# A new petabyte-scale data derivation framework for ATLAS



Journal of Physics: Conference Series **664** (2015) 072007 doi:10.1088/1742-6596/664/7/072007

# A new petabyte-scale data derivation framework for ATLAS

**James Catmore[1], Jack Cranshaw[2], Thomas Gillam[3], Eirik Gramstad[1], Paul Laycock[4], Nurcan Ozturk[5], Graeme Andrew Stewart[6] on behalf of the ATLAS Collaboration**

[1] Fysikkbygningen, University of Oslo, P.O. Box 1048 Blindern, N-0316, Oslo, Norway
[2] Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, Illinois 60439, U.S.A.
[3] Cavendish Laboratory, 19 J J Thomson Avenue, Cambridge, CB3 0HE, U.K.
[4] University of Liverpool, The Oliver Lodge Laboratory, Liverpool L69 7ZE, U.K.
[5] University of Texas at Arlington, Box 19059, Arlington, Texas 76019, U.S.A.
[6] School of Physics and Astronomy, University of Glasgow, Glasgow G12 8QQ, U.K.

E-mail: james.catmore@cern.ch

**Abstract.** During the Long Shutdown of the LHC, the ATLAS collaboration overhauled its analysis model based on experience gained during Run 1. A significant component of the model is a "Derivation Framework" that takes the petabyte-scale AOD output from ATLAS reconstruction and produces samples, typically terabytes in size, targeted at specific analyses. The framework incorporates all of the functionality of the core reconstruction software, while producing outputs that are simply configured. Event selections are specified via a concise domain-specific language, including support for logical operations. The output content can be highly optimised to minimise disk requirements, while maintaining the same C++ interface. The framework includes an interface to the late-stage physics analysis tools, ensuring that the final outputs are consistent with tool requirements. Finally, the framework allows several outputs to be produced for the same input, providing the possibility to optimise configurations to computing resources.

## 1. Introduction

A feature common to many physics analyses is the use of intermediate-sized data formats at some stage of the analysis procedure. Typically these formats are made directly from the retained output of the reconstruction (known in ATLAS [1] as Analysis Object Data or AOD) and often have the following features:

(i) their size is usually around a few percent to a few per mille of the input data
(ii) they are typically aimed at one analysis or perhaps a group of related analyses (e.g. sharing the same final state)
(iii) they usually contain all of the information necessary to perform smearing, scaling, selection, calibration and other operations on reconstructed objects (collectively known in ATLAS as combined performance operations), and the systematic uncertainties related to these operations
(iv) they are used privately by physicists or groups of physicists to produce their small n-tuples, on which the final analysis is done

(v) they are typically reproduced 10-12 times per year, but are often read several times per week by the analysis teams

The third point above is particularly important since an optimal understanding of the reconstruction, which feeds into the combined performance recommendations, tends only to be achieved after many months of study of the data and Monte Carlo. Moreover different domains of the reconstruction update their recommendations asynchronously, so it is usual practice to store all the information needed to allow the application of these recommendations at the user analysis level.

In Run 1 the AOD files were not ROOT-readable and so had to be converted to a series of ROOT formats which were typically a significant fraction of the input data size [2]. Beyond the production of these large n-tuples, ATLAS did not further coordinate the design and production of intermediate formats, so it was left to users to both write and produce them privately. This led to a plethora of different incompatible intermediate formats, and more importantly imposed large demands on the users who were required to run across large datasets (the full AOD or the large n-tuples) via user grid jobs. To ameliorate these problems, in Run 2 the large n-tuples will not be produced and the intermediate formats will be made centrally from the AOD, using common software. This software, known as the "derivation framework", is the topic of this paper.

Since the terminology differs from one experiment to another, it is necessary to state the ATLAS definitions for the three standard operations for information removal:

- **Skimming** is the removal of whole events, based on some criteria related to the features of the event.
- **Thinning** is the removal of individual objects within an event, based on some criteria related to the features of the object.
- **Slimming** is the removal of variables within a given object type, uniformly across all objects of that type and all events. Unlike the other operations, slimming does not vary depending on any event/object properties: the same variables are removed for every event and object.

