ATLAS Software and Computing for Run 3

Frank Berghaus and Alaettin Serhan Mete
Argonne National Laboratory

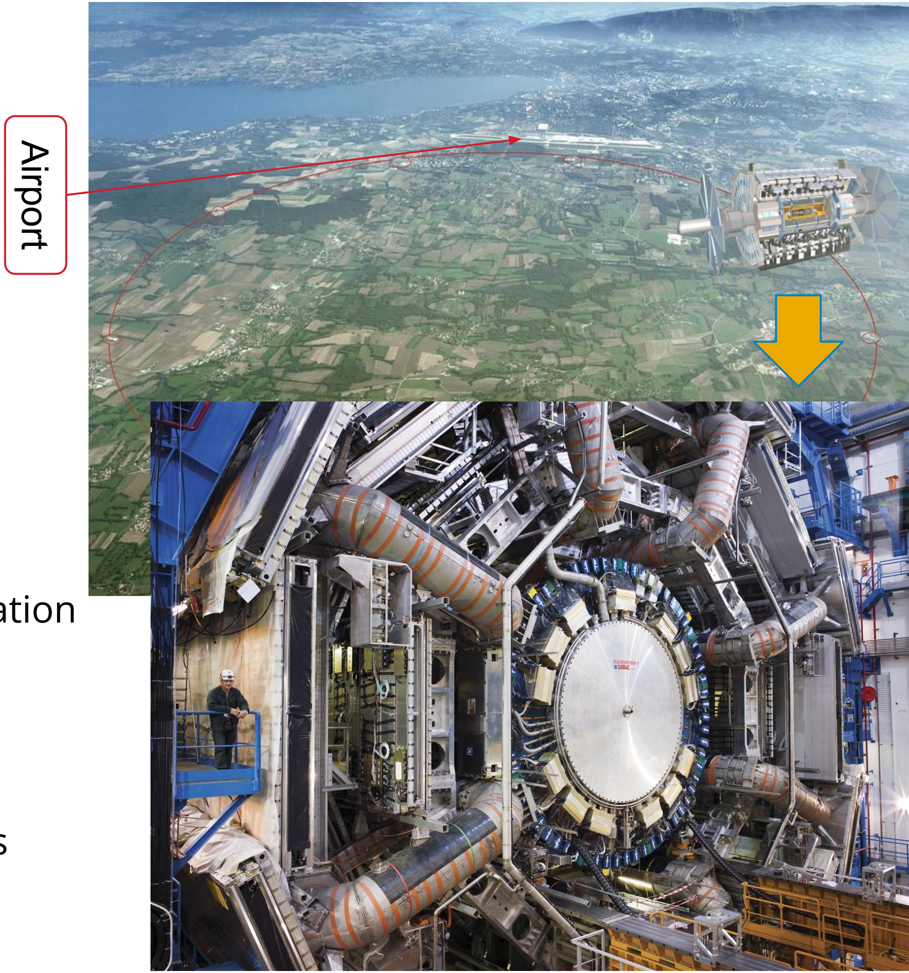
ANL HEP Divisional Seminar
29 September 2021

Outline

- **Introduction**
  - Reminder of the ATLAS Experiment/Detector
  - Overview of the ATLAS Software and Computing (**S&C**) and Argonne's Involvement
- **ATLAS Computing**
  - Introduction to the **W**orldwide **L**HC **C**omputing **G**rid (**WLCG**)
  - Discussion of the Main Aspects of the ATLAS Computing
- **ATLAS Software**
  - Introduction to ATLAS workflows and resource usage statics from Run-2
  - Introduction to ATLAS' main software framework, **Athena**
  - Discussion of the multi-threaded Athena (**AthenaMT**) migration
  - Discussion of preliminary results from the data reconstruction with AthenaMT
- **Argonne's Contributions/Leadership in ATLAS S&C**
  - Discussion of Argonne's significant role in the success of ATLAS S&C
- **Outlook & Conclusions**

The ATLAS Experiment

- Measures LHC collisions
  - Proton-proton at $\sqrt{s}$ = *13 TeV*
  - Heavy ions at $\sqrt{s}$ = *5.12 TeV/u*
- Multipurpose Particle Detector
  - Inner detector: vertex finding & tracking
  - Calorimeters: energy (EM and hadronic)
  - Muon Spectrometers: tracking & identification
  - Magnets: central dipole & air core toriod
- Global collaboration
  - ~5,700 physicists, engineers, and students



ATLAS Collaboration, ATLAS-CONF-2019-021

$\mathcal{L} = 139\ fb^{-1} \Leftrightarrow 439\ PB$

Barisits *et al. Comput Softw Big Sci* **3,** 11 (2019)

The LHC Timeline and the ATLAS S&C

- **The LHC went into its second long shutdown (LS2) in 2019**
  - Originally planned to last for **2 years** but extended for several months
  - **The physics program will resume in 2022** (accelerator commissioning underway)
- **LS2 provided a great opportunity for various improvements**
  - Installing new detectors (e.g. **N**ew **S**mall **W**heel)
- **ATLAS S&C undertook a set of major development work**
  - **Main goal:** Upgrading the ATLAS software infrastructure, ***Athena***, to be **multithreaded**
  - Performing software optimizations to have more efficient simulation, reconstruction, etc.

