CHALMERS

A Cross-Platform, High Performance Shared Storage System

Master of Science Thesis in the Programme Networks and Distributed Systems

RICKARD NORDSTRAND

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden, December 2009

A Cross-Platform, High Performance Shared Storage System

RICKARD NORDSTRAND

Examiner: ROGER JOHANSSON

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
Sketch of a SweDisk rack case © Al Briscoe

Department of Computer Science and Engineering
Göteborg, Sweden, December 2009

Abstract

Advancements in information technology pushes the requirements of networks and storage solutions to new levels. The digital media industry is one particular area where the increased performance demands conflicts with the requirements of multi-user, cross-platform systems. While the concept of Storage Area Networks (SAN) sets new records on data throughput speeds, the interopability between Windows and OS X systems offered by Network Attached Storage (NAS) is lost. SweDisk is a startup company that works closely with the digital media industry to develop new storage solutions to fit their constantly higher demands. A new prototype system is designed, implemented and deployed at customer sites as distributed systems and user-space filesystems play key roles when the super-computing concepts of clustering and scalable storage are brought to the world of digital media production.

Sammanfattning

Stora framsteg inom informationstekniken ställer nya krav på nätverk och lagringslösningar. Den digitala mediabranschen är ett av områdena där de ökade prestandakraven kolliderar med krave på fleranvändarstöd och plattformsoberoende. Medan SAN-lösningar slår nya rekord i överföringshastighet går man miste om det så givna plattformsoberoendet i NAS-systemen. SweDisk är ett ungt företag som med hjälp av ett nära samarbete med den digitala mediabranschen utvecklar nya lagringssystem för att uppfylla de allt högre kraven. Distribuerade system och user-space-filsystem spelar huvudrollerna när superdator-koncepten kluster och skalbar lagring introduceras i den digitala mediabranschen när ett nytt prototypsystem tas fram och driftsätts hos kund.

Preface

This Master's Thesis is the final work at the M. Sc. Programme in Networks and Distributed Systems at the Department of Computer Science and Engineering at Chalmers University of Technology. The thesis consists of 30 credits, which is equivalent to 20 weeks of work. The thesis and work was carried out at SweDisk AB in Gothenburg. This thesis is a part of a project at SweDisk consisting of developing a shared storage product for commercial use. The software solution developed during this thesis project was based on work done by Ergodata AB during early 2009. Ergodata's work consisted of developing and evaluating the hardware prototype that this thesis is based on.

This report was written with the great help and support of the whole SweDisk team. I would especially like to thank my supervisor Håkan Edler for his great expertise and support as well as my examiner Roger Johansson. I also would like to personally thank all the people involved in SweDisk, past and present, including: Simon Givre, Tuve Nordius, Marcus Söderström, Karin Broman, Bengt-Erik Olsson, Goran Rajkovic and Jörgen Hansson.

Contents

Chapter 1

Introduction

This thesis constitutes a significant part of the early development process for new storage solutionsfor the digital media industry. SweDisk AB is a start-up company established in the autumn of 2008 as part of the business incubator Chalmers Innovation at Lindholmen Science Park in Gothenburg, Sweden. The company's aim is "creating robust, logic and efficient solutions that will provide the user with quicker, simpler, cheaper and more safe network storage solutions than he/she currently has." [2]. The first SweDisk product has been in development since the beginning of 2009.

This thesis is structured as seven chapters which includes this introduction, an analysis of shared storage technlogies, a description of reference sites, the methodology used, achieved results, as well the ending discussion and conclusion chapters.

1.1 Background

The amount of data stored in the world increases extensively every year. According to a study carried out by the market research firm IDC, there were a total of 281 exabytes (281 million terabytes) [9] of data existing in the world in 2007. Combined with the astonishing growth rate of nearly 60 % per year, IDC expects the produced data amounts in the world to reach ten times as much as early as in 2011. Advancements in digital media production is a significant contributor to the extended requirements of data storage. The increased adoption of high-definition (HD) video for television and film results in storage requirements seen in few other areas. When the video resolution increases, the storage requirements explode!

As a reference, uncompressed standard-definition video (SD), commonly consisting of a resolution of 720x576, has a bit rate of nearly 30 MB/s[1]. The current top-of-the-line standard used on various digital TV channels and BluRay discs is called 1080p, or "Full HD". This standard gives an uncompressed bit rate of approximately 140 MB/s[2].

[1] SD bit rate calculated from 720x576, 25 FPS, 10 bit YUV, 48.0 kHz/16 bit stereo audio

[2] HD bit rate calculated from 1920x1080, 25 FPS, 10 bit YUV, 48.0 kHz/16 bit stereo audio.

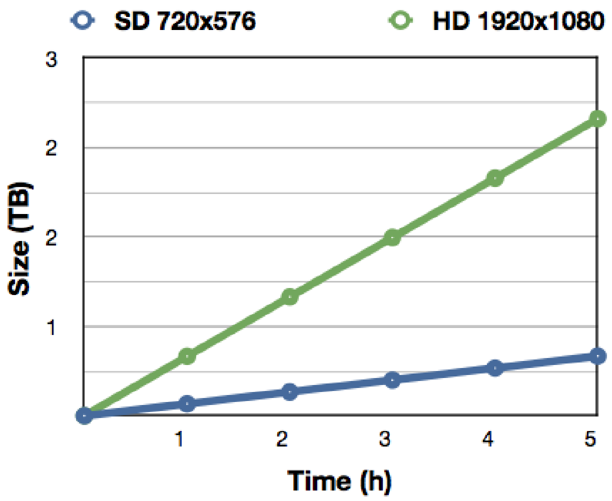

Figure 1.1. SD and HD video storage requirements compared.

