SWIFT-HEP Generators area & projects

Andy Buckley, Marek Schoenherr

SWIFT-HEP kick-off, 14 Jan 2021

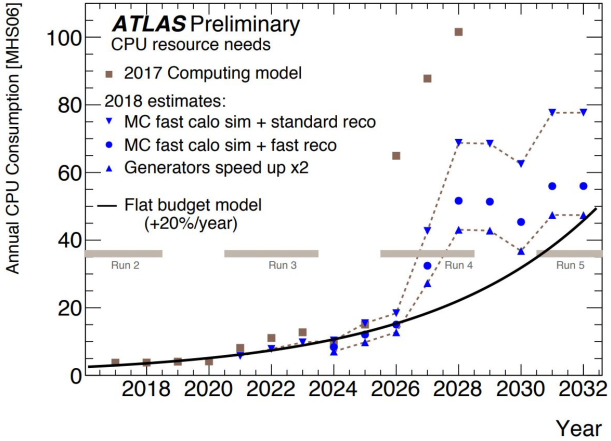
MCnet University of Glasgow Durham University THE ROYAL SOCIETY

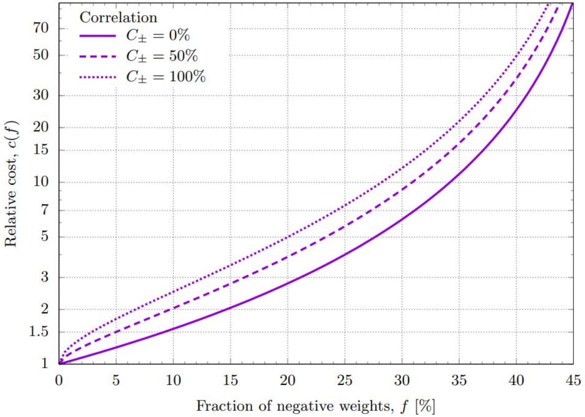
Big issues in MC event generation

Challenge: generation of large, high-precision SM samples alone will saturate HL-LHC compute budget!

Main issues are physical, technical, and logistical:

- low efficiency in ME phase-space sampling
  NNLO ~ 10x NLO ~ 100x LO!
- high rates of negative event weights from NLO ME/shower matching kill statistical power
- legacy code designs mismatch modern architectures and new usage patterns (e.g. O(100) weights/event)
- low-scale physics can also be problematic when scaled up: heavy quarks, exclusive hadron production & decay channels, colour reconnection algorithms
- evgen duplication between experiments





Ways forward

Better ME sampling and negative-weight elimination

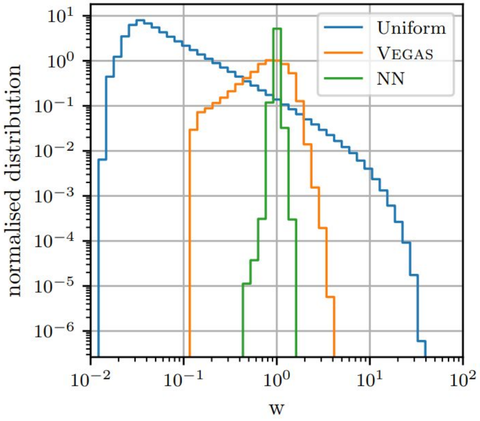
- Machine learning for ME-sampling proposal functions. Needs theory depth, dominant cases remain hard so far
- New matching schemes for -ve weights, and positive-weight resampling (+ this) ⇒ will be investigated by experiments

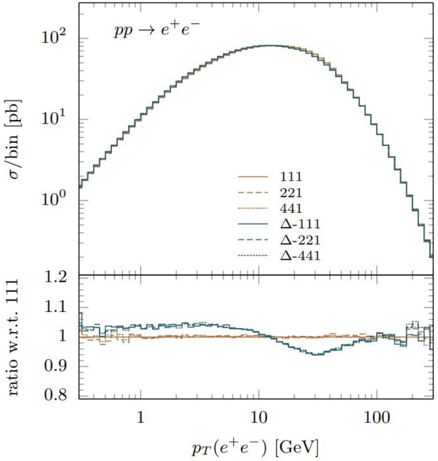
Computational efficiency and new architectures

- Efficiency not highest priority for theory. Problems amenable to sw eng / exp physicist effort: profiling, caching, vectorising
- Similar for efforts to take advantage of HPC facilities, hardware accelerators, etc.: US HPC efforts, HDF5 formats

Inefficient low-scale physics modelling

- Efficient hadronisation biasing (decays not a bottleneck) ✓
- Accelerators for expensive colour-reconnection algorithms? Latest physics models are too expensive for LHCb





Activity under ECHEP

Details in Tim Martin's talk

Profiling Sherpa in experiment-like setups:

- preliminary findings:
  ⟹ primary cpu sinks are PDF access and transcendental functions
- identified as targets on short-time frame
- potentially large improvements due to heavily multiweighted samples. Gain depends on use-case, e.g. ME/shower or integration/generation: behavioural changes?

