



# FairRoot and $O^2$ Multi-threading Simulation

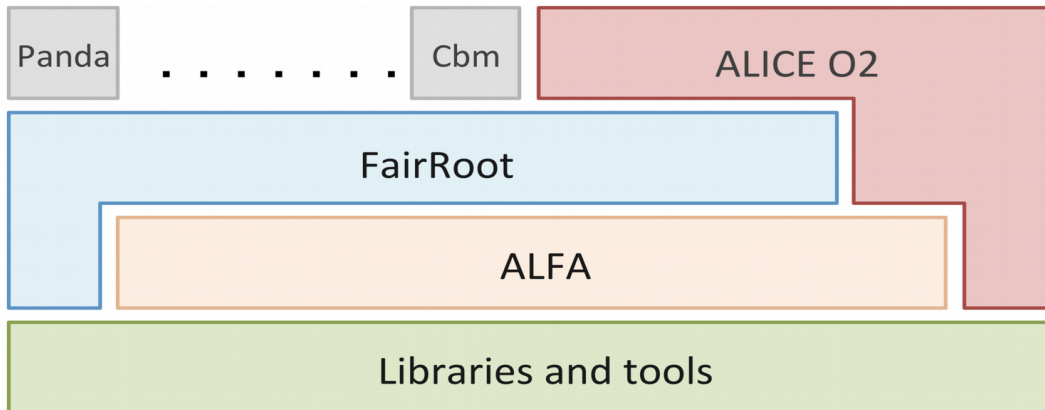
I. Hřivnáčová

Institut de Physique Nucléaire (IPNO), Université Paris-Sud,  
CNRS-IN2P3

23<sup>rd</sup> International Conference on Computing in High Energy and Nuclear Physics,  
July 9-13 2018, Sofia

# ALICE O<sup>2</sup> Software Architecture

- The new ALICE O2 software, besides general libraries and tools such as Boost , ROOT or CMake, is based on 2 other frameworks: ALFA and FairRoot.
  - FairRoot - standard simulation, reconstruction and data analysis framework for the FAIR experiments
  - ALFA – the new message queuing based framework is the result of a common effort of the ALICE and FAIR experiments

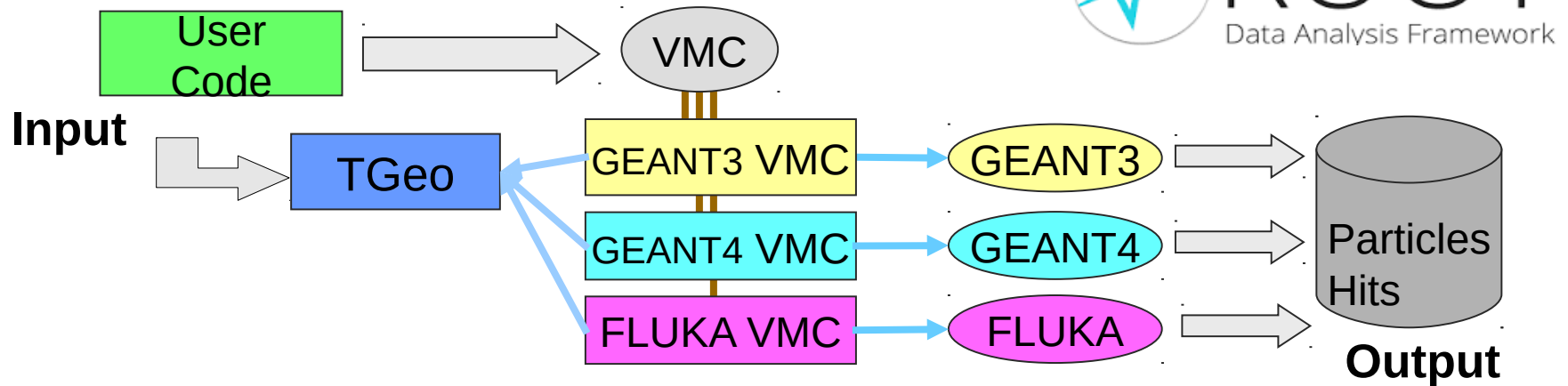


The multi-process approach combined with multi-threading within processes when required



