

# Energy-Efficient Deep Learning with PyTorch and



Jae-Won Chung  
October 17<sup>th</sup>, 2023



ML.ENERGY



# Why Energy?

## Outlooks for energy consumption

- IT accounts for 7-8% of global electricity demand today<sup>[1]</sup>
- Before GenAI, 10-14% by 2030 was the usual projection<sup>[1,2]</sup>
- GenAI is likely to accelerate increase, without focused efforts

[1] "Digital Economy and Climate Impact – White Paper," Schneider Electric, 2021

[2] "Hypothesis for primary energy use, electricity use and CO2 emissions of global computing and its shares of the total between 2020 and 2030," Andrae et al., 2020

# How Do We Optimize Energy?

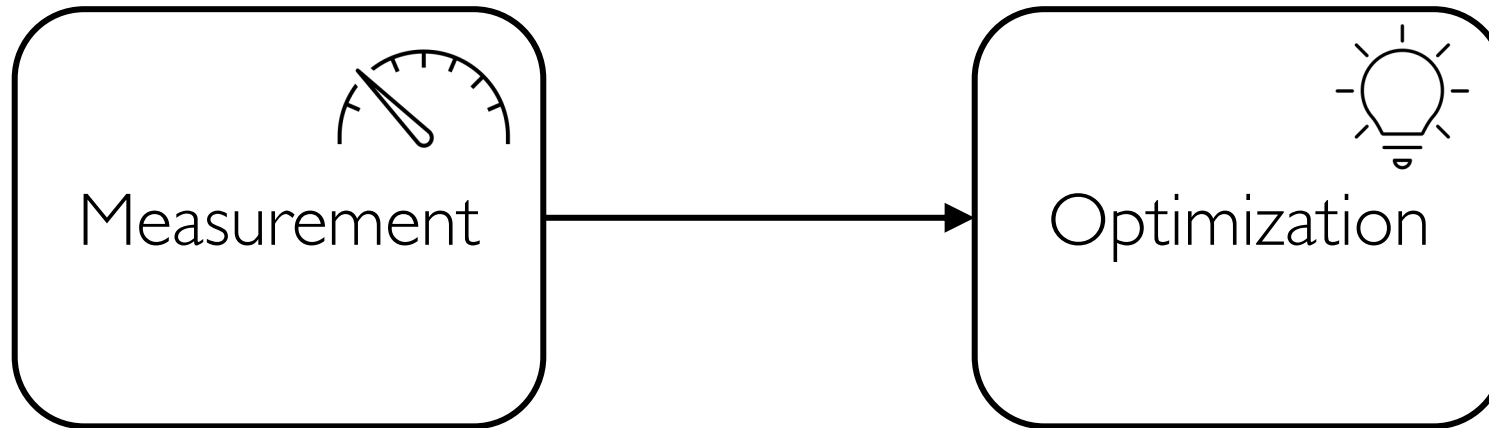
## GPU-side knobs

- Power limit
- Frequency locking

Neither changes **what** is computed by the GPU.



<https://ml.energy/zeus>



# Measuring Energy with Zeus

## Zeus Monitor

- Measure the **time and energy consumption** of arbitrary code ranges

```
1 from zeus.monitor import ZeusMonitor
2
3
4 monitor = ZeusMonitor(gpu_indices=[0,1,2,3])
5
6 monitor.begin_window("training")
7 for e in range(epochs):
8     monitor.begin_window(f"epoch {e}")
9     for x, y in train_data_loader:
10
11         y_hat = model(x)
12         loss = criterion(y, y_hat)
13         ...
14     mes = monitor.end_window(f"epoch {e}")
15 mes = monitor.end_window("training")
16
```

# Optimizing Energy with Zeus

## Power Limit Optimizer

- Automatically optimizes GPU power limit

```
1 from zeus.monitor import ZeusMonitor
2 from zeus.optimizer import GlobalPowerLimitOptimizer
3
4 monitor = ZeusMonitor(gpu_indices=[0,1,2,3])
5 plo = GlobalPowerLimitOptimizer(monitor)
6
7 for e in range(epochs):
8     plo.on_epoch_begin()
9     for x, y in train_data_loader:
10        plo.on_step_begin()
11        y_hat = model(x)
12        loss = criterion(y, y_hat)
13        ...
14        plo.on_step_end()
15    plo.on_epoch_end()
16
```

