NANOAOD: a new compact event data format in CMS

Karl Ehatäht[1,*] for the CMS collaboration

[1]National Institute of Chemical Physics and Biophysics, Tallinn, Estonia

Abstract. The CMS Collaboration has recently commissioned a new compact data format, named NANOAOD, reducing the per-event storage space requirement to about 1-2 kB. This represents a factor 20 reduction in storage space compared to the MINIAOD data format used previously for physics analysis at CMS. We envisage that the information stored in the NANOAOD data format is sufficient to support the majority of CMS physics analyses. NANOAOD also facilitates the dissemination of analysis methods and the automation of standard workflows for deriving conditions and object calibrations. The latest developments of this project will be presented.

1 Introduction

NANOAOD is a new data format recently commissioned by the CMS Collaboration [1]. The format aims to alleviate the need to produce user-defined ROOT [2] Ntuples that have proliferated during the Run 2 data-taking period of the LHC. In particular, the NANOAOD format aims to reduce the growing demands on disk capacity and CPU resources, to simplify increasingly complex data analysis recipes, and avoid sophisticated dependencies on CMS software when performing physics analysis. A prototype of the new data format was first presented in [3].

The content of NANOAOD averages to about 1-2 kB per event which is about 20 times less than that of MINIAOD, the next most compact data tier [4]. The reduction in the event size is achieved by storing only the most common high-level physics objects, and by limiting the numerical precision of stored variables, taking into account the experimental resolution of the detector. The main principle that is followed in the design of the NANOAOD format is to satisfy the requirements of as many physics analyses as possible, while keeping the event size to a minimum. More details about the design rationale and implementation of the format, impact on physics analysis and software models, and future goals are presented in the following.

2 Design

2.1 Data tiers

The data model employed by CMS has a tiered structure, where each subsequent format contains a more compact summary of the event data than its predecessor. There are two possibilities to reduce the size of data: by selecting a subset of events ("skimming"), or by

*e-mail: karl.ehataht@cern.ch

reducing the event content. The first option is employed by the CMS trigger system that lowers the rate at which data gets recorded from a collision rate of 40 MHz to a few 100 Hz. The information read out from the detector is stored in the RAW data format which consumes roughly 1 MB of disk space per event, assuming the pileup conditions of Run 2 data-taking period. The events in RAW format are then passed to event reconstruction algorithms, the very detailed outputs of which are stored in RECO format. The RECO data tier requires about 2-3 times as much storage space compared to the RAW format.

In order to provide the data in a suitable format for any physics analysis, the AOD (Analysis Object Data) tier was created in the beginning of Run 1. The AOD format is derived from a subset of the RECO format which reduces the data size by 85%. The conversion from the RAW format to the AOD format can be rerun every few years, with the aim to incorporate the latest developments in object reconstruction, identification and calibration into the data available for physics analysis. During Run 1, each individual physics analysis group typically produced custom Ntuples based on the AOD data, which turned out to be too demanding on the CMS infrastructure for it to be sustainable during Run 2. It was also observed that custom built Ntuples produced by one group were typically shared with other groups who used them for different purposes. In order to make the creation of custom Ntuples more efficient and sustainable, the CMS collaboration created the MINIAOD format. The aim of the MINIAOD format was to provide sufficient event information to cover 80% of physics analyses, while reducing the event size by a about a factor 10 compared to the AOD. The MINIAOD data tier was commissioned after Run 1 and has been successfully used for about 95% of physics analyses performed by CMS during Run 2. There can be a few MINIAOD production campaigns in a single year.

Naturally, the next logical step was to provide the data directly in the format most commonly consumed by physics analysis groups. Thus, by the end of Run 2, a new, centrally produced data format, the NANOAOD Ntuple format, was created, and commissioned shortly thereafter. The new format aims to be sufficient to cover the needs of 50%-70% of physics analyses, potentially more.

