

ATLAS I/O Performance Optimization in As-Deployed Environments

Thomas Maier

on behalf of the ATLAS Collaboration

Ludwig-Maximilians-Universität München

LS Schaile

13 April 2015



Co-authors: D. Benjamin, W. Bhimji, J. Elmsheuser, P. van Gemmeren,

D. Malon, N. Krumnack

Introduction

- ATLAS produces huge amounts of data during physics data taking periods
- Grid sites deploy a wide variety of storage technologies → require also a wide range and reliable ways to access data for prompt physics analysis
- ATLAS has established a working group to address a range of areas related to I/O performance
 - ▶ Monitoring, measurement, and data collection of I/O performance, both in cleanroom(local) and Grid environments
 - ▶ Evaluate implications for decision-making on many fronts → persistent data organisation, caching, best practices, framework interactions with underlying service layers, and settings at many levels (application code, Grid sites,...)
 - ▶ Improving robustness of distributed data access → failover mechanisms for error recovery → proper propagation of non-recoverable errors
- This talk will only present a portion of this work today

Range of Analysis Computing in ATLAS

- Local processing
- Distributed data analysis
 - ▶ Running on Grid sites using PanDA
 - ▶ Running on batch systems
- Access patterns
 - ▶ Remote access protocols → dcap, XRootD, WebDAV (Talk by Johannes Elmsheuser)
 - ▶ Copy-to-scratch
 - ▶ Local disk access

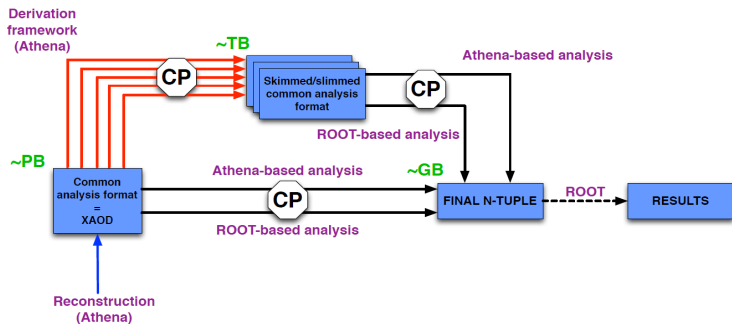
Instrumentation for Performance Monitoring

- Local tests
 - ▶ Direct (manual) way to test performance
 - ▶ Allows to test on a very basic level, but does not necessarily represent the way how a data analysis is run in reality
 - ▶ Can be easily modified for other access cases
- Hammercloud
 - ▶ Automated system to run stress and functional tests on Grid sites
 - ▶ Allows implementation of tests to monitor performance
- Analysis environments for new Event Data Model (EDM)
 - ▶ "Enforce" centralisation of analysis usage
 - ▶ Provide hooks for central monitoring

The New ATLAS xAOD Event Data Model in a Nutshell

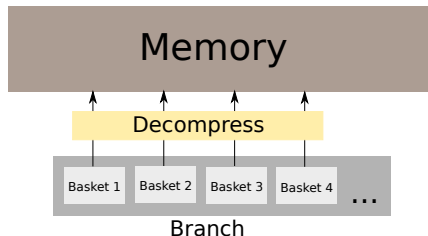
- Problems with Run I analysis model
 - ▶ Disconnect between data reconstruction output (AOD - Analysis Object Data) and data format used by physics analyses (DPD - Derived Physics Data)
 - ▶ Huge amount of data/software duplication
- Requirements for Run II
 - ▶ Prepare for increased data rates ($\sim 2\times$ that of Run I)
 - ▶ Provide similar I/O performance for physics analyses
 - ▶ In general homogenisation \rightarrow less steps from data preparation to physics results
- Development of new data model (Talk by Scott Snyder)
 - ▶ Merging of AOD and DPD to new format called xAOD
 - ▶ Class based information storing \rightarrow directly analysable in ROOT and ATLAS software framework Athena

The New ATLAS Analysis Model in a Nutshell



- Data preparation after reconstruction → DxAOD recommended data format
 - ▶ Centrally produced, trimmed down xAOD
 - ▶ Heavily reduced content, customised to the needs of different physics groups
 - ▶ Talk by James Catmore on the Derivation Framework

