Experience with NANOAOD

Andrea Rizzi (CMS Collaboration)

June 1st, Workshop on Analysis Tools

Istituto Nazionale di Fisica Nucleare

UNIVERSITÀ DI PISA

Outline

- Reminder: what are NANOAOD
- Current experience with it
  - Analysis coverage
  - Production process
  - New features integration
- The road ahead

A brief history

- In Run1 CMS central production was finishing with AOD
  - A set of tools (PAT: Physics Analysis Tools) was provided for object calibration or to rerun some high level algorithms
- For Run2 CMS developed MINIAOD format
  - Centrally running relevant PAT algorithms
  - Reducing size per event by 1 order of magnitude retaining large flexibility
  - MINIAOD now reached > 90% analysis coverage (and growing)
  - Lightweight ntuples created from MINIAOD by individual groups
- At the end of Run2 a common ntuple like format (NANOAOD) has been proposed in order to test it and possibly widely adopt it for Run3
  - Size reduced to ~1-2kb/event
  - Expect initial coverage of 30-50% of analyses
  - Retaining flexibility for many analyses choices

Analysis Data formats in CMS today

RAW: Full event information directly from T0 containing "raw" detector info, not used for Analysis

RECO: reconstructed data; contains physics objects with many details stored [hits, etc..] , Mainly for low level developments

AOD(**A**nalysis **O**bject **D**ata): a subset of RECO data tier. Used for physics analyses in Run1, Run 2: Used for searches with non-standard signatures e.g., displaced objects

miniAOD: default datatier for the Run2 analyses
"EDM object type" format , can be processed by CMS fwk

nanoAOD: light weight data tier introduced in 2017
"fundamental type and arrays thereof" format, can be read from bare root or even python tools

Analysis Data formats in CMS (2)

miniAOD: default datatier for the Run2 analyses,

1. *"EDM object type"* format

I.e. `std::vector<pat::Muon>`

2. Full information to allow developments

C++ object

nanoAOD: light weight data tier introduced in 2017

1. *"fundamental type and arrays thereof"* format,

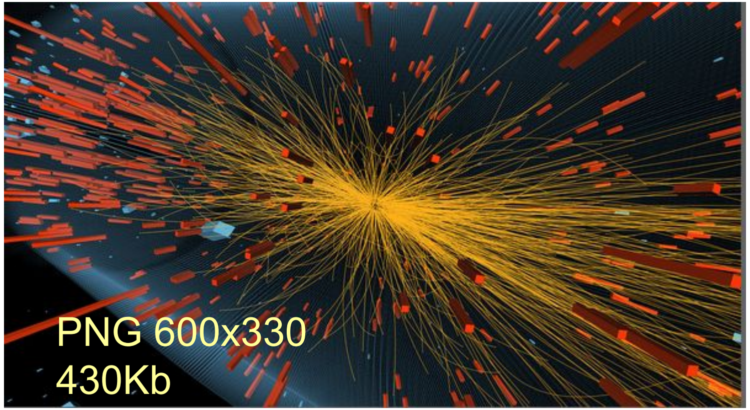
```
Int_t nMuons;
Float_t Muon_pt[nMuons];
Float_t Muon_eta[nMuons];
```

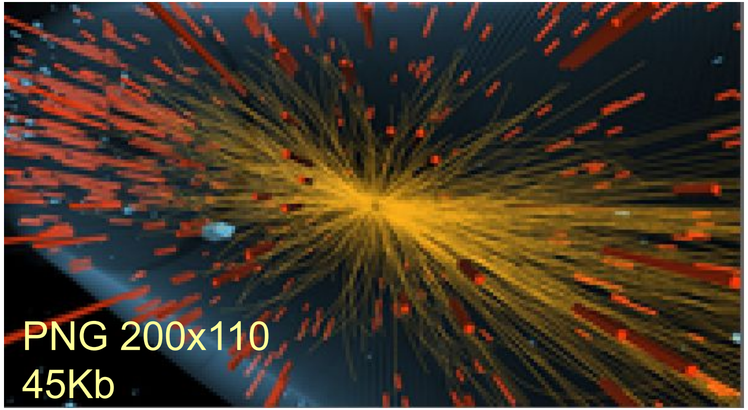
2. Store high level physics objects with precomputed ID/variables subset of generated particle and LHE weights, trigger bits , with reduced precision when needed
3. drop particle flow candidates and tracks, most detector level information

