# PR #2543 完整报告

radixark/miles

Allow --stream-optimizer-state-to-disk without trainer offload

合并时间：2026-08-15 05:29

原文链接：http://prhub.com.cn/radixark/miles/pull/2543

## 执行摘要

- 一句话：解除流式优化器状态与训练器卸载的强耦合
- 推荐动作：值得精读，尤其关注参数校验层面的设计权衡，以及“放宽断言但未补测试”的后续跟进。该修复展示了在大型集群真实负载约束下，参数级耦合假设如何被实际拓扑推翻的过程。

## 功能与动机

运行 examples/experimental/openenv/glm52_tbench2 时发现该示例完全无法启动，这是阻断该示例的三个问题中的第一个。--offload-train 只在 colocate 下默认开启，分离式运行时训练器根本没有可卸载的内容（engine 在独立节点），为通过断言而开启 --offload-train 会在每个阶段边界做无意义的整模型 NVMe 往返。而 DP1 下优化器状态不切分时高达 270 GB/rank（GPU 仅 276 GB），流式化是让它装下的唯一途径。

## 实现拆解

实现拆解分为三步：

1. 目录默认值提升：在 miles/utils/arguments.py 的 miles_validate_args 中，把 offload_train_disk_dir 的默认值设置从 offload_train_target == "disk" 分支中提升出来，改为在 offload_train_target == "disk" or stream_optimizer_state_to_disk 时都设置。这是因为 miles_plugins/optimizers/nvme_stream.py:453 会无条件把 NVMe store 挂到这个路径，此前流式单独启用时该值为 None，会在 step 内触发 TypeError。
2. 放宽断言：将 assert args.offload_train_target == "disk" 改为 assert args.offload_train_target == "disk" or not args.offload_train，并更新错误信息说明“分离式运行完全不卸载训练器，target 在那里未被使用”。其余与 indep_dp、LoRA、distributed optimizer 等相关的断言保持不变。
3. 文档同步：docs/advanced/disk-offload.md 把“流式必须与 offload-train 成对”改为“流式单独使用是分离式场景；在 --offload-train 下两者才成对”；docs/user-guide/training-backend.md 把“--stream-optimizer-state-to-disk 构建在 disk offload 之上”改为“两者都断言 Megatron 后端”。

未新增测试文件。PR 说明原有 colocate 测试 tests/e2e/megatron/test_qwen3_4B_offload_disk_stream.py 两个参数都设置，不受影响；验证方式是在 16x GB300 上跑通参考配置，确认 store 正常参与训练步骤。

关键文件：

- miles/utils/arguments.py（模块 参数校验；类别 source；类型 core-logic；符号 miles_validate_args）：核心改动：放宽流式优化器状态与 train offload 的耦合断言，并提升磁盘目录默认值设置，是本次变更的主路径。
- docs/advanced/disk-offload.md（模块 文档；类别 docs；类型 documentation）：文档同步：改写流式单独使用的说明，从“两者成对部署”改为“分离式场景下流式可单独使用，offload-train 下两者成对”，是理解新断言行为的关键文档。
- docs/user-guide/training-backend.md（模块 文档；类别 docs；类型 documentation）：修正训练后端文档中关于流式优化器状态与 Megatron 后端约束的描述，保持文档与实现一致。

