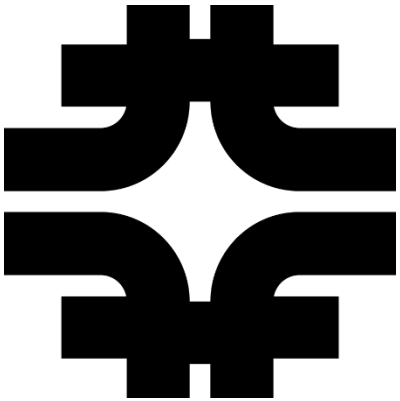
ML Inference Integration in CMS

Kevin Pedro (FNAL)
on behalf of the CMS Collaboration

November 23, 2021



CMS Simulation Status

- CMS FullSim is *4–6× faster* than baseline Geant4
  - Numerous technical optimizations & physics-preserving approximations
  - Sustained effort to commission and adopt new Geant4 versions
- CMS FastSim application: *10–20× faster* than FullSim
  - Includes sim- and reco-level optimizations (tracking)
  - Currently used for generation of large supersymmetric model scans, some studies of systematic uncertainties

	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
VecGeom (scalar)	0.87	0.93
Mag. field step,track	0.92	0.90
All optimizations	0.16	0.24

Eur. Phys. J. Web Conf. 214 (2019) 02036

ML for CMS Simulation

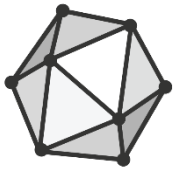
- Well-positioned for Run 3, but further acceleration crucial for Phase 2
  - HGCal full simulation expected to be ~3× slower because of more complex geometry and more precise physics
  - May deploy some ML prototypes for specific use cases during Run 3
  - Provides natural avenue to utilize *heterogeneous computing resources*
- In this talk: discuss general/relevant considerations for ML integration

Relative CPU usage

EMN: current state of art physics list		**Run 2**	**Run 4 (range)**	
Minimum Bias	**FTFP_BERT_EMM**	1.00	1.18	1.24
(10.5.ref08)	**FTFP_BERT_EMN**	1.06	2.01	2.15
ttbar	**FTFP_BERT_EMM**	1.00	1.64	1.75
(10.5.ref08)	**FTFP_BERT_EMN**	1.14	2.97	3.25

Eur. Phys. J. Web Conf. 245 (2020) 02020

ML Frameworks in CMS

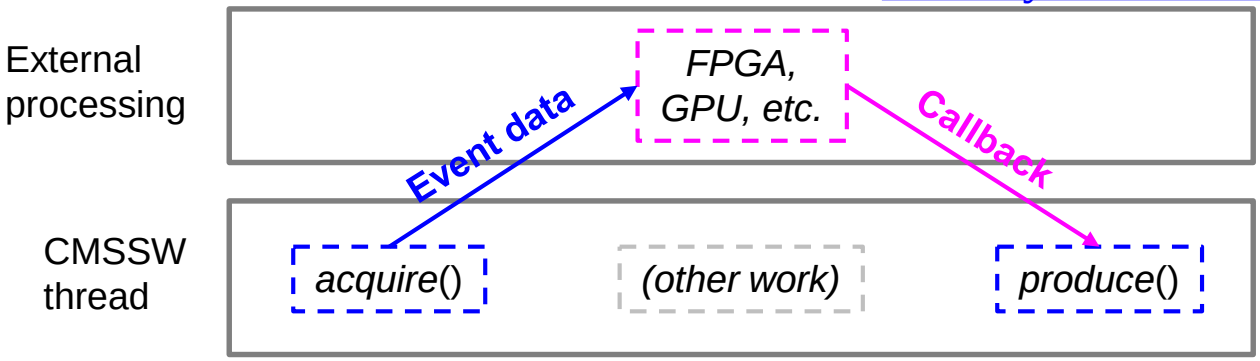



- CMSSW currently includes C++ APIs (as externals) for:
  - TensorFlow, ONNX, LWTNN, TMVA (in ROOT)
  - Ongoing work to add PyTorch
- Maintenance burden for these externals is high:
  - Difficult build tools (especially Bazel)
  - Complex and inflexible build configurations
    - Takes significant work to use existing versions of external dependencies instead of reinstalling everything from scratch
  - Thread-safety, etc. problems are common
    - External maintainers may not accept contributed solutions
- Significant work in performance optimization for CPU inference, e.g.:
  - Use OpenBLAS rather than GSL CBLAS
  - Enable multi-vectorization to support multiple instruction sets
  - Reduce memory usage by sharing some objects
  - CMS custom BDT reader is ~5× faster than TMVA

Coprocessors

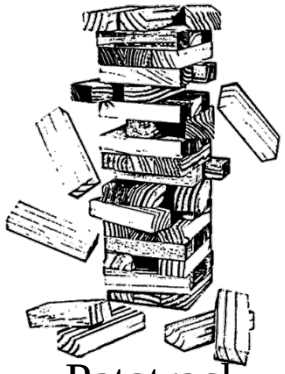
- Two approaches to utilize coprocessors: direct connect or as a service
  - Both use ExternalWork: asynchronous, non-blocking task-based processing (below)

