PYTHIA report for LHCC Review

Stephen Mrenna

Fermilab

May 20, 2021

Who we are:

	Status	Special Role	Location
Christian Bierlich	Senior	Webmaster, ALICE contact	EUR
Christian Preuss	Junior		EUR
Ilkka Helenius	Senior		EUR
Leif Gellersen	Junior		EUR
Leif Lönnblad	Senior		EUR
Marius Utheim	Junior		EUR
Nishita Desai	Senior		India
Peter Skands	Senior	Spokesperson	PAC
Philip Ilten	Senior	Release Manager, LHCb	US
Rob Verheyen	Junior		EUR
Smita Chakraborty	Junior		EUR
Stefan Prestel	Senior	ATLAS contact	EUR
Stephen Mrenna	Senior	CMS	US
Torbjörn Sjöstrand	Senior		EUR

The PYTHIA framework

Hard Interactions

- LO/Born (QCD, DY, Z')
- LHE interface (text,HDF5)

Radiative Corrections

- Dipole-like PS (with some ME corrections)
- DIRE (co-developed with Sherpa)
- Vincia (antenna evolution, including EW)
- EW Sudakovs

Multiparton Interactions

- Based on T. Sjöstrand and M. van Zijl, many variants

Hadronization

- Lund string, variants

Decays

- MEs, EFT, Phase Space
- Dedicated τ-package

BSM

- Hidden Valley, R-hadrons, generic new resonances
- SLHA interface

Other

- γ-beams/collisions, Heavy Ions

The Pythia Recipe

Good Documentation

Prompt Response to Bug Sightings, User Questions

Lack of External Dependencies (such as ThePEG, Root, etc)

Easy extensions via UserHooks and (S)LHA interfaces

Don’t "duplicate" efforts of other groups unless there is a good physics or computing reason

General Considerations

Are there plans/funds in place to continue support through HL-LHC?

- Dependent on funding agencies and/or lab priorities
- International, multi-institutional collaboration with no formal constitution
- Big challenge is to find permanent positions for junior people (the future of the collaboration)

What major physics updates do you foresee for HL-LHC?

- NLO parton shower
- Resonance-aware matching and NLO subtraction
- Fully differential NNLO matching for important processes
- Full SM parton shower and NLO EW matching
- Automated ME-corrections interfacing with ME generators
- Forced hadronization/rare hadron production
- QED shower that also works for decays

What major software updates are foreseen for HL-LHC? Main bottlenecks?

- Few of the proposed physics updates are trivial – will require some large code additions and a validation process
- The addition of VINCIA and DIRE as core code was already a major (challenging) update!
- Changes for MP/GPU/Concurrency?
- Main bottleneck: finding time and people

Are there issues/areas of work where help from HSF/expts may be needed?

- Restructuring of code
- Porting of code/algorithms to GPUs(?) Is there any use case?
- Profiling for things people actually use our code to do in the HL-LHC era
- HDF5 usage/exploitation

ME+PS generator specific considerations

What are the current CPU performance bottlenecks & how being worked on?

- Need profiling
- Veto efficiency
- See slides

What improvements in compute performance are expected for Run 4/5?

