

PR #2653 完整报告

radixark/miles

Stream optimizer main initialization to NVMe

合并时间: 2026-09-01 02:57

原文链接: <http://prhub.com.cn/radixark/miles/pull/2653>

执行摘要

- 一句话: main 参数初始化逐桶直写 NVMe, 训练启动显存峰值下降
- 推荐动作:

功能与动机

PR body 开篇即点明目标: Extend `--stream-optimizer-state-to-disk` to distributed-optimizer construction。在此前实现中, NVMe streaming 只覆盖优化器 step 阶段的读写, 而 DistributedOptimizer 构造时会一次性在 HBM 中累积全部 FP32 main 参数; 配对 Megatron#86 的 body 明确其边界: the largest individual FP32 shard must still fit briefly; this removes cumulative main-param residency, not the largest single allocation。本 PR 正是把「不常驻累计 main 状态」的承诺延伸到 Miles 训练链路, 使 744B 级别模型的启动显存峰值从全量 FP32 main 降到单 bucket 量级。

实现拆解

变更入口是 `miles/backends/megatron_utils/model.py` 的 `setup_model_and_optimizer` 与 `miles_plugins/optimizers/nvme_stream.py` 的 `setup_optimizer_state_streaming`, 实现按以下步骤拆解:

1. 上层配置入口 (data-contract): 在 `setup_model_and_optimizer` 构造 `OptimizerConfig` 时, 若 `args.stream_optimizer_state_to_disk` 且优化器非 Muon, 则设置 `config.defer_main_param_initialization = True`。这使 Megatron 以 deferred 模式构造 DistributedOptimizer: 创建同形状 FP32 main 张量句柄后立即释放 backing storage, 只保留逻辑形状、dtype、device 与对象身份。Muon 被显式排除, 因为上游已给 Muon 独立的 file-backed chunked-state 路径, 两条路径不应叠加。
2. Store 构造调整: `NVMeOptimizerStateStore.__init__` 删除了构造末尾对 main 段的 flush 循环。原因是 deferred 模式下 main 张量此时没有底层存储, 原来的 flush 只会写出空数据; main 段的落盘职责被转移到新增的 `initialize_main_from_model_params`。
3. 核心新增方法 `initialize_main_from_model_params()`: 遍历 buckets, 对每个 bucket 先 `materialize_main()` 物化其 FP32 main 存储, 再逐 entry 通过 `_get_model_param_range_map(entry.model_param)["param"]` 取得 BF16 源分片范围并 `copy_` 到 main, 随后 `bucket.flush(segments=("main",))` 直写最终 bucket 文件, 并依次执行 `os.fdatasync` 与 `os.posix_fadvise(POSIX_FADV_DONTNEED)` 驱逐页缓存。每个

bucket处理完毕，其CUDA存储回到零字节（测试断言`untyped_storage().nbytes()==0`）。该设计复用既有 200M 元素 soft bucket 上限与最终 store 布局，全程没有临时 full-main 文件。

4. 装配点调度：`setup_optimizer_state_streaming` 对每个非 stub 分布式优化器先断言 `dist_opt.config.defer_main_param_initialization`（fail-fast 契约校验，防止未开启 deferred 模式时初始化读到空存储），构造 store 后立即调用初始化并记录 wrote GB 日志，最后 `_bind`。
5. reload 路径修正（第二 commit）：`_bind.reload_model_params` 增加分支——`state_dict is None` 时直接调用 `initialize_main_from_model_params()`（无 checkpoint 可加载时逐 bucket 重建 main，保持内存受控）；有 `state_dict` 时仍走原有 `refresh_main_from_model_params` 全量物化 + 拷贝 + flush 路径。这保持了 post-load 刷新语义与初始化路径的行为一致。
6. 测试与文档配套：新增 `tests/fast-gpu/test_nvme_optimizer_main_init.py`（注册 stage-b-2-gpu-h200、30 秒、miles-plugin 标签），用最小 bucket 模拟 deferred 场景，验证初始化返回字节数、文件字节与 BF16→FP32 零容差一致、CUDA storage 归零；`tests/e2e/megatron/test_qwen3_4B_offload_disk_stream.py` 的 `_assert_streamed` 增加对 NVMe optimizer main-param initialization: 日志的断言，要求全部 4 个 rank 的 worker 日志都出现该记录；`docs/advanced/disk-offload.md` 补充设计说明（无临时状态文件、逐 bucket 直写、每段同步并驱逐页缓存）。

关键文件：

- `miles_plugins/optimizers/nvme_stream.py`（模块 `miles_plugins/optimizers`；类别 `source`；类型 `core-logic`；符号 `initialize_main_from_model_params`）：源码主路径；涉及符号 `initialize_main_from_model_params`；包含 控制流调整、配置键调整、符号定义调整；+33/-8
- `tests/fast-gpu/test_nvme_optimizer_main_init.py`（模块 `nvme/optimizer/main/init`；类别 `test`；类型 `test-coverage`；符号 `test_bucketwise_main_initialization_preserves_bytes_and_releases_cuda_storage`）：测试配套；涉及符号 `test_bucketwise_main_initialization_preserves_bytes_and_releases_cuda_storage`；包含 测试覆盖调整、导入关系调整、配置键调整；+44/-0
- `miles/backends/megatron_utils/model.py`（模块 `miles/backends`；类别 `source`；类型 `data-contract`）：源码主路径；包含 控制流调整、配置键调整；+2/-0
- `tests/e2e/megatron/test_qwen3_4B_offload_disk_stream.py`（模块 `qwen3/4B/offload/disk/stream`；类别 `test`；类型 `test-coverage`）：测试配套；包含 测试覆盖调整、控制流调整、配置键调整；+9/-1
- `docs/advanced/disk-offload.md`（模块 `docs/advanced`；类别 `docs`；类型 `documentation`）：文档说明；包含 控制流调整、异常路径调整；+7/-0

关键符号：`initialize_main_from_model_params`, `test_bucketwise_main_initialization_preserves_bytes_and_releases_cuda_storage`

评论区精华

仓库内没有实质性的技术 review 讨论：该 PR 来自 fork，claude[bot] 自动 review 被禁用（评论提示维护者可发 `@claude review` 触发一次性审核），合并者 yueming-yuan 直接给出 APPROVED，且无任何 inline review comment。真正的设计权衡记录在 PR body 中：逐 bucket 物化直写最终文件（无临时 full-main 文件）、初始化复用既有 200M 元素 soft bucket 上限、每 bucket 处理完毕即 fdatsync 并 fadvise 驱逐页缓存、以及 Muon 不进入 deferred-main 路径（`optimizer=dist_muon is_muon=True defer_main_param_initialization=False` 的直查结果）。第二个 commit 由合并者提交，修复了 `reload_model_params(state_dict=None)` 的语义：从

- 暂无高价值评论线程

风险与影响

- 风险：
- 影响：
 - 风险标记：暂无

关联脉络

- 暂无明显关联 PR