Experience with HEP analysis on mounted filesystems

Patrick Fuhrmann, Martin Gasthuber,
Yves Kemp, Dmitry Ozerov
DESY IT & dCache.org
CHEP 2012, 05/21/2012

HELMHOLTZ ASSOCIATION

DESY

A brief (and maybe incomplete) overview on access protocols

- **dCache**: dcap, ftp, xrootd, NFS v4.1, HTTP/WebDav
- **CASTOR**: RFIO, ftp, xrootd
- **DPM**: RFIO, xrootd, ftp, (NFS v.4.1)
- **EOS**: xrootd, ftp
- **XROOTD server**: xrootd (ftp/SRM via Bestman)
- **Lustre, GPFS, HDFS** : Filesystem (ftp/SRM via StoRM)

- Many different protocols in use – some protocols more or less tied to one or few storage products.
- The majority of HEP analysis done via HEP home-grown protocols – clients provided and maintained by HEP community

DESY

General benefits from using mounted filesystems

- Mounted FS provide POSIX IO
  - “Can I run Matlab on the FS?”
- Using most common file access method
  - Storage backend interchangeable
- Kernel VFS cache comes for free
  - HEP does not need to take care – VFS has greater persistency
- Files can be browsed and accessed easily from Linux (and some other OS)
  - Important for end-user analysis
- … performance!

- Fuse somehow in-between these filesystems and protocols:
  - can provide a filesystem for selected protocols without native FS. (e.g. xrootd)
  - Usually performance drop compared to native protocol
  - Two protocols for different usage scenarios (one for transfer, one for metadata)

DESY

Some reasons why one should care about NFS 4.1

1) High latency link performance
   - Batching of several components, reducing number of network ops, bidirectional RPC
2) Proper authentication and authorization
   - Kerberos, X509 under investigation, ACL
3) Introduction of sessions with NFS 4.1
   - Decoupling transport from client
4) Parallel NFS
5) Standardized and Industry backed
   - story goes on: NFS v4.2 waiting for standardization
6) Client & server available – from industry!
   - Real POSIX IO, caching provided by OS & tuned by experts, no apps modifications
8) In HEP: Funding secured
   - EMI funds NFS 4.1/pNFS in DPM and dCache, HGF (D) additional funds for dCache

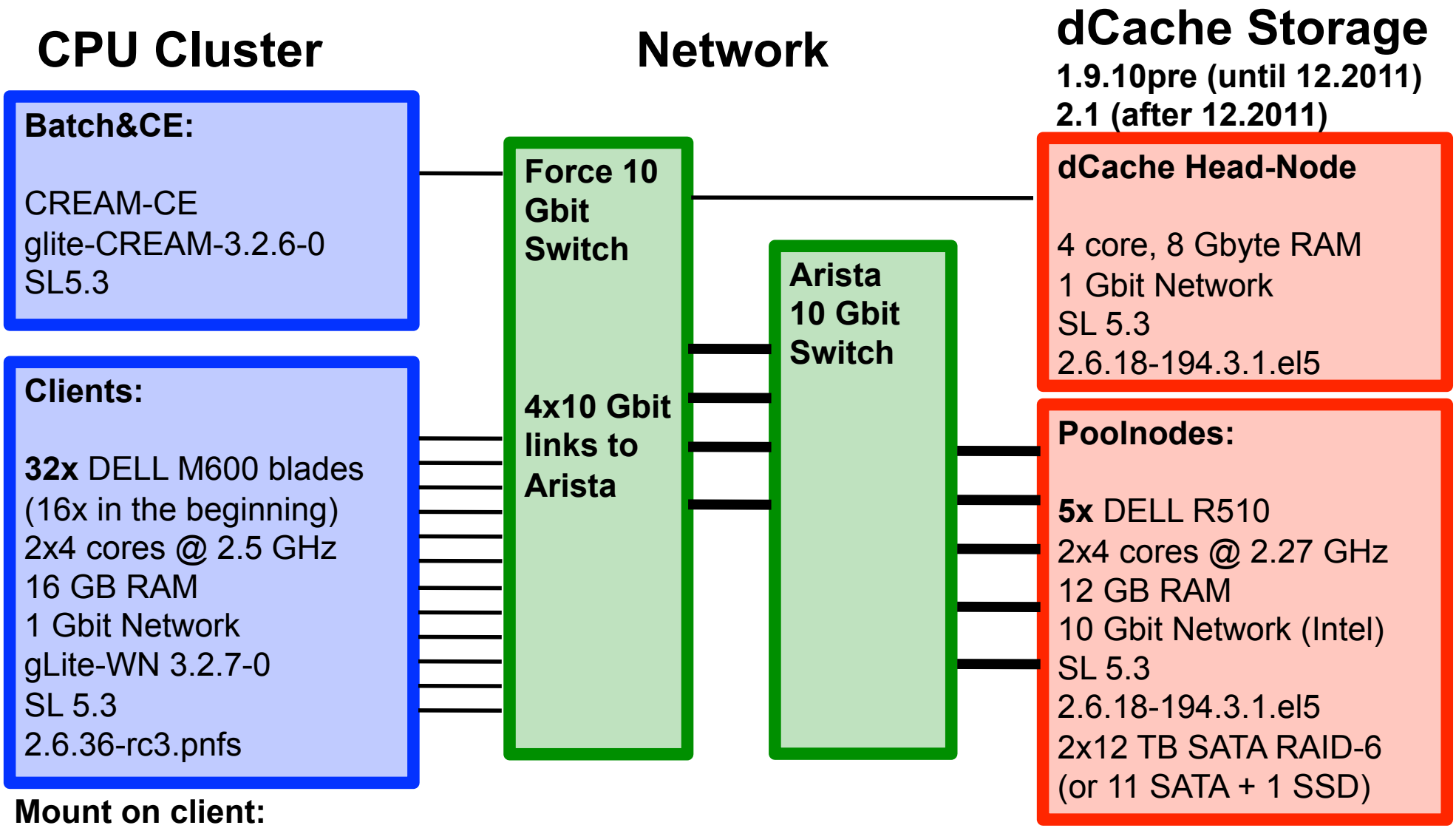
Status of NFS v4.1/pNFS: Clients and Server(s)

