Simulation Code Modernization

Witek Pokorski

EP-SFT, CERN

20.03.2019 HOW

Based on input from several experiments and projects.

Overview

- Motivation
- Experiments status and plans
- Simulation toolkits status and plans
- Conclusion

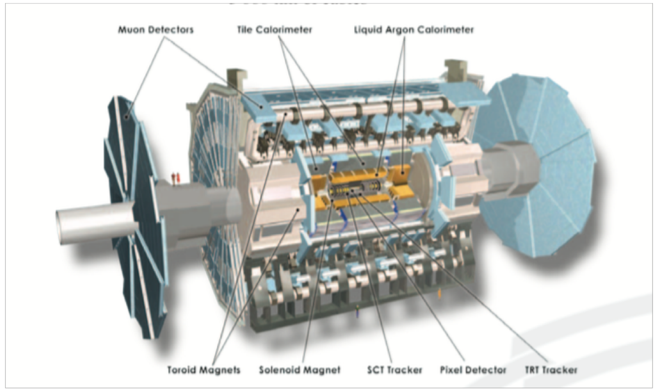
Motivation

- future accelerators (HL-LHC, FCC) experiments need a large speed-up in detector simulation (one of dominant CPU-time consumers)
    - requirement of at least an order of magnitude speed-up in simulation (more to simulate pile-up)
        - HSF Community White Paper
            - https://arxiv.org/pdf/1712.06982.pdf
            - https://arxiv.org/pdf/1803.04165.pdf

- we need
    - better algorithms
    - better code
    - efficient use of current (and future) computing architectures
- **we need to modernize our code!**

Experiments status and plans

ATLAS



- starting to use multi-threaded capabilities of Geant4
  - running in the multithreaded version of their general software framework (AthenaMT)
  - validation against single-threaded version successful
  - would appreciate if Geant4 event loop could be opened up a bit more for user interaction and different classes more suitable for inheritance (protected members)
- looking at the possibility of running their stand-alone fast-simulation code (FastCaloSim V2) on GPUs
- exploring the usage of Machine Learning techniques (Variational Auto Encoders and Generative Adversarial Networks) for fast calorimeter simulation

