Stato e prospettive del computing e tecniche di ricostruzione e analisi ad alta luminosita

Andrea Rizzi, Tommaso Boccali Uni/INFN Pisa

CMS Italia, Piacenza/Pavia, 1 Dicembre 2017

Outline

- Stato del computing 2017
- Prospettive per i prossimi anni
- Evoluzione del modello di analisi: MINIAOD → NANOAOD
- Nuove tecniche di analisi: Deep Learning in CMSSW

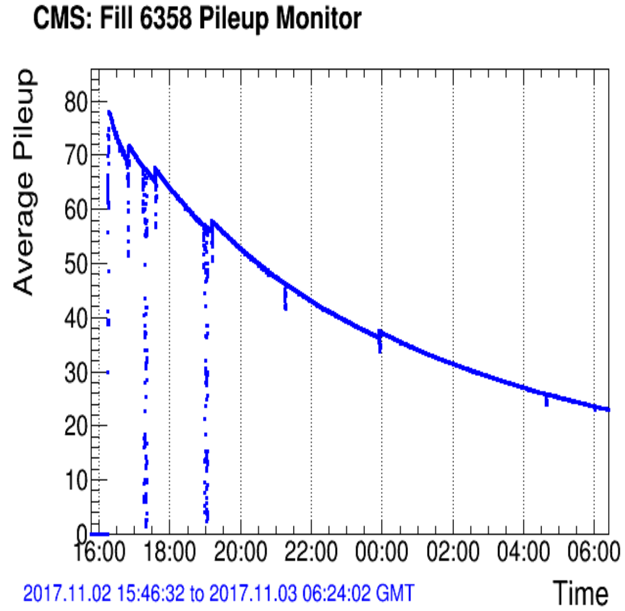
2017 – anno complicato …

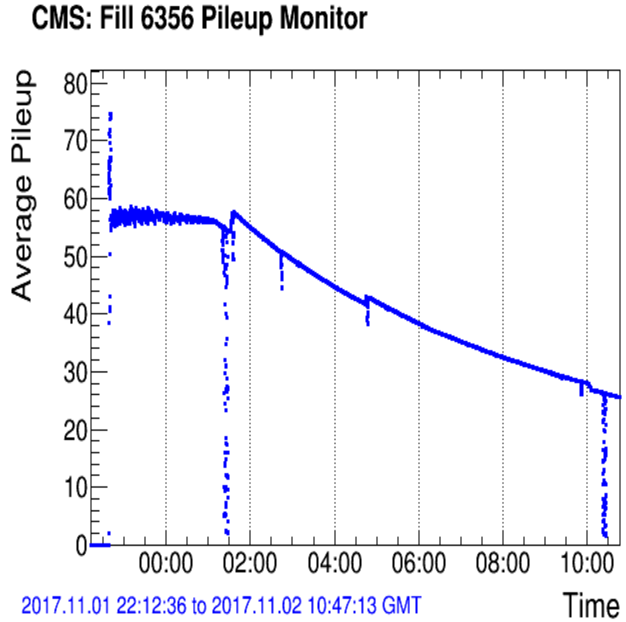
- Come doveva essere:
  - Risorse chieste per:
    - 1.7e34 su 2500-2800 bunches
    - <PU> = 35
    - HLT rate a 1 kHz
    - ~7.5Msec di stable beams; sostanzialmente costante durante l'anno
  - Produzione MC da giugno, nessun reprocessing fino alla fine dell'anno

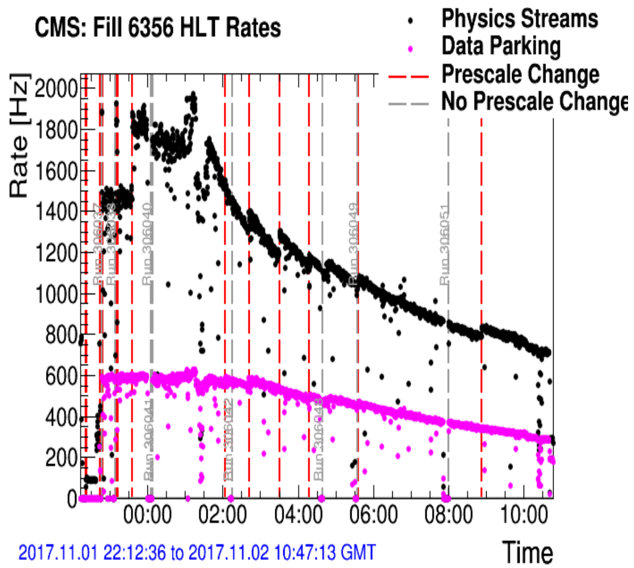
- Come e' stato:
  - Risorse ridotte del 20% (ancora effetto del cambio di parametri di LHC nel 2016, troppo tardi per avere effetto sulle risorse)
  - <PU> = 45 dopo il 20 settembre (e anche peggio per il PU effettivo)
  - ~7.2Msec di presa dati
  - Inoltre, luminosita' / efficienza di LHC sbilanciata negli ultimi 2 mesi
    - Fino a settembre poco o nulla
    - Da settembre una valanga di dati
  - HLT a > 1.5 kHz
  - Produzione MC parte in pratica ORA (a parte una piccola V1)
    - Commissioning dei pixel principale problema, ma non solo
  - Follie di fine anno: ppRef a 40 kHz di HLT rate ….

Presa dati – dal 20 settembre in poi

- In prtaica siamo almeno a RunIII…
  - 2.5e34 → abbiamo girato stabilmente a 1.5e34 su 1800 bunches. La lumi per bunch (e il PU) sono quelli di 2.3e34
  - PU a inizio fill 80, livellato a 55
  - PU medio durante il fill ~ 45
  - PU effettivo (pesato sul trigger rate) superiore a 50, visto il rate di HLT durante levelling





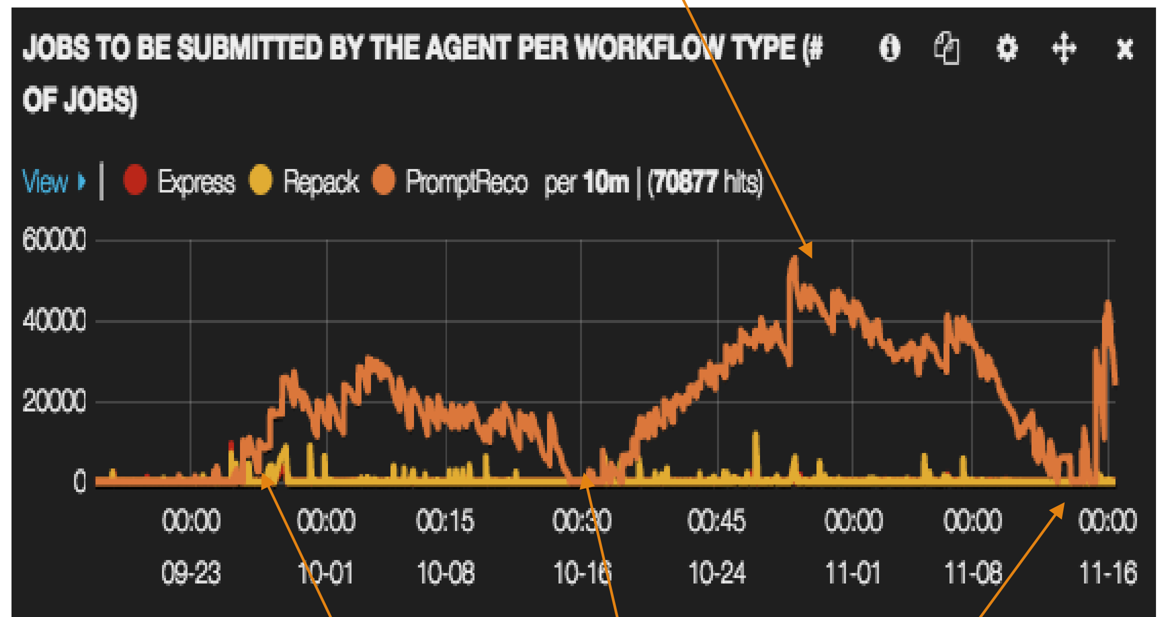


Conseguenze? Al Tier-0 ….

- PU 45 vs PU 35 e’ almeno il 50% in piu’ di sec/ev + almeno 30% in piu’ di rate medio rispetto a 1 kHz + availability di LHC > 60%
- → Tier0 non riesce a digerire in tempo il processing Prompt, e diverge
  - Prompt delay da 2 gg puo’ arrivare a molto di piu’
  - Ma se cosi’ succede, I buffer del Tier-0 si riempiono e non si ricevono piu’ dati da P5
    - CMS si ferma
- Non e’ successo!
  - Usato T2 del CERN come supplemento al Tier-0
  - Diminuiti eventi in formato RECO

  - **→ sistema ha retto l’urto dei dati; importante soprattutto se assumiamo che 2018 sara’ cosi’**

T2_CH_CERN in the game



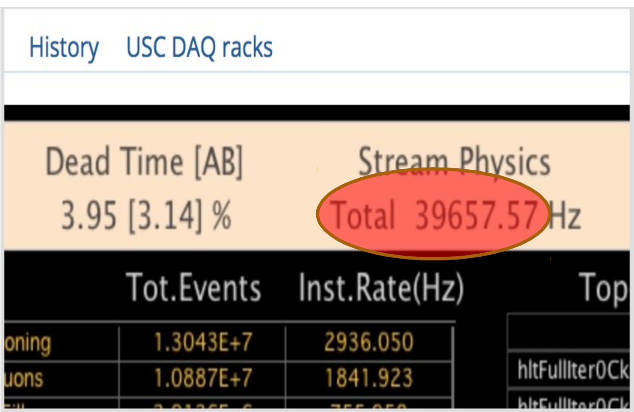
Levelling start XeXe Run ppRef Ru

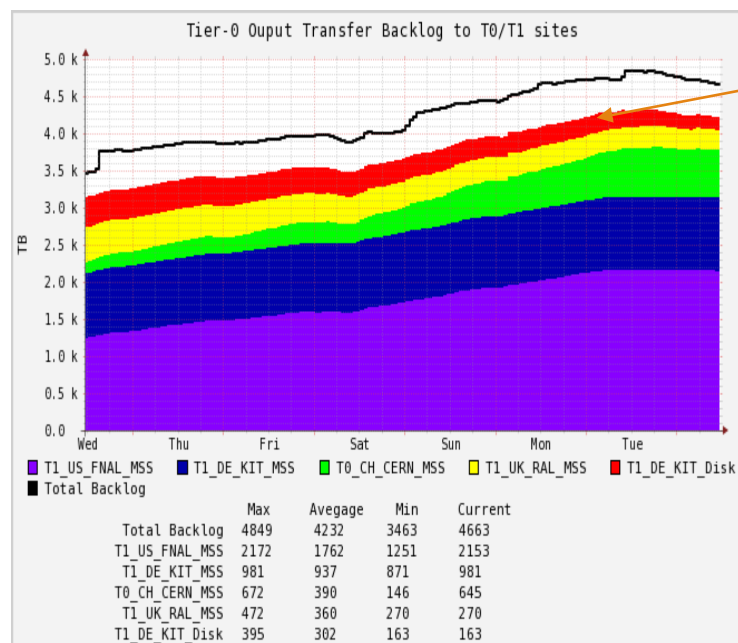
di Tier-0- jobs in ritardo (“backlog”)

Divertimenti di fine anno…

- ppRef run:
  - PU ~ 3, eventi a 180 kB l'uno
    - 1/5 di pp(PU=35)
  - Quindi 5 kHz e' equivalente al design Tier-0 (1 kHz @ PU35); invece ….

