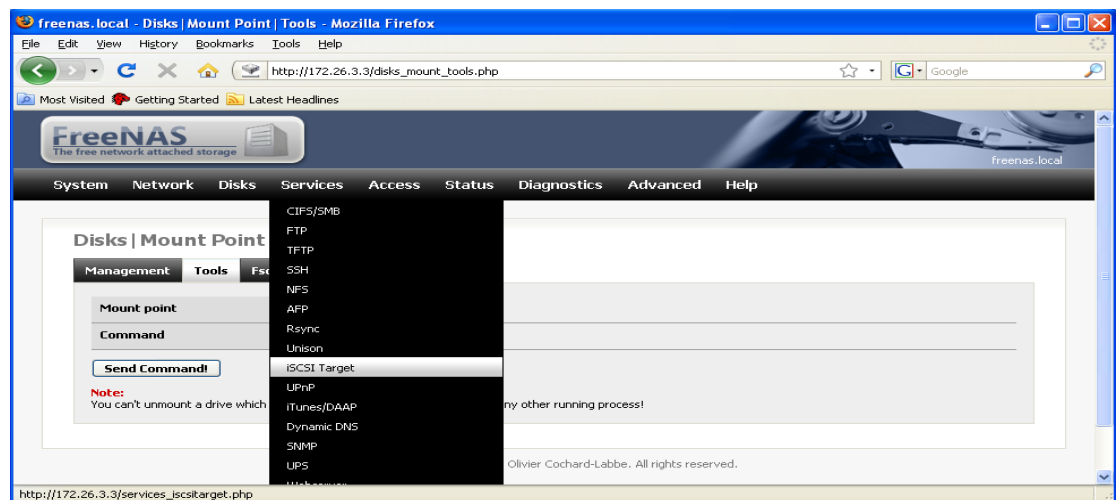


Fig. 13 iSCSI target starts

### Step 7 Create RAID targets in FreeNAS PCs.

To be sure that the single disk shall be recognized by the front-end PC, it has to be defined as a RAID target first. From the FreeNAS management panel, we establish each target for each PC, see Figure 13. Therefore, there are now three large single disks each of which has a volume of 228 GB from the back end PCs, as shown in Figure 14. Figure 15 shows how each disk becomes available from the Front-End.

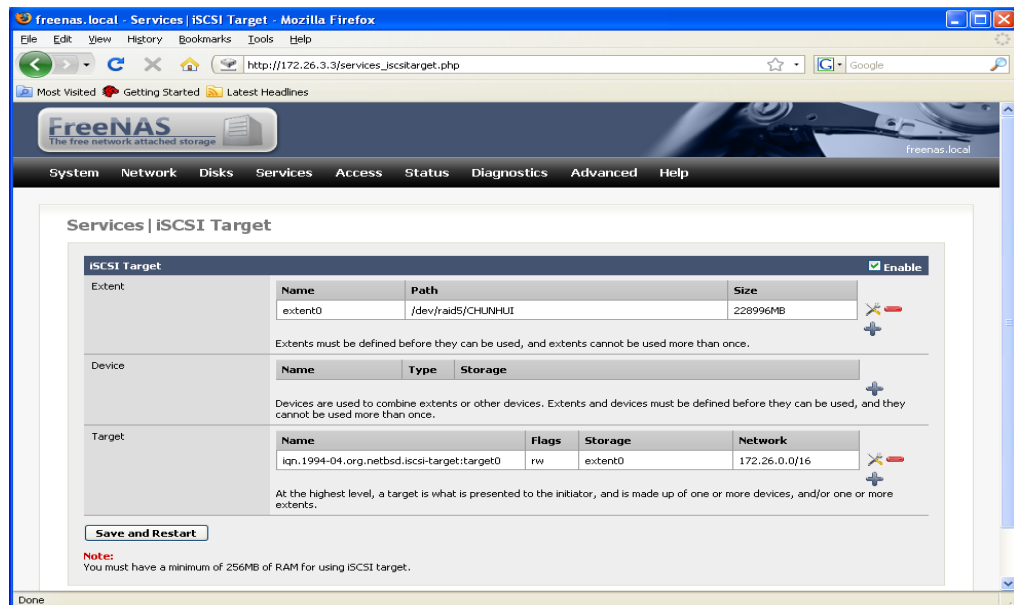


Fig. 14 RAID 5 Array of 228GB out of three drives



Fig. 15 RAID 5 Pack Complete

**Step 8** Create RAID Initiator in the front-end PC.

A Synaptic Package Manager can help to start the RAID Initiator for the iSCSI target on FreeNAS machines. Both versions of “iscsitarget-modules-2.6.26-1-6 2.6.26+0.4.16+svn” and “iscsitarget-modules-2.6.26-2-6 2.6.26+0.4.16+svn” were installed in my lab machine. If there is no GUI mode, then “apt-get” command can be used to install this program.

