PAUL SCHERRER INSTITUT

PSI

Wir schaffen Wissen – heute für morgen

Paul Scherrer Institut

Heiner Billich

DAQ at high data rates with 2d-detectors: Storage, network, compute nodes at SLS – Pilatus6M and beyond

ESRFUP WP10 Workshop 13./14. January 2011

Outline

What did work for ~100MB/s incoming data rate?

What will need to change to achieve 1-10GB/s incoming data rate?

Optional:

Details on the SLS beamline compute and storage setup

Very short introduction to GPFS

The standard data flow – set the scene

1. Detector connected to detector server at beamline
2. Detector server at beamline gets all data from detector and writes to beamline storage, located in the computer room. One file per image, .cbd, .edf, .tif, …
3. Analysis jobs read data from beamline storage and write results and intermediate files back
4. Data export and archiving reads from beamline storage and writes to external disk, tape or sends via network/WAN
5. Quality control/visualization on beamline consoles reads data from beamline storage

The transfer from detector server to beamline storage is specific for each beamline and detector

- NFS mount beamline storage on detector server – marCCD, PX-III
- CIFS mount - PCO camera, Tomcat
- Native GPFS client on detector server - Pilatus2M and Pilatus 300W, cSAXS
- Sync_daemon (rsync based, Ezequiel Panepucci) - Pilatus6M at PX-I and PX-II

Pilatus6M – some simple math in 2007 ….

Back in 2007 - Pilatus6M at beamline PX-I

Requirements

- 12Hz@6MB - 72MB/s peak data rate from detector at beamline to computer room
- Two processing jobs in parallel, data export to external disk, data influx from detector – need to provide 470MB/s sustained i/o to beamline storage
- Max. 5 minutes runtime per processing job – need 32cores, 8 nodes, 2GB per core
- Keep 2 months' worth of data – 30TB beamline storage

Rule of thumb:

- 500GB SATA disk at 7'200rpm provides15MB/s
- Entry level Raid6 controller: 150MB/s per controller

Conclusion

- 470MB/s - we need 32+ data disks and 4+ Raid6 controller
- 30TB - we need 60+ data disks
- 72MB/s - 1GbE Network from beamline to compute room is sufficient
- 4Gb/s fibre channel storage network is sufficient

What did I learn?

We were limited by **metadata performance** in the beginning, i.e. stat(), readdir(), open(), close(), ….try “ls –ls” on 100’000 files …
Our Sata storage was very peculiar about metadata i/o to Sata Raid6. Once I moved metadata to separate FC disks maximum performance increased from ~600MB/s to ~900MB/s for 96 disks in 16x 4+2Raid6.

It’s the **Raid controllers which limit the maximum data rate**, not the disks.I measured 150-300MB/s max. throughput per controller using entry/medium class controllers in 2007-2009. Hence 10-20 data disks (+N x parity disk) can saturate at controller. You scale by adding more controllers with a fixed number of disks by controller, you don’t scale by adding more disks to an existing controller.

GPFS gives real posix semantics and avoids NFS caching issues – files are visible and accessible on all nodes at the same time, you don’t need to wait for files to show up, you never get outdated content. We did put GPFS on some beamline consoles and detector servers mainly for this reason.

Why did it work – is it that easy?

Data compression on the detector server , about factor 4. Things would not have worked out without compression a the source: 280MB/s data rate from detector to computer room, 120TB disk space, 1000+ MB/s i/o to beamline storage – I probably can handle all this numbers in 2011, but definitely not in 2007. All credits to Erik Eikenberry (Dectris).

GPFS did prove to be the right choice – it gives access to the full performance of the underlying hardware and otherwise stays out of the way – it delivers what it says it can deliver with a very reasonable amount of sysadmin work involved. Credits to IBM and those at PSI which choose GPFS as parallel file system – the decision was made before I joined PSI. .