2.2 Event contents

A NANOAOD Ntuple is a ROOT file that contains a flat `TTree` (the ROOT columnar format), where entries correspond to events and branches to physics observables. For every event, the properties of high-level objects (such as mass of jets) are stored as arrays in the branches. The length of those arrays is stored in a separate branch, and is determined by the number of objects in an event. Depending on the nature of the observable, event level variables may be stored as scalar values (for instance, missing transverse momentum) or as arrays of event weights. Every stored variable has a fundamental type which can be a signed or an unsigned integer, a float or a boolean. The branch names follow a naming convention that helps to identify variables belonging to the same object collection, as illustrated in Table 1. The branches are equipped with additional metadata accessible via the `TBranch::GetTitle()` function. This information serves the purpose of providing a brief documentation for each variable.

One of the major benefits of the NANOAOD format is its small size of about 1-2 kB per event. The reduction in size is achieved by storing only the high-level objects used by most physics analyses in the NANOAOD Ntuples, such as electrons, photons, muons, hadronic τ decays (τ_h) and jets. Low-level information such as constituents of these high-level objects, as well as low-level detector information such as tracks and clusters are omitted. All high-level objects stored in the NANOAOD format are required to pass preselection criteria, which are suitable for most physics analyses. For instance, most physics analyses in CMS use τ_h of $p_T >$

20 GeV or higher. The τ_h stored in NANOAOD Ntuples are required to pass the condition $p_T > 18$ GeV. The p_T threshold is lowered by 2 GeV in order to allow physics analyses to compute the effect on systematic uncertainties on the τ_h energy scale. Figure 1 illustrates the storage space taken by different types of high-level objects stored in NANOAOD Ntuples.

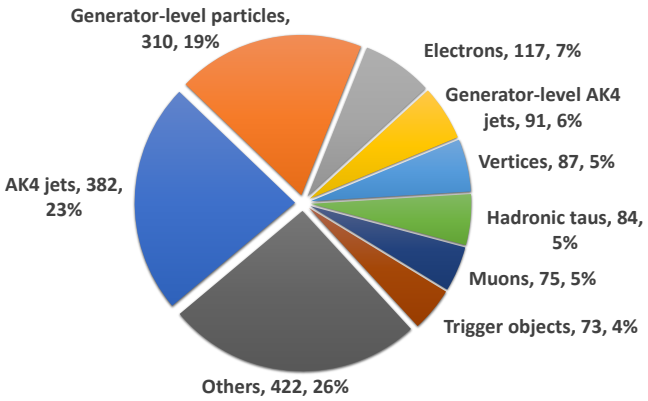

Figure 1. Breakdown of storage spaces taken by different high-level objects, on average per event, in a $t\bar{t}$ event sample produced by Monte Carlo simulation. The storage space is shown in bytes and in percentages relative to compressed event size.

Two more key principles are employed in order to reduce the size of NANOAOD files: store the output of e/γ, τ_h and b-jet identification algorithms, but drop their input variables, and omit redundant information that can be later derived from the existing event content. The latter principle, for instance, concerns systematic variations of variables such as jet energy scale which can be recomputed based on the four-momentum of the jet. The file size is further reduced using the LZMA compression algorithm. In order to improve the compression of the NANOAOD files, the precision of 32-bit floating point variables is reduced to match the resolution of the detector on the physics observables they represent. The experimental resolution on most observables does not reach the 10^{-6} level precision of a floating point number. For this reason, the least significant bits in the mantissa of the floating point numbers are zeroed in the NANOAOD files. The number of boolean variables in the NANOAOD format is reduced by combining multiple boolean variables into a single integer bitmask where possible. The latter optimization technique typically applies to working points of particle identification discriminants.