**Step 9** Establish the connection between the initiator and the target.

A. After the initiator software has been installed, it is necessary to start the software. The command “/etc/init.d/open-iscsi start” will initiate the open-iSCSI connection between the iSCSI target and initiator. And whether it is done or not, the result information will pop out:

```
Chunhui:~# /etc/init.d/open-iscsi start
Starting iSCSI initiator service: iscsid
iSCSI daemon already running
```

B. Then the targets from the back end port need to be discovered by the front-end:

```
Chunhui:~# iscsiadm -m discovery -t st -p 172.26.3.3
172.26.3.3:3260,1 iqn.1994-04.org.netbsd.iscsi-target:target0
Chunhui:~# iscsiadm -m node
172.26.3.3:3260,1 iqn.1994-04.org.netbsd.iscsi-target:target0
```

And all the three targets are already there connecting:

```
Chunhui:~# ls /etc/iscsi/send_targets
172.26.3.3,3260 172.26.3.4,3260 172.26.3.5,3260
```

From the kernel messages, it can be shown:

```
Chunhui:~# tail /var/log/messages
Mar 27 13:57:37 Chunhui kernel: [605813.381468] RAID5 conf printout:
Mar 27 13:57:37 Chunhui kernel: [605813.381473] --- rd:3 wd:2
Mar 27 13:57:37 Chunhui kernel: [605813.381477] disk 0, o:1, dev:sda1
Mar 27 13:57:37 Chunhui kernel: [605813.381479] disk 1, o:1, dev:sdb1
Mar 27 13:57:37 Chunhui kernel: [605813.381482] disk 2, o:1, dev:sdcl
```

**Step 10** The targets need to be partitioned and make a file system on them.

```
380 fdisk /dev/sdb
```

```

281 fdisk /dev/sda
382 fdisk /dev/sdc
389 mkfs -t ext3 /dev/sda1
390 mkfs -t ext3 /dev/sdb1
391 mkfs -t ext3 /dev/sdc1

```

**Step 11** Install the mdadm package.

The sda1 and sdb1 commands are used to create a RAID 0 package, and then the sdc1 command is just directly defined as a target for the remote customer PC. Firstly, we install the “mdadm” package on front-end PC:

```
416 apt-get install mdadm
```

**Step 12** Create RAID 0 out of two RAID 5 arrays.

```

#:mdadm --assemble /dev/md0 /dev/sda1 /dev/sdb1
#:mdadm --detail --scan:
ARRAY      /dev/md0      level=raid0      num-devices=2      metadata=00.90
UUID=640e551d:e3fe764a:fa9c896f:eca7b28e

```

**Step 13** Set the RAID 0 array as a target for the client PC.

The RAID 0 array md0 can be directly set as a target in the Debian system by redefining some configuration files.

The configuration file needs to be modified:

To show the contents of the file:

```
more /etc/mdadm.conf
```

To change it, using “vi” or “nano” commands:

```
nano /etc/mdadm.conf
```

**Step 14** Set the single array disk as a target for the client PC.

The single disk of RAID 5, here sdc1, can also be directly configured as a target to the initiator, similarly to the previous step. The target name is given differently as described in the previous step.

**Step 15** Set the single disk as a target for the client PC.



The single disk from FreeNAS which will be used to test the reading speed can be directly set as a target as well as RAID packs.

The target name is different from the above two.

**Step 16** Install initiator software on the client PC (Windows XP).

The Initiator software connects the client Window XP and the Debian operating system. Here, free Initiator software was downloaded from the Internet. The target PC IP address is needed to confirm the connection and there are two targets of RAID virtual disks and one single disk target from FreeNAS via the arrangement of Debian system. The version used here is “initiator-2.08-build3825-x86fre.exe”, a free Microsoft iSCSI Initiator downloaded from the MS website.

**Step 17** Initiate the target of RAID 0 and single disk of RAID 5.

First, change the configuration files, and then start the service:

```
nano /etc/ietd.conf
nano /etc/initiators.allow
/etc/init.d/iscsitarget start
```

**Step 18** Check the boot loader sequence if the system does not work properly as this is one of the reasons why the system cannot find the target.

In the booting procedure, “open-iSCSI” in the back-end should be initiated before the “mdadm” module starts, thus the devices in backend can be connected before the “mdadm” devices can be introduced to the client end, confirming the whole chain integrity.

The sequence is checked in /etc/rc2.d, to see if the order is right to establish the final target. If the “mdadm” module is already introduced to the Debian operating system behind the “open-iSCSI” module, then there is no need to make change. After the change,

**Step 19** Reboot the system.

Having completed the above steps the experiment environment is ready.

The new volume is connected to the customers' computer, but can not be used yet. The second disk in system is the new virtual disk that has just been created. Each PC has a 228 GB, RAID 5, three PCs form another RAID 5, then the total disk size is near 447 GB, see Figure 16.

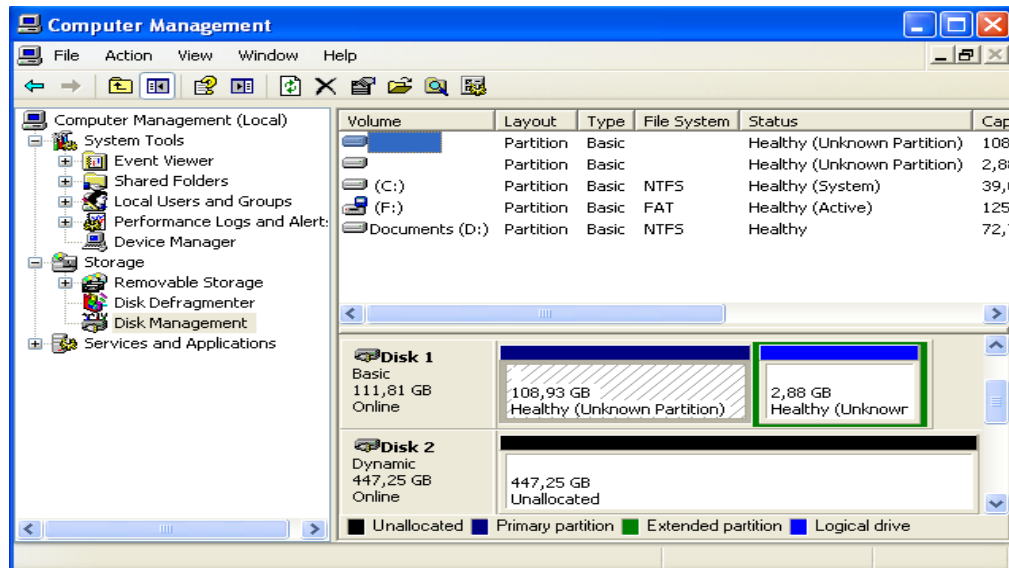


Fig. 16 New disk is connected (disk2 -- 447,25GB)

