HEP Software: Overview and Future Challenges

Heather M. Gray

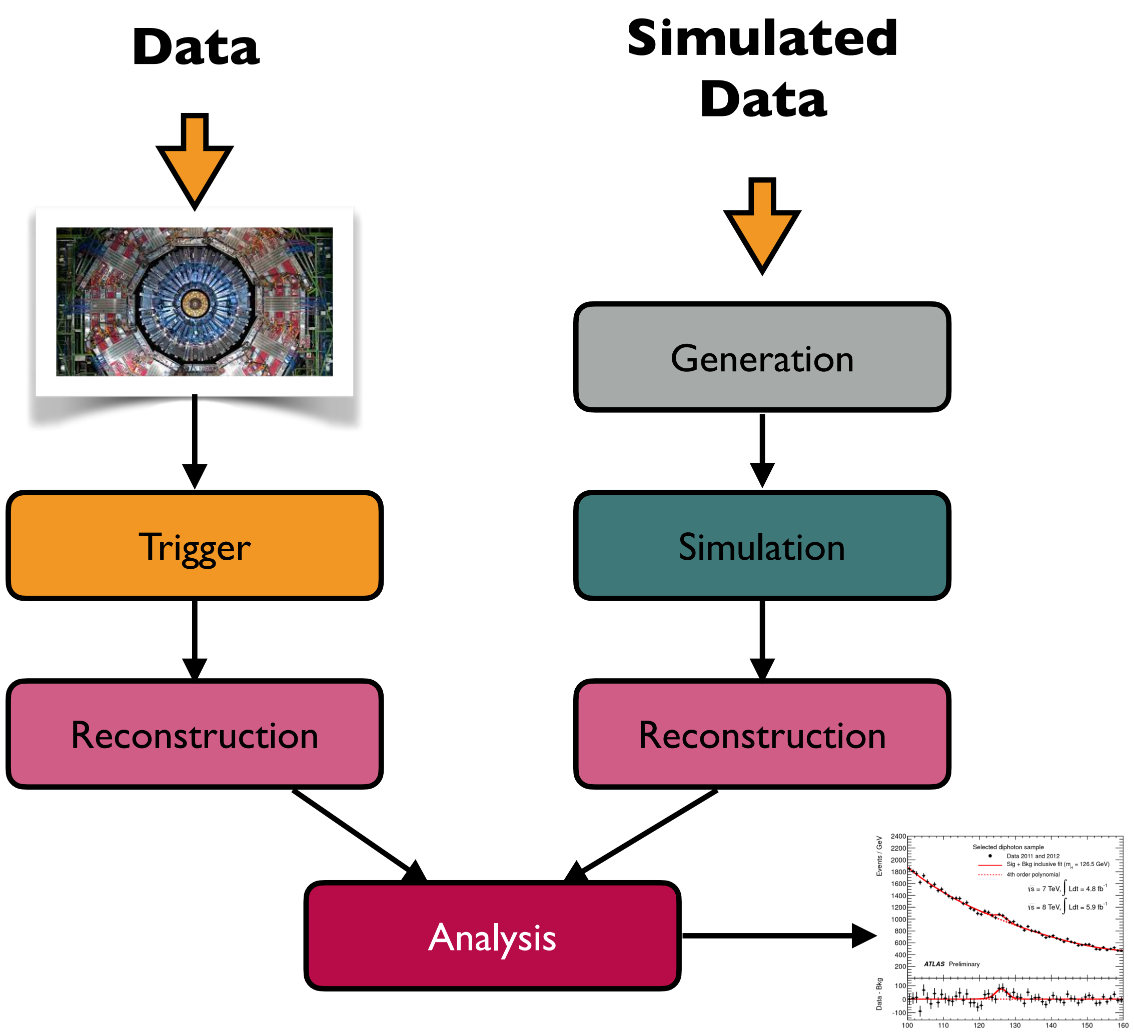
Berkeley
UNIVERSITY OF CALIFORNIA

BERKELEY LAB

Introduction

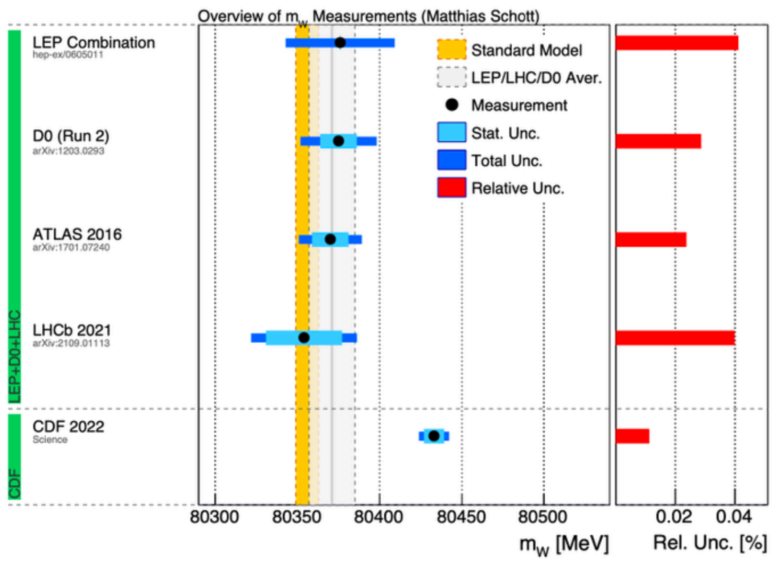
- Software is used ever increasingly in high-energy physics during **every step** of the **data processing chain**
  - From detector control, through trigger, to reconstruction and analysis
- The **code base** is enormous
  - ~50M lines of C++
  - Also large (but size) unknown python code base

Typical data processing chain at the LHC

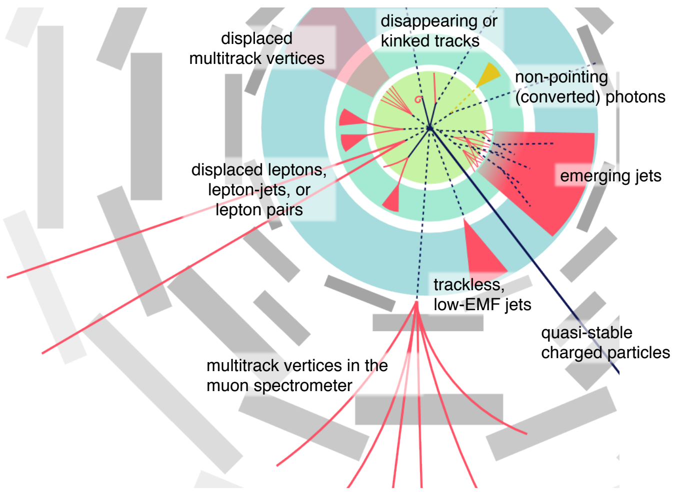


Characteristics and Challenges

- **Characteristics** of HEP experiments over the next decade
  - Increasingly sophisticated detectors, increased event data volume
  - Higher data rates
  - Increasing demands in physics precision
  - Need to explore unconventional signatures
- **Challenges/Opportunities**
  - Technology evolution
    - Increased concurrency
    - Increasingly diverse architectures
  - Machine learning
  - Data science, including python for scientific computing
  - Open Source Software
  - Funding constraints

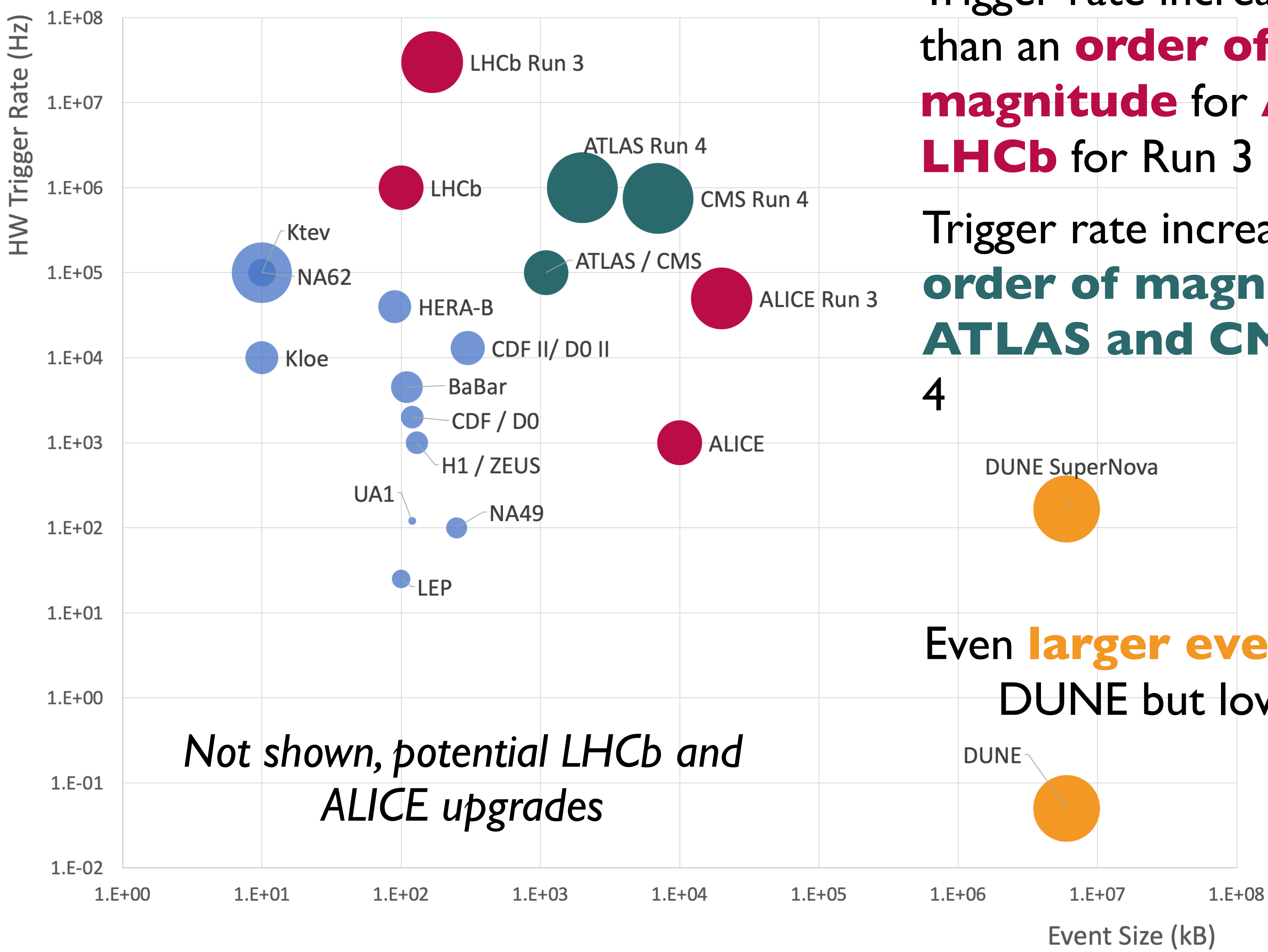


M. Schott



H. Russell

HEP Event Rates and Sizes



Trigger rate increases by more than an **order of magnitude** for **ALICE and LHCb** for Run 3

Trigger rate increase by an **order of magnitude** for **ATLAS and CMS** for Run 4

Even **larger event** sizes for DUNE but lower rate

Not shown, potential LHCb and ALICE upgrades

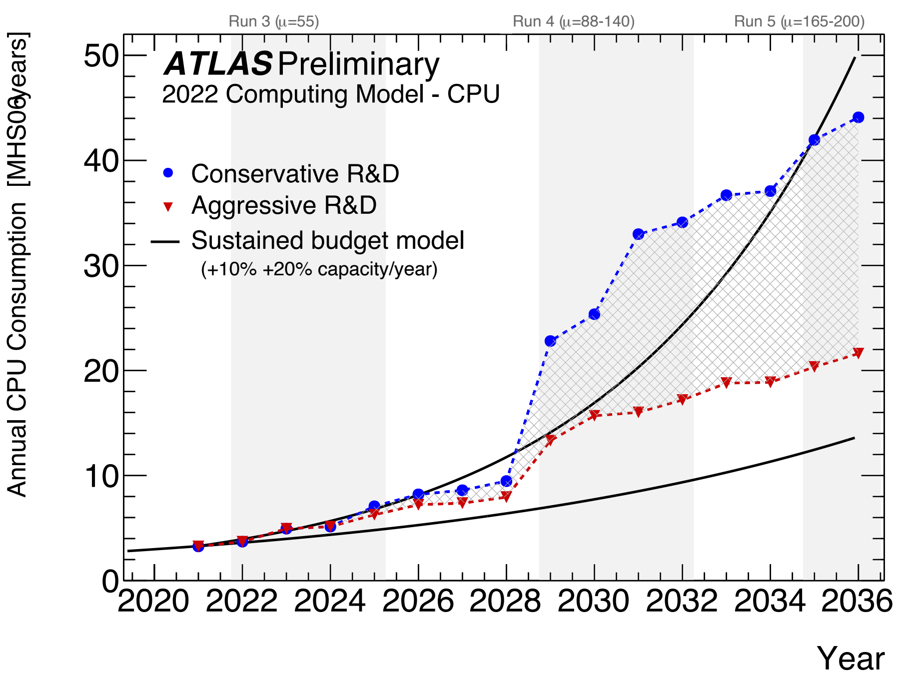
A. Cerri

Looking ahead to Run 4 (HL-LHC)

- For ATLAS and CMS we expect
  - **5-7x** increase in luminosity (LHC upgrade)
  - **4-5x** increase in event size (new detectors)
  - **10x** increase in event rate (trigger upgrade)
- However, **flat computing budgets** mean that **new techniques** and **new ideas** are required