- Stress estremo di storage manager e Tier-0; sistema chiaramente non convergente
- Run possibile solo perche' e' durato 7 gg
- … nei quali abbiamo accumulato 5 PB di bakclog di trasferimento verso I Tier-1

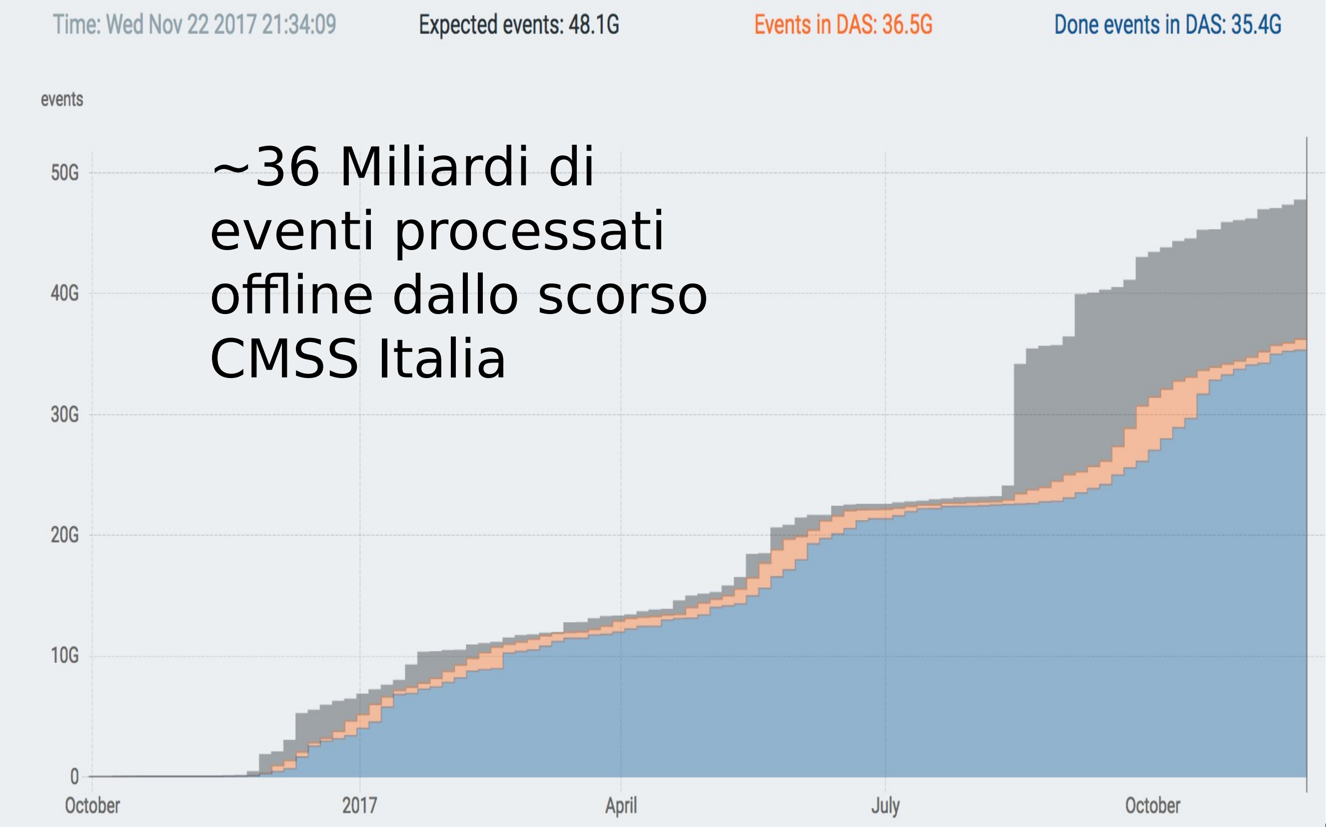




End of ppRef

Detto questo …

- I sistemi di Computing e Offline hanno retto, con un tetris molto complicato di attivita'
- Lista veloce:
  - **PhaseII TDR** - prodotti in totale 500M di eventi, fino a PU=200



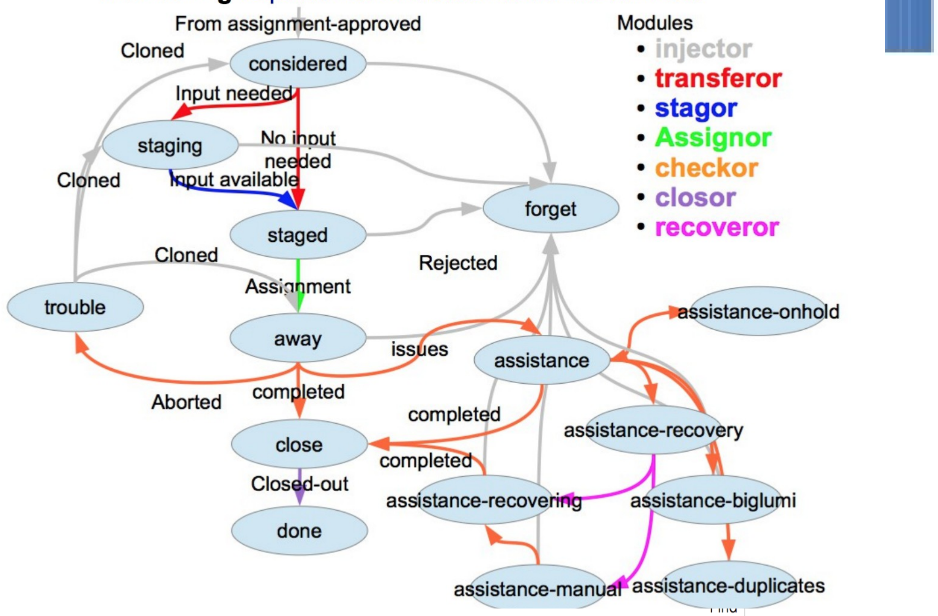
~36 Miliardi di eventi processati offline dallo scorso CMSS Italia

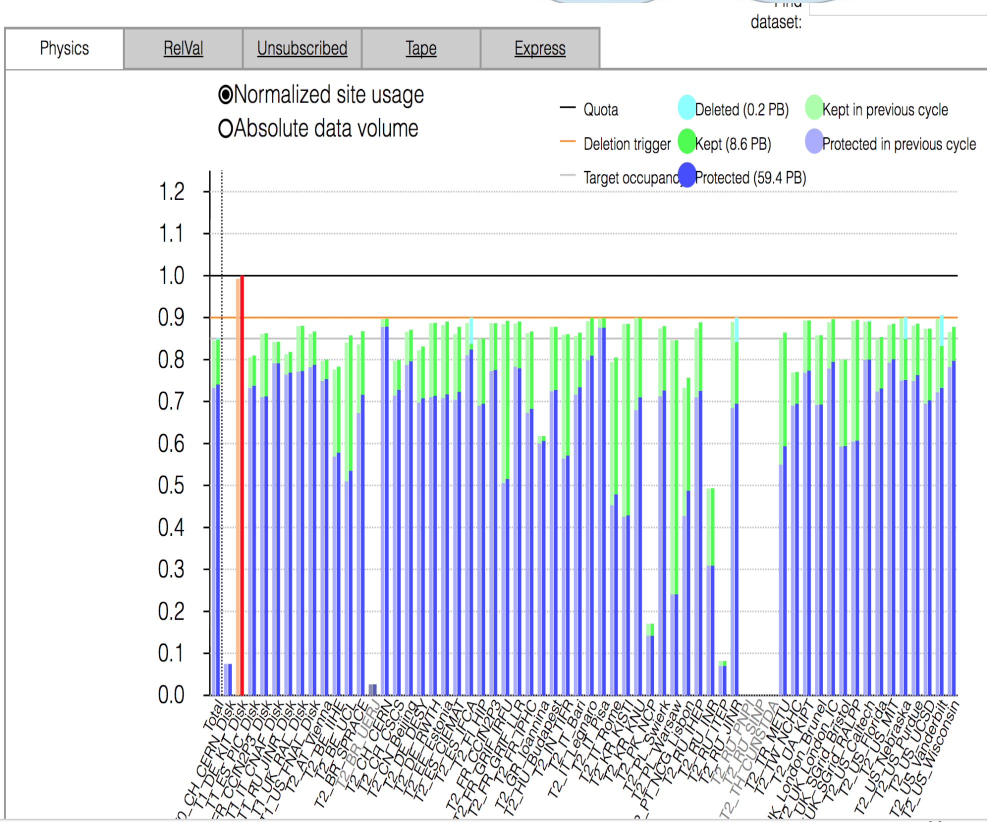
- **MC2016**:
  - per le analisi di Moriond17, 15B di eventi
- **Legacy Rereco DT2016**: fatta 2 volte
  - Totale 13B di eventi
- **PhaseI*:** preparazione run 2017
  - 3B di eventi
- **MC2017v1**:
  - 2B di eventi
- **ReReco 2017**:
  - Parziale di 2B si eventi fatta
  - Iniziata quella totale (7B di eventi)
- **MC2017V2**:
  - 10B di eventi entro Winter Conf

  - (me ne saro' dimenticati molti ….)

