

Analysis Ecosystem at the HL-LHC

Gordon Watts*

September 10, 2020

Abstract

The HL-LHC will see a $\times 10$ more data than Run 2 and Run 3. Disk space, one of the most expensive aspects of computing hardware, will be at a premium. The experiments are responding by building single corrected and un-skimmed datasets. A single processing of these datasets will be many 100's of TB or PB's as the HL-LHC progresses. The field does not have the disk space or i/o bandwidth for each analysis to skim large parts of these datasets locally, at scale. Nor should graduate students have to babysit the jobs required to do this. Many new tools are being developed to address this, from ones based in the traditional ROOT environment, like RDataFrame and RNTuple, to those built in the python ecosystem like awkward-array and coffea and ServiceX and func.adl. This LOI discusses the eco-system for analysis in the HL-LHC - how all the tools will be put together in a usable, responsive, and reproducible way.

Introduction

The ATLAS and CMS HL-LHC[1] are settling on computing models that rely on two datasets. The first is a highly compressed dataset, with minimal information - enough to satisfy at least 50% of analyses use cases (the nanoAOD at CMS[2] and PHYSLITE at ATLAS[2]) and the second is $\times 10$ larger (the miniAOD at CMS and PHYSDAOD at ATLAS) that can satisfy the rest of the analyses. The larger of the datasets range in the low numbers of PB/year, and the smaller range from 100's of TBs up to a PB/year. Many other experiments are facing similar challenges.

This increase in dataset sizes will force the nature of analysis at the experiments to change. The most common path currently starts GRID jobs run on output of an experiment's production system. These files are downloaded to a group or analysis common disk area. Those outputs are skimmed again to a smaller set of files - the analysis ntuples. These files are usually small enough a complete run can be made on a single desktop PC or even a laptop in some cases in a few hours or a day. Control of this process is mostly ad-hoc, using bash, python, ROOT[3] macros, some Jupyter notebooks[4], and hand-art. The HL-LHC will apply a $\times 10$ to each of these stages, perhaps more for many of the precision analyses that have to be done to realize the full physics potential of the HL-LHC.

The automation likely required to run on datasets of the HL-LHC size will require analyses to evolve from their common current state of a collection of scripts and C++ source and ROOT macros and python scripts and Jupyter Notebooks to something that is much more organized. Further, it isn't clear the best use of a new graduate student's time is learning how to run on 100's of TB's of files rather than making a plot. The field must move to a simpler way to express its physics intent. This simpler way must have high fidelity, be flexible, sustainable, use resources efficiently, and archival and reproducible.

The community has already started exploring aspects of this. Projects like ServiceX[5] are exploring data discovery and rapid delivery. The 'awkward' array package[6] enables columnar analysis in python. 'RDataFrame'[7] is a declarative analysis work-flow based in C++ and enabled in python, part of the ROOT ecosystem. 'coffea'[8] is a python-based columnar analysis framework which has interfaces to Spark[9] and DASK[10] and other systems for rapid parallel processing and production of histograms. Jupyter Notebooks enable code, text, and graphics to co-exist in a single document, transforming how we interact with our analysis code. New, more powerful, techniques for histogramming are being developed in the ROOT7 and 'boost.histogram'[11] projects. Recent years have shown a sudden flood of declarative analysis languages - allowing the analyzer to specify what they want done rather than how they want it done. There is also an effort (represented by another LOI[12]) to make an analysis differentiable[13] - and that will have impacts up and down the analysis tool-set.

*University of Washington, Seattle

Work has also begun understanding how to host these tools. A separate LOI on Analysis Facilities[14] has been submitted to Snowmass. An analysis facility is a modern compute cluster that will furnish a range of services, including data delivery (ServiceX, XRootD[15], etc.), batch jobs, Jupyter Notebook servers, hosted terminals with experiment’s programming environments, etc. That LOI discusses problems such as how to best share resources between interactive and notebook users and batch facilities that are part of the facility, or located in an adjacent Tier 2, how to best deliver data quickly to analyzers, how to configure hardware to enable the proper amount of caching, etc.

Challenges to a Coherent Analysis Vision for the HL-LHC

Work will progress on these individual tools over the course of the next year. This LOI is meant to collect (a small subset of) the ecosystem-level issues and challenges that should be addressed to make a physics analyzer’s job as easy as possible. What follows are some of the eco-system level issues to be thought through.

What services should we expect an analysis facility to provide? How tied should its design be to the field? To an experiment? To an analysis? How easily can a physicist move their analysis from one facility to the next?

How can we support both columnar and procedural approaches in the same analysis? Some algorithms are more straight forward when written in a procedural method (track finding, jet building, etc.). Others are most concisely expressed in a columnar format. Many existing, well understood and debug utilities exist in C++. How do we build all of this into an ecosystem that supports the new methods of analysis - avoiding rewriting all of the code we currently have?

Systematic errors and physics-object corrections present some of the most thorny problems, partly due to combinatorics and partly due to the many different ways they are applied and derived. Different experiments have taken very different approaches to applying systematics, some requiring going all the way back to the larger data sample. Solving this in a way that does not stress our data access facilities and achieves the precision required for HL-LHC analysis will likely require getting a broad range of people working together, from physics management to object identification groups to physics groups.

Machine Learning (ML) will figure more prominently in our analysis workflows. The standard ML workflow starts by extracting data into local files. Then designing a ML network, and using the local files to train. Then taking the results of the training and moving it back into the data extraction workflow. What is needed from an Analysis Facility to support this? From the software eco-system to make these transitions as seamless (and trackable and reproducible) as possible?