Eur. Phys. J. Web Conf. 245 (2020) 05009



Direct Connect

- Use algorithms written in CUDA or compatibility libraries (Kokkos, Alpaka)
- Will be deployed for Run 3 HLT (non-ML)
- Challenge: integrating external software

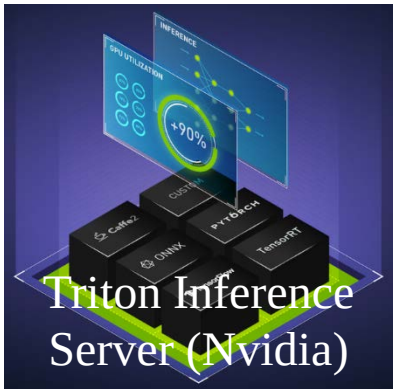


Patatrack

arXiv:2008.13461, arXiv:2104.06573

As a Service

- Connect to remote (or local) GPUs running on separate server
- Advantages: flexibility, portability, containerization of ML fwks
- Challenges: network usage, complexity



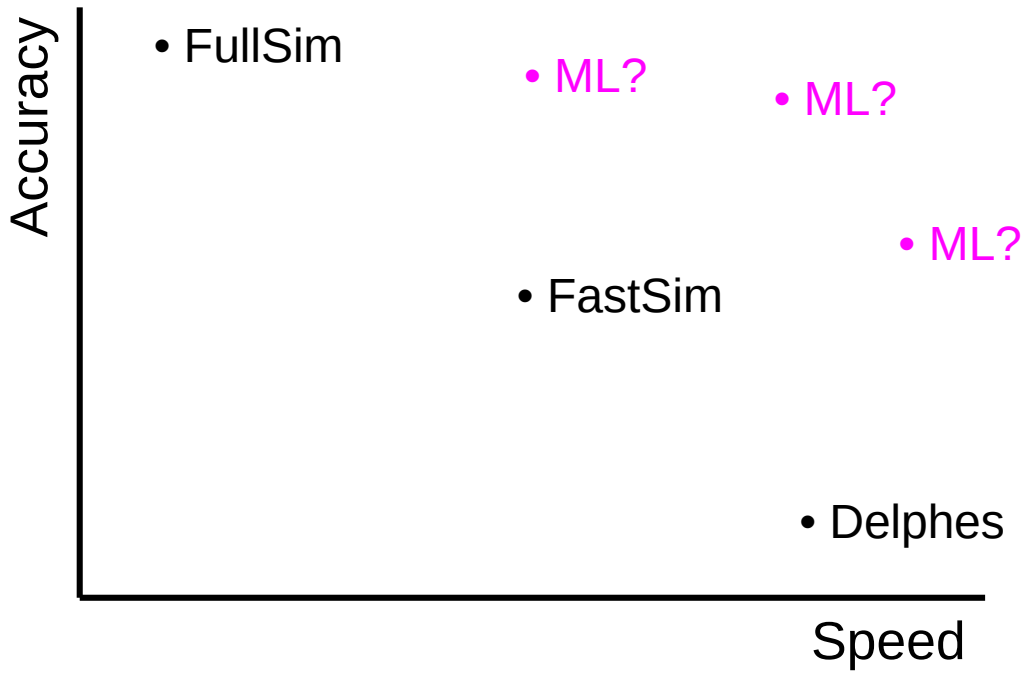
CSBS 3 (2019) 13, Proc. H2RC (2020) 38, Front. Big Data 3 (2021) 48, MLST 2 (2021) 035005

ML + Coprocessors

- TF & ONNX direct inference on GPU: now supported in CMS software!
  - Improving CUDA compatibility support based on experience w/ HLT
- hls4ml can translate BDTs, DNNs, etc. into FPGA instructions
  - Planned for use in CMS L1 trigger for Run 3
  - Also explored: Xilinx ML Suite, Microsoft Brainwave, Intel OneAPI, …
  - Currently used outside of CMS software
- As a Service approach being tested in CMS (also protoDUNE, LIGO, …)
  - SONIC (Services for Optimized Network Inference on Coprocessors)
  - Supports TF, ONNX, PyTorch, FIL, custom backends
  - Also provides dynamic batching, load balancing, etc.
  - Can be used w/ CPUs, GPUs, FPGAs, IPUs, and more

Simulation Considerations

- Numerous ways to apply ML for simulation:
  - Replace Geant4
  - Augment Geant4
  - Augment FastSim
  - Replace entire chain ("end-to-end")
- Flexibility is paramount
  - Make it easy to test and benchmark different ML applications
  - Important for Geant4 to engage in co-development w/ experiments to ensure compatibility (as with other R&D projects)



Conclusions

- CMS is developing more advanced infrastructure to support ML
  - Many frameworks, coprocessors, optimizations, offloading methods, etc.
- Several ongoing projects to use ML for simulation
  - Aimed at Run 4 / HL-LHC: HGCal is primary motivation
- Offloading substantial portion of detector simulation computing to ML: most promising approach to utilize next-generation high performance computing (HPC) centers, which will have various GPU or other coprocessor resources
- Look forward to co-development with Geant4 (and other experiments) to deploy these new approaches

Backup