EvtGen report for LHCC review

Michal Kreps

HSF generator WG meeting, 6th May 2021

WARWICK

Team behind and funding

- Core team consists of John Back, Tom Latham, Fernando Abudinen and myself
- We have fraction of our time funded for work on EvtGen
  - Mostly UK STFC Consolidated grant
  - Part of the funding is attached to LHCb as part of our responsibility within experiment, part being for general maintenance
  - Fernando is currently funded by Warwick University (as part of collaboration with Monash)

- Other people contribute by writing new decay models driven by their needs
  - Mostly done in the context of measurements they are doing

What is EvtGen and its status

- EvtGen is package specialised for heavy flavour hadron decays
- It consists of about 130 decay models which implement specific dynamics of various decays
- Maintains detailed decay table with large number of explicit decays
  - Known decays do not add up to 100% BF, what is missing is filled up by generating quark configurations and passing those to Pythia8 for fragmentation
    - Amount passed to Pythia8 varies depending on particle with b-baryons relaying more on Pythia8 than others
- τ decays are done using Tauola
- Photos used for radiative corrections
- Code has been largely stable over past 10 years with most of the changes due to added models
- Some modernisation and cleanup was done rather recently

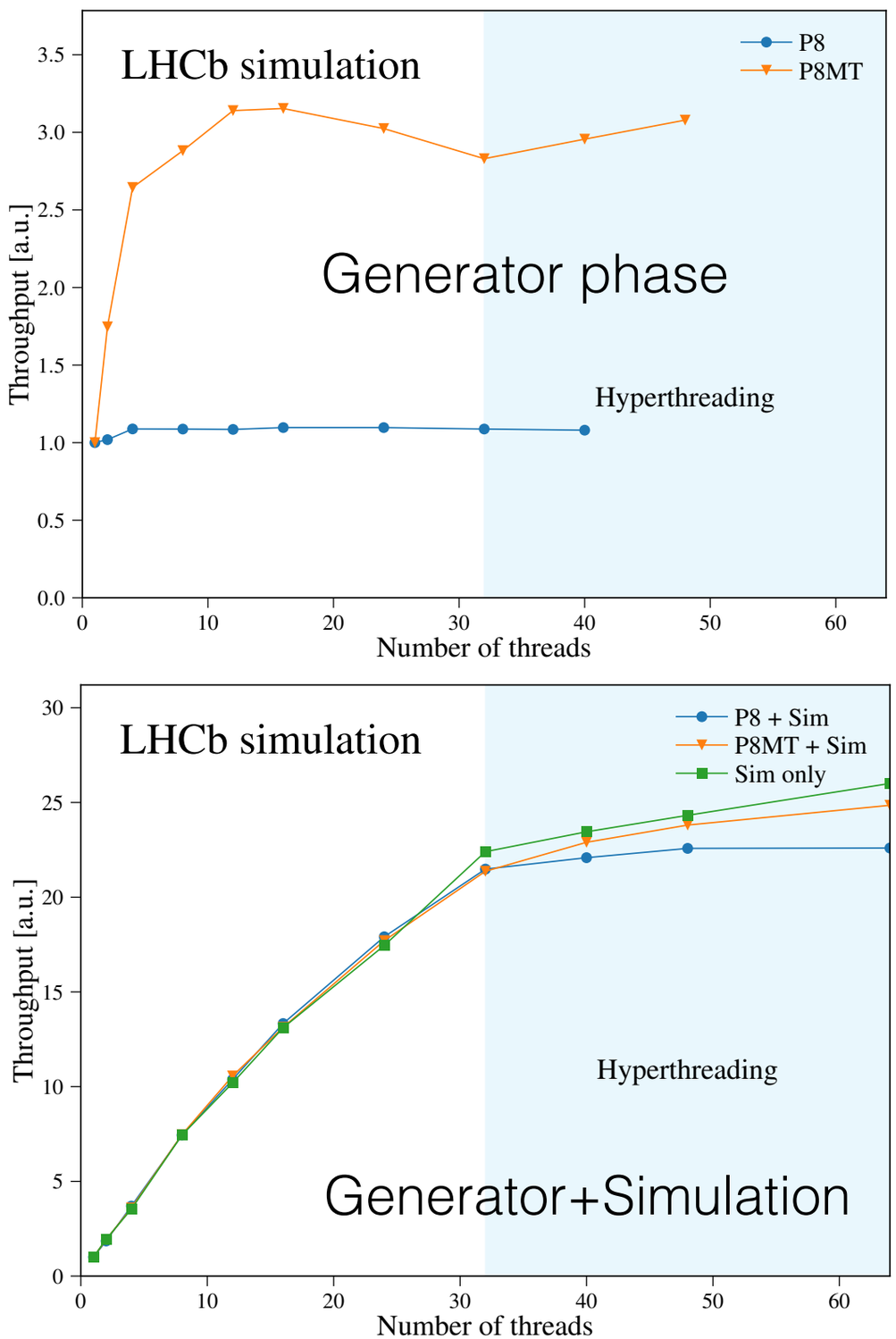
Plan for future developments

- Physics wise no plan for changes in near future
- Working on some code consolidation
  - Unify coding style, C++ modernisation
  - Plan to decrease code duplication within decay models
  - Improve/update documentation (Doxygen and paper/guide)
  - None of these is urgent and we are working through it slowly as time allows
- Plan to allow event level multithreading
  - Event here actually means particle to be decayed through full decay chain
  - Main blocking points here are Tauola and Photos, none of which allows multithreading
  - We have some funding for computing engineer to work on code redesign
  - Potentially not enough for full adaptation, core should be doable

Multithreading environment within LHCb

- New version of simulation under development uses single EvtGen instance which is locked as needed
- With only generator, there is limitation from few threads
- When detector simulation runs at the same time, there is no visible effect
- LHCb by default runs Generator+simulation in the same job
  - This is probably different to other experiments
  - There could still be limit with more threads

