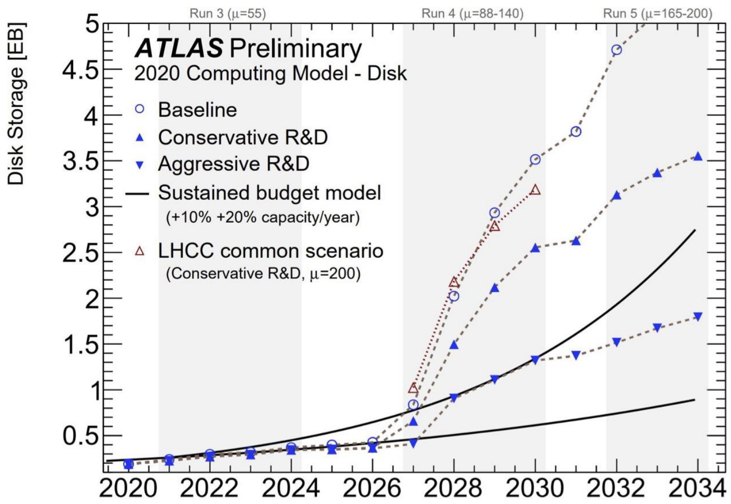
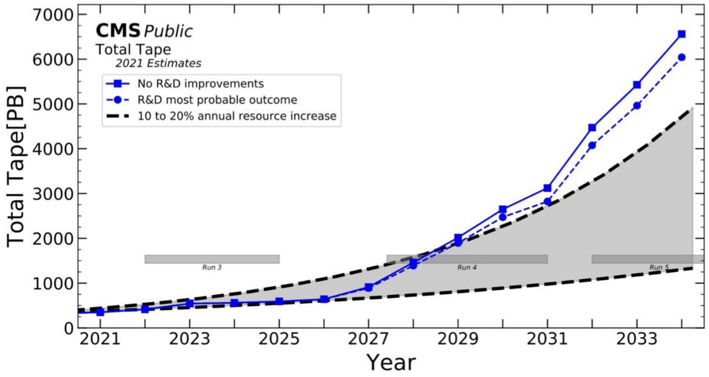
CompF4 Storage Summary

Bo Jayatilaka (FNAL), Carlos Maltzahn (UCSC), Peter van Gemmeren (ANL)

Storage needs

- Meeting the storage needs of future HEP experiments with increasing data rates
  - E.g. HL-LHC experiments at Exabyte scales
- Topic is not just about storage volumes
  - Storage must serve heterogeneous compute architectures and resources
    - Common interfaces/portable edge-services must work with storage HEP sites deploy





Community Input/Relevant Whitepapers

Submitted Snowmass white paper:

- "Data Storage for HEP Experiments in the Era of High-Performance Computing" [arXiv:2203.07885] A. Bashyal, P. Van Gemmeren, S. Sehrish, K. Knoepfel, S. Byna, Q. Kang

Other inputs:

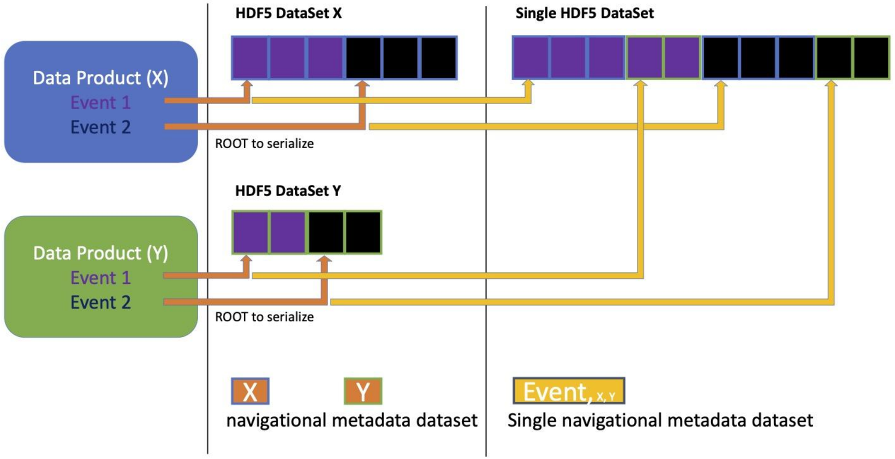
- HSF Community White Paper "A Roadmap for HEP Software and Computing R&D for the 2020s", The HEP Software Foundation, Albrecht, J. et al., Comput Softw Big Sci (2019) 3, 7
- January 2022 DOE ASCR Workshop on the Management and Storage of Scientific Data
- Contributions to the CompF4 topical group workshop [https://indico.fnal.gov/event/53251/]

