Performance Analysis of RAIDs in Storage Area Network

Sneha M.
Assistant Professor, Department of Computer Science and Engineering,
R V College of Engineering Bengaluru-560059

ABSTRACT

Direct Attached Storage, Network Attached Storage and Storage Area Network are the different kind of networking and storage facilities used in different domains from small scale industries to large scale industries. Storage area network is a disk system concentrated on network which allows clients or servers to use the disk system as a local disk. The speed of transmission of data to and from these systems is a crucial element in measuring the availability and reliability of the services provided by the industry. The redundancy of data is very important to provide high data availability; this can be achieved by having RAID (Redundant Array of Inexpensive Disks) methods. This paper describes the performance analysis on RAID levels 0,1,5,6 and10 through IOR benchmark tests on SAN which is implemented using ATA(Advanced Technology Attachment) over Ethernet protocol AoE(ATA over Ethernet). All these tests were performed on eight, 2-TB SAS (Serial Attached Storage) hard drives.

General Terms

Storage Area Network.

Keywords

SAN, RAID, ATA, IOR tests, SAS drives.

1. INTRODUCTION

RAID management tools are used to configure storage on Storage Area Network. They serve as an interface between the client machine and the server which gives the administrator the liberty of accessing the storage on the server and deciding on how the data can be stored in a secure and an efficient manner. This project aims at developing a generic tool for managing the storage on the server using the concept of RAID. The project also aims to carry out the performance analysis of the storage using the interface mentioned above.

The security, accessibility and reliability are very important since storage system has become a main concern in today's industry [1]. Any business values its data as one of its most important asset. Thus, managing storage and ensuring efficient and fault tolerant distribution of data over the storage has become of paramount importance. Thus, the interface developed for managing the storage using RAID can be a revolutionary approach in the industry because of its generic nature.

The whole idea of the project is to develop a single interface that can cater to the needs of establishing a Secure Shell connection, managing the storage of the server and carrying out performance analysis all complying to the standards of the IOR benchmark test [2].

1.1 Storage Area Network

SAN stands for Storage Area Network. A storage area network is a specialized high-speed network that enables fast, cheap and reliable access among servers and external or independent storage resources. The primary purpose of Storage Area Network is to enable storage devices to communicate with computer systems and with each other [3]. It provides block-level storage that can be accessed by the applications running on any networked servers. SAN storage devices can disk-based devices, like RAID hardware. SANs are particularly helpful in failure-backup and disaster recovery settings. Within a SAN, data can be transferred from one storage to another without interacting with a server. This speeds up the backup process and eliminates the need to use server CPU cycles for backup.

Recognized as a high performing technology, SAN is considered as one of the best choices for storing large data because of reliability, security and availability provided by it [1]. The backup software is connected to the storage infrastructure provided by the storage vendors through the SAN. Also, many SANs use Fiber channel technology or other networking protocols that allow the networks to span longer distances geographically. That makes it more feasible for companies to keep their backup data in remote locations [4].

The diagram in Figure 1[3] shows an overview of a SAN that connects multiple servers to multiple storage systems, all of which are interconnected by networks and LANs. The clients having different operating systems are connected to a network or LAN. The network is further connected to Storage Area Network. When the client sends the data to be stored through the network, it is distributed on the RAID levels configured on the storage in SAN with the help of the controller.

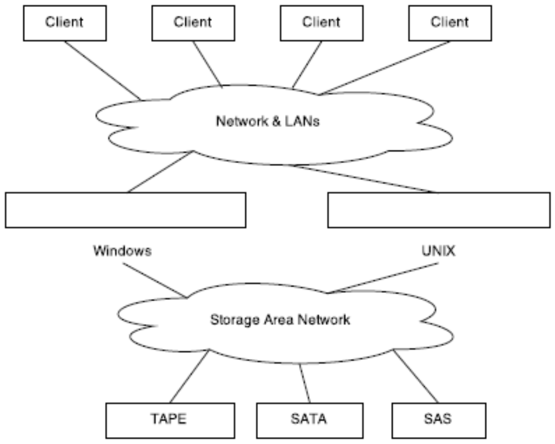


Figure 1: Storage Area Network Architecture

1.2 RAID (Redundant Array of inexpensive Disks)

Please RAID (redundant array of inexpensive disks) is a data storage technology that integrates multiple disk drive components into a logical unit for the purposes of data redundancy or performance improvement [5]. Some of the RAID configurations are described below:

RAID 0 (striping): RAID 0 consists of striping without mirroring or parity. The capacity of a RAID 0 volume is the total capacities of the disks in the set, the same as with a spanned volume. There is no added redundancy for risking disk failures, just as with a spanned volume. Failure of one disk causes the loss of the entire RAID 0 volumes, with reduced possibilities of data recovery when compared to a broken spanned volume [1].

There are few advantages of RAID 0. It offers great performance, both in read and write operations. There is no overhead caused by parity controls. Entire storage capacity is used without any overhead and it’s easy to implement. In addition to this there is a disadvantage i.e. zero tolerance against drive failures. Therefore, it should not be used for mission critical systems [6].

Figure 2 depicts the schematic drawing of RAID 0[1]. The diagram clearly states that the data is distributed in form of blocks which are stored on the drives without any redundancy. It is used for non critical data storage that has to be read/written at high speeds, such as image retouching or video editing systems.

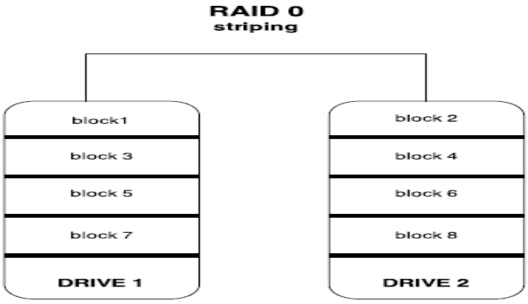


Figure 2. Schematic Diagram of RAID 0

RAID 1(mirroring): This configuration consists of data mirroring, without parity or striping. Same data is written to two (or more) drives, thereby producing a mirrored set of drives. The array continues to operate as long as at least one drive is functioning. If a drive fails, the controller uses either the data drive or the mirror drive for data recovery and continues operation. At least two drives are needed for RAID 1[1].

The advantages of having RAID 1 is that, it offers efficient read speed and write speed that can be compared to that of a single drive. In caseof failure, data does not have to be rebuilt, they just have to be copied to the replacement drive. In addition to all this it’s a very simple technology to implement. The main disadvantage of RAID is the effective storage capacity which drops to half of the total drive capacity because all data get written twice. Software RAID 1 solution do not always allows redundancy of a failed drive i.e. it cannot be replaced while the server keeps running[6].

Figure 3 shows the schematic diagram of RAID 1[1]. The diagram clearly states that the data is stored in form of blocks in the storage and a copy of each block is maintained to provide the redundancy. It is used for mission critical storage, such as accounting systems. It is also applicable for smaller servers in which only two data drives are to be used.

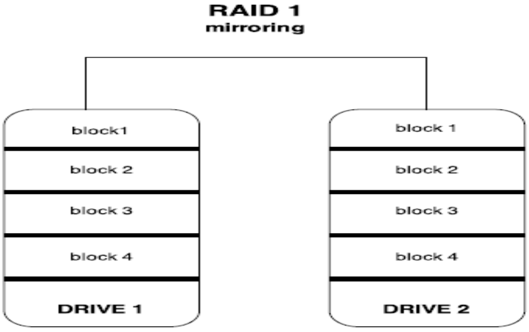


Figure 3: Schematic Diagram of RAID 1

RAID 5(Striping with Parity): It consists of block-level striping with distributed parity. It requires that all drives but one be present to operate. Upon failure of a single drive, subsequent reads can be calculated from the distributed parity such that no data is lost. RAID 5 requires at least three disks. RAID 5 is the most common secure RAID level. The parity data are not written to a fixed drive, they are spread across all drives. Using the parity data the computer can recalculate the data of one of the other data blocks, if those data blocks are unavailable. That means RAID 5 can withstand a single drive failure without losing data or access to data. Often extra cache memory is used on these controllers to improve the write performance [1].

