Optimization of simulation and reconstruction algorithms in BmnRoot

S.P.Merts, S.A.Nemnyugin, V.A.Roudnev, M.M.Stepanova

Saint-Petersburg State University Joint Institute for Nuclear Research

4th Collaboration Meeting of the BM@N Experiment at the NICA Facility, 14-15 October 2019

















Simulation



Reconstruction





- 1. «Handmade» (source level) timing.
- 2. Google Performance Tools (GPT).
- 3. Advanced tools for deep analysis of the software code bottlenecks.





Testbench 1

CPU: Intel(R) Core(TM) i5-2400 @ 3.10GHz (4C noHT, L3 Cache 6MB) RAM: 16GB OS: Linux (Ubuntu)

Testbench 2

CPU: Intel Xeon E-2136 @ 4.5GHz Turbo (6C 2xHT, L3 Cache 8MB) RAM: 32GB OS: Linux (Ubuntu)

Testcase 1

Simulation with BOX generator 5000 events for hotspot analyses Macros run_sim_bmn.C

Testcase 2

Simulation with DCM-QGSM generator 5000 events for hotspots/1000 events to study scalability Macros run_reco_bmn.C

Testcase 3

Reconstruction for data simulated by DCM-QGSM generator 5000 events for hotspot analyses/1000 to study scalability Macros run_rec_bmn.C







Reconstruction

4th Collaboration Meeting of the BM@N Experiment at the NICA Facility



Advanced hotspot analysis



Simulation (Testbench 2 + Testcase 2)

sincos	399.4 s
Trandom::Gaus	368.9 s
DeadZoneOfStripLayer::IsInside	365.2 s
TRandom3::Rndm	231.6 s
deflate	167.3 s
BmnGemStripModule::AddRealPointFull	120.8 s
BmnGemStripLayer::ConvertPointToStripPosition	97.7 s
•••	





