Software for particle physics UK feedback to ECFA

26 September 2024

Davide Costanzo and Sam Harper

Multi-experiment software

The UK Particle Physics Technology Advisory Panel (PPTAP) (link) 2021

Closely follows similar recommendations from the CERN review of particle physics

[34] Likewise within software and computing the UK has significant leadership in a significant number of important areas, including in exploitation of computing accelerators, exploitation of low power compute units, computing operations, enabling software and computing, reconstruction algorithms, software framework development, development of cross- experiment development tools, use of HPC and development of collision simulation/generation programmes.

Layer	Domain	Experiment 1	Experiment 2	Experiment 3
6	Physicists	Analysis code	...	...
5	Experiment Physicists programmer and software engineers	Analysis framework. Simulation, Reconstruction, Calibration Code	...	...
4	Experiment Software Engineers	Software Frameworks	...	...
3	Common Software HSF / SWIFT-HEP	Common software components (Data management, Generators, Geant4, Accelerator integration)		
2	GridPP / WLCG	Middleware infrastructure for Distributed Computing		
1	GridPP / WLCG	Physical Hardware		

Moving Software down the stack

Driving factors in the evolution of software

Online/Offline convergence

- Same code, tools and technology for trigger (low latency) and offline (high latency)

Computing architectures evolution

- Designed for parallel computing, eg GPUs. Chips innovation driven by AI
- Software needs to be portable (e.g arXiv 2409.09228, 2306.15869).
- Tools like alpaka, SYCL, kokkos, oneAPI currently in development
- Will we use an open standard for GPU code?

HEP code is mostly Array of Structure (AoS) and needs to move to Structure of arrays (SoA)

- In progress, lots of experimental software evolving this way
- We should not think in terms of "porting to GPU" rather "re-engineering"

HEP-CCE: arXiv 2306.15869

	CUDA	HIP	OpenMP Offload	Kokkos	dpc++ / SYCL	alpaka	std::par
NVidia GPU					codeplay and intel/llvm		nvc++
AMD GPU				feature complete for select GPUs	via openSYCL and intel/llvm	hip 4.0.1 / clang	
Intel GPU		CHIP-SPV early prototype		native and via OpenMP target offload		prototype	oneAPI::dpl
multicore CPU							g++ & tbb
FPGA						via SYCL	

Portability example from CMS

CMS DP -2024/026

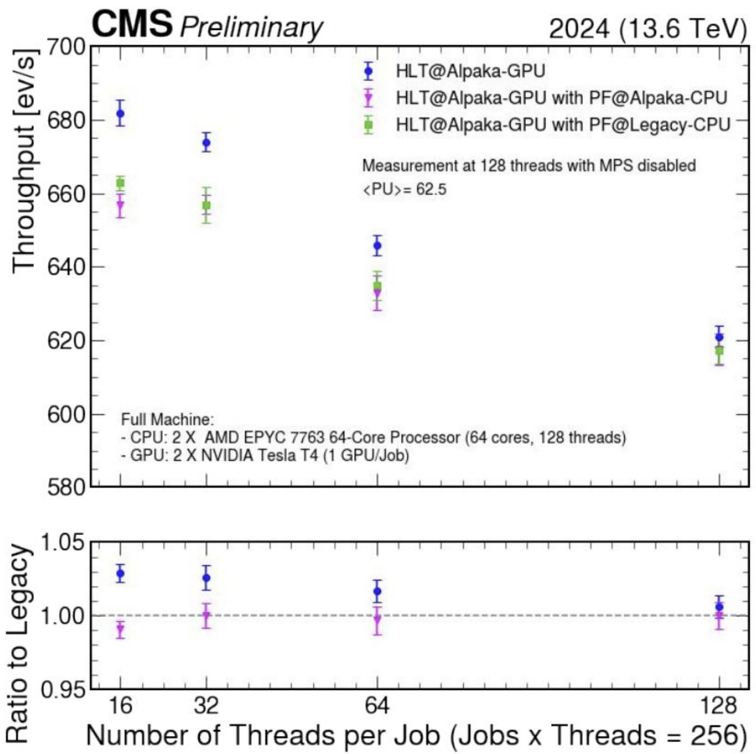
Like LHCb, ALICE, CMS runs HLT on GPUs

- initially a GPU coded and CPU coded algo

Now uses abstraction library alpaka

- one source file compiled run to CPU and GPU object code
- performance ~equivalent to native