Configuring RAID 5 can be advantageous. Read data transaction are extremely fast while write data transactions are relatively slower due to parity which has to be calculated. If one drive fails, still all data can be accessed, even while the failed drive is being replaced and the storage controller rebuilds the data on the new drive. Since this is a complex technology, if one of the disks in an array fails and is replaced, restoring the data may take a day or longer depending on the load on the array and the speed of the controller. If another disk goes bad during that time, data are lost forever.

Figure 4 shows the schematic diagram of RAID 5[4]. The introduction of parity block adds to the fault tolerance if any of the data block go missing. It is an overall good system that combines efficient storage with excellent security and decent performance. It is ideal for file and application servers that have a limited number of data drives.

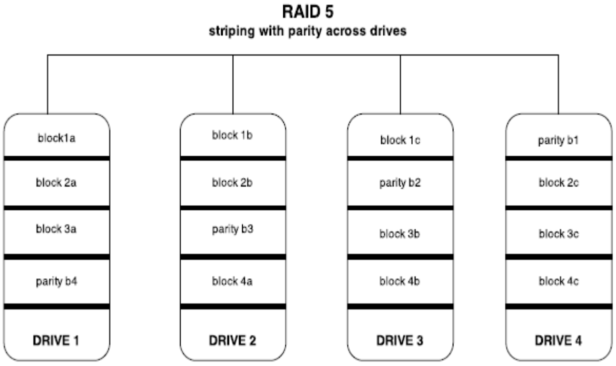


Figure 4: Schematic Diagram of RAID 5

RAID 6 (Striping with double Parity): It consists of block-level striping with double distributed parity [1]. RAID 6 is like RAID 5 but it writes the parity data to two drives. That means it requires at least four drives and can withstand two drive failures simultaneously. This makes larger RAID groups more practical, especially for high-availability systems, as large-capacity drives take longer to restore. As with RAID 5, a single drive failure results in reduced performance of the entire array until the failed drive has been replaced. With a RAID 6 array, using drives from multiple sources and manufacturers, it is possible to mitigate most of the problems associated with RAID 5[4].

Like the RAID 5, read transactions are very fast. If two drives fail, the user still has access to all data, even while the failed drives are being replaced. So RAID 6 is more secure than RAID 5. Drive failures have an effect on throughput, still it is acceptable. As compared to RAID 5 write data transactions are slowed down due to the parity that has to be calculated. Presence of a double parity makes it a complex technology; therefore, rebuilding a drive can take a long time.

Figure 5 shows the schematic diagram of RAID 6. It can be noted that it uses double parity and thus provides better redundancy than RAID 5. RAID 6 is an overall good system that combines efficient storage with excellent security and decent performance. It is preferable over RAID 5 in file and application servers that use many large drives for data storage.

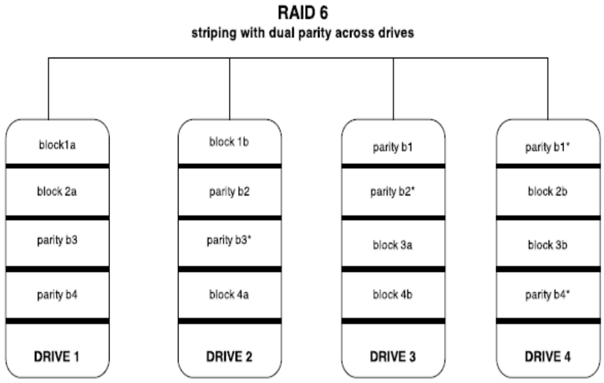


Figure 5: Schematic diagram of RAID 5

RAID 10 (Combining mirroring and striping): It is possible to integrate the advantages and disadvantages of RAID 0 and RAID 1 in one single system. This is a hybrid or nested RAID configuration. It provides security by mirroring all data on secondary drives while using striping across each set of drives to speed up data transfers [6].