sincos,"399.411","0","0","libm.so.6","sincos","[Unknown]","0x23db0" BmnGemStripLayer::ConvertNormalPointToStripX,"380.587","0","0","libGem.so.0.00","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","IMnkown]","0x15be30" TGeoPgon::GetPhiCrossList,"1.69609","0","0","libGeom.so.6.12","TGeoPgon::GetPhiCrossList(double const*, double const*, int, double*, int*, double) const","[Unknown]","0x14fae0" phpnuc_ <- nucrec_,"0.920032","0","0","libGeom.so.6.12","TGeoPgon::GetPhiCrossList(double const*, double const*, int, int, bool, double, double) const","[Unknown]","0x14fae0" TGeoPgon::SafetyToSegment <- TGeoPgon::Safety,"0.763982","0","0","libGeom.so.6.12","TGeoPgon::SafetyToSegment(double const*, int, int, bool, double, double) const","[Unknown]","0x15 TGeoPgon::Contains,"0.415969","0","0","libGeom.so.6.12","TGeoPgon::Contains(double const*) const","[Unknown]","0x155280" TGeoPgon::SliceCrossingIn <- TGeoPgon::DistFromInside <- TGeoNavigator::FindNextBoundary <- gtnextTGeo,"0.20401","0","libGeom.so.6.12","TGeoPgon::SliceCrossingIn(double const*, int, double, double) const","[Unknown]","0x154130" TGeoPgon::SliceCrossingSlice <- TGeoPgon::SliceCrossing <- TGeoPgon::DistFromOutside <- TGeoShapeAssembly::DistFromOutside <- TGeoNavigator::FindNextDaughterBoundary <- TGeoNavigator::F
BmnGemStripLayer::ConvertNormalPointToStripX,"380.587","0","0","libGem.so.0.0.0","BmnGemStripLayer::ConvertNormalPointToStripX(double, double)","BmnGemStripLayer.cxx","0x822b0 TGeoShape::SafetyPhi,"14.7765","0","0","libGeom.so.6.12","TGeoShape::SafetyPhi(double const*, bool, double, double)","[Unknown]","0x15be30" TGeoPgon::GetPhiCrossList,"1.69609","0","0","libGeom.so.6.12","TGeoPgon::GetPhiCrossList(double const*, double const*, int, double*, int*, double) const","[Unknown]","0x14fae0" phpnuc_< nucrec_,"0.920032","0","0","0","libGeom.so.6.12","TGeoPgon::GetPhiCrossList(double const*, double const*, int, double*, int*, double), double, double) const","[Unknown]","0x14fae0"
TGeoShape::SafetyPhi,"14.7765","0","0","libGeom.so.6.12","TGeoShape::SafetyPhi(double const*, bool, double, double),"[Unknown]","0x15be30" Image: Constant to the type of type of the type of type of the type of ty
TGeoPgon::GetPhiCrossList,"1.69609","0","0","10","10","0","10","10","0","10","10","0","
phpnuc_<- nucrec_"0.920032","0","0","libgeant321.so","phpnuc_","[Unknown]","0xc6cc0"
TGeoPgon::SafetyToSegment <- TGeoPgon::Safety,"0.763982","0","0","libGeom.so.6.12","TGeoPgon::SafetyToSegment(double const*, int, int, bool, double, double) const","[Unknown]","0x157280"
TGeoPgon::Contains, "0.415969", "0", "0", "libGeom.so.6.12", "TGeoPgon::Contains(double const") const", "[Unknown]", "0x155280" Image: Contains, "0.415969", "0", "0", "libGeom.so.6.12", "TGeoPgon::DistFromInside <- TGeoNavigator::FindNextBoundary <- gtnextTGeo, "0.20401", "0", "0", "libGeom.so.6.12", "TGeoPgon::SliceCrossingIn(double const", TGeoPgon::DistFromOutside, "0.040001", "0", "0", "libGeom.so.6.12", "TGeoPgon::DistFromOutside, "0.040001", "0", "0", "libGeom.so.6.12", "TGeoPgon::DistFromOutside, "0.040001", "0", "0", "libGeom.so.6.12", "TGeoPgon::DistFromOutside, "0.040001", "0", "0", "libGeom.so.6.12", "TGeoPgon::DistFromOutside(double const", double const", int, double, double*) const", "[Unknown]", "0x154130"
TGeoPgon::SliceCrossingIn <- TGeoPgon::DistFromInside <- TGeoNavigator::FindNextBoundary <- gtnextTGeo,"0.20401","0","0","libGeom.so.6.12","TGeoPgon::SliceCrossingIn(double const*, TGeoPgon::DistFromOutside,"0.040001","0","0","libGeom.so.6.12","TGeoPgon::DistFromOutside(double const*, double const*, int, double, double*) const","[Unknown]","0x154130"
