DALHOUSIE UNIVERSITY

CASCON 2024

PINet Programmable and Intelligent Networking

GreenStream: Enabling Sustainable LLM Inference in Stream Processing

Md. Monzurul Amin Ifath and Israat Haque

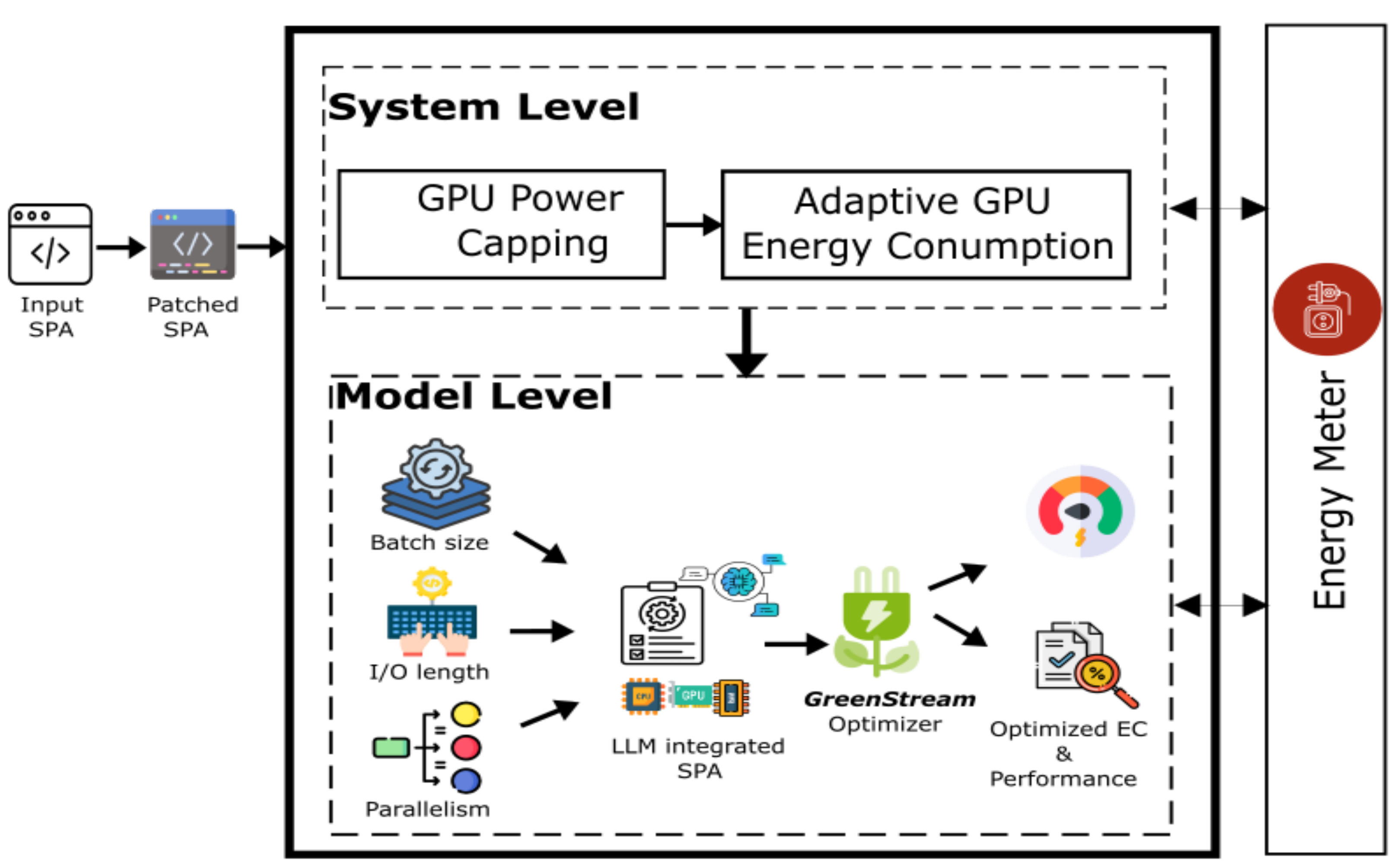
1. Problem Statement

- Large Language Models (LLMs) are increasingly integrated into stream processing applications (SPAs) (e.g., predictive fraud detection, personalized recommendations, and GenAI travel assistants).
- LLM inference consumes significant energy, posing a sustainability challenge.
- Traditional optimization methods are hindered by the complexity of distributed nature of SPAs.