Unified State Machine

Monitoring https://cmst2.web.cern.ch/cmst2/unified/





Alcuni highlights

- Evolution of Computing Model 2017 (ECom17):
  - Chairs T. Boccali, F. Wuerthwein
  - 20 raccomandazioni da qui fino a RunIII, fra cui
    1. Definizione di ~~Micro~~NanoAOD
    2. Lifetime model
    3. Definizione delle priorita' in R&D per RunIV
    4. Studi generatori Sherpa vs Madgraph
    5. Ottimizzazione di workflow (StepChain, TaskChains, …)
    6. …

- Gestione ormai automatizzata:
- di workflow: **Unified** gestisce in autonomia campagne di processing (dallo stagein dei dati alla finalizzazione)
- Dello storage: **Dynamo** gestisce lo storage, I trasferimenti, le cancellazioni

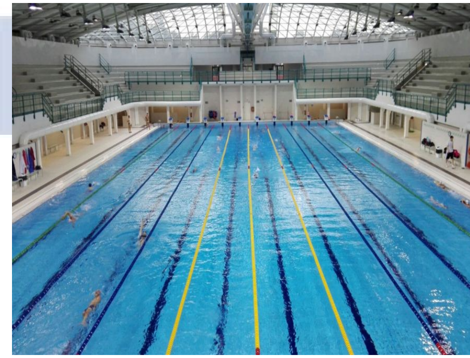
Fine di RunII - e oltre…

- 2018:
  - Risorse richieste per parametri Pre-Settembre 2017 di LHC
    - Non sara' banale sopravvivere se PU=45 e' confermato, ma possiamo
      - CPU: usare calcolo distribuito per aiutare Tier-0
      - Tape: se alle strette, smettere parking
      - Disco: bisogna essere estremamente parchi, per esempio ridurre al minimo la generazione di RECO
  - 2019:
    - Piano attuale e' di effettuare un reprocessing legacy di tutto il RunII (25B dati e 35B simulazioni)
    - Richieste "preapprovate" a ottobre, aumenti limitati dall'uitlizzo del Tier-0 e dell'HLT come risorse offline

- RunIII:
  - Non chiaro cosa sara' RunIII, sulla carta al modello molto simile a 2018 + 1 TeV
  - Aumenti al max flat budget, se non meno (~ 2x da qui al 2022)

- RunIV:
  - Ancora modelli molto approssimativi (almeno per un altro paio di anni)
  - Pero' introduzione di MiniAOD permette di abbassare stime di richieste storage
  - Al momento siamo "fuori" di un fattore 4 (eravamo fuori di un fattore 10 2 anni fa …)
    - Con calma si converge …

Il CNAF!



?

- 9 Novembre – il disastro
- Stato attuale
  - Impianto elettrico da buttare (> 1 mese)
  - Tape library: 150 cassette su ~ 7000 bagnate
    - 40 di CMS, di cui 6 di RAW data
    - Questi RAW data in copia CERN → IN2P3
    - Le altre cassette hanno vissuto in un ambiente ad alta umidita' → non gatantito siano ok, sotto ricertificazione Oracle
    - Problemi maggiori per altri esperimenti: dati di CDF RunI?
  - Dischi: problemi ai dischi piu' in basso
    - LHC: il sistema di parita' dovrebbe garantire la leggibilita' dei dati
    - Astroparticle / nucleari: meno fortunati, potrebbero aver perso roba
  - CPU: perso il 10%, ma onestamente sono il problema minore

- Piani:
  - CNAF online a Febbraio (quando pronto il nuovo impianto elettrico)
  - Prima di allora: test di tutte le apparecchiature con alimentazione "di fortuna" e ricertificazione Tape system
  - Entro 10 gg dovrebbe essere possibile una stima di quello che si e' perso
  - In generale: ricomprare tutto quello che
    - Non funziona
    - Funziona me e' stato "bagnato"
- Gli esperimenti LHC paradossalmente sono I meno colpiti, perche' si basano su almeno 10 siti pari al CNAF – problemi maggiori per I piccoli esperimenti (CSN 2 e 3)

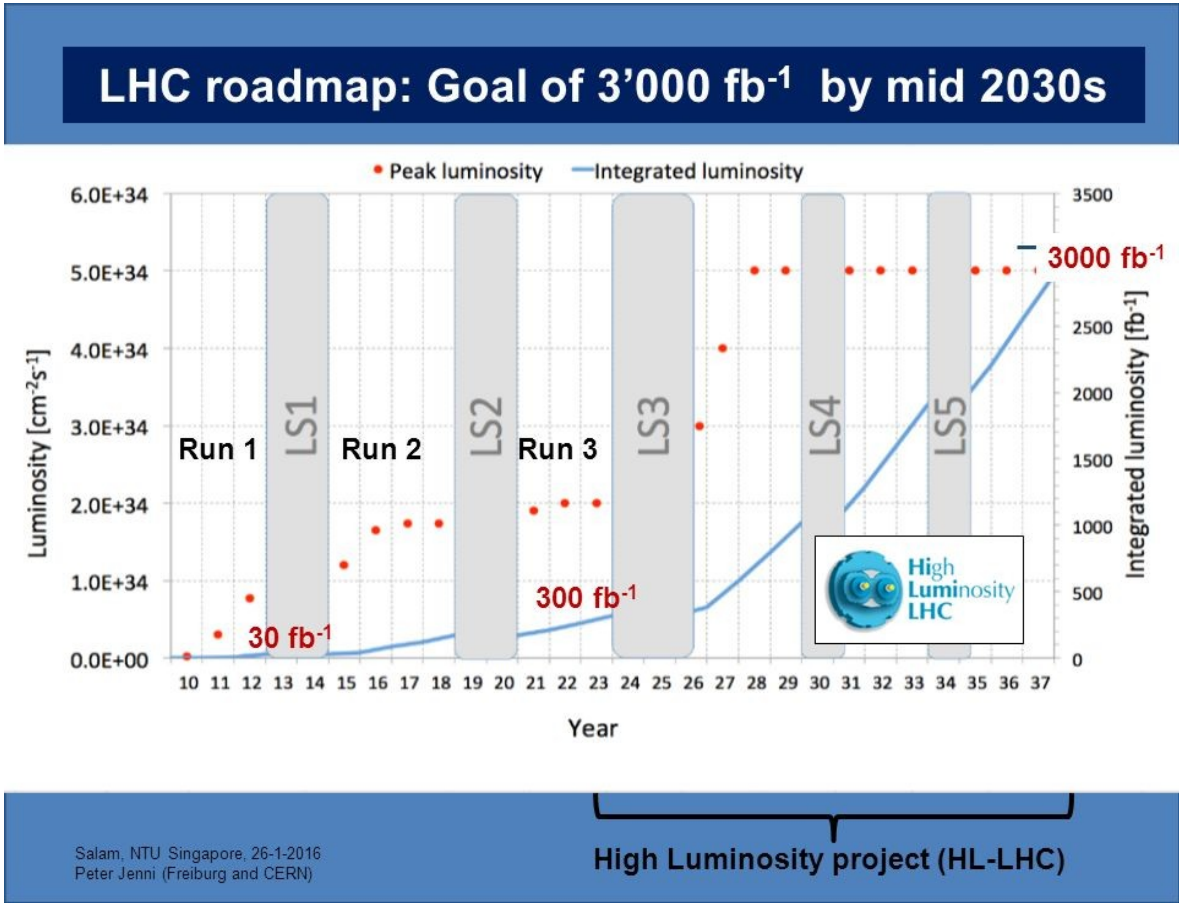
CMS analysis model in Run2 and beyond

Datasets size

- Run2 data: ~10 billions events/year
  - Some analyses can do tight skimming to get to mangeable dataset/ntuples sizes (typically final states with many/high pt isolated leptons)
  - Other analyses cannot preselect much more than HLT already does
    - Hadronic final states
    - Soft final states (not just SMP, FSQ, BPH... Higgs mass is “soft” wrt today's trigger thresholds!!)
    - A very large fraction of the 15 billion events are used in the various steps of the analysis
- Needed MC samples: ~10 billions events/year
  - Modern data analysis (MVA) techniques often required 2-3 fold splitting of the datasets (=> generate more events)
  - NLO and NNLO samples statistic significance killed by negative weights (=> generate more events)

Run3 and HL-LHC

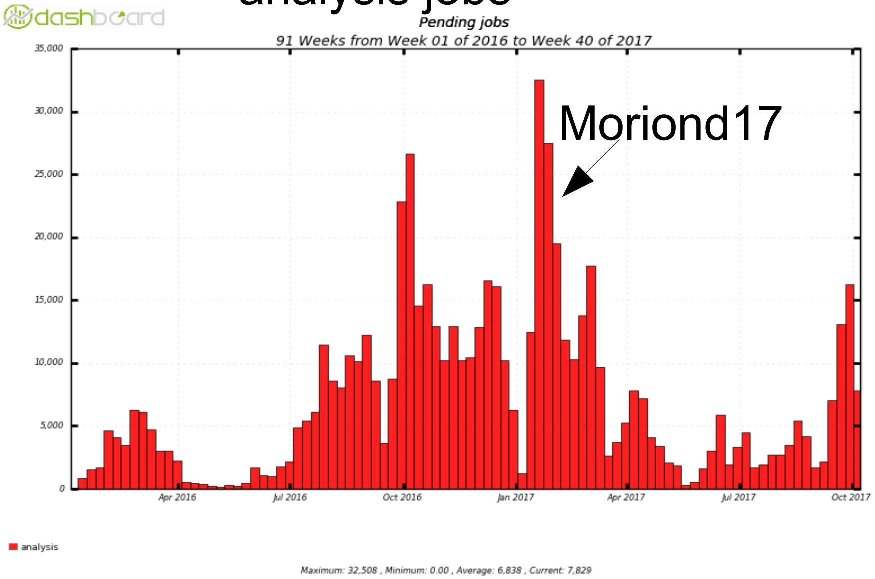
- Run2 lumi "not negligible" wrt to Run3
  - Cannot "forget Run2" (need to access data of ~ 6 years)
- Cannot increase trigger thresholds too much without cutting into physics
  - Increase in trigger rates
- Larger Pileup
  - Larger event size
- Higher detector granularity
  - HGCAL => larger event size
- MC: more precise theory calculations needed
  - More CPU for NNLO generators, filter efficiencies, negative weights



Run2 → Run3 and beyond?