The major advantage of having RAID10 is, if something goes wrong with one of the disks in a RAID 10 configuration, the rebuild time is very fast since all that is needed is copying all the data from the surviving mirror to a new drive. This can take as little as 30 minutes for drive of 1 TB. This also comes with a trade-off, since the concept of mirroring is being used, half of the storage capacity goes to mirroring, so compared to large RAID 5 or RAID 6 arrays, this is not a cost-effective way to have redundancy.

The diagram in figure 6 shows the schematic architecture of RAID 10. It can be seen as a combination of two already available RAIDs i.e. RAID 1, RAID 0. The data first gets stripped and then mirrored.

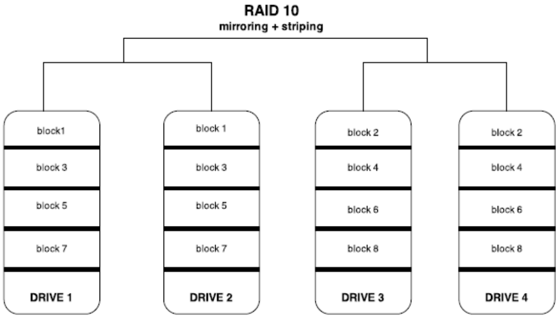


Figure 6: Schematic diagram of RAID 10

The diagram in Figure 7 shows how the RAID arrays work in tandem with Storage Area Network, along with different type of operating systems used to configure the RAID. The client accesses the Storage Area Network through protocols such as Fibre Channel Protocol or iSCSI and starts a session with it. The storage which is present on the SAN is managed by either hardware or software RAID controllers present on the server, which configures RAID levels on the storage and is also thus responsible for the data distribution on these disks.

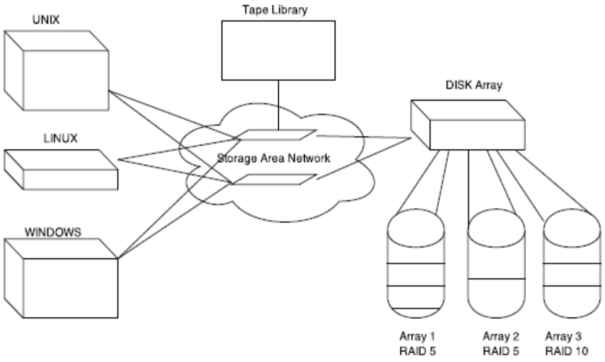


Figure 7: Architecture of SAN comprising of RAID

2. IMPLEMENTATION MODEL

The whole system is designed in Python and its variants. The user interface is coded in Python along with the basic modules. The command line interface is linked to a front end, coded in python which is presented to the user. Pycharm is a software development kit that supports Python for implementing standards and has a strong base for scripting and designing. Figure 8 shows the system architecture of the interface developed.