## 2. The ATLAS analysis model in Run 2

The analysis model developed for Run 2 is shown in Figure 1. As can be seen the model is based on a new ROOT-readable format (xAOD) [3] produced directly by the reconstruction and which replaces the old AOD. As well as the format itself the xAOD also brings with it an Event Data Model (EDM) which is used consistently across all reconstruction and analysis domains. The derivation framework, the second ingredient in the model, is used to create the intermediate data products from the xAOD by removing (and adding) information whilst maintaining the structure and EDM used in the original xAOD. The third and final component of the model is the analysis framework, which is used by physicists to read the derived data formats, apply various combined performance tools and produce the final small n-tuples, from which plots are produced and upon which statistical analyses are based.

The role of the derivation framework should therefore be seen both in terms of software and computing: it provides physicists with tools to define the intermediate formats, and these tools are run on the central production system. Analysts are thereby freed from the trouble of designing their own intermediate formats, and the considerable labour involved with running over the entire data sample themselves. Users will still have access to the full xAOD as is implied by the lower route in Figure 1, but it is expected that very few will work in this way and the large majority will use centrally produced derived formats.

Based on the practices of analysts in Run 1, approximately 100 derived formats are estimated to be needed for the full range of activities of ATLAS in Run 2. Collectively these are required to occupy no more disk space than a single copy of the full xAOD. This implies that overall,

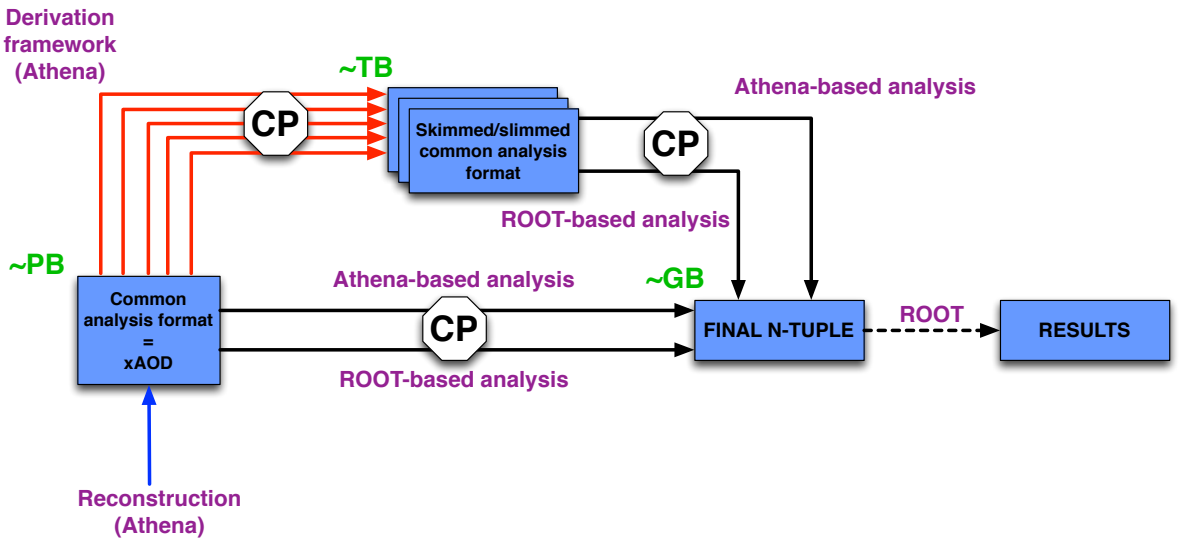


**Figure 1.** The ATLAS analysis model for Run 2

a given derived format should have around 1% (or less) of the volume of the input xAOD. An additional constraint is the processing time: the success of the model depends on achieving a decent turn-around, which in practice means a given production should complete in a small number of days. This imposes a requirement that together the derived formats should take less than 2 seconds per event to build. This is especially important during data taking operations, when derivation production will launch as soon as new data arrives at the Grid sites.

As well as fitting into this resource envelope, there are a number of requirements that the framework must satisfy:

- It must be able to run reconstruction-level operations, to apply fixes in the derived formats that cannot be applied in the main reconstruction due to the policy of freezing the software during data taking.
- It must be able to produce multiple output formats from a single input file, to reduce the number of times very large xAOD datasets must be accessed and to share the I/O burden. This is referred to as *train production*.
- It must provide tools to monitor the sizes of the output formats, the overlaps between them and the CPU usage.
- It must provide software to allow all groups to build their derived formats in a simple, modular and easily re-usable way.