- Clients:
  - Linux Kernel: File Layout in vanilla kernel 2.6.39, block layout since 3.0
  - Linux distro NFS v4.1 (pNFS) in RHEL 6.2 as “technology preview” -> Also in SL 6.2
  - Windows 7 from CITI (file layout) published LGPL
  - Solaris client: Availability date not yet published
  - VMware hypervisor integrated client, not public yet
  - Windows 8: SMB 2.2 and NFS v4.1 client+server (Microsoft statement at SDC 2011)
- Server:
  - NetApp: ONTAP 8.1 ClusterMode (since 19. April 2012): File layout
  - dcache: 1.9.12 (released April 2011): File layout
  - DPM server development ongoing – not published yet
  - IBM, Panasas, BlueARC, EMC, Solaris working on prototypes – no release dates yet

➔ **Talk 250– T. Mkrtchyan, Tuesday 4:35-5:00**
➔ **Talk 253 – P. Millar, Thursday 4:35-5:00**

DESY

Evaluation: The testbed in the DESY GridLab



Mount on client:

```
dcache-head:/pnfs on /pnfs type nfs4 (rw,minorversion=1,rsize=32768,wsize=32768)
```


➔ Poster 503 – D. Ozerov, Tuesday

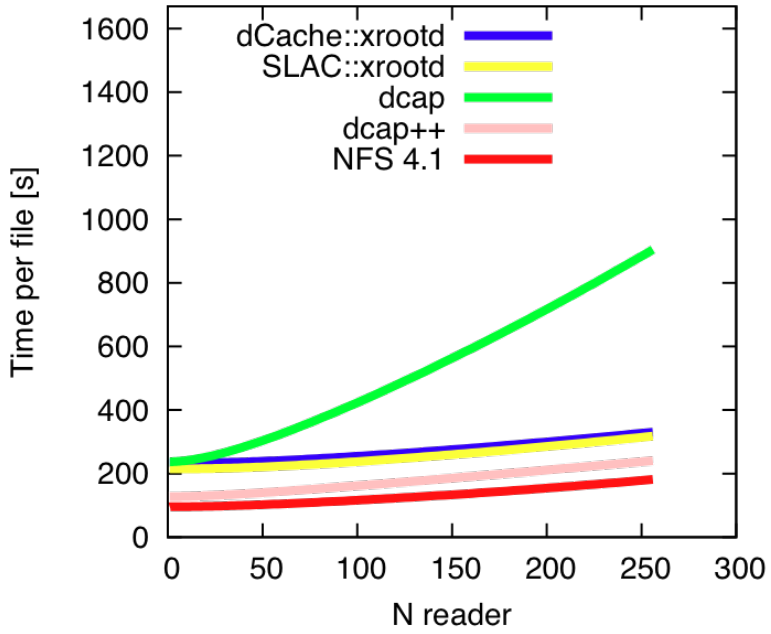
Comparing Protocols: Reading ROOT files

- ROOT version 5.27.06, compiled with dCap support
- Files provided by René Brun: `atlasFlushed.root` (re-organized files with optimized buffers) and `AOD.067184.big.pool_4.root` (some other original file) (flushed: 1GByte, original 1.3 GByte)
- Test script provided by René: simple script reading events: `taodr.C`
- Different test runs:
  - Reading with 60MByte TreeCache, or with 0Byte TreeCache
  - Reading all branches or only 2 branches
  - 1, 8, 32, 64, 128, 192 or 256 jobs running in parallel
  - Reading via NFS, dCap, xrootd (dCache server), xrootd (SLAC server) and dCap++ (a patched dCap with caching)

- Leads to eight different scenarios
  - Will show two on next slide

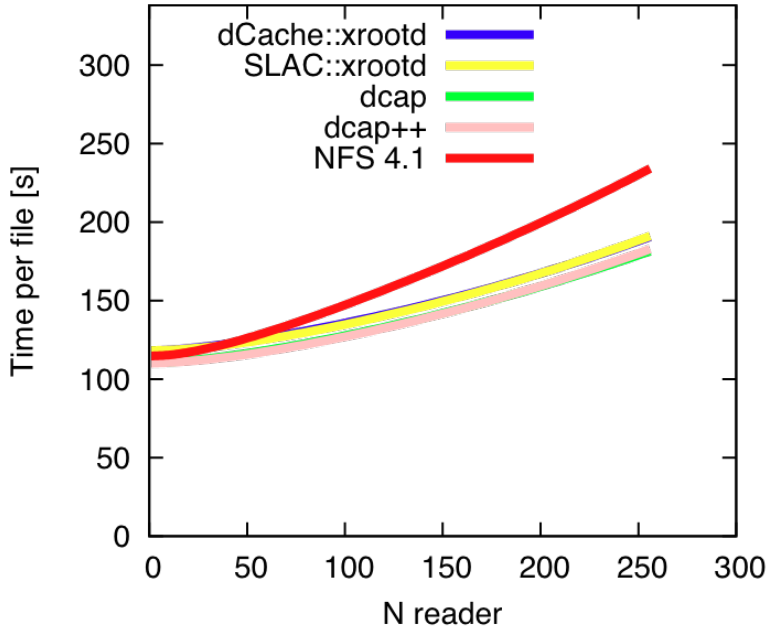
Results of protocol comparisons

- No clear winner: Depends on the read scenario
- NFS generally one of the fastest in this test setup



