Computing and software beyond HL-LHC

James Catmore, UiO

Introduction

- The next big accelerator won't be operational until the late 2030s at the earliest
- Computational hardware and software evolve at a fast pace (although slowing…?)
  - The best we can do is make an educated guess about what things will look like by then unless we have a crystal ball or a magic pool
  - Run 5 of HL-LHC will probably be computationally more demanding than FCC &c
  - All of the points in the next slides also apply to Run 5 of the HL-LHC

What can we expect by the 2030s?

More use of common software

- Currently each experiment has its own framework, reconstruction/analysis software, event data model, I/O &c
    - Some components shared via the LCG installations which are managed by CERN's SFT group: simulation (Geant4), event generators, ROOT &c
- Recently SFT launched the Key4HEP project which is a toolkit for new experiments (either beyond LHC or smaller experiments), which will include a common framework (Gaudi), common tracking software (ACTS), common event data model, build scripts &c &c
- We can expect that future experiments will not have to build all of their own software from scratch but will be able to import and then adapt common tools
- Success depends on the HEP Software Foundation and SFT coordinating this common software so the community needs to ensure they have adequate support
- Can't see any current trends away from C++ for heavy duty core software and python as an interface layer
    - Usability of C++ improves significantly with each major update

What can we expect by the 2030s?

More use of massive parallelization

- At the moment CPUs and GPUs are separate hardware types with data being shipped between them
  - CPUs: small number of high-power cores; GPUs: large number of low-power cores
  - Many groups are working on migrating algorithms/workflow to GPUs where this is practicable
  - Most grid sites don't yet have GPUs; most HPC centres are primarily constructed from them
- Will this separation continue? Probably not → "system on a chip"
  - Already we see integration between CPU and GPU in Apple silicon and architectures like Nvidia Grace Hopper
  - By the 2030s it's reasonable to assume that compute hardware will contain elements of both and software will need to make use of large numbers of threads to get the full performance
- Consequently the ongoing work to migrate as many algorithms/workflows as possible to GPUs is very important
  - The language is largely unimportant - what matters are the algorithmic changes
  - Often the process of migrating algorithms to many threads yields improvements for CPU running as well

What can we expect by the 2030s?

More use of ML-based components

- ML-based algorithms will almost certainly be used more widely than now
  - Already close to ubiquitous in data analyses where the backgrounds are non-trivial; ML-based clustering has also been in use for years
  - Now being used for higher-level tasks such as jet classification and flavour tagging (graph NNs, transformer models), calorimeter simulation (generative methods) → often see better computing *and* physics performance
  - Ongoing work to train graph NNs to do pattern recognition for inner tracking is more challenging
  - Easy to envisage increasing use of generative techniques in simulation beyond calorimetry
    - Could similar techniques be applied to physics event generation as well?
  - Quite possible that most steps of the production workflow will be ML-based by then
- Even for human-written procedural software, very likely that ML-based components will be embedded in commonly used toolkits and run behind the scenes

Discussion - where might we contribute?

- Too far in the future to say anything concrete
- Possible activities
  - Join relevant HSF workshops
  - Lobby to ensure that HSF and SFT are supported so that common software projects are able to flourish
  - Contribute to developing common training material
- Key to all of this is ensuring that our current and future undergraduates and master students are given a solid basis of knowledge around scientific computing, data analysis and machine learning