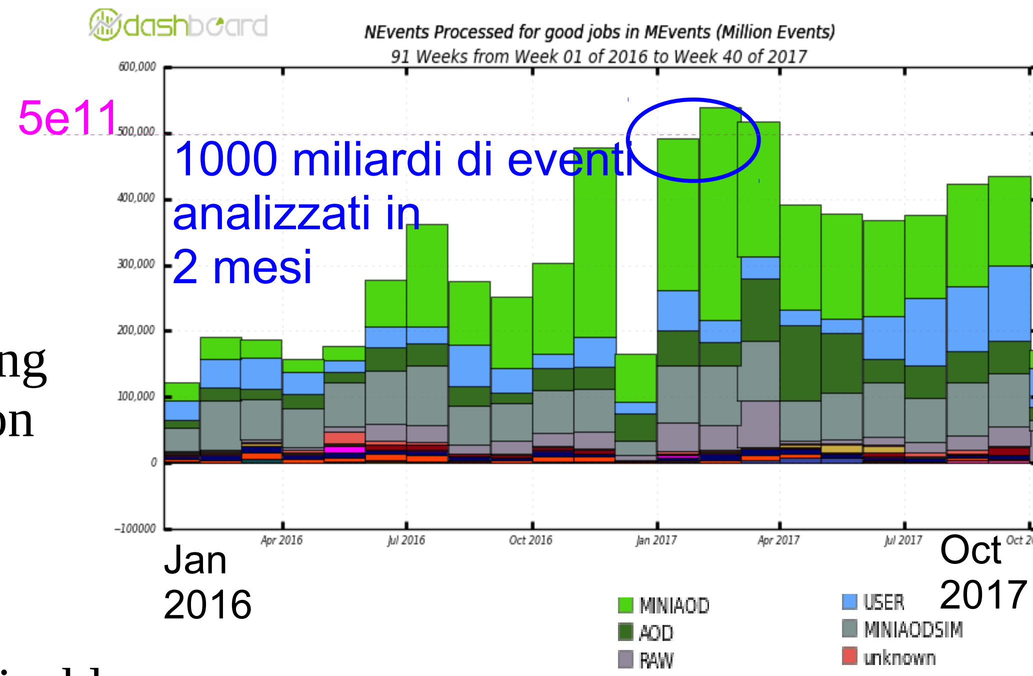
- Can we just rely on the better CPUs, cheaper disks, and in general more resources ?
  - God knows: too many unknowns (e.g. HGCAL reco-time?)
- Experience: Run1 → Run2
  - We could not substain Run1 data distribution / analysis model for 2016 analysis
  - Introduced MINIAOD for a large fraction of the analyses (factor x10 event size reduction)
    - The actual saving in disk was smaller because we “only” reduced the number of AOD copies on disk
  - 2017 critical, see previous slides
  - Analysis CPU usage in peak period (pre-Moriond17) was “critical” (took time to access the full dataset)
- Can we be more efficient?

Number of Pending analysis jobs



Analysis model

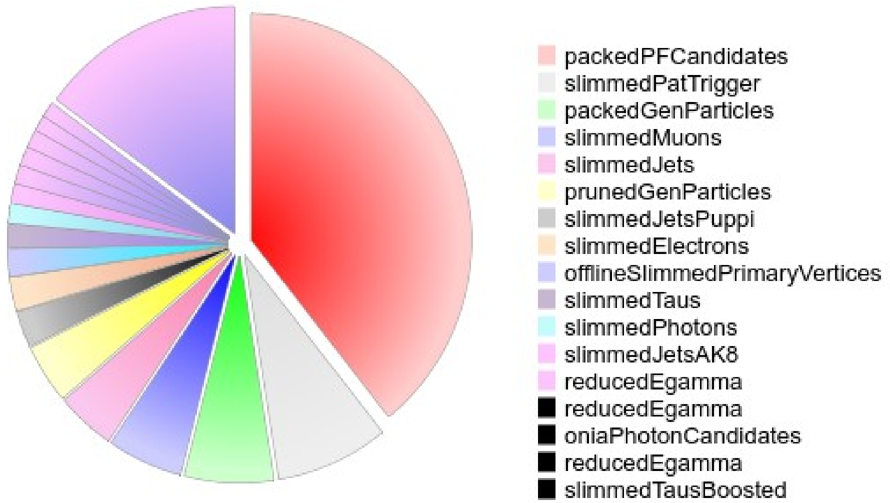
- Datasets size can be reduced vertically/horizontally
  - Reduce event size (RECO->AOD->MiniAOD->Ntuples)
  - Reduce number of events (HLT, Skims, analysis selection)
- No single recipe serving all needs: cannot distribute RECO+RAW for 15billion events
- Central data processing stops at MiniAOD and Skims



  - Every analysis "group" implements ntuplizers, analysis frameworks etc..
  - N similar sets of "ntuples", produced running NxM times on the grid to produce M version of their ntuples (with new JECs, with Moriond2021 lepton ID, with HCAL noise MET cleaing V2134)
  - Code for "group frameworks" is un-maintainable
    - Written by 1-2 people that typical change group, institute, interest, job every a couple of years
    - Filled with un-reviewd, emergency mode written, cut&pasted code

MINIAOD

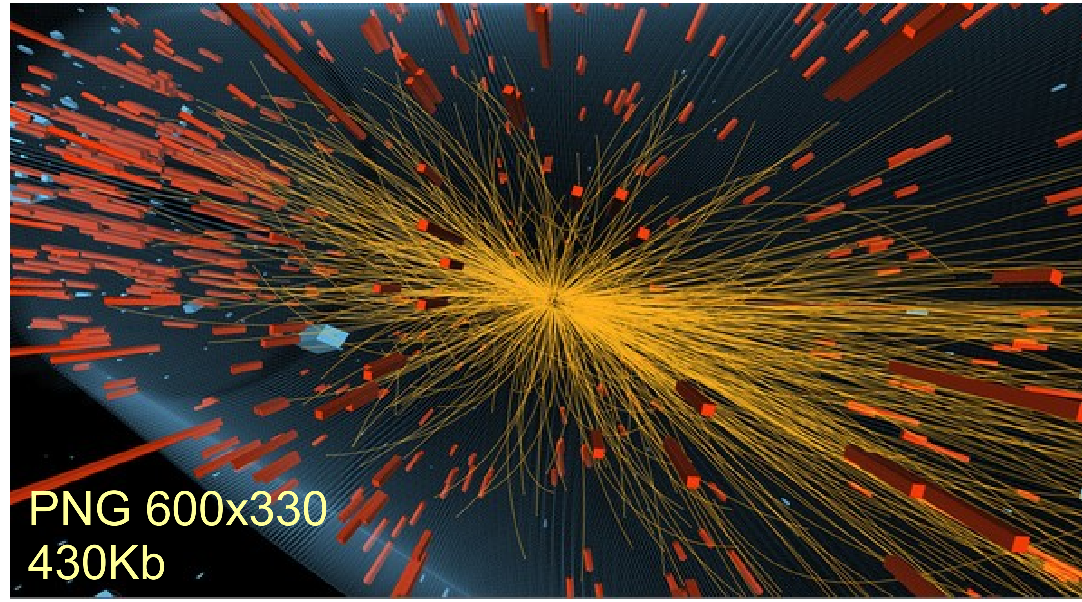
- MiniAOD was a first step to unify one of the "middle layers" (the ~50kb/ev steps that every analysis had) between AOD and final histos
  - It worked well and more analyses switch to using it
- MINIAOD content:
  - All single particles information (in some more or less compressed form)
    - Track + PFcandidate information unified
    - Details up to ECAL rechits or muon segments
  - Allow to recalculate all POG quantities that need fine tuning after the data has been taken
    - ID, isolation, energy corrections, pu rejection
    - All so called "POG recipes"
  - Allow some "special analysis needs"
    - Different jet clustering, ….

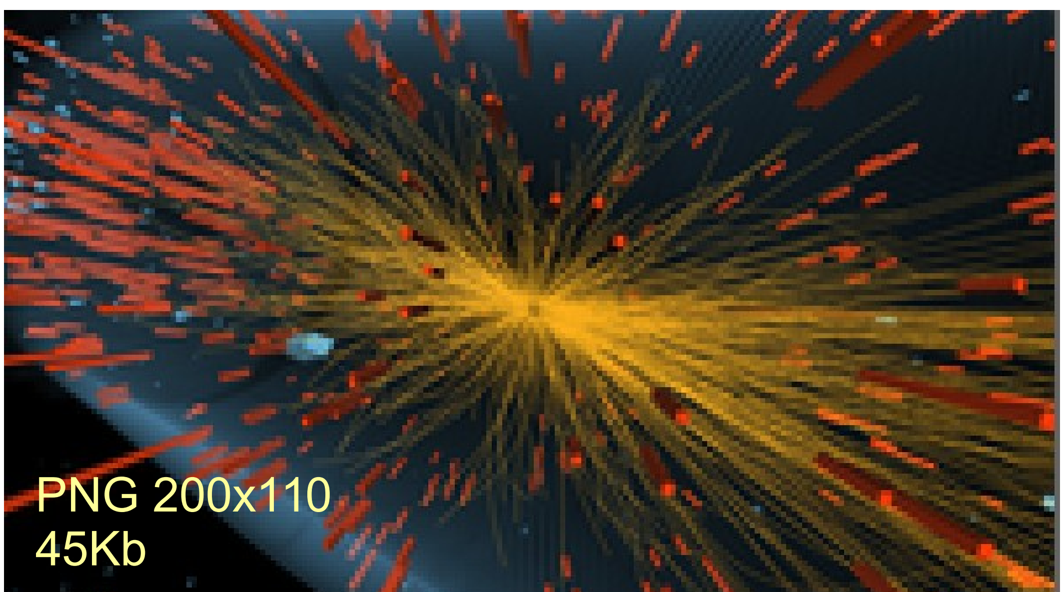


	kb/event
patPackedCandidates_packedPFCandidates__PAT	19.90
patTriggerObjectStandAlones_slimmedPatTrigger__PAT	4.17
patPackedGenParticles_packedGenParticles__PAT	3.30
patMuons_slimmedMuons__PAT	2.77
patJets_slimmedJets__PAT	2.16
recoGenParticles_prunedGenParticles__PAT	2.13
patJets_slimmedJetsPuppi__PAT	1.28
patElectrons_slimmedElectrons__PAT	1.18
recoVertexs_offlineSlimmedPrimaryVertices__PAT	0.97
patTaus_slimmedTaus__PAT	0.84
patPhotons_slimmedPhotons__PAT	0.70
patJets_slimmedJetsAK8__PAT	0.68

Next Step: *nano*AOD

- Can we reduce the event size by at least one more order of magnitude?
- AOD: 450kb/ev
- MINIAOD: 45kb/ev
- NANOAOD (target): ~~4kb/ev~~ 1kb/ev





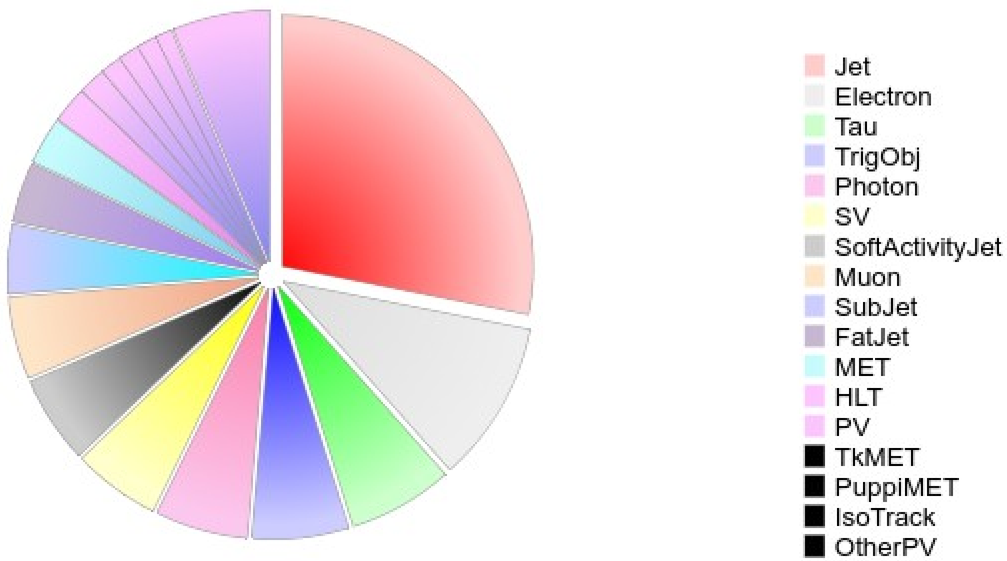
How to spend only 1kb/ev ?

