Report of progress on HPO service with iDDS

Wen Guan, Fernando Barreiro Megino, Tadashi Maeno, Paul Nilsson, Tuan Pham, Torre Wenus, Rui Zhang
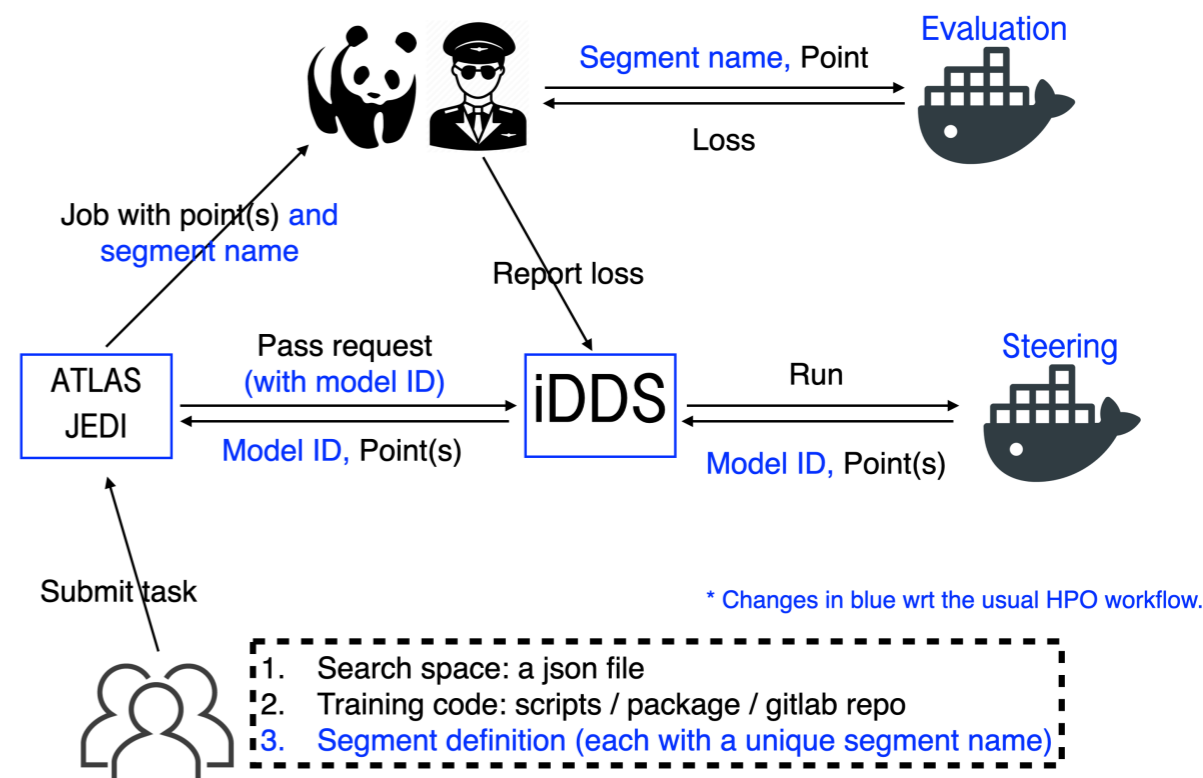
University of Wisconsin-Madison

ADC WFMS weekly meeting

Jun 03, 2021

Introduction

✤ <u>Last time</u> showed a new workflow: Segmented HPO

- Useful for similar but slightly different models that can be parametrised by a segment name

