Experience with HEP analysis on mounted filesystems.

Patrick Fuhrmann\(^1\), Martin Gasthuber\(^2\), Yves Kemp\(^2\), Dmitry Ozerov\(^2\)

\(^1\)dCache.org, Notkestrasse 85, D-22607 Hamburg
\(^2\)DESY, Notkestrasse 85, D-22607 Hamburg

E-mail: yves.kemp@desy.de

Abstract. We present results on different approaches on mounted filesystems in use or under investigation at DESY. dCache, established since long as a storage system for physics data has implemented the NFS v4.1/pNFS protocol. New performance results will be shown with the most current version of the dCache server. In addition to the native usage of the mounted filesystem in a LAN environment, the results are given for the performance of the dCache NFS v4.1/pNFS in WAN case. Several commercial vendors are currently in alpha or beta phase of adding the NFS v4.1/pNFS protocol to their storage appliances. We will test some of these vendor solutions for their readiness for HEP analysis. DESY has recently purchased an IBM Sonas system. We will present the result of a thorough performance evaluation using the native protocols NFS (v3 or v4) and GPFS. As the emphasis is on the usability for end user analysis, we will use latest ROOT versions and current end user analysis code for benchmark scenarios.

1. Introduction

In High Energy Physics (HEP), different systems are used for data storage and data analysis. The ones currently holding most LHC data are for example CASTOR and dCache. In addition, DPM, EOS and XROOTD server systems are used. Only a minority of sites uses systems based on Lustre or GPFS. Only those products feature a filesystem that can be mounted and be used directly. The other systems use dedicated protocols to access data and meta data like dcap, RFIO or xrootd (some server systems support more than one of these protocols). These protocols all have in common that they are proprietary protocols in the sense that they are not standardized, provided by the HEP community and maintained by it. The usage of these systems outside of the HEP world is small - if there is any. One reason certainly is the lack of standardization, the other one is the absence of a mounted filesystem and its advantages.

The advantages for user analysis on a mounted filesystem are obvious:

- All kind of even closed source software can be used for analyzing data on the respective storage system, like e.g. Matlab.
- Direct file access via a mounted filesystem is certainly the most common file access method, hence the storage backends are most interchangeable.
- If using the Linux OS, the Kernel VFS cache can be used without further configurations or adaptions. Neither a HEP site administrator needs to setup cache parameters, nor does the HEP community need to develop or maintain this cache: This is done by the Linux
developer community. In addition, the Linux VFS cache has greater persistency than any application/job-based cache.

- Files can be browsed and accessed easily from Linux - and some other OS. This is of special importance for end-user-analysis.
- As we will show in these proceedings, users will also benefit in terms of performance.

Some of the network protocols used in HEP can be mounted using the Fuse filesystem. It has shown however that data access via Fuse often has severe performance issues. So Fuse does not present a real advantage, as it is only used for metadata operations and the native protocol for the actual data transfer [3].

In the last years, many vendors of storage systems have invested into NFS v4.1 and the pNFS extension. In the HEP community, dCache and DPM are following this path and support or plan to support access to their system via NFS v4.1/pNFS so that users can easily mount these storage systems on their client.

2. Status of NFS v4.1/pNFS
During the past months, many developments around NFS v4.1/pNFS took place, which might be worth summarizing. We can only give the status quo as of CHEP 2012 and cannot reflect the developments that will come after CHEP 2012.

2.1. NFS v4.1/pNFS client availability
The OS most used in HEP for analysis certainly is Linux. The NFS v4.1/pNFS file layout is included in the vanilla kernel since version 2.6.39, block layout is included since 3.0. The Linux distribution most used in HEP is ScientificLinux which is derived from RHEL. RHEL 6.2 features NFS v4.1/pNFS as a technology preview [4], SL 6.2 follows this path.

CITI provides a client for Windows 7 (file layout) and has published it under LGPL. Microsoft announced support for both SMB 2.2 and NFS v4.1 as clients and as servers for the upcoming Windows 8 [5].

A Solaris client is being worked on, no availability date is published yet. The VMware hypervisor will have an integrated client, which also is not yet public.

2.2. NFS v4.1/pNFS server availability
The dCache server supports NFS v4.1/pNFS file layout since release 1.9.12 (April 2011). DPM server development is ongoing and not published yet.

NetApp has announced that ONTAP 8.1 ClusterMode available since April 19th 2012 support NFS v4.1/pNFS file layout [9]. IBM, Panasas, BlueArc, EMC and Solaris are working on prototypes, no release dates are known yet.

