Fancy Filesystems

A look at some of the many options we have for serving up data to users.

With thanks to Winnie, Sam, Rob and Raul.

Disclaimer - I don’t know much, but this is about starting conversations.

Old School (1) - NFS/ClassicSE style

- A single server with a chunky volume, presented over nfs or behind a single protocol endpoint.
  - I'd say this is how we did things "back in the day", but I'd put money on everyone in this room having an nfs box or two sitting in their cluster - or even back at home!
  - The ClassicSE is less common.
- Retro is back in fashion though, and that's not necessarily a bad thing.
  - It's easy to have over 100TB in a 2U box.
  - Decent in-rack networks are commonplace, so these boxes can be well connected.
  - Easy to admin, well understood.
  - Xroot (or XCache) servers are the new ClassicSE
- Of course there are some limitations and concerns.
  - IOPs, Bandwidth and Resilience are all factors when dealing with single servers.

Old School (2) - The not so classic SEs.

The early naughties saw sites deploy what is a very griddy solution for their storage with DPM and DCache SEs.

- From a distance they both look very similar - protocol gateways on every pool server, a single metadata server pulling things together in a giant virtual filesystem.
- Very grid-specific solution.
- Easiest access via gfal (or lcg…) tools.
- Protocols very grid focused.

Rise of ZFS

In recent years some of us have been ditching raid cards for ZFS. Raid 6 is dead, long live RaidZ3!

- Although I for one do not think I’m leveraging enough of the ZFS features (or whether or not we can even leverage them).
- I’ve heard many praises for ZFS and it’s ability to recover where raid would not (just don’t reboot the server).
- Although ZFS servers might be a little more expensive (no raid card, but more RAM - at least in the Lancaster ZFS servers).

However this makes our ZFS servers JBODS with a bit of computing umph to them - which would make repurposing them down the line easier.

cvmfs

It would be remiss of me not to mention cvmfs when we're talking about fancy filesystems.

cvmfs removed at least one nfs mount from our compute - one that could often get hammered!

With experiments experimenting in disseminating data as well as software over cvmfs the lines blur further.

And maybe there are other squid-like solutions out there? (If you're being fancy you could describe Xcache as a squid-like solution I suppose…)

New users, new expectations

Astronomers, low-energy physicists, non-WLCG experiments - some under the IRIS banner, some coming from across the pond - all aren’t used to our particular paradigm.

- gfal-tools, x509 in general.
- Access protocols - might expect an Industry Standard.
- Expectations of data resilience.
- There’s a lot to be said for just being able to mount a volume.
- And then there’s these Object Store things

Protocol Soup, now with added “Industry” Croutons.

- GridFTP (how much longer?)
- XRoot (Xcache…)
- https/dav (are they the same?)
- SRM (phasing out)
- S3/Swift (lots of other communities using this sort of thing)
- And more that I couldn’t think of off the top of my head...

SRM/GridFTP were the Third Party Copy tools of choice, but we need to move on.

New School SE

So it’s looking like the future Grid Storage element will be:

- A big, high performance bucket of data.
  - Or a smaller bucket for cache-like services.
- With a (thin) layer of (griddy) access protocols over the top.
  - Endpoint advertisement likely to be done via something akin to entries in the gocdb/agis, and json-like information files stored directly on the server.

What do you know, STORM sites have been living in the future all along...

Filesystems A la carte

- CEPH
  - Subject of many a talk, an object store that can do many other things
  - Tier 1, Glasgow - a possible future SE solution for the medium-large Tier 2s.
- BeejeeFS
  - Apparently the leading parallel cluster filesystem (their websites says so).
  - I’ve heard it mentioned a few times, but has anyone considered a large scale application?
- Lustre
  - I actually neglected to ask our hosts about how their lustre was doing.
- GPFS
  - Or IBM Spectrum Scale as wikipedia informs me it should be called, since 2015.
  - Often out of our price range, but used at ECDF.
- HDFS
  - Part of the Hadoop “Ecosystem”.
  - Big on the other side of the Atlantic, only used at Bristol that I know of.

And of course there’s many more.

- Although my brain couldn’t dig any examples up over the weekend, does anyone have any to watch out for?
- Vip mentioned EOS

CEPH @ Glasgow

https://ceph.io/

Glasgow are trailblazing the use of CEPH at a Tier 2, using the Tier 1's experiences as a guiding star.

(Does that make Gareth, Gordon and Sam the three wise men?).

