HEP Software R&D

Graeme A Stewart, CERN EP-SFT

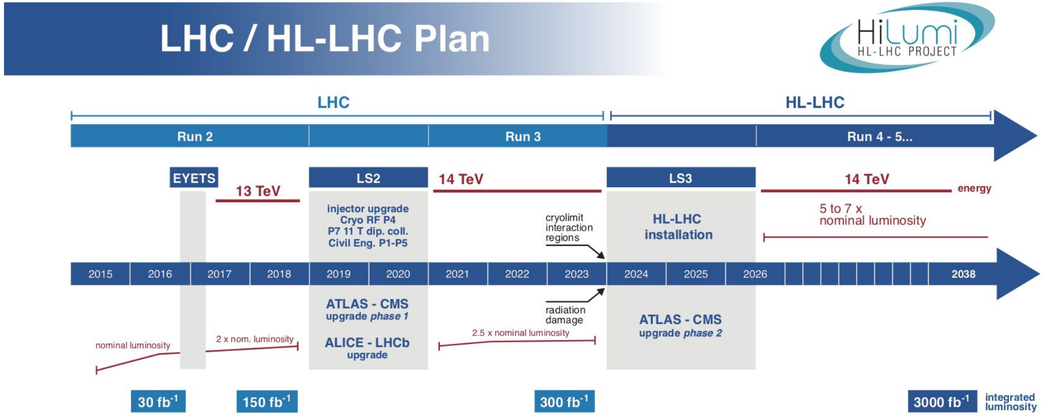
2019 International Workshop on the High-Energy Circular Electron-Positron Collider
2019-11-18

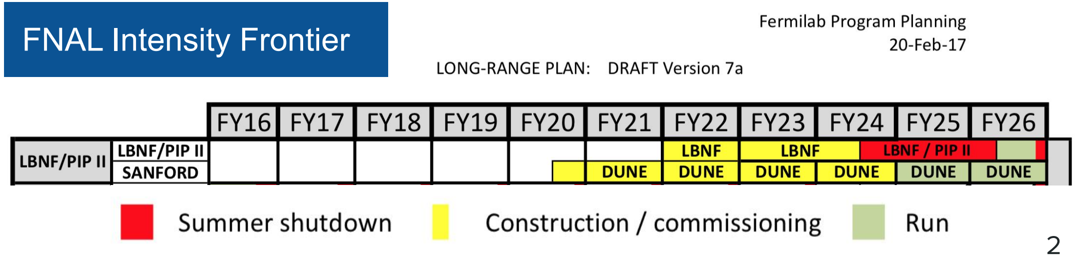
HL-LHC and the Intensity Frontier

The mission:

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

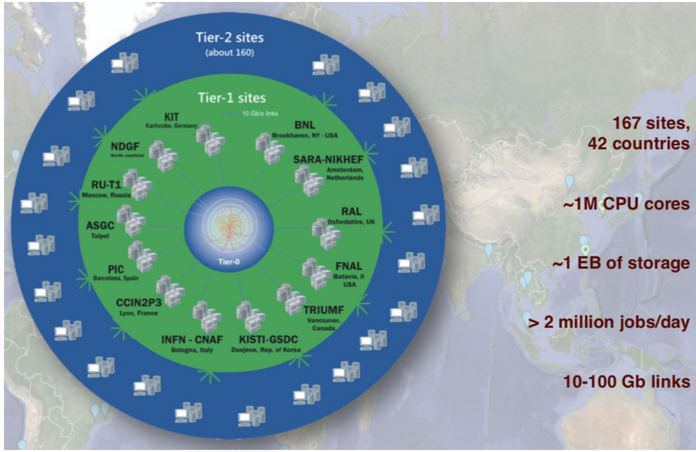
Rich physics programme that is highly reliant on software to achieve its goals





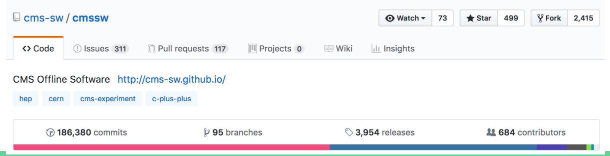
HEP Software and Computing Today

- 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 100PB 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



athena

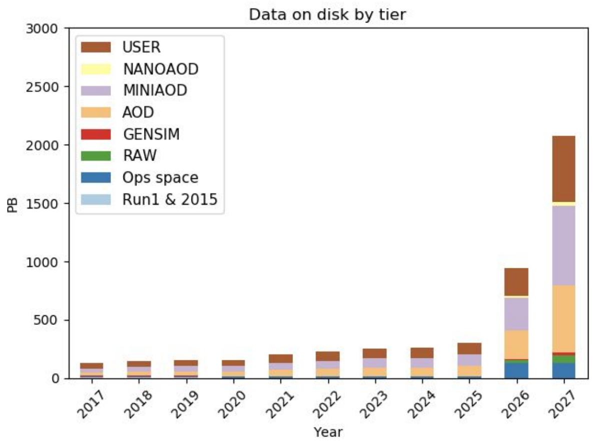
ATLAS Experiment main repository for Athena code