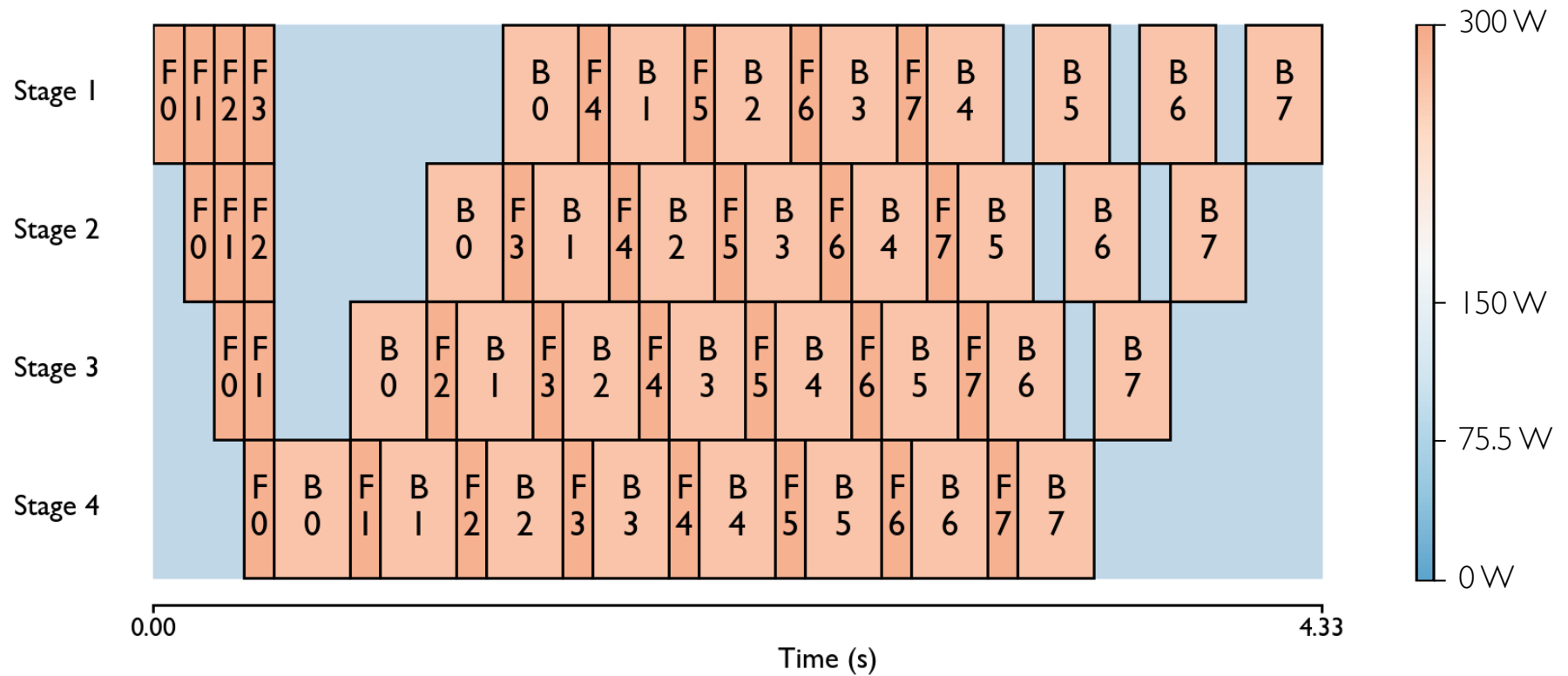
# Optimizing Energy with Zeus

## Power Limit Optimizer

- You define what **optimal** means

```
1 from zeus.optimizer import GlobalPowerLimitOptimizer
2 from zeus.optimizer.power_limit import (
3     Time,
4     Energy,
5     MaxSlowdownConstraint,
6 )
7
8 plo = GlobalPowerLimitOptimizer(
9     monitor,
10    MaxSlowdownConstraint(factor=1.1),
11 )
12
```

