



Adapting and Scaling Storage for NHEP

Tejas Rao <raot@bnl.gov>

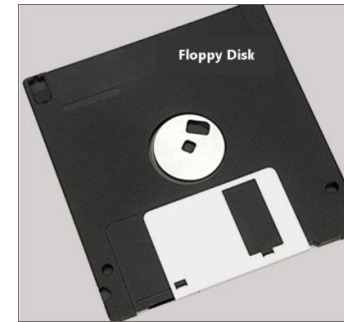
Scientific Data and Computing Center (SDCC)



@BrookhavenLab

Evolution of storage

- Over the past 70 years, data storage has evolved.
 - 1956 - IBM ships first hard drive - 5MB in size
 - 1972 - First Floppy disk - 4MB
 - 1980 - World's first 1GB+ hard disk
 - 1985 - World's first CD ROM - 900MB.
 - 1996 - First DVD ROMs - 4.7GB
 - 2001 - First USB flash drive - 8MB
 - 2007 - Hitachi unveils World's first 1TB hard drive
 - 2016 - Samsung ships World's largest SSD drive - 15TB



**IBM 5 MB hard drive.
Size of a refrigerator.**

Data is growing and so is the Data storage technology

Disk Storage at SDCC

- Total of 120 PB of raw storage and > 2 billion files
- Storage is a mix of Lustre , Spectrum scale (GPFS) , dCache and XrootD.
- Total performance requirements are 300GB/s + in streaming sequential IO.
- More than 7000 disks and hundreds of storage servers.



IBM **Spectrum Scale**

MINIO

l.u.s.t.r.e.[®]

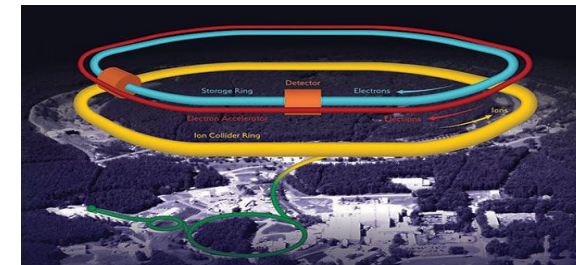
Tape Storage (HPSS)

- 217 PiB accumulated data 30 PB purged in 2021
- 9 Oracle Silos , 5 IBM TS4500 currently in production with 34 library frames.
- Total 320 LTO tape drives and 120K tapes.
- **Expected to have over one exabyte of data on tape by 2025.**



Initial storage projections for an EIC detector (FY 34 - FY 36)

ECCE Runs	year-1	year-2	year-3
Luminosity	$10^{33} \text{cm}^{-2} \text{s}^{-1}$	$2 \times 10^{33} \text{cm}^{-2} \text{s}^{-1}$	$10^{34} \text{cm}^{-2} \text{s}^{-1}$
Weeks of Running	10	20	30
Operational efficiency	40%	50%	60%
Disk (temporary)	1.2PB	3.0PB	18.1PB
Disk (permanent)	0.4PB	2.4PB	20.6PB
Data Rate to Storage	6.7Gbps	16.7Gbps	100Gbps
Raw Data Storage (no duplicates)	4PB	20PB	181PB
Recon process time/core	5.4s/ev	5.4s/ev	5.4s/ev
Streaming-unpacked event size	33kB	33kB	33kB
Number of events produced	121 billion	605 billion	5,443 billion
Recon Storage	0.4PB	2PB	18PB
CPU-core hours (recon+calib)	191Mcore-hrs	953Mcore-hrs	8,573Mcore-hrs
2020-cores needed to process in 30 weeks	38k	189k	1,701k



Storage requirements for SPHENIX

	FY20	FY21	FY22	FY23	FY24	FY25
MC DISK deployed (FY25 is driven purely by HW retirement), PB	5.5	10.9	16.4	21.8	27.3	21.8
RAW DISK deployed (FY24-25 capacity is kept leveled with that of FY23; no HW retirement in FY20-25 period), PB	0.0	0.0	10.9	21.8	21.8	21.8
DST DISK deployed (no HW retirement in FY20-25), PB	0.0	0.0	10.9	21.8	32.7	43.6
Total sPHENIX DISK deployed in SDCC, PB	5.5	10.9	38.2	65.4	81.8	87.2
Combined number JBOD + head node pairs	5	10	35	60	75	80
Combined DISK max. unidirectional bandwidth	20 GB/s	40 GB/s	140 GB/s	240 GB/s	300 GB/s	320 GB/s