Optimized file, no TTreeCache, reading all branches

- VFS cache enhances analysis speed



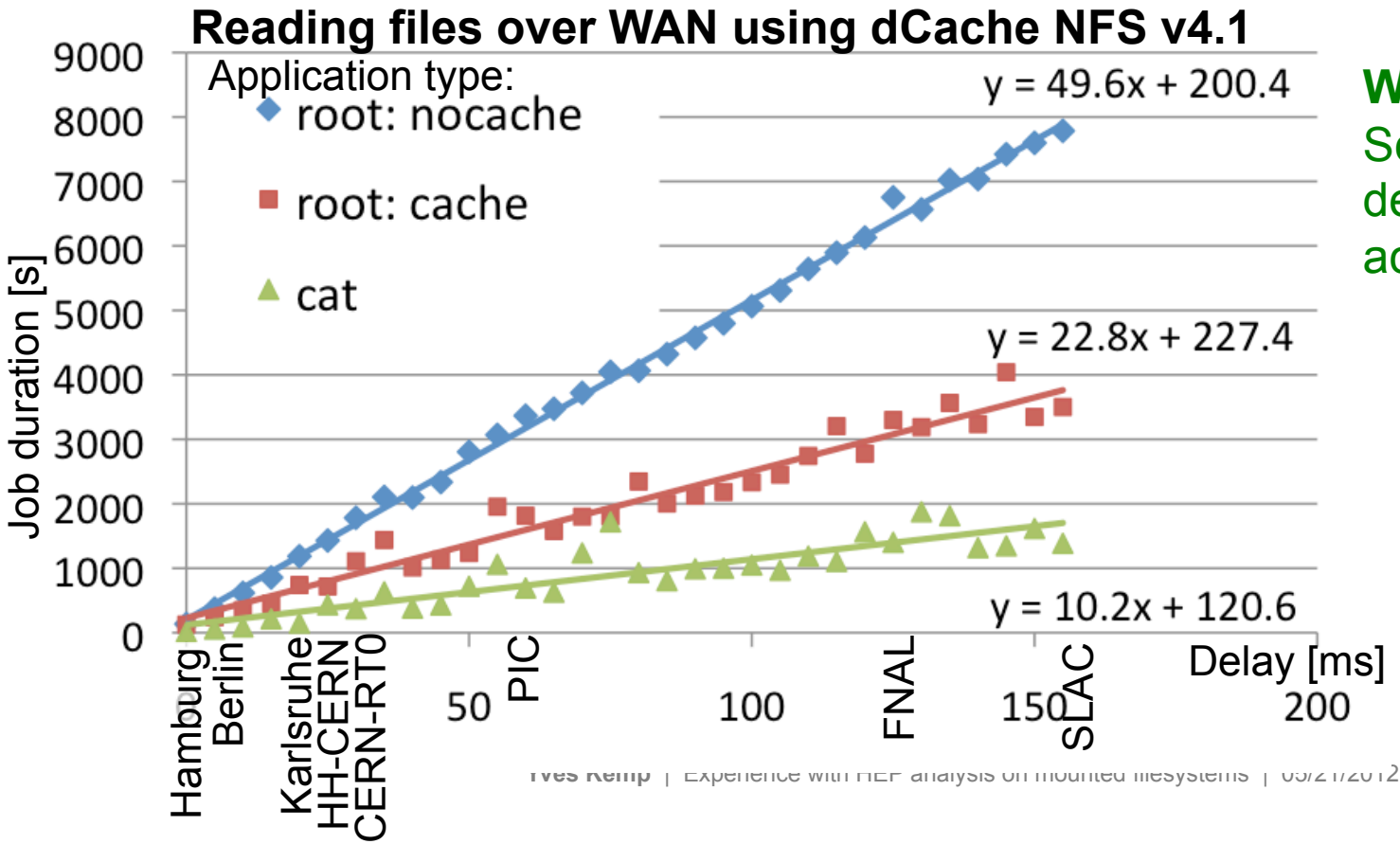
Non-optimized file, 60MB TTreeCache, reading all branches