While storage capacity is cheap and easy to extend as needed, the transfer speeds of available storage media lags behind significantly. As of today, a single Serial ATA disk with a rotational speed of 7,200 RPM reaches a maximum of 132 MB/s [13]. The next generation of hard drives, the flash memory based Solid State Drives (SSDs) has taken a significant step in performance and offers read and write speeds up to 232 MB/s [14]. SSDs are however, yet far from as affordable as the Serial ATA drives.

For bandwidth heavy task such as HD video editing, regular hard drives in a desktop computer are simply not suitable for the job. While single hard drives may be sufficient for lower quality video production, there is still another problem with sharing the material among co-workers.

The problem itself is not new and there are several possible solutions involving high-speed networks and servers with multiple disks combined into RAID arrays. That does however not solve all the problems. An additional issue in media production is cross-platform interoperability. A video production site is commonly a mixed computer environment consisting of Mac or Windows platforms as well as Linux servers, all dedicated to their specific tasks, such as video editing, 3D modeling, 3D rendering, data storage, archiving and backup. In a perfect world, these platforms would all work perfectly together.

1.2 Problem Description

There already exist several ways of how to approach the problem of shared storage. It seems however that there are no clear winner that sufficiently fulfills all the requirements needed for smooth and efficient production of HD video in a mixed computer environment.

The aim is to develop and implement a high performance, cross-platform, shared storage software solution for the SweDisk hardware prototype, primarily based on free open-source components. The results should comply to the vision outlined by the SweDisk company with the main objectives and selling points for the products being high performance, cross-platform interoperability, data safety and easy maintenance.

1.3 Goals

The goal of this thesis is to design and implement a shared storage system for use with Mac OS X, Windows and Linux clients. This shared-storage system should achieve a level of performance suitable for editing of HD quality video. The system should handle scaling to approximately ten client workstations.

The exact performance goals depend on the requirements of reference customers with respect to their previous solutions as well as the available hardware configuration at the customer site. Besides the performance and cross-platform interoperability, a trustworthy data storage is necessary for customer interest in the product, which includes hardware redundancy as well as a reliable backup functionality.

Successful implementation on a real-word customer site is a significant milestone in the development of the SweDisk products and therefore of the company itself. Consequently, successfully implementing a reference system at a customer site is a natural aim for this project.

1.4 Purpose

The SweDisk company pushes four Unique Selling Points (USPs) for their products. These USPs represent the essence of the products and should be outstanding factors of all product released by SweDisk. The four USPs are:

- Performance
- Security
- Reliability
- User-Friendliness

By spending less time on slow data transfers and time-consuming maintenance, while ensuring that data is always kept safe and consistent, customers will increase their efficiency and be able to spend more time and energy on creative tasks.

1.5 Delimitation

The scope of this thesis is limited to the design, implementation and validation of a new SweDisk prototype to be implemented at the reference sites. The work will be based on an existing hardware prototype that has been configured and evaluated by a company named Ergodata AB. Their work consisted of evaluating the RAID and 10 Gb Ethernet (10 GbE) technologies and the results are documented in an internal report [8] that will be used for further development of the reference prototypes.

The main tasks of this thesis project is to research and evaluate available software technologies for possible inclusion in the new reference prototype. The evaluation process consists of systematically measuring the performance of available components to get an overview of the possible design choices.

Extensive programming tasks such as implementing new filesystems or applications and porting software to new platforms are not regarded as a part of this thesis project.

Chapter 2

Shared Storage

There are two established concepts of shared storage systems: the easily confused Network Attached Storage (NAS) and Storage Area Network (SAN) systems. These two approaches differ in the way the filesystem and network layers are separated between server and client while the presentation to the end-user is however not significantly different. To understand the differences of SAN and NAS and their advantages and disadvantages is necessary when designing shared storage system.

This chapter starts out by describing the characteristics of different approaches to shared storage and then goes into briefly analyzing specific technologies with respect to the requirements of the SweDisk system.

2.1 Network Attached Storage

A NAS is the most common form of shared storage that every computer professional is familiar with while not necessary recognising this term. With a NAS, communication is done on a file level, which means that clients simply request a filename and a byte offset for reading or writing.

The communication between client and server is done with a network protocol on the application level of the OSI model and is sometimes referred to as a network filesystem. There are several established alternatives, with the three following protocols being the most common ones:

- Common Internet File System (CIFS), also known as the Server Message Block (SMB). Developed by Microsoft and hence the native file sharing protocol on the Windows platform.

- Network File System (NFS). The native protocol for UNIX platforms. Originally developed by Sun Microsystems.

- Apple Filing Protocol (AFP). The native Macintosh alternative as a part of the AppleTalk protocol suite.

The most prominent advantage of the NAS approach is that the client is not dependent on the actual filesystem used for storing the files on the server, as the communication is done on a filename-based level. Although each protocol is being tightly associated with its OS origins, there are implementations available for each of the other OSes, making them all cross-platform operable. However, due to the complexity involved in the NAS approach and the workload put on the server, the performance - and especially the ability to scale with multiple clients - is a questionable property that has to be investigated.

