Design of store-and-forward servers for digital media distribution

University of Amsterdam
Master of Science in System and Network Engineering

Class of 2006-2007

Daniël Sánchez (dsanchez@os3.nl)

27th August 2007

Abstract

Production of high quality digital media is increasing in both the commercial and academic world. This content needs to be distributed to end users on demand and efficiently. Initiatives like CineGrid [1] push the limit looking at the creation of content distribution centres connected through dedicated optical circuits.
The research question of this project is the following: *"What is the optimal architecture for the (CineGrid) storage systems that store and forward content files of a size of hundreds of GBs?"*

First I made an overview of the current situation. At the moment the Rembrandt cluster nodes [16] are used in the storage architecture. All data has to be transferred manually to the nodes via FTP. This is not preferred, because administration is difficult. Therefore a list of criteria is made for the new storage architecture. Important criteria are bandwidth (6.4 Gb/s) and space (31.2 TB a year and expandable).
I made a comparison between open source distributed parallel file systems based on these criteria. Lustre and GlusterFS turned out to be the best of these file systems according to the criteria. After that I proposed two architectures which use these file systems. The first architecture contains only cluster nodes and the second architecture contains cluster nodes and a SAN.
In the end it is recommended to install GlusterFS in the first architecture on the existing DAS-3 nodes [15] with Ethernet as interconnect network. This way costs can be saved for a SAN (considering the limited budget). Further tests are necessary to determine the exact configuration.

Contents

Chapter 1

Preface

I am a student at the University of Amsterdam (UvA)[1], and study System and Network Engineering (SNE)[2]. Part of the master's curriculum is to carry out a research project to ensure that students become acquainted with problems from the field of practice and gain collaborative skills, which require the development of non-trivial methods, concepts and solutions. This course is named Research Project 2. The duration of this project is four weeks.

To fulfil these requirements, I researched into the field of storage architectures. I am doing this project at the UvA. My supervisors are dr. Paola Grosso and Jeroen Roodhart from the University of Amsterdam. This project originates from the SNE Research Group[3] lead by dr. ir. Cees de Laat. The results of this project can be used by the UvA to develop a storage architecture.

This report contains abbreviations and terms with which is assumed the reader is familiar. For those who are not familiar with these terms, there is a definition in Appendix A. The symbol [A] is used throughout the report to indicate the definitions.

Acknowledgements: I would like to thank Jeroen Roodhart, Cees de Laat, Jaap van Ginkel and especially Paola Grosso for their supporting role during the project. They helped with making this project possible in the first place and also guided me during the project.

[1] UvA: http://www.uva.nl/
[2] OS3: http://www.os3.nl/
[3] SNE Research Group: http://www.science.uva.nl/research/sne/

Chapter 2

Project introduction

Production of high quality digital media is increasing in both the commercial and academic world. This content needs to be distributed to end users on demand and efficiently. Initiatives like CineGrid [1] push the limit looking at the creation of content distribution centres connected through dedicated optical circuits.

2.1 Research question

"What is the optimal architecture for the (CineGrid) storage systems that store and forward content files of a size of hundreds of GBs?"

With the storage architecture the following is meant: the way data is stored, the used technology. Examples are: a Storage Area Network (SAN) or a Distributed File System (DFS) [A] cluster.

With optimal is meant: matching the most criteria for the storage architecture. In any case performance will be an issue that has to be taken into consideration. These exact criteria will be described later.

2.2 Scope

The scope is to find a suitable solution for the store-and-forward servers to be used by the CineGrid collaboration project in Amsterdam. However, this solution can also be used for other projects similar to CineGrid. Furthermore, I only research the storage architecture. An exact implementation with a detailed description of equipment is not part of the research. An other important part is networking in a storage architecture: this is not covered in this project.

2.3 Approach

The following approach to the problem was chosen:

1. Search for related work through the internet and current documentation.
2. Describe the current storage architecture.
3. Make a list of criteria for the new storage architecture.
4. Search for different possibilities for a new storage architecture.
5. Make a comparison of the found possibilities.
6. Describe the optimal solution in more detail.

Chapter 3

Background

In this chapter I will explain the background of the project. The purpose of this chapter is to understand the research question and the research itself.

Before describing the more technical background, I will explain the basic components that are needed in a storage architecture:

- A location to store the data: the physical storage.
- A way to implement reliability for the data: RAID.
- A way to provide access to the data: a file system.

These topics will be described in this chapter. About performance: this also will be an important topic, which is included in the three components above. But first some background about CineGrid is explained to place the other topics in a context.

3.1 CineGrid

In this section I will explain the background of the CineGrid project initiative. Parts of the text in this section are quoted from the CineGrid.org website [1].

3.1.1 What is CineGrid?

Quoted from the CineGrid website [1]:

> *CineGrid is a non-profit international membership organization administratively based in California.*
>
> *CineGrid has been established to promote research, development and deployment of new distributed applications of ultra-high performance digital media (sound and picture) over advanced networks, using Grid computing technologies for networked collaboration.*
>
> *CineGrid members explore the feasibility and trade-offs of different networking approaches suitable for emerging applications of rich-media intensive forms of art, entertainment, distance learning, scientific visualization, remote collaboration and international cultural exchange. To support members' research, CineGrid organizes network test beds prepared to host a variety of experimental digital media projects. Designed for very high bandwidth requirements, these network environments provide appropriate security safeguards between a limited numbers of 'trusted' users around the globe.*
>
> *CineGrid periodically organizes inter-disciplinary workshops and demonstrations to share results and identify new avenues of research. Education and training of next-generation digital media professionals is an explicit goal.*
>
> *CineGrid activities are designed to help members foster new user communities that*

are eager to share their cultural, technical and human creativity with colleagues across the globe, linked by CineGrid.

In the Netherlands a consortium consisting of De Waag, UvA, SARA, SURFnet and a number of cultural organizations designed and built an infrastructure to support the CineGrid content in Amsterdam.

3.1.2 CineGrid use case scenarios

There are different use case scenarios: for example storing data on the CineGrid nodes or forwarding data from the CineGrid nodes. The storing of data can be seen in the logical overview in figure 3.1. This is not the real architecture of CineGrid, but an example of a smaller scale.

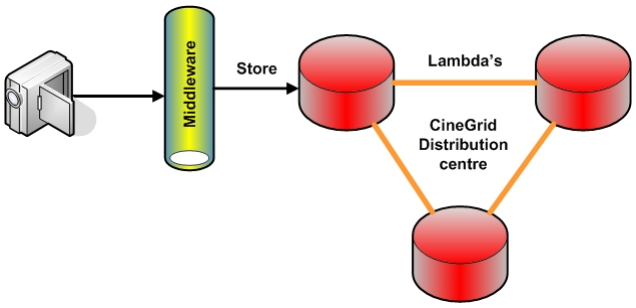

Figure 3.1: Storing data at CineGrid nodes

The idea is that an event is filmed and the gathered data is stored at one of the storage nodes to make it available for editing and viewing at remote locations. Middleware is needed to communicate with the CineGrid infrastructure.

Another use case is the forwarding of data. This can be seen in the logical overview in figure 3.2.

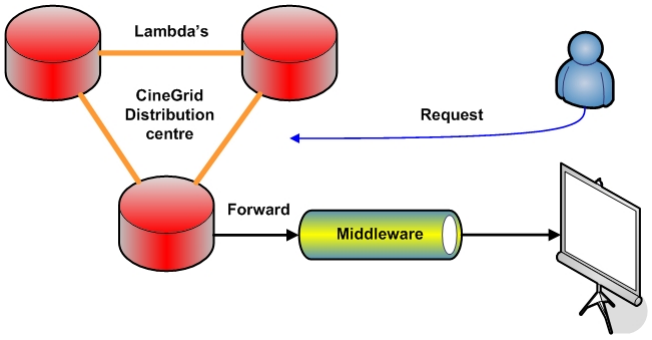

Figure 3.2: Forwarding data from CineGrid nodes to users

The idea is that someone can ask for certain geographically distributed data stored somewhere in the CineGrid architecture. With the help of a distributed information system can be determined from which location in the world the data should be retrieved. Also rendering of the video material is done via middleware.

Another use case scenario is the streaming of video: an event is filmed and the content is sent in real-time to a certain destination and possibly buffered or stored at a store-and-forward server (see CineGrid @ Holland Festival in 3.1.4.

3.1.3 Why CineGrid now?

Quoted from the CineGrid website [1]:

The CineGrid initiative is aligned with three major trends:

- *Spreading deployment of a new generation of 1 Gbps and 10 Gbps digital networks, capable of moving extremely high-quality digital media very quickly between devices, systems, users and collaborators.*
- *Maturing implementation of Grid computing, which fosters the development of software tools to securely manage the high performance distributed workflows needed in support of digital media applications.*
- *Increasing demand for higher quality digital media exchange among remote collaborators in science, education, research, entertainment and art, as well as increased demand for networked distribution of high quality digital media around the world.*

3.1.4 Latest event: CineGrid @ Holland Festival

Quoted (parts) from Calit2.net [2]:

On June 20, 2007, the first successful demonstration of trans-Atlantic streaming over photonic IP networks of 4K digital motion pictures and 5.1 surround sound was achieved by the international research consortium, CineGrid. This demonstration, part of the CineGrid @ Holland Festival 2007 project, was the latest in a series of ground-breaking CineGrid experiments using very high quality digital media running over very high speed digital networks.

