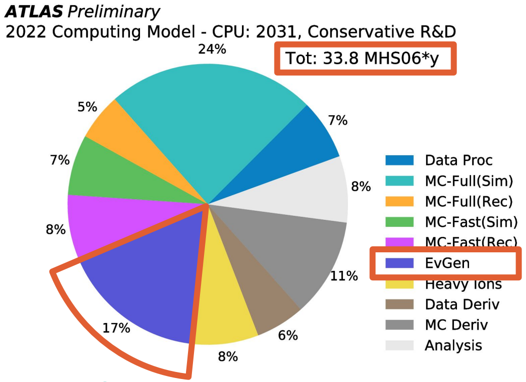
Capitalise on existing HEP software expertise to upgrade event generation

Event generators' and N(n)LO codes' acceleration Workshop 13/11/2023

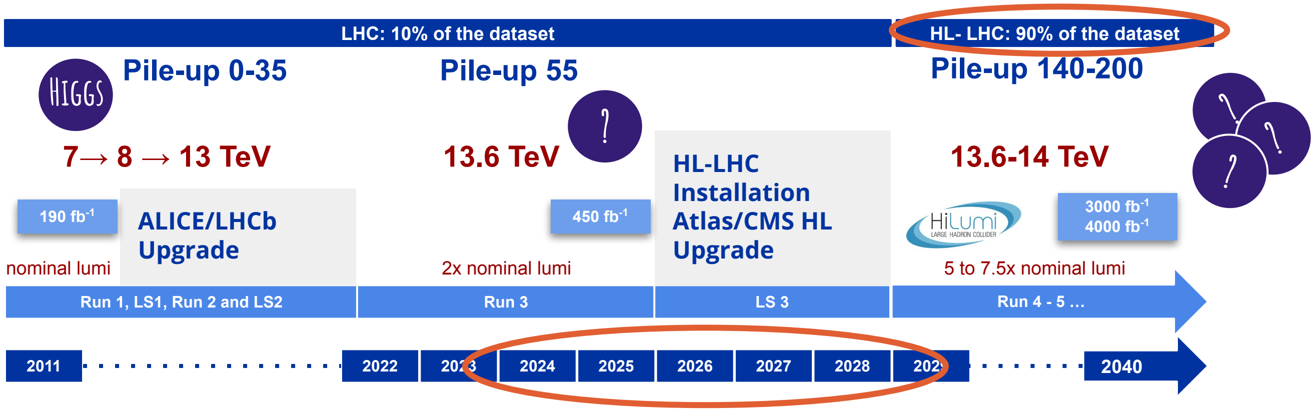
Danilo Piparo CERN for the EP-SFT group

Full Exploitation and “Flat Budget” for Computing Resources

- Today more than 1 billion events per week are generated by ATLAS and CMS
- The European Strategy for Particle Physics **recommends the full exploitation of the physics potential of the HL-LHC**
  - **Honouring that recommendation also implies generating ~1 trillion of events per year** ([1], [2])
- Not only a plan for HL-LHC, but **also a Run 3 matter**: e.g. legacy processing during LS3 or particular analyses (see pdf independent W mass measurement)
- We cannot realistically plan on **CPU capacity at disposal of experiments incremented much more than 10-15% per year to address future MC event generation needs…**

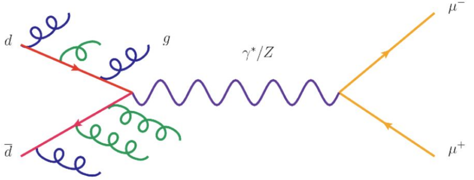


ATLAS public results Twiki



The Opportunity: Event Generator Software must Improve

- **A challenge is posed to several aspects of N(N)LO event generator software, e.g.**
  - Runtime performance
  - Interfaces and integration for experiment frameworks
  - Flexibility in the deployment (easy access, multiple platforms)



S. Qian talk

- **Meeting such a challenge requires sizeable expertise**
- **All this while the hardware landscape evolves**: x86_64 / ARM / RISC CPUs, and accelerators provided by different vendors (see S. Hageboeck's talk)
- Example: High Performance Computing centres (HPCs) - supported by multibillion national and supranational investments (see HPC section of this workshop)
  - They are there to stay, and HEP needs to continue to use them
  - Most of the computing capacity at HPCs is expressed by accelerators
  - Accelerators are not only provided by HPCs: they are already present in experiments' (WLCG, trigger farms) infrastructure

epi European Processor Initiative

Full exploitation of HL-LHC physics potential poses a challenge to several aspects of event generators software

