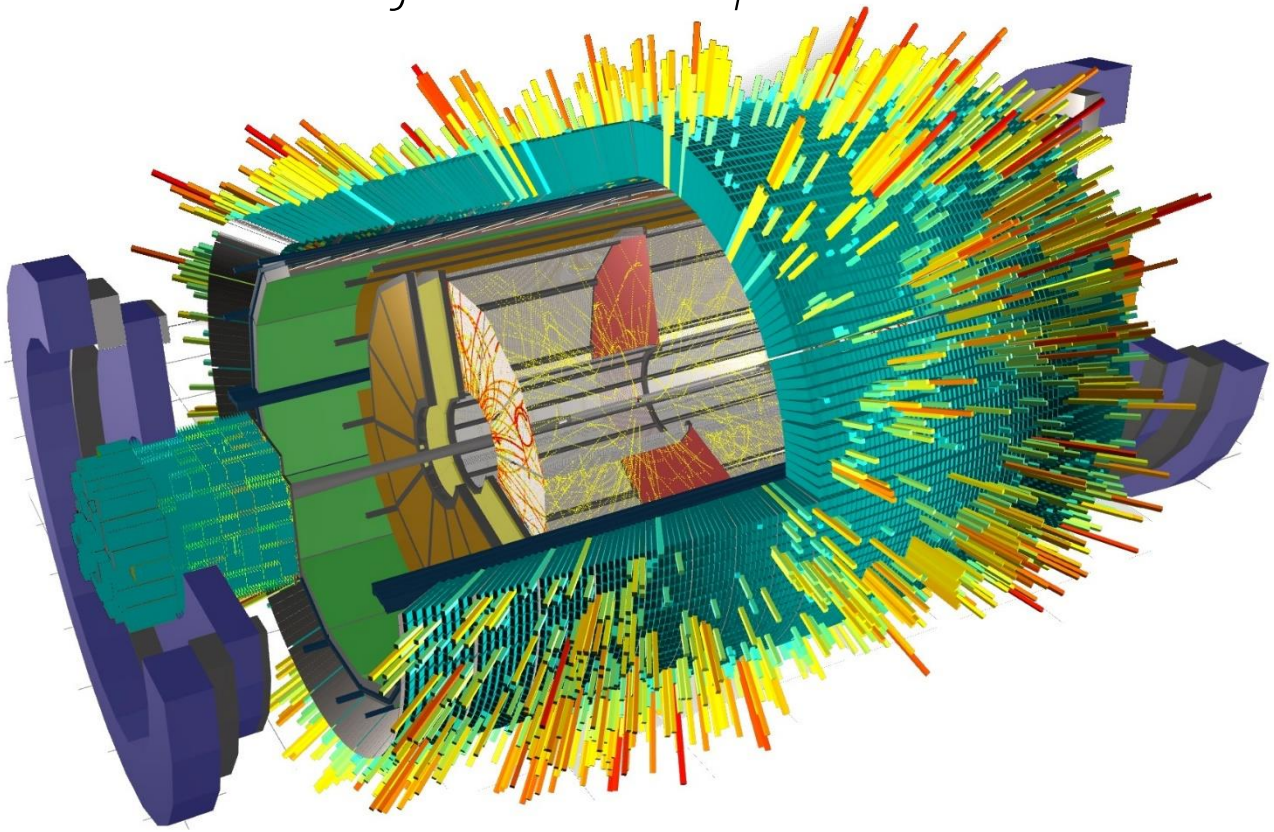
Integration of JupyterLab and ACTS Tracker into MPDRoot

HNATIC Slavomir

MPD Software Development Team



OUTLINE

- Quick Recap (April 2023 status)
- Software Development summary
- R & D → SOFTWARE
- QA ENGINE
- MPD LAB
- Rapid Development
- ACTS tracker integration
- Preliminary multimodule benchmarks
- Near future plans, perspectives

QUICK RECAP (April 2023)

NICADIST
- separate build system
- dependencies handling

CVMFS
- software distribution
- unified environment

Project Management & Support/User Interaction

GITLAB
- codebase
- CI
- testing

SUPPORT
- helpdesk
- telegram channel

WEBSITE
- howtos
- docs
- general info

R & D

MPDRoot

ANALYSIS | SIMULATION | RECONSTRUCTION

Mass Production

PWG REQUESTS HANDLING | DIRAC INTERWARE

Computing Infrastructure
(MICC & friends)
- supercomputer
- clusters
- storage systems