The new disk has to be partitioned first:

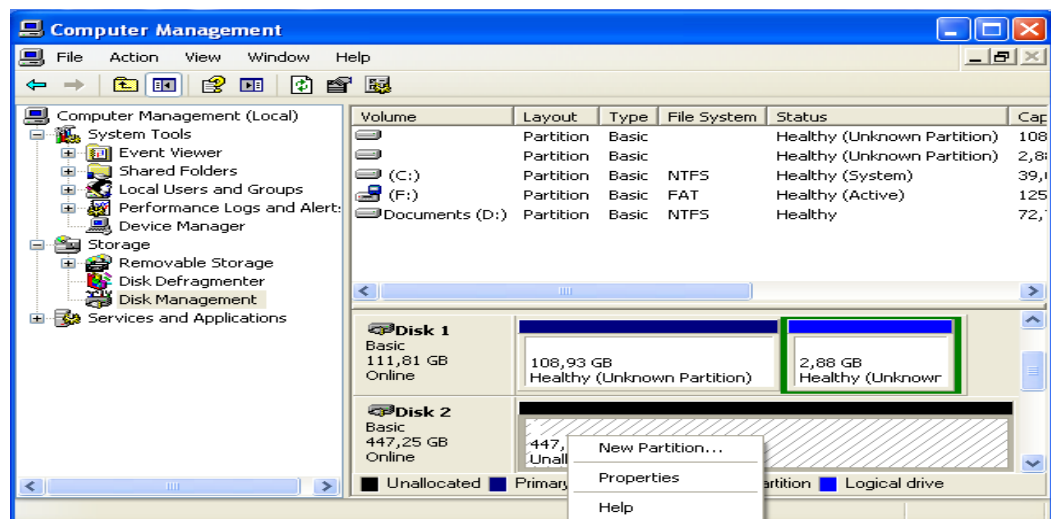


Fig. 17 Partitioning of new disk (disk 2)

And then the virtual disk (I) is created:

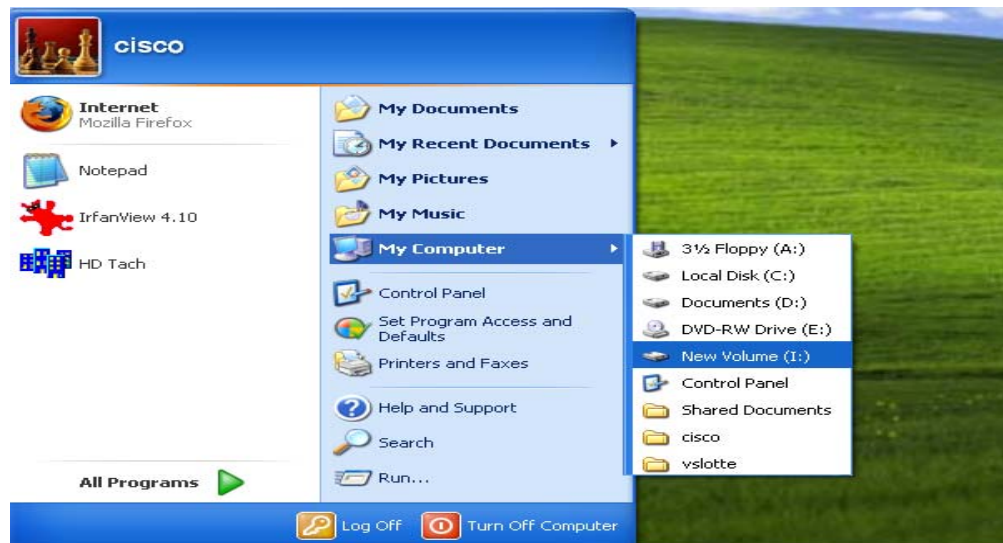


Fig. 18 New volume of virtual disk (I) created

And in the same way, Disk (J) is connected to the customers' computer as Disk I: Figure 19 shows that the remote disk has been successfully connected to the host computer, and the capacity is 447,25GB. There are two virtual disks now available to the customers.