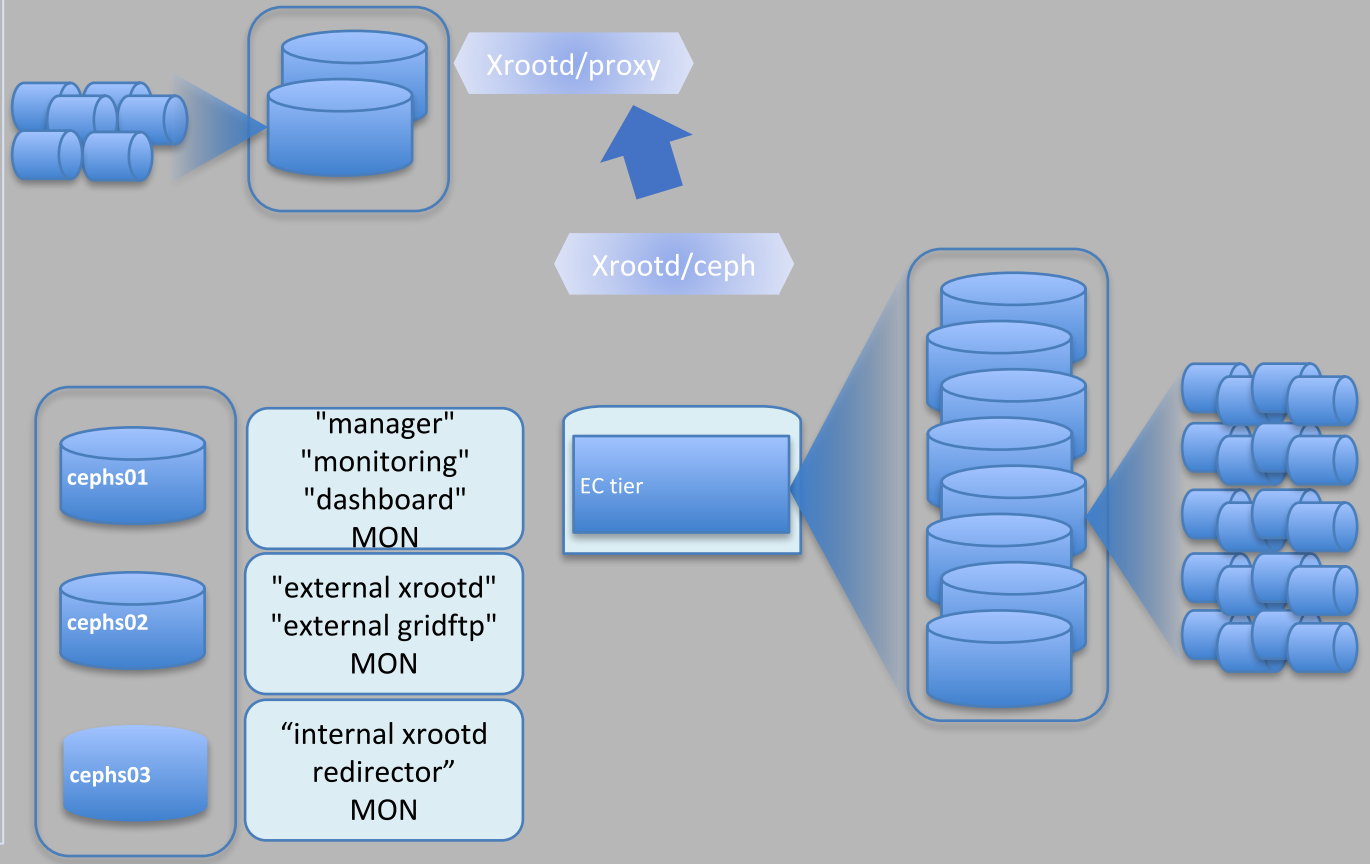
There are many motivations, but para-phrasing a half-remembered) chat with Gareth at at GridPP meeting "I can have storage that is usable by everyone, not just the grid".

Ceph

- Underlying “Ceph Storage Cluster” (a low-level distributed object store)
- “End-user” products supported on top of this as a common backend (you can have more than one per cluster, in their own pools):
  - Ceph File System (CephFS)
    - Posix-ish filesystem layer, with directories and metadata and stuff.
  - Ceph Block Device (RBD)
    - Block-device like layer, for hosting virtual machine images etc
  - Ceph Object Gateway (COG)
    - “full featured” Object Store with S3 and Swift API support.
- Can also write your own layers if you want to put something else on top of the low-level store.
  - RAL ECHO and Glasgow grid setup use plugins to do this, for “efficiency”.

GLASGOW

Post-DPM Transition: CEPH



Trade-offs

- Overheads / failure domains:
  - Traditional Grid Storage: file localised to a given server, striped over disks (resilience at disk level).
    - Stripe length ~ 12+2 to 16+2 ?
    - Resilience overhead ~14%?
  - Distributed Filestores: file striped across all servers (implicitly over disks), resilience at server level.
    - Stripe length ~ 8+2 to 10+3
    - Resilience overhead ~20%
  - Failure domain is also important: you can lose *all* the disks in 1 whole server, without losing any files at all.

