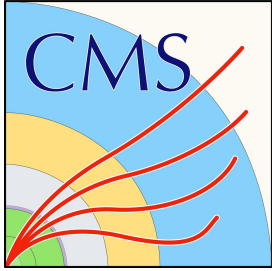


HL-LHC analysis mini-workshop: inputs from CMS

Mariarosaria D'Alfonso
Massachusetts Institute of Technology

On behalf of CMS collaboration

Challenges of HL-LHC analyses

Integrated luminosity L = 160 fb^{-1} in Run 2; expected to reach L > 3000 fb^{-1} during High-Luminosity LHC (HL-LHC)

New physics opportunities ahead with analysis challenges:

- Higher pile-up (Run5 ~200)
- Higher trigger rates
  - Record tailored signatures, going into the tails
- More MC simulated events to match the data luminosity
- More unconventional signatures
- More precisions physics
  - Need the calibration constants follows demands
  - More parametrizations ⟶ improved/flexible storage

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

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

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; drop particle flow candidates and tracks, detector level informations

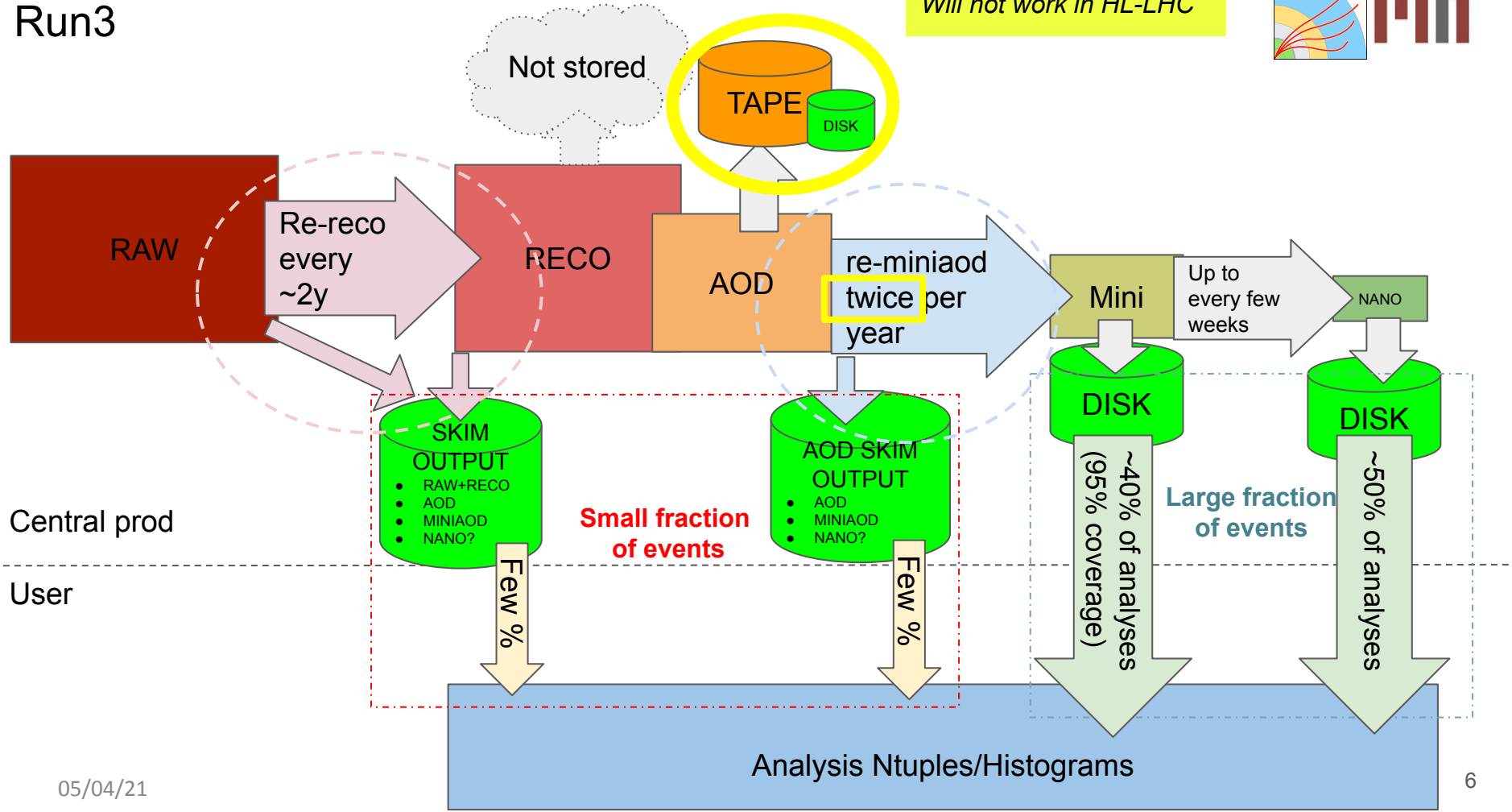
Analysis Data formats in CMS (3)