# Energy-Efficient Large Model Training



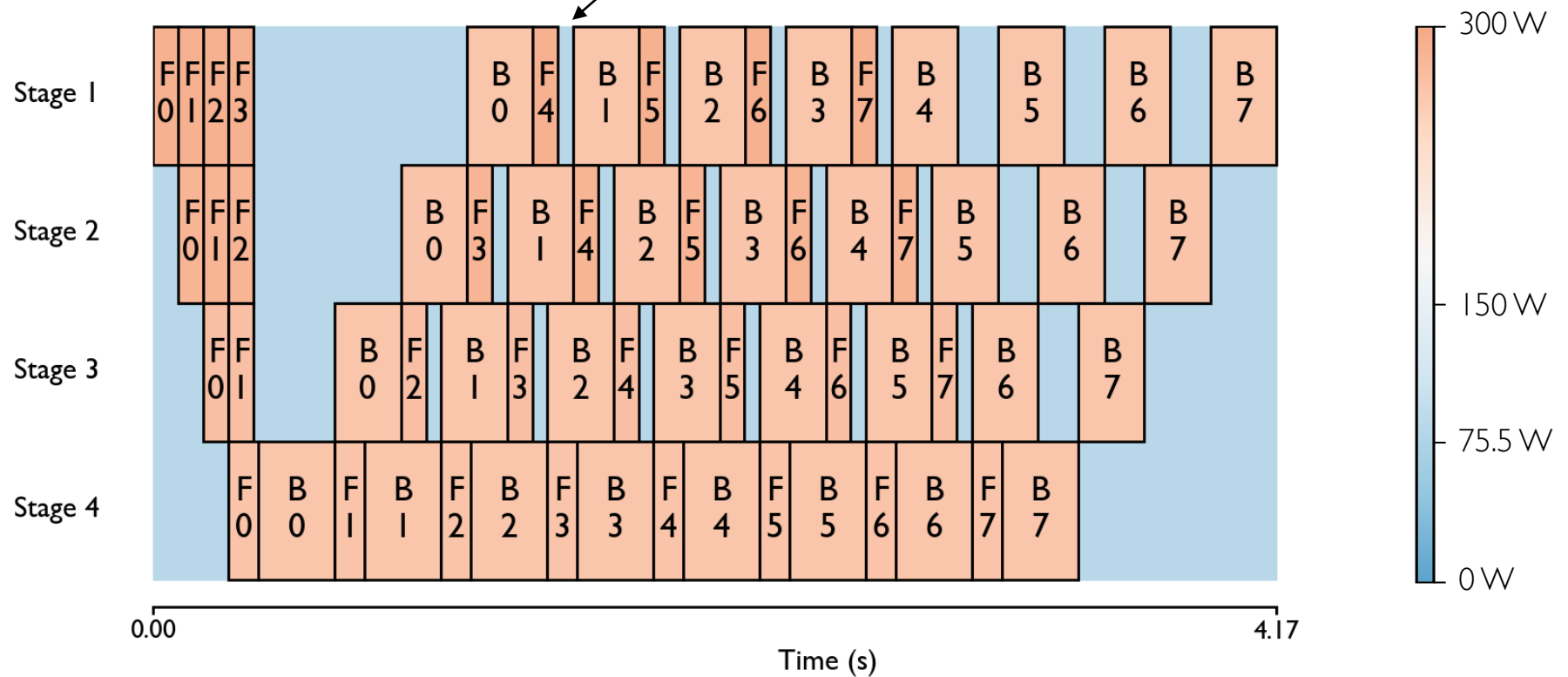
F = Forward, B = Backward

One training iteration, 4 stage 8 microbatch IFIB pipeline



# Energy-Efficient Large Model Training

Idle times – Not all computations need to run at full speed!

