International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 2– No.2, September 2013 – www.ijais.org

File App Data Protection via NFS

Latesh Kumar K J
Research Scholar
AEU University
Kaulalampur, Malaysia

Shwetha C S
Department of CSE
SIT, Karnataka
India

Reshma M
Department of CSE
SIT, Karnataka
India

ABSTRACT
This paper introduces a new method for implementing File RAID for text and programmable files on FAT, NTFS and Ext3 file system. This approach is currently involving two techniques FileSnapImitate (FSI) and FileSnapStripe (FSS) hash and indexing Parity calculation (HIPC) is anticipated to trim down the standard classic raid techniques over general file systems like, FAT, NTFS and Ext2. This paper introduces and proposes FILERAID, software which can achieve significantly higher WRITE performance and can also recover data of files.

Keywords
RAID, FSM, FSS

1. INTRODUCTION
In this approach the foremost advantage of using an Application Level File RAID (ALFR) is that it increases the performance and reliability of the system. The ALFR application is a credible example that could be used on a desktop or server with or without a storage connected to it. In general a RAID activity will happen if it is enabled on the volume, partition or a disk drive, if not RAID will not be active and assures a data guarantee on the environment.

The ALFR is the new method where in doesn't requires a volume/partition/diskdrive should be under RAID filesystem to guarantee the data being posted on that disk/volume/partition by user, this just a tool needs to be started before the process is started, and then user can go on, without any worries of being data lost, the smart tool will take care of handling data being posted by user/s on the environment.

2. IMPLEMENTATION OF FILE SNAP STRIPE
The figure shows the implementation of application level file raid – File Snap Stripe. This approach is quietly unique compared to native RAID functionality, the smart file raid controller is the intermediate between the Source File(s) created by user and the StripeStore(S) on the internal disks of server/workstation. When files are started by user over on internal disk the smart file raid is triggered on for a small scale activity of highly available application. Any Workstation will communicate via a protocol called NFS to storage appliance whereas the Windows workstations will communicate using CIFS (if any). The content being supplied by user/s to the file handle is now controlled by the Application File Raid system (Smart Raid Controller) and they are being processed by this and also pushed to its vault volume which is firmly sitting on one of the Network File system shares.

The performance is much highlighting and increases a lot when the disk stripping is done. The performance increases a lot by the content stripping? This is actually done by the interleaving of the bytes or the group of bytes. The interleaving of this sort is done across the multiple files. By this procedure only one disk is reading or writing the data. The reading and writing of the data are done in a simultaneous process.

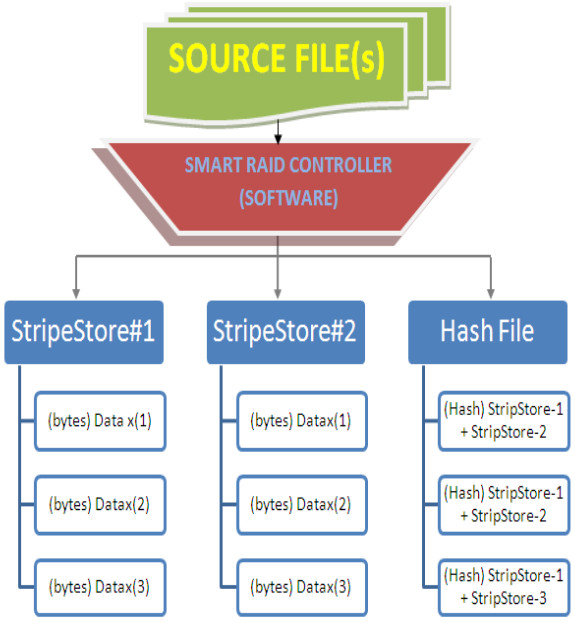


Fig 1: Application Level File Raid – File Snap Stripe via NFS

Striping is the concept of hooping the data without parity or mirroring. Here striping of data is done across two files. When user activates RAID0 service, RAID0 configuration file seeks two file where striped data will be be stored namely "a.$$$" and "b.$$$". When user chooses the option of building RAID0, striping of source file across a.$$$ and b.$$$ get started. Striping of data can be done based on 'line by line' or 'space by space' across two files. A reference file is maintained called "ref.txt" which contains the information regarding the striped data of source file and the file name where it stored.

