VecGeom navigation on the GPUs

Guilherme Lima (FNAL)

VecGeom Project Meeting
December 16, 2019
Introduction

- **US is investing in GPU-based HPC for HEP computing**
  - Exascale: GPU for the foreseeable future
  - ECP: Investing in software for Exascale
  - Geant On GPU: prior efforts
  - Exascale Geant Pilot: a collaboration between ECP and HEP
    Fermilab, L Berkeley NL, Oak Ridge NL
  - Now is the time to prepare for the next generation of supercomputers

- Reference: [J. Madsen’s talk at 24th Geant4 collaboration meeting](#)
ExaScale Pilot – some goals

• study and characterize architecture and performance
  - Explore memory access, computation ordering, CPU/GPU communication patterns, ...

• Investigate how to best use GPUs for full HEP simulation
  - estimate higher limit of speed-ups

• Strategies:
  - reuse or leverage existing packages (VecCore, VecGeom, GeantV)
  - require flexibility on data and coding structure
  - avoid bw compatibility and *upstream* tracking constraints
  - Focus on NVidia, while keeping an eye on Intel, AMD GPU developments

  - Reference: [J. Madsen's talk on the 24th Geant4 collaboration meeting](https://example.com)
Geometry: solids and navigation

• Geometry system mostly absorbed from VecGeom
  – VecGeom package is a git submodule
    • libraries fully available at link time
  – Integration test executable available, including geometry
    • builds a TestEm3 geometry
    • used for integrating other components (framework infrastructure, scheduling, track model, EM physics model)
  – How about navigation?
    • VecGeom’s NavigationBenchmark worked in GPUs (3-4 years ago)
    • Broken for long time (since upgrade to indexed navigation states?)
    • Good candidate, if it can be “revived” on the GPU
Reviving navigation on the GPU

- **What is now available?**
  - **NavigationBenchmark was disabled**
    - depended on SimpleNavigator, now deprecated since it depended on obsolete function SafetyToInMinimize()
    - also on point location methods from navigators → new classes like GlobalLocator, SafetyEstimator, etc.
    - testVectorSafety ran some navigation tests – also disabled, probably for the same reasons.
  - Sandro pointed me to NavigationKernelBenchmark
    - it does not seem to have any GPU support
    - several navigators: NewSimpleNavigator, SimpleABBoxNavigator, VoxelNavigator, HybridNavigator

- **Plan of action**
  - Look at source code to learn from what runs now
  - revive testVectorSafety and NavigationBenchmark (finishing up now, passes CI tests)
  - add navigation capabilities to the existing pilot’s integration test (before end of year)
  - go back and fix what is not working (vectorized navigation, located volume names)
So what was wrong with old tests?

- **testVectorSafety:**
  - memory corruption due to inverted arguments: AlignedAlloc(size, 32) – [fixed on 501d80b0]
  - deprecated stuff, due to being disabled for long time
  - vector safety still broken, probably due to incorrect way to drop SafetyToInMinimize() with a SafetyToIn() call

- **NavigationBenchmark:**
  - upgrades to use more recent stuff (e.g. GlobalLocator, NewSimpleNavigator, SafetyEstimator)
  - new methods to use NavStatePool instead of NavigationState**
  - fixes to NavStatePool’s CopyToGpu() and CopyFromGpu(), from SizeOf() to SizeOfAlignAware()
  - Corresponding “align-aware” fixes to NavigationBenchmarker’s cuda kernel to rebuild NavigationState instances on the GPU
  - navigation issue: VecGeom vs. Root mismatch for location volume names (1% level)
  - GPU navigation stepping is mostly validated – only a 0.5% validation failure rate!
Current status: testVectorSafety

# safety mismatches: 539/1024

# navigation mismatches: 491/1024
*** Validating VecGeom navigation...
validateAgainstROOT: pos=(-5.17231, 7.88185, 0.527992) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(-2.15661, 3.49906, 9.23149) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(-2.68548, -9.17899, -3.05581) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(-2.70118, -9.90599, -1.63701) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(9.72857, -2.45692, -3.47319) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(-9.4389, -7.56387, -0.192487) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(9.81287, -6.84684, -2.94348) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(9.57142, 7.67596, -5.34047) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(-1.03633, 9.3275, -9.37924) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: pos=(1.77207, 7.20055, 9.33555) -> vol name mismatch: VName=<NULL>, ROOTName=<world_1>
validateAgainstROOT: more mismatches detected, but further reports dropped!
VecGeom navigation – serial interface: # ROOT mismatches (step lengths) = 0 / 1024
VecGeom validation passed.

*** Running navigation benchmarks with ntracks=1024 and nreps=3.
Uncontained capacity for world_1: 3904 units
PlacedVolume created after geometry is closed --> will not be registered
VolumeUtilities: FillBiasedDirs: nhits/size = 611/1024 and requested bias=0.800000
really hits 820, virtually hits 820 CPU elapsed time: 0.401302 ms for locating and setting steps
CPU elapsed time: 0.558743 ms for serialSafety<NewSimpleNavigator>>
CPU elapsed time: 0.268010 ms for serialSafety<SimpleABBoxNavigator>>
CPU elapsed time: 0.108868 ms for vectorSafety<NewSimpleNavigator>>
CPU elapsed time: 0.167458 ms for vectorSafety<SimpleABBoxNavigator>>
CPU elapsed time: 0.863356 ms for ROOT
CPU elapsed time: 1.236051 ms for serialized navigation
CPU elapsed time: 0.623282 ms for vectorized navigation
CPU elapsed time: 1.694365 ms for ROOT navigation

lima@mac-131320: build $ ccmake ..

Current status: NavigationBenchmark

Volume name mismatch

CPU performance as curiosity only for now (pending on validation)
current status – NavigBenchmark on the GPU

GPU performance as curiosity only for now (pending on validation)

* transfer times are not included!
Summary

• Good progress on reviving navigation on the GPU
  - Fixes pushed to git branch lima/navigOnGpu
  - merge request #718 passes code checks on CI

• Working this week on adding this navigation feature to the Geant exascale pilot integration

• Next step will be to understand and fix the issues with vectorized safeties and navigation mismatches