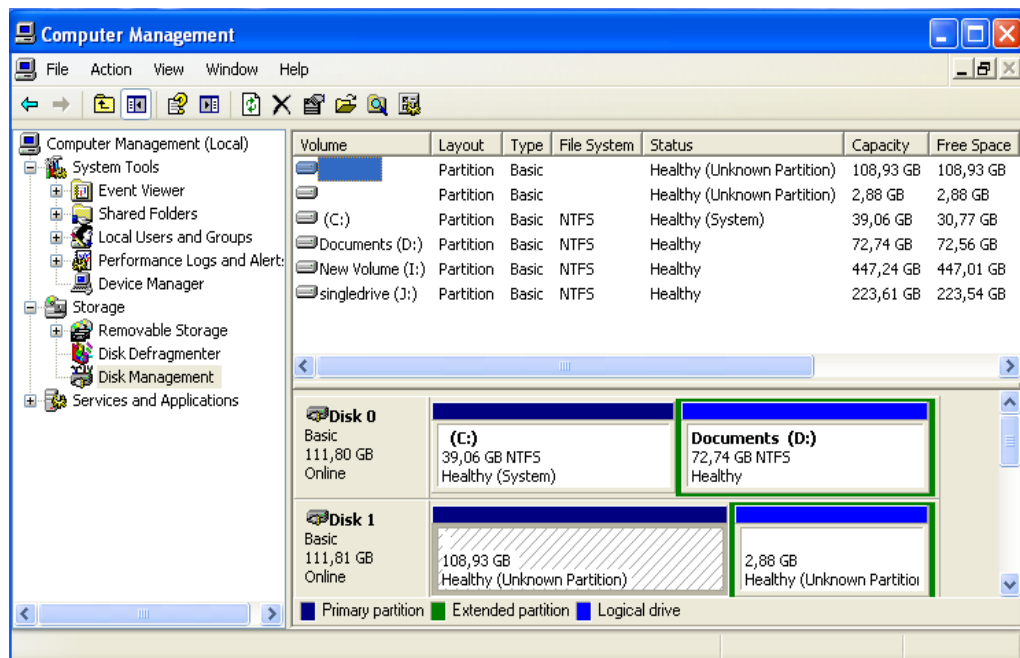


Fig. 19 Two virtual disks of (I) and (J) are available to use.

## 4.2 Experiment 1

Two FreeNAS PC disks (each formed of RAID5) form an md0 in this experiment, we ran quick bench testing on both the single virtual disk (RAID 5) and the md0 (RAID 0). The test tool was HD Tach downloaded free from the Internet.



Fig. 20 Quick bench testing

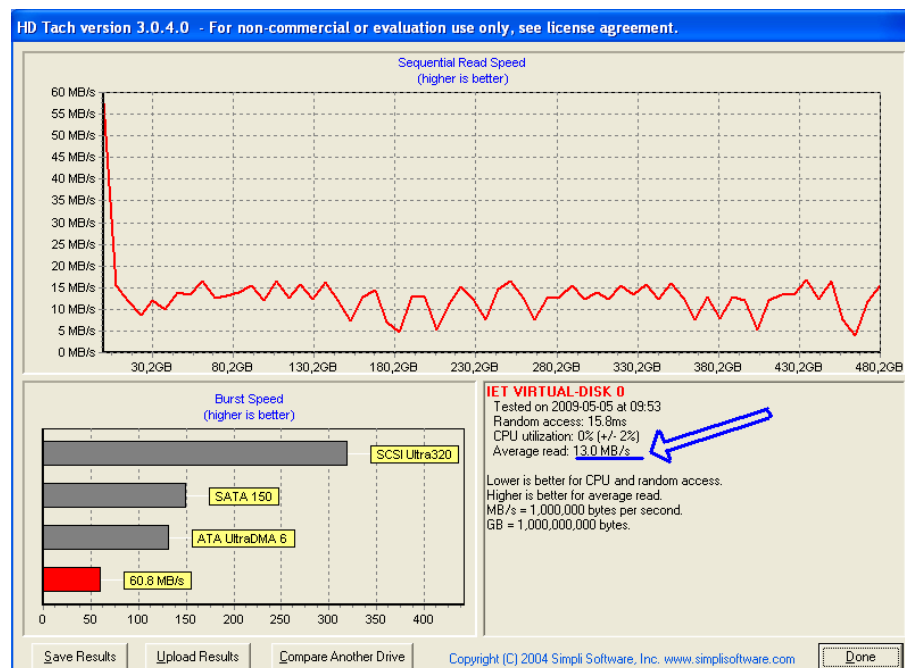


Fig. 21 The reading speed of Disk (I)

Figure 20 shows that the virtual disk I is selected for quick bench testing. The result of the quick bench testing shows that the average reading speed is 13.0MB/s (Fig.21).

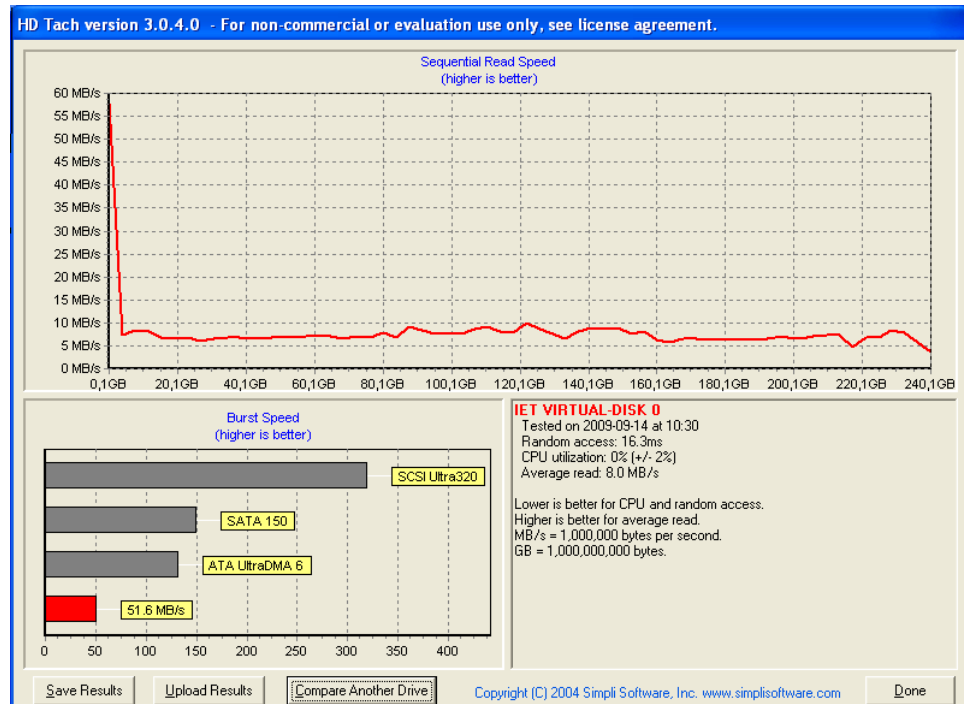


Fig. 22 The reading speed of Disk (J) of single RAID 5 pack



Fig. 23 Single disk (G) with no RAID is available.

