pyhf Hardware Acceleration Benchmarking with CPUs and GPUs

IRIS-HEP Fellows Presentations

Research Fellow: Bo Zheng
(Master CS student at Rice)

Mentor: Matthew Feickert
01

Project Description
Project Description

- The pyhf library is a pure-python implementation of the HistFactory statistical model for multi-bin histogram-based likelihoods.

- The pyhf-benchmark library I developed this summer is a benchmarking suite in Python to test and benchmark the performance increase of the hardware accelerated backend (Tensorflow, JAX, Pytorch) on CPUs and GPUs on openly published physics data from LHC experiments.
Summary of work

- Wrote monitoring code for the performance of CPU and GPU based on NVIDIA Management Library and wrote comparison plot generation code using matplotlib
- Made library be a pip installable package and supported Python API as well as a CLI API
- Added unit tests for pyhf-benchmark package
- Wrote documentation using Sphinx that is deployed as a public website
02
Research & Solution
Research & Solution

System Metrics captured during benchmarking

- CPU Utilization
- System Memory Utilization
- Disk I/O Utilization
- Network traffic (bytes sent and received)
- GPU Utilization
- GPU Temperature
- GPU Time Spent Accessing Memory (as a percentage of the sample time)
- GPU Memory Allocated

Reference
wandb package & NVIDIA Management Library
Achievement Exhibition
Achievement Exhibition

pyhf-benchmark github: https://github.com/pyhf/pyhf-benchmark

Installation:

$ git clone https://github.com/pyhf/pyhf-benchmark.git
$ python -m pip install .

PYPI installation is coming soon
Python API can be used as well

CLI Usage:

$ pyhf-benchmark run -c [-b] [-p] [-u] [-m] [-n] [-mm]

Examples:

$ pyhf-benchmark run -c mle -b numpy -u
https://www.hepdata.net/record/resource/1267798?view=true -m [750,100]
$ pyhf-benchmark run -c interpolation -b [numpy, jax] -n 0 -mm fast
Future work
Future work

- Perform benchmark test cases for model analyzing different hardwares

- Current research on TPUs is limited to the evaluation to the performance of TPUs on Google Colab platform. More research over TPUs is needed.

- I have wrote code for plots visualization on websites using D3.js and Flask. Further optimization of a nicer data visualization can be taken if needed.
Future plan

- Go back to Rice for a master degree of computer science
- Look for a SDE (software developer engineer) full-time job after graduation