CMS AI Infrastructure Needs

Kevin Pedro (FNAL)

April 6, 2023

CMS AI Workflows

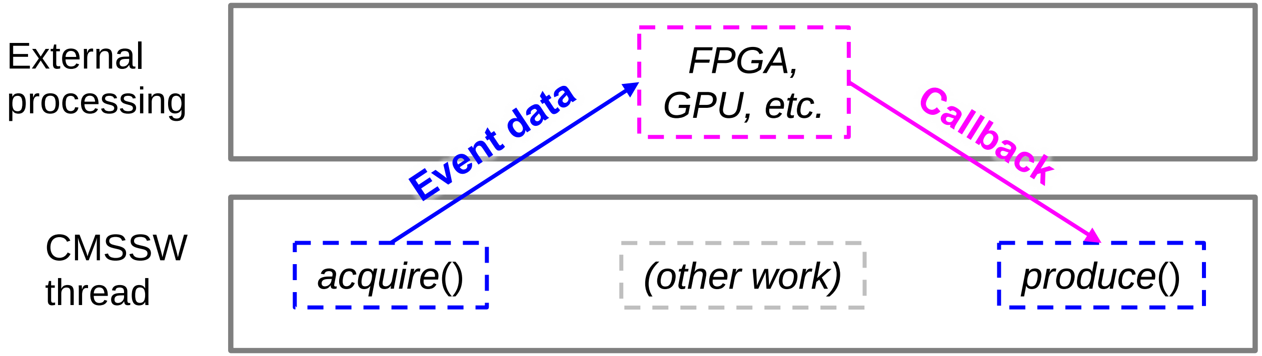
1. Training
   - Performed by individuals/small groups
   - Uses ML framework (TensorFlow, PyTorch, etc.)
   - Often compute- and memory-intensive
     - May need long batch reservations etc.
2. Inference
   a. Production: in CMS software
      - Usually on CPU (slow); GPU just being implemented
      - Alternative: as-a-service using SONIC & Triton Inference Server

   b. Analysis: in Python etc.
      - Usually on CPU (slow)
      - Growing interest in Triton Inference Server (often w/ coffea)
   - Need to use *batching* for efficient utilization of GPUs
   - Triton/aaS: ethernet connectivity & bandwidth requirements

(brief summary; more detail in 2022 Institutional Cluster Acquisition Planning Committee Report)

CMS AI in Production

- Training: currently no centralized handling of workflows etc.
  - Probably needs to change in the future as AI becomes more important
- Inference:
  - Mainly CPU-based inference for (mostly) relatively small models
  - Direct (local) GPU inference with ONNX and TensorFlow being tested
    - Integrating & supporting multiple ML frameworks is a pain point
  - Arbitrary (local or remote) GPU inference available via SONIC (Services for Optimized Network Inference on Coprocessors) inference-as-a-service
- All GPU access in CMS software relies on ExternalWork: asynchronous, non-blocking, task-based processing (built on Intel TBB)
  - CPU and coprocessor work simultaneously: minimize impact of latency



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

CMS Computing Resources @ FNAL

cmslpc
78 login nodes
2 "heavy" dev nodes
~5K batch nodes
3 P100 GPUs (interactive)

EAF
? login nodes
2 A100 GPUs (interactive)

ailab
3 login nodes
3 T4 GPUs (interactive)

Wilson
2 login nodes
100 batch nodes
10 P100 GPUs (batch)
12 V100 GPUs (batch)
4 A100 GPUs (batch)

Other resources:

- Personal GPUs (usually consumer-level, e.g. 2080 RTX Super)
- University GPUs: Grid access (CMS Connect or CRAB)
- CERN: cms-ml docs
- HPCs: Argonne, etc.; need a proposal/allocation
- Cloud GPUs:
  - AWS, GCP, Azure: usually require credits or $$$
  - Google Colab: free K80, paid T4/V100/A100

CMS AI Software Resources @ FNAL

- Software environments:
  - LPC: TensorFlow & PyTorch containers (GitHub, DockerHub)
    - Maintained by KJP (created by Alexx Perloff)
      - Updated infrequently…
    - Converted to Apptainer & synced to cvmfs by unpacked
  - EAF: GPU notebook image
    - Maintained by Burt Holzman (?)
  - Wilson: use of Apptainer recommended
- CUDA & drivers:
  - Maintained/updated by system administrators
    - Not necessarily consistent version or frequency of updates across different resource hubs
  - Datacenter GPUs have forward compatibility
    - Not universal
    - Are compatibility drivers always provided?

Opportunities for Improvement (1)

- Expand GPU resources
  - 34 GPUs provided by FNAL (from T4 to A100)
    - Some dedicated to CMS, others shared across lab
    - **None** dedicated to *FNAL CMS*: LPC resources shared by ~200 active users from US & international universities
  - In AI research, results $\propto$ money
    - Bigger networks, more data, longer training
    - Examples:
      - Nvidia StyleGAN3: 92 V100 GPU-years, including exploration
      - DALL·E 2: 23 V100 GPU-years just for one training
      - Stable Diffusion: 17 A100 GPU-years just for one training
      - CaloScore: 16 A100 GPUs (@ Perlmutter) for a one-off paper
- Better handling of interactive ("wild west") vs. batch usage
  - Exploration/experimentation vs. long trainings
  - Enforcement of fair share, priority, etc.

Opportunities for Improvement (2)

- Disk access/interaction
  - CMS users keep source files on EOS
    - Not directly readable through ML frameworks
      - Preprocessing often required to change data formats
      - Maybe solved with fsspec-xrootd, but not widely used yet
    - Slow to read over xrootd (IO-bound training → poor GPU utilization)
- Hyperparameter scanning
  - Example: scan for ParticleNet (DGCNN), w/ just 96 hyperparameter variations, takes >1 week (close to 2 weeks) to run on Wilson cluster
  - Can be improved using distributed frameworks like DeepHyper or Optuna
- Larger-scale training w/ multiple GPUs
  - Also needs some kind of distributed framework (& associated support)
    - e.g. MLFlow, Kubeflow, determined.ai; establish a standard, scalable solution across the lab
- Clearer delegation of maintenance & documentation responsibilities
  - "User-centric" docs very important to make facilities accessible
  - Possible to reduce duplication of effort w/ standardization?

Conclusion

- In AI research, results $\propto$ money
- FNAL leadership in AI requires investment
  - In both hardware and personpower/processes
- More GPUs please! (Maybe even an H100?)
  - Easier access to remote GPUs? (grid, HPC, cloud, etc.)
    - From a pure research perspective: perhaps most cost-effective approach
    - From a funding opportunity perspective: DOE frowns upon proposals that request $$$ for resources the lab is “supposed” to have already

➢ If FNAL wants to be a leader in AI, need to invest in AI facilities

Backup