30th International Symposium on Lepton Photon Interactions at High Energies

Software and Computing R&D

Graeme A Stewart, CERN EP-SFT

HEP Software Foundation

2022-01-14

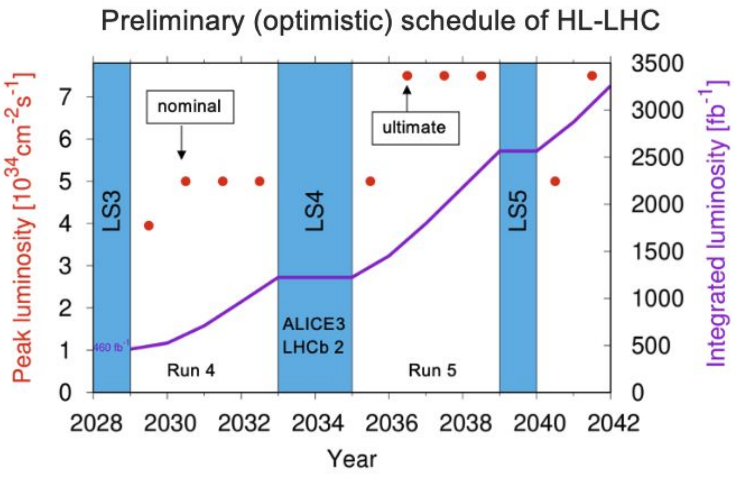
HL-LHC, the Intensity Frontier, and beyond

Our mission:

- Exploit the Higgs for SM and BSM physics
- b, c, tau physics to study BSM and matter/antimatter
- Dark matter
- QGP in heavy ion collisions
- Neutrino oscillations and mass
- Explore the unknown

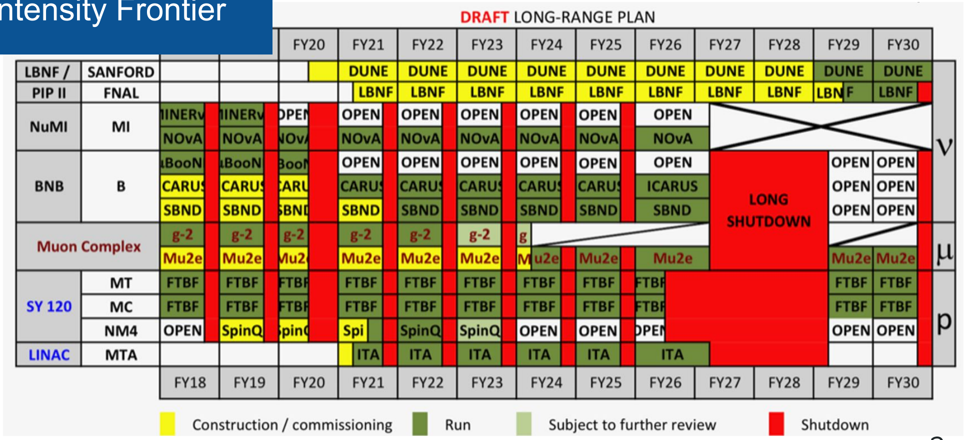
Our Tools:

- [present] (HL-)LHC, DUNE, Belle II
- [longer-term] ILC, FCC, CEPC, BEPC
- smaller LHC-adjacent experiments (e.g. FASER)
- nuclear physics experiments (e.g. FAIR, EIC)

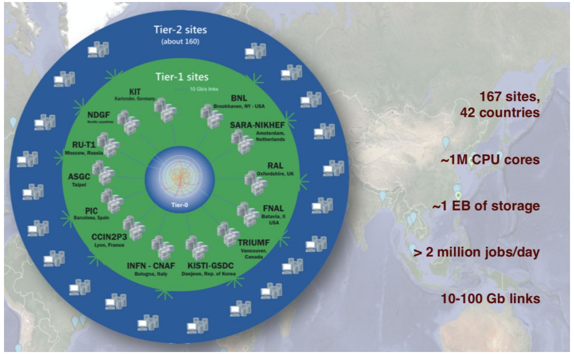


Updated HL-LHC Schedule

FNAL Intensity Frontier



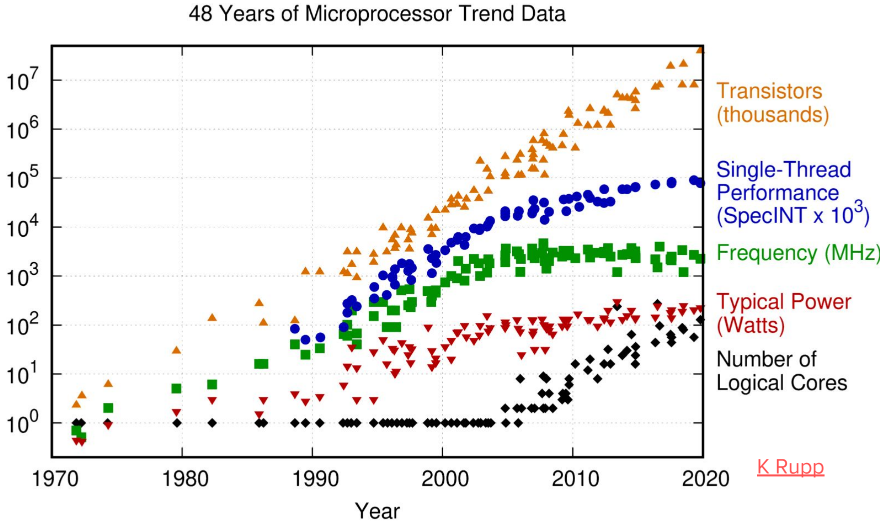
HEP Software and Computing



- High Energy Physics has a vast investment in software
  - Estimated to be around 50M lines of C++
  - Which would cost more than 500M$ to develop commercially
- It is a critical part of our physics production pipeline, from triggering all the way to analysis and final plots as well as simulation
- LHC experiments use about 1M CPU cores every hour of every day, we have around 1000PB of data with 1000PB of data transfers per year (10-100Gb links)
  - We are in the exabyte era already
- This is a *huge* and *ongoing* cost in hardware and human effort
- With significant challenges ahead of us to support our ongoing physics programme

