Gaussino
a Gaudi-based core simulation framework

Dominik Muller
On behalf of the LHCb collaboration

CERN
CHEP 2019
Outline

Motivation

Gaussino
Generation
Simulation

Use in LHCb

All plots can be found in [LHCb-FIGURE-2019-012](LHCb-FIGURE-2019-012)
Motivation

Current simulation framework: Gauss
- ~15 years of age
- Purely single-threaded
- Memory usage becoming issue
Motivation

Current simulation framework: Gauss
- ~15 years of age
- Purely single-threaded
- Memory usage becoming issue

- General modernisation of LHCb software ongoing
- Idea: separate core functionality for simulation
Gaussino
Main features

  - Modular
  - Integrated generation and simulation phase
- Multi-threaded Gaudi
  - Easy Python configuration
- Multi-threaded Geant4
Execution structure

Gaudi functional

- Every algorithm a task
- Constant execution
  - Random engines created per call
- Fixed input/output

```cpp
/*output data*/ operator()(/*const input data*/) const
```
Execution structure

Gaudi functional

- Every algorithm a task
- Constant execution
  - Random engines created per call
- Fixed input/output

```cpp
/*output data*/ operator()/*const input data*/) const
```

Diagram:
- GenAlg → SimAlg
- SimAlg → GenMonitor → HepMCWriter
- SimMonitor → G4Hit Cnv → MCTruth → Output
Generation
Generation structure

- From Gauss
- Highly modular
Generation structure

- From Gauss
- Highly modular

See: NSS2010 - CD Conference Record, N42-284; LHCb-PROC-2010-056
Multithreading

- Thread-safety of generators
  - Production and decay tool
- Examples in Gaussino: Pythia8
  - Shared (locking) & thread-local (still locking memory allocation)
Multithreading

- Thread-safety of generators
  - Production and decay tool
- Examples in Gaussino: **Pythia8**
  - Shared (locking) & thread-local (still locking memory allocation)
Simulation
(with Geant4)
Simulation

- Modular!
- Configurable with python!
- Simulation service hiding backends
- Flexible configuration, e.g. different settings for
  - Pile-up - Spillover - Main event
  - Signal - other particles
  - For LHCb examples: A. Davis yesterday!
Geant4 interface
Geant4 interface

- **Gaudi-tools ➔ factories**
  - Easy python configuration of G4 settings
- **Dynamic assignment possible**
  - Entire Gaudi event
  - Split Gaudi Event into multiple G4 workloads
  - Plan to investigate new track based parallelism (in G4)
Truth tracking

- Consistent MC truth for particles from:
  - Generator level only
  - Generator level but modified by Geant4
  - Created in Geant4

\[
\begin{align*}
\text{omega}(782) \quad \text{HepMC}|\text{G4Primary}|\text{G4Truth} &= 1|0|0 \quad \text{CONV} = \text{MC} \\
|--- \quad \text{pi}\text{-} \quad \text{HepMC}|\text{G4Primary}|\text{G4Truth} &= 1|1|1 \quad \text{CONV} = \text{G4} \\
|--- \quad \text{pi}\text{+} \quad \text{HepMC}|\text{G4Primary}|\text{G4Truth} &= 1|1|1 \quad \text{CONV} = \text{G4} \\
|--- \quad \text{pi}\text{0} \quad \text{HepMC}|\text{G4Primary}|\text{G4Truth} &= 1|0|0 \quad \text{CONV} = \text{MC}
\end{align*}
\]

- Conversion to final event model
- Linking of hits to particles
Truth tracking

- Consistent MC truth for particles from:
  - Generator level only
  - Generator level but modified by Geant4
  - Created in Geant4

```
omega(782) HepMC|G4Primary|G4Truth = 1|0|0 CONV  = MC
|--- pi- HepMC|G4Primary|G4Truth = 1|1|1 CONV  = G4
|---|--- e- HepMC|G4Primary|G4Truth = 0|0|1 CONV  = FROMG4
|--- pi+ HepMC|G4Primary|G4Truth = 1|1|1 CONV  = G4
|--- pi0 HepMC|G4Primary|G4Truth = 1|0|0 CONV  = MC
```

