# How HL-LHC challenges inform workload management R&D: An ATLAS view

Torre Wenaus (BNL), Davide Costanzo (Sheffield), Alessandro Di Girolamo (CERN), Johannes Elmsheuser (BNL)

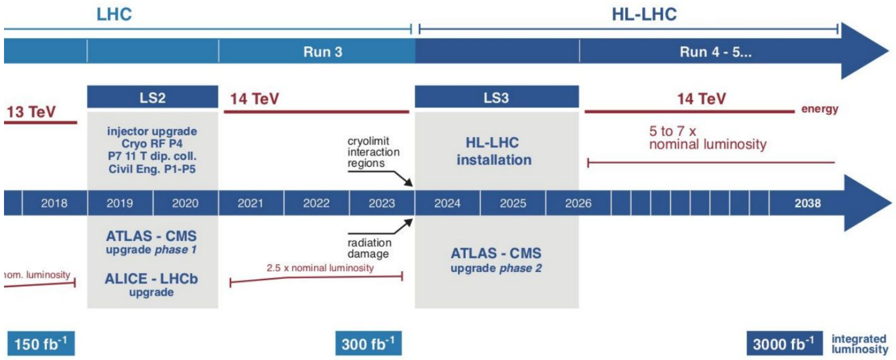
Joint WLCG & HSF Workshop 2018
Naples
March 27 2018

# ATLAS computing challenge at HL-LHC

- HL-LHC computing far outstrips the budget
- ATLAS physics already is **compute-limited**
  - Drives our computing approach today
- Motivated **tools suited to evolving** to HL-LHC
  - PanDA workload management (WM)
  - Rucio data management
  - Centralized ever-growing info sources
  - All working in a tightly integrated way
- On this foundation, developing new approaches that
  - **leverage powerful networks**
    - The most crucial factor to LHC computing success
  - **minimize storage** needs
  - make the most of **opportunistic resources**

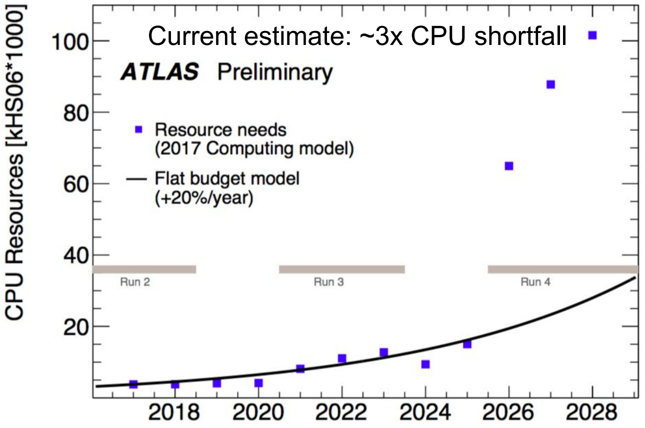
**Working assumptions:**

- Flat facilities budget
- 20%/yr capacity growth, constant dollars
- Development effort is sustained
  - The need for this has been affirmed by ATLAS, LHC management, LHCC, resource board scrutiny, ...