CineGrid @ Holland Festival 2007 recorded a performance of "Era la Notte" at the Holland Festival, featuring soprano Anna Caterina Antonacci performing works of Monteverdi, Strozzi and Giramo at the Muziekgebouw aan 't IJ concert hall in Amsterdam. The 75-minute live performance was transmitted nearly 10,000 kilometres, in real-time, to the University of California San Diego where it was viewed in 4K (at four time the resolution of HDTV) on a large screen with surround-sound by an audience in the 200-seat auditorium of the California Institute for Telecommunications and Information Technology (Calit2). Calit2 built the first CineGrid node in North America, fully equipped to handle networked digital media at this extremely high quality.

The 4K transmission from Amsterdam to San Diego on June 20 utilized 4K real-time JPEG 2000 codecs originally designed by NTT Network Innovation Labs to send a compressed 4K x 30 fps stream at bit rates of approximately 500 Mbps.

'To securely store the many terabytes of data recorded each day by CineGrid @ Holland Festival 2007 was a challenge,' said Paul Wielinga, manager of networking at SARA. 'An equivalent of 750 DVDs of data was transferred after each recording from the Muziekgebouw to SARA, where it was copied to two separate high performance storage systems for redundant protection.'

In total about 12 TB was collected in that week.

3.1.5 Relation of CineGrid with UvA

The University of Amsterdam is one of the founding members of the CineGrid project. This information originates from a presentation from Cees de Laat [4]. The UvA has the following roles concerning CineGrid:

- Linking communities.
- The System and Network Engineering group is doing research in the following areas: optical photonic networks, store & forward, DRM / AAA / security and grids.
- Metadata and make it searchable.

3.1.6 Relation of CineGrid with this project

A logical view of activities that are related to CineGrid, can be seen in figure 3.3. This project is about designing the architecture that is needed for a storage system to be used in CineGrid. CineGrid users will have the following demands relating to CineGrid:

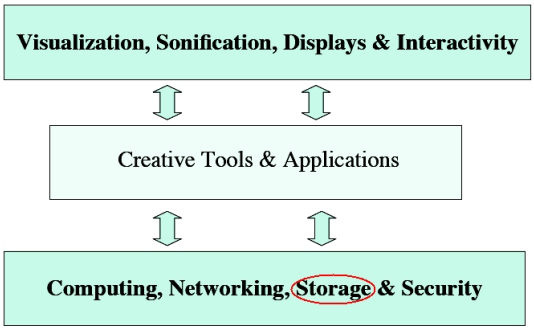

Figure 3.3: Activities related to CineGrid. Source: adjusted from Firstmile.us [3]

- Streaming of content after which it will be discarded. This could be a performance that is encoded and in some cases compressed immediately and streamed to the destination.
- Store the content to forward it at a later time. This could be a performance which is first recorded, then stored at the storage system, and finally is transferred to the destination.

This project focuses on the second user demand: store and forward. To store and forward 4K [A] video material, a lot of space in the first place is needed. Besides that the performance is an issue: the storage architecture must be able to deliver data of large bandwidth rates.

3.2 Physical storage

In this section I will describe the general solutions in the area of physical storage. I will describe Direct Attached Storage (DAS), Network Attached Storage (NAS) and Storage Area Network (SAN). A lot of information is taken from the website Storageresearch.com [12] [18].

In figure 3.4 a total overview of DAS, NAS and SAN is given.
Now each of the three solutions will be described.

3.2.1 DAS

DAS (not to be confused with DAS-3 in 5.3.1) is the most basic level of storage, where the data is stored at the host computer itself, which then also acts as a server. To access data, users have to access the server. Normally, DAS is used with local data sharing requirements. Often this solution is cheaper than the others. In figure 3.5 a DAS topology can be seen.

3.2.2 NAS

The data is stored at a central component in the network: the NAS server. This way the storage management is easier. NFS [A] and CIFS [A] are mostly used as file systems. Clients access the NAS through the NAS head, a 'gateway' that connects all NAS servers. Normally this solution is used to share file-level data across the enterprise. In figure 3.6 a NAS topology can be seen. In this project NAS does not play a significant role, because a NAS can not meet the high performance requirements (see 4.2). This is because the centralized NAS head in combination with Ethernet can not deliver enough bandwidth.

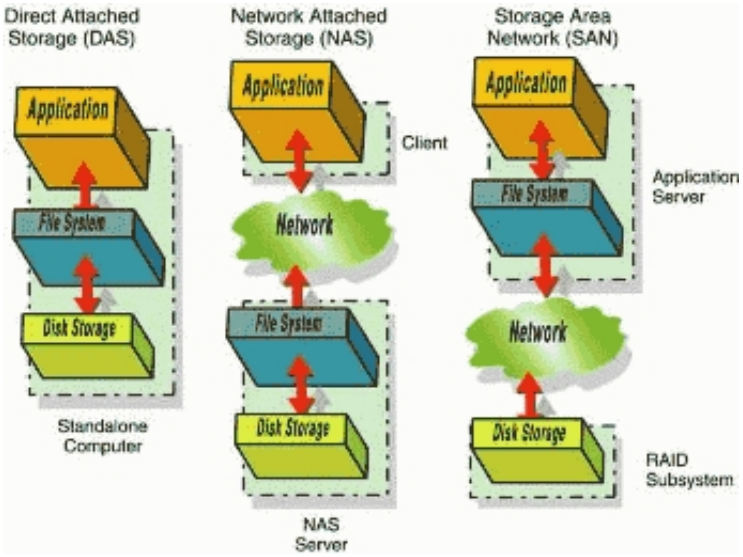

Figure 3.4: Overview of DAS, NAS and SAN. Source: Storageresearch.com [12].

3.2.3 SAN

A SAN is a dedicated, high performance storage network. Normally this network is separate from the local area network. Often fibre channel is used as medium to transfer data. Fibre channel is ideal for moving large volumes of data over long distances. High performance and reliability is achieved. While DAS and NAS are optimized for data sharing at file level, a SAN is optimized for data sharing at block level. This is important for high bandwidth demanding applications and transaction processing. In figure 3.7 a SAN topology can be seen.

3.3 RAID

Redundant Array of Independent Disks (RAID) is a method to store data divided over multiple hard drives. Goals can be higher performance and/or higher reliability of the data. There are different RAID levels: from 0 to 6. There also exist combinations like 0+1 and 5+0. The most used RAID levels are now shortly described:

- RAID-0: known as striping. A minimum of two disks is configured in one array [A]. Data is read and written from and to all disks. This increases performance a lot. Disadvantage is that if just one disk fails, all data on all participating disks is lost.
- RAID-1: known as mirroring. Data is written to two disks: one disk is the backup. Reliability is increased at the cost of half of the capacity.
- RAID-5: parities are used to increase reliability. A minimum of three disks is required. At each participating disk a parity block is used to recover data when a disk fails. For critical applications often RAID-5 is used.
- RAID-6: similar to RAID-5, but at each disk are two parity blocks stored. This means two disks can fail and still all data can be recovered. This increases reliability even more. Disadvantage is that RAID-6 requires more processing power, costs and space.

Another way to increase reliability is a so called ‘hot spare’. This means a pre-installed disk is ready and can immediately replace a malfunctioning disk. With a hot spare it is possible to minimize the time frame a system is vulnerable to a second or third disk that is failing. A hot spare is often used with RAID-5. The RAID-5 system then needs a minimum of time to get the new disk fully synchronized.

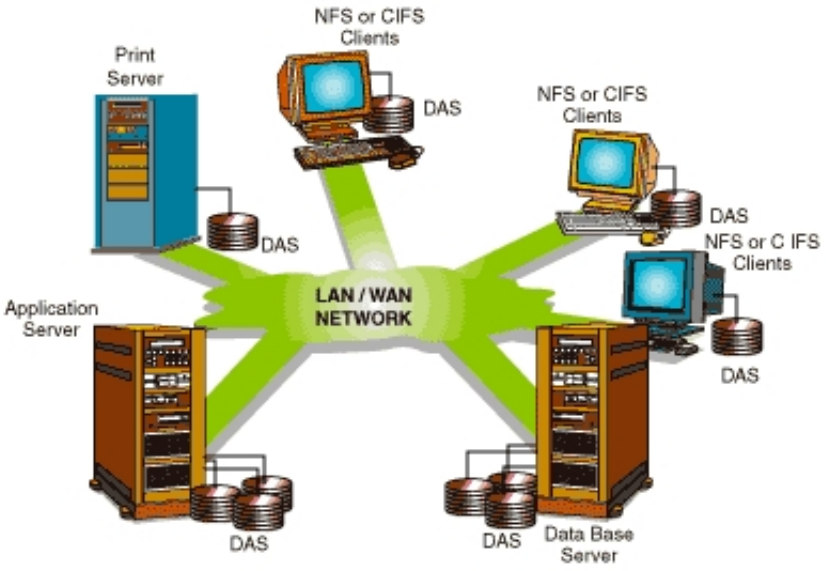

Figure 3.5: DAS topology. Source: Storageresearch.com [12].