mini/nano

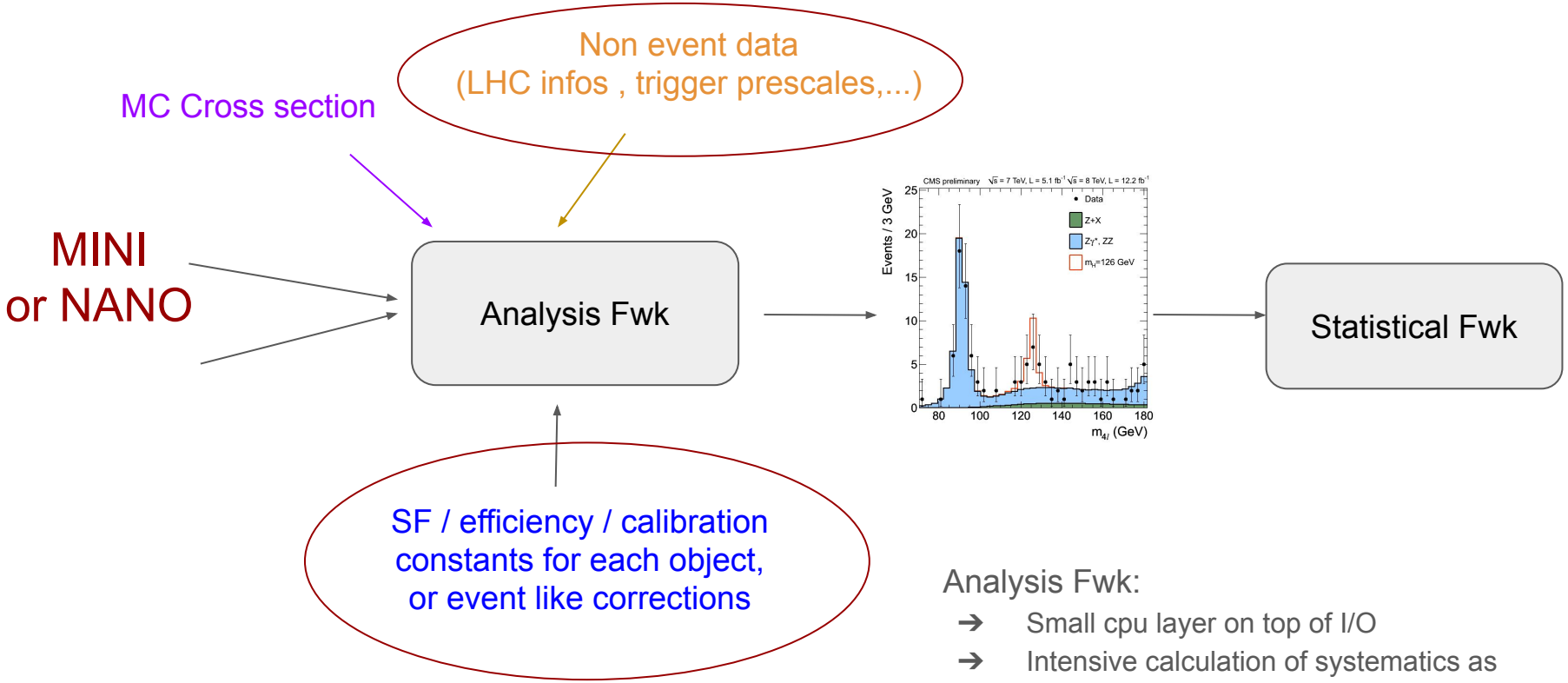
- Run2 small event data formats for analysis.
  - miniAODSIM (~70 kB/evt) - covers 90% of the Run2 analyses needs
    - miniAOD being adopted also for the Heavy Ion and Flavor-physics analyses
  - nanoAODSIM (~2 kB/evt) - progressively adopted in Run2 analysis → aim for 50% in Run3
- Different CPU requirement to produce.
  - miniAOD (~ 750 ms)
  - nanoAOD from miniAOD 10 Hz on one CPU core