Grid Interfaces

- XROOTD:
  - [Posix-like FS] supported natively
  - [Ceph Storage Cluster] supported by plugin (maintained by??)
  - [HDFS] supported by plugin (maintained by OSG)
  - [S3 stores] supported by plugin / some native support.

- GridFTP:
  - [Posix-like FS] supported natively
  - [Ceph Storage Cluster] supported by plugin (maintained by RAL?)
  - [HDFS] supported by plugin (maintained by OSG)

- HTTPS:
  - [Posix-like FTS] supported natively
  - [S3 interfaces] supported by Dynafed

Security (AuthZ and AuthN)

- GridFTP and Xrootd both support LCMAPS for account authorization.
- Real problem is mapping DNs to capabilities on underlying object stores.

  - We (at Glasgow) use LCMAPS ARGUS plugin to do DN level banning.
  - Then LCMAPS VOMS parsing to map VOMS role to a “proxy user”
  - (that user account exists to be a name for a set of capabilities)
  - We then use Xrootd’s authdb file syntax to map each proxy user to a set of capabilities (r,w,x,etc) on pools & object names within those pools.

BeegeeFS impressions

https://www.beegfs.io

Having no experience with I’m going to ask you lot what your impressions are, and turn over to Raul if he’s able.

Raul’s BeeJee thoughts

Niceties

- Fast(est?) parallel cluster filesystem
- linear scalability documented and demoed at USENIX/291(8/9)
- Does CEPH have a test case showing linear growth with increase of number of servers? - I'm inquiring, not doubting
- Excels on I/O intensive workload
- Also showcased in USENIX
- decent performance even on poor hardware
- I need it for my old Brunel test bed
- *Reportedly* scales from a few servers to cluster of thousands (?)
- Reported at USENIX: 8GB/s writes on 100G network plus HDD

Raul’s BeeGee thoughts continued

A few more technicalities

- Linux or FreeBSD
- runs on top of standard FS (xfs, zfs, btrfs)
- user pinning to server or set of
- replicated data with server buddying
- automatic recover after server downtime
- fault tolerance
- on-demand, real-time creation of parallel FS
- alternative for HDFS on Hadoop clusters (Hadoop connector)[https://www.beegfs.io/wiki/HadoopConnector]
- Could we use it as storage on compute nodes?

Lustre @ STORM sites (QM, until recently Sussex)

http://lustre.org/

As mentioned before, I neglected to ask ahead of time for some input from Dan about this, and I think he's in a different hemisphere. Maybe Terry or Alex could add to this? (sorry to put you on the spot)

The lessons learned by observing from the back seat over the years seem to be make sure your metadata/<insert protocol name> gateways have enough umph - and make sure that the users know just to use a cp rather then a gfal-cp!

HDFS @ Bristol

https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

- At Bristol the storage is on the compute (very hadoop-y).
- The HDFS volume is mounted across all compute nodes, from physical storage that is on all nodes.
  - All data is replicated, for resilience.
  - The mounts are read-only for “regular”, posix-like data access - writes are performed using hadoop tools
- The DPM SE uses some of this volume - but the dmlite plugin that allowed this to work is no longer supported.
- The future is using a xroot plugin to turn their SE into an XrootD server with an HDFS backend.
  - This is a popular solution in OSG, so this plugin will be around for a while.

HDFS @ Bristol (2)

Lukasz notes that their batch system (HTCondor) isn't "hadoop-aware", which sadly means that a lot of the benefits of HDFS can't be fully realised. But a HTC plugin may yet come.

There's also the problem that putting your disk on your compute means some CPU cycles are "lost" serving out data. Nothing is free!

Thanks to Winnie for pointing me to these slides:

https://seis.bristol.ac.uk/~phpwl/GridPP40_Hadoop_Bristol.pdf

https://seis.bristol.ac.uk/~phpwl/2019.10.15_Bristol_Computing.pdf

GPFS impressions

Rob kindly gave me his impressions of GPFS, from the standpoint of a “power-user” which I will now give to you verbally.

Other buckets of bits at Lancaster.

I asked my counterparts at Lancaster what they used (and planned on using) for serving data - mainly out of idle curiosity (and to weep at the £/TB others can afford).

The local home area mount on the shared cluster is currently a single **panasus** shelf.

This is soon to be replaced by a “curated **GPFS/Arcastream** solution”.

The University central filestore is a Dell EMC **Isilon** system of tiered storage. It was “spendy”, but seems to work well.

Getting to the point of all this.

To use management speak, there is a paradigm shift happening - particularly with respect grid storage.

As a community we need experience with these “new”[1] technologies, and from that document what works, what doesn’t - and then build **recipes** for others to follow.

What we don’t need is to duplicate effort or set off down paths that others have already tried, tested, and realised just lead to peril and doom.

Or at least wasted time and a lot of swearing.

[1] Most of these have been around for ages, but not used with a grid-shim over the top in a WLCG setting.