PR #22998 完整报告

sgl-project/sglang

Skip torch.cuda.empty_cache() in weight update flush path

合并时间：2026-04-25 12:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22998

执行摘要

- 一句话：权重更新路径跳过 torch.cuda.empty_cache()
- 推荐动作：建议精读。该 PR 展示了如何通过细粒度控制同步 CUDA 操作来优化性能，值得关注的设计决策是：将 torch.cuda.empty_cache() 从 flush 路径中分离，而不是全局移除，保持了灵活性。

功能与动机

torch.cuda.empty_cache() 会同步所有 CUDA 流，在并发推理负载下调用时会导致显著延迟。PR 描述明确指出"skipping it in the weight update path avoids this without affecting KV cache pool clearing"——权重更新时只需要刷新 KV 缓存池的数据结构，而无需额外回收 GPU 缓存，因此可以安全跳过。

实现拆解

1. 修改 flush_cache() 签名（scheduler.py） - 新增参数 empty_cache: bool = True，控制是否执行 torch.cuda.empty_cache()。 - 将原本注释掉的 TODO 替换为条件执行。
2. 在权重更新 mixin 中透传参数（scheduler_update_weights_mixin.py） - 在 update_weights_from_disk()、update_weights_from_distributed()、update_weights_from_tensor()、update_weights_from_ipc() 四个方法中，将 self.flush_cache() 调用改为 self.flush_cache(empty_cache=recv_req.torch_empty_cache)。 - 这样，调用方可以通过请求的 torch_empty_cache 字段决定是否需要触发 CUDA 缓存回收。
3. 新增请求字段（io_struct.py） - 为 UpdateWeightsFromDistributedReqInput、UpdateWeightsFromTensorReqInput、UpdateWeightsFromIPCReqInput 三个 dataclass 添加 torch_empty_cache: bool = False 字段，默认不执行 empty_cache。 - 注意到 UpdateWeightFromDiskReqInput 之前已经存在该字段，本次保持一致性（其余请求类型也统一了协议）。
4. 文档注释补充：为新增字段添加注释说明"Whether to call torch.cuda.empty_cache() during flush"。

无测试、配置或部署配套改动。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 flush_cache）：核心调度器，修改了 flush_cache() 方法，增加参数控制是否执行 empty_cache
- python/sglang/srt/managers/scheduler_update_weights_mixin.py（模块 调度器；类别 source；类型 core-logic）：权重更新 mixin，在四个更新路径方法中将 flush_cache 调用改为传递 empty_cache 参数
- python/sglang/srt/managers/io_struct.py（模块 调度器；类别 source；类型 core-logic）：定义请求输入结构体，为三种权重更新请求添加 torch_empty_cache 字段，默认 False

关键符号：Scheduler.flush_cache, SchedulerUpdateWeightsMixin.update_weights_from_disk, SchedulerUpdateWeightsMixin.update_weights_from_distributed, SchedulerUpdateWeightsMixin.update_weights_from_tensor, SchedulerUpdateWeightsMixin.update_weights_from_ipc

关键源码片段

python/sglang/srt/managers/scheduler.py

核心调度器，修改了 flush_cache() 方法，增加参数控制是否执行 empty_cache

```
def flush_cache(self, empty_cache: bool = True):
    """Flush the memory pool and cache."""
    if self.is_fully_idle():
        self.cur_batch = None
        self.last_batch = None
        self.tree_cache.reset()
        self.req_to_token_pool.clear()
        self.token_to_kv_pool_allocator.clear()
        self.grammar_manager.clear()
        self.reset_metrics()

        if self.draft_worker:
            self.draft_worker.clear_cache_pool()

        # 根据参数决定是否调用 torch.cuda.empty_cache()
        # 该调用会同步所有 CUDA 流，在并发推理负载下会造成显著延迟
        if empty_cache:
            torch.cuda.empty_cache()
        logger.info("Cache flushed successfully!")
        success = True
    else:
        logging.warning(
            f"Cache not flushed because there are pending requests. "
            f"#queue-req: {len(self.waiting_queue)}, "
            f"#running-req: {len(self.running_batch.reqs)}"
        )
        success = False
    return success
```

评论区精华

该 PR 无 review 评论，也未在 Issue 中引发讨论。PR 由作者直接合并，说明改动在该团队内部已达成共识。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 回归风险低：flush_cache() 默认参数 empty_cache=True，所有非权重更新的调用路径行为不变。
  - 性能风险：跳过 empty_cache 可能使 GPU 显存碎片化程度更高，但对权重更新场景而言，后续通常会加载新权重并重新分配显存，影响有限。
  - 兼容性风险：新增的 torch_empty_cache 字段默认值为 False，与旧版权重更新请求（未携带该字段）的默认行为（旧版会执行 empty_cache）不同；但旧请求一旦升级到新版本 scheduler，就会自动跳过 empty_cache，这可能带来微妙的显存行为变化。
  - 未添加测试覆盖。
  - 影响：用户 / 系统影响：对于使用在线权重更新的场景（如在训练和推理之间切换），此改动可减少激活 torch.cuda.empty_cache() 所带来的同步延迟，提高并发推理的稳定性和吞吐量。由于跳过的是 GPU 级缓存回收，仅影响显存碎片管理，不影响正确性。

影响范围：权重更新相关的四个路径全部受影响。如果这些路径被频繁调用（例如在强化学习训练中），延迟优化效果明显。

团队影响：无。改动仅涉及 3 个文件，逻辑简单。

- 风险标记：默认行为变更，缺少测试覆盖

关联脉络

- PR #21985 perf: eliminate attention DtoD copy by passing pre-allocated output to FA: 同为性能优化 PR，涉及 CUDA 同步操作的消除
- PR #22218 [Experimental] Breakable Piecewise Cuda Graph: 涉及 CUDA 图执行流程优化，与此 PR 同属 CUDA 性能调优方向