Software Expertise and Monte Carlo Event Generators 1/2

- **MC event generators: a real asset for our community, built through decades of hard work**
  - At the heart of the understanding of the data acquired at LHC
  - A concentrate of theoretical knowledge, testing and measurements
- **A limited number of specialists are able to develop MC event generators**
- Typical profile: phenomenologists, theorists, ~~**software engineers, and computer scientists**~~
  - They have very good programming skills
  - Often do not have the desired funding nor time to be at the bleeding edge of all software process best practices
  - Collaboration of event generators authors and software experts has been proven to work (see several talks at this workshop)
- **Expertise requirements do not seem to get any looser**, e.g.:
  - Need to take advantage of new (extreme) hardware architectures
  - Need to evolve the functionality of the event generators to match available computing and software technologies, as well as experiments practices (e.g. read-only gridpacks distributed via CVMFS)

Rarely know how to develop evt generators

Software Expertise and Monte Carlo Event Generators 2/2

- For a theorist, it is not always possible (funding, time, reward in terms of positions or grants) to heavily invest in software development best practices
  - **The interests of theory and needs of HEP experimental community seem to diverge on the point of software engineering practices**
- **Not everybody is getting what they need: is a path needed to improve this situation?**
  - Brillant efforts of a limited number of individuals (see talks about generation on GPUs)
  - Is this sufficient or just a piece of the puzzle which can be further consolidated, e.g. across CERN and other entities?

Is there a risk that the effort expended on achieving precise MC event generation is wasted unless it is accompanied by software engineering best practices?

The Software Development Unit of CERN's Experimental Physics Department

- The SFT (Software) group of the Experimental Physics (EP) Department at CERN
- SFT provides leading contributions to major community software packages such as CVMFS, FCC SW, Geant4, Gaudi, DD4HEP and ROOT
- **SFT has been supporting experiments long before the start of LHC: consolidated links and communication channels are established** (e.g. core software and simulation teams)
  - Long standing collaborations with major laboratories (FNAL, GSI) as well as universities
- Not only development/support/management of production software
  - **Software packaging and distribution** of community software packages through LCG Releases (collections of ~500 common sw packages compiled coherently available through cvmfs)
  - **Robust R&D programme**, e.g. simulation on GPUs (AdePT), data processing/analysis with SYCL, (ML) Fast Simulation techniques…

SFT Software Expertise

A wide range of software expertise available in SFT, among which:

- Software design and development for parallel hardware architectures (e.g. multicore, heterogeneous)
- **Profiling, runtime performance improvements** and debugging
- **Packaging and distributing** production software (including MC event generators)
- **GPU programming and performance portability**
- Expression of **data and task parallelism for HEP data processing and analysis problems**
- Compiler development
- **Machine learning**
- **Physics validation** of simulation and physics processes models
- **Integration of production software** in large experiments' software stacks

Can SFT help to upgrade HEP's event generators suite? If yes, how?

A Platform for Software Related Work on Event Generators?

Based on the expertise available in SFT, some example of *potential ideas* for activities and resulting benefits can be easily compiled

Extend the current validation workflows for G4/Fast simulation to event generators

- Faster feedback loop with generators developers, *before* their product reaches experiments

Package and distribute a more complete suite of generators on CVMFS as part of LCG Releases

- Reach even wider audience, provide a reference platform, test integration in a package suite, profit from existing continuous integration infrastructure for catching bugs, track runtime performance, stress-test generators.

Apply knowledge and know-how gained with GPU usage in production to generators

- Create a critical mass of GPU expertise in the RCS sector, further accelerate generators upgrade, including support for different accelerator types

