Further Sherpa and LHAPDF improvements

Enrico Bothmann, Andy Buckley, Christian Gütschow,
Stefan Höche, Max Knobbe, Marek Schönherr

Slides originally by Christian Gütschow, slightly adapted

HSF Generator WG Meeting #19

7 April 2022

Recap (copy-pasted from Matthew Gignac's talk last time)

Configuration	SHERPA 2.2.1	SHERPA 2.2.11
Generator version	SHERPA 2.2.1	SHERPA 2.2.11
PDF set	NNPDF3.0NNLO	NNPDF3.0NNLO
EW input scheme	Effective	$\sin^2\theta_{\text{eff}}$
QCD accuracy	0–2j@NLO+3,4j@LO	0–2j@NLO+3,4,5j@LO
NLO EW_{virt} corrections	No	Yes
Subtraction scheme	Default	Modified Catani–Seymour
Unordered histories allowed	Yes	No
Scale for $\mathbb{H}$-events	STRICT_METS	H'_{T}
Gluon colour/spin exact matching	Yes	No
Core process for K-factor	$2 \to 4$	$2 \to 2$
Phase-space strategy	Sliced in $\max(H_{\text{T}}, p_{\text{T}}^{V})$	Analytic enhancement

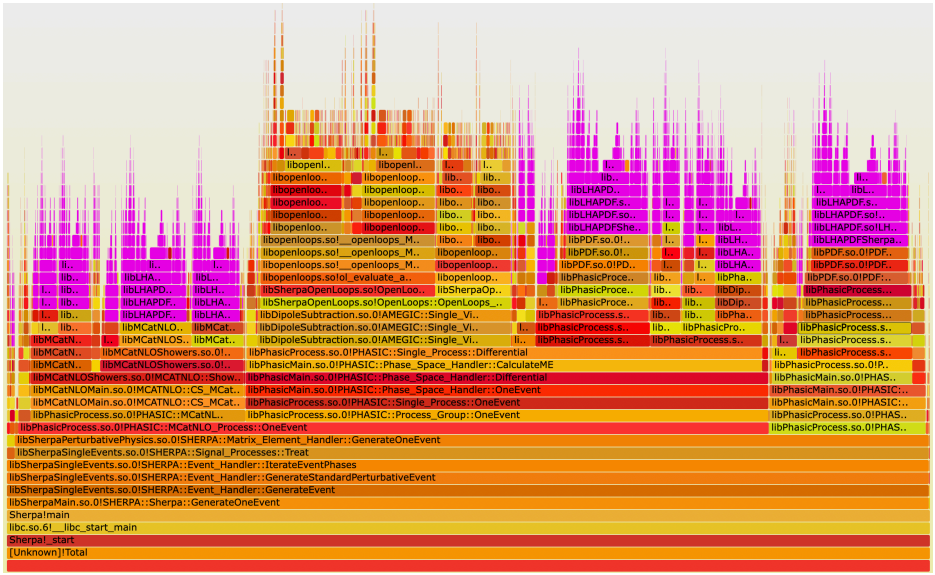
Some of these add additional CPU time, some of them reduce it → how do the setups compare when everything is simultaneously deployed?

➜ "Achieved nearly a factor of two reduction in the CPU resources compared to a legacy setup used by ATLAS for more than 5 years, while simultaneously improving the modelling & accuracy"

Recap

- first ECHEP generator profilings by Tim Martin showed LHAPDF uses a lot of the per-event CPU time (60–70%) (Tim's talk)
  - maybe not completely surprising: multiweights are a releatively recent feature and code originally not designed with hundreds of variations in mind (EB, M. Schönherr, S. Schumann [arXiv:1606.08753])

