Sustainability and Future of Software Frameworks

Graeme Stewart, CERN EP-SFT & HSF

JENAS Symposium, Madrid, 2022-05-05

Caveat Emptor

- **I will speak about some specific software in High-Energy Physics**

    - Which has a significant overlap with nuclear physics

    - And I can't mention all projects

- **Generally, the arguments made here are applicable to other scientific software domains**

    - Astro-particle observatories and projects face many of the same issues

        - Particularly as data rates and volumes rise

    - Despite domain specific matters, many of the development trends and solutions will be very similar

    - We all, in common, face the problem of training, recruitment and retention of excellent people who can write and maintain scientific software
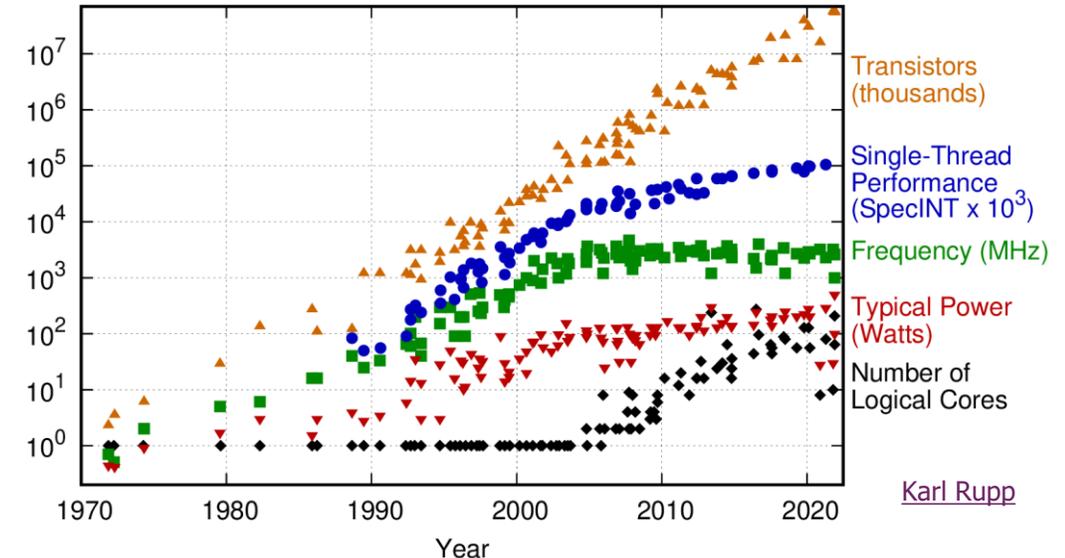
Technology Context

- **Once we move away from the detectors all software and computing is being done mostly commodity off-the-shelf hardware (COTS)***

- **The evolution of this technology over the last decades has continued to follow Moore's Law**

  - We still get more transistors with each processor generation

  - But clock speed has been stalled now for many years

  - This limits the throughput of any serial codes

    - We have to adopt parallel processing paradigms

* Data centre GPUs, FPGAs, tape systems would be the exceptions

50 Years of Microprocessor Trend Data

Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
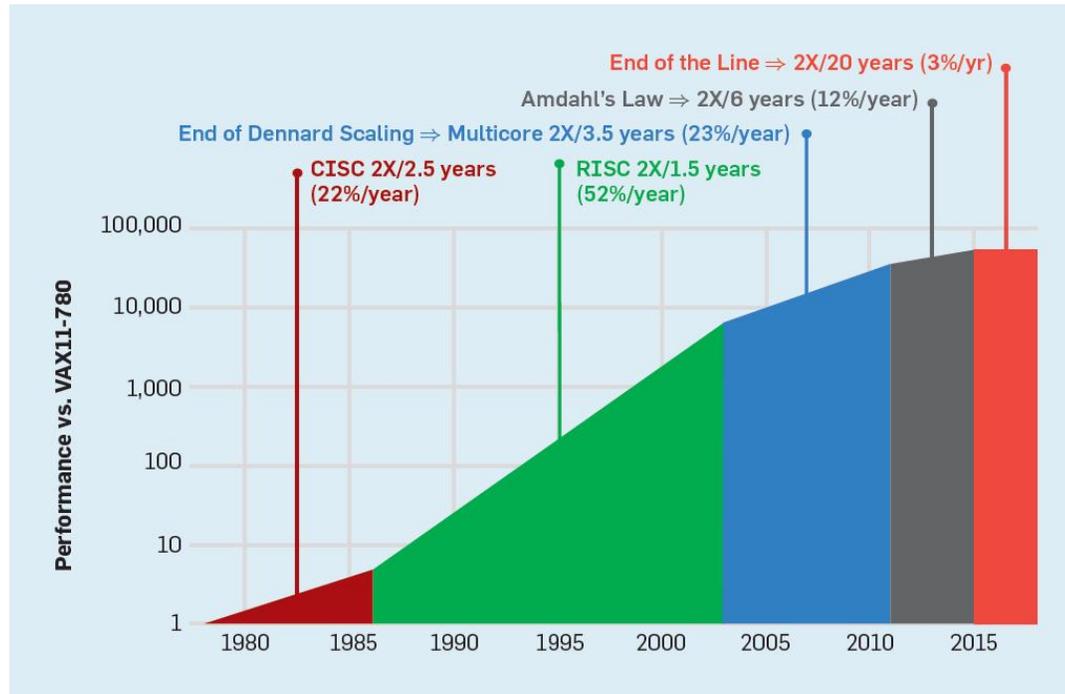Typical Power (Watts)
Number of Logical Cores

Karl Rupp

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 2010-2021 by K. Rupp

New process schedules for future chips

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

Alternative Processor Architectures

End of the Line ⇒ 2X/20 years (3%/yr)
Amdahl's Law ⇒ 2X/6 years (12%/year)
End of Dennard Scaling ⇒ Multicore 2X/3.5 years (23%/year)
CISC 2X/2.5 years (22%/year)
RISC 2X/1.5 years (52%/year)
Performance vs. VAX11-780

A New Golden Age for Computer Architecture, Hennessy, Patterson 2019

Efficient use of these computing architectures significantly lowers energy costs – Green Computing

- **Traditional CPU architectures are no longer the optimal choice for many problems**
  - Albeit that there is innovation here, e.g., ARM processors such as Apple M1, Fujitsu A64FX; AMD x86 CPUs. (CPU has a SIMD model, a.k.a. vectorisation)

- **By devoting more transistors to arithmetic operations, GPUs can achieve a higher computational throughput**
  - Single instruction multiple thread model (SIMT)
  - Albeit with significantly more work from the software developer
  - GPUs are not the only option either: FPGAs, TPUs, etc.