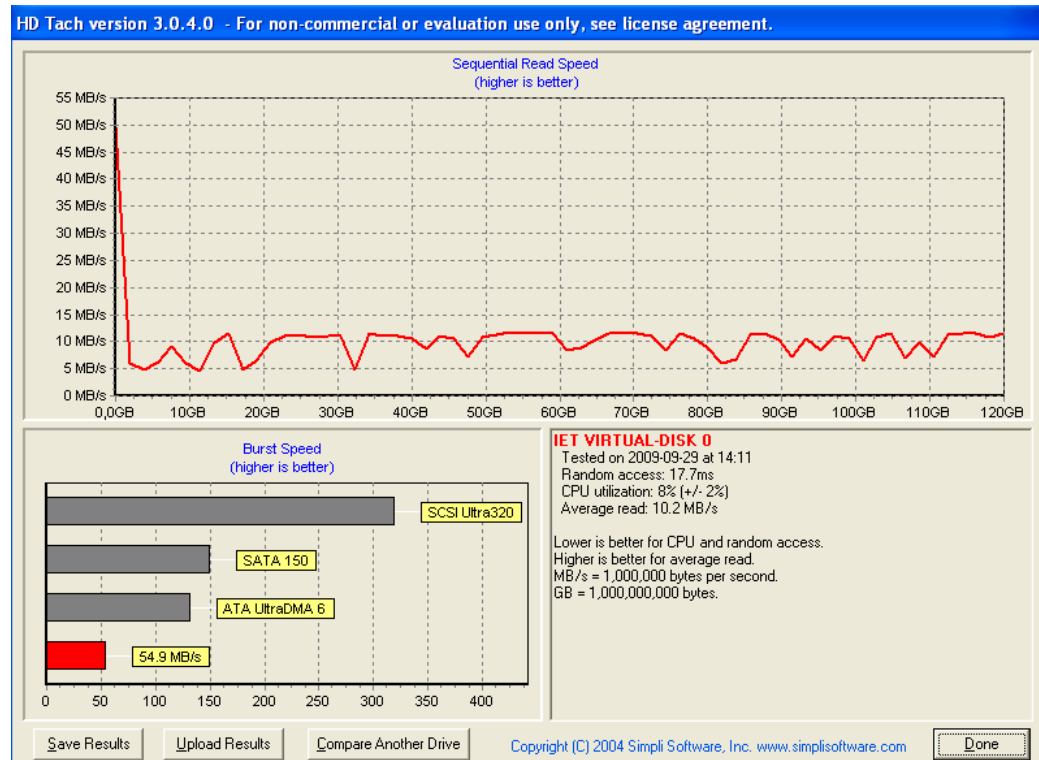


Fig. 24 The reading speed of a single disk (G) from back-end

### 4.3 The result analysis

The experiment shows that the RAID 0 of two RAID 5 has the read speed of 13.0MB/s(see Fig.21), and the read speed of a single RAID 5 is 8.0MB/s(see Fig. 22), while the single disk from the same back-end PC transfers at 10.2MB/s(see fig.24). RAID 0 has a better performance than a single disk. Since the disks inside RAID 0 have their own data channel, each one can operate independently. When reading, the system will select the disk which has shorter response time to process, thus a faster reading speed can be achieved than single disk can. In this experiment, RAID 0 consists of two lower parts of RAID 5. In addition, we can see that 62.5% increase of performance can be achieved. In theory, the increase should be 100%. The RAID 5 disk array performance is slower than the single disk transfer speed by about 27.5%. RAID 5 provides redundancy to the data storage, so the transfer speed is slower.

## 4.4 Experiment 2

In this experiment a CDROM is used as the testing data. The experiment is to copy this CDROM to md0/1, and then take away one of the disks from RAID 5 in one FreeNAS PC. How is the data now? Is the data lost or can it be found after some processing? The result should be that the lost data can be found after the RAID 5 package completes the rebuilding. Figure 24 shows copying a large size of file from the local PC to the virtual disk (I):