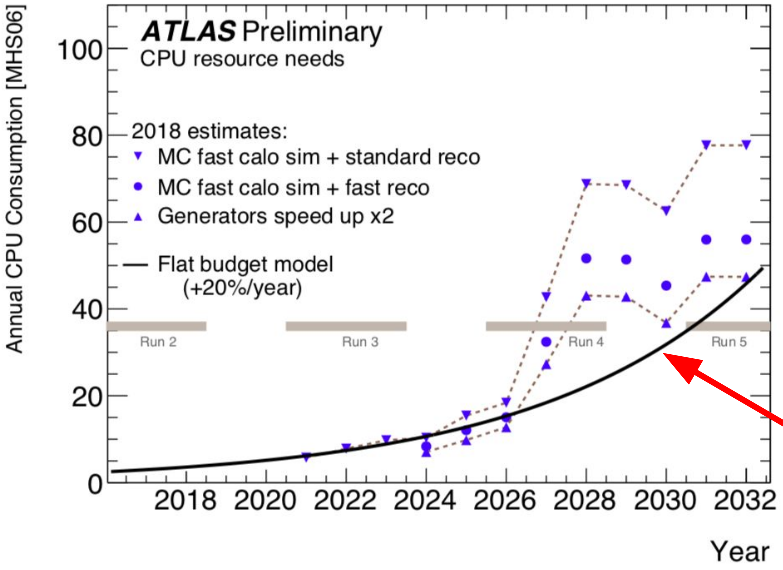


Challenges for the Next Decade

- HL-LHC brings a huge challenge to software and computing
  - Both rate and complexity rise

CMS



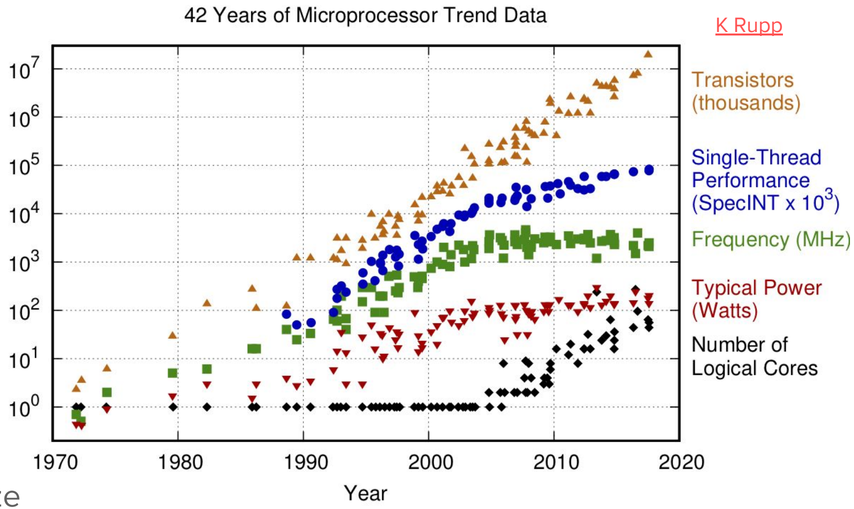


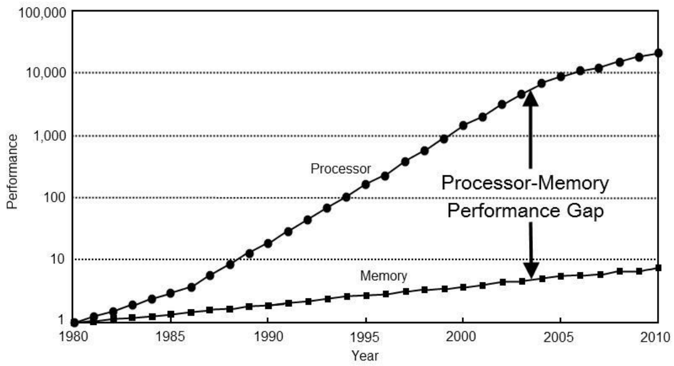
- Not just a simple extrapolation of Run 2 software and computing
  - Resources needed would hugely exceed those from technology evolution alone

This is even probably too optimistic, ~5-10%?