- **Data intensive codes have to work hard to mitigate data movement costs**
  - On the CPU as well as the GPU
  - Poor memory access patterns will destroy performance

Machine Learning and Other Technologies

- **Machine Learning**
  - Used in HEP for a long time
  - Discrimination, classification, anomaly detection are all in use in HEP
  - Centre of cutting edge software outside HEP, developed and supported by industry
    - Our models are generally very simple, cf. GPT-3 (autoregressive language generator) that has 175 billion parameters

- **Auto-differentiation**
  - Neural networks (amongst other things) are differentiable
  - Can evaluate the relationship between underlying calculations and observables in both directions
    - Requires the development of *surrogate models* for non-differentiable parts
  - But with this could do an end-to-end optimisation of everything from analysis to experiment design (MODE talk, Pietro Vischia)

- **Quantum Computing**
  - Definitely a very hot topic (see previous talk from Philippe Chomaz)
  - Some models now appearing for HEP applications, many training NNs
  - Remains to be seen if this will really be a revolution for software (and on what timescale)
    - Doing theory calculations directly on a quantum computer would be really exciting!

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%

Radovic et al., [10.1038/s41586-018-0361-2]
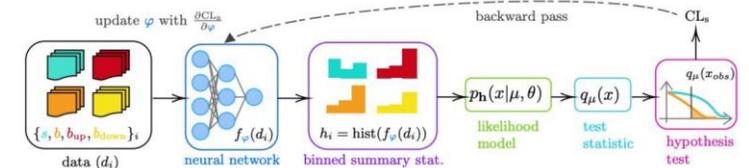
Auto-differentiation adds a backwards pass to optimise an analysis, Simpson and Heinrich [2203.05570]

End-to-End Data Storage and Networks

- **For the LHC storage is actually the dominant capital cost**

- **Hierarchical storage is necessary to control this cost and adapt to technology changes**
  - Tape systems offer the highest capacity per unit cost and highest data safety
  - Spinning hard disk drives are still the best choice for bulk online storage
    - Even if the consumer market has gone to SSDs, this technology does not go away
    - HAMR drives offer even greater capacity, up to 20TB
  - SSDs become faster (NVMe2) and larger capacity
    - But for now remain more costly than HDDs

- **Network capacity has grown enormously**
 - This has significantly changed how we manage data and computing for big experiments

ATLAS Data managed by Rucio, more than 500PB

Monday, Jan 25, 2021
• Bytes: 501 739 234 047 197 500
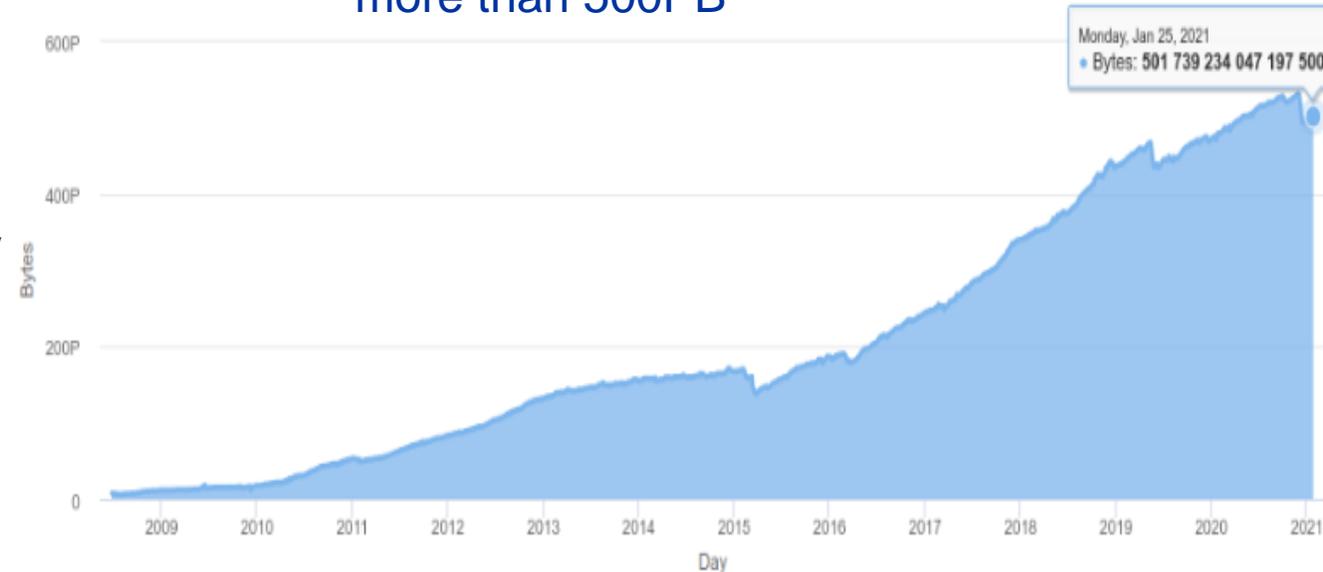
Resource scaling and computing

- **Data intensive sciences use significant computing resources**

  - The number of high data-rate experiments and projects is only growing

  - ESCAPE has helped to bring considerable harmony in the data management area

- **Computer centres serve multiple communities and provide resources that match their integrated needs**

  - E.g., now commonly including GPUs (which handle ML tasks very well)

  - Larger centres in the future, but more diverse hardware probable

- **High-performance computing centres (HPCs) are funded and used in many regions now**