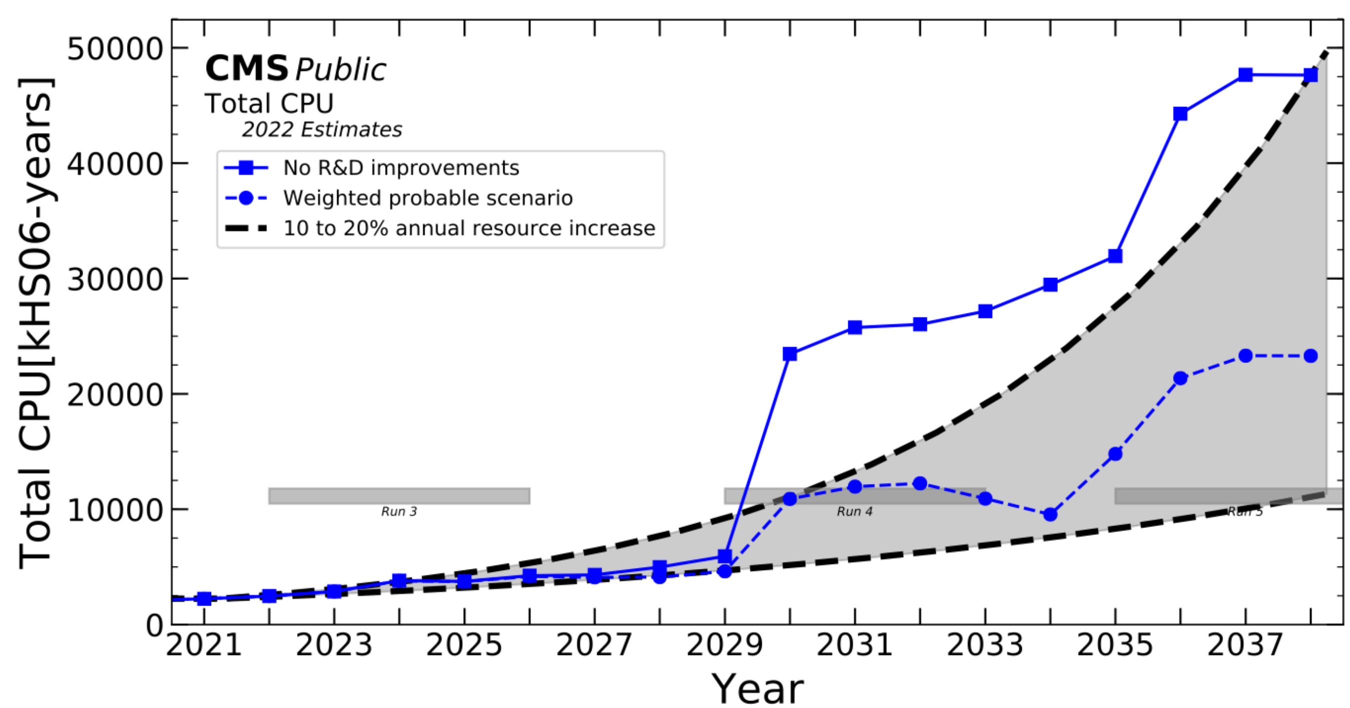
Similar results for disk and tape

CPU, ATLAS



ATLAS S&C Public Results

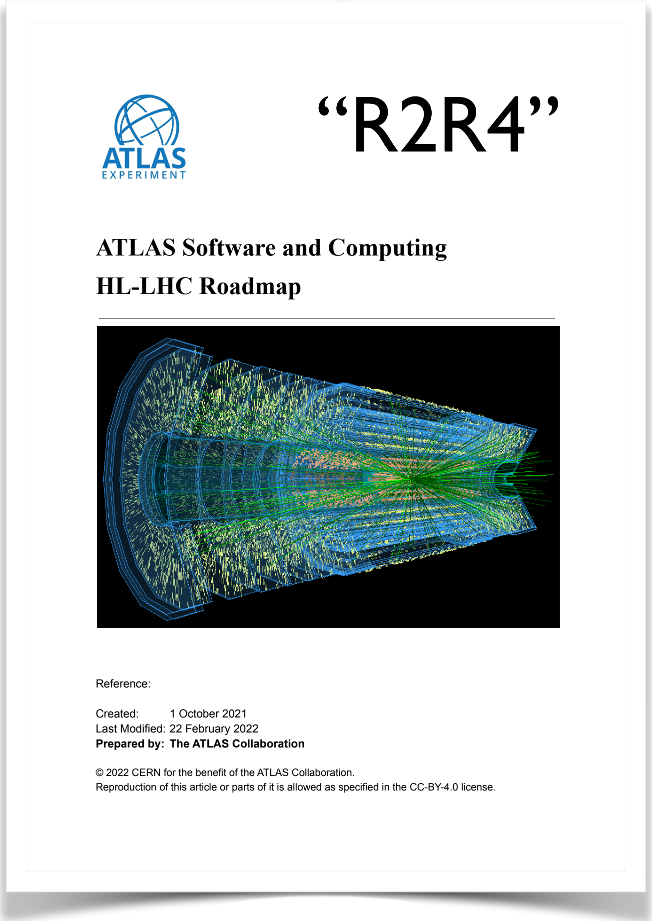
CPU, CMS

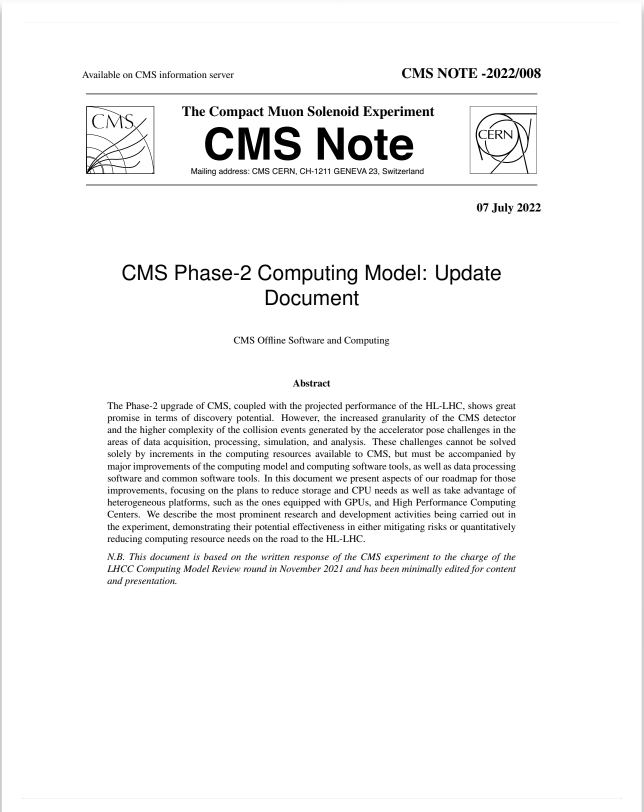


CMS O&C Public Results

Project Management towards Run 4

- ATLAS and CMS have undergone an extensive **planning process** to identify the software development needs towards HL-LHC
  - ATLAS HL-LHC Conceptual Design Report (2020)
  - Evolution of the CMS Computing Model towards Phase-2 (2021)
  - November LHCC Review
    - ATLAS Software and Computing HL-LHC Roadmap
    - CMS Phase-2 Computing Model: Update Document
- ATLAS and CMS have defined a set of **projects** and **milestones** that are tracked and reviewed regularly





Available on CMS information server — CMS NOTE -2022/008

The Compact Muon Solenoid Experiment

CMS Note

07 July 2022

CMS Phase-2 Computing Model: Update Document

CMS Offline Software and Computing

Abstract

The Phase-2 upgrade of CMS, coupled with the projected performance of the HL-LHC, shows great promise in terms of discovery potential. However, the increased granularity of the CMS detector and the higher complexity of the collision events generated by the accelerator pose challenges in the areas of data acquisition, processing, simulation, and analysis. These challenges cannot be solved solely by increments in the computing resources available to CMS, but must be accompanied by major improvements of the computing model and computing software tools, as well as data processing software and common software tools. In this document we present aspects of our roadmap for those improvements, focusing on the plans to reduce storage and CPU needs as well as take advantage of heterogeneous platforms, such as the ones equipped with GPUs, and High Performance Computing Centers. We describe the most prominent research and development activities being carried out in the experiment, demonstrating their potential effectiveness in either mitigating risks or quantitatively reducing computing resource needs on the road to the HL-LHC.

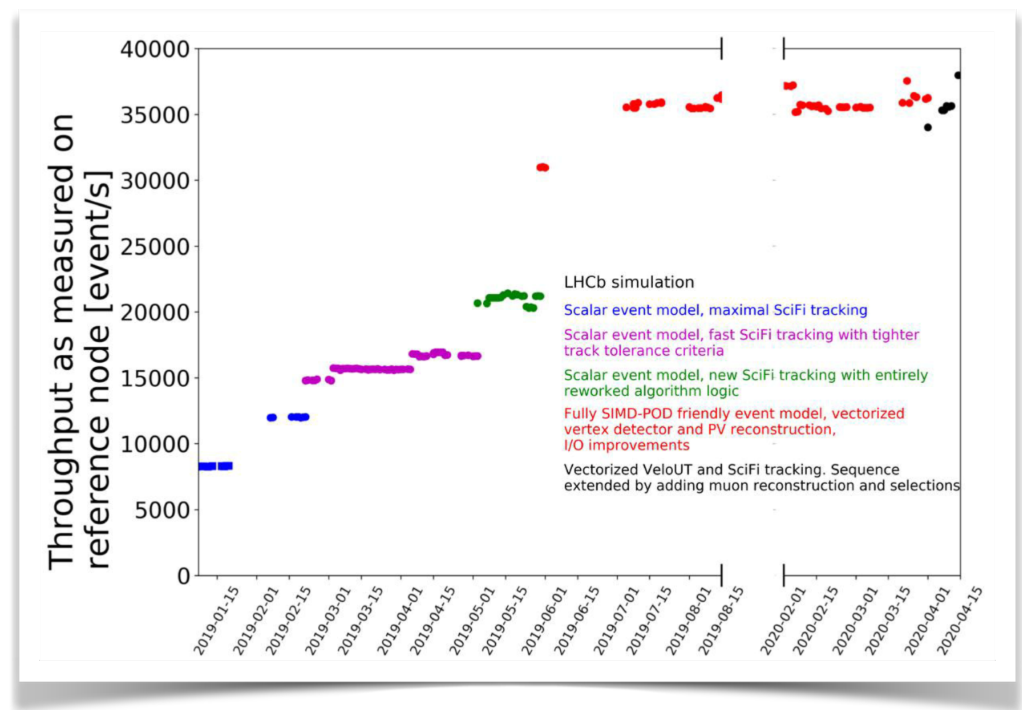
N.B. This document is based on the written response of the CMS experiment to the charge of the LHCC Computing Model Review round in November 2021 and has been minimally edited for content and presentation.

Code Optimization/Software Modernization

Code Optimization and Modernization

- Optimization and modernization bring significant CPU gains
  - **vectorization**, multi-threading, memory architecture and allocation
- CMS continues to use **multithreaded** applications extensively in reconstruction
  - Multicore generation, simulation, digitization/pile up mixing, reconstruction, creation of analysis formats (8 threaded jobs on the GRID)
- ALICE can perform simulation with parallel processing of **sub-events**
  - Exploit opportunistic HPC resources

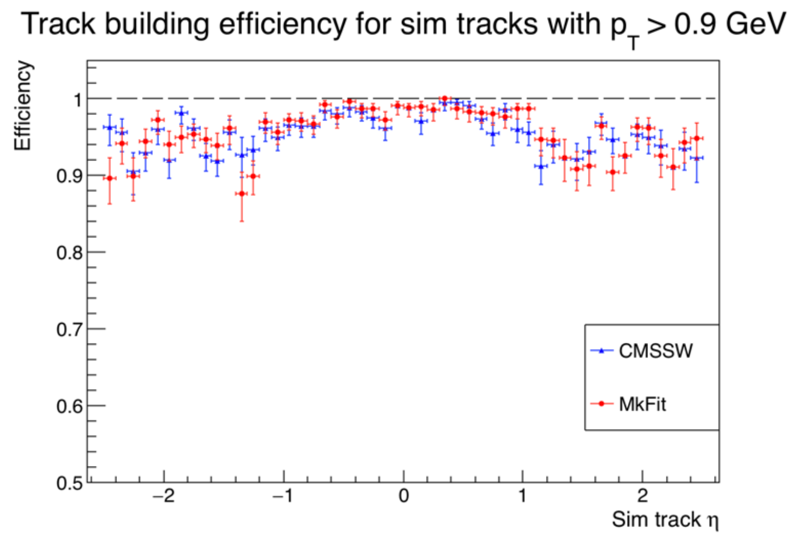
- Through optimization the ATLAS **Geant4 code** has been sped up by 30%

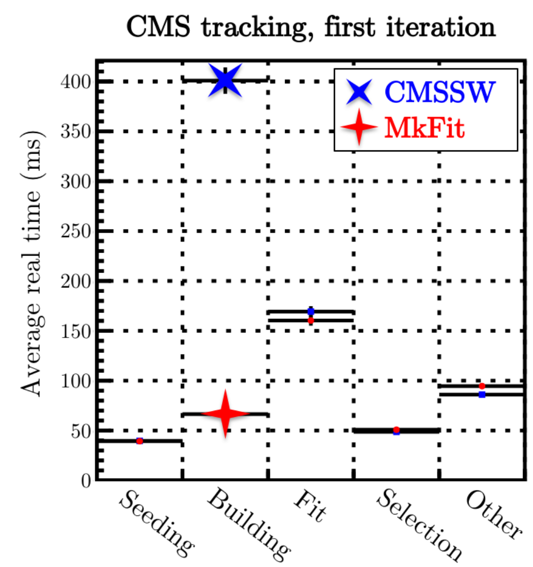


LHCb Tracking ~4x

Tracking Modernization

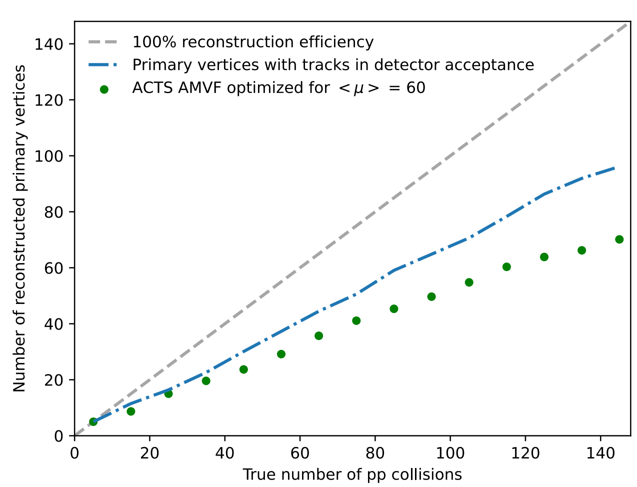
- Complete revamp for CMS tracking for Run 3
  - New CMS tracking (**mkFit**) reduced time needed for **tracking by 20%** and the full **reconstruction by 10%**