The effort to better integrate the ROOT and python ecosystems has a long history in our field[16]. As new tools appear on both sides (ROOT’s RDataFrame and RNTuple, python distributed analysis tools like DASK and Spark, analysis pipelines like coffea, histogram packages like boost.histogram), what is needed to continue to integrate the two so the best tool can be used for this task?

Reproducible Analysis have already to paid dividends in Run 2[17]. What work needs to be done to make this work in this new era? What will it take to preserve RDataFrame analyses, or columnar analyses? What about micro-service architectures for delivering data?

There is also the last mile. For example, making it easy to push likelihoods, plots, tables generated in an analysis into HEPDATA[18]. Making sure a new graduate student can quickly make plots on an analysis group’s ”derived” data. Can you pick up work from multiple places in the world without some complex setup or large data download required?

Goals

As work progresses on building a coherent analysis ecosystem the field will discover gaps as well as new ways to address old problems - that span more than just one tool or how it interacts with the Analysis Facility or the physicist. In some cases, only as prototypes are built will the field understand the issues and how they fit into the overall puzzle. Lastly the field has to keep in mind that the tool-set we start the HL-LHC with may not be the one we finish with. However, if we design the components well enough, we can allow for a smooth evolution as new and better tools are written. The work underway in ATLAS, CMS, CERN, US ATLAS and US CMS, LHCb, IRIS-HEP, and other entities is building this new ecosystem. The field has enough pieces in place to start looking at the big picture.

References

- [1] G. Apollinari *et al.*, *High Luminosity Large Hadron Collider (HL-LHC)*. CERN Yellow Rep.Monogr. 4 (2017) 1. CERN, 2017.
- [2] Tommaso Boccali, “Atlas and cms at the hl-lhc computing models.” https://indico.cern.ch/event/719844/contributions/3047940/attachments/1746734/2828521/quantum_computing_atlas_and_cms_Nov_2018.pdf, 2018. Presented at the Quantum Computing for High Energy Physics Workshops.
- [3] R. Brun and F. Rademakers, “ROOT: An object oriented data analysis framework,” *Nucl. Instrum. Meth. A* **389** (1997) 81–86.
- [4] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. D. Team, “Jupyter notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87–90. IOS, 2016.
- [5] B. Galewsky, “Servicex – a distributed, caching, columnar data delivery service.” <https://indico.cern.ch/event/773049/contributions/3474438/>, 2019.
- [6] J. Pivarski, P. Elmer, and D. Lange, “Awkward Arrays in Python, C++, and Numba,” in *24th International Conference on Computing in High Energy and Nuclear Physics*. 1, 2020. [arXiv:2001.06307](https://arxiv.org/abs/2001.06307) [cs.MS].
- [7] D. Piparo, P. Canal, E. Guiraud, X. Valls Pla, G. Ganis, G. Amadio, A. Naumann, and E. Tejedor, “RDataFrame: Easy Parallel ROOT Analysis at 100 Threads,” *EPJ Web Conf.* **214** (2019) 06029.
- [8] N. Smith *et al.*, “Coffea – Columnar Object Framework For Effective Analysis,” in *24th International Conference on Computing in High Energy and Nuclear Physics*. 8, 2020. [arXiv:2008.12712](https://arxiv.org/abs/2008.12712) [cs.DC].
- [9] Zaharia, Matei and Xin, Reynold S. and Wendell, Patrick and Das, Tathagata and Armbrust, Michael and Dave, Ankur and Meng, Xiangrui and Rosen, Josh and Venkataraman, Shivaram and Franklin, Michael J. and Ghodsi, Ali and Gonzalez, Joseph and Shenker, Scott and Stoica, Ion, “Apache spark: A unified engine for big data processing,” *Commun. ACM* **59** no. 11, (Oct., 2016) 56–65. <https://doi.org/10.1145/2934664>.
- [10] Dask Development Team, *Dask: Library for dynamic task scheduling*, 2016. <https://dask.org>.
- [11] H. Schreiner, H. Dembinski, Nino, C. K. Maji, C. Burr, D. Davis, and P. Grimaud, “scikit-hep/boost-histogram: Version 0.10.2,” July, 2020. <https://doi.org/10.5281/zenodo.3967982>.
- [12] A. G. Baydin, K. Cranmer, M. Feickert, L. Gray, L. Heinrich, A. Held, A. Melo, M. Neubauer, J. Pearkes, N. Simpson, N. Smith, G. Stark, S. Thais, V. Vassilev, and G. Watts, “Differentiable programming in high-energy physics.”. Submitted as a Snowmass LOI.
- [13] Y. LeCun, “Deep Learning est mort. Vive Differentiable Programming!” <https://www.facebook.com/yann.lecun/posts/10155003011462143>, Jan, 2018. <https://www.facebook.com/yann.lecun/posts/10155003011462143andhttps://techburst.io/deep-learning-est-mort-vive-differentiable-programming-5060d3c55074>.
- [14] R. Gardner *et al.*, “Analysis facilities in the hl-lhc eta.”. Submitted as a Snowmass LOI.
- [15] “Xroot project page.” <http://www.xrootd.org/>.
- [16] “Hep analysis ecosystem workshop.” <https://indico.cern.ch/event/613842/timetable/>, 2017.
- [17] “Reinterpretation of the atlas search for displaced hadronic jets with the recast framework.” <https://cds.cern.ch/record/2714064>, 2020.
- [18] E. Maguire, L. Heinrich, and G. Watt, “HEPData: a repository for high energy physics data,” *J. Phys. Conf. Ser.* **898** no. 10, (2017) 102006, [arXiv:1704.05473](https://arxiv.org/abs/1704.05473) [hep-ex].