John Chapman, Heather Gray

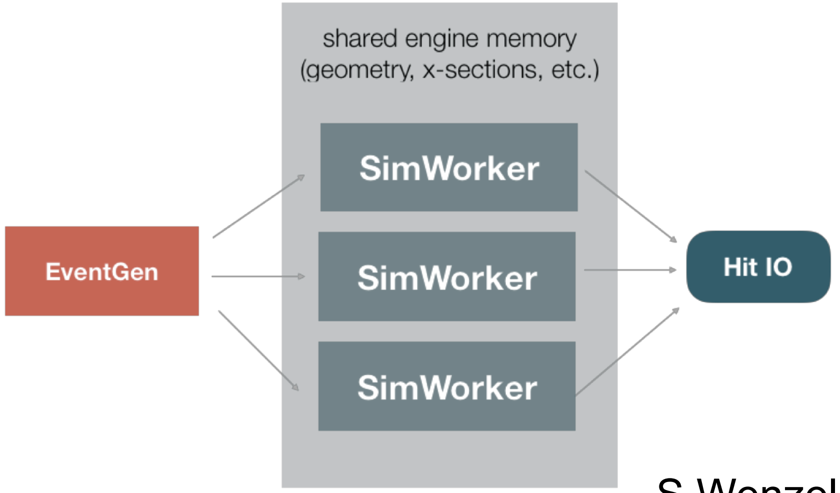
A Large Ion Collider Experiment

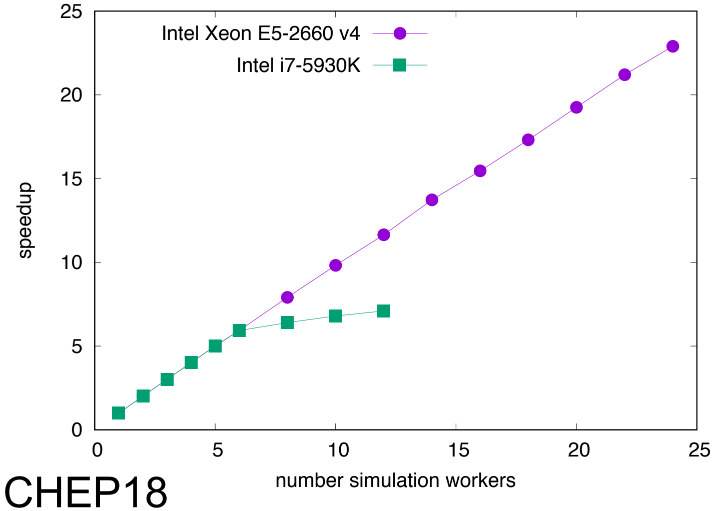
ALICE
Parallel high-performance simulation framework

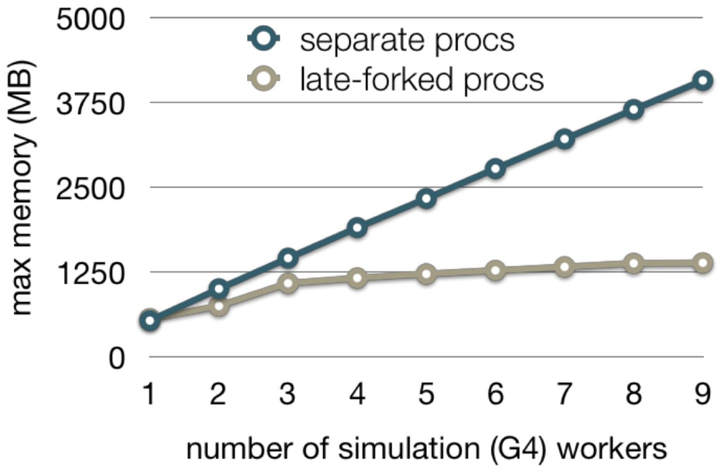
Sandro Wenzel

- Development of a scalable and asynchronous parallel simulation system based on independent actors and FairMQ messaging
- Supports parallelization of simulation for any VMC engine
- Supports sub-event parallelism
  - Make simulation jobs more fine-granular for improved scheduling and resource utilization

- Demonstrated strong scaling speedup (24 core server) for workers collaborating on few large Pb-Pb event
- Small memory footprint due to particular ”late-forking” technique (demonstrated with Geant4)
- In result, reduce wall-time to treat a Pb-Pb events from O(h) to few minutes and consequently gain access to opportunistic resources





