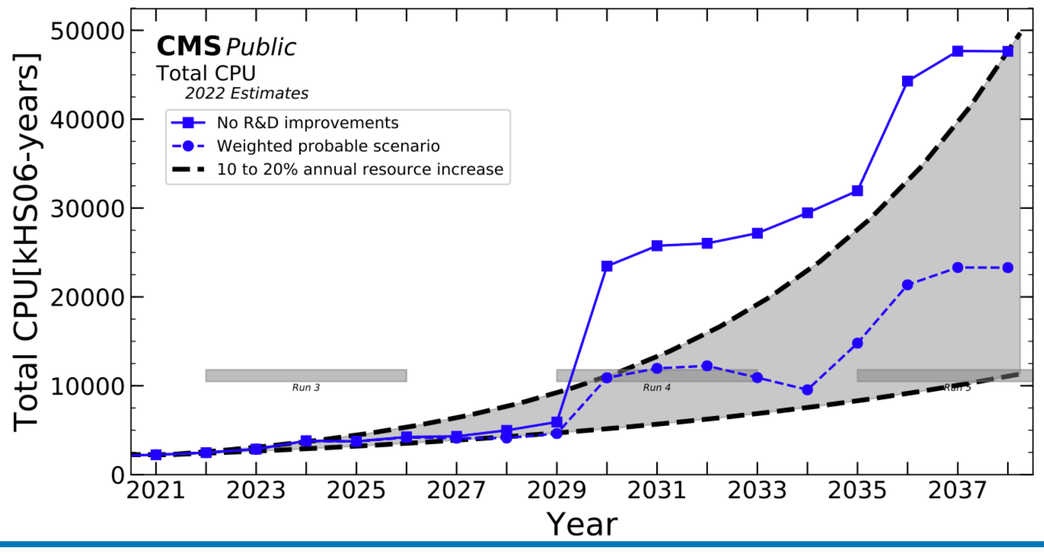
Portable Acceleration of CMS Computing Workflow with Coprocessors as a Service

Yongbin Feng (Fermilab)
on behalf of the CMS Collaboration

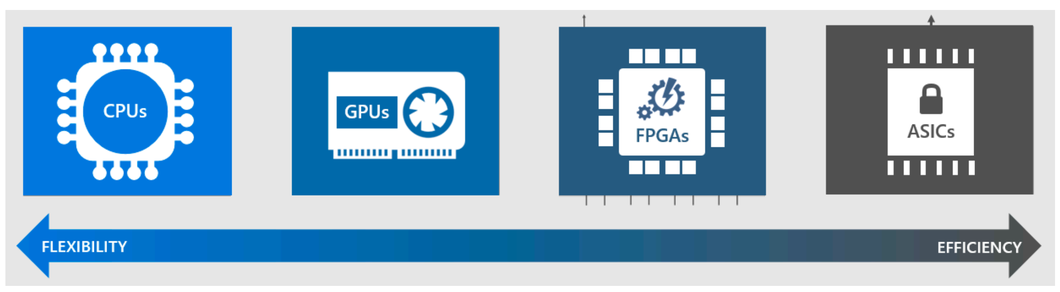
Computing Demands

- Large computing demands for HL-LHC but limited CPU performance increase

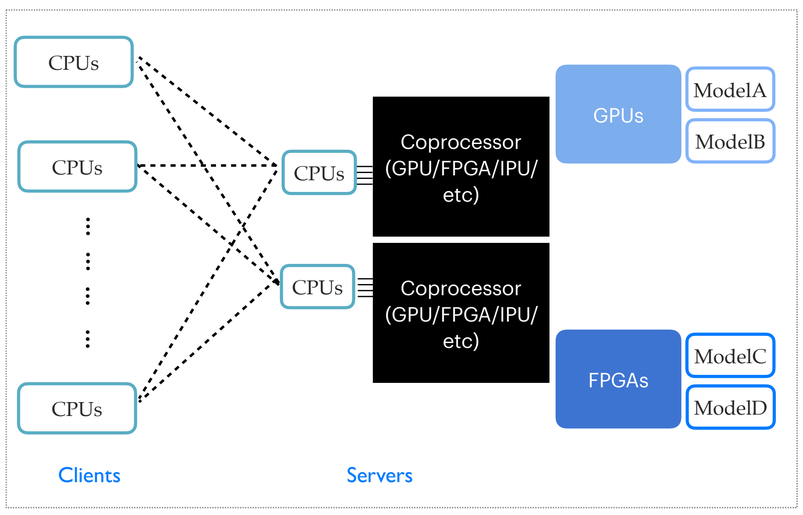


Variety of Coprocessors

- Fast developments in industry on different types of coprocessors
- Each type of coprocessor has its own uniqueness and is suited to process certain types of processing tasks

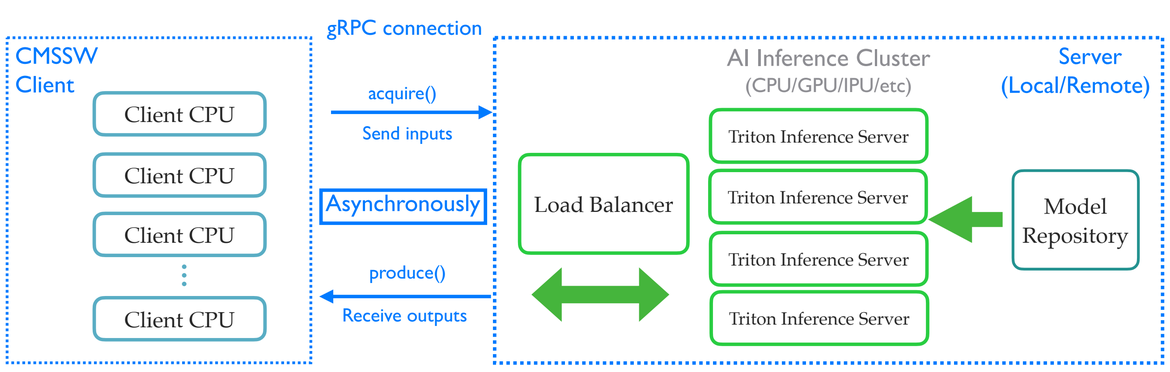


- Explore ways to easily, flexibly, and efficiently use different types of coprocessors for HEP data processing
- As a service (aaS) approach provides us such a solution