Technology Evolution

- Moore's Law continues to deliver increases in transistor density
    - Increasingly challenging technical issues, but there is a roadmap to 2nm by 2025
- Clock speed scaling failed around 2006
    - No longer possible to ramp the clock speed as process size shrinks
    - Leak currents become important source of power consumption
- So we are basically stuck at ~3GHz clocks from the underlying Wm^{-2} limit
    - This is the *Power Wall*
    - Limits the capabilities of serial processing
- Memory access times are now ~100s of clock cycles



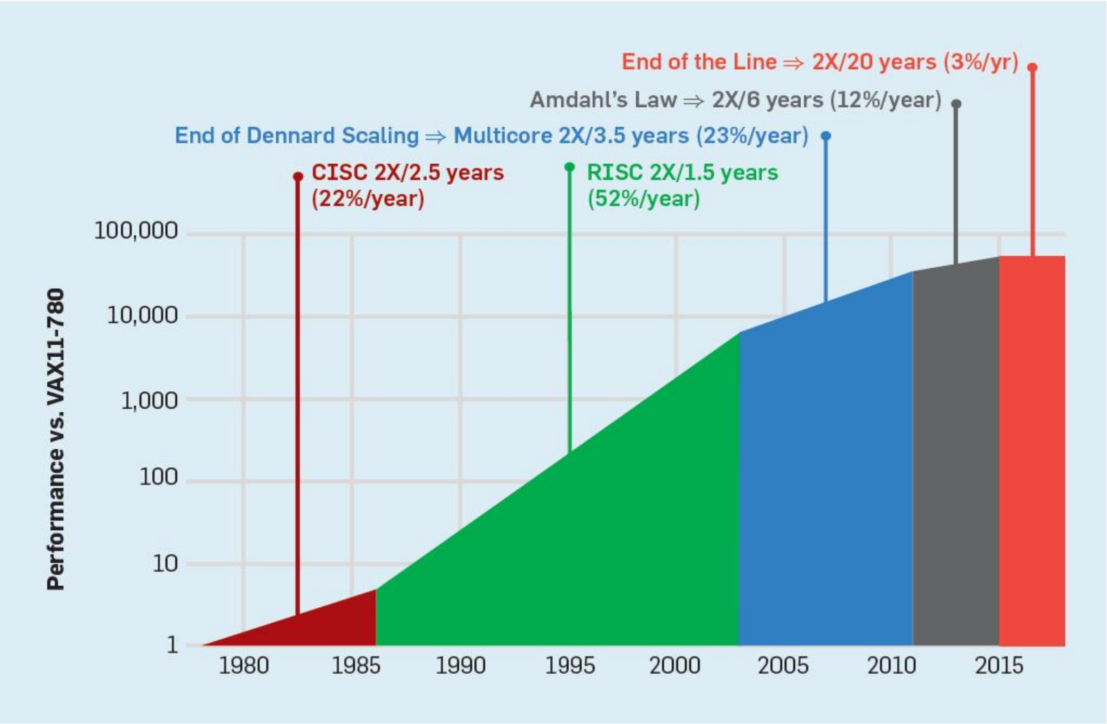
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

New process schedules for future chips

	2018	2020	2021	2022	2023	2024	2025
TSMC	7nm	5nm/6 nm		3nm	3nm+		2nm
Samsung	8nm	5nm/4 nm			3nm		2nm
Intel	Intel 10		Intel 7		Intel 4/Intel 3	Intel 20A	Intel 18A

Decreasing Returns over Time

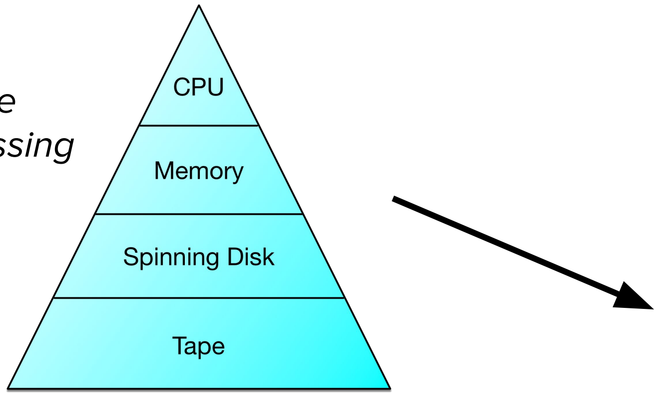
- Conclusion: diversity of new architectures will only grow
- Best known example is of GPUs
- Also FPGAs, TPUs
- As well as non-trivial innovations for CPUs
  - Apple M1
  - Fujitsu A64FX
  - Google Tensor