  - Our software needs to adapt to these resources

    - GPUs are very prevalent now – in multiple flavours

    - Non x86 CPUs

  - *Large scale heterogeneous computing is the new model*
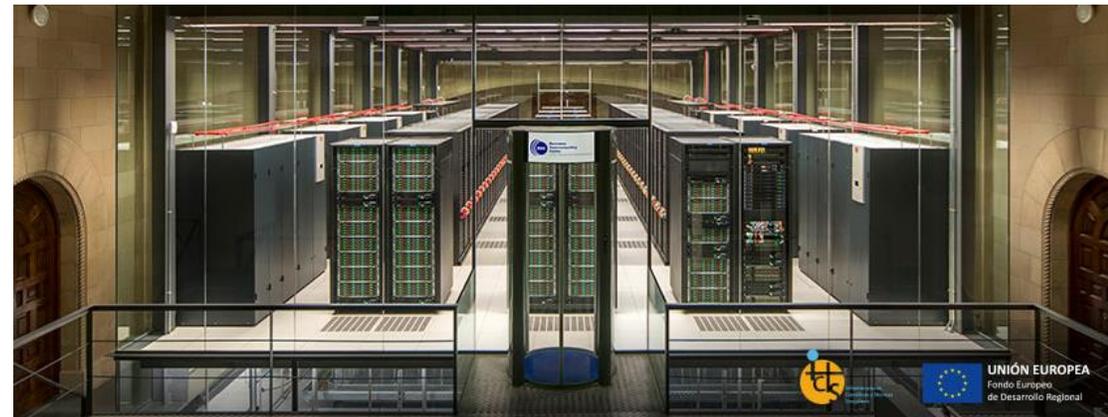
European Strategy Update 2020

2020 Strategy Statements

European Strategy Update

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.*
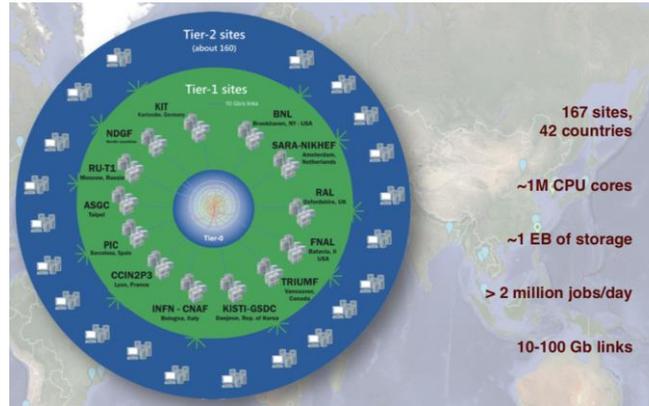
HEP Software Frameworks in a nutshell...

- **Software is involved at every level**
  - Event generation
  - Detector Simulation
  - DAQ and Triggers
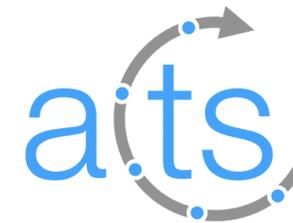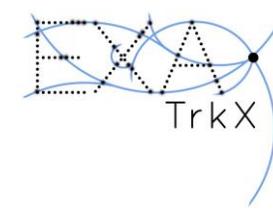 - Reconstruction
  - Analysis

  >50M lines of code

- **Resource consumption implied can be massive, e.g., for the LHC experiments now with DUNE on the horizon**
  - Use of exabytes of storage and millions of CPU cores

- **Balance of consumption in each step can vary from experiment to experiment**
  - Guide optimisation by total cost (e.g., EM calorimeter simulation)

Event Generation Challenges

- **High luminosity LHC will require a very good control over uncertainties**

  - Theory errors are very significant

  - Need to move from LO calculations to NLO and even NNLO

  - These calculations are inherently more demanding

    - And introduce problems of negative weight events from MC@NLO matching

    - Dilutes statistical power and also computationally heavy as negative probability events need to be simulated and reconstructed as well

$$\frac{1}{\sqrt{N} \cdot (1 - 2f_{nw})}$$

- **Generally there is a mismatch between the incentives for the theory community and the needs of the experimental community**

  - Work in the HSF and SWIFT-HEP is helping to bridge this gap
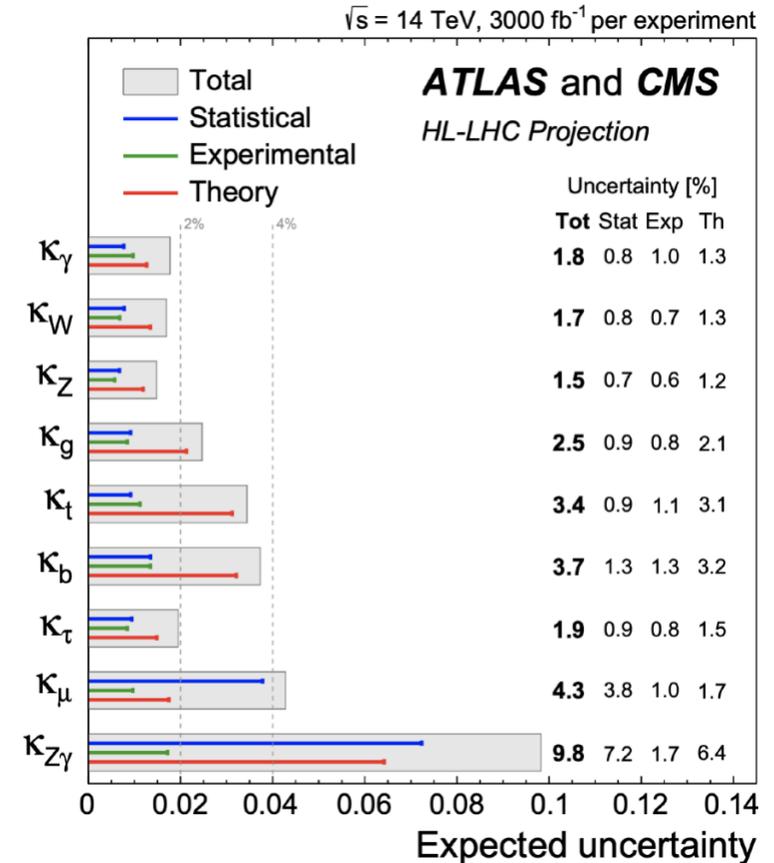
Event Generator Improvements

- **Negative weights**
  - 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	MC@NLO-Δ
	111	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)
$pp \to H b\bar{b}$	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)

Also excellent results on vectorised CPUs, x3.5 on avx2

Further Event Generator Improvements

- **SWIFT-HEP profiling 2 years ago showed a suspicious amount of LHAPDF CPU**

  - 60-70% of CPU time used when generating multi-jet events in Sherpa+OpenLoops for ll+0,1,2@NLO+3,4,5@LO

- **How to optimise this?**

  - Make LHAPDF faster (optimise interpolation and transcendental calls)

    - Restructured LHAPDF calls to improve interpolation code, 30% speed-up

  - Call LHAPDF less – do a minimum generation for the event, up to unweighting

    - Then only calculate the full event when it is accepted (rewind RNG state to do this)