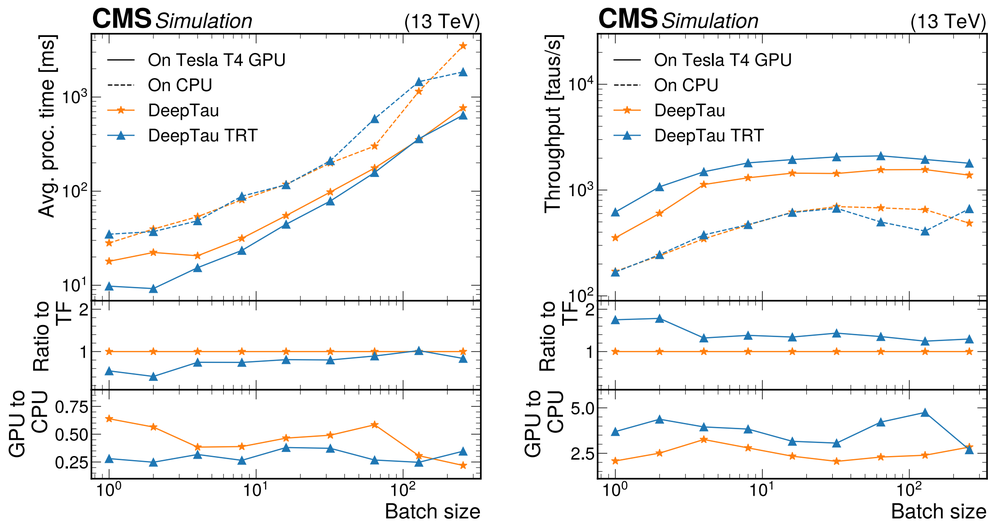
CMS Implementation

- CMSSW clients communicate with servers via gRPC calls
- CMSSW processes regular data processings on CPUs, and offload certain tasks to (remote) servers running on different types of coprocessors
- Servers process the offloaded tasks, and send outputs back to clients
- In the studies we chose NVIDIA Triton Inference Server
- Asynchronous processing implemented, so the data transfer latency can be hidden



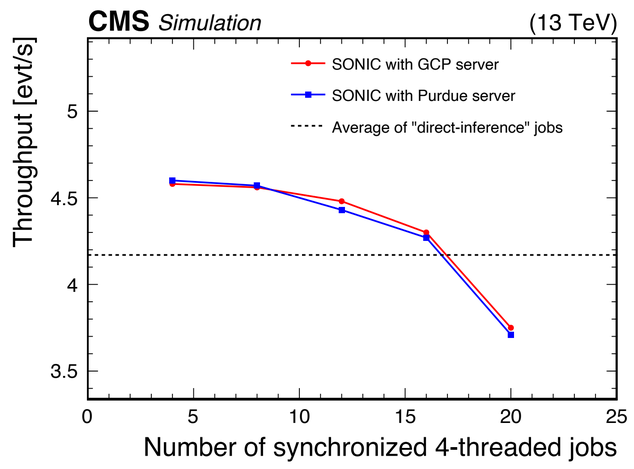
Per-algorithm Optimization

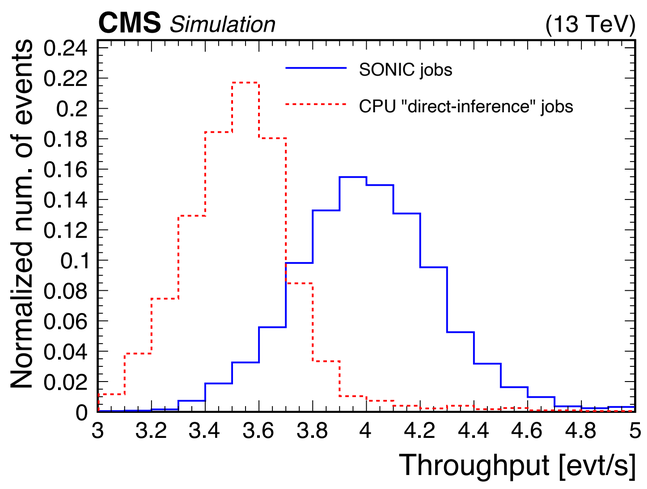
- Optimizing the server inference performance (e.g., processing time and throughput), through "pseudo" clients keeping sending inference requests
- Large flexibility with various parameters that can be optimized: batch size, number of model instances, choice of backends, choice of coprocessors, etc



Small and large-scale Tests

- Check how many clients can communicate with one server simultaneously by varying the number of synchronized clients pinging one server
- Varying the physics distance between servers and clients, confirmed it has little impact on the performance within a few hundred kilometers





- Large-scale tests at Google cloud, with 10,000 CPU jobs and 100 NVIDIA Tesla T4 GPUs
- Observed around 13% throughput gains, which matches the expected from the tasks offloaded

Portability Studies

- Running servers and CPU clients on the same CPUs, with CPU being saturated
- Confirmed that after optimizations there is no throughput decrease running extra servers
- Also tested running on Graphcore IPUs. No extra change in the workflow is needed.
- Observed 3 times throughput improvement compared with NVIDIA V100