

The Evolution of Analysis Models for HL-LHC

Andrea Rizzi^{1,*}

¹INFN and University of Pisa, Italy

Abstract. A quick review of how analyses were and are performed at LHC in Run1 and Run2 is proposed. A discussion then follow on how this could scale and what can be improved for Run3 and for High Lumi LHC. Critical items are highlighted and some interesting emerging technologies are discussed.

1 Introduction

The Large Hadron Collider is going to be upgraded, in the next decade, to the higher luminosity version named High Luminosity LHC (HL-LHC). As the name says the emphasis is on large delivered luminosity, hence large volumes of data will need to be processed and analyzed by the experimental collaborations. During LHC lifetime several steps in luminosity have been already performed: Run1 was composed by an initial 7 TeV data taking with only 5/fb of data recorded, followed by an 8 TeV period with about 20/fb; Run2 is composed by multiple years of data taking delivering in multiple steps a total of 160/fb at 13 TeV. The luminosity as function of the year of data taking for LHC and the prediction for HL-LHC are shown in Figure 1. It is interesting then to study how the analysis models evolved from Run1 to Run2 and during Run2.

Before trying to understand the evolution of analysis models we should try to define what an analysis model is. From a computing point of view, an analysis model is about defining the access to resources and the amount of resources needed for the so called “analysis” part of the data processing. This is, in practice, the part that is customized to specific analysis needs and that is largely executed by individual physicists or groups of physicists rather than centrally handled by the collaboration production teams. The point of view of the user, i.e. a physicist mostly performing analysis, is slightly different. What could be called “analysis model” from his standpoint is the set of tools and the organization and interplay among those tools used for data reduction and plot making.

An analysis models should be studied both from the point of view of the access to resources and from the perspective of the available tools and their interaction with users and resources.

2 The evolution of analysis model in Run1 and Run2 of the LHC

During Run1 the model followed by the LHC experiment was to centrally process data for what concern the so called “Reconstruction” step and then let the user handle any subsequent processing or derivation of smaller datasets with direct access to grid services. Most analyses

*e-mail: andrea.rizzi@cern.ch

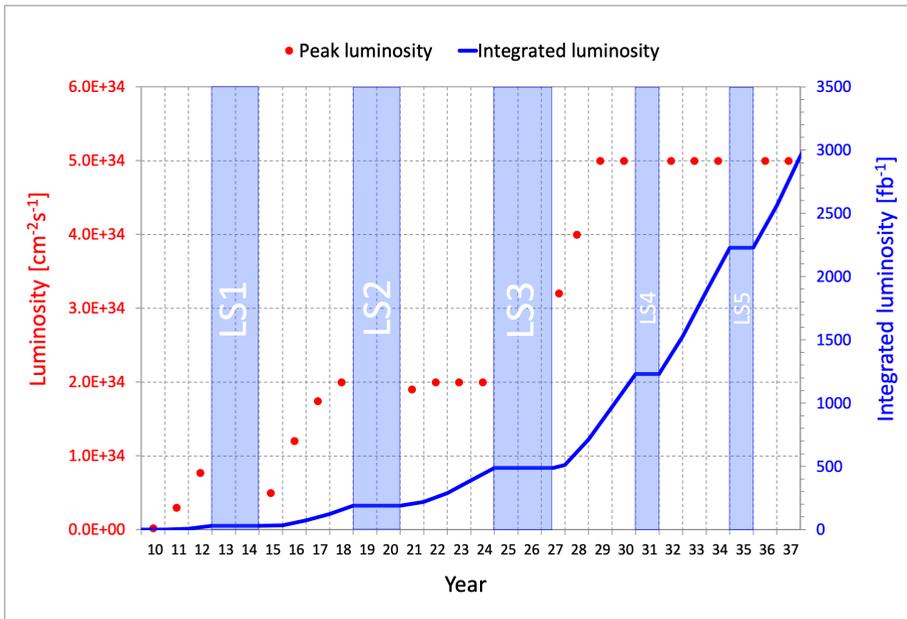


Figure 1. LHC luminosity in Run1 and Run2 compared to predictions for Run3 and for High Lumi LHC [9]

were using the “AOD” format [1], that, with several hundreds kB per event, was providing both a description of high-level objects and many detector details useful for recalibration and detector studies.

