

PR #56157 完整报告

vllm-project/vllm

[PCP][DCP] Enable PCP+DCP on sparse-MLA models

合并时间: 2026-09-13 07:24

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56157>

执行摘要

- 一句话: 稀疏 MLA 模型开启 PCP+DCP: 统一 chunk 排布修复集体通信挂起
- 推荐动作: 值得精读。这是 vLLM 分布式并行 (PCP×DCP 组合) 在稀疏注意力模型上的一次关键落地, 核心看点有三: ①"所有 rank 派生一致的 collective 形状" 这一不变式的工程化——通过确定性行序、短 prefill 复制、专用 shard 元数据字段三个手段共同实现; ②PCPGlobalChunkPlan 的 deinterleave 索引设计, 把 rank-major all-gather 布局无通信地还原为全局 token 顺序, 是理解 PCP+DCP 数据流的最佳入口; ③review 过程中从 "全局 PCP schedule" 到 "按请求分组 + req_idx 传递" 的重构, 展示了如何在分布式代码中做减法。若要在生产使用 PCP+DCP, 请同时追踪 fused all-reduce for CP groups 的 followup PR, 并关注 decode 性能是否被接受。

功能与动机

Issue #53573 描述了 PCP+DCP 组合在 MLA 模型上的根本性 bug: PCPManager 按 rank-local 虚拟 batch 独立派生 prefill 元数据, 不同 PCP rank 拥有不同的 query 位置, 导致本地 context 长度和虚拟请求行布局不同, 最终 DCP KV-gather 的 collective 在数量、顺序、形状上发散。对两 token prefill 会直接造成 **direct DCP kv-gather multimem timeout**, 对长 prefill 则会把 KV payload 关联到错误的 query 行, 静默破坏注意力结果。PR body 明确目标是「Enable DCP to run alongside PCP on sparse-MLA models」, 并指出 dense MLA 路径无法在不重做 chunk 计划的情况下修复, 因此改为在配置校验期显式拒绝 dense MLA 的 PCP+DCP 组合。

实现拆解

实现按 5 个步骤推进:

1. 建立跨 rank 一致的 PCP 行排布不变式 (vllm/v1/worker/gpu/pcp_manager.py)。
_reorder_segments 从基于 num_computed_tokens 的纯 prefill 置后排序, 改为确定性的 sort_key = (is_prefilling, req_idx, global_batch_slice.start), 保证每个 PCP rank 发布的行到全局请求的映射完全一致, 且 decode 永远在 prefill 之前 (这是 split_decodes_and_prefills 的边界假设)。新增 replicated_requests: 当 DCP>1 时, 若一个 prefill 太短、无法填满 2*PCP 个 chunk (即某些 rank 会拿到空或短 chunk), 就整体复制到所有 rank 而非切分, 使每个 region 的行数在所有 rank 上一致。
partition_batch 新增专用字段 dcp_local_seq_lens_cpu_upper_bound 发布每个请求的最大 DCP shard 长度, 不再借用通用的 seq_lens_cpu_upper_bound。