  - Use analytic one-loop amplitudes, when available

Detector Simulation – Machine Learning

- **A major consumer of LHC grid resources today**

  - Experiments with higher data rates will need more simulation

  - Simulation will also need to be more accurate (e.g., recent Geant4 improvements in release 11.0)

- **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)

- **Recent LPCC workshop on fast simulation provides a good overview**

  - *Machine learning* lends itself to problems like this

    - Calorimeter simulations usually targeted, due to their high resource consumption

  - 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/s41781-021-00056-0]

  - Retrainable networks allow faster adaption to new layouts, e.g., a different calorimeter geometry

- **Hybrid models are also an option, e.g., ATLAS AtlFast3, combining parametric and ML approaches [2109.02551]**

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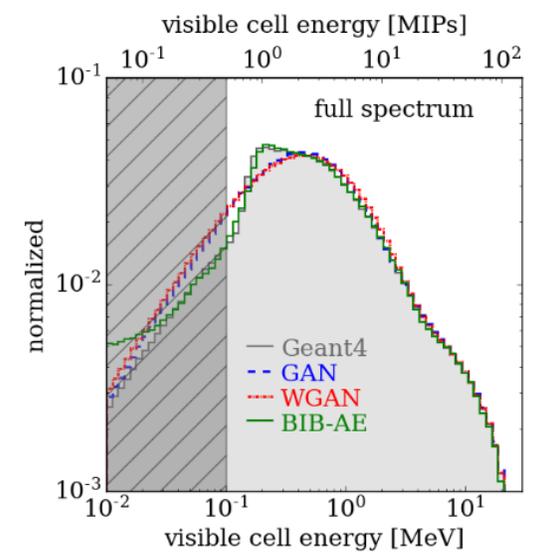

Meta Learning allows for fast adaption of a model from one calorimeter to another, Salamini & Zabrowska
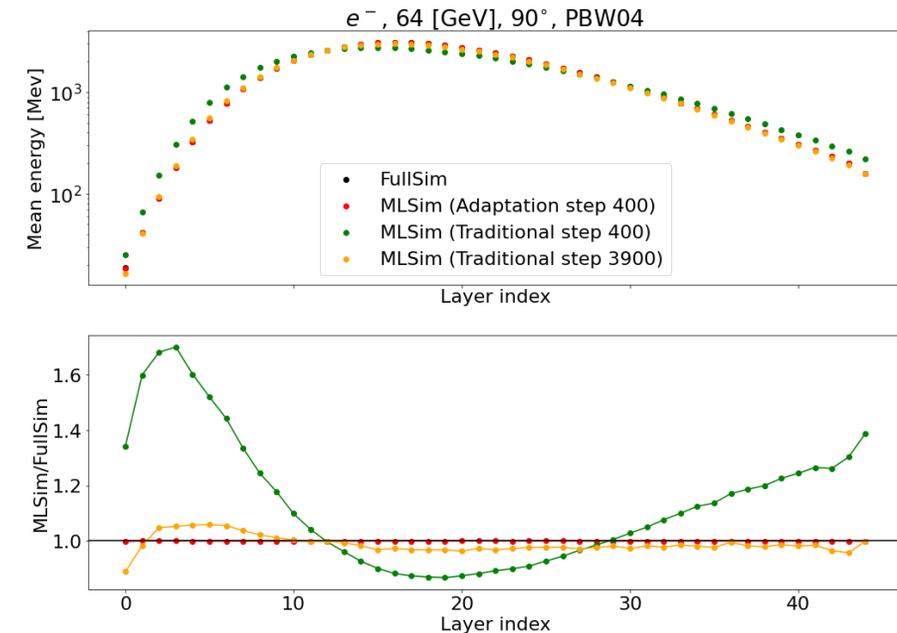
Detector Simulation on GPUs

- **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)
    - HSF Community Meeting this week to review progress
      - Can duplicate HEP EM physics extremely well cf. Geant4
      - Can **beat the speed of a multi-core CPU on a GPU** in a simple geometry
      - Complex geometry currently is a significant bottleneck

- **Opticks, using ray tracing on GPUs for optical photons, takes advantage of native ray tracing on GPUs**
  - Developed for JUNO (Simon Blyth, 10.1051/epjconf/202125103009) and integrated into Geant4 for DUNE (w. Hans Wenzel)

Physics results with G4HepEm library running on CPU and GPU, cf. Geant4
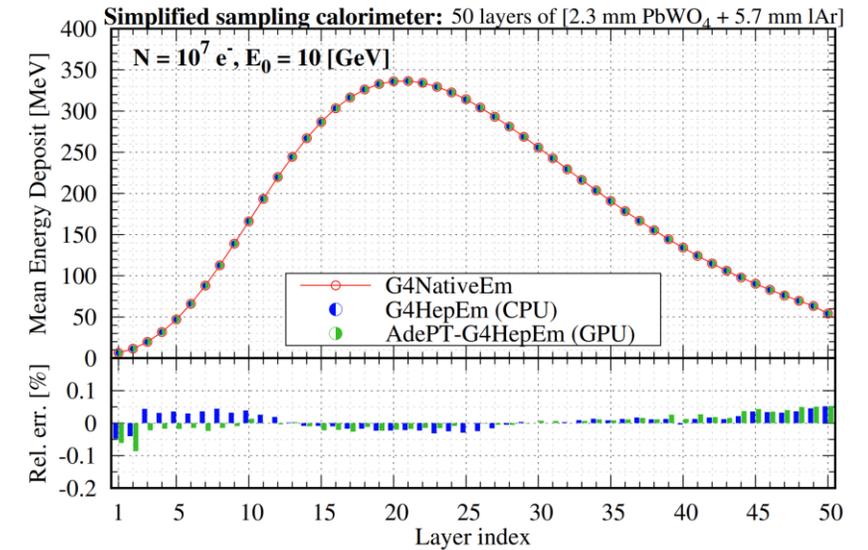
Speedup in optical photon simulation in LAr with GPU (x378) and ray tracing GPU (x900) cf. Geant4 on single CPU core

| 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

Reconstruction

- **New detectors are optimised for high-rate data acquisition**
  - Physics needs require high quality selection at close to beam-crossing rates – software triggers
  - This is drinking from the fire-hose at 30MHz (LHCb have MHz rates of charm and beauty)

- **Break with the past and rewrite software targeting GPU architectures**
  - Pioneered by ALICE in Run 2, now revamped for Run 3; being introduced by CMS too (Patatrack project)
  - LHCb have a new implementation of their HLT1 running on *GPU*, the Allen framework
    - GPUs integrated into event builder nodes, up to 3 GPUs per server
  - Lessons learned: keep data model simple, bulk data, be asynchronous, minimise data transfers
    - Most work is in re-thinking algorithms and data flow, not in cross-architecture portability