Technology Evolution

- Moore's Law continues to deliver increases in transistor density
  - But, doubling time is lengthening
- 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

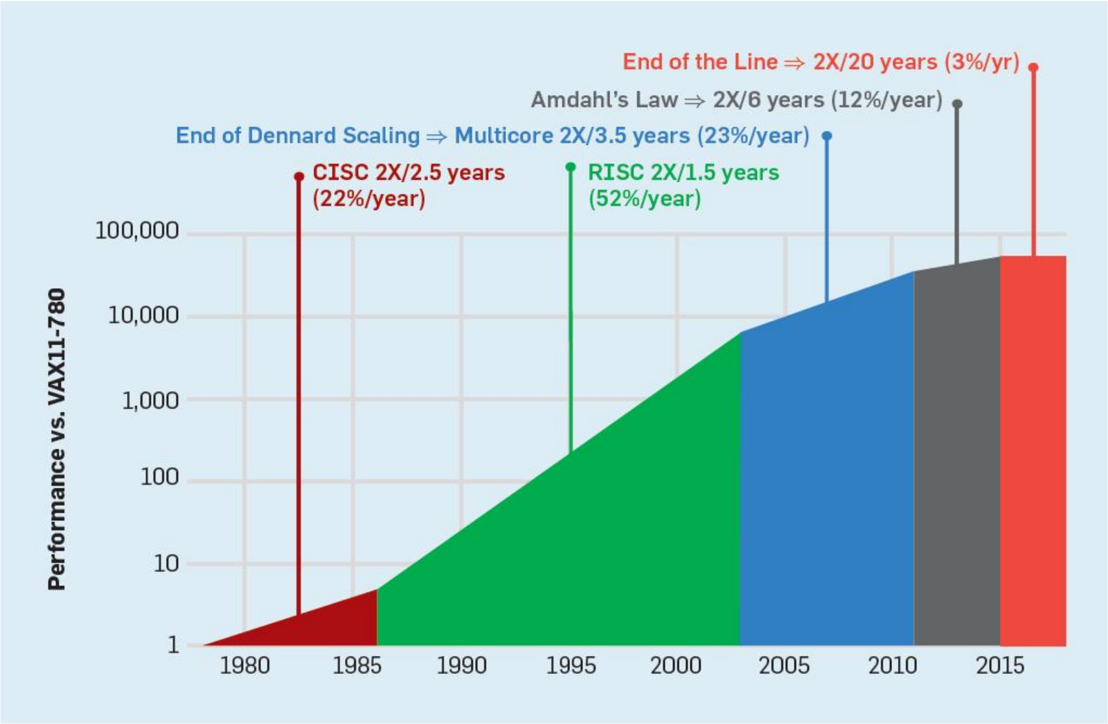




Decreasing Returns over Time

- Conclusion is that diversity of new architectures will only grow
- Best known example is of GPUs

[link]



GPUs dedicate far more transistors to arithmetic

Drivers of Technology Evolution

- Low power devices
  - Driven by mobile technology and Internet of Things
- Data centre processing
  - Extremely large clusters running fairly specialist applications
- Machine learning
  - New silicon devices specialised for training machine learning algorithms, particularly lower precision calculations (e.g. `bfloat`)
- Exascale computing
  - Not in itself general purpose, but poses many technical problems whose solutions can be general - HEP pushed to use HPC centres, especially in US; most computing power in accelerators
- Energy efficiency is a driver for all of these developments
  - Specialist processors would be designed for very specific tasks
  - Chips would be unable to power all transistors at once: dark silicon is unlit when not used

Future Higgs Factories

- Next machine after HL-LHC likely to be a Higgs factory, e^+e^- collider
  - Circular or linear not known at this point
  - European Strategy Update ongoing and US Snowmass just beginning
- Event complexity is not at all as high as for hadronic machines
- However, detector resolutions and reconstruction complexity can be high
  - High-granularity calorimetry is very likely
  - Pure software reconstruction is quite likely
  - Interaction rates for lower energy phases of circular machines (Z-pole) can be extremely high
- We would ignore software and computing challenges at our peril
  - Need to be flexible at this stage to manage uncertainty and share efforts between different options
  - Learn a lot from the experience of the LHC experiments, who have led the transition into a modern computing environment *and continues to evolve*