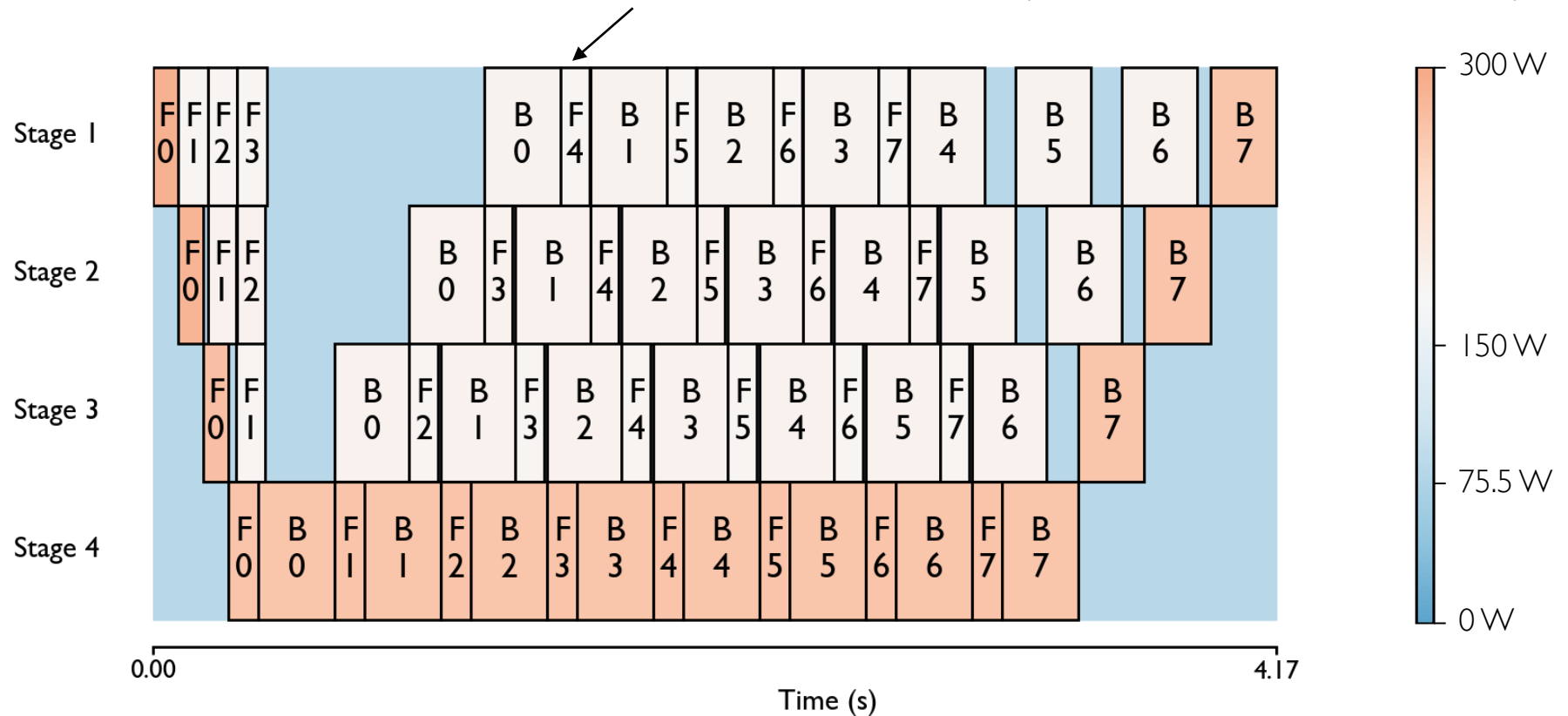


F = Forward, B = Backward

One training iteration, 4 stage 8 microbatch IFIB pipeline  
 Computation drawn to scale for GPT3-large on NVIDIA A40 GPUs

# Energy-Efficient Large Model Training

Idle times – Not all computations need to run at full speed!