Whenever source file gets corrupted, one of the file along with the reference file is used to recover data and it will be

done. In general RAID0 is successfully done it results in destination file which is a block level content accessed by any network file access protocol (NFS, CIFS) which helps in extracting the striped data successfully.

3. IMPLEMENTATION OF FILE SNAP IMITATE

The figure [2] shows the File Snap Imitate method for our research case on File RAID application for data protection across a business requirement. In this approach whenever a client connects to server and starts of file delegation (Create) automatically the file snap imitate gets activated and then it will start byte by byte file copying to scattered remote file system with a file name along with index, and this file is scattered and copied onto multiple hosts with the metadata of the file no different client via network file system protocol, so that any time when the source file gets corrupted, user or client can raise a request to recover his file data, when it is done so the data will be recovered by looking up to all the hosts connected across the clients to the node along with indexing the metadata.

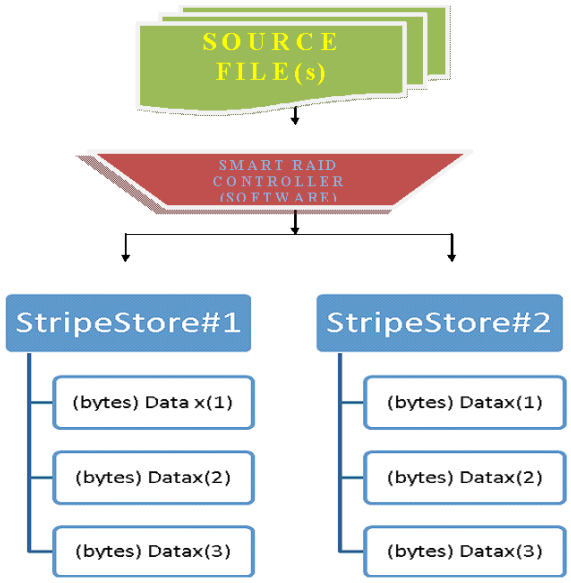


Fig 2: Application Level File Raid – File Snap Imitate via NFS

Mirroring is the concept of duplicating the file. When a user activates RAID1 service, an exact copy of the source file is created. The destination file name and its path are accepted by user as per his convenience. Once the copy of source file is over it asks the user whether to save the destination file or not. This option is provided to user since he may not need the back up of each file. So only when the user chooses the option of "save", the contents of source file gets saved in the destination file.

4. WHY AND HOW NFS FOR DATA PROTECTION

NFS operates as a typical client server application. The server receives remote-procedure-call (RPC) requests from its various clients. An RPC operates much like a local procedure call: The client makes a procedure call, then waits for the result while the procedure executes. For a remote procedure call, the parameters must be marshaled together into a message. Marshaling includes replacing pointers by the data to which they point and converting binary data to the canonical network byte order. The message is then sent to the server, where it is unmarshalled (separated out into its original pieces) and processed as a local file system operation.

The result must be similarly marshaled and sent back to the client. The client splits up the result and returns that result to the calling process as though the result were being returned from a local procedure call. The NFS protocol uses the Sun's RPC and external data representation (XDR) protocols. Although the kernel implementation is done by hand to get maximum performance, the user-level daemons described later in this section use Sun's public-domain RPC and XDR libraries.

The NFS protocol can run over any available stream- or datagram-oriented protocol. Common choices are the TCP stream protocol and the UDP datagram protocol. Each NFS RPC message Need to be broken into multiple packets to be sent across the network. A big performance problem for NFS running under UDP on an Ethernet is that the message may be broken into up to six packets; if any of these packets are lost; the entire message is lost and must be resent. When running under TCP on an Ethernet, the message may also be broken into up to six packets; however, individual lost packets, rather than the entire message, can be retransmitted.

NFS clients A and B are communicating to primary data center, the primary data center/ facility is well connected to identical distant/remote data centre/facility that can serve the NFS client request all the time during fail over cases. The heart beat connection between these to data centre/facility is a dedicated IP cover by a firewall, this network is also encapsulated by fail over ether setup by adding one of the special nodes (Network Switch) to a dedicated port, which involves Active cluster and each data centre/facility is covered up by RAID service at the Disk. The Multi-path disk level connectivity and strong heart beat connection between the special nodes (Switch) make it a highly available setup for any cluster activity. NFS clients are making request over a TCP channel to their local data centre, each of these requests are handled by the NFS server at PDC and request is served, any repository is snapped at the aggregate levels of the disk pools placed at the primary data centre.

