为PyTorch原生模型提供自动并行的训练框架

veScale Team

ByteDance

2024-8-22

关于我

朱虹宇

PhD毕业于多伦多大学
（导师：Gennady Pekhimenko）

2022年3月加入字节跳动

目前主要专注于大语言模型训练架构相关工作

议程

- 为什么需要VeScale
- VeScale设计与实现
- 初步测试结果
- 未来展望

为什么需要VeScale

Company: 100s~1000s New Models Each Week

Industrial Training Framework

Only Performance

Ease of Use

单个model
往往需要数
周时间开发

当前的框架的使用痛点

非PyTorch

系统代码与
模型代码纠缠

自动化程度低

GradBuffer Defrag
AllReduce Overla

nn.Linear

ColumnParallelLinear

Debug难度

相互纠缠的bu

无分布式
checkpoint

人力维护成本繁重

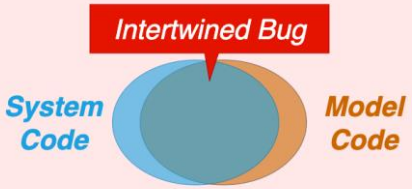
当前的框架的使用痛点

Not PyTorch

Only 8% non-PyTorch

HuggingFace Models

Intertwined System and Model Design

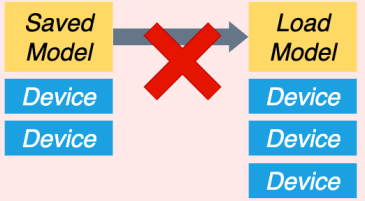


Not Automated Enough