MPD assembly

TPC installation: March/May 2025

ONLINE EVENT DISPLAY
- experiment visualization
- slow control

DATA STORAGE & RETRIEVAL

DETECTOR CALIBRATION
- alignment
- noise level
- digitalization delay

RELEASES SINCE SUMMER 2023

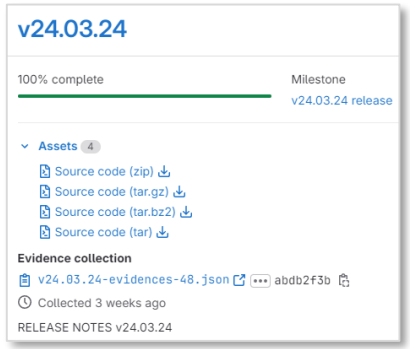
MOST IMPORTANT CHANGES

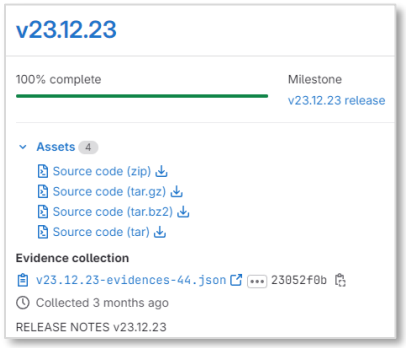
New features
- Analysis updates (physicists)
- ACTS tracker integration
- JupyterLab integration
- QA Engine
- AlmaLinux 9 default container

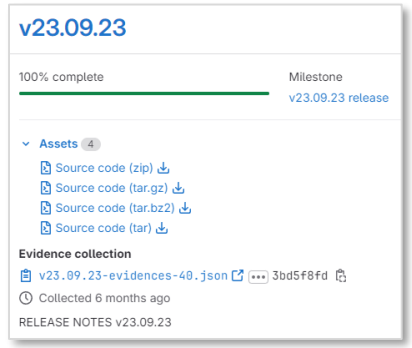
Latest dependencies
- ROOT …….. 6.28/12
- GCC ……..13.2.0
- Boost ……..1.83.0
- FairRoot … 18.6.10
- GEANT3 … 4.2
- GEANT4 … 11.1.3
- Python ….. 3.11.6
- Fedora 39, Debian 12

DETAILED INFO in RELEASE NOTES

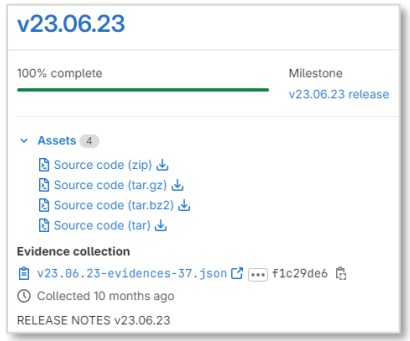
git.jinr.ru/nica/mpdroot/-/releases







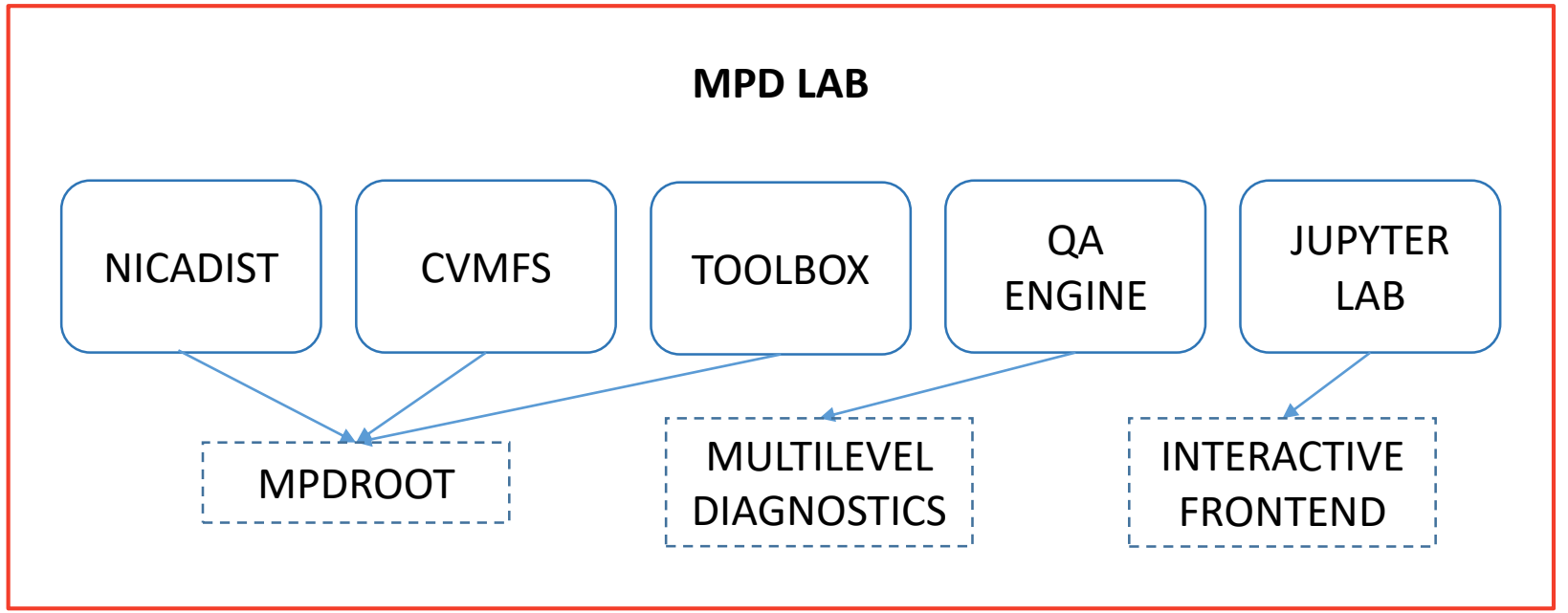
Preliminary summer 2024 release
(module add mpddev/v24.06.24-pre-1)
- ROOT ……. 6.30.06
- Python …. 3.12.2
- GEANT4 .. 11.2.1
- Pythia8 …. 8.3.11
- Pythia6 removed
- Last release with Centos7 support