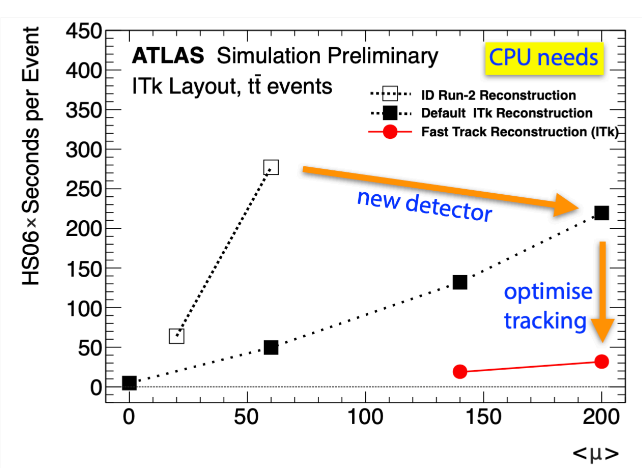




- **ACTS** is an experiment-independent toolkit for track reconstruction in implemented in modern C++
  - **Components of ACTS** will be used by ATLAS for Run 3, but full integration targets Run 4

Ai et al, arXiv:2106.13593



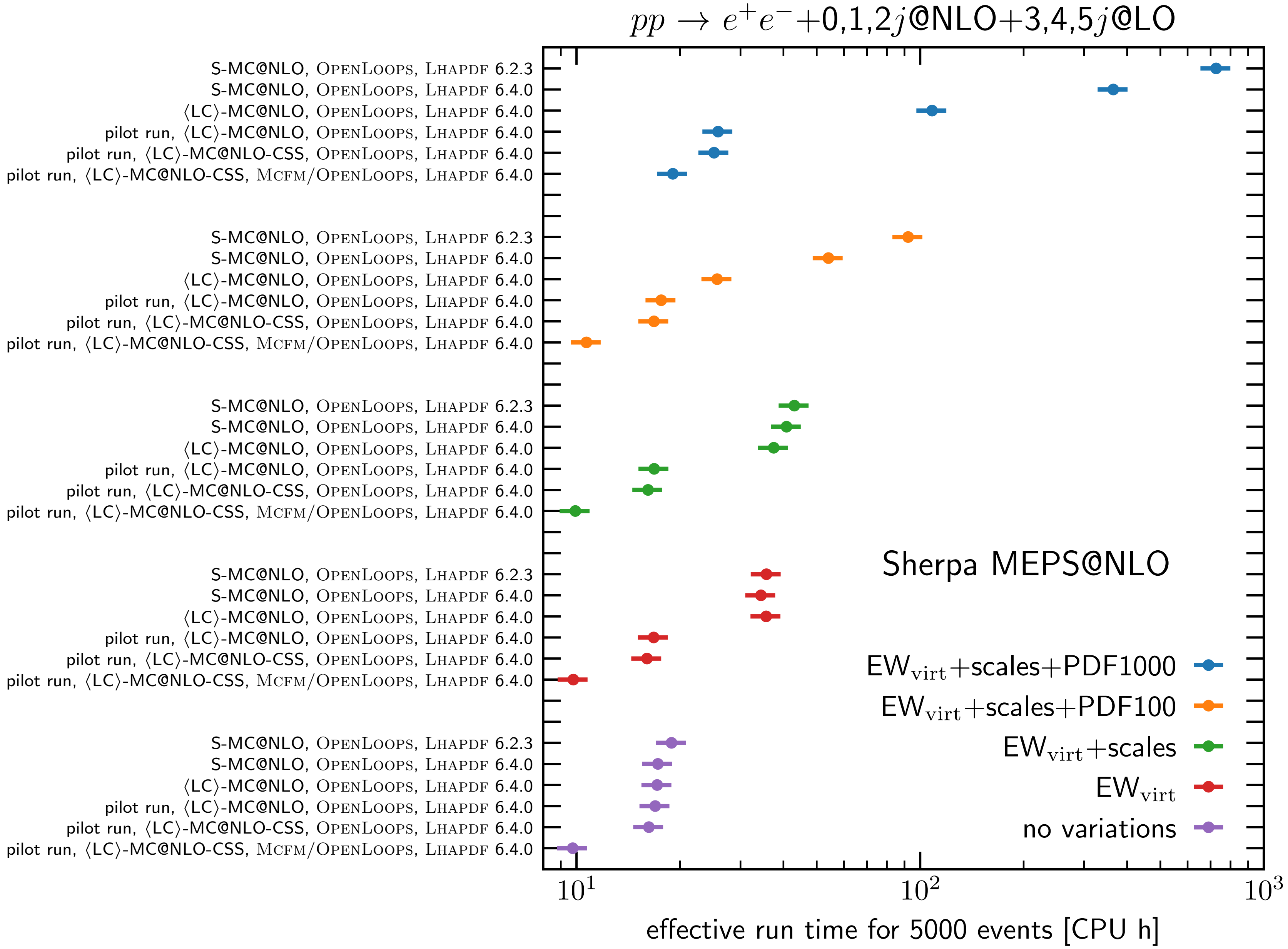


ATLAS Tracking

~9x

Generator Optimization

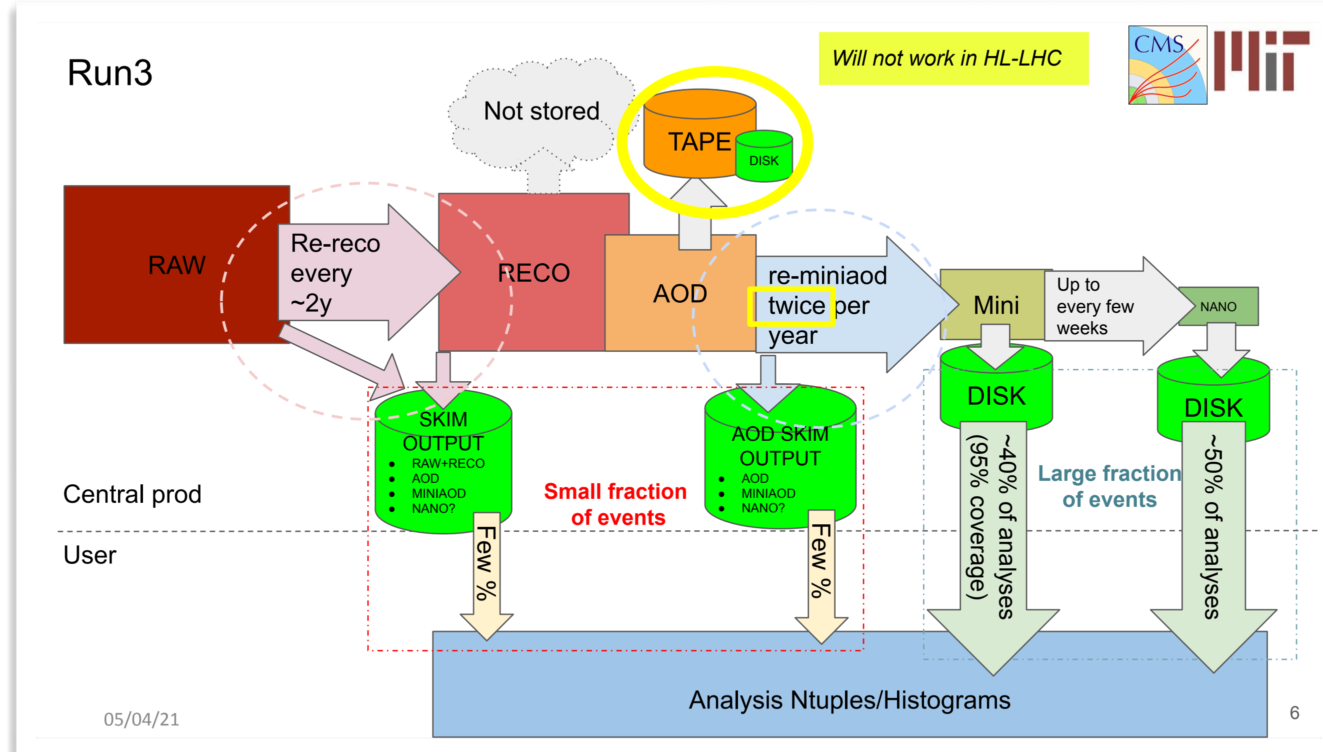
- Example: Optimization of Sherpa brings approximately **order of magnitude** improvement for V+jets for ATLAS
  - LHAPDF optimization (~5x), weight optimization (~5x)



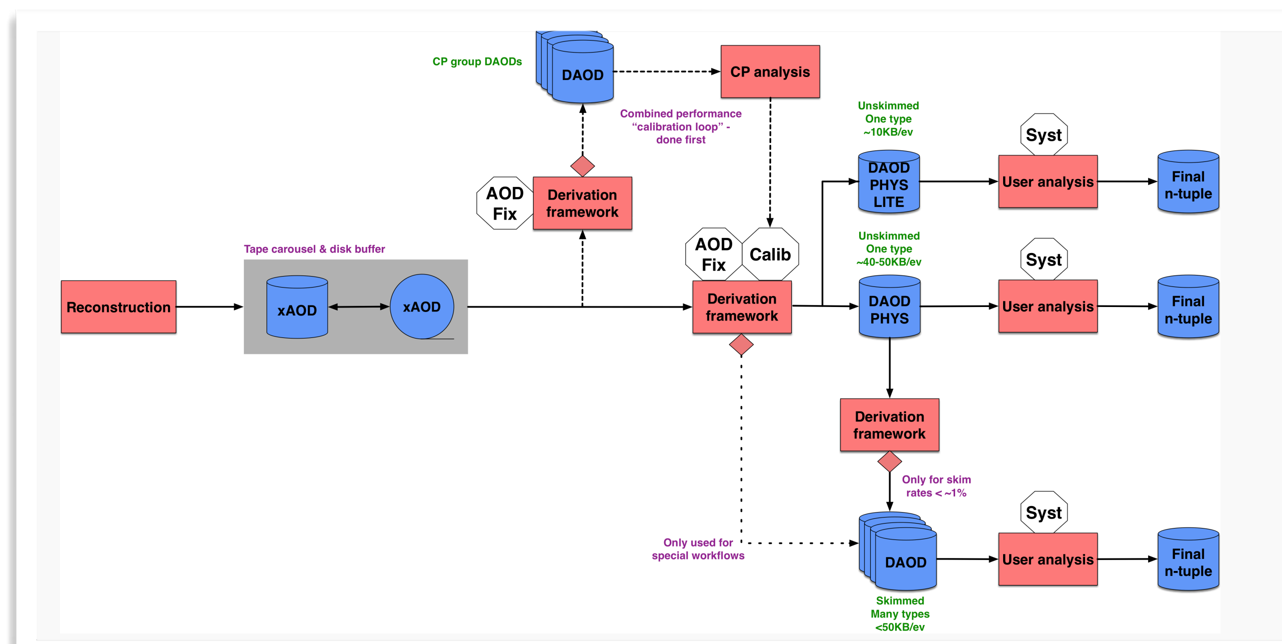
E. Bothmann et al

Analysis Models

- Many **competing factors** keep analysis models fluid
- Flexibility and ease of analysis vs computing needs
- Tiny formats for specific analyses vs shared formats for multiple analyses
- Framework readable vs laptop readable
- High-level physics analysis objects vs novel techniques
- Stability and consistency (e.g. reanalyse old datasets)
- Analysis precision vs disk space (including detailed systematics)
- Lossy compression



M. D'Alfonso

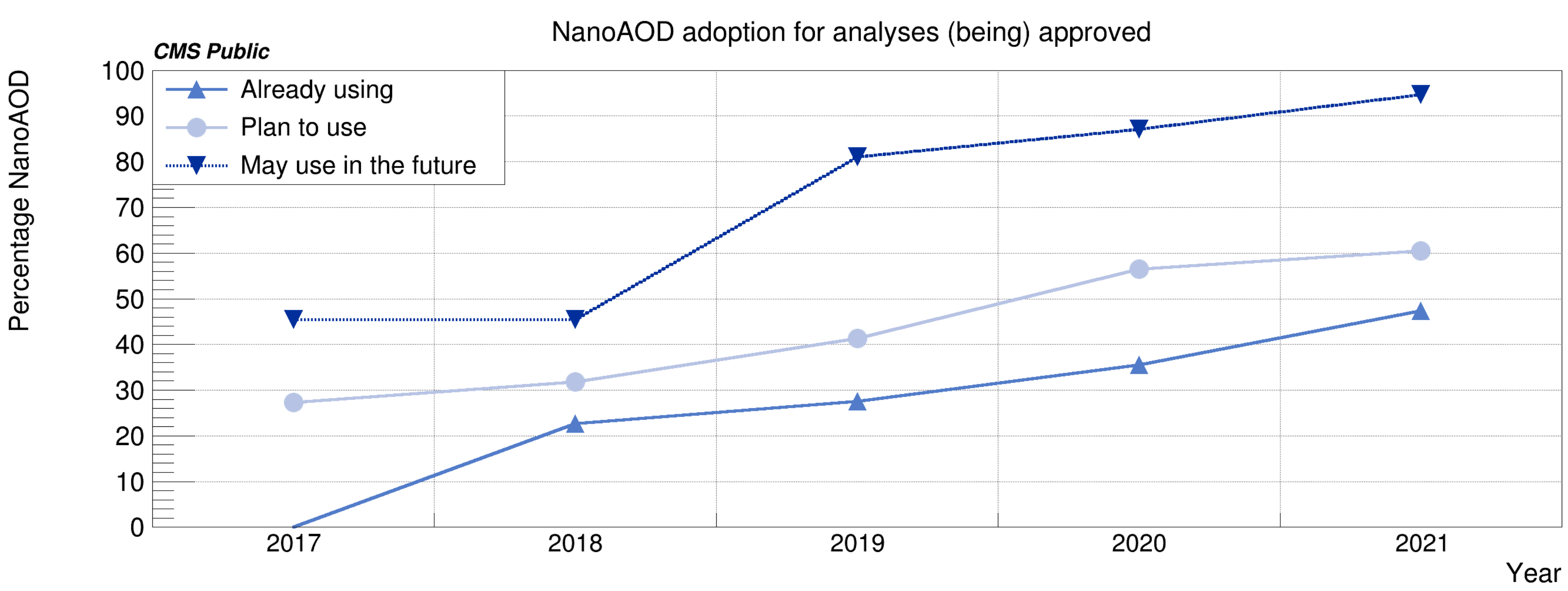


J. Elmsheuer

Analysis Model Mini-Workshop

Recent improvements to analysis models

- **Centralized** analysis production
  - e.g. analysis trains (D0/ATLAS/ALICE) or DIRAC (LHCb) transformation system
- **LHCb** has centralized skimming and trimming (sprucing)
- **CMS** a NanoAOD (1-2 kB/event) for ~50% of analyses
- **ALICE** has a highly optimized AOD format based on tables
  - Declarative analysis
  - Complexity hidden from users
- **ATLAS** is introducing DAOD_PHYS and DAOD_PHYSLITE (10 kB/event)