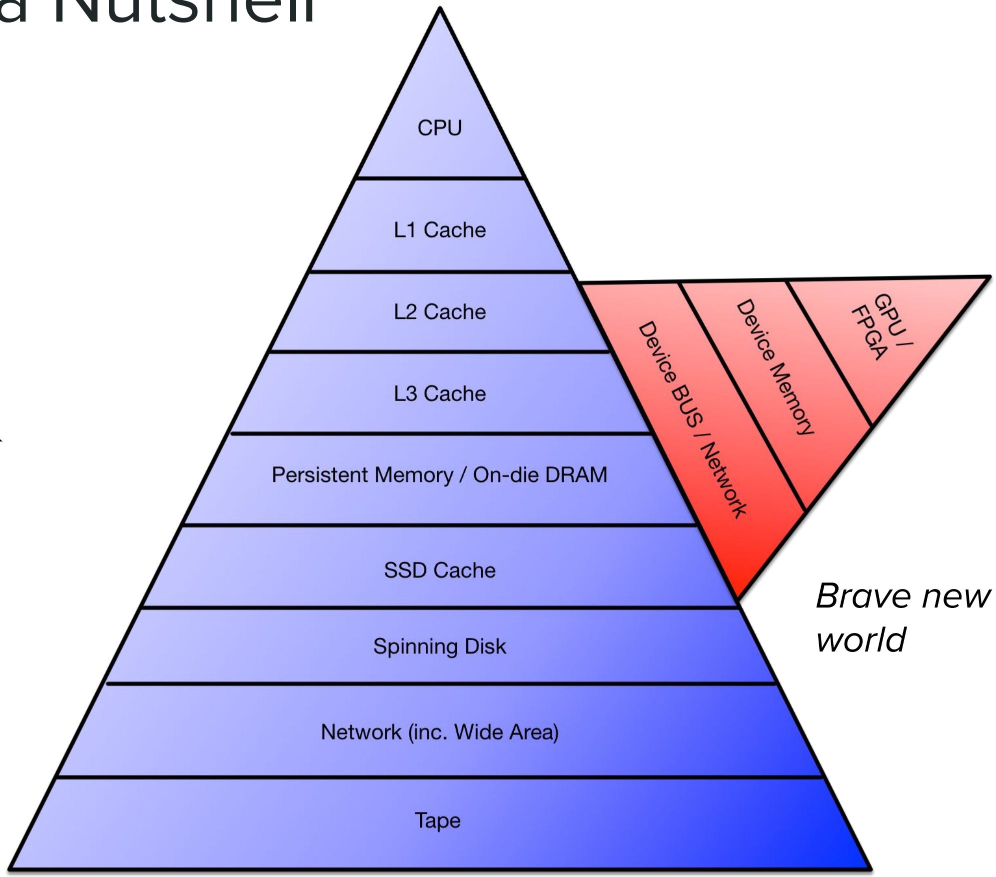


GPUs dedicate far more transistors to arithmetic

Hardware Evolution in a Nutshell

Ye olde processing model





Brave new world

“We’re approaching the limits of computer power – we need new programmers now”
[John Naughton, Guardian]

2020 Strategy Statements

4. Other essential scientific activities for particle physics

Computing and software infrastructure

- There is a need for strong community-wide coordination for computing and software R&D activities, and for the development of common coordinating structures that will promote coherence in these activities, long-term planning and effective means of exploiting synergies with other disciplines and industry
- A significant role for artificial intelligence is emerging in detector design, detector operation, online data processing and data analysis
- Computing and software are profound R&D topics in their own right and are essential to sustain and enhance particle physics research capabilities
- More experts need to be trained to address the essential needs, especially with the increased data volume and complexity in the upcoming HL-LHC era, and will also help in experiments in adjacent fields.

d) Large-scale data-intensive software and computing infrastructures are an essential ingredient to particle physics research programmes. The community faces major challenges in this area, notably with a view to the HL-LHC. As a result, the software and computing models used in particle physics research must evolve to meet the future needs of the field. *The community must vigorously pursue common, coordinated R&D efforts in collaboration with other fields of science and industry to develop software and computing infrastructures that exploit recent advances in information technology and data science. Further development of internal policies on open data and data preservation should be encouraged, and an adequate level of resources invested in their implementation.*

HEP Software Foundation (HSF)

- The LHC experiments, Belle II, DUNE and future experiments face the same challenges
  - HEP software must evolve to meet these challenges and exploit all expertise
  - Avoid further duplicated efforts
  - New experiments should not be starting from scratch, but building on best-of-breed
- The role of the HSF, started in 2015, and now well established in the field, is to facilitate coordination and common efforts in software and computing across HEP in general
  - Our philosophy is bottom up, a.k.a. Do-ocracy
- Early HSF goal to describe a global vision for software and computing for the HL-LHC era and HEP in the 2020s: Community White Paper [10.1007/s41781-018-0018-8]
  - Community engagement: 310 authors from 124 institutes, 14 chapters
  - Recent updates address better understanding of the HL-LHC challenge
- Engaged with European Strategy, ECFA, LHCC, Snowmass, etc to advocate for software

Software and Computing International R&D Projects

There has been recent investment specifically into software and computing R&D in recent years (many supported directly by the HSF):

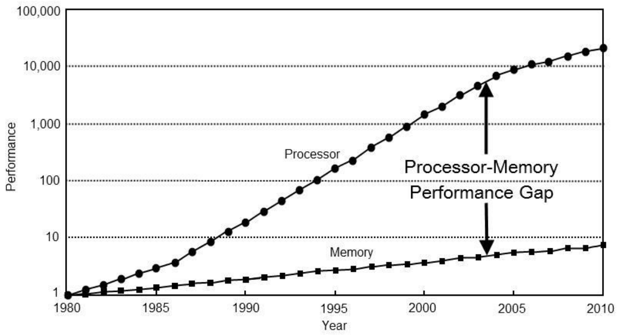
- IRIS-HEP, NSF USA
    - Analysis systems, innovative algorithms, DOMA
- ErUM-DATA, Helmholtz Institute DE
    - Heterogeneous computing and virtualized environments, machine learning for reconstruction and simulation
- EP R&D, CERN
    - Turnkey software systems, faster simulation, track and calo reconstruction,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

Software Challenges and Opportunities

Concurrency

- The one overriding characteristic of modern processor hardware is concurrency
  - SIMD - Single Instruction Multiple Data (a.k.a. vectorisation)
    - Doing exactly the same operation on multiple data objects
  - MIMD - Multiple Instruction Multiple Data (a.k.a. multi-theading or multi-processing)
    - Performing different operations on different data objects, but at the same time
  - SIMT - Single Instruction Multiple Threads
    - GPU running a block of threads in instruction lock-step (masking allowed, more flexible than SIMD)
- Because of the inherently parallel nature of HEP processing a lot of concurrency can be exploited at rough granularity
  - Run many jobs from the same task in parallel; Run different events from the same job in parallel
- However, the push to highly parallel processing (1000s of GPU cores) requires **parallel algorithms**
  - This often requires completely rethinking problems that had sequential solutions previously, e.g.,
    - Finding track seeds via cellular automata (TrickTrack library, CMS and FCC)
    - Evaluation of multiple phase space points in event generation in MadGraph

Heterogeneity and Data Layout



- There are a lot of possible parallel architectures on the market
  - CPUs with multiple cores and wide registers
    - SSE4.2, AVX, AVX2, AVX512, Neon, SVE, Altivec/VMX, VSX
  - GPUs with many cores; FPGAs
    - NVIDIA (many generations - often significantly different), AMD, Intel, ...
- Many options for coding, both generic and specific:
  - CUDA, HIP, TBB, HPX, OpenACC, OpenMP, OpenCL, SYCL, Alpaka, Kokkos, oneAPI, ...
- Frustratingly no clear winner, mutually exclusive solutions and many niches - actively investigated in HEP-CCE project
- Data layout critical to performance - needs to be supported now on multiple devices
  - But with efficient translation (ideally no change) and hiding latency