Mini and Nano will be the focus for Phase2

Run3


Will not work in HL-LHC
Not stored
TAPE
DISK
RAW
Re-reco every ~2y
RECO
AOD
re-miniaod twice per year
Mini
Up to every few weeks
NANO
DISK
DISK
SKIM OUTPUT
RAW+RECO
AOD
MINIAOD
NANO?
AOD SKIM OUTPUT
AOD
MINIAOD
NANO?
Small fraction of events
Large fraction of events
~40% of analyses (95% coverage)
~50% of analyses
Few %
Few %
Central prod
User
Analysis Ntuples/Histograms

Analysis ingredients



Analysis Fwk:

- ➔ Small cpu layer on top of I/O
- ➔ Intensive calculation of systematics as input of the fit

Volume of input data

- Number of events expected to be analysed
  - Studies before data taking, each year: 4B Monte Carlo events (commissioning/trigger menu)
  - Physics analysis, each year: 5B Monte Carlo events + 0.2 B per fb^{-1} of luminosity collected
- Number of distinct samples (data, MC) analyzed per average analyses
  - One Primary Dataset , few dominant background used often for optimization, more dataset for final analysis
  - Initial subset of MC then full MC production/analysis with “extended” statistics (i.e. x5)
- Latency of the analysis respect to the data taking and various reconstructions:
  - Keep flexibility with calibration precision need: Prompt vs EOY vs legacy processing
  - 2016 completed with EOY, 2017-2018 bulk of the analyses after the data taking was completed
- Frequency of the production:
  - miniAOD remade once a year,
  - nanoAOD: Fast production (~ 2 weeks) , every 3-4 months or on demand

Accessibility of the datasets

Desiderata:

- Flexibility: Remote running (xrootD remote reads) vs local mass storage (eos) vs cache (content aware) vs local disk should be optimized
- nanoAOD (de)compression algorithm important to reduce the resources required to store and transmit data
- Smaller nanoAOD size help to increase the number of replica on disk
- Efficient tools for “skimming” needed at user level significant i.e. when analyzing high level object multiplicity events
- Possibility of “pruning” and “extending” dataset:
  - i.e. allow production of private custom nano with central code with mixed input (nano code + extra Input)

Analysis fwk

Variety of computing languages:

Root , C++ and python ecosystems

Variety of analysis type:

loop analysis : Load relevant values for a specific event

columnar analysis: Load relevant values for many events into contiguous array

Parse object correction files and provide the correction function casted in terms of loop/columnar operations

Wishlist for SW:

- ❏ Minimize the data conversion
- ❏ Process many events simultaneously
- ❏ Minimizes disk space
- ❏ Efficient memory access
- ❏ Communication and collaboration between teams providing analysis packages and interoperability between the products provided

Analysis facilities

Plans for several CMS **analysis facilities** with services, software, hardware for analysis and dedicated support team

- Reliable platform to plug in technologies and enable efficient analysis
- Services:
  - Access to experimental data products
  - Storage space for per-group or per-user data (often ntuples)
  - User support
- Physics software: ROOT and the growing Python-based ecosystem
- Computing hardware: available/new CPUs and disks (maybe GPUs)