This high level of reduction in the file size is desirable due to the growing demand on the disk space required to store the CMS data as well as event samples produced by Monte Carlo simulation, as the LHC continues to operate. The 17 billion events that have been recorded in Run 2 data-taking period, plus accompanying 60 billion events produced by Monte Carlo simulations fit in just under 140 TB when stored in the NANOAOD format. An additional benefit of having a compact data tier is that the reading of events becomes much faster. The events stored in NANOAOD format can be read at rates in the order of kHz, including the time needed for LZMA decompression. The time required to convert events from MINIAOD to NANOAOD format amounts to typically 100 ms per event, which is fast enough to allow several production campaigns per year without a major impact on CPU utilization.

3 User experience

3.1 Analysis model

The reconstruction and identification algorithms developed by dedicated physics object groups typically become more complex as the detector conditions evolve over time. The NANOAOD format provides a way to consolidate, manage, and therefore simplify the usage of such recipes by the physics analysis groups. The standardized format that is used to store the high-level objects in NANOAOD Ntuples makes these objects readily available for analysis without further processing, which allows to use the same analysis software to analyze data recorded during different data-taking periods. This is a necessary prerequisite in automating physics analyses, which lessens human involvement and therefore errors in the process.

None of the high-level objects are cleaned with respect to one another, or removed by some analysis-specific criteria. Instead, the objects are "linked" to one another if there is a meaningful association between the particles. For instance, a jet that a given muon is constituent of can be found using the `Muon_jetIdx` branch, as described in Table 1. This flexibility makes it possible to use the same set of Ntuples in different analyses, regardless of analysis-specific object selections and object cleaning criteria employed. Furthermore, the adoption of the standardized NANOAOD format renders it unnecessary for analysis groups to run custom Ntuple production, or spend time on "synchronization", i.e. spend time comparing the Ntuple contents between different groups, in order to ensure that all participants use the same definition of high-level objects in their analysis. Since less time is spent on such tasks, the time between data recording and final publication is shortened.

Table 1. Naming convention of object collections established in the NANOAOD format. The rows corresponds to branches in the `TTree` that is associated with the event content. Object attributes can have any of the following data types: signed or unsigned integer, float or boolean.

Branch name	Type	Data type	Function	Example
`nObject`	Scalar	Unsigned integer	Number of objects in collection `Object`	`nMuon`, `nJet`, `nGenPart`
`Object_var[i]`	Array	Any	Attribute `var` of the *i*-th object in collection `Object`	`Muon_pt[i]`, `Jet_mass[i]`, `GenPart_pdgId[i]`
`Object_otherIdx[i]`	Array	Signed integer	Index of the object from collection `Other` if it is linked to the *i*-th object from collection `Object`, and −1 otherwise	`Muon_jetIdx[i]`, `Muon_genPartIdx[i]`

3.2 Software model

The physics object groups, responsible for developing and publishing new revisions of reconstruction and identification algorithms, are encouraged to propagate the necessary changes to the NANOAOD format. The software used to convert events from MINIAOD to NANOAOD format has a modular structure, so that multiple people can easily contribute simultaneously without interfering with one another. Each new contribution is required to pass integration tests, which ensure that the impact to file sizes remains reasonable and that new features do not break some other functionality in the CMS software.

One of the main advantages of the NANOAOD data tier is that it does not require any special software other than ROOT to access its contents. This makes it possible to perform data analysis without any dependencies on CMS software, and even perform the analysis with

tools that are not traditionally used in HEP. This flexibility lowers the barrier of entry for new students and allows to make CMS data publicly available for analysis by people who are not members of the CMS collaboration.

Every centrally produced NANOAOD Ntuple includes provenance information, which is a recorded summary of the history of the full production chain. It is possible to trace back the exact conditions that were used to reconstruct the event, thereby improving the integrity of the data compared to custom Ntuples. This, combined with the other properties of the format such as simple and standardized access to its event contents, creates necessary prerequisites for a reproducable analysis.