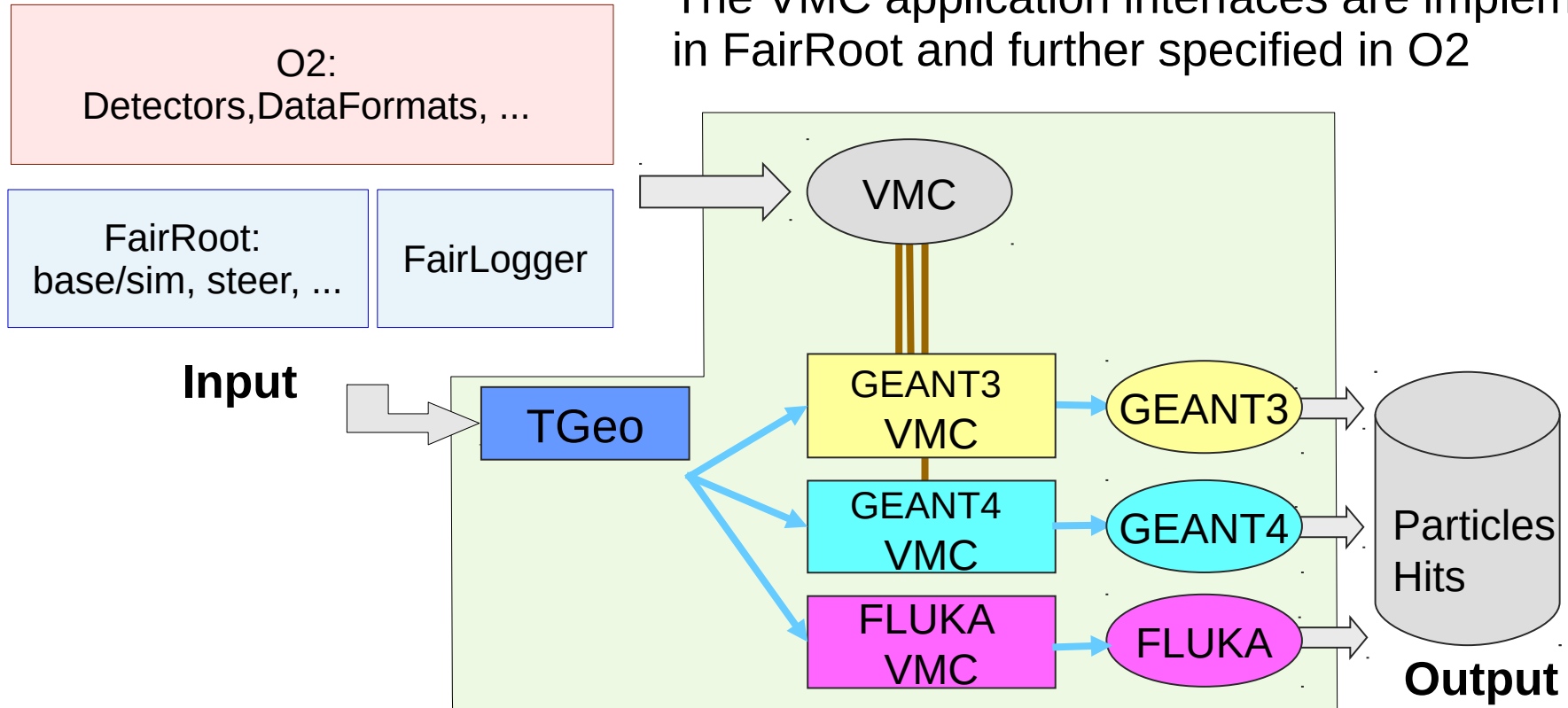
# Virtual Monte Carlo

- The O2 simulation framework is based on Virtual Monte Carlo (VMC)
- The VMC interface allows to run simulation with different Monte Carlo codes from the same user application
  - Developed by the ALICE Offline Project and further included in ROOT
  - <http://root.cern.ch/drupal/content/vmc>



# O2 + VMC

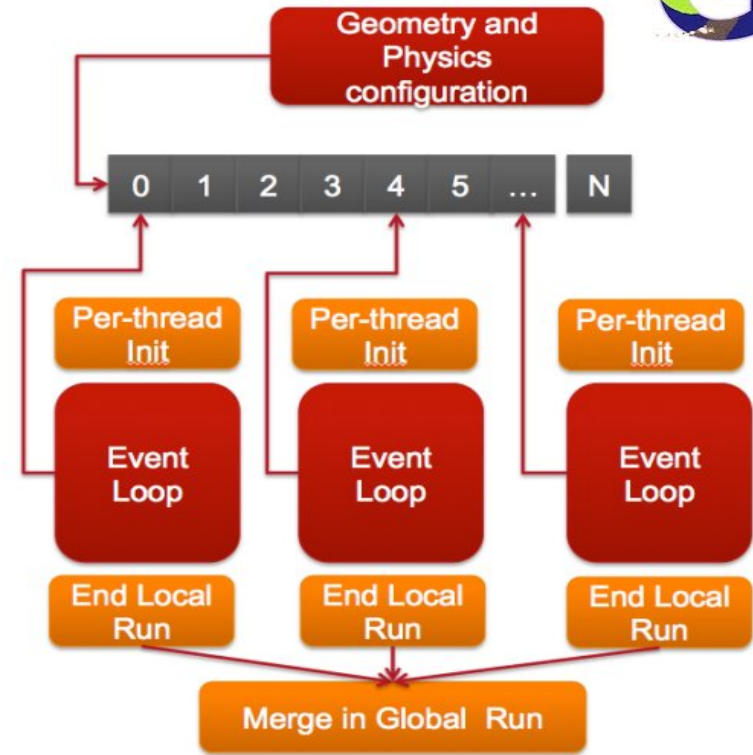
The VMC application interfaces are implemented in FairRoot and further specified in O2



