IMPLEMENTATION OF A RELIABLE AND EXPANDABLE ON-LINE STORAGE FOR COMPUTE CLUSTERS

Jos van Wezel, Forschungszentrum Karlsruhe, Karlsruhe, Germany
jvw@iwr.fzk.de

Abstract

The cluster at GridKa uses the General Parallel File System (GPFS) on a 20 node file server farm that connects to over 1000 FC disks via a Storage Area Network. The current 220 TB of on-line storage is distributed to the 510 node cluster via NFS. A load balancing system ensures an even load distribution and additionally allows for on-line file server exchange. Discussed are the components of the storage area network, specific Linux tools, and the construction and optimisation of the cluster file system along with the RAID groups. A high availability is obtained and measurements prove high throughput under different conditions. The use of the file system administration and management possibilities is presented as is the implementation and effectiveness of the load balancing system.

Current HEP and future LHC experiments that use the compute centre at GridKa need to handle large amounts of data. Traditional access methods via local disks or large network storage servers show limitations in size, throughput or lack of data management flexibility. High speed interconnects like Fibre Channel, iSCSI or Infiniband as well as parallel file systems are becoming increasingly important in large cluster installations to offer the scalable size and throughput needed for PetaByte storage. At the same time the reliable and proven NFS protocol allows local area storage access via traditional Ethernet at moderate costs.

COMPONENTS

The components of the on-line storage configuration are servers, disks, interconnecting Storage Area Network (SAN) fabric and the GPFS file system software. The disks are connected to RAID controllers that connect to the SAN.

Servers

The servers are 1-unit height rack mounted IBM x335 machines with the following specifications: dual Xeon 2.4 Ghz, 1.5 MB memory, 36 GB SCSI disk, two Broadcom Ethernet interfaces, Qlogic 2 Gb Fibre Channel HBA. The machines run RedHat Linux 8 with kernel 2.4.20.8-18. The computers of the server cluster are functioning as server only. The total number of available servers is 20 of which 7 machines are currently in production.

Disks and disk controllers

Disk storage is arranged in 9 racks. Each rack holds 1 controller and 10 trays with 14 drives for a total of 1260 drives. 36 drives are used as hot spare (4 per rack). The drives are 146 GB FC at 10 krpm. Disks in a rack are distributed over the 4 independent disk connections on the dual controller. The disk controller has 4 independent connections to the SAN. The hot spare disks are able to take over a failing drive in any of the locations of the rack. Read ahead caching as well as write caching was turned off for the created RAID volumes.

Although the disk controllers have NiCd batteries for backup of cache memory at power failure, write caching is not enabled. This frees up memory for the read cache. More important, the write cache on dual controllers can be used safely only when it is mirrored. Write cache mirroring in the particular controller is implemented by using the disk side fibre channel links for copying. Large writes will therefore see a reduced throughput if mirroring is enabled.

RAID volumes

Each disk rack holds 136 usable drives which allowed the creation of 17 RAID-5 volumes (7 drives + 1 drive parity) The usable capacity of each volume is just below the 1 TB limit required by the GPFS system (see below) and results in a reasonable loss of 13% in exchange for the RAID-5 reliability. The RAID volumes were created with 256 MB segment size.

GPFS

The General Parallel File System (GPFS) is a product of IBM and in use in many large computation centres in the world. Its key features are high parallel IO performance with a POSIX interface, the ability to sustain loss of multiple nodes, the capability to grow and shrink files systems and replace or exchange disk on-line and the possibility to export the file system via NFS. GPFS allows striping the data over many volumes. For a detailed description see [1].

The maximum size of a single disk volume in GPFS is 1 TB. Aggregation of volumes to 70 TB is supported. GridKa currently uses file system sizes in the range between 2 and 15 TB. All file systems are created with a 512 MB block size and striped over 2 to 15 volumes. Metadata is mirrored to allow recovery of the file system in case a volume is lost. Data mirroring is not used.

GPFS nodes can function in two configurations. In a SAN configuration all nodes have direct access to the storage and share the same storage. In a DAS configuration the storage attached to one node is made available via a high speed network to the other nodes. The SAN configuration offers the highest bandwidth to the storage and offers a higher availability. The SAN configuration is in production at GridKa.