2. sparse indexer 的全局 chunk plan (vllm/v1/attention/backends/mla/indexer.py、sparse_utils.py)。新增 `request_row_bounds` 把 PCP rank 上相邻属于同一请求的行归为一个 region；新增 dataclass `PCPGlobalChunkPlan` 和 `build_pcp_global_chunk_plan`，基于所有 rank 一致的 shard 行数计算全局布局（按 `shard rows × dcp_world_size` 对齐），生成 rank-major gathered buffer 的 `deinterleave_idx`，并用 pinned host 缓冲区 + `async_copy_to_gpu` 避免 CPU-GPU 同步。`_split_pcp_dcp_prefill_chunks` 按请求而非按行切 chunk，让 split prefill 的两个相邻行共享一次 context 计费。`DeepseekV32IndexerPrefillChunkMetadata` 增加 `pcp_deinterleave_idx` 字段。
3. FlashMLA-sparse 后端的 gathered prefill 路径 (vllm/v1/attention/backends/mla/flashmla_sparse.py)。新增 `pcp_dcp_kv_gather` 标志：PCP+DCP 时强制走 separate prefill/decode 路径（mixed-batch 路径无法在 gathered KV 上做 prefill），并要求 `fp8_ds_mla` cache dtype 与 `ag_rs` comm backend。`gathered_prefill_shards` 按请求分组取 shard 行数，workspace 每请求一项、每项持有本 rank 的 KV shard；`_gather_prefill_chunk` 在 per-chunk 粒度 all-gather upconvert 后的 shard。`sparse_mla_attention.py` 配套在 PCP+DCP 下禁用 dense MHA 快速路径。
4. KV all-gather 执行与注意力合并 (vllm/model_executor/layers/sparse_attn_indexer.py、vllm/models/deepseek_v32/attention.py、mla_attention.py)。新增 `dcp_gather_kv_rows` 执行 `all_gather_into_tensor` + `index_select` deinterleave，workspace 管理为 gathered/deinterleaved 结果各申请一对 buffer；PCP 路径下跳过 `_merge_dcp_topk_global` (top-k 已在全局 context 上完成)。`MLAAttention` 增加 `supports_pcp_dcp` ClassVar 并在构造时拒绝不支持的后端（回应 GirasoleY 的 review），`DeepseekV32Attention` 声明支持并只对 decode 行做 LSE merge (prefill 行已看过全量 KV)。顺带修复 kernel/reference 分歧：prefill-workspace 行的 block-table bound 判定修正为 `is_remote & ~is_prefill`。
5. 配置约束与测试配套。`validate_config` 显式拒绝 dense MLA 的 PCP+DCP、sparse MLA 无 DCP 时的 CUDA graph、非 `ag_rs` backend，`SparseAttnIndexerKpool` 与 ROCm 路径各自增加拒绝分支。测试方面：`test_gpu_pcp_manager.py` 新增 5 个不变式测试（行序一致、decode 先行、重复行携带完整 extent、短 prefill 复制、首行永远非短 chunk），`test_indexer_dcp_localize.py` 新增 deinterleave 顺序还原测试，并新增 GSM8K eval 配置 `GLM-5.2-NVFP4-TP1-PCP4-DCP4-EP.yaml`；后续 commit 还修复了稀疏 MLA 测试 fake 配置缺失的问题。

关键文件：

- vllm/v1/attention/backends/mla/indexer.py (模块 索引器；类别 source；类型 core-logic；符号 `PCPGlobalChunkPlan`, `build_pcp_global_chunk_plan`, `_split_pcp_dcp_prefill_chunks`)：PCP+DCP 的核心算法所在：新增 `PCPGlobalChunkPlan`、`build_pcp_global_chunk_plan` 和 `_split_pcp_dcp_prefill_chunks`，基于全 rank 一致的 shard 行数构造全局 chunk 布局与 deinterleave 索引，是整个方案的不变式支柱。
- vllm/v1/attention/backends/mla/flashmla_sparse.py (模块 FlashMLA；类别 source；类型 dependency-wiring；符号 `gathered_prefill_shards`, `_gather_prefill_chunk`)：PCP+DCP 的 FlashMLA-sparse 消费端：新增 `pcp_dcp_kv_gather` 标志强制走 separate prefill/decode 路径，`gathered_prefill_shards` 把 workspace 按请求组织、每个请求持有

本 rank 的 KV shard, `_gather_prefill_chunk` 在 chunk 粒度执行 all-gather。

- `vllm/v1/worker/gpu/pcp_manager.py` (模块 PCP 调度; 类别 source; 类型 dependency-wiring; 符号 `is_pure_prefill`, `sort_key`, `replicated_requests`) : 跨 rank 一致性的源头: `replicated_requests` 决定短 prefill 复制策略, `_reorder_segments` 改为确定性排序 (decode 先行、按请求序), `partition_batch` 发布专用字段 `dcp_local_seq_lens_cpu_upper_bound`, 并显式拒绝 dense MLA 的 PCP+DCP (#53573 未修复路径)。
- `vllm/model_executor/layers/sparse_attn_indexer.py` (模块 稀疏索引; 类别 source; 类型 data-contract; 符号 `dcp_gather_kv_rows`) : PCP+DCP 的 KV all-gather 执行点: 新增 `dcp_gather_kv_rows` (`all_gather_into_tensor` + `index_select` deinterleave), `workspace` 管理器为 `gathered/deinterleaved` 结果申请配对 buffer, PCP 路径跳过 `_merge_dcp_topk_global`。
- `tests/v1/worker/test_gpu_pcp_manager.py` (模块 PCP 测试; 类别 test; 类型 test-coverage; 符号 `_rank_rows`, `test_published_row_order_is_identical_on_every_pcp_rank`, `test_published_row_order_puts_every_decode_before_every_prefill`, `test_split_prefill_rows_repeat_the_request_and_its_full_extent`) : 守护本 PR 全部核心不变式的测试: 行序全 rank 一致、decode 先行、拆分行重复请求与完整 extent、短 prefill 复制、首行永不短 chunk, 是未来 PCP 分区策略变更时的回归防线。
- `vllm/v1/attention/backends/mla/sparse_utils.py` (模块 稀疏工具; 类别 source; 类型 core-logic; 符号 `request_row_bounds`) : 提供 `request_row_bounds` 区域划分工具, 并修正 prefill workspace 行不再应用 block-table bound 的 kernel 行为 (`is_remote` & `~is_prefill`), 同时给 `triton_convert_req_index_to_global_index` 增加 `workspace_rank_stride` 与 DCP deinterleave 参数。
- `vllm/models/deepseek_v32/attention.py` (模块 模型注意力; 类别 source; 类型 data-contract; 符号 `supports_pcp_dcp`) : DeepSeek V3.2 注意力层声明 `supports_pcp_dcp=True`, 并在 PCP+DCP 下只对 decode 行的 LSE 做 DCP merge (prefill 行已看过全量 gathered KV), 是模型侧适配的关键。
- `vllm/model_executor/layers/attention/mla_attention.py` (模块 MLA 注意力; 类别 source; 类型 data-contract; 符号 `supports_pcp_dcp`) : 新增 `supports_pcp_dcp` 契约并在构造期拒绝不支持 PCP+DCP 的 MLA 子类, 落实 GirasoleY 的 review 建议 (构造期失败优于 forward 期失败)。

