Modernizing and improving simulators for HEP – Pythia, EvtGen

Fernando Abudinén, John Back, Michal Kreps, Thomas Latham

MONASH University

Warwick RSE meeting
November 10, 2022

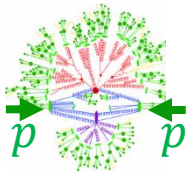
WARWICK
THE UNIVERSITY OF WARWICK

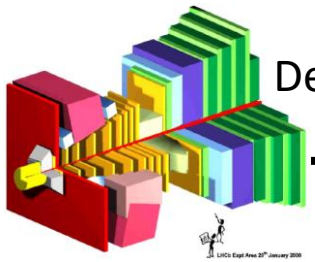
Overview

Simulation is essential since we interpret measurements by comparing simulation with collision data
⇒ Ideally, simulation should mirror data differing only by the knowledge of the “truth”.

Data

Collision





Detector

Electronics → Hardware trigger

Simplified conceptual sketch for LHCb Run I + II

Software trigger → Reconstruction → Stripping

Moore — Brunel — DaVinci

Turbo lines → Physics analysis

ROOT, Numpy, …

Simulation

Event generation → Interaction of particles with detector material → Detector response

Geant 4 — Boole

This talk

Collision and production → Decay

Pythia — EvtGen

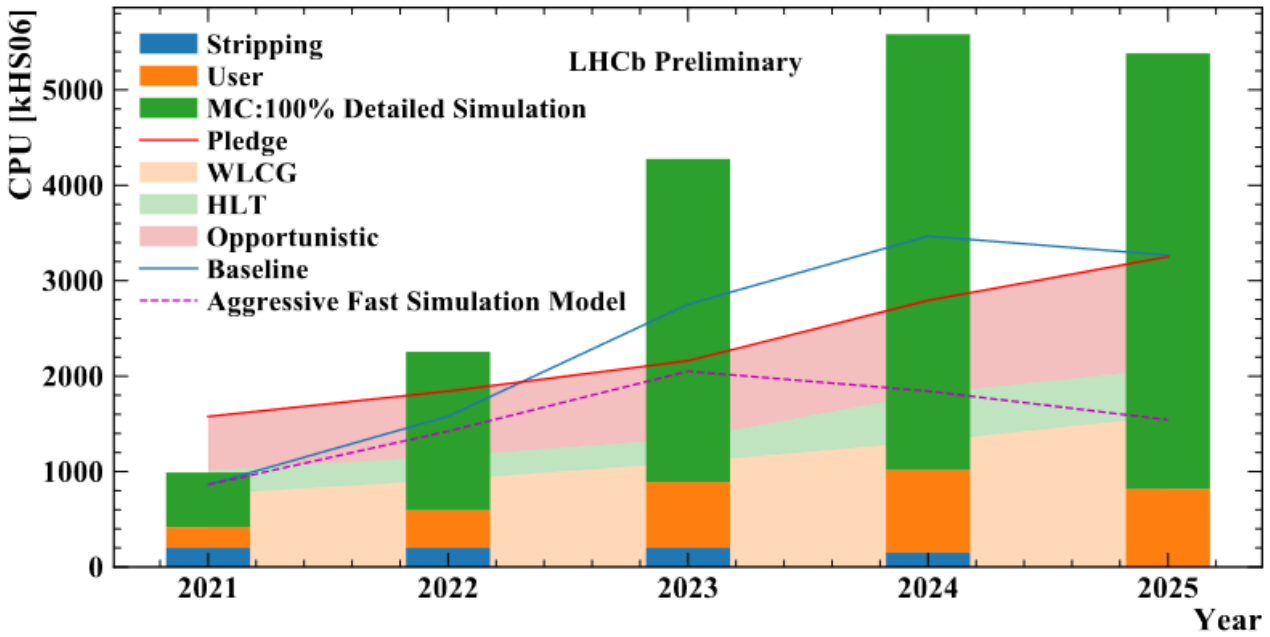
Gauss

Monte Carlo

Simulation is produced by a collection of specialized packages performing various assumptions and approximations depending on specifications

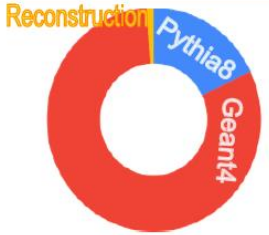
How much CPU does simulation consume?

