

Future Evolution of Computing/Reconstruction in High Energy Physics

David J Lange*

Princeton University

Princeton, NJ 08544, USA

E-mail: David.Lange@princeton.edu

We discuss the needs and plans for evolving event reconstruction and computing infrastructures of high-energy physics experiments towards the high-luminosity LHC era. Realizing the full physics potential of programs of the high-luminosity LHC experiments will require the high-energy physics community to address a number of challenges in the area of reconstruction algorithms, software and computing. These challenges center around a number of themes, including: complexity of event environment; enhancements in detector capabilities; changes in commodity computing technologies; legacy software; and the need for increased sensitivity in high-luminosity LHC analyses. We summarize some of the research and development opportunities in these areas.

*Sixth Annual Conference on Large Hadron Collider Physics (LHCP2018)
4-9 June 2018
Bologna, Italy*

* Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

<https://pos.sissa.it/>

1. Introduction

The enormous productivity of physics researchers working as part of the Large Hadron Collider (LHC) experiments is enabled by a robust infrastructure that spans a broad set of detector and computing technologies. Purpose designed and built software is a critical part of the physics production pipeline for any high-energy physics (HEP) experiment. The usage of software that is custom designed and developed by each scientific collaboration spans from selecting events in real time all the way to simulation, analysis, and the final published plots. The flow from detector data down to analysis is illustrated in Figure 1 for the LHC experiments.

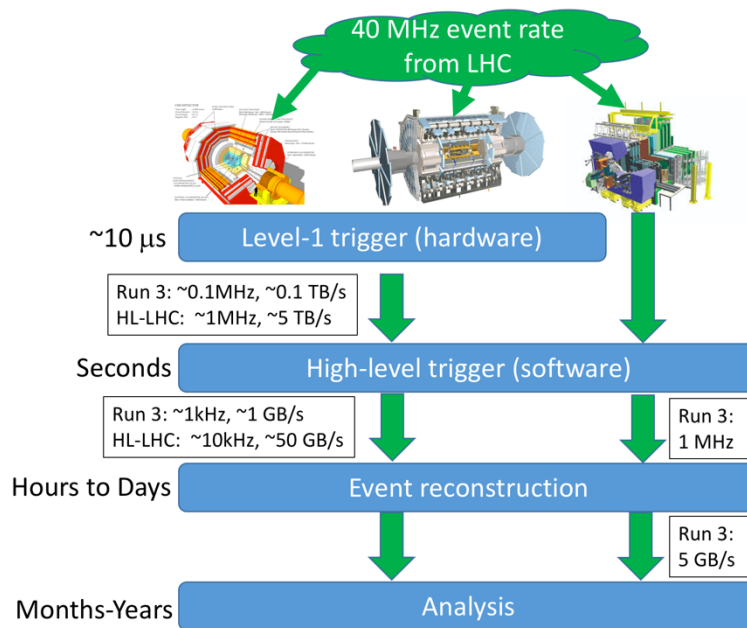


Figure 1: Data flow and data rates for Run 3 (2021-2023 LHC operations) and HL-LHC eras for the CMS and ATLAS experiments (left), and LHCb experiment (right).

This software development and maintenance represents a major undertaking by the collaboration. The LHC experiments each support millions of lines of code written and maintained by everyone from software professionals to graduate students. These codebases in turn rely on numerous other software packages that were created with the use cases of experimental high-energy physics as the primary motivator. Examples of these packages include ROOT [1], Geant4 [2], and numerous event generation packages. Each product contains functionality essential for the HEP scientific research community.

These proceedings focus on needs and requirements for LHC event reconstruction applications, which are a major part of the software and computing efforts in LHC experiments and the research and development towards HL-LHC [3]. These applications are responsible for turning raw detector data into fully processed data formatted for easy use by HEP researchers. This processing includes many facets, including: interpreting raw detector channel data using detector geometry and electronics channel mapping information; grouping detector hits and doing pattern recognition; track reconstruction and fitting; identification and classification of likely

physics object candidates (e.g., muons, jets, electrons, photons); deriving observables for the classification of these physics objects; and creating and storing data structures for use by analysts in condensed formats.

2. Evolution of Event Reconstruction at the LHC

The upgrade program of the LHC brings a significant new challenges to the authors of event-reconstruction algorithms. These include handling major increases in event rates, event complexity, and sophistication of detector design, while increasing performance and sensitivity to provide analysts with tools capable of probing the smaller and smaller signals which are of interest as data sets increase in size.

