GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Institut für Theoretische Physik

Portable event generation on GPU-accelerated hardware

Max Knobbe
@ WLCG/HSF Workshop at DESY

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

About

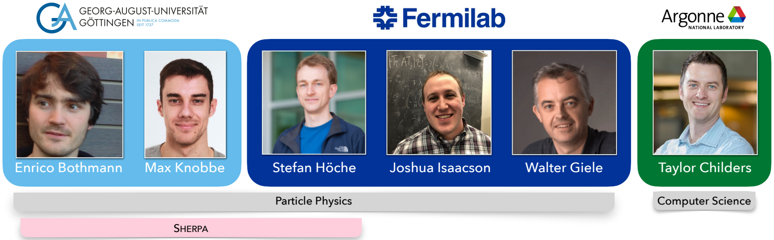
Touch on multiple Community tools:

- PEPPER: **P**ortable **E**ngine for the **P**roduction of **P**arton-level **E**vent **R**ecords
- LHAPDF
- (Sherpa?)

About this Talk

- General overview, discuss performance/portability aspects
- Some slides borrowed from E. Bothmann's talk at ACAT@2024

About the Team



Introduction

Why improve computing performance?

- High statistics at HL-LHC & excellent detector performance

  → Need for precise MCEG simulations

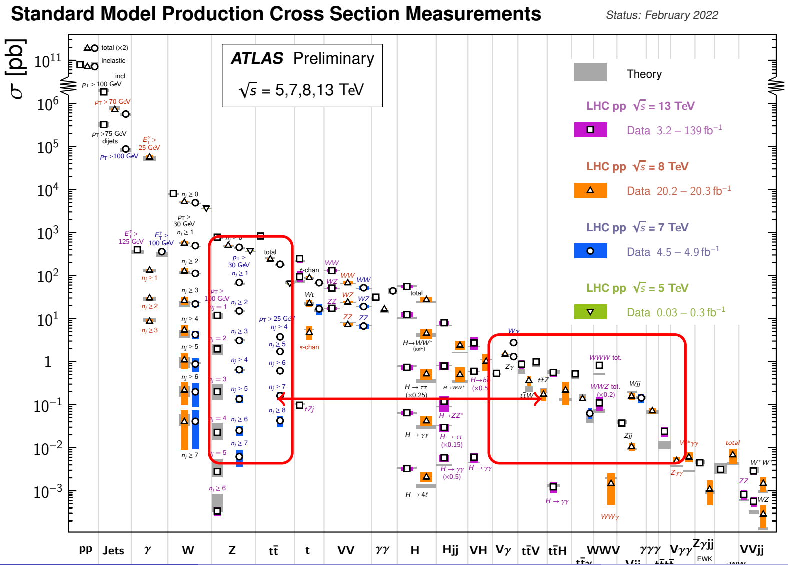
  → Poor MCEG performance can limit experimental success

  [HSF Physics Event Generator WG] arXiv:2004.13687, arXiv:2109.14938

What dominates the computing budget?

- Which physics processes?
- Parton or particle level?
- Which final-state jet multiplicities?

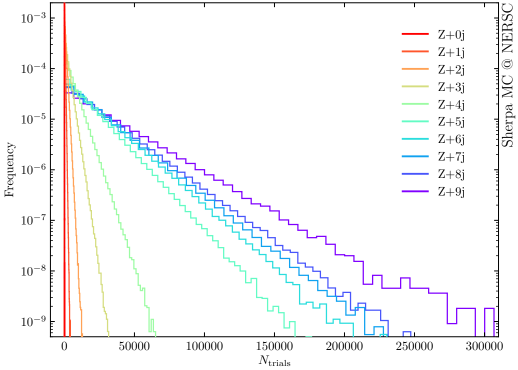
Which physics processes?



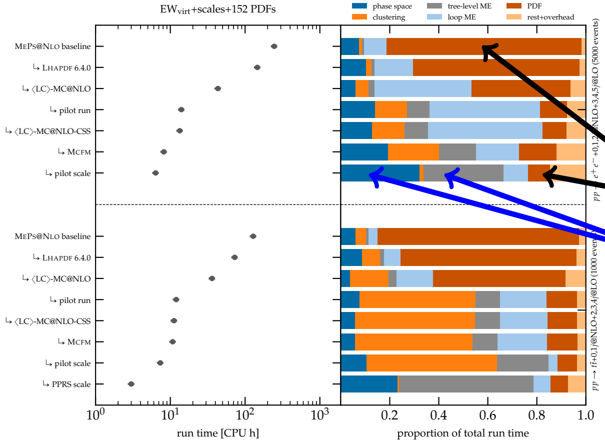
Current state of the art

- ATLAS' state-of-the-art SHERPA samples
  - $pp \to e^+e^- + 0, 1, 2\text{j@NLO} + 3, 4, 5\text{j@LO}$
  - $pp \to t\bar{t} + 0, 1\text{j@NLO} + 2, 3, 4\text{j@LO}$
- Reason: low unweighting efficiencies and expensive ME for high jet multiplicities