Currently SPHENIX Lustre filesystem which is **35PB** in usable capacity, is capable of pushing **210GB/s** in pure sequential reads/writes

Challenges with Enterprise storage

- Massively growing data.
- Aging storage systems and hardware refresh.
- Data security concerns.
- Continuous uptime expectations.
- Excellent performance for all workloads.
- Data protection and Data recovery.
- Lowest possible cost.



Challenges with Distributed POSIX filesystems

- POSIX IO is stateful
 - Reading and writing data is governed by persistent state that is maintained by the OS in the form of file descriptors.
 - OS keeping track of state of every file descriptor becomes a major scalability bottleneck.
 - Cost of opening a file on most parallel filesystems' scales linearly with the number of clients making the request.
- POSIX IO has strong consistency
 - Write() has to block application execution until the system can guarantee that any other read() call will see the data just written.

POSIX

Challenges with Distributed POSIX filesystems (continued..)

- The pains of ensuring POSIX consistency.
 - To adhere to POSIX's strong consistency, parallel filesystems must ensure page-cache (dirty pages) are flushed completed to the backend before other nodes as able to read the data
 - Most parallel filesystems get around this issue by implementing complex locking mechanism.
 - Locking mechanisms can suffer from scalability limitations where multiple nodes are fighting over locks for overlapping extents.
- We see frequent client evictions or node expels due to the POSIX strong consistency requirement.

Cost

- Hardware RAID
 - Easy to maintain.
 - Easy to share block devices over multiple servers
 - Cost is usually 2X compared to software RAID solutions.
 - Random burst performance is excellent.
- Software RAID
 - More complex to maintain long term.
 - Sharing of block devices can be done with corosync/pacemaker.
 - Cost is 1/2 compared to Hardware RAID.
 - Streaming sequential performance is very good.



Currently using both at BNL for different use cases.

Costs (continued)

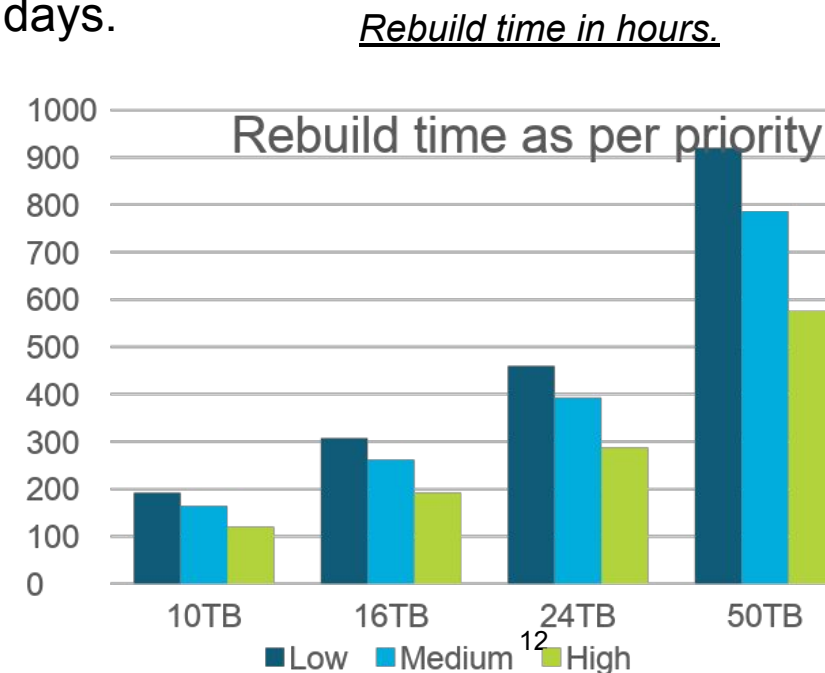
- Currently deployed more than 100PB raw disk storage using 102bay/60bay JBODs.
- JBODs/software RAID solution need linux/storage expertise.
- Developed in-house expertise and tools to manage software RAID solutions like Linux mdraid and ZFS.
- Dense storage systems reduces cost. Currently packing 102 drives in 4U space.
- Dual CPU socket servers are capable of driving 25GB/s or upto 300 drives.

