Brainstorming: Fermilab Storage Evolution

Goal

This document is the start of forming a plan for evolving Fermilab's storage setup for the future. Fermilab is supporting the storage needs of its many High Energy, Muon and Neutrino experiments and other facilities like the test beam experiments. Fermilab's storage facilities store experiment data and simulated data and provide access to it in a variety of ways. This document should be seen as a list of technologies and architectures that are promising to be part of a future storage system setup at Fermilab. This document does not present a concrete plan nor does it prioritize and choose amongst the listed technologies and architectures.

Participants

Gerard Bernabeu
Stu Fuess
Oliver Gutsche
Burt Holzman
Dirk Hufnagel
Robert Illingworth
Bonnie King
Jim Kowalkowski
Briant Lawson
Tanya Levshina
Dmitry Litvintsev
Dave Mason
Tim Messer
Gene Oleynik
Patrick Riehecky
Andy Romero
Ed Simmonds
Amitoj Singh
Jeny Teheran
Steve Timm
Yujun Wu

Trends

- We see a trend to establish more storage tiers in the overall setup. The current architecture consists of tape as the lowest, highest latency tier, with disk as the highest and lowest latency storage tier. In the future, we should investigate adding more tiers of more specialized storage technologies that provide different levels of latency at different cost points.
- Specialized storage tiers like mid-range block stores both for virtualized and bare-metal services and extra-high-bandwidth online buffers need special attention and need to be added to the portfolio.
- Many interesting storage solutions and technologies exists, need cost/benefit analysis including maintenance/operations costs, networking costs, conversion and life-cycle costs. A big topic is comparing commercial solutions with open source solutions. Same metrics apply. Every analysis needs to be accompanied by a full security assessment, especially checking for countermeasures protecting against insider attacks.
- Use cases define how storage needs to be architected to fulfill the needs of the experiments and scientists. A use case analysis is needed to determine future trends, determining with major experiments what they think their future storage needs will be.
- Industry is investing a lot of money into block storage technologies, transitioning from the purely file based storage systems. A detailed analysis of advantages and disadvantages of block storage technologies for our use cases is needed.
- POSIX access is still the easiest way of interacting with storage for scientists. It needs to be determined how POSIX access can be provided for which storage tiers and what the related support costs are.
- Interesting question that influences the design of storage system is the sharing between different entities (VOs). It needs to be studied if VOs are better served by providing them with own instances of storage solutions or if it is more cost effective (also from the maintenance/support point of view) to share storage instances between VOs.
- Investigate replacement of traditional hardware Raid-6 disk boxes with JBOD solutions with SW RAID or hardware based object stores and ethernet-attached disk like Seagate Kinetic.

POSIX Storage Solutions

1. Lustre
    - Lustre is a massively scalable, high performance file-system. It is tightly coupled to the kernel. It supports HSM and POSIX.
    - Comments:
        - Expertise at the lab exists (HPC) and its use is trending on other HEP sites
        - CMS used it many versions ago, tuned for infiniband, maybe now more mature with ethernet
            - Potential problem: clients have to be low latency
            - But works with 10 Gbps (needs to be studied)?
        - (Giant but safe leap approach): Replace Bluearc data disks with a Lustre appliance.
            - Many vendors now sell LustreFS appliances with multi-year support
            - These commercial solutions would not have to be assembled like the Lattice QCD lustre instance.
        - Potential issues
            - Requires a lot of maintenance
            - Problems with many small files
            - Need to explore HSM capabilities
2. GlusterFS
    - GlusterFS is a scalable network filesystem. GlusterFS is free and open source software. It is a Red Hat sponsored project.
    - Native clients are bundled with EL6 and EL7
    - Comments:
        - It is supposed to fully scale out, metadata included
        - Anything that depends on Redhat Clustering is more trouble to maintain than what it is worth (GFS, GlusterFS, CLVM, etc).
3. Beegfs
    - BeeGFS (formerly FhGFS) is a parallel cluster file system “developed with a strong focus on performance and designed for very easy installation and management.” (from http://www.beegfs.com/)
    - Comments:
        - Similar to Lustre and should only be looked at if Lustre wouldn't work
4. Clustered HNAS
    - (NFSv3 NFSv4, smb) for onsite, low-latency, general purpose interactive work
    - Mature, proven, reliable productized technology on the server-side
    - Mature, proven reliable (built-in) client-side support (Linux, Mac, Windows)