Why did it work – finally it's the people, not technology

In 2007 and ever since then I found

a working, stable, well designed and maintained **linux and network environment with all needed services readily available** (scalable and adaptable configuration management and kickstart scripts, yum repositories, accounts/directory service, naming conventions, clean and scalable network design, firewall services, user support, monitoring, …).If you need to compensate any deficiencies in these services you loose time, energy and focus. If this services work well you get a big leverage for your own work. Credits and many thanks to **René Kapeller**, Remo Rickli, Urs Beyerle, Tobias Marx, Karel Stadler, Frank Lendzian, Valeri Markushin, Mark Gasser, Hans-Christian Stadler, Mauro Bianchi, Edgar Barabas, Marco Kohler, …

and: Beamline staff and management that fully supported and trusted in myefforts. Ezequiel Panepucchi and all MX , Federica Marone and all Tomcat, Oliver Bunk and all cSAXS, and many more.

Beyond Pilatus – data rates of 1-10GB/s

Eiger and other upcoming detectors will deliver uncompressed data at rates of 1-10GB/s.

This is a 10-100 fold increase compared to todays detectors in use at SLS.

What changes?

Data rates of 1-10GB/s – What will change?

How Eiger and others differ from current 2d-detectors

- Parallel modular detector architecture and readout, no more single detector server which receives all data
- Detector itself sends data via several parallel 1/10 GbE links – most likely simple UDP streams. No more detector servers with special interfaces to the detetcor like cameralink, gigastar,

Data rates of 1-10GB/s – What needs to change?

What needs to change for storage, compute and network?

- Use parallel processing from data influx over processing/analysis to storage.
- Write large files from 50MB to a few GB – hence no more “one file per image”. Use HDF5
- No longer read any data back from disk that you can keep in memory and distribute via message passing as well.
- Adapt all data-handling software to do “large block sequential i/o” and apply other HPC storage best practice, naïve approaches as often used today will no longer work
- Fast network link between beamline and computer room – most likely n-times 10GbE
- Fast storage and message passing network inside the computer room – QDR Infiniband (or 10GbE??)

Does compression come to rescue - again?

Compression at the source, i.e. on the detector itself, would be ideal. This requires fast compression algorithms which an be implemented in hardware (FPGA) with a deterministic - real time? - behavior. Ideally this would allow fast on-the-fly decompression on compute nodes as well. I don't know whether this is possible, at least there seems to be no obvious well known candidate.

Next best choice would be data compression before you write to disk. Here you need to handle incoming data rates of 1-1GB/s, but time behavior can be a bit less restrictive, you can have a few 100GB of buffer memory. With HDF5 I know two promising candidates

- LZF compression filter by Andrew Colette (http://h5py.alfven.org/lzf/)
- LZO by Markus Oberhumer (http://www.oberhumer.com/opensource/lzo/) and used wit PyTables (http://www.pytables.org)

We did not investigate any further in this direction yet.

Compresion may be extremly beneficial for some data sets– in some occasion I've seen a scale ranging from 1-10^{-4} counts per pixel and image – i.e. the vast majority of pixels report just zero counts.

Data rates of 1-10GB/s – current state

What do I know

- GPFS on DDN storage - 4+ GB/s sustained large 4MB block random i/o (300 disks, 30x 8+2 Raid6)
- Infiniband QDR and current Intel Xeon X5600 hardware – 3GB/s sustained i/o for a single process – MPI message passing and i/o to disk.
- 10GbE and data incoming as UDP stream – can handle up to 1GB/s with almost no packet loss in direct back-to-back benchmark setup.
- HDF5 – 1'000-1'800MB/s
- Python – you can get very close to native (C language) MPI and HDF5 performance

Need to investigate, missing components

- HDF5 – can I reach 2'000+ MB/s ?
- 10GbE and UDP packets – will our switches drop packets, does it work in real life?
- Fast compression with HDF5
- Software …..

