

PR #6916 完整报告

verl-project/verl

[trainer] fix: Update latest TorchTitanEngine

合并时间: 2026-07-06 10:11

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6916>

执行摘要

- 一句话: 更新 TorchTitan 引擎并添加端到端测试脚本
- 推荐动作: 建议团队确认默认值变更后通知用户更新 torchtitan 版本; 尽快修复测试脚本中的 Shell 问题以提高稳健性; 后续可基于 spmd_types 做进一步的性能调优, 并跟进 CI 集成。

功能与动机

为了支持最新的 torchtitan nightly 版本, 需要更新引擎 API 兼容性, 并引入 spmd_types 后端以获得更好性能。同时为后续 CI 集成提供端到端测试脚本。

实现拆解

1. 在 verl/workers/config/engine.py 的 TorchTitanEngineConfig 中新增 spmd_backend 和 activation_checkpoint 配置字段, 并在 __post_init__ 中添加合法性断言。
2. 在 verl/workers/engine/torchtitan/transformer_impl.py 中更新 OptimizersContainer.Config 使用新的 ParamGroupConfig API, 将 spmd_backend 传入 ParallelismConfig, 并根据 activation_checkpoint 配置映射到对应的 AC 策略 (SelectiveAC.Config、FullAC.Config 或 None)。
3. 在 verl/workers/engine/torchtitan/utils.py 的 get_attention_masks 中支持 flex_flash 注意力类型, 复用 flex 的 BlockMask。
4. 更新 verl/trainer/config/engine/torchtitan.yaml 和 _generated_ppo_torchtitan_trainer.yaml, 添加新的配置项默认值。
5. 新增 tests/special_e2e/run_ppo_trainer_torchtitan.sh 端到端测试脚本, 允许通过环境变量配置并行度、注意力类型、SPMD 后端和激活检查点。

关键文件:

- verl/workers/engine/torchtitan/transformer_impl.py (模块 引擎实现; 类别 source; 类型 core-logic; 符号 init) : 核心引擎实现, 更新 OptimizersContainer API、集成 spmd_backend 和 activation_checkpoint 配置, 驱动整个 TorchTitan 训练流程。
- verl/workers/config/engine.py (模块 配置; 类别 source; 类型 core-logic; 符号 TorchTitanEngineConfig.post_init) : TorchTitanEngineConfig 新增 spmd_backend 和 activation_checkpoint 字段及验证逻辑, 是配置入口。

- tests/special_e2e/run_ppo_trainer_torchtitan.sh (模块 测试; 类别 test; 类型 test-coverage) : 新增的端到端测试脚本, 覆盖 TorchTitan 引擎的多维度配置, 为 CI 集成提供基础。
- verl/workers/engine/torchtitan/utils.py (模块 工具函数; 类别 source; 类型 core-logic ; 符号 get_attention_masks) : 支持 flex_flash 注意力类型, 扩展注意力 mask 生成逻辑。
- verl/trainer/config/engine/torchtitan.yaml (模块 配置文件; 类别 config; 类型 configuration) : 配置文件新增 spmd_backend 和 activation_checkpoint 默认值, 是用户可见的配置接口。
- verl/trainer/config/_generated_ppo_torchtitan_trainer.yaml (模块 配置文件; 类别 config; 类型 configuration) : 自动生成的训练配置, 同步更新新配置项以确保一致性。

关键符号: TorchtitanEngine.init, get_attention_masks, TorchtitanEngineConfig.post_init

关键源码片段

verl/workers/engine/torchtitan/transformer_impl.py

核心引擎实现, 更新 OptimizersContainer API、集成 spmd_backend 和 activation_checkpoint 配置, 驱动整个 TorchTitan 训练流程。

```
# verl/workers/engine/torchtitan/transformer_impl.py (head)

# 使用新的 ParamGroupConfig 构建 optimizer 配置
optimizer = OptimizersContainer.Config(
    param_groups=[
        ParamGroupConfig(
            pattern=r".*", # 匹配所有参数
            optimizer_name=self.optimizer_config.name,
            optimizer_kwargs={
                "lr": self.optimizer_config.lr,
                "eps": self.optimizer_config.eps,
                "betas": (self.optimizer_config.betas[0], self.optimizer_config.betas[1]),
                "weight_decay": self.optimizer_config.weight_decay,
            },
        )
    ],
)

# ...

# 将 spmd_backend 传入 parallel 配置
parallelism = ParallelismConfig(
    # ... 其他参数 ...
    spmd_backend=self.engine_config.spmd_backend, # 新增
)

# 根据 activation_checkpoint 配置映射对应的 AC 策略
# 注意: spmd_backend="spmd_types" 且 eager 下选择性 / 全 AC 会崩溃, 需设为 "none"
```

```

activation_checkpoint = {
    "selective": SelectiveAC.Config,
    "full": FullAC.Config,
    "none": lambda: None, # 不启用 AC
}[self.engine_config.activation_checkpoint]()

# 将 activation_checkpoint 传入 Trainer.Config
self.config = Trainer.Config(
    # ...
    activation_checkpoint=activation_checkpoint,
)