LHCB-FIGURE-2019-012

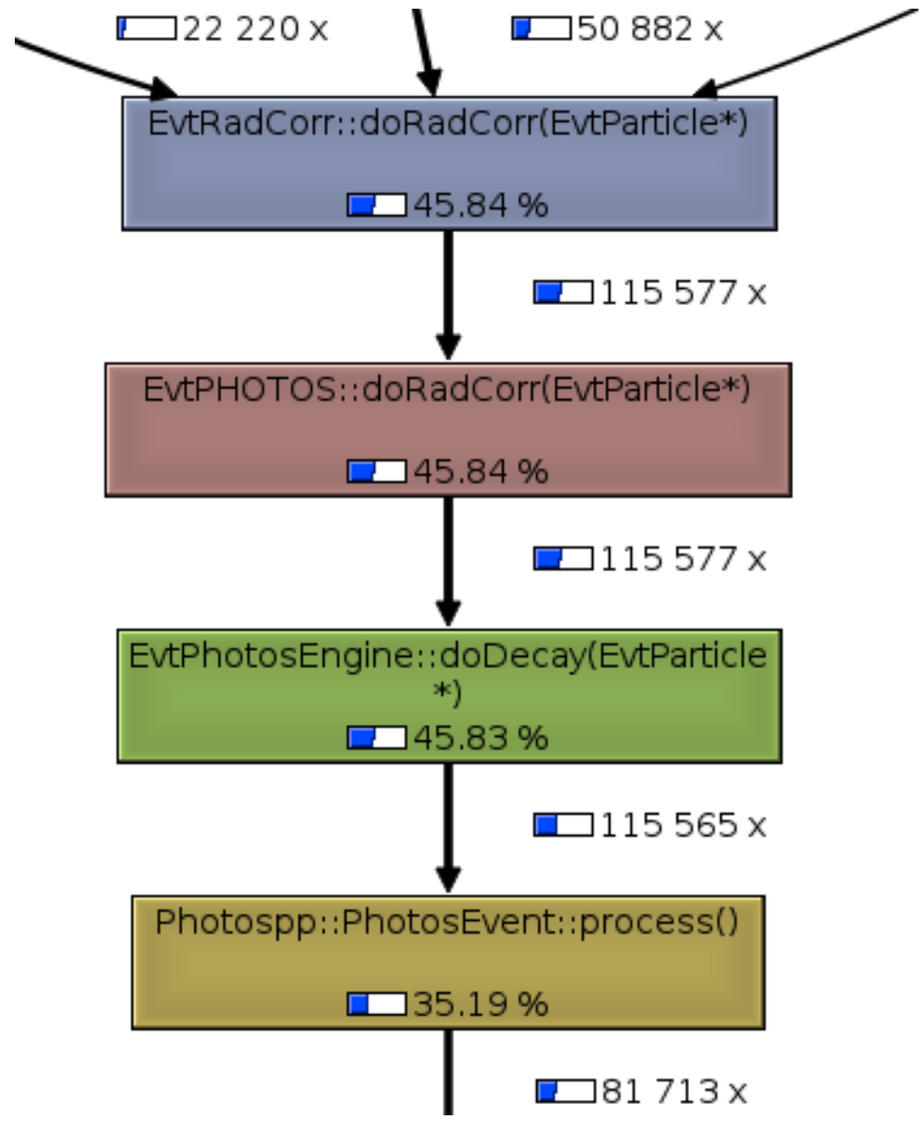


Issues with multithreading in EvtGen

- Structural limitations inside EvtGen for running multithreaded, e.g.
  - Global instance of random number generator
  - Global instance of particle properties and decay table (probably less of an issue
- Limitations from dependencies
  - Tauola
  - Photos
- Structural changes are fully in our hands, but require touching practically everything
- Dependencies more difficult as these are external
  - Replace Tauola with Pythia8
  - Look for alternative to Photos - discussing with Peter Skands and investigate SOPHTY