LHC experiments continuously use about 600k CPU cores and have around 400PB of data stored on disk and 600PB on tape [4]. Relative to current operations, HL-LHC brings a dramatic increase in event rates and event complexity: $\sim 4x$ increase in pileup and $\sim 10x$ higher projected input rates to the offline reconstruction [5,6,7] as illustrated in Figure 2. These expected operating conditions would require a large increase in resources if we simply scale up how LHC reconstruction algorithms and facilities work today. The trigger and event reconstruction application CPU needs increase significantly relative to other large components of the overall CPU budget for LHC due to the combinatorical nature of their algorithms.

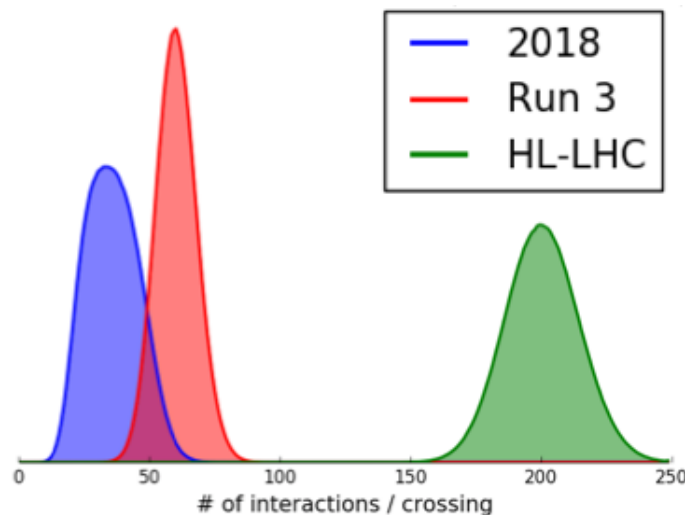


Figure 2: Current and expected pileup conditions in 2018, Run 3 and HL-LHC.

We see four primary challenges related to the HL-LHC detector, accelerator and analysis requirements. Specifically:

1. **Higher detector occupancy:** To cope with a more complex event environment (e.g., more interactions per bunch crossing), experiments have adopted more sophisticated detector technologies with a higher channel count. This evolution is essential to preserve the pattern recognition capabilities for HL-LHC. However, reconstruction algorithms will be faced with greater combinatorics that tend to drive CPU requirements.
2. **Trigger rate:** Higher rates are needed to preserve current physics reach given the increase in instantaneous luminosity. Online, this has meant that advanced capabilities will be used earlier in the processing chain, as well as the introduction of “real-time” analysis concepts. Examples

of these advancements include the use of tracking at first-level trigger (instead of limited to the second-level trigger), or even systems without a hardware trigger that rely on a purely software-based system to make trigger decisions on the full LHC event rate.

3. **Particle reconstruction:** As pileup increases, it becomes more and more difficult to separate detector hits corresponding to each unique particle in the event. This more challenging reconstruction problem means either a tradeoff in physics impact (e.g., a worsening of the achievable efficiency vs fake rate curve for identification) or in technical performance (e.g., increased CPU time to achieve the same physics performance), or both.
4. **Analysis sensitivity:** Searches for lower cross section processes demand higher precision and more robust approaches to data reconstruction. At the same time that the detection and reconstruction environment becomes more challenging, the goals of the HL-LHC physics program demand increased sensitivity. This is another driver for the more sophisticated detector designs for HL-LHC, but also an opportunity for algorithm improvements.

3. Software and Technology challenges

After nearly two decades of exploiting x86 hardware with Linux-based operating systems, the landscape of commodity computing hardware is evolving [8]. HEP does not drive the hardware market, so instead it must evolve to continue to derive sufficient event processing throughput per cost to enable our physics programs at reasonable computing cost. Market trends include more processing cores, as well as slower to access and more expensive memory. These trends are directly at odds with traditional HEP applications that typically exploit single threaded and memory-heavy algorithms. Another evolution is towards heterogeneous hardware systems exploiting low-power processing cores, or accelerators. An example is the increased focus on efficiently using high-performance computing (HPC) systems that typically include considerable graphics-processing unit (GPU) components.

As the timescale for experimental data taking and analysis increases, the needs of legacy code support will increase. Rewriting all of the software developed by the LHC experiments is not likely to be considered for HL-LHC. This means that pieces of the code base will be 15-30 years in age, implying an increased need for sustainable software development and investment in software education for experimental teams. The move towards open source software development and continuous integration systems brings a number of important opportunities to assist developers of software trigger and event reconstruction algorithms. Continuous integration systems have already brought the ability to automate code quality and performance checks, both for algorithm developers and code integration teams. While it is straightforward to test changes where no regression is expected, fully developed infrastructure for assisting developers in confirming the physics and technical performance of their algorithms during development is a work in progress.