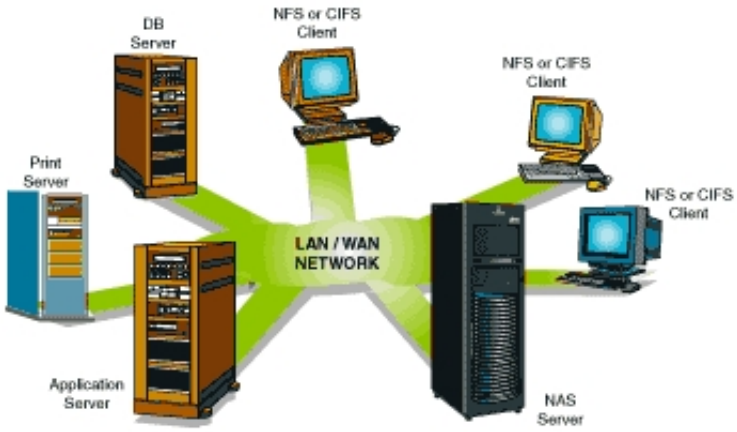

Figure 3.6: NAS topology. Source: Storageresearch.com [12].

3.4 File systems

In this section I will describe some basics about file systems in the context of storage. Information comes partly from Wikipedia [19].

There exist three kinds of file systems:

- Disk file systems: designed for the storage of files on a certain storage device. Examples are NTFS, ext3, ReizerFS and ZFS[1]. Subcategories of disk file systems are:
  - Solid state media file systems: file systems especially for solid state media like Flash memory. Error detection and correction algorithms are needed. Examples are FAT, JFFS2 and YAFFS.
  - Record-oriented file systems: meant for mainframe and minicomputer systems. Files are stored as a collection of records. Examples are Files-11 and VSAM.

[1]Though ZFS supports dynamic striping across multiple devices, it still remains a local file system, because concurrent access from multiple hosts is not supported.

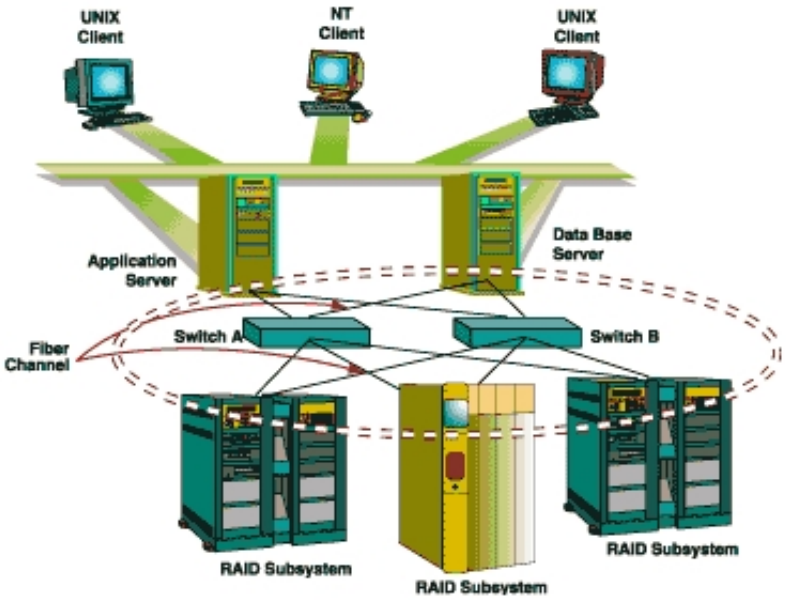

Figure 3.7: SAN topology. Source: Storageresearch.com [12].

  - Shared disk file systems: mostly used in a storage area network where all nodes directly access the block storage where the file system is located. Examples are EMC Celerra HighRoad, GFS, GPFS, OCFS, SAN file system and VMFS3.

- Distributed file systems: a network file system. There are multiple and independent storage devices, which look like they are in one place to the clients. Examples are OpenAFS, AFP and NFS. Subcategories of distributed file systems are:
  - Distributed fault tolerant file systems: they replicate data between nodes. Examples are Coda and DFS.
  - Distributed parallel file systems: they stripe data over multiple servers for higher performance. Examples are Ceph, Lustre and PVFS.
  - Distributed parallel fault tolerant file systems: they combine the two file systems above. Examples are Gfarm file system, GlusterFS, Google File System, PeerFS and TerraGrid.
- Special purpose file systems: they have special purposes like encryption, fault tolerance or a virtual file system. Examples are FUSE [A], EncFS and Callback File System.

In this project the focus is on distributed (fault tolerant) parallel file systems, because they achieve high performance.

Chapter 4

Research

The following topics will be described in this chapter: the current storage implementation, the criteria for the new storage architecture and a comparison between the found file systems for the new environment.

4.1 Current storage implementation

In this section the current storage implementation will be described and why this solution is not sufficient.

Before looking to the current storage implementation, I give an overview of the network topology to see where the storage part fits (see figure 4.1).

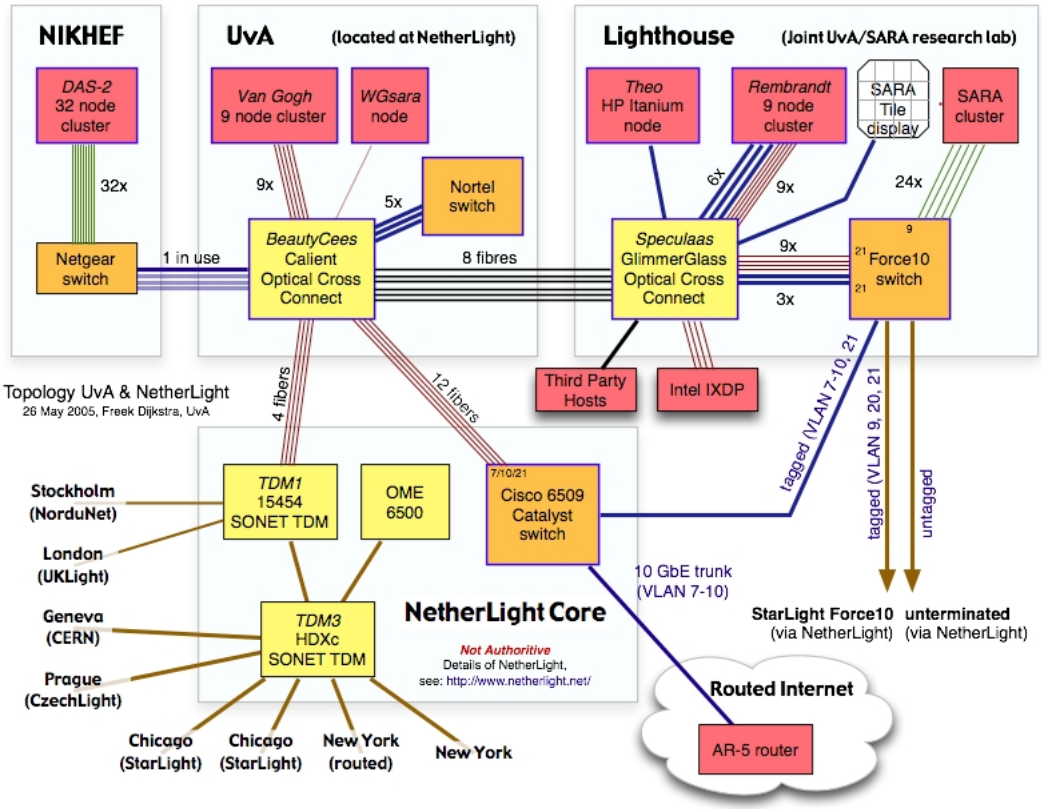

Figure 4.1: Network topology Advanced Internet Research Group. Source: Freek Dijkstra, SNE (formely AIR) [13].

The storage as it was used at the Holland Festival 2007 (described in 3.1.4) is located in the *Lighthouse*: the *Rembrandt* cluster. The *Rembrandt* cluster consists of 9 nodes: 8 of them are used for CineGrid content. Each node has a 1 Gb/s and a 10 Gb/s connection to the GlimmerGlass switch. There is a 10 Gb/s link from the GlimmerGlass switch to the SARA tile display for displaying the content. The same connections are linked to the *Force10* switch. From there the content is routed to the internet through the Cisco 6509 switch. In figure 4.2 the components related to CineGrid can be seen.

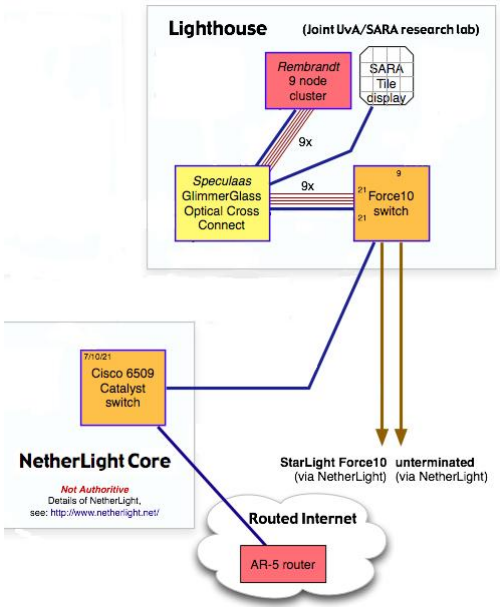

Figure 4.2: Network topology CineGrid components. Source: adjusted from Freek Dijkstra, SNE (formely AIR) [13].