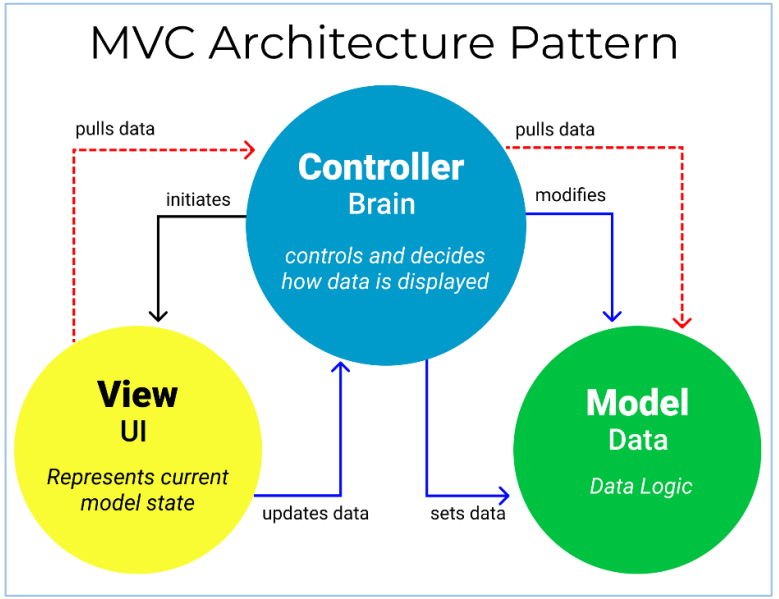
R&D → SOFTWARE

The need to have modern data analysis tool

- development **potential** (the variety of possibilities to innovate) directly depends on the properties of development environment
- developing/integrating the best of latest know-how & technologies for the needs of MPD experiment
- clarity, user friendliness, ability to learn on-the-fly



QA ENGINE



QA ENGINE PROPERTIES

- pluggable/switchable reconstruction modules
- QA modes to choose Diagnostics depth
- writing output in terms of MPD primitives into multiple structured root files for modular diagnostics and postprocessing

RUNRECO.C

Options:

tpcClustering = ETpcClustering::MLEM
= ETpcClustering::FAST
= ETpcClustering::WAVELET (future)

tpcTracking = ETpcTracking::DEFAULT
= ETpcTracking::ACTS

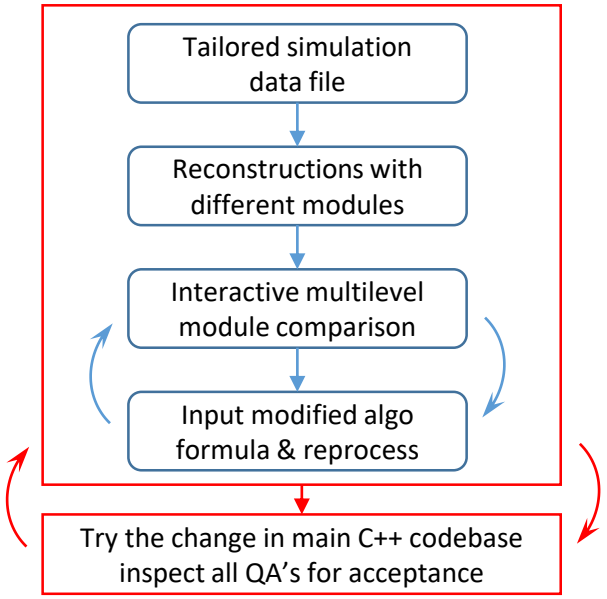
qaSetting = EQAMode::OFF
= EQAMode::BASIC
= EQAMode::TPCCLUSTERHITFINDER
= EQAMode::TRACKER (future)