- Alternative is to run each model by hand and bookkeeping oneself

✤ Now is testing on ATLAS FastCaloGan

- Will be part of AF3 for ATLAS Run3 simulation

- One GAN for a particle type and an η range (in total 300 GANs)
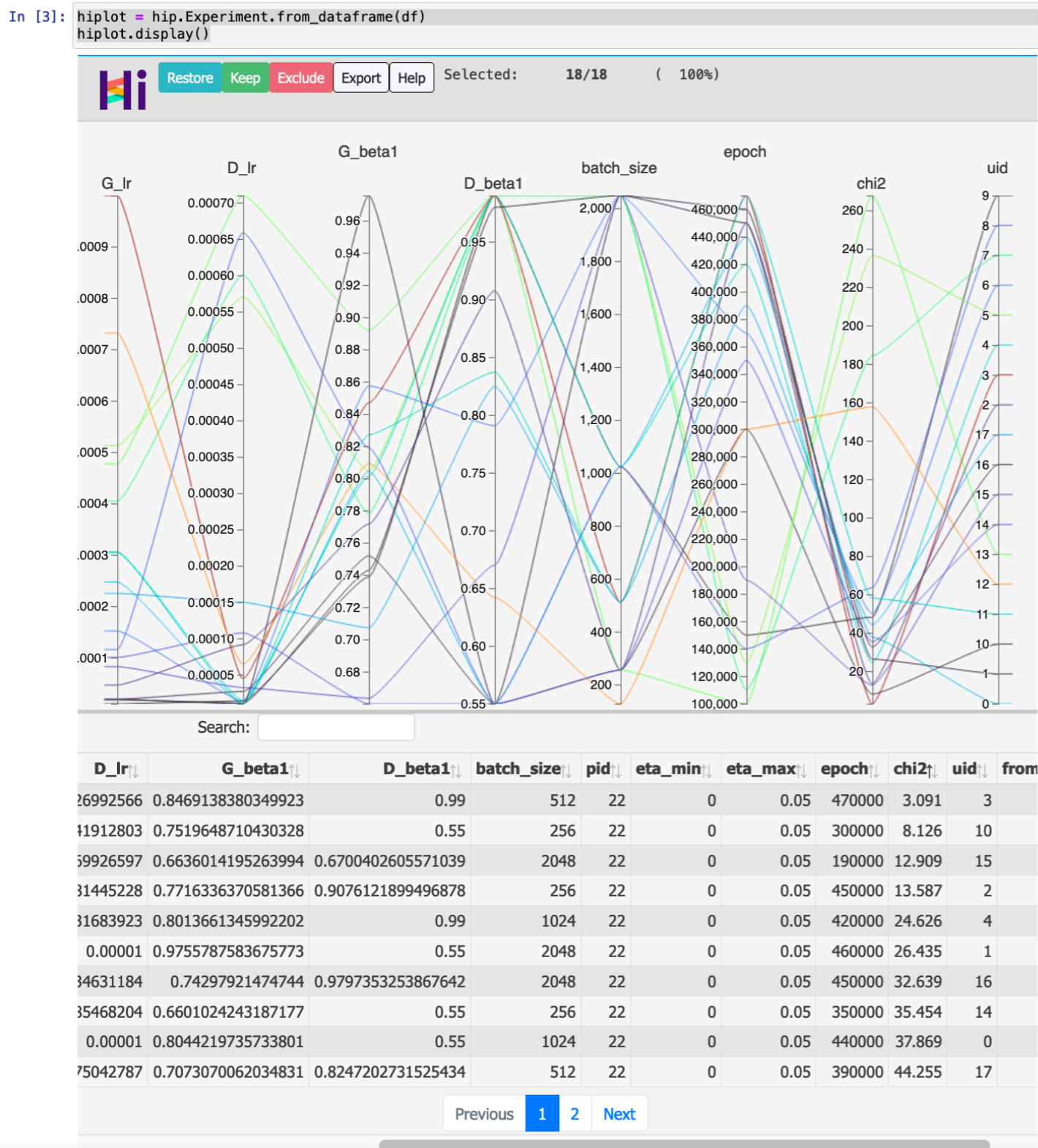
✤ A reminder of the workflow:

Evaluation

Segment name, Point

Loss

Job with point(s) and segment name

Report loss

Steering

| ATLAS JEDI | Pass request (with model ID) | iDDS | Run |

Model ID, Point(s)

Model ID, Point(s)

Submit task

* Changes in blue wrt the usual HPO workflow.

1. Search space: a json file
2. Training code: scripts / package / gitlab repo
3. Segment definition (each with a unique segment name)

Testing and scalability

✤ Orchestration worked last time with small number of segments and short training jobs

✤ Scalability tests were performed

- Now can run realistically long training ~1M epochs.
- Updated to FastCaloGan V2 which changes the strategy of its HPO searching
- Suffered from some incompatibility of the cuda version on site and the cuda version required by the container
  - Job still runs but will be very slow since it does not use GPU
 - Was not straightforward to detect as there is no error

Post training analysis

✤ FastCaloGan requires a post training analysis

- It trains a very long epoch and stores checkpoints; the best model can be any checkpoint
- A post training analysis is to select the best model
- In HPO, this means to select the best model in all hyperparameters

✤ Incorporated a visualisation tool for this purpose

- If more jobs finished, rerunning this will include more curves
- Downloading of the output is required