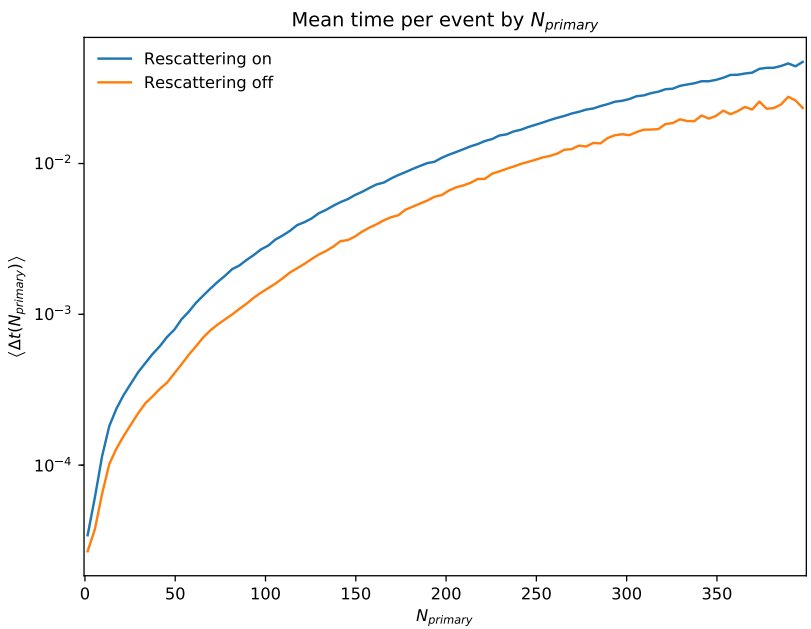
- Unknown
- We need to learn more about VINCIA and DIRE performance
- We know that VINCIA shower (see slides about merging) is an order of magnitude slower – no people to work on this

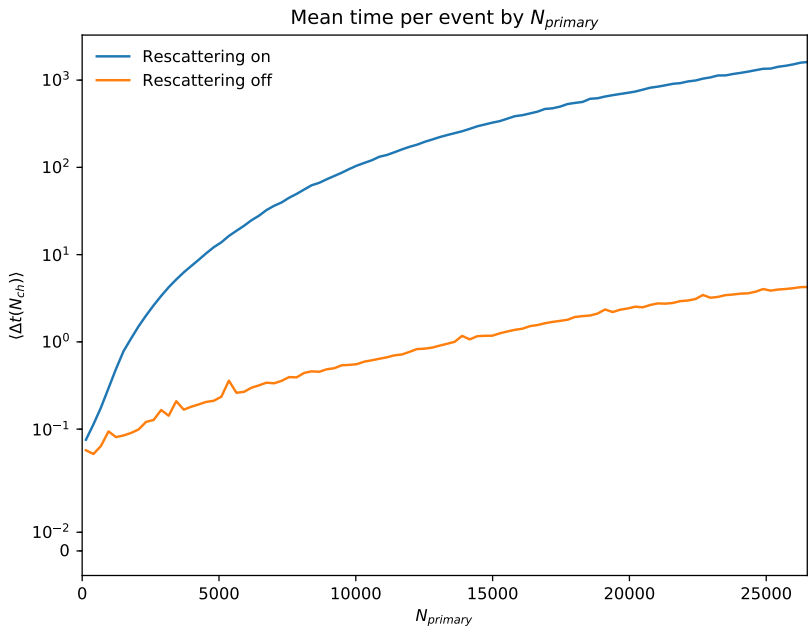
LHCb/CMS care about HI generators

$\langle$time/event$\rangle$: Intel(R) Core(TM) i7-6700K CPU at 4.00GHz.

Case	Resc. off	Resc. on	Ratio
pp	2.24 ms	4.02 ms	1.79
pPb	6.40 ms	25.6 ms	4.00
PbPb	0.594 s	150.4 s	253

$\langle$time/event$\rangle$ pp (Left) PbPb (Right) at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV.





Performance Analysis

No one complains. Never told we were a bottleneck. If we are, we are willing to partner to mitigate problems.