The transactions committed at primary data centre are copied as a form of timely snapshots for every 3 seconds across the disk pools to the distant/remote data centre/facility continuously, each copy of transaction is authenticated and acknowledged to guarantee the mode of communication between the heart beat signal connectivity between the data centers. The NFS clients can also be plugged directly to the distant/remote data centre with parallel support connectivity to perform direct communication during the local disk failures, this can happen vice versa between the data centers.

A physical snapshot mapping is also made possible to keep the data identical at every point of time to hold the highly available status to the clients of outside world. All NFS clients are configured with auto mount feature enabled so that they can do very quick access to local server and also allow mounting and unmounting by loading less procedure calls, and file handles across the server and themselves.

The NLM and NSM are the two protocols responsible for stateful and provides a notification mechanism following client or server failure, it also notifies during recovery across server restart requires NFS Server. Whenever a server goes down and comes back it notifies all the clients connected to the server about the healthy status, locks and claims in due with the clients when server went down. The NFS standards allows one lock file per volume, which contains caller name – identification of client host and lock file name, range, client address, protocol version and procedure.

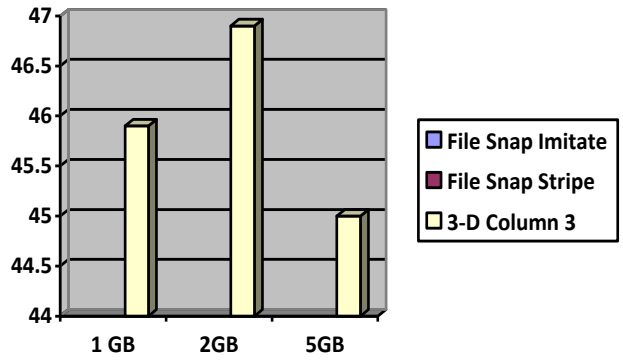
NFS Server enforces grace period to allow lock reclaims by NFS Clients after server reboot (about 60 seconds) standard timeout by all storage vendors like (NetApp, EMC, HP, IBM), during the grace period NFS Server registers for IP address PnP notifications from the operating system, When an IP address online notification is received, an LPC is performed to query Failover Cluster for the netname (and hence the NFS Virtual Server) that it should be scoped to All unknown endpoints are put in a deferred list while waiting for the Failover Cluster resource group, NFS Resource DLL and the NFS Virtual Server to come online.

5. RESULTS AND DISCUSSIONS

The tests are conducted on both UNIX and Windows file system and environments to figure out whether induced methods are able to reach the expected standard results. Since crucial issue in the design of very large disk arrays is the protection of data against catastrophic disk failures. Although today single disks are highly reliable, when a disk array consists of 100 or 1000 disks, the probability that at least one disk will fail within a day or a week is high. Hence the tests are conducted on a flat file system design and environment.

Table 1. Performance Report

RAID Type	1 GB	2GB	5GB
File Snap Imitate	18.17 Sec	23.05s Sec	41.01 Sec
File Snap Stripe	27.15 Sec	31.45 Sec	57.11 Sec



6. CONCLUSION

The need for the deployment of data protection technology is most essential in today's IT infrastructure maintenance, business specific data protection needs to be pushed based on the customer requirement is most essential.. However, when the implied costs are considered, data protections technology often either boycott the business need altogether or provide very poor performing solutions. In this paper, it is shown that this data protection technology method allows us to flexibly establish and secure the data of simple business to highly skilled customer business data centres.

It is also shown that the proposed solution allows flexibility in the control of the data protection in two different ways between all created networks and indeed the global network. To further this research work, considerations can be given to the implication of extending this to all kinds of block level and file level data onto storage. The researcher must also consider the benefits and implications of applying this data protection method in and with various other different file systems and storage of a data center.

7. REFERENCES

[1] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=364531&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D364531

[2] http://www.linuxjournal.com/article/2391

[3] http://www.smbitjournal.com/2012/11/choosing-a-raid-level-by-drive-count/

[4] http://www.cypress.com/?docID=37074

[5] http://gridsec.usc.edu/hwang/papers/JSA'00.pdf

[6] http://www.hpl.hp.com/hpjournal/95jun/jun95a11a.pdf