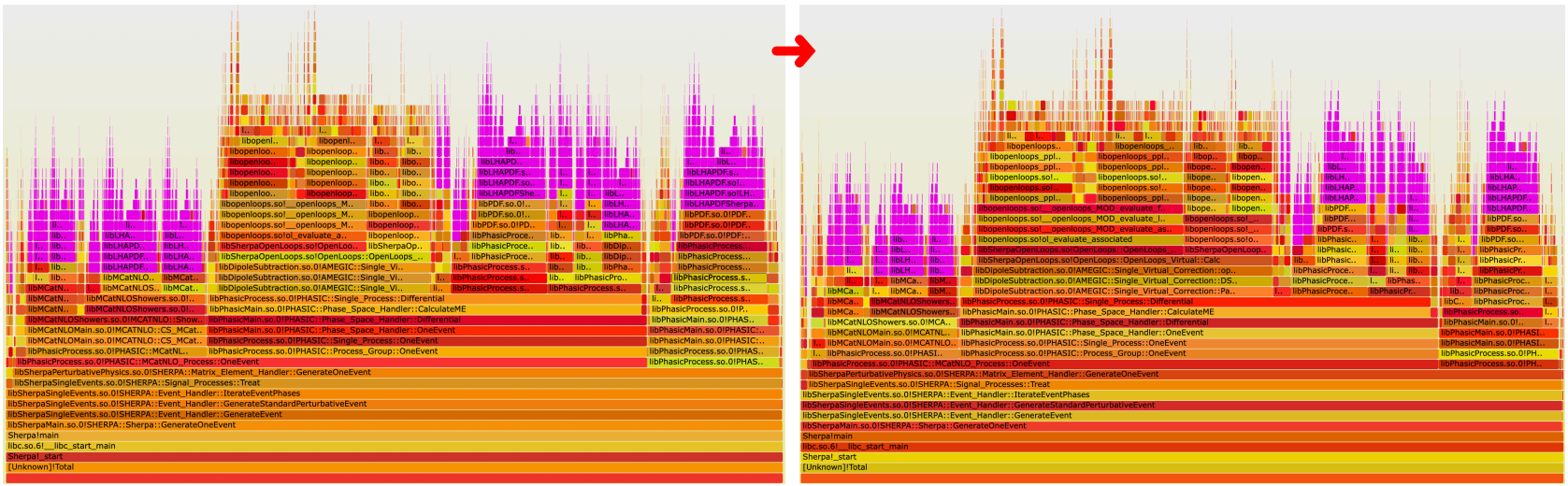
- RHS: Sherpa+OpenLoops $\ell\ell$+0,1,2j@NLO+3,4,5j@LO with PDF calls (60%) highlighted in magenta
- using LHAPDF 6.2.3



- explore two approaches in parallel:
  - make LHAPDF faster (lots of time spent interpolating and evaluating transcendental functions)
  - rework LHAPDF call strategy