The computation of redundant information, such as systematic uncertainties that are excluded from the NANOAOD data tier in order to keep file sizes as small as possible, is based on standard algorithms in many cases. For this reason, a companion tool has been developed [5], which allows users to recompute this information easily. Additional information that the companion tool can compute includes: systematic variations of muon energy, jet energy and jet resolution, event weights for pileup reweighting, and corrective "scale factors" for various *b*-tagging discriminants. The tool also provides an easy-to-use interface that allows to perform object-level skimming and event selection, and to compute new object- and event-level variables. The companion tool is integrated into the CMS-based software stack, but can be used as standalone, without further CMS software. Many analysis groups have forked the companion tool in order to customize it for their analysis. Tracking the number of forks of this tool can give some insight about the adoption rate of NANOAOD, which as seen from Figure 2 has progressed at a relatively constant rate since the introduction of NANOAOD.

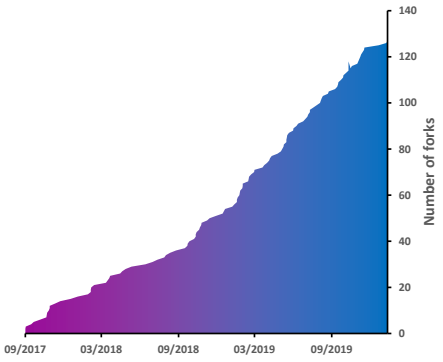

Figure 2. Number of forks of the NANOAOD companion tool as a function of time since its inception.

4 Future prospects and conclusions

The CMS data centers can benefit from the small size of the NANOAOD format only if its adoption reaches a high enough level. Such level allows to reduce the number of copies of MINIAOD files on disk, and to free up CPU processing power that was previously consumed by custom Ntuple production. The focus is expected to shift towards improving the companion tool once the NANOAOD format becomes widely adopted. The frequency of the centrally managed conversion campaigns from MINIAOD to NANOAOD format is foreseen to increase in the future.

Thanks to its high customizability, the NANOAOD format has also found some application in object calibration studies. One such example is the jet energy calibration, which was previously based on custom Ntuples. The workflow for the jet energy calibration will be based on NANOAOD: a set of custom NANOAOD Ntuples are automatically produced from MINIAOD files which, upon finishing, triggers the automatic execution of the software that performs the actual jet energy calibration. Object calibration workflows may become automated based on NANOAOD format in the future, providing more prompt feedback on the quality of the data than previously possible.

The NANOAOD format could be modified in order to accept the direct output of Monte Carlo event generators. Ntuples produced from such information would be useful in technical validation of the event generators. In this application, the validation would automatically start before any other following step in the full production chain, thus bypassing time-consuming detector simulation and event reconstruction steps. These changes would therefore facilitate the quality monitoring of simulated events.

Currently, information specific to Monte Carlo generators and to parton distribution functions are stored as event weights in the NANOAOD Ntuples. At present, it is not possible to store alternative choices of said information in the NANOAOD data tier without increasing the file sizes substantially. Improving on that limitation is another avenue for improvement in the future.

In conclusion, a new compact data format, NANOAOD, has been recently commissioned by the CMS collaboration. The format is designed to alleviate the increasing demand on computing resources, and to simplify the approach to physics analyses. The NANOAOD format has also found to be useful in specialized tasks such as the automation of object calibration frameworks. We anticipate growing interest and usage of the format as we move towards LHC Run 3.

References

[1] CMS Collaboration, The CMS experiment at the CERN LHC, JINST **3**, S08004 (2008)

[2] R. Brun, F. Rademakers, ROOT - an object-oriented data analysis framework, Nucl. Inst. & Meth. in Phys. Res. A **389**, 81-86 (1997). See also https://root.cern.ch/.

[3] A. Rizzi, G. Petrucciani, M. Peruzzi, A further reduction in CMS event data for analysis: the NANOAOD format, EPJ WOC **214**, 06021 (2019)

[4] G. Petrucciani, A. Rizzi, C. Vuosalo, MINIAOD: a new analysis data format for CMS, JPCS **664**, 072052 (2015)

[5] CMS Collaboration, Tools for working with NANOAOD, website, https://github.com/cms-nanoAOD/nanoAOD-tools