2.2 Storage Area Networks

The Storage Area Network, not to be confused with the NAS, is a shared storage system working on the storage device level, below the filesystems, as seen in Figure 2.1.

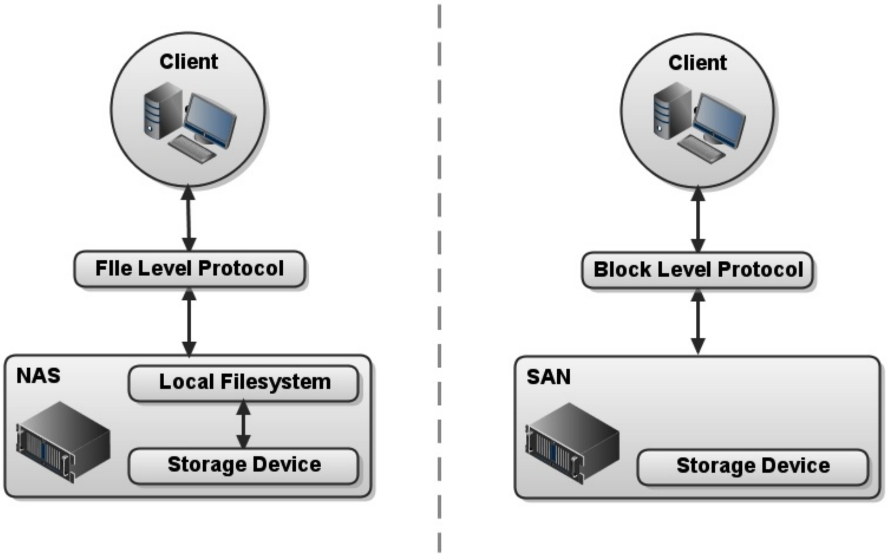

Figure 2.1. SAN and NAS compared.

In contrast to the NAS, the SAN clients requests raw data blocks for their operations, meaning that filesystem actions are only handled at the client side and the network communication between server and client consists only of the low-level hardware commands that would normally be sent to the storage device itself. In order to send low-level hardware commands on the network, an encapsulating SAN protocol is needed. The most common protocols for SAN traffic are:

- Internet SCSI (iSCSI) - Encapsulates SCSI commands in IP packets on ordinary Ethernet hardware.

- ATA over Ethernet (AoE) - Encapsulates ATA commands directly into Ethernet frames.

- FibreChannel - A Proprietary hardware/software technology not considered an alternative by SweDisk, due to expensive hardware and licensing costs.

- InfiniBand - Another hardware/software technology excluded for the same reasons as FibreChannel.

Using a SAN protocol means that storage devices present themselves to the client OS the same way as local devices do. This allows for performance gains, as several network layers are omitted from the SAN system, including the transport and application layer protocols used in the SAN approach. However, sharing a storage device on the block level introduces another problem, which is file locking.

2.2.1 Concurrent Write Access

A major limitation of the basic SAN approach is the problems with concurrent write access. While reading files is not an issue, concurrent writes are more problematic. When leaving out the filename-based application layer protocol of the NAS system in favor of the raw block access of the SAN, the server is no longer capable of managing the locking of files used by the clients, which has to be done on the filesystem level. Ordinary local filesystems like Microsoft NTFS, Apple HFS or Linux ext3 has no support for safely operating in a shared storage environment where multiple users may write data at the same. Failing to prevent multiple clients from concurrently writing data may result in an inconsistent filesystem with corrupt data. In order to ensure data integrity, clients must handle concurrency and file locking among themselves in a distributed manner. There are several available filesystems offering such functionality.

2.3 Clusters and Distributed Filesystems

A distributed system may be defined as "one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages". [7], while a cluster is a subset of distributed systems. The concept of computer clusters was introduced in the 70s by Tandem Computers, Inc. and can simply be defined as a group of computer nodes interconnected to share a common resource such as CPU and RAM [5]. A common application for clusters is super-computing, where hundreds, or even thousands of nodes collaborate to solve some kind of computing intensive task.

In the super-computing scenario, the performance scaling is the biggest issue, combined with requirements of high availability and data redundancy. In the case of filesystems, the shared resource is a storage device and the distributed

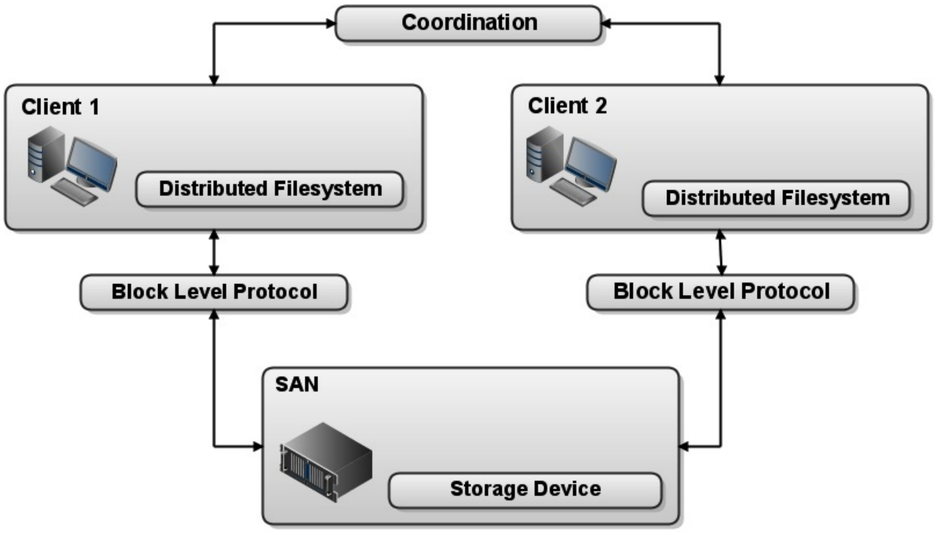

Figure 2.2. A shared-disk filesystem operating with two clients.

mechanism for solving the problem is called the Cluster Lock Manager (CLM) [12] or by developers more commonly called the Distributed Lock Manager (DLM).