Output example: BaseQA_Fast.root, QA_TpcClusterHitFinder_Fast.root
Settings: EQAMode::TPCCLUSTERHITFINDER, ETpcClustering::FAST

MPD LAB

QA / ATDD ENVIRONMENT

Interactive workflow example



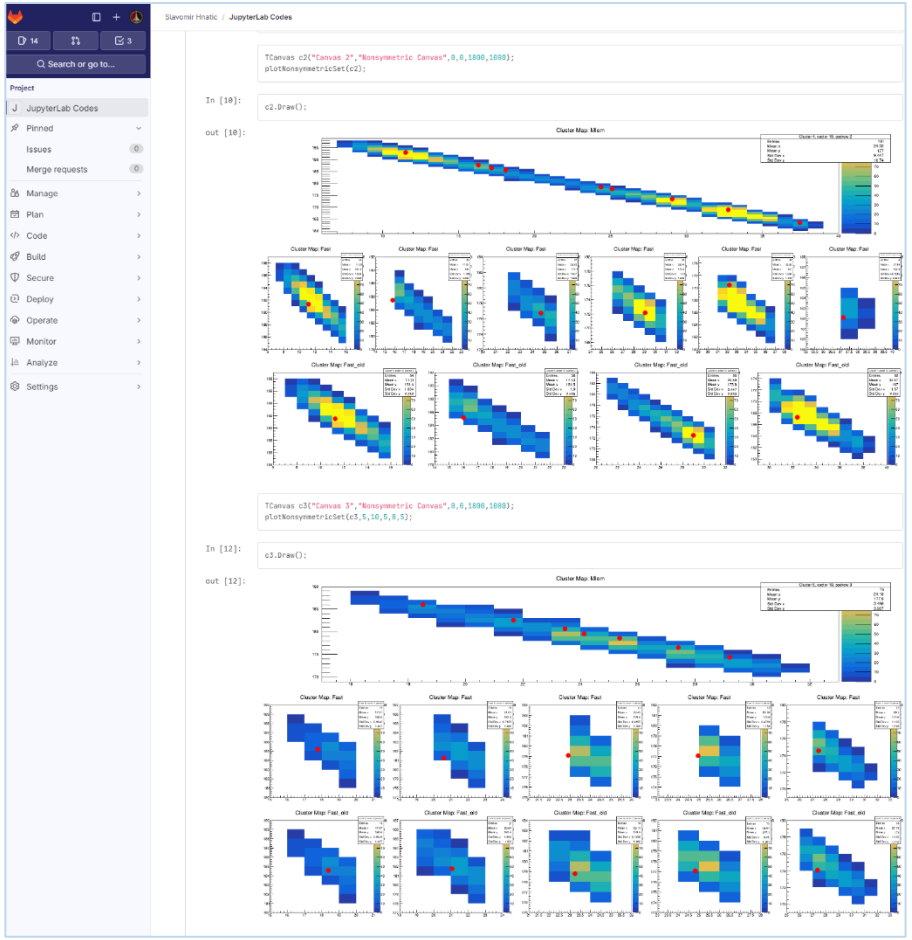
- Jupyter-Lab with JSRoot
- Custom code injection
- Cell structure with reprocess option
- Graphical output customized on demand
- Algo tuning to real experiment data

CLUSTERHITFINDER COMPARISON

- Mlem
- Fast

ABSTRACTION LEVELS

- Top ……..bench……..Reconstruction
- Middle…..component….ClusterHitFinder
- Bottom ……..units……..Clustering,
  Topology,
  Hits Extraction

