PR #1731 完整报告

THUDM/slime

Fix CUDA IPC cache leaks during weight updates

合并时间：2026-03-17 10:37

原文链接：http://prhub.com.cn/THUDM/slime/pull/1731

执行摘要

修复了 SLIME 中权重更新过程中的 CUDA IPC 缓存泄漏，通过添加内存释放和 CUDA IPC 收集调用，避免 GPU 内存累积，提升系统长期运行的稳定性。

功能与动机

本 PR 旨在解决一个特定场景下的 GPU 内存泄漏问题。根据 PR body，根因在于使用 ForkingPickler 时，GPU 张量通过 storage._share_cuda_() 创建了 CUDA IPC 缓存条目，这些条目在消费者未关闭句柄前不会释放，导致内存累积。修复动机是确保内存能被及时回收，防止资源浪费。

实现拆解

修改集中在 update_weight_from_tensor.py 文件的 update_weights 函数中：

- 在循环中添加 del long_lived_tensors, hf_named_tensors 来明确释放 GPU 张量。
- 调用 torch.cuda.ipc_collect() 两次：一次在每次 chunk 处理后，释放已完成 chunk 的 IPC 缓存条目；一次在所有 chunk 完成后（barrier 后），释放最后一个 chunk 的 IPC 条目。 代码块示例：

```
for hf_named_tensors in self._hf_weight_iterator.get_hf_weight_chunks(megatron_local_weights):
    refs, long_lived_tensors = self._send_hf_params(hf_named_tensors)
    ray.get(refs)
    del long_lived_tensors, hf_named_tensors
    torch.cuda.ipc_collect()
    dist.barrier(group=get_gloo_group())
    torch.cuda.ipc_collect()
```

评论区精华

本 PR 未经过 review 讨论，直接由作者提交并合并，表明变更可能较为紧急或已在内部验证，无需额外争议。

风险与影响

风险：引入的 torch.cuda.ipc_collect() 调用可能带来轻微性能开销，尤其是在高频率权重更新场景中。此外，依赖 PyTorch CUDA IPC 机制，需确保与现有环境的兼容性。 影响：对用户透明地减少内存泄漏，改善系统稳定性；对开发团队，提供了一个针对 CUDA 内存管理的修复案例，有助于规避类似问题。

关联脉络

与近期 PR 1765（涉及 megatron_utils update_weight 模块的优化）相关，表明团队正持续改进权重更新相关的性能和稳定性。建议结合阅读以理解该模块的整体演进方向。