Will show a few slides on what we know

- Timing and Memory Usage in some merging configurations
- Timing for parton shower variations
- Multi-threading

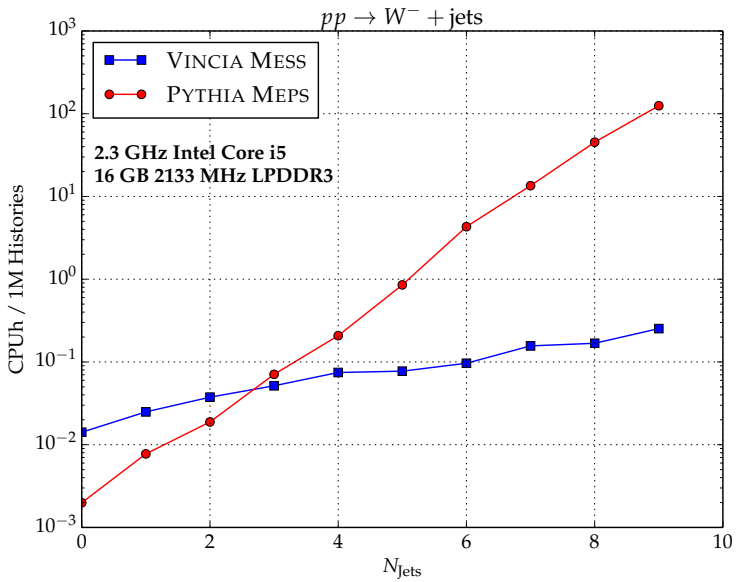
Note: VINCIA and DIRE do not have currently a large user-base, but now integrated directly as core code

Some Timing Information on Merging

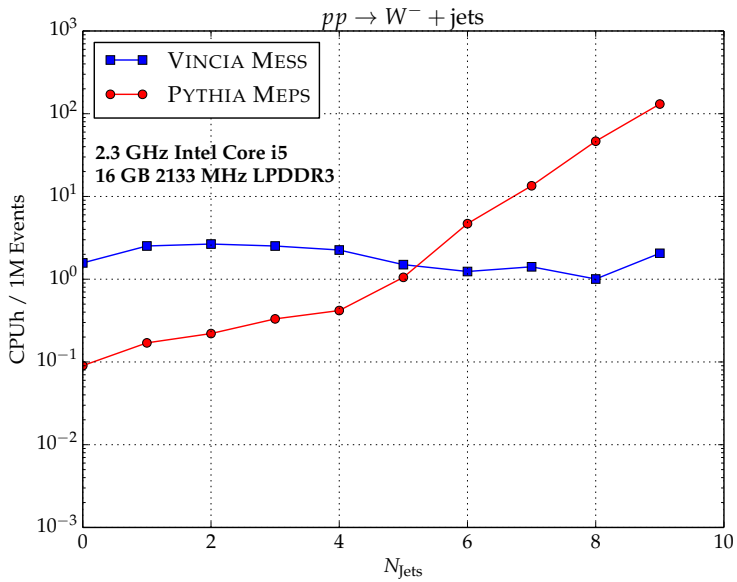
CKKW-L for PYTHIA vs VINCIA

VINCIA sector shower (SS): # histories limited by sector

History construction



Parton-level event generation



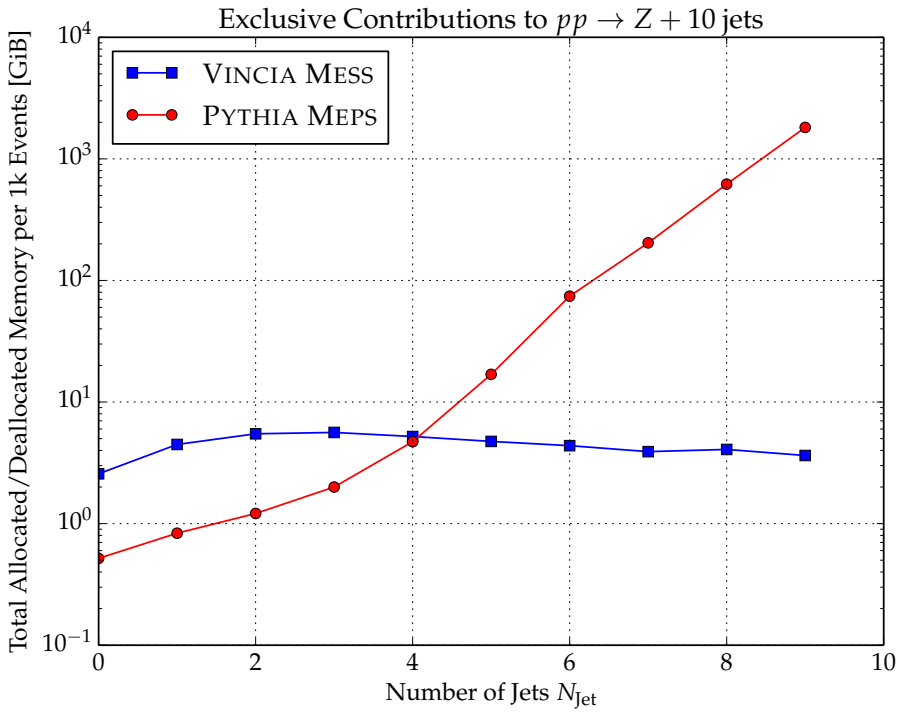
PYTHIA/VINCIA CPU time scaling; $pp \to W^- + \text{jets}$ merging at $\sqrt{s} = 14$ TeV.