There are several types of clustered filesystems, aiming to solve different problems in a network. In this thesis, the following two definitions will be used:

- Shared-disk filesystem: A clustered filesystem in its simplest form: A distributed filesystem coordinating simultaneous read/write access among multiple clients to one unified storage area, with the help of a DLM, as illustrated in Figure 2.2.
- Distributed parallel filesystems [11]: A distributed filesystem where multiple clients coordinate access of a storage volume that constitutes of multiple servers, each contributing physical storage to the same logical volume. The logical volume is transparently presented to the clients as one single storage entity, as seen in Figure 2.3. By eliminating the performance bottleneck of a single server and instead distributing the storage among multiple servers, there are better possibilities for data redundancy and scalability.

2.4 Open-Source Clustering

The open-source community offers a wide range of clustered filesystems for running on top of a SAN, each focusing on different applications and scales of clusters. Table 2.1 contains a brief summary of the most prominent alternatives followed

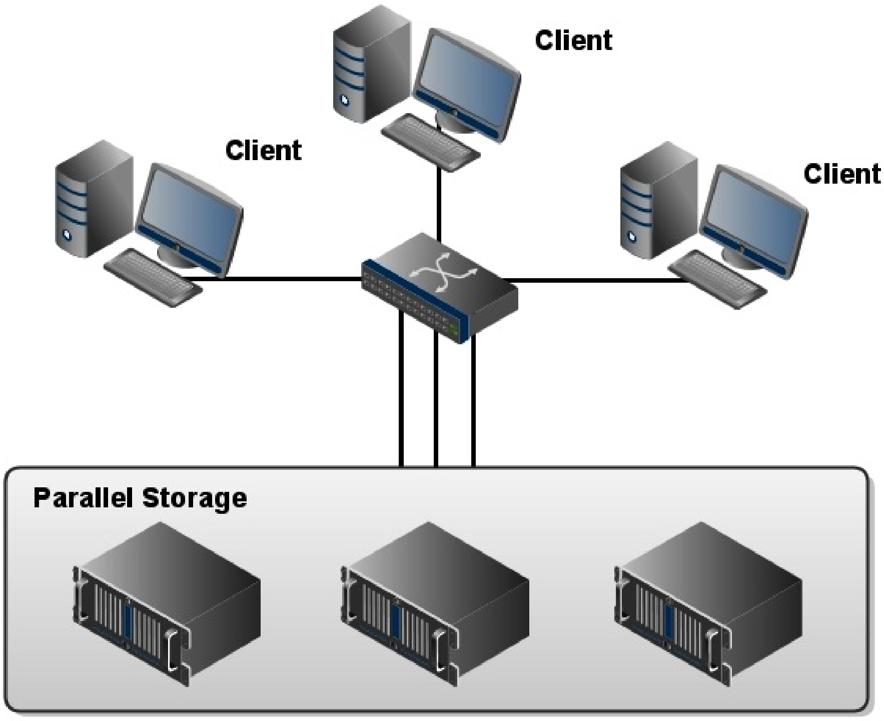

Figure 2.3. A distributed parallel storage system operating with three clients.

by short introductions to the filesystems, including their origin and development focus. In this part it will also be determined which filesystems will qualify as appropriate candidates for further evaluation when designing the SweDisk system.

None of the mentioned projects requires special network hardware and runs on top of regular Ethernet networks.

2.4.1 GFS2

The Global Filesystem is a clustered file system originally developed by Sistina Software in 1996, a company that was later acquired by Red Hat in 2003. In November 2006 GFS version 2 was merged into the official Linux kernel. GFS is a clustered file system allowing multiple nodes to read and write to a single shared filesystem. No separation is made between server and clients and each node is simply a peer in the cluster, coordinating their actions using a DLM. GFS2 is a part of the Red Hat Clustering Suite which contains tools for cluster membership management, fail-over support, load balancing, quorum and other features. Red Hat claims that GFS2 supports scaling to 100 nodes, while the Clustering Suite states support for 32 nodes.

Name	Filesystem	OS Support	License
Red Hat	GFS2	Linux	GPL
GlusterFS	Gluster	Linux, Windows, OS X	GPL
Lustre	Sun Microsystems	Linux	GPL
OCFS2	Oracle	Linux	GPL
PVFS	PVFS.org	Linux	LGPL

Table 2.1. Clustered filesystems licensed as open-source.

GFS2 appears to be a mature and capable filesystem, suitable for testing Linux clustering performance.

