PR #19890 完整报告

sgl-project/sglang

[Disagg] GPU staging buffer with dynamic ring allocator for heterogeneous TP KV transfer

合并时间：2026-04-02 05:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/19890

执行摘要

本 PR 为 sglang 的 disaggregated serving 引入 GPU staging buffer 机制，针对异构 TP 布局下的 KV 传输优化。通过将分散的 head slices 收集到连续内存进行批量 RDMA 传输，减少请求数量约 1000 倍，提升高并发下吞吐量 2-3 倍。实现包括动态环形分配器和专用 kernels，设计隔离良好，但仅支持 mooncake 后端。

功能与动机

在异构 TP 场景（如 TP4 prefill + DEP4 decode），原有每令牌 RDMA 方法生成约 30,000 小请求，限制吞吐量。如 PR body 所述：'severely limiting transfer throughput at high concurrency'。新机制旨在通过批量传输解决此瓶颈，基准测试显示在 128 并发下吞吐量从 2,466 tok/s 提升至 4,914 tok/s。

实现拆解

- staging_buffer.py: 新增 Triton kernels _fused_gather_to_staging_kernel 和 _fused_scatter_from_staging_kernel，执行 GPU gather/scatter，内核延迟低至微秒级。
- staging_handler.py: 定义 DecodeStagingHandler 和 PrefillStagingContext，管理 staging 生命周期，隔离于 decode.py 和 conn.py。
- mooncake/conn.py: 集成 send_kvcache_staged 方法，处理 staging 传输路径，更新 bootstrap 协议支持动态偏移。
- decode.py: 添加 _poll_with_staging 函数（后移至 utils.py），处理 staging 感知的轮询，异步 scatter 在专用 CUDA 流执行。
- prefill.py: 添加 maybe_prefetch_staging_for_batch 预取逻辑，优化 decode 侧内存分配。
- environ.py: 新增环境变量 SGLANG_DISAGG_STAGING_BUFFER 控制功能开关，默认禁用。

评论区精华

- 设计优化: ShangmingCai 建议：'Can we add an attr in PrefillServerInfo, let us call it require_staging'，最终将逻辑移至 CommonKVReceiver，提升多实例支持。YAMY1234 回复：'Fully agree. I've updated the code to support cases where prefill uses different TP sizes'。
- 代码可读性: 讨论中强调使用 enable_staging 直接检查，避免 getattr hack，如 ShangmingCai 评论：'I think we should let prefill/decode both have this attr'。

YAMY1234 采纳并调整代码。

- 性能权衡: 关于 queue 大小，ShangmingCai 询问：'Have you checked whether fixing the queue size to 1 affects the performance?' YAMY1234 解释：'With a single queue there's only one dispatcher... can become a single-threaded bottleneck'，决定保持多 queue 设计。

风险与影响

- 风险:
  1. 兼容性：staging 仅支持 mooncake 传输后端，其他后端会报错（server_args.py 验证）。
  2. 内存：动态环形分配器可能在高并发下 OOM，但通过分块传输和默认 4GB 池缓解。
  3. 正确性：长上下文场景 buffer 不足，PR 通过 per-chunk 传输解决，已验证 128k ISL 场景。
  4. 性能：多线程调度增加复杂性，但基准测试显示显著提升。
- 影响:
  - 用户：异构 TP 传输吞吐量提升 2-3 倍，接近同构性能（基准测试数据）。
  - 系统：新增 GPU 缓冲区和环境变量，增加内存使用但可控（环境变量可调）。
  - 团队：代码库增加新模块，需维护 staging 相关逻辑，但设计隔离良好（staging_handler.py）。

关联脉络

与历史 PR #18158 和 #18163 相关，后者曾尝试类似方法但未解决长上下文问题，如 issue 评论中 Ch3ngY1 提到：'in long-context scenarios, the buffer can hardly accommodate the entire context's kv cache'。本 PR 通过分块传输和动态分配器克服限制，体现了 disaggregation 性能优化的持续演进。近期 PR 如 #21233（清理冗余代码）和 #21834（JIT 内核优化）也显示团队在核心路径上的改进趋势。