High-Energy Physics Software Foundation

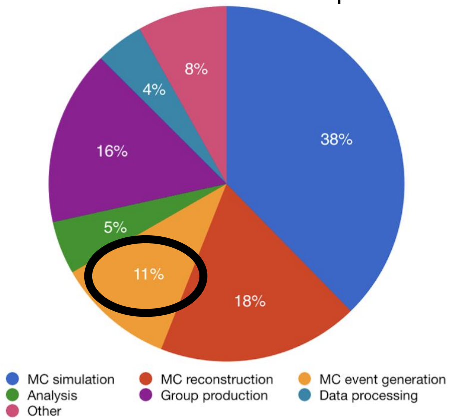
HEP Software Foundation

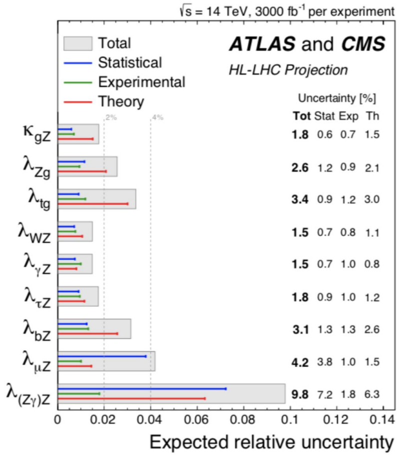
- HSF setup in 2014 as a bottom-up community organisation
- Promote *common efforts* to solve the software problems of high-energy physics
- Produced the Community White Paper in 2017 (https://doi.org/10.1007/s41781-018-0018-8), giving a *roadmap* for HEP software and computing into the 2020s
- Since then HSF Working Groups have been formed, tackling major areas of interest for the field
  - N.B. nominations still open for convenors in a number of working groups
- Regular HSF workshops, often held together with WLCG

Event Generators

- Event generators are the start of the simulation chain
  - At the LHC Run1 only leading order generators were used
  - Negligable CPU consumption compared with detector simulation - no pressure to optimise
- However, with LHC upgrades coming higher order generators become much more important
  - These are inherently much more costly to run
  - Problems of negative weights can increase hugely the samples needed for weighted event samples
- In addition, the theory community, who develop these codes usually work in small teams
  - Recognition for technical improvements is limited/missing

ATLAS 2018 CPU Report

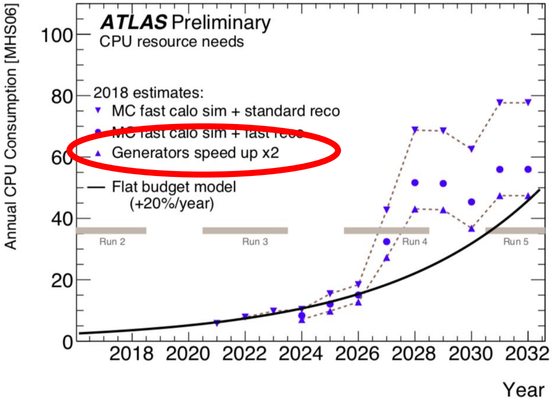


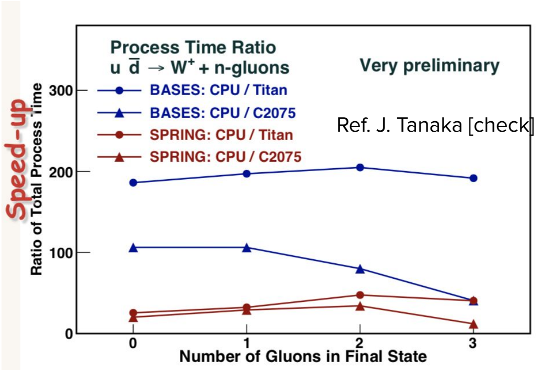


Many electroweak measurement errors dominated by theory (red). B. Hinemann

Event Generators - Technical Improvements

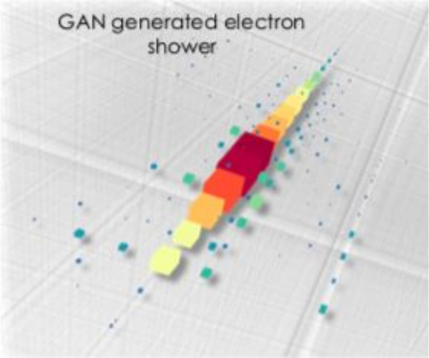
- HSF/LPCC workshop in November brought theory and experiment together to look at computing challenges of event generation
  - This was the first workshop of its kind
- Working group tackling technical challenges
  - Setting a baseline for further comparisons
  - Understanding how to run generators for best efficiency
  - Support for technical improvements (e.g. thread safety)
  - Porting to other architectures
    - Could be very suitable code to do this with (smaller, self contained code bases, numerically intensive)
    - e.g. building on the work done so far in MadGraph with GPUs