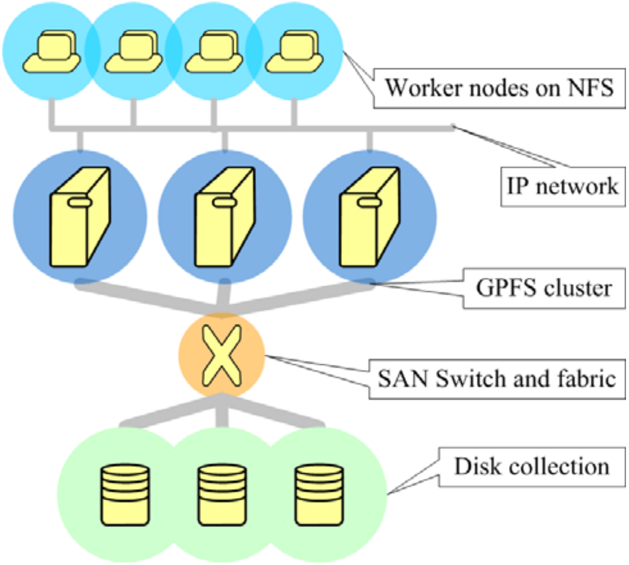

Figure 1: GPFS nodes export via NFS.

Integration with Linux

The used kernel is compiled with support for a large (200) number of SCSI disks, 4 GB memory and NFS over TCP. Several components of the Linux operating system are important in the presented configuration. For a system that needs to be on-line 24*7, changes to the system configuration have to be carried out while it is running. SCSI drivers and failover Fibre Channel drivers allow for this feature. Important are the following components.

- Fibre Channel host bus adapter (HBA) driver connects SAN based disks or logical units (LU) to the SCSI driver. A LU is seen as SCSI disk from the OS. The HBA driver takes care of a) changing the IO path if the current path fails and b) mapping SAN based LUs to SCSI id's.
- SCSI driver allows for hot adding disks. In a SAN storage devices can be added and removed without interrupting existing services. The Linux SCSI drivers allow a rescan of the bus to detect newly added devices.
- NFS server components are part of all Linux distributions. Because possibly hundreds of clients connect to the server the number of nfsd-threads was raised from the default 2 to 1024. A script at boot time reduces the time between wake-ups of the kernel page daemon from 30 to 5 seconds.
- Automounter autofs4 [2], used on the clients, comes with a user and kernel space part. Versions before 4.1 showed various problems in large installations. These are fixed in recent versions. Installed was 4.1.3.

The above components together with the GPFS logical volume manager (LVM) enable the removal and addition of disks. GPFS allows for the on-line exchange of disks. The failure of a disk controller or the loss of a path to a disk controller is handled automatically without interruption of the system operation. In fact it is possible to change a complete controller or its firmware while the system is on-line.

STORAGE LOAD BALANCING

There are two different mechanisms used to load balance the IO access.

File system level

All of the file systems in use have their data striped over different RAID volumes. Currently the largest file system in the GridKa cluster has a capacity of 15 TB and is striped over 15 RAID-5 volumes. These RAID volumes are located in different racks and behind different controllers.

To optimise throughput the volumes of large file systems are distributed over as many as 5 controllers. Stripes of a file lay distributed over controllers and different volumes. The highest throughput for reading as well as writing is reached when each stripe is on a volume managed by a different controller. Throughput tests have shown that a scalable throughput performance can be achieved. Reading simultaneously from 5 volumes with 10 GPFS cluster nodes reached beyond 1000 MB/s aggregated throughput [3].

Server level

The export of the file systems via NFS enables the possibility to spread the connections from clients over a cluster of servers. As GPFS offers a uniform view of the file systems on each of its nodes, NFS clients can select any node in the cluster to read and write files to.

A combination of DNS, the autofs4 automounter and NIS maps is used to construct a simple but very effective system to spread the client connections over the production servers.

All servers are listed in the DNS with a single name and corresponding IP address. This list of address records can be adapted when servers are taken off line or when servers are added to the server cluster. The autofs daemon mounts file systems upon mount point entry on a client. The NIS maps list a single host name for a given file system.

When the automounter commences a file system mount it runs a programmable map script. The script interrogates the specific NIS-map, requests the list of host names from the DNS for the entry in the NIS-map and selects an entry at random. The NIS-map returns one host name / mount point pair for a given key.

In Figure 2 it can be seen this results in an even pattern of throughput for the servers in production. They have an almost equal number of clients to serve and the activity mirrors this fact.

