HSF Simulation Meeting with Lightning talks on accelerators in simulation

Simulation Notebook

The notebook here is for comments, questions, discussions and any issues that should be raised at the meeting and could be followed up afterwards. Anyone is welcome to add their comments and observations also during the meeting

If you wish to raise a question or discussion point here ***please give your name*** so that we know who asked the question and give you the 'microphone'. Questions listed here will be given the priority in the discussion

Lightning talks on 10 June 2020

Prototyping simulation workflows on GPUs - an exploration phase

by Andrei Gheata

Q. Gloria Corti - The resources that will be available will be a combination of CPUs and GPUs. Are you thinking about separate scheduling?

A: (Andrei Gheata) While we definitely need a CPU workload scheduler that is aware of GPUs, I think we will also need a GPU-specific scheduler to handle processing full transportation steps as a pipeline of kernels doing geometry, field propagation, physics and scoring.

Q: Should we investigate code specific to given architectures or use the performance portability tools that are coming on the market for addressing different type of GPUs?
A: (AG) It looks like a prerequisite is adapting the data model/algorithms targeting a single architecture/programming model (GPU/CUDA), but investigating portability tools can/should be done meanwhile I think. Definitely we shouldn't try to convert fully the simulation framework in CUDA, but at least try to understand the workflow and data model, and use maybe portability tools after that.

A: We need to adapt first the application to GPUs and then address the portability problem. Expect to learn from the HEP-CCE project's investigations on this.

Q. Witek Pokorski - We need to 'flatten' the geometry if we want to use vendor libraries like Optix, however, if use VecGeom, navigation of hierarchical geometry seems to work fine, right? Or to put it differently, do you see limitations of hierarchical geometries in VecGeom on GPU.

A: With deep hierarchies we may hit some limits (e.g. using recursions), but we have to experiment them first.

Q. Ianna Osborne - What would you suggest Reducing geometry size by algorithmically placing Assemblies?

A: Assemblies is one of the way of flattening the geometries, but are not solving the complexity problem.

Comment : Simon Blyth, on Opticks geometry hierarchy

One thing I think I should clarify is about the hierarchy of the Opticks geometry. Although it is very flat as you give it to
NVIDIA OptiX, it then constructs a BVH with a hierarchy designed to optimize traversal speed.

It makes sense for the hierarchy of the acceleration structure to be flexibly optimized for traversal, rather than just being for the convenience of organization.

A: Clearly, the BVH hierarchy is OK as a fast geometry traversal for minimizing the number of effective individual shape checks. The problem is the current deep logical volume hierarchy based on containment that we use in our detector setups to describe the geometry, which hinders on building an appropriate BVH w/o preliminary (potentially heavy) transformations for flattening.

Pre-learning geometry on GPUs

By Vangelis Kourlitis

John A> Can you compare the number of arithmetic operations required for particular sets of volumes with the number of operations for a NN (especially a DNN) ? It would seem that this approach would be relevant only the most complex surfaces / sets of volumes might be of interest, unless a small network is used.

Response/Walter Hopkins (WH): It is correct that we expect this only would probably be mainly useful for a complex geometry. We have done some comparison for the calculation time with simple geometries such as a box and found that Geant4 is quite fast and actually faster than a NN. We want to test this for ATLAS or polycone geometries.

JA> Maybe the most interesting type of solid(s) would be the Twisted Solids, which are similar to parts of the EMEC. TwistedTrap is one.

WH: Thanks, we will investigate the twisted solids example.

Q. Gloria Corti - I did not understand your comment that this may not be necessary if you are inside a box: what about particles produced inside the box they would still need to know how far they can go

Response/WH: that is correct. You have to calculate the distance to the nearest boundary for every step you consider. On slide 4, B has also undergone the same calculation but the process had a shorter step length than the geometric boundary. If I'm understanding the question correctly.

Response/VK: maybe my point was that if a particle stands in the center of a cube box, the step shouldn't be really limited by geometry. Can we decide on this safely (e.g. based on position) prior of doing the distance calculation? In this way we can save function/NN calls.

Q: Witek Pokorski: There is not only the calculation of the distance involved, but also the navigation (finding out in which volume you are). I guess, your approach can provide more speed up if you also take into account the navigation.