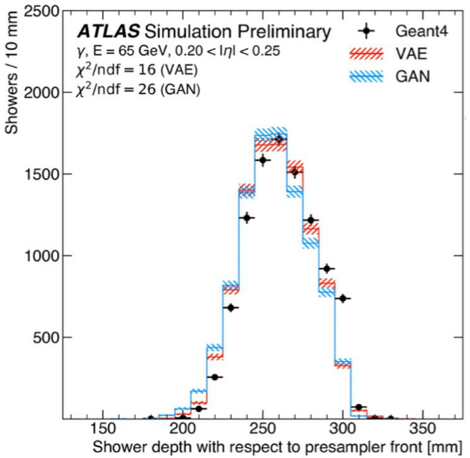
Detector Simulation

- A major consumer of LHC grid resources today
  - Experiments with higher data rates will need to more simulat
- Faster simulation, with no or minimal loss of accuracy, is the goal
  - Range of techniques have been used for a long time (frozen showers, parametric response)
  - Key point is deciding when it’s good enough for physics
- Machine learning lends itself to problems like this
  - Calorimeter simulations usually targeted
  - Variational Auto Encoders (VAEs) attempt to compress the data down to a ‘latent space’ - can be randomly sampled to generate new events
  - Generative Adverserial Networks (GANs) train two networks, one to generate events, the other to try to classify as real/fake
  - R&D on lifecycle integration into Geant4 is starting



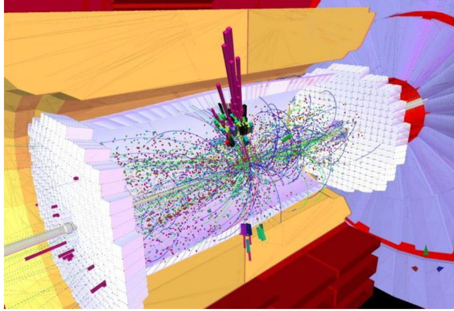
Use of Generative Adversarial Networks to simulate calorimeter showers, trained on G4 events (S. Vallacorsa)

Energy = 65 GeV



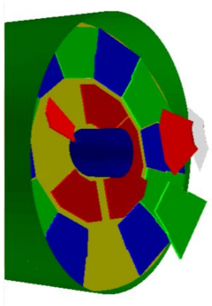
ATLAS VAE and GAN cf. Geant4 simulation [ATL-SOFT-PUB-2018-001.]

Detector Simulation

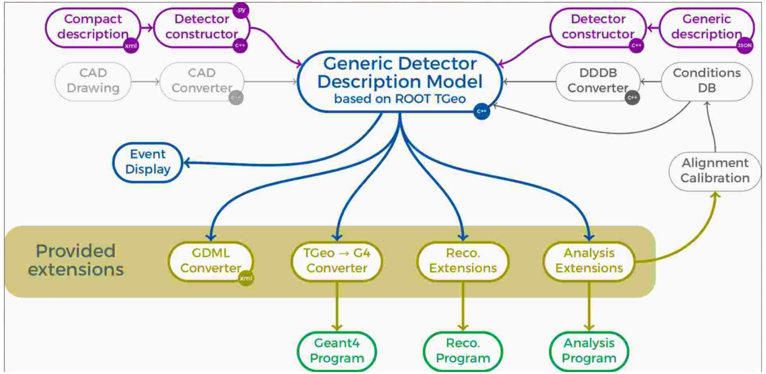


- Technical improvement programme helps (and helps *everyone*)
- GeantV R&D modernises code and introduces vectorisation
  - Speed-ups observed
  - Vectorisation introduces some gains
  - Code modernisation seems to help a lot
    - Reduce complexity and layers of object orientation
- Geant4 now have a new R&D working group that will take studies forward
- Some studies of running Geant4 on GPUs have begun
  - US Exascale Computing Project is funding this
  - Motivated by the next generation of US supercomputers that target exaflop
    - 90% of FLOP capacity in GPUs

DD4hep, DDG4, DDcond



- Detector description toolkit aimed at the full lifecycle of an experiment
  - Conceptualisation, Construction, Operations
- Geometry consisting of a ‘tree’ of detector elements, defined in a single place
  - Simulation (GDML)
  - Reconstruction geometry
  - Analysis extensions



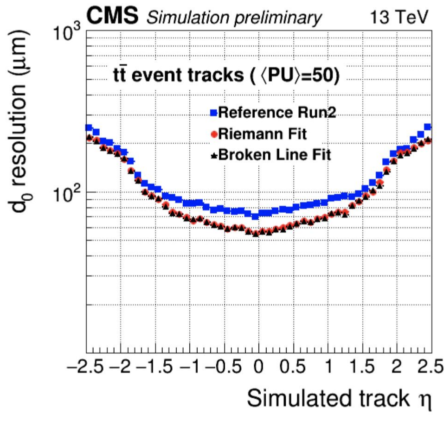
- DDG4
  - Hooks for user actions to generate detector response
- DDcond
  - Shifts detector elements from ideal position
  - Supports IoVs efficiently without locking (allows multi-threaded reconstruction across IoVs)
- Used by ILC, CLIC, CMS, LHCb, FCC, CALICE, SCTF, CEPC

This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.