✤ Further simplification

- Incorporate the post-training selection to the workflow

Post training analysis results

❖ Can easily identify the best model and track the results

ATLAS
Simulation internal
γ, 0.00<|η|<0.05
— Geant4
- - - GAN
Epoch: 300000
χ^2/NDF = 2194/270 = 8.1
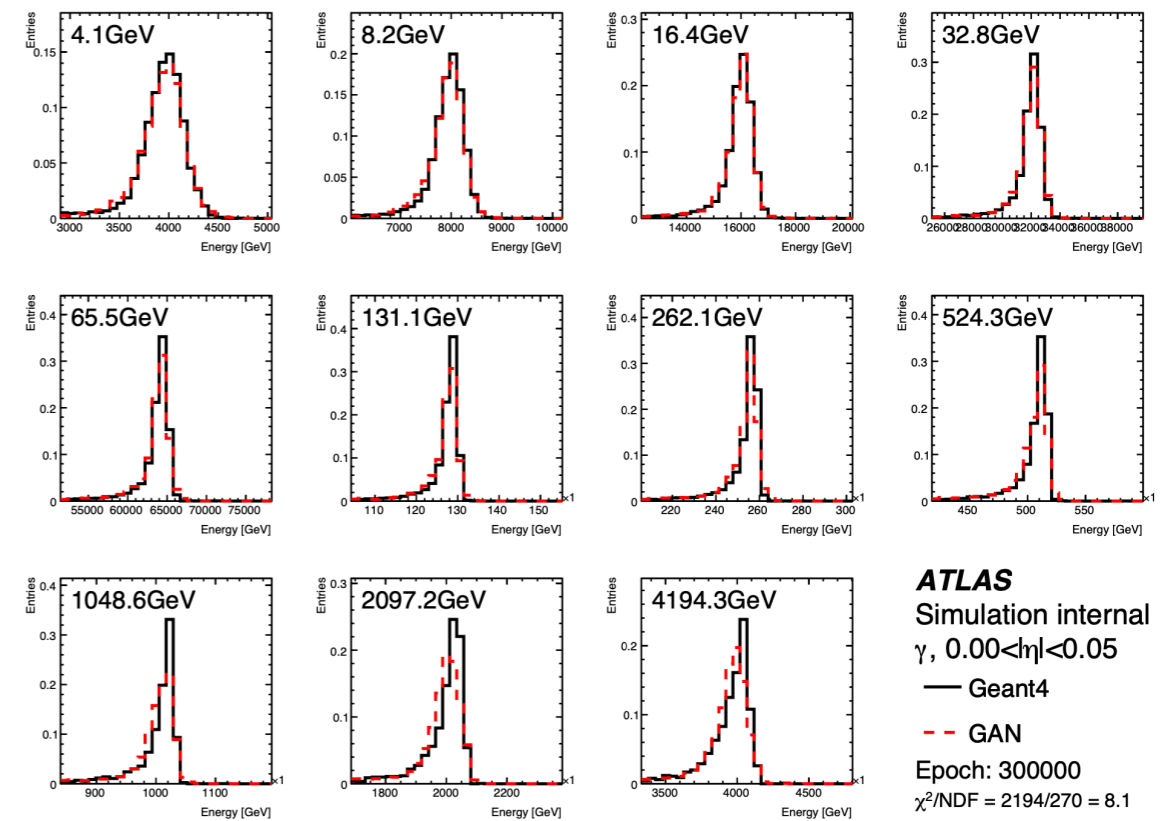
Distributed HPO

✤ We have also been working on distributed HPO using commercial cloud.

- Fernando has been working very hard to bridge the resources with the ALTAS infrastructure (<u>his talk today</u>)
- Testing Horovod as a first step

✤ Testing payload are available

- Not a big community in ATLAS - we are not in the regime of "very" big training model
  - Maybe just because we lack resources?
- An available ATLAS payload is not fully taking advantage of it
  - It has several steps, only one of the step is using multiple GPUs
- For now, we focus on a mnist payload

Distributed HPO (2)