- no tracks / candidate
- no detector details for objects (no segments, cells, rechits)
- no "variables needed to compute ID" (ID and iso are precomputed)
- complex event quantities should be stored rather than providing the needed inputs (even if used by few analyses)
- limit information in long collections (e.g. jets)
- systematic variations not persistently stored (JEC, btag SF, etc.. etc..)
  - They can be computed later with a simple function

    f_corr(pt,eta, and few other variables we can store)
- Do not store 32bit precision floats (1e-7 relative precision) because we do not measure with this precision!

Current prototype size/content

t (8.379 Mb, 10000 events, 0.86 kb/event)



collection	items/evt	kb/evt	b/item
Jet	5.46	0.164	30.8
Electron	0.66	0.061	94.9
Tau	0.64	0.039	63.0
TrigObj	2.93	0.036	12.7
Photon	0.85	0.035	42.0
SV	1.09	0.033	30.7
SoftActivityJet	5.82	0.033	5.8
Muon	0.48	0.031	66.3
SubJet	1.08	0.026	24.3
FatJet	0.60	0.022	38.0
MET	1.00	0.017	17.9
HLT	1.00	0.013	13.6

Electron

Electron_pt
Electron_mvaSpring16HZZ
Electron_mvaSpring16GP
Electron_pfRelIso03_all
Electron_pfRelIso03_chg
Electron_miniPFRelIso_all
Electron_eCorr
Electron_eta
Electron_phi
Electron_dz
Electron_miniPFRelIso_chg
Electron_dxy
Electron_eInvMinusPInv
Electron_deltaEtaSC
Electron_ip3d
Electron_mass
Electron_sip3d
Electron_mvaTTH
Electron_sieie
Electron_r9
Electron_energyErr
Electron_hoe
Electron_dzErr
Electron_dxyErr
Electron_vidNestedWPBitmap
Electron_dr03EcalRecHitSumEt
Electron_dr03HcalDepth1TowerSumEt
Electron_dr03TkSumPt
Electron_jetIdx
Electron_photonIdx
Electron_tightCharge
Electron_cutBased
Electron_pdgId
Electron_charge
Electron_cutBased_HLTPreSel
Electron_lostHits
Electron_isPFcand
Electron_mvaSpring16HZZ_WPL
Electron_mvaSpring16GP_WP90
Electron_mvaSpring16GP_WP80
Electron_cutBased_HEEP
Electron_convVeto
Electron_cleanmask

Jet

Jet_eta
Jet_phi
Jet_bReg
Jet_pt
Jet_mass
Jet_qgl
Jet_btagCSVV2
Jet_rawFactor
Jet_btagDeepB
Jet_btagDeepC
Jet_neEmEF
Jet_neHEF
Jet_btagCMVA
Jet_chHEF
Jet_nConstituents
Jet_area
Jet_puId
Jet_chEmEF
Jet_electronIdx1
Jet_nElectrons
Jet_muonIdx1
Jet_nMuons
Jet_jetId
Jet_cleanmask
Jet_electronIdx2
Jet_muonIdx2
nJet

NanoAOD Format

- NANOAOD format is a bare root ntuple
  - Typical reasonable ntuple format (Muon_pt[nMuons], Muon_eta[nMuons] etc...)
- Even if it is a bare root ntuple, it has some additional goodies
  - Contains Events, Lumi, Run trees to keep track of non-event information
  - Has provenance information
  - It is compatible with most edmXYZ tools (edmProvDump, edmEventSize, etc...)
  - ...and especially, can be produced by central production

NanoAOD features

- Features
  - No cross cleaning is applied (because each analysis needs different criteria)
  - But cross-linking done (using PF constituents)
  - Linking from collections as simple as Jet_pt[Muon_jetIdx]
- NanoAOD-Tools (useful to process nanoaod)
  - Fast and efficient skimming or friend-trees creation
  - JEC uncertainties, jet smearing, btag uncertainties
  - Lepton scale factors etc..
  - Central location for "recipes"
- Auto generated 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

https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookNanoAOD

NanoAOD production

- NanoAOD integrated in CMSSW_9_4_X
- First tests of central production started.. DAS screenshot:

Dataset: /JetHT/CMSSW_10_0_0_pre1-94X_dataRun2_relval_v5_RelVal_nanoin_jetHT2016H-v1/NANOAODSIM
Creation time: 2017-11-23 16:24:50, Physics group: NoGroup, Status: VALID, Type: data
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx , XsecDB Sources: dbs3 show

Dataset: /RelValMinBias_13/CMSSW_9_4_0_pre3-94X_mc2017_realistic_v4_RelVal_nanoaod94X-v1/NANOAODSIM
Creation time: 2017-10-25 15:40:52, Physics group: NoGroup, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx , XsecDB Sources: dbs3 show

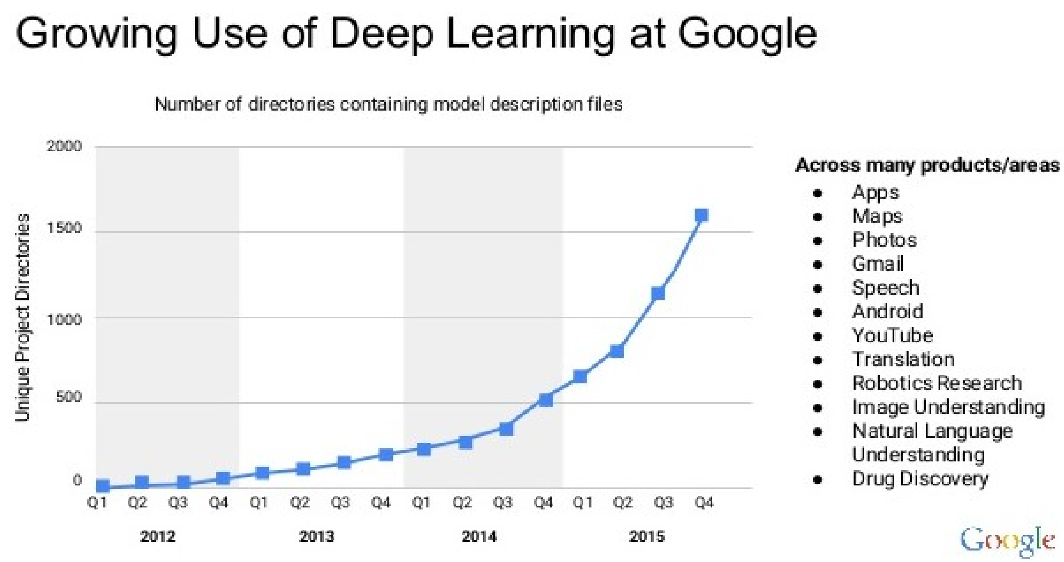
Dataset: /RelValTTbar_13/CMSSW_10_0_0_pre1-94X_mc2017_realistic_v5_RelVal_nanoaod92X-v1/NANOAODSIM
Creation time: 2017-11-20 08:06:59, Physics group: NoGroup, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx , XsecDB Sources: dbs3 show

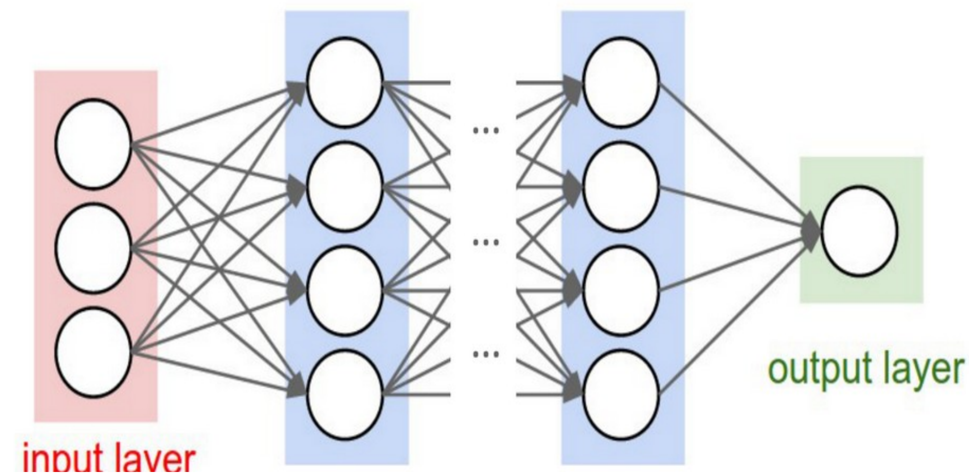
Dataset: /RelValTTbar_13/CMSSW_10_0_0_pre1-94X_mc2017_realistic_v5_RelVal_nanoaod94X-v1/NANOAODSIM
Creation time: 2017-11-20 08:02:33, Physics group: NoGroup, Status: VALID, Type: mc
Release, Blocks, Files, Runs, Configs, Parents, Children, Sites, Physics Groups, py , Subscribe to PhEDEx , XsecDB Sources: dbs3 show

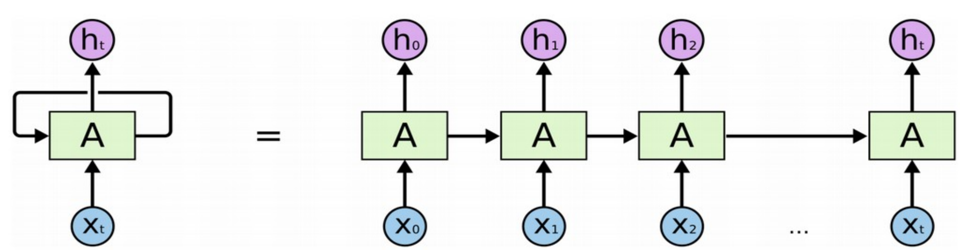
- Central production to start in (early) 2018
- Meanwhile
  - run with CRAB or use nanoaod created by others and published with CRAB
  - Test test test
- Expect bugs
  - and report/fix them !! (issues/PR on github.com/cms-nanoAOD)

Deep Learning

- Deep Learning is the new IT buzzword
  - correlates with “big data”
  - Extract features from “raw” data
- Can be applied to many different problems
  - Classification, regression, feature extraction, clustering, sample generation, ...
- Example of the day:
  - https://arxiv.org/pdf/1711.00043.pdf
  - Translate between two languages just reading a few books in a language then a few (different) books in another language