- Low total cost (dev-tinker:0, ops:<1 FTE, disks:slightly-higher-than-rock-bottom, nas-appliances:reasonable … only a small number of nas gateway appliances are needed)
- High availability, High performance (when implemented properly and used for intended role)
- Easy and efficient storage management / allocation
- Feature rich, snapshots, replication
- Planned improvements
  - SSD usage will be increased: meta data and Tier-1 cache
  - Planned 10% of SSD
  - Would allow to match the backend to the frontend capabilities

5. GPFS
   - General Parallel File System (GPFS) is a high-performance clustered file system developed by IBM
   - Comments
     - It is widely used nationally (for example NCSA) and internationally (for example Italian Tier-1 in Bologna)
       - NCSA is happy with it, 2-3 people exclusively for maintenance and operations
     - It is expensive
     - FNAL looked at it before, decided against it
     - Potential issues
       - Cannot be subdivided into enough pieces
       - Striping means that loosing a storage unit causes loss of many files

POSIX-like Storage Solution

1. dCache
   - xrootd, nfs v4.1
2. CephFS
   - Ceph Filesystem (Ceph FS) is a POSIX-compliant filesystem that uses a Ceph Storage Cluster to store its data. It is a red hat sponsored project.
   - Comments
     - Question if it is production quality?
       1. Website says: CephFS is production quality for single name space, concerns though about metadata backup

Non-Posix solutions

1. CEPH
    - Ceph delivers object, block, and file storage in one unified system.
    - Plugin structure. Many plugins have been developed for it (xrootd, gridftp, etc).
    - Comments
        - With proper plugins it's usable for physics data.
        - Lots of traction and potential
            - Some laboratories (like RAL) are thinking of production deployments for physics data soon (before end of 2016).
        - Could use it as block storage for VMs that need large amounts of disk (iscsi mount style)
        - Same CEPH deployment can be used to have 'toy' cephfs instances.
        - Native clients in EL7.
        - The object store is supposed to be good
        - Potential issues
            - This seems to be the FS du jour but is it really stable?
            - manpower-intensive and still rather bleeding edge
2. EMC block storage solution
    - better than Ceph, but expensive
3. EOS (https://github.com/cern-eos/eos).
    - From the website: “EOS is a software solution that aims to provide fast and reliable multi-PB disk-only storage technology for both LHC and non-LHC use-cases at CERN. The core of the implementation is the XRootD framework which provides feature-rich remote access protocol. The storage system is running on commodity hardware with disks in JBOD configuration. It is written mostly in C/C++, with some of the extra modules in Python. Files can be accessed via native XRootD protocol, a POSIX-like FUSE client or HTTP(S) & WebDav protocol.”
    - Comments
        - Large investments by CERN, other sites are using or testing it
        - It offers very good xrootd support and supports HSM.
        - With some development effort on our side can be a great solution.
        - CERNs plans work out well the user experience could be very good
        - Should keep an eye on it if resources are available. Useful both for physics data and interactive usage of physics data.
4. dCache
    - System for storing and retrieving huge amounts of data, distributed among a large number of heterogenous server nodes, under a single virtual filesystem tree with a variety of standard access methods.
    - Software defined storage with
        - Flexible, policy driven data flows and QoS

        - Hot spot detection
        - Policy driven data replication
        - Rich and easily expandable Authentication and Authorization Infrastructure
        - Support for industry standard and domain specific transfer protocols (xrootd).
        - Good support for HSM
    - Comments
        - The fact that files are immutable may be an issue for some use cases.
        - Could be used on top of block storage solutions (eg: CEPH) that would take care of lowering the $$/TB.
            - I believe there are some development efforts to be used on top of block storage solutions that I'm not following but last time I heard about were promising.
            - The development commitment was not strong last I heard from Fuhrman.
            - If dCache just uses CEPH distributed block storage, all data will be moved over the network twice, doubling the network requirements.
            - There is effort underway to get dCache to work with CEPH natively (via RADOS API). Results from a prototype will be presented at CHEP 16.
5. Cleversafe(recently acquired by IBM to build a storage for public and private cloud):
    - High reliability with low storage overhead based on erasure coding
    - High availability, fault tolerance and performance stability under adverse conditions such as network outages, hardware failures (particularly disks)
    - Unlimited scalability with a single namespace, could grow in size transparent to applications
    - Full S3 support and Swift OpenStack open standard
    - Upcoming NFS and CIFS support for POSIX like access with S3 or Swift access to the same data
    - Current implementation is not efficient for data mutation but will be improved in the future
    - Built-in and pluggable authentication schemes
    - Easy management: deployment, monitoring, configurability
    - Geographically distributed deployments
    - High performance for large objects
        - IOPS constrained small object performance
    - Significant low prices compared with other commercial products
    - Software only with certified vendor’s hardware or full software/hardware option
    - Gartner 2016 market leader in object storage
    - Biggest object storage installments (100s PB), referenceable large clients Shutterfly, Comcast

