ATLAS Data Carousel and WLCG Data Lake R&D Projects

Alexei Klimentov

BigPanDA Technical Interchange Meeting

BNL

April 25, 2019

Thanks

- Simone Campana, Kaushik De, Andrey Kiryanov, Torre Wenaus, Andrey Zarochentsev, Xin Zhao for slides and materials

Data Carousel and Data Lake Projects Motivations.

The HL-LHC will be a multi-Exabyte challenge where the anticipated storage and compute needs are a factor of ten above the projected technology evolution and flat funding.

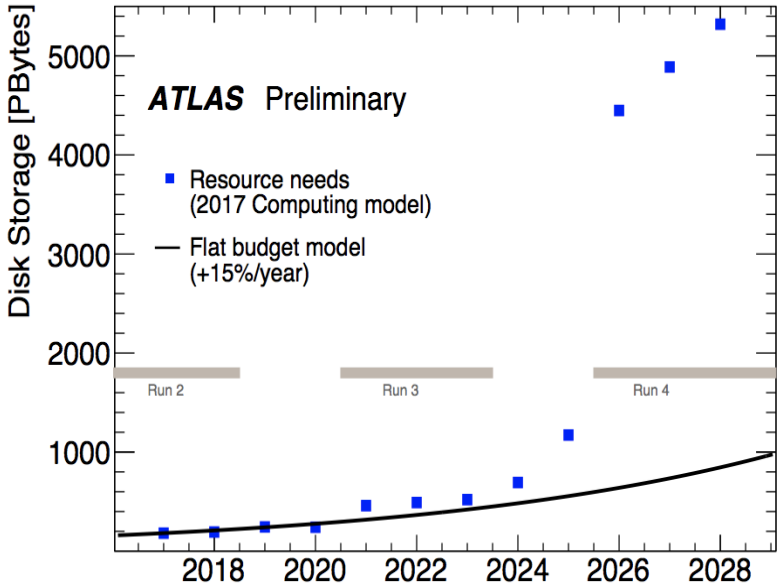
The WLCG community needs to evolve current computing model to manage data and storage efficiently.

Technologies that will address the HL-LHC computing challenges may be applicable for other communities to manage large-scale data volumes (SKA, DUNE, LSST, BELLE-II, JUNO, etc). Co-operation is in progress

ATLAS was very successful with opportunistic CPUs, but ‘Opportunistic storage’ basically doesn’t exist

Format size reduction and data compression are both long-term goals, require significant efforts from the software and distributed computing teams (some R&Ds are planned together with commercial partners)

Tape storage is 3~5 times cheaper than disk storage, increasing tape usage is a natural way to cut into the gap of storage shortage for HL-LHC



Data Carousel. Initial Project Motivation.

- Ultimate goal : use tape more efficient and active
  - Data carousel… sliding window…
    - Cycle through tape data, processing all queued jobs requiring currently staged data
    - ‘carousel engine’: job queue regulating tape staging for efficient data matching to jobs?
      - Brokerage must be globally aware of all jobs hitting tape to aggregate those using staged data
- ‘Data Carousel’ R&D started in 2018 → to study the feasibility to use tape as the input to various I/O intensive workflows.
  - …and “tape” could be any “cold” storage

Data Carousel. Primary Project Objectives.

- DDM system : Rucio → more intelligent tape I/O
    - Bulk data staging requests handling
- File Transfer Service → optimize scheduling of transfers between tape and other storage endpoints
- DDM / WFM integration. Optimize data placement to tape
    - Define tape families for files known to be re-read from tape (data grouping)
    - Optimize file size (Larger file size, 10GB+ preferred)

Data Caroseul. Project phases.

- First phase – Completed
  - Understand tape system performance at CERN and Tier1s
    - Collect metrics for each site and use them in DDM and WFM
  - Identify workflows and possible scenarios for data pre-staging

    - ADC Data Carousel (live google doc)
    - Tape Tests
    - Initial document to trace the progress and results of the tape performance testing. (X
    - Report to DOMA

- Second phase – In progress
  - Deeper integration between workload and data management systems
- Third phase
  - Run data carousel in production at scale, for selected workflows before LHC Run3 (2022)

Data Carousel Phase II

- Data Carousel Phase II - Facilities / DDM / WFM integration
  - Use metrics obtained at Phase I
    - Define sites storage characteristics : min/max I/O limit, average throughput. Use it in DDM and WFM for data staging
  - Setup and implement DDM / WMS communication protocol
  - Develop and implement algorithms
    - For intelligent data pre-staging
      - Respect priorities, shares, computing and storage availability
    - Tasks brokering
  - Provide monitoring to sites and users
  - Do intelligent data writing
  - Address data carousel in hot/cold storage context