Optional slides

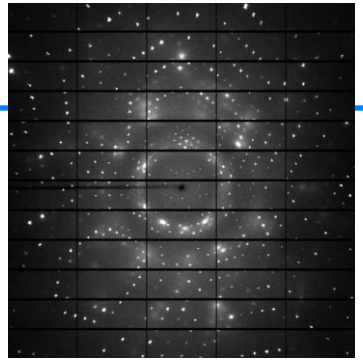
Optional Slides

First presented at hpc-ch community meeting “Parallel file systems for HPC”

2010-10-28, Zürich

http://www.hpc-ch.org

Lots of data ...



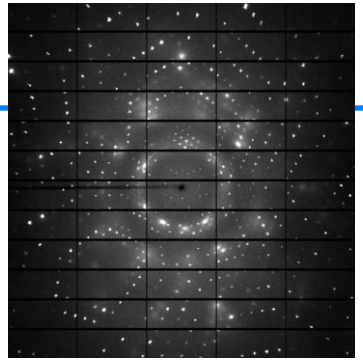
- About 1TB/day/beamline regularily at “high throughput” beamlines
- Several million files created per day at a single beamline (worst case)
- Small files:130KB – 18MB, about 100-10’000 files per dataset
- take data at 10-100Hz - create files with 10-100Hz
- Concurrent - random ! – access to the same datasets – take data, run analysis on compute nodes, export to external disks or via network, write to tape archive, visualize, compress, ...

- in 2007 we had demand for 500MB/s sustained throughput and 30TB space for a single beamline.
- Next generation detectors will deliver data at 3GB/s and above (3GByte/s, not Gbit/s !!) in 2012 and later

Traditional Linux based NFS servers with Sata disk arrays could no longer handle these demands, expecially with random/concurrent access.

Couldn’t easily scale up by factors of 5-10.
Additionally NFS caching issues showed up, clients took seconds to “see” new files.

Lots of data - GPFS and HPC



In 2007 decided to

- Use GPFS as parallel file system at SLS, don't try to beef-up our traditional NFS servers but replace them.
- Use storage and compute solutions and ideas from HPC and high throughput computing.
- Go for a common solution for all experiments.
- Keep all storage and compute ressources in a single place (serverroom), don't fragment.
- Pool and share ressources as much as possible.

Constant change ...

2007

- Implement first GPFS cluster for a single beamline, 30TB usable, 96x500GB SATA, 8 compute nodes, 4GB FC, SAN topology for GPFS, i.e. direct storage access for each node.
- Implement separate GPFS clusters for 5 more beamlines, total of 120TB usable,16 FC storage systems

2008

- Use **CTDB to add high availabililty NFS and CIFS clustering** to GPFS
- start to use more GPFS features: Separate metadata disks, mirroring, multicluster, storage pools, filesets, storage policies, ...

2009

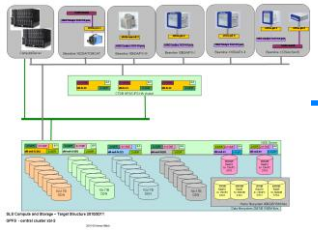
- Make one GPFS cluster a single central HA NFS and CIFS cluster for all “small throughput” experiments.
- Use GPFS for $HOME
- **Add Infiniband network and storage** to replace 4Gb/s Fibrechannel

2010

- start to **merge five “high throughput”** experimental stations’ **storage** into one single filesystem
- switch to GPFS LAN topology to get better scalability
- Start to decomission storage purchased in 2010

In 2009 and 2010 I did migrate/move all data at least once per year.

Target Structure End of 2010



Just two GPFS clusters, 6 NFS and CIFS servers in total.