Machine Learning

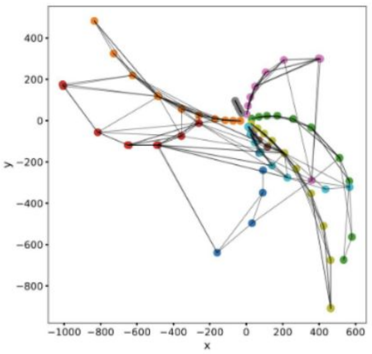
- Machine learning, or artificial intelligence, used for many years in HEP
  - Algorithms learn by example (training) how to perform tasks instead of being programmed
- Significant advances in the last years in ‘deep learning’
  - Deep means many neural network layers
  - Fast differentiability and use of GPUs have made this practical
- Rapid development driven by industry
  - Vibrant ecosystem of tools and techniques
  - *Highly optimised for modern, specialised hardware*
- HEP exploiting this domain, but also pushes development in, e.g., graph neural networks
- Now generalising this approach to *differentiable computing*

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] $H \to \gamma\gamma$	2011–2012	2.2σ, $P = 0.014$	2.7σ, $P = 0.0035$	4.0	51%
ATLAS[43] $H \to \tau^+\tau^-$	2011–2012	2.5σ, $P = 0.0062$	3.4σ, $P = 0.00034$	18	85%
ATLAS[99] $VH \to bb$	2011–2012	1.9σ, $P = 0.029$	2.5σ, $P = 0.0062$	4.7	73%
ATLAS[41] $VH \to bb$	2015–2016	2.8σ, $P = 0.0026$	3.0σ, $P = 0.00135$	1.9	15%
CMS[100] $VH \to bb$	2011–2012	1.4σ, $P = 0.081$	2.1σ, $P = 0.018$	4.5	125%

Machine learning at the energy and intensity frontiers of particle physics,

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

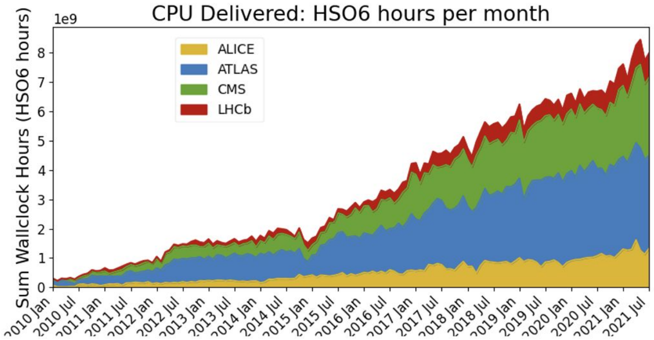


Graph Neural Network used in charged particle tracking

```
int main {
  cout << “write software” << endl;
  return 0;
}
```

Computing R&D

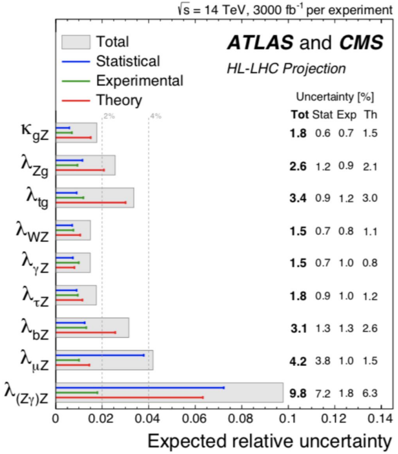
- Largest scales of HEP computing dominated by the LHC experiments
  - Operations and R&D under the umbrella of WLCG and the Data Organisation, Management and Access group
  - Increasingly use by non-LHC experiments, e.g., Belle II
- Infrastructure software consolidation is very helpful for scaling and sustainability
  - Increasing use of Rucio across experiments: ATLAS, CMS, Belle II, Square Kilometer Array Telescope
  - FTS for lower level transfers
    - Deprecate GridFTP for http and xrootd
  - Move from x509 certificates to tokens - easier for end users and fits in to wider ecosystems better
- Increasing use of non-HEP facilities, like HPCs
  - Specialist analysis facilities?



RUCIO
SCIENTIFIC DATA MANAGEMENT

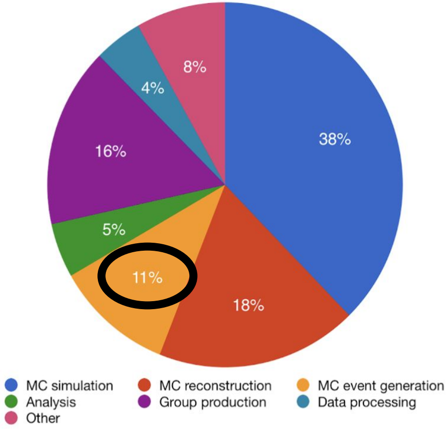
Event Generators

- First step of all simulation
  - LHC Run-1 leading order generators and little contribution to overall CPU budgets
- Increasing importance for LHC precision measurements
  - ATLAS and CMS now use higher order generators like Madgraph and Sherpa
- HSF/LPCC organised the first ever [workshop devoted to software and computing issues] in the generators domain in 2018
  - Identifying a combination of physics challenges and technical challenges
  - Sociological challenges complicate matters
    - Generators written by theory colleagues in small teams
    - Little incentive to improve the computing performance for experiments
- HSF WG continued this work with overview papers on the challenges [10.1007/s41781-021-00055-1]