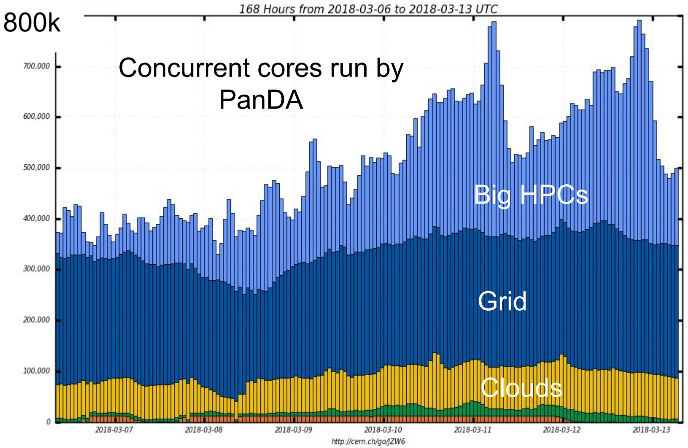


# Scaling to HL-LHC: CPU

- In 2016, HL-LHC CPU estimations showed a ~6x shortfall, dominated by track reconstruction combinatorics, mu=200
- Since then, new optimized tracking code has sped it up ~4-5x
- Thanks to this, today's estimations show only a **~3x shortfall**
- With **important optimizations still to come**, e.g.
  - Optimizing reconstruction for the other detectors
  - Use of truth info in MC track reconstruction
- This is still **without leveraging co-processors**/GPUs
- ATLAS is ever more broadly harvesting diverse sources of **opportunistic computing**
  - Good HPC citizens: fully parallel MPI applications, keeping every core busy for the full job duration
  - Adding a common provisioning layer to uniformly integrate resources (Harvester - Alexei's talk)
- Bottom line: **CPU shortfall is serious but we appear to be on track, not the biggest problem**



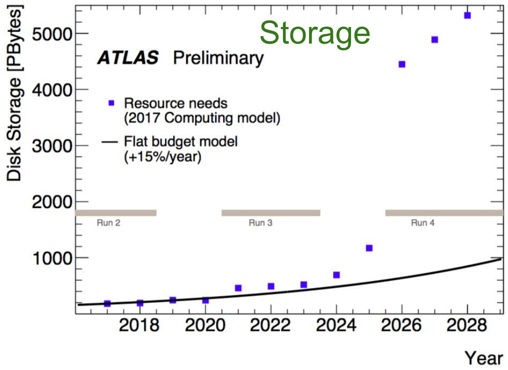

**WM scales smoothly today to 800k+ cores**



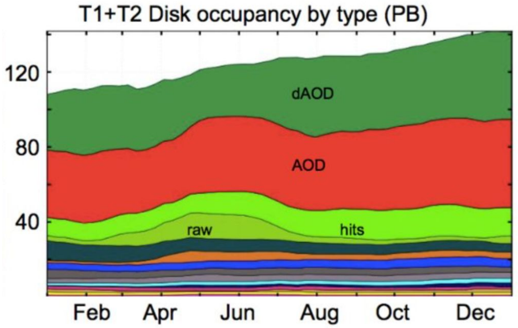
# Scaling to HL-LHC: Storage

- ~6x shortfall by today's estimate, a level that has held ~steady
- 'Opportunistic storage' basically doesn't exist
- Working on format size reductions, but hard to achieve large gains
- Replica counts already squeezed, hard to achieve large gains
- **Storage shortfall is our biggest problem. We need new approaches!**
- ATLAS disk usage is currently ⅔ analysis formats and ⅓ everything else
- Within the '⅓ everything else' are samples that reside mostly on tape, staged onto disk cache when needed for processing
- A way to **dramatically reduce our storage footprint** is to **grow the use of tape** (it looks like our cheap storage will remain tape)
  - Use a '**tape carousel**' approach for the analysis formats
  - A moving window of say ~10% staged to disk at any one time
- **This is hard:** tape is slow and complicates workflow orchestration
  - Analysis workflows are time critical and already complex
- Tape is geographically limited, while processing happens everywhere
- **Fertile ground for R&D...**