2.4.2 GlusterFS

GlusterFS differs from the other systems by its user-space design. The self-explanatory Filesystems in Userspace (FUSE) library makes GlusterFS a highly-portable filesystem, separated from complex OS kernel internals. The FUSE library (originally a Linux application) has been ported to both OS X and Windows. GlusterFS claims to be designed for parallel storage clusters as well as smaller SAN systems. [1]

GlusterFS performance was evaluated by Ergodata as the only tested SAN technology in their report [8]. The tests were performed between two Linux machines on a 10 Gbit Ethernet (10 GbE) network connection resulted in a maximum of 300 MB/s reading and 420 MB/s writing. With a Mac and a Linux machine however, reading reached only 140 MB/s and writing was as low as 17 MB/s. There was no clear conclusion of what caused the problems.

The poor Mac performance rules out GlusterFS as a candidate for this study but should still be observed for further developments.

2.4.3 Lustre

Lustre is a distributed parallel storage cluster introduced in 2003 and acquired by Sun Microsystems in 2007. According to Sun it is powering seven of the world's ten largest computing clusters while handling tens of thousands of nodes, petabytes of storage and hundreds of gigabytes in throughput [15]. Being a distributed parallel storage system, the infrastructure in a Lustre system differs significantly from other open-source alternatives. Not only does Lustre support parallel storage nodes, labeled as Object Storage Targets (OST), also server tasks may be distributed among nodes. Multiple nodes taking the roles of Management Servers (MGS), Metadata Servers (MDS) and Object Storage Servers (OSS) gives the impression of a highly-scalable storage cluster. An example configuration is shown in Figure 2.4 where each MDS represents a logical volume on the network.

In contrast to the most of the other clustered open-source filesystems, Lustre does not rely on any third-party SAN protocols such as iSCSI or AoE. Sun has developed their own equivalent, LNET, running directly on top of Ethernet or

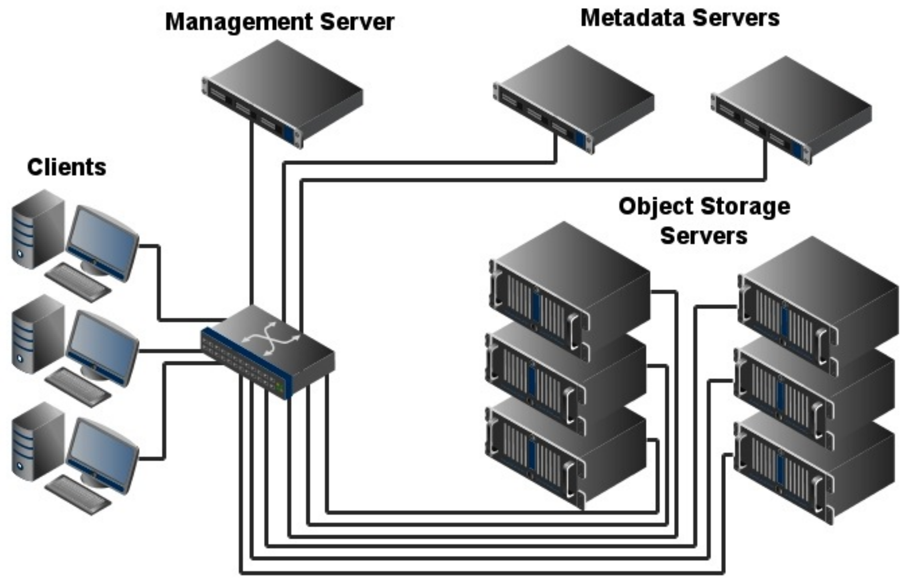

Figure 2.4. An example of a large Lustre system topology providing two different filesystems, each constituting of three OSTs.

InfiniBand links. The underlying filesystem is however currently based on the ext3 Linux filesystem, with plans to incorporate Sun's own next-generation filesystem ZFS exists. [16].

Lustre is licensed under the GPL but is not a part of the official Linux kernel.

While being targeted mainly at large cluster systems, Lustre bears unique functionality that makes it an interesting candidate for future development. The possibilities of Lustre is further discussed in Chapter 2.9.

2.4.4 OCFS2

The Oracle Clustered Filesystem was merged into the Linux kernel as an experimental feature in March 2006. OCFS2 was declared a stable feature of the Linux kernel in November 2006, coinciding with the merge of the previously mentioned GFS2. OCFS1 originated as a clustered filesystem for Oracle's database systems only, but has since developed to a fully functional filesystem in its 2nd incarnation.

OCFS2 is together with GFS2 considered the most interesting candidates for Linux cluster evaluation.

2.4.5 PVFS2

The Parallel Virtual Filesystem started development in 1993 at the Parallel Computing Research Group at Clemson University, South Carolina. PVFS2's focus lies on a modular design for adoption to new hardware and algorithms suitable for research in parallel computing. PVFS2 is licensed as Lesser GPL (LGPL) and not included in the official Linux kernel.

While having a long history of development, PVFS2 has been omitted from further evaluation due to other filesystems appearing more suitable for this project.

2.4.6 ZFS

While not being a clustered filesystem, the "Zettabyte Filesystem" (ZFS) supports built-in NAS exports as well as iSCSI SAN capabilities. Developed by Sun Microsystems, ZFS incorporates storage pools, volume management, snapshots, software RAID, strong data integrity and a lot of other modern filesystem features, without the need of accompanying tools. While being open-source, the CDDL licensing is a controversial issue among the Linux community [4], as it is not compatible with the GPL used by the Linux kernel, making it impossible for legal inclusion of ZFS into the Linux kernel. [1]

2.5 Open NAS/SAN Solutions

There are a couple of open-source projects offering a complete software solution for setting up a NAS or a SAN. These are OS distributions focusing on the single purpose of sharing files, much like the concept of stand-alone firewall distributions.