Now that the location of the storage cluster is known, a deeper look into it can be given. Parts of the following text about the *Rembrandt* cluster are quoted from the SNE website [16]:

The nodes contain the following:

- Computing power: each node has two 64-bit Opteron processors 2.0 GHz, and 4 GB memory.
- Storage: Besides a 80 GB system disk, all nodes have a Serial ATA RAID system. Rembrandt0 comes with eight 250 GB drives, totalling 2 TB storage using RAID 5. The other nodes have eleven 250 GB disks, totalling 2.3 TB storage using RAID 0.

Nodes 1 to 8 are used for CineGrid content. On each node 2350 GB can be used as storage. This means a total amount of 18,4 TB. The 10 Gb/s link comes from node 1. As software *GlusterFS* is tried. "GlusterFS is a clustered file-system capable of scaling to several petabytes. It aggregates various storage bricks over Infiniband RDMA or TCP/IP interconnect into one large parallel network file system. GlusterFS is based on a stackable user space design without compromising performance."[1] But GlusterFS never worked in a stable way at the Rembrandt cluster. This was caused by a crashing node that caused GlusterFS to stop. The assumption is that this problem was related to the 10 Gb/s interface and there actually was a driver problem or a limitation in the bus architecture, but this has not been tested. At the moment GlusterFS is not used anymore. Instead, all participating nodes are configured with RAID-5 arrays and all files are copied manually via FTP.

About the files and size of the files: the extension of the files is *umf*. This is a proprietary video format used to store the 4K video content. In section 4.2 more can be read about umf and its needed bandwidth. The files vary in size from approximately 0.4 TB to 1.6 TB.

[1] Website GlusterFS: http://www.gluster.org/

Although the *Rembrandt* cluster at the moment is functioning as the storage component, this is not a preferred configuration. All files have to be copied manually through FTP. This is not easy to administer. Other issues are the performance and maximum file size on the *Rembrandt* cluster. Therefore a storage architecture is needed. Researching the GlusterFS problem or fixing it is not a part of this research. In the following section the criteria for this storage architecture are described.

4.2 Criteria

To design a storage architecture, a list of criteria must be made. The following criteria are important: bandwidth, IOs per second (IOPS), budget, reliability, space, scalability, administration, response time, complexity, maturity and flexibility. These criteria are formulated together with the supervisors of the project. Now each of these criteria will be determined.

Bandwidth

Bandwidth is an important criterion in this project. To determine the bandwidth that is needed, we should look at the material that needs to be transferred: that would be 4K video. To calculate the bandwidth, we used the following parameters:

- HZP (Horizontal pixels per frame) = 4096
- VTP (Vertical pixels per frame) = 2160
- CBP (Colour bits per pixel) = 3 bytes (Red, Green, Blue) = 3 x 8 bits = 24 bits
- FPS (Frames per second) = 30

Now we can calculate the bandwidth:

$$Bandwidth(Gb/s) = (HZP * VTP * CBP * FPS)/(10^9) = (4096 * 2160 * 24 * 30)/(10^9) = 6.4\ Gb/s$$

Sound is not taken into account in this calculation!
About compression (see related work about DXT compression [6]): the bandwidth of 6.4 Gb/s is a minimum requirement. There must be some headroom and the choice for multiple streams in the future should be an option. This project is about the storage architecture: the data has to be stored in real-time. Compression is a possibility in a later stage and is not part of this project.
Note: b means bit and B means Byte (8 bits).

IOPS / block size

The number of IOPS [A] (IOs Per Second) is not really important: bandwidth is far more important. However, something can be said about this topic.
For calculating the number of IOPS we use the following formula [17]:
IO/s per disk x No. of disks = Bandwidth / Block size
The bandwidth is already calculated: 6.4 Gb/s or to be more precise: 6370099 Kb/s = 777600 KB/s. Normally the block size [A] is dependent on the used application which reads the content from the storage. But in this project the only thing that is important is storing large amounts of digital media data at a very high speed (real-time). This means a large block size is preferred over a smaller block size. The larger the block size, the higher the bandwidth can be, but the lower the number of IOPS [11]. Since IOPS is not as important as bandwidth in this project, this is not a problem.
We can use the formula above to calculate the minimum block size that has to be used when the disk configuration is known. The block size would be: Block size = Bandwidth / (IO/s per disk x No. of disks) = 777600 KB/s / (IO/s per disk x No. of disks).
Note: b means bit and B means Byte (8 bits).

Budget

Although there is no real budget given for this project, the estimated costs for the new storage architecture should be at a maximum of approximately 50,000 Euro including network costs. This budget is indicated by the supervisors.

Reliability

The storage system has to be stable. This means robustness: it has to be able to cope with one or more failing components. The system has to be as robust (or more) as a RAID-5 configuration.

Space

To determine the needed space, we need the calculated bandwidth and the duration of the video material. To calculate the space of one hour of 4K video material, we use the following formula:

$$Space(TB/hour) = (Bandwidth(Gbps) * 10^9 * Time(s))/(8 * 1024^4) =$$

$$(6.4 * 10^9 * 60 * 60)/(8 * 1024^4) = \textit{2.6 TB/hour}$$

This is the space that is needed for the proprietary umf format, which was talked about in 4.1. If a two-hour movie has to be stored, then 5.2 TB would be needed. Normally in each year four to six performances are stored. This means a total of 6 * 5.2 TB = 31.2 TB/year.
Note: b means bit and B means Byte (8 bits).

Scalability

The storage system needs to be scalable. This means it has to be possible to expand it in such a way it supports more space and higher performance. For the space this means it has to be expandable for another year, which means around 60 TB would be needed in total.

Administration

The storage system must be able to cope with central administration. A lot of servers that have to be managed in a decentralized way is not preferred.

Response time

The response time of the storage, while clients access it, has to be as low as possible. One way to realize this is to separate the data and the metadata: this way the traffic can be managed more efficiently and response times can be lower.

Complexity

The complexity of the solution has to be as low as possible, but the other criteria have to be maintained first. This involves the number of servers and disks, but also the needed expertise. Possible problems and how they should be solved can also add in complexity.

Maturity

It is preferred that the solution is mature, meaning it is proven in other environments. A product that is at the start of development, is not considered mature. Due to the other criteria, such as bandwidth this may not be possible.

Flexibility

The solution has to be as flexible as possible (looking at the other requirements). This means different operating systems, hardware and network configurations can be used with the same storage architecture.

As a summary all criteria are given in the table in figure 4.3. Now that the list of criteria is known, a comparison can be made between possible candidates for the new storage architecture.

Criterion	How?
Bandwidth	>= 6.4 Gb/s
IOPS / block size	Dependent on disk configuration
Budget	50,000 Euro
Reliability	>= RAID-5
Space	31.2 TB/year
Scalability	Expand space and performance
Administration	Central
Response time	As low as possible
Complexity	As low as possible
Maturity	Proven technology
Flexibility	Different software and hardware

Figure 4.3: Criteria for the new storage architecture

4.3 File system comparison

To begin, I will compare different Distributed Parallel File Systems (DPFS). Later I will propose a solution for the location of the data and a way to implement reliability as introduced in the beginning of chapter 3. I choose for a DPFS because performance is the key in this project and a DPFS can deliver performance. I choose to compare the file systems first, because each of them has a lot of different properties, which may influence the total architecture. First I give a short description of each found file system [8]. After that, the file systems will be held against the described criteria. Finally the architectures for the best file systems are described.

4.3.1 File systems

- Ceph: a distributed, parallel, fault tolerant file system by the Storage Systems Research Center at the University of California, Santa Cruz. Fault tolerance is realized by data replication. When new nodes join the cluster, all data is automatically distributed to the new nodes. Ceph is installed at kernel-level. Although Ceph looks promising, it is still a prototype at this moment. Ceph is open source and falls under the GPL license. Website: http://ceph.sourceforge.net/.

- Gfarm Grid File System: a distributed, parallel, fault tolerant file system by ApGrid. Gfarm is installed at user-level with the help of FUSE. Gfarm has a stable version, but there are known bugs and the program still is in development. Gfarm is open source and falls under the X11 license. Website: http://datafarm.apgrid.org/.

- GFS: Global File System by Red Hat. GFS is a shared disk file system available with Red Hat cluster suite. Only a SAN and only Linux Red Hat is supported. GFS is open source and falls under the GPL license. Website: http://www.redhat.com/gfs/.

- GlusterFS: a distributed, parallel, fault tolerant file system by Z RESEARCH. GlusterFS is installed at user-level on top of an existing file system. This makes it very flexible in use. No single point of failure (SPOF) exists and all metadata is handled by the underlying file system. Performance is good. Installation is easy. GlusterFS is open source and falls under the GPL license. Website: http://www.gluster.org/.

- Google File System: a distributed, parallel, fault tolerant file system by Google. This file system is only used internally at Google. Website: http://labs.google.com/papers/gfs.html.

- GPFS: General Parallel File System by IBM. GPFS is a proprietary shared disk file system. Website: http://www.ibm.com/systems/clusters/software/gpfs.html.

- Hadoop Distributed File System: a distributed, parallel, fault tolerant file system by Apache. Hadoop DFS runs on JAVA. This software is open source and the license is free. Website: http://lucene.apache.org/hadoop/.