- No single point of failure
- Comments
  - It is a tape replacement but is also cached file retrieval (“slow disk”)
    - Example discussion with Shutterfly: deep-store is cleversafe, upper-tiers are disk systems like bluearc
  - Pushing NFS now
  - Selling software separate from hardware
  - IBM requested meeting with Amitoj to talk about cleversafe ➜ will be setup to include more people

6. Cloud based solutions
   - Two purposes for cloud-based solutions: Off-premises to support work on commercial clouds, and on-premises emulation to have similar protocols supported.. Could be small installations for test or large installations for enterprise storage
     - Meant to be a service ➜ storage as a service
   - Azure
     - Lab already using this and paying for it--Microsoft OneDrive
   - Amazon S3
     - can buy service from Amazon or use a number of appliances or open-source emulations on-premises
   - Scality
     - Similar architecture to cleversafe (erasure coding, object storage, ring namespace)
     - Good performance on small objects (avoid erasure coding on small objects)
     - Posix access is claimed to be mature
     - Appliances use SSD drives for better performance
     - No single point of failure: Uses distributed hash map
     - S3 support
     - A contract with Los Alamos (size?)
   - OwnCloud
     - Used by DESY--web frontend that can be applied to many storage backends including dCache
7. Panzura
   - Caching NAS with cloud backend
8. MarFS,
   - hybrid storage system that uses an object based store for data storage and GPFS for namespace metadata (developed at Los Alamos National Lab)
     - Described as "A scalable near-POSIX metadata file system with cloud based object backend"
     - It keeps the parts of POSIX that end-users appreciate for data management (chown, chmod, rename, mv, etc) without inheriting the

complexity of managing POSIX semantics for data handling. It leverages on cloud storage semantics to scale out easily.
        - https://github.com/mar-file-system/marfs
        - http://storageconference.us/2016/Presentations/DavidBonnie.pdf
9. IRODS
    - implements data virtualization, allowing access to distributed storage assets under a unified namespace
    - enables data discovery using a metadata catalog
    - Existing plug-ins support a variety of hardware, communication technologies, database technologies, and storage topologies. Templates are available for new, custom plug-ins.
    - Storage resources could be UNIX FS, S3, Ceph, universal mass storage
    - Native iRODS password access, Pluggable Authentication Module (PAM),Grid Security Infrastructure (GSI)
    - Used by many academic and gov institution, installments 6PB, 1K users
10. Current VMware based VM environment storage:
    - Storage and Virtual Services currently leverages COTS block storage for its VM storage.
    - All VM storage is hosted on this block storage and VMware Enterprise level licenses are used to allow live VM migration between LUNs on a single storage array as well as between LUNs on separate storage arrays.
    - This provides exceptional VM mobility as well as resiliency.
    - Standard procedures like host maintenance, host configuration, compute and storage capacity additions and changes cause virtually no impact on running VMs.

Data Transfer and Access Protocols Landscape

1. NFSv4 (v4.1, 4.2 and beyond)
    - actively developed
    - built-in client-side support
    - users are familiar with the technology
2. HTTP/WebDAV protocol
    - Overwhelming majority of the data transfers across the WAN are using http, more sophistication added to WebDAV protocol
        - Domain specific protocols like GridFTP and xrootd gradually dying off.
    - Additionally HTTP based approach provides easy interoperability with existing cloud solutions like Amazon S3.
3. S3
4. CDMI
5. Not listing the established protocols we use like GridFTP, ...

