A PyTorch-Native Auto-Parallel Framework for *Ease of Use*

veScale Team

ByteDance

2024-8-22

About Me

Hongyu Zhu

Received my PhD degree from University of Toronto (advisor: Gennady Pekhimenko)

Joined ByteDance AML group in 2022.3

Currently working on LLM training frameworks

Agenda

- Why veScale
- What is veScale
- Preliminary Results of veScale
- Future of veScale

Why veScale

Company: 100s~1000s New Models Each Week

Industrial Training Framework

Only Performance

Ease of Use

Many weeks to write one model

But Current Frameworks are Hard to Use

Not PyTorch

Intertwined System and Model Design

Not Automated Enough

GradBuffer Defrag
AllReduce Overla

nn.Linear

ColumnParallelLinear

Hard to Debu

Distributed
kpoint

Intertwined Bugs

Heavy Maintainance Effort

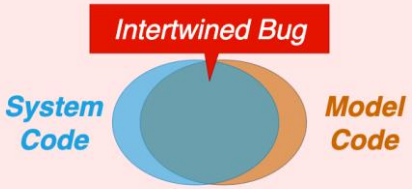
But Current Frameworks are Hard to Use

Not PyTorch

Only 8% non-PyTorch

HuggingFace Models

Intertwined System and Model Design

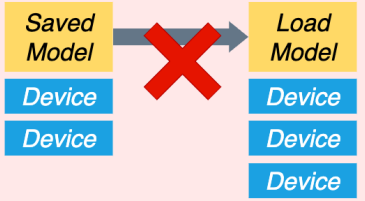


Not Automated Enough

Manual Effort

Hard to Debug

No Distributed Checkpoint



veScale

A PyTorch-Native Auto-Parallel Framework for *Ease of Use*

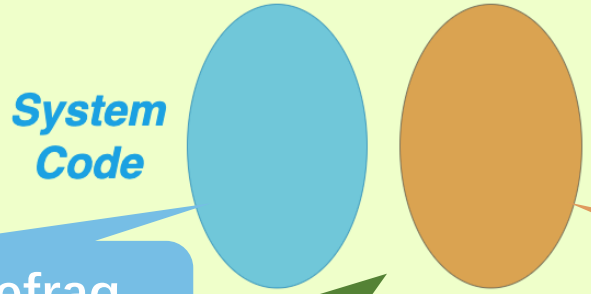
veScale

PyTorch Native

92% PyTorch

HuggingFac

Decoupled System and Model Design



GradBuffer Defrag
AllReduce Overla

nn.Linear

No Intertwining

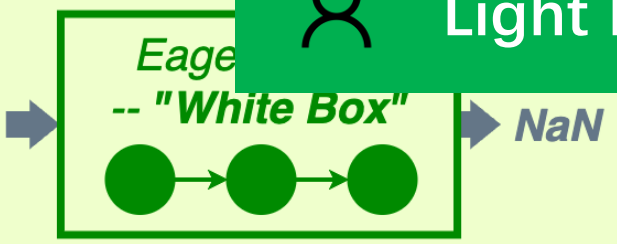
Automatic Parallelism

Tensor, Sequence, Data, ZeRO, Pipeline Parallelism

Minimal Manual Effort

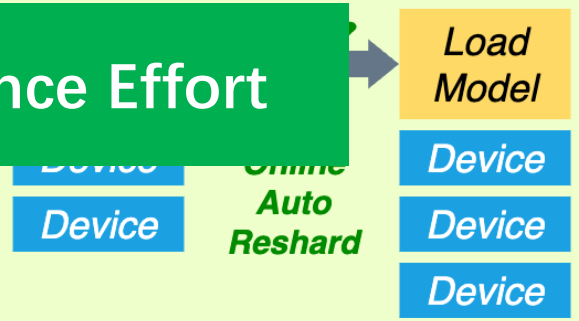
Easy to Debu

Line-by-Line Debug



Light Maintainance Effort

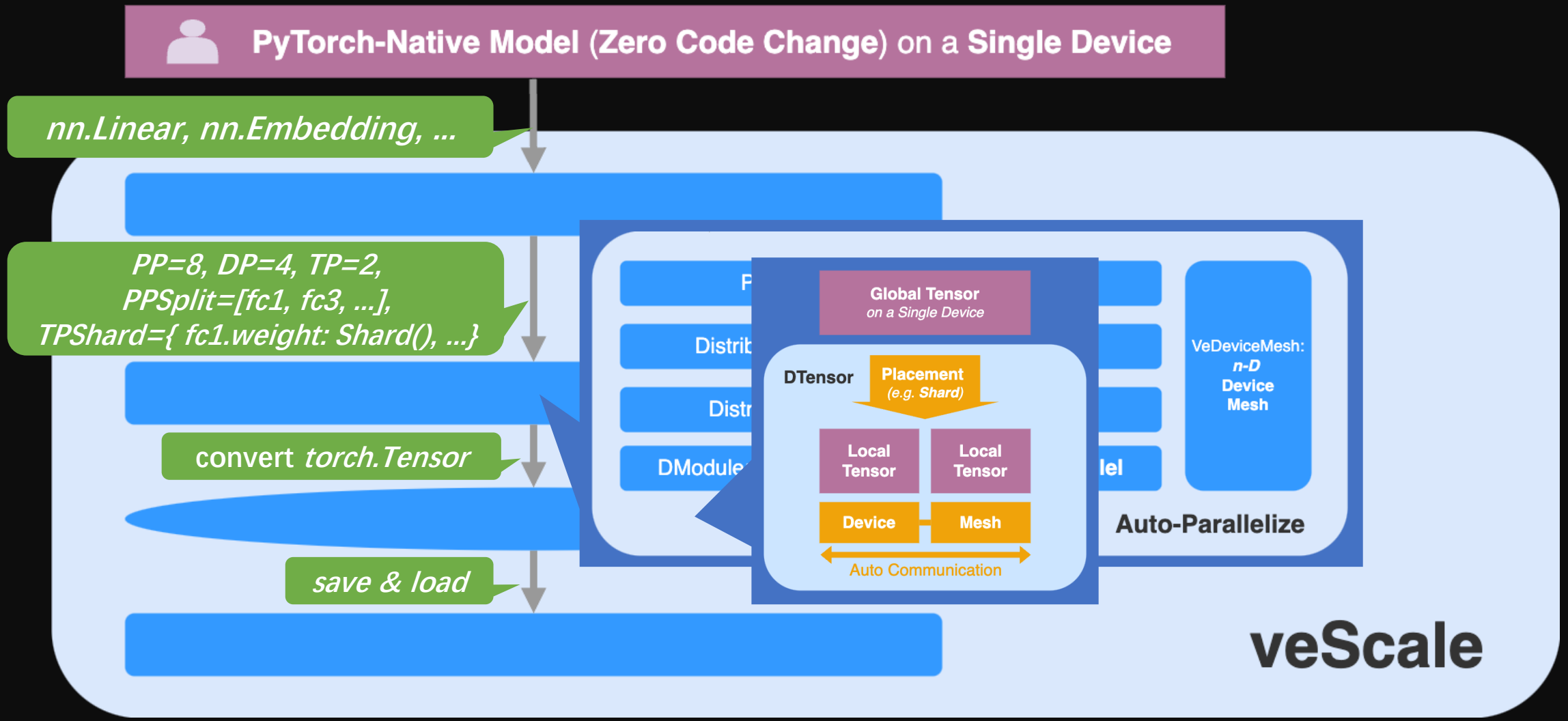
o Distributed Checkpoint



Agenda

- Why veScale
- What is veScale
- Preliminary Results of veScale
- Future of veScale

What is veScale



Agenda

- Why veScale
- What is veScale
- Preliminary Results of veScale
- Future of veScale