Disk storage ~6x short at HL-LHC



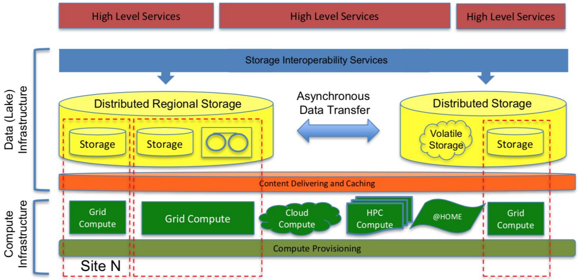

ATLAS disk usage 2017



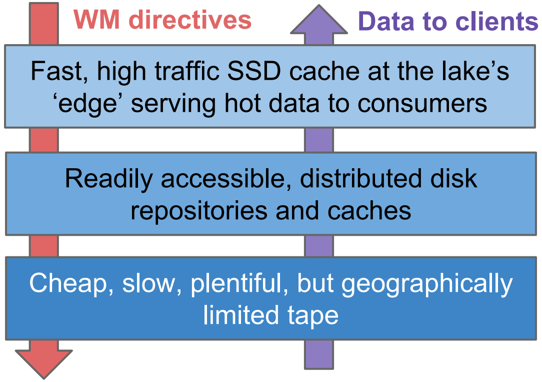
# Data lakes and workload management

- 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**

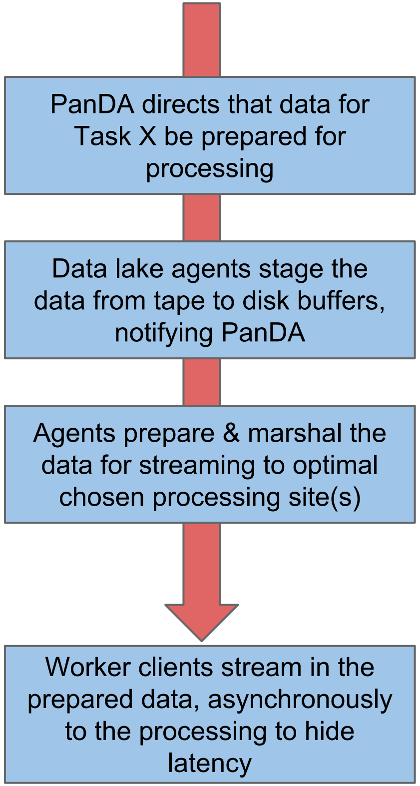



**Data lake interactions**



# 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



# Implementing streaming: granular event processing

- Event service + event streaming service = granular processing several benefits
  - Fine grained work assignments: **utilize CPU fully by ‘filling the cracks’**
    - Keep all multiprocessing cores busy all the time
    - Keep a ramping-down grid site busy until it’s offline,
  - Fine grained inputs: **stream remote inputs asynchronously**
    - Without the up-front latency and complexity of pre-staging big files
    - Without WAN latency impacting the processing
  - Fine grained outputs: **stream outputs asynchronously in quasi real time**
    - Avoid losing good data on a resource that vanishes (spot market cloud, preemptible grid queue, BOINC)
    - Hide WAN latencies, consolidate distributed outputs to one destination
- **Streaming = copying small files**, not a remote open and read across WAN
  - Files are robust, cacheable, easier to diagnose when problems appear, allows pre-marshaling of input data at the source

# The pieces, the path, the R&D prospects

- **Event service:** experience has proven the concept, flexible granularity is worthwhile, but more work is needed to complete a clean implementation
    - Goal: one ATLAS workflow with tunable granularity to optimize
    - Currently implemented over a largely coarse-grained foundation
    - Plan to re-engineer to fine-grained all the way down
        - While preserving conventional mode
    - Also enabling fine-grained mode **for use beyond ATLAS??**
- **Event streaming service:** intimately coupled to data lake (and I/O) R&D
    - Very early stage in ATLAS: **good moment to team up**
    - Ideas outlined here nicely expanded in Data Lakes white paper from US ATLAS (R. Gardner et al)
- **Data management via Rucio:** close ally in all this work. Intimate coupling between workload management and data management
    - **Much commonality and collaboration in R&D**

# The pieces, the path, the R&D prospects

- **Metadata management**
  - ATLAS Data Characterization and Curation (DCC) group consolidates event metadata activities, with a substantial R&D component
    - Bringing together established systems (e.g. AMI) and new developments (e.g. Data Knowledge Base R&D)
  - New R&D project important to granular processing is the **Event Whiteboard**
    - Database of event (more accurately, object) records with user defined metadata, and collections of them
    - Pursue as basis for metadata needs of event service, ESS, …
    - **Ripe for commonality** (avoid repeat of 7 different file catalogs 15 years ago :-)