Currently in production at P5



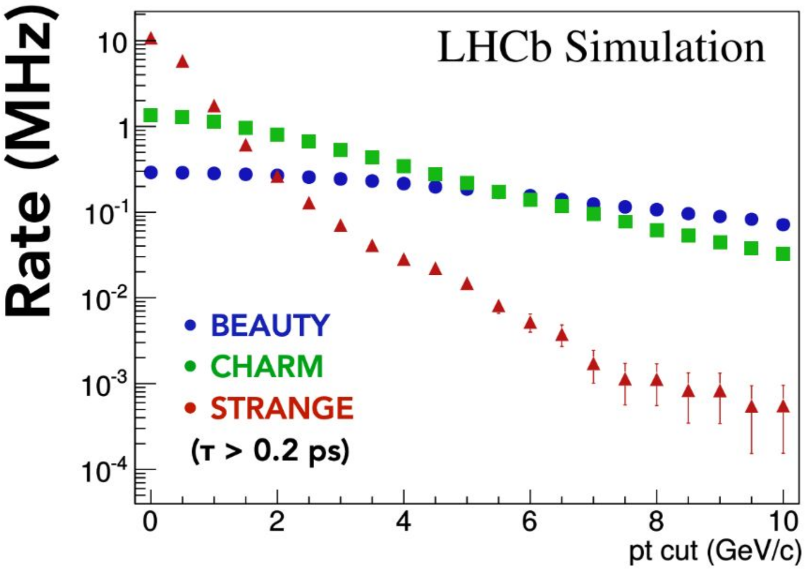
Reconstructing events

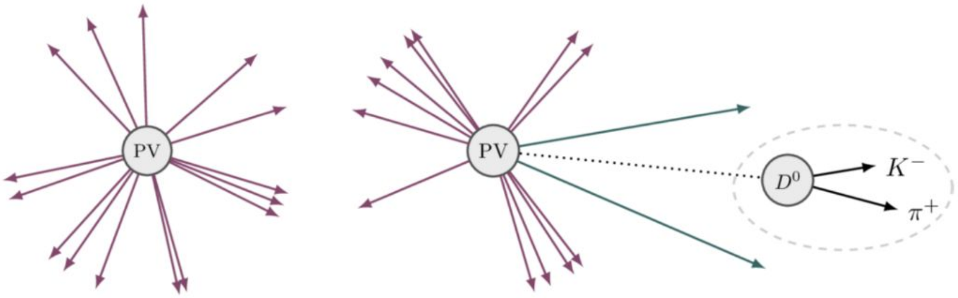
What reconstruction code will we use in 10+ years time?

- C++ based? Which portability tools?
- Skills and expertise of people who know how to reconstruct particle collisions
- Role of AI and ML driven algorithms (see next talk)