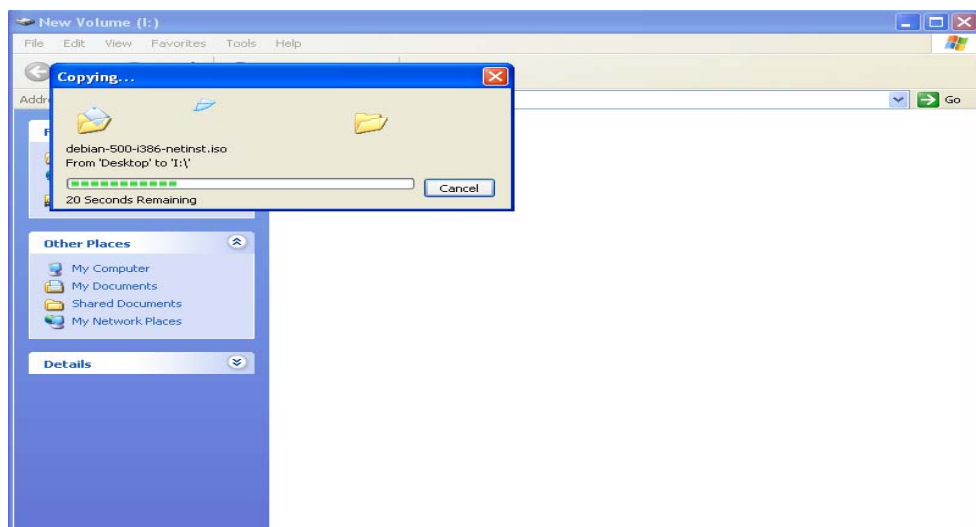


Fig. 25 Copying data

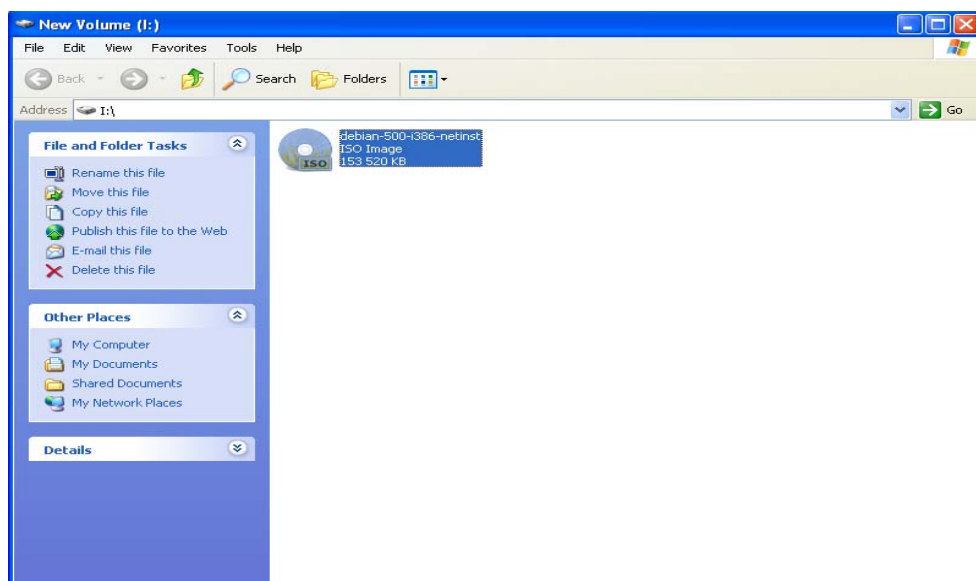


Fig. 26 The data is copied to disk (I).

Then I remove one of the RAID 5 pack disks (Fig 27):

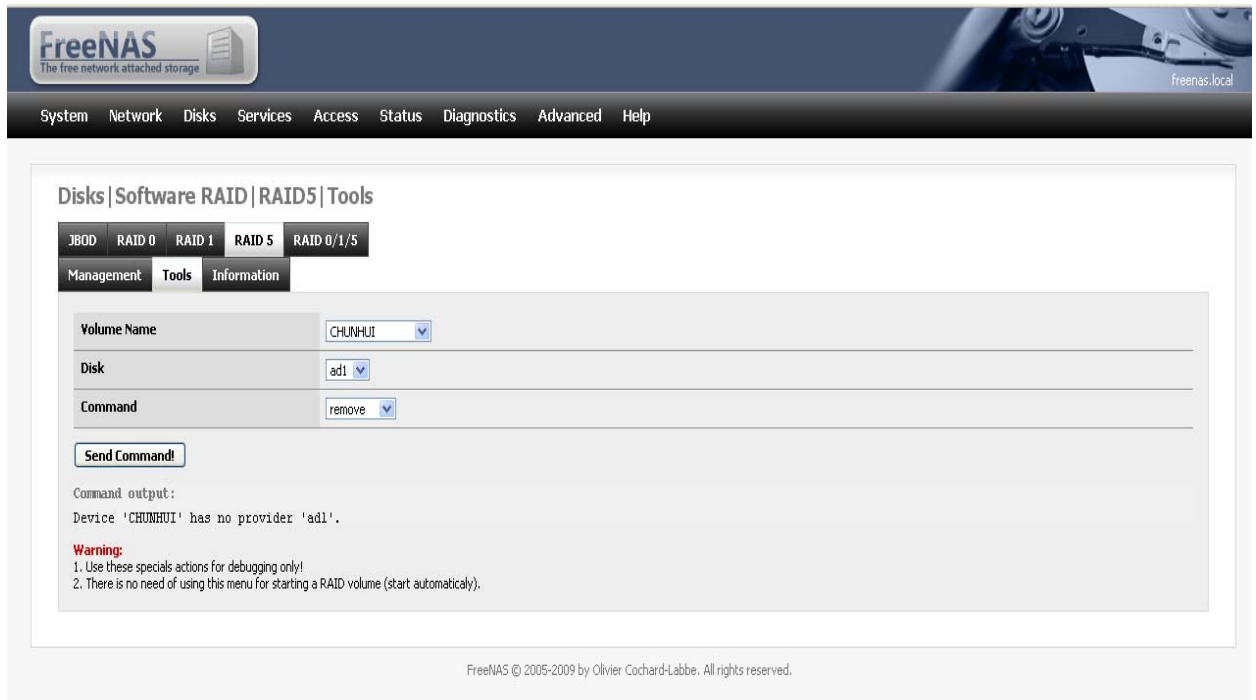


Fig. 27 Removing one disk from RAID 5

The RAID 5 pack is in degraded mode (Fig28)