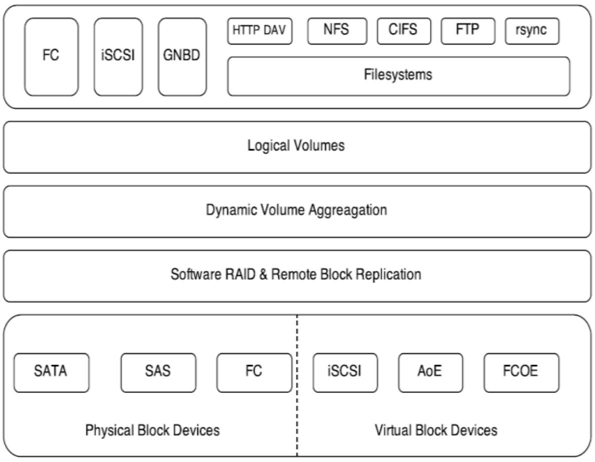


Figure 8: System Architecture of the interface

The client must establish a SSH connection with the SAN server at the storage array. The layer can connect over simple transfer protocols to the server and client can configure port numbers or IP addresses through which the communication can take place. The Linux based operating system Centos 6.5 is a centric part of the environment of storage optimized operating system. Block devices referred to here are any disks or raw volumes. Block devices can be directly attached devices to the host machine or remotely attached block devices which can be designated as virtual block devices. SATA, SAS, SCSI, FC disks are the various supported physical block devices. RAID volumes implemented in this project also fall under the designation of physical block devices. The upper layers of this architecture refer to these remote block storage devices. The remote block devices accessed using the protocols iSCSI, AOE or FCOE protocols are referred to as virtual block devices. There is no difference between physical or virtual block devices according to the upper layers of Dynamic Volume Aggregation, software RAID and remote block replication layer.

Block devices mentioned in the lower layers can be aggregated using the protocols used to configure RAID by the software RAID management layer. The replication done at the block level is done over TCP/IP protocol which is secured by the SSH established on the network.

The software RAID facilitates the user to manage the aggregation of various physical and virtual block devices. This increases the usable capacity, performance of block I/O and also provides multi fold availability and reliability. The various RAID configurations possible using the interface developed are RAID 0, RAID 1, RAID 6 and RAID (1+0).

The RAIDs configured are grouped together into storage pools by the dynamic volume aggregation. These storage pools are grouped into various logical volumes which are the ones that are virtually and physically visible to the administrator or the user.

These volumes that are visible to the administrator can support multiple ODF (on-disk File System). The administrator after creating RAIDs can apply file system to it that readily satisfies the needs of specific applications and is optimal for carrying out performance analysis. The file system used in this project is the ext4 file system. Apart from this, XFS and ext3 can also be used. These are referred to as journaled file system which has a greater data security and also reduces regular file system check needs. Any data which is written to these disks is first logged into log files which here are referred to as journals which are shown in the Figure 9, before the data being written is finalized. This is how data consistency is taken care of, if a system gets crashed or there is a sudden power outage.

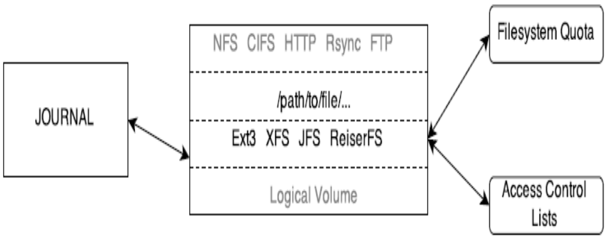


Figure 9: logging of the data into journals

3. EXPERIMENTAL ANALYSIS AND RESULTS

3.1 Evaluation Metric

The generic interface was developed in this project with the main intention of automating the process of RAID configuration and thus performing the IOR benchmark test on it. IOR is used for testing performance of parallel file systems using various interfaces and access patterns. The necessary parameters to be considered while performing the IOR tests were the BLOCK SIZE and TRANSFER SIZE, which were to be provided as parameters. It is necessary for the block size to be a multiple of the transfer size because the chunk size transferred at once is the transfer size.