AOD vs MINIAOD vs NANOAOD in a picture

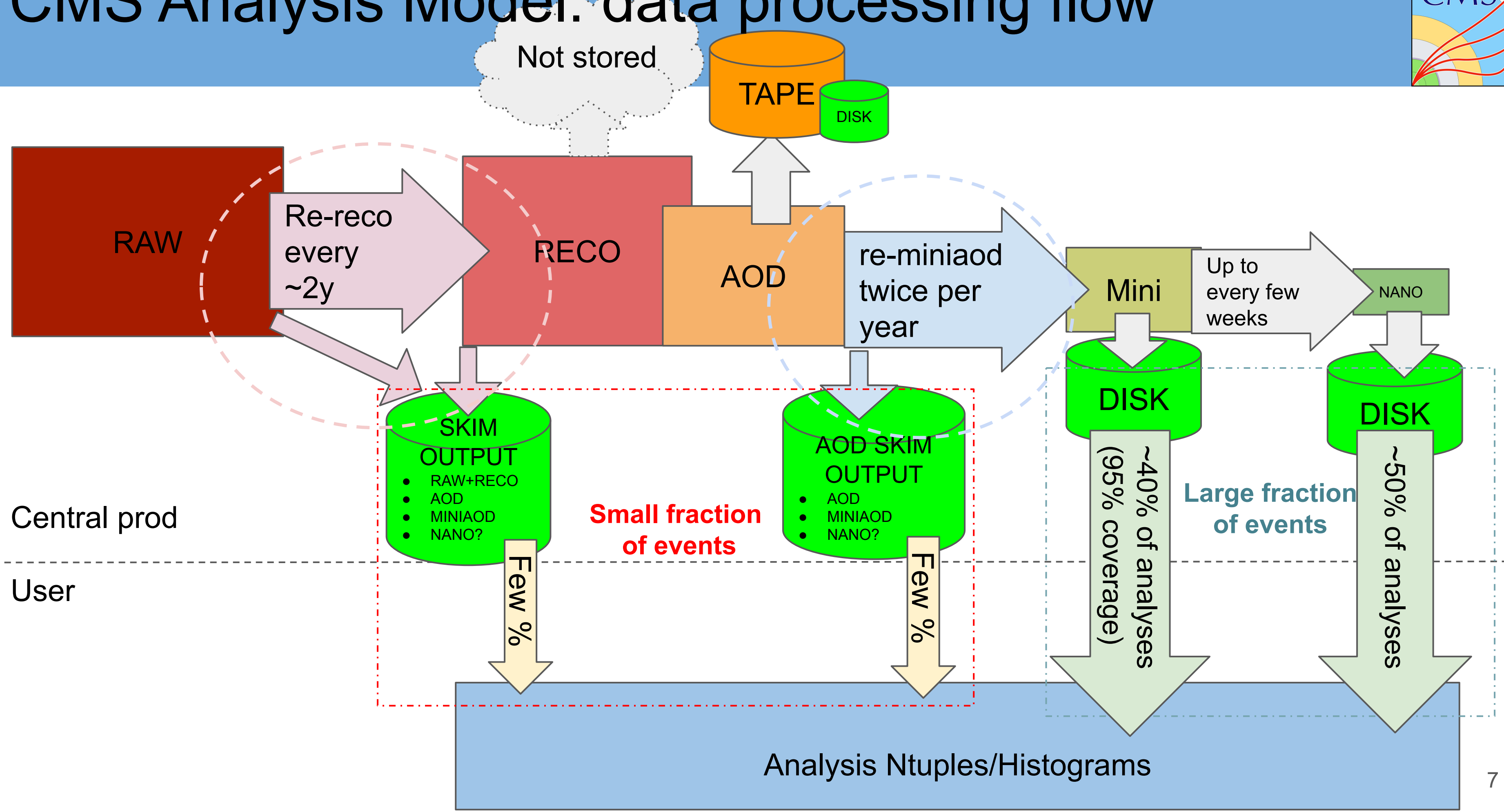
How would an event display (in PNG format) of the experiment would like using the per event budget of:

- AOD
- MINIAOD
- NANOAOD





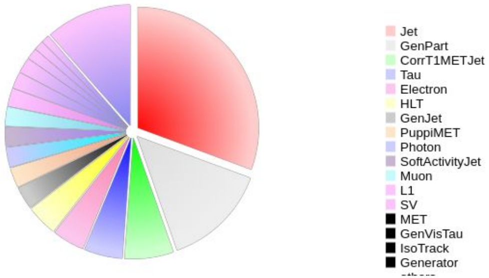
CMS Analysis Model: data processing flow



The content of NANOAOD

- Cannot afford/do not need “all tracks”
  - But some track details for leptons
- Jet collection is the biggest
  - Cannot afford storing ~50 systematic variations of jet energy (but we can recompute them on the fly!)
- Gen level description reduced to the minimum
  - “Important” particles
  - Matrix Element level initial and final state information
  - Flavour information
- Additional information specific for few analyses can be added too as long as it is cheap
  - $\frac{\text{Additional Size}}{\text{number of analyses}} <$ few bytes

mc106Xul18_NANO.root (13.569 Mb, 9000 events, 1.54 kb/event)



Event data

collection	kind	vars	items/evt	kb/evt	b/item	plot	%	cumulative %	
Jet	collection	51	7.35	0.352	49.1		30.6%	30.6%	100.0%
GenPart	collection	9	28.02	0.161	5.9		14.0%	44.6%	69.4%
CorrT1METJet	collection	6	8.89	0.075	8.6		6.5%	51.1%	55.4%
Tau	collection	48	0.67	0.059	89.7		5.1%	56.3%	48.9%
Electron	collection	64	0.33	0.049	154.3		4.3%	60.6%	43.7%
HLT	singleton	685	1.00	0.048	49.3		4.2%	64.7%	39.4%
GenJet	collection	7	2.92	0.035	12.2		3.0%	67.8%	35.3%
PuppiMET	singleton	15	1.00	0.031	31.2		2.7%	70.4%	32.2%
Photon	collection	31	0.50	0.030	62.4		2.6%	73.0%	29.6%
SoftActivityJet	collection	4	5.72	0.030	5.4		2.6%	75.7%	27.0%
Muon	collection	57	0.21	0.029	144.5		2.5%	78.2%	24.3%
L1	singleton	344	1.00	0.027	28.0		2.4%	80.6%	21.8%
SV	collection	16	0.63	0.025	40.9		2.2%	82.7%	19.4%
MET	singleton	12	1.00	0.023	23.9		2.0%	84.8%	17.3%
GenVisTau	collection	8	1.64	0.017	10.5		1.5%	86.2%	15.2%
IsoTrack	collection	15	0.42	0.014	34.9		1.2%	87.5%	13.8%
Generator	singleton	9	1.00	0.012	12.6		1.1%	88.5%	12.5%
PV	singleton	8	1.00	0.011	11.7		1.0%	89.5%	11.5%

Some additional NANOAOD features

- Features
  - No cross cleaning is applied (because each analysis needs different criteria)
  - But cross-linking done (using “shared” PF constituents)
    - DeltaR matching can be performed a posteriori
  - Linking from collections as simple as using indices (e.g. Jet_pt[Muon_jetIdx])
- Set of (non mandatory) tools to further process the format: NanoAOD-Tools
  - Fast(ish) and efficient skimming or friend-trees creation
  - Pluggable modules to add JEC uncertainties, jet smearing, btag uncertainties
  - Lepton scale factors etc..
- In-file documentation