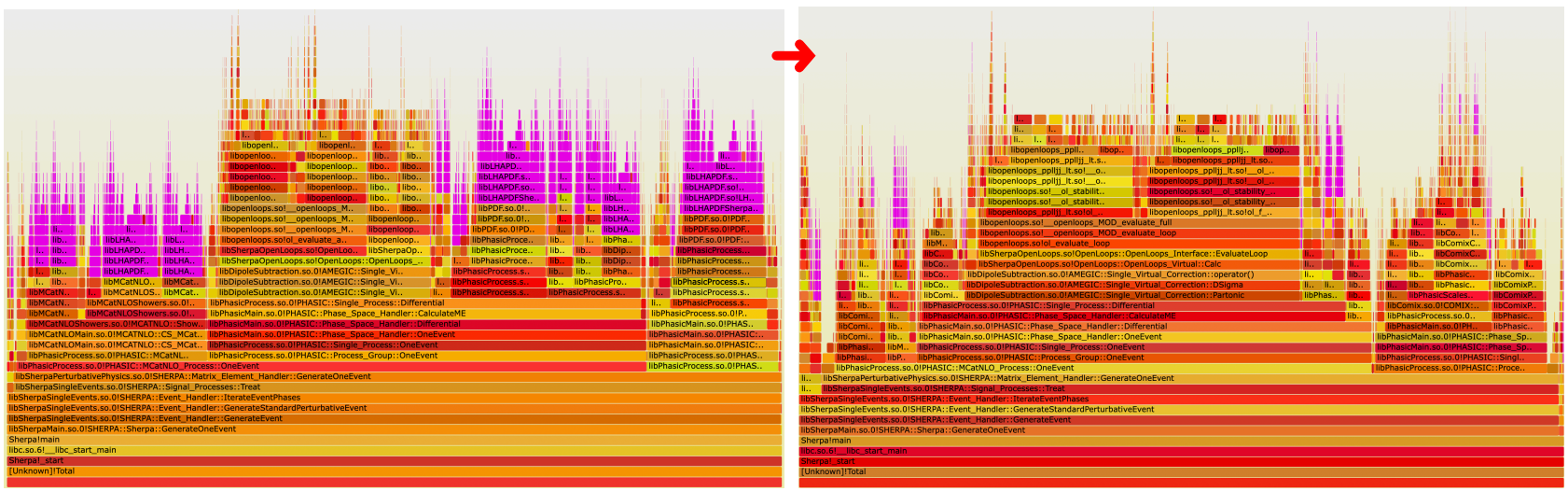
LHAPDF 6.4.0

- ➜ lots of work went into reviewing the LHAPDF interpolation logic over the summer (MCnet short-term studentship of M. Knobbe in Glasgow with A. Buckley)
  - ➜ new release out since end of last September
- ➜ latest ATLAS V+jets setup overall 30% faster using new LHAPDF release



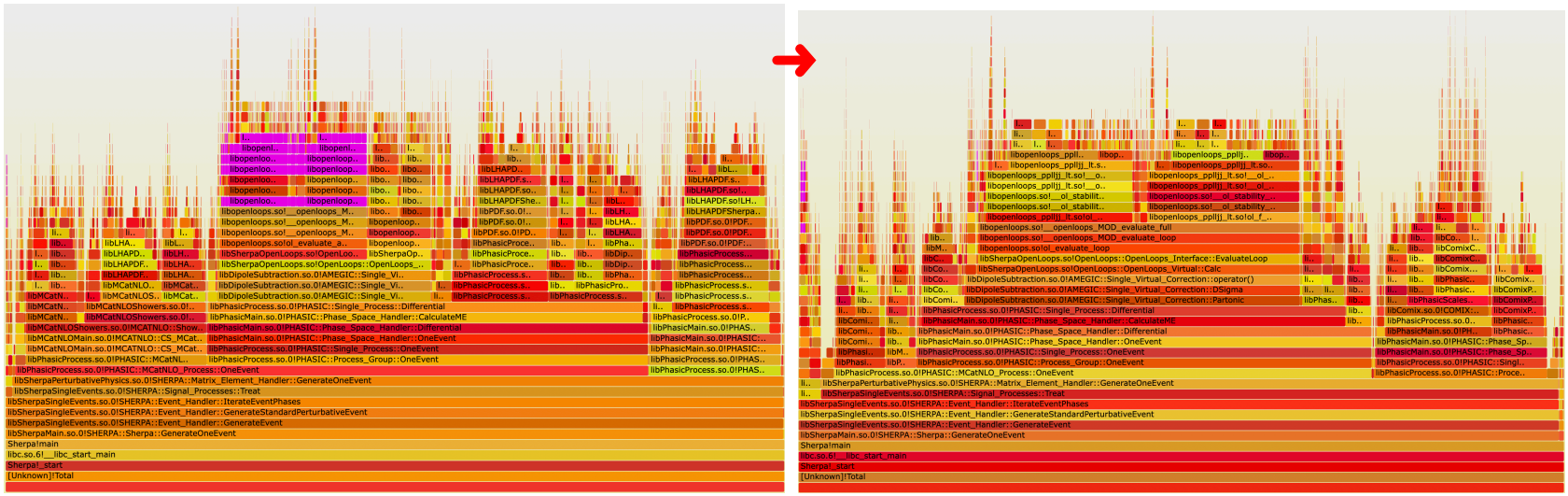
Multi-weight restructuring: the pilot run

- in Sherpa, variations are calculated at the same time with the nominal result; this makes the implementation localised, more robust and leads to less code duplication; but this approach makes it hard to postpone variations until after unweighting
- perform the unweighting using a minimal setup and once an event is accepted, rewind RNG state and re-calculate accepted event using all the bells and whistles
- achieves factor 5 speed improvement for ATLAS setup
  (using LHAPDF 6.4.0 yields additional 6% speed-up)
- pilot run reduces CPU spent on evaluating PDFs to below 10%
- `External_RNG` must implement `[CanRestore|Save|Restore]Status()`