(comment/JA) In almost no cases can you ignore which volume is hit by a 'ray'. If a particle crosses that surface you must always be able to identify what volume / material (etc) is on the other side of that surface.

Response/WH: currently we only calculate the distance to the next volume, given the starting position and the direction proposed by a process. We don't actually do the transportation to the new volume. We found that the distance calculation is what takes a long time for a complex geometry like the ATLAS EMEC. We have detailed profiling studies for ATLAS that we could point you to.

EM physics with CUDA/portable APIs in the Excalibur-HEP project

by Benjamin Morgan

Q: Witek Pokorski: so, you are going to implement those few physics processes in CUDA?

- Some 'toy' processes might still demonstrate key features. However, Mihaly has some modernised EM physics code that might be more portable to GPU.

Q: Gloria Corti: how do you plan to make the outcome available? Even the inventory of what work/doesn't, what is out there would be useful

Q. Graeme Stewart: do you have particles of the same type in a structure that can execute on the same warp?

- Fill warps at first, allow particles to deplete. At some point reorganising and garbage collection would become worthwhile (but not every step!)

e/γ calorimeter simulation using native GPU navigation libraries & shaders: a feasibility study

by John Apostolakis

Q: Gloria Corti - Extracting information from the GPUs: I agree that from a calorimeter ‘hits’ (what do you define as a hit?) are what you want with a reference to the particle originating the shower. But what when you are transporting on a tracker? There you may also want the secondary particles…

[Ioana Osborne: Are you planning to use realistic calorimeters as testbeds or simplified ones?]
John> Am seeking to evaluate realistic LHC calorimeters, and the ability of Optix / Opticks to navigate in these. If only 1-2 extra types of solids are needed, one (or more) of these should make a good benchmark geometry to use.

GATE talks on 24 June 2020

Experience with hGATE on GPUs

by Julien Bert

Q. Slide 9: why the speed up is going down to 2. What are the parts of the code which are not run on GPUs?

A. Only the phantom voxel navigation is done on the GPU in hGATE.

Q. Can the current code be compiled for CPU as well?

A. No, GGEMS current code only runs on GPU; this was a significant drawback for validation and motivates the move to OpenCL

Q. How did you measure the factor 7-10 speed up from C++ to CUDA/C

A. Running the cuda code on the CPU has significant inefficiencies so we made some standalone prototype.

Q. Isn’t OpenCL no longer supported and thus a sub-optimal choice?

A. We really needs its ability to run on both CPU and GPU and the simplification in maintenance. It also allows us to run on older cards.

Q. Did you look at different strategies for reducing thread divergence - not sorting everytime, but only when the warp efficiency had dropped beneath a threshold?

A. Cost to sort (to reduce divergence) was higher than the gain (from less divergence). We did not investigate variation on the sort granularity and frequency.

Q. Did you investigate the effect of occupancy/kernel size on performance?

A. Especially with hGATE (and older cards) we needed to optimized the kernel size as the number of register available was very low. Also we have one generator per threads is costly due to the size of the states (i.e. lots of memory operation for which random number generation).

Lightning talks on 24 June 2020

Porting the ATLAS fast calorimeter simulation to CUDA and SYCL

by Vincent Pascuzzi

Q. Only part of the code is ported to the GPU. Do you do buffering to make efficient use of the GPU?

A. This is the idea, but the initial samples only had single particles.

Q. What are the next steps?

A. Complete simulation (on SYCL) and perform benchmarks. Want to use new Intel cards (non-shared memory).

Q. Is the geometry on the GPU flattened?

A. It's read in from a text file on the host, then massaged and transferred. This is a *read-out geometry*, not a simulation geometry.

Geometry for GPUs

by Akanksha Vishwakarma

Q. Have you started looking at the possibility of tessellating the geometry yet?

A. Not yet.

Q. What is the timeframe of the project?

A. Funded for a couple of years from now.

Prospects for RICH detector simulation using OptiX in GPUs

by Sajan Easo

Q. Do you have any estimate of the speedup?

A. Not yet, this is one of the next things to come out.

Q. How difficult it will be to port the RICH geometry to Optix?

A. I don't expect any major problems. Related LHCb is migrating to DD4Hep.