2. Approach

- We propose *GreenStream*, a framework to optimize the energy efficiency of LLM inference without compromising the performance of SPAs.

Conceptual Overview



Workflow

- Take an existing SPA script as input.
- Apply patches to the SPA to enable energy meter and adjust batch size, I/O length, and parallelism.
- Impose GPU power capping and adaptive GPU energy consumption to optimize energy usage.
- Run the patched SPA through optimizer to confirm optimal balance between energy usage and performance.

3. Setup Details

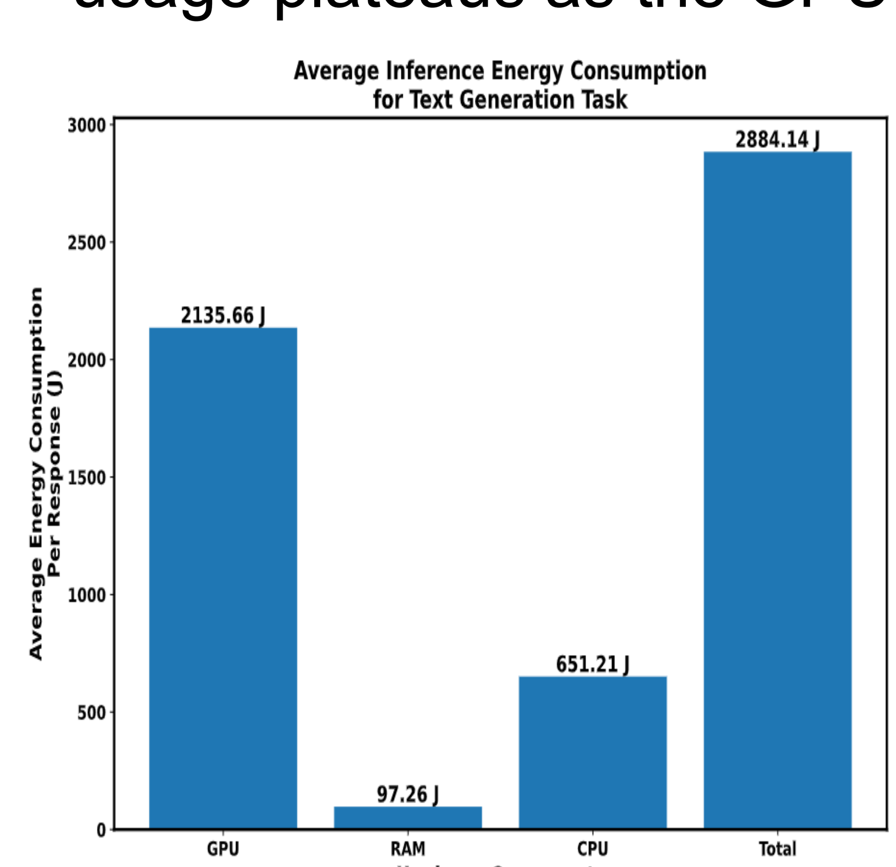
- Energy meter is based on PyRAPL, pynvml and only compatible with Intel processors and NVIDIA GPUs.
- Perform experiments on Intel Xeon Silver 4310 CPU and NVIDIA A100 GPU.
- Evaluate Meta Llama 3.1 8B for preliminary results.