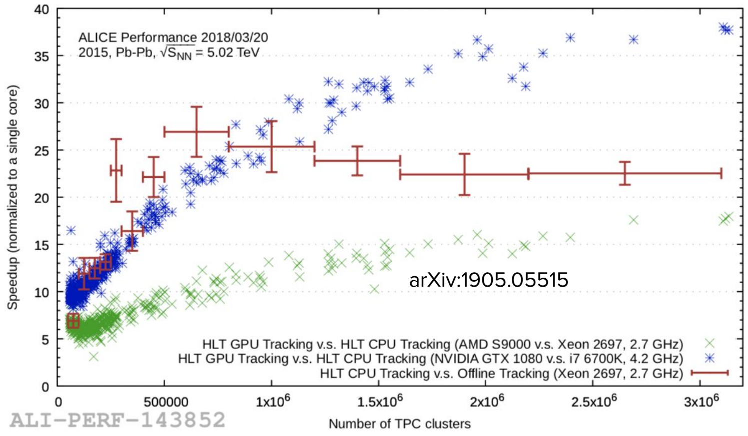
AIDA 2020

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 experimentation with rewriting code for GPUs
  - ALICE have ported a lot of reconstruction to GPUs and also improved the algorithms a lot
  - CMS Patatrack project has improved physics performance as well
    - Revisiting old code helps!
- Lessons learned keep data model simple, bulk data, be asynchonous, minimise data transfers



(a) d_0 resolution vs η

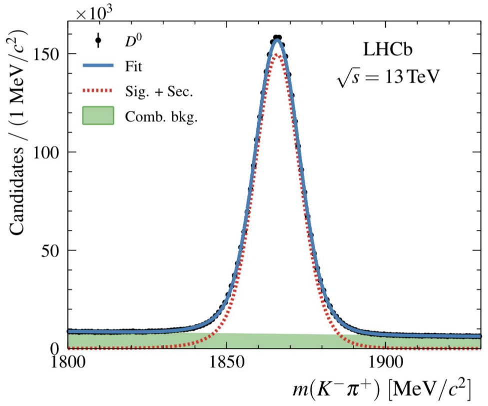


Reconstruction and Software Triggers

Persistence method	Average event size (kB)
Turbo	7
Selective persistence	16
Complete persistence	48
Raw event	69

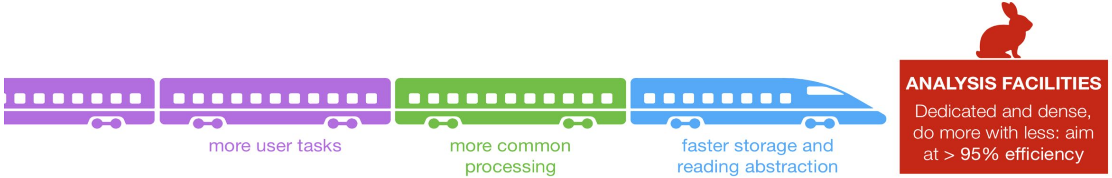
LHCb Run2 Turbo took 25% of events for only 10% of bandwidth

- 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 has been introduced in Run2 and will be dominant in Run3
- Whole ALICE data reduction scheme is based around keeping 'useful' parts of events (no more binary trigger)
- ATLAS and CMS have schemes under development for special handling of samples for which full raw data is unaffordable

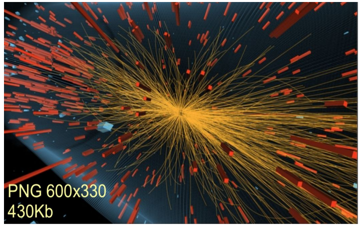


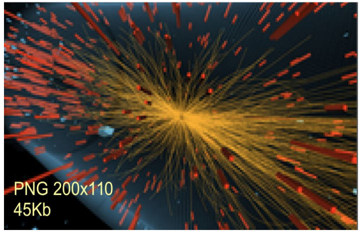
LHCb charm physics analysis using Turbo Stream (arXiv:1510.01707)