ATLAS Software & Computing

- Two environments:
  - *Online:* near the detector. Fast & reliable.
  - *Offline:* away from detector. Precise & reproducible.
- ATLAS software products
  - The main software, *Athena*, is used online and offline
  - Other software:
    - Trigger and data acquisition tools (online)
    - Analyses and their frameworks
    - Services & tools interfacing with computing resources
- Computing can also be divided into general categories
  - ~80% of ATLAS computing operate offline
    - Provided by the *WLCG* (Worldwide LHC Computing Grid)
  - ~20% of ATLAS computing operate online

ATLAS Computing

The WLCG

- Hardware and services for LHC Computing
  - Widely adopted in HEP (e.g. Belle-II, DUNE)
  - Prospective users from other communities (e.g. SKA)

- Components of the WLCG
  - *Sites* are data centers at universities and laboratories
    - Provide storage and computing resources
  - Research networks (e.g. ESnet, GÉANT)
    - Connect the sites
  - Software operating computing resources
    - Authentication & authorization
    - Software distribution
    - Scheduling and distributed storage systems

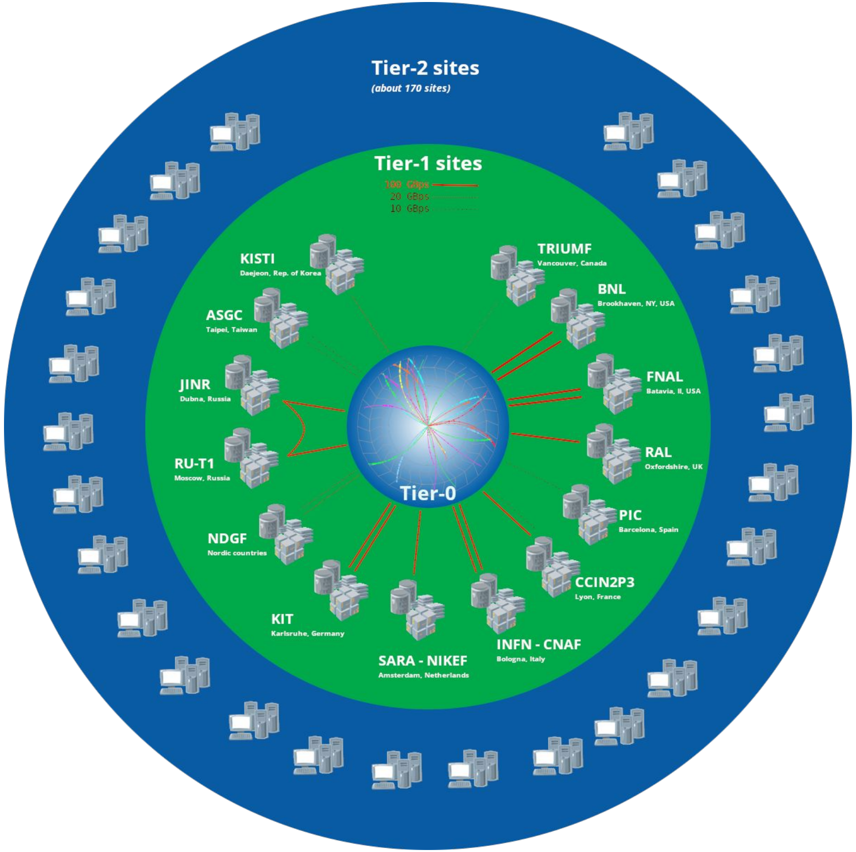
WLCG for the experiments (e.g. ATLAS)

- Workload management
  - Accepts high level request to execute a certain task (e.g. JEDI in ATLAS)
    - Example: Run this sequence of code on this data, generate these events
  - Plans out cascades of jobs to meet request with intermediate data (PanDA in ATLAS)
    - *Sends jobs to data*

- Data management systems: Rucio
  - Data access: Provide global file index with file information (location, checksum, etc)
  - Data management: Enforce policies on data replication and deletion
  - *Note:* Widely adopted, for example by CMS as well as ATLAS

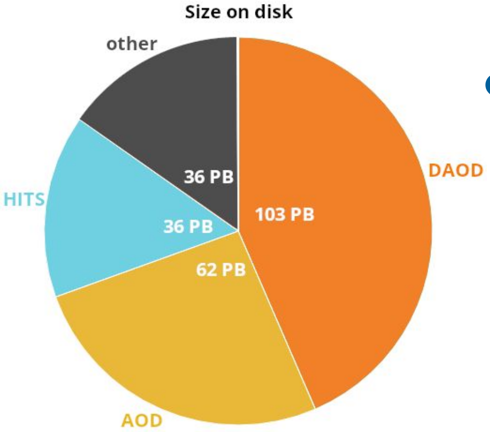
- Content delivery system: CVMFS/Frontier
  - CVMFS distributes software and small (<*2 G*) files to the sites
  - Frontier provides remote access to the central databases (e.g. detector conditions)

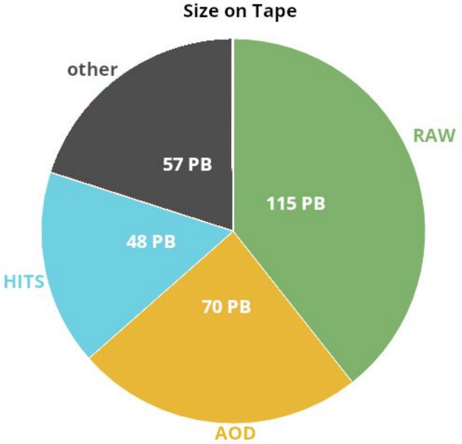
Taxonomy of WLCG sites



- Three categories of site:
  - Tier0 (1)
    - Site of the experiment
    - Hosts full copy of RAW data
    - All workflows
  - Tier1 (14)
    - Hosts fraction of RAW data
    - All workflows
  - Tier2 (~170)
    - Simulation and analysis