Architecture

1. one single shared storage system vs. multiple similar/identical storage systems (eg: one per VO)
    - Comments:
        - General opinion: **Per-VO storage system is preferable**
        - Multi-VO storage system is very difficult to manage and VOs can interfere with each others work.
        - Per-VO storage system is not very cost effective, especially when experiment needs are not well known.
        - Seamless horizontal scaling with minimum VO-specific configuration overheads helps to alleviate cost effectiveness penalty of Per-VO storage systems
        - Requires cost analysis of Multi-VO vs. Per-VO storage systems
        - Large experiments, e.g., CMS, can have its own VO storage system. It will be hard for many small experiments (VO) due to cost, utilization and admin considerations. May be helpful to study the automatic, dynamic storage resource (capability, network, IO) utilization to better suit multiple VO need. It should be achievable---cloud computing is already doing this (for many different users).
2. Automation like we have for WN
3. Allow to federate with other globally distributed, possibly heterogeneous, storage system using agreed on protocols
    - Maybe this is a bit above the level we want to go here, but this is something to look at later
    - More meant to discuss the inter-site storage solutions
    - Global or federated namespaces are part of this discussion
4. POSIX vs. Non-POSIX
    - Other labs (eg CERN) do have POSIX (like) access to physics data on their midterm schedule
        - and I believe it's an important feature.
    - Data globalization and sheer amount of data distributed across the world call for non-POSIX protocols.
    - POSIX compliance is becoming more and more a limiting factor for any modern storage solution and support for it is introduced only to support legacy applications that rely on POSIX.
    - Industry moves to object store, read-modify-write the whole object (or file) scheme which is far more efficient at large scale.
    - Having POSIX access can be very valuable, but experience so far shows that this is an all or nothing decision. Once the capability of POSIX access is there, due to the convenience of it, eventually most/all users will want to use it. As such this either scales to your whole infrastructure (potentially even allowing mounting

it on all worker nodes), you have hooks to severely restrict the amount of use or you might as well completely abandon it and not provide it. The latter is perfectly fine IMO as long as you provide some other smaller scale storage system with POSIX access that can be used for testing, debugging and (small scale) interactive work. Requiring POSIX access at full computing scales is a cop-out IMO for users that are too lazy to adopt their procedures and workflows from the interactive to the production use case.
    - About non-POSIX access, different storage systems allow different access paths and it's not guaranteed that every user software will be able to read from a storage system natively. Being able to at least copy the data a job works on into local storage on the node or into a scratch space area could work around a lot of these issues though. Users running at smaller scales should be ok with this, users running at massive scale should invest in supporting native reads from the storage system. We should of course list current and expected requirements and make sure the chosen technology supports most of it, but since this is an outlook into the future, this will only get us so far.
5. Tiered storage
    - following HPC developments/deployments
    - Current tape storage architecture is over 10-15 years old. Now the disk storage cost (using JBOD+SW Raid) is much lower.
    - Could we add a layer (similar to disk-style cloud storage) and only use the tape as a real backup?
    - Underlying data storage architecture has to be tiered and storage system that runs atop of it must be able to take advantage of it.
    - Comment:
        - Hope is that tape, although cheap, will finally die off and we will have:
            - Tier 1 : Flash Storage, Flash memory (NAND), SSD
            - Tier 2 : High performance rotating disks in either h/w RAID or s/w RAID configurations
            - Tier 3: Low performance rotating disk, that turn off when not in use. Shingled disks.
6. Small files and workflows
    - Experience shows that small file aggregation encourages users to write a lot of small files into storage system
    - Discourages them from using file catalogs (a directory tree and file names becoming catalogs)
    - Leads to namespace database growth
    - Need a solution that tackles the small file problem at scale.
    - Can we get rid of files completely? Would computing models incorporate different strategies if technology is available?
7. Networking

    - While identifying an evolution of the Fermilab MSS we should not forget a key component which is "behind-the-scenes" such as campus-wide networking which can be a bottleneck when moving data across campus.
8. Vendor solution (commercial, off the shelf) vs. DIY
    - Are we looking at costing of open source vs. commercial, all effort included
    - Cost of conversion ➜ if you move away from dCache or Enstore, what is the cost
    - Cost/Benefit analysis
    - Cost of supporting multiple technologies

Security Analysis

- Modern storage solutions provide multiple security services: authentication, authorization, auditing service, encryption service and access controller supported by robust technologies
- However, there are security gaps in the implementations, specially the lack of countermeasures applied for insider attacks.
    - For example, the administration interfaces are not being built using good security practices and can lead to unauthenticated and unauthorized access by exploiting known vulnerabilities.
    - See http://www.theregister.co.uk/2016/04/18/gartner_object_storage_rankings/
- Most of the solutions provide a good set of base security capabilities, but a deeper analysis is needed to establish if these solutions are susceptible to attacks where the intruder has taken the storage device.

