GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Institut für Theoretische Physik

Portable event generation with PEPPER

Max Knobbe

@ Event generators and N(n)LO codes acceleration

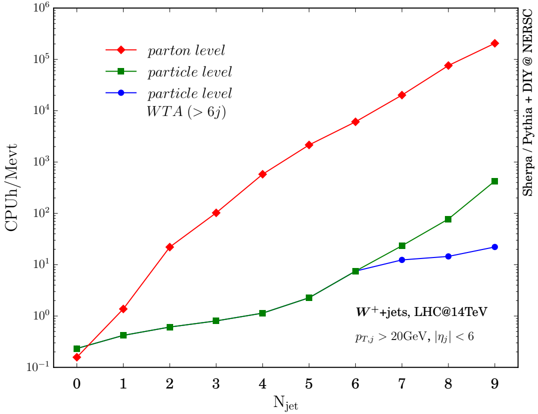
MCnet

Gefördert durch DFG Deutsche Forschungsgemeinschaft

Defining objectives

- Majority of time spend in parton level components
  [Höche,Prestel,Schulz] arXiv:1905.05120
- Scales exponentially with multiplicity
- Experiments require large number of unweighted events
- Relevant processes e.g. $V + j, t\bar{t} + j$
  [ATLAS] arXiv:2112.09588

Objective:
high-multiplicity unweighted events

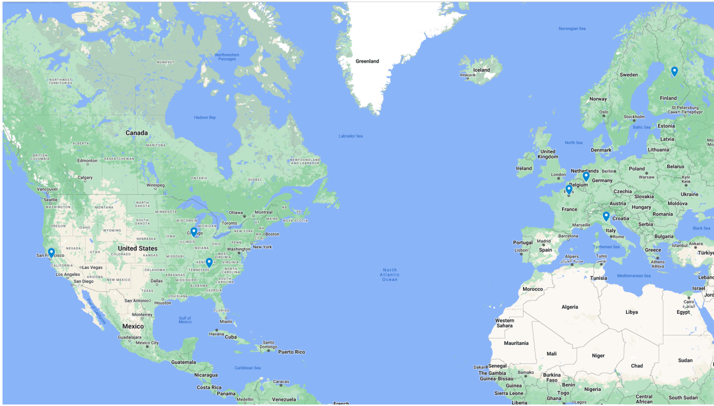


Challenges and opportunities in the modern HPC landscape

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

Challenges and opportunities in the modern HPC landscape

- Many vendors, fairly heterogeneous architectures
- (Pre-)Exascale computing systems intentionally diverse
- MC-Generation runs on none of these
- Monte Carlo generation ideal testground/pathfinder for HEP-Software



The Pepper approach

Scale of problem justifies redevelopment!

Clearly defined objectives + deliverables:

1. Identify and work on bottlenecks
   $\rightarrow$ Current candidates: V+jets, $t\bar{t}$+jets, pure jets
   $\rightarrow$ No intent to be general purpose, but needs to feed into existing tool-chain
2. Figure of merit: performance of unweighted event generation
3. Scalability
   $\rightarrow$ Large problems require large number of computing units
4. Portability
   $\rightarrow$ Make use of existing Hpc

Code Release: [Bothmann, Childers, Giele, Höche, Isaacson, MK; arXiv: 2023:06198]

(Some) Components of a MC Computation

$$\sigma_{pp\to X_n} = \sum_{ab} \int dx_a dx_b d\phi_n f_a(x_a,\mu_F^2) f_b(x_b,\mu_F^2) \times \left|\mathcal{M}_{ab\to X_n}\right|^2 \Theta(p_1,...,p_n)$$

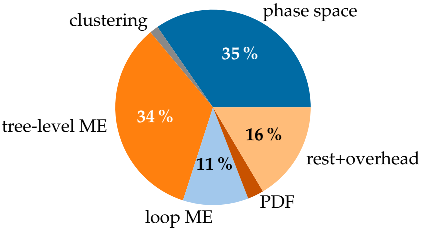
- Large portion of MC time spend in ME + PS

cf. [2209.00843], Chris Gütschows talk

Components we need to consider:

- Tree-level Matrix elements
- Phase space generation
- PDF's

$pp \to e^+e^-$+0,1,2j@NLO+3,4,5j@LO



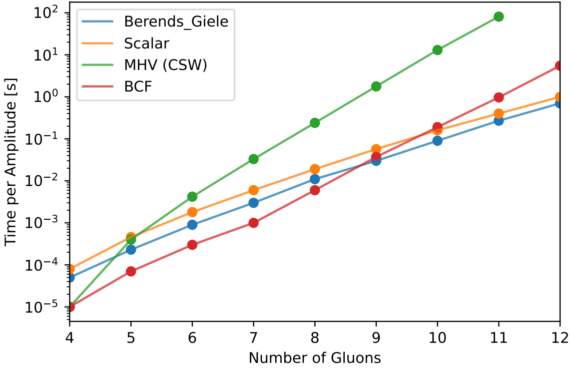
Bothmann et al. [2209.00843]

The Amplitudes

- Strategies to compute tree-level amplitudes
  1. Berends-Giele like recursion
  2. Scalar
  3. MHV (CSW)
  4. BCF
- Rely on performance studies from early 2000's

  hep-ph/0602204,hep-ph/0607057
- We are interested in best scaling behaviour / performance for multi-jet processes

⇒ **Choice: Berends-Giele recursion**



Based on numbers from hep-ph/0602204

The Color & Helicity Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

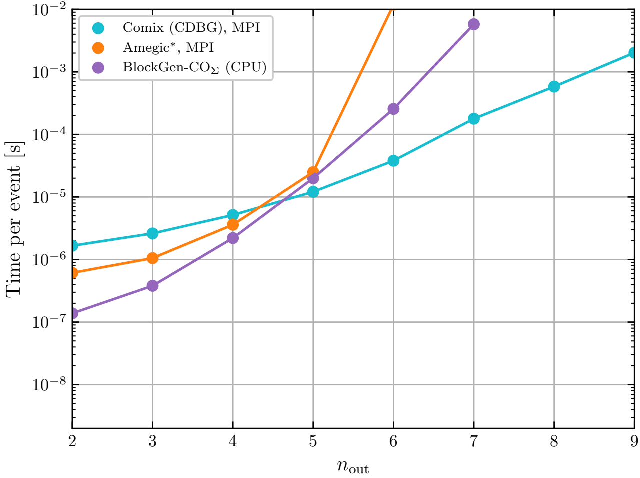
Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors**

Helicity-treatment:

- Picture less clear, still allow multiple options



The Color & Helicity Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

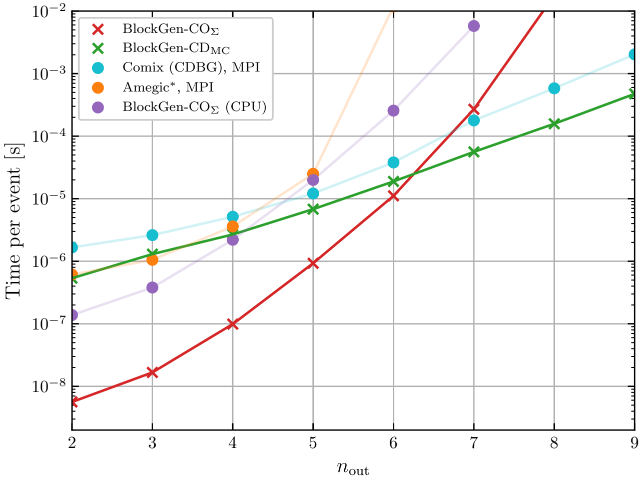
Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors**

Helicity-treatment:

- Picture less clear, still allow multiple options



The Color & Helicity Sum [Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

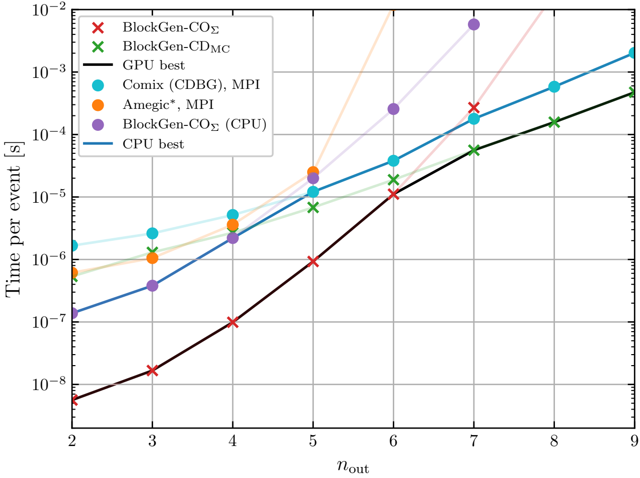
Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors**