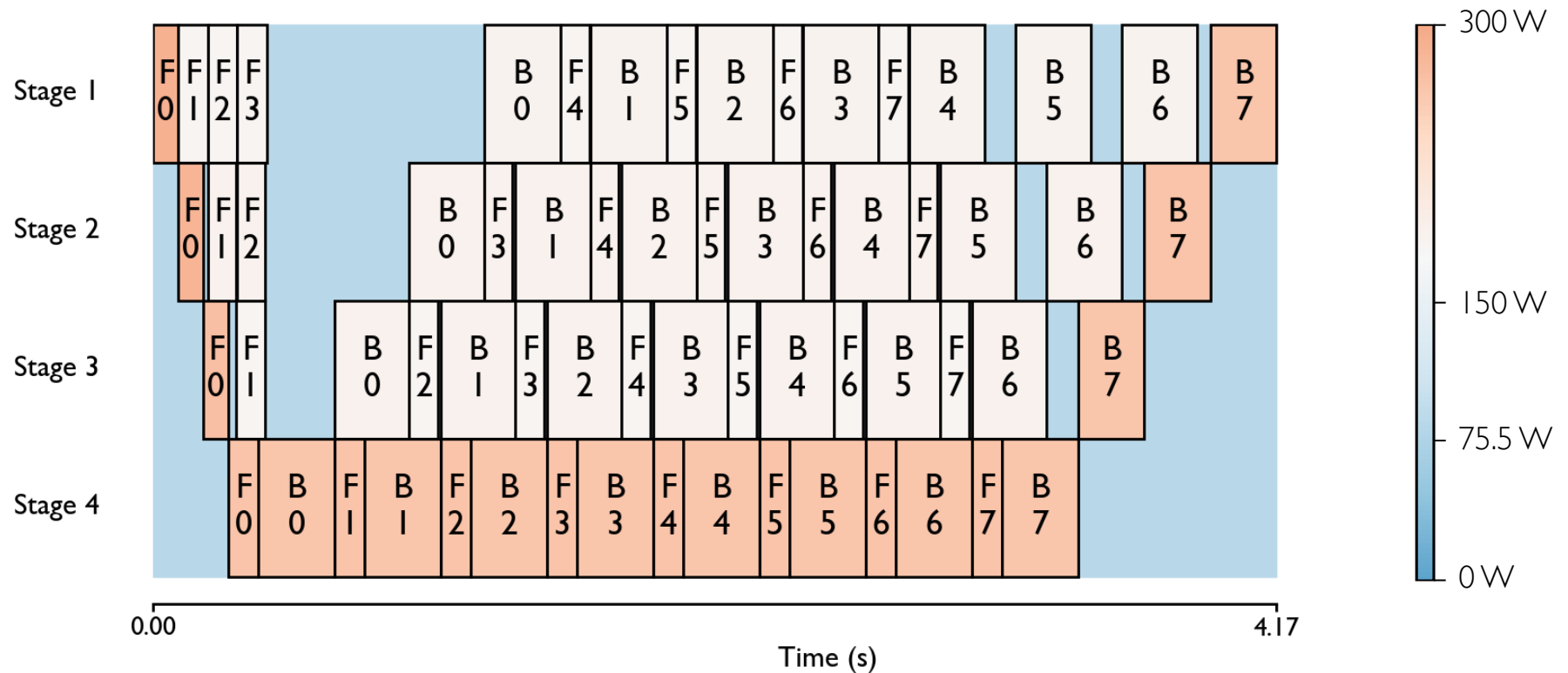


F = Forward, B = Backward

One training iteration, 4 stage 8 microbatch IFIB pipeline  
 Computation drawn to scale for GPT3-large on NVIDIA A40 GPUs