CMS Offline Computing Results

Common Software

Common Software R&D Institutes

- HEP experiments at the LHC and beyond face **similar changes**
  - Formation of the **HEP Software Foundation (HSF)** in 2015
    - Provides a common forum for software for HEP experiments
- Funded R&D efforts in **common software** in a number of countries
  - Examples are listed on the next page
- Activity encouraged by the European Strategy
  - "[…] vigorously pursue **common**, coordinated R&D efforts […], to develop **software** […] that exploit the recent advances in information technology and data science […]"
- Common projects can aid software maintainability
  - More likely to have a pool of people available for maintenance

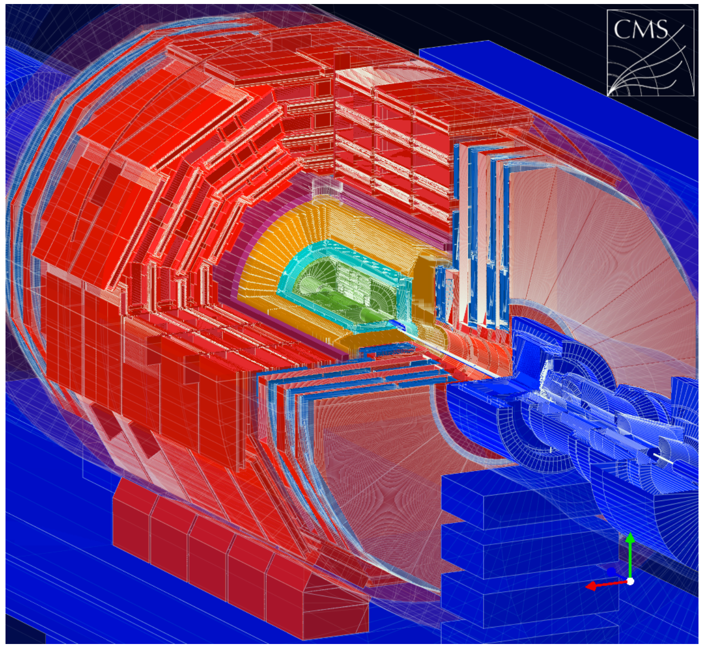
Examples of Software Institutes

- IRIS-HEP, NSF, 2018
  - Analysis systems, innovative algorithms, DOMA, training
- ErUM-DATA, Helmholtz Institute, Germany
  - Heterogeneous computing and virtualized environments, machine learning for reconstruction and simulation
- EP R&D, CERN, Switzerland, 2020
  - Turnkey software systems, faster simulation, track and calo reconstruction, efficient analysis
- HEP-CCE, DOE, USA, 2019
  - Portable Parallelization Strategies, I/O Strategy on HPC, Event generators
- AIDAInnova, European Commission EU, 2021
  - Turnkey software, track reconstruction, particle flow, ML simulation
- SWIFT-HEP STFC, 2021 and ExCALIBUR-HEP, 2020, UKRI UK
  - Exascale data management, Event generators, detector simulation on GPUs, FPGA tracking for HLT

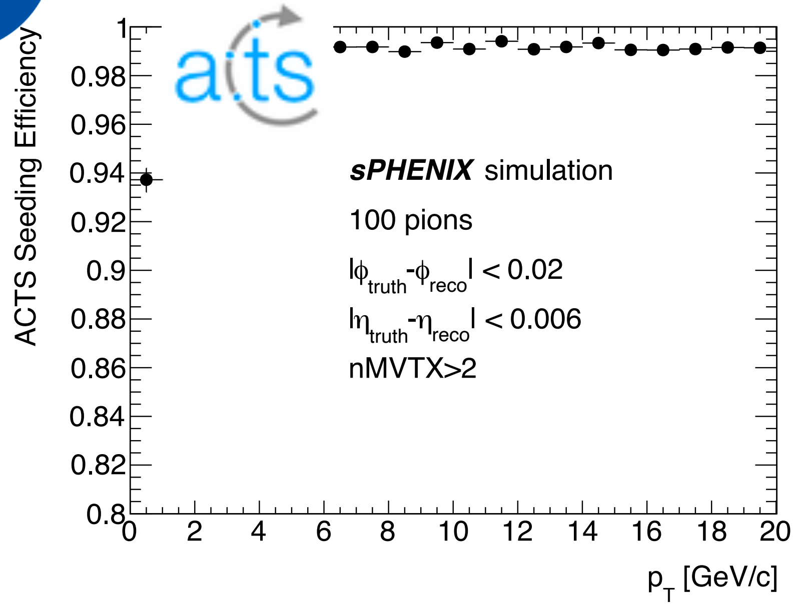
Slide Credit: G. Stewart

Software for Multiple Experiments

- **Common packages** have been used extensively by many experiments over many years including CLHEP, ROOT, Geant4, GAUDI
- For Run-3, ALICE uses **ALFA**, framework developed with GSI (FAIR) as common integration platform for online/offline processing
  - Online reconstruction using heterogeneous farm
  - Enables parallel data processing
- **DD4HEP** is now used by CMS, LHCb among other experiments for the detector description
- **ACTS** has origins in ATLAS tracking software, but currently being explored by different experiments
- LHCb is splitting of **Gaussino** as experiment-independent part of Gauss simulation framework (w. CERN SFT/FCC)



Vuosalo et al.

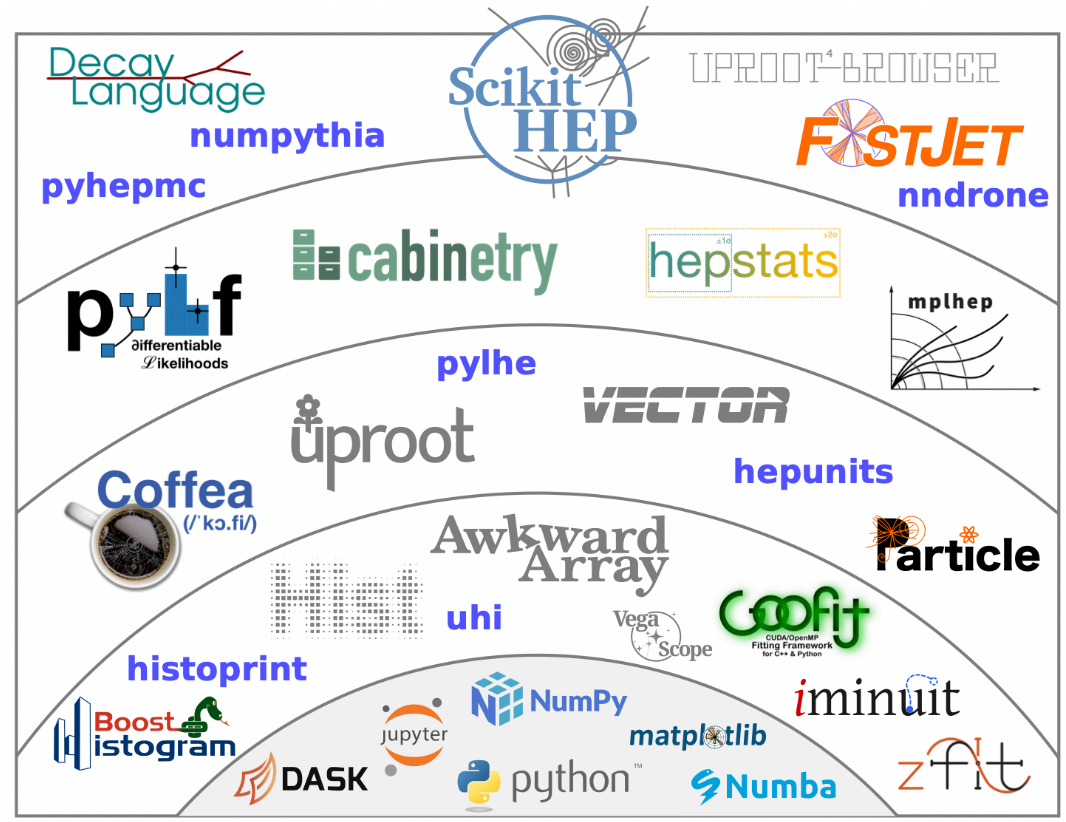


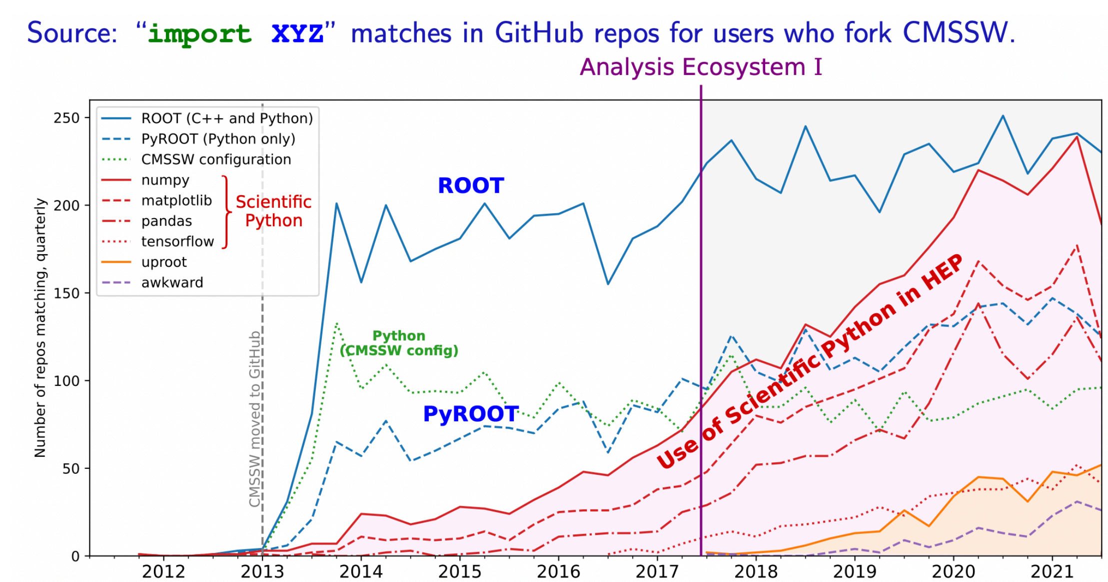
Osborn et al, arXiv:2103.06703

Python for Analysis

- Ongoing **boom** in the field of data science
- **Python** has become the language of choice for data science applications
  - Huge community has developed well-documented tools
  - numpy, matplotlib, pytorch, tensorflow, etc
- Balanced against our **own** designed-to-purpose and customized tools, in particular, ROOT
- **Python** is becoming increasing popular for analysis especially amongst the younger members of our community

ROOT Data Analysis Framework



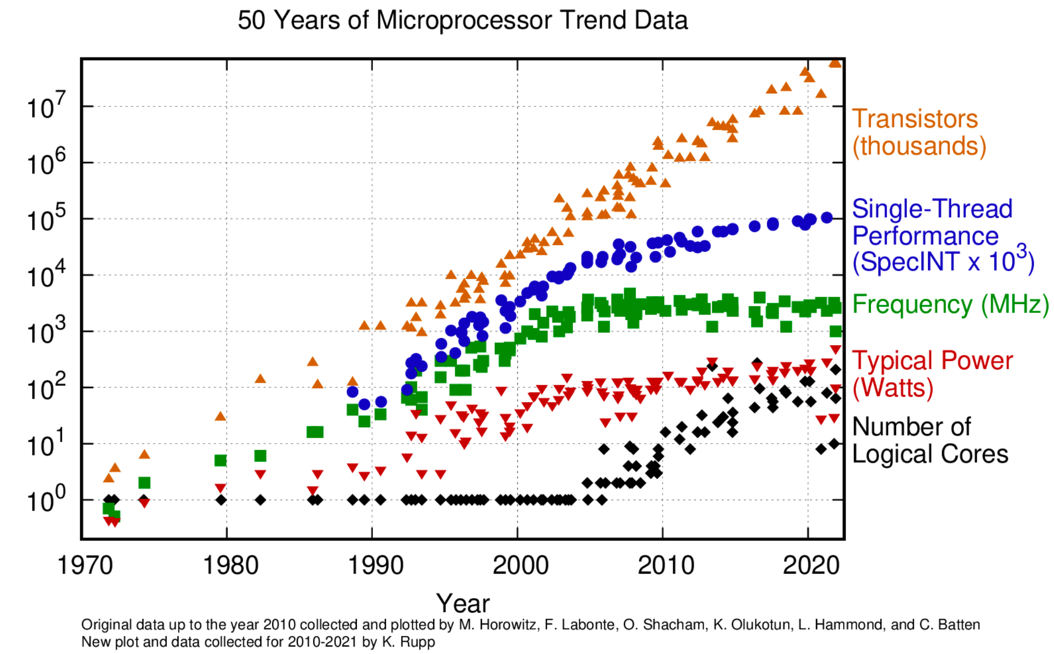


J. Pivarski

Hardware Evolution

Hardware Evolution

- **Transistor density** still increasing according to Moore's Law
  - since ~2006 increase due to **more cores** rather than increased chip clock speed
- Increasingly **diverse** set of computer architectures, e.g.
  - Graphical Processing Units (**GPUs**)
  - Field Programmable Gate Arrays (**FPGAs**)
  - Tensor Processing Units (**TPUs**)
- Require **parallel programming**
  - Extensive set of R&D projects
- GPUs and FPGAs can be used either to run a full **standalone** application or to **offload** specific applications
- **GPU** or **FPGA** clusters **as a service** for deep learning training or inference has been explored by a number of groups
  - Krupa et al, Duarte et al, Rankin et al, Wang et al