TGeoPgon::DistFromOutside,"0.040001","0","0","libGeom.so.6.12","TGeoPgon::DistFromOutside(double const*, double const*, int, double, double*) const","[Unknown]","0x154130" TGeoPgon::IsCrossingSlice <- TGeoPgon::SliceCrossing <- TGeoPgon::DistFromOutside <- TGeoShapeAssembly::DistFromOutside <- TGeoNavigator::FindNextDaughterBoundary <- TGeoNavigator::FindNextDaughterBoundary <- TGeoNavigator::Gaus,"368.883","0","0","libMathCore.so.6.12.06","TRandom::Gaus(double, double)","[Unknown]","0x10dc80"
TGeoPgon::IsCrossingSlice <- TGeoPgon::SliceCrossing <- TGeoPgon::DistFromOutside <- TGeoShapeAssembly::DistFromOutside <- TGeoNavigator::FindNextDaughterBoundary <- TGeoNavigator::FindNextDaughterBoundary <- TGeoNavigator::Gaus,"368.883","0","0","19,"19, 19, 19, 19, 19, 19, 19, 19, 19, 19,
TRandom::Gaus,"368.883","0","0","0","libMathCore.so.6.12.06","TRandom::Gaus(double, double)","[Unknown]","0x10dc80" BmnGemStripModule::AddRealPointFull <- BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCPoints <- BmnGemStripStationStationStationSet::AddPointToDetector <- BmnSiliconStationSet::AddPointToDetector <- BmnSiliconStationDigitizer::ProcessMCPoints <- BmnSiliconDigitizer::ProcessMCPointFullOne <- BmnSiliconStation::AddPointToStation <- BmnSiliconStationSet::AddPointToDetector <- BmnSiliconDigitizer::ProcessMCPointFullOne <- BmnSiliconStation::AddPointToStation <- BmnSiliconStationSet::AddPointToDetector <- BmnSiliconDigitizer::ProcessMCPointFullOne <- BmnSiliconDigitizer::ProcessMCPointFoDetector <- BmnSiliconDigitizer::ProcessMCPointFo
BmnGemStripModule::AddRealPointFull <- BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCPoints <- BmnGemS BmnSiliconModule::MakeCluster <- BmnSiliconModule::AddRealPointFullOne <- BmnSiliconStation::AddPointToStation <- BmnSiliconStationSet::AddPointToDetector <- BmnSiliconDigitizer::ProcessMCPointSet::Pr
BmnSiliconModule::MakeCluster <- BmnSiliconModule::AddRealPointFullOne <- BmnSiliconStation::AddPointToStationStat
BmnSiliconModule::AddRealPointFullOne <- BmnSiliconStation::AddPointToStation <- BmnSiliconStationSet::AddPointToDetector <- BmnSiliconDigitizer::ProcessMCPoints <- BmnSiliconDigitiz
DeadZoneOfStripLayer::IsInside,"362.539","0","0","libGem.so.0.0.0","DeadZoneOfStripLayer::IsInside(double, double)","BmnStripData.h","0x81ca0"
BmnGemStripLayer::IsPointInsideDeadZones <- BmnGemStripLayer::IsPointInsideStripLayer,"362.539","0","0","libGem.so.0.0.0","BmnGemStripLayer::IsPointInsideDeadZones(double, double)"
TRandom3::Rndm,"231.622","0","0","libMathCore.so.6.12.06","TRandom3::Rndm(void)","[Unknown]","0x10ffa0"
TRandom::Gaus <- BmnGemStripModule::AddRealPointFull <- BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCI
grndm_,"26.5638","0","0","libgeant321.so","grndm_","[Unknown]","0x26b790"
TRandom3::SetSeed,"0.368046","0","0","libMathCore.so.6.12.06","TRandom3::SetSeed(unsigned long)","[Unknown]","0x110300"
TRandom::Exp <- BmnGemStripModule::AddRealPointFull <- BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCPc
deflate,"167.345","0","0","libz.so.1","deflate","[Unknown]","0x4800"
R_zipMultipleAlgorithm,"167.345","0","0","libCore.so.6.12","R_zipMultipleAlgorithm","[Unknown]","0x2c1750"
BmnGemStripModule::AddRealPointFull,"120.852","0","0","libGem.so.0.0.0","BmnGemStripModule::AddRealPointFull(double, double, double, double, double, double, double, int)","BmnGemStripModule::AddRealPointFull(double, double,
BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCPoints <- BmnGemStripDigitizer::Exec <- FairTask::ExecuteTask
BmnGemStripLayer::ConvertPointToStripPosition,"97.6891","0","1ibGem.so.0.0.0","BmnGemStripLayer::ConvertPointToStripPosition(double, double)","BmnGemStripLayer.cxx","0x824b0"
BmnGemStripModule::AddRealPointFull <- BmnGemStripStation::AddPointToStation <- BmnGemStripStationSet::AddPointToDetector <- BmnGemStripDigitizer::ProcessMCPoints <- BmnGemS