Preliminary Results of veScale

Simple API of nD Parallel Training (WIP)

```python
### user provides model on single device
from internal_model/huggingface.transformers import AutoConfig, AutoModel
config = AutoConfig.from_pretrained('/path/to/config')
import vescale
model = AutoModel.from_config(config)

### vescale creates nD parallel plan
plan = vescale.generate_plan(model, settings_and_constraints, ...)

### vescale creates nD parallel model
model, optimizer, ... = vescale.parallelize(plan, model, optimizer_fn, ...)

### vescale loads nD parallel model
vescale.load("/path", { "plan": plan, "model" : model, "optimizer" : optimizer})

### user trains nD parallel model as if on single device
for batch in dataloader:
    loss = model(batch)
    loss.backward()
    optimizer.step()
    optimizer.zero_grad()
    ...

### vescale saves nD parallel model
vescale.save("/path", { "plan": plan, "model" : model, "optimizer" : optimizer })
```

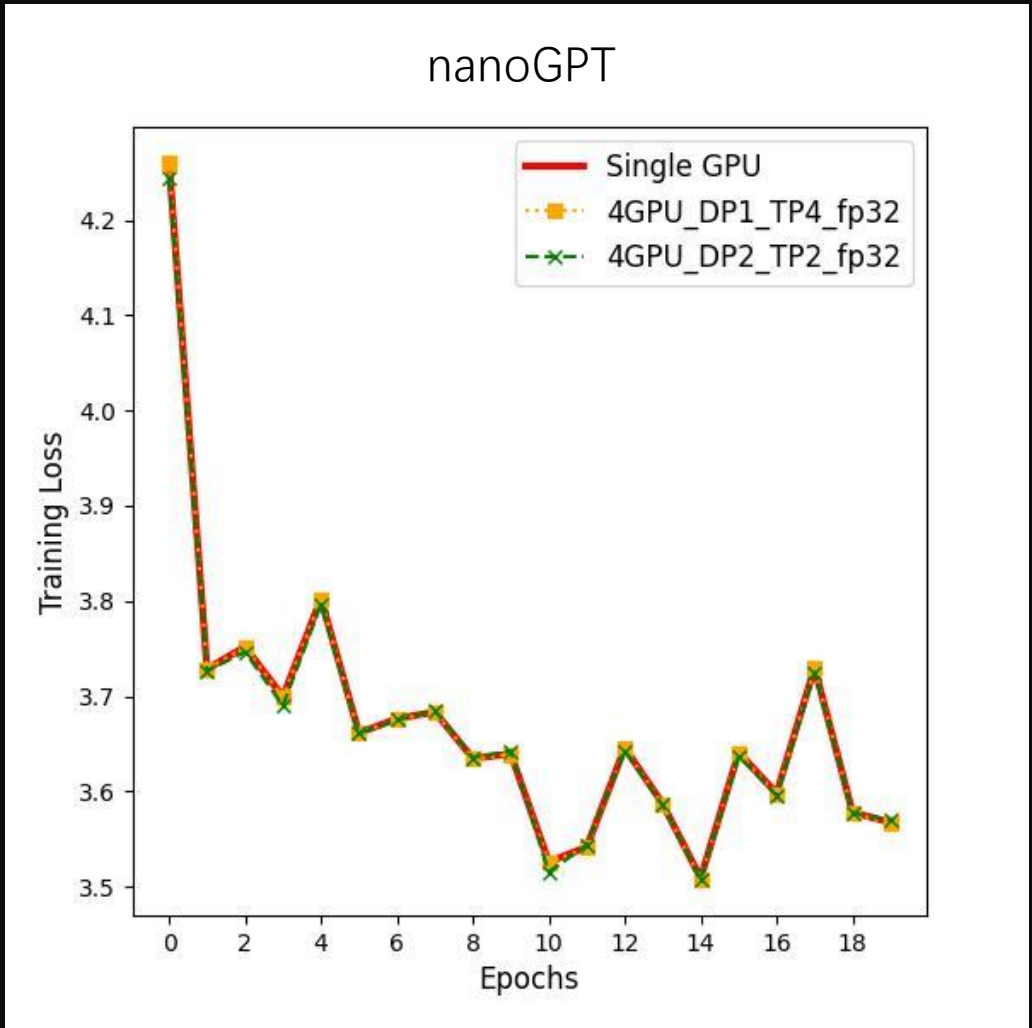
Zero Code Change of Model

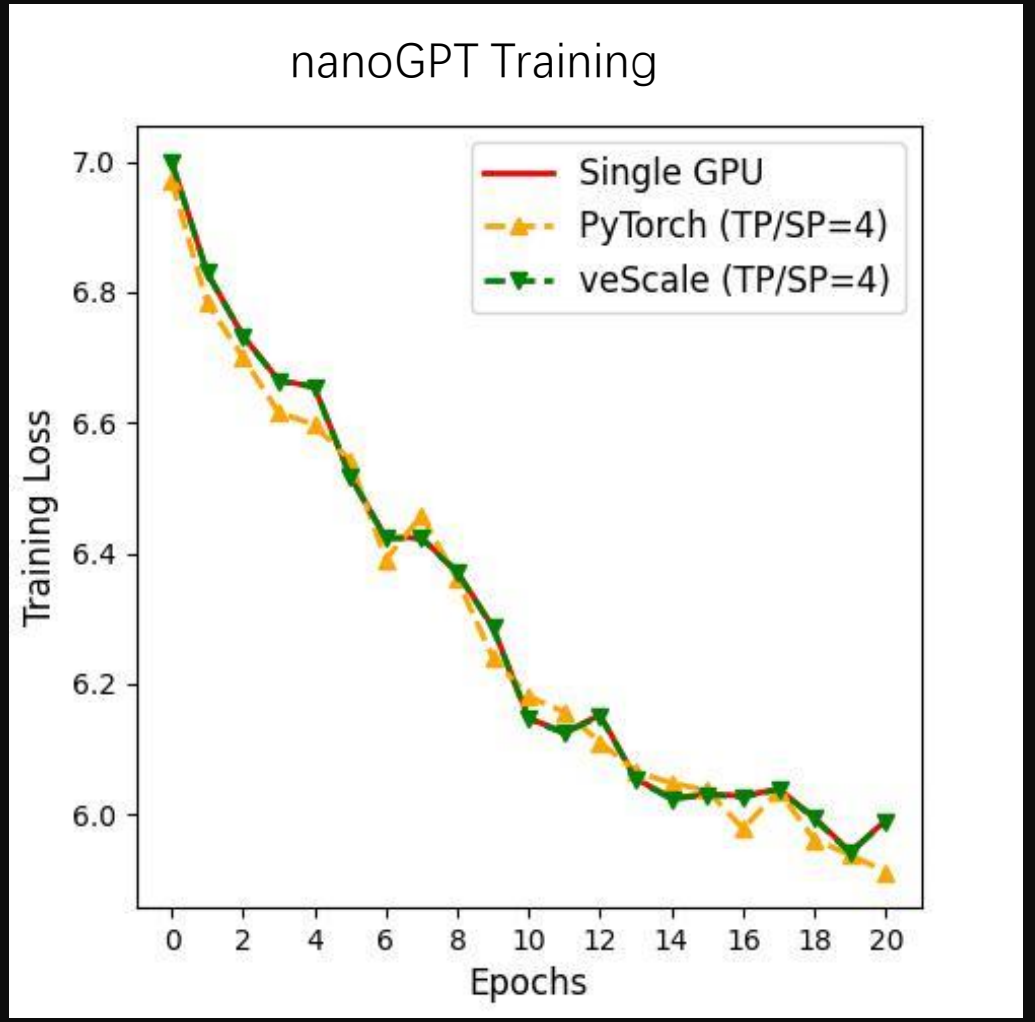
Zero Code Change of Training Loop

nD Parallel Training in 5 LoC

Preliminary Results of veScale

Bitwise Correctness of 4D Parallel Training

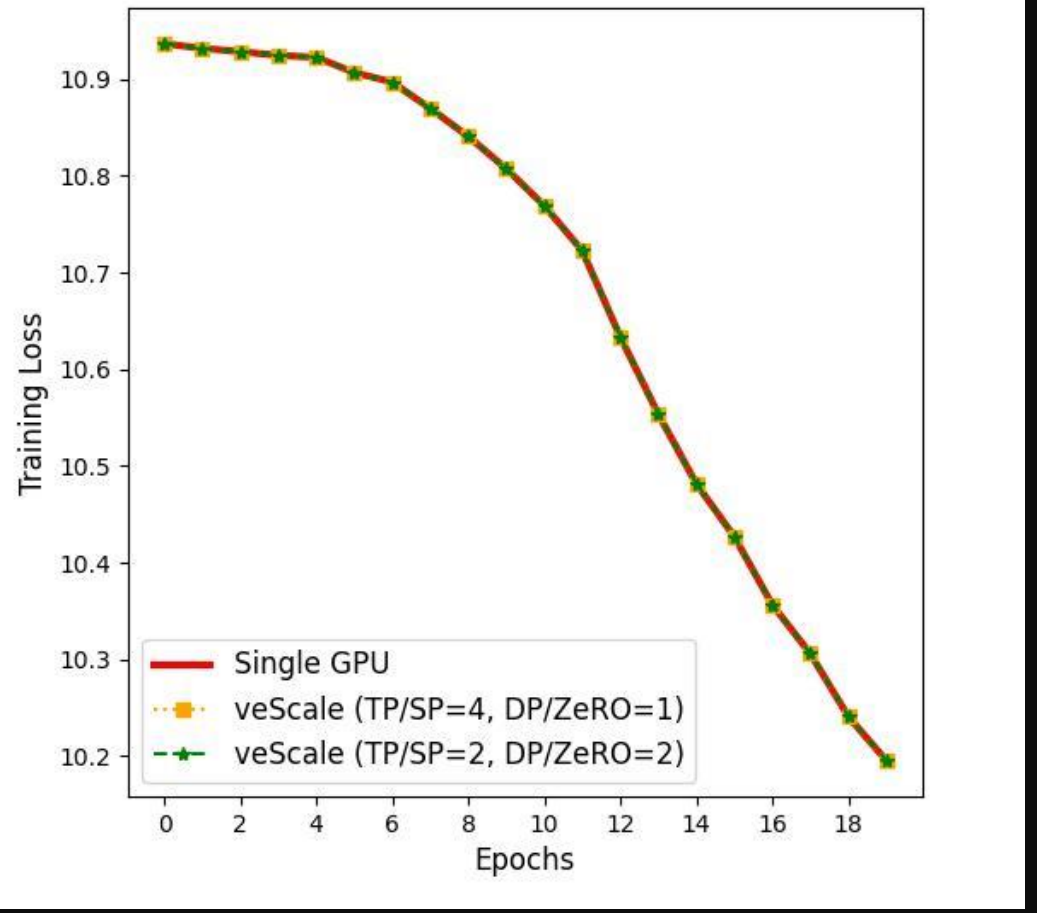