3.2 Experimental Dataset

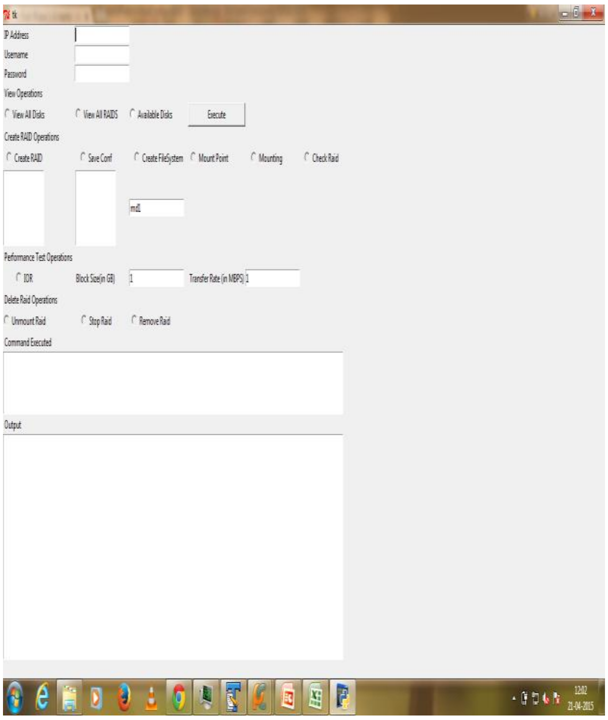
The dataset on which the experiment was conducted consisted of eight 2 TB SATA hard drives. While configuring the RAID, after the RAID is created it needs to be mounted on the system. For mounting the RAID, the file system of the configured RAID needs to be specified because each and every mounted hard drive needs to have a file system to create, manage and allow the reading and writing operations of various files to be performed on it. Mounting is necessary because it creates a specific mount point which serves as location of test.dat file which is read while performing the IOR benchmark test. This test.dat file serves as a major data set for the experimental analysis. This file is a binary file which is written to and read from the RAID level configured, and is analyzed for providing performance results of various RAIDS. The data obtained after performing the tests were further stored in various log files on the server, which served as further dataset for the final experimental analysis that needs to be performed.

3.3 Performance Analysis

IOR can be used for testing performance of parallel file systems using various interfaces and access patterns. IOR uses MPI (Message Passing Interface) for process synchronization.

All RAID levels were configured on two 2TB Serial ATA 7200rpm enterprise edition 6 gbps drives. After mounting the RAID, IOR tests were performed with different values for the parameters Block Size and Transfer Rate.

The Figure 10 shows the GUI of the generic interface developed. Whenever the user wants to perform some operation he should select the option and click on “Execute” button.



tk

IP Address

Username

Password

View Operations

○ View All Disks ○ View All RAIDS ○ Available Disks

Create RAID Operations

○ Create RAID ○ Save Conf ○ Create FileSystem ○ Mount Point ○ Mounting ○ Check Raid

md1

Performance Test Operations

○ IOR Block Size(in GB) 1 Transfer Rate (in MBPS) 1

Delete Raid Operations

○ Unmount Raid ○ Stop Raid ○ Remove Raid

Command Executed

Output

Figure 10: Snapshot of User GUI

After performing the IOR tests on different RAID levels with different block sizes and transfer rates the results are tabulated as shown in Table 1. The results show the average maximum read and write speeds for each of the RAID levels.

Table 1. Results of Performance Analysis

	Avg. Max Write	Avg. Max Read
RAID 0	1909.0836 Mbps	7061.8156 Mbps
RAID 1	1753.4112 Mbps	6182.516 Mbps
RAID 5	1721.0532 Mbps	6064.0756 Mbps
RAID 6	1622.1784 Mbps	5650.4436 Mbps
RAID (1+0)	1717.069565 Mbps	5939.504783 Mbps
Optimum: **RAID 0**	1909.0836 Mbps	7061.8156 Mbps

After the various IOR benchmark tests performed, it was observed that the most efficient RAID level was RAID 0 with the maximum reading and writing values. This was also verified by the graphs for MAX. WRITE and MAX. READ in Figure 11, values which were finally obtained after the analysis of the IOR benchmark tests. Another anomaly that was noticed by running these test on 7200rpm hard drives, which was the RAID 5 maximum writing speed was greater than the max write speed of RAID (1+0), when it is generally lower, which is because the number of drives for parity in RAID 5 is lower than in RAID (1+0) and is also dependent upon the rpms of the hard drives.