Reconstruction (Testbench 2 + Testcase 3)

BmnCellAutoTracking::CellsConnection	239.3 s
inflate	48.2 s
BmnKalmanFilter::RK4Order	22.1 s
BmnNewFieldMap::FieldInterpolate	16.6 s
•••	
BmnNewFieldMap::IsInside	12.0 s
BmnKalmanFilter::TransportC	10.1 s
•••	





1	Function / Call Stack,"Effective Time","Spin Time","Overhead Time","Module","Function (Full)","Source File","Start Address"
2	BmnCellAutoTracking::CellsConnection,"239.312","0","0","libGem.so.0.0.0","BmnCellAutoTracking::CellsConnection(std::vector <bmncellduet, std::allocator<bmncellduet="">>*, std:</bmncellduet,>
3	BmnCellAutoTracking::Exec <- FairTask::ExecuteTasks <- FairTask::ExecuteTask <- FairRunAna::Run <- [Unknown stack frame(s)] <- cling::MetaParser::isXCommand <- cling::MetaP
4	inflate,"48.2329","0","libz.so.1","inflate","[Unknown]","0x9d20"
5	Runzip,"48.2329","0","libCore.so.6.12","Runzip","[Unknown]","0x2c1a20"
6	BmnKalmanFilter::RK4Order,"22.1161","0","0","libBmnData.so.0","BmnKalmanFilter::RK4Order(std::vector <double, std::allocator<double="">> const&, double, std::vector<double, std:<="" td=""></double,></double,>
7	BmnKalmanFilter::RK4TrackExtrapolate <- BmnKalmanFilter::TGeoTrackPropagate,"22.1161","0","libBmnData.so.0","BmnKalmanFilter::RK4TrackExtrapolate(FairTrackParam*,
8	BmnNewFieldMap::FieldInterpolate, "16.552", "0", "0", "libBmnField.so.0.0.0", "BmnNewFieldMap::FieldInterpolate(TArrayF*, double, double, double)", "BmnNewFieldMap.cxx", "0x1b
9	BmnNewFieldMap::GetBy,"5.80009","0","0","libBmnField.so.0.0.0","BmnNewFieldMap::GetBy(double, double, double)","BmnNewFieldMap.cxx","0x1b92c"
10	BmnNewFieldMap::GetBx <- BmnKalmanFilter::RK4Order <- BmnKalmanFilter::RK4TrackExtrapolate <- BmnKalmanFilter::TGeoTrackPropagate,"5.50002","0","0","libBmnField.so.0
11	BmnNewFieldMap::GetBz <- BmnKalmanFilter::RK4Order <- BmnKalmanFilter::RK4TrackExtrapolate <- BmnKalmanFilter::TGeoTrackPropagate,"5.25186","0","0","libBmnField.so.0
12	std::map <float, int,="" std::less<float="">, std::allocator<std::pair<float const,="" int="">>>::operator[],"16.5514","0","0","libBmnData.so.0","std::map<float, int,="" std::less<float="">, std::allocator</float,></std::pair<float></float,>
13	BmnTrack::AddHit,"16.5514","0","0","libBmnData.so.0","BmnTrack::AddHit(int, FairHit*)","BmnTrack.cxx","0x13e270"
14	operator new,"13.5602","0","libstdc++.so.6","operator new(unsigned long)","[Unknown]","0x8de60"
15	gnu_cxx::new_allocator <double>::allocate,"5.33216","0","0","libBmnData.so.0","gnu_cxx::new_allocator<double>::allocate(unsigned long, void const*)","new_allocator.h","</double></double>
16	gnu_cxx::new_allocator <std::_rb_tree_node<std::pair<float const,="" int="">>>::allocate,"2.35993","0","0","libBmnData.so.0","gnu_cxx::new_allocator<std::_rb_tree_node<std::pair<float const,="" int="">>>::allocate,"2.35993","0","0","libBmnData.so.0","gnu_cxx::new_allocator<std::_rb_tree_node<std::pair< p=""></std::_rb_tree_node<std::pair<></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float></std::_rb_tree_node<std::pair<float>
17	gnu_cxx::new_allocator <bmnmaterialinfo>::allocate <- std::_Vector_base<bmnmaterialinfo, std::allocator<bmnmaterialinfo="">>::_M_allocate <- std::vector<bmnmaterialinfo,< td=""></bmnmaterialinfo,<></bmnmaterialinfo,></bmnmaterialinfo>
18	TStorage::ObjectAlloc,"0.608008","0","0","libCore.so.6.12","TStorage::ObjectAlloc(unsigned long)","[Unknown]","0x23eb10"
19	std::vector <double, std::allocator<double="">>::_M_emplace_back_aux<double const&=""> <- std::vector<double, std::allocator<double="">>::push_back,"0.452001","0","libMathCore</double,></double></double,>
20	std::vector <double, std::allocator<double="">>::_M_emplace_back_aux<double> <- std::vector<double, std::allocator<double="">>::push_back <- dist <- BmnCellAutoTracking::CalcQp</double,></double></double,>
21	gnu_cxx::new_allocator <tline*>::allocate <- std::vector<double, std::allocator<double="">>::vector <- BmnCellAutoTracking::CalcQp <- BmnCellAutoTracking::CalculateTrackPara</double,></tline*>
22	gnu_cxx::new_allocator <bmnlink>::allocate,"0.199997","0","0","libBmnData.so.0","gnu_cxx::new_allocator<bmnlink>::allocate(unsigned long, void const*)","new_allocator</bmnlink></bmnlink>
23	gnu_cxx::new_allocator <tline*>::allocate <- std::vector<double, std::allocator<double="">>::vector <- BmnCellAutoTracking::CalcQp <- BmnCellAutoTracking::CalculateTrackPara</double,></tline*>
24	





gcc compiler optimization

-O2/-O3 by default

+

Aggressive vectorization. Autoparallelization of loops. Profile-guided optimization. Data alignment. Various kinds of loops optimization etc

→

No impressive effects.

Microarchitecture analysis + advanced hotspot analysis

Inefficiencies in the code: dependencies, inefficient use of pipelines and so on.





OpenMP parallelization

Simple OpenMP-parallelization of the most "hot" loop in CellsConnection method must be verified by Quality Assurance module. It doesn't work (data races).

Threadsafe parallelization requires changes of algorithm.

Modified algorithm is parallelized.



QA&Scalability (Testbench 1 + Testcase 3)







Scalability (Run-4699)









MWPC – schematic

Simulation requires the description of detector geometry, its proportions, materials and location. The **ROOT** geometry package is used.





Z distribution of secondary particles

















- Dynamic analysis of the BmnRoot modules which are used in simulation and track reconstruction has been performed and hotspots are localized with different approaches.
- One of the most significant hotspots in reconstruction module has been parallelized for shared memory computing systems.
- Parallelized version has been verified. Scalabilty was studied both on simulated and experimental data.
- Functionality of the BmnRoot has been extended by more complete description of the MWPC detector. Digitizer, hit finder and track reconstruction algorithm have been developed for simulated data and implemented as C++ classes for BmnRoot.
- We plan deeper optimization of the performance both of the BmnRoot simulation and reconstruction modules.
- Sources will be uploaded in repository of the BM@N (BmnRoot).

Presented results are part of the project supported by the RFBR grant #18-02-40104.

THANK YOU FOR ATTENTION