Technology drivers: storage media

- **Magnetic tape, rotating hard drives, and solid state** will be the primary storage media for HEP in the coming decade
    - Other technologies on the horizon are not likely to factor in that timescale
- Tape will continue to form the archival backbone
    - Tape storage densities increase faster than access rates
    - Aggregate bandwidth may be more limiting with tape than capacity in HL-LHC era
- HDDs still provide bulk of enterprise storage and at small/medium HEP sites
    - PMR is reaching limits in storage density
- Solid state storage continues to be costly compared to HDD
    - HEP applications that require higher IOPS benefit from dedicated solid-state storage

Storage for HEP experiments at HPCs [arXiv:2203.07885]

- Future HEP experiments require large processing resources and will rely on HPCs to provide those
- Running HEP workflows on HPCs will benefit from using HPC-friendly storage software, such as HDF5 (in addition to ROOT)
- HEP-CCE/IOS group has studied using HDF5 for ROOT serialized data.
  - ROOT serialization allows to handle complex data models
  - Different mappings to HDF are being investigated



Compression, custom content, and streaming

- Most HEP data is stored with **lossless compression and highest precision**
  - Limited storage budgets may require revisiting this practice
    - E.g., only store RAW data with high precision; identify variables whose precision can be limited without reducing physics potential
- Most disk storage at HEP experiments is used for **derived data**
  - Often multiple copies and multiple formats with partial overlap
  - Development of standard derived formats that meet most physics needs reduces storage
  - Adding additional information for analysis often requires rederivation of entire datasets
- Ability to read **additional data on-demand** is more efficient
  - Would need standard access of **sub-event** level information and robust streaming

Computational Storage

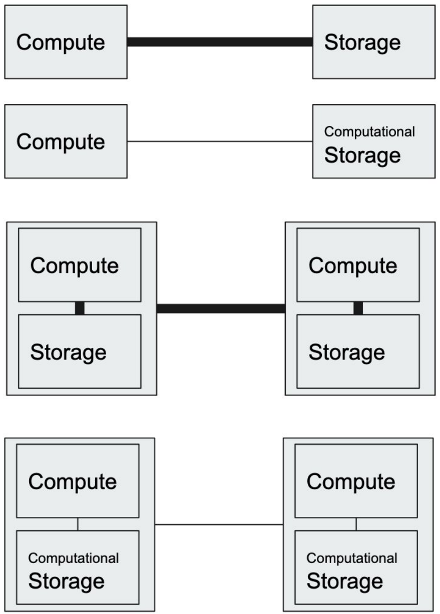
Trends

- Hardware accelerators are becoming available within NICs, storage devices, PCIe devices, storage arrays
- Driven by disaggregation and faster storage devices (NVMe Devices)
- Everyone wants common data management and access ecosystems (Apache Arrow, HDF5)
- Standardization efforts (NVMe 2.0, CXL) and Open source ecosystems reveal internal strategies of hyperscalers:
  - Leverage embedded processing and accelerators to reduce data movement while reducing latency
  - Converge memory, flash, and spinning media to one interface (NVMe, CXL)

Aligning *some of* data management and its context with data placement

Co-location of compute and storage is complex:

- **failures** change location of stored data;
- computation **unaligned** to physical data layout



Computational Storage: Technologies to watch

Access libraries mapping datasets to multiple data sources,

- Examples: Apache Arrow Dataset Interface, HDF5 Virtual Object Layer (VOL).

Intermediate representation standard for relational algebra (e.g., substrait.io).

Sandboxing technologies eBPF and WebAssembly with extensions that go beyond what is allowed within the Linux kernel

Distributed resource management for computational storage balancing locality and occupancy.

Identified some DOMA/Storage R&D topics for the 2020s

- Sub-file granularity storage access and management
- Data organization and analysis used by other big data users (then Spark, now “Lake-housing”)
- Data placement and caching, including for ML applications
- Minimize infrastructure cost by coordinating tiered storage
- Globally minimize data access latency

HSF Community White Paper: A Roadmap for HEP Software and Computing R&D for the 2020s, The HEP Software Foundation, Albrecht, J. et al., Comput Softw Big Sci (2019) 3, 7 https://doi.org/10.1007/s41781-018-0018-8.

Recommendations/Research Directions

High-level CompF4 recommendations all specifically apply to storage

- Efficiently exploit specialized compute architectures and systems
- Invest in portable and reproducible software and computing solutions to allow exploitation of diverse facilities
- Embrace disaggregation of systems and facilities
- Extend common interfaces to diverse facilities