Helicity-treatment:

- Picture less clear, still allow multiple options

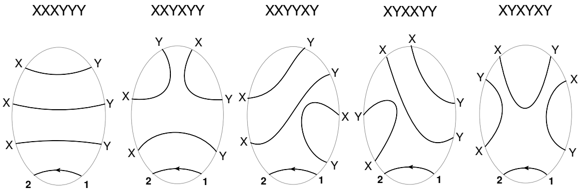


From Gluon-only to V+Jets

- Introduce spinors (Weyl for massless, Dirac for massive particles)
- Add more general QCD three point vertices
- Straight-forward for helicity-sum and Berends-Giele recursion
- First time in a code aimed for production: use minimal QCD color-basis $\{A(1, 2, \sigma), \sigma \in \text{Dyck}\}$

  [T. Melia 1304.7809 & 1312.0599 & 1509.03297; H. Johansson, A. Ochirov, 1507.00332]

  → Allows to fix one fermion line, remaining permutations are given by Dyck-Words
  → Four particle Dyck Words: ()(), (())
  → Significantly fewer amplitudes to compute
- Include EW particles after QCD basis has been set up



[1304.7809]

Phase space generator: Chili (SciPost Phys. 15 (2023) 169)

- Differential phase space element for an n-particle final state

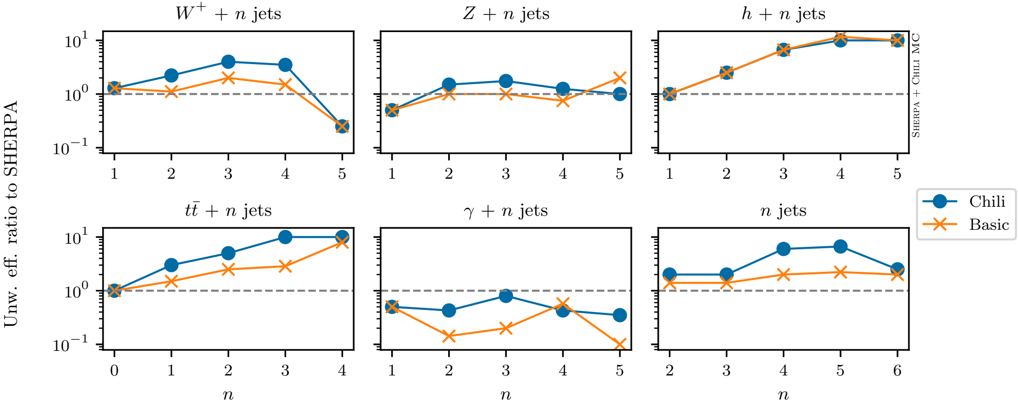
$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \left[\prod_{i=1}^{n} \frac{\mathrm{d}^3\vec{p}_i}{(2\pi)^3\,2E_i}\right](2\pi)^4\delta^{(4)}\left(p_a+p_b-\sum_{i=1}^{n}p_i\right).$$

- Standard factorization formula

$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \mathrm{d}\Phi_{n-m+1}(a,b;\{1,..,m\},m+1,\ldots,n)\,\frac{\mathrm{d}s_{\{1,..,m\}}}{2\pi}\,\mathrm{d}\Phi_m(\{1,..,m\};1,\ldots,m)\,.$$

- Use t-channel + adjustable number of s-channels
- Basic strategy: use single t-channel and only add s-channel resonances when required
  → easy to combine with Vegas
  → lean implementation allows for portability

Chili performance compared to Sherpa default

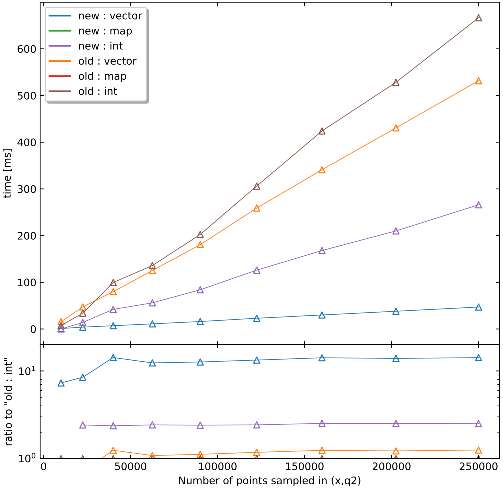


- Efficiency en par with basic Chili on par with complex recursive Comix phase-space
- Basic version excellent choice for modern architectures