Many electroweak measurement errors dominated by theory (red). B. Hinemann

ATLAS 2018 CPU Report



Event Generators

- Negative weights
  - Dilute the statistical power of measurements, events need to be simulated and reconstructed
  - New NLO matching schemes proposed that reduce negative weight fraction [2002.12716], MC@NLO-Delta
  - Resampling before further processing reduces CPU "wasted" (positive resampling [2002.12716], NN-based resampling [2002.12716])
  - Efforts from the experiments to improve the sampling schemes to reduce CPU costs and increase statistical precision [2112.09588 - x2 cost reduction with better accuracy for V+jets event in Sherpa]
- Porting to alternative architectures
  - Matrix element calculations are quite suitable for GPUs and CPU SIMD - many repetitive and independent calculations on multiple phases space points
  - Early results from porting limited set of processes of MG5+aMC to GPUs are promising
  - Work also going on in Sherpa team [2106.06507] showing several factor speedups

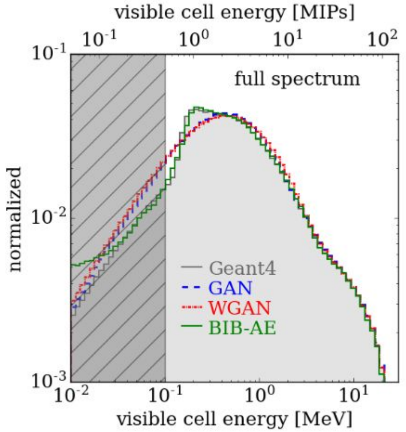
[2002.12716]	MC@NLO 111	MC@NLO-Δ Δ-441
$pp \to e^+e^-$	6.9% (1.3)	2.0% (1.1)
$pp \to e^+\nu_e$	7.2% (1.4)	2.3% (1.1)
$pp \to H$	10.4% (1.6)	0.5% (1.0)
$pp \to Hb\bar{b}$	40.3% (27)	31.3% (7.2)
$pp \to W^+j$	21.7% (3.1)	7.4% (1.4)
$pp \to W^+t\bar{t}$	16.2% (2.2)	11.5% (1.7)
$pp \to t\bar{t}$	23.0% (3.4)	7.7% (1.4)

Implementation ($e^+e^- \to \mu^+\mu^-$)	MEs / second Double
1-core MadEvent Fortran scalar	1.50E6 (x1.15)
1-core Standalone C++ scalar	1.31E6 **(x1.00)**
1-core Standalone C++ 128-bit SSE4.2 (x2 doubles, x4 floats)	2.52E6 (x1.9)
1-core Standalone C++ 256-bit AVX2 (x4 doubles, x8 floats)	4.58E6 (x3.5)
1-core Standalone C++ "256-bit" AVX512 (x4 doubles, x8 floats)	4.91E6 (x3.7)
1-core Standalone C++ 512-bit AVX512 (x8 doubles, x16 floats)	3.74E6 (x2.9)
Standalone CUDA NVidia V100S-PCIE-32GB (2560 FP64 cores*)	7.25E8 **(x550)**

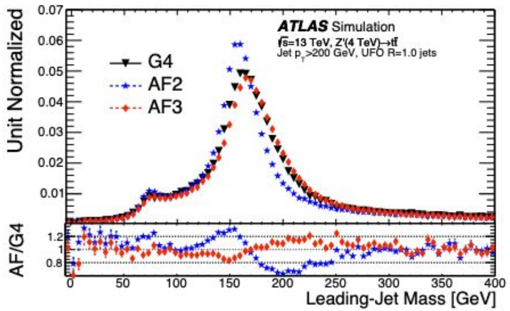
A Valassi et al.

Detector Simulation

Performance of different ML architectures for photons in ILC Calorimeter, S Diefenbacher et al.



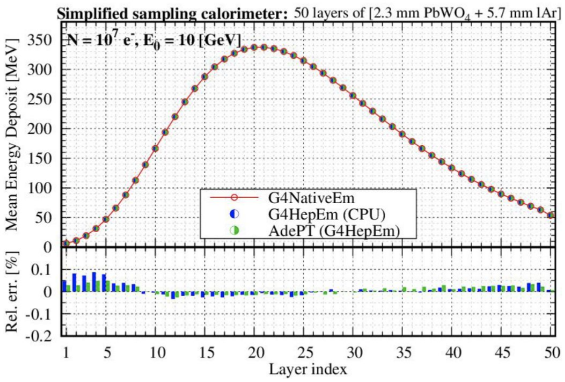
- A major consumer of LHC grid resources today
  - Experiments with higher data rates will need *more simulation*
- Faster simulation, with minimal loss of accuracy, is the goal
  - Range of techniques have been used successfully for a long time (frozen showers, parametric response)
  - Key point is deciding when it's good enough for physics
  - Recent LPCC workshop on fast simulation provides a good overview
- Machine learning lends itself to problems like this
  - Calorimeter simulations usually targeted
  - Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), but also other architectures used (BIB-AE)
    - This is *not as easy as we thought* - traditional parametric approaches are hard to beat, but can be done [10.1007/s41781-021-0(
  - Development from Geant4 targets easier integration of ML
    - Export of events for training
    - Import of trained models in industry standard ways, like ONNX
  - Hybrid models are also an option, e.g., ATLAS AtlFast3, combining parametric and ML approaches [2109.02551]



Improved jet mass modelling when incorporating ML into ATLAS fast simulation

Detector Simulation

- Speeding up particle tracking simulation is also needed
  - Even if fast simulation is inevitable, still need full sim
- Significant effort continually goes into improving Geant4's performance
  - E.g., neutron Russian Roulette, optimised/vectorised geometry
- Particle tracking on GPUs is very challenging, but work has started
  - Naively there is a lot of parallelism in the problem, but stochastic simulation naturally introduces divergence
  - As for fast simulation, calorimeters are a first target
    - AdePT demonstrator (CERN EP-SFT and SWIFT-HEP)
    - Celeritas (DOE)
  - Opticks, using ray tracing on GPUs for optical photons, takes advantage of native ray tracing on GPUs
    - Developer for JUNO (Simon Blythe, 10.1051/epjconf/202125103009) and integrated into Geant4 for DUNE (w. Hans Wenzel)