LHCb-FIGURE-2019-018



Today

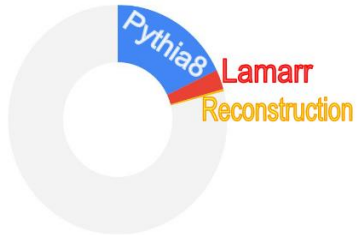
Detailed simulation: Pythia8 + Geant4
1M events @ 2.5 kHS06.s/event ≈ 80 HS06.y



Geant4: detailed simulation of interaction with detector material

Eventually in the future

Ultra-fast simulation: Pythia8 + Lamarr
1 M events @ 0.5 kHS06.s/event ≈ 15 HS06.y



Lamarr: simulation of detector response based on neural networks

- CPU consumption of generators (and analysis tools) not yet critical in general
- But developments in full sim (e.g. use of parametric sim) could soon change the picture
- Hardware developments (hyperthreading, GPUs, etc) call for modernization

Pythia

Ulrik Egede, Tom Hadavizadeh, Philip Ilten,
Minni Singla, Peter Skands

- General purpose generator for simulation of collision events of particles (electrons, protons, photons, heavy nuclei) at high-energies.
- Contains models for several aspects: hard/soft interactions, parton distributions, initial/final-state parton showers, multiparton interactions, fragmentation and decay.

Heavy (double heavy) hadron production occurs in only a (small) fraction of pp collisions
Simulation in Pythia is very inefficient (just keep trying until one is produced)
$\Rightarrow$ Currently working on different aspects aimed at making simulation of b-hadrons faster:

- B enhancement
- Forced hadronization/Doubly-heavy hadrons
- Optimize simulation for color-reconnection modes

MWAPP meeting
Tuesday 25 Oct 2022, 08:00 → 10:10 Europe/London

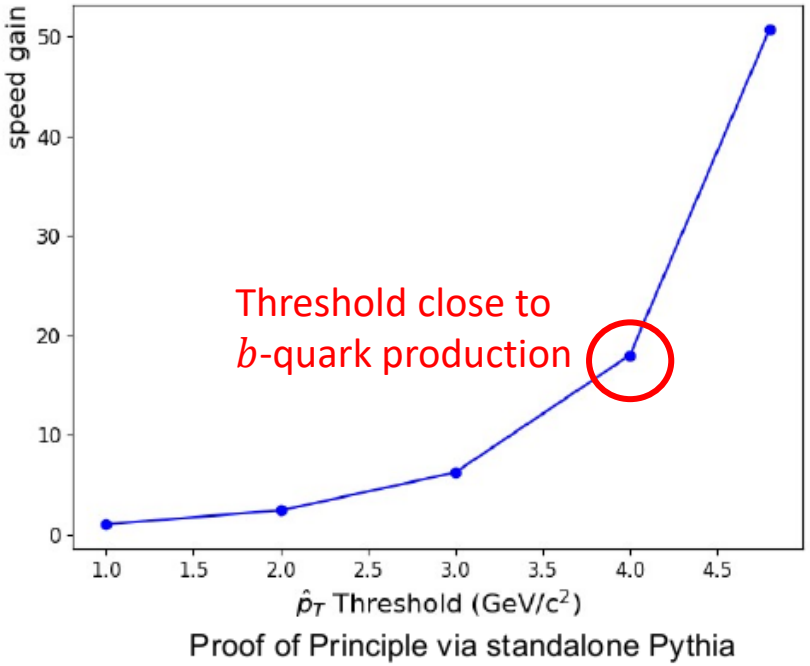
Fields with potential active collaboration with Monash

B enhancement

- Particularly when current generator exceeds CPU use wrt. detector simulation

$\Rightarrow$ Veto partons below p_{T} threshold

- Module made flexible for user to enhance $g \to c\bar{c}, b\bar{b}$ splitting
- Should be kinematically unbiased