- **Allows for real-time analysis**
  - Keep partial event information for many more events recorded

Reconstruction

cern.ch/go/Q897
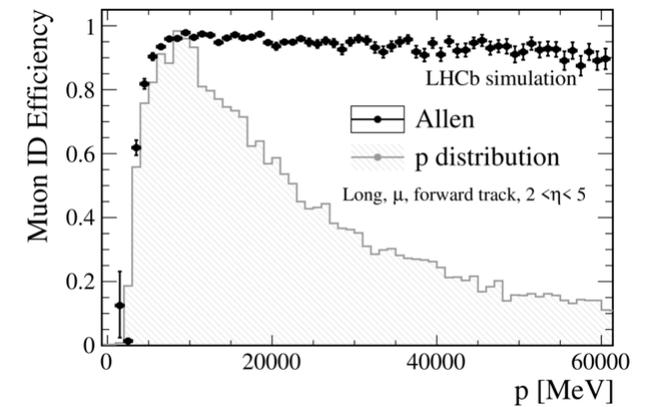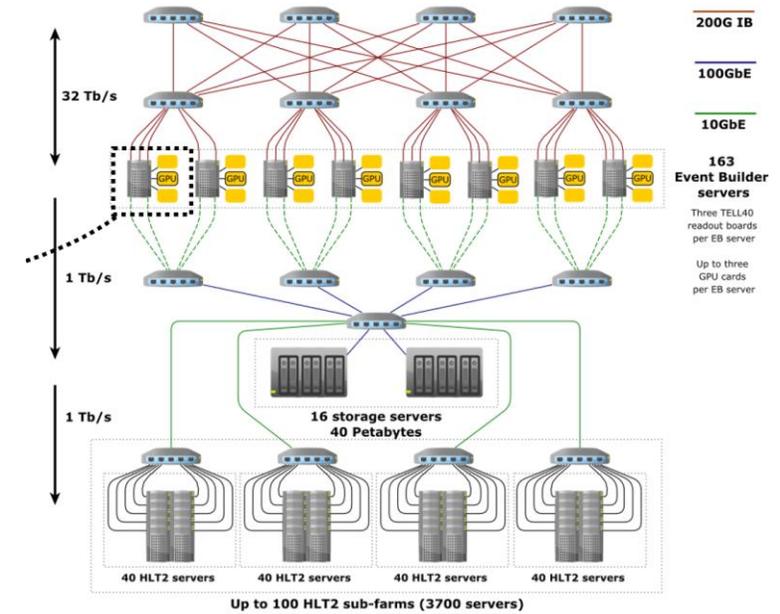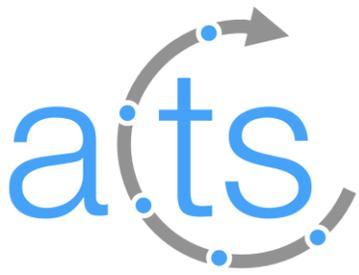
- **Traditionally reconstruction software has been rather experiment specific**
  - Grimy details of sub-detectors, calibrations and specific geometries drove specific implementations…

- **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]
    - *Now incorporated into Acts stack*
  - Impressive parallelisation results from 'ground up' thread safety
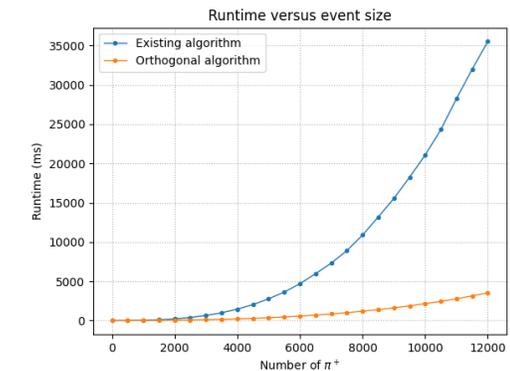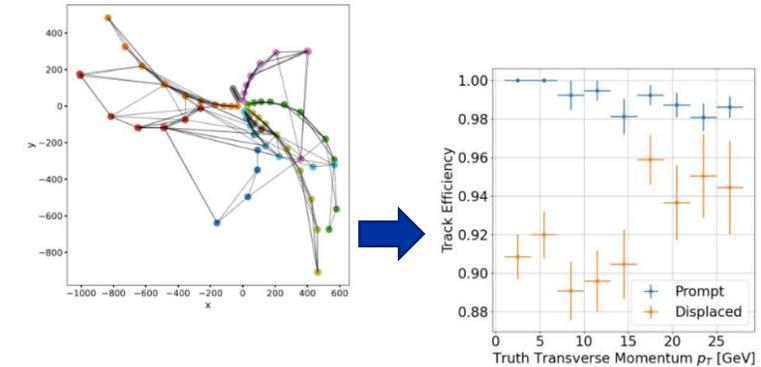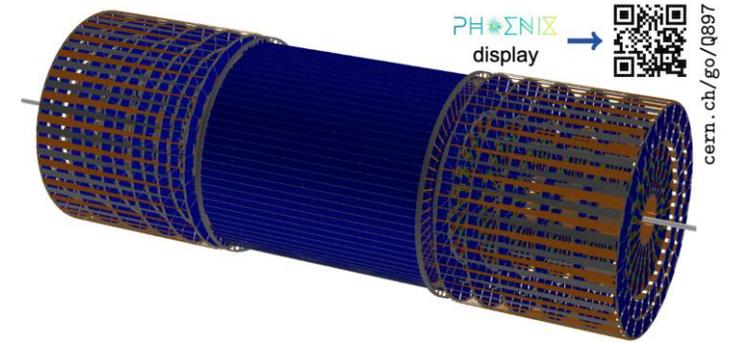
New seed finding strategy on based of KD-trees, achieving very similar tracking performance

HEP Data Analysis

- **Scaling for analysis level data also a huge challenge for experiments**

  - Data must also be kept readable and accessible for decades afterwards (FAIR principles)

    - Nothing is sustainable if we can't read our data!