Challenges with magnetic disks and RAID

- Drive capacities are increasing every year. 22TB NAS PMR drives being shipped currently.
- Performance continues to decrease - 13MB/s/TB (22TB drives) vs 21MB/s/TB (12TB drives). Multiple actuators in the future will help increase IOPS.
- 50TB drives predicted in 3 years and 100TB drives in year 2030 .
- MTBF are usually 2.5M hours for newer drives.
- If your storage system has 7000 drives then a failure occurs every 15 days.
- It would take 37 days to build a 50TB at low priority.

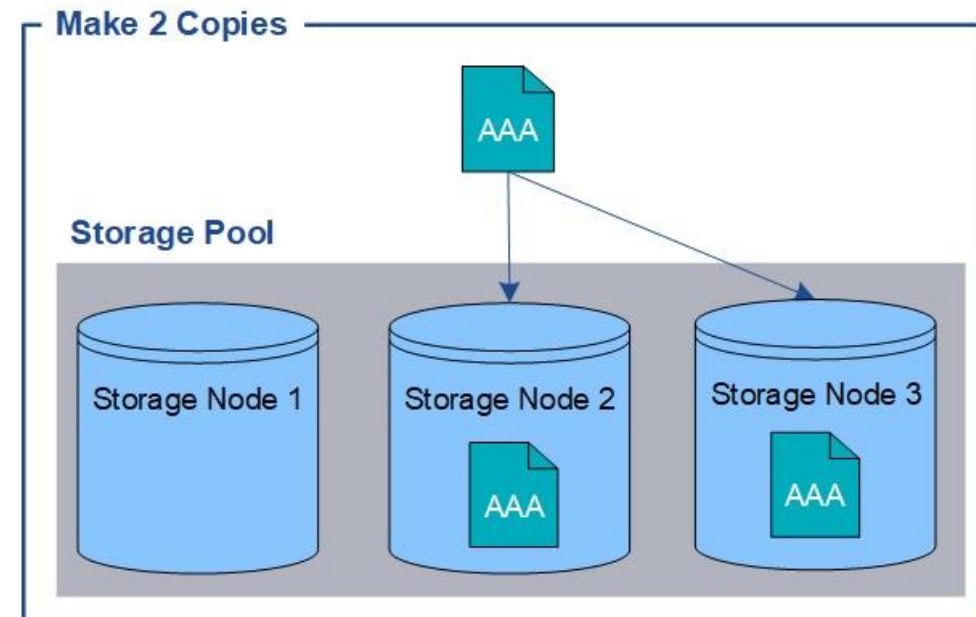
RAID can't scale. Risk exposure.

Large scale organizations need other data protection mechanisms



1st Solution – Data replication – Local or Geo mode.

- Multi-copies with special consistency approaches.
- No more consistency with the write() operation, now controlled with the read() operation.
 - *CAP Theorem: Strong Consistency when $R + W > N$ (R : Number of Replicas in a Read, W : Number of Replicas that must send an ACK for Writes and N : Number of nodes for the operation) or Eventually Consistent when $R + W \leq N$*
- No data modification, so data is easy to access.
- Ideal model but becomes expensive at very large scale.