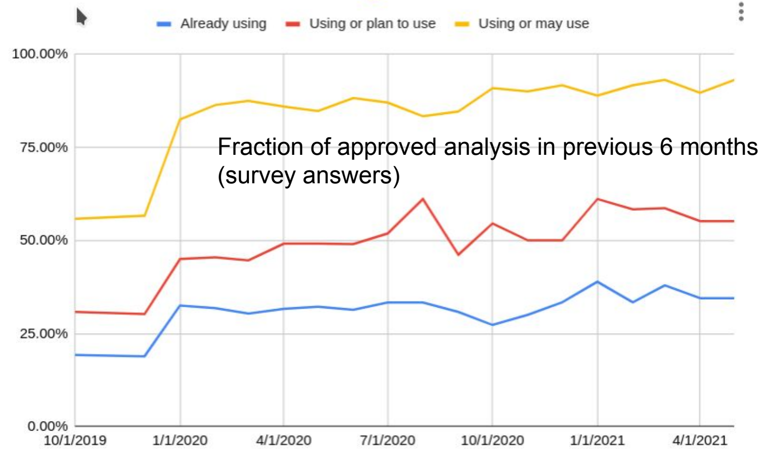
Muon_genPartIdx	Int_t(index to Genpart)	Index into genParticle list for MC matching to statu
Muon_highPtId	UChar_t	high-pT cut-based ID (1 = tracker high pT, 2 = globa
Muon_ip3d	Float_t	3D impact parameter wrt first PV, in cm
Muon_isPFcand	Bool_t	muon is PF candidate
Muon_jetIdx	Int_t(index to Jet)	index of the associated jet (-1 if none)
Muon_mass	Float_t	mass
Muon_mediumId	Bool_t	cut-based ID, medium WP

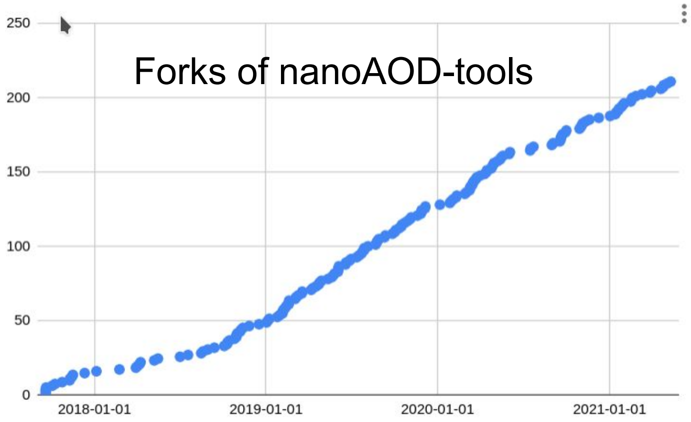
Experience with NANOAOD

- NANOAOD was introduced at the end of Run2, hence:
  - Most analysis efforts were already ongoing with some fwk/ntuples already setup (no reasons to switch)
  - Initial central production campaigns were more for testing than actual usage
  - Initial content had some simple to fix missing content for some analyses (but production was not foreseen too often)
- Some people adopted the format with a few private changes/adaptations:
  - Expected in the analysis model at least for limited (signal) samples
  - Still useful to start from a common baseline with 99% shared content with other analyses
  - ...eventually contribute the additions to the central common format
- A survey is performed with a google form filled each time an analysis approaches physics approval

