PR #2089 完整报告

THUDM/slime

[2/n] Disaggregated rollout: disk-level delta weight sync

合并时间：2026-07-02 17:20

原文链接：http://prhub.com.cn/THUDM/slime/pull/2089

执行摘要

- 一句话：磁盘级 delta 权重同步，替代 NCCL 传输
- 推荐动作：值得精读，特别是 disk_delta.py 中的工具函数设计以及 UpdateWeightFromDiskDelta 如何通过继承复用 UpdateWeightFromDistributed。配置参数的变化也需关注以便兼容旧脚本。

功能与动机

PR 描述：For non-colocated training/inference across clusters, ship only the changed bytes between weight syncs instead of a full checkpoint; replaces the NCCL delta transport from #1806. 使 rollout 侧与 trainer 解耦，可自由使用任何低精度或并行方案。

实现拆解

1. 基础工具层：新增 slime/utils/disk_delta.py，提供字节级 delta 编解码（overwrite_encode）、多种校验和（xxh3-128/blake3/adler32）、本地检查点初始化（init_local_checkpoint）和 delta 应用（apply_deltas），使用线程池并行处理张量。
2. 发布端核心：新增 slime/backends/megatron_utils/update_weight/update_weight_from_disk_delta.py，实现 UpdateWeightFromDiskDelta 类。首次调用捕获基线快照，后续调用对每个 HF 张量进行字节级 diff，压缩后写入标准 HF 检查点目录。支持自定义 pre-push 钩子。
3. Rollout 引擎端：修改 slime/backends/sglang_utils/sglang_engine.py，新增 sync_local_checkpoint() 方法，调用 init_local_checkpoint 确保本地副本存在，然后调用 apply_deltas 应用发布的最新 delta。启动时在后台线程预拷贝基座检查点以重叠启动时间。
4. 配置与验证：修改 slime/utils/arguments.py，将 --update-weight-encoding 替换为 --update-weight-delta-encoding（choices: xor/overwrite），新增 --update-weight-delta-checksum，删除 --update-weight-delta-dir，调整帮助文本明确 delta 只支持 disk transport。验证逻辑内联化。
5. 清理与依赖：删除旧文件 update_weight_from_distributed_delta.py（864 行），删除旧的示例脚本 examples/delta_weight_sync/run-glm4.7-355B-A32B-delta.sh，更新 sglang patch 去除 delta 接收端代码，调整 actor.py 和 rollout.py 的导入。文档同步更新。

关键文件：

- slime/backends/megatron_utils/update_weight/update_weight_from_distributed_delta.py（模块 增量传输（旧）；类别 source；类型 deletion；符号 ParamDiff, EncodedChunk, empty, _checksum）：被完全删除的旧 delta 同步实现，包含 NCCL 和磁盘双传输方式，是本次重构的直接替代目标。
- slime/backends/megatron_utils/update_weight/update_weight_from_disk_delta.py（模块 权重同步；类别 source；类型 dependency-wiring；符号 UpdateWeightFromDiskDelta, init, connect_rollout_engines, disconnect_rollout_engines）：新增的磁盘 delta 同步实现，核心发布端类，继承 UpdateWeightFromDistributed，只支持 disk transport。
- slime/utils/disk_delta.py（模块 增量工具；类别 source；类型 dependency-wiring；符号 overwrite_encode, _Adler32, checksum, init_local_checkpoint）：新增的磁盘 delta 工具模块，提供 overwrite_encode、checksum、init_local_checkpoint 等核心函数。
- slime/backends/sglang_utils/sglang_engine.py（模块 SGLang 引擎；类别 source；类型 core-logic；符号 sync_local_checkpoint）：修改 SGLang 引擎，新增 sync_local_checkpoint 方法，去掉旧 delta 代码。

关键符号：UpdateWeightFromDiskDelta.init, UpdateWeightFromDiskDelta.update_weights, UpdateWeightFromDiskDelta._capture_baseline, UpdateWeightFromDiskDelta._publish, UpdateWeightFromDiskDelta._write_delta_files, overwrite_encode, checksum, init_local_checkpoint, apply_deltas, sync_local_checkpoint