✤ The payload is validated on Summit with 1, 3, 4 GPUs

- An "almost" linear acceleration is seen

- Caveat: results are from multiple GPUs but all on a single node
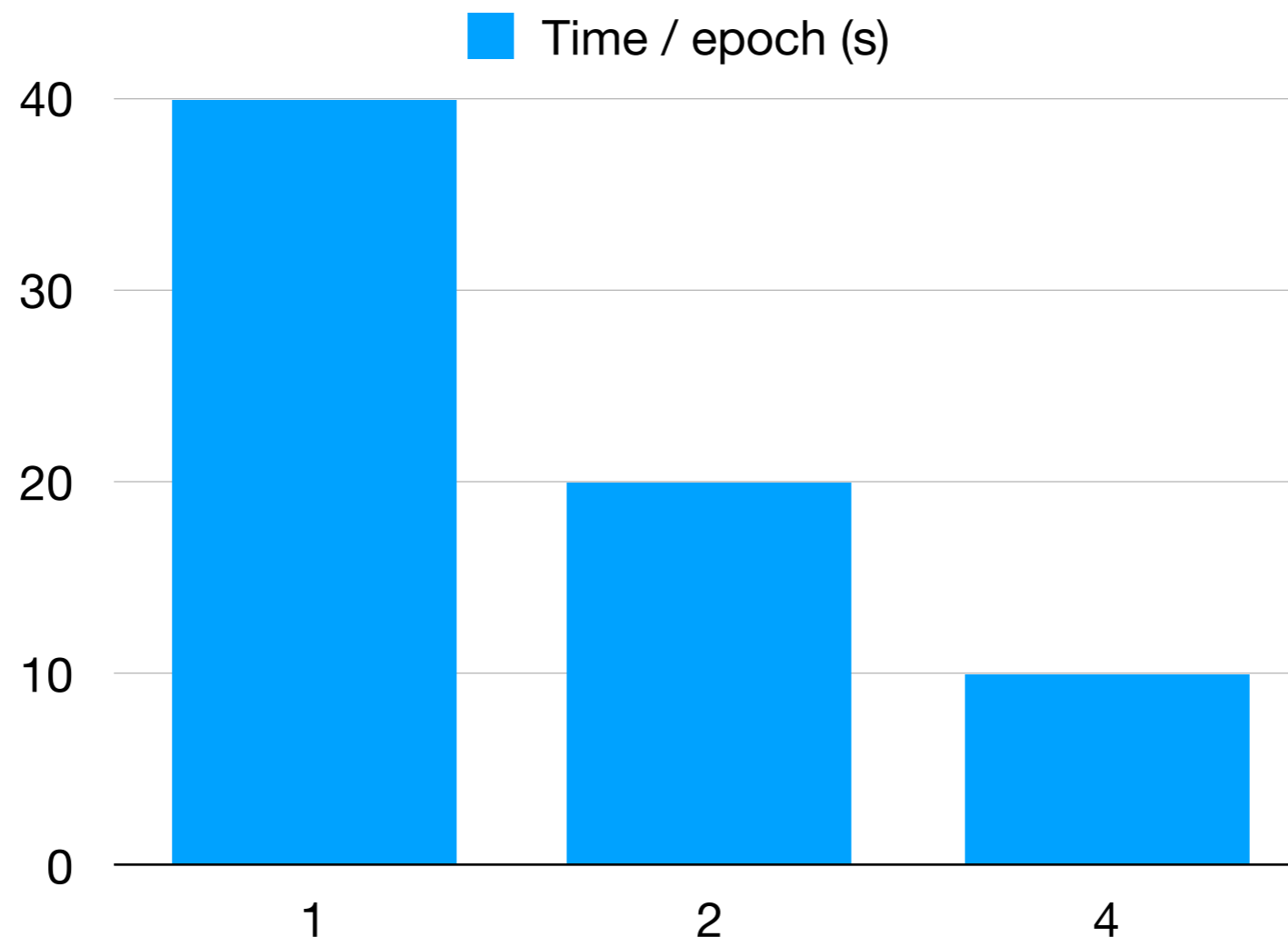
Visualisation on task submission

✤ We are developing a Jupyter-based job submission tool

- Tuan's QT: <u>Presentation last time</u>

- The idea is to hind the JSON configuration details from users, simplify and guild as much their choices

- Now migrating from widget to a dashboard (almost done)

- Further development is to fetch the output from PanDA via the dashboard, and (potentially) monitor jobs

Summary

✤ Testing scalability of the whole HPO workflow (based on segmented HPO)

- Went smoothing, various issues were solved during the test
- Will extend to more sites besides BNL
- At BNL, Harvester jobs are evicted by Jupyter jobs

✤ Some activities on distributed HPO

- Making good progress as well
- Looking for ATLAS payload for multiple GPU training

✤ Various activities are happening in the past couple of months

- Good progresses were made in several directions in parallel