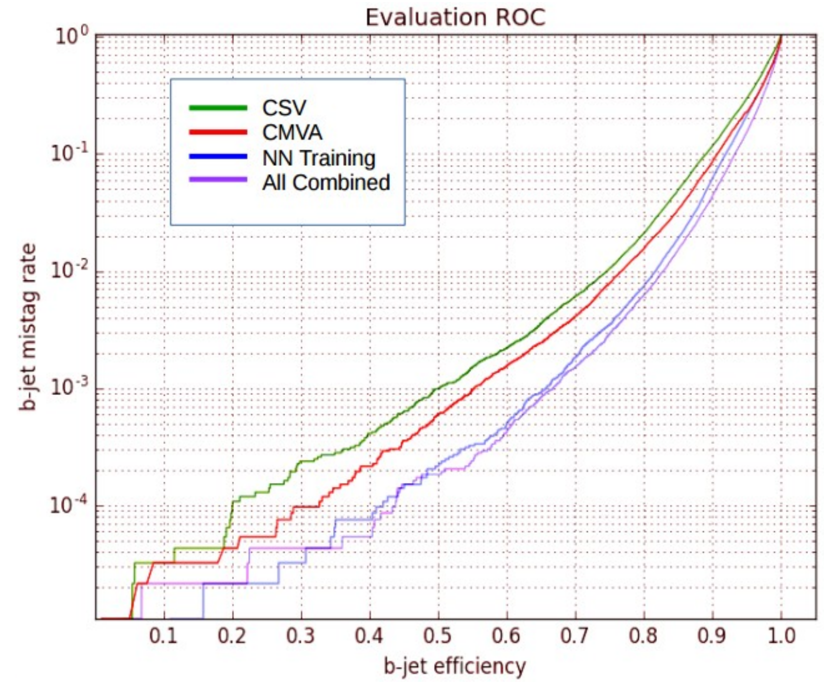


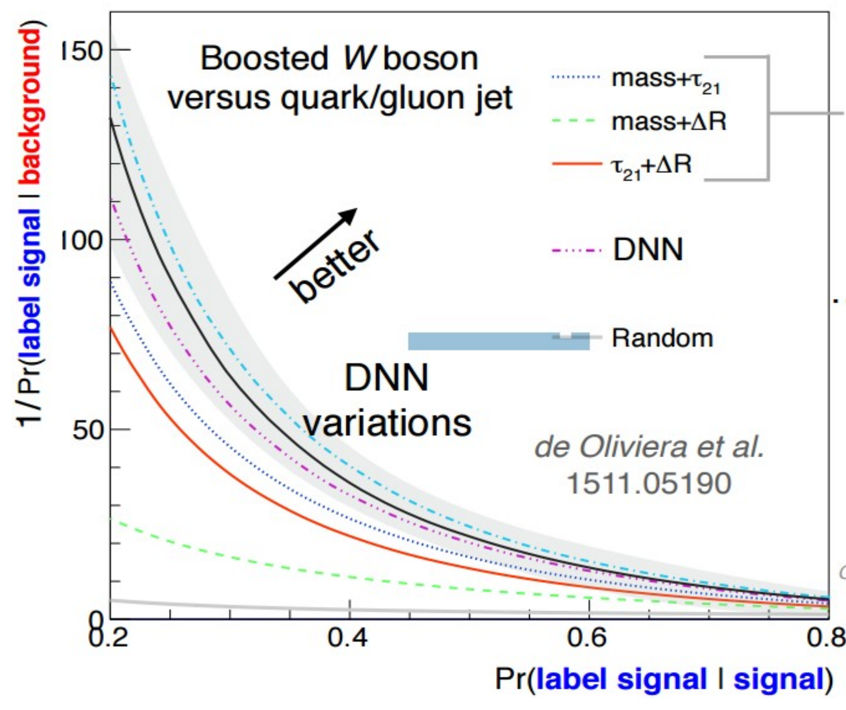


Deep Learning in HEP/CMS

- In CMS and HEP several people looking at DeepNN
  - CERN and CMS dedicated maching learning forum
- Applications
  - Analysis S/B
  - B-tagging
  - Track quality
  - Pixel cluster recovery
  - …
- Integration of most important tool (from google) in CMSSW

TensorFlow

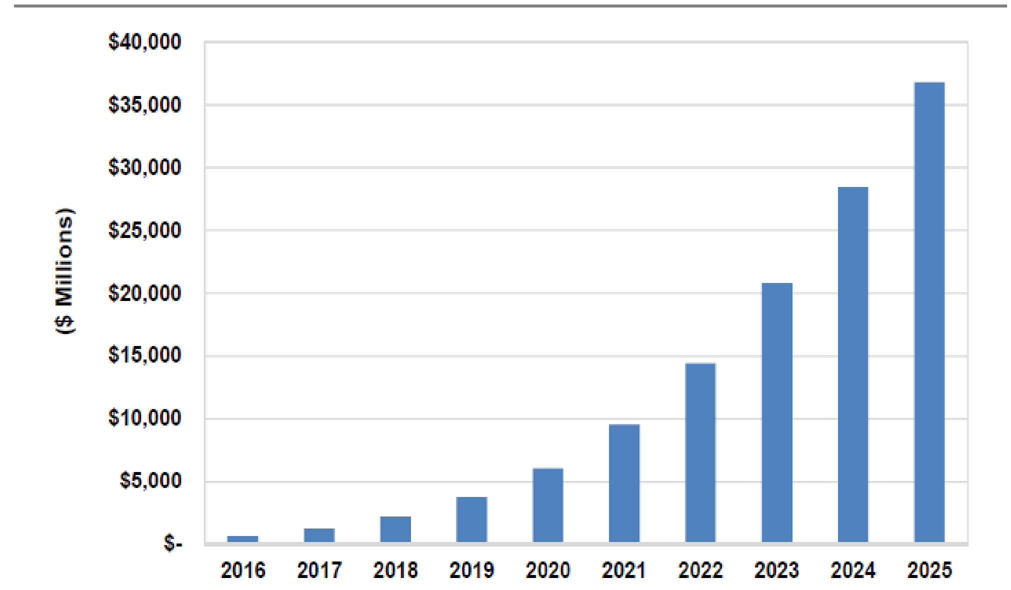


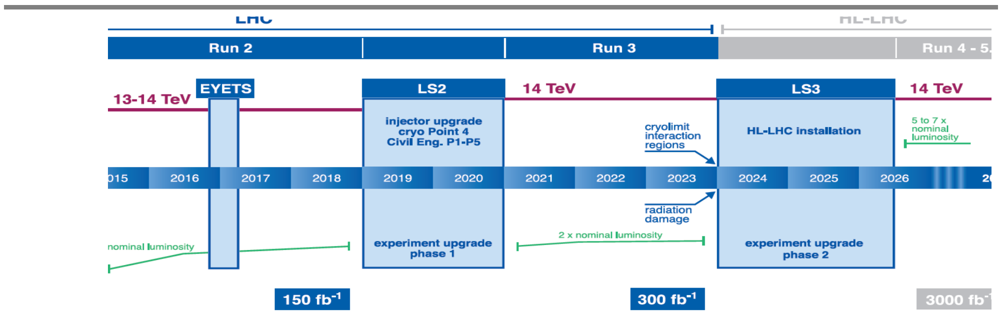


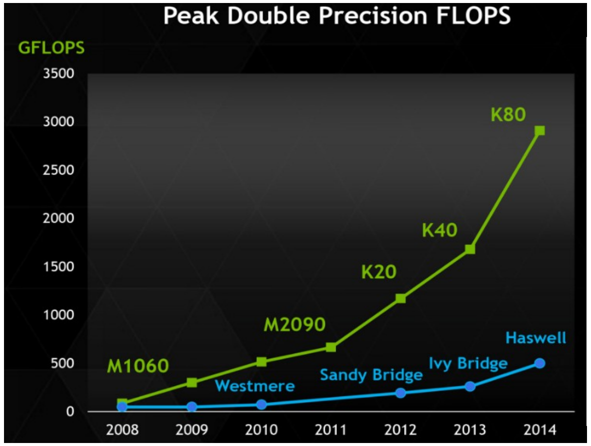
Outlook and Hardware

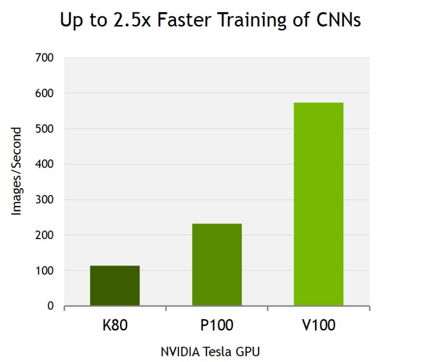
- Prediction of exponentially growing usage of DeepNN
- CMS only has a couple of DNN applications so far
- Where could we use DNN in cmssw? (a random set of examples)
  - Particle flow
  - Track seeding
  - Object ID
  - HG-CAL reco
  - PV identification
  - SV reconstruction
- DeepNN training benefit from GPUs
  - Not needed for “evaluation”
- Should we buy some powerful machine with GPUs for training?

Chart 1.1 Artificial Intelligence Revenue, World Markets: 2016-2025









Conclusions

- 2017 was above thComputing is changing rapidly with Deep NN, e specs, somehow survived
- 2018 need planning on HLT output rate, datasets format to distribute / store on disk / move to tape
- Longer term is getting more complicated but we have a new tool, the nanoAOD... we should try to test them in 2018
- Deep NN growing everywhere in “bigdata” (and we are big data!), need to understand potential applications and hardware needs
- Next years are not going to be boring in software and computing

backup

What are Micro/NanoAOD?