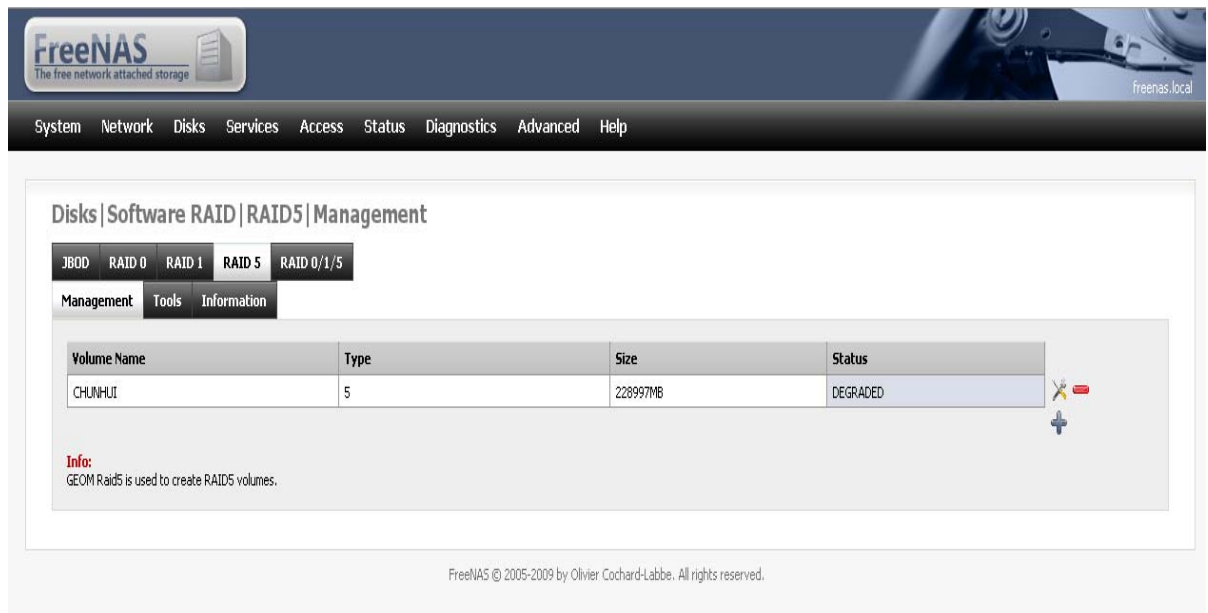


Fig. 28 RAID 5 in degraded mode

After the RAID 5 pack completes the rebuilding, the volume (I) is healthy (Fig29):



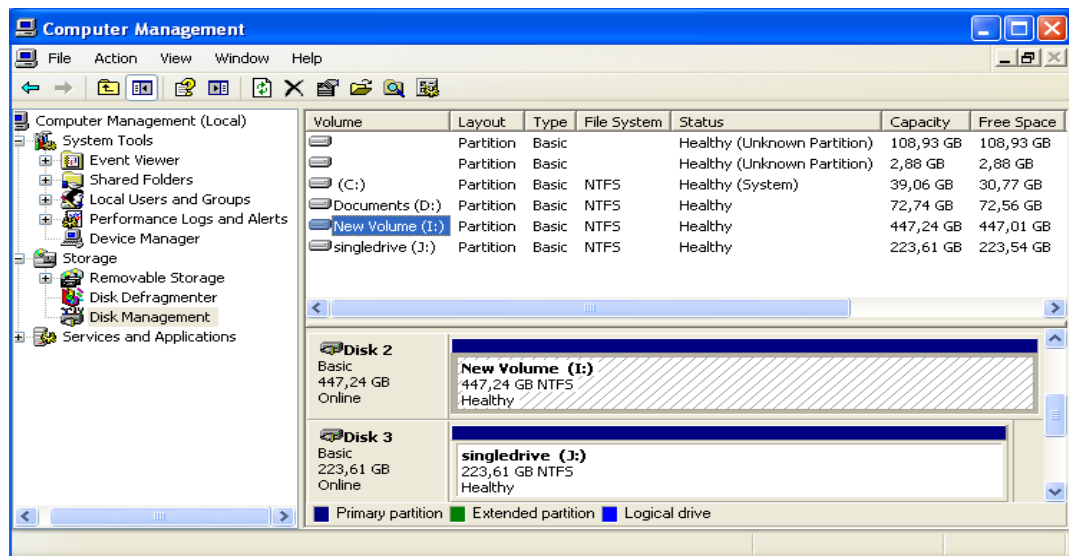


Fig. 29 The volume (I) is healthy after rebuilding.

And the lost file can be obtained again (Fig.30):

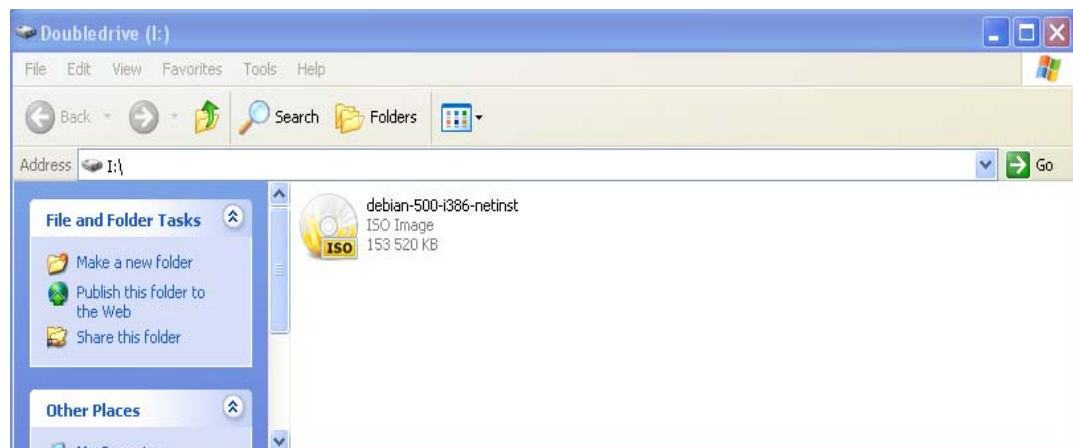


Fig. 30 The lost data is found.