Storage Hardware

1. Disk: Identify current MSS (Mass Storage System) bottlenecks and apply faster and better storage solutions to remove such bottlenecks.
    - This could include NVMe Solid State drives, CPU-controllers instead of hardware RAID controllers, Ethernet hard drives, etc. Note, in this case we are not replacing the existing software stacks (dCache, Enstore), so legacy and current users will continue to access the Fermilab MSS as before.
2. Disk: Look into JBOD solutions with SW RAID
    - ZFS, mdraid, etc
        - Already using in enstore small-file-aggregator and Lustre for LatticeQCD (ZFS over NFS). ZFS-over-NFS allowed us to use a mix of SSD and spinning disks where the SSD was used as cache besides 30% of RAM. This is a loose tiered storage approach.
    - Tiered storage will become more and more important
    - Should use an explicit combination of (MB/s)/TB and IOPS/TB figures in our procurement
    - Need to cut disk costs as they are dropping slower that the capacity requirements are increasing (see INSIC roadmap)

- Comments:
  - We are very conservative and that means we are getting less performance/$$. I'm not sure about TB/$$ but that figure may be slightly low for us too. I'm worried that using the same technology will lead us to higher IO contention as disks grow.
  - Some solutions (for example CEPH) do erasure coding across servers, so effectively RAID across arrays - more robust against failures beyond single disks.

9. Hardware-based object stores
   - one ARM-cpu per drive hardware.
   - Ethernet-attached disk: Seagate Kinetic
10. Long-term active archival storage back-end
    - Is tape the right solution going forward?
      - Tape media costs are dropping faster than capacity requirements (see INSIC roadmap).
      - Performance is a real issue. Accesses are dominated by seek times and front-end disk rates (heavy read/write spindle contention) resulting in 10s of MB/s tape i/o rates instead of their maximum performance of 250 MB/s.
      - Just trying to scale CMS up by a factor of 8 currently would require 240 drives at a cost of over $4.8M and would require another tape library to house the drives.
      - Eliminate disk contention between front-end disk and tape drive to improve utilization using spindle-less flash drives (SSD, PCIx flash cards, storage more like RAM) to buffer i/o between the tape and disk system. E.g. this could easily be done in dCache pool nodes with a modified real-encp and a smallish SSD.
      - Tape library vendors
        - Spectra Logic. Support LTO and IBM Enterprise drives
        - IBM - Support IBM Enterprise drives
        - Oracle Storage Tek - Support LTO and Storage Tek Enterprise drives
      - Tape media
        - LTO or Enterprise? Total cost / TB is the measure.
      - Tape storage system options
        - What are the options? Enstore, HPSS, CASTOR, ?
        - What are the costs and issues (development staff, sustainability, support)?
    - Alternative: fully tapeless back-end?
    - Alternative: adding in a new low latency tier.
      - Could be constructed of lower cost shingled drives with erasure coding e.g. Seagate, WDC, Spectra Logic and many others have deep storage solutions.

Storage Categories

1. Need for mid-range block stores (large servers, small-medium databases)
    - Back end VM image and data block store for easily migratable virtual machines is needed. Right now 3 different virtualization stacks (VMware, RHEV, OpenNebula) doing three different things.
2. Also mid-range reconfigurable store needed for bare metal services--mid-to-large databases (witness Gratia DB servers which are on-board the machine and all full at 4TB with no upgrade path).
3. Need for extra-high-bandwidth online buffers
    - Current protoDune plans for online system call for a system that can source and sink 20Gbit/s minimum, maybe much more than that.
    - May require SSD drives, certainly will require one of the fancy file systems. Often overlooked that solid state storage is of benefit where disk spindle contention becomes a bottleneck. they can play a role as buffers.
    - Some experiments are having problems now and solutions are needed
4. Need local object store
    - Appliances which speak Amazon S3 protocol now readily available
    - Open Source emulations of same also now readily available (as add-ons and gateways to Ceph for instance)
    - Xrootd already has plugins to read block N of multi-GB root object, successfully tested in Amazon runs this year. Could be more cost-efficient to store and access large bulk data this way, less overhead than keeping them on a POSIX based system.

Glossary

- **HSM** (as 'something' that automatically moves data between high-cost and low-cost storage media) is 'just another' cache layer. It may be more precise to talk about automated tiered storage (https://en.wikipedia.org/wiki/Automated_tiered_storage) which is an automated and fine-grained HSM implementation.