Challenges choosing storage systems for experimental data

Nick Rees

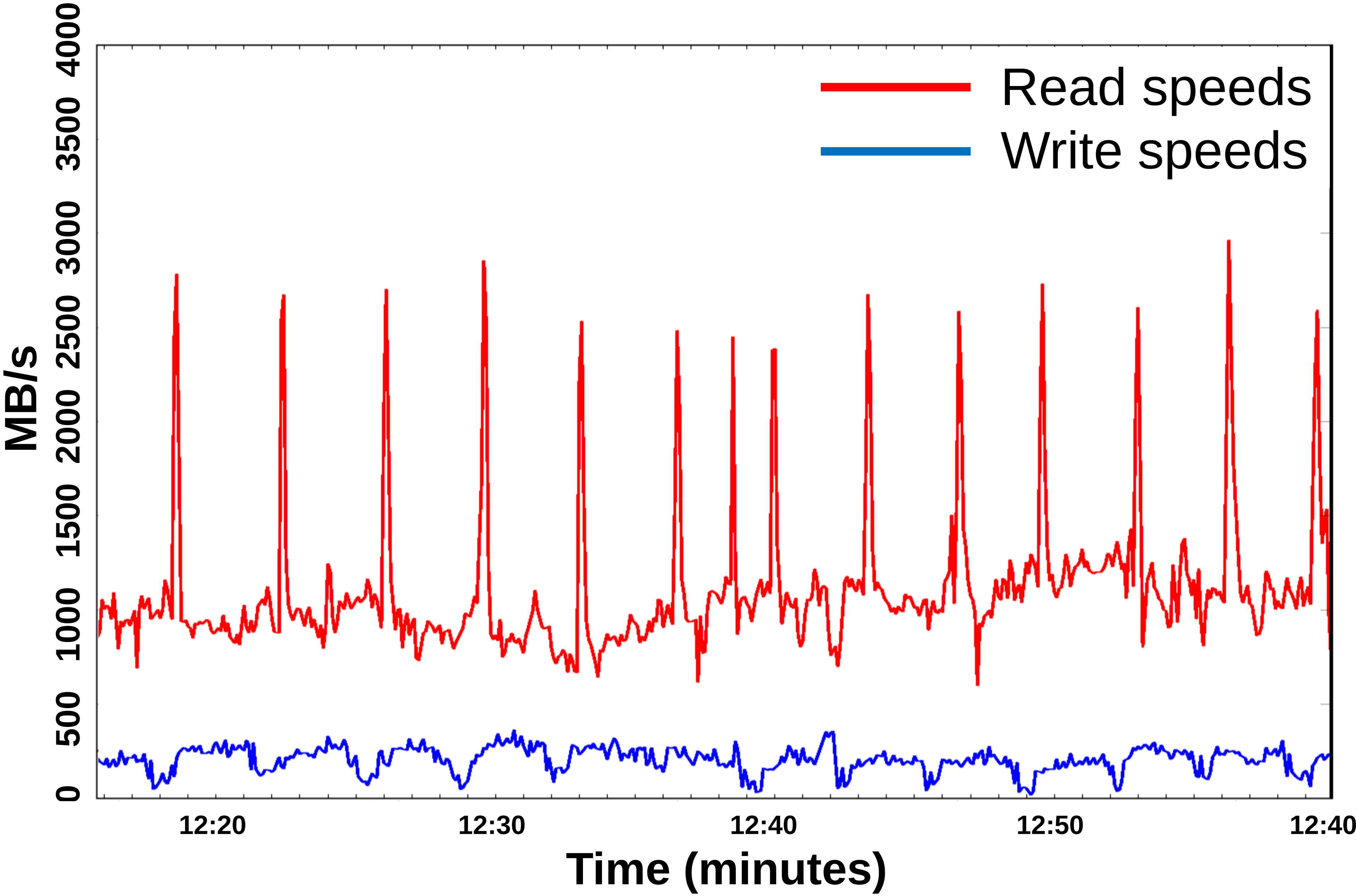
Overview

- Fast parallel storage
- Comments on storage purchases
- General purpose storage
- Caveats for large storage systems.