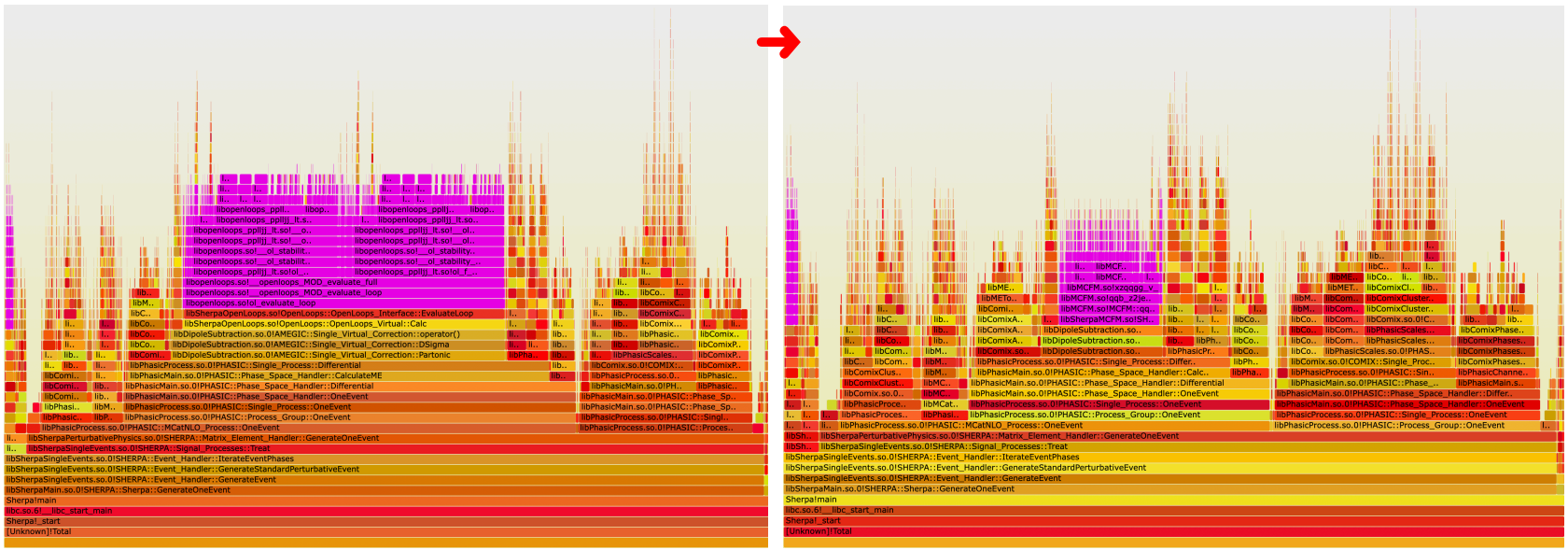
Multi-weight restructuring: the pilot run

- ➜ CPU spent on calculating EW one-loop amplitudes going from 19% down to 0.8% when using the pilot run with the ATLAS V+jets setup
- ➜ nevertheless, ∼40% of the CPU still spent on calculating QCD loops