3. Test procedures and results
In the following, we will perform some qualitative and quantitative investigations on some mounted filesystems. The emphasis was on NFS v4.1/pNFS, but also other filesystems and access protocols were looked at. The testbed used was the DESY GridLab, which is described elsewhere [1, 2]. Recent versions of the gLite WN installation 3.2.12-1 and dCache version 1.9.10pre were used (some tests were performed with dCache version 2.1). On the client machines, the Linux Kernel 2.6.39-39.1 was used, compiled for the otherwise standard SL 5 distribution. For some tests comparing the dCache xrootd protocol implementation with the XRootD/SLAC xrootd protocol implementation, the files on the dCache pool nodes were also served using an XRootD/SLAC server installed in parallel. More details an this setup are found in [6].
3.1. **ROOT reads**

We repeat the tests presented in [1] using the new ROOT version 5.30.02. The ROOT team provided us with two example files: An ATLAS file with the original, non-optimized basket structure (AOD.067184.big.pool.root), and another ATLAS file with reorganized baskets (atlasFlushed.root). We were also provided with a ROOT macro, taodr.C. This macro was slightly modify by us to allow easy change of reading parameters:

- File type.
- xrootd server from dCache, xrootd server from XRootD/SLAC, dcap, dcap++ (a modified dcap version providing smart block caching.) [7] and dCache NFS v4.1 server.
- read all branches or only two branches.
- A TTTreeCache of 60 MBYTE, or no TTTreeCache.
- Reading 1, 8, 16, 32, 64, 128, 192 or 256 files in parallel.

Comparing the scaling behavior of the five different protocols over varying parallel file reads results in eight plots. Two example plots are shown in figure 1.

### 3.2. **WAN reads**

In the Grid, wide area transfers are done using well established transfer mechanisms based on GridFTP and steered using SRM and FTS. It is improbable that mounted filesystems will replace these mechanisms for data transfer. It is however important to investigate whether a mount over WAN would work in principle and what the performance penalty is. Two imaginable examples for WAN mount would be for example dCache mounts between the two DESY sites in Hamburg and Zeuthen over their dedicated 10 GBit line, or the planned remote Tier0 in Budapest reading from CERN. In a first step, we have done qualitative tests: From one worker node (WN) in Hamburg, a small test dCache instance in Taipeh and a production ATLAS dCache instance in Vancouver were used. The connection worked without errors or interruptions.

In a second step, a quantitative investigation was performed. The Linux Kernel tool netem [8] was used to simulate different latencies as well as some other network parameters like jitter and
packet loss on the WNs. The test itself was performed in the DESY GridLab and three different read scenarios analyzed:

- Reading all branches of a ROOT file with read-optimized buffers without a TTreeCache.
- Reading all branches of a ROOT file with read-optimized buffers using a 60 MB TTreeCache.
- Reading the complete file using "cat" (equivalent) to streaming the content over the network.

One can clearly see the positive effect of an application cache in figure 2. The fastest reads are however the simple streaming over the network. This is obvious since seek operations over network are known to be slow. The main message is clear tough: The performance penalty grows linearly with the latency. Grid WNs with a NFS mounted dCache storage system can perform file reading and analysis even when the network has a very high latency.

3.3. File open time

When using a storage system for analysis, especially when it is mounted, users expect a fast opening of files. The faster, the better the interactive experience with the filesystem is. Some analysis code has shown that it needs opening all necessary files before starting analysis - and the number of necessary files can reach 1000 or more. In such a scenario, a fast open time directly translates into a short turn-around time for analysis jobs.

For dCache, the open time depends on many factors, including the load on the namespace database and the performance of the hosting machine - factors that are to some extent difficult to quantify. Also comparing the open time from dCache to the open time of another product is difficult as both products can be tuned to faster open times. We therefore opted for an investigation on the dCache installation given in the GridLab. No special tuning like SSD disks or RAM caching was applied to the hardware or the dCache namespace database. It was taken care that the client caches were empty while the files were present in the VFS cache of the file server to exclude hard disk seek times. The open times from a NFS v4.1 mounted dCache storage system were compared to xrootd and dcap reads. The total read times (time to copy the file to /dev/null) were measured for files of different file sizes. We define the file open time as the time it would take to read a file with 0 byte file size. This time was obtained by linear
regression of the recorded values. In figure 3 one can see that the the file open time for NFS is somewhat smaller than the open time for the dcap and xrootd protocols. No large differences are seen. The dCache installation in the GridLab shows around 50 ms file open times.

3.4. The parallel feature in pNFS: Scaling over many file server
Together with NFS v4.1, the protocol extension pNFS - parallel NFS - was defined. This decouples the metadata from the data transport. Data are transferred directly between clients and machines hosting the files in the file layout case. This approach is well known from storage products like dCache and others in use in HEP, but is new to NFS. We investigated on the scaling behavior of dCache in that respect.