Anatomy of a WLCG site

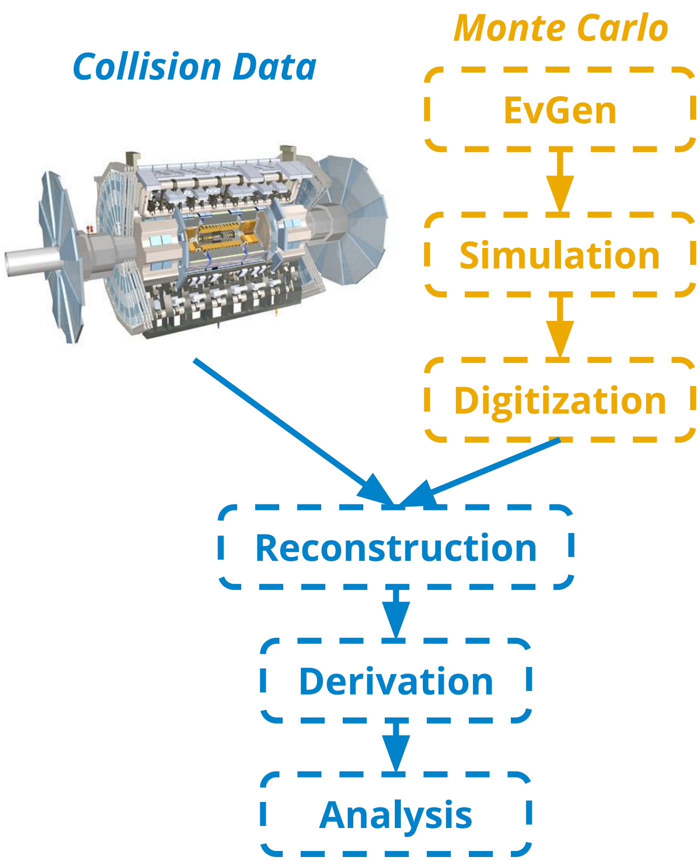




- Storage
  - Disk: Frequently used data. E.g.
    - (D)AODs analysis inputs
  - Tape: Rarely used data. E.g.
    - RAW data, inputs to published analysis
  - Provide an API for remote file operations
- Processing
  - Batch systems exposing a uniform API for remote scheduling
    - Commodity hardware: 20 GB disk & ***2 GB Memory per core***
    - Few systems offer more memory or disk
  - Argonne initiatives:
    - High Performance Computers [Doug B and Rui W]
    - High Throughput applications on clouds [Kate K at MCS]

ATLAS Software

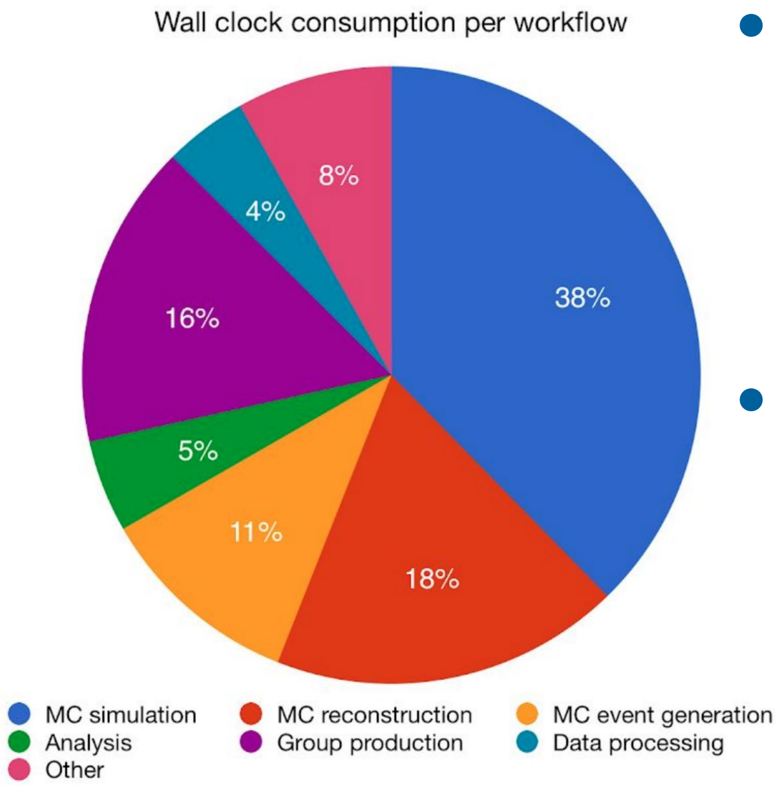
ATLAS Workflows



- **ATLAS data processing chain:**
  - **Event Generation :** Generating Monte Carlo (MC) events
    - _ → EVNT
  - **Simulation :** Simulate interaction w/ detector (MC-only)
    - EVNT→ HITS
  - **Digitization :** Simulate detector output & pile-up (MC-only)
    - HITS→ RawDataObject
  - **Reconstruction :** Reconstruct physics objects
    - RAW → AnalysisObjectData
  - **Derivation :** Refine physics objects
    - AOD → DerivedAOD
  - **Analysis :** Perform final physics analysis
- **ATLAS computing GRID is used for all**
  - Analysis @ local resources (institute, laptop etc.)

■ Data formats

ATLAS Resource Usage



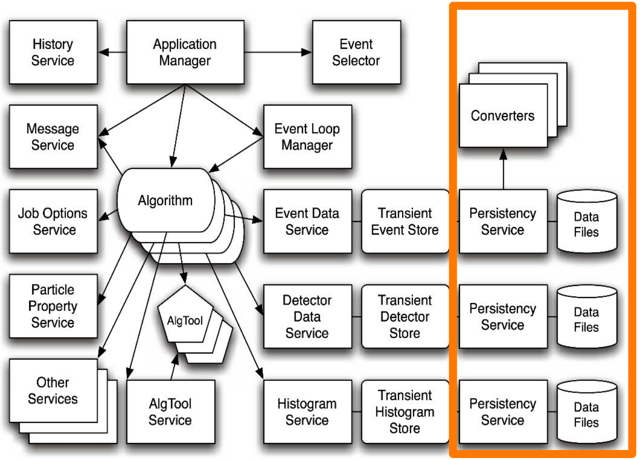
- **CPU usage %-age per workflow (2018):**
  - **Simulation :** ~40%
  - **Reconstruction :** ~25% (Data + MC)
  - **Group Production (inc. Derivation) :** ~15%
  - **Event Generation :** ~10 %
  - **Analysis :** ~5%
- **Disk-space usage (today):**
  - About **200 PB** of data on disk (+10-20 PB/year)
    - **~50%** (**100 PB**) in the form of **DAODs,** *O(10 KB/evt)*
    - **~30% (60 PB)** in the form of **AODs,** *O(100 KB/evt)*
  - Frequently accessed data are kept on disk (e.g. analysis)
  - **Using practically all of the pledged resources**
    - *Never enough disk-space...*

