HSF November 2020 Workshop - Outcomes and Followup

OVERVIEW

This document is a summary of outcomes and follow-up points for the HSF part of the WLCG-HSF November 2020 Virtual Workshop. It should be used by appropriate HSF working groups as items that can be followed up on in future meetings.

For more details of the discussions, see the Workshop Notebook,

Day 1 - Thursday 19 November, HSF Plenary

Future trends in Nuclear Physics Software and Computing. Training identified as a fruitful area of collaboration [Training WG]. Python [PyHEP] should be a topical focus. Proposal to have the HSF join the Software and Computing Roundtable organisation [HSF Coordination], to broaden it's audience and scope.

PyHEP. Encourage citation of software packages [HSF in general] so that people are encouraged to do this. N.B. we could again make this appeal connected to vCHEP2021. PyHEP2021 should be organised! Explore ways to improve engagement with people who register. [PyHEP].

Spack and Key4hep. Quite a number of technical issues to follow-up on with the [Tools and Packaging WG]: package relocatability (what can be improved on the Spack side? what's a bug in the package?); testing and giving feedback on the new concretiser (feeds into the discussion on common base layers can be used, cf. Gentoo Prefix builds; EESSI project); helping people move away from LD_LIBRARY_PATH to RPATH based distributions; supporting a community of upstream builds that smaller experiments could use.

Training. Captioning is supported (but requires some money). If lessons are unsuitable or need extended fork (and maybe PR back changes) and we should encourage people to do that, instead of starting from scratch. Improving interactive participation is a challenge for online events - at least encourage sharing the experience of what could work (key is to build engagement, but having micro-payments not necessarily the best). University departments and

Snowmass Process are areas where we can try to engage more. We should look at research on quantifying success. Discuss with The Carpentries. [Training WG].

Day 2 - Friday 20 November, Event Generators

GPU developments

- Madgraph on GPU progressing well, but still several components missing for production use at LHC, including PDFs, better phase space sampling and more.
- PDFFlow and VegasFlow look like mature products. Even if they are based on ML, they do not introduce additional approximations with respect to LHAPDF or Vegas, and should have comparable physics precision. They would fit nicely in what Madgraph on GPU still needs to address, and will certainly be tested in that context.

Negative weights

- Very good progress in MC@NLO-Delta. This is an approach based on fundamental theory, with a comparable level of physics precision as MC@NLO. This could lead to some significant savings in CPU budgets (for simulation and reconstruction, because fewer events are needed). The experiments should test it and give feedback. It is in a beta release and one may need to pay attention to external dependencies such as Pythia8. Experiments should also check if the matching uncertainty comes up correct from this method. We should also understand what are the CPU costs to the different folding schemes.
- Neural resampler is a complementary approach, based on approximating the multi-dimensional distributions from NLO generators (which are the result of the sum of positive and negative weight events) using ML methods. This introduces additional approximations which may impact physics precision (the problem is not dissimilar from the proposed use of GANs for detector simulation, to a certain extent). Experimentalists and theorists should evaluate more carefully its benefits and limitations. This is a physics task much more than a software or computing task at this stage however it would be good to follow up the progress, e.g. CPU cost of event generation, or identifying regions of negative weights.

Day 3 - Monday 21 November, Detector Simulation

Geant4 is and will remain in the foreseeable future the de-facto standard for HEP detector simulation. Attracting a new generation of young developers is crucial and needs to be addressed in a proactive manner.

In HEP experiments a wide range of fast simulations are more and more complementing Geant4 in order to address the ever growing need in size of simulated samples. Even if the implementations are specific to a given detector technology the methods are often very similar. Work is starting on understanding and quantifying the quality and fidelity of the deep learning fast simulation and on evaluating the cost of both generating the input and running the training.

There are three axes along which the development needs to happen:

- Improvements and modernisation of Geant4 code-basis
- Fast simulation methods development and integration
- R&D on the particle transport on new hardware platforms (GPUs)

R&D activities are taking place as far as running simulation on GPUs are concerned and it is expected to have first prototypes demonstrating the possibility (or not) of full HEP simulation on those platforms within a year time-scale.

Integration, customization and tuning of all three development's axes into HEP experiments simulation applications also need to be addressed. Information exchange and close collaboration between Geant4 and the experiments' software developers is thus essential.

Day 4 - Tuesday 22 November, Software R&D

Takeaways and follow-ups from the R&D talks

- Phoenix event display offers a modern high-performance web based event display suitable for many different experiments, easy to get started. Promote Phoenix more widely to potential users. [HSF Coordination, Visualisation Activity]
- There is a lot of R&D going into building efficient high-performance analysis environments (including RDataFrame and tools built on top, like bamboo). Analysis description languages are promising in some aspects, but unclear which technologies are best to use. [Analysis WG]. Analysis 'standard candles' would help compare different systems. Scaling issues, including multiple users, should be addressed. Also need to prioritise UI and transparency. [Analysis WG, PyHEP WG].
- PODIO interesting for new experiments, should offer multiple backends and integration into Key4hep. [Tools and Packaging WG]. Keep developing PODIO with functionality and performance measurements, for transient as well as serialised representations. [Key4hep WG]
- Auto-diff could be an effective tool for verifying implementations of (semi-)analytical solutions more efficiently & precisely than numerical validation. [Reco and Trigger WG]

- CPU -> GPU code conversion appears not to be performant (with SYCL/Kokkos) -- launch latency in particular is bad, memory management overheads as well. Build configuration is another challenge, with diverse backends. [Tools and Packaging WG, Frameworks WG, Trigger and Reco WG]
- GPU-as-service potentially very powerful if processing time is dominated by individual steps that can be offloaded (case: LArTPC ML shower discrimination). Preprocessing on CPU is limiting?
- ACTS and sim experts should talk to each other on adapting their code for GPUs. A few possible avenues for collaboration [Reco WG, Sim WG]:
    - Handling geometry in VecGeom in a GPU-friendly manner, consistently between simulation and tracking. [Compute Accelerator Forum]
    - Does the work on GPU-based propagation code (AdePT, Celeritas in particular) carry over to charged particle transport in tracking? [Compute Accelerator Forum, Reco WG, SimWG]]