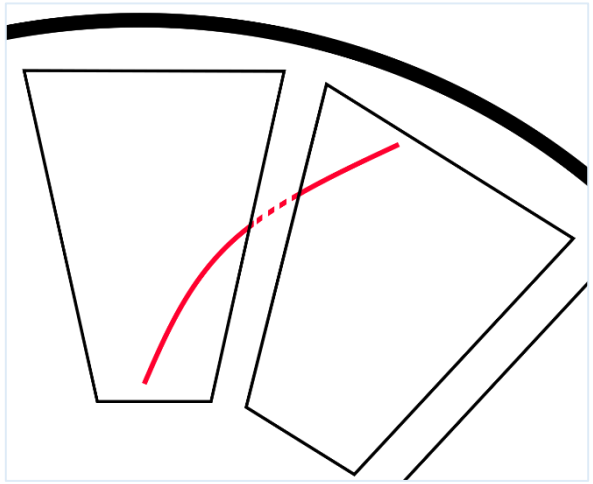


DIAGNOSTICS & RAPID DEVELOPMENT

EXAMPLE: DISCONNECTED TRACKS RETRIEVAL

MC trackID → TPC tracks

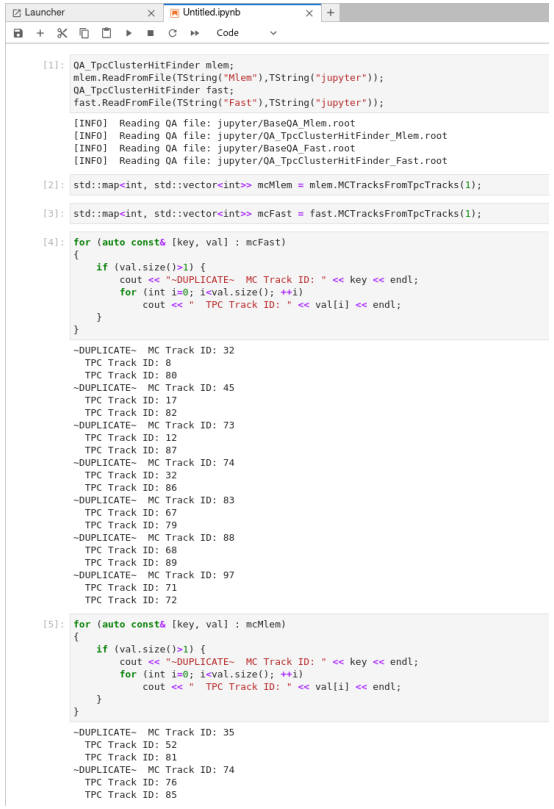
```
map <int, vector<int>> MCTracksFromTpcTracks(int event);
```



- to be then used to write, test and evaluate algorithm connecting disconnected tracks
- because of the considerable technical simplification, this work can be outsourced to juniors

RAPID DEVELOPMENT

- Prototyping method – 15 minutes
- Integrating properly into main codebase – several hours



Launcher | Untitled.ipynb | Code

```
[1]: QA_TpcClusterHitFinder mlem;
     mlem.ReadFromFile(TString("Mlem"),TString("jupyter"));
     QA_TpcClusterHitFinder fast;
     fast.ReadFromFile(TString("Fast"),TString("jupyter"));

[INFO]  Reading QA file: jupyter/BaseQA_Mlem.root
[INFO]  Reading QA file: jupyter/QA_TpcClusterHitFinder_Mlem.root
[INFO]  Reading QA file: jupyter/BaseQA_Fast.root
[INFO]  Reading QA file: jupyter/QA_TpcClusterHitFinder_Fast.root

[2]: std::map<int, std::vector<int>> mcMlem = mlem.MCTracksFromTpcTracks(1);

[3]: std::map<int, std::vector<int>> mcFast = fast.MCTracksFromTpcTracks(1);

[4]: for (auto const& [key, val] : mcFast)
     {
         if (val.size()>1) {
             cout << "~DUPLICATE~  MC Track ID: " << key << endl;
             for (int i=0; i<val.size(); ++i)
                 cout << "  TPC Track ID: " << val[i] << endl;
         }
     }

~DUPLICATE~  MC Track ID: 32
  TPC Track ID: 8
  TPC Track ID: 80
~DUPLICATE~  MC Track ID: 45
  TPC Track ID: 17
  TPC Track ID: 82
~DUPLICATE~  MC Track ID: 73
  TPC Track ID: 12
  TPC Track ID: 87
~DUPLICATE~  MC Track ID: 74
  TPC Track ID: 32
  TPC Track ID: 86
~DUPLICATE~  MC Track ID: 83
  TPC Track ID: 67
  TPC Track ID: 79
~DUPLICATE~  MC Track ID: 88
  TPC Track ID: 68
  TPC Track ID: 89
~DUPLICATE~  MC Track ID: 97
  TPC Track ID: 71
  TPC Track ID: 72

[5]: for (auto const& [key, val] : mcMlem)
     {
         if (val.size()>1) {
             cout << "~DUPLICATE~  MC Track ID: " << key << endl;
             for (int i=0; i<val.size(); ++i)
                 cout << "  TPC Track ID: " << val[i] << endl;
         }
     }

~DUPLICATE~  MC Track ID: 35
  TPC Track ID: 52
  TPC Track ID: 81
~DUPLICATE~  MC Track ID: 74
  TPC Track ID: 76
  TPC Track ID: 85
```

ACTS TRACKER INTEGRATION

INITIAL VERSION
(Authors: A.Kamkin, P. Belecky)