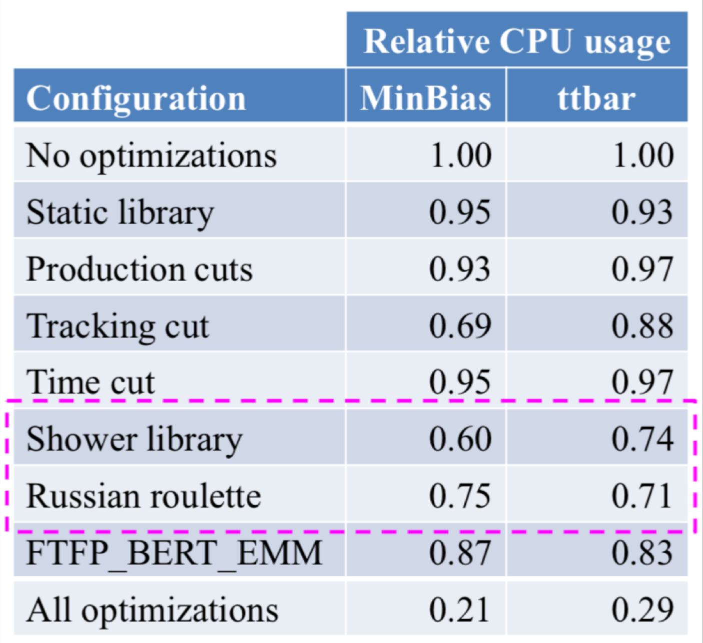


S.Wenzel @ CHEP18

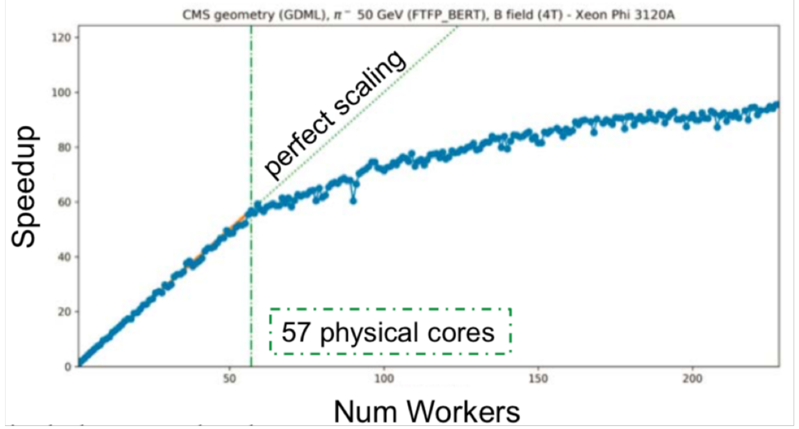
CMS

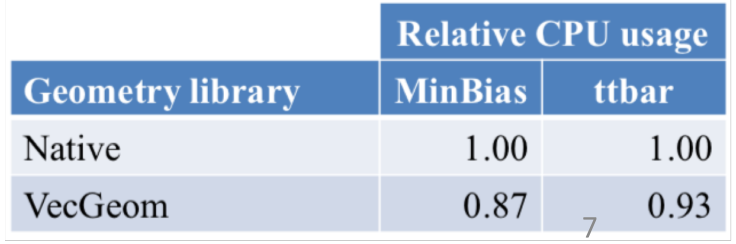
- several improvements implemented giving up to x5 speed up
- Geant4 includes event-level multithreading
  - CMSSW framework supports multithreading
- CMS switched to using VecGeom (in scalar mode) for the Geant4 geometry
- ongoing effort to develop and test some of the elements involved in the integration of the GeantV transport engine within CMSSW
  - aim to run the CMS detector simulation on heterogeneous computing and new architectures
  - see Kevin's presentation tomorrow
- following efforts within the Exascale/Geant project in the US with ATLAS and CMS participation to study a Geant4 proxy for opportunities for effective use of accelerators in the context of HPC systems

Daniel Elvira, Kevin Pedro (FNAL),
Vladimir Ivantcheko (CERN)



	Relative CPU usage	
Configuration	MinBias	ttbar
No optimizations	1.00	1.00
Static library	0.95	0.93
Production cuts	0.93	0.97
Tracking cut	0.69	0.88
Time cut	0.95	0.97
Shower library	0.60	0.74
Russian roulette	0.75	0.71
FTFP_BERT_EMM	0.87	0.83
All optimizations	0.21	0.29