  - 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 *smaller on disk* and *faster to read* than older formats or industry alternatives

    - Adapted for modern object stores systems

    - Comes with a 'lifetime guarantee' to be supported
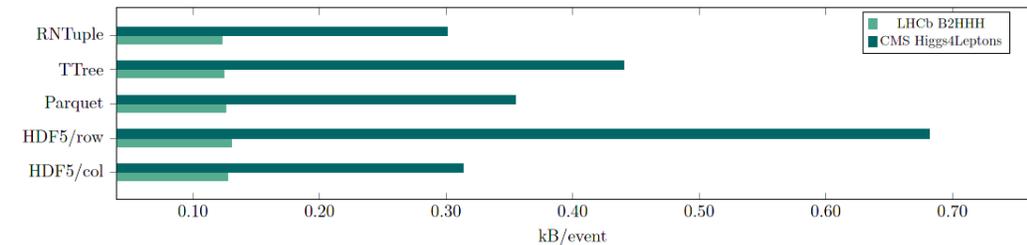
- **Reducing volume of data needed helps hugely**

  - CMS ~1kB nanoAOD [10.1051/epjconf/201921406021] makes a vast difference to analysis efficiency and "papers per petabyte"
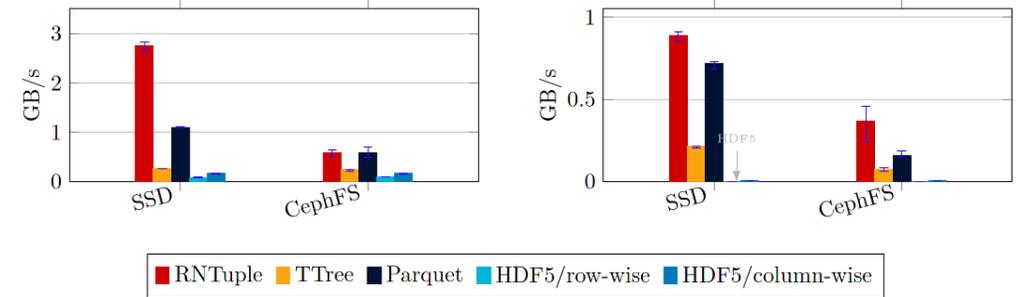
Read speeds comparing new ROOT RNTuple with older HEP TTree format and other industry formats [Gomez, Blomer] (note y-axis is log10 scale)

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

# Analysis Ergonomics and Scaling

```python
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()
```
Python

Declarative approach to systematics with
RDataFrame in ROOT

- **Improving how analysis is done increases productivity**
  - Rise of declarative models
    - E.g., ROOT's RDataFrame, Coffea
  - 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

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

Over **52 GB/s**
of peak throughput
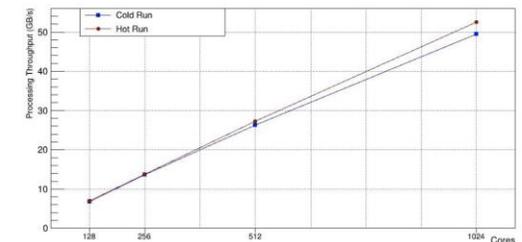
RDataFrame analysis parallelised with Dask
backend

Analysis Ecosystem

- **Analysis facilities may offer specialist solutions to the different working point of analysis vs. other workflows**

  - coffea-casa prototype with columnar backend

  - SWAN with EOS backed storage

  - Here notebooks are commonly used as interfaces for the analyst

- **Standard candle analysis benchmarks help compare approaches and Analysis Grand Challenges (IRIS-HEP) test solutions end-to-end**

- **Upcoming Analysis Ecosystems Workshop 2 (HSF, IRIS-HEP) will discuss all of this**
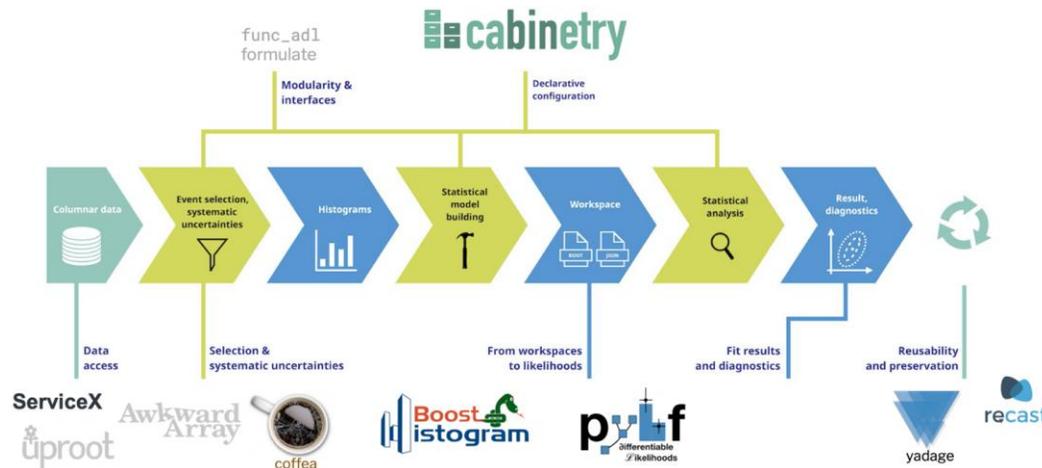
Analysis Grand Challenge incorporates many software elements into a full workflow, from source data access to analysis preservation