Example from LHCb upgrade

- Requires full event reconstruction at 30 MHz

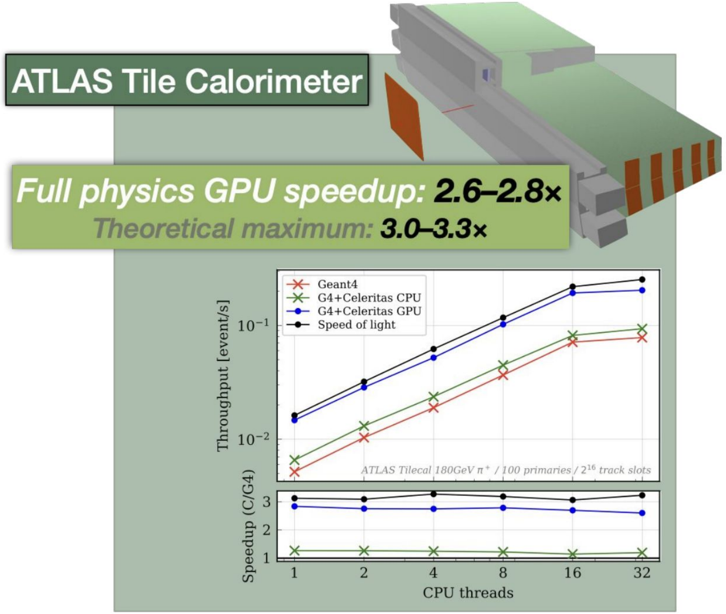


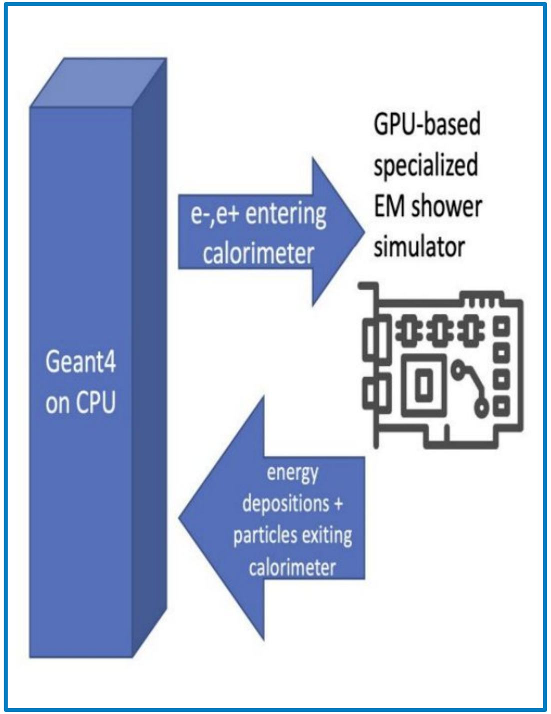


Simulation and Monte Carlo generation

Simulation requirements for high precision measurements

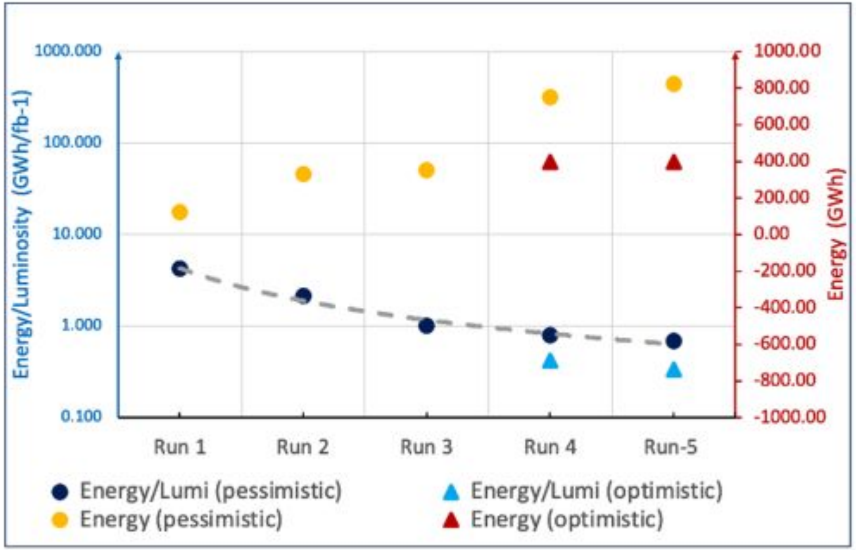
- Event generators evolving with more "N"s NNLO, etc. Software engineering plays a central role
- Role of non-perturbative methods (e.g. Lattice models)
- Simulation of particle interactions is very power intensive. GPU prototypes under development for major LHC experiments.
- Will this change for Run-4?





Power usage and the role of the software

Energy needs on WLCG for ATLAS and CMS (CHEP 2023)



Electricity may be carbon neutral from 2040,
But it will still cost money!