Analysis groups were typically deriving from those datasets some more compact per event representations, possibly also reducing the number of events with harder selection on main signature objects compared to those applied in the online trigger.

During the shutdown between Run1 and Run2 it was clear that such model was not sustainable from the computing point of view and different solutions were proposed.

A widely adopted solution has been that of a “train model” [2]. In this model the custom software written for each analysis is treated as a wagon in a train that runs at a fixed schedule. The single train with multiple wagons allows a reduction of the needed I/O as long as wagons share the input data. The controlled schedule in addition ensure that analysis resources are not saturated. Each wagon finally produce a derived dataset that can be used by analyzers.

While initially planning to adopt the train model, a different approach was preferred in CMS thanks to the definition of a reduced event format (MiniAOD [3]). The reduced event format was designed to be an order of magnitude smaller than the original AOD format, to be able to serve a large fraction of the CMS analyses and to replace the privately derived formats that all analysis groups were using during Run1. Also with this approach the overall need for resources was highly reduced, in this case both in term of CPU needs for analysis processing and for storage of analysis datasets.

For both CMS and ATLAS experiments, during Run2, the analysis steps following the derivation with train model or the production of MiniAOD were completely left to user tastes. Multiple toolkits and frameworks, developed by some institute or analysis groups, have been adopted to produce histograms, plots and statistical interpretation of the data.

It must be noted anyhow that further intermediate event level datasets are typically produced in most of those toolkits starting from input datasets and before creating histograms. Those intermediate format are usually just ROOT n-tuples (TTrees) with minimal metadata information.

Why were those intermediate datasets needed? Why not reducing directly to histograms and plots? The reason for intermediate datasets is mostly speed and human time saving as discussed in the next section.

3 Human machine interactions and workflows

When trying to make benchmarks for a new analysis tool we are used to take an existing analysis, e.g. some Higgs discovery example, and see how that can be written in such tool. This anyhow does not tell anything about how good such tool is for analysis.

What we fail to test, with such approach, is how good the tool is for the development and evolution of the data analysis. An analysis idea is first tested on some generator level samples, then exercised on some full simulation MonteCarlo, than studied in data control regions for limited data taking period, then expanded, tuned, optimized and finalized with systematic studies and possibly cross checked with alternative approaches.

The time needed from first idea to publication is typically between one and several years and in this period of time a lot of tuning and intermediate studies are performed. The idea that the main deliverable of an analysis model is some input datasets and a way to produce histograms from this is pretty naive and does not catch the main point of everyday analysis work.

What tools for analysis should provide is a quick and efficient way to answer questions that will arise during the analysis development process (from initial idea to final peer review). This is the very reason why most analysis groups implemented, both in Run1 and in Run2, a multi-level caching of the data processing with one or more intermediate datasets (ntuples) splitting the analysis processing in chunks (e.g. first select events passing trigger criteria and calibrate the objects, then interpret the event and build derived objects such as resonances, finally apply an MVA and store a dozen relevant variable for the defined signal and control regions).

Having multiple checkpoints to resume the analysis process from is a convenient way to quickly answer questions going back only up to the first usable checkpoint (i.e. if you just want to verify how the result would change tightening a region definition cut, you do not need to recalibrate the physics objects or to recompute the invariant masses).

It should also be noted that analysis steps can have very different CPU needs ranging from microseconds per event to seconds per events as shown in Figure 2, hence it is very natural to place checkpoints right after time consuming steps.