HEP Software Stacks – Key4hep

- **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**
- **Key4hep 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
- **Developed in AIDAinnova and EP R&D; contributions from ILC, CLIC, FCC and CEPC communities**
- **Deployed via CVMFS to resources worldwide**
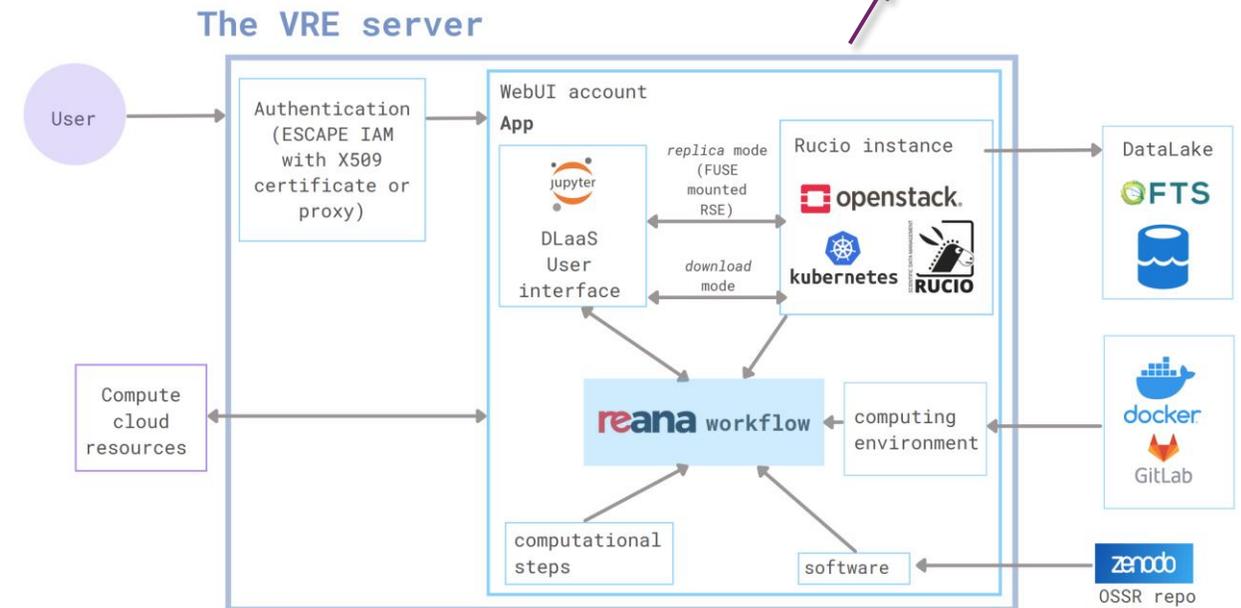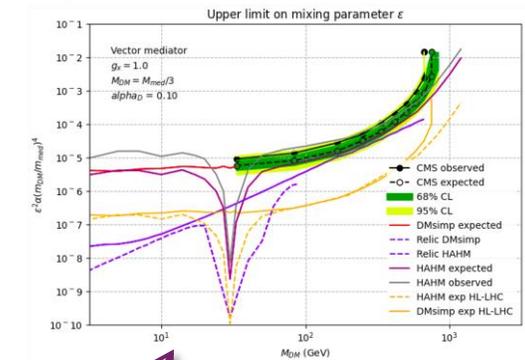 - This also is a stepping stone to preservation
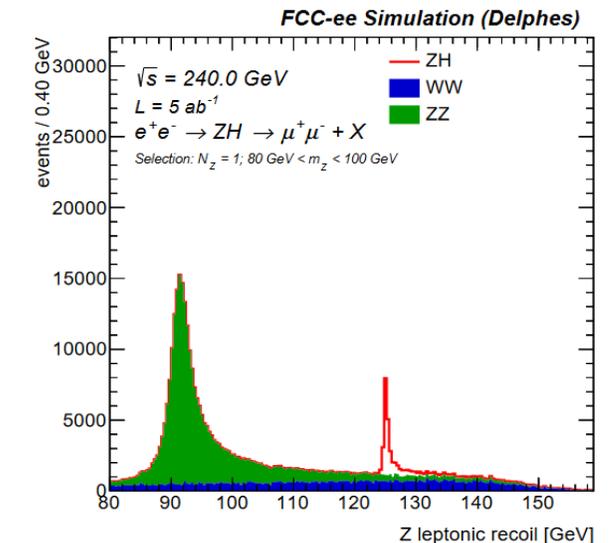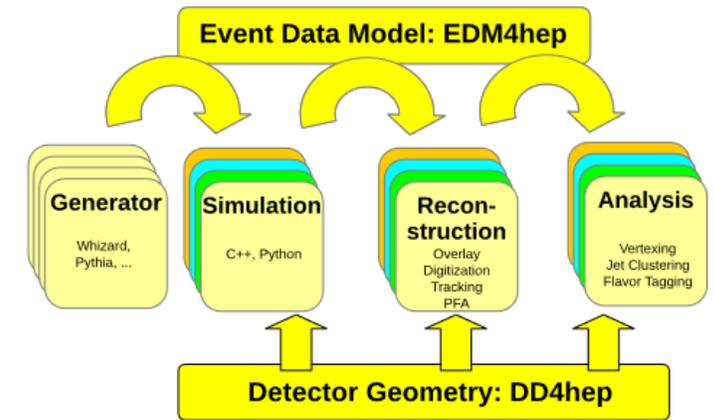
courtesy of C. Helsens

EOSC-Future Test Science Projects

For an example of the physics cases and VRE usage, see posters by J. Greaves and J. Little

- **Two test science projects** (see also G. Lamanna's **ESCAPE talk**)
  - Extreme Universe
  - Dark Matter
- **Test science projects will connect resources and enable scientific analysis at scale**
  - Prototyping use of the European Open Science Cloud
- **Use/develop a Virtual Resource Environment**
  - Experimental data (detector and simulation)
    - Ambition to starting from different data formats (raw/derived/likelihood level)
  - Software for analysis
  - Computing resources
  - Workflow management
  - Demonstrate end-to-end analysis
   - With preservation 'built-in'

E. Gazzarrini

Software Sustainability

- **Sustainability is a complex matter, with interactions between many parties**
  - Funding agencies, funded projects, software engineers and developers, resource providers, …