Name	**OS Base**	**License**
FreeNAS	FreeBSD	BSD
Openfiler	Linux	GPL
NexentaStor	Nexenta OS	CDDL/GPL

Table 2.2. Complete NAS/SAN distributions with open-source licensing.

2.5.1 FreeNAS

FreeNAS is a minimal OS distribution based on FreeBSD. As the name implies it is primarily a NAS system supporting all the common protocols. Simple SAN functionality is also supported through the iSCSI protocol. The BSD nature of this system gives it no support for clustering. It has however support for ZFS, due to the BSD license being less restrictive than the GPL used by the Linux kernel. The FreeNAS system uses a web interface for administration.

[1] A FUSE implementation of ZFS was developed as a part of the annual Google Summer of Code in 2006 and allows ZFS to be run on Linux by circumventing the licensing issues of incorporating the filesystem into the kernel. It is however regarded as experimental and not suitable for production use. [6]

2.5.2 Openfiler

A Linux based SAN/NAS system with features comparable to the FreeNAS system. In addition, it claims High Availability (HA) cluster support. In practice Openfiler incorporates the DRBD software, which allows mirroring of block devices through the network, "similar to a network-based RAID1" as stated on the site. This approach of clustering is meant for storage redundancy, but does not function as a cluster for the purpose of sharing a storage resource to multiple users.

2.5.3 NexentaStor

Based on an OpenSolaris kernel incorporated into an Ubuntu Linux environment, Nexenta has "the power of OpenSolaris with the usability of Linux" [3]. NexentaStor is a commercial derivative of Nexenta, focusing on SAN and NAS services powered by ZFS, but lacks clustering capabilities.

Just like the aforementioned open-source NAS/SAN distributions in mentioned in this sub-chapter, NexentaStor lacks real clustering functionality, which makes it unsuitable for this project and will not be evaluated further.

2.6 Proprietary Clustered Filesystems

Apart from the free, open-source clustering software primarily considered for this project, there are also several proprietary, closed-source clustered filesystems.

Name	**OS supported**	**Network technology**
Apple Xsan	OS X	FibreChannel
IBM GPFS	Linux, Windows	FibreChannel, InfiniBand, Ethernet
SGI CXFS	Linux, Windows	FibreChannel, InfiniBand
Quantum StorNext	Linux, Windows, OS X (with Xsan)	FibreChannel

Table 2.3. Proprietary cluster technologies.

As seen in Table 2.3, only one of the proprietary SAN solution claims cross-platform support as well as Ethernet compatibility, which is GPFS from IBM.

2.6.1 IBM GPFS

The General Parallel Filesystem (GPFS) is as the name implies a distributed parallel clustered filesystem. GPFS emerged as the Tiger Shark filesystem in 1993 [10] and has since been developed to one of the most common filesystems used in super-computing.

It should be noted that the Windows support is limited to Windows Server 2003 R2 x64 only. But, the fact that GPFS has at least limited Windows support makes it the only clustered technology that officially supports both Ethernet networks and more than one of the SweDisk target systems. However, the proprietary closed-source format makes it impossible to customize and extend as needed.

2.7 NAS Clusters

An alternative idea to the pure SAN and NAS technologies is to share the load of a heavily used NAS system by setting up multiple NAS servers combined with an underlying shared-disk filesystem, distributing the workload across multiple NAS servers. In theory, this would solve any scaling issues with the expenses of adding additional servers.

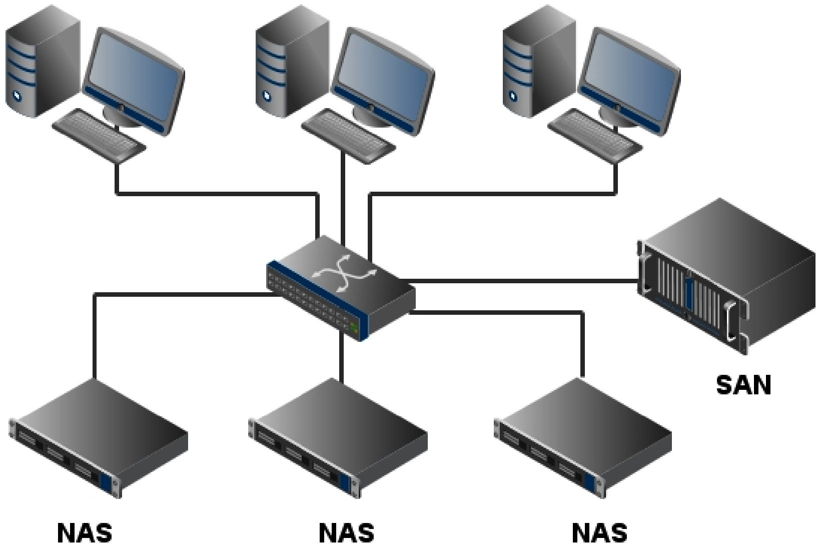

Figure 2.5. A distributed NAS with an underlying shared-disk cluster.

There are derivations of both SMB and NFS servers offering support for distributed deployment. However, the multi-server nature of the NAS cluster approach makes it unsuitable for a single-server SweDisk system. NAS clusters will not be evaluated further in this thesis.