## 3. Implementation

The derivation framework is implemented entirely in the ATLAS bulk data processing framework Athena [4]. This choice means the first three requirements above can be met almost immediately, since Athena has these capabilities already. Athena also provides the definitions for algorithms, tools and services, as well as providing crucial components such as the transient data store and thinning service. On top of the core software, the derivation framework then provides:

- interfaces to enable users to implement tools for skimming, thinning and augmenting the data
- a set of centrally provided tools for performing common selections, in particular a text-based event and object selection mechanism

- built-in lists of variables needed for the combined performance operations
- detailed monitoring of CPU, skimming and size fractions and overlaps between formats

The implementation of the derivation framework within Athena is shown in Figure 2. The design is based around a "kernel", which is an Athena event filter algorithm which additionally loops over the skimming, thinning and augmentation tools passed to it by the user, and applies them sequentially. Reconstruction fixes are applied to data immediately after being read in from disk, before the derivation process begins in the kernel, thereby ensuring that all such fixes appear in the output derived data. The slimming of variables occurs separately on the output stream, once the skimming decision has been made, and as such is not steered by the kernel. The configuration of the kernel, the tools fed to it and the slimming are done in standard job options files, and are left to the user or group defining the format.

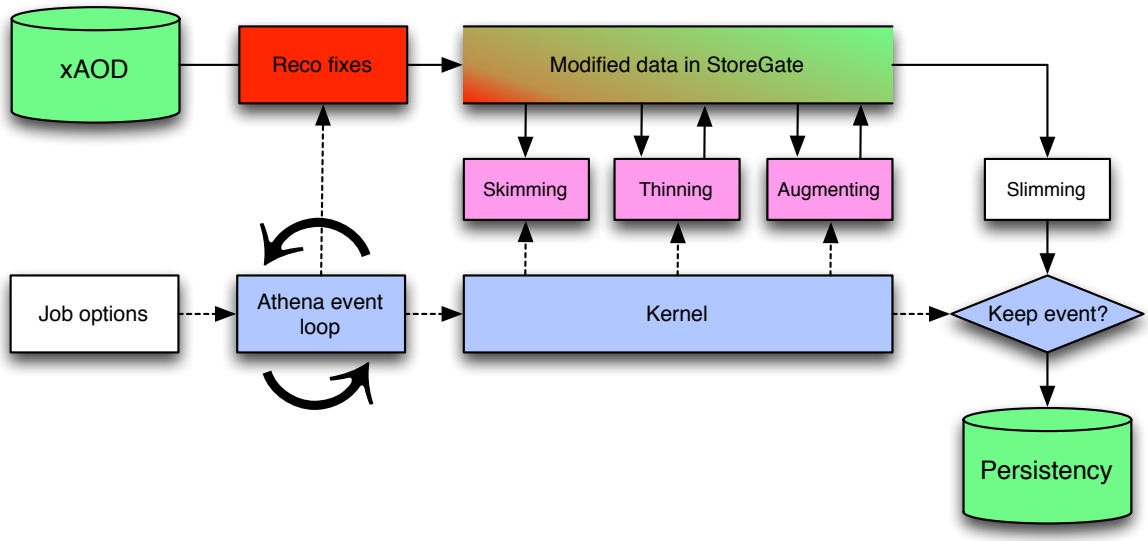


**Figure 2.** Implementation of the derivation framework in Athena

Figure 3 shows show multiple output formats are produced. Instead of a single kernel, several are instantiated (each with their own independent configurations) and write to separate output files, using the multiple output stream manager provided by Athena. It should be stressed that this does not imply that the kernels run on separate cores: this has not yet been attempted. It should also be stated that whilst the separate streams are independent, the transient data store is shared, so variables written by one kernel can be accessed by another.

### *3.1. Expression Evaluation*

Whilst it would be perfectly possible to implement each and every event and object selection as an independent C++ tool, this would lead to a large proliferation of code with the associated maintenance difficulties. Instead, ATLAS has sought to minimize the need for such dedicated tools by developing a generic tool, based on string parsing, for performing both object and event selection. Figure 4 shows the design of this tool: based on the BOOST Spirit Library [5], it performs the string parsing operation once at the start of the job, and then creates a virtual machine to execute the selection for each event/object. This avoids multiple parsings of the same string. The tool can access any variable from any container in the xAOD and can also interpret operators, common unary mathematical functions and constants. Examples of string selections are:

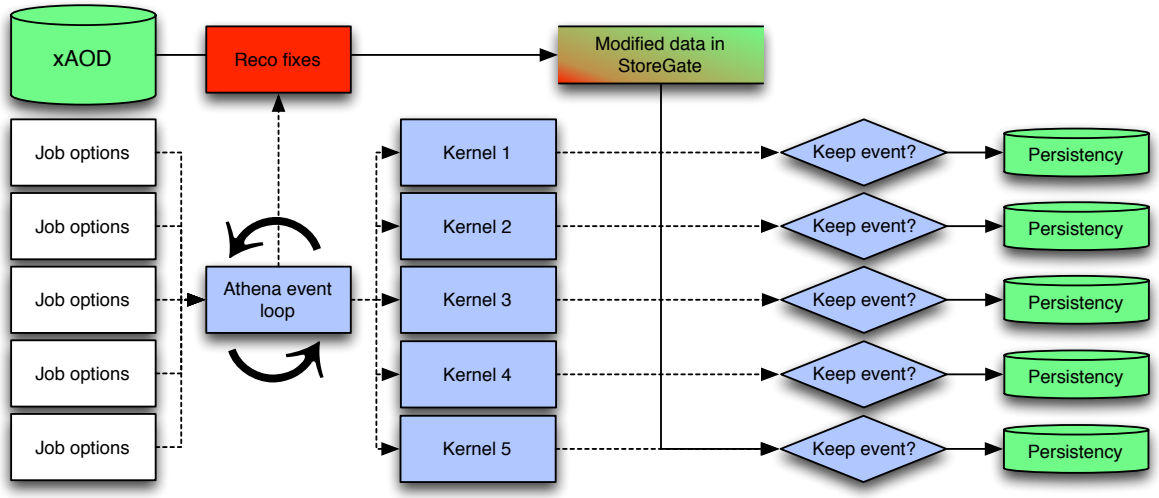


**Figure 3.** Derivation framework with multiple output streams

- event selection: `count(Muons.pt > 25.0*GeV && abs(Muons.eta) < 2.5) >= 4`
- object selection (for thinning): `InDetTrackParticles.pt > 5.0*GeV`

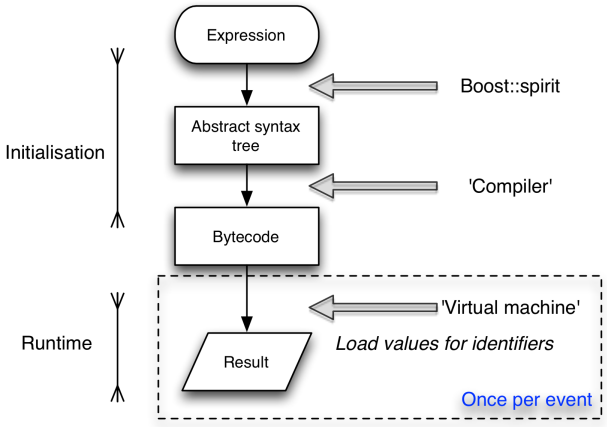


**Figure 4.** The expression evaluation tool design

*3.2. Slimming*

A procedure has been developed to ensure that the variables required for analysis are stored automatically in the derived data. This procedure uses a built in feature of the xAOD, known as PrintStats, which produces a list of all variables extracted from the transient store in a given job. Sets of tools that are known to be needed for a given domain of combined performance (e.g. muons, jets, electrons) are run once per analysis release on the full xAOD, in their various configurations. The resulting lists from the PrintStats service are then installed in the derivation framework, such that the required variables are automatically included. Additional variables, required for special applications, can be included via the job options.

*3.3. Augmentation*

Augmentation can take two forms in the derivation framework:

- Adding entirely new containers of reconstructed objects that are not available in the primary xAOD. This typically concerns jets and missing energy containers, which can be made from the information in the primary xAOD.
- Adding new variables to existing objects (decoration). This uses the built-in decoration facilities of the xAOD. Decorations may be added by individual kernels for use in a single format, or they may be added to all formats. Typically such common decorations are flags indicating whether a given object has passed certain combined performance criteria: by adding them centrally the tools making the decisions only need to be run once per train, rather than once for each format.

## 4. Readiness for Run 2

The derivation framework has already been used successfully during the 2014 Data Challenge to produce derived formats for the physics and combined performance groups, who have used them to prepare and commission their Run 2 analyses. The derivation production workflow has been fully integrated into the new ATLAS production system (ProdSys2) [6]. Approximately 60 derived formats are defined, with a further 20 under development. Figure 5 shows the size of each of these derived formats as a fraction of the input xAOD dataset size, for part of the 2012 collision data. Note that each run appears as a separate entry, so the total number of entries in the histogram is the number of formats, multiplied by the number of runs and the number of trigger streams (three in 2012). As can be seen from the plot, a large number of the derived formats are hitting the target of not occupying more than 1% of the input data volume, and the total size of all formats is just below the total size of the input. Nevertheless a tail can be seen and these derived formats must be reduced in size before data taking begins.