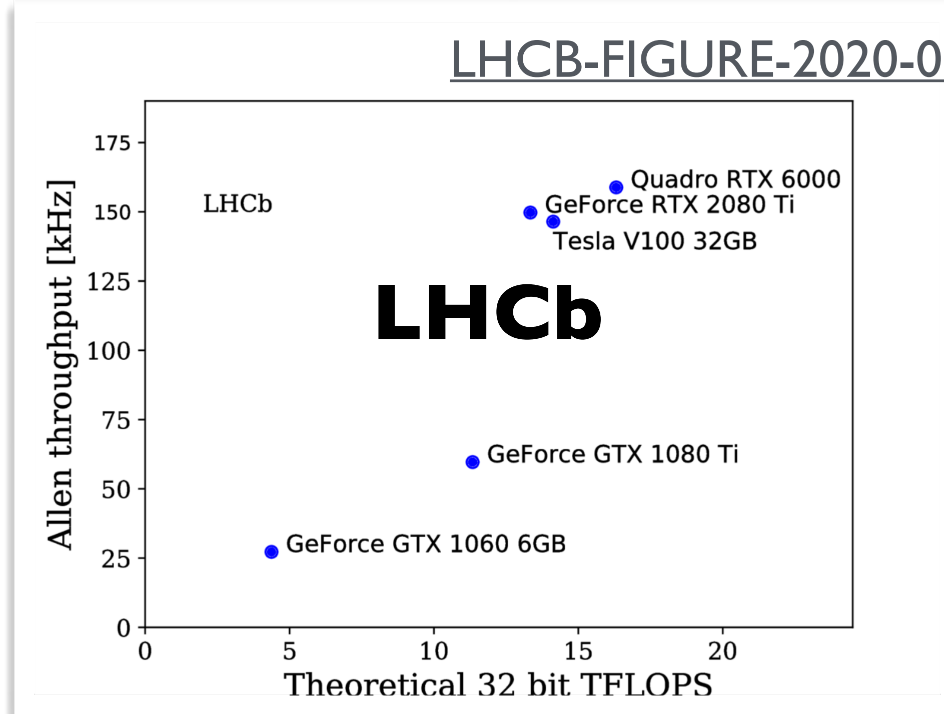
K. Rupp

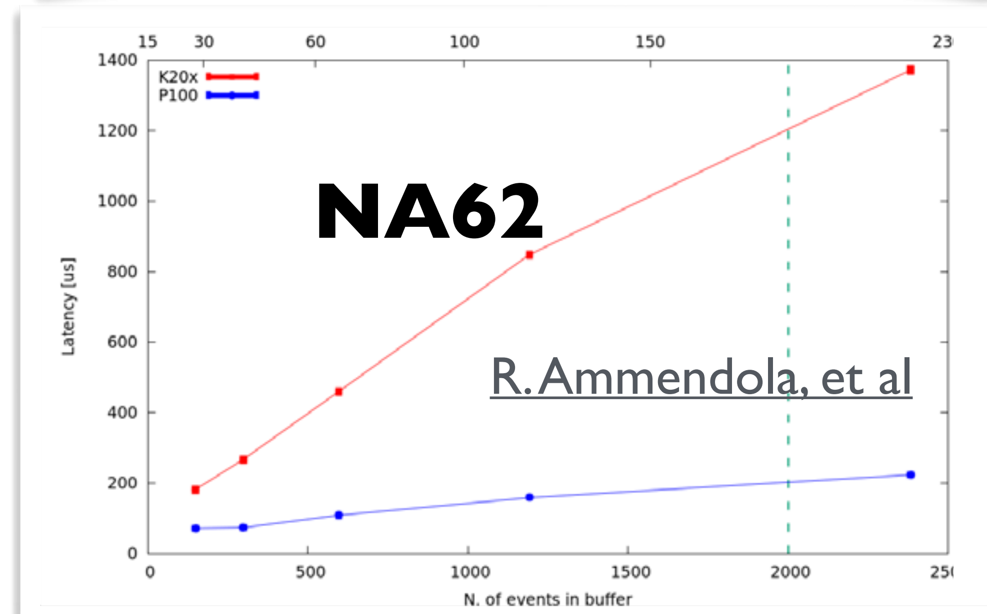
Image Source

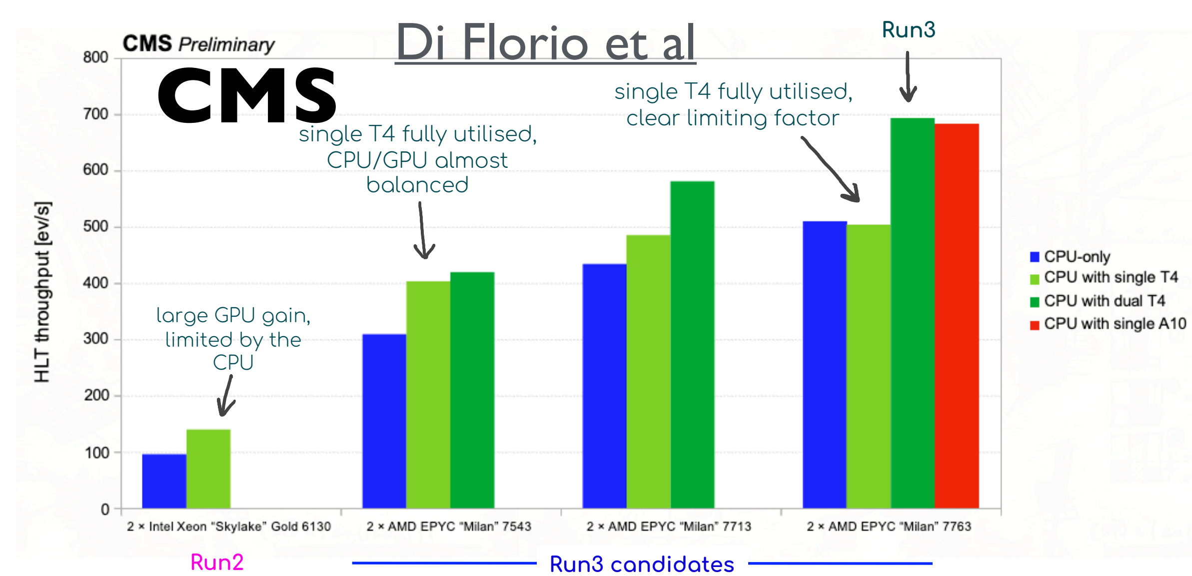
Trigger Applications

- LHCb's HLT1 is a **fully software trigger** that performs partial event reconstruction, in particular tracking, with the **Allen framework*** using GPUs
- Processes events at **30 MHz** on **<500 GPUs** with **improved physics** performance
    - 40 Tbit/s ➡ 1-2 Tbits/s
- **NA62** uses GPUs in their **trigger** relying on an FPGA-based interface card to directly transfer data between CPU and GPU (NaNeT)
- CMS has offloaded **30% of HLT sequence** to GPUs (NVidia T4)
    - Ongoing work to offload offline reconstruction to GPUs

*general framework for heterogeneous computing

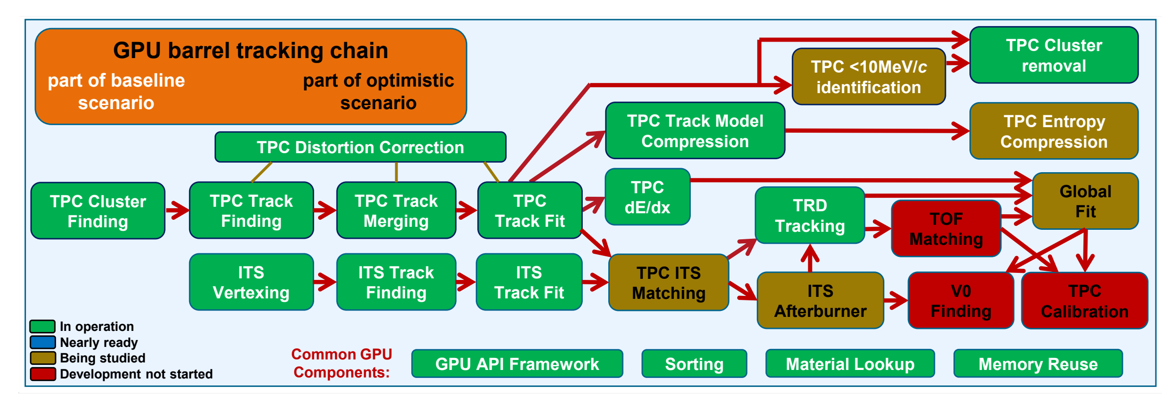




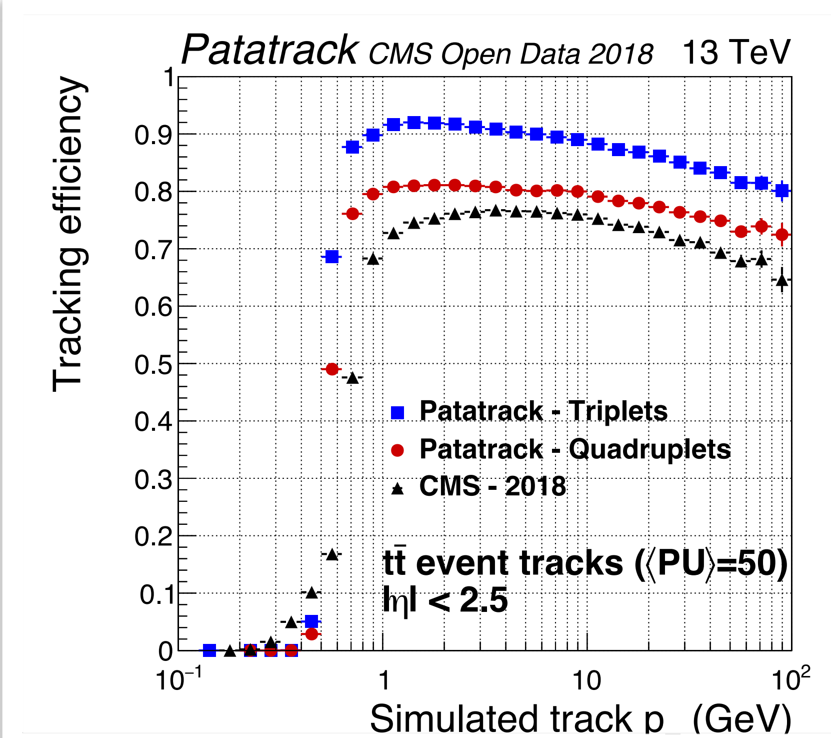


Tracking

- Early studies from **ALICE** exploring **GPUs**
    - Used for **TPC tracking** since 2012
    - Run-3 also use for **additional tracking detectors**
- CMS pixel tracking and vertexing algorithms running on CPU and GPU (**Patatrack**)
    - Superior **technical** performance, and equal or better **physics** performance
- **traccc** project (connected to ACTS) developing end-to-end track reconstruction on GPUs
- LHCb **RETINA** project on FPGAs, aims for track reconstruction, currently vertex clustering



D. Rohr

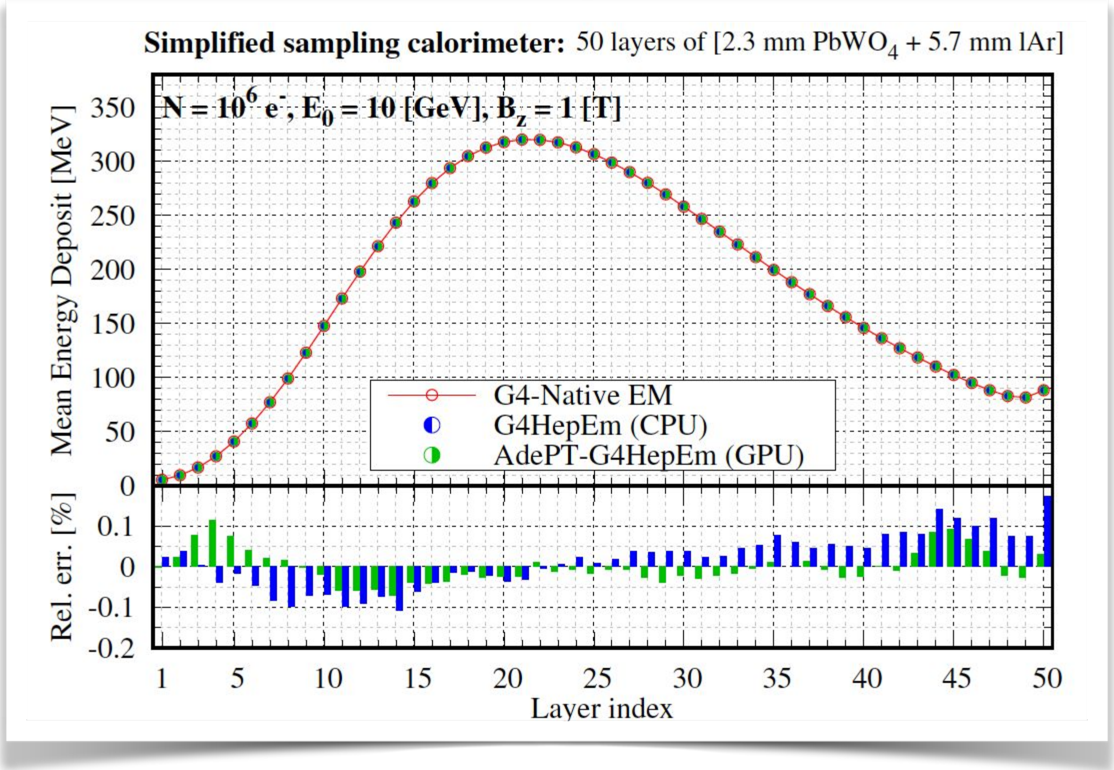


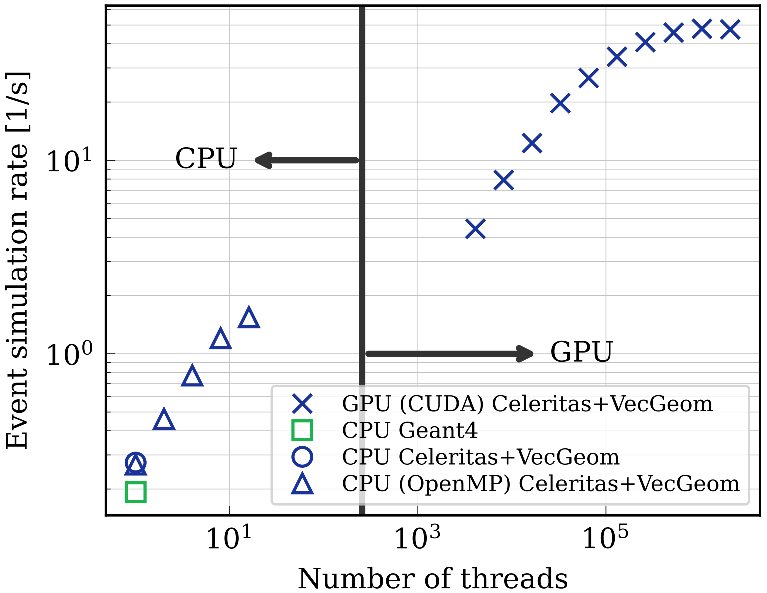
Configuration	throughput in events/s				
	Triplets CPU	Triplets GPU	Quadruplets CPU	Quadruplets GPU	CMS 2018
no copy	611	870	892	1386	476
copy, no conv.	—	867	—	1372	—
conversion	585	861	855	1352	—