- Conversion to final event model
- Linking of hits to particles
Truth tracking

- Consistent MC truth for particles from:
  - Generator level only
  - Generator level but modified by Geant4
  - Created in Geant4

\[
\text{omega(782)} \quad \text{HepMC|G4Primary|G4Truth} = 1|0|0 \quad \text{CONV} = \text{MC}
\]

\[
|--- \text{pi-} \quad \text{HepMC|G4Primary|G4Truth} = 1|1|1 \quad \text{CONV} = \text{G4}
\]

\[
|---|--- \text{e-} \quad \text{HepMC|G4Primary|G4Truth} = 0|0|1 \quad \text{CONV} = \text{FROMG4}
\]

\[
|--- \text{pi+} \quad \text{HepMC|G4Primary|G4Truth} = 1|1|1 \quad \text{CONV} = \text{G4}
\]

\[
|--- \text{pi0} \quad \text{HepMC|G4Primary|G4Truth} = 1|0|0 \quad \text{CONV} = \text{MC}
\]

\[
|---|--- \text{gamma} \quad \text{HepMC|G4Primary|G4Truth} = 1|1|1 \quad \text{CONV} = \text{G4}
\]

- Conversion to final event model
- Linking of hits to particles
Truth tracking

- Consistent MC truth for particles from:
  - Generator level only
  - Generator level but modified by Geant4
  - Created in Geant4

omega(782) HepMC|G4Primary|G4Truth = 1|0|0 CONV = MC
|--- pi- HepMC|G4Primary|G4Truth = 1|1|1 CONV = G4
|---|--- e- HepMC|G4Primary|G4Truth = 0|0|1 CONV = FROMG4
|--- pi+ HepMC|G4Primary|G4Truth = 1|1|1 CONV = G4
|--- p|0 HepMC|G4Primary|G4Truth = 1|0|0 CONV = MC
|---|--- gamma HepMC|G4Primary|G4Truth = 1|1|1 CONV = G4
|---|---|--- e+ HepMC|G4Primary|G4Truth = 0|0|1 CONV = FROMG4
|---|---|--- e- HepMC|G4Primary|G4Truth = 0|0|1 CONV = FROMG4
|---|--- gamma HepMC|G4Primary|G4Truth = 1|1|1 CONV = G4

- Conversion to final event model
- Linking of hits to particles
Detector geometry

- User must provide:
  - A Gaudi tool that provides the world pointer
  - Logical volume - sensitive detector mapping (python configurable)
- Basically python configurable G4 UserDetectorConstruction
- DD4hep prototype being tested
Integration in LHCb
A future Gauss prototype
Future Gauss prototype

- Built on Gaussino
- Migrated additional components:
  - Integrated EvtGen
  - LHCb geometry description (DD4hep in progress)
  - LHCb specific G4 components (sensitive detectors, physics, ...)
- Integrated with performance and regression tests
- Full stack studies and validation to begin
Future Gauss prototype

- Built on Gaussino
- Migrated additional components:
  - Integrated EvtGen
  - LHCb geometry description (DD4hep in progress)
  - LHCb specific G4 components (sensitive detectors, physics, ...)
- Integrated with performance and regression tests
- Full stack studies and validation to begin

Covers 90% of use-cases in LHCb!
Migrated Generation tools

- Integrated EvtGen (single, locking instance)
  - Handles all decays
- Logic for signal generation, e.g. specific D0 decay from minimum bias
- LHCb specific generator level cuts
Migrated Generation tools

- Integrated EvtGen (single, locking instance)
  - Handles all decays
- Logic for signal generation, e.g. specific D0 decay from minimum bias
- LHCb specific generator level cuts

Generation phase only, signal D0 decays from minimum bias, LHCb config
Validation studies

- Performed in LHCbPR (our performance and regression monitoring tool)

See: EPJ Web of Conferences 214, 02043 (2019)
LHCb detector simulation - Performance

- Tested with locking / thread-local generation phase
LHCb detector simulation - Performance

- Tested with locking / thread-local generation phase

Simulation with LHCb 2016 conditions, signal D0 decays from minimum bias
LHCb detector simulation - Performance

- Tested with locking / thread-local generation phase
- Overall good scaling in throughput
- Very good scaling in memory usage

Simulation with LHCb 2016 conditions, signal D0 decays from minimum bias
Conclusion

Showed core simulation framework Gaussino

Fully functional prototype of future LHCb Gauss framework built on it being validated