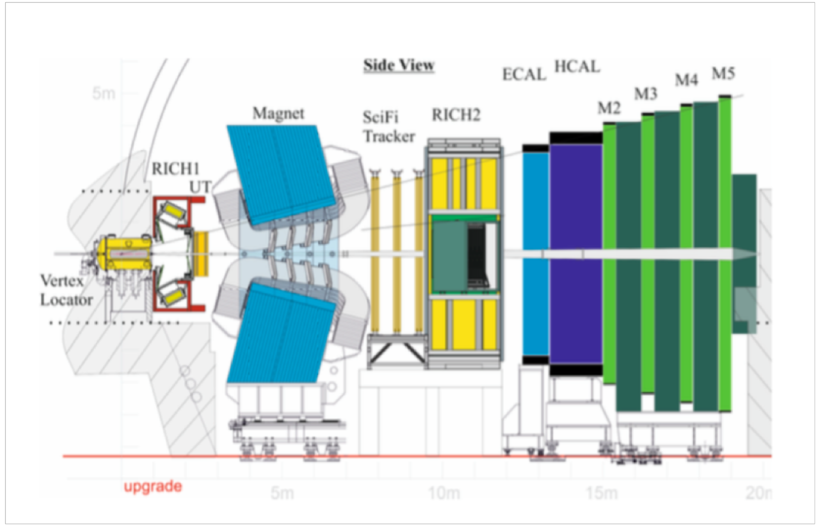


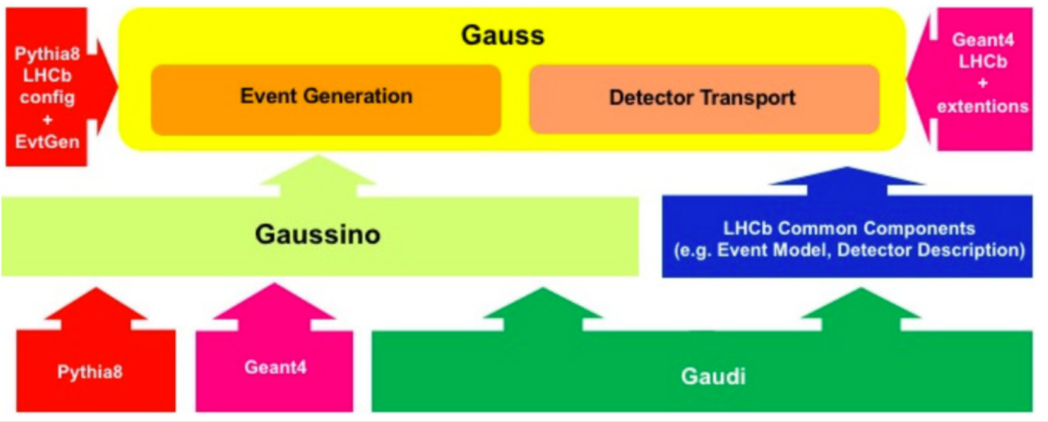


	Relative CPU usage	
Geometry library	MinBias	ttbar
Native	1.00	1.00
VecGeom	0.87	0.93

LHCb

- For LHC Run 3, updating (rewritting) large parts of the code base necessary to run the software trigger during the data taking
  - his context, also upgrading GAUSS, our simulation software framework
  - cornerstone of this upgrade effort is the full utilisation of multi-threading
  - implementing the interface to the multi-threaded version of Geant4
    - based on new framework Gaussino
- Concerning HPCs, studying the performance of the current Gauss version (using Gaudi multiprocessing) on KNL chips
  - no plan for any dedicated changes in the code
- currently no plan to investigate use of GPUs for simulation
  - with the exception of potential application to ML-based fast simulation





Gauss

Event Generation

Detector Transport

Pythia8 LHCb config + EvtGen

Geant4 LHCb + extentions

Gaussino

LHCb Common Components (e.g. Event Model, Detector Description)

Pythia8

Geant4

Gaudi

Gloria Corti,
Dominik Muller

Doris Y. Kim

Belle II

- Core of the Belle II simulation library is Geant4.
  - The simulation library is stable and ready for Phase III of SuperKEKB/Belle II, started just a few days ago.
- Next objective is optimizing the performance, to fit it into the limited computing resources.
  - The Geant4 experts prepared a new physics list tailored for the Belle II needs, which runs at a lower energy than LHC. (V. Daniel Elvira's talk)
  - The Geant4 will be upgraded from version 10.1 to version 10.5 to utilize various new geometry features.
- Geant4 multi-thread option is not turned on yet
  - basf2 has its own multi-process mode based on forking
- Various deep learning strategies are being tested to boost performances.
  - Filter generated events to prevent unnecessary events from being simulated. (Currently less than 10% of generated physics events survive reconstruction and skimming processes.)
  - Create beam background hits by GAN, to reduce disk space used to store background hits. (For example, for vertex detectors)
- May have to come back to the multi-thread / multi-processing later and see how to improve, but not on the near future agenda

Neutrino experiments

Heidi Schellman

- DUNE: in the formative stage
  - great advantage that detectors are big vats of a single material, which makes the Geant4 geometry part much simpler, leading to pretty nice efficiency in generation
  - looking at:
    - Simulation of TPC behavior - electric field, space charge, fluid flow, electron lifetime - turns out having a charged liquid flowing in the detector is challenging
    - Simulation of scintillation and light propagation in noble liquids
    - Handling large shower and photon libraries in a heterogeneous environment