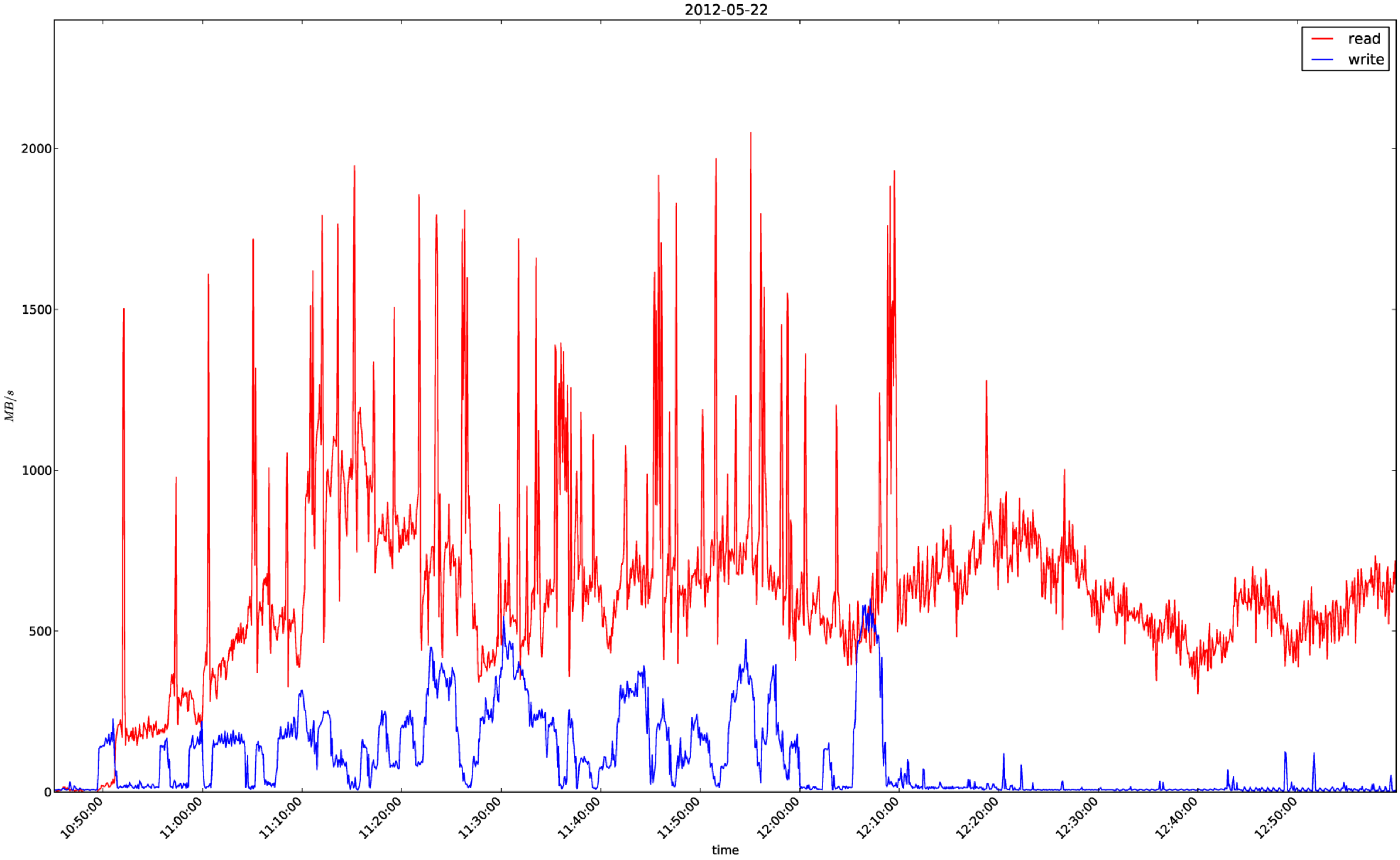
Fast Storage

- First storage (2006) was installed separately all on beamlines
  - slow (30 MB/sec) and difficult to manage
- Bought central Lustre/DDN system in 2008
  - 3 GB/sec
  - worked OK for MX and cluster processing
  - had problems with metadata and small files
- Bought second Lustre/DDN system in early 2011
  - 6 GB/sec
  - Faster metadata
  - Used mainly for MX:
    - 3 x 25 Hz Pilatus 6M (150 MB/sec each)
    - 1x30 Hz Pilatus 2M
    - 1 ADSC system
- Old system is still used for for tomography
  - 4 Hz PCO4000 (90 MB/sec)

