ADAPter: Power Efficient Model Inference on Nvidia GPUs

Ante Tonkovic-Capin, Ashu Kumar, Drishan Poovaya, Pratik Sanghavi

WISCONSIN
UNIVERSITY OF WISCONSIN - MADISON

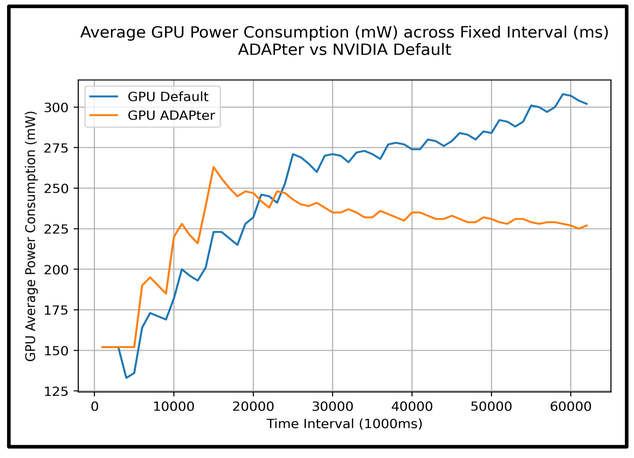
Motivation

- Growing popularity of AI models +17 billion mobile devices on the planet (~2 devices/person) = High demand for power
- Efficient energy consumption for inference on such devices is crucial
- More focus has been on efficient training, but NVIDIA and Amazon both estimate that **inference accounts for 90% of the costs** for deployed models

Our Approach

- Make the inference server SLO aware by forcing client to specify a desired latency with the payload.
- Method used must be quick and minimally impede inference jobs
- Uses a simple moving average to estimate whether to step up or step-down GPU frequency
- At low frequencies, requests will be dropped; use the queues to retry with exponential backoff
- We will build middleware that sits between the clients and the Triton Server that measures average deviation and tunes frequency every few seconds.

Early results

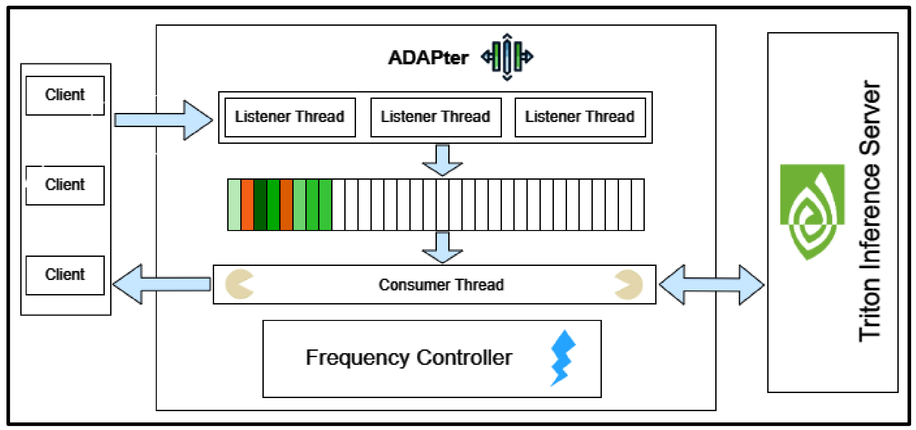


Push a constant stream of inference jobs with a fixed latency comparing default versus ADAPter GPU consumption

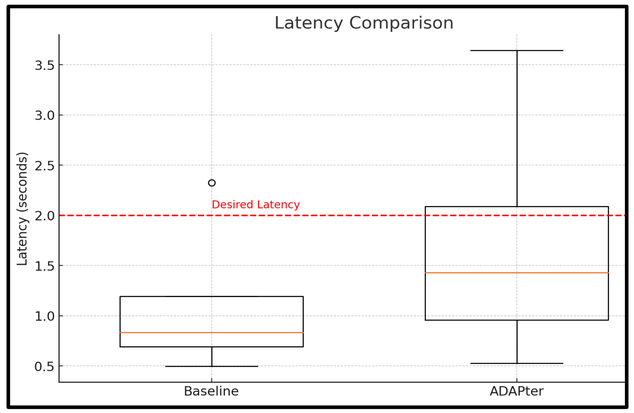
Related work

- Most of the research focus has been on improving efficiency during model training
- Estimates on cost of inference have been done by J. Barr and McDonald J. et al. Clearly demonstrating the need for more efficient approaches
- Samuel S. Ogden and Tian Guo created "MODI" targeting efficient inference using edge computing

ADAPter



Frequency Controller adjusts the GPU frequency to ensure that inference is served efficiently



Power draw reduces with our simple module with variance of latency increased. This happens because at lower frequencies, we go for multiple rounds of retry and backoff, frequency floor set to improve this behavior

Challenges

- Initial project focused on *Zeus* package and research on Qualcomm Innovators Development Kit (QIDK) with Snapdragon SoC, however the device and kit was never received
- Project goals modified several times as alternate approaches using available devices faced several obstacles
- Even with access, modifying device hardware parameters such as power consumption can be indeterministic and difficult to validate, with other settings such as fan and cooling systems being out-of-reach completely.

Contributions / WIP

- Tune the constants currently in use
- Batching requests and test with “bursty” traffic
- Integrate logic within the triton inference server code
- Incorporate learned frequency floor & ceiling for each model
- Incorporate model selection with different quantization