2.8 Virtual NAS Clusters

The virtual NAS clusters was brought up within the SweDisk company. The concept is basically the aforementioned NAS cluster implemented using virtualization software. By using virtualization software, such as VMware or Sun VirtualBox, it is possible to run any OS inside a virtual machine within the already installed system. See Figure 2.6.

A Linux system running inside a virtual machine can easily be set up on a Windows client as well as an one running OS X. This allows the virtual Linux machine to be set up in a cluster, just like real physical clients. The idea to re-export the attached SAN storage to the real OS using a standard NAS protocol.

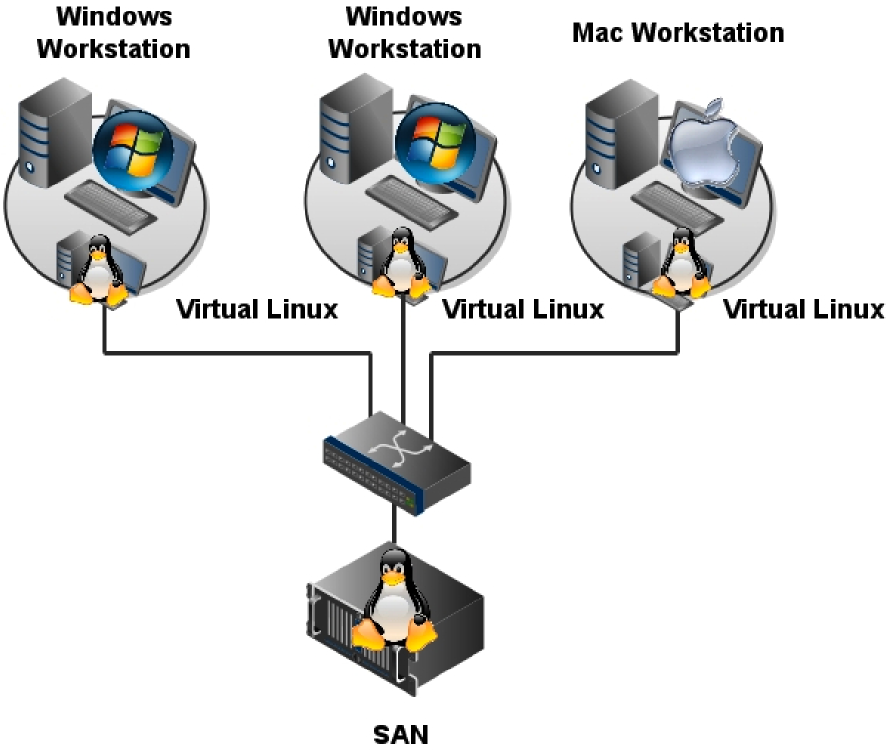

Figure 2.6. A distributed NAS and SAN cluster running on top of virtual machines.

The advantage of this virtual NAS solution is the portability of the virtualization software that allows a Linux cluster to be set up within a Windows or OS X machine, overcoming the lack of complete cross-platform supported clusters, while the distribution of the NAS servers to each client avoids any scaling issues with the NAS technology.

2.9 Porting

Undoubtedly the most effective, but also most difficult and time consuming way to design a cross-platform high performance shared storage system, is to port an already existing cluster software to new OS platforms. The many available open-source clusters with generous licenses makes this approach theoretically possible.

However, the large differences in OS kernel architectures makes this a very complicated task, especially considering the amount of code involved. OCFS2 and GFS2 occupies 25,000 and 40,000 lines of kernel-space code, respectively. On top of the kernel-space come the user-space services required to set up and manage the cluster.

While OS X and Linux have some limited UNIX heritage in common, the Windows platform has a completely different kernel architecture, making it necessary to develop two completely new and independent implementations of an existing open-source Linux cluster in order to reach full cross-platform support.

2.9.1 Lustre

When investigating the possibilities of developing a port of one of the previously mentioned open-source cluster filesystem, Lustre has a feature that makes it stand out among the others: Sun provides LibLustre, which is a user-space library interface to the Lustre architecture, aimed to provide a possibility of developing user-space applications with Lustre functionality built-in, without mounting a Lustre volume on filesystem level via kernel-space. LibLustre is considered experimental code and contains certain limitations regarding multi-threading and the fact that it is still tied to the Linux environment.

2.10 SweDisk Prototype 1

Confidential.

Chapter 3

Reference Sites

Confidential.

Chapter 4

Method

The approach chosen for designing the high performance, cross-platform shared storage system is largely based on an extensive evaluation of the available shared storage technologies introduced in Chapter 2. The performance of a number of candidating SAN and NAS technologies will be evaluated by measuring the throughput on different levels in the system to study the efficiency and the overhead factors caused by possible bottlenecks.

As an example of this, the filesystem performance will first of all be measured locally to ensure optimal configuration at the server side. The remainder of the tests will incorporate the client performance using both 1 and 10 GbE network connections. Both SAN and NAS technologies will be measured and compared in order to get an overview of the current state of shared storage and the performance potential of different approaches.

The results given by these tests form a base for the design and implementation of the next SweDisk reference system. When the prototypes are ready and put in place at customer sites, the systems will be validated by performing additional tests in their real-world environment. The systems will then be left for the company users to test and evaluate their everyday tasks, while being connected online for remote control and 24/7 monitoring.