Summary of experiments input

- Full simulation using multi-threading as current (near future) ‘production mode’
  - ALICE (and Belle II) using multi-process framework with ‘late forking’ and messaging system
- new geometry library (VecGeom) demonstrates how modern code (and internal vectorization) can help
  - although in case of geometries dominated by simple solids (like LHCb) the gain would come from navigation and not just solids
- Fast simulation is (very) seriously taken into account by all the experiments (can’t survive without it)
  - see Vince’s talk
- GPUs fit naturally in conjunction with Machine Learning techniques being explored in the context of fast simulation
  - not yet possible to use in full simulation for HEP use-case, but efforts ongoing

Simulation toolkits status and plans

(Some of the) Geant4 developments

- Ongoing development in all Geant4 Working Groups
  - from simple technical improvements to new architectural choices
- Few selected topics on next slides
  - Sub-event parallelism
  - Task based Geant4
  - VecGeom navigation
  - Transportation and physics ‘framework’ review
- Much more in different Working Groups plans on the Geant4 web

Sub-event parallelism – short term

- Sub-event parallelism generalizes Geant4-MT event parallelism approach to serve the case of applications requesting large memory per event.
- One event is split into "sub-events", e.g. each few primary tracks = a sub-event
  - Split method is obviously user dependent.
- Each sub-event is sent to a worker thread, and merged back to the original full event later.
  - No communication over threads required while processing a sub-event.
- Constraint – all the current API's must be preserved.
- Time scale of the development :
  - first prototype by May 2019

Makoto Asai (SLAC)

Sub-event parallelism (and more) – longer term

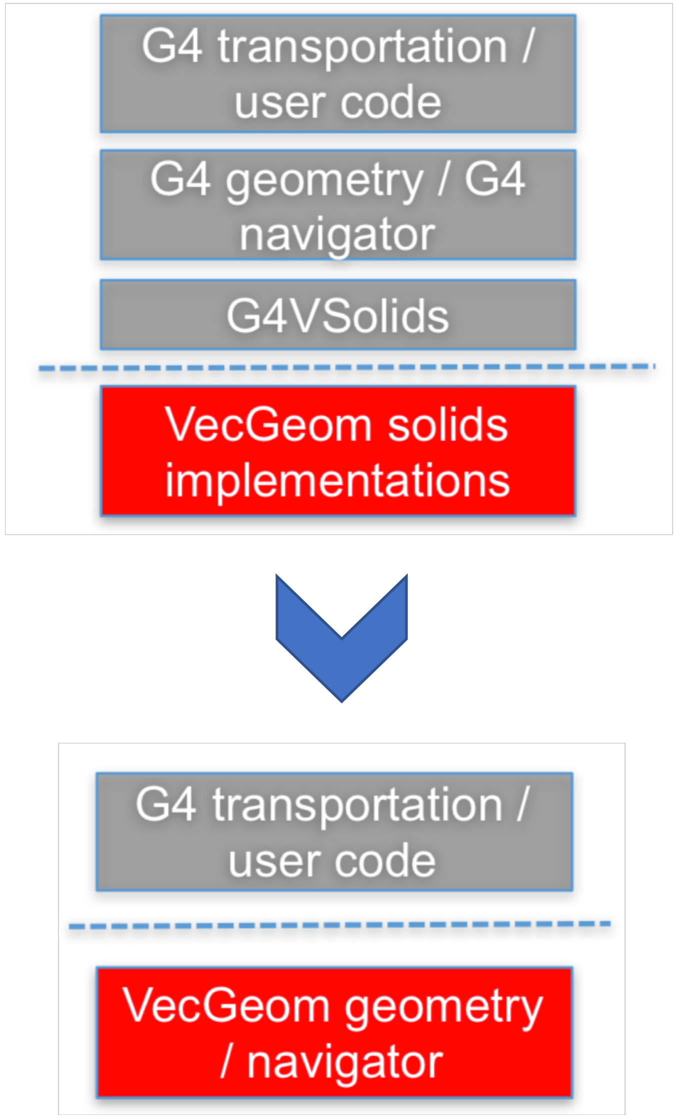
- New “tasking” framework for Geant4
  - Pool of threads without a predefined call-stack
  - Tasks are essentially function calls that are placed in a queue
  - Threads in pool are idle until tasks are placed in the queue
  - When the queue is empty, threads go back to sleep
- Naturally allows fine-grained parallelism
  - Every G4Track could a be ‘task’
  - Could play a critical role in how Geant4 could leverage the GPU
    - one or more thread-pool instances per master-thread, “task groups", and CUDA streams
      - balanced workload between the CPU and the GPU
      - nearly-perfect scaling on the CPU is maintained in addition to computation boost from the GPU