- Scenario for which NFS v4.1 is slower than other protocols

DESY

Reading via WAN

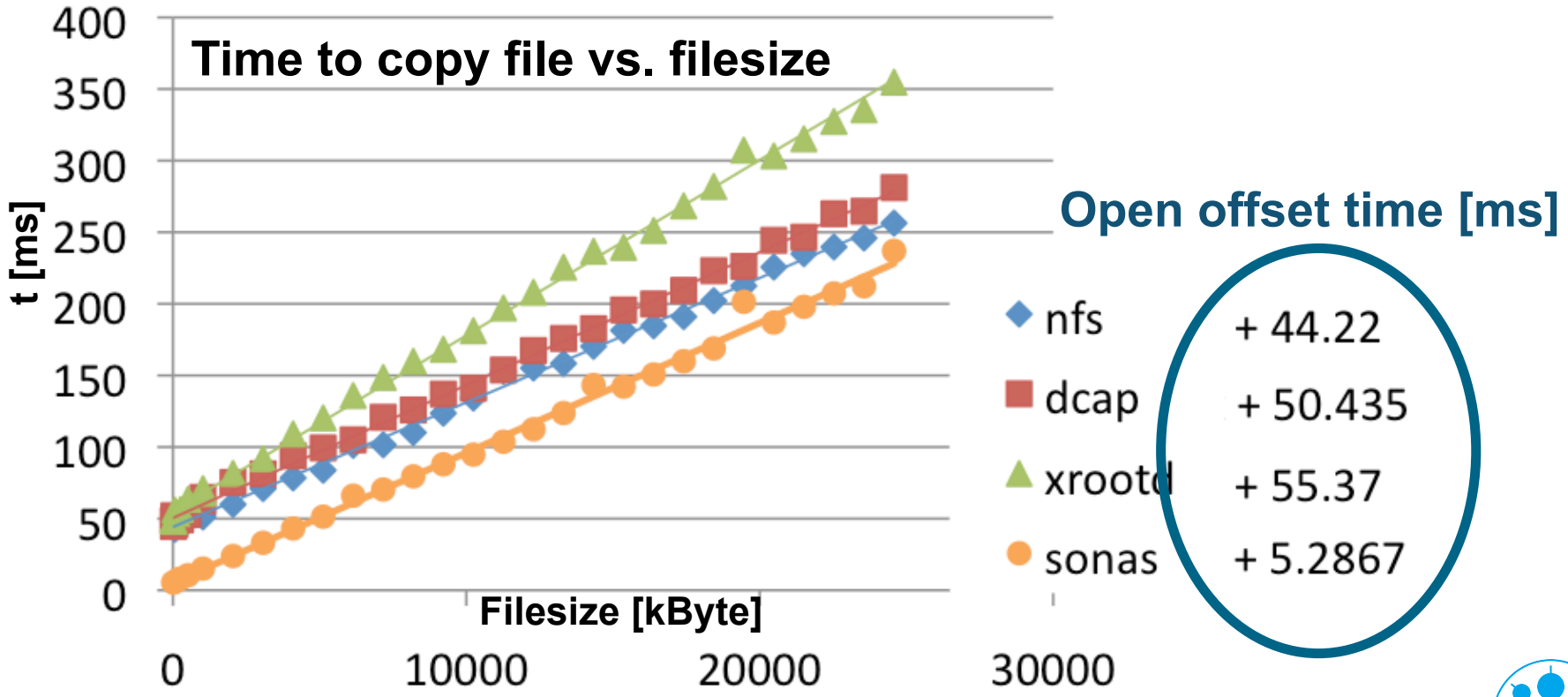
- Qualitative tests over real WAN:
  - WN in Hamburg, Small dCache in Taipeh / productive ATLAS instance Vancouver: Works
- Quantitative test over LAN+latency:
  - Using netem to emulate WAN (latency, jitter, packet loss,…)
  - Using GridLab as test setup



Works!
Scaling behavior depends on exact access pattern.