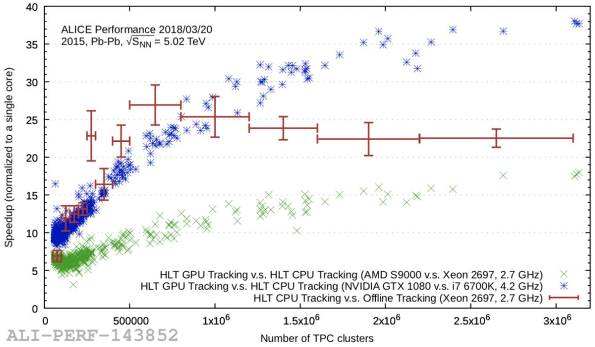
New GPU friendly EM Physics library, G4HepEM, integrated into the AdePT prototype (M Novak, J Hanfeld) - per mil agreement with Geant4

Timing results (Geant4 10.7.p01):	
Geant4 optical physics	2438 sec/event
G4Opticks, RNGmax[1] 10	6.45 sec/event
G4Opticks RTX enabled, RNGmax[1] 10	2.72 sec/event
G4Opticks, RNGmax[1] 100	6.86 sec/event
G4Opticks RTX enabled, RNGmax[1] 100	2.87 sec/event

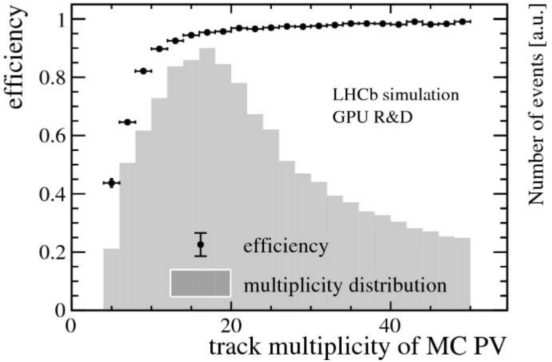
Speedup in optical photon simulation in LAr with GPU (x378) and ray tracing GPU (x900) cf. Geant4 on CPU

Reconstruction and Software Triggers

- Hardware triggers no longer sufficient for modern experiments
  - More and more initial reconstruction needs to happen in software
- Close to the machine, need to deal with tremendous rates and get sufficient discrimination
  - Pressure to break with legacy code is high
  - Lots of developments rewriting code for GPUs
    - Physics can get better!
  - Lessons learned: keep data model simple, bulk data, be asynchronous, minimise data transfers
- This work is driving more and more interest in GPUs in HEP
  - ALICE pioneered this in Run-2 - other LHC experiments also active
  - Choice of LHCb to use Allen for HLT1 baseline is a boost for this R&D line and a general retooling of HEP software

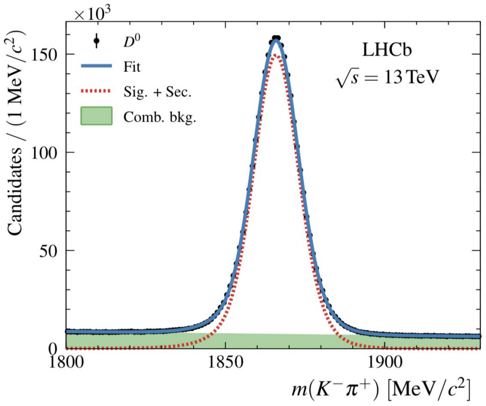


[1905.05515]



Allen: A High-Level Trigger on GPUs for LHCb, doi:10.1007/s41781-020-00039-7

Reconstruction and Software Triggers

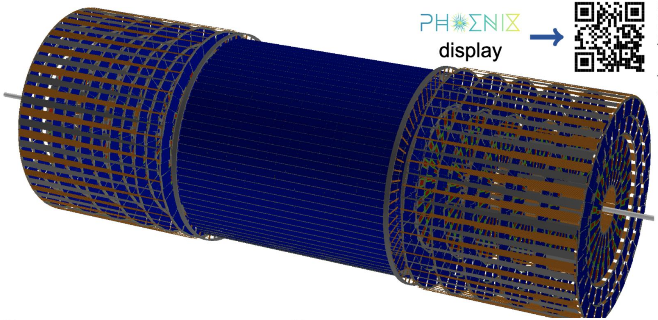


LHCb charm physics analysis using Turbo Stream (arXiv:1510.01707)

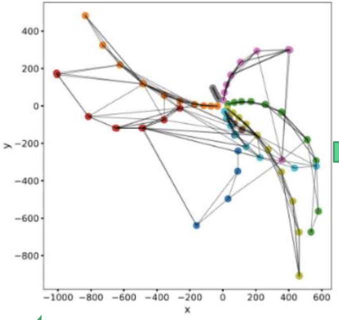
- Real Time Analysis (HEP Version)
  - Design a system that can produce analysis useful outputs as part of the trigger decision
    - If this captures the most useful information from the event, can dispense with raw information
  - *This is a way to fit more physics into the budget*
- LHCb Turbo Stream was a radical way to do this and its use in Run3 is a vindication of the approach
- Whole ALICE data reduction scheme is based around keeping 'useful' parts of events (no more binary trigger)
  - O2 → Online/Offline Data Reduction Farm
- ATLAS and CMS have schemes under development for special handling of samples for which full raw data is unaffordable (e.g., low P_T di-jets)
- For future detectors handling timing information becomes more and more important

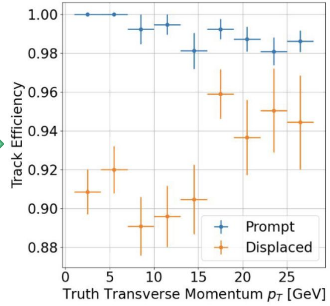
Tracking - ACTS

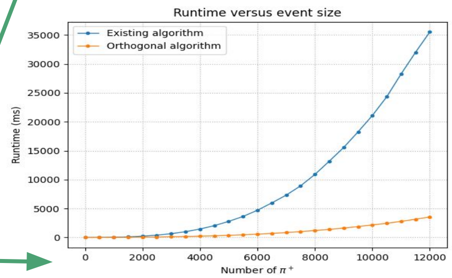
PHOENIX display → cern.ch/go/Q897



- Traditionally reconstruction software has been rather experiment specific
  - Grimy details of sub-detectors, calibrations and specific geometries drove specific implementations...
- The problems of efficient solutions on challenging hardware drive in the opposite direction - towards common solutions that can adapt to different experiments
  - One example is the ACTS Tracking Project [2106.13593]
- Drives developments in parallelisation, compute accelerator implementations and machine learning
  - Development of the Open Data Detector (developed from TrackML) provides an open platform for development and comparison of different approaches
  - Graph neural network approach [1, 2], also explored by Exa.TrkX project [3]
  - Impressive parallelisation results from 'ground up' thread safety