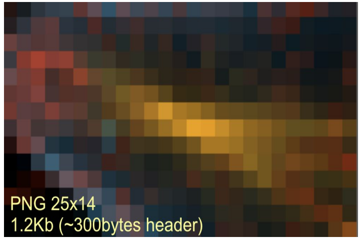
Analysis



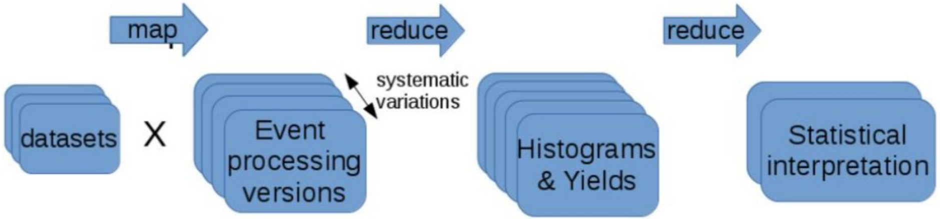
- Scaling for analysis level data also a huge challenge for all LHC experiments
- Efficient use of analysis data can come with combining many analyses as carriages in a train like model (pioneered by PHENIX then ALICE)
  - Also goes well with techniques like tape carousels (ATLAS scheme for rotating primary AOD data from tape systems into a disk buffer)
  - Interest in *analysis clusters*, specialised for analysis operations over the generic grid resources (WLCG/HSF pre-CHEP workshop 2-3 November)
- Reducing volume of data needed helps hugely
  - CMS ~1kB nanoAOD makes a vast difference to analysis efficiency and “papers per petabyte”
  - Smaller EDM is easier to make efficient
  - Requires analyst agreement on corrections, scale factors, etc.
    - However the alternative is perhaps that your analysis never gets done







Analysis



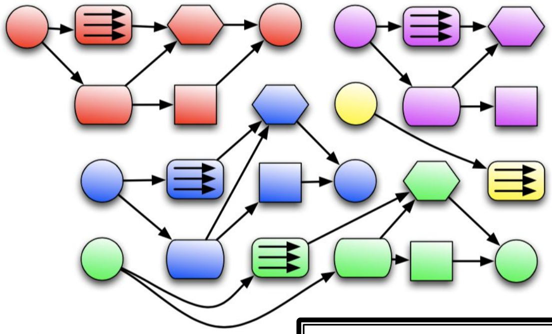
- Improve analysis ergonomics - how the user interacts with the system to express their analysis
  - Streamline common tasks
    - Handle all input datasets; Corrections and systematics
    - Compute per event and accumulate; Statistical interpretations
  - Declarative models, building on ROOT's RDataFrame
    - Say *what*, not *how* and let the backend optimise
    - E.g. split and merge, GPU execution
- Notebook like interfaces gain ground, as do containers - lots of high level Python
- Interest in data science tools and machine learning is significant for this community - inspiring new approaches (e.g. uproot, awkward array, Coffea, scikit-hep)
  - This is an ecosystem into which HEP can contribute
  - PyHEP Working Group coordinates activities in this area

Many analysis frameworks, multiple per experiment, not well generalised

```
# * Jet select/cleaning against loose leptons , jet pt > 25 , jet id
flow.DefaultConfig(jetPtCut=25,jetIdCut=0,jetPUIdCut=0)
flow.SubCollection("CleanJet","Jet",'''
 Jet_pt > jetPtCut &&
 Jet_jetId > jetIdCut &&
 Jet_puId > jetPUIdCut &&
 (Jet_LeptonIdx==-1 || Jet_LeptonDr > 0.3)
''')
```

A. Rizzi, NAIL prototype

Frameworks and Integration

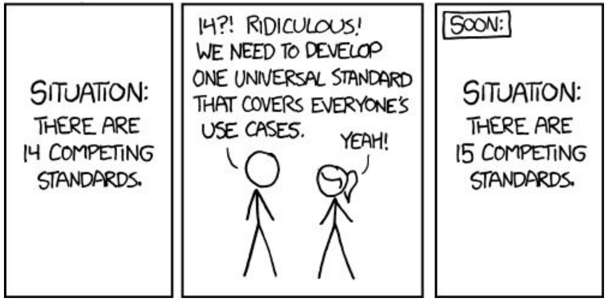


Cartoon of a single job, processing multiple events (colours) through different modules (shapes)

- Increasingly heterogeneous world requires advanced software support infrastructure
    - Software frameworks support use of different devices as well as insulate developers from many of the details of concurrency and threading models
        - Adapt to the new heterogeneous landscape
        - Latency hiding is critical to maintaining throughout
    - Framework development has traditionally been quite fragmented, but new experiments should offer a chance to increase convergence
        - Better to start off together than try to re-converge later (iLCSoft, LArSoft examples of success, albeit without concurrency; Gaudi for LHCb, ATLAS)
        - ALFA for ALICE and FAIR experiments
- New HSF Frameworks Working Group now established