Embrace low power computing

- Evolution driven by AI (chatGPT4 required ~50 GWh to train)
- Low power CPUs available, eg ARM and RiscV
- Lots of potential in using GPUs or FPGA
- Reduce operating costs, environmental impact and cooling power needed on site

We should start quoting software and computing performance in events/kWh

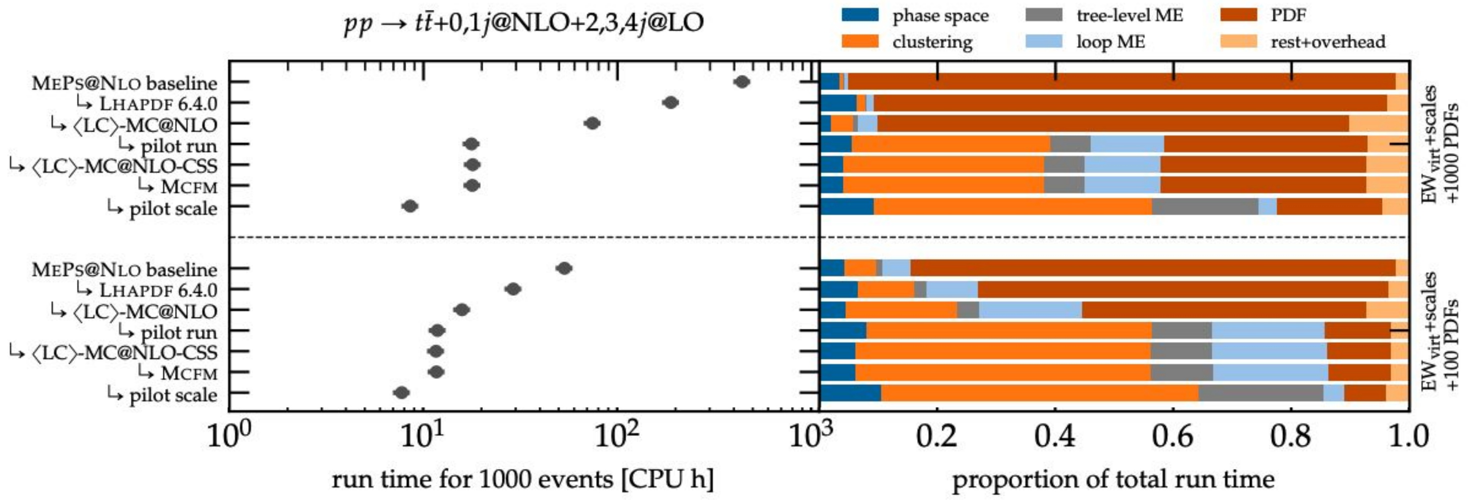
- Or event events/£, which requires an understanding of the total cost of ownership
- Software should be taken into account in the experiment design

Software improvements. An example from event generators

- Event generators are a key ingredient at the LHC
  - Higher precision generators needed for high precision physics
  - Not just the LHC, also neutrino physics (e.g. Genie)

UCL, Glasgow, Durham
Chris Gutschow et al,
EPJ C82 (2022) 12

- Same physics results with less computing
  - Example from Sherpa (possible thanks to SWIFT-HEP funding)
  - Improve loop over PDF
  - Pilot run
  - …