Data Rates while Data Taking



A less regular example

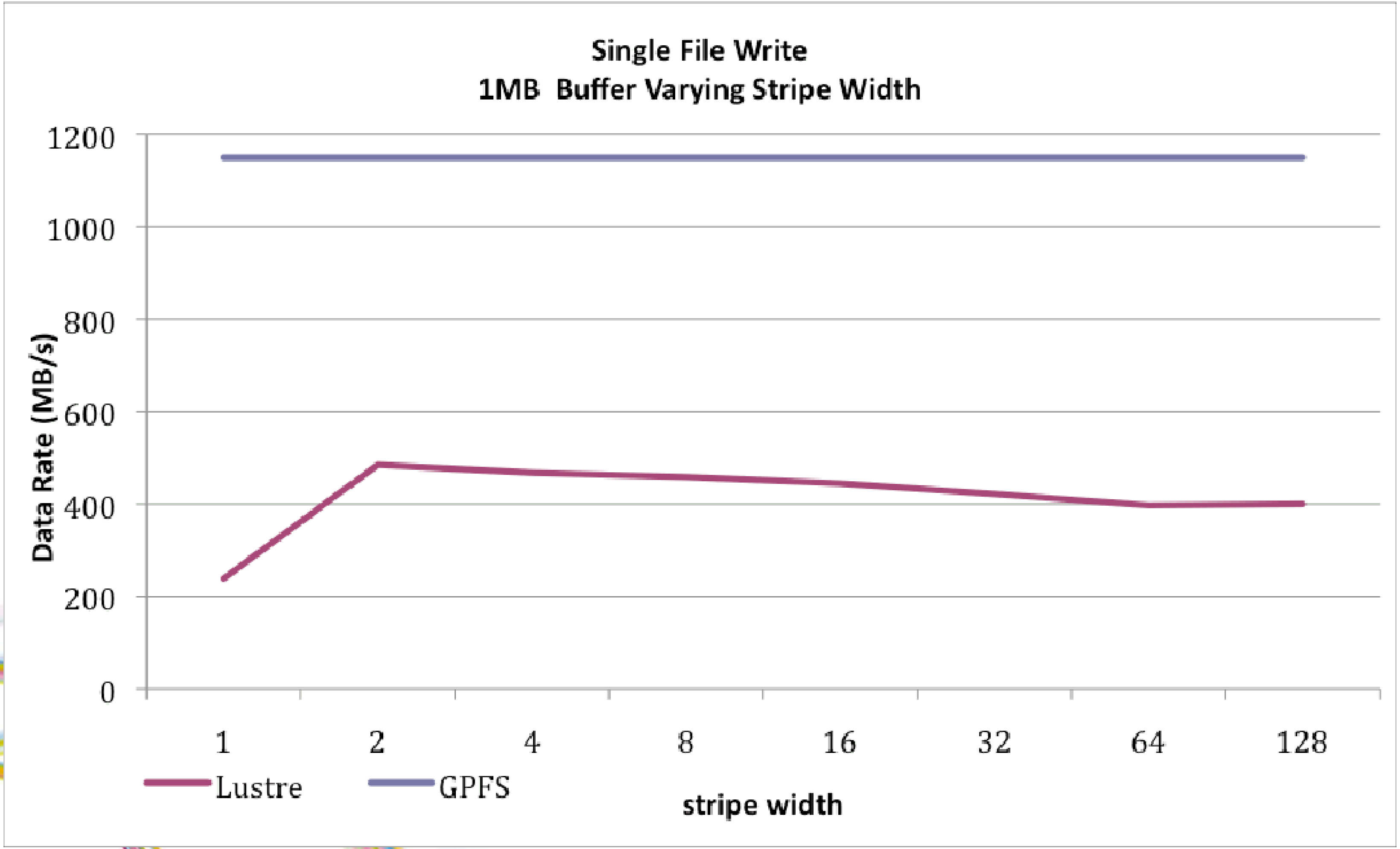


Next challenge

- Faster detectors
  - 100 Hz Pilatus (600 MB/sec write).
  - Tomography detector with 2 PCO.edge systems writing simultaneously (2x900 MB/sec).
- Looking at next generation DDN system
  - SFA12K-20 ~ 16 GB/sec
  - SFA12K-40 ~ 32 GB/sec
- But problem is with client write speed.
  - Lustre 1.X client write speed is limited to ~400 MB/sec (or ~750 MB/sec with checksums off).
    - One core in the client is pegged at 100% usage.
  - GPFS is much better (~3 GB/sec)
  - Lustre 2.0 is meant to be better (we are just starting testing)

diamond

Client Write Speed



From: R Hedges K Fitzgerald M G and Stearman D “Comparison of leading parallel NAS file systems on commodity hardware” https://e-reports-ext.llnl.gov/pdf/457620.pdf

Storage Purchases

- Storage purchases are complicated because:
  - Storage is expensive and complicated.
  - Procurement want you to go to tender to prove cost competitiveness.
  - Vendors have a “bid registration” process in which they will guarantee one supplier 10-20% better pricing than anyone else.
  - Vendor price lists are fairy-tales. A good price is normally > 70% discount off list prices.
- The result is that if you know what you want and the vendor is interested, you can get better prices by negotiation than by tender.
  - and save everyone a lot of time and effort.

New storage system

- In late 2011/early 2012 we tried to buy a general purpose storage system.
- Requirements were:
  - Network attached storage,
  - Reasonable performance (not high speed or parallel)
  - Windows (CIFS) and Linux (NFSv3) clients,
  - ACL’s, snapshots and replication
- All suppliers claimed they could meet all technical requirements before the bid, but in the end only one was left
  - We required draft Posix 1e standard ACL’s, and most suppliers provided NFSv4 ACL’s

Posix 1e vs NFSv4 ACLs

- Posix 1e ACL’s
  - Use Unix uidNumbers and gidNumbers internally
  - Are order independent
  - Were available on Solaris since mid 1990’s and on Linux for at least 10 years
  - Were never ratified as a standard.
- NFSv4 ACL’s
  - use user@domain strings internally
  - Closely matches Windows ACL’s (order dependent).