In general analyses can be described as complex workflows with heterogeneous steps of event processing, data reduction and data interpretation. The data to handle is not limited to the “datasets” of event-by-event measurements, in fact there is a relevant component of the input data originating from other sources that we can generically refer to as non-event data.

4 Non-event data

While HEP data analysis can be considered an example of big-data, there are many specific aspects that make it quite unique. One of the peculiarities is that the bulk of the data is represented by the event-data, i.e. information that is different event by event, but on the

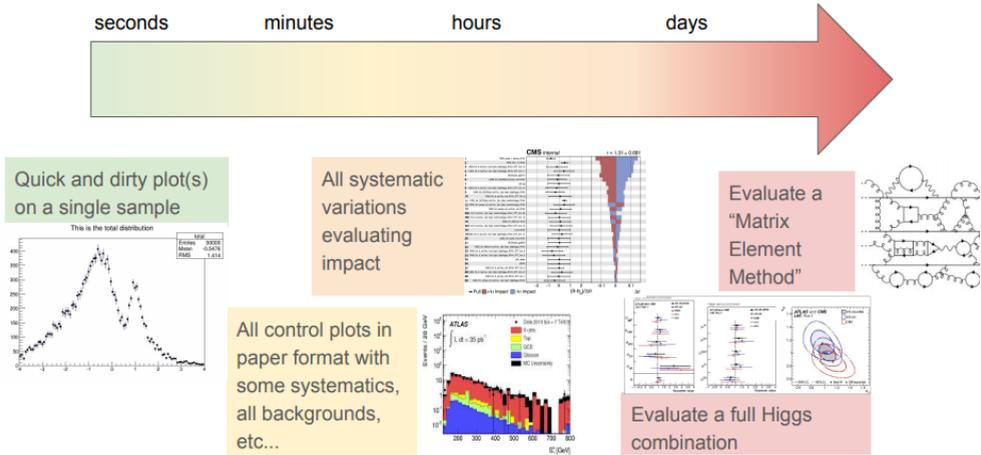


Figure 2. CPU time needed for different analysis level processing tasks: single histograms can typically be filled at multi KHz rate while on the other extreme Matrix Element Method for event classification can have sub-Hz rates.

other hand such information is useless without access and proper usage of a pretty large set of meta-data or non-event data.

Examples of non-event data are:

- Cross section and normalization information for simulated samples
- Generated processes included in a given simulated sample
- Phase space of the generated process, including relevant variable for combined usage of multiple samples
- Integrated luminosity and running conditions for data
- Corrections to the simulation, such as data to simulation scale factors
- Efficiency maps and calibration corrections
- Various other details that could be sample or experiment specific

In HEP data analysis in addition we attempt to discover new processes or measure known ones with well defined uncertainty. The need to calculate the uncertainty is a major difference compared to many other fields of big data analysis. The propagation of multiple uncertainty sources is a delicate process that is exploiting two possible techniques: recomputing the whole analysis (or at least the relevant fraction of it) with varied input objects or reweight the events with weights representing distortion of the input variables distributions as predicted by the uncertainty source. A typical analysis usually have to use both techniques for different uncertainties, with the first technique being much more expensive, in term of CPU power, than the latter.

5 Common physics objects definitions and analysis variability

The observables related to a reconstructed particles are usually grouped in what we call “physics objects”. For example a muon particle can be reconstructed by different detectors and properties such as flight direction, momentum, impact parameter with respect to the

interaction point, number of detection points, etc. are measured. Those are the properties of a muon object and in a single event multiple objects are usually grouped in uniform containers (muons, electrons, photons, jets, etc...).

Quality selections can be used to distinguish leptons or photons from jets, electrons from photons or particles originating from the main interaction from particles originated in pileup. Each analysis usually defines a different set of cuts for selecting physics objects and may prefer one reconstruction algorithm over another for a specific task. As an example jets are objects created with clustering algorithms, these clustering algorithms have tunable parameters and what is optimal for low energy analyses is likely not good for the so called “boosted regime” where ~ 100 GeV mass particles have momenta above the TeV and decay hadronically producing collimated jets with internal substructures.