4. Evolution in HEP frameworks and algorithms

Given the increased demands from both the experiment's requirements and from the evolution in computing technology, it is no surprise that software frameworks in all experiments are evolving to provide the needed flexibility to algorithm developers. Reconstruction applications in HEP already consist of many small kernels. These kernels together with the

independence of events create natural mechanisms for processing frameworks to leverage for parallelize. Specifically, LHC frameworks are evolving to allow multiple events in flight, multiple algorithms running on the same event, parallelism within algorithms, and offloading to accelerators and external resources to better take advantage of computing hardware. Threaded frameworks [9] are one approach to drastically reducing the memory requirement per computing core, which is important both due to the evolution in memory speed and to the increasing cost of RAM.

5. Evolution in HEP algorithms and event-reconstruction techniques

Next is the evolution of techniques and requirements to follow for reconstruction algorithm developers. The computing whitepaper (CWP) working group on event reconstruction and software trigger set out the following primary research and development areas to ensure that algorithms can perform at the technical performance levels needed for HL-LHC [10]:

1. **Enhanced vectorization programming techniques:** Modern CPUs have larger and larger vector units enabling simultaneous computations when the same instruction sequence can be exploited for multiple data inputs (SIMD). Challenges faced by HEP applications to make better use of vector units include that manipulation of small matrices (e.g. 5x5 matrices are common in tracking algorithms) mean that usual approaches to vectorizing matrix algebra do not work as well as in other applications; HEP algorithms typically rely on branch points which prevent vectorization; and the use of wide vector units (e.g., AVX512) often slows the clock frequency of the CPU. This last point means that a significant fraction of the computational kernel in HEP applications must be vectorized to benefit from its use. Localized changes do not give an immediate speed up. An example R&D project in this area is the MatrixPlex project recasts sets of small matrices to fill a vector unit (or accelerator) while operating on them in sync. It has been shown to provide significant speedups in track building [11].
2. **Algorithms and data structures to efficiently exploit many-core architectures:** Threaded applications are becoming common place in HEP. This has been driven in large part by the evolution towards more, but not faster (and thus lower power) processing cores, and by the evolution of CPU memory characteristics. Memory access speeds have not improved at the same rate as for processors. This means there is a larger cost for memory access. Together with increases in fast on-chip cache memory, memoy-local algorithms are of increasing importance (avoiding large latency for cache misses). There are potentially large gains for using more data aware programming methods, such as exploiting structures of arrays rather than arrays of structures.
3. **Algorithms and data structures for non-x86 computing architectures (eg, GPUs or FPGAs).** Accelerators have become notably easier to use and much more common in across data science in reach years. Algorithmic issues that must be addressed for their effective use include: managing data movement; achieving massive parallelism within local algorithms, independence from thread ordering in algorithms; avoiding unneeded data transfers and transformations; and simple data formats optimized for parallel memory access. These issues are issues that affect performance on CPUs (as described in items 1 and 2), however are even more important when using accelerators. An example project is describe in Ref. [12].

4. **Enhanced QA/QC for reconstruction techniques:** Reducing the rate at which HEP applications waste resources due to incorrect results is another approach to reducing the computing resources needed. Mistakes are increasingly hard to find in complex applications, and have a potentially large consequence to both the amount of computing resources needed, but also the time at which data is available to the analysis community. Improved and more automated quality assurance and control processes would reduce the rate of bugs and even sub-optimal physics performance issues that enter HEP applications.
5. **Real-time analysis:** HL-LHC experiments will operate in a regime where interesting “signal” events dominate data rates. This is clear from the move away from traditional hardware triggers, and towards ideas such as real-time analysis that allow a much increased rate of events to be saved for analysis. Techniques include saving an already processed and very reduced output that allows many more events to be saved within a given offline resource envelope. The idea is to perform the first steps of analysis directly on the output of the trigger rather than a truly offline reconstruction [13,14,15]. By dropping the raw detector data, a large factor in data reduction is achieved, however, all stages of processing must be ready from the start of data taking. This includes both software and other inputs such as calibrations, and represents a large change from traditional HEP and LHC data processing techniques.
6. **High precision physics-object reconstruction, identification and measurement techniques:** As discussed above, HL-LHC algorithms need to take advantage of new detection capabilities to determine properties of physics-object candidates of sufficient quality, efficiency, and purity for HL-LHC analysis requirements. Beyond incorporating the new detector technologies themselves, new measurement techniques may provide considerable algorithmic improvements. Machine learning techniques are a currently topical example [16,17]. HEP has used neural networks and other machine learning techniques for a long time, but recent advancements have brought new capabilities for dealing with noisy data. In an extreme vision of algorithm developments, training deep learning networks could become the primary activity of HEP researchers.
7. **Fast software trigger and reconstruction algorithms for high-density environments:** Algorithm developers must also examine ways to improve implementations and tunings for the pileup environment expected during HL-LHC.