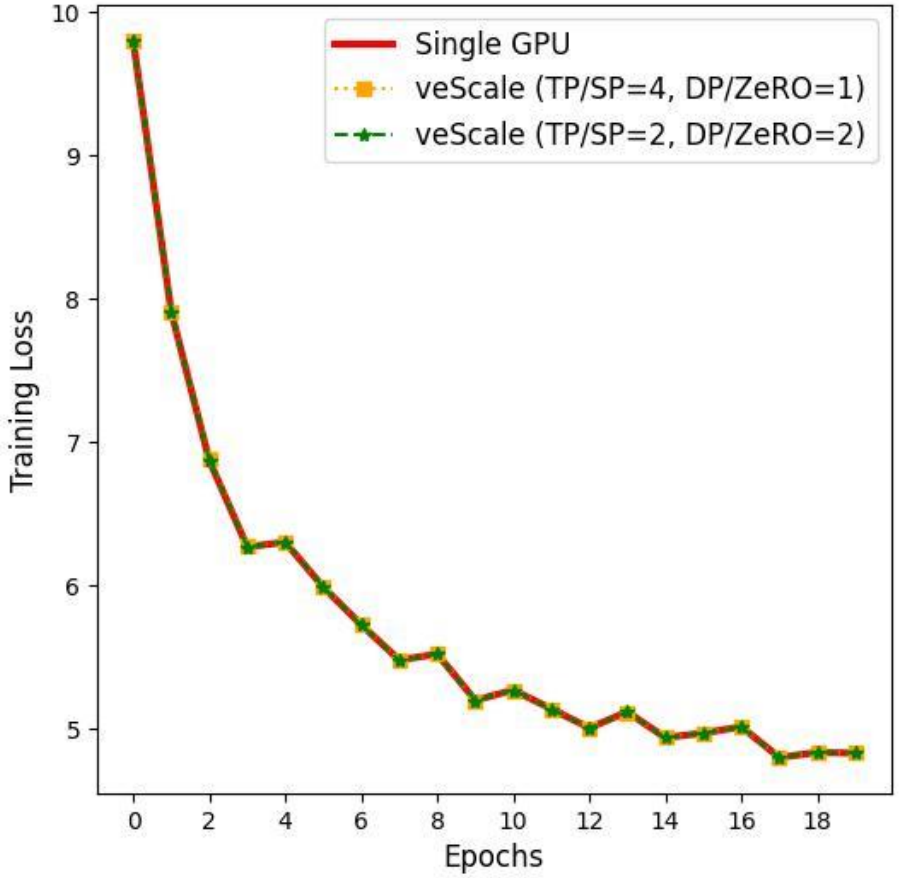
Preliminary Results of veScale

Bitwise Correctness of 4D Parallel Training

Mixtral

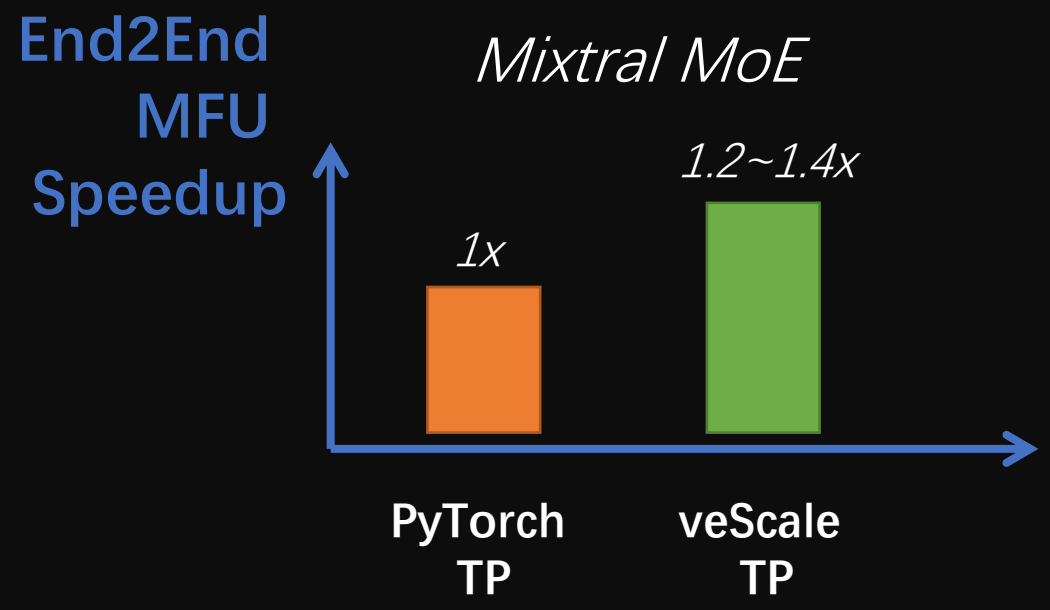


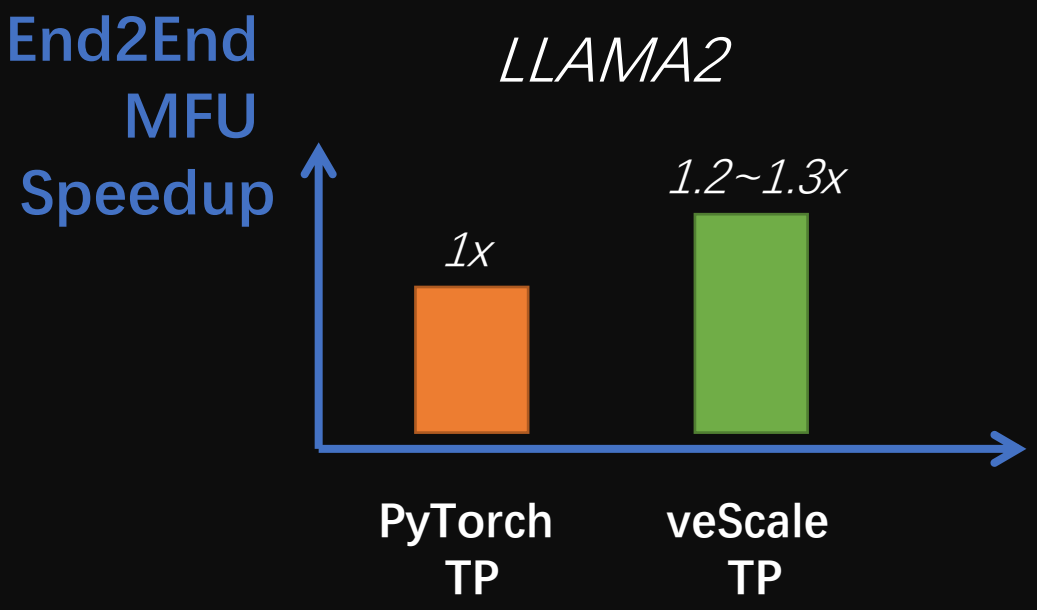
LLama2



Preliminary Results of veScale

Decent Performance of TP (WIP)