- based on ACTS v20.1.0 (Sept. 2022)
- Centos 7

Virtual geometry

Input Hits

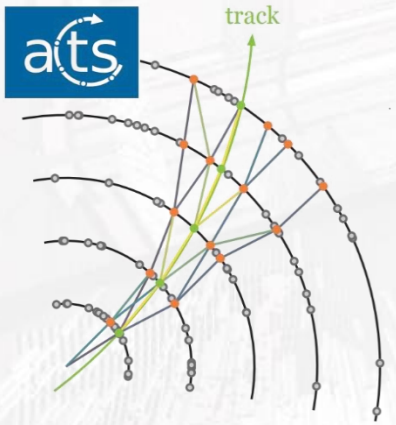
Projection

Seeding

Input KF parameters

Track finding

Selector



INTEGRATION
(Authors: S.Hnatic, J. Busa)

- ACTS v33.0.0 (03.2024)
- AlmaLinux 9
- VM with ACTS alibuild (debug)
- adapted to each new major ACTS release

How to run:

toolbox enter a9-nica-dev

module add mpddev ACTS/v33.0.0-1

build mpdroot's dev branch

runReco.C with ETpcTracking::ACTS

PERFORMANCE COMPARISON

LOCAL MACHINE

Au-Au 7GeV, 500 events (AMD 5825U)

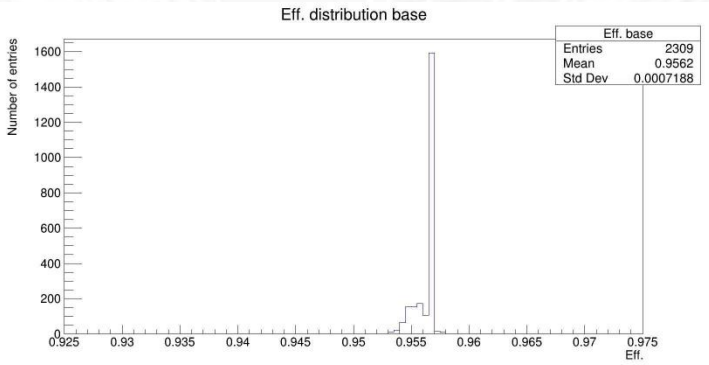
Module Combination	Efficiency	Speed
MLEM + DEFAULT KALMAN	96.53 % (62903 / 65162)	12.51 s / event
MLEM + ACTS	96.47 % (62854 / 65162)	16.67 s / event
FAST + DEFAULT KALMAN	95.52 % (62245 / 65162)	7.91 s / event
FAST + ACTS	95.81 % (62432 / 65162)	10.60 s / event

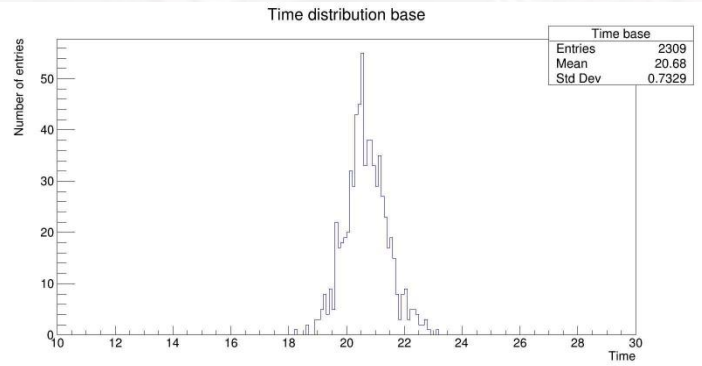
MLEM + DEFAULT KALMAN … tuned
FAST + ACTS …….. no tuning

MASS PRODUCTION

Bi-Bi 9.2GeV, > million events (Govorun)