https://doi.org/10.1016/j.cpc.2021.107985

Some Memory Usage Information

CKKW-L for PYTHIA vs VINCIA

VINCIA sector shower (SS): # histories limited by sector



PYTHIA/VINCIA memory usage scaling in $pp \to Z +$ jets merging at $\sqrt{s} = 14$ TeV.

Impact of Parton Shower Variations

Estimating theory uncertainties event-by-event

Veto method of PS is delayed until late in the algorithm – allows us to calculate the no-branching probability and conserve unitarity at a loss of efficiency

(some numbers from CMS)

Dataset Tag	Total Events [1e6]	Total cpu s [1e9]	Total cpu s/evt	HS06
pythiaOnly	1724.69	7.36	4.27	89.59
pythiaOnly_PSW	293.47	3.15	10.74	225.45
pythiaOnly_noPSW	1431.22	4.21	2.94	61.73

*PSW : with PSWeights
*noPSW : without PSWeights

In the CMSSW environment (so includes the production of CMS data products), but indicative that PSW increases run time by a factor of 3-4 – but one run corresponds to many independent runs, so a net gain

Pythia8ConcurrentHadronizerFilter

- Pythia8 is ‘thread friendly’
  - Multiple instances of the Pythia class can be used concurrently on different threads

- Pythia8HadronizerFilter
  - Only handles one Event at a time
  - This is because the Decayers are all ‘thread-hostile’
    - E.g. using multiple instances of EvtGen running on different threads interfere with each other

- Pythia8ConcurrentHadronizerFilter
  - Creates a Pythia class instance per each cmsRun Stream
  - Does not contain any Decayer
  - Can be used wherever Pythia8HadronizerFilter was configured without a Decayer

Speed of ExternalGeneratorFilter

- I started from a simple configuration

```
process.generator = cms.EDFilter("Pythia8GeneratorFilter",
    comEnergy = cms.double(7000.),
    PythiaParameters = cms.PSet(
        pythia8_example02 = cms.vstring('HardQCD:all = on',
                                        'PhaseSpace:pTHatMin = 20.'),
        parameterSets = cms.vstring('pythia8_example02')
    )
```

- Ran jobs for 100,000 events
  - 1 thread and running the generator directly: 1815 seconds for job
  - 1 thread and use the ExternalGeneratorFilter: 3035 seconds for job (0.60x speed)
  - 4 threads and the ExternalGeneratorFilter: 903 seconds for job (2.0x speed)

- Serializing/deserializing data products takes considerable time

Takeaway Message

PYTHIA would benefit from HSF support on issues that increase the common good

Basic profiling for relevant use-cases would serve us well

VINCIA and are major updates: need to be test-driven at a larger scale by experiments to provide feedback (both physics and computing)