# Multi-threading in Geant4



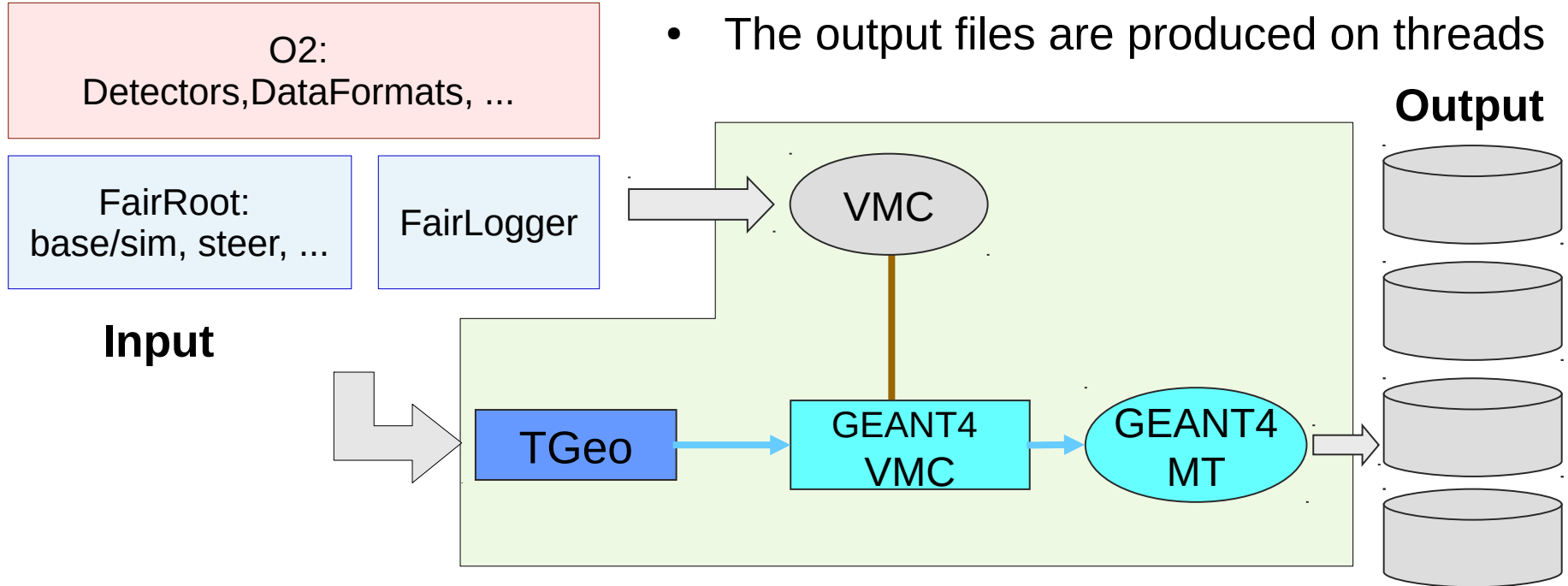
- Geant4 offers MT since version 10.0, released on December 6th, 2013
- Event level parallelism
- G4MTRunManager takes care of worker threads management
- Heterogeneous parallelism also possible and well tested
  - MPI (Message Passing Interface) works together with MT
  - Intel Thread Building Block (TBB): task based parallelism framework
- This release has triggered developments in the experiment frameworks towards MT



From Geant4 Toolkit Developers Guide

# O2 + Geant4 VMC MT

- Only Geant4 offers the multi-threading mode (the other MC options will not be discussed in this talk)
- Migration to MT concerned all base packages



