AGC Demo Day

16 December 2022

First steps using inference server at coffea-casa facility

Elliott Kauffman (Princeton University)

Oksana Shadura (UNL)

Alexander Held (UW-Madison)

IRIS-HEP Topical Meeting on January 18

https://indico.cern.ch/event/1232532/

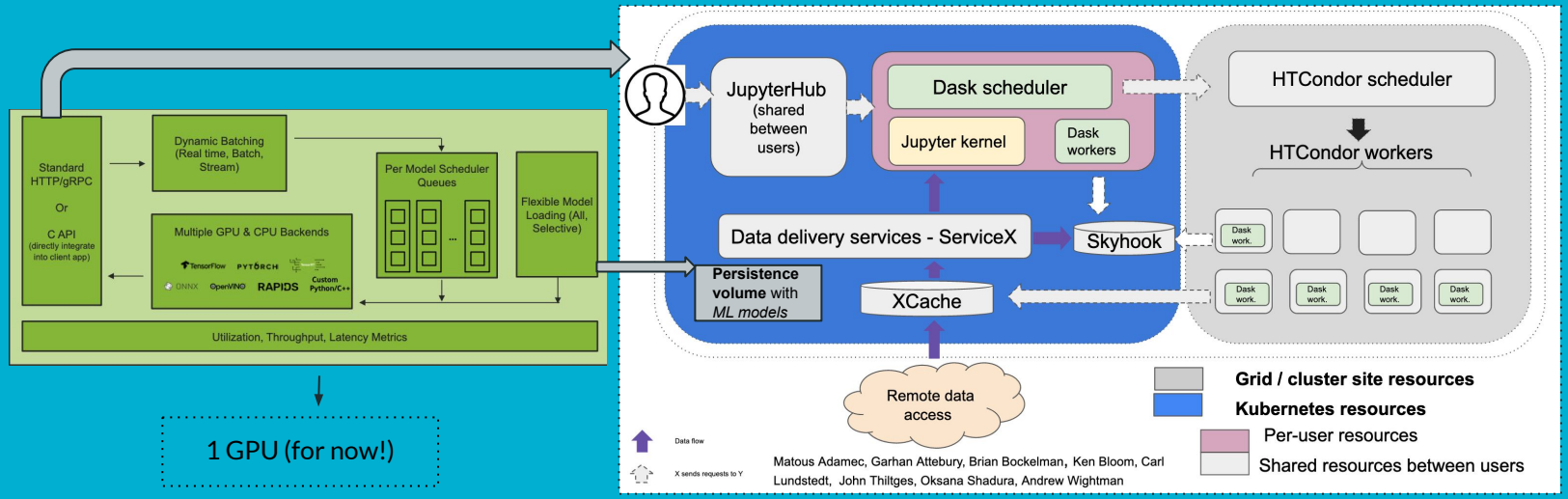
Inference with Triton at Fermilab

Speakers: Claire Savard (University of Colorado, Boulder), Lindsey Gray (Fermilab)

nvidia-triton

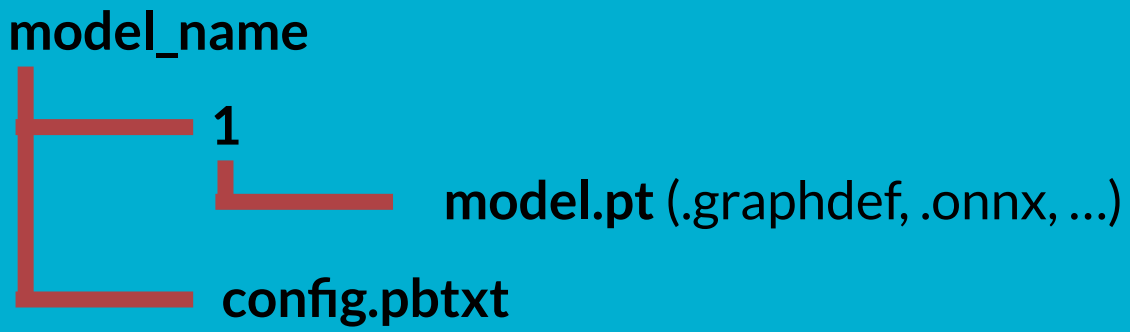
- Inference server that takes columns as input, then outputs ML predictions
- Can use with GPU or CPU
  - Currently have 1 GPU available at UNL
- Useful for inference on a large amount of data with complex models
- Can run inference using multiple models at the same time
- Two client options: gRPC and HTTP

Diagram


Standard HTTP/gRPC
Or
C API
(directly integrate into client app)
Dynamic Batching (Real time, Batch, Stream)
Per Model Scheduler Queues
Flexible Model Loading (All, Selective)
Multiple GPU & CPU Backends
TensorFlow
PYTORCH
ONNX
OpenVINO
RAPIDS
Custom Python/C++
Utilization, Throughput, Latency Metrics
1 GPU (for now!)
JupyterHub (shared between users)
Dask scheduler
Jupyter kernel
Dask workers
HTCondor scheduler
HTCondor workers
Data delivery services - ServiceX
Skyhook
Persistence volume with ML models
XCache
Dask work.
Remote data access
Grid / cluster site resources
Kubernetes resources
Per-user resources
Shared resources between users
Data flow
X sends requests to Y
Matous Adamec, Garhan Attebury, Brian Bockelman, Ken Bloom, Carl Lundstedt, John Thiltges, Oksana Shadura, Andrew Wightman

Loading a model

- Train and save in the appropriate format
  - Supported formats include **ONNX Runtime**, **Tensorflow**, **PyTorch TorchScript**, **TensorRT**
- Create model directory



Loading a model

- Create config file

```
name: "binary_classifier"
platform: "pytorch_libtorch"
max_batch_size: 1000
input [
  {
    name: "x__0"
    data_type: TYPE_FP32
    dims: [ 4 ]
  }
]
output [
  {
    name: "logits__0"
    data_type: TYPE_FP32
    dims: [ 1  ]
  }
]
```

model name (same as name of model directory)

model type/backend

Maximum batch size to use for inference. Must be compatible with model

Names must be in the form <name>__<index>

Equivalent to [-1, 4] with max_batch_size: 0 (assumes any size is allowed for batch dimension)

If batch dimension is first, can use dynamic batching by adding
dynamic_batching {} to config and setting max_batch_size: 0

Loading a model (specific to coffea-casa OD)

- Add model directory to /mnt (available directly from notebook pod)
- Load model into Triton using **curl -v -X POST agc-triton-inference-server:8000/v2/repository/models/binary_classifier/load**

DEMO

https://github.com/ekauffma/coffeacasa-triton-test/

Current Issues and Future Steps

- User cannot see server logs
- Need to find most user-friendly way to load models
  - Although copying the model repository to /mnt is easy, it is less easy for a user to check if a model is loaded or to load/unload it
  - We are using *explicit update* model policy but still load/unload seems somewhat broken from time to time
  - Issue with sharing the same PV between users: https://github.com/ceph/ceph-csi/issues/3562
- Can currently only use local workers, since they need access to the persistent volume (or S3 access through cephfs)

Next steps:

- Define how we will do training and how to integrate it in cc af / analysis pipeline?
- Experiment with requesting inference for multiple models at one
- Experiment with more complex models
- Run with AGC notebook