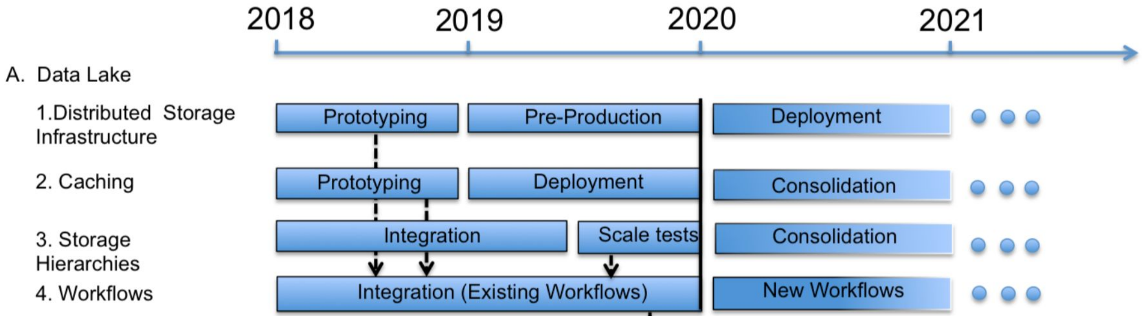
# More R&D prospects

- Understand **optimal granularity** for data management and access for all storage tiers and workloads. What are the **elementary data objects**, how are they aggregated, moved between tiers, served to the client.
- Related to ESS: storage-side **intelligent data marshalling**, informed by the scientific content of the data and the client’s actual needs. What **transformations are applied** to data, how, and where. How do they relate to compression, and effective use of cache hierarchies. Virtual data support.
- In latency hiding, data marshaling & packaging, compression management, caching, … **how much do we do via ROOT**
- Tape based **data carousels** with strong couplings between workload management, DDM, storage for intelligent automation. How fast can they go.
- **Data lake API and service design** supporting workload management automation and thoroughly isolating clients from special knowledge of storage organization. A smooth, intelligent global CDN (or Data Delivery Network)

# More R&D prospects

- Breaking co-location boundaries. How much will latency hiding & caching allow us to **relax co-location**? Commonality in more relaxed brokerage?
    - Continued value in preferentially processing close to data
- Workload management for **cloud based analysis**
- Mine deep data on system operation for **intelligent automation via ML**
- **Good metrics** to measure performance, efficiency and drive improvements
- Scalable, uniform **resource provisioning** among dynamic diverse resources

For data lakes (among other things), Ian Bird and Simone Campana have started to lay out an R&D timeline towards writing an informed HL-LHC computing TDR in 2020



# Near term benefits from long-view R&D

- Fine grained processing
  - Full and efficient utilization of **opportunistic resources**
  - **Relaxing co-location**, simplifying data placement and brokerage
- Tape carousels
  - **Lessen pressure on storage** resources
- Data lake
  - Use prototypes to provide relatively low-volume but high-value functions like hosting **distributed analysis outputs** in a robust, location-agnostic way
- Metadata
  - More powerful **tools for analysts** to follow and annotate the course of analysis work
- Next-gen resource provisioning
  - Better control and knowledge flow between WM and resource to better **match resources to tasks** and their requirements

# Conclusion

- Commissioning prior to Run 4 in ~2025 is only **7 years away**!
- ATLAS's **compute-limited science** has driven us to (start to) develop early some of the capabilities we'll need for resource-constrained HL-LHC
- **CPU is probably on track** as long as we keep smart and keep working
- **Storage is the greater danger**
- **The Network** is our enabling foundation and will remain so
- Addressing the storage problem in the context of global computing on a high powered network foundation is in part **a big workload management challenge**
- ATLAS's work on **fine grained processing**, motivated in the near term by effective use of opportunistic resources, will also pay off in efficient use of storage in the data lake context
- **R&D prospects are rich and varied** with near and long term benefits

# Thank you

- Much thanks to a great many in ATLAS Distributed Computing (ADC) for inputs and discussions, and to Simone Campana, Ian Bird
- Draws on parts of the “WLCG Strategy towards HL-LHC” currently in preparation by Simone et al
- For interesting reading very much aligned with ideas presented here see the “Organizing, Orchestrating, and Delivering Data From Lakes” white paper by Rob Gardner et al