Manual Effort

Hard to Debug

No Distributed Checkpoint



veScale

为实现PyTorch原生模型的自动并行的训练框架

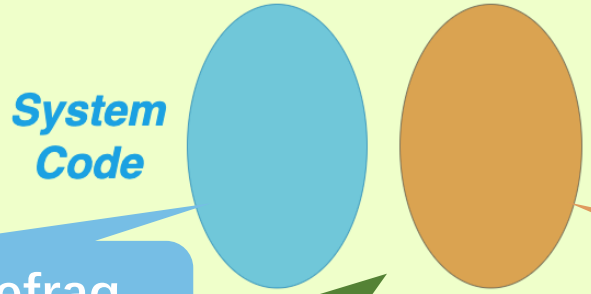
veScale

PyTorch Native

92% PyTorch

HuggingFac

Decoupled System and Model Design



GradBuffer Defrag
AllReduce Overla

nn.Linear

No Intertwining

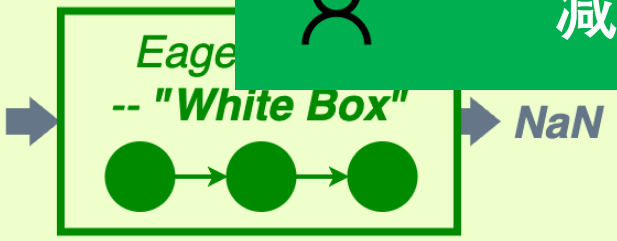
Automatic Parallelism