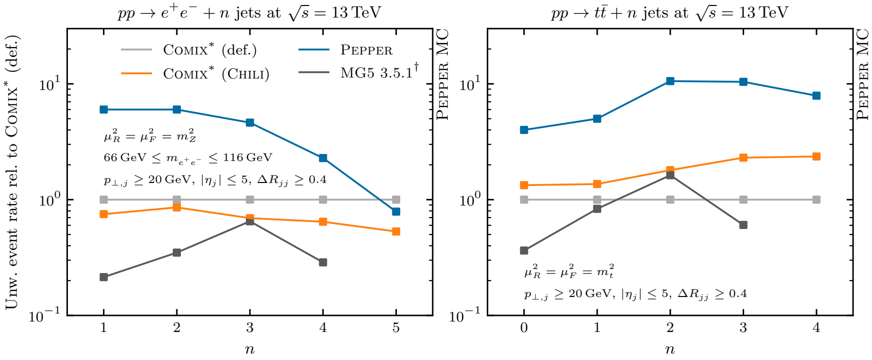
Making LHAPDF ready

- Byproduct of porting exercise
  → PDF evaluations not critical
  → copying of data is
- 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
- Added CUDA + Kokkos interface/version
  → excellent computing performance / accuracy
  → portable version used for the remaining talk



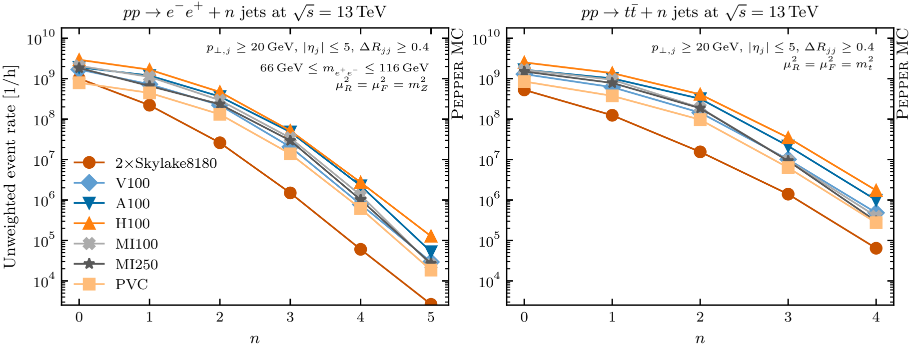
PEPPER+ CHILI: baseline CPU performance



- First complete CHILI + PEPPER benchmark
  → PEPPER + LHAPDF, colour-summed matrix elements
  → Physics motivated change in parametrization of azimuthal angle integrals
  → helicity treatment: sampled

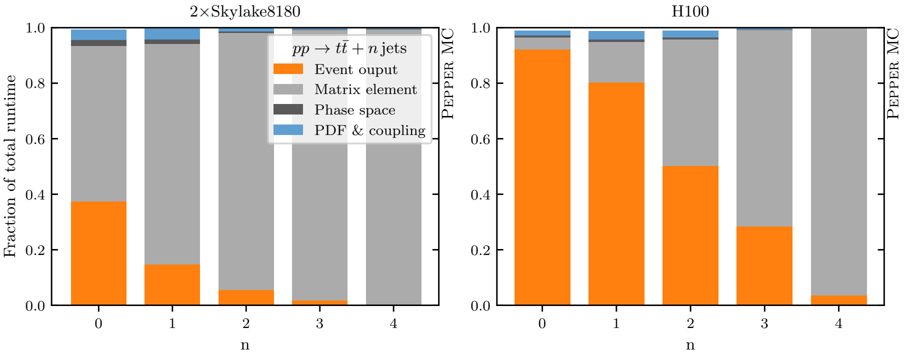
Comparing runtimes on relevant architectures