  [Höche,Prestel,Schulz] arXiv:1905.05120

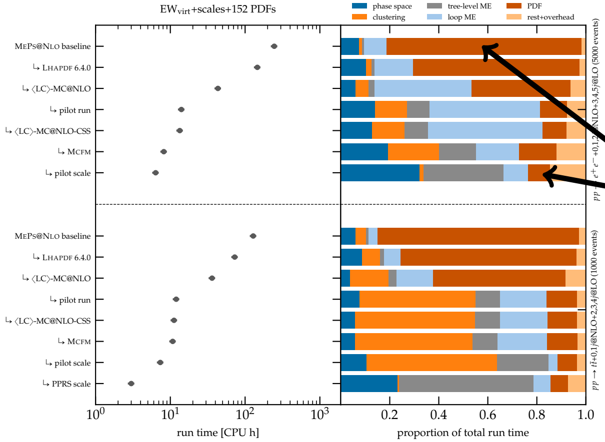


What components should we focus on?



- C.f. Chris' talk:
  Sherpa baseline performance for
- 1. PDF performance
- 2. How to further improve?
  (Phase-space + matrix elements)

What components should we focus on?

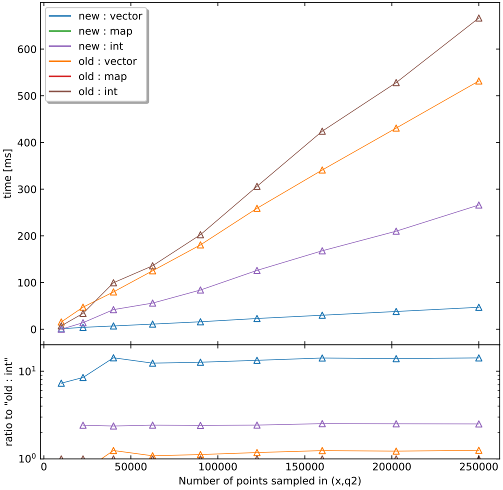


- C.f. Chris' talk:
  Sherpa baseline performance for
- **1. PDF performance**
- 2. How to further improve?
  (Phase-space + matrix elements)

LHAPDF improvements

- PDF interpolation library
- Performance gain, cf. Chris Gütschow's talk
  $\rightarrow$ 3-10x speedup per pdf-evaluation
  $\rightarrow$ more possible by changing MC workflows
- OpenMPI used for efficient initialisation
  $\rightarrow$ constant init time vs. infinite init time

  [Höche,Prestel,Schulz] arXiv:1905.05120
- Added CUDA + Kokkos interface/version
  $\rightarrow$ excellent computing performance / accuracy
  $\rightarrow$ portable version used for the remaining talk
  $\rightarrow$ performance numbers in the end