One for small throughput experiments – focus on availability
all data and metadata mirrored, fully redundant

- 4Gb/s FC SAN
- 22TiB work + 400GiB home
- 34M files work 10M files home
- 300MB/s sustained throughput

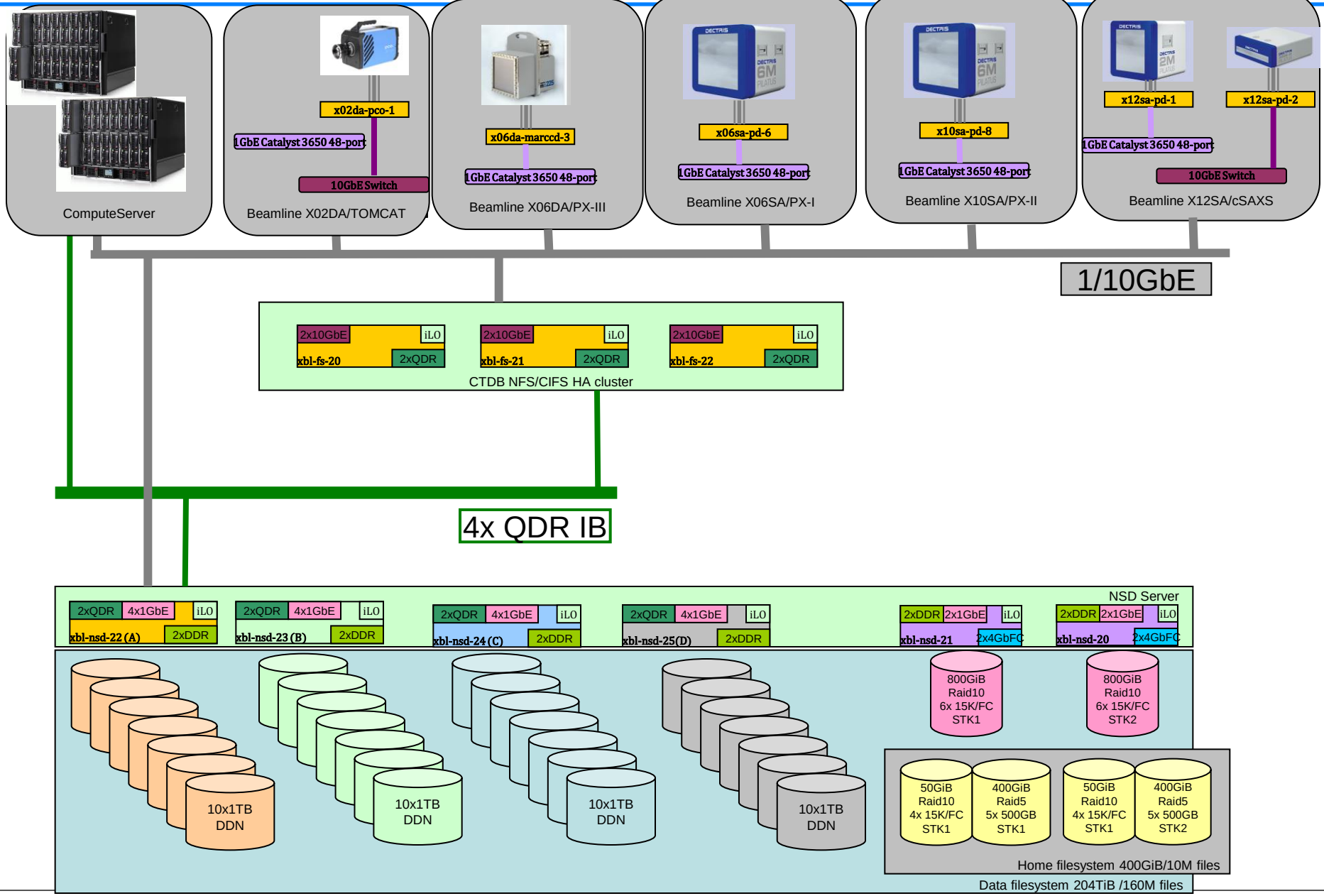
One for five high throughput experiments – focus on performance.

- Single file system for work
- 280x 1TB Sata / 204TiB usable for work + 400GiB home
- 160M files work and 10M files home.
- 4x QDR Infiniband based, IB on all file servers and compute nodes
- 25 compute nodes
- 2-3GB/s sustained throughput.