EDM4hep

- Levels of software interoperability vary – from common file formats to deep framework level integration as “components”
- Sharing an EDM makes many tasks much easier
  - Common interface for algorithms
  - Serialised events easier to read when in standard format
- Very positive experience from Linear Collider community with LCIO
- We want to revisit this, in light of what has been learned in the last decade
  - EDM4hep project (GitHub, Indico)
  - LCIO + FCC EDM as basis
  - Use data model *generator* to write optimal data layouts for modern hardware
  - PODIO, Plain Old Data IO



Key4hep

- Build on EDM4hep to a complete *software stack* (easily 100s of packages)
- HSF Packaging Group has been looking at packaging issues for the community for several years
    - Problem seems naively simple, but quickly gets complex when addressing full set of use cases
        - N x M complexity (versions x build options / target platforms)
        - Reproducibility
            - Better to build deep and not depend on underlying OS
        - Relocatable binary products
            - Target: CVMFS, Containers, Local Installations
- Many possible solutions for building a *software stack* with a *build orchestrator*
    - FOSS, Scientific Community, HEP Specific, Experiment Specific

Spack Package Manager

Picking up momentum in HEP

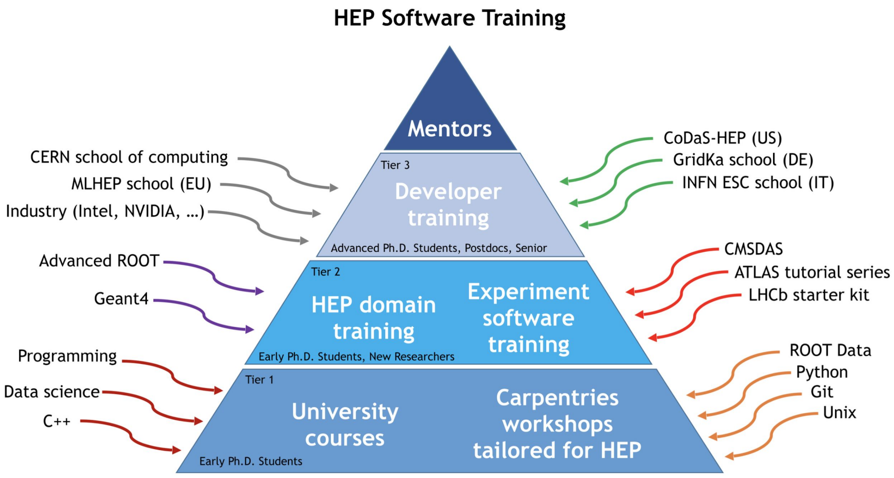
- used by FCC, Key4HEP, SupeNEMO
- FAIR moving to spack
- CMS has proof of concept
- ATLAS considering but things still to be understood
- LHCb and Belle II willing to follow

spack.io

Key4hep Prototype

- Using Spack build orchestrator from LLNL
  - Fulfils our use cases, strong scientific package support, good developer dialogue
- Building on work done to provide FCC and SuperNEMO stacks with Spack
- Prototype stack covering main use cases, from event generation to analysis
  - Spack first builds its own compiler (currently gcc9.2.0), for full self-consistency
  - Key top level packages:
    - Pythia, Geant4, DD4hep, Gaudi, ROOT
    - Use Spack's `packages.yaml` to set reproducible build options
  - Binary packages uploaded to build cache for deployment to CVMFS
  - Relocation via RPATH manipulation of libraries and binaries, configuration sed-iting
  - Runtime setup via standard *module system*

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
    - Links to 'Carpentries' being remade and to Astroparticle training events (ESCAPE project)
    - Software Carpentry training event at CERN, 27-29 November
- New areas of challenge
    - Concurrency, accelerators, data science, evolving C++ requirements
- Careers area for HEP software experts is an area of great concern
    - Need a functioning career path that retains skills and rewards passing them on
    - Recognition that software is a key part of HEP now

Summary and Outlook

- There is a continuing active R&D programme for software in HEP
  - Much activity in all experiments and in support groups, such as EP-SFT
  - New support from initiatives like IRIS_HEP in the US
- Increasingly well linked to end-to-end systems that run our production at scale
- HSF plays a prominent role in developing cross experiment communication in key areas
  - Working groups are open to all experiments and nominations for convenors to take a leadership role are welcome
- Links between the LHC experiments and future growing stronger

We welcome future collaboration and engagement with CEPC colleagues