Issues with multithreading in EvtGen

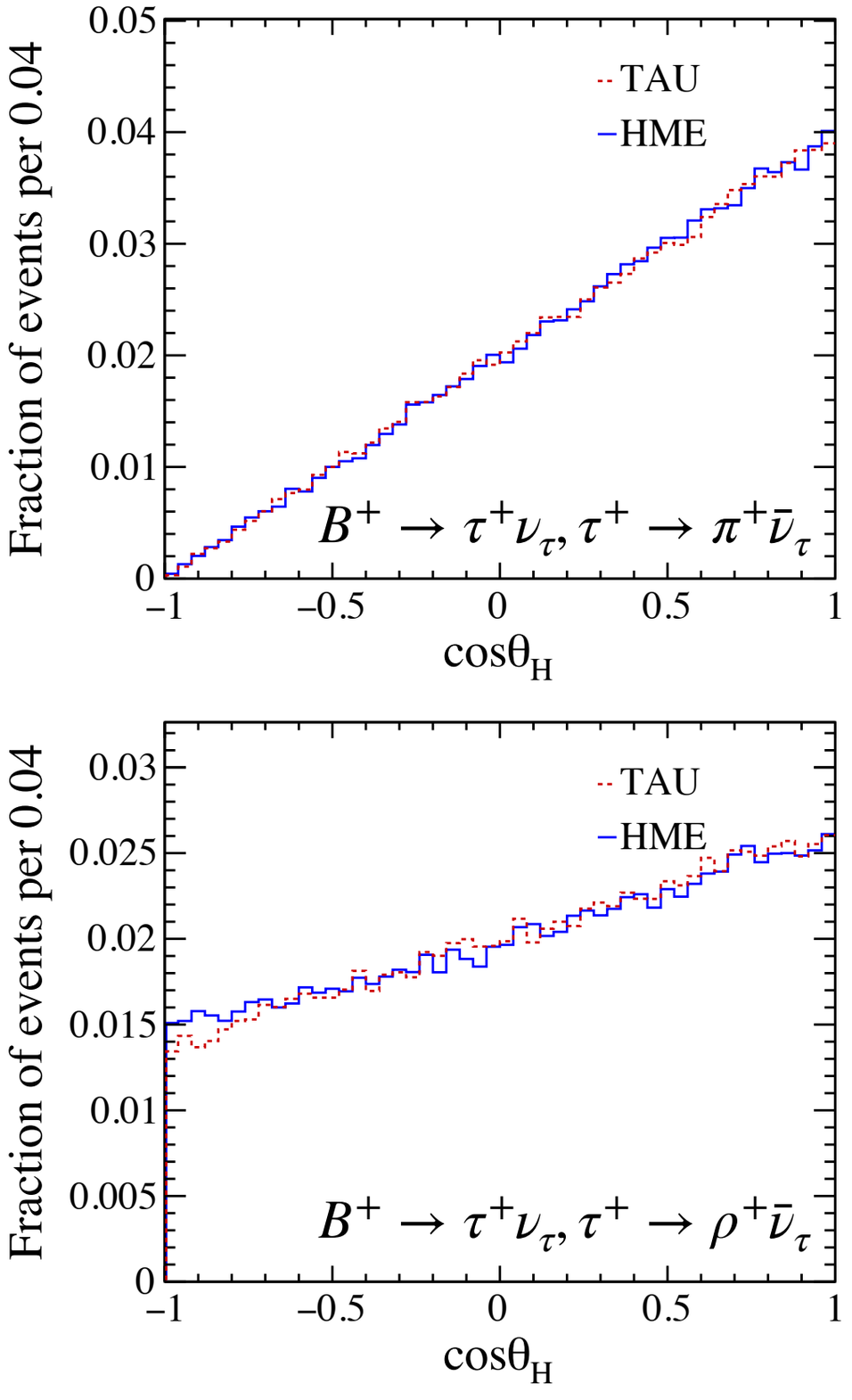
- Photos is used on practically every decay in usual use case
- From profiling, we spend significant amount of time in Photos itself
- What we see from EvtGen side is that conversion to HepMC and from HepMC is significant on EvtGen side
  - Similar conversion happens inside Photos
  - Overall, probably half of the time needed for radiative correction is event conversion
  - Need to try to bypass HepMC to see how much we can gain there



- Likely about 1/3 of the time in EvtGen is on radiative correction
- Locking Photos inside EvtGen unlikely to be satisfactory