CERN-LHCC-2020-015

Athena in a Nutshell

- **Athena is the main ATLAS software framework (open-source)**
  - Based on the Gaudi framework, a common LHCb and ATLAS effort (also open-source)
- Athena consists of about **4 (1.5) million lines of C++ (python)** code
  - CMake is used for *building*, python for *configuration*, and C++ for *algorithms*
- It has been in use since the early days of the ATLAS experiment
- The main features of the Gaudi/Athena software paradigm are:
  - Based on Microsoft's Component Object Model (COM)
  - Components implement an **interface** and use other components through an **interface**
    - Main components are **services**, **tools**, and **algorithms**
      - **Algorithm :** The main building block of the **Event Loop**, called once per event
      - **AlgTool :** A plugin that helps an algorithm perform certain actions
      - **Service :** A plugin providing a common service to multiple components

Athena in a Nutshell (cont'd)



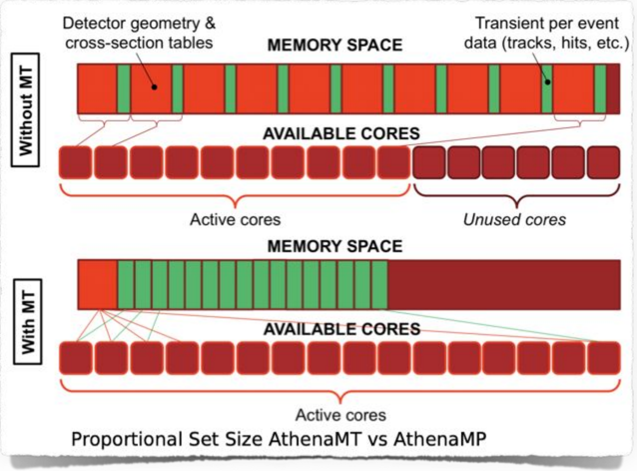
- An Athena job comprises **four** main steps:
  - **Configuration:** Parsing of configuration scripts/user input
  - **Initialization** : Initializing all job components
  - **Execution** : Executing the algorithms (Event Loop)
  - **Finalization** : Finalizing all job components
- **Event data are shared across components via a dedicated Store**
  - **Algorithm A** *reads* data **X** from the Event Store and *writes* data **Y** to the Event Store
  - **Algorithm B** *reads* data **Y** from the Event Store and *writes* data **Z** to the Event Store
  - Algorithms can be chained but the execution order needs to be carefully coordinated!
- **Argonne is leading the core I/O software development in Athena**
  - Responsible for our most valuable asset: Our Data
  - An integral part of **every** ATLAS workflow

Different Mode of Operations in Athena

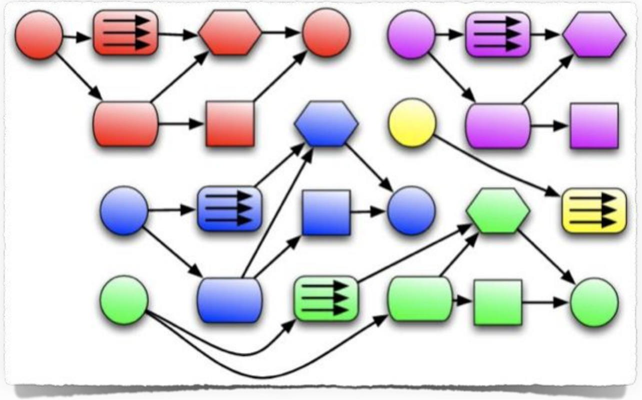
- **Serial Athena:**
  - This is the original mode of operation in Gaudi/Athena
  - A single process executes all job steps sequentially
  - The execution of algorithms in the Event Loop are determined by the job configuration
- **Multi-process Athena (AthenaMP):**
  - This mode builds on top of the serial Athena and was the primary mode of operation in Run-2
  - After *initialization*, multiple **processes** are created, each processing a unique set of events
  - Allows sharing a significant amount of memory (allocated during *initialization*) between processes
    - More efficient than running multiple serial Athena jobs in parallel
    - For some workflows (e.g. reconstruction) still not "good enough" memory sharing
    - For some workflows (e.g. derivation production) still an excellent choice
- **Multi-thread Athena (AthenaMT):**
  - This will be the primary mode of operation in Run-3 for otherwise memory-bound workflows
  - After *initialization*, multiple **threads** are created, algorithms are then executed on these threads
    - Allows not only inter-event parallelism but also intra-event parallelism
    - Maximizes the amount of memory shared across threads improving memory footprint