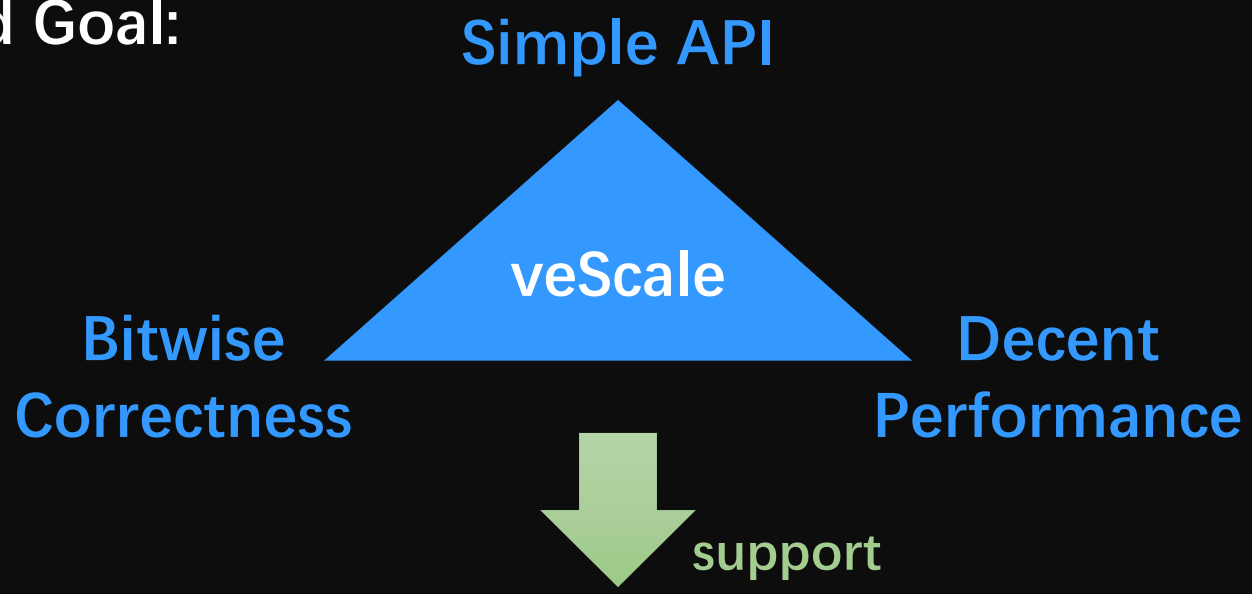


Agenda

- Why veScale
- What is veScale
- Preliminary Results of veScale
- Future of veScale

Future of veScale

End Goal:



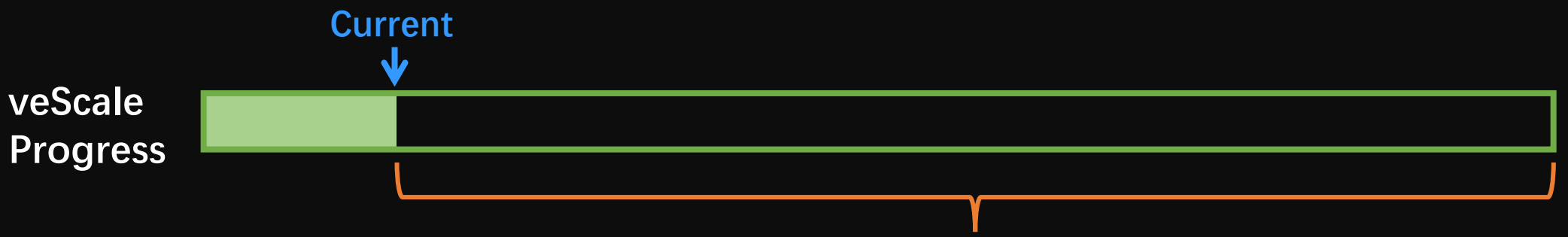
"A Promising Work!"
-- AWS AI Lab
-- Octo.AI
-- Boson AI

"An Ambitious Work!"
-- Llama Training Lead
-- PyTorch Training Lead

"But Many Effort Ahead; Long-Term Effort Ahead ..."
-- Llama Training Lead

What's next for veScale

- Better eager-mode n-D parallelism
  - Ease of use & performance
- Better fsdp2
  - Performance & fsdp2+pp+tp
- Compile mode for performance
- Auto-planner



Future Challenges

- “800” Operator Support for PyTorch
- Bitwise Correctness for “nD Parallel”
- “Easy to Use” vs “High Performance”

Acknowledgement

(random order)

[Leaders] **Li-wen Chang**, **Yanghua Peng**, **Haibin Lin**, **Xin Liu**

[Contributers] **Xinyi Di**, **Jiawei Wu**, **Hongyu Zhu**, **Ziang Song**, **Jiacheng Yang**, **Youjie Li**

[Collaborators] **Minji Han**, **Chengji Yao**, **Chenyuan Wang**, **Yan Xu**, **Changming Yu**, **Wenlei Bao**, **Hao Gong**, **Ming Zhang**, **Ningxin Zheng**, **Xuanrun Zhang**

veScale

Open Source for All

vescale.xyz