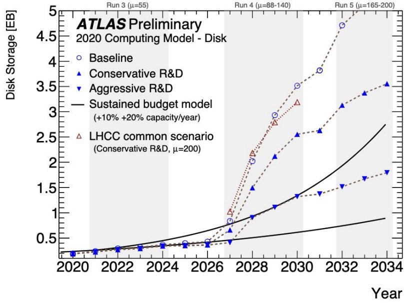


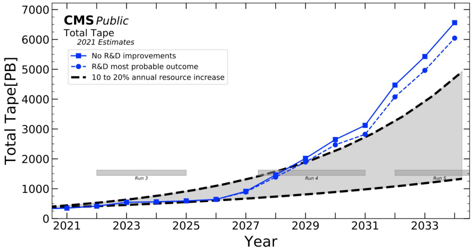




New seed finding strategy on based of KD-trees, achieving very similar tracking performance

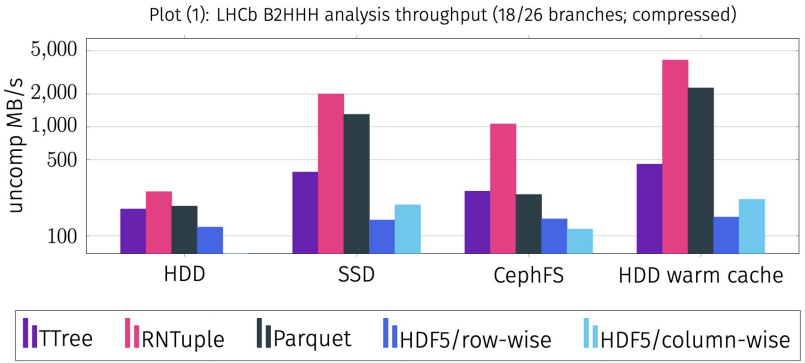
Analysis



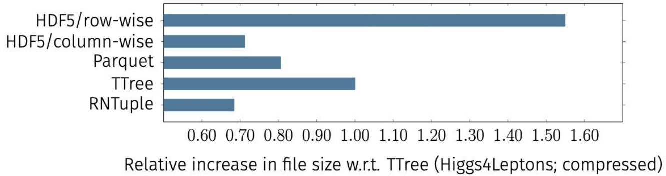


Reminder of the challenge of HL-LHC storage volumes [ATLAS, CMS]

- Scaling for analysis level data also a huge challenge for experiments
- Reducing volume of data needed helps hugely
  - CMS ~1kB nanoAOD [10.1051/epjconf/201921406021] makes a vast difference to analysis efficiency and "papers per petabyte"
- Re-inventing data formats for modern devices is a key piece of re-engineering by ROOT to scale up in speed (and down in size!)
  - New RNTuple format is considerably smaller on disk
  - Faster to read than older formats or industry alternatives
  - Also adapted for modern object stores systems



Read speeds comparing new ROOT RNTuple with older HEP TTree format and other industry formats [Gomez, Blomer] (note y-axis is log10 scale)



On disk serialised sizes comparing new ROOT RNTuple with older HEP TTree format and other industry formats [Gomez, Blomer]

Analysis Ergonomics and Scaling

- Improving how analysis is done increases productivity
  - Rise of declarative models
    - E.g., ROOT's RDataFrame
  - Say what, not how and let the backend optimise
    - More natural treatment of the problem
    - No event loop!
  - E.g. split and merge, GPU execution, cluster-wide distribution, systematics
- Front end is increasingly Python, C++ optimised behind
  - This gives an excellent avenue into the Python Data Science tools ecosystem
  - Many new HEP specific Python packages are contributing to exploiting this area and addressing HEP specific needs
    - Fitting, histogramming, statistics, …
  - Very active field, 1300 registrants at last PyHEP workshop

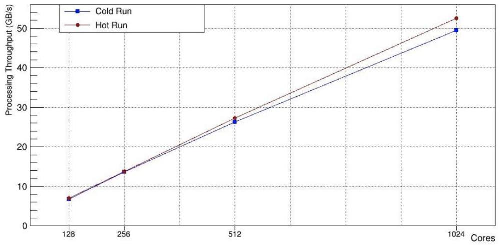
```
nominal_hx =
  df.Vary("pt", "RVecD{pt*0.9, pt*1.1}",  ["down", "up"])
    .Filter("pt > k")
    .Define("x", someFunc, ["pt"])
    .Histo1D("x")

hx = ROOT.RDF.VariationsFor(nominal_hx)
hx["nominal"].Draw()
hx["pt:down"].Draw()
```