- **Key pieces**
  - **Software Engineering**
    - Good code, built with best practices
    - Even existing code needs to be maintained (new compilers, platforms, library versions)
  - **Community**
    - People feel involved in the projects and between projects
    - Open source and welcoming – lower barriers to entry
  - **Training**
    - Equipping developers with the skills they need
    - Recognising that scientific software is developed by many people at different expertise levels (domain knowledge is critical!)
  - **Policy**
    - Is software recognised and funded?
    - Are there career paths for those at the heart of our software projects?
      - Research software engineer model is a step forward
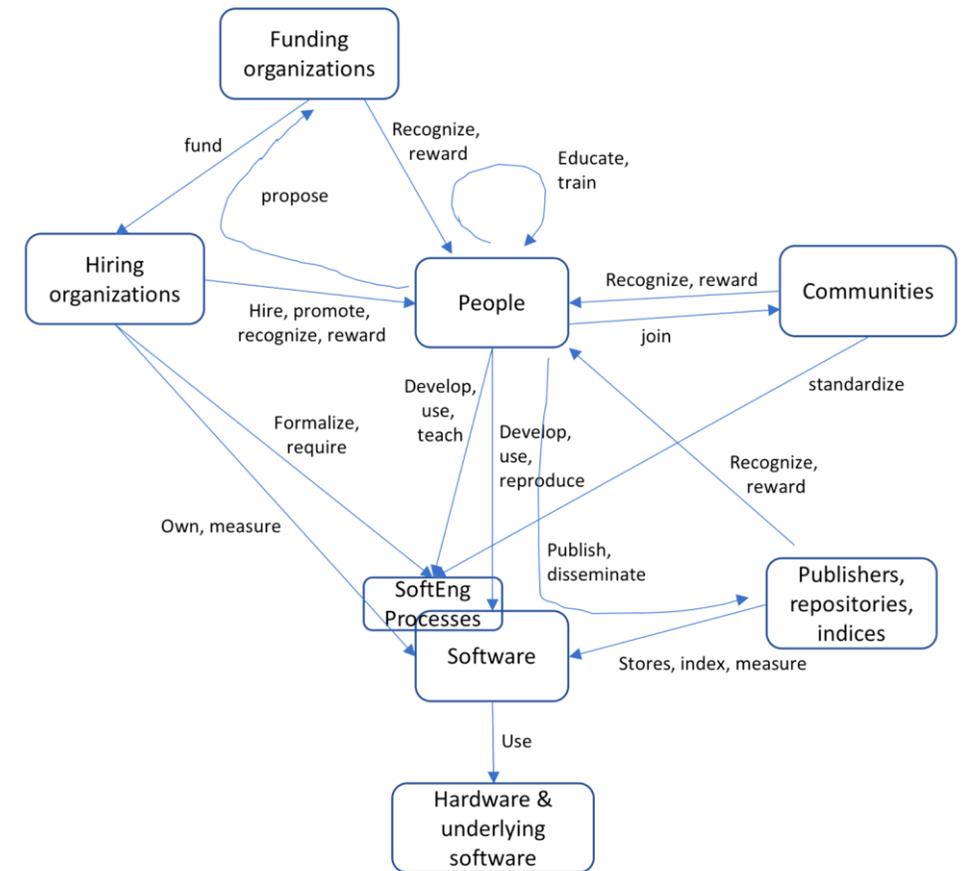
Mapping the research software sustainability space, Druskat and Katz
https://doi.org/10.1109/eScience.2018.00014

Software and Workflow Sustainability

- **Sustainability requires commitment**
  - There is often a chicken and egg problem…
    - *I won't use your software because it's not maintained long term* – Experiment computing coordinator
    - *I won't fund your software long term because it's not used by the experiments* – Funding agency

- **Projects that have staff on long-term/permanent positions can make firmer commitments into the future**
  - E.g. ROOT and Geant4 supported by CERN (and others!)
  - From these core commitments communities are certainly easier to sustain long term
    - Communities develop: User ➔ Contributor ➔ Developer ➔ Leader (reduce friction along this path)
  - These projects are critical for the long term health of the field – stability here matters!

- **We need also to promote innovation and allow projects to take risks**
  - But with credible paths to sustainable outcomes

- **Just sustaining software isn't enough, we also need to preserve data and analysis workflows – for many decades**
  - Requires support for (open) data, support infrastructure (e.g., REANA) and documentation

Training

HEP Software Training

- **Many new skills are needed for today's software developers and users**

- **Base has relatively uniform demands**
  - Any common components help us

- **HSF Training Group runs Software Carpentries and other tutorials (co-organised between the HSF and IRIS-HEP, with key projects like ROOT)**

- **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]**
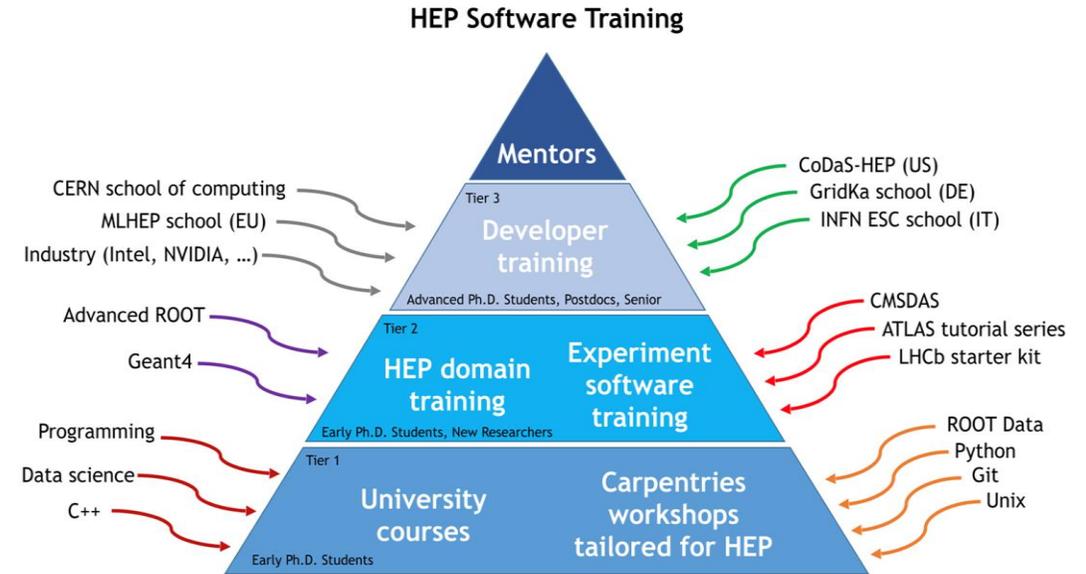
HSF and Software Projects

- **Since the publication of the Community White Paper in 2018 [10.1007/s41781-018-0018-8] there has been growing awareness of the need to invest in software for the future**
  - A number of important projects have been funded
    - IRIS-HEP, NSF
    - SWIFT-HEP, STFC
    - HEP-CCE, DOE
    - EP R&D Software, CERN
    - AIDAinnova, EU

- **These projects are having a significant positive impact on software for HEP**
  - The sustainability of the software and people they develop needs to be addressed

- **HSF provides a meeting point for exchange of information and interaction between these activities, through working groups, fora and workshops**
  - Our key objective is to form a community for HEP software

Conclusions

- **HEP has a huge investment in software**

  - Our science just wouldn't be possible without it

- **We face challenges from**

  - Ambitious physics programmes, with higher data rates and more stringent requirements
  - Technology evolution towards parallel architectures, particularly GPUs
  - This is common across disciplines

- **Important software projects have been funded in recent years**

  - In combination with the experiments and the existing projects these are helping our software to advance
  - The HSF, though largely unfunded, has become a place to help build the community and share knowledge

- **Sustainability requires a commitment to people more than anything**

  - We will take care of the technical matters, but there is a real need to find people sustainable careers