- Lustre: a distributed, parallel file system by Cluster File Systems. There is no replication of data: reliability must be handled in external hardware. Performance is good. This product is very mature and used on several supercomputers. Documentation and support are good. Lustre is installed at kernel-level. The software still is open source, but it may be possible proprietary add-ons will be developed in the future. Website: http://www.lustre.org/.
- NFS: Network File System. A distributed file system that comes with the Linux kernel. Performance is less than with distributed parallel file systems. NFS is open source and falls under the GPL license. Website: http://nfs.sourceforge.net/.
- OpenAFS: Open Andrew File System by IBM. This is a distributed file system with read-only replication. Limits with OpenAFS are not as high as with the other file systems: 8 GB per volume is supported and the maximum file size is 2 GB. IPv6 is not supported. OpenAFS is open source and falls under the IBM Public License. Website: http://www.openafs.org/.
- OCFS: Oracle Cluster File System. OCFS is a shared file system by Oracle. OCFS is open source and falls under the GPL license. Website: http://oss.oracle.com/projects/ocfs2/.
- PeerFS: a distributed, parallel, fault tolerant file system by Radiant Data Corp. In the area of reliability the software only supports mirroring. PeerFS is proprietary software. Website: http://www.radiantdata.com/English/Products/PeerFS.html.
- PVFS: Parallel Virtual File System. A distributed, parallel file system by several contributors. PVFS is installed at user-level. The software is open source and falls under the GPL license. Website: http://www.pvfs.org/.
- TerraGrid: a distributed, parallel, fault tolerant file system by Terrascale. This software is proprietary. Development is not active anymore. Website: http://www.terrascale.com/.

4.3.2 Comparison

From the found file systems, the following eight will be compared against the criteria: Ceph, Gfarm Grid File System, GFS, GlusterFS, Hadoop Distributed File System, Lustre, OpenAFS and PVFS. I choose to only compare open source distributed parallel file systems, because this is one aspect of flexibility, which is an important criterion in this project. Installations of the file systems were not possible due to limited time. They will be compared against the following criteria: bandwidth, block size, budget, reliability, space, scalability, administration, response time, complexity, maturity and flexibility. Comments may be added. The file systems are only compared in theory by searching on the internet for information and user experiences. An installation of the file systems may give more accurate and maybe different results.

Below a summary is given of found articles (excluding the file system's manuals) concerning distributed parallel file systems and their performance:

- The article "Building A High Performance Parallel File System Using Grid Datafarm and ROOT I/O" [21] covers some performance evaluations about Gfarm. A bandwidth of 350 MB/s, which is around 2.7 Gb/s can be found here.
- The article "Wide Area Filesystem Performance using Lustre on the TeraGrid" [20] covers a performance evaluation about Lustre. Using two connections a speed of 8 Gb/s has been accomplished.
- The article "Scalable Security for High Performance, Petascale Storage" [22] covers the impact of security measures on the performance of Ceph. A bandwidth of 350 MB/s for 10 nodes is presented, which is around 2.7 Gb/s.
- The article "PVFS: A Parallel File System for Linux Clusters" [23] covers performance tests with PVFS. A bandwidth of 700 MB/s with Myrinet is accomplished, which is around 5.5 Gb/s.

- The article "10Gb Ethernet with TCP Offload - The High Performance Interconnect" [24] covers performance tests about (among others) PVFS. A bandwidth of 1700 Mb/s is accomplished, which is around 1.7 Gb/s. Besides that, TCP Offload Engine (TOE) tests are shown, which may be interesting for higher performance when a file system is implemented through Ethernet.

It is clear that some of the found articles are outdated. The only thing that can be concluded is that the found values are a minimum of what the file systems can deliver. At the websites of GlusterFS and Lustre very high bandwidths of 130 Gb/s are given. It may be possible that the other file systems can also deliver these bandwidths, but I have not found results about this.
In the table in figure 4.4 the results can be seen.

Criterion/Product	Ceph	Gfarm	GFS	GlusterFS	Hadoop	Lustre	OpenAFS	PVFS
Bandwidth	+ 2.7 Gb/s	+ 2.7 Gb/s	+ 6.5 Gb/s	130 Gb/s	+ 1 Gb/s	90% of raw bandwidth, max 130 Gb/s	Unknown	+ 5.5 Gb/s
Block size	1 MB	8 KB?	8 KB	1 MB	+ 64 MB	1 MB	Unknown	64 KB?
Budget	Free	Free	Free	Free	Free	Free (still)	Free	Free
Reliability	Replication of data	Replication, but not auto	No replication	Replication; no SPOF	Replication of data	Later implemented	No replication	No replication
Space	Petabytes	Petabytes	16 TB 32 bit 8 EB 64 bit	Petabytes	Petabytes	+ 32 PB	8 GB/volume, max file size is 2 GB	Petabytes
Scalability	Data: auto migration to new nodes	Manually?	New node: reboot LVM	New nodes: automatic after reboot	New nodes not used for old files	Easy and linear	Unknown	New node: stop servers
Administration	Central	Central	Central	Central	Central	Central	Central	Central
Response time	Separation of data and metadata	Separation of data and metadata	Shared FS	No metadata: handled by normal FS	No separation of data and metadata	2 Central metadata servers	No separation of data and metadata	Separation of data and metadata
Complexity	Reasonable	Easy install	Easy install	Easy install	Easy install	Complex	Reasonable	Reasonable
Maturity	Prototype, promising; not many doc.	Stable, but development	Mature, built in RedHat	Stable, but development	Development just started	Very mature, development, good support and doc.	Development, old doc.	Stable, but development
Flexibility	Commodity hardware	Commodity hardware	Only RedHat, only SAN	On top of existing FS, Infiniband	Linux and Windows	Infiniband, Myrinet, FC, SAN	No IPv6	Infiniband, Myrinet
Comments	Kernel-level	User-level	Kernel-level	User-level	Runs in JAVA	Kernel-level	Kernel-level	User-level

Figure 4.4: Comparison of Distributed Parallel File Systems

Note about the results: all values are either found in the manual or in articles. The current version of all file systems may actually generate different results. Furthermore the criterion *reliability* is implemented using RAID. However, some of the file systems are fault tolerant, which means they support another form of reliability: replication of data among the nodes. To be complete, I choose to indicate what form of reliability the file systems support. RAID is chosen as reliability mechanism because it is implemented at a lower level than file system security. This is considered more reliable and is proven.
Based on the table in figure 4.4, Lustre and GlusterFS rank as the best DPFSes. They are the only two file systems that can deliver the bandwidth requirement (based on found articles). Lustre is very mature and has good documentation and support. It is used in many other environments successfully. Furthermore it is a flexible product: many technologies are supported, like installation with a SAN and native Myrinet. Performance is good and most criteria are met. On the other hand Lustre is difficult to install: a specific Linux kernel is needed.
GlusterFS is still in development, but has a stable version. Big advantage is the ease of installation. This is done at user-level on top of the existing file system. Performance is good and most criteria are met.
An alternative is PVFS: the found performance results look promising. Advantage is that PVFS supports native Myrinet (GlusterFS does not), meaning the file system can directly talk to Myrinet. An other possibility may be to use IP over Myrinet.

4.3.3 Lustre and GlusterFS architectures

In this section the architectures for Lustre and GlusterFS are made clear.

Lustre

A Lustre file system consists of four components:

- Management Server: defines information for all Lustre file systems at a site.
- Meta Data Target: provides back-end storage for metadata.
- Object Storage Targets: provides back-end storage for file object data.
- Lustre clients: the "users" of the file system.

An example of an architecture containing Lustre (copied from the manual) can be seen in figure 4.5. Furthermore it is important that the used configuration matches the following:

- Operating systems: Red Hat Enterprise Linux 3+, SuSE Linux Enterprise Server 9 and 10 or Linux 2.6 kernel.
- Platforms: IA-32, IA-64, x86-64, PowerPC architectures or mixed-endian clusters.
- Interconnect: TCP/IP, Quadrics Elan 3 and 4, Myrinet, Mellanox or Infiniband.

GlusterFS

An architecture containing GlusterFS contains servers and clients. The servers are providing access to the data and the clients are accessing the servers for the data. There is no metadata server: the metadata is handled by the underlying file system. An example of an architecture containing GlusterFS (copied from the website) can be seen in figure 4.6. Furthermore it is important to match the following configuration:

- On the client side FUSE (Filesystem in userspace) kernel support is needed.
- Operating system: GlusterFS server can run on any POSIX compliant OS. Linux, FreeBSD, OpenSolaris and Mac OS X are supported.
- Interconnect: TCP/IP or Infiniband.