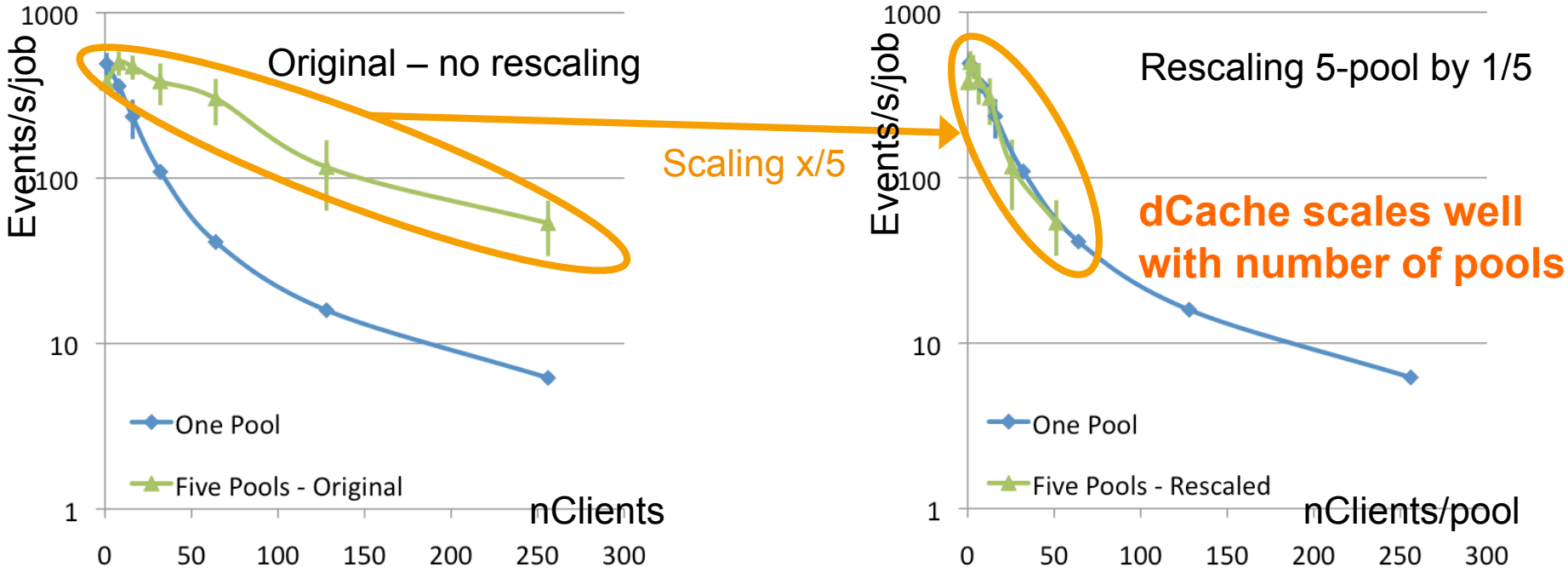
Time to open a file

- How fast can dCache open a file?
  - … will of course depend on load on Chimera DB – use idle GridLab
- Compare reading via NFSv4.1, dCap and XROOTD (file copy to /dev/null)
  - No GSI security involved
  - Compare with Sonas as an NFS appliance



Performance scaling over multiple pools w. dCache

- Does the aggregated dCache performance scale with number of pools?
  - Especially of importance since NFS v4.1/pNFS takes profit of distributed storage nodes
- Methodology: Have two dCache instances, test with ATLAS HammerCloud
  - One with 5 pools – test with 1-256 clients. X-Axis=nClients, Y-Axis=Events/Sec/Client
  - One with only 1 pool – test with 1-256 clients.
  - Scale X-Axis of 5-pool measurement down by 1/5 to have same nClients/pool



➔ Poster 503 – D. Ozerov, Tuesday

dCache and NFS v4.1 / pNFS: Qualitative statements

- Usage at DESY is increasing in Photon Science community
  - Data import from PETRA 3 beamline experiments.
  - Data archive for local groups (CFEL, HASYLAB) of remote data.
  - Photon Science is not bound to SL 5 – Better Linux client availability
  - Analysis on stored data over NFS v4.1 starting only with SL 6.2 – needed AFS
- General good experience
  - Active linux kernel development on pNFS part – many changes
  - Minor problems could be solved.

- **Linux client (SL 6.2) and dCache server for NFS v4.1 / pNFS : Works in production**

Other vendors: NetApp NFSv4.1 and pNFS

- NetApp 3270 with ONTAP 8.1rc3 cluster mode in early testing since beginning of year at DESY
- Results superseded by release ONTAP 8.1 Cluster Mode on 19.4.2012
  - No time to repeat measurements
- NFS v4.1 / pNFS finally made it to a commercial product!

Screenshot from NetApp ONTAP 8.1 Cluster Mode release notes:

File access protocol enhancements

This Data ONTAP release includes a number of new features and enhancements for file access and protocols ma enhancements.

.....

Support for NFSv4.1

Beginning in Data ONTAP 8.1, clients can use the NFSv4.1 protocol to access files on the storage system.

Support for pNFS

Beginning in Data ONTAP 8.1, parallel NFS (pNFS) is supported. It offers performance improvements by giving cli a NFSv4.1 feature and requires NFSv4.1 to be enabled.

Lustre and Sonas for National Analysis Facility

- NAF storage in a nutshell:
  - AFS for the small files, O(1GB)/user – global file system
  - dCache for the large common datasets (currently no NFS v4.1 mount)
  - Currently Lustre in-between, O(1-10TB)/user, mount using Lustre kernel module
- In 2011 – looking for replacement for Lustre in Hamburg
  - Staying with the concept of mounted filesystem for low-latency & high-BW user IO
  - Strong user request to have the convenience of a mounted FS for (some) analysis data
- End 2011/Beginning 2012: Purchase of IBM Sonas
  - ~500 TB size / 300k IOPS
  - Will be mounted via NFS 3
  - Currently early-bird Usage – awaiting final HW config and upgrade to Sonas 1.3.1 for public availability

➔ **Poster 375 – M. Gasthuber, Tuesday**
➔ **Poster 213 – A. Haupt, Tuesday**
➔ **Poster 288 – S. Gonzalez de la Hoz, Tuesday**

Summary and Outlook

- NFS v4.1 / pNFS is there
  - Clients are available: SL 6.2 vanilla kernel.
- dCache NFS v4.1 / pNFS server is ready for production
  - Waiting for HEP code / Grid-WN working on SL 6
  - Some non-HEP communities already using it for production.
- NFS v4.1 / pNFS: First *commercial* server is there!
- … and other mountable filesystems also exist and are used for HEP analysis!

Overall quite “boring” statement:

Mounted filesystems simply work for HEP analysis

DESY