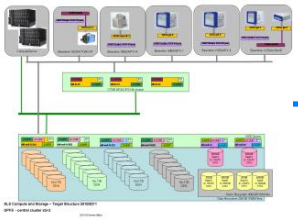
Compute nodes in separate storage-less gpfs cluster, use multicluster to grant access. Same for certain remote nodes in the experimental hutches.

PAUL SCHERRER INSTITUT

Target Structure End of 2010



Target Structure 2010 - Hardware



1x DDN S2A9900 Couplet, 5 disk enclosures,300x 1TB, Infiniband DDR hostports, Raid6 8+2
4x HP DI380G6 with QDR and DDR Infiniband HCA as GPFS NSD servers for DDN storage

2x SUN STK6140 with 12x300GB 15K amd 8x 146GB 15K for metadata (Raid10)
2x HP DL360G5 with FC HBA and IB DDR HCA as GPFS NSD servers for FC storage (access to metadata)

6x Xyratex F5402E. 24x500GB SATA each. (3+ years old. Raid6 4+2)
4x Qlogic SANBox 5600 16-port FC switches, 4Gb/s, dual-fabric setup

2x HP C7000 Blade Center with internal IB QDR switch, one with 10GbE internal switch, too.
1x HP C7000 Blade Center with FC pass-through
25x HB BL460G6 as compute nodes
6x HP BL460G6/G1 as NFS/CIFS file servers

1x Mellanox IS5030Q IB Switch 36port, 4x QDR

What is GPFS?

Disclaimer:
I can't give a complete introduction to GPFS in the short time available. Follow the links at the end of this presenstation for more details. Here I just highlight some features which make GPFS special or outstanding.

I copied a lot from Raymond L. Paden's presentation "*GPFS Best Practices*"

What I GPFS? **GPFS is a *shared disk*, *parallel clustered*, *posix compliant* file system.**

Shared disk
all user and metadata are accessible from any disk to any node

Parallel
user data and metadata flows between all disks and all nodes in parallel

Clustered
1 to 10's to 1000's of nodes (I can't speak for more then ~30 nodes)

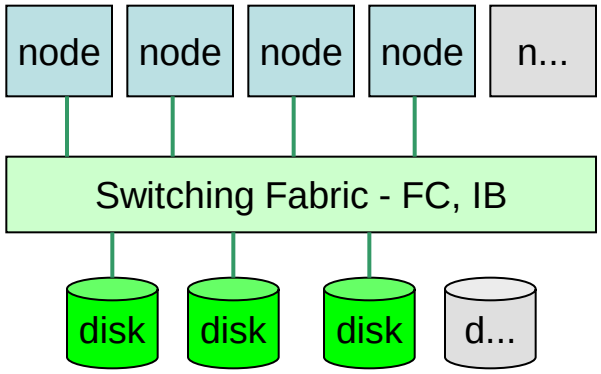
Posix Compliant
Processes on *different nodes* accessing the same file get posix semantics.

GPFS – Shared Disk, SAN Topology, LAN Topology

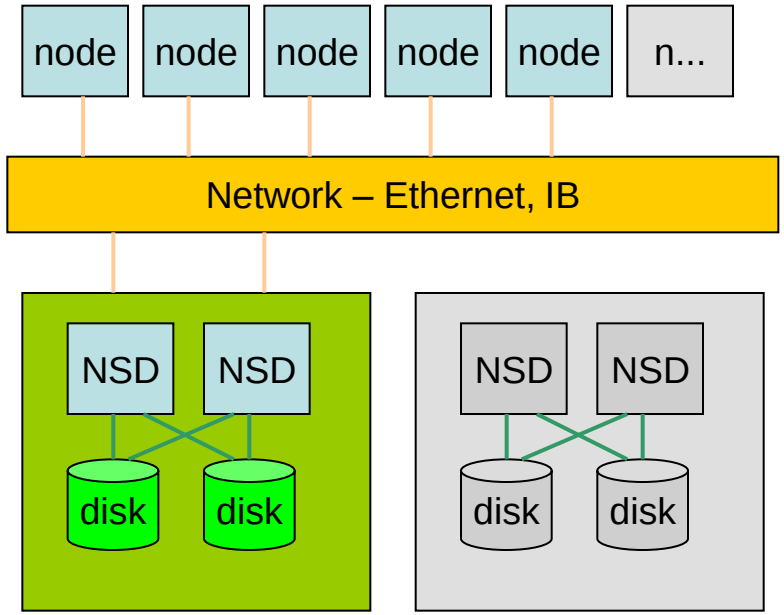
In GPFS each node can access each disk (block device) concurrently, either directly through some kind of direct connection (Fibrechannel, SAN, InfiniBand SRP, ...) or through *NSD servers*.