Proof of Principle via standalone Pythia

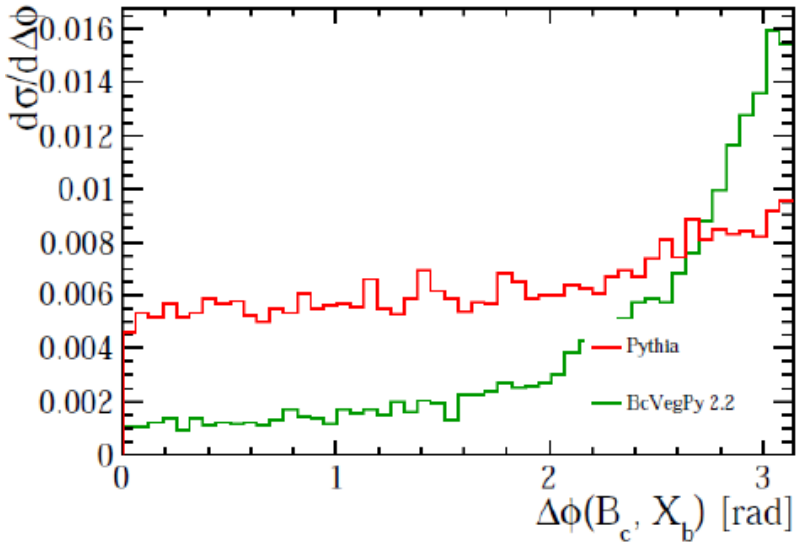
Double-heavy hadrons

- Production currently very inefficient in Pythia

$\Rightarrow$ Study vetoes based on presence of correctly colour-connected heavy quarks at early stages

- Currently comparing geometrical B_c distributions with dedicated generators like BcVegPy with limited list of supported production mechanisms

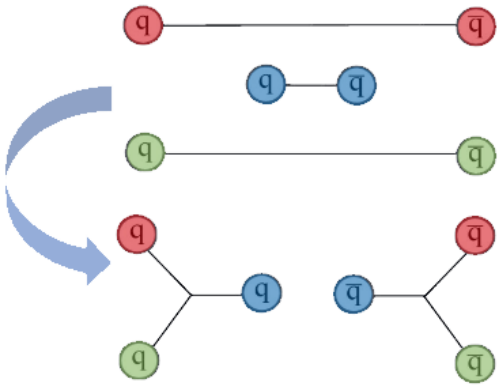
$\Delta\phi$: angle between B_c and $X_{b(c)}$ hadron on transverse plane