Analysis, storage, other comments

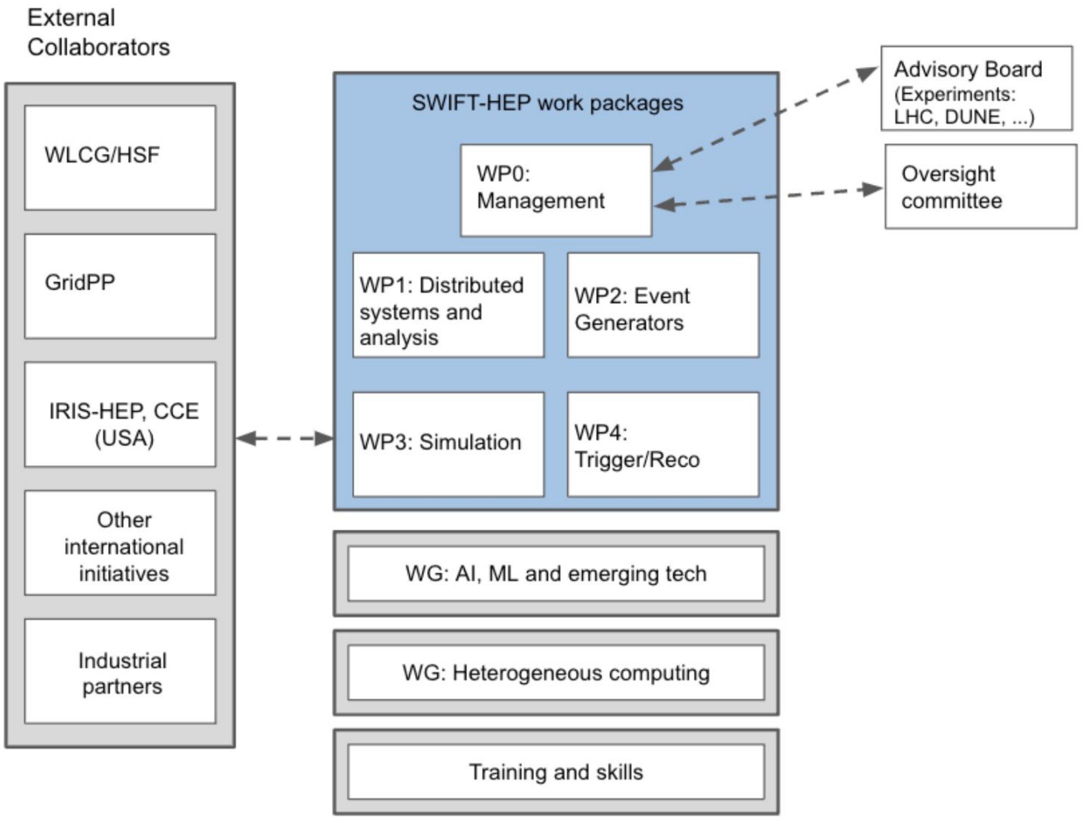
Analysis and user interaction with the software ecosystem

- Particle physics used to be leaders on this
- How do we use modern data science tools? What is the role of AI?
- Needs to remain very attractive for the new generation
- How do we improve turnaround time?
- More in Tim’s talk

Storage is a large fraction of the total cost (See Alastair’s talk)

- We store ExaBytes of data
- Quality of service is crucial for users
- Strong constraint on ATLAS&CMS trigger rates
- Better compression algorithms?
  - lossless, lossy , ML assisted
- Better data placement?
  - re-engineer data format objects to be more efficiently stored

Cross experiment software development in UK (SWIFT-HEP)



Crucial activity to deliver the computing needed in the 2030s

- £3M funding for 2020-25
- 20 institutes signed the proposal
- 8 institutes received direct funding
- Part fund people at Universities and national Labs to match with other projects

5 FTE funded by STFC from Sep 2020 to 2025 + 2 FTE funded by the ExCALIBUR programme (software for exascale facilities)

Larger proposal for up to 20 FTE (2025-29)

- Statement of intent not supported by STFC
- In discussion for a continuation at the current level for 2025-27

Conclusions

Software is an important technology in the evolution of future HEP experiments

- "Co-design" experiment to take into account what it takes to process the data output
- And to consider how detectors are simulated
- We need to work with DRD and the rest of the community
- Software and computing **will** be a major bottleneck without adequate personpower and development

Technology evolution is ever more important

- We are at a point in history where software developments have a great impact on society
- Great opportunities are available (see next talk). But also lots of competition for talent!
- We (particle physics) are an excellent training ground for future data scientists
- Will we code in C++? How do we train the new generation?

We are preparing a proposal to a UKRI call to write a roadmap for HEP software in the UK

- As part of CoSeC (Computational Science Centre for Research Communities)
- Idea: to have a particle physics "collaborative computational project" (CCP) as part of CoSeC portfolio
- I'll chat with people about this in the next few days