  - Needs Linux to provide uidNumber and gidNumber mapping functions.
  - Only recently available on Linux – not supported on many target file systems.

File system evaluation

- We spent 4 months evaluating the file system.
- Found numerous little problems – many claimed “to be fixed in the next release”
- Problems with GUI and command line not matching.
  - GUI was an add-on that never really worked
- Spent many days on phone with support in US, China and India.
- Ultimately found system hanging for long periods at times.
  - Turned out that whenever a snapshot was taken when a file had extended attributes the whole file system was locked while a copy was made of the extended attributes.
- Snapshots took 10 minutes with 38 million files...

diamond

The result

- Ultimately we rejected the product
  - No money, but a lot of time was spent.
- Existing systems were replaced with a short-term XFS/NFS solution
  - No replication or snapshots.
- Soon after implementation, we started getting compilations failing in non-reproducible ways with an error of:
  - “Value too large for defined data type”

File system sizes

- Unix file systems have a concept of an inode
- inode number is often the offset of the inode in the file system (in units of the inode size)
  - if the inode size is 512 bytes (2^9), inode address > 2TB (2^{41}) from the start of the file system is > 32 bits
- > 32 bit inodes creates problems with 32 bit system calls
  - Linux stat and readdir
  - VxWorks readdir
- Core 64 bit operating system software is safe, but you need to check your 32 bit binaries. For example:
  - VxWorks cross-compiler on Linux needs rebuilding with CFLAGS set to:
  - -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

Morals of the story

- Fast parallel file systems usually have slow metadata.
- Parallel file systems may be fast in aggregate performance, but single writer performance is limited by the client.
- Don’t believe the manufacturer’s – you need to test everything.
- The storage market is such that the tender process is often not optimal.
- Most commercial NAS systems use NFSv4 ACL’s and this isn’t mainstream in Linux yet.
- Large file system support is not just large files, but also large inode numbers.
- 32 bit applications may need to be recompiled with large file systems.

diamond