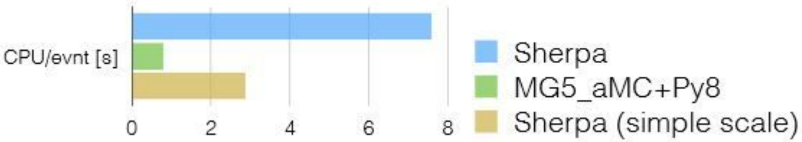
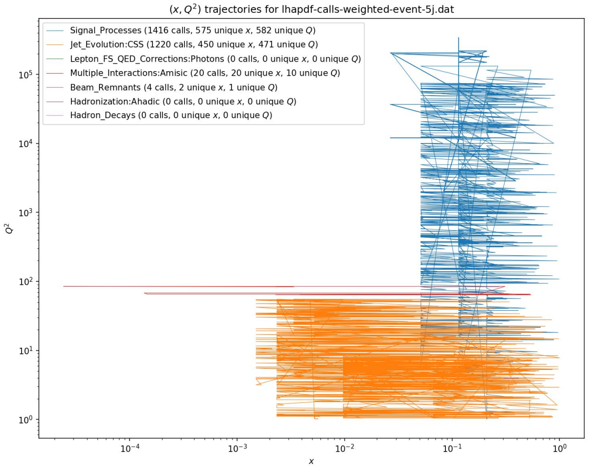


Figure 4: CPU per event $W \to e\nu$+0-2jets@NLO (using pre-integration grids).

Leveraging accelerators for colour reconnection:

- investigated SYCL accelerator API for Pythia8's colour reconnection string-annealing model
- technical and algorithmic lessons learned
- hadronisation and hadron decay biasing → SWIFT-HEP



ECHEP/SWIFT-HEP event generator programme

12+6 RA person months, 3+4 RSE person months at UCL & Warwick. Deliverables:

Hard-scattering performance study and technical optimisation

1) Detailed reports (first step soon) on CPU-performance profiling of high-precision Monte Carlo event generation to identify bottlenecks for cost-driving setups used by experiments
2) Optimisation of PDF evaluation through major contributions to LHAPDF, Sherpa, Herwig, MG5, etc. following (1)
3) Other generator technical performance refinements following (1)

Development of efficient hadronisation models

4) Prototype of a biased hadronisation method for Pythia8
5) Develop a vectorised colour-reconnection module for Pythia 8
6) EvtGen modernisation… toward common hadron decay pkg? [not critical path, but aligned]

Outlook

A big step to have MC software funding, albeit small-scale in this round

- Can only tackle a subset of the big issues, but it’s a start
- SWIFT-HEP isn’t just the funded posts — all MC performance activity (e.g. touching MG5, which isn’t a “UK based” generator) encouraged to join the community Mattermost, etc.

Real-world problems — structural incentives are often misaligned

- MC generators are developed under HEP theory grants: funds and career progression depend on “theory papers”, not sw performance.
  Less institutional room & reward than in experiment for technical work
- Ensuring career paths & attractive prospects for fractional FTEs
- Organising how (remote?) fractional FTEs will embed in MC gen groups
- Experiment service credit: MC community work needs expt recognition

Exciting times! Let’s do some nice, impactful work

Backup

Broader programme starter possibilities

JIT optimisation

- event generators are universal codes at prototype stage, featuring plenty of recurring branch points of constant evaluation outcome in a given run (eg. check that hadron collider setup and both initial states need PDFs, ME has n legs, etc)
- in many cases (probably) not caught/anticipated by current CPUs (esp. In light of spectre/meltdown fixes)

Large-scale refactorings of generator codes

- different parts of event generation lend themselves to different types of acceleration
  ME -- constant program flow, large expressions (multithreading, vectorisation, GPU)
  rest -- variable program flow (Markov Chains, etc), best parallelised trivially
- needs software engineering expertise, probably also target machine specific opts

Common decay package (including systematics)

- hadron decays mostly non-controversial, provide common hadron decay package