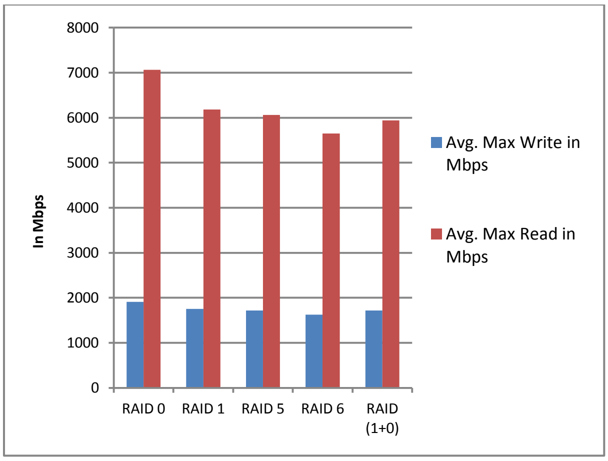


Figure 11: Graph for MAX. WRITE and Max. READ

4. CONCLUSION AND FUTURE WORK

The goal of the project is to develop a generic interface to facilitate the interaction between the SAN server and a host machine. This interface is unique in a way because most of the interfaces developed by the companies like IBM, HP, etc do not allow external agents to access the hard drives. These interfaces only accept the proprietary commands and do not allow the user to bypass the controller. The interface mentioned in this project allows user to access the drives irrespective of the device drivers using the open source commands and allows them to manage it abstracting the background details. The RAIDs are configured on the SAN and performance testing is done using benchmark tools.

The connection between the client and the server was established using a SSH connection. This was achieved using Paramiko which is a python implementation of the SSHv2 protocol. The RAIDs were configured using the mdadm commands embedded in the python script. These commands interacted with the interface to configure the storage on the server. RAIDs configured were tested for performance using IOR benchmark test. The commands and parameters for performing this test were embedded in the python script which interacted with the interface developed and thus performed the required tests. Once all the results are compiled together, analysis is done to find the most efficient RAID level.

As future work one can work on the following concepts:

1. Currently this project involves using Software RAID controller which can be replaced by a Hardware RAID controller for better performance clubbed with greater rpm hard drives.
2. Using storage which has an electronic mechanism such as pen drives instead of drives which have a mechanical spindle; the performance can be increased considerably.
3. The integrity of data stored in the drives should be maintained even if all the drives fail. This includes the security of data, backing up the data before changing the RAID levels.

5. REFERENCES

[1] TodyAriefianto W, LeannaVidyaYovita, 2014. DidinOlviovitha P, Performance Analysis of AoE-SAN Using Bonding Interface over RAID, 2nd International Conference on Information and Communication Technology (ICoICT), Indonesia.

[2] Hongzhang Shan, John Shalf, 2008. Using IOR to Analyze the I/O performance for HPC Platform. ACM/IEEE Conference on Supercomputing, Hong Kong.

[3] Weijun Xiao, Yinan Liu, Qing Yang, Jin Ren, ChangsenXie, 2006. Implementation and Performance Evaluation of Two Snapshot Methods on iSCSI, in Proceedings of 14th NASA Goddard/23rd IEEE Conference on Mass Storage Systems and Technologies (MSST '06), Seoul.

[4] Jae-Chang Namgoong and Chan-IkPark, 2001. Design and Implementation of a Fibre Channel Network Driver for SAN-Attached RAID Controllers. IEEE Conference on Information and communication Technology, Manchester.

[5] Jin hai, Wang hai, 2014. Research of the Router Scheme for Virtual Storage, 7thInternational Conference on Intelligent Computation Technology and Automation, China.

[6] Liu Xiao-Guang, Wang Gang and Liu Jing, 2002. A research on multi-level Networked RAID Based On Cluster Architecture, 5th IEEE International Conference on Algorithm and Architectures for Parallel and Processing, London.