NSD Server

GPFS's implementation of a block device server. Like iSCSI. Works over 1/10GbE and Infinband. Can provides very high throughput. Usually used in large cluster with more then 30-60 nodes. Several NSD servers can serve the same disk to provide redundancy.



GPFS SAN Topology



GPFS LAN Topology

GPFS – Performance – Short Answer

GPFS gives you access to all throughput the underlying storage can provide and it scales linearily if you add more storage controllers.

```
dd if=/dev/zero of=test bs=1M count=32768
32768+0 records in
32768+0 records out
34359738368 bytes (34 GB) copied, 15.2419 seconds, 2.3 GB/s

dd if=test of=32GB bs=1M
32768+0 records in
32768+0 records out
34359738368 bytes (34 GB) copied, 11.7497 seconds, 2.9 GB/s
```

40x 1TB Sata disks in 4x Raid6 8+2.
QDR IB as storage network.
DDN S2A 9900 controller
HP Bl460cG6 Blades with X5650 CPU

Performance – the usual disclaimer

```
dd if=/dev/zero of=test bs=1M count=32768
32768+0 records in
32768+0 records out
34359738368 bytes (34 GB) copied, 15.2419 seconds, 2.3 GB/s

dd if=test of=32GB bs=1M
32768+0 records in
32768+0 records out
34359738368 bytes (34 GB) copied, 11.7497 seconds, 2.9 GB/s
```

GPFS gives you access to all throughput the underlying storage network can provide and it scales linearily if you add more storage controllers

as long as you

configured your hard- and software stack right

- raid controller parameters (caching, block size, ...)
- operating system (LUN queuedepth, srpdaemon.conf, /sys/block/sd*/queue/*
- GPFS (block size 1MB or larger, pagepool, ...)

and if your jobs show suitable i/o patterns

- large files
- sequential or strided access
- i/o in multiples of file system block size
- i/o aligned to file system block boundaries

This is not bad and probably is the same for all file systems. There are several “best practice” and tuning guides for GPFS available. There is no magic involved, in general you understand what’s going on.

GPFS – Reliable? maintainable? serviceable? ....

You can setup GPFS in a fully redundant way, failover and recovery mechanisms are build in. With CTDB on top of GPFS you can easily build a high-available NFS and CIFS cluster (http://ctdb.samba.org). This is a big plus in our environment.

I did never loose data due to a failure of GPFS, I consider it to be as stable as the linux kernel, samba, apache, ...

GPFS provides a complete set of administrative commands and all the functuality I expect, like

- add a disk and enlarge a filesystem during operation
- add and remove nodes during operation
- emty and remove a disk from a filesystem during operation
- migrate data from one disk to another during operation
- do rolling upgrades of GPFS, i.e. update one node at a time while the others stay up
- get detailed configuration and performance information
- snmp interface

Informal support through the gpfs mailing list and gpfs forum is very good, you get in direct contact to the gpfs developers. Formal support through IBM is very professional.

GPFS – no buzzwords?

GPFS is a serious no-nonse technical product with a very low buzzword density.
But there are ***cool*** features:

- It really scales linearily with the number of disk systems
- It has no separate metadata server or any other single controlling or coordinating instance which may easily become a bottleneck. Of course there is an overhead involved in metadata operations.
- GPFS uses RDMA over InfiniBand, hence you can get very high throughput on Infiniband-base clusters without the need to deploy a separate storage network.