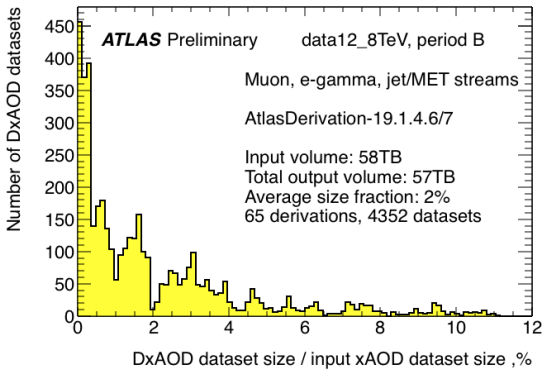


**Figure 5.** Size fractions for all defined derivations for part of the 2012 data

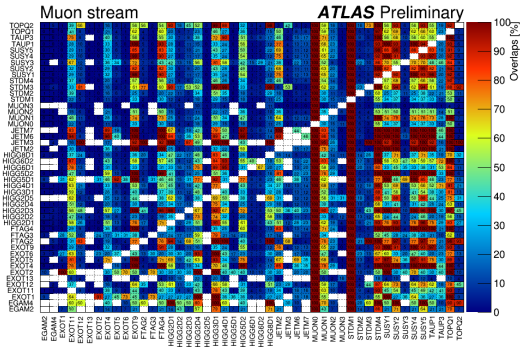


**Figure 6.** Overlap fractions for all defined derivations for part of the 2012 data, muon stream

Figure 6 shows the proportion of events in each format which also appear in the others (since the formats contain very different numbers of events, the table is not symmetric). It can be seen that in the most part the formats do not overlap excessively. Note that some accept very high proportions of the events and so overlap in event-wise terms very heavily with the others - to obtain a complete picture it is also necessary to look at variable-wise overlaps. Such information is also available from the metadata. As well as being extracted directly from the derivation jobs, the event-wise overlap information can also be obtained on a global basis (from all data sets) using the EventIndex database [7].

## 5. Conclusions

The Run 2 analysis model for ATLAS includes the centralised production of analysis specific data formats containing less information than the reconstruction output, but in the same format. Of order 100 derivations are foreseen, each with a size of approximately 1% of the input. A software framework, built on Athena, has been developed to produce these formats in bulk. The framework is in use already and the output data products are within the resource limits. It will be deployed from the start of Run 2 data taking in June 2015 and will reduce the computing workload on individual physicists, who will no longer have to run large private productions.

## References

[1] ATLAS Collaboration 2008. **The ATLAS Experiment at the CERN Large Hadron Collider**, JINST 3 S08003 doi:10.1088/1748-0221/3/08/S08003

[2] P. Laycock, N. Ozturk, M. Beckingham, R. Henderson, L. Zhou 2014 *J. Phys.: Conf. Ser.* **513**

[3] T. Eifert, M. Elsing, D. Gillberg, K. Koeneke, A. Krasznahorkay, E. Moyse, M. Nowak, S. Snyder, P. Van Gemmeren 2015. **Implementation of the ATLAS Run 2 event data model**. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) *J. Phys.: Conf. Ser.*

[4] M. Elsing, R. Seuster, G. Stewart, V. Tsulaia 2015. **Status and future evolution of the ATLAS offline software**. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) *J. Phys.: Conf. Ser.*

[5] `http://boost-spirit.com`

[6] M. Borodin, K. De, J. Navarro, A. Klimentov, T. Maeno, A. Vaniachine 2015 **Scaling up ATLAS production system for the LHC Run 2 and beyond: project ProdSys2**. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) *J. Phys.: Conf. Ser.*

[7] D. Barberis, J. Cranshaw, A. Favareto, A. Casani, E. Gallas, C. Glasman, S. Gonzalez, J. Hrivnac, D. Malon, F. Prokoshin, R. Yuan, J. Sanchez, J. Salt, R. Toebbicke 2015. **The ATLAS EventIndex: architecture, design choices, deployment and first operation experience**. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) *J. Phys.: Conf. Ser.*