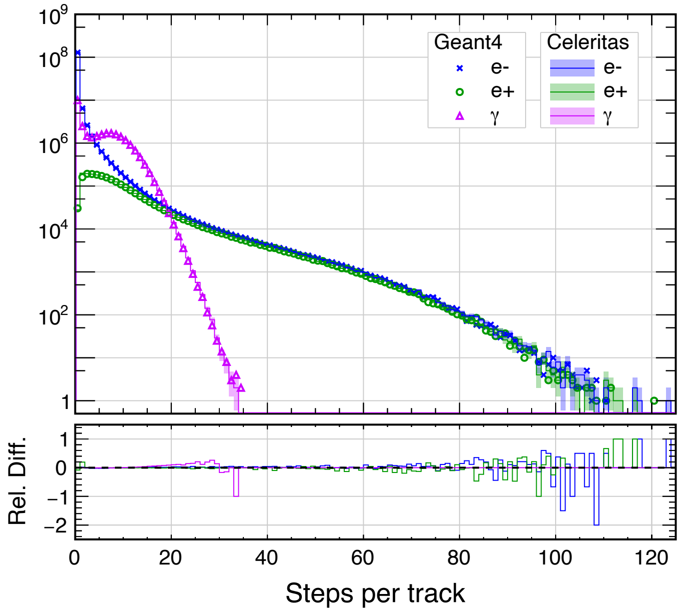
A. Bocci et al

Simulation

- G4 plugins for EM transport on GPUs: **AdePT**
- Very early performance studies
  - G4 CPU 1 (24) threads : 497 (43) s
  - AdePT GPU: 115 s
- **Celeritas** also aims to offload EM physics
  - Reproduces Geant4
  - Large performance gains
- A number of groups currently exploring **Opticks**



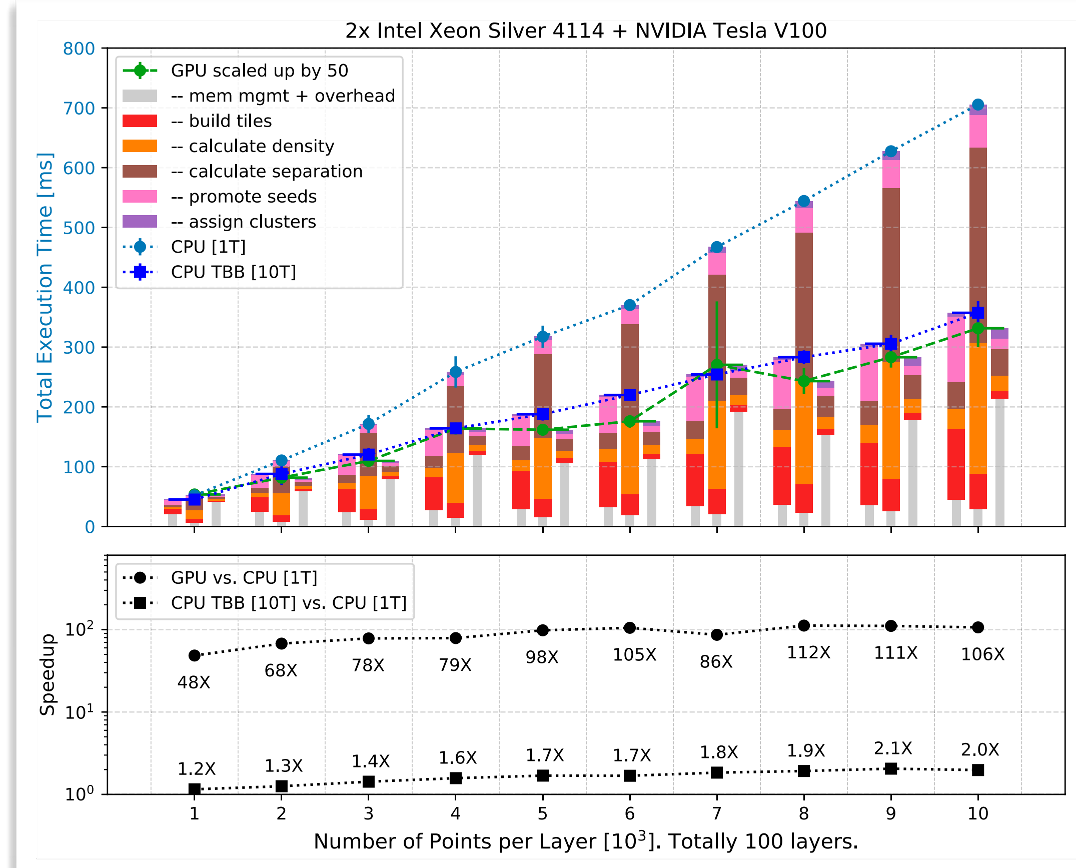


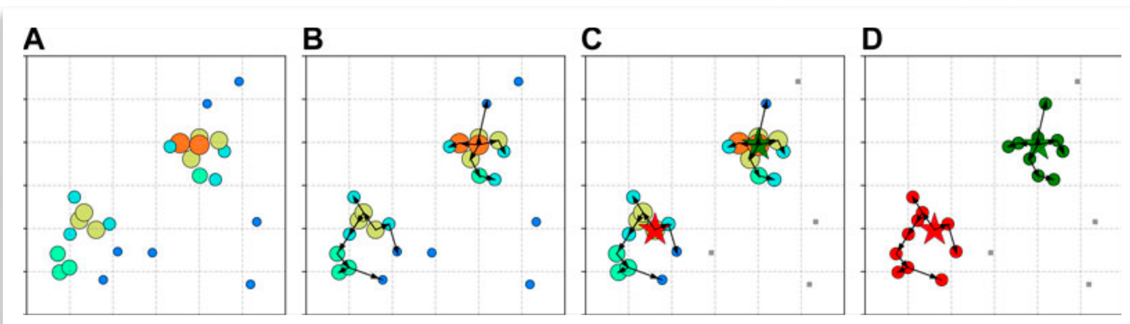


Togini et al, arXiv:2203.09467

Calo Reconstruction

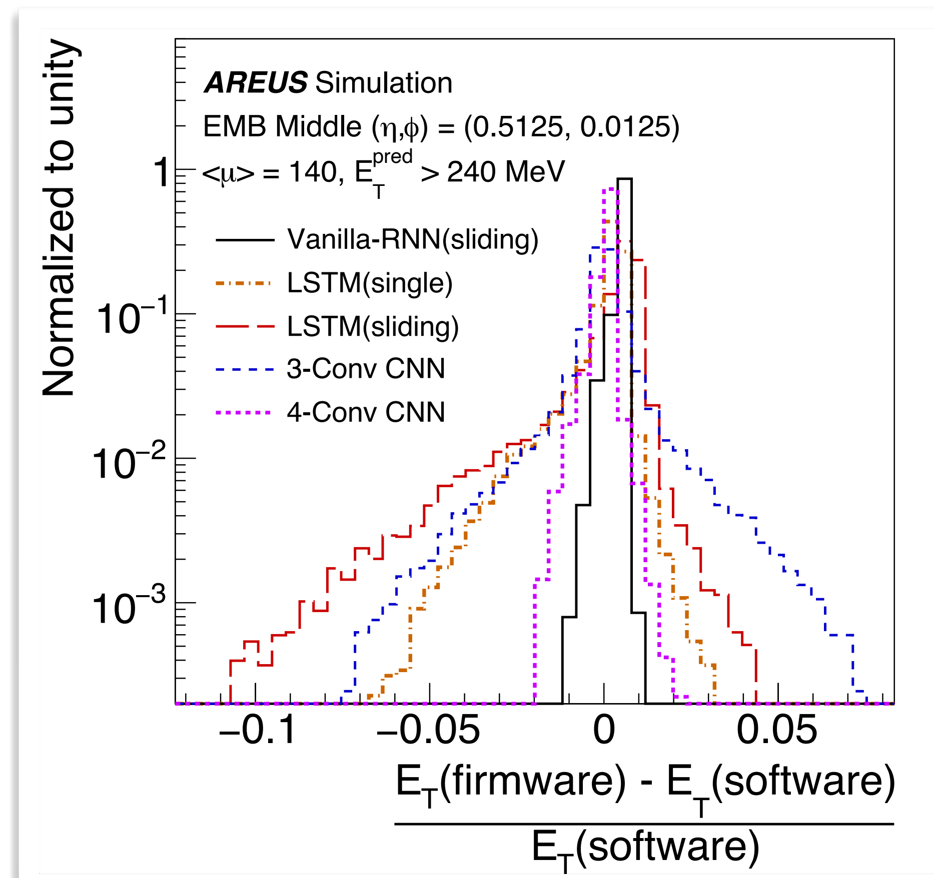
- High-granularity calorimeters: **clustering** becomes a computational challenge
- CMS has developed the highly-parallelizable CLUE algorithm which obtains speed increases of 48-112 (1.2 - 2.0) on **GPUs** compared to single (10 threaded) CPUs





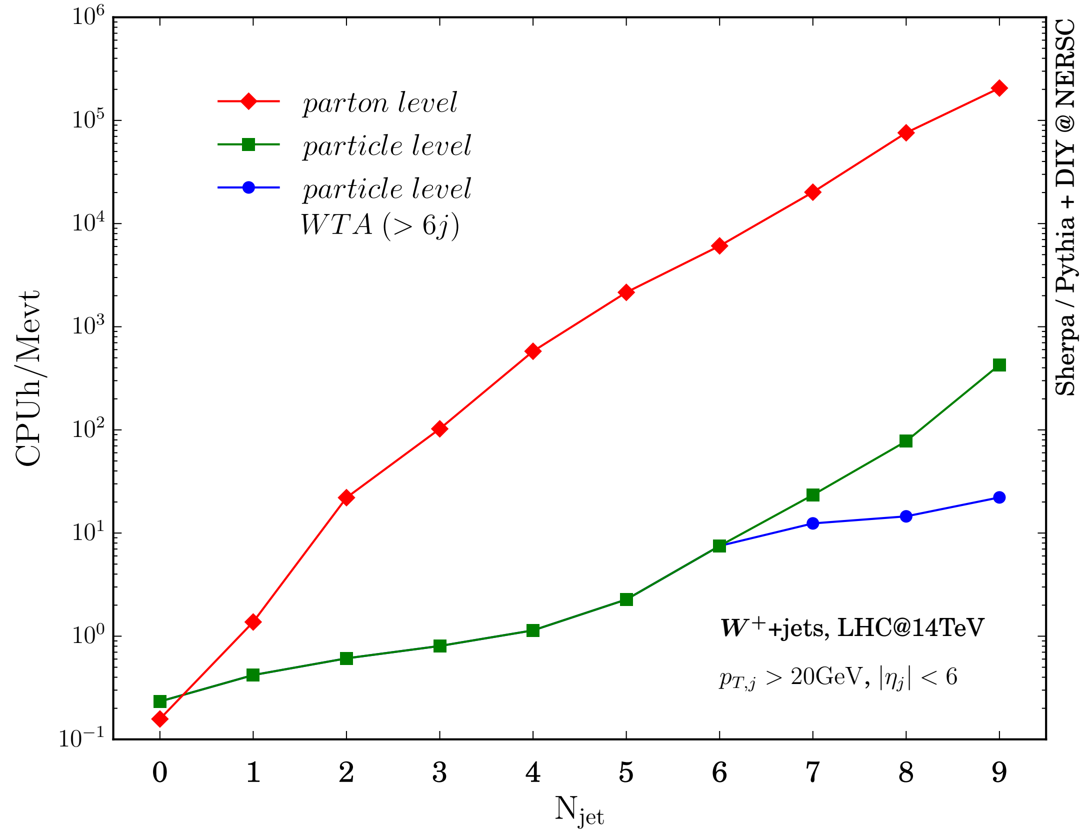
[Rovere at al](#)

- ATLAS implementation of calorimeter signal processing on **FPGAs** using ML
  - Promising results, but requires further optimization to improve resource usage and latency

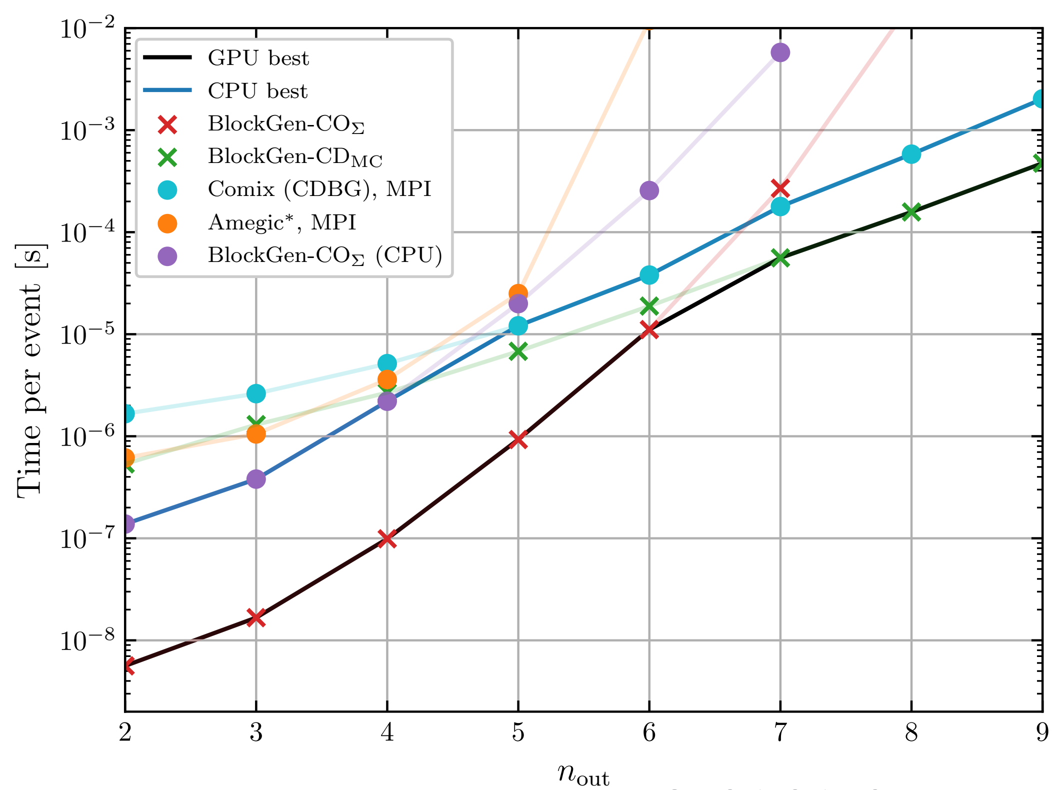


Generators

- Generation time becomes significant particularly as the **event complexity** increases (e.g. events with many jets)
- GPUs are being explored for **matrix element calculations** to improve speed, e.g.
  - BlockGen
  - MadFlow
  - MadGraph-GPU
- Recent review, Isaacson, arXiv: 2202.05991