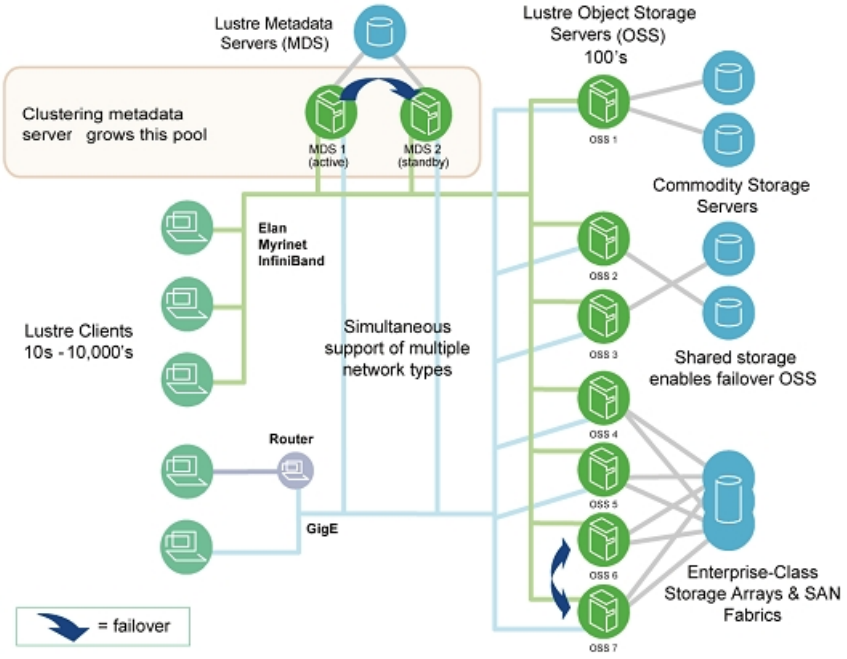

Figure 4.5: Example of Lustre file system architecture. Source: Lustre manual.

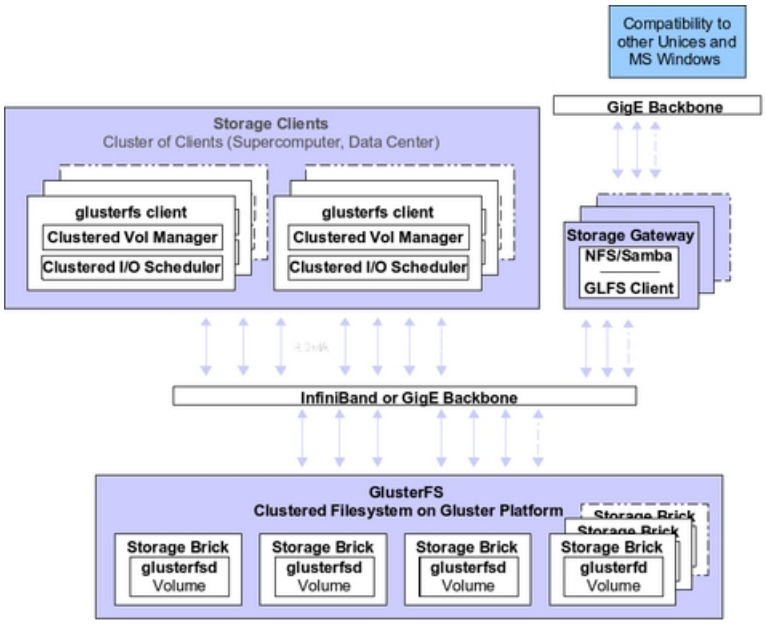

Figure 4.6: Example of GlusterFS file system architecture. Source: GlusterFS website.

Chapter 5

Store-and-forward architectures

Of course, the delivered performance is not only dependent on the used file system, but on the total storage architecture. Now a couple of total architectures will be proposed and in the end one will be proposed as solution based on the criteria. In these architectures the three components from chapter 3 will play a role: the physical storage location, the reliability mechanism (RAID) and the file system.

5.1 Architecture 1

This architecture consists of multiple cluster nodes, which are connected through a high speed network. This can be Ethernet, Infiniband or Myrinet. The idea is to install a distributed parallel file system at the clusters. This could be GlusterFS or Lustre. Native Myrinet is not supported by GlusterFS. In the case of Lustre there could be installed two metadata servers nodes. Reliability can be realized by adding RAID controllers to the servers. They should at least support RAID-5, but it is recommended (for future use) to support RAID-6. RAID-5 with hot spares is a good alternative. RAID-6 is not necessarily: if we took a MTBF [A] of 1000 days for the used disks, it only begins to become a must to use RAID-6 with very large requirements for space (petabytes). Quick calculation: with a repair time of a disk close to 1 day, the expected failure rate with 100 disks (we need less) is 100/1000 = 0,1 disk per day. If we would need 2000 disks in a petabyte environment, then the expected failure rate would be 2000/1000 = 2 disks per day. The disks should be for example SATA-2 or SAS 1 TB disks: at least they should be fast enough and large enough. In the current situation 8 nodes are dedicated for the use of CineGrid. If we continue this idea, then in each node can exist 8 disks in hotswap [A] containers, which means 8 TB per node and a total of 64 TB. Because RAID-5 is used actually 56 TB of space can be used: one RAID array contains 8 nodes so 1/8 of total space is used for parities. As also been said in the Lustre manual, it is best to achieve high performance (bandwidth) by using a block size of 1 MB. Additional nodes could be added for maybe higher performance. Furthermore: the nodes need to have fast hardware: a high speed NIC (at least 1 Gb/s), a modern CPU and enough memory.
Figure 5.1 shows this solution. Advantages of this architecture:

- Costs can be relatively low, because a SAN can be avoided.
- In the case of GlusterFS: installation is very easy. GlusterFS installs on top of an existing file system at user-level.
- In the case of Lustre: if a SAN would be used in the future, then Lustre can be maintained. Furthermore Lustre is more mature and support is good.

Disadvantages of this architecture:

- Data is not physically stored in a central place like a SAN. Data is not separated from the servers. These topics make administration less easy in the future.
- In the case of GlusterFS: this software is still in early development. This may cause problems.
- In the case of Lustre: installation is less flexible. There are more components involved and Lustre must be installed at kernel-level.

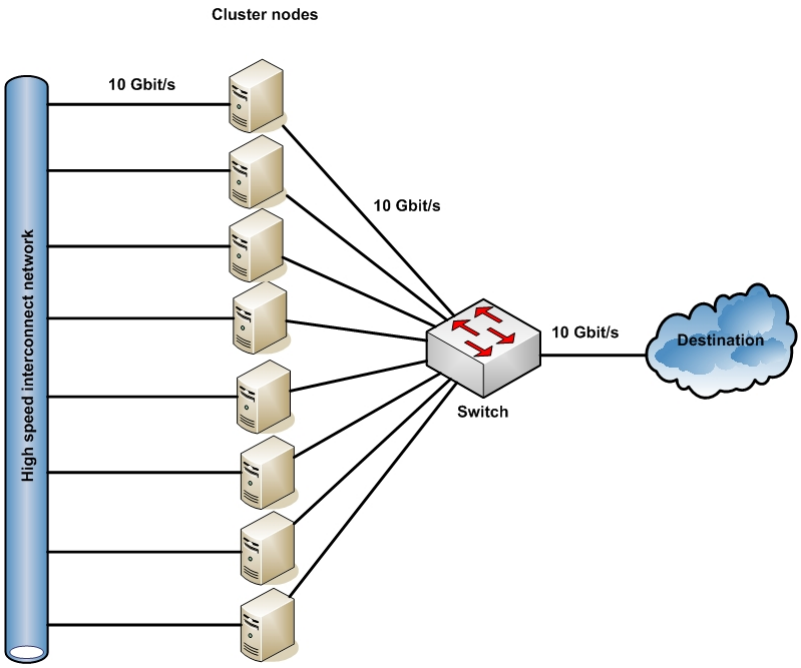

Figure 5.1: Cluster architecture

The following costs play a role in this architecture:

- The cluster nodes.
- Hard disks: 64 x 1 TB disks.
- RAID controllers: 8 x 8 ports supporting RAID-5 and preferable RAID-6.
- In the case of Lustre: support.

5.2 Architecture 2

This architecture also consists of multiple cluster nodes, connected through a high speed network like Ethernet, Infiniband or Myrinet. The data is stored at a SAN. This SAN should have enough ports for the participating servers at 4 Gb/s (fibre channel). The SAN should also have enough space: at least 30 TB and it must be able to scale to at least 60 TB. The servers are connected to a SAN switch having HBA-interfaces. This should be HBA-interface cards that could deliver high bandwidth. The idea is to install Lustre in this architecture, because Lustre has support for SANs. 8 Nodes are the Object Storage Servers (they may be extended with more nodes). Tests [20] showed that 8 nodes give good results in comparison with less nodes and similar results in comparison with 16 nodes. 2 Nodes are meant as Meta Data Servers. They should contain a RAID-5/6 controller and enough hard disk space. For high performance a block size of 1 MB should be used. This architecture has similarities with the architecture from TeraGrid [20]. The nodes need to have fast hardware: a high speed NIC (at least 1 Gb/s), a modern CPU and enough memory.
Figure 5.2 shows this solution. Advantages of this architecture:

- A SAN is used. This means data is separated from the servers, which makes future administration easier.
- Lustre is used, which means more mature software and good support.

Disadvantages of this architecture:

- Costs are higher because a SAN has to be bought.

