# Geant4 performance aspects in ALICE Run3

Sandro Wenzel (CERN), Ivana Hrivnacova (IJClab IN2P3/CNRS)

For the ALICE Collaboration

29th Geant4 Collaboration Meeting,Catania,
9 October 2024

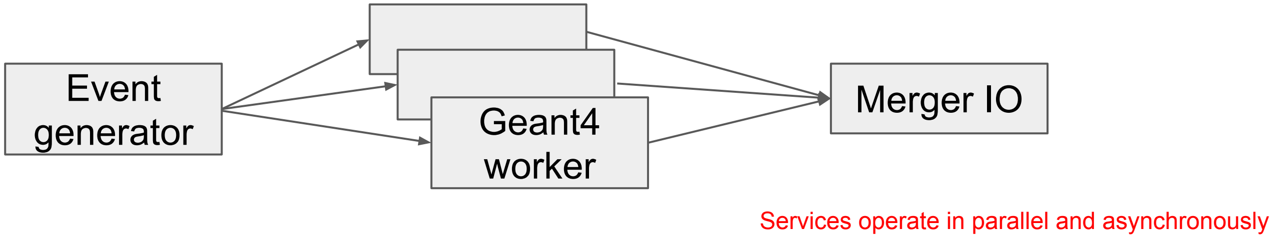
# Setting the scene

1. ALICE is using Virtual Monte Carlo (VMC) ecosystem for detector simulation
   - User code based on TGeo + VMC API (scoring); No direct coding against Geant4
   - Geant4 can be used as one transport backend (next to Geant3 and FLUKA)
2. Geant4 is the default simulation engine since LHC Run3
   - Previously, Geant3 was used
   - Started with **v11.0.4**, now with **v11.2.0**
   - These slides reports the updates since the ALICE presentation at the last Geant4 CM
3. Geant4 is using the TGeo navigator
   - Translation between TGeo geometry structure to G4 necessary (in-memory tables)
   - Initialization overhead + some small additional runtime cost
   - Currently no VecGeom usage possible since TGeo not yet able to use VecGeom

# Main Geant4 configuration choices

- NystromRK4 stepper
- Physics lists:
  - FTFP_BERT_EMV+optical
  - FTFP_BERT_EMV+optical+biasing (INCLXX physics in ITS region)
    - With PAI and SpecialUrbanMsc models in selected regions

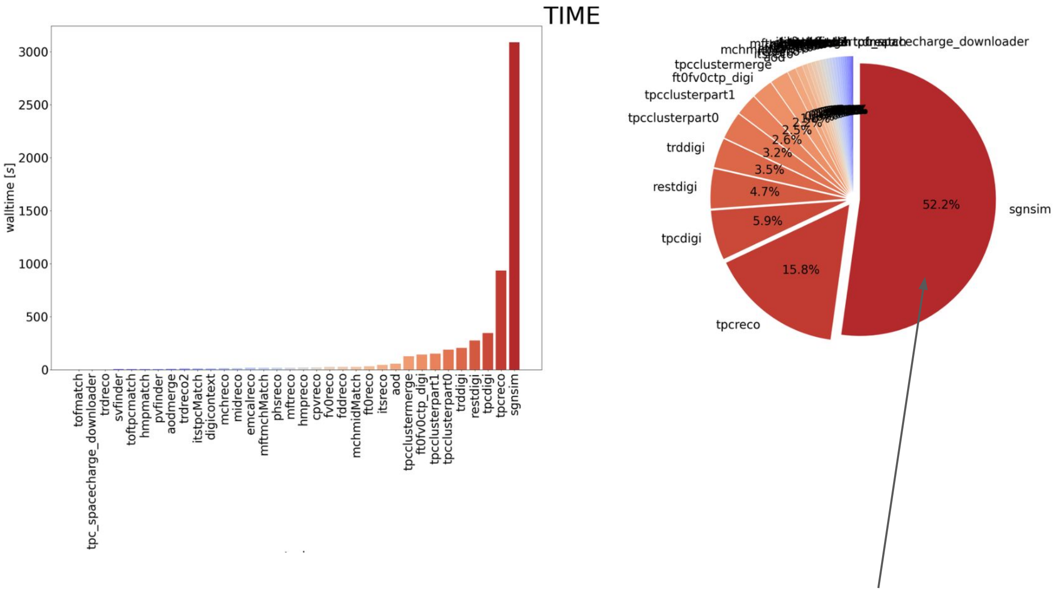
# Architectural overview of ALICE detector sim




- The o2-sim detector simulator is an executable that spawns various sub-processes as microservices that communicate via messaging
    - Event generator
    - Geant transport workers
    - Data merger for simulation output (final ROOT IO)
- Parallel Geant worker processes collaborate on simulation at the event level
    - Workers transport **sub-events** (or event chunks)
    - Processes are **memory shared due to late fork** (similar to multi-threading mode)
    - Parallelism works seamlessly also for FLUKA and Geant3

