Simulation R&D PoW 2021

Witek Pokorski for the Simulation R&D team
01.02.2021
Motivation - Forecast Simulation Needs

- The upgrade to the HL-LHC for Run 4 produces a step change for **ATLAS** and **CMS**.
- The beam intensity will rise substantially, giving bunch crossings where the number of discrete proton-proton interactions (pileup) will rise to about 200, from about 35 today (2018 and foreseen for 2022)
- Accurate simulations and larger Monte Carlo samples will be needed to achieve the desired precision in physics measurements, while avoiding that simulation dominates the systematic uncertainties

**Simulation is the leading CPU consumer** for LHCb and ALICE and will continue to be in the near future

- Only with adoption of aggressive alternatives to detailed Geant4 based simulation will we be able to stay within pledged resources

See for instance:
‘Detector simulation requirements from HEP Experiments’
M. Bandieramonte
HSF/WLCG/Geant4 workshop 23rd November 2020
Approach - Three Main Axes of Development

- **Improve, optimise and modernise** the existing Geant4 code to gain in performance and precision for the detailed simulation.

- **Trade precision for performance** using fast simulation techniques both with parameterisations and with ML methods, and integrate them seamlessly in Geant4.

- Investigate the **use of ‘accelerators’** such as GPUs for performance gain.
People involved

- SFT team: Alberto, Andrei, Anna, Dalila, Gabriele, Graeme, Gunter, Guilherme, Ioana, John, Jonas, Mihaly, Pere, Placido, Predrag, Vladimir, Witek

- UK Excalibur team: Ben Morgan, Davide Contanzo, Mark Hodgkinson

- IT-SC-RD: Dirk Duellmann, Stephan Hageboeck, Stefan Roiser, Markus Schultz, Andrea Valassi

- EP-AIP: Sandro Wenzel

- University Bucharest: Adrian Petre

Most of those people split their time between numerous other projects.
Activities in 2020

- Finalization of the GeantV paper, wrapping up the R&D activity on using vectorization and other novel concepts in simulation

- Fast simulation
  - improvement to classical parameterization, Machine-Learning based fast simulation training and inference integration

- Geant4 modernization and improvements
  - ‘stateless prototype’, profiling and code modernization

- R&D on compute accelerators in HEP simulation
  - to identify the approach of exploiting GPU potential in full simulation
GeantV: Results from the prototype of concurrent vector particle transport simulation in HEP


Full detector simulation was among the largest CPU consumer in all CERN experiment software stacks for the first two runs of the Large Hadron Collider (LHC). In the early 2010's, the projections were that simulation demands would grow rapidly with luminosity increase, compensated only partially by an increase of computing resources. The extension of fast simulation approaches to more use cases, covering a larger fraction of the simulation budget, is only part of the solution due to intrinsic precision limitations. The remainder corresponds to speeding-up the simulation software by several factors, which is out of reach using simple optimizations on the current code base. In this context, the GeantV R&D project was launched, aiming to redesign the legacy particle transport codes in order to make them benefit from fine-grained parallelism features such as vectorization, but also from increased code and data locality. This paper presents extensively the results and achievements of this R&D, as well as the conclusions and lessons learnt from the beta prototype.

Comments: 34 pages, 26 figures, 24 tables
Subjects: Computational Physics (physics.comp-ph); High Energy Physics - Experiment (hep-ex)

Submission history
From: Soon Yung Jun [view email]
[v1] Sun, 3 May 2020 00:05:51 UTC (6,981 KB)
[v2] Tue, 2 Jun 2020 13:00:20 UTC (3,774 KB)
Fast simulation in 2020
General tool for energy deposition in Fast Sim

Status in Geant4:

- Currently no general tool for depositing energy.
- GFlash implementation includes models, sensitive detector, …
- They are all interlinked and connected to GFlash* classes.

New developments:

- Assume simple implementation of fast simulation models.
  - energy $E_i$ is deposited in $\overline{r}_i$, $i \in 1, \ldots N$ (no other information)
- G4FastHit binding energy deposit and its position;
- G4FastSimHitMaker that keeps track of volumes, readout geometry, and calls sensitive detector to deposit energy;
- G4VFastSimSensitiveDetector is a base class for user’s SD (additional to G4VSensitiveDetector) that includes method for processing hits based on G4FastHit (instead of the usual G4Step)
Energy deposition - application to GFlash

**Green** Energy deposited by current implementation of GFlash in Geant4.

**Red** Additionally, assign deposits to (mesh) cells and deposit accumulated energy.

- Maintaining cell sizes relatively small (wrt readout cells) should not compromise accuracy.
- But should speed up the simulation.
- Size of cells subject of optimisation by users (specific detectors).
On-going tuning of GFlash-inspired parameterisations