Comparing the file after rebuilding the original files, we find out that the two files are completely the same:

*C:\Documents and Settings\cisco>*

*fc c:\debian-500-i386-netinst.iso I:\debian-500i386-netinst1.iso*

*Comparing files C:\debian-500-i386-netinst.iso and I:\DEBIAN-500-I386-NETINST1.ISO*

***FC: no differences encountered***

#### **4.5 The result analysis**

RAID 5 keeps redundant parity information inside the array. When one disk fails, the system runs in degraded mode. If users want to access the data on the failed disk, RAID 5 will read the data in the same stripe on other disks and the parity, and then calculate the data out. If a good disk is inserted and replaces the fail disk, RAID 5 starts rebuilding, recovers all the data and writes into the new disk. The result proves that RAID 5 can really tolerate one disk failure and keep data correct in degrade mode, thus provides high reliability and availability.

## 5 Conclusion

In this thesis, a design method to set up a Software RAID system is introduced. This kind of design is suitable for small companies that need a cheaper and massive capacity storage system. As shown in the design, there can be many servers on the front-end, which means more users can share the RAID system. Furthermore, at the back-end, many arrays can be connected. The system is scalable, when needed, users can add more arrays as long as the Ethernet switch can support them.

Via the switch, we establish a data backup connection between FreeNAS and the customer. Customers can use the remote hard drives to do the data management including reading, writing, and backup.

These experiments prove that RAID 0 has quite good reading performance that a single drive can provide and with RAID 5 arrangement, the system can tolerate one disk failure and the lost data can be recovered, after it has rebuilt the package.

The deficiency of this system is that the transfer speed is restrained. The reasons are:

1) One ATA channel can only support two disks, one disk has to be defined as the master and the other as the slave. ATA bus is not sharable which means the two disks can only run in serial not in parallel. 2) The speed of the switch interface also limits the system's whole speed. 3) Since the ATA bus is monopolized in the RAID 5 degraded mode, users can not use the virtual disk. It takes long time waiting for it to recover. Besides, ATA disks do not support hot swappable function.

If the whole system for some reason is turned off, the RAID package needs to be manually assembled although the data is safe.

## 6 References

- [1] D.A.Patterson, G.A.Gibson and R.Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In: ACM SIGMOD. *Proceedings of 1988 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM Press, 1988 : 109~116
- [2] R.H.Katz, G.A.Gibson and D.A.Patterson. Disk System Architecture for High Performance Computing. *Proc. of the IEEE*, 1989, 77(12): 1842~1858
- [3] Derek Vadaia *Managing Raid on Linux* O'Reilly & Associates First Edition 2002
- [4] P.M.Chen, E.K.Lee, G.A.Gibson, et al. RAID: High-Performance, Reliable Secondary Storage. *ACM Computing Surveys*, 1994, 26(2): 145~185
- [5] G.A.Gibson and D.A.Patterson. Designing Disk Arrays for High Data Reliability. *Journal of Parallel and Distributed Computing* , 1993 , 17(1) : 4~27

### On line references

- [6] <http://linux.yyz.us/why-software-raid.html> (23.Mar.2009)
- [7] <http://www.pcguide.com/ref/hdd/perf/raid/conf/ctrlHardware-c.html> (28.Oct.2009)
- [8] <http://linuxdevcenter.com> (27.Mar.2009)
- [9] <http://howtoforge.org> (20.Apr.2009)
- [10] <http://www.bestpricecomputers.co.uk/glossary/raid-0.htm> (16.Jul.2009)
- [11] <http://www.scsita.org> (28.Oct.2009)
- [12] <http://www.intel.com/products/processor/corei7/index.htm> (01.11.2009)
- [13] <http://www.mbnet.fi/uudet/uoitteet/?id=485> (01.11.2009)
- [14] [http://www.ugrad.cs.ubc.ca/~cs318/notes2005/Lect19b-WhitePaper-RAID-ACSP\\_RAID\\_Ch4-328-04.pdf](http://www.ugrad.cs.ubc.ca/~cs318/notes2005/Lect19b-WhitePaper-RAID-ACSP_RAID_Ch4-328-04.pdf) (26.Oct.2009)
- [15] <http://storageconference.org/2003/presentations/C01-Meth.pdf> (28.Oct.2009)

[16]

[http://www.enterprisestorageforum.com/technology/features/article.php/11192\\_137894\\_1\\_2](http://www.enterprisestorageforum.com/technology/features/article.php/11192_137894_1_2) (29.Oct.2009)

[17] <http://www.sharpened.net/glossary/definition.php?sata> (30.Oct.2009)

[18] <http://h10010.www1.hp.com/wwpc/uk/en/sm/WF06b/12169-304616-241493-241493-241493-3971478-3882370.html> (02.Nov.2009)

[19] <http://www.freenas.org/downloads/docs/user-docs/FreeNAS-SUG.pdf>