# Transport within the whole MC pipeline

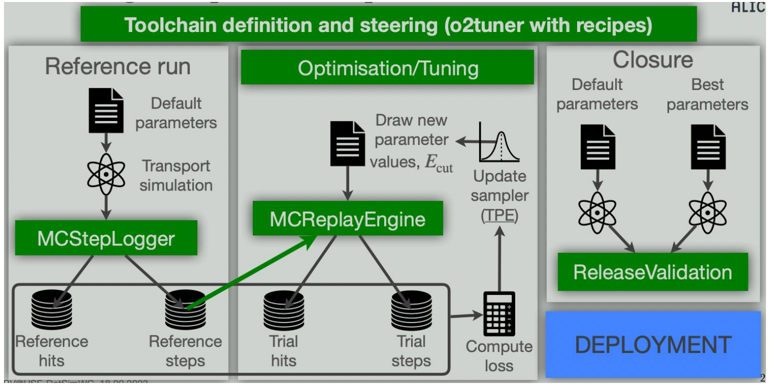
- Weight of Geant4 transport in the complete MC pipeline is very important
  - 40 - 55ish % depending on configuration
- But not the absolute hotspot. Other algorithms play major role as well, somehow limiting the global impact of improvements in transport

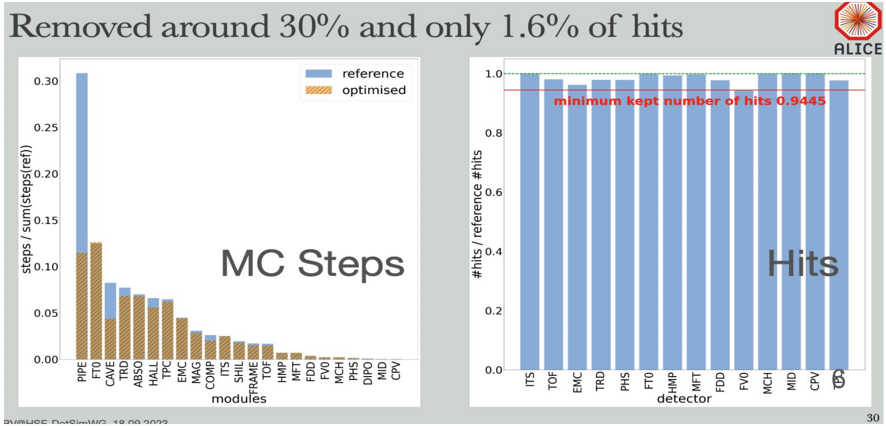



transport

# Optimizing MC steps (“process cuts”)

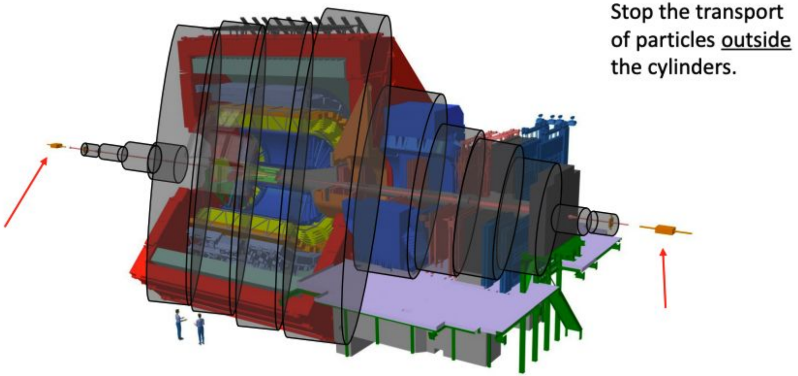
- An **optimization framework** developed which automatically tunes “production cuts” under the constraint to keep hit output constant
  - The talk at CHEP 2023 or [HSF meeting]
- **Achieved 30% reduction** in total steps by tuning electromagnetic production thresholds our PIPE geometry
- **Integrated in the O2DPG framework**
  - Via an external JSON file which is deployed in CCDB (ALICE calibration database) which will overwrite the values hardcoded in O2
- **Can be extended** to more material parameters



# Optimizing MC steps ("geometry region killer")

- Similar project on idea to absorb/kill tracks early once they exit a certain core part of the detector
- Last year summer student project to find optimal arrangement of "track-killing regions"
    - Or in other words a tightly fitting transport region
- Use of same auto-optimization framework to find optimal size of "bounding" cylinders
- Potential for another 10% reduction in steps
- Also **integrated in the O2DPG framework**