- 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

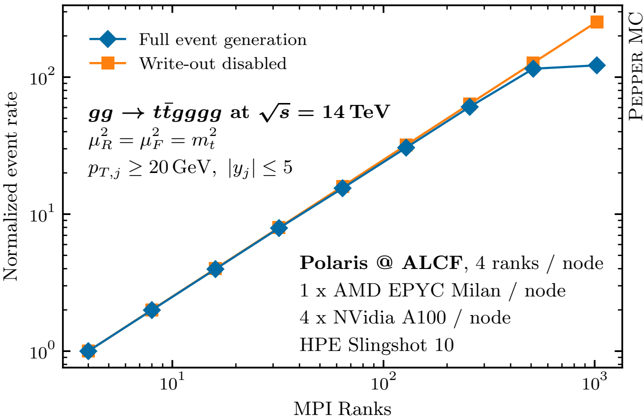
Timing details



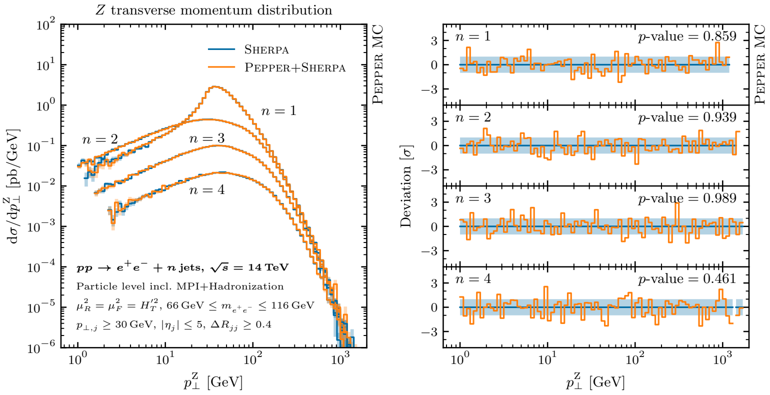
- Lower multiplicities are limited by write-out speed
  $\to$ No more need for computing improvements, but faster I/O
- Computing becomes relevant component only for large multiplicities

Scalability

- Test scalability for up to 1024×A100's
- Scaling violation due to I/O problems
  → currently investigated at ALCF
- Equivalent technology to [2309.13154]
  → established scaling to up to 16k threads at NERSC
- These problems don't show up for a couple of nodes or low data volume
  → important benchmark



Validation + Pipeline into existing tools



- Validated against SHERPA
  $\rightarrow$ for $V + j$, $t\bar{t}$+j, single multiplicity and multi-jet merged
- Writeout of HDF5 files, processable via SHERPA & PYTHIA

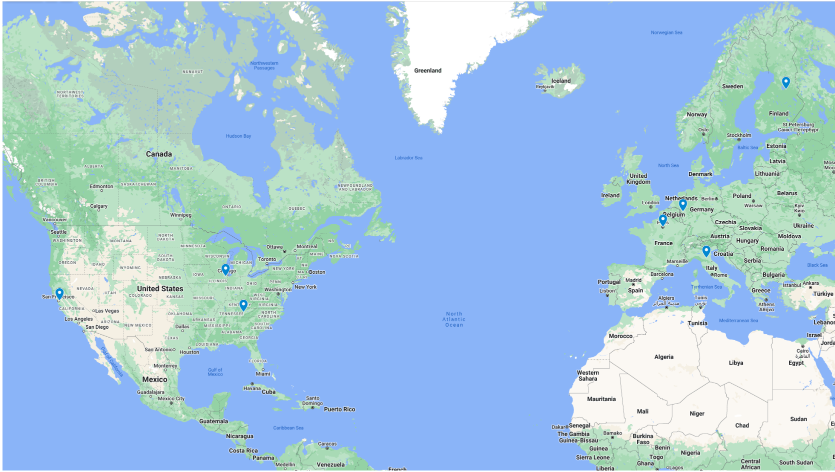
Example application: Atlas V+jets sample for HL-LHC

Disclaimer: Order of magnitude computation, clearly this should be done by experts

- Atlas V+jets sample: 330×10^9 unweighted events [ATLAS] arXiv:2112.09588
- Complex setup, good proxy: $pp \to e^+e^- + 4j$
- How much computing time would this require on current machines?

Time required for Atlas V+jets sample for HL-LHC

- Back to the HPC map



Time required for Atlas V+jets sample for HL-LHC

- Back to the HPC map
- PEPPER runs on all leading systems in EU and U.S.
- Time for V+j sample:
  - 4h Frontier
  - 6h Aurora
  - 8h Leonardo
  - 15h Lumi

Conclusion

- Pepper is ready for production-level use
  in many processes to be simulated at high fidelity at the LHC
- Parton-level event generation will now no longer contribute to the projected shortfall in computing resources during Run 4 and 5 and the high-luminosity phase of the LHC
- Enables the LHC experiments to use most HPC facilities

- Need clear identification of other critical bottlenecks
- Remaining problem: Embedding in experimental workflows