Multi-threading in Athena

Memory



Concurrency

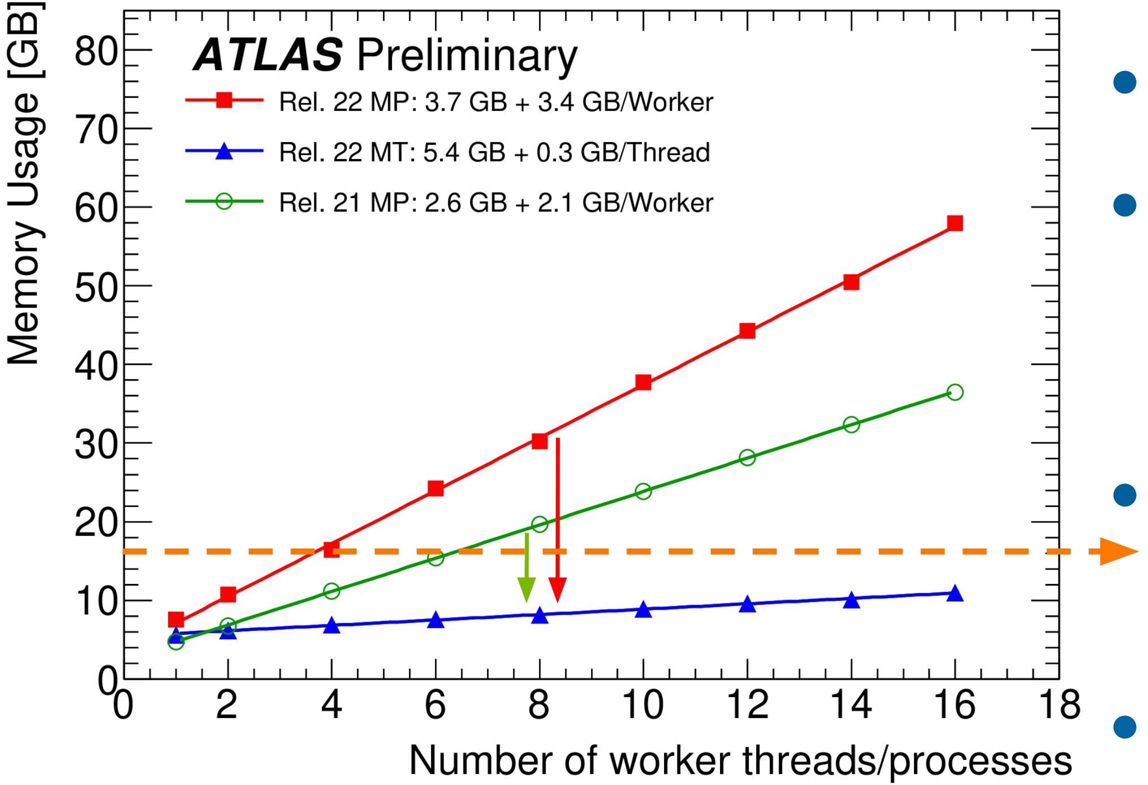


Color: Event
Shape: Algorithm

- **Only Event Loop is multi-threaded/multi-processed**
  - All remaining steps are executed serially
- **Main challenges:**
  - More complex task scheduling, race conditions, memory corruptions, lock contentions
  - From the I/O perspective:
    - AthenaMP: Handling of multiple parallel output files from the worker processes
    - AthenaMT: Handling of concurrent data from the worker threads

Performance of Multi-threaded Reconstruction

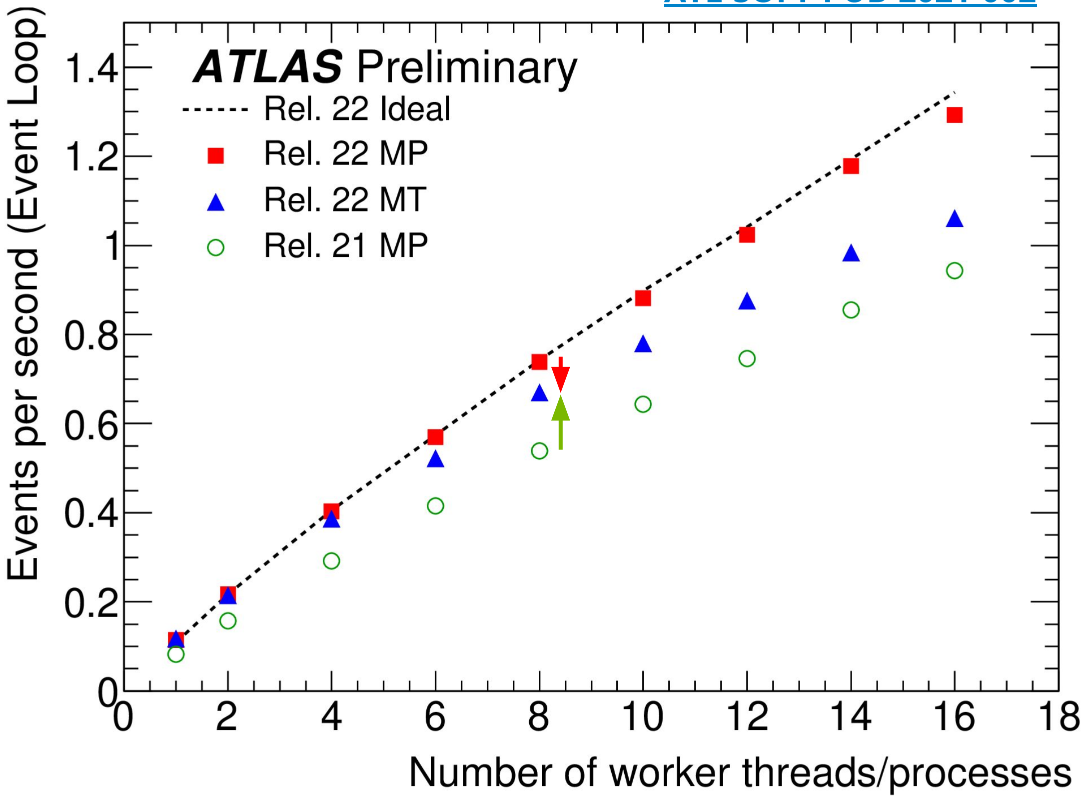
ATL-SOFT-PUB-2021-002



- **Main goal: Memory**
- Memory scaling:
  - **Rel. 21 MP :** **2.1 GB/process**
  - **Rel. 22 MP :** **3.4 GB/process**
  - **Rel. 22 MT :** **0.3 GB/thread**
- At 8 processes/threads:
  - **Resource limit 2GB/core → 16 GB**
  - **~60-70% gain in overall memory**
- **All in all great success!**

