# A non-proprietary RAID replacement for long term preservation systems

Samuel Goebert
Hochschule Darmstadt
University of Applied Science
Darmstadt, Germany
samuel.goebert@bigcurl.de

Alain Sarti
Hessen Main State Archive
Wiesbaden, Germany
alain.sarti@hhstaw.hessen.de

## 1. INTRODUCTION

Disk-based storage, as suggested by Rosenthal et al., is the de facto standard for long term storage solutions [1]. Established by Patterson et al., RAID-based systems are a basic building block and have been a common best practice for building large scale storage systems [2].

Increasing disk failure rates, as experienced by Pinheiro et al. and proprietary ways to access the file system render the benefits of a RAID-based system questionable for longterm preservation [3].

A system with a RAID-level of 6 loses all its data if three disks are unavailable at the same time. Otherwise the lost content can be recovered by replacing one or both of the unavailable disks (with either a new one or a hot spare one).

We developed a RAID-free storage system that is able to replace RAID as a fundamental building block. The approach named NRN (No-RAID-Necessary) distributes a configurable number instances of data over a number of drives. To experience data loss all disks which hold an instance of a file have to fail. Even in this case only the data on those disks is lost. The data on the other discs is still accessible.

## 2. OVERVIEW

In large scale network storage systems like Amazon S3 and LOCKSS, the concept of treating a file as a whole object is part of the strategy against data loss. Splitting the file into chunks and putting them on different nodes in the network raises the overall speed, while accessing the file but also raises the number of machines necessary to fully recover a file [6] [7].

While it is possible to recalculate missing chunks of a file if redundant information is added, it still depends on the algorithm, how many of the chunks have to be intact to recalculate the missing chunks in the file. In a one chunk per node distribution strategy, the number of nodes that have to be intact to fully retrieve a file is determined by the algorithms ability to recover the file.

This is in contrast to a whole object approach, where one node holds a complete copy of a file. While this approach takes up more storage space since multiple copies of a file are stored in the system, only one node is needed to fully retrieve a copy. This makes the overall system robust against node failures.

While this approach is widely used for nodes in a network, the nodes themselves follow a different pattern for storing the files on disk. In large systems up to 48 hard drives are used per machine to form a node in the system. In most cases a RAID system is used to let the drives appear as a single large volume to the software that stores files into these disks.

Instead of taking the same approach as the network level and translate it to the disks instead of nodes, the files are split up and distributed over several disks with all the disadvantages that are avoided on the network level. A single failing drive can take down a complete node, which might result in many network traffic since the minimal number of copies of a file is enforced by the managing system.

What NRN does is taking the lessons learned from the network level and applies them to the node level. Individual disk fulfill the same role as nodes on the network and store full copies of the file on more than one disk. If a drive fails, only the missing content from the drive has to be replicated. Replication on the local bus happens with maximum bandwidth provided by the drives and does not utilize CPU cycles while copying data.

Also only one disk is affected during the recovery stage and not the whole system, as it would be the case in a RAID system where the missing data is recalculated from the remaining disks. Every workflow that might be enforced during the boot process like a fsck disk check up can be started in parallel on all disks which greatly improves boot up time of a node.

## 3. APPROACH

In the presented approach, several consumer grade disks are connected individually to a system, which, in contrast to a RAID system, are not logically combined to form one big drive. The disks are accessed through a thin software layer,


iPRES2011, Nov. 1–4, 2011, Singapore.

which is responsible for replicating incoming files over several disks. Several instances of a file establish the basis to achieve high availability for the files and durability against disk failures.

The abstraction through which the overlaying software access NRN is provided by FUSE *http://fuse.sourceforge.net*. FUSE allows a filesystem implementation in user space without coding directly in the kernel. The resulting filesystem hides the complexity of dealing with all disks individually and provides a single folder interface for the software that wants to store data. This makes it also possible to utilize the storage method also with software that is not optimized for the usage of more than one disk.

NRN always returns one of the available data instances when a client requests a file. This ensures that a file can be retrieved, even if the system has detected a disk failure and while redistributing the lost data. In case of a disk failure, it tries to comply with the predefined number of instances by recreating them on a hard disk which does not contain one already.

Recovery time in a NRN system depends on the amount of data on the failed disk, not on total system capacity or even individual disk capacity. If remaining total system capacity allows it, the system does not need a hot spare or replacement disk to start issuing new instances of the lost data onto the remaining drives.

The problem of distributing the instances onto the disks is solved by using one dimensional bin packing problem algorithms. For this purpose Lee et al. provided a first fit algorithm [4]. We identified two approaches, which suits long term archives best. They only vary by the number of disks available to the algorithm.

In the first approach we put a strong emphasis on high availability. Data is distributed equally onto all available disks. The drive with the lowest total capacity stores a new file. Due to the fact that only a fraction of files have to be restored, the recovery time from a drive failure is minimal. A higher number of disks means better protection against total data loss.

The second approach puts the emphasis on growing the capacity as needed. Disks are filled one by one. Initially only the minimum number of disks have to be attached to the system. If full capacity is reached, more disks are attached to the system to expand the overall capacity. This approach enables to start with a small upfront investment and only add drives when they are really needed.

# 4. CONCLUSION

Our research revealed that by replacing the RAID components with a system running NRN we have to accept a lower space usage efficiency and throughput. It is possible to keep most benefits from the RAID approach like robustness against individual disk failure and hot spare disks to lower maintenance reaction times. In addition we removed the proprietary file system, decoupled the system recovery time from the total disk capacity and lowered the probability of total data loss.

# 5. REFERENCES


[1] D. S. H. Rosenthal, M. Roussopoulos, TJ Giuli, P. Maniatis, and M. Baker. Using hard disks for digital preservation. In *Imaging Sci. and Tech. Archiving Conference*, 2004.

[2] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). In *SIGMOD88 International Conference On Management of Data*, SIGMOD '88, pages 443, Chicago, IL, USA - June 01 - 03, 1988. ACM.

[3] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX conference on File and Storage Technologies*, pages 2–2, Berkeley, CA, USA, 2007. USENIX Association.

[4] C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. *JACM*, 32:562–572, July 1985.

[5] P. Constantopoulos, M. Doerr, and M. Petraki. Reliability modeling for long term digital preservation abstract. In *9th DELOS Network of Excellence thematic workshop "Digital Repositories: Interoperability and Common Services"*, Foundation for Research and Technology, Hellas (FORTH), Heraklion, Crete 11-13 May, 2005

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41:205–220, October 2007.

[7] V. A. Reich and D. S. H. Rosenthal. Lockss: Building permanent access for e-journals â practical steps towards and affordable, cooperative, e-preservation, and e-archivingprogram. In *ELPUB*, 2003.