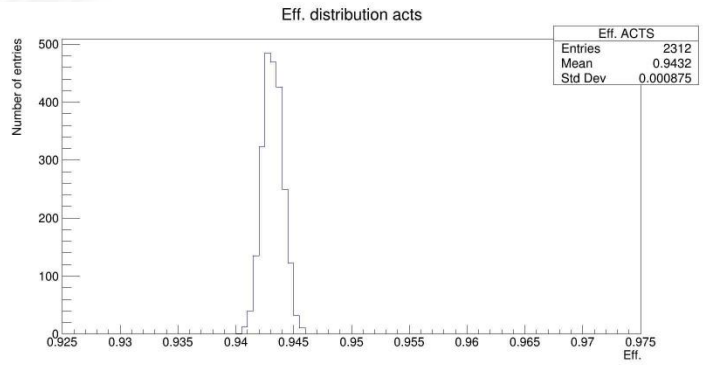
MLEM + DEFAULT KALMAN

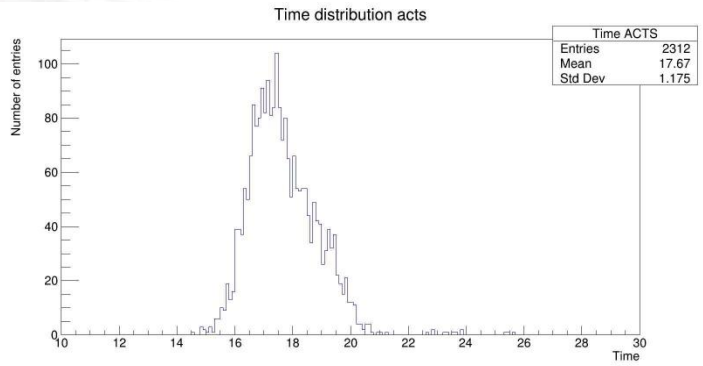




1.3% more reconstructed tracks

FAST + ACTS





17% faster reconstruction

PERFORMANCE COMPARISON

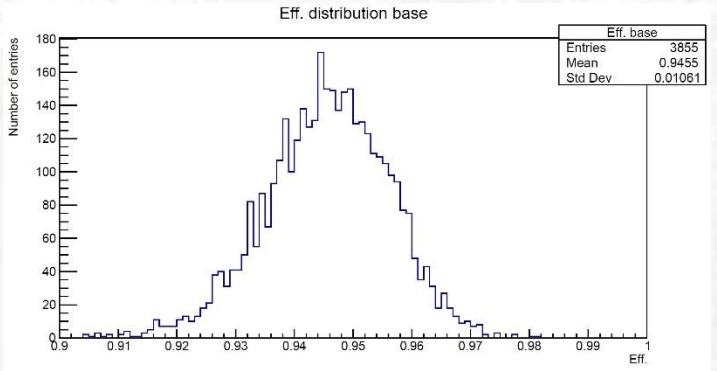
- comparable results
- full potential of new algorithms to be utilized in the future as:

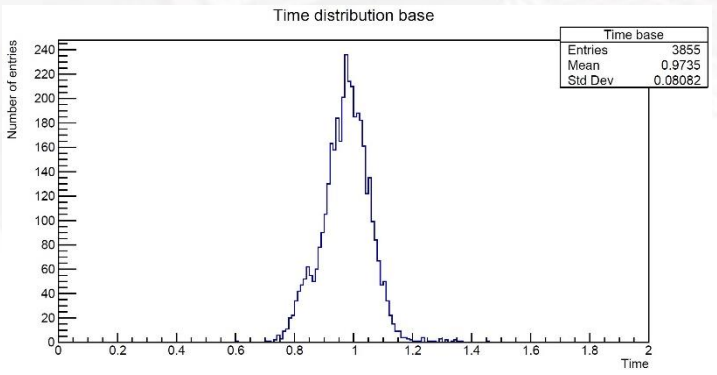
ACTS tracker & FAST clustering are YET TO BE TUNED

MASS PRODUCTION

p-p 9.2GeV, > million events (Govorun)

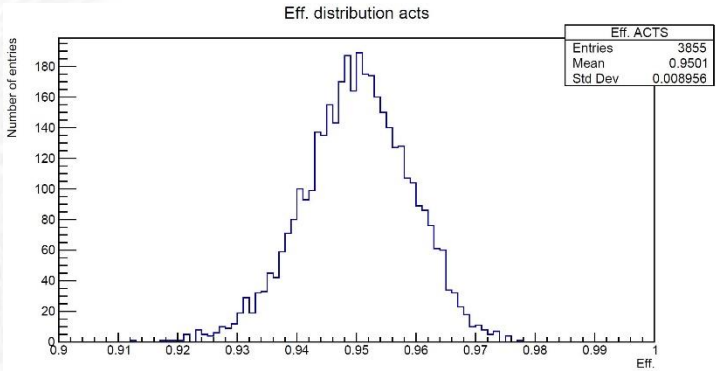
MLEM + DEFAULT KALMAN

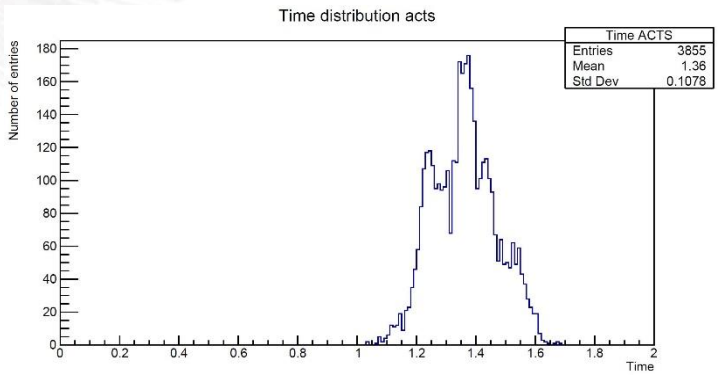




0.46% less reconstructed tracks

FAST + ACTS





t = 1.12 s (geometry correction)
15% slower reconstruction

PERSPECTIVES, FUTURE PLAN

MULTI-MODULE ANALYSIS

CLUSTERING

- Improve accuracy level of Fast algorithm (edge cases, benchmarking)
- Integration of Wavelet algorithm