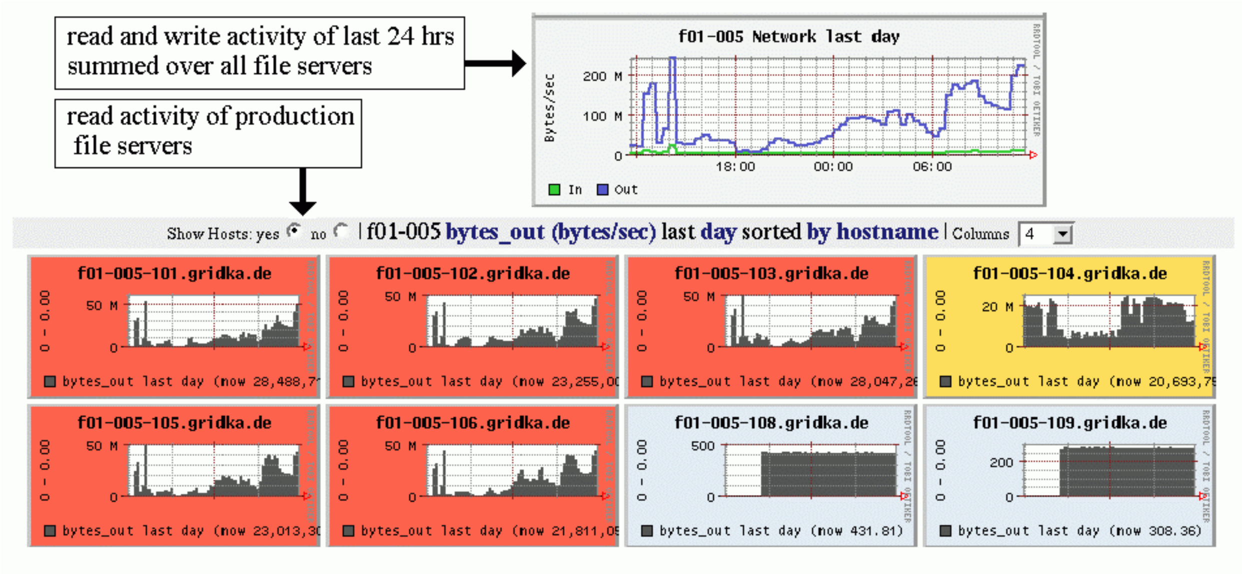
read and write activity of last 24 hrs summed over all file servers
read activity of production file servers
f01-005 Network last day
Bytes/sec
200 M
100 M
0
18:00
00:00
06:00
In
Out
RRDTOOL / TOBI OETIKER
Show Hosts: yes no | f01-005 bytes_out (bytes/sec) last day sorted by hostname | Columns 4
f01-005-101.gridka.de
bytes_out last day (now 28,488,7
f01-005-102.gridka.de
bytes_out last day (now 23,255,0
f01-005-103.gridka.de
bytes_out last day (now 28,047,2
f01-005-104.gridka.de
bytes_out last day (now 20,693,7
f01-005-105.gridka.de
bytes_out last day (now 23,013,3
f01-005-106.gridka.de
bytes_out last day (now 21,811,0
f01-005-108.gridka.de
bytes_out last day (now 431.81)
f01-005-109.gridka.de
bytes_out last day (now 308.36)
0 - 0.00
50 M
20 M
500
200
0

Figure 2: effectiveness of server based load balancing

SUMMARY

The described configuration has proven its viability at the GridKa tier 1 centre. This on-line storage solution minimises costs by using standard NFS functionality over Ethernet, while offering scalable high throughput and availability at the same time.

It is possible to manage 220 TB of disk space and 20 servers with a minimal operator effort. The export of the cluster file system via NFS removes the need for expensive components (fibre channel HBA, large memory, PCI slots on separate busses.) from the clients. NFS offers a stateless POSIX compatible file server connection. Applications do not have to be changed. The temporary loss of a server is handled by the NFS-client. IO is stalled until the server or a replacement is put back on–line and continues where it left off.

The GPFS cluster file system supports file system sizes up to 70 TB. A total of more than 1 PB of storage space within one rooted file system is possible. A file system that is unevenly striped after the exchange of disks or the addition of disks is re-striped on-line by a separate process.

Initial problems were encountered with the volume exchange function of GPFS and the functionality of the autofs4 automounter. Newer versions of both software packages have resolved these.

FUTURE WORK

The continuous improvement of the storage configuration at GridKa has led to the situation where a single Ethernet connection is easily saturated. It is possible to combine 2 or more Ethernet interfaces in order to improve throughput per server. The number of possible interfaces per server has to be experimentally determined.

The load balancing system allows for the introduction of load policies. Servers with less capacity or with single Ethernet connection would have a smaller change of getting selected by the load balancing system. It also allows the use of different hardware in the server cluster.

Focus will also be on getting Scientific Linux and the recent GPFS version to work.

ACKNOWLEDGEMENTS

The author wishes to thank Holger Marten, Doris Ressmann and people of the departments GIS, GES and DASI from IWR at Forschungszentrum Karlsruhe and the German Federal Ministry of Education and Research (BMBF), for their support at various stages in the preparation of the work presented.

REFERENCES

[1] http://www-1.ibm.com/servers/eserver/\ clusters/software/gpfs.html

[2] http://www.kernel.org/pub/linux/daemons/autofs

[3] J. van Wezel, B. Verstege, A. Jaeger and H. Marten, “High throughput cluster computing with Linux,” To be published in proceedings of 9th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT03)