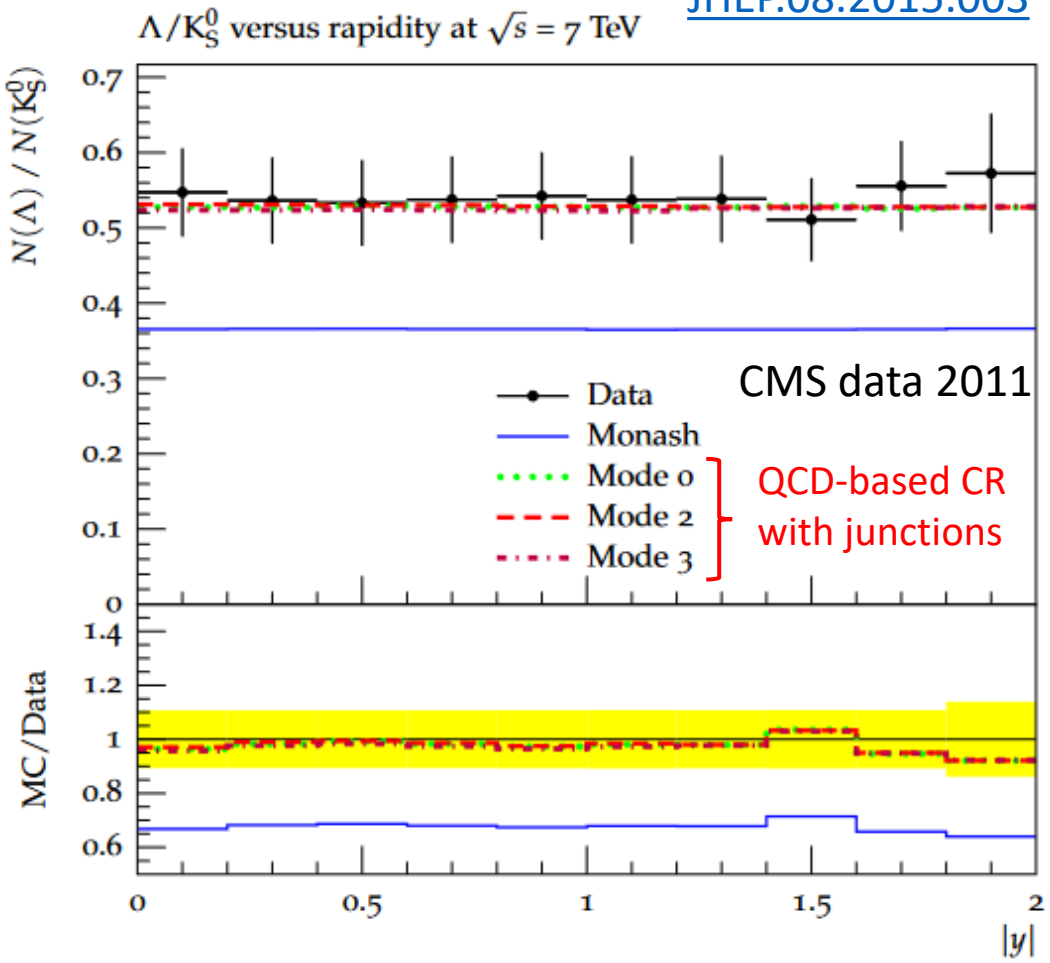
Improving color-reconnection models

- QCD-based color-reconnection models with junction agree well with collision data (without particular tuning)

CR junction: choose "shortest" string configuration



JHEP.08.2015.003



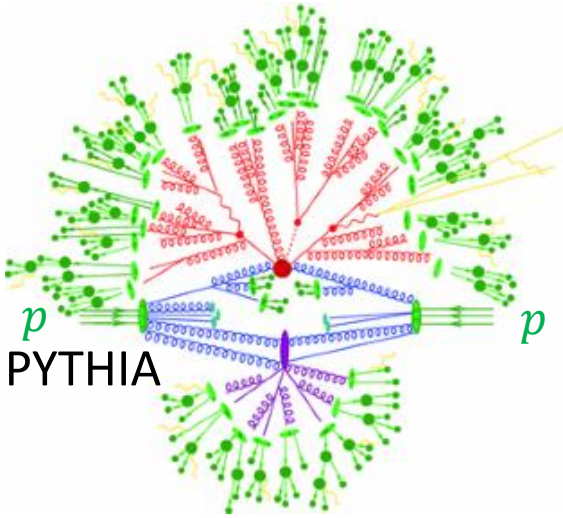
- However inefficient and CPU expensive
- Structural changes and efficient alternative (Hungarian algorithm) for minimization identified
- Need to be implemented and tested, but promising for enhancement of baryon production

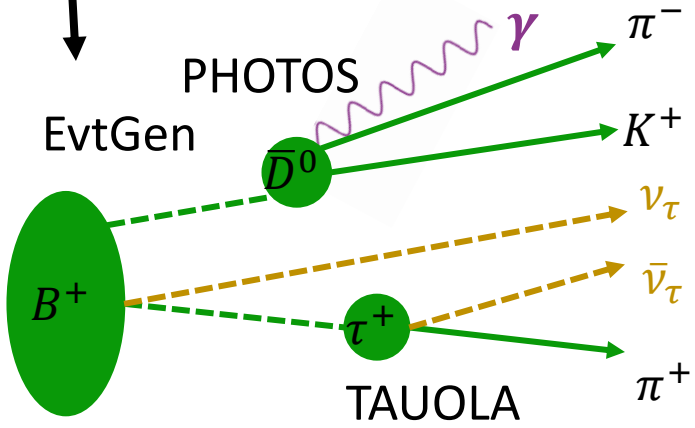
EvtGen

- Generator package specialised for heavy-flavour hadron decays
  - Used as well inside simulation of b jets
- Contains about 130 decay models implementing specific dynamics of various decays
- Maintains detailed decay table with large number of explicit decays
- τ decays simulated using TAUOLA
- PHOTOS used for simulation of final-state radiation

External

- Developed in the 90's, stable over past 10 years (changes mostly additions of new models)





Plans for EvtGen

- Physics wise no plan for changes in near future
- Currently working on code consolidation
  - Unify coding style, C++ modernisation
  - Plan to decrease code duplication within decay models
  - Improve/Update documentation (Doxygen and paper/guide)
  - Implemented common testing framework for validation
- Currently making EvtGen thread safe
  - Event ≙ particle whose decay is simulated (through full decay chain)
  - Main blockers are TAUOLA and PHOTOS, which are not yet thread safe
  - Work on source-code redesign currently ongoing with help of RSE engineers
  - Full adaptation is challenging (first core code adaptation almost finalized)