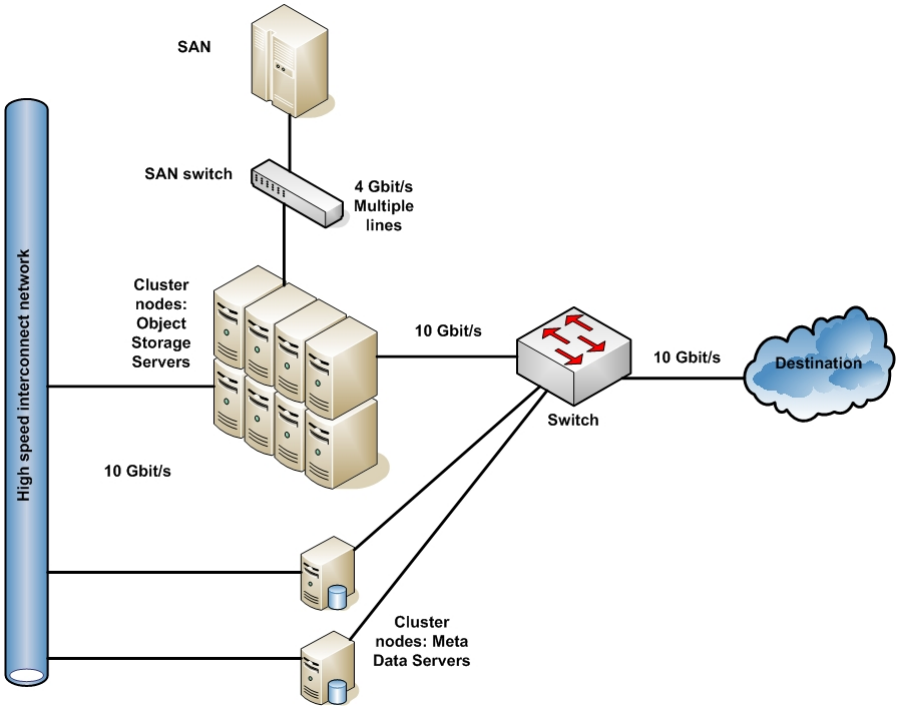

Figure 5.2: SAN together with cluster architecture

- Lustre is used, which means a more difficult installation.

The following costs play a role in this architecture:

- The SAN: this would be the biggest part of the costs. It is not certain whether it is possible to stay within the budget and meet the SAN requirements.
- The cluster nodes.
- The HBA-interface cards for the servers.
- RAID-controllers and hard disks for the Meta Data Servers.
- Support for Lustre.

5.3 Implementation

Before recommending one of the architectures for the CineGrid store-and-forward servers in Amsterdam, some consequences associated with the implementation must be highlighted. The following topics will be looked at: the cluster nodes, a SAN alternative and the interconnect network.

5.3.1 Cluster nodes

In both architectures cluster nodes must be available. One possibility is to buy new nodes, but another is to use existing nodes. At the moment the UvA can use part of the Distributed ASCI Supercomputer 3 (DAS-3) cluster [15]. This is a cluster that is used for all kinds of intensive processing tasks. The UvA can use 41 1U [A] nodes and has an additionally 8 4U nodes. An advantage of using these nodes is reduced costs. However: extra hard disk space still is needed. A disadvantage of using these nodes may be less flexibility: it is not preferred that the nodes are re-installed with a new OS/kernel. This would be the case with Lustre, because it needs an adjusted kernel. The alternative would be to use GlusterFS which can be installed at user-level on top of an existing file system without the need for an adjusted kernel.

5.3.2 SAN alternative

Although this research does not cover a detailed SAN product comparison, the idea came up to use some kind of alternative for a SAN: the Sun Fire X4500 server[1], also known as Thumper. This is a device running Solaris 10 and is powered by two dual-core Opteron processors. This device is not suited as an alternative for a SAN, because it does not meet all criteria. Storage capacity is only scalable to 24 TB, which is not enough. It has four 1 Gbit Ethernet ports, which is not enough to deliver a bandwidth of 6 Gb/s.

5.3.3 Interconnect network

As interconnect network can be chosen between Ethernet, Infiniband or Myrinet. At the moment the DAS-3 cluster described above already can use an existing 10 Gb Myrinet infrastructure. Advantage is a reduction in costs. Disadvantage is less flexibility: GlusterFS does not support the use of native Myrinet. An alternative is to use (the cheaper) Ethernet. Advantage is increased flexibility: interconnectivity with other infrastructures may be easier and GlusterFS is supported. Disadvantage is less performance than for instance Myrinet. Ethernet is not suited for very high bandwidth demands, because of the generated processing overhead of the network stack. Myrinet is designed to overcome this problem.

There exists another technology: the TCP Offload Engine (TOE). This is a technology used in network interface cards to offload processing of the entire TCP/IP stack to the network controller. Test results [23] show that much higher speeds can be achieved. Though it may not be necessary to use this technology - maybe the bandwidth can be high enough with the distributed parallel file system - it could be helpful when the limits of Ethernet are reached.

5.4 Result

If budget was not an issue, then architecture 2 would be the optimal solution. With the SAN it is very easy to expand the storage capacity and because the data and the nodes are separated, administration is easier. When Lustre would be used, a very mature and proven distributed parallel file system is chosen. Performance requirements can be handled. This architecture has proven itself and can deliver the performance requirements [20].
But as always, budget actually is an important criterion. With a budget of 50,000 Euro the second architecture is no option. The SAN would be the largest part of the costs and besides that it may be necessary to buy new nodes, because Lustre needs an adjusted kernel.
Because of the budget we would have to choose for architecture 1. However, there still are several questions when implementing architecture 1:

- Which file system is used: GlusterFS or Lustre?
- Is the DAS-3 cluster used or should there be bought new nodes?
- Is Myrinet or Ethernet (with maybe TOE) used?

To answer these questions, we again have to look at the budget. The common hardware that in any case has to be bought is hard disk space: 8 x 8 x 1 TB, which means 64 disks and appropriate RAID controllers. This would be a minimal of 25% of the budget if we look at today's prices. This means there would be enough of the budget left for buying new nodes: now the budget is no longer the key criterion.

Now we can look at the most important differences between GlusterFS and Lustre. In short: GlusterFS is very easy to install, but less mature and does not support native Myrinet. GlusterFS is flexible because it can be installed at user-level on top of an existing file system. Lustre is harder to install, but very mature and it does support more technologies like native Myrinet and SAN (for maybe future use). Lustre is less flexible, because it needs an adjusted kernel. Furthermore it is likely when choosing Lustre, new nodes had to be bought considering the installation of Lustre. Performance should be comparable.

[1] http://www.sun.com/servers/x64/x4500/index.xml

For the near future I recommend to install GlusterFS on the DAS-3 nodes using Ethernet. This way no new nodes have to be bought and money does not have to be spent for investments that maybe are not necessary. When GlusterFS is installed, it is possible to have quick results and do more tests.
For the further future I recommend to make a new decision based on the test results with GlusterFS on the DAS-3 nodes. It may turn out that GlusterFS contains too many bugs, which makes Lustre the better choice. Another conclusion may be that the performance through Ethernet cannot be achieved, which means it may be better to use Ethernet with TOE or Lustre with Myrinet.
In figure 5.3 the architecture can be seen.

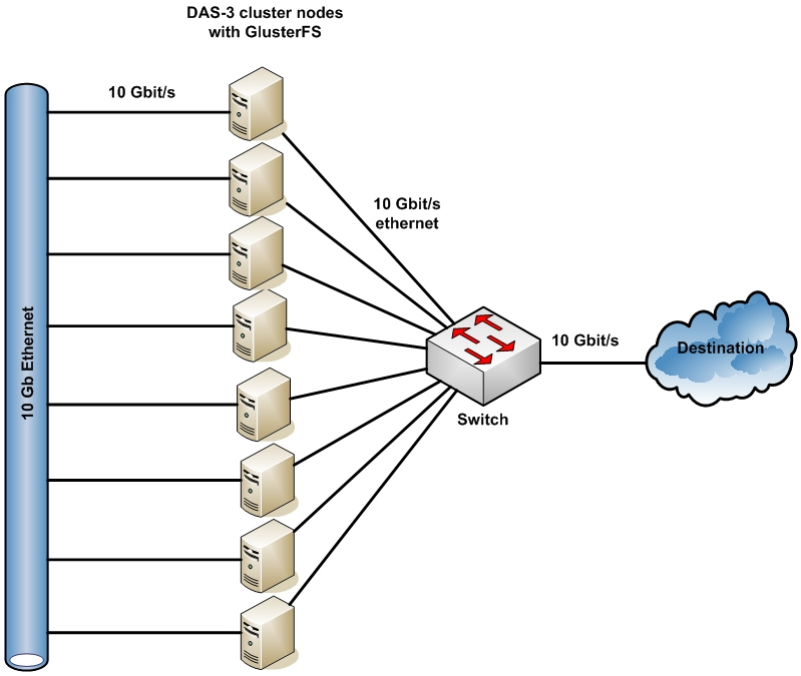

Figure 5.3: DAS-3 cluster with GlusterFS

Note: although native Myrinet is not supported by GlusterFS, another possibility is to use IP over Myrinet. Performance of such a configuration should be tested.

Chapter 6

Conclusion

The research question was the following: *"What is the optimal architecture for the (CineGrid) storage systems that store and forward content files of a size of hundreds of GBs?"*. I answered this question in specific for the store-and-forward servers of the CineGrid collaboration project in Amsterdam.

After doing some basic research about physical storage solutions and file systems, several things became clear. I made a comparison between distributed parallel file systems. For each found open source distributed parallel file system some criteria are checked, which are mentioned before. The best file systems according to these criteria are Lustre and GlusterFS. Each of them has its own advantages and disadvantages. Lustre is very mature, has proven itself to be reliable and deliver good performance in other recent projects [20]. It also contains good documentation and support. Many technologies like SAN and native Myrinet are supported. Disadvantages are that Lustre is difficult to install because it needs a specific adjusted kernel. GlusterFS is a good competitor to Lustre: performance tests showed similar results as with Lustre. There is a stable version, but GlusterFS is still in active development and not as mature as Lustre. Advantage is the ease of installation, because GlusterFS can be installed at user-level on top of an existing file system.

