IRIS-HEP Fellow Proposal

Project: Estimation of the energy cost and efficiency of the HEP data compression ML algorithm (Baler)

Author:Leonid Didukh (ledidukh@gmail.com)
Mentor: Caterina Doglioni (caterina.doglioni@gmail.com)
Duration: 05.06.2023 - 15.09.2023

Project Description:

Energy consumption and efficient green software are very crucial points in data preservation, management, and computing workflow organization in HEP collaboration. With the increasing amount of data that can be processed and limited storage facilities, a question naturally arises - is it possible to store and retrieve the data in an efficient way?

Baler [1, 2] is an Open Source ML python package that aims to alleviate the issue of storing large data files, by compressing data from HEP and other scientific fields. At the same time, the efficiency, memory consumption, computing time, and energy cost of this software have not been studied, and could potentially be improved. For this purpose, in this project, we will study profiling metrics of Baler utilizing both CPUs and GPUs. Moreover, we will provide suggestions how performance of various part of Baler can be optimized.

Deliverables and Goals:

1. Provide the initial document that contains a list of improvements for parallelization in Baler. Baler is CPU/GPU executable, where parallelization is done in the data fashion. Nevertheless, we will approbate the distributed computational methods to speed up ML inference using the model parallel approach [3,4].
2. Deliver the code and results from profiling the software:
    a. on CPU - lxplus and computing cluster at the University of Manchester (Blackett).
    b. on GPUs - computing cluster at the University of Manchester, with lxplus as a backup, using code/metrics from Pratik Jawahar, PhD at the University of Manchester.

    In order to profile the performance of the software on CPU/GPU, we will use Pytorch profiler [5] and open source profilers for deep learning algorithm [15, 16]. We will use these tools for computational time and memory consumption estimation [6,11]. Also the relevant test will be constructed in order to define the CPU bottleneck.
3. Deliver code and results on energy consumption metrics from literature [7,8,9,10,11,12,13], linked to the previous profiling exercise. Our primary focus will be estimation of GPU energy consumption using code/metrics from Renato Cardoso (OpenLab, CERN), but if time allows we will also repeat this exercise for CPU.

Timeline:
The research and development process is split into two weeks periods. The mentor and I will have weekly meetings for updates and issue resolution, and we will be regularly communicating with the rest of the team via Mattermost/Slack. Every two weeks we will present a status report at the Baler weekly meeting and at the Manchester ATLAS jets and trigger group meeting.

- **05.06 - 19.06**
  Preparatory phase for reading literature and setting up code and accounts.
- **19.06 - 30.06**
  Hands-on tests and code analysis of Baler and writing of document on parallelization improvements. Test the framework locally using laptop/desktop. Conduct the research about possible ways how to improve the efficiency of the infrerence of Baler.
- **03.07 - 17.07**
  Work on the deliverable number 2a. Define and test profiling metrics for CPU (list metrics, tests different ones, choose a final metric), compiling results into a presentation. The task was performed in collaboration with Google Summer of Code HSF student Manas Pratim Biswas.
- **17.07 - 31.07**
  Work on the deliverable number 2b. Define and test if of the profiling metrics for GPU. Perform GPU tests and compile results into a presentation.
- **31.07 - 14.08**
  Test code and scripts (from R. Cardoso) related to the green software analysis for estimation of GPU energy consumption, link to GPU profiling done before.
- **14.08 - 28.08**
  Compare results from R. Cardoso's scripts and references to green software metrics. Compile findings into a presentation.
- **28.08 - 31.08**
  Apply ML-specific energy consumption metrics to Baler and compile results into a presentation.
- **31.08 - 15.09**
  Wrap up results into a report, presentation and publish on Zenodo (see e.g. https://zenodo.org/record/5482611#.ZHcYaS3MKAP)

Communication: Since there will be more students working on topics connected to this project (energy efficiency of different software and further developments of the Baler algorithm), we will meet with them regularly and keep a common flipchart (trello/JIRA) board for tasks.

References:
[1]Baler code - https://github.com/baler-collaboration/baler

[2]Baler -- Machine Learning Based Compression of Scientific Data - https://arxiv.org/abs/2305.02283
[3]Distributed computing pytorch - https://pytorch.org/tutorials/beginner/dist_overview.html
[4]Run Pytorch Faster- https://sebastianraschka.com/blog/2023/pytorch-faster.html
[5]Pytorch Profiler - https://pytorch.org/tutorials/recipes/recipes/profiler_recipe.html
[6] Compute and Energy Consumption Trends in Deep Learning Inference https://arxiv.org/abs/2109.05472
[7]Green Software Tools and Resources - https://github.com/Green-Software-Foundation/awesome-green-software
[8] Deploying a machine learning model catalog at CERN https://indico.jlab.org/event/459/contributions/11656/
[9]Accelerating GAN training using highly parallel hardware on public cloud - https://arxiv.org/abs/2111.04628
[10] Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models - http://arxiv.org/abs/2007.03051
[11]Measuring the Algorithmic Efficiency of Neural Networks - https://arxiv.org/abs/2005.04305
[12]AI and Compute - https://openai.com/research/ai-and-compute
[13] Object Metrics for Green Software - http://www.jsoftware.us/vol16/452-JSW15412.pdf
[14]Ftlops library - https://pypi.org/project/ptflops/
[15] Deeplite profiler - https://github.com/Deeplite/deeplite-profiler