Simplified Picture of Data Storage in ROOT Files

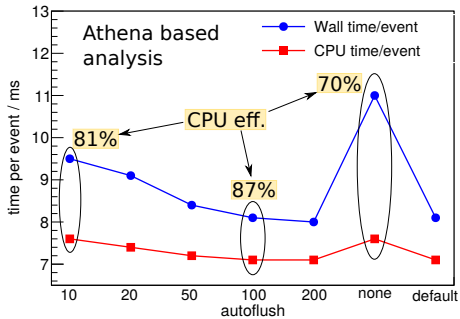
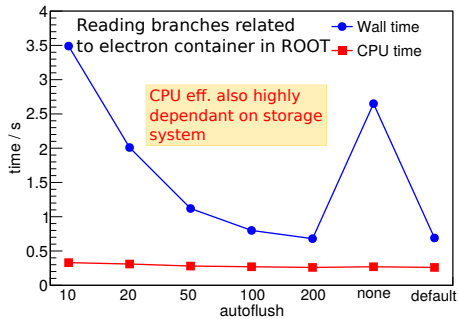


- Properties (e.g. electron p_T , eta, phi, ...) are stored in separate branches
- Information within branches is stored in multiple, separate baskets
- Accessing a property contained in a given basket \rightarrow whole basket is loaded into memory \rightarrow process has to wait until I/O operation is completed

Autoflushing as a Handle on Number of Baskets

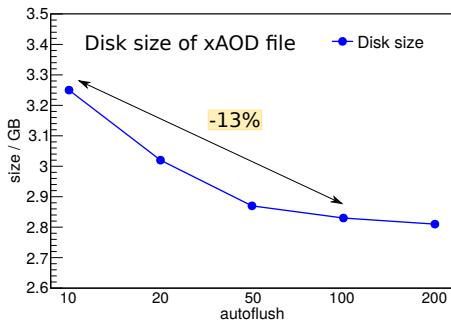
- Number of baskets heavily affects reading speed if all events are accessed
- Using Autoflush to steer number of baskets → while writing, flush buffered data to disk
 - ▶ after a certain number of events have been processed
 - ▶ after a certain amount of bytes have been processed
- Has been found to be a very effective handle in the past → value of 10 found to be most practical for old AOD format
- Old AOD (~ 300 branches) ↔ new xAOD (> 2000 branches) → requires higher autoflush setting
- Re-optimisation needed for new xAOD format to adapt to new requirements

Impact of Autoflush on I/O



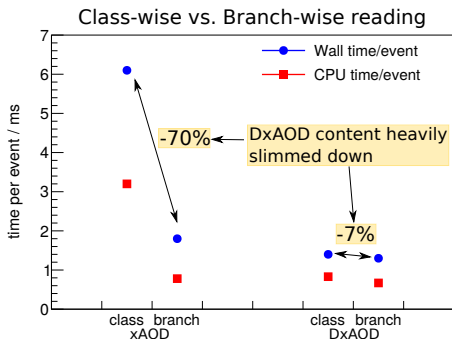
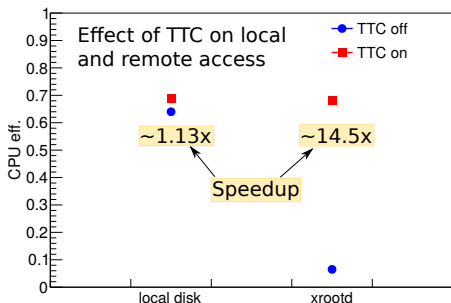
- Noticeable impact of Autoflush configuration on reading speed
 - ▶ "None": no autoflushing, number of baskets determined by default basket size
 - ▶ "Default": flushing according to amount of bytes in buffer (30MB)
- Old Autoflush setting of 10 clearly not suitable for new xAOD format

Further Observations



- Higher Autoflush values also reduces the disk size of the xAOD file
 - ▶ More compressable data per basket
 - ▶ Higher compression rates
- Slight increase in Virtual Memory foot print → acceptable tradeoff
- New Autoflush value of 100 is used for (D)xAODs

Additional Handles - TTC & Branch-wise Reading



- Pre-caching of data via the TTreeCache (TTC) feature of ROOT
 - ▶ In general beneficial to analysis speed
 - ▶ Very important for remote access → running on Grid sites
- Feature in xAOD EDM to toggle access mode for ROOT access
 - ▶ Class-wise → all branches connected to the container are read
 - ▶ Branch-wise → branches are read when respective properties are accessed

Conclusions

- New ATLAS analysis model and data format introduced in preparation for next period of data taking
- Old configurations and handles on I/O performance need to be revisited and re-optimised
- First improvements already found their way in the new xAOD format
- Further plans
 - ▶ Investigate benefit from more differentiated settings (xAOD \leftrightarrow DxAOD)
 - ▶ Extend monitoring of data access patterns and performance in user jobs
 - ★ Integrate with production job reporting
 - ★ Integrate with ATLAS analytics infrastructure for decision support
 - ▶ Further establish monitoring via Hammercloud