- "storage pools" – you can use different storage types in the same file system – like SAS, SATA, 1TB, 2TB
- storage policies – you can setup rules on which files to place in which storage pool, which files to migrate from one pool to another
- external storage pools – you can extend GPFS with tape storage to build a HSM, this is available for Tivoli Storage Manager and HPSS (We never did try this).

- It's no black box, you get documentation and access for many parameters and internals required to tune and adapt to special needs.

- You don't need to use IBM hardware to get support from IBM, GPFS is not locked to IBM hardware.

Key questions from Michele and Olivier

Why GPFS?
Back in 2006 it was much more mature then Lustre.

Custom drivers?
You need to compile kernel modules – for the latest kernel.org kernels you may need to wait long until they are supported.

Usage?
Home directories and work – temporary storage for beamline data

What is most important?
1 – Reliability – we are a user lab and production faciltiy
1 – Performance -
3 – Costs, as long as they stay in a reasonable range (~ 1'500-2'000CHF/TB)

Benchmarks?
just plain dd
run application benchmarks which do a lot of IO
use strace, iostat, dstat, ... to watch production

Key questions

build yourself or turnkey?

We did build ourself and yes, I would do it again.

We take an incremental evolutionary approach and frequently improve and extend. There are more or less continuous changes required to follow the developments at SLS. This is difficult to do with an external supplier.

But: It takes time to build up the skills and knowledge required, you need dedicated staff, it can't be done part-time easily.

Costs? Open source vs. proprietary?

In our case license costs are a very small fraction of the overall costs including hardware, network, serverroom, ... Support for GPFS through IBM, mailing lists and the support forum is almost optimal.

How would we extend?

I did change storage hardware, storage network, compute nodes, during the last three years. GPFS, Linux, ctdb,samba did stay and I expect that they'll stay for some more time. We probably change to different hardware in 2012 or 2013, just the usual upgrade after three years..

Summary

GPFS is a mature product with good features, but it is not a “silver bullet”.

To get a reliable and performing storage setup you need to do a careful design and spend considerable work, either by yourself or through consultants.

Don’t expect to get things right at the first time – if possible start small and grow incrementally to allow major changes during the first phases.

GPFS is very flexible, you can adapt it to your needs - but there are usage patterns which just don’t fit - like random i/o in small unaligned chunks (VMWare virtual disk images ...).

References

Paul ScherrerInstitut http://www.psi.ch
Swiss Light Sourc http://www.psi.ch/sls

GPFS Best Practices – Raymond L. Paden
http://www.oscer.ou.edu/GPFS_Materials_20100422_23/tutorial.v17.2.pdf
GPFS main page
http://www.ibm.com/systems/gpfs
Documentation
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html
FAQ
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfs_faqs.html
GPFS Forum
http://www.ibm.com/developerworks/forums/forum.jspa?forumID=479

GPFS Wiki
http://www.ibm.com/developerworks/wikis/display/hpccentral/General+Parallel+File+System+(GPFS)
GPFS Mailing List
https://lists.sdsc.edu/mailman/listinfo.cgi/gpfs-general
GPFS at IBM research
http://www.almaden.ibm.com/StorageSystems/file_systems/GPFS/
GPFS – A clustered Filesystem
http://www.nsc.liu.se/nsc08/pres/gottschalk.pdf

CTDB home page
http://ctdb.samba.org
CTDB presentation from SambaXP 2010
http://sambaxp.org/files/SambaXP2010-DATA/Michael_Adam_Tutorial.pdf
CTDB presentation SambaXP 2009
http://sambaxp.org/files/SambaXP2009-DATA/Michael_Adam_tutorial-paper.pdf

PAUL SCHERRER INSTITUT

PSI

Contact: Heiner.Billich@psi.ch, http://www.psi.ch/sls