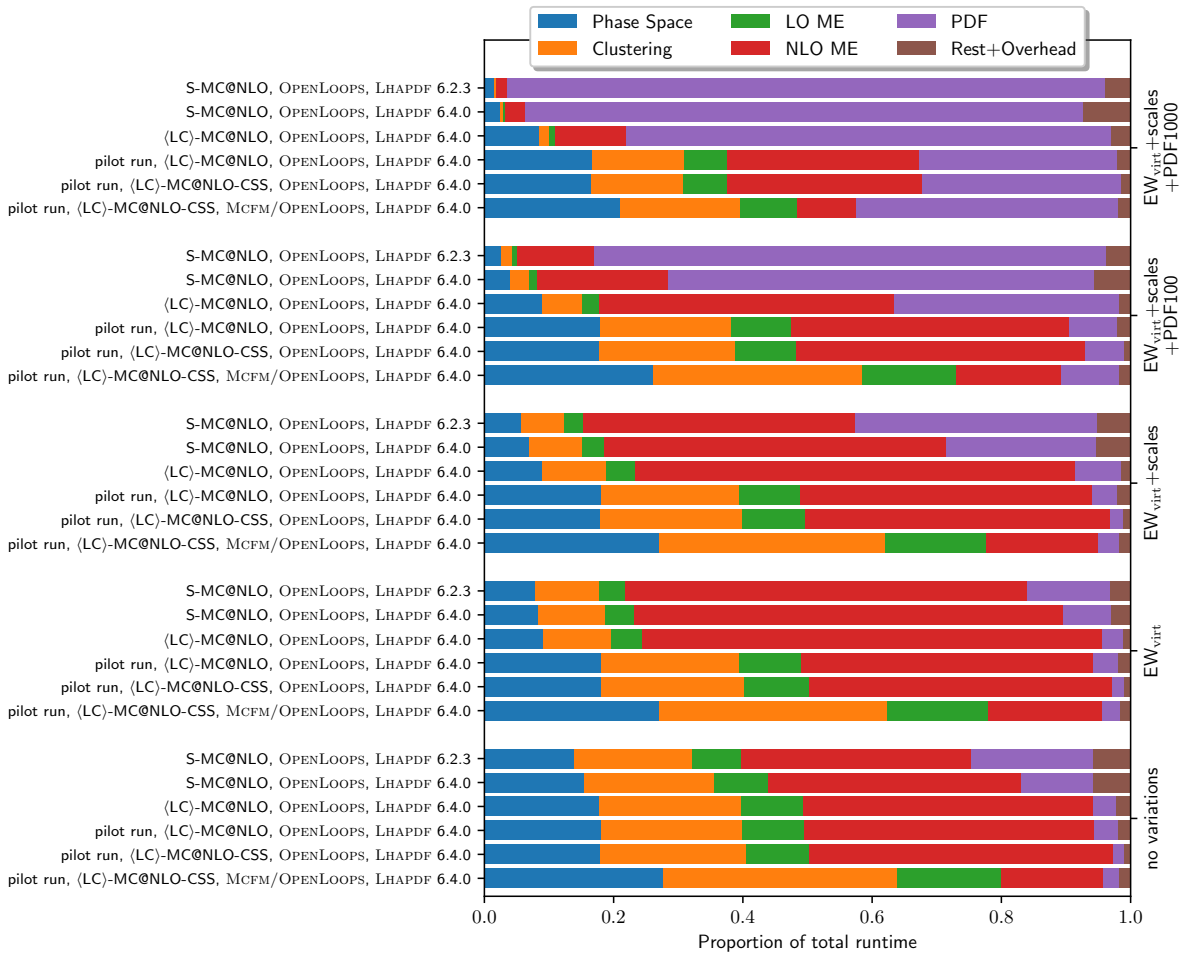
Analytic vs numerical QCD loop amplitudes

- using analytic one-loop amplitudes (if available) in the pilot run
  - using Sherpa-MCFM interface
    (J. Campbell, S. Höche, C. Preuss [arXiv:2107.04472])
- yields additional ∼35% speed improvement for the ATLAS V+jets setup



