# PR #1991 完整报告

THUDM/slime

[ci] Add e2e test for delta weight update

合并时间：2026-05-30 11:55

原文链接：http://prhub.com.cn/THUDM/slime/pull/1991

## 执行摘要

- 一句话：为 delta 权重更新添加端到端 CI 测试
- 推荐动作：建议合并。该 PR 不仅增强了 CI 覆盖，还修复了一个潜在的版本同步 Bug，设计决策（如 set_weight_version 方法、_published_any 标志）值得在类似场景中借鉴。

## 功能与动机

确保 delta 权重更新功能在 CI 中得到验证，避免零 diff 场景（如全零梯度）导致引擎版本与更新器不匹配，进而触发 CI 版本一致性检查失败。

## 实现拆解

1. 新增端到端测试文件(tests/test_delta_weight_update.py): 包含 prepare 和 execute 两个函数，自动下载模型和数据集，转换 checkpoint，然后使用临时目录运行一次训练，验证 delta 目录下生成 .safetensors 文件。
2. 新增 sglang 引擎方法(slime/backends/sglang_utils/sglang_engine.py): 添加 set_weight_version 方法，通过 HTTP 请求更新引擎记录的权重版本，无需重新加载权重。
3. 修改 delta 更新器(slime/backends/megatron_utils/update_weight/update_weight_from_distributed_delta.py): 新增 _published_any 标志，在种子调用时和 _finalize_sync 结束时调用引擎的 set_weight_version，确保版本同步。
4. 更新 CI 配置(.github/workflows/pr-test.yml 和 pr-test.yml.j2): 将测试添加到 4 GPU 测试矩阵中，同时清理了一个重复的 r3 测试条目。

关键文件：

- tests/test_delta_weight_update.py（模块 测试；类别 test；类型 test-coverage；符号 prepare, execute）：新增的端到端测试文件，验证 delta 权重更新的完整流程，是 PR 的核心。
- slime/backends/sglang_utils/sglang_engine.py（模块 推理引擎；类别 source；类型 core-logic；符号 set_weight_version）：新增 set_weight_version 方法，允许手动设置引擎的权重版本，用于零 diff 场景。
- slime/backends/megatron_utils/update_weight/update_weight_from_distributed_delta.py（模块 训练后端；类别 source；类型 core-logic；符号 _published_any, update_weights, _finalize_sync）：Delta 权重更新器的核心逻辑修改，添加 _published_any 标志并在零 diff 时同步版本。

- .github/workflows/pr-test.yml（模块 CI；类别 infra；类型 infrastructure）：将测试添加到 CI 的 4 GPU 测试矩阵中，确保每次 PR 都会运行。
- .github/workflows/pr-test.yml.j2（模块 CI；类别 infra；类型 infrastructure）：模板文件同步添加测试条目，并修复一个重复的 r3 测试条目。

关键符号：prepare, execute, set_weight_version, update_weights, _finalize_sync

## 关键源码片段

### tests/test_delta_weight_update.py

新增的端到端测试文件，验证 delta 权重更新的完整流程，是 PR 的核心。

```
"""E2E smoke test for disk-backed delta weight updates.

Runs a tiny Qwen3.5-0.8B job so the first weight update seeds the delta
snapshot and the post-train update publishes sparse delta files through
``update_weights_from_disk(load_format="delta", files=...)``.
"""

import os
import tempfile
from pathlib import Path

import slime.utils.external_utils.command_utils as U

MODEL_NAME = "Qwen3.5-0.8B"
MODEL_TYPE = "qwen3.5-0.8B"
NUM_GPUS = 4
TORCH_DIST_CKPT = f"/dev/shm/{MODEL_NAME}_torch_dist"


def prepare():
    # 准备模型和数据集
    U.exec_command("mkdir -p /root/models /root/datasets")
    U.exec_command(f"hf download Qwen/{MODEL_NAME} --local-dir /root/models/{MODEL_
    NAME}")
    U.hf_download_dataset("zhuzilin/gsm8k")
    U.convert_checkpoint(
        model_name=MODEL_NAME,
        megatron_model_type=MODEL_TYPE,
        num_gpus_per_node=NUM_GPUS,
        dir_dst="/dev/shm",
    )


def execute():
    with tempfile.TemporaryDirectory(prefix="slime_delta_weight_update_") as delta_dir:
        # 构造参数：包括 rollout、优化器、GRPO、sglang 和 delta 相关参数
        ckpt_args = f"--hf-checkpoint /root/models/{MODEL_NAME}/ " f"--ref-load {TORCH_DIST_
```

```
        CKPT} "
        rollout_args = (
            "--prompt-data /root/datasets/gsm8k/train.parquet "
            # ... 其他 rollout 参数
        )
        # ... 其他参数组
        delta_args = (
            "--update-weight-mode delta "
            "--update-weight-transport disk "
            "--update-weight-encoding deltas "
            f"--update-weight-delta-dir {delta_dir} "
            "--update-weight-delta-keep-files "
        )
        # ... 组合并执行训练
        U.execute_train(
            train_args=train_args,
            num_gpus_per_node=NUM_GPUS,
            megatron_model_type=MODEL_TYPE,
        )
        # 验证 delta 文件生成
        delta_files = list(Path(delta_dir).glob("weight_v*/*.safetensors"))
        assert delta_files, f"No disk delta safetensors were written under {delta_dir}"
```

## slime/backends/megatron_utils/update_weight/update_weight_from_distributed_delta.py

Delta 权重更新器的核心逻辑修改，添加 _published_any 标志并在零 diff 时同步版本。

```
# 在 __init__ 中新增
self._published_any: bool = False

# 在 update_weights 的种子调用部分
if not self._snapshot_seeded:
    self._seed_snapshot()
    self._snapshot_seeded = True
    # Pin the engine's recorded version to ours (0) on the seed call
    if dist.get_rank() == 0 and self.transport == "disk" and self.rollout_engines:
        weight_version = str(self.weight_version)
        ray.get([engine.set_weight_version.remote(weight_version) for engine in self.rollout_
        engines])
    return

# 在 update_weights 后续重置 _published_any
self._published_any = False

# 在 _publish_batch 中标记
self._published_any = True

# 在 _finalize_sync 中处理零 diff
if not self._published_any:
```

```
# No delta files needed publishing this sync (e.g. all-zero diff).
# Engines never saw the new version via update_weights_from_disk, so
# bump it explicitly to keep their recorded version in sync with ours.
weight_version = str(self.weight_version)
ray.get([engine.set_weight_version.remote(weight_version) for engine in self.rollout_engines])
```

## 评论区精华

该 PR 无 review 评论。

- 暂无高价值评论线程

## 风险与影响

- 风险：风险较低：测试依赖外部模型下载和 GPU 资源，可能因网络或资源不足失败；CI 执行时间略有增加。set_weight_version 的 HTTP 调用可能因引擎异常而失败，但已有错误处理。
- 影响：对用户无直接影响；对系统而言，CI 覆盖了 delta 权重更新的关键路径，未来修改 delta 相关逻辑时能及早发现问题；对团队而言，新增的 e2e 测试可作为类似功能的参考。
- 风险标记：外部依赖（模型下载），CI 执行时间增加

## 关联脉络

- 暂无明显关联 PR