6. Conclusion

We should expect HL-LHC reconstruction applications and their usage to look very different from traditional HEP or even LHC Run 2 applications. The community is challenged by the opportunity to enable the vast physics program of the HL-LHC, and must adapt to the corresponding analysis requirements, the detector evolution, and increased event complexity. In most aspects, the best approach is still unknown, so it is an exciting time to work on new ideas and new concepts. This need for innovation also indicates that HEP researchers must be trained adequately in both algorithmic approaches and technical implementations to develop the needed algorithms for HL-LHC.

The authors contribution to this work was supported by the National Science Foundation under grants ACI-1450377 and PHY-1624356.

References

- [1] R. Brun and F. Rademakers, *ROOT – An Object Oriented Data Analysis Framework*, NIM in Phys. Res. A389 (1997) 81-86 [[DOI: 10.1016/S0168-9002\(97\)00048-X](https://doi.org/10.1016/S0168-9002(97)00048-X)]. See also <http://root.cern.ch/>.
- [2] S. Agostinelli, et al., Geant4 – a simulation toolkit, NIM A503 (2003) 250 [[DOI:10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)]. J. Allison, et. al., *Recent developments in Geant4*, NIM A835 (2016) 186 [[DOI:10.1016/j.nima.2016.06.125](https://doi.org/10.1016/j.nima.2016.06.125)].
- [3] A. A. Alves Jr et al., *A Roadmap for HEP Software and Computing R&D for the 2020s*, 2017 [[arXiv:1712.06982](https://arxiv.org/abs/1712.06982)].
- [4] WLCG Resource, Balance and Usage (REBUS) summary for 2018 [[Pledged resources to LHC experiments](#)]
- [5] CMS Collaboration, *Technical Proposal for the Phase-II Upgrade of the CMS Detector*, CERN-LHCC-2015-010, LHCC-P-008, CMS-TDR-15-02 (2015) [[CDS:2020886](https://cds.cern.ch/record/2020886)].
- [6] ATLAS Collaboration, *ATLAS Phase-II Upgrade Scoping Document*, CERN-LHCC-2015-020, LHCC-G-166 (2015) [[CDS:2055248](https://cds.cern.ch/record/2055248)].
- [7] LHCb Collaboration, *Expression of Interest for a Phase-II LHCb Upgrade: Opportunities in flavour physics, and beyond, in the HL-LHC era*, CERN-LHCC-2017-003 (2017) [[CDS:2244311](https://cds.cern.ch/record/2244311)].
- [8] I. Bird, et al., *Upgrade of the Computing Models of the WLCG and the LHC experiments*, CERN-LHCC-2014-014 (2014) [[CDS:1695401](https://cds.cern.ch/record/1695401)].
- [9] E. Sexton-Kennedy, et al., *Implementation of a Multi-threaded Framework for Large-scale Scientific Applications*, J. Phys.: Conf. Ser. 608 (2015) 012034 [[IOPP](https://arxiv.org/abs/1507.08001)].
- [10] J. Albrecht, et al., *HEP Community White Paper on software trigger and event reconstruction*, 2018 [[arXiv:1802.08640](https://arxiv.org/abs/1802.08640)].
- [11] G. Cerati, et al., *Parallelized Kalman-Filter-Based Reconstruction of Particle Tracks on Many-Core Processors and GPUs*, EPJ Web Conf 150 (2017) [[CTD/WIT 2017](https://arxiv.org/abs/1703.09441)].
- [12] F. Pantaleo, *Patatrack*, [[CHEP 2018](https://arxiv.org/abs/1803.09441)]. <https://patatrack.web.cern.ch/patatrack/>.
- [13] R. Aaij, et al., *Tesla : an application for real-time data analysis in High Energy Physics*, Comput. Phys. Commun. 208 (2016) 35 [[CDS:2147693](https://cds.cern.ch/record/2147693)].
- [14] ATLAS Collaboration, *Trigger-object Level Analysis with the ATLAS detector at the Large Hadron Collider: summary and perspectives*, ATL-DAQ-PUB-2017-003 (2017) [[CDS:2295739](https://cds.cern.ch/record/2295739)].
- [15] CMS Collaboration, *Search for narrow resonances in dijet final states at $s = 8$ TeV with the novel CMS technique of data scouting*, Phys. Rev. Lett. 117 (2016) 117 [[CDS:2149625](https://cds.cern.ch/record/2149625)].
- [16] S. Farrell, et al., *The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking*, EPJ Web Conf. 150 (2017) 00003 [[CTD/WIT 2017](https://arxiv.org/abs/1703.09441)].
- [17] K. Albertsson, et al., *Machine Learning in High Energy Physics Community White Paper*, [[arXiv:1807.02876](https://arxiv.org/abs/1807.02876)].