Performance of Multi-threaded Reconstruction

ATL-SOFT-PUB-2021-002



- Metric: Event throughput
- Rel. 22 improves Rel. 21
  - Thanks to various optimizations
    - Primarily in track reconstruction
- **MT mostly keeps w/ MP**
  - Similar throughput, less memory
- At 8 processes/threads:
  - **MT is ~90% efficient w.r.t MP**

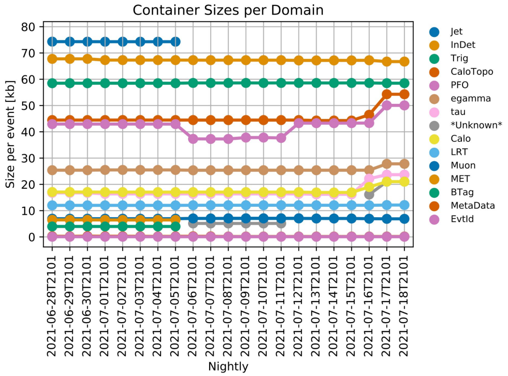
Argonne @ ATLAS S&C

Argonne @ ATLAS Software & Computing

- **Argonne is heavily involved in ATLAS Software & Computing**
    - **We're leading the core Input/Output (I/O) software effort:**
        - Peter Van Gemmeren (Leadership, Shared I/O)
        - Frank Berghaus (MetaData, Bytestream)
        - Alaettin Serhan Mete (Shared I/O, Storage)
    - **We're leading the software performance optimization effort:**
        - Alaettin Serhan Mete (**S**oftware **P**erformance **O**ptimization **T**eam [SPOT] coordinator)
        - Walter Hopkins and Evangelos Kourlitis (Simulation optimization)
    - **We're strongly involved in emerging workflows and the HPC efforts:**
        - Doug Benjamin and Rui Wang

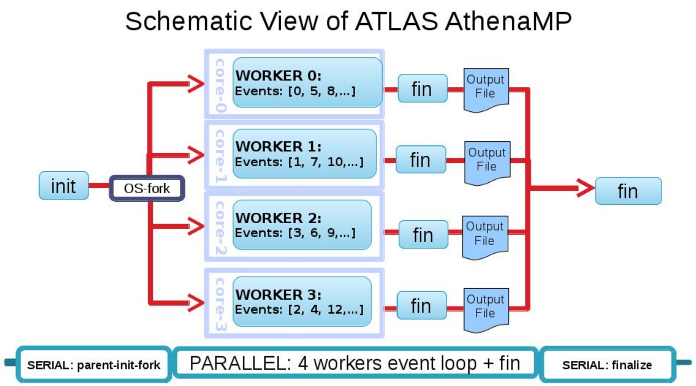
Software Performance Optimization in ATLAS

- **Software Performance Optimization is integral to ATLAS S&C**
  - **The relevant team (SPOT) is coordinated by Alaettin Serhan Mete since 2018**
    - In a nutshell, it's tasked to coordinate all the relevant work for official ATLAS workflows
- **The team has multiple responsibilities**
  - Ensuring the resource usage of official ATLAS jobs meet the production system constraints
  - Developing and integrating the necessary software to perform (regular) monitoring/profiling
  - Helping other developers, organizing tutorials, hackathons etc.
- **SPOT played a significant role in the recent migration effort**



- **A recent example from daily monitoring**
  - Evolution of AOD content as a function of days
  - Closely monitor what/how much data we store
    - Removing Jet/MET/FTag, changing noise thresholds etc.
  - Similar monitoring is done for CPU and memory
    - Catch and fix issues before they make it to production

Shared I/O



- **Shared I/O was designed for AthenaMP**
  - Suggested, developed, and deployed by **Peter VG**
  - Originally outputs from workers were merged separately
  - Shared I/O enabled doing this "*on-the-fly*"
  - Not only improves throughput but also job success rates
    - **Reduces wall-time by 20-30% in derivation prod.**
    - No additional (merging) jobs
- **In Run-2 Shared I/O was successfully used in official production**
  - Primarily used for derivation production but supports all AthenaMP workflows
- **For Run-3, various improvements were implemented**
  - Taking better advantage of parallel data compression provided by *ROOT*
  - **Further boosting processing times by 20% (reconstruction) to 30% (derivation)!**
  - The effort is lead by **Peter Van Gemmeren** and **Alaettin Serhan Mete**

Simulation Optimization Studies

- **Simulation led the CPU usage in 2018 by ~40%**
  - Various optimization are performed towards Run 3:
    - Increasing the “fast” simulation (parametrized Calorimeter response) usage
      - Towards Run-4 : Fast-chain (merged simulation + reconstruction) - **Rui Wang**
    - Optimizing pile-up simulation (using MC overlay)
    - Adopting various tuning and technical optimizations
  - **The target is to improve simulation performance by up-to 50% in Run 3 w.r.t. Run 2**
    - Most of these are either already deployed or in validation phase
- **Argonne is playing an important role** in simulation optimization
  - **Evangelos Kourlitis** and **Walter Hopkins** work on various Geant4 performance optimizations
    - Geant4 is a toolkit for simulating the passage of particles through matter
  - Three main focus points:
    - Magnetic field tailored switch-off (up to 10% speed-up in full simulation)
    - Woodcock tracking for photons (aims to simplify particle propagation via reduced iterations)
    - Machine learning (ML) correction for photons (aims to speed-up calorimeter simulation)