Heather Ratcliffe
Chris Brady

Progress on thread-safety

- Internal challenges: global instances (random number generator, particle properties and decay table)
- External challenges: limitations from dependences (TAUOLA, PHOTOS)

Set of solutions to reach thread-safety (preliminary):

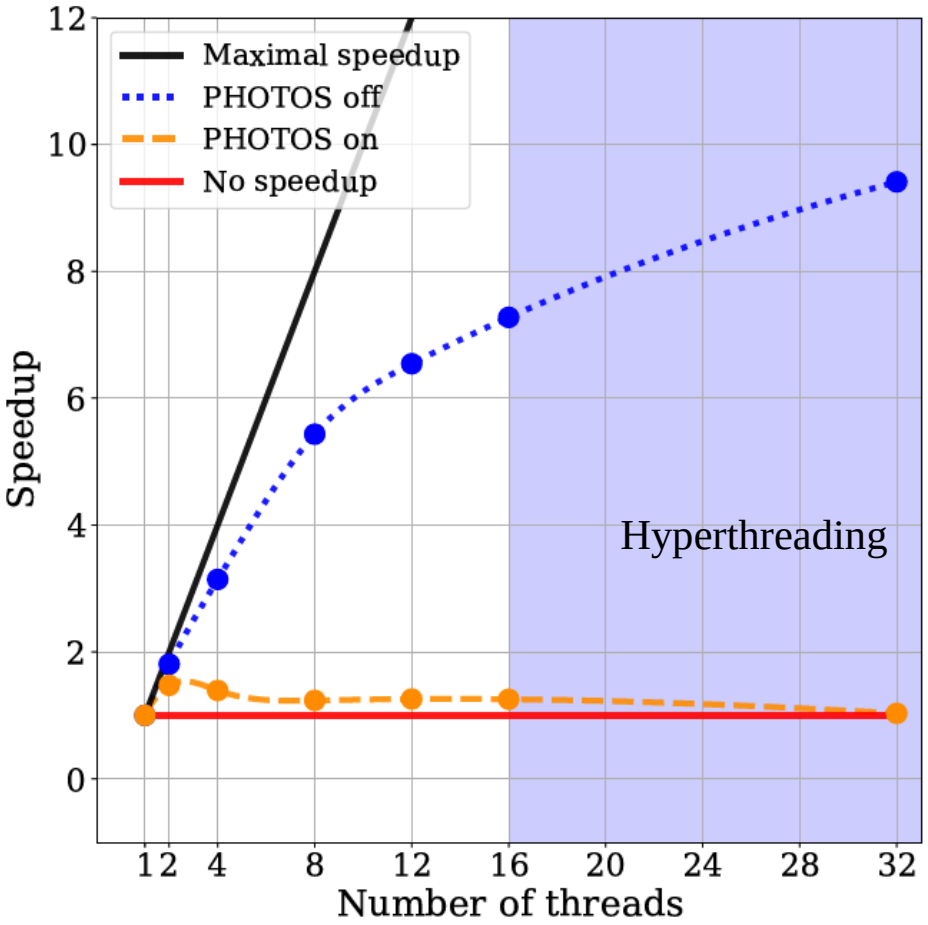
- Converted **static** objects to **static const** where possible
- Global singleton objects made thread-local
- Serialized (mutexed) calls to PHOTOS and TAUOLA

⇒ Deeper structural changes needed to fully exploit multi-threading (plan to continue working on it)

⇒ Current preliminary status reached thread-safety, passing physics tests for all decay models

⇒ But performance limited by external dependencies

⇒ Overcoming limitations from dependencies are more challenging as they are external (exploring alternatives)



Summary and outlook

Pythia:

- Working on enhancement of b-hadron production to make simulation faster
- Exploring how to improve efficiency for events with multiple heavy quarks

EvtGen:

- Currently making EvtGen threadsafe

$\Rightarrow$ Finalized common testing framework for validation

$\Rightarrow$ Converged on preliminary set of solutions to enable thread-safety of generator (exploitation of multi-threading will require further structural changes)

$\Rightarrow$ Performance limited by external dependencies (exploring alternatives)