After that a comparison between architectures is made. The first architecture consists of multiple cluster nodes with Lustre or GlusterFS installed. The second architecture consists of a SAN and multiple cluster nodes with Lustre installed. RAID-5 with a hot spare is used for reliability. A total of around 60 TB of disk space is needed.
When budget would be not an issue, then the second architecture is recommended. This kind of architecture is proven already and meets the performance requirements. Advantage is the use of a SAN, which makes administration easier. Data and nodes are separated. But the SAN is the most expensive component, which is the cause for recommending the first architecture. For the near future it is advised to install GlusterFS on the DAS-3 [15] nodes with Ethernet as interconnecting network. This way money can be saved, because the DAS-3 nodes already exist. Installation of GlusterFS is easy. Further tests have to be done to determine whether the performance requirements are met. After these tests a new proposal could be given for the further future.

Chapter 7

Future work

In relation to this project, there are several topics for future work:

- Implement the architecture. It is likely certain problems arise, that are not covered in the architecture. With several tests it must become clear whether the recommended architecture can be implemented or adjustments have to be made.
- Analyze what impact several security measures, like authentication and authorization, would have on performance. A paper is written on this subject [22], which examines the impact on performance when security is implemented. It may be interesting to research this topic in the context of the CineGrid store-and-forward servers.
- Make a comparison of distributed parallel file systems by installing them and execute several performance tests with them. In this project time was limited to do this. It may be possible that from these tests another result is concluded.
- Execute performance tests to see how many nodes give the best performance / cost relation. This has been tested on TeraGrid [20]. Here it becomes clear with 8 nodes a higher speed can be reached than with 16 nodes.

Chapter 8

Related work

The following articles and presentations are related to this research:

- The presentation "Walking the Line" [5] from the University of Amsterdam is about high performance network infrastructures. At page 16 a concept for a new storage architecture for CineGrid is proposed using the DAS-3 cluster [14] and Rembrandt cluster [16]. This may be one possible solution that can be researched.
- The article "Realtime compression for high-resolution content" [6] from the University of Illinois is about using "software DXT compression for high-resolution content at an interactive speed. DXT compression allows to stream full HD video over a gigabit connection where multiple gigabits were required. Moreover, it enables 4K streaming without the need for an high-end storage system or an expensive codec." This may be an alternative for any high-end storage solution.
- The presentation "Unification of Data Center Fabrics with Ethernet" [7] from the Storage Networking Industry Association (SNIA) is about the future in networking and storage. A market and technical overview of the competitive landscape for next generation 10 Gb technologies is provided with particular focus on the operational characteristics and implementation aspects of Ethernet. This may be a source for ideas for finding possible solutions for new storage architectures.
- The article "Distributed Parallel Fault Tolerant File Systems" [8] from Bitvis Datakonsult gives an overview of available file systems for Linux that may be considered for use in high-availability clusters and high performance computing, with a focus on open source. This may be a help in finding a file system.
- The presentation (in Dutch) "KAN het een NAS (met IP-SAN) zijn, of MOET het een FC-SAN worden....." [9] from Stork Fokker AESP B.V. describes a comparison between different storage technologies, such as DAS, NAS and SAN. This could be a source for ideas for finding possible solutions for a new storage architecture.
- The paper "A High-Performance Cluster Storage Server" [10] from the University of California describes a high-performance cluster server built around the SDSC Storage Resource Broker (SRB) and commodity workstations. A number of performance critical design issues and solutions to them are described. This may be a source for ideas for a storage architecture.
- The article "What Is 'Good' Application Performance" [11] from Imperial Technology describes the difference between IOPS and MB/sec and how applications are involved in this. This may help with making the list of criteria for the new storage architecture.
- The article "A Storage Architecture Guide" [12] from Auspex is about different storage technologies such as DAS, SAN and NAS and mentions advantages and disadvantages. Though this article is from the year 2000, it still is useful. This also can be used as a source for new ideas about storage architectures.

Bibliography

[1] CineGrid http://www.cinegrid.org/

[2] CineGrid Demonstrates Real-Time 4K Trans-Atlantic Streaming of Live Performance from Holland Festival in Amsterdam to San Diego, 2007, http://www.calit2.net/newsroom/release.php?id=1122

[3] CineGrid, Laurin Herr, 2006, http://www.firstmile.us/events/conf/spr06/presentations/CineGrid%20at%20First%20Mile%202006.pdf

[4] CineGrid: Global Facility for very high quality digital Cinema, Cees de Laat, 2007, http://staff.science.uva.nl/~delaat/talks/cdl-2007-01-17.pdf

[5] Walking the Line, Cees de Laat, University of Amsterdam, 2007, http://staff.science.uva.nl/~delaat/talks/cdl-2007-06-12.pdf

[6] Realtime compression for high-resolution content, Luc Renambot et. al., University of Illinois at Chicago, 2007, http://www.evl.uic.edu/files/pdf/ag2007-renambot.pdf

[7] Unification of Data Center Fabrics with Ethernet, Matthew Brisse, Dell Inc., Storage Networking Industry Association (SNIA), 2007, http://www.snia.org/education/tutorials/2007/spring/applications/Unification_of_Data_Center_Fabrics_with_Ethernet.pdf

[8] Distributed Parallel Fault Tolerant File Systems, Jerker Nyberg, Bitvis Datakonsult, 2007, http://bitvis.se/fs_overview.php

[9] KAN het een NAS (met IP-SAN) zijn, of MOET het een FC-SAN worden....., Gert Stolte, Stork Fokker AESP B.V., 2006, http://beurzen.jem.nl/fsmk/images/img.asp?src=pages&id=13124&number=1&type=1&wf=20

[10] A High-Performance Cluster Storage Server, Keith Bell et. al., University of California, San Diego, 2002, http://www.hipersoft.rice.edu/grads/publications/bell-cluster-storage2.pdf

[11] What Is "Good" Application Performance, Craig Harries, Imperial Technology, Inc., 2002, http://www.imperialtechnology.com/technology_whitepapers_Good_Performance.htm

[12] A Storage Architecture Guide, Auspex, 2000, http://www.storagesearch.com/auspexart.html

[13] Advanced Internet Research (AIR) Group University of Amsterdam (UvA) http://www.science.uva.nl/research/air/

[14] The Distributed ASCI Supercomputer 3 http://www.cs.vu.nl/das3/

[15] Distributed ASCI Supercomputer: DAS-3 http://www.starplane.org/das3/cluster.html

[16] Rembrandt cluster http://www.science.uva.nl/research/air/network/rembrandt/index_en.html (Only viewable with correct username and password!)

[17] Storage advisors weblog, Tom Treadway et. al., Adaptec http://storageadvisors.adaptec.com/

[18] NAS, DAS or SAN? - Choosing the Right Storage Technology for your Organization, Duran Alabi, Xtore http://www.storagesearch.com/xtore-art1.html

[19] List of file systems, Wikipedia, 2007, http://en.wikipedia.org/wiki/List_of_file_systems

[20] Wide Area Filesystem Performance using Lustre on the TeraGrid, Stephen C. Simms et. al., 2007, http://datacapacitor.researchtechnologies.uits.iu.edu/lustre_wan_tg07.pdf

[21] Building A High Performance Parallel File System Using Grid Datafarm and ROOT I/O, Y. Morita et. al., 2003, http://arxiv.org/pdf/cs.DC/0306092.pdf

[22] Scalable Security for High Performance, Petascale Storage, Andrew W. Leung, University of California, 2007, http://www.ssrc.ucsc.edu/Papers/ssrctr-07-07.pdf

[23] 10Gb Ethernet with TCP Offload - The High Performance Interconnect, Chelsio Communications, 2007, http://www.chelsio.com/solutions/pdf/Chelsio_10GbE_TOE_Perf_Bmarks.pdf

[24] PVFS: A Parallel File System for Linux Clusters, Philip H. Carns et. al., 2000, http://www.parl.clemson.edu/pvfs/el2000/extreme2000.html

Appendix A

Definitions

Term	*Explanation*
1U	U is the standard unit for specifying how many physical space is being used by a device in a rack. 1U is equal to 1.75 inch, which is 44.45 mm.
4K	4K in this context means 4K digital video. It is called 4K, because of the number of horizontal pixels: around 4,000.
Array	Array in this context means a RAID array. This means multiple hard disks drives are grouped into one array with the same function, or RAID level.
Block size	The size of blocks of data that are transferred by a file system. Usually in the order of KBs.
CIFS	Common Internet File System: a protocol that defines a standard for (remote) file sharing. Typically used in Windows systems.
DFS	Distributed File System: a file system able to provide access to files that are distributed to many servers. For end users it looks like they can access the files locally.
FUSE	Filesystem in Userspace: software which makes it possible to implement a fully functional filesystem in a userspace program.
Hotswap	A feature that makes it possible to add and remove drives from a computer, while the computer is running. The drives are automatically recognized by the OS.
IOPS	Input/output Operations Per Second: used as a method for defining CPU and server performance metrics.
MTBF	Mean Time Between Failures: the average time a failure occurs in a system.
NFS	Network File System: a protocol that defines a standard for (remote) file sharing. Typically used in Unix systems.