Jonathan R. Madsen
✉ jrmadsen@lbl.gov
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

VecGeom navigation in Geant4

- VecGeom is evolution of G4/TGeo/Usolids geometry with the goal to
  - use SIMD acceleration as much as possible (multi-particle API, single-particle API)
  - modernize / revise / optimize algorithms in general
- Two main components provided for use in simulation:
  1. elementary and composite **geometry primitives** (box, tube, ...) ✓
     - distance / containment / etc algorithms
     - bricks to build complex geometries
  2. geometry modelling + **navigation** ⬅
     - "fast" determination of the next (straight) line intersection of a ray and the distance
     - determination of next "geometry path" at boundary traversal
- Two options for interfacing navigation:
  1. User friendly, easy-to-implement, but "some overhead" option:
     - Simultaneous existence of Geant4 and VecGeom geometry with necessary synchronization/translation of states/objects
  2. No overhead, full integration but much more complex and potentially user-disturbing
     - Complex changes required in VecGeom and in Geant4; API and type evolution, more abstraction layers, ...

under development


G4 transportation / user code
G4 geometry / G4 navigator
G4VSolids
VecGeom solids implementations
G4 transportation / user code
VecGeom geometry / navigator

Sandro Wenzel

Geant4 transportation and physics ‘framework’ review

- Transportation
  - Currently, only one transportation object exists in the memory
    - It deals with all particle types:
      - neutral and charged particles, optical photons, phonons, etc
  - Idea is to provide at least two flavors of transportation that co-exist:
    - One for charged particles, one for neutral particles
    - Eventually one also for optical photons
      - As velocity calculations differ from other particles

Makoto Asai,
Gabriele Cosmo

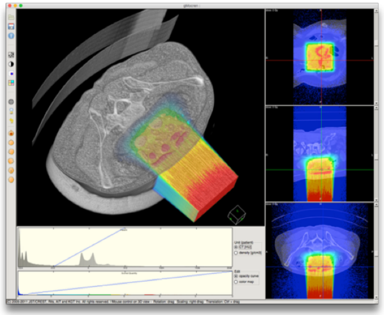
- Physics ‘framework’
  - Investigate the possibility of using a more lightweight physics framework in Geant4
    - simpler and streamlined design
    - less virtual calls
    - potential gains from data and instruction locality

Mihaly Novak,
Alberto Ribon

Geant4 and HPCs

- Create “official” Geant4 docker repository
  - e.g.,https://hub.docker.com/r/geant4
  - Tags for compilers, versions, etc.
  - Use this as standard for per-machine performance comparison
- Create a public CDash dashboard at NERSC: cdash.nersc.gov
  - Built with Docker and deployed in Spin – a containers-as-service (CaaS) platform at NERSC
- Use Geant4 + TiMemory + CDash for performance characterization with respect to HPC system, compiler, OS
  - Nightly and continuous performance testing
  - Separate from our build testing – avoid adding more clutter
  - Longer-term additions:
    - Valgrind reports
    - Code coverage
    - VTune reports (e.g., attach VTune summary as CTest note)

Jonathan R. Madsen
✉ jrmadsen@lbl.gov
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory



The MPEXS Project

(NOT OPEN SOURCE)

A state-of-the-art radiation simulator running on **GPU** devices

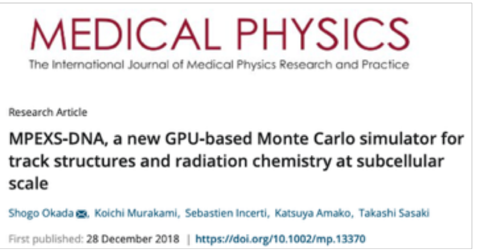
- Developed as a dose calculation engine for radiotherapy
  to improve the calculation time drastically. (*Application-oriented*)
- Ultra parallel tracking mechanism in GPU threads : *N x 100k of threads concurrently running*
- Efficient usage of modern computing architecture : **Parallel Computing**