A large variability in object definitions is then due to intrinsic differences of analyses, the final state of interest and the typical energy regime. On the other hand pretty large equivalence classes can be identified looking at analyses that target the same phase space or the same kind of physics objects with different goals.

Some differences in the object definitions, looking especially at Run1 and Run2 experience in CMS and ATLAS, anyhow remains even within those equivalence classes. The source of those difference can be either some minor optimization (i.e. like squeezing the last percent of performance improvement) or just stochastic noise (i.e. two analysis reaching different optimal cuts but the two being equivalent within the uncertainties).

While during Run1 and Run2 we could afford this high level of customization, a great potential for saving resources lays in the reduction of the unnecessary variability. This can be achieved by defining common datasets with standardized format and contents for analyses of the same equivalent class. A step further can be done having a single format containing the superset of the information required by all (or a large part) of the analysis equivalence classes.

The CMS NanoAOD format [4] and the effort in ATLAS to develop a lighter analysis format go exactly in that direction. Taking the NanoAOD as an example, the idea is to cover a large fraction of the analyses (more than 50%) with a single format. I.e. it should cover many of the analysis equivalence classes but not all of them. In order to do so, the NanoAOD format contains both a collection of jets for low momentum analysis and a collection for the boosted regime; the leptons are reconstructed with a single algorithm but few quality cuts are applied so that this kind of customization can happen in a later step of the analyses.

In order to cover a larger phase space than what CMS NanoAOD is aiming at today, a possibility is to have multiple (few) formats serving different classes of analyses. As an example a format tailored for exclusive B physics can be imagined as an alternative to the current NanoAOD that is more general purpose and is expected to cover analyses in Higgs physics, Standard Model measurements, SUSY and other not-too-exotic searches or top physics.

In any case, analyses that need access to very low level information for customized reconstruction algorithms (e.g. Heavy Stable Charged Particle search) will need to follow a different path, skimming the relevant data directly from original reco or raw datasets.

6 Towards HL-LHC

During Run3 the LHC collaborations will have a way to test the feasibility of the data reduction ideas and fine tune the common event formats. On top of this, there are other interesting technologies that are being developed, prototyped or even deployed for production that could play a relevant role in analysis at HL-LHC.

6.1 Highly parallel analysis tools

The number of events to handle at analysis level with HL-LHC will increase given the higher luminosity, the larger trigger bandwidth and the fact that a large part of the physics program is at relatively low energies (100 GeV scale). While computing resources will also somehow scale up, it is clear that a large fraction of the computing power will be in form of multi-core processors or GPU accelerators. Currently the HEP software stack for analysis is not very parallel friendly (e.g. thread safety is often not granted) and physicists taking care of analysis optimization often lack specific skills to design highly parallel software.

In this context new tools are being developed to express basic event processing operations that could be efficiently parallelized. Two examples of this kind of tools are:

- **RDataFrame**: framework in C++ developed by the ROOT team (and part of the most recent ROOT Toolkit release) [5]
- **Coffea/awkward-arrays/uproot**: framework in python developed at FNAL for columnar analysis [6]

The idea is that the concrete code executed to perform a set of event processing operations is not written by the physicist expert in analysis but rather by a skilled software developer that can optimize the code for multi-threading, for GPUs, for a HPC or for a Spark cluster.

The usage of those tools requires to avoid explicit for-loops. While this is pretty trivial for the “event loop” as in an analysis the events are usually not correlated, the explicit loops on collections of objects cannot be avoided unless predefined functions are provided for a given collection-task, hiding the actual implementation to the user. For example the task of finding the maximum in an array should be performed with a “findmax” operation rather than explicitly coding the maximum finding in a for-loop.