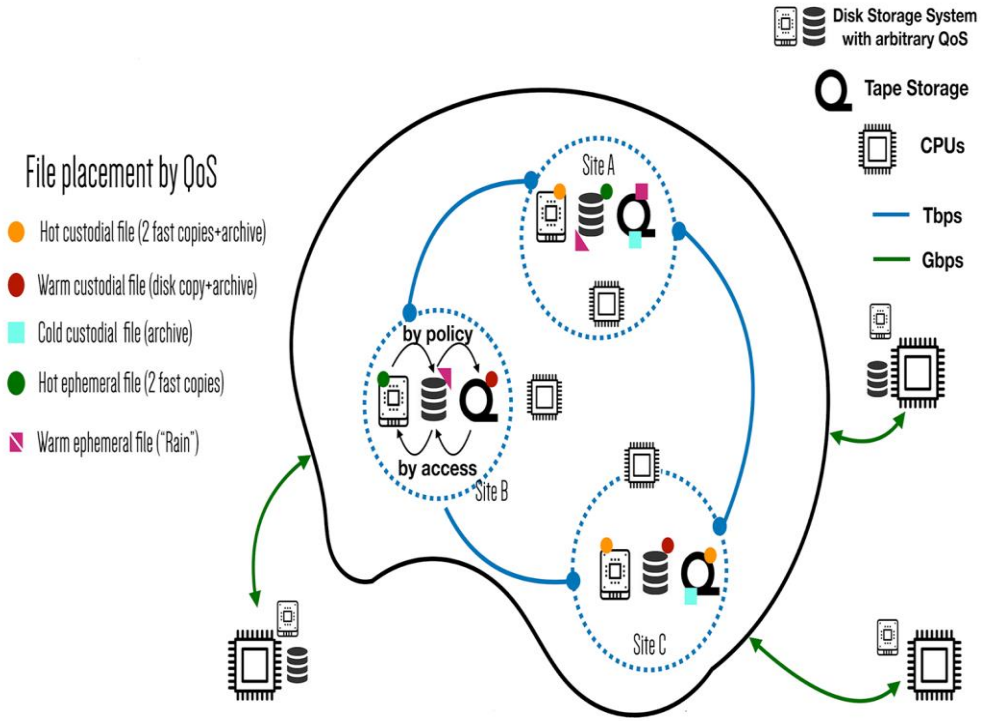
Ultimate goal to have it in production before Run3

Data Lake. Storage software emerged from HENP scientific community

- **DPM** – WLCG storage solution for small sites (T2s). Initially supported GridFTP+SRM, but now undergoing a reincarnation phase as **DOME** with HTTP/WebDAV/xrootd support as well. No tapes.
- **dCache** – a versatile storage system from DESY for both disks and tapes. Used by many T1s.
- **xrootd** – both a protocol and a storage system optimized for physics' data access. Can be vastly extended by plug-ins. Used as a basis for ATLAS FAX and CMS AAA federations.
- **EOS** – based on xrootd, adds smart namespace and lots of extra features like automatic redundancy and geo-awareness.
- **DynaFed** – designed as a dynamic federation layer on top of HTTP/WebDAV-based storages.
- **CASTOR** – CERN's tape storage solution, to be replaced by **CTA**.
- On top of that various data management solutions exist: **FTS**, **Rucio**, **FedEx,** etc.
- On top of that various WFM and WMS systems : **PanDA, Dirac, Pegasus**, etc

Data Lake Concept

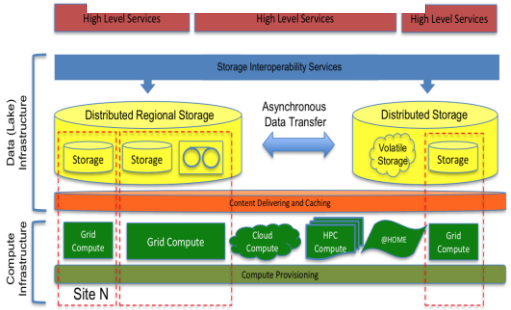
- Not another software or storage solution.
- It is a way of organizing a group of Data and Computing centers so that it can perform an effective data processing.
- A scientific community defines a “shape” of their Data Lake, which may be different for different communities.
- We see the Data Lake model as an evolution of the current infrastructure bringing reduction of the storage costs.



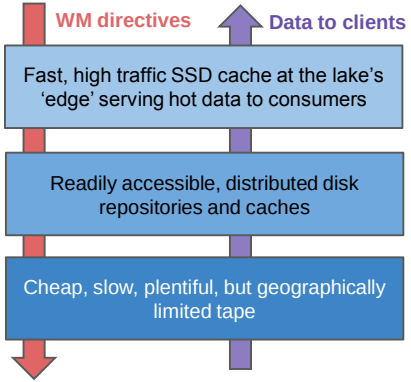
Data lakes and workload management

- Our sites are linked with (ever higher) high-bandwidth networking
  - We can expect **~100x bandwidth growth** by HL-LHC
- **Data lakes**: integrated consolidation of distributed storage (and compute) facilities, leveraging high-bandwidth networks
- Data lake encompasses facilities with several levels of storage
  - **Tape**, at a relatively limited number of sites
  - **Standard disk**, at large storage repositories and smaller caches
  - Fast SSD '**edge cache**' for the hottest data
  - Should be able to **place data optimally** based on (dynamic) need
- Workload management knows the hot popular data in use
  - Use that knowledge to drive preparing data in the lake, asynchronously to the processing, e.g.
    - tape staging in a **carousel workflow**
    - placing hot data in SSD cache **'close' to available CPU**
    - **transforming/marshaling data** optimally for client delivery
  - Requires APIs supporting WM directives
- **Instead of 1.8 replicas on disk today, WM + data lake manages dynamic availability of actively used data with replica count <<1**

Data lake schematic



Data lake interactions



Serving from the lake: streaming data flows

- **Move only the data you need**, to a client ready to consume it
  - **Hide the latencies** involved from the processing
- Can achieve this with **streaming data flows**
  - Don’t require big files to move from A to B before processing starts at B
  - Be **agile, asynchronous, adaptive** to current resource availability
- Data streaming to the client can
  - use **knowledge of the task to marshal** and send only needed data
  - **begin processing immediately** without a long staging wait
  - be **(re)directed** to workers at different or multiple processing resources to complete tasks ASAP, without long slow tails
- Enabled by **fine-grained processing** that ATLAS has been developing
  - Flexible partitioning of the work to enable **optimizing the granularity** from full files down to single events
  - Granular processing: **Event Service** in early production for simulation
  - Granular marshaled inputs: **Event Streaming Service** in early R&D
  - Both are clients for the **Event Whiteboard** to manage associated metadata, in early R&D