4.1 Measuring Methods

To obtain a better understanding of the advantages and disadvantages of the described SAN and NAS technologies, their capabilities will go through rigorous tests, measuring their performance during multiple scenarios using the newly assembled second prototype hardware. In order for the tests to best represent the application of digital media applications in general and video production in particular, certain assumptions must be made about the assumptions that the tests will be carried out within.

Reading and writing video and audio data is done sequentially. In contrast to most other uses of storage, where data are read and written at random locations in the address space, audio and video files are always read and written in a sequence of blocks. Since traditional mechanical hard drives uses an arm to move the read head across the platter, random access requires longer seek times which in turn decreases the throughput of the drive. All read and write tests in this report were done sequentially, without any considerations of random read and write performance. Also, all read and write tests were performed separately, and never mixed, in order to limit the number of tests.

4.1.1 Measuring Tools

To measure the performance, i.e. read and write throughput rates, primarily two command-line applications has been chosen:

- For network performance, the nuttcp tool will be used. Nuttcp measures TCP or UDP throughput in both directions using the client/server model. The default mode of operation is sending the maximum amount of packets for ten seconds and then returning the average speed. Just like the disk tests, the network experiments will only be done in one direction at a time to keep it simple, and never in duplex. When in some cases, nuttcp did not execute properly due to platform specific problems, the very similar tool iperf was used instead.
- Data read and write throughput will primarily be tested using the standard UNIX dd tool. It is a general-purpose tool to read bytes from one source and write to another, while returning the average speed. Apart from source and destination, the parameters supplied to the program is the block-size in which data should be read and written, as well as the number of blocks that should be read. Due to the UNIX nature of dd, all Windows read and write tests where conducted using the xdd benchmark application.

4.2 Conditions

All experiments will be performed using the second SweDisk prototype which is the hardware configuration used for both reference sites. The RAID array consists of eight disks, in contrast to 14 on the first prototype.

Chapter 5

Results

Confidential.

Chapter 6

Discussion

Confidential.

Chapter 7

Conclusions

Shared storage systems are facing a new era as the network hardware takes a new leap in bandwidth. The old trusty, centralized NAS storage will have to step aside for new scalable, distributed concepts.

The surface starts to crack when the NAS is exposed to demanding multi-user environments and bandwidths that exceeds the current 1 GbE standard. SAN systems on the other hand, has shown that the light protocols and distributed coordination makes it possible to achieve an affordable shared storage system that is much faster than before.

While the SAN technology seems to address the performance issue, the cross-platform interoperability issue has turned more complicated as the low-level approach of the SAN requires more specific software solutions that cannot easily be ported to other platforms.

Computer users have always found a way to make use of higher bandwidth, and will continue to do so as high performance 10 GbE starts to replace the current standard. This change will force the software industry into adapting the super-computing technologies to new areas with high demands. Sun's plans of moving towards a system where the clustering functionality is more separated from the underlying operating system is one indication of this – making it possible for any Windows user to benefit from the impressive scalability of Lustre systems.

By migrating to distributed and highly scalable storage solutions, we will be able to more efficently utilize the advancements in network technology and to handle the explosive rate of increased data amounts in the world.

Bibliography

[1] GlusterFS User Guide. http://www.gluster.com/index.php. Retrieved: 27 November 2009.

[2] SweDisk AB. http://www.swedisk.se.

[3] The Nexenta Project. http://www.nexenta.org/.

[4] Jeremy Andrews. Linux: ZFS, Licenses and Patents. *KernelTrap*, http://kerneltrap.org/node/8066, 2007. Retrieved: 27 November 2009.

[5] Rene J. Chevance. *Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, and Storage Solutions*. Chapter 5 - Clusters and Massively Parallel Machines. Digital Press, 2005.

[6] Ricardo Correia. ZFS on FUSE/Linux. http://zfs-on-fuse.blogspot.com/.

[7] Tim Kindberg George Coulouris, Jean Dollimore. *Distributed Systems - Concepts and Design*. Addison Wesley/Pearson Education, 4th edition, 2005.

[8] J. Holmsbo and K. Sanden. Rapport Swedisk. *Ergodata AB*, 2009.

[9] IDC. As the Economy Contracts, the Digital Universe Expands. http://www.emc.com/leadership/digital-universe/expanding-digital-universe.htm, 2009. Retrieved: 17 November 2009.

[10] John M. May. *Parallel I/O for High Performance Computing*. Morgan Kaufman, 2000.

[11] Sunil Mushran. OCFS2 - A Cluster Filesystem for Linux: User's Guide for Release 1.4. 2008.

[12] Dilip M. Ranade. *Shared Data Clusters: Scalable, Manageable, and Highly Available Systems*. Chapter 12 - Cluster Lock Manager. John Wiley & Sons, 2002.

[13] Patrick Schmid and Achim Roos. New Desktop Hard Drives: Speed Or Capacity? *Tom's Hardware Guide*, http://www.tomshardware.com/reviews/2tb-hdd-caviar,2261-7.html, 2009. Retrieved: 11 September 2009.

[14] Patrick Schmid and Achim Roos. SSD Summer Slam: 12 New 2.5" And 1.8" Drives Rounded-Up. *Tom's Hardware Guide*, http://www.tomshardware.com/reviews/ssd-x25-m-vertex,2399-14.html, 2009. Retrieved: 11 September 2009.

[15] Sun Microsystems, Inc. Lustre. http://wiki.lustre.org/.

[16] Sun Microsystems, Inc. Lustre Roadmap. 2009.