- GFlash starts parametrisation as soon as particle enters volume;
- But first interaction (shower start) may happen further;
- Including this shift in the longitudinal profile calculation does not impact significantly the parameters of the distribution, but certainly has impact on longitudinal profile for $t < 5X_0$;
- Needs to be implemented for parametrisation;
Deep Learning for Fast Simulation

- developed new model for calorimeter shower simulation using autoregressive network (see next slide)

- worked on full cycle system for Geant4 integration of Deep Learning utilities, from data production through inference integration and results validation
Autoregressive networks for Fast Simulation

Ioana Ifrim

The choice of our study

Explicitly tractable: AutoRegressive Models

- Complex modelling of the data distribution
- Able to encode long term dependencies between energy cells
- Training is highly parallelizable given the type of operations
- The training is stable
- Autoregressive factorisation is general: expressivity of model
- Meaningful parameter sharing has good inductive bias => good generalisation

Validation Results - 10 GeV

- Correlations between layers are kept given that cell $x_i$ is dependant on preceding $z$ layers cells, thus physics relevant dependencies between calorimeter cells are implemented
Geant4 modernization and improvements R&Ds in 2020

- activities in all Geant4 Working groups
- some examples on next slides
Stateless Geant4 prototype

• to explore benefits of instruction-level locality and vectorization, sub-step-level parallelism needs to be possible
  • to group tracks ‘doing the same thing’
  • requires Geant4 engine to be ‘stateless’ (not keeping state of the current track)

• R&D work demonstrated the necessary changes in the code to achieve this
  • removing ‘state’ information from managers, navigators and some processes and attaching it to the G4Track
  • splitting stepping into several stages
  • introducing containers for each stage

• changes are substantial, but feasible
  • to be eventually revisited

• could allow potential performance improvements of certain computational phase by locality and vectorization
  • magnetic field propagation, MSC, some physics models, etc
Investigated cost to eliminate TLS in geometry
- Key attributes of G4 parameterised physical volumes (PV) depend on the volume’s identity
  - Geant4 MT split PV & LV in two - moved changeable parts into per-thread containers.

Challenge: All relevant objects/types are user-visible: physical & logical volume
- seek to minimize interface changes & need for migration of user code

First prototype
- copy data into ‘proxy’ object & use only that. Check: “does it provide a CPU benefit?”

Answer: small extra cost in ‘global dynamic’ mode (+0.7%)
Geant4 profiling

- Created Geant4 application to run and profile simulations
  - Simple to run, single script runs the full build + performance reports
  - Easy to interpret view of performance changes to be run in continuous integration tests
  - Currently using only **perf**, but reports based on VTune also planned for near future
  - Would like to integrate also the magnetic field map, main difference with CMSSW
**G4HepEm: motivations & description in a nutshell**

- **initiated by the Geant4 EM physics working group** as part of looking for solutions to **reduce the computing performance bottleneck** experienced by the HEP detector simulation applications.
- **targeting** the most performance critical part of the HEP detector simulation applications, i.e. the **EM shower generation** covering (initially) $e^-/e^+$ and $\gamma$ particle transport.
- The main goal is to investigate the **possible computing performance benefits** of replacing the current general particle transport simulation stepping-loop of Geant4 by **alternatives, highly specialised for particle types** (i.e. for $e^-/e^+$ and $\gamma$) and **tailored for HEP detector simulations**.
- **isolates and extracts all the data and functionalities**, required for **EM shower simulation** in HEP detectors, in a very simple and **compact form**.
- This clean and **compact and well documented environment** for EM shower generation also **provides an excellent domain** for further related R&D activities.
- **G4HepEm** provides **special support for the related R&D activities targeting** EM shower simulation on **GPU devices**.
- It has been made available in order to **facilitate and catalyse correlated R&D activities** by providing and **sharing the related expertise and specific knowledge**.
Investigating Use of Accelerators

- need to efficiently use the evolving hardware
- general HEP simulation code is **not a natural candidate to run on GPUs**
  - Large complex codes, computation spread in many areas, many branches and special cases
  - Work needed to be done not known a priori (stochasticity)
- some successes on reduced and simpler problems
  - Low energy electromagnetic, medical app. with simple geometries, neutrons transport, optical photons, etc.
- there may be other alternatives (non-GPU based HPC), but GPUs are certainly widely available
  - Pressure from funding agencies to make efficient use of large HPC installations
  - We won’t get the necessary speedup by running on CPUs
- big issue on the sustainability of the code
  - No standard GPU programming language
Simulation on GPUs – R&D in 2020

● several discussions, presentations, brain-storming meetings over the first 6 months
  ○ goal: transform the very heterogenous Geant4 HEP particle transportation into a more homogenous computational problem
    ■ impractical to ‘port’ full Geant4 in its current form to GPU
  ○ discussing/learning how to deal with GPUs in some simulation-specific workflows (e.g. geometry, scheduling)