The core algorithms and physics data taken from Geant4

- Geant4 Standard EM physics processes are reengineered and reimplemented in **CUDA**.
- Data structure is redesigned from scratch to suite for GPU processing.
- No usage of GPU-offloading and automatic code translation

Current functionality:

- Standard EM physics processes for electrons and photons below 1 GeV
- Water equivalent material
- Transportation in voxelized geometry



MEDICAL PHYSICS

The International Journal of Medical Physics Research and Practice

Research Article

MPEXS-DNA, a new GPU-based Monte Carlo simulator for track structures and radiation chemistry at subcellular scale

Shogo Okada, Koichi Murakami, Sebastien Incerti, Katsuya Amako, Takashi Sasaki

First published: 28 December 2018 | https://doi.org/10.1002/mp.13370

The joint project among:

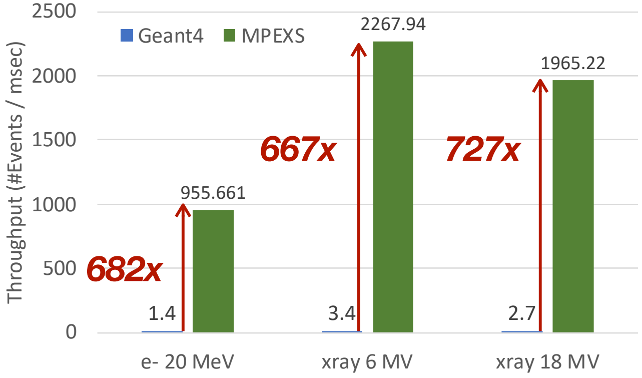
doi:10.1002/mp.13370

MPEXS Performance

Up to ***730x faster*** than Geant4 simulation with single-core CPU

NVIDIA TITAN V GPU for ***MPEXS*** vs.

Intel Xeon E5-2643 v2 (3.5 GHz) for ***Geant4***



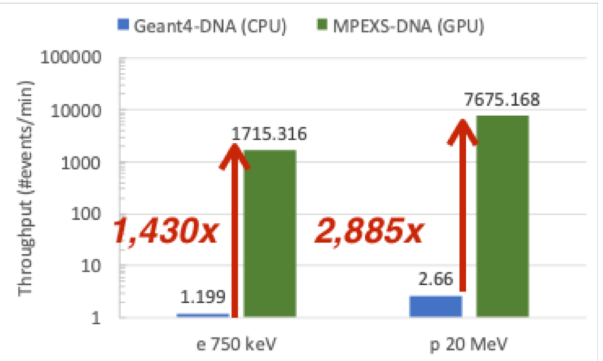
MPEXS Physics Extensions

MPEXS-DNA

- Could facilitate microdosimetry simulation and contribute scientific achievement in radiobiology

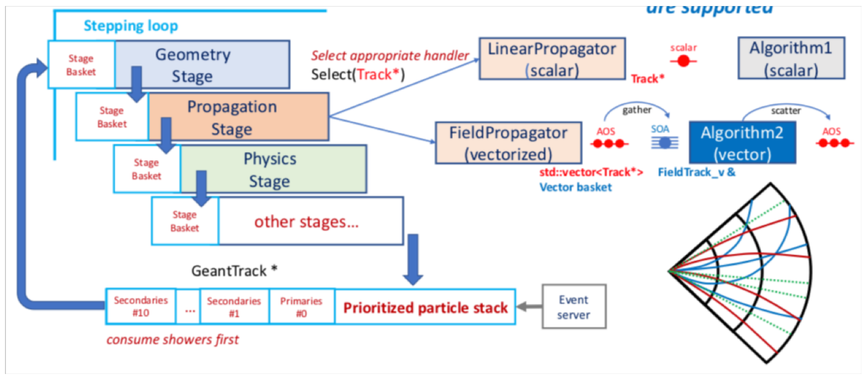
MPEXS-h

- Hadron physics extension for proton and carbon therapy
- Prototype : *30x faster* realized