Tensor, Sequence, Data, ZeRO, Pipeline Parallelism

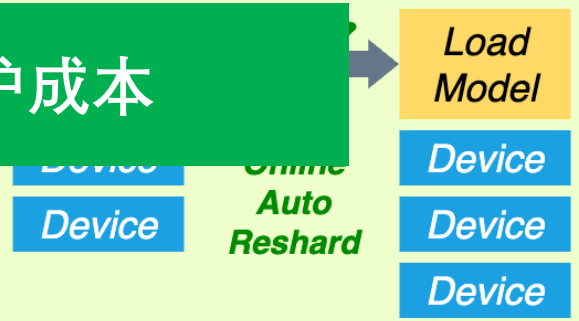
Minimal Manual Effort

Easy to Debu

Line-by-Line Debug



o Distributed Checkpoint

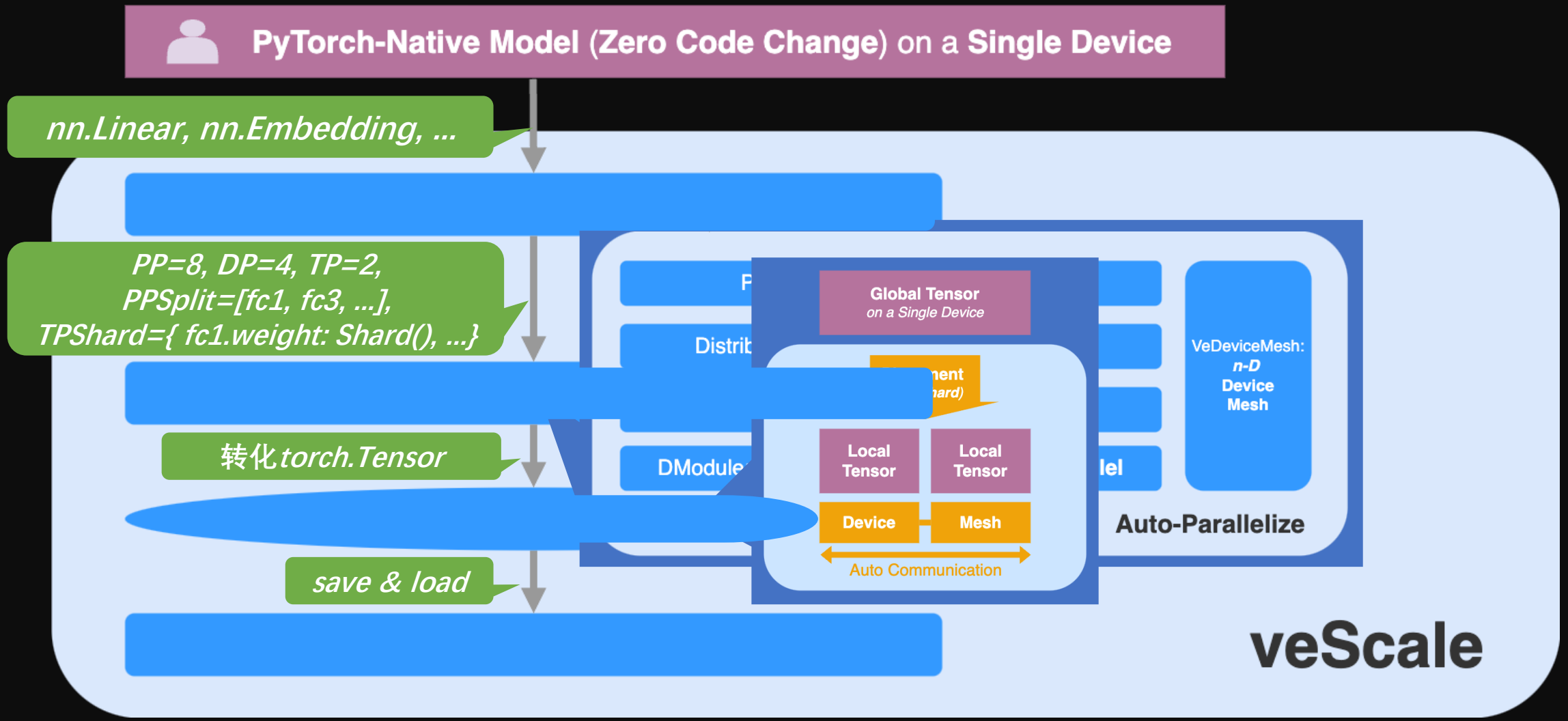


减少人力维护成本

议程

- 为什么需要VeScale
- VeScale设计与实现
- 初步测试结果
- 未来展望

VeScale设计与实现



议程

- 为什么需要VeScale
- VeScale设计与实现
- 初步测试结果
- 未来展望