Key features:

→ access the nanoAOD samples with low latency

→ scheduling computation with max efficiency (including options beyond batch jobs+merging output)

→ Aiming at a common repository of code for routines/workflows

Variety of user cases:

→ derivation of corrections/calibrations → next slides

→ efficient training of the ML-based objects ID → next slides

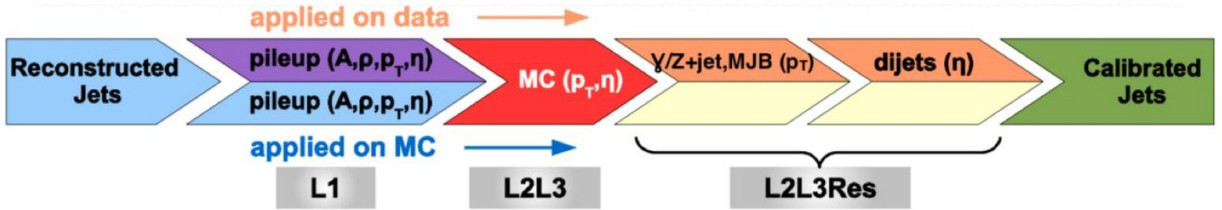
→ user end analysis

Calibration workflow analysis

Physics object calibration repeated often (once for each year, more times during the datataking).

SW requirements:

1. Code change are usually very small
2. Computations sw need to encapsulate the calibration step dependency
3. Require stable computations environments
4. Automated and reproducible



ML trainings (1)

Advanced ML techniques become standard tools in CMS analyses

- The majority of those focusing on object identification
  - Current state-of-the-art developments utilize low-level info: e.g., Particle Flow Candidates directly
- Lots more of potential: for current and foreseen applications
  - E.g., condensate the granular informations from the improved detectors

→ Let's pick jet classification as example; A typical workflow:

- Training dataset: flat root ntuples starting from miniAOD or privately produced nanoAOD
  - 100M of jets of various types/flavour
- Algorithm design and network architecture:
  - Based solely on PF candidates and Secondary vertices (low-level)
  - Network: Graph Neural Networks
- Training details: ~5 days on 4 GPUs
- Inference: Use of ONNXRuntime -> 30 msec / jet on CPUs
  - General idea: Include computationally expensive tasks in mini/nanoAOD so analyzers just read a branch

ML trainings (2)

- ML-based tools show significant improvement wrt more traditional techniques.
  - The trend is to exploit their potential in more application.
- Improvements in many areas beyond the algorithm design are needed to meet HL-LHC requirements
  - Improved inference time: currently >25% of the miniAOD processing time is used for ML-based jet tagging algorithms
    - Access to GPU/FPGAs; expect > O(10) speed-up
  - Improved flexibility & simplify integration: ML inference as a service [techniques already used in industry]
    - Efficient sharing of resources: CPUs, GPUs
    - Transparent to the analyzer [i.e., no need to translate algorithms in coprocessor-specific languages

Complexity in analysis

Likelihood function needed to extract the results, usually huge as all the variations are nuisance parameters

→ reduced version with template/binned

Computing analysis challenges are in the bookkeeping of templates for systematics variation of uncertainty weights for both background and signal

→ simple case of pT and eta variation of one object
→ theoretical inputs i.e/ NNPDF3.1 has ~ 103 input variation, anomalopus coupling O(~ 1500) inputs
→ Transforming the data : take binning info for each dimension

Fitting strategy for HL-LHC might evolve into unbinned fits and/or improved minimization methods

→ new challenges for SW

Final remarks

CMS working towards the physics and technical analysis challenges of HL-LHC:

- continuously optimizing the dataformats and datasets definition already for the Run2/Run3.
- maintain the flexibility to meet different goals
- Identifying sw desiderata such as *data accessibility, minimize disk space, process many events simultaneously, efficient memory access , easy and intuitive programming model, ...*

Essential to strengthen

- communication and collaboration between teams providing analysis packages and interoperability between the products provided
- active engagement within experiments' communities