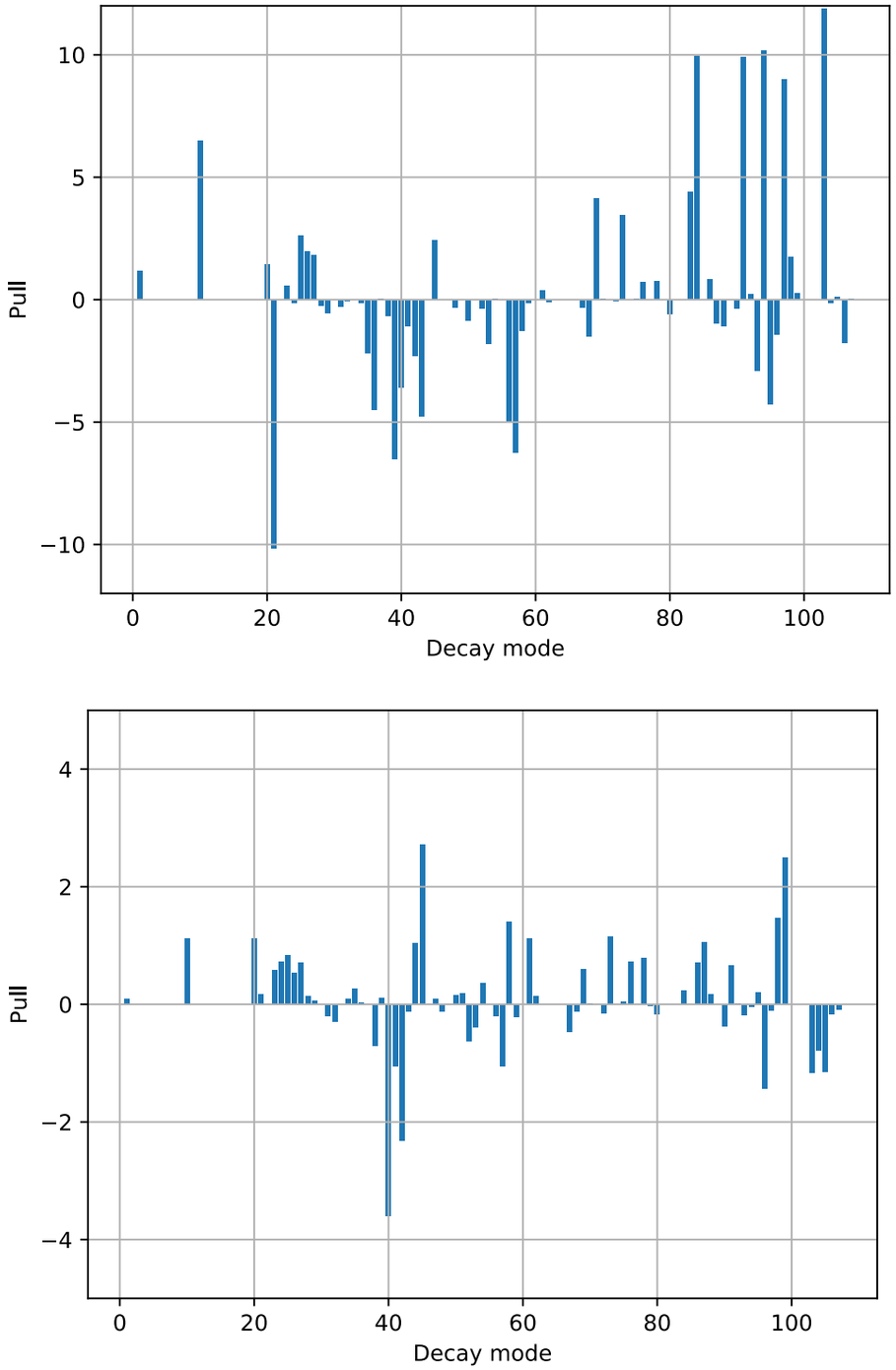
Tauola replacement

- Pythia8 provides τ decays using helicity formalisms
- EvtGen already depends on Pythia8 for other reasons, so try to use their functionality for τ decays
- Main interface is ready and we are just finalising minor details
- Unlike with Tauola, Pythia8 provides amplitude for particular spin state and sum over spin states done by EvtGen
- Almost done, but will need to iron out some details with Pythia8 authors (1 method needs to be made public)



Decay table

- Decay table is one thing which is difficult to update
- At some point we tried
  - Outcome had some mistakes as data are not always fully clear
- Attempt with D_s as testbed to update by hand to summary tables from RPP by minimising discrepancies
- Issue with unobserved explicitly listed decays as it is hard to track down original assumptions
  - Would require considerable amount of time to try to reconstruct assumptions
- Other issue is consistency of particle properties between generators



Decay channel in RPP 2018

Summary

- EvtGen package rather stable over years
- We hope to be able to make core thread safe within about year
  - Radiative correction where Photos is used is main concern
- Existing funding allows enough support to fix bugs and do necessary changes for newer OS, dependencies or compiler versions
  - Large developments would need dedicated funding