Disk-space Usage Optimization Studies

- Helping with the storage problem: **Lossy Float Compression**
  - The main data structure that is used for persistency is single precision floating point numbers
    - Each number occupies **4 bytes** in memory → **7 decimal places of accuracy**
  - This is well beyond detector/physics precision for most of the variables
  - Goal: Zero out the “redundant” bits to help the compression algorithms do a better job
    - It’s possible to gain **up to 30% in disk-space for primary AODs**, i.e. **O(1 PB/year)**

- **Important to have a good synergy within the Argonne group!**
  - **Walter Hopkins** and **Alaettin Serhan Mete** mentored **Robert Snuggs (SULI)** last year
    - Check the impact of lossy float compression on physics analyses
    - In most cases, the impact is found to be well within the expectations…

Metadata in ATLAS

Data that provides information about other data

- What we mean:
  - Detector conditions
  - Run parameters
  - Simulation parameters
- Our specific responsibility:
  - **Metadata stored in data files about the events in that file**
  - *Note*: ATLAS also has metadata in central databases
- Argonne project from the start:
  - Leadership from Jack Cranshaw and David Malon

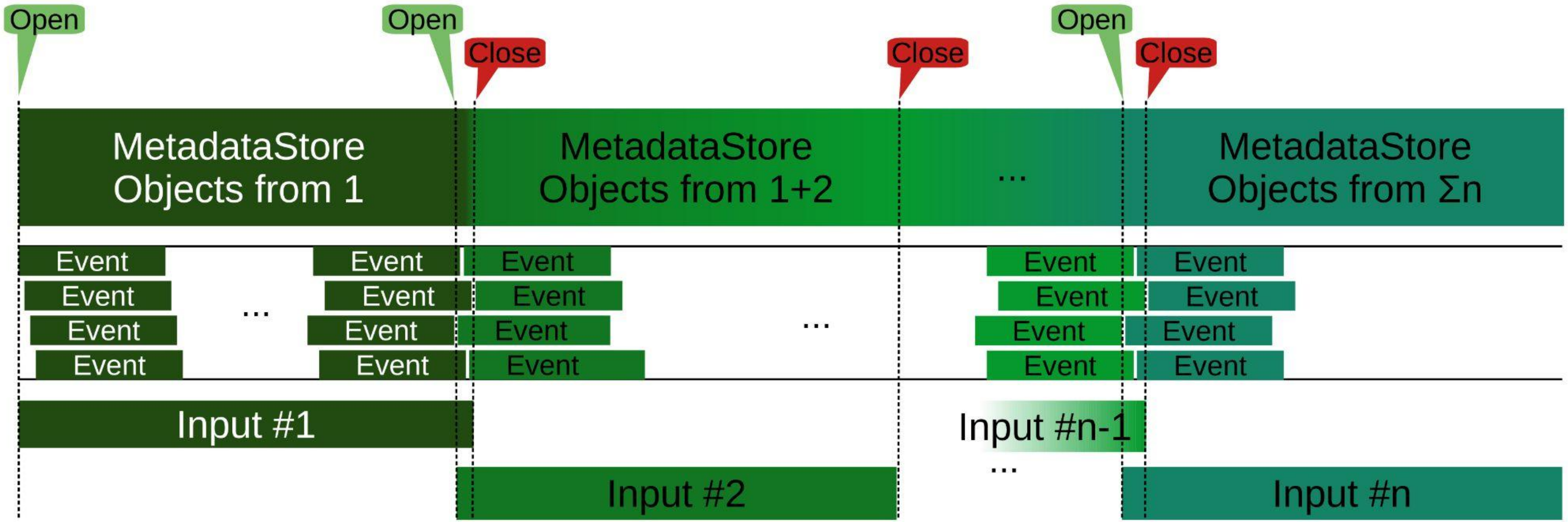
Traditional metadata

ATLAS in-file metadata

- In-file metadata is used to
  - *Configure* the software: input files provide configuration information
  - *Initialize* software components: e.g. event bookkeeping about past selection
  - *Map*ping: What input content maps onto what type in the running software
  - *Trigger* decoding: what was the trigger configuration and menu during data taking
  - *Normalization*: tracking event selection and luminosity blocks
  - *Annotations*: information added by analysers

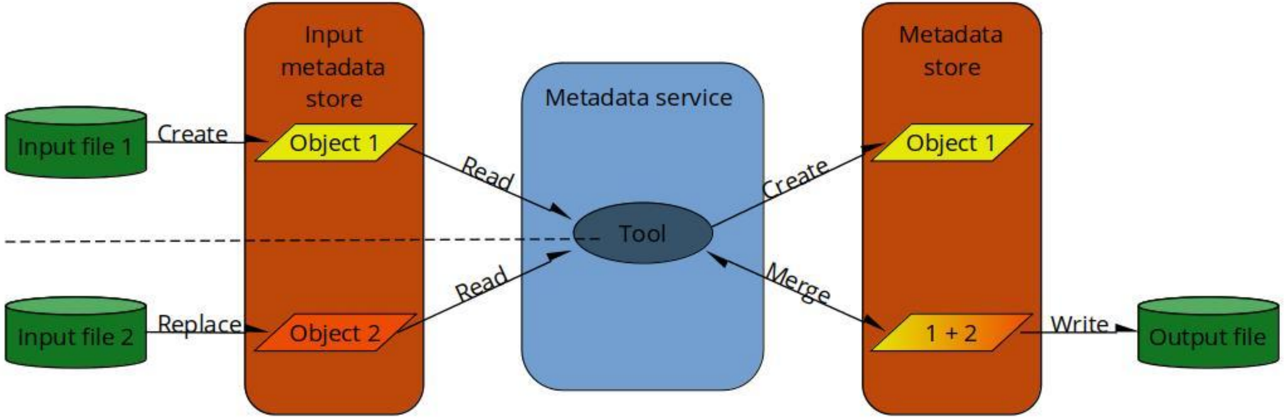
- Not managing content, but infrastructure to
  - Read from input
  - Propagate through job
  - Make information available to clients
  - Write to output