VeScale用户代码Demo

简易的多维度并行训练API (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 })
```

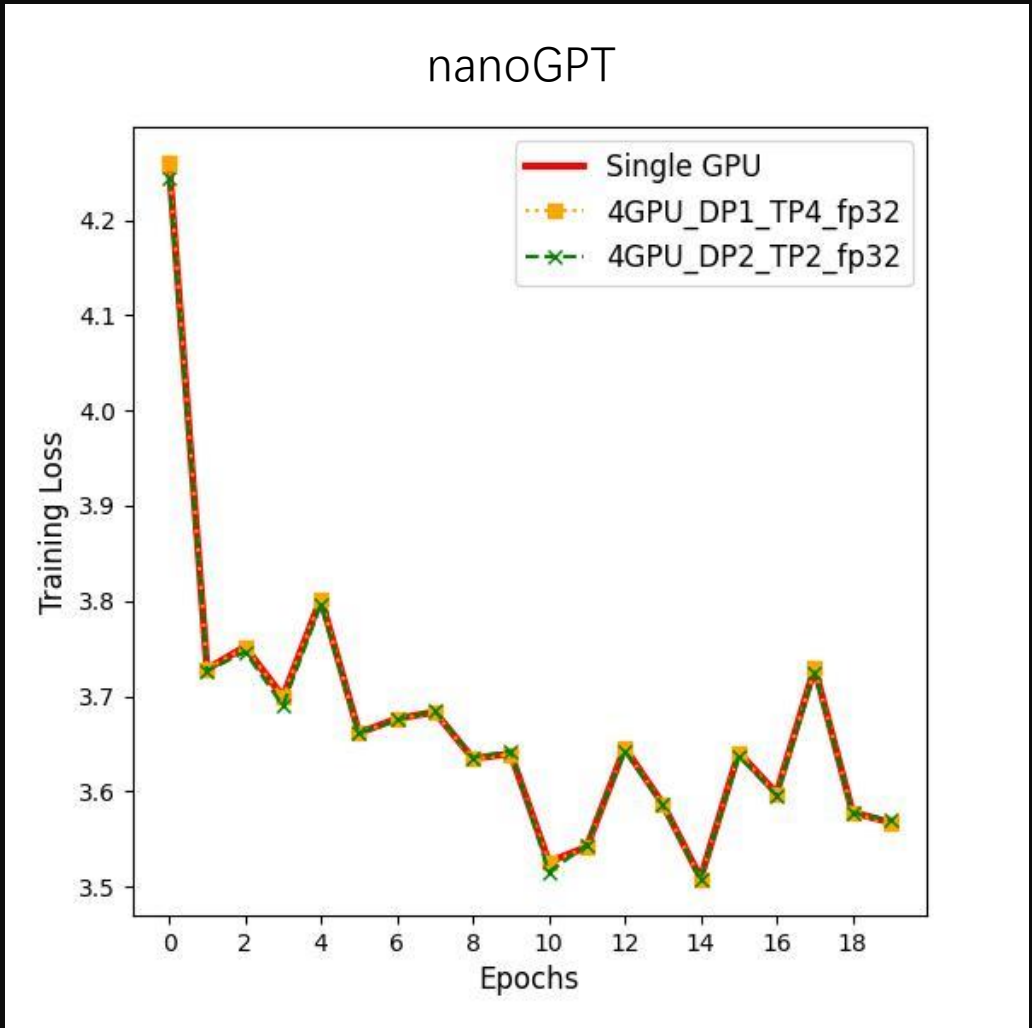
模型代码零改动

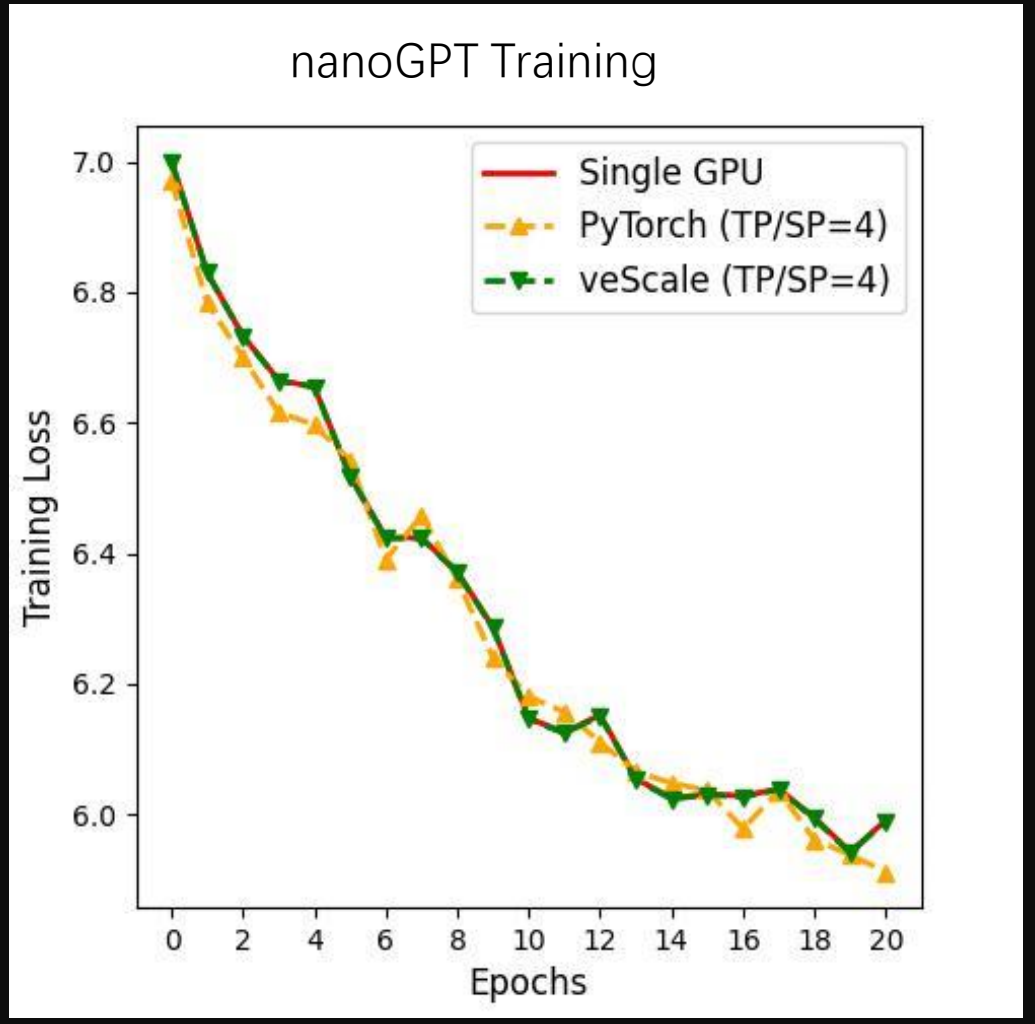
训练代码零改动

5行代码实现
多维度并行

VeScale初步测试结果

4D分布式训练下的Bitwise正确性

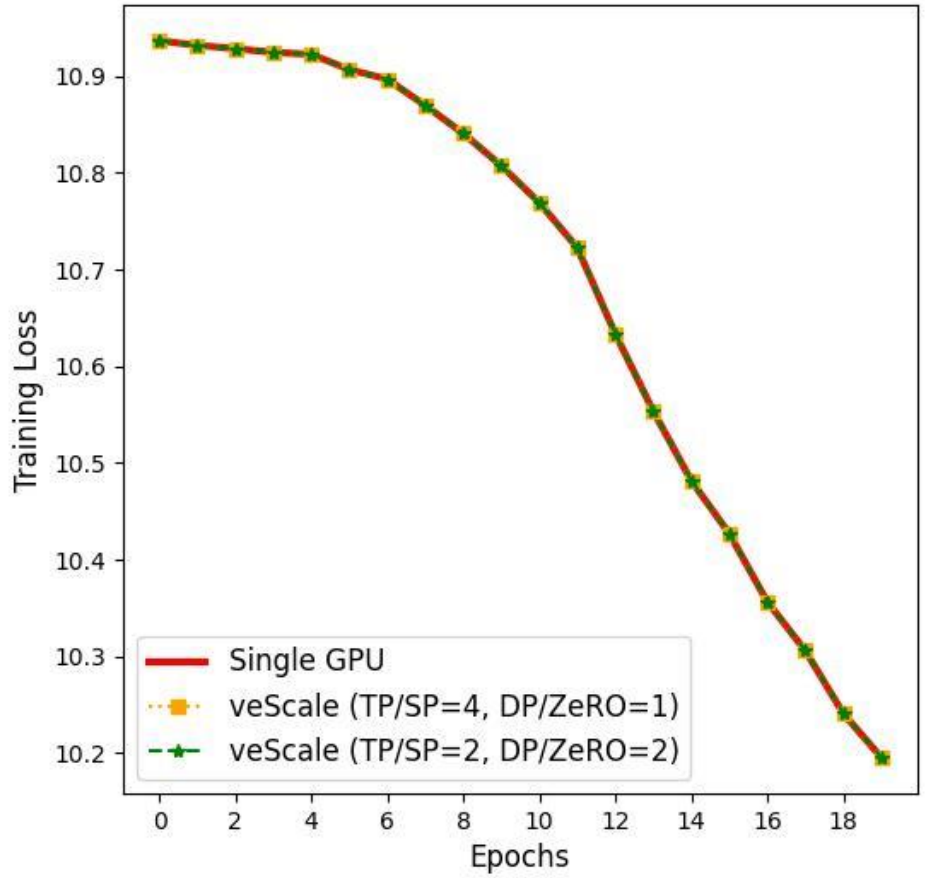




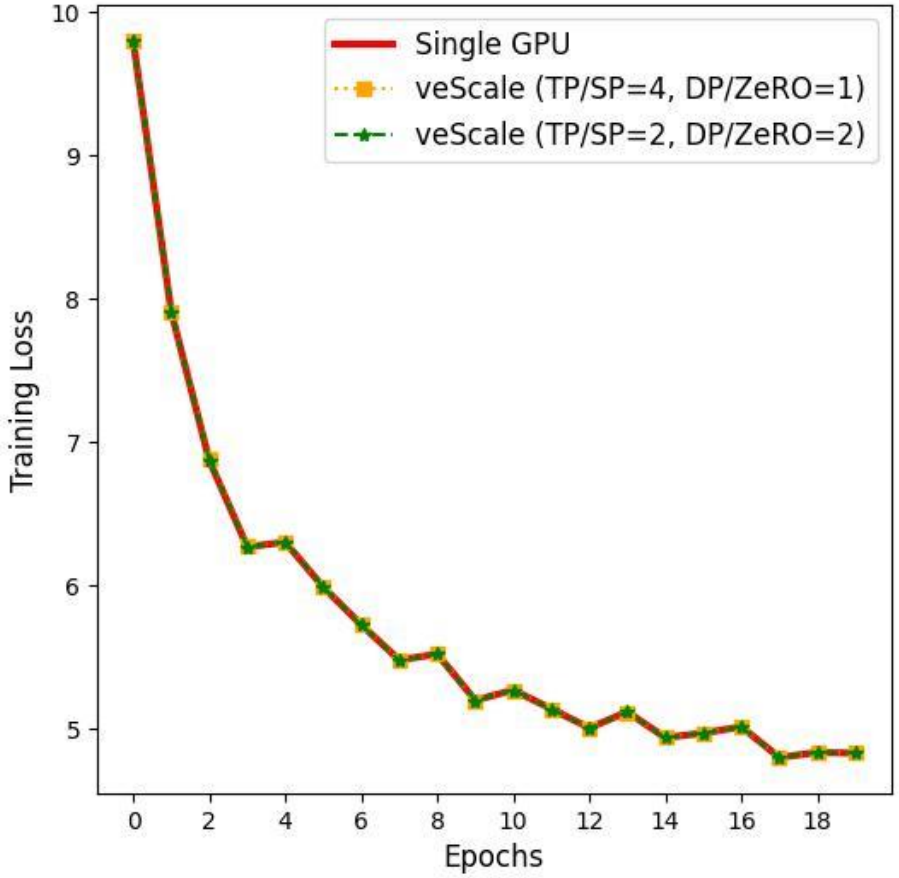
VeScale初步测试结果