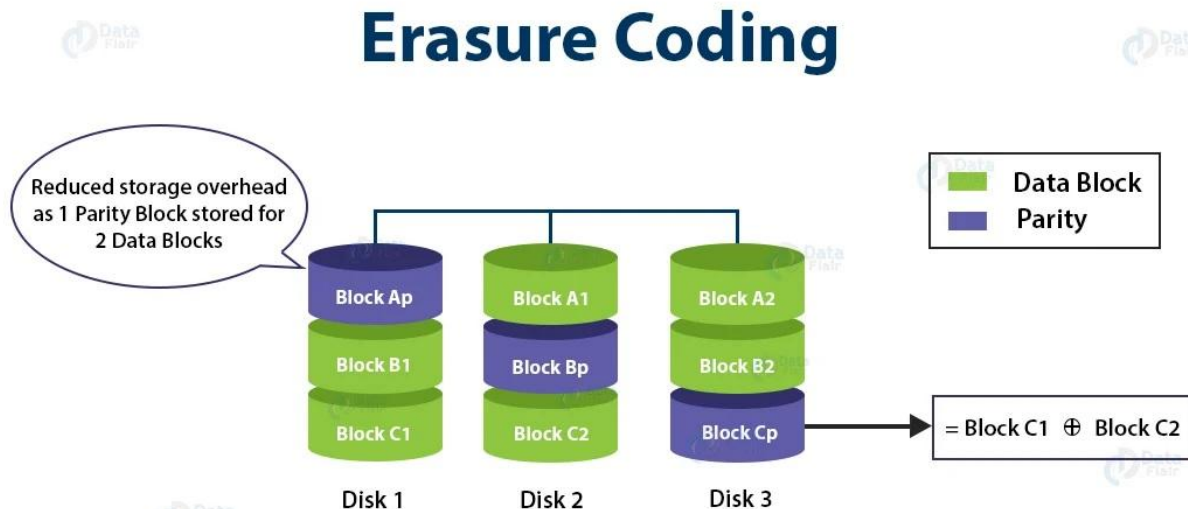


2nd Solution – Erasure coding.

- Goal: tolerate high number of concurrent failures and continue to deliver data on requests

Examples: 4 failures among 10 data elements, 6 on 16...

- Various notation: Data (n) and Parity/Checksum (k) chunks noted (n,k)
- Ratio $(n+k)/k$ is key: common from 1.2 to 1.6 (capability to tolerate loss) and storage efficiency $k/(n+k)$
- Local within a Cluster/Ring or Geo (Dispersed), Local coupled with Replication for Geo
- Cost effective solution to control the cost associated with the data volume growth
- Erasure coding is generally meant for cold storage and delivers lower performance.



Possible solutions

Object Storage

- Object storage is a flat (as opposed to hierarchical) file storage system and is designed for massive scalability.
- Rather than organizing files into folders, files can be tagged with metadata, thereby allowing files to be self-described and easy to find.
- Object storage – Many opensource and commercial solutions available.
 - Ceph , MinIO , Scality , OpenIO, Hitachi (HCP).






Possible solutions (Continued)

Object Storage

Advantages —

- Massive scalability
 - Filesystems/filestorage work best for less than 1 billion files.
 - Data management becomes a problem with > 500 million files.
- Reduced Cost (Compared to Netapp/Hitachi hardware RAID solutions)
- Searchability
- Sharing of data, High security - Federated Storage
- Loose coupling of clients.

STORAGE TYPES

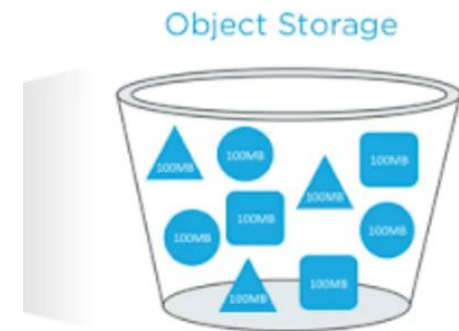
	BLOCK STORAGE	FILE STORAGE	OBJECT STORAGE
			
TRANSPORT:	FC or iSCSI	TCP/IP	TCP/IP
INTERFACE:	Direct Attached or SAN	NFS, SMB	HTTP, REST
USE CASE:	Low Latency Best for Structured Data	Good Performance File Sharing, Global File Locking	Easy Scaling with No Limits Accessible across LAN & WAN

Possible solutions (Continued)

Object Storage

Disadvantages –

- IO interface is the the primary drawback.
 - Mounting of object storage is only possible through fuse interfaces (low performance and compatibility)
- Modifying data is tedious, suitable for static data.
- Performance in terms of IO throughput is lower compared to traditional storage systems.