How can we get to such target:

- no tracks / candidate
- no detector details for objects (no segments, cells, rechits)
- no cross cleaning, but cross referencing (i.e. the user should decide what to keep/drop)
- (almost)no "variables needed to compute ID" (ID and iso are precomputed)
- rather complex event quantities should be stored rather than providing the needed inputs (even if used by few analyses)
- systematic variations not stored (JEC, btag SF, etc..): provide instead simple setup to compute them on the fly

Format details

- The format is that of a bare ROOT ntuple, i.e.
  - nMuon
  - Muon_pt[nMuon]
  - Muon_eta[nMuon]
  - ...etc..
- The system supports several types of variables:
  - simple values per event: e.g. rho
  - vectors of values: e.g. lhe weights lheWeights[n]
  - single objects: e.g. met (met_et, met_phi, etc...)
  - collections of objects: e.g. jets (jet_pt[n], jet_eta[n], ...)
- We include also some Lumi/Run information (e.g. processed lumisections, sum of generator weights), and minimal config/provenance information

NanoAOD support library

- NanoAOD format can be used directly in bare ROOT, but for convenience we will also provide a standard library of tools to work with it:
  - Support for simple standard tasks like filtering with a JSON file, merging files, skimming, …
  - Tools to apply at read time standard recipes for calibration and systematic uncertainties (e.g. JEC variations, b-tag Sfs)
  - The library will be usable both from CMSSW and from bare ROOT + PyROOT

Content details

- We had a few brainstorming session with random CMS collaborators to try to draft a possible prototype content
  - First draft created during Moriond 17
  - Everything that is small (as in “few bytes”) but can add an analysis to the pool of “covered analyses” can be added
  - Analyses that would need +1kb/ev to run → use miniAOD
    - Same for analysis tasks that need detailed information and do not need to be run on all 20B data+MC events (e.g. tag & probe, commissioning, tuning of IDs, trigger studies, ...)
  - Anyhow, content is not cast in stone
    - And what we present today is a prototype
- Here the link to a spreadsheet we use to define and keep track of the content (please use it to comment!)
  - http://cern.ch/go/hq6t

Cross ~~cleaning~~ linking

- MiniAOD collections are not cross cleaned
  - i.e. leptons appears as jets, both a photon and an electron can originate from a single SC, almost all jets are taus etc...
- Cross collection cleaning is a typical example of “analysis dependent choice”
  - We do not want to enter your analysis freedom
  - So we do not cross clean.
- … but we cross link!
  - With PF/GED we have an obvious way to know what should be cleaned (are two objects sharing PFCandidates? Then your analysis should decide where the candidate belong...)
  - We save links (i.e. just indices) among POG objects in the final format

Cross *cleaning*

- Some associations are to a single object
  - Jet_pt[Muon_jet] is the pt of the jet linked to a given muon
- Some associations are to multiple (max 2) objects
  - Muon_pt[Jet_muon1] and Muon_pt[Jet_muon2] are the pt of the two muons contained in a jet
- When no association is available the index is -1
- How about the common SUSY-Had cleaning we had in our centrally produced ntuples?
  - Bit masks: we support pre-calculated crosscleaning decisions that can be proposed by PAGs/SubGroups (they take 1 bit!)
  - Example producer for this is given so that for a PAG maintaining this is a simple as writing down the definition of the cleaning algorithm

More linking

- Linking with bare indices will also be used to
  - Match gen information
  - Associate subjets to fatjets
  - Associate Sec Vertices to jets
  - ….
- Because those are not “framework references/ptr” but just indices...
  - The name of the index tells you the collection it points to
  - Muon_collectionIdxAdditionalSuffix
    - Muon_jetIdx
    - GenPart_genPartIdxMother

Current prototype

- The current prototype implements about 93% (in size) of the content we “planned”

	expected <N>	Size	Fraction implemented	bytes impl	bytes missing	
Jets	10	235	92.34%	217.00	18.00	
Electron	2	61.5	91.79%	56.45	5.05	
Muons	2	57.25	89.52%	51.25	6.00	
Tau	2	62.25	74.43%	46.33	15.92	
Photons	2	40.25	89.44%	36.00	4.25	
MET	1	30	80.00%	24.00	6.00	
Prim Vertices	1	16	100.00%	16.00	0.00	
Sec Vertices	2	44	86.36%	38.00	6.00	
Gen Part	50	381.25	100.00%	381.25	0.00	
Other	1	20.5	92.68%	19.00	1.50	
OtherMC	1	54	90.74%	49.00	5.00	
Trigger bits	300	75	100.00%	75.00	0.00	
Trigger obj	20	135	88.89%	120.00	15.00	
Boosted Jets	1	40	95.00%	38.00	2.00	
SoftActJets	5	33	100.00%	33.00	0.00	
GenJets	10	90	88.89%	80.00	10.00	
Isotracks?	1	10	50.00%	5.00	5.00	
			92.80%	**1,285.28**	bytes/ev	
	total expected	**1385**	bytes/event			c
	projected zlib	*2313.6*	measured zlib:	**2147.00**	bytes/ev	
	projected lzma	*1761.8*	measured LZMA	1635.00	bytes/ev	

Documentation

- Each variable comes with some minimal documentation
  - Newcomers can immediately understand what a variable is
  - Accessible directly from root file, from the title of the root branch where the variable is stored
  - Easy to dump into any format one may like, e.g. to produce html documentation pages, or compare different versions
- The config file can also be extracted from the provenance using standard CMSSW tools (edmProvDump)

Documentation

- Example html documentation extracted from a root file (just proof of concept, a better tool is being developed)

Content

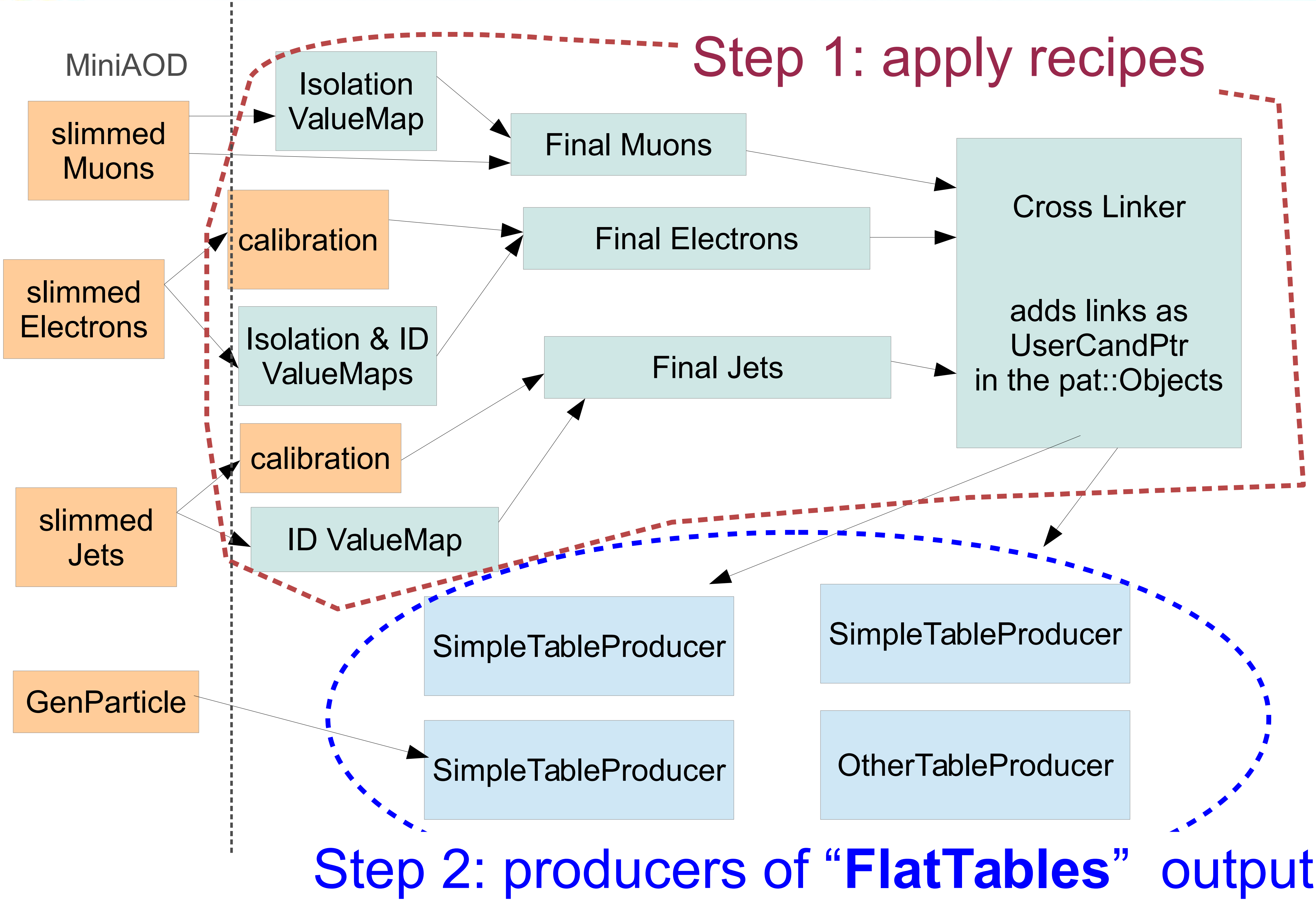
Collection	
GenPart	interesting gen particles
Jet	slimmedJets, i.e. ak4 PFJets CHS with JECs applied, after basic
Electron	slimmedElectrons after basic selection (pt > 5)
TrigObj	
LHEPdfWeight	
SV	
GenJet	
Tau	deFi oose
Photon	
HLT	
Muon	l &&

Object property	Type	
Muon_pt	Float_t	pt
Muon_PFIso03_all	Float_t	PF isolation dR=0.3
Muon_PFIso03_chg	Float_t	PF isolation dR=0.3
Muon_miniPFIso_all	Float_t	mini PF isolation,
Muon_eta	Float_t	eta
Muon_phi	Float_t	phi
Muon_dxy	Float_t	dxy (with sign) wrt
Muon_ip3d	Float_t	3D impact parameter
Muon_sip3d	Float_t	3D impact parameter
Muon_dz	Float_t	dz (with sign) wrt
Muon_mvaTTH	Float_t	TTH MVA lepton ID s