● some prototypes implemented to demonstrate specific functionalities
  ○ VecGeom-based ray tracing
  ○ ‘stateless’ Geant4
  ○ VectorFlow framework prototype

● discussed with the authors of several working applications
  ○ Neutron transport (Oak Ridge team)
  ○ Opticks
  ○ GATE

● took part in Nvidia hackathon allowing better understanding of efficient use of GPUs
  ○ strong motivation for single precision

● started a new R&D project (AdePT) aiming to demonstrate EM shower simulation on GPU
  ○ many of the initial ideas presented in the HSF workshop and Geant4 Collaboration meeting
AdePT - Accelerated demonstrator of electromagnetic Particle Transport

- Significant amount of work need to be performed in one go on the device due to high cost of transferring data between CPU and GPU

- Decided to focus on prototyping specialized GPU code to perform electromagnetic shower simulation in a calorimeter
  - Specialized set of physics models and geometry
  - Pre-defined scoring capabilities

- Geant4 would off-load simulation of EM showers to the GPU library
  - Similar concept to ‘fast-simulation’ processes, but doing full simulation on GPU
AdePT status

- started with CUDA utilities for track data handling
- implemented several toy examples of increasing complexity
  - selecting ‘physics process’ based on random number
  - dummy energy loss and pair production processes as kernels consuming queues of particles
  - running ‘shower’ of particles
- started integrating geometry (VecGeom)
  - several improvement to VecGeom code to make it GPU-aware (and efficient)
    - implemented new way of handling navigation states as indices
- implemented first version of magnetic field propagator
- started interfacing to G4HepEm library
Plans for 2021

• Fast simulation

• AdePT prototype development
Fast simulation – ‘classical’ parameterization

• continuation of tuning of parameters of Gflash-inspired models
  • start of shower dependent tuning
  • transverse shower profile tuning

• automatic tuning tools and generalization procedures
Fast simulation – ML

1. **Integration of inference into C++**
   a. Provide G4 example extending its simulation facilities to ML-based fast simulation designed with long-term maintainability
   b. [Already started]
      i. FastMLSim class integrated in G4: loads the saved ML model and simulates energy depositions in 3D coordinates.
      ii. Comparative study of existing tools (LWTNN, ONNX, TWVA, TF light) in terms of supported ML models, stability, memory footprint, inference time ...

2. **Provide detector-agnostic models for easy application and extension to detector geometries facilitating their use by various experiments**
   a. Design a ML model with predefined architecture & condition on generic parameters (energy of the truth particle, $\eta$,..) to study and track model’s performance on custom detector geometries to evaluate the changes & limitations,..
   b. This allows to better design a generic, detector-agnostic ML simulator based on the ML concept of learning to learn fast or “Meta learning”.

3. **Validation & optimization of ML models**
   a. Design of generic metric for validating the ML model performance during the optimization to automatically select the best set of model parameters.
AdePT goal for 2021

- develop a demonstrator of EM calorimeter simulation on GPU with as many realistic components as possible
  - geometry
  - main EM physics processes
  - magnetic field
  - calorimeter-specific scoring

- perform first assessment of possible speed-up with respect to equivalent CPU-based simulation
AdePT plans for 2021

- Development of a core GPU transport engine
  - Dynamic track population scheduling (dynamic data management)
  - Stepping loop split into a sequence of kernels
  - Targets: functionality (complete EM shower transport), performance (maximize occupancy, minimize memory overheads), abstraction (run on GPU and CPU)

- Concrete work items
  - Geometry and transport
    - Add missing solids on GPU
    - Improve navigation layer on GPU, add missing functionality
    - Single precision geometry prototype
    - Kernels for transport of neutral and charged particles (with integration of motion in field.)
    - Integration to transportation (abstracting the propagation layer) and connection to physics
  - Workflow implementation/optimisation
    - Types allowing dynamic track management (e.g. containers allowing parallel selections, compacting, managing sparse data)
    - Management of tracks in the stepping loop, data management (track block handling, coalescing vs. handling in-place)
    - Sustainability (handling exponentially-growing track population) via priority policies, data partitioning (e.g. by generation or energy)
Miscellaneous

- Geant4 R&D Task Force
  - coordination and contribution

- EP R&D
  - Fast simulation project on ML-based parameterizations

- AIDAinnova
  - Fast Simulation task coordination

- HSF Detector Simulation working group
  - coordination (until December 2020) and contribution
Summary

- R&D on simulation software essential to meet the HL-LHC (and post) requirements
- three axis of further development
  - improve and modernize Geant4
  - Fast simulation R&D
  - compute accelerators usage in simulation
- major goals for 2021
  - fast simulation ML-based generalized prototype
  - AdePT: GPU-based EM calorimeter simulation prototype