TRACKING

- Optimize virtual geometry
- Fine tune Acts tracker configuration (speed, efficiency)
- Disconnected tracks

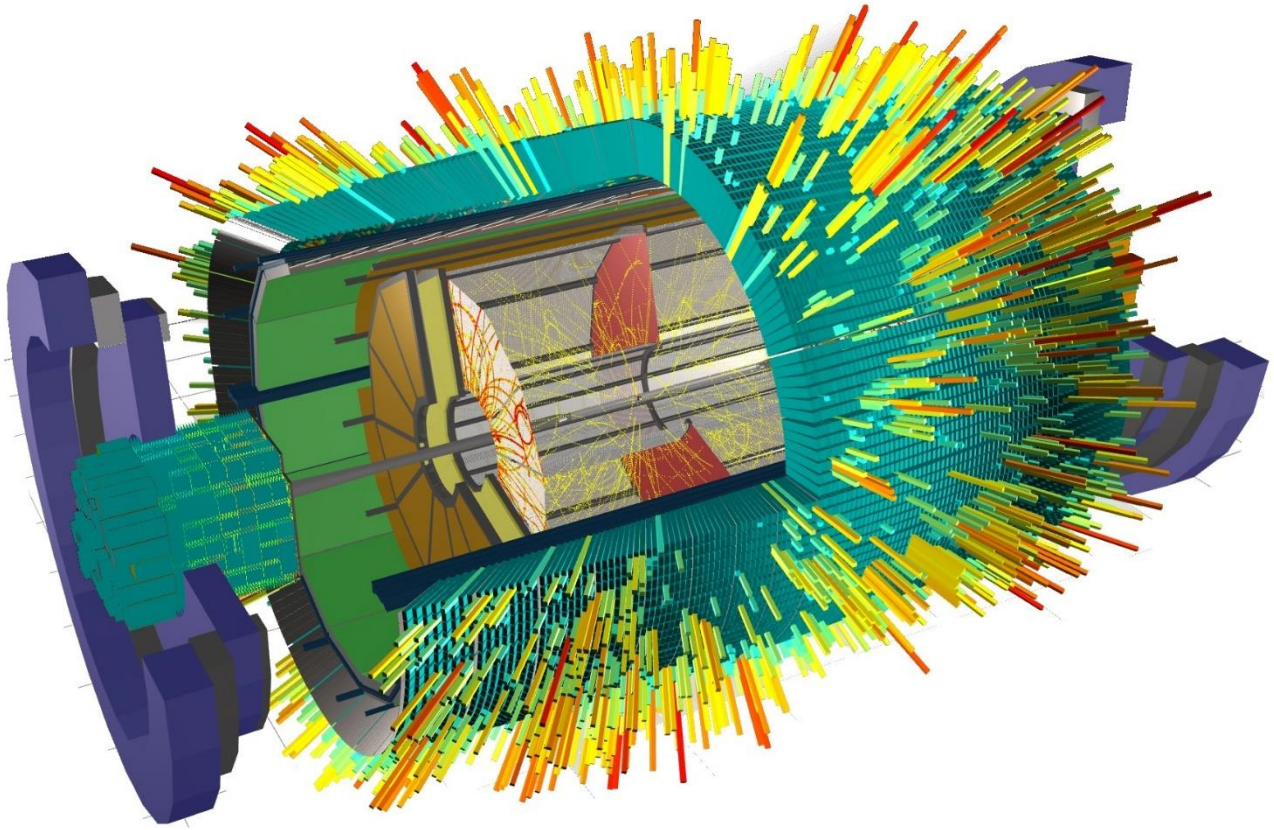
ESSENTIAL CRITERION – real experiment reusability (QA toolkit)

SOFTWARE DEVELOPMENT

- up to date with latest packages (Acts – breaking changes every 1-2 months)
- regular release schedule
- automated tests
- cleanup
- refactoring

Thank You !

Q & A



MPD Software Development & Computing Team

Rogachevsky O.	Coordinator
Krylov V., Krylov A.	Online MPD Event Display
Moshkin A., Pelevanyuk I.	Mass Production
Bychkov A.	Detector Simulation
Kuzmin V.	Detector Alignment
Podgainy D., Zuev M.	Supercomputing
Alexandrov E., Alexandrov I.	Databases
Balashov N.	Gitlab Support
Belyakov D.	Network Infrastructure
Belecky P., Kamkin A.	Acts Tracker
Busa J.	Build System
Hnatic S.	Architecture