While the most trivial operations and algorithms are already available in the languages used by the two tools (C++ `std::` algorithms or python `numpy`), there are once again HEP domain specific operations that are common in the field but not trivial to implement with basic container algorithms.

In order to address this problem domain specific analysis languages can be implemented to translate complex, but rather common, analysis operations into the basic operations natively implemented by the toolkits such as RDataFrame and Coffea.

6.2 Analysis languages

Domain specific analysis language are a long time, and long term, dream proposed by several people in the past decades in HEP. The idea is to create some language to describe **what** we want to do in an analysis rather than **how** we do it.

Several attempts and prototypes have been discussed in the past (e.g. [7]) especially addressing the event processing part and the creation of histograms. A more complete language should also describe how to obtain the input data and how to run statistical interpretation of the data running final fits and signal extraction.

Analysis languages could be a key element for HL-LHC for several reasons: they could guarantee high performance and parallel code (because the actual code is not written by the analyzer but rather by the language compiler/interpreter/translator), they can simplify the creation of new analysis exploring new ideas (e.g. reducing the time from the original idea to publication) but, most important, they can free manpower (currently wasting time in re-implementation of analysis plotters and macros) for fundamental tasks in detector operations, reconstruction and simulation software developments, etc..

Another advantage with analysis languages, combined with common analysis formats, could be seamless analysis preservation, reproducibility and analysis portability across experiments or across different running periods of the same experiment.

In the recent years there has been multiple attempts to develop an analysis language and dedicated events to discuss them (such as a 4 days workshop in FNAL [8]), so it could be a concrete possibility for HL-LHC.

6.3 Deep learning

The last emerging technology that can be a game changer for HL-LHC analysis is Deep Learning (DL). As of today machine learning, and DL in particular, is used in the final discrimination of signal vs background, in search analysis or in object definition and identification. New applications of DL in HEP are foreseen for reconstruction and simulation tasks, for data quality monitoring, for resource usage optimization but also for analysis. In particular anomaly detection techniques can be used for unbiased searches or for real time analysis at trigger level.

7 Conclusions

In conclusion, HL-LHC will pose new challenges for analysis especially in term of data volumes. A better organization of the data reduction steps should include a careful evaluation of the common practices to avoid duplications retaining the flexibility needed to ensure that any kind of analysis has access to the needed data.

Parallelism at analysis level will be unavoidable and proper interfaces for physics analysis to parallel resources will be needed. Several efforts along this direction already started and can be tested during LHC Run3.

A better human to computer interface to describe the analysis, to fetch the data and metadata, to schedule execution on available resources could be very useful to improve the scientific output of HL-LHC.

References

- [1] B. Hegner, “The CMS Analysis Model” , Journal of Physics: Conference Series **331** (2011) 032014
- [2] J. Catmore et al, “ A new petabyte-scale data derivation framework for ATLAS”, Journal of Physics: Conference Series. **664** 072007
- [3] G. Petrucciani, A. Rizzi, C. Vuosalo, “Mini-AOD: A New Analysis Data Format for CMS” **J.Phys.Conf.Ser. 664 (2015) no.7, 072052**
- [4] A. Rizzi, G. Petrucciani, M. Peruzzi, “A further reduction in CMS event data for analysis the NANO AOD format”, **EPJ Web of Conferences 214, 06021 (2019)**
- [5] ROOT Team, RDataFrame documentation at https://root.cern/doc/master/classROOT_1_1RDataFrame.html
- [6] Coffea project, <https://coffeateam.github.io/coffea/>, doi:10.5281/zenodo.3345994
- [7] G. Watts, Developing a Declarative Analysis Language: LINQToROOT, **EPJ Web of Conferences 214, 06039 (2019)**
- [8] Analysis Description Languages for the LHC, <https://indico.cern.ch/event/769263/>
- [9] Taken on march 2020 from <https://lhc-commissioning.web.cern.ch/lhc-commissioning/schedule/images/LHC-nominal-lumi-projection.png>