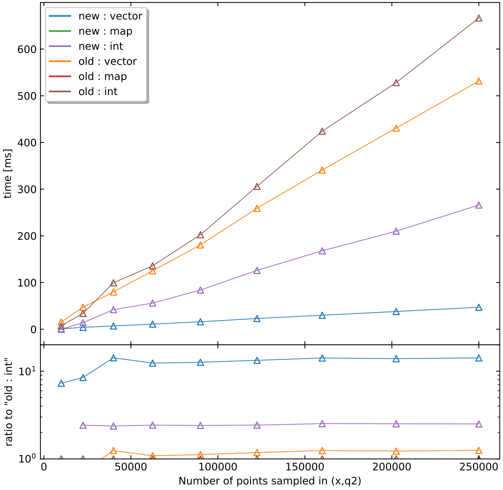


LHAPDF improvements

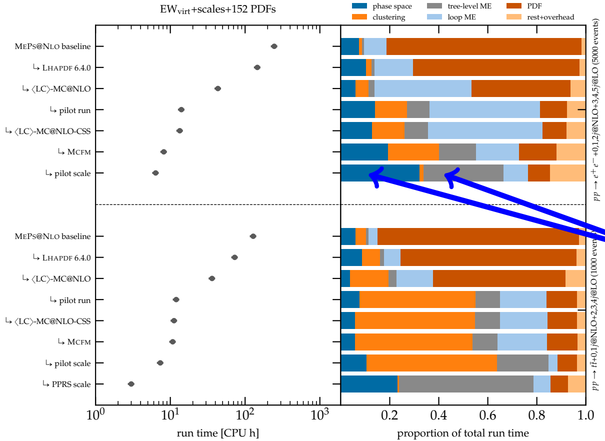
- PDF interpolation library
- Performance gain, cf. Chris Gütschow's talk
  → 3-10x speedup per pdf-evaluation
  → more possible by changing MC workflows
- OpenMPI used for efficient initialisation
  → constant init time vs. infinite init time

[Höche,Prestel,Schulz] arXiv:1905.05120

- A… erface/version
  → … rmance / accuracy
  → … the remaining talk
  → … the end

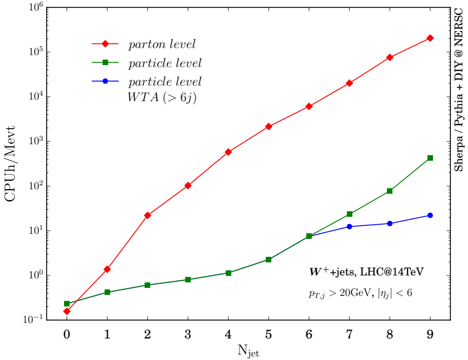


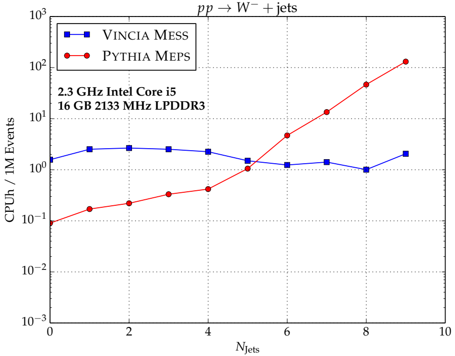
What components should we focus on?



- C.f. Chris' talk:
  Sherpa baseline performance for
- 1. PDF performance
- **2. How to further improve?**
  (Phase-space + matrix elements)

Timing distribution: scaling with multiplicity





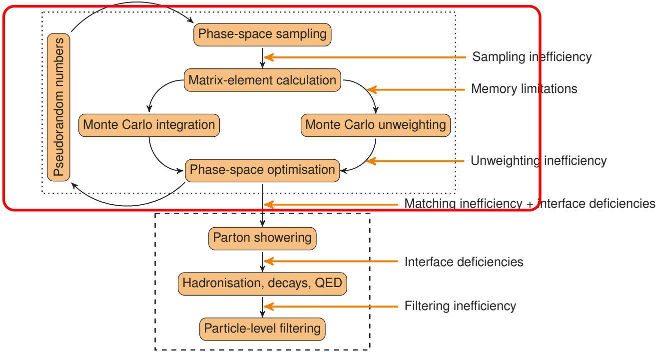
- Hard scattering simulation much more demanding than particle-level remainder [Höche,Prestel,Schulz] arXiv:1905.05120
- Complexity of merging ME&PS can be reduced to achieve linear scaling using sector showers [Brooks,Preuss] arXiv:2008.09468 so not a problem in principle

C.f. Chris:

ACCELERATING MC EVENT GENERATOR ANALYSIS
CHRISTIAN GÜTSCHOW

UCL

Other bottlenecks



➜ Lack of active development on infrastructure tools (LHE, HepMC, ...) set to become major bottleneck going forward (➜ upcoming [IPPP workshop] on MC support tools)

WLCG/HSF workshop, Hamburg, 16 May 2024 chris.g@cern.ch 5/16

Algorithms: matrix element

Figure of merit

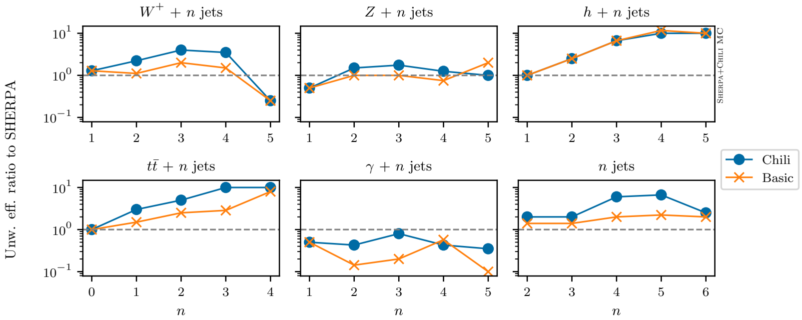
unweighted event generation throughput for highest jet multiplicity