- The output files are produced on threads

# Geant4 VMC MT

- The first Geant4 VMC MT version (3.0) was released in 2014
  - The developments in this version were presented at CHEP 2015 (Okinawa)
- Geant4 VMC code was adapted for multi-threading using the same approach as in Geant4 MT
  - Replacement of the singleton objects in Geant4 VMC with singletons per thread, including the main classes: TVirtualMC and TVirtualMCApplication
  - New VMC package MTRoot – implements the ROOT output per thread with locking critical ROOT operations
  - G4Root (TGeo Geant4 navigation packages) migration – by A. Gheata

# Migration of VMC Applications

- Users need to implement new functions of `TVirtualMCApplication` which are then used to clone the application and its containing objects on workers
- Creating of the objects on worker threads is then triggered from the Geant4 VMC classes
  - Examples and more detailed instructions are available from the VMC Web site

`class TVirtualMCApplication`

```
// required for running in MT                                CHEP'15, Okinawa
virtual TVirtualMCApplication* CloneForWorker() const;
// optional
virtual void InitForWorker() const;
virtual void BeginWorkerRun() const;
virtual void FinishWorkerRun() const;
virtual void Merge(TVirtualMCApplication* localMCApplication);
```



# Migration of VMC Applications (2)

- A new set of non-constant functions had to be added to `TVirtualMCApplication` for FairRoot migration
  - The old const functions are now deprecated and will be removed in the future

class TVirtualMCApplication, ROOT6.10/00

```
// optional  
virtual void InitOnWorker() {}  
virtual void BeginRunOnWorker() {}  
virtual void FinishRunOnWorker() {}
```

# Migration of FairRoot

- **FairMCApplication**
  - Implemented new TVirtualMCApplication functions added for multi-threading
  - Revisited the application initialization so that the objects relevant to event processing are created and initialized on workers
- Cloning the scoring objects on workers:
  - Added new virtual function in **FairModule** class which overriding is in MT mode obligatory:  
`FairModule* CloneModule() const;`

# Migration of FairRoot (2)

- Singleton classes used in simulation - FairRunSim, FairStack, FairRootManager, FairLogger, FairTrajFilter
  - Changed to singletons per threads
- FairRootManager
  - Implements the ROOT output per thread
  - Merged the MT features as in MTRoot
  - Most of locking could have been removed thanks to ROOT thread-safety improvements in ROOT 6

# FairRoot Examples

- Four examples in FairRoot/examples/simulation migrated to multi-threading
- A new option to select MT mode was added in the examples run macros
  - The option is then passed to `FairMCApplication` via `FairRunSim` singleton which holds the information about the MC setting selected by the user
- The default number of threads (2) can be changed in a configuration macro using the standard Geant4 command or the `G4FORCENUMBEROFTHREADS` environment variable

`run_tutorial1.C`

```
void run_tutorial1(Int_t nEvents = 10,
                  TString mcEngine = "TGeant3",
                  Bool_t isMT = false)
{
  ...
  FairRunSim* run = new FairRunSim();
  run->SetIsMT(isMT); // Multi-threading mode (Geant4 only)
  ...
}
```

# Migration of O2

- All O2 detectors and passive modules have been migrated:
  - PIPE, MAG, DIPO, ABSO
  - ITS, EMC, TRD, TOF, TPC, MCH, FIT
  - Implemented `CloneModule()` method
  - The initialization of sensitive volumes in detectors were revisited to get the necessary information available on threads
  - Fixed thread-safety issues
- Migration of the particles stack class, `o2::Data::Stack`
  - Enabled and implemented cloning the object on workers, fixed thread-safety issues
- Added an option `--isMT` to the O2 simulation application program, `o2sim`, which allows to activate MT mode

# O2 Tests

- The O2 build configuration files were updated to allow building a test with Geant4 libraries build in MT mode
- A new test `o2sim_G4_mt` has been added in the O2 standard test suites
  - The same simulation setup (detectors, primary event generator) as in `o2sim_G3` and `o2sim_G4` but with MT enabled
  - Keep Geant4 MT mode running with the future updates
- Packages versions in the O2 code integration tests:
  - ROOT 6.12/06, Geant4 10.3.p03, Geant4 VMC 3.5,
  - FairRoot: dev, O2: dev

# Plans

- Output buffering per thread like G4cout in Geant4
- Support for reading primary particles from a file
- A larger scale testing should follow to achieve the code robustness
- An integration with the new parallel processing framework
  - See S. Wenzel, Monday, 9 July 2018 12:15:  
A scalable and asynchronous detector simulation system based on ALFA

# Conclusions

- FairRoot and O2 framework classes were migrated to multi-threading simulation mode available via Geant4 VMC
- All FairRoot simulation examples and the O2 simulation program were enhanced with the multi-threading run option
- The MT mode was added in the standard O2 tests and is being added in FairRoot tests what should preserve this mode running in spite of intensive development in both frameworks