S. Hoeche et al, arXiv: 1905.05120v1



Bothmann et al, arXiv 2106.06507

Machine Learning

Machine Learning

- Machine learning methods have been used in HEP since the **1990s** [see Bhat, 2011 for a review]
  - Recent advent of **deep learning** has boosted performance
- Classification and regression used in **all steps** of the HEP software pipeline
  - Examples on the following slides
  - Also many interesting talks in parallels
- Developments in machine learning are often driven by **industry**
  - HEP benefits through the application of these techniques
- In most cases, aim for improved **physics performance** rather than improved speed

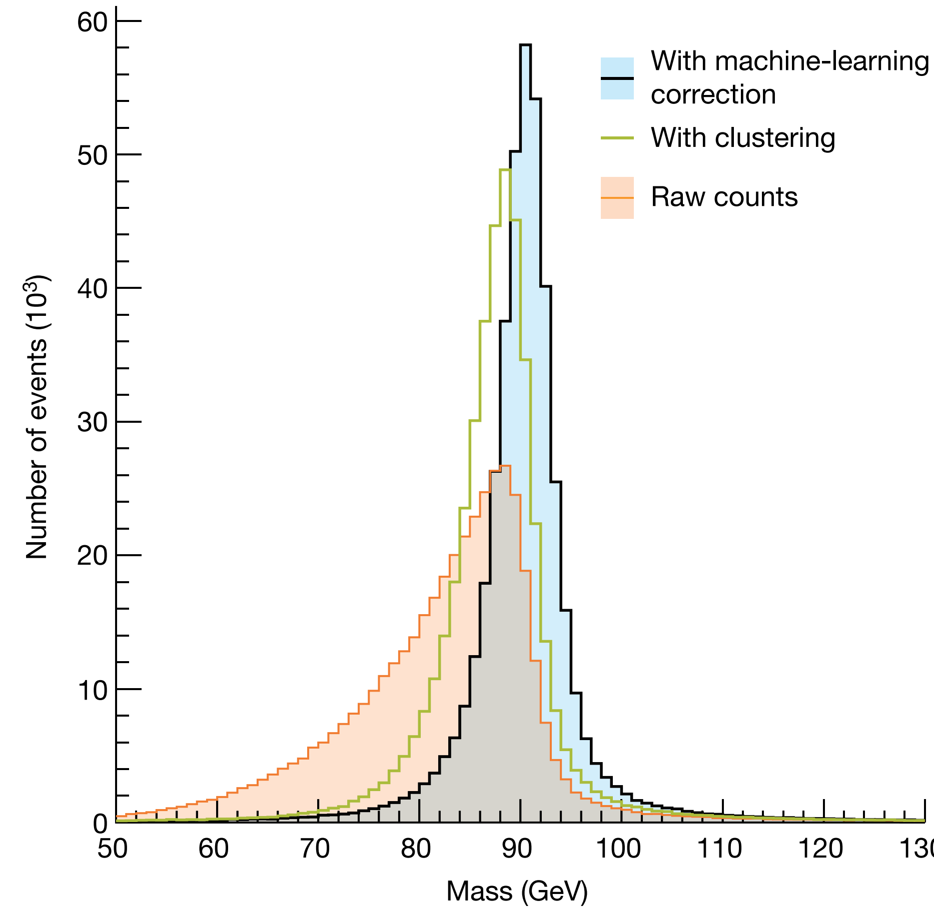


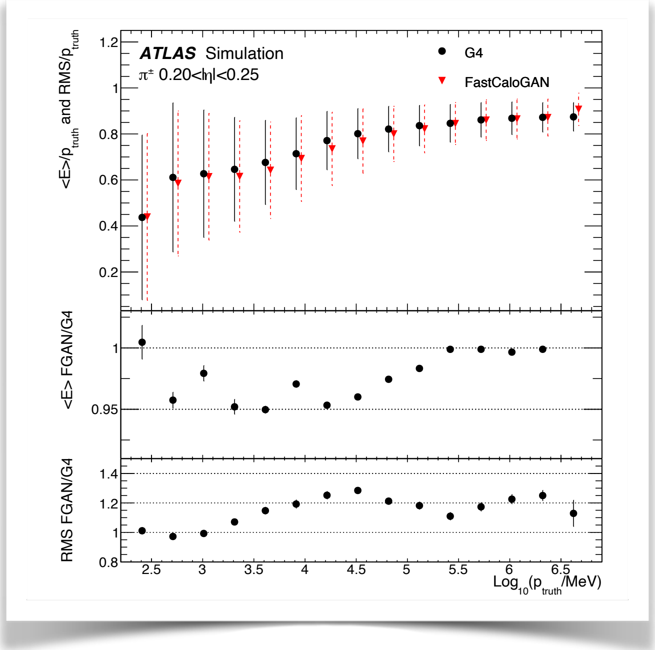
Table 1 | Effect of machine learning on the discovery and study of the Higgs boson

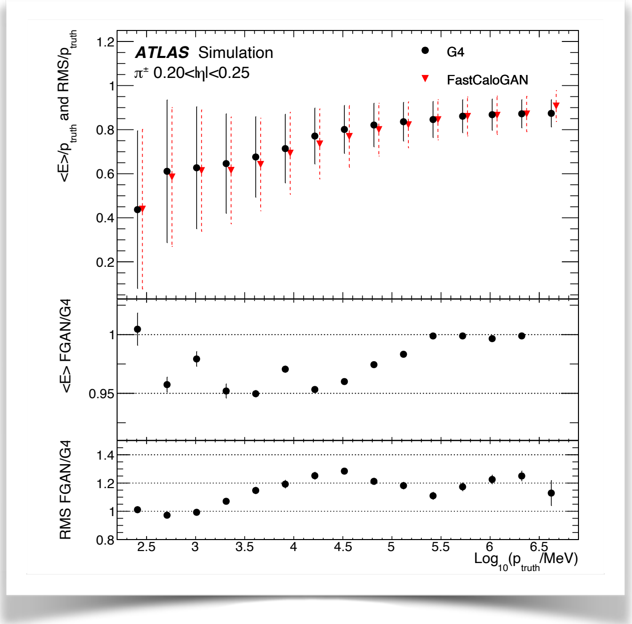
Analysis	Years of data collection	Sensitivity without machine learning	Sensitivity with machine learning	Ratio of P values	Additional data required
CMS[24] $H \rightarrow \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \rightarrow \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \rightarrow bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \rightarrow bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \rightarrow bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%

https://doi.org/10.1038/s41586-018-0361-2

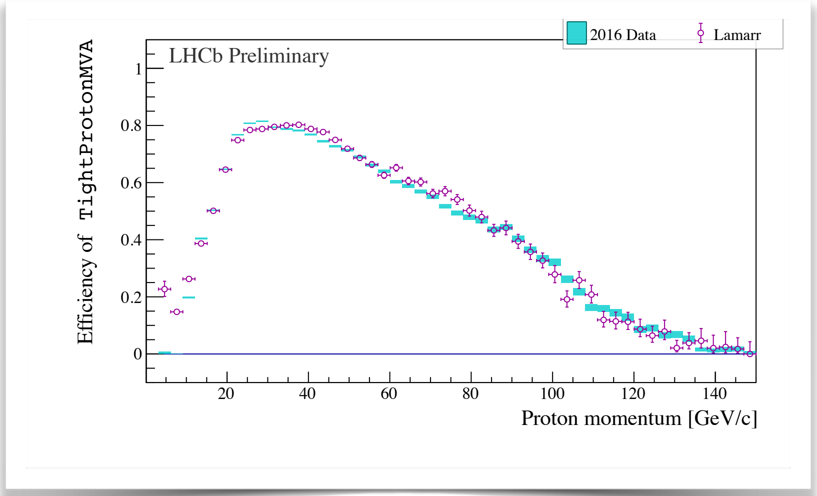
Simulation

- For simulation, the ML method of choice is currently Generative Adversarial Networks (GANs)
- ATLAS **fast calorimeter** simulation uses GANs for selected phase space
    - improves the modeling of hadronic shower fluctuations
- LHCb **Lamarr** uses GANs for particle identification simulation, tracking efficiency and resolution
    - Ongoing work for calorimeter simulation
- ALICE uses a GAN for fast simulation of their **zero degree calorimeter**





Comput Softw Big Sci 6 (2022) 7

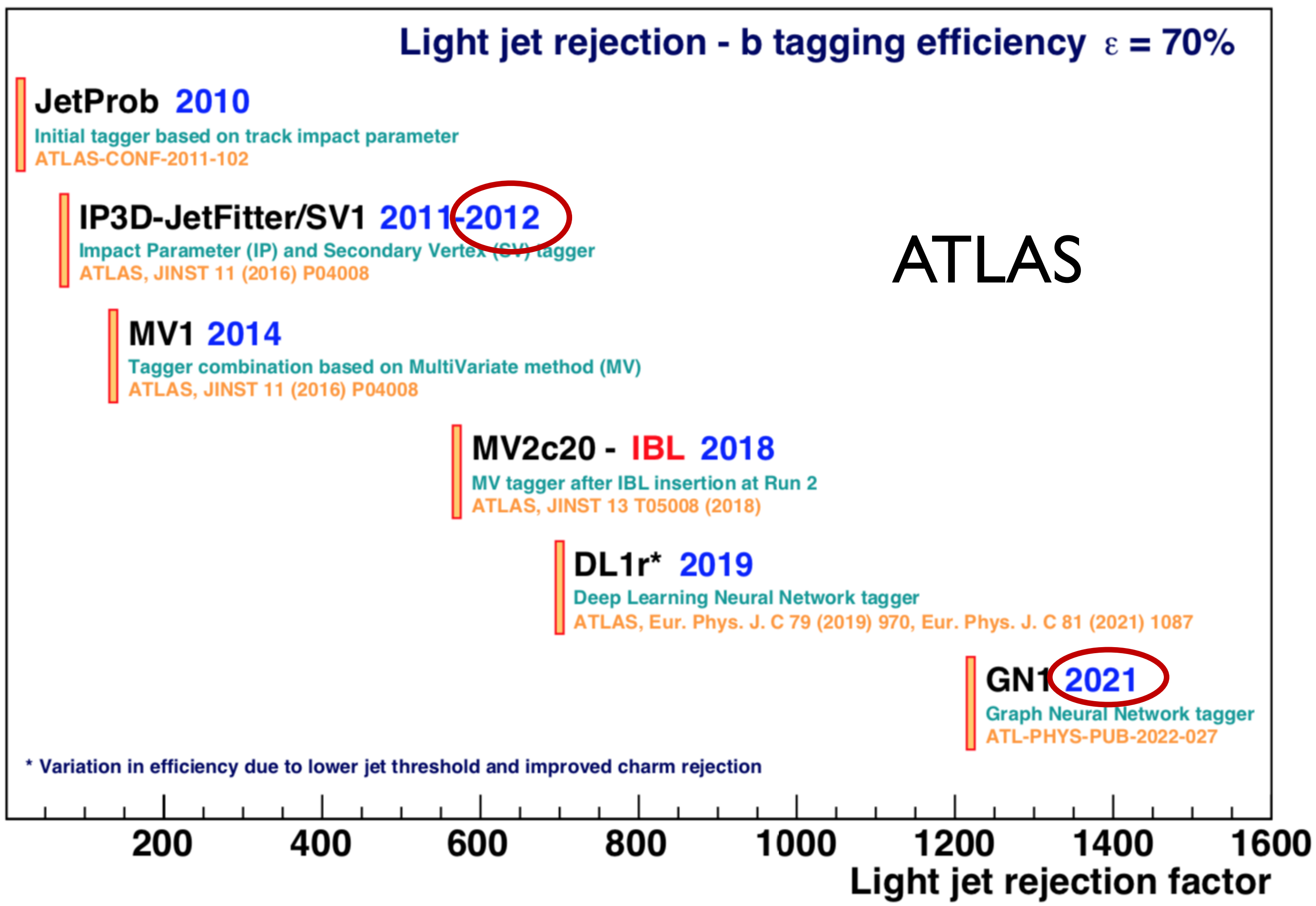


Other ML simulation techniques include variational auto encoders (VAEs), regression (e.g. NNs)

LHCb-FIGURE-2019-017

Flavor tagging

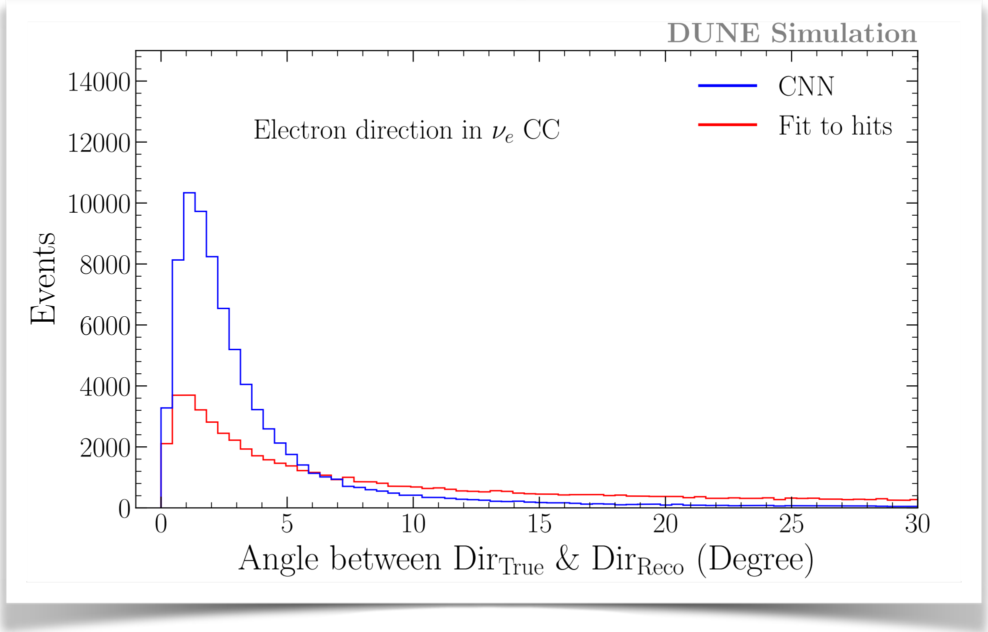
- Extensive (and exclusive) use of ML for **flavor tagging** for many years
- **Example:** Improvement in light jet rejection for ATLAS over the years
  - Large improvement by the use of deep learning and GNNs