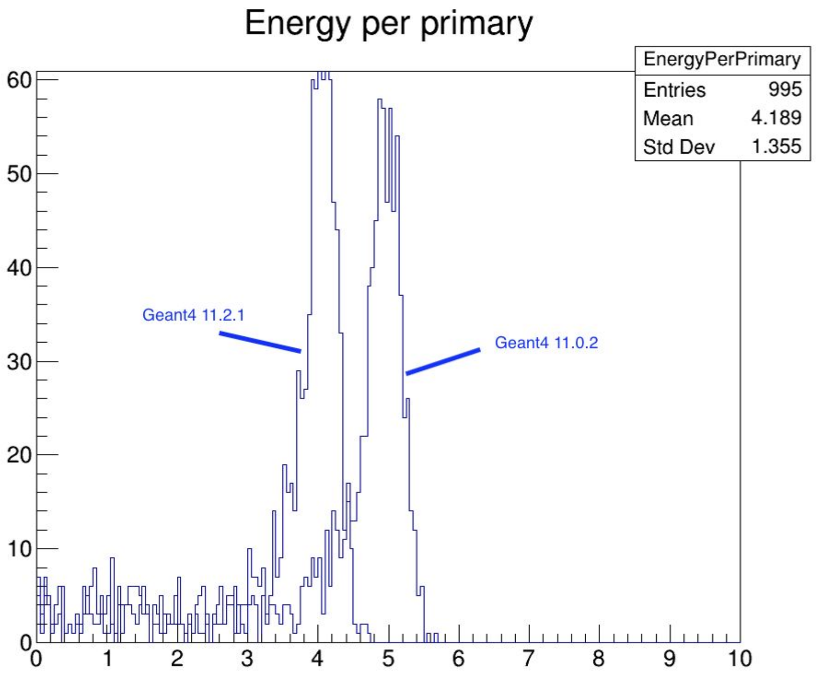


# Updates in Geant4 VMC Cuts

- The ALICE requirement to have a possibility to define thresholds in energy was addressed
  - Redundant double conversion Energy => Range => Energy
  - The final energy threshold does not always match exactly this initial value due to the approximations
- in Geant4 v11.2:
  - Added new function G4ProductionCutsTable::SetEnergyCutVector(…) and one more that allow to set productions cuts in energy and so to avoid double conversion energy - ranges - energy
- In Geant4 VMC v6.6:
  - A new method G4RangeManager::DefineRegions2() implemented
  - The inconsistencies due to double conversion are removed

# Issue With Geant4 11.2

- After switching to 11.2, a Pi0 mass shift was observed in the EMCAL detector, that could be reproduced with a simple macro
- The cause of this shift was identified in a **backward incompatibility** in newly introduced G4TransportationWithMsc process, activated by default in the EMV physics list option
    - When the new process is used, the setting of an extra EM model via G4EmConfigurator fails due to which **the SpecialUrbanMscModel tuned for ALICE was not taking into account**



# Issue With Geant4 11.2 - 2

- The first solution: switch off the new combined process with the UI command:
    - `/process/em/transportationWithMsc Disabled`
- The complete solution: Geant4 VMC code was adapted to support the explicit setting of an Em model to the G4TransportationWithMsc process, in v6.6.p2
    - Both options then produced the same result
    - From quick tests we did not see a noticeable speed-up with the new process

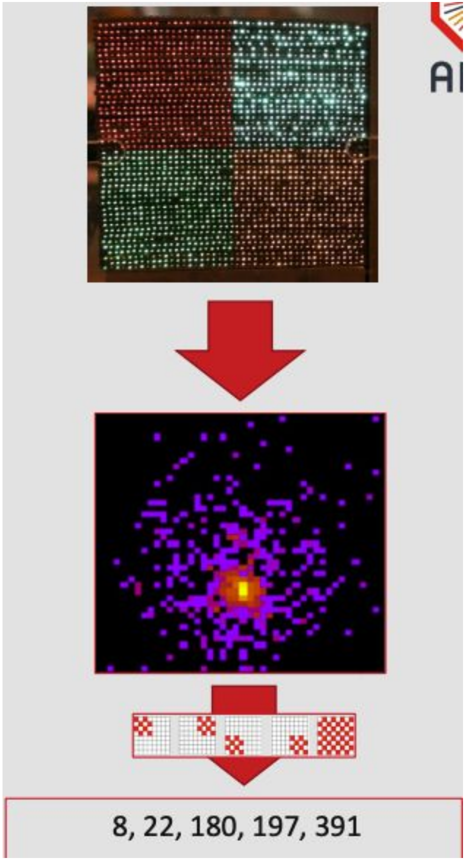
# More ideas that are on the list

- Magnetic field seems expensive for us, so tuning access to it is a natural idea
    - Avoid magnetic field calls whenever possible
        - For instance, check that all materials outside of field are marked as “non-field”
    - Play with parameters for caching etc.
- Stop tracks based on more deeply learned ML criteria
    - kill particles early based on particle properties (location, direction, material, …)
- VecGeom integration into TGeo or use of native Geant4 geometry

- *Still only a wish list due to lack of manpower*

# FastSim of ZDC

- ALICE is investigating also the use of ML techniques to replace full simulation in the ZDC (zero degree calorimeter)
- When ZDC is switched on, the Geant4 transport time approx doubles - triples due to showering
- Working on ML models that **avoid any transport to ZDC at all**
  - "Predict ZDC output = 2D image directly **based on primary particle properties"**
- Good recent progress with GAN models but more work towards production + validation needed
- Technique would allow to include ZDC without additional cost




ZDC has 2D arrangement of optical fiber tubes

Impeding particles induce photon showers which are essentially 2D images with color == number of photons

The idea is to generate these images with ML tools

Link to recent status update