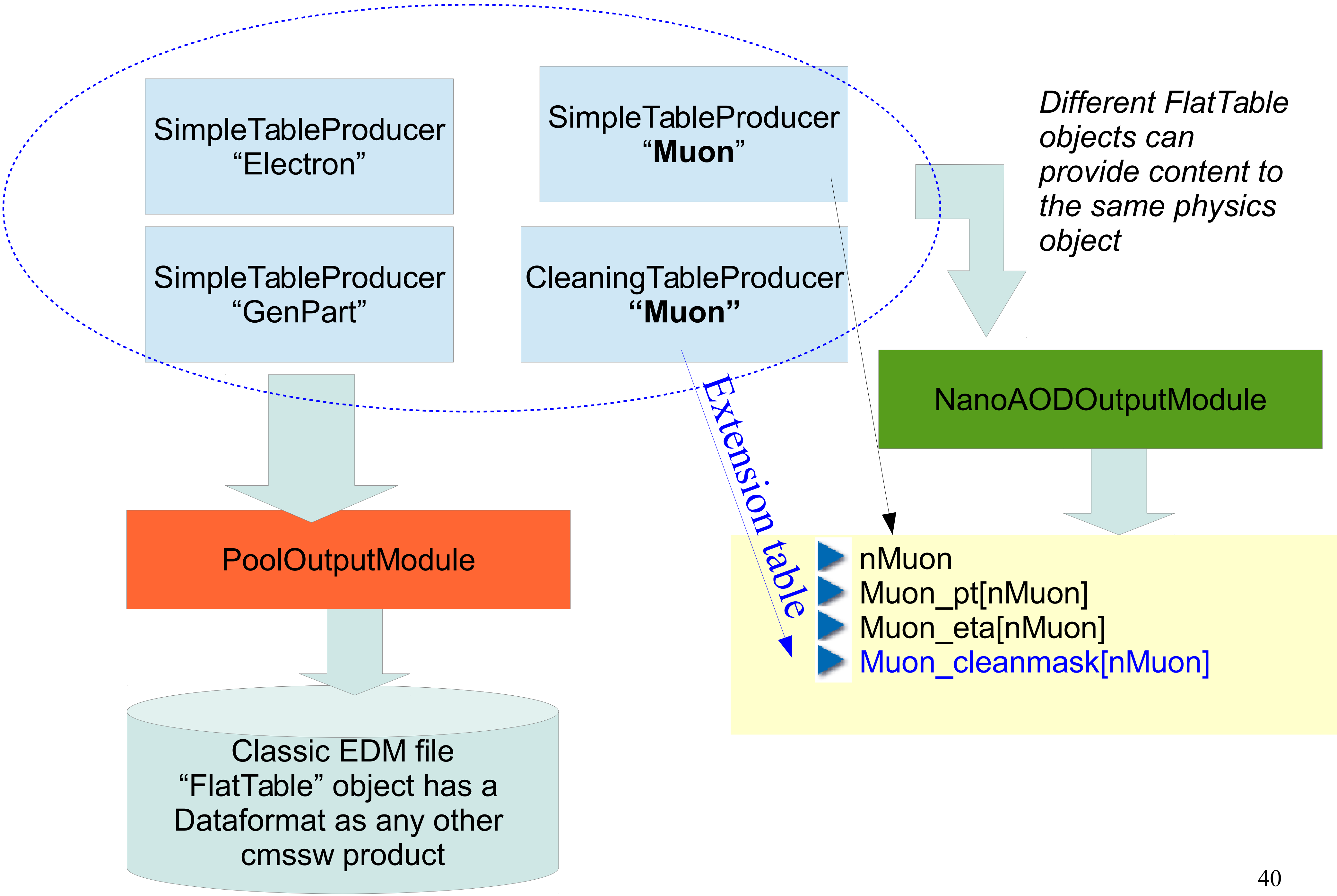
NanoAOD production workflow

- NanoAOD will be produced from MiniAOD via CMSSW with a standard workflow
  1) Run POG & PAG recipes on top of miniAOD objects (perform some basic selection, apply calibrations, produce extra information in ValueMaps, ...)
  2) Convert objects into flat tables of values
- A special OutputModule will then read all the tables write out a ROOT file in NANOAOD format
  - The format is not EDM, but will be fully supported by DAS
- To support central production, flat tables can also be saved into intermediate EDM files with the PoolOutputModule
  - The merge CMSSW job can read the tables in all the input files, and write the output in NANOAOD format

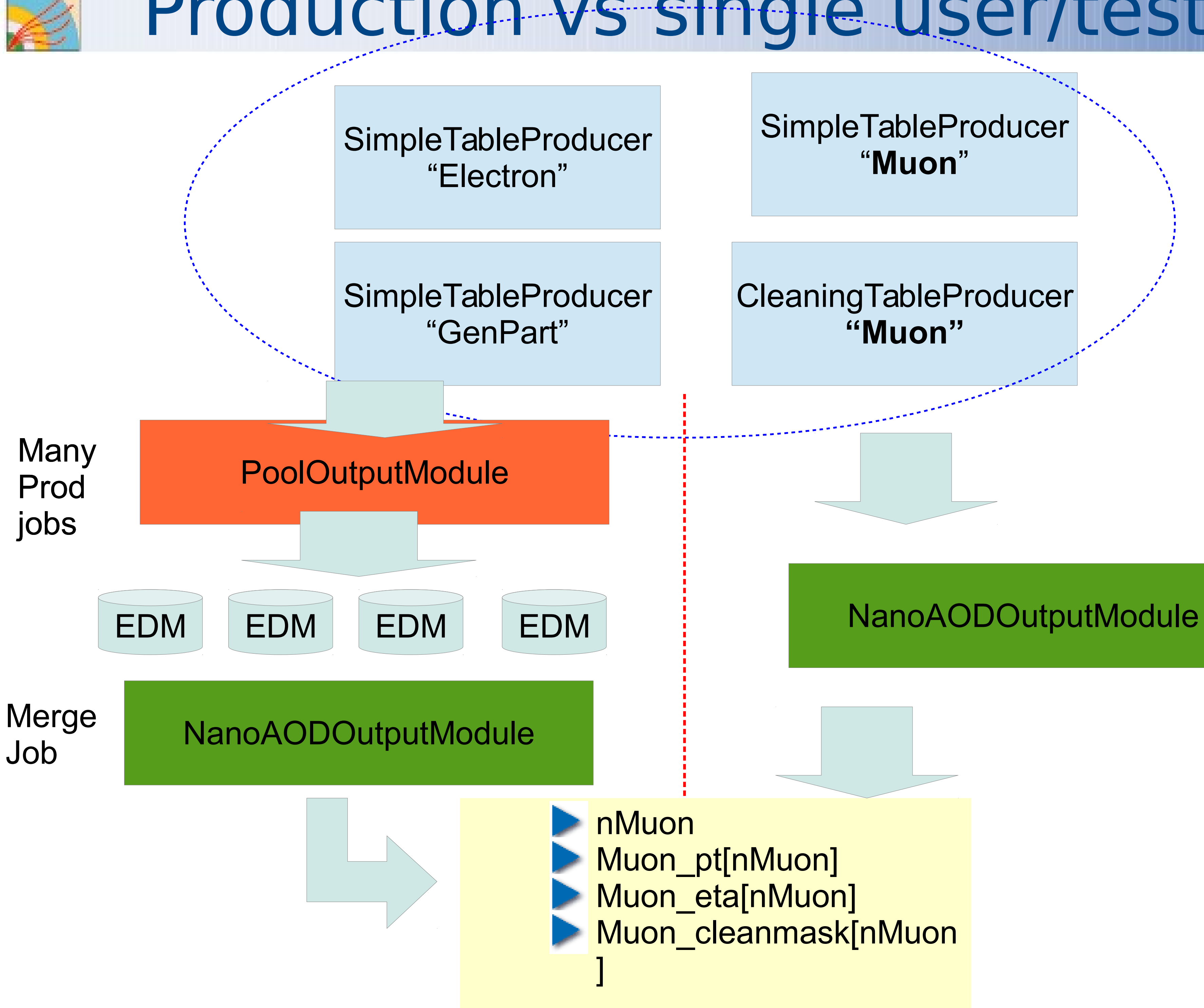
NanoAOD cartoon workflow



Output modules



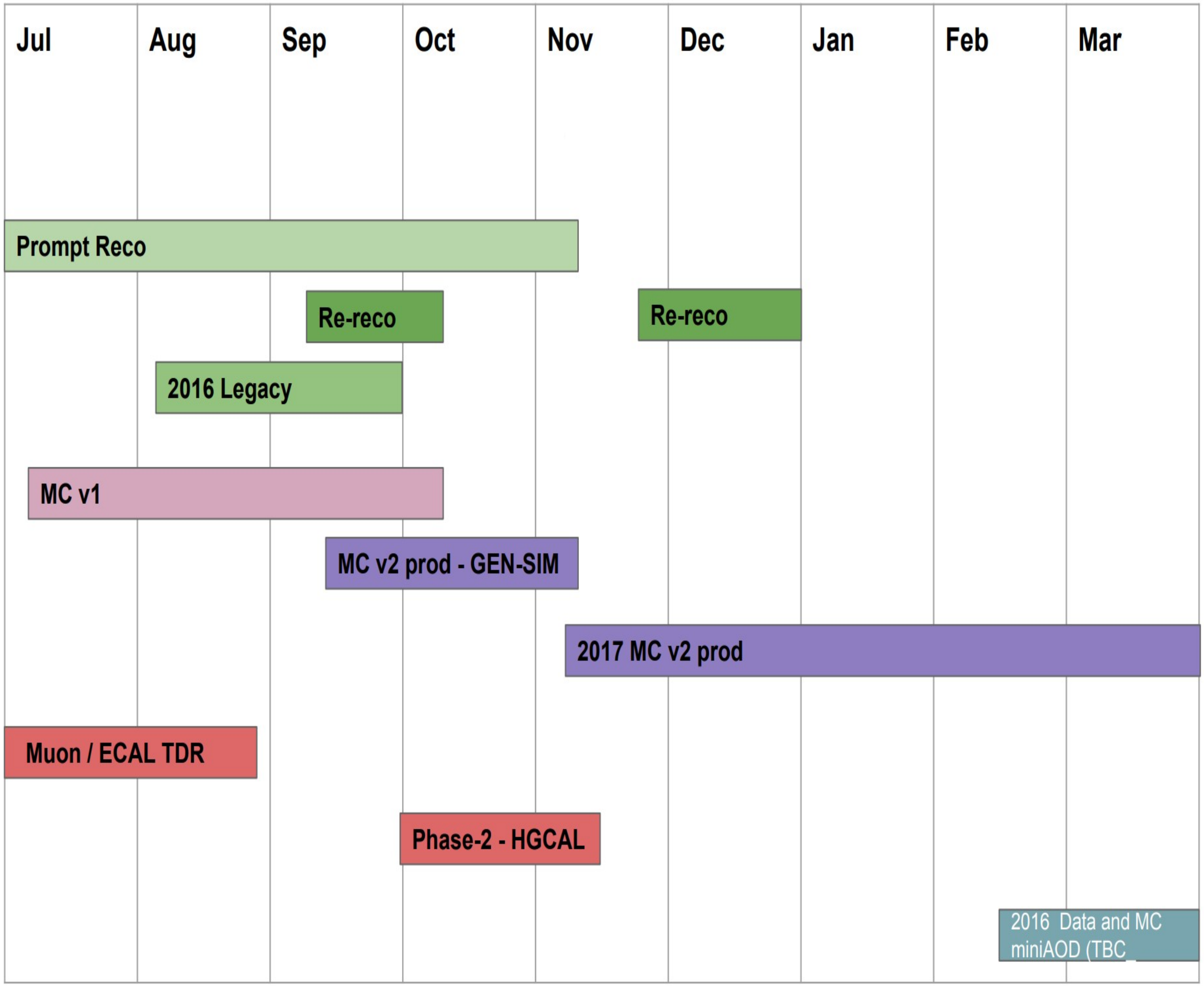
Production vs single user/testing



Using NanoAOD (postprocessing)

- Nano AOD are a common format so by design there are tasks typical of ntuplization that are left out of the central processing
- A postprocessing system is being setup to simplify common tasks in that step too, e.g.
  - compute SF for leptons from txt files
  - compute JEC and basic variations
  - compute factorized jec variations
  - further match objects across collections
  - check if event passes the json lumi selection
  - run SVFit
  - Provide functions to compute invariant masses, deltaR, etc..
  - … others … (see the NanoAOD spreadsheet under Utilities tab)
- Some post processing tasks could be slow
  - Run with or after a “skimming” step

Ultima versione del piano generale – obsoleta per definizione



Organigramma 2018-2019

** Joint with Physics