Other applications in reconstruction

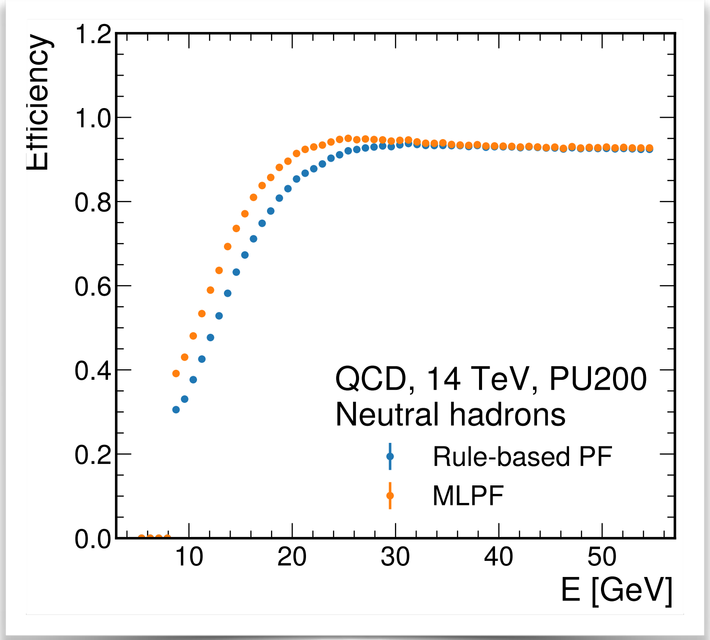
- DUNE obtains superior performance for convolutional neural networks (CNNs) for **energy** and **direction reconstruction**
- Improved performance CMS-like detector with **particle flow reconstruction** with Graph Neural Networks (GNNs)
- Belle II **full event interpretation** with multivariate classifiers

DUNE electron direction reconstruction



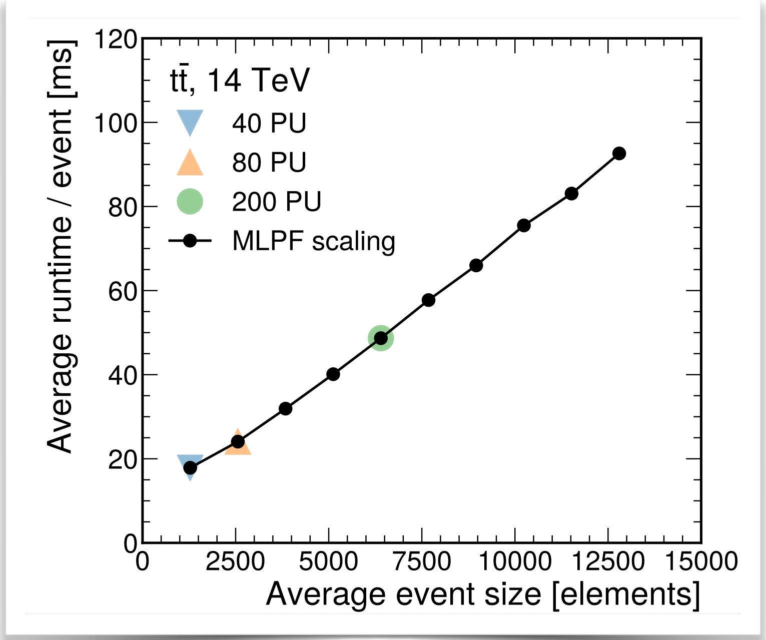
Liu et al

CMS-like improved physics performance



Keck et al

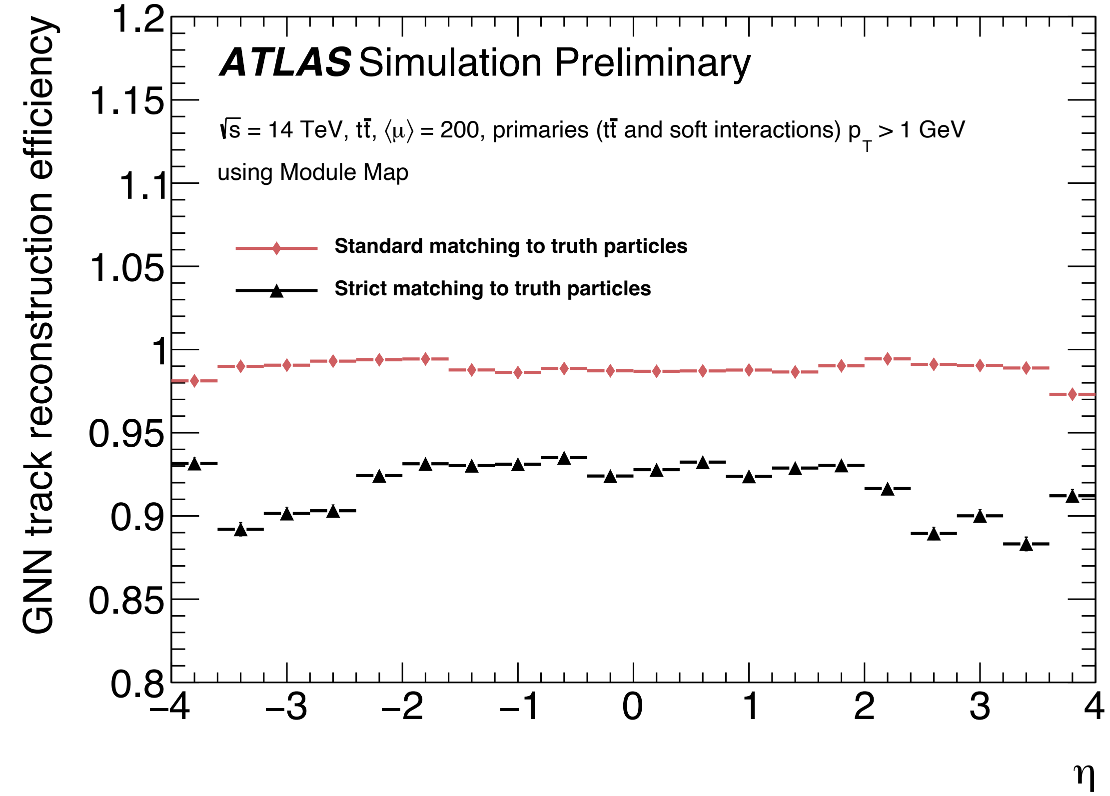
CMS-like linear timing scaling

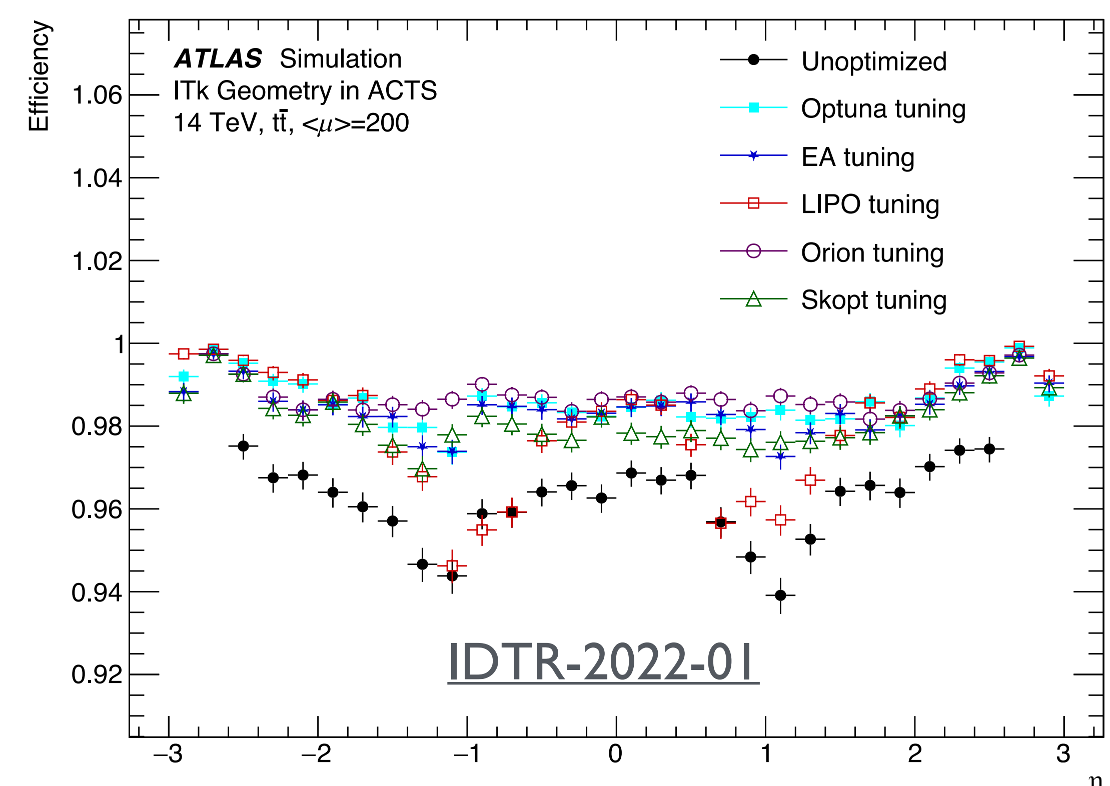


Pata et al

Track reconstruction

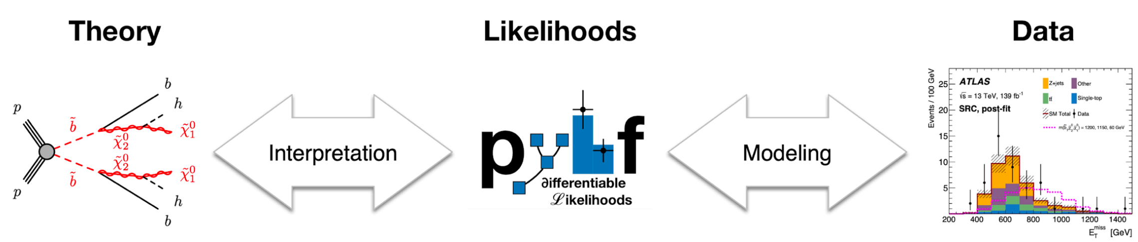
- Tracking algorithms make the largest demands on **CPU requirements**
- Algorithms using machine learning are being explored by a number of groups, predominantly **Graph Neural Networks**, e.g.arXiv:1810.06111, arXiv:2003.11603, arXiv:2103.16701
  - See arXiv:2012.01249 for a recent review
- Can also use machine learning to **automatically tune** the parameters of track reconstruction algorithms





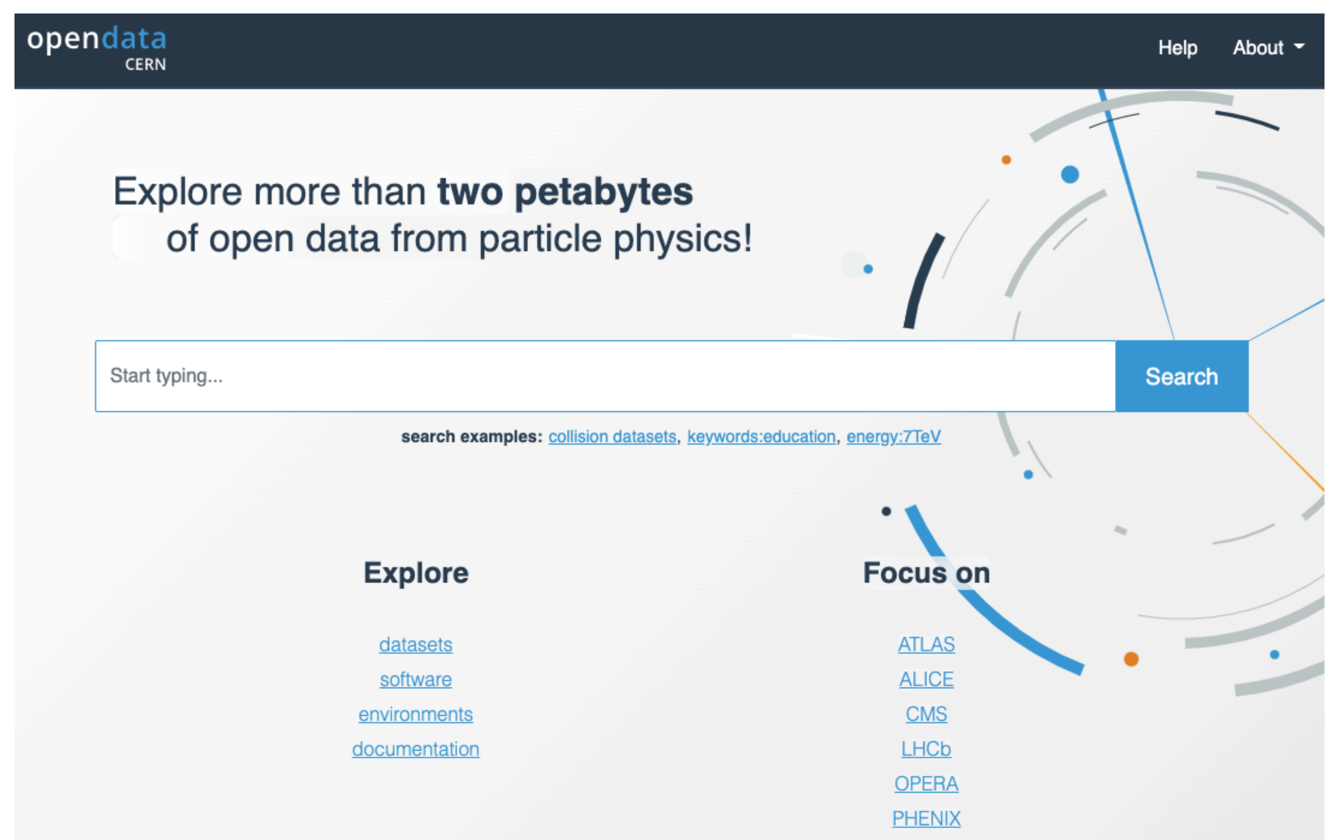
Open [Software, Data]

- The open source philosophy has long played an important role in software development
- At the LHC, first the **results**, then the **software**, then **data** and most recently the **likelihoods** of the LHC experiments have become open
  - **Reinterpretation** can probe additional models
  - However: can be challenging to use our software/data if you don't have direct access to experts and significant hardware resources
- CERN **Open Data Policy**



reana

https://reanahub.io/



Conclusion

- Software plays a key role in essentially **every component** of modern HEP experiments
- Within HEP, software been going through a period of **rapid evolution** due to more demanding **experimental requirements** and changing **hardware environment**
- Key features include
  - **Optimization** and **modernization**
  - Movement towards **common software**
  - Increasing diversity of **hardware architectures**
  - Impact of **machine learning**
- This rapid development will need to continue in preparation for **future upgrades** such as the HL-LHC
- For further details, I encourage you to consult the excellent talks from the **parallel sessions**

Acknowledgments

- Many thanks to C. Bozzi, A. Cerri, A. diGirolamo, J. Elmsheuer, V. Gligorov, M. Kreps, D. Lange, P. Laycock, J. Letts, Z. Marshall, A. Morsch, D. Piparo, E. Rodrigues Figueiredo, N. Styles, G. Stewart for helpful discussions and material used in this talk