Combine generators with distributed analysis tools

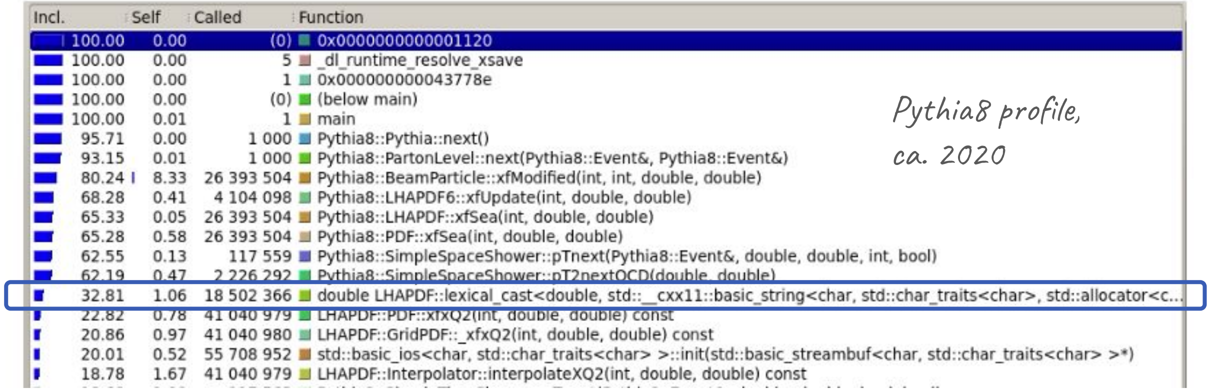
- Provide a more modern approach wrt large scale job submission to deliver datasets for generator-level studies

Ultimately, this could be an opportunity to build up a critical mass to apply software knowledge (notably about GPU programming and parallelism) to generators codes

Success Stories from the Past

SFT is not new to Monte Carlo event generators

- **GENerator SERvice GENSER** (now re-scoped and absorbed in LCG Releases provision) From the original mandate *“[...]collaborate with the authors of Monte Carlo generators and with the LHC experiments in order to prepare validated code for both the theoretical and experimental communities at the LHC”*
- D. Konstantinov, **20%-50% speedup of Pythia8** (depending on the workflow) for LHCb)
- D. P. in collaboration with G. Chahal and **Sherpa authors: 15% memory saving for DY + 01234 jets** (presentation in CMS , Sherpa Repo)



```
Incl.     Self    Called       Function
100.00    0.00          (0)    0x0000000000001120
100.00    0.00            5    _dl_runtime_resolve_xsave
100.00    0.00            1    0x000000000043778e
100.00    0.00          (0)    (below main)
100.00    0.01            1    main
 95.71    0.00        1 000    Pythia8::Pythia::next()
 93.15    0.01        1 000    Pythia8::PartonLevel::next(Pythia8::Event&, Pythia8::Event&)
 80.24    8.33   26 393 504    Pythia8::BeamParticle::xfModified(int, int, double, double)
 68.28    0.41    4 104 098    Pythia8::LHAPDF6::xfUpdate(int, double, double)
 65.33    0.05   26 393 504    Pythia8::LHAPDF::xfSea(int, double, double)
 65.28    0.58   26 393 504    Pythia8::PDF::xfSea(int, double, double)
 62.55    0.13      117 559    Pythia8::SimpleSpaceShower::pTnext(Pythia8::Event&, double, double, int, bool)
 62.19    0.47    2 226 292    Pythia8::SimpleSpaceShower::pT2nextQCD(double, double)
 32.81    1.06   18 502 366    double LHAPDF::lexical_cast<double, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<c...
 22.82    0.78   41 040 979    LHAPDF::PDF::xfxQ2(int, double, double) const
 20.86    0.97   41 040 980    LHAPDF::GridPDF::_xfxQ2(int, double, double) const
 20.01    0.52   55 708 952    std::basic_ios<char, std::char_traits<char> >::init(std::basic_streambuf<char, std::char_traits<char> >*)
 18.78    1.67   41 040 979    LHAPDF::Interpolator::interpolateXQ2(int, double, double) const
```

Pythia8 profile,
ca. 2020

Executive Summary

- The full exploitation of the physics potential of the (HL-)LHC poses **a challenge to Monte Carlo generators' software: not only runtime performance, but engineering, adaptation to experiments frameworks and new hardware platforms (e.g. GPUs)**
  - HEP will need trillions of Monte Carlo events per year in just a few years time
  - **This challenge should not scare us, but become a strong motivational trigger!**
- **Difficult to find all the necessary software expertise to face such a challenge in the Monte Carlo generators community alone**
- **Consolidation of the available software expertise is key, at CERN and elsewhere**
  - **We need to create a critical mass of developers** to apply sw related knowledge to the event generation codes, most notably about GPU and parallel programming

Given its expertise in critical software domains and links to experiments, the CERN EP-SFT group might be well positioned to act as a focal point for re-engineering the HEP Monte Carlo event generators suite