Adoption of NANOAOD

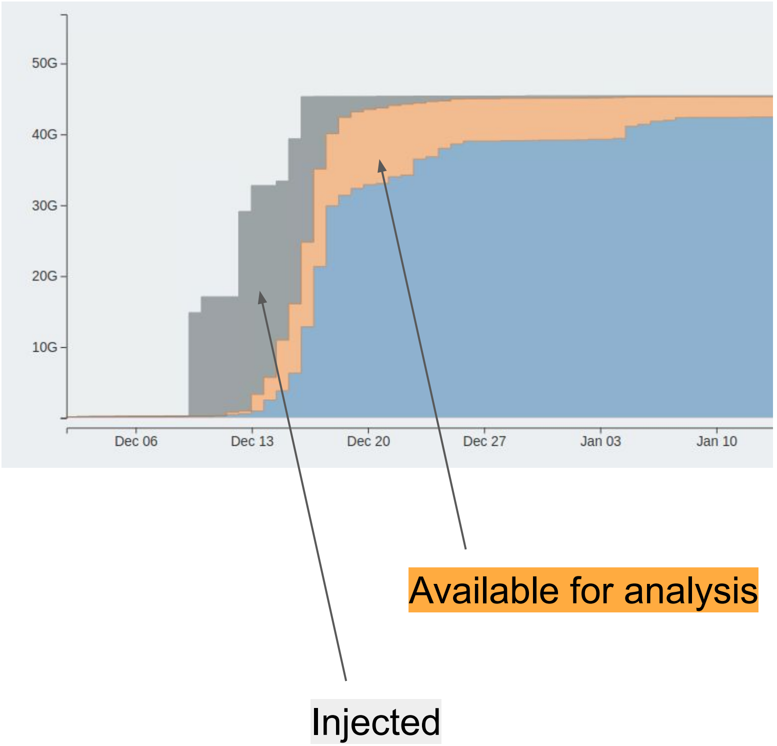
- More than 30% of the analyses in the past year used nanoaod
  - In some cases multiple groups doing xcheck analyses used different formats
  - Some slightly modifying the content to adapt to their needs
- Another ~20% are willing to switch to nanoaod as soon as they are done with Run2 analyses
  - And counting also those that said they may use nanoaod for Run3 (i.e. they see no showstopper) the total figure goes >80%
- As an index of how many people are looking into nanoaod we can check the number of forks of nanoaod-tools (more or less needed for analyses willing to use this tools)
  - Steady increase since beginning of 2019





Production time

- NANOAOD is a dynamic format
  - People are expected to add their newly needed observable
  - New algorithms (e.g. btag) or new calibrations can be made available
  - Bug fixes!
- NANOAOD are produced “often”
  - CPU time is not an issue (~10-20 Hz per core)
  - Handling of >10k datasets, ~50B events is now possible in ~1-2weeks
  - Currently producing NANOAOD multiple times every few months
  - Frequency for scheduled NANOAOD production expected to increase for Run3
  - Possibly implement a more “on demand” policy
- Time for developing new feature, validation, deployment of sw release
  - Automatic tools for CI
  - Documentation and tracking of features
  - => dedicated CMS group following this



Experience gained with end of Run2 analysis

- Event weights (e.g. for PDF) are a pain
  - Way too many (only few analyses properly using them in the end)
  - Changing representations in generator headers (even between when only changing version minor number)=> bug for a particular sample often discovered very late
- Too many people would like to use NANOAOD, even those that cannot really make it with a few kb/ev format
  - Should probably use MINIAOD
  - Main issue being “all tracks” (or better “all particle flow candidates”)
    - An example is high pT “fat jets”
    - Some reasonable subsets are being explored, but it is not yet clear if the trade-off could be satisfactory
  - Many ML applications want “more raw features”
    - Train on MINIAOD
    - Save output on NANOAOD

Experience gained with end of Run2 analysis

- Nanoaod-tools is not so satisfactory
  - It is python event loop based (beside the pre-skimming based on TTree::Draw like syntax)
  - Coffea and RDataFrame can go much faster than that
    - Need C++ (or other way accelerated) modules to compute uncertainties and calibration corrections
  - It is ok to use nanoaod-tools each time a significant skimming is needed
    - In fact, most of the cases
- Possibility to create local “analysis facilities” as full datasets in nanoaod format (possibly skimmed) can fit a single SSD few TB drive
  - Local clusters
  - Single multicore machines

Customization

- Additional customization developed for specific purposes
  - Calibration workflows (e.g. for Jet Energy calibration)
    - Larger “per event” size
    - Limited number of samples
  - Special analysis groups
    - When additional content is not small enough to be fit in the general purpose format
    - Some skimming?
    - Limited number of samples?
    - Privately produced?
- The “common base” is still useful in order to implement full analysis quality selection / physics object reconstruction into calibration workflows
- Other nano-like formats?
  - B-Physics has completely different needs (tracks, refitting, particle ID, secondary vertices, soft pions, etc…)
    - A possibility could be to investigate a common format with similar footprint of nanoaod

Conclusions

- NANOAOD format seems to work as planned
- Adoption for and of Run2 in line with expectations
- Plans for Run3
    - Increase coverage of analyses by extending nanoaod content
    - Increase frequency to make easy additions of newly developed observables
    - Improve automatization of validation workflow / preparation of new nanoaod releases