Concurrency challenge for in-file metadata



- Input files are opened and closed in sequence
  - Overlap required to handle data access on-demand
- Events need metadata from multiple sources around file boundary

Pragmatic approach to concurrency



- Simplify by remove unused content
- React to file operations during the job
- Provide merge metadata safely & provide thread safe interfaces

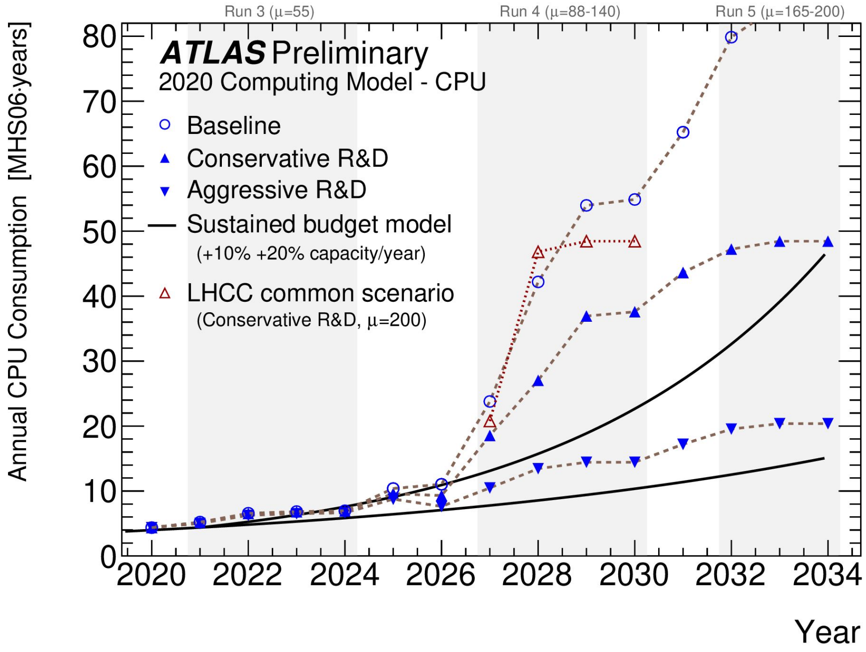
RAW data format

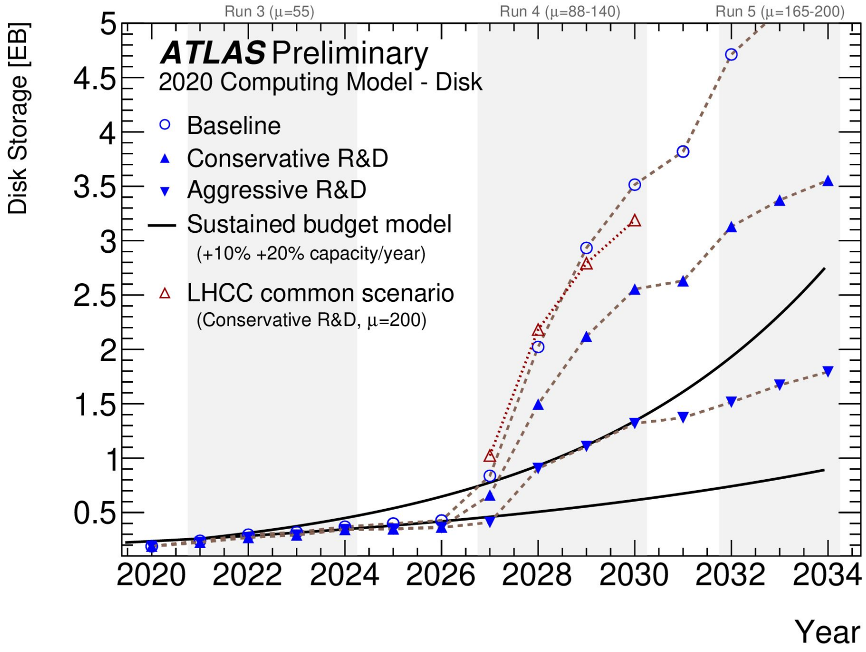
- Designed to fit the requirements of trigger and data acquisition
  - Fast and flexible merging of information from many sources
- Event fragments inspired by network packets:
  - Header describing source, event, and payload size
  - Payload of actual detector readout
- Event building
  - Many fragments are merged into a full event, with its own header
- Outside of the data acquisition process this is required for
  - Processing of the RAW data
  - Simulating the trigger

Towards Future

Looking into the future…

[CERN-LHCC-2020-015]





- **Run-4 and beyond requires aggressive R&D work in S&C**
  - Both compute (CPU) and disk-space requirements are quite high…

Conclusions

- **AthenaMT migration is being finalized prior to Run-3**
  - Run-2 reprocessing campaign is just around the corner
  - **A huge amount of work but it definitely pays off!**
- **Argonne plays a significant role in ATLAS S&C**
  - **Our contributions/leadership are vital for the success of the ATLAS S&C project**
  - We lead the effort in many topics: Core I/O, Software Performance Optimization etc.
- **The future brings many new challenges**
  - Heterogeneous architectures, emerging workflows, storage problem etc.
  - **There are a lot of challenges but also fun projects/opportunities ahead**
- **Argonne will continue leading the way in scientific computing!**
  - **... and we're certainly looking forward to it!**


Argonne
NATIONAL LABORATORY
75
1946–2021