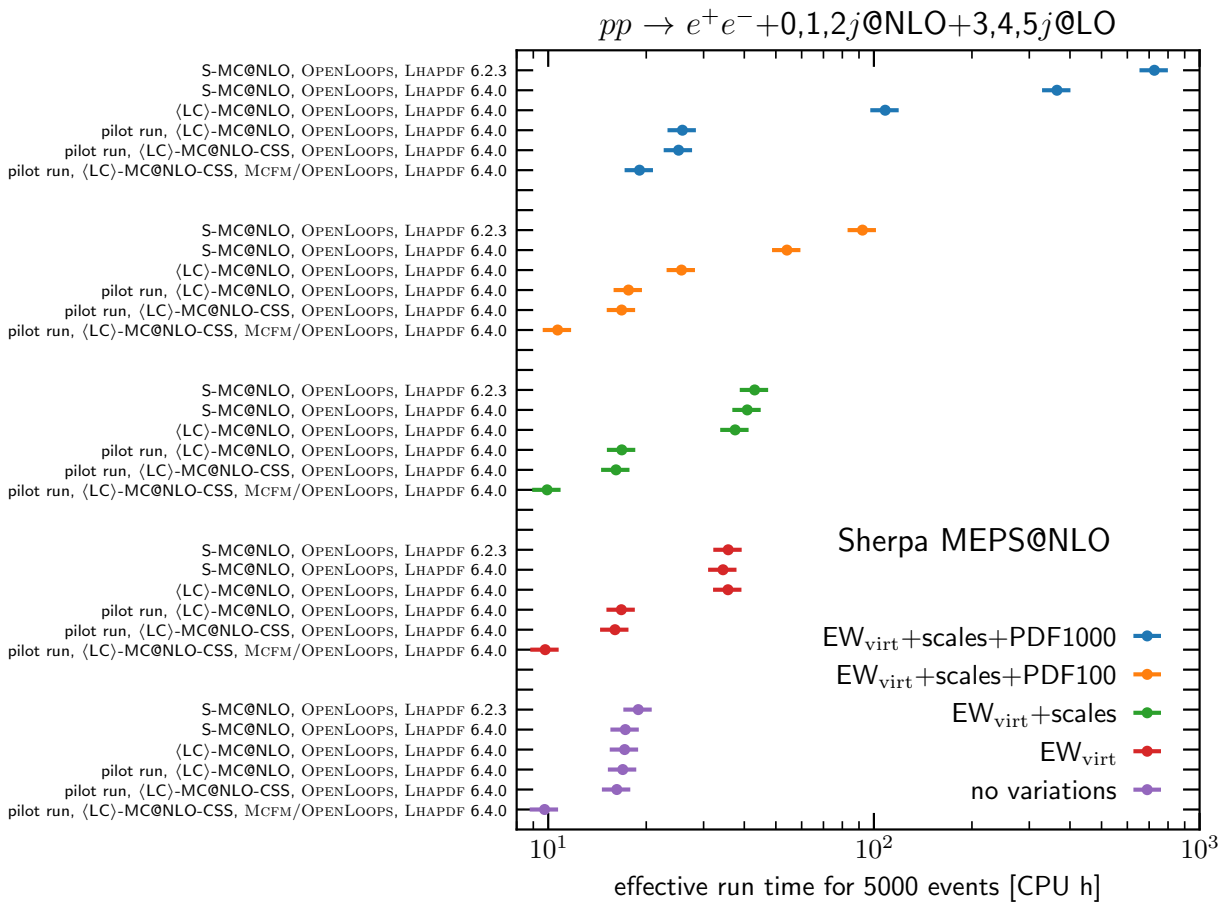
Breakdown of CPU budget in V+jets

- study the impact of improvements sequentially:
  - improved efficiency in LHAPDF (6.2.3 → 6.4.0)
  - disable subleading colour corrections in MC@NLO 1-step parton shower (`PSMODE` 0 → 1)
  - introduce pilot run in Sherpa (2.2.11 → 2.2.12)
  - defer (LC) 1-step shower until after the unweighting (`PSMODE` 1 → 2)
  - use analytic one-loop amplitudes in pilot run (OL → MCFM)



Impact on overall run time in V+jets

- study the impact of improvements sequentially:
  - improved efficiency in LHAPDF (6.2.3 → 6.4.0)
  - disable subleading colour corrections in MC@NLO 1-step parton shower (`PSMODE` 0 → 1)
  - introduce pilot run in Sherpa (2.2.11 → 2.2.12)
  - defer (LC) 1-step shower until after the unweighting (`PSMODE` 1 → 2)
  - use analytic one-loop amplitudes in pilot run (OL → MCFM)