# Energy-Efficient Large Model Training

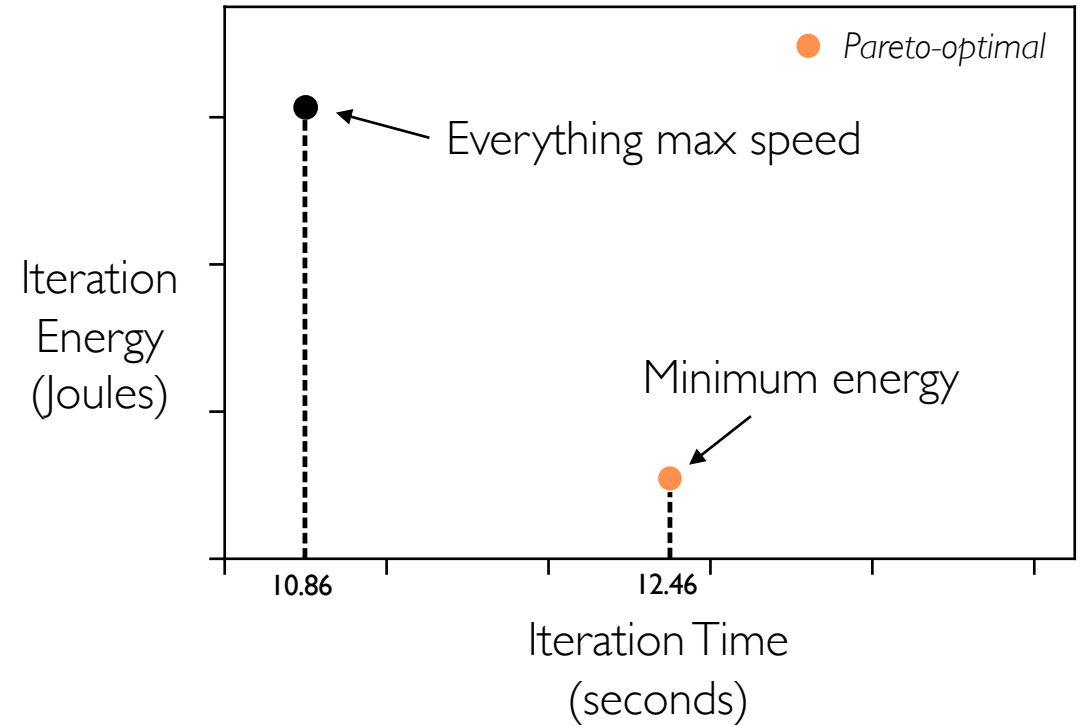
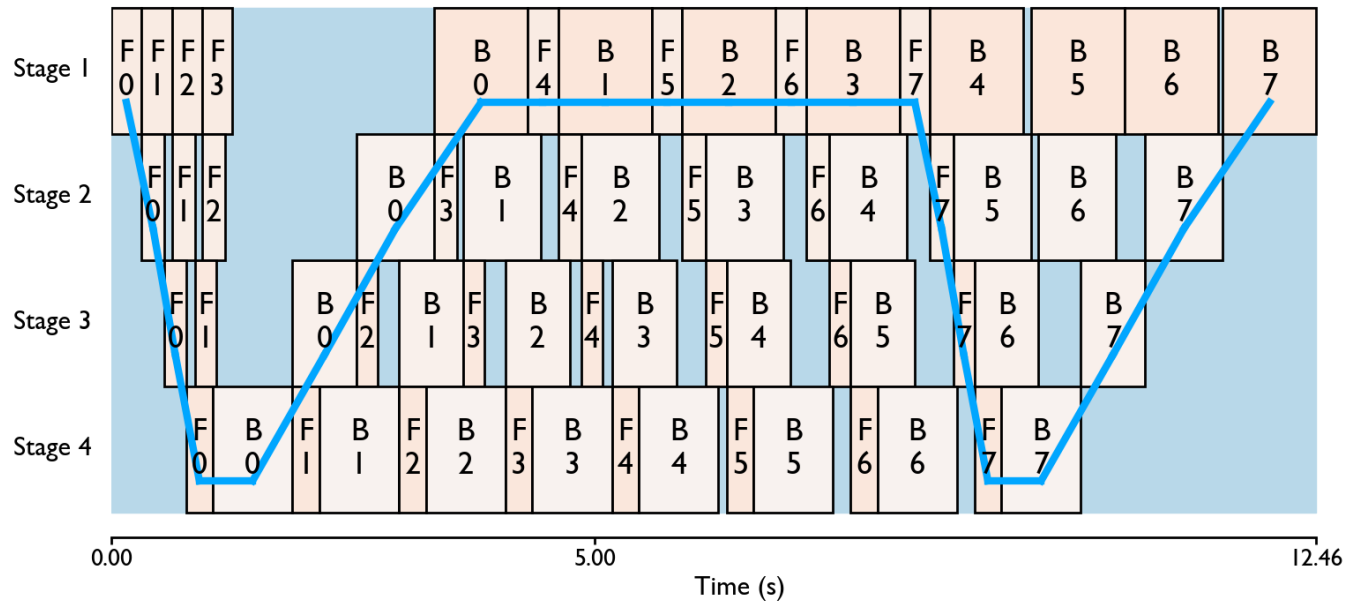
**0% slowdown, 24% less energy**



