[Project Check-in]
Power Efficient Model Inference on Nvidia GPUs

Ante Tonkovic-Capin, Ashu Kumar, Drishan Poovaya, Pratik Sanghavi
Group Name: ADAP's 11

16th April 2024

Progress so Far

- **Literature review on relevant power tuning techniques for inference**: We started with some background research on existing research on energy-minded strategies for model training and serving taking us through model-level optimizations, efficient model training and model selection strategies for inference. The majority of the research thus far has primarily focused on training, with surprisingly little spent on inference when it comes to effecient power consumption.
- **Exploration of inference on QIDK**: We started with a plan to run inference, profile and then adapt a power efficient methodology for model inference on Qualcomm Innovator Development Kit.
  - We did a review of the SNPE APIs as well as its Python wrapper (PySNPE) for understanding the pipeline for converting models to the appropriate format, specifying targets, optimization techniques and troubleshooting of deployed models.
  - We connected with an employee at Qualcomm to understand power tuning options available on QIDK and to avail ourselves of any additional resources or documentation that could be available for this topic.
- **Exploration of inference on NVIDIA GPUs**: We later pivoted to Nvidia GPUs since there is an abundance of documentation on using these devices for inference and generally greater accessibility.
  - *Initial Experiment*: Starting with a simple K80 instance on cloud, we ran an inference server and observed the power and GPU memory utilization of the inference jobs submitted. Our observations about the power consumption and memory footprint aligned with our expectations. Time to move to phase 2
  - *Setup for Jetson TX2*: The team also worked on setting up the jetson device for running the triton inference server. Unbeknownst to us, the model backends and the inference server was already built on the device albeit not on the system path. Only after a deep dive in the jetpack release notes, we were able to locate these crucial packages. Precious man hours lost here!

Challenges

- Insufficient documentation on power tuning on QIDK as well as delays in accessing Qualcomnm HDK led us to pivot to Nvidia Jetson as our device of choice.
- Initial issue with accessing Jetson device due to campus VPN access not enabled for the device.
- Packages not installing due to proxy not set up properly. Got the solution for setting the same from CSL team.
- Pytorch and Tensorflow packages are not getting installed in the Jetson device. However, after digging deeper into jetpack release notes, we were able to locate the model backends and inference server which we will try to set up and use for the project going forward.
- Unable to access PyTriton API on provided resource due to Ubunutu 18 restriction on machine with incompatible glibc requirement leading to additional hours spent debugging and trying to build compatible versions from source.

Timeline

Action	Time
Develop robust code to run inference on a continuous batch of data during observation period	1 week
Work through feasibility of modifying Zeus for inference; pivot to a offline model if infeasible	1 week
Findings and analysis, report results and summarize challenges faced	0.5 week

Table 1: Timeline for the Project

Help Needed

- Need access to Chameleon Cloud to try out the GPU power level monitoring in parallel along with Jetson TX2 device.
- Setup session with Minghao to understand procedure to perform a power sweep on Jetson. Data collected will be crucial to training a simple model to set power level during inference.