关键源码片段

slime/backends/megatron_utils/update_weight/update_weight_from_disk_delta.py

新增的磁盘 delta 同步实现，核心发布端类，继承 UpdateWeightFromDistributed，只支持 disk transport。

```
from slime.utils.disk_delta import NUM_WORKERS, checksum, make_tensor_reader, overwrite_
encode

class UpdateWeightFromDiskDelta(UpdateWeightFromDistributed):
    """
    Delta weight sync over a shared filesystem.
    PP-src ranks diff each gathered HF tensor against a CPU snapshot of the previous sync
    and publish the changes as a canonical HF checkpoint dir;
    every rollout host applies the delta into its local checkpoint and reloads via the ordinary
    update_weights_from_disk path, so sglang needs no delta support.
    """

    def __init__(self, args, model, weights_getter, *, model_name, quantization_config):
        super().__init__(args, model, weights_getter, model_name=model_name, quantization_
        config=quantization_config)
        self.delta_dir = args.update_weight_disk_dir
        os.makedirs(self.delta_dir, exist_ok=True)
        self.delta_encoding = args.update_weight_delta_encoding # "xor" or "overwrite"
```

```
        self.checksum_algorithm = args.update_weight_delta_checksum # e.g. "xxh3-128"
        self._snapshot: dict[str, np.ndarray] = {}
        self._baseline_captured = False
        self._commit_hook: Callable | None = None
        if args.custom_delta_pre_push_path:
            from slime.utils.misc import load_function
            self._commit_hook = load_function(args.custom_delta_pre_push_path)

    @torch.no_grad()
    def update_weights(self) -> None:
        if not self._baseline_captured:
            self._capture_baseline()
            self._baseline_captured = True
            return
        self.weight_version += 1
        if dist.get_rank() == 0:
            ray.get([engine.pause_generation.remote() for engine in self.rollout_engines])
            ray.get([engine.flush_cache.remote() for engine in self.rollout_engines])
        dist.barrier(group=get_gloo_group())
        self._publish()
        self._reload_engines()
        self._record_metrics()

    def _capture_baseline(self) -> None:
        if dist.get_rank() == 0:
            shutil.rmtree(self.delta_dir, ignore_errors=True)
            os.makedirs(self.delta_dir, exist_ok=True)
            if self._commit_hook is not None:
                self._commit_hook(self.args, self.delta_dir, list(self.rollout_engines))
        dist.barrier(group=get_gloo_group())
        read_hf = make_tensor_reader(self.args.hf_checkpoint)
        for name, tensor in self._iter_hf_tensors():
            try:
                self._snapshot[name] = read_hf(name)
            except KeyError:
                self._snapshot[name] = tensor.cpu().numpy()
```

slime/backends/sglang_utils/sglang_engine.py

修改 SGLang 引擎，新增 sync_local_checkpoint 方法，去掉旧 delta 代码。

```
  def sync_local_checkpoint(self, target_version: int):
      """Apply the published deltas into this host's local checkpoint up to target_version;
      the engine reloads it afterwards. Assumes this actor shares the checkpoint filesystem
      with the sglang it drives (true for slime-launched engines)."""
      from slime.utils.disk_delta import apply_deltas, init_local_checkpoint

      # idempotent: ensure local checkpoint exists
      init_local_checkpoint(self.args.update_weight_local_checkpoint_dir, self.args.hf_checkpoint)
      # non-POSIX filesystems may need a pre-read hook for consistency
```

```
if self.args.custom_delta_pre_read_path:
    from slime.utils.misc import load_function
    load_function(self.args.custom_delta_pre_read_path)(
        self.args.update_weight_disk_dir, target_version)
apply_deltas(
    self.args.update_weight_local_checkpoint_dir,
    self.args.update_weight_disk_dir,
    target_version,
)
```

评论区精华

Review 评论只有 1 条，无实质性内容；设计决定已在 PR body 和 commit message 中明确记录。主要讨论包括：将 delta 传输从 NCCL 迁移到磁盘，使得 rollout 引擎无需了解 delta 细节；配置参数简化，移除旧 alias。

- 整体设计评估 (design): 设计被接受，无需修改。

风险与影响

- 风险:
  1. 文件系统一致性：非 POSIX 文件系统可能出现跨主机写后读不一致，通过 custom_delta_pre_read_path 钩子解决。
  2. 锁竞争：使用 fcntl.flock 进行主机级互斥，高并发时可能成为瓶颈。
  3. 配置兼容性：旧配置中使用 --update-weight-encoding 和 --update-weight-delta-dir 的脚本需要迁移到新参数。
  4. 性能风险：磁盘 I/O 和压缩 / 解压缩可能增加同步延迟，但利用线程池和 zstd 轻量压缩缓解。 - 影响：用户 / 系统：需要更新启动参数；delta 模式仅支持 disk transport，不再支持 NCCL delta；跨集群训练 / 推理支持更灵活部署，但依赖共享文件系统。团队：代码结构更清晰，decoupled design 便于后续维护；新增 disk_delta.py 工具模块可供其他组件复用。
- 风险标记：核心路径变更，配置兼容性，跨文件系统依赖，锁竞争风险

关联脉络

- PR #1806 PR #1806（前序：NCCL delta weight sync): 本 PR 替代了 #1806 中的 NCCL delta 传输方案，采用纯磁盘方式。
- PR #2181 PR #2181（后续：增量权重同步优化）: 作为系列 PR 的一部分，本 PR 实现磁盘 delta 基础，后续 PR 将在此基础上优化。