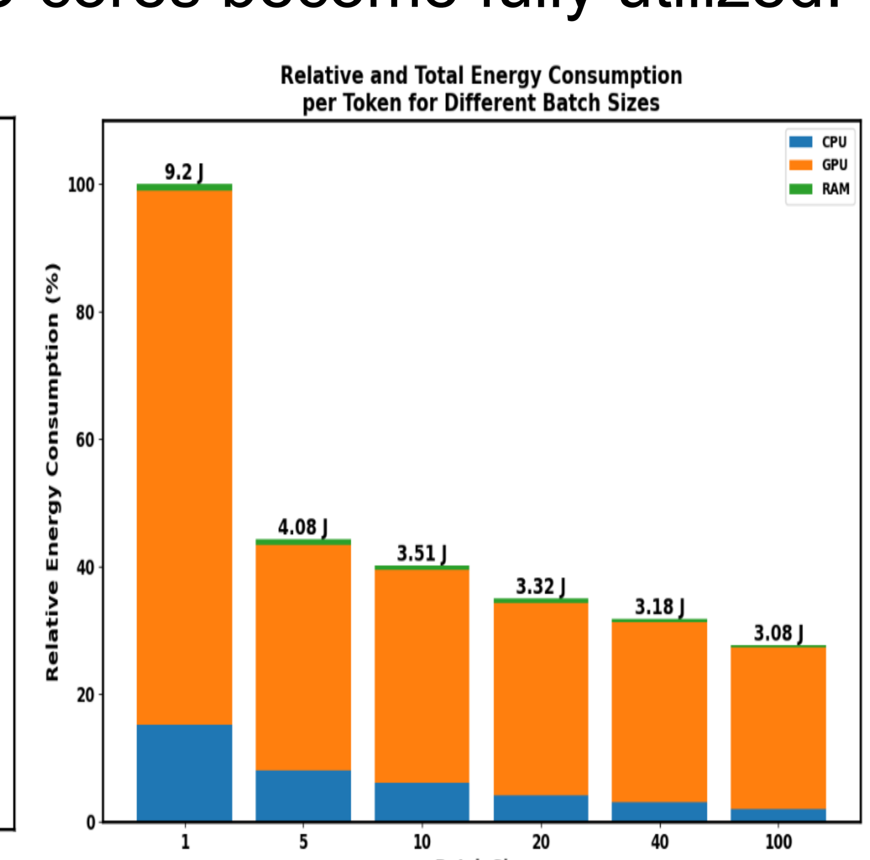
4. Example Use Case

```
# Custom energy meter for sustainable ML
from GreenStream import EnergyMeter

# Initialize the GreenStream energy meter
energy_meter = EnergyMeter()

# Load the pre-trained model,tokenizer
model_name = "meta-llama/Llama-3-8B-text-completion"
....

# Define prompt input and tokenize the prompt
prompt = [...]
input_ids = tokenizer(prompt, return_tensors="pt").input_ids

# Set generation parameters
generation_params = {
    "max_seq_length": 512,
    "top_p": 0.9,
    "temperature": 0.6,
    "max_gen_len": 64,
}

# Generate output with energy measurement
with torch.no_grad():
    energy_meter.begin()  # Start measuring energy usage
    output_ids = model.generate(input_ids, **generation_params)
    energy_meter.end()    # End measuring energy usage

# Decode generated tokens and print in text
....

# Total energy consumption is tracked by energy_meter
Average total energy consumption for text generation: ~2900J
```

5. Preliminary Results

- During inference, main energy consumer is the GPU.
- 3x energy consumption reduction with increasing batch size of 1 to 100 (in cost of higher GPU memory usage).
- Beyond a certain batch size, the decrease in energy usage plateaus as the GPU cores become fully utilized.





6.Next Steps

- Automatically identify optimal power caps and model parameters for minimal energy consumption.
- Systematically assess the impact of different LLM variants on performance and energy usage (e.g., Llama 8B vs 70B).
- Evaluate energy efficiency of LLM inference on energy optimized inference servers (e.g., NVIDIA TensorRT).