```

verl/workers/config/engine.py

TorchTitanEngineConfig 新增 spmd_backend 和 activation_checkpoint 字段及验证逻辑，是配置入口。

```

# verl/workers/config/engine.py (head, TorchTitanEngineConfig 部分)
@dataclass
class TorchTitanEngineConfig(EngineConfig):
    # ... 其他字段 ...

    # SPMD 后端: "default" | "full_dtensor" | "spmd_types"
    spmd_backend: str = "spmd_types"
    # 激活检查点模式: "selective" | "full" | "none"
    activation_checkpoint: str = "selective"

    def __post_init__(self):
        super().__post_init__()
        # 校验 attn_type 仅支持 "flex"、"flex_flash" 和 "varlen"
        assert self.attn_type in ["flex", "flex_flash", "varlen"], (
            f"attn_type {self.attn_type} not supported"
        )
        # 校验 spmd_backend 合法值
        assert self.spmd_backend in ["default", "full_dtensor", "spmd_types"], (
            f"spmd_backend {self.spmd_backend} not supported"
        )
        # 校验 activation_checkpoint 合法值
        assert self.activation_checkpoint in ["selective", "full", "none"], (
            f"activation_checkpoint {self.activation_checkpoint} not supported"
        )

```

tests/special_e2e/run_ppo_trainer_torchTitan.sh

新增的端到端测试脚本，覆盖 TorchTitan 引擎的多维度配置，为 CI 集成提供基础。

```

# tests/special_e2e/run_ppo_trainer_torchTitan.sh (head)
#!/usr/bin/env bash
set -xeuo pipefail

# 模型和数据路径

```

```

MODEL_ID=${MODEL_ID:-Qwen/Qwen3-0.6B}
MODEL_PATH=${MODEL_PATH:-${HOME}/models/${MODEL_ID}} # 存在未引用问题

# torchtitan 并行配置
FSDP_SIZE=${FSDP_SIZE:-1}
TP_SIZE=${TP_SIZE:-1}
EP_SIZE=${EP_SIZE:-1}

# 注意力后端: flex | flex_flash | varlen
ATTN_TYPE=${ATTN_TYPE:-flex}
# 激活检查点模式: selective | full | none
AC_MODE=${AC_MODE:-none}
# SPMD 后端: default | full_dtensor | spmd_types
SPMD_BACKEND=${SPMD_BACKEND:-spmd_types}

common_params=(
    model_engine=torchtitan
    # ... 其他参数 ...
    actor_rollout_ref.actor.torchtitan.attn_type="${ATTN_TYPE}"
    actor_rollout_ref.actor.torchtitan.activation_checkpoint="${AC_MODE}"
    actor_rollout_ref.actor.torchtitan.spmd_backend="${SPMD_BACKEND}"
    # ...
)

python3 -m verl.trainer.main_ppo "${common_params[@]}" "$@" # 未引用 $@ 问题

```

评论区精华

Review 中 `gemini-code-assist[bot]` 指出新增的测试脚本存在 Shell 脚本最佳实践问题:

- `$HOME` 变量未加引号可能导致路径解析错误。
- `trainer.logger` 的方括号未引号包裹, `bash` 可能执行 `glob` 展开。
- `$@` 未加引号会破坏包含空格的参数。这些建议已记录但未在 PR 中修改, 合并者已批准, 可能留待后续修复。
- Shell 脚本中 `$HOME` 未引用 (correctness): 未在 PR 中修改, 仍为 `${HOME}`, 但合并者已批准, 可能后续修复。
- Shell 脚本中 `trainer.logger` 方括号未引用 (correctness): 未修改, 同样未解决。
- Shell 脚本中 `$@` 未引用 (correctness): 未修改, 仍为 `$@`。

风险与影响

- 风险:
 - `spmd_backend` 默认改为 `spmd_types`, 旧配置 (未显式设置且依赖 `DTensor` 行为) 可能不兼容, 需要用户更新 `torchtitan` 版本。
 - `activation_checkpoint` 在 `spmd_types` + `eager` 模式下可能导致崩溃 (已在注释中说明), 用户需设为 `none` 或启用 `torch.compile`。

- 测试脚本中未引用的变量和数组可能在高版本 bash 或特殊路径下出错。
- 影响：影响所有使用 TorchTitan 引擎的用户：需要升级 torchtitan 到 nightly ($\geq 0.6.24$)，并可能需要调整配置（默认值变更）。新配置项提供更细粒度的控制，性能预期与 DTensor 持平或更优。新增的端到端测试脚本为 CI 集成提供了基础。
- 风险标记：配置默认变更，测试脚本稳健性

关联脉络

- 暂无明显关联 PR