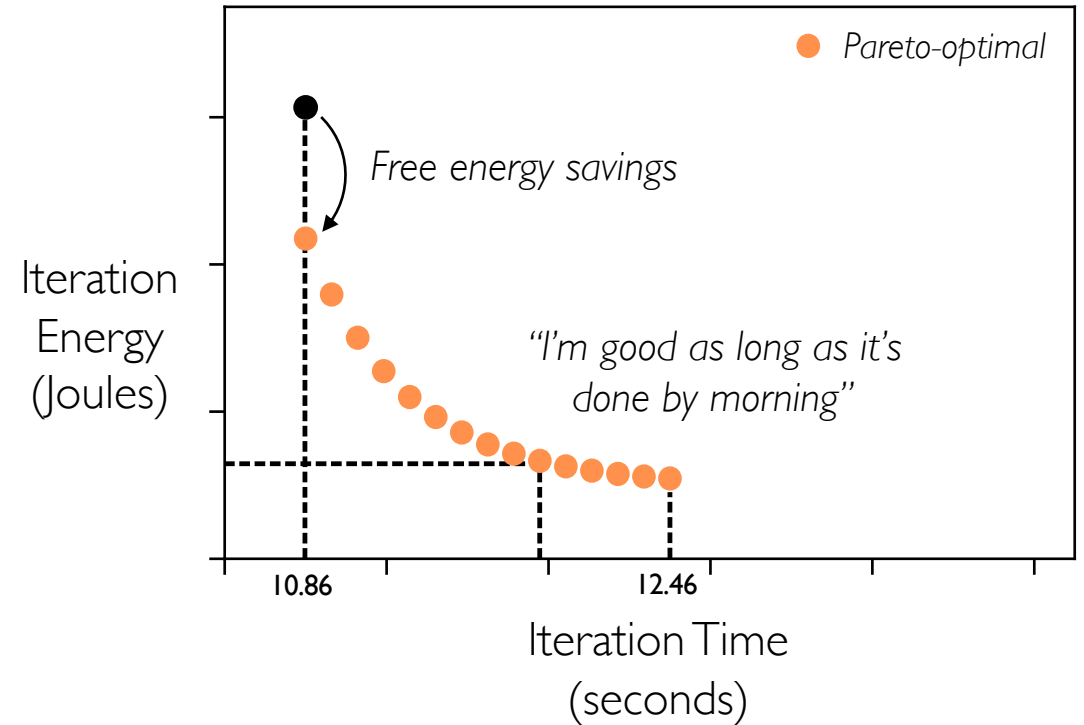
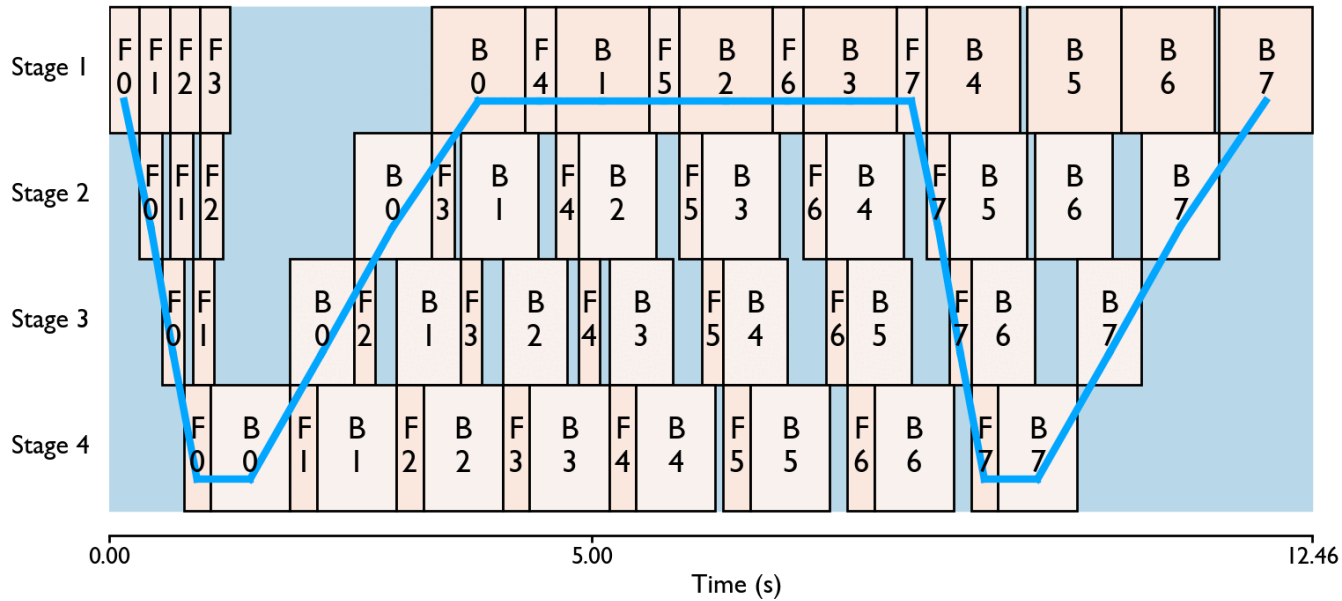
F = Forward, B = Backward

One training iteration, 4 stage 8 microbatch IFIB pipeline  
Computation drawn to scale for GPT3-large on NVIDIA A40 GPUs