4D分布式训练下的Bitwise正确性

Mixtral

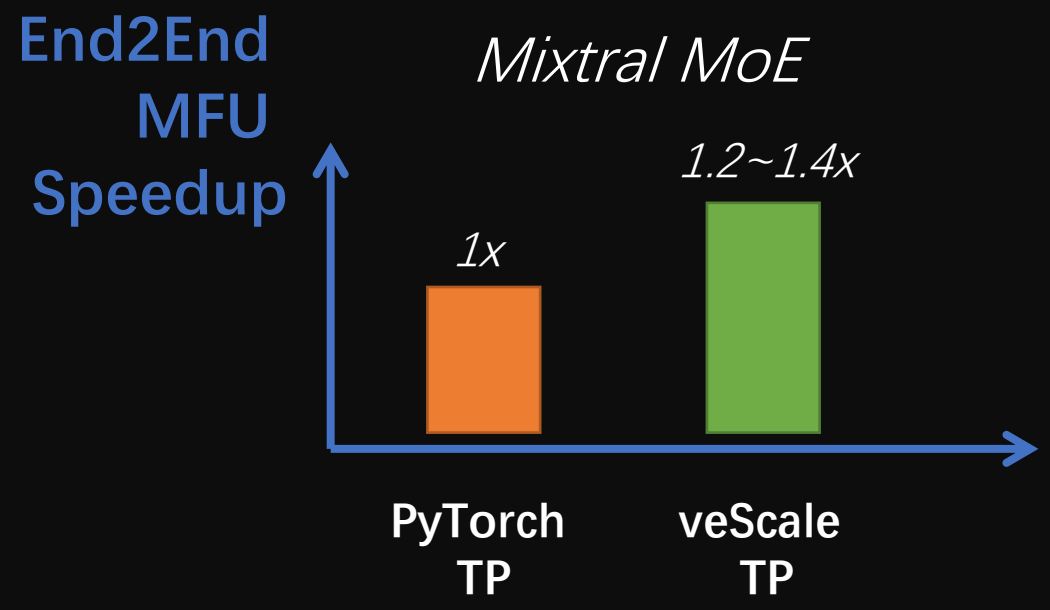


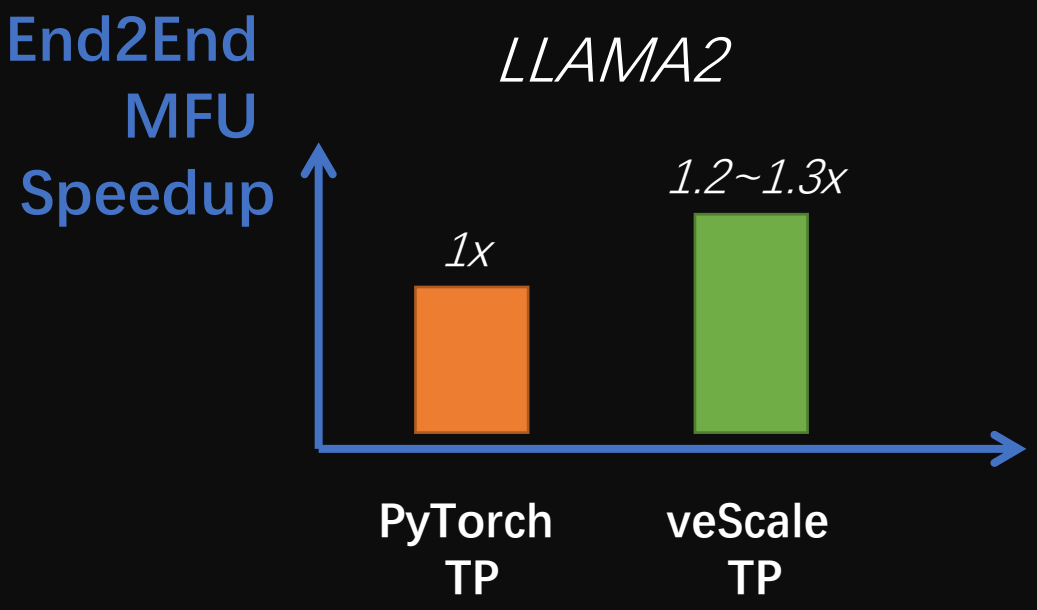
LLama2



VeScale初步测试结果

Tensor Parallelism的性能优势 (WIP)



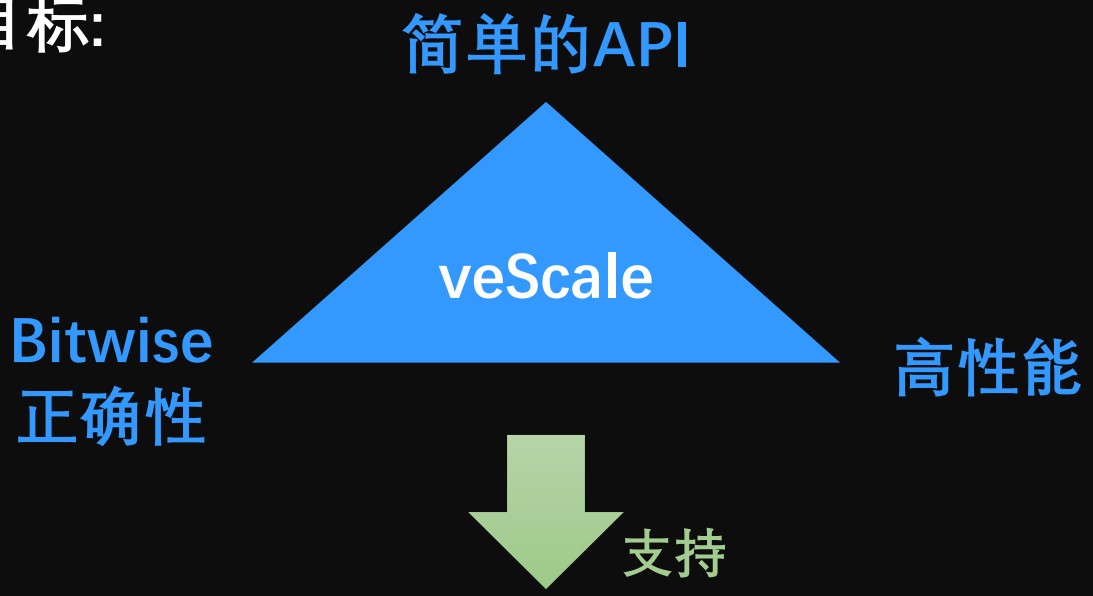


议程

- 为什么需要VeScale
- VeScale设计与实现
- 初步测试结果
- 未来展望

未来展望

最终目标:



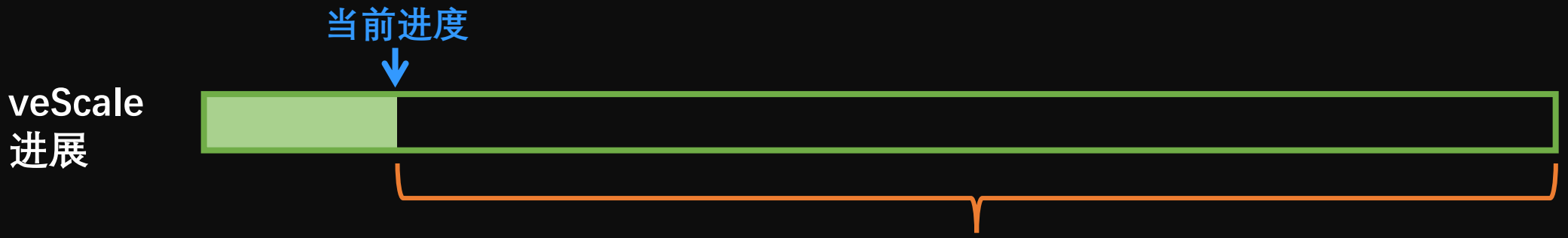
"A Promising Work!"
-- AWS AI Lab
-- Octol AI
-- Boson AI

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

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

VeScale的下一步计划

- Eager模式下的易用性和性能提升
- 更强的fsdp2（性能，易用性以及fsdp2+pp+tp支持）
- 更强的Compile支持
- 自动生成多维并行Plan



未来的挑战

多达800个PyTorch算子支持

多维度并行下的bitwise正确性

易用性与性能权衡

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