e.g. $pp \to e^+e^- + 5j$, $pp \to t\bar{t} + 4j$ or more

- Berends–Giele recursion for best multi-jet scaling behaviour
- Colour summing for lockstep GPU evaluation
  - Use minimal colour basis developed by amplitude community $\mathcal{O}((n-1)!^2) \to \mathcal{O}((n-2)!^2)$
    [Melia] arXiv:1304.7809 arXiv:1312.0599 arXiv:1509.03297
    [Johansson,Ochirov] arXiv:1507.00332
  - Our implementation generalises it to $\ell\ell$+jets amplitudes
- Helicity sampling to avoid additional 2^n scaling

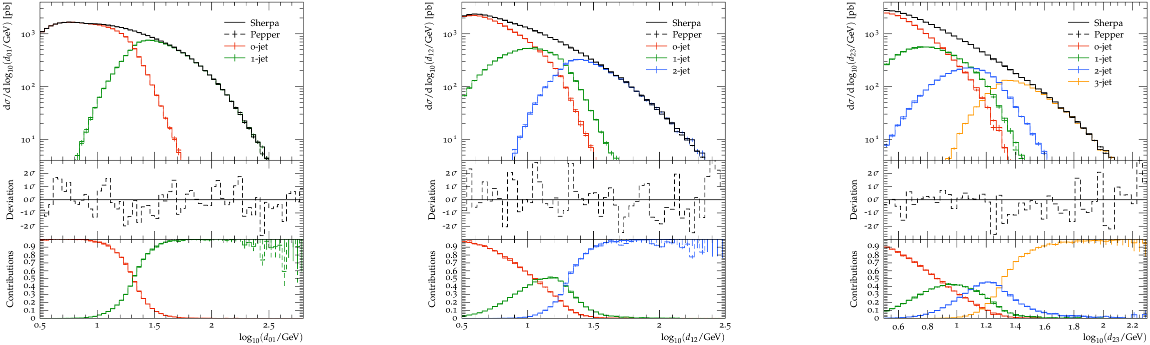
Algorithms: phase space

- CHILI phase-space generator uses simple (MCFM-inspired) structure: one t-channel + adjustable number of s channels [EB et al.] arXiv:2302.10449
  - Portable (ported builtin CHILI in PEPPER)
  - RAMBO-like speed
  - Efficiency on par with recursive COMIX phase-space
    - → Ideal to provide on-device ML training data for many jets



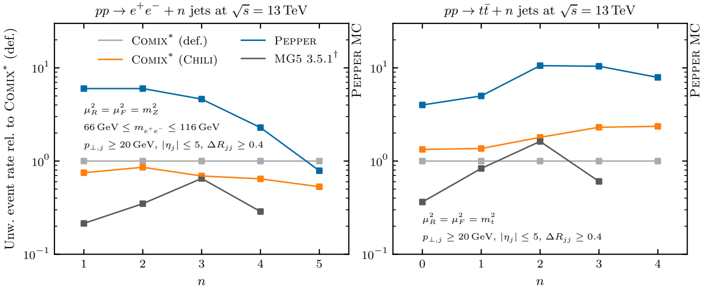
Algorithms: I/O and toolchain integration

- PDF via LHAPDF, ported to CUDA and Kokkos
- Particle-level simulation via Sherpa or Pythia
  → LHEH5-based framework [EB et al.] arXiv:2309.13154 → Chris' talk



- Test of complete LHEH5-based simulation pipeline with Pepper+Sherpa [EB et al.] arXiv:2309.13154
- Additional 3× speed-up for ATLAS MEPS@NLO $pp \to e^+e^- +$ jets set-up → Sherpa v2.3.0 (Sep '23)

Baseline unweighted event generation performance



- Unweighted event throughput compared to Comix*
- Constitutes baseline single-threaded performance of currently available competitive algorithms
- Novel standalone Pepper performs better than Comix, but Pepper's real goal is portability [EB et al.] arXiv:2311.06198

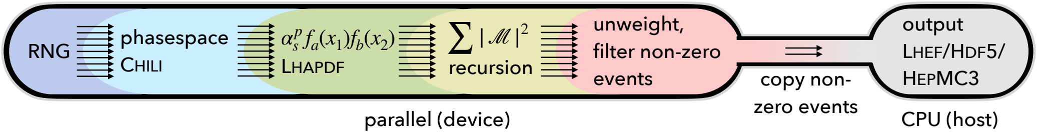
Numbers generated on Intel Xeon E5-2650 v2
* Partonic processes split into to g/q groups (not Sherpa standard)
† Modified to match efficiency convention of [Gao et. al] arXiv:2001.10028

Why portability?