# Energy-Efficient Large Model Training



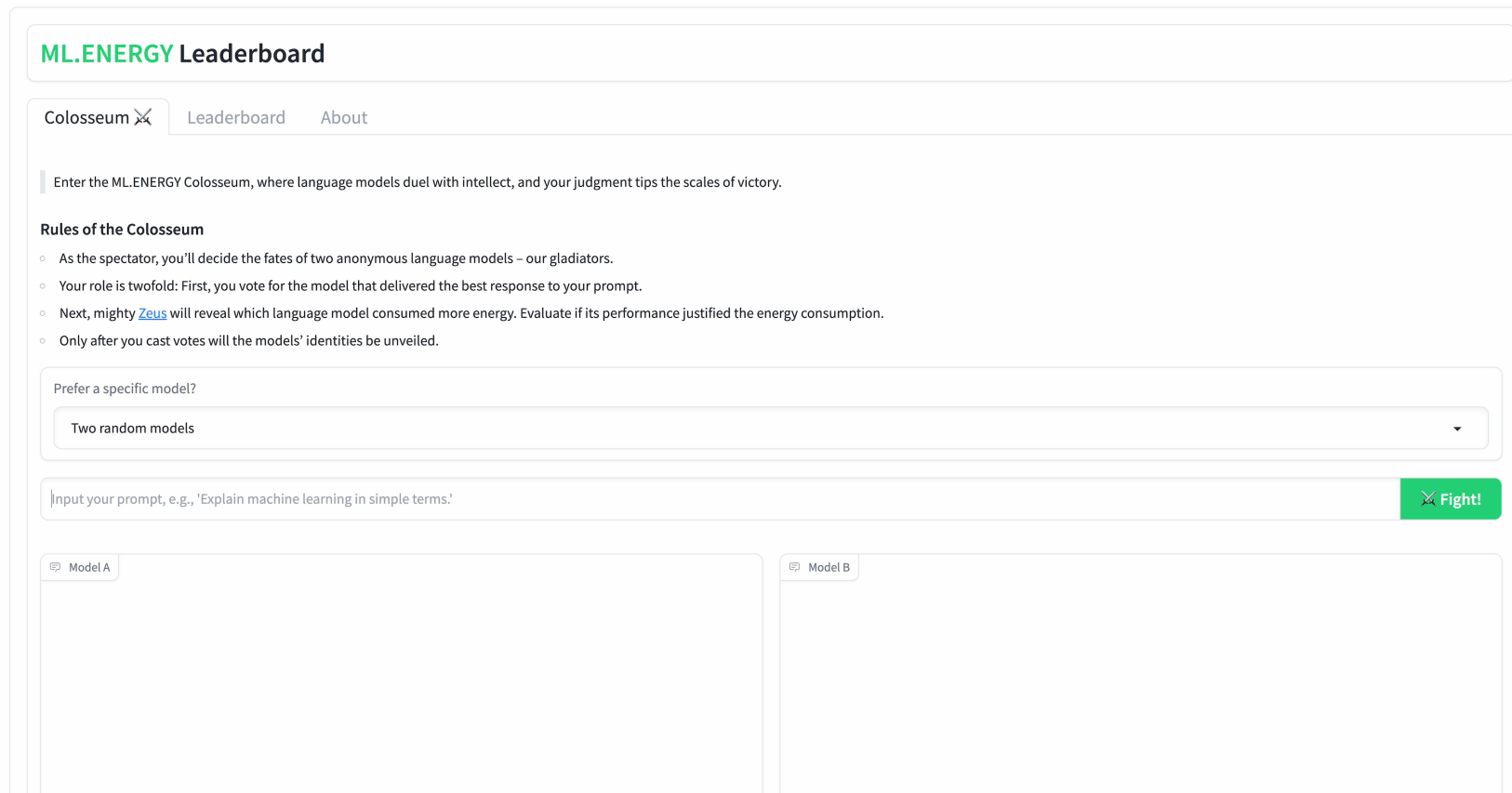
# Energy-Efficient Large Model Training



# Monitoring Real Time LLM Inference

Real time energy measurements with Zeus for LLM responses on your prompt

<https://ml.energy/leaderboard>



The screenshot shows the ML.ENERGY Leaderboard interface. At the top, there's a navigation bar with 'Colosseum' (selected), 'Leaderboard', and 'About'. Below this, a text box says 'Enter the ML.ENERGY Colosseum, where language models duel with intellect, and your judgment tips the scales of victory.' Underneath is a section titled 'Rules of the Colosseum' with four bullet points: 'As the spectator, you'll decide the fates of two anonymous language models – our gladiators.', 'Your role is twofold: First, you vote for the model that delivered the best response to your prompt.', 'Next, mighty Zeus will reveal which language model consumed more energy. Evaluate if its performance justified the energy consumption.', and 'Only after you cast votes will the models' identities be unveiled.' Below the rules is a dropdown menu labeled 'Prefer a specific model?' with the option 'Two random models' selected. At the bottom, there's a text input field for a prompt, with a green 'Fight!' button to its right. Below the input field are two empty chat boxes labeled 'Model A' and 'Model B'.

# Where We're Headed with ZEUS

<https://ml.energy/zeus>

## Generality is at our core value

- Very clear **tradeoffs** related to energy and **allowing control**
- Minimal and explicit **assumptions** on workload
- **Agnostic** to HW/SW environments