We used two dCache installations, which were identical except for one point: the first dCache installation comprised five pool nodes (the normal GridLab dCache installation described above), while the second dCache installation only comprises one pool (hardware identical to the five pools). We varied the number of simultaneously reading clients from 1 to 256 for both installations. As can be seen in the left plot in figure 4 the performance of the one-pool dCache does not scale as well as the five-pool dCache, which was expected. We rescaled the values of the five-pool dCache such that the number of clients per pool can be compared to the one-pool dCache [2]. This is obtained by dividing the number of clients in the five-pool scenario by 5. As can be seen in the right plot in figure 4 the modified five-pool dCache performance is identical to the one-pool dCache performance, so that we can conclude that the dCache performance scales linearly with the number of pools in the NFS v4.1/pNFS case.

3.5. Production use of dCache NFS v4.1 at DESY
Before the availability of SL 6.2, some systems were manually equipped with a special kernel including NFS v4.1 support. These clients are used in production for data import of some beamlines at the PETRA 3 storage ring. Some CFEL and HASYLAB groups are using such
specially prepared clients for archiving remote data. With the availability of SL 6.2, usage of NFS v4.1 for a more diverse variety of use cases such as end user analysis has just started. The general experience with both client and server is very good, in terms of usability, stability and performance. We do not expect a backport of the NFS v4.1/pNFS clients to the kernels used in the SL 5 series. It is therefore important that HEP switches to SL 6 as soon as possible to take advantage of NFS v4.1 capabilities.

3.6. Commercial vendors
In the beginning of 2012, NetApp provided us with a test installation for NFS v4.1/pNFS testing. This comprised a NetApp 3270 with ONTAP 8.1rc3 cluster mode. We have performed different tests, both performance, stability and feature oriented. We will not show results of these tests as they were superseded by the general availability of ONTAP 8.1 cluster mode on April 19th 2012 [9]. Due to time constraints, we were not able to repeat the tests with the new version. It is however very important to notice that the first commercially available product featuring NFS v4.1 and the protocol extension pNFS is released.

3.7. Other mounted filesystems than NFS v4.1
DESY recently has purchased an IBM Sonas system, designed in an initial phase for a total capacity of 500 TB and 300k IOPS. The intended use is the NAF (National Analysis Facility) as a replacement for Lustre at the DESY Hamburg site [10]. Users need a storage system with a high throughput and low latency for their analysis. NAF users also request the convenience and ease of use of a mounted file system for parts of their analysis data and workflows. When the final setup of Sonas will be done, it will be mounted using NFS v3 to the NAF clients. At time of CHEP 2012, only selected early-bird users were accessing the system as the final hardware and firmware configuration was not yet achieved, hence no performance results and user experience can be reported.

4. Summary
Many improvements have been performed by many actors in the NFS v4.1/pNFS ecosystem: dCache server development has continued, the first commercial server is available - more to come. NFS v4.1/pNFS has found its way into the vanilla Linux kernel, and into Linux distributions.
Other OS vendors also are working on clients. We have shown that NFS v4.1/pNFS presents advantages over other access methods used in HEP. We are waiting for a switch to SL 6 in the HEP world to take fully advantage of this protocol. Also other mounted filesystems are successfully used for HEP analysis.

References

Note: All HTTP links checked 06/22/2012.
[1] Kemp Y et al 2011 LHC data analysis using NFSv4.1 (pNFS): A detailed evaluation J. Phys. Conf. Ser. 331 (2011) 052010.
[2] Ozerov D Kemp Y 2012 The DESY Grid Lab in action, to appear in these proceedings.
[3] Yang W Introduction to the XrootdFS
   http://wt2.slac.stanford.edu/xrootdfs/xrootdfs.html
[4] RHEL 6.2 release notes http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/ch12s02.html
[5] McDonald A 2012 NFS v4 usenix ;login: February 2012
   http://static.usenix.org/publications/login/2012-02/pdfs/McDonald.pdf
[6] Kalinin S 2011 XRootD dCache Workshop March 2011
   http://www.dcache.org/manuals/2011/goettingen/20110314-02-xrootd-sergey.pdf
[7] Fuhrmann P et al. 2010 Hepix 19.4.2010
   http://w3.hepix.org/storage/hep_pdf/2010/Spring/fuhrmann.dcache+NFS.v4.pdf
[8] netem (network emulation) webpage
   http://www.linuxfoundation.org/collaborate/workgroups/networking/netem
[9] NetApp ONTAP 8.1 ClusterMode release notes (NetApp login needed)
   http://support.netapp.com/NOW/download/software/ontap/8.1/
[10] Haupt A et al. 2012 The DESY Grid Centre, to appear in these proceedings.