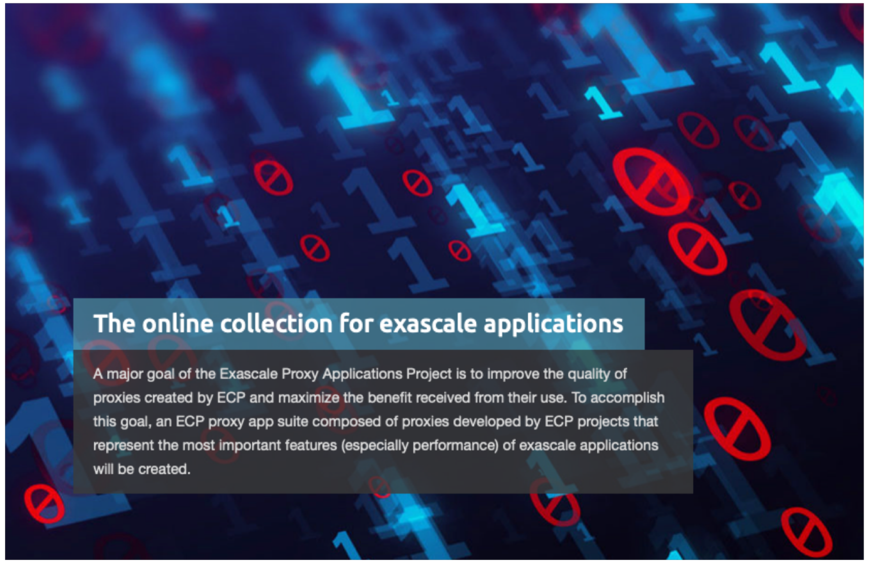
Takashi Sasaki, Koichi Murakami, Shogo Okada

GeantV R&D (see Andrei’s Gheata talk)



- GeantV: performance study for a vector simulation workflow
  - An attempt to improve computation performance of Geant4
    - Track-level parallelism, “basket” workflow
    - Improving instruction and data locality, leverage vectorization
  - Making simulation components more portable and vector friendly
    - VecGeom: modern geometry modeler handling single/multi particle queries
    - New physics framework, more simple and efficient
    - VecCore, VecMath: new SIMD API, SIMD-aware RNG and math algorithms
- in its final phase, making performance measurements and identifying the most promising aspects to work on to improve Geant4
  - Performance to be recovered by library restructuring (better fitting caches)
  - Code simplification: physics framework and step management
  - Better compromise between data and instruction locality, by adopting basket- like workflow in certain areas

Geant Exascale Proxy



The online collection for exascale applications

A major goal of the Exascale Proxy Applications Project is to improve the quality of proxies created by ECP and maximize the benefit received from their use. To accomplish this goal, an ECP proxy app suite composed of proxies developed by ECP projects that represent the most important features (especially performance) of exascale applications will be created.

- https://proxyapps.exascaleproject.org/
- Main objective
  - Identify [further] re-engineering opportunities that will increase event throughput by improving single node performance and being able to make efficient use of of the next generation of accelerators available in exascale facilities.
- Approach
  - proxy application to be a "workload/runtime emulator” for simulation
  - provide a context for exploring new off-loading methods for MC particle transport in general while minimizing the possibility that the offloading concepts won’t scale in the full complexity of Geant4

Philippe Canal,
Jonathan Madsen

Geant4 Task Force for R&Ds

- new Task Force created to promote and survey research activities on on the exploitation of emerging technologies, computing architectures or software architectural revisions that would be beneficial to Geant4
- starting now
  - brainstorming meetings, collecting ideas, topics, ongoing projects
  - form groups to assess and prepare concrete proposals for specific R&Ds
  - quick prototypes to demonstrate feasibility and benefits
  - implement production code and merge..

Conclusion

- experiments are moving (sooner or later…) to new solutions offered by the toolkit developers
    - they have no choice… they need it…
- several different R&Ds for simulation toolkits ongoing, but maybe too disjoined?
- new technologies available on the market (HPCs, GPUs, Machine Learning) but we seem not to be using them so far, why?
    - not adapted to our use case?
    - or are we too slow with modernizing our code…?
- seems that we (toolkits developers) are dragging a bit…?
- essential to have common, dedicated effort on simulation R&Ds
    - agile development with quick prototyping and testing of ideas
    - implementation of the successful ones in the production
- personally, I see it extremely positive to have the new R&D Geant4 Task Force
    - really looking forward to a lot of new, interesting developments