关键符号：miles_validate_args

## 关键源码片段

### miles/utils/arguments.py

核心改动：放宽流式优化器状态与 train offload 的耦合断言，并提升磁盘目录默认值设置，是本次变更的主路径。

```
# miles/utils/arguments.py 中的关键改动
# 1) 目录默认值提升：流式单独使用时也能拿到非 None 的磁盘目录，
# 否则 nvme_stream.py 的 NVMe store 挂到 None 路径会在 step 内触发 TypeError。
if (args.offload_train_target == "disk" or args.stream_optimizer_state_to_disk) and (
    args.offload_train_disk_dir is None
):
    uid = os.getuid() if hasattr(os, "getuid") else 0
    args.offload_train_disk_dir = os.path.join(
        os.environ.get("SCRATCH", "/scratch"), f"miles_train_offload_{uid}"
    )

# 2) 断言放宽：分离式运行不卸载训练器，offload_train_target 未被读取，
# 因此 stream-optimizer-state-to-disk 可以单独启用；
# 但若开启 --offload-train，仍要求 target=disk，避免整模型驻留与流式共存。
if args.stream_optimizer_state_to_disk:
    assert args.offload_train_target == "disk" or not args.offload_train, (
        "--stream-optimizer-state-to-disk with --offload-train requires "
        "--offload-train-target=disk: a run that cannot hold the optimizer state on GPU for "
        "the duration of the step will not hold a pinned host copy of the whole actor either. "
        "Disaggregated runs do not offload the trainer at all, and the target is unused there."
    )
    assert not args.indep_dp, (
        "--stream-optimizer-state-to-disk does not support --indep-dp: each cell has its own "
        "process group, so torch.distributed.get_rank() restarts at 0 per cell and two cells "
        "on one node would share a store directory"
    )
    assert args.use_distributed_optimizer, "--stream-optimizer-state-to-disk requires the
    distributed optimizer"
```

```
    assert (
        args.optimizer == "adam"
    ), f"--stream-optimizer-state-to-disk requires --optimizer adam, got {args.optimizer}"
    assert not (args.multi_lora or is_lora_enabled(args)), (
        "--stream-optimizer-state-to-disk does not support LoRA: the LoRA checkpoint path "
        "persists optimizer.state_dict(), which the store leaves empty, and restores the "
        "adapter into the model params without refreshing the streamed main params"
    )
    assert not args.optimizer_cpu_offload, "--stream-optimizer-state-to-disk excludes --optimizer-cpu-offload"
    assert (
        not args.offload_optimizer_states
    ), "--stream-optimizer-state-to-disk excludes --offload-optimizer-states"
    assert (
        not args.use_precision_aware_optimizer
    ), "--stream-optimizer-state-to-disk requires mcore to hold the fp32 main params"
    assert not args.reset_optimizer_states, (
        "--reset-optimizer-states walks the master optimizer's state, which the NVMe store "
        "leaves empty, so the reset would silently do nothing"
    )
    assert not args.save_local_weight_checksum, (
        "--save-local-weight-checksum reads param.main_param, whose storage the NVMe store "
        "resizes to 0 between steps"
    )
```

## 评论区精华

仅一条审核评论：Zhichenzzz 批准（APPROVED）并要求同步更新相关文档。提出者照做，PR 以 1 个 commit 合并。

- 文档是否需要同步更新 (documentation): PR 已同步更新 docs/advanced/disk-offload.md 和 docs/user-guide/training-backend.md，并成功合并。

## 风险与影响

- 风险：
  1. 缺少新测试：本 PR 放宽了断言但没有新增针对“流式单独使用”的测试，现有 colocate 测试 test_qwen3_4B_offload_disk_stream.py 两个参数都设置，无法覆盖新放开的路径。
  2. 非 Megatron 后端未验证：原断言链中 offload_train_target == "disk" 会隐式要求 Megatron 后端，放宽到 not args.offload_train 后，流式单独使用在非 Megatron 后端上不再被参数校验拦截，但 NVMe store 是否支持未知，可能到运行时才暴露。
  3. 文档一致性：文档同步改动较小，需要确认与其它章节的表述持续一致。 - 影响：影响用户配置自由度：分离式训练（disaggregated）可直接使用流式优化器状态，避免为满足断言而开启 --offload-train 导致的全模型 NVMe 往返。对 colocate 用户无行为变化；对文档读者理解更准确。这是让 glm52_tbench2 示例在 16x GB300 上可运行的第一个阻塞问题修复。 - 风险标记：原约束被放宽但缺少新测试，非 megatron 后端未验证，参数校验放宽需运行时验证

# 关联脉络

- PR #2484 docs: add disaggregated RL rollout guide: 与本 PR 同为分离式训练（disaggregated）拓扑支持相关，本 PR 是让分离式示例（glm52_tbench2）可运行的首个阻塞问题的修复。