Declarative approach to systematics with RDataFrame in ROOT

9 TB processed
in **3 min** on **1024 cores (32 nodes)**

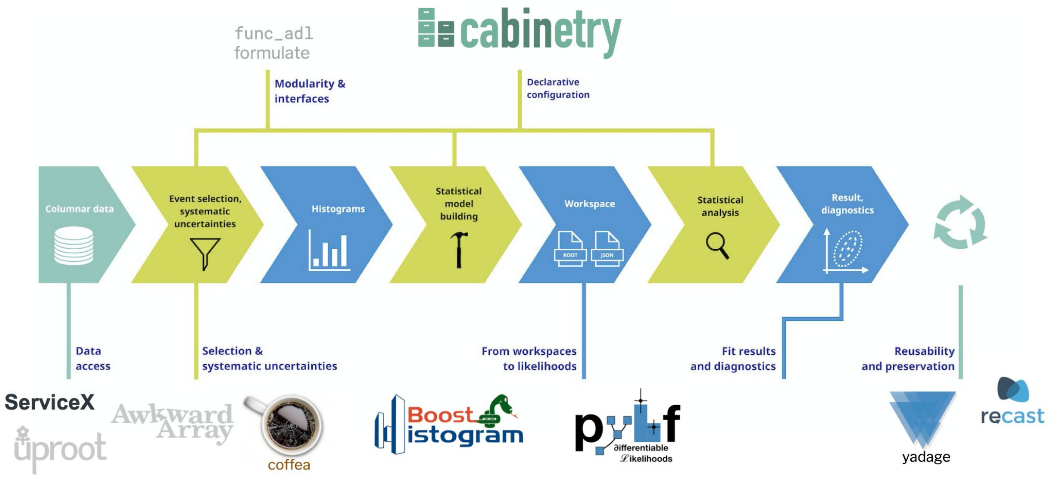
Over **52 GB/s**
of peak throughput



RDataFrame analysis parallelised with Dask backend

Analysis Ergonomics and Scaling

- Analysis facilities may offer specialist solutions to the different working point of analysis vs. other workflows
  - E.g. coffea-casa prototype with columnar backend
- Standard candle analysis benchmarks help compare approaches and Analysis Grand Challenges (IRIS-HEP) test solutions end-to-end

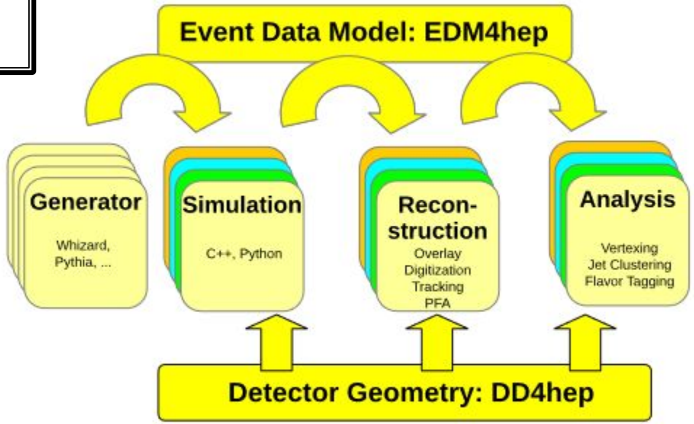


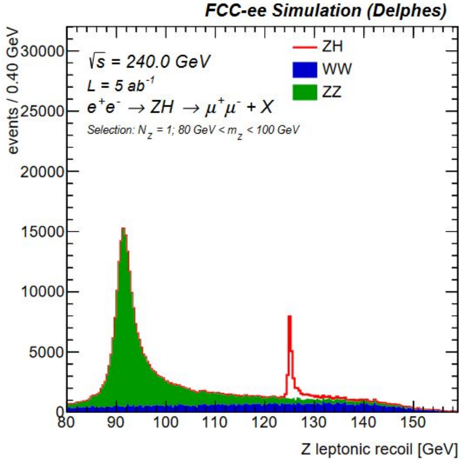
Analysis Grand Challenge incorporates many software elements into a full workflow, from source data access to analysis preservation

Key4hep

More details in Key4hep talk

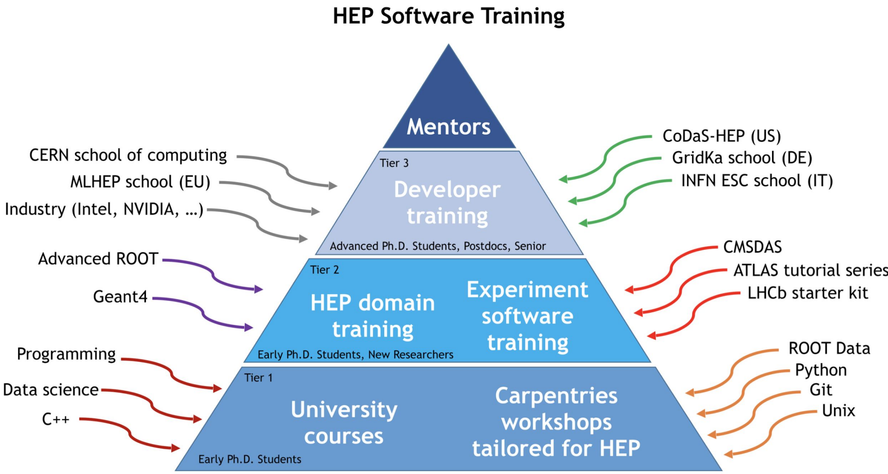
- HEP software stacks are wide and deep - many dependencies
- Want to be able to run full chains for detector design studies easily and in a validated setup
- Ingredients
  - Event data model, EDM4hep based on LCIO and FCC-EDM
  - DD4hep for geometry
  - Gaudi event processing framework
  - Packaged and deployed using Spack
  - Fast (Delphes) and full (Geant4) simulation available
- Contributions from ILC, CLIC, FCC and CEPC communities and from CERN EP R&D and AIDAinnova





courtesy of C. Helsens

Training and Careers



- Many new skills are needed for today's software developers and users
- Base has relatively uniform demands
  - Any common components help us
- LHCb StarterKit initiative taken up by several experiments, sharing training material
- HSF Training Group runs Software Carpentries and other tutorials (co-organised between the HSF and IRIS-HEP)
- Highly successful C++ training courses (from SIDIS and HSF)
  - Inspires continued curriculum development and sharing material
- Assembling a complete curriculum for training in HEP, using Carpentries templates as part of addressing this need [10.1007/s41781-021-00069-9]

Conclusions

- Future physics projects, from upgrades to new facilities and experiments requires first class software and computing to succeed
  - Has to address the challenges of effectively using modern processing hardware
- Many R&D projects have been successful at receiving funding and are working on the *wide range* of problems to be tackled
  - From event generation to analysis preservation
  - Much of this work happens cooperatively
  - Strongly embedded within experiments
  - Even work on far-out things like *Quantum Computing* are starting
- HEP Software Foundation helps collaboration and building community and tacking specific challenges like training

Software and Computing remains a vibrant and essential part of the future of High-Energy Physics

iris hep

HSF
HEP Software Foundation

SWIFT HEP
SoftWare InFrastructure and Technology for High Energy Physics

ErUM-Data IDT

SIDIS
Software Institute for Data Intensive Sciences

ESCAPE

AIDA innova

EP-R&D
Programme on Technologies for Future Experiments