- WORM storage
- Massive scale
- Single namespace

Federated access to Storage

- Use case - Access to storage to users using their own institutional accounts.
- Sharing POSIX filesystem data over WAN/Internet.
 - Possible by using Kerberos (krb5/krb5i/krb5p).
 - File locking/cache coherency makes this complex.

Object Storage

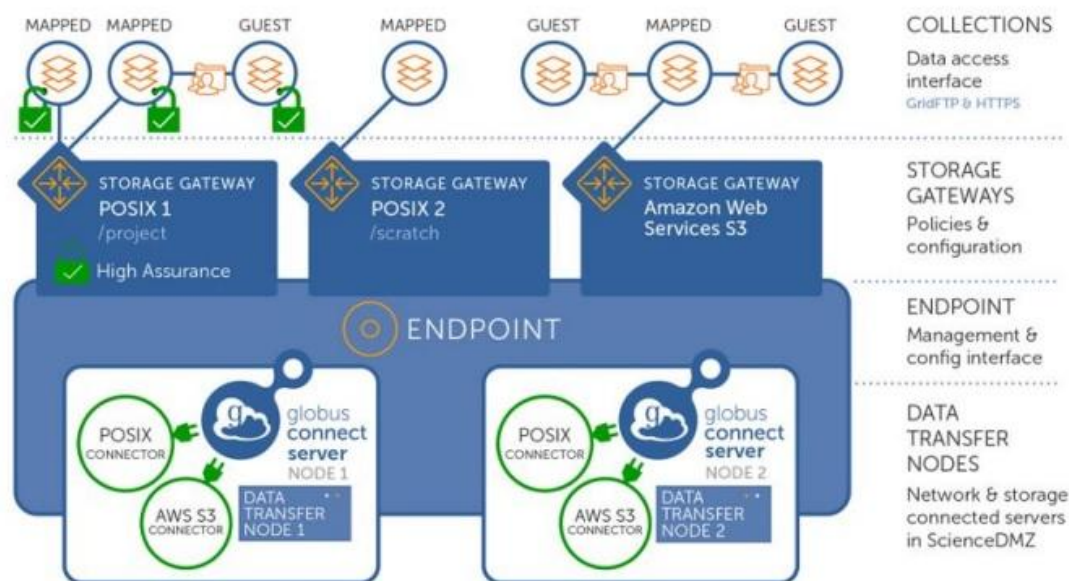
- Users can use their institutional credentials to get access to federated storage.
- Possible to integrate to CILogon for federated access.
- Investigating this in context of EIC.
- Identity Federation
 - Client grants - Let applications request client_grants using any well-known third party identity provider such as KeyCloak, WSO2
 - Returns a set of temporary security credentials for applications/clients

Conclusion

- Very hard to predict storage trends and solutions 10 years from now.
- Storage requirements are expected to increase exponentially.
- Object storage solutions seem promising specially for 10's of billions of files.
- Current Storage solutions are working well but there is a need to adapt and keep up with newer technologies.
- Managing large active storage systems requires expertise developed over time.
- Minimizing costs is important when requirements are increasing exponentially.

Back up slide (Globus Data transfers)

- Lustre data filesystems and NFS home directories available worldwide via Globus endpoint “NSLS2”.
- Oauth based access management architecture.
- Multi-DTN endpoints for load balancing/scalability and redundancy.
- Configurable Identity mapping system.
 - Map Identities from different domains to local accounts.
- Sharing of files is through mapped collections or guest shared collections.
- Shared site-license with central BNL IT.



Backup Slides (Backups)

- NSLS2 backup policy is to backup all files modified/created in the last 1 year.
- ZFS is used for backups since it supports snapshots.
- MYSQL database ingesting all Lustre changelogs from MDS.
- Applying mass actions to filesystems entries quickly using various criteria and actions.
- Lustre changelogs used to keep robinhood database updated in near real time.

Scanning 500 million files takes few hours.

1 million + new files every day.