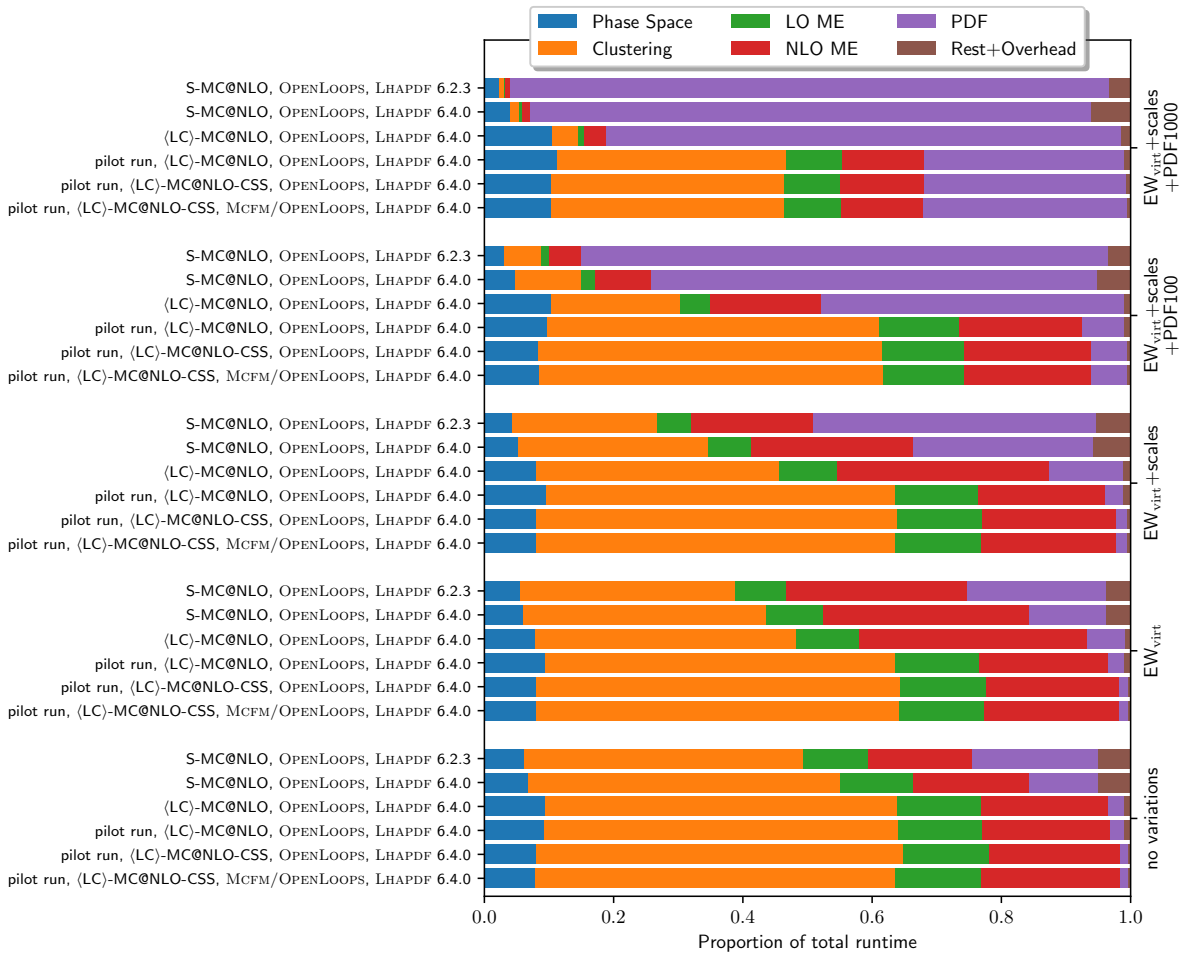


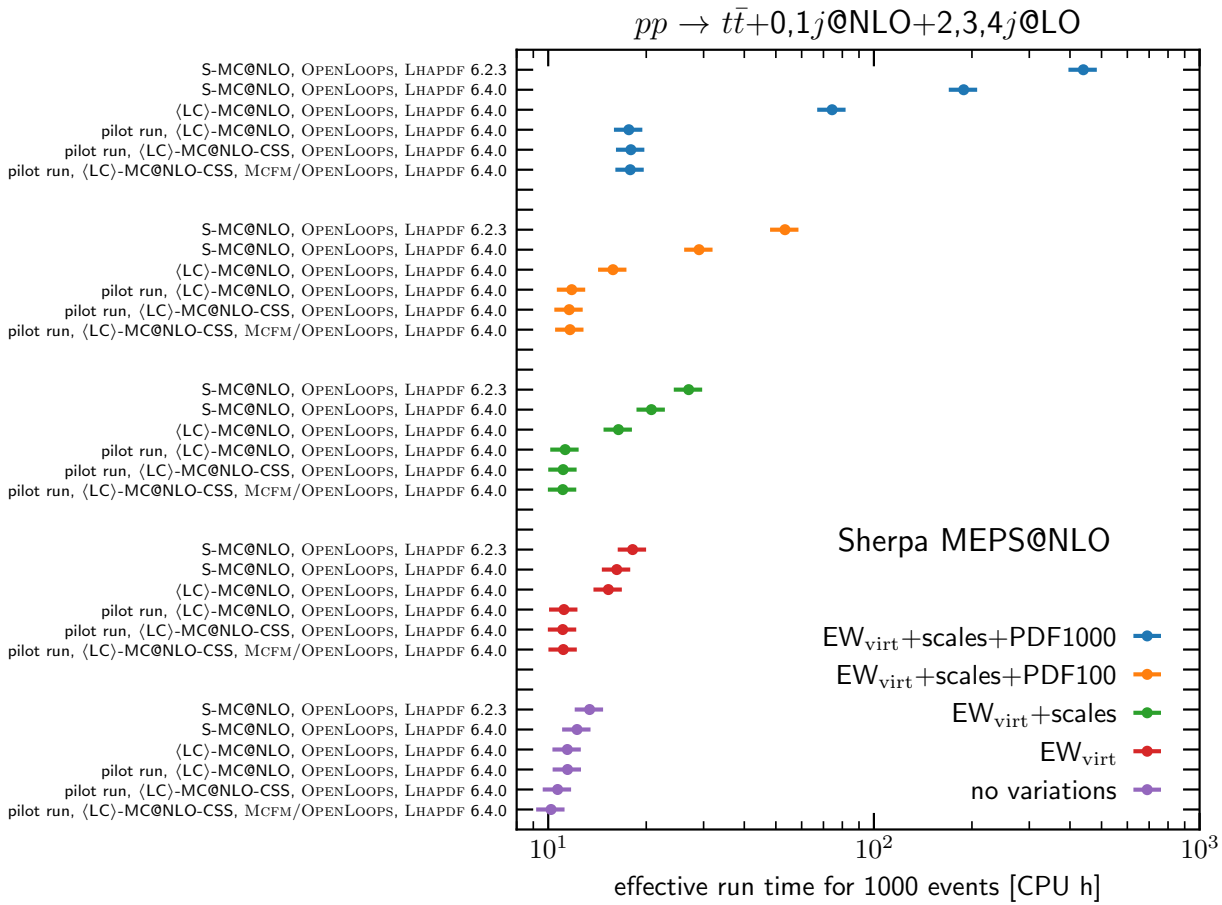
Summary

- latest LHAPDF release brings major performance improvements
  that already have a noticeable impact on overall run time
- multiweight restructuring brings a factor 5 improvement for the ATLAS V+jets setup
- using analytic QCD loop amplitudes in the unweighting brings another factor 1.5
- overall CPU gains depend on the number of multiweights
  - order of magnitude improvement in case of ATLAS V+jets setup
  - similar story for $t\bar{t}$+jets
- next steps
  - phase-space and clustering next biggest CPU sink
- related efforts
  - Reducing negative weights in Monte Carlo event generation with Sherpa
    (K. Danziger, S. Höche, F. Siegert [arXiv:2110.15211])
  - Rejection sampling using neural network event-weight estimates
    (K. Danziger, T. Janßen, S. Schumann, F. Siegert [arXiv:2109.11964])
  - GPU-accelerated ME evaluation
    (EB, W. Giele, S. Höche, J. Isaacson, M. Knobbe [arXiv:2106.06507])

Backup: Breakdown of CPU budget in $t\bar{t}$+jets

- study the impact of improvements sequentially:
  - improved efficiency in LHAPDF (6.2.3 → 6.4.0)
  - disable subleading colour corrections in MC@NLO 1-step parton shower (`PSMODE` 0 → 1)
  - introduce pilot run in Sherpa (2.2.11 → 2.2.12)
  - defer (LC) 1-step shower until after the unweighting (`PSMODE` 1 → 2)
  - use analytic one-loop amplitudes in pilot run (OL → MCFM)



Backup: Impact on overall run time in $t\bar{t}$+jets

- study the impact of improvements sequentially:
  - improved efficiency in LHAPDF (6.2.3 → 6.4.0)
  - disable subleading colour corrections in MC@NLO 1-step parton shower (`PSMODE` 0 → 1)
  - introduce pilot run in Sherpa (2.2.11 → 2.2.12)
  - defer (LC) 1-step shower until after the unweighting (`PSMODE` 1 → 2)
  - use analytic one-loop amplitudes in pilot run (OL → MCFM)