- Many computing vendors, heterogeneous architectures
- (Pre-)Exascale computing systems intentionally diverse

Portability is baked into PEPPER

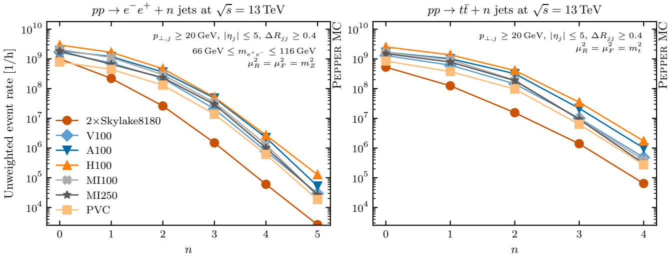
- Focus on highest multi (e.g. $e^+e^- + 5$, $t\bar{t} + 4$)
  this is beyond small scale computing → WLCG / HPC
- 10–20 years ago: Homogeneous CPU+RAM architectures
- This is undergoing a big change (partly due to AI trends)
  - HPC moves to exascale era → scalability
  - GPU acceleration → portability
- PEPPER addresses both aspects with MPI, HDF5 and Kokkos
- PEPPER parallelises the entire parton-level event generation:



- Tested Xeon CPU, Intel/AMD/Nvidia GPU, HPC systems
  - ✓ Covers all (pre-)exascale architectures on previous slide
  - ✓ Scalable from a laptop to a Leadership Computing Facility

Comparing runtimes on relevant architectures

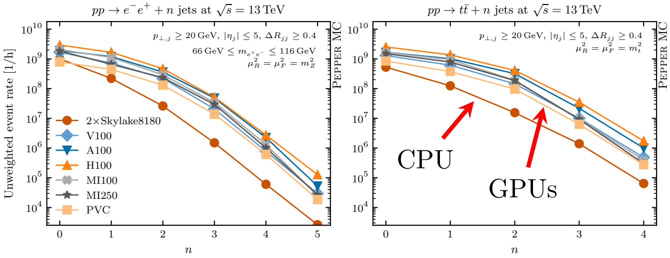
- Excellent performance across a wide range of architectures
- Portability provided by Kokkos: one code-base compiled for different architectures



MEvents / hour	2×Skylake8180	V100	A100	H100	MI100	MI250	PVC
$pp \to t\bar{t} + 4j$	0.06	0.5	1.0	1.7	0.4	0.3	0.3
$pp \to e^- e^+ + 5j$	0.003	0.03	0.05	0.1	0.03	0.03	0.02

Comparing runtimes on relevant architectures

- Excellent performance across a wide range of architectures
- Portability provided by Kokkos: one code-base compiled for different architectures



MEvents / hour	2×Skylake8180	V100	A100	H100	MI100	MI250	PVC
$pp \to t\bar{t} + 4j$	0.06	0.5	1.0	1.7	0.4	0.3	0.3
$pp \to e^- e^+ + 5j$	0.003	0.03	0.05	0.1	0.03	0.03	0.02

Portability: Aurora example

- Estimate "roughly 330 billion [leptonically decaying V+jets] events" required for HL-LHC
  [ATLAS] arXiv:2112.09588
  - "**Sherpa 2.2.11 setup would exceed budget by 16%**"
  - Assume all 330 billion events are Z+4j
    Production cost at parton-level would be:
    - ⋆ 240M CPUh COMIX @ Intel E5-2650 v2 CPU
    - ⋆ 380k GPUh PEPPER @ Nvidia A100 →

      This would be 8h on Aurora (with PVC)

Conclusions and outlook

Status

✓ Portable parton-level multi-jet event generator Pepper
  Achieves scalability from a laptop to a Leadership Computing Facility

Outlook

- Use synergies Pepper/Chili ↔ on-device training ↔ ML
- Add more processes to Pepper, work towards NLO

Discussion points

- Regularly updated per-process event generation cost data from ATLAS & CMS?
  (time/energy/money/...)
- Can we get together and establish HPC/GPU workflows
  with hep-ex & LCFs? (Usability ⇜⇝ Flexibility, Portability ...)
- Expected adoption of HPC resources by LHC computing?