关键符号: `build_pcp_global_chunk_plan`, `_split_pcp_dcp_prefill_chunks`, `replicated_requests`, `_iter_rank_chunks`, `_reorder_segments`, `gathered_prefill_shards`, `_gather_prefill_chunk`, `dcp_gather_kv_rows`, `request_row_bounds`

关键源码片段

`vllm/v1/worker/gpu/pcp_manager.py`

跨 rank 一致性的源头: `replicated_requests` 决定短 prefill 复制策略, `_reorder_segments` 改为确定性排序 (decode 先行、按请求序), `partition_batch` 发布专用字段 `dcp_local_seq_lens_cpu_upper_bound`, 并显式拒绝 dense MLA 的 PCP+DCP (#53573 未修复路径)。

```

def replicated_requests(
    self,
    num_scheduled_tokens: np.ndarray,
    is_prefilling: np.ndarray,
) -> np.ndarray:
    """每个全局请求是否让所有 PCP rank 都拿到完整 query。"""
    num_chunks = 2 * self.pcp_world_size
    query_lens = np.asarray(num_scheduled_tokens, dtype=np.int64)
    # decode 天然复制到所有 rank
    replicated = ~np.asarray(is_prefilling, dtype=np.bool_)
    if self.dcp_world_size > 1:
        chunk_sizes = (query_lens + num_chunks - 1) // num_chunks
        # 若前 num_chunks-1 个 chunk 已覆盖整个 query, 说明最后一个 chunk
        # 为空或过短, 某些 rank 会拿到空行, 导致各 rank 的 DCP KV-gather
        # collective 形状不一致 (#53573 的根因之一)。此时整体复制。
        drops_a_chunk = (num_chunks - 1) * chunk_sizes >= query_lens
        replicated |= drops_a_chunk
    return replicated

```

```

def _iter_rank_chunks(
    self,
    rank: int,
    num_scheduled_tokens: np.ndarray,
    is_prefilling: np.ndarray,
) -> Iterator[tuple[int, int, int]]:
    """产出该 PCP rank 的 ``(请求索引, query 偏移, 长度)``。

```

PCP=4 时每个 prefill 被切成 8 个 chunk (DualChunkSwap) :

rank 0 拿首尾、rank 1 拿第 2/倒数第 2……

"""

```

num_chunks = 2 * self.pcp_world_size
replicated = self.replicated_requests(num_scheduled_tokens, is_prefilling)
for global_batch_req_idx, num_tokens in enumerate(num_scheduled_tokens):
    query_len = int(num_tokens)
    if query_len == 0:
        continue
    chunk_indices: tuple[int, ...]
    if not replicated[global_batch_req_idx]:
        chunk_size = (query_len + num_chunks - 1) // num_chunks
        chunk_indices = (rank, num_chunks - 1 - rank)
    else:
        # decode, 以及 DCP 下过短的 prefill, 都整体复制。
        chunk_size = query_len
        chunk_indices = (0,)
    for chunk_idx in chunk_indices:
        chunk_offset = chunk_idx * chunk_size
        chunk_len = min(chunk_size, query_len - chunk_offset)
        if chunk_len <= 0:

```

```
        continue
    yield global_batch_req_idx, chunk_offset, chunk_len
```

评论区精华

LucasWilkinson 的重构主导了最终形态。他提出四点原则：最小化对核心文件（`model_runner.py`、`cuda_graph_utils.py`）的改动、移除新引入的全局状态 `_current_pcp_schedule`、避免过多新数据结构、尽量与已有 PCP/DCP 路径统一。作者 PatrykSaffer 明确接受（"LGTM, for the *Make the gathered prefill a mode of the separate prefill/decode path* I intentionally put it separately to avoid if/else branching"），随后合入了 LucasWilkinson 的建议分支（commit 7553c5f、8871b3a）。

foraxe 的两个设计关切在 review 评论中被直接提出（issue 评论留言）：一是 `seq_lens_cpu_upper_bound` 在 PCP+DCP 下被悄悄赋予 "whole global-request extent" 新语义，该字段被多个 backend 消费，建议要么加强契约 / 断言、要么用专用字段——LucasWilkinson 在 commit df370c7 中落实了后者，改为发布专用字段 `dcp_local_seq_lens_cpu_upper_bound`；二是 `_split_pcp_dcp_prefill_chunks` 假设首行永远是 full chunk、短尾在最后，是 PCP 分区策略的隐藏耦合——作者回应 "added asserts and test"，新增了 `test_pcp_first_chunk_row_is_never_short`。

GirasoleY 的三条评论：①在 `mla_attention.py` 建议把 forward 时的 raise 提前到类构造时拒绝（"Instead of raise during forward could we just reject at class construction?"）——最终以 `supports_pcp_dcp` ClassVar + `__init__` 检查实现；②在 `indexer.py:344` 指出 plan 构建中的 CPU-GPU 同步问题（"These will cause CPU-GPU sync, use host pinned buffer and non_blocking here"）——最终用 `PIN_MEMORY` + `async_copy_to_gpu` 解决；③对 `sparse_attn_indexer_kpool.py` 的拒绝提出疑问（"I thought kpool indexer is needed for GLM5.3"），该点没有完全解决，代码保留了拒绝分支。

- LucasWilkinson 的减法式重构建议 (design): 作者采纳并合入建议分支，最终实现改为传递 `req_idx`、去掉模块级 PCP schedule，diff 显著缩小。
- `seq_lens_cpu_upper_bound` 字段语义被静默扩展 (design): LucasWilkinson 在 commit df370c7 改为发布专用字段 `dcp_local_seq_lens_cpu_upper_bound`（携带每行所属请求的最大 DCP shard 长度），不再覆盖通用字段。
- chunk plan 对 PCP 分区策略的隐藏耦合 (testing): PatrykSaffer 回应 added asserts and test，新增 `test_pcp_first_chunk_row_is_never_short` 覆盖 2/4/8 PCP 多组 query 长度。
- forward 期 raise 改为构造期拒绝 (design): 最终以 `supports_pcp_dcp` ClassVar（默认 False）+ `MLAAttention.__init__` 检查实现，DeepseekV32Attention 显式 opt-in。
- chunk plan 构建中的 CPU-GPU 同步 (performance): 改为 `PIN_MEMORY` pinned host 缓冲 + `async_copy_to_gpu`，commit a59ce95 落实。
- SparseAttnIndexerKpool 是否需要拒绝 PCP+DCP (question): 代码保留拒绝分支，评论未见进一步回复；GLM5.3 用户若使用 kpool 索引器则无法开启 PCP+DCP，属未完全解决的前置疑虑。
- PCP+DCP decode 性能退化与 fused all-reduce followup (performance): fused all-reduce 拆到独立 PR（commit 590e9b7 移出本 PR），合并版以明确限制换正确性，

decode 性能留待 followup 优化。

风险与影响

• 风险:

1. decode 性能显著退化: 合并版移除了 FULL decode graph 与 MoE 去重优化, PR body 自测数据显示 pcp4dcp4 在 PIECEWISE 下 TPOT 达 86.7ms (仅 0.22x TP4), 远差于纯 DCP4 的 28.4ms。根因是 CudaCommunicator.__init__ 只为 TP group 启用 custom all-reduce / symm-mem / FlashInfer fused all-reduce, PCP+DCP 的 per-layer all-reduce 走 NCCL ring; fused all-reduce 被拆到单独 PR (commit 590e9b7 已移出), 在合入前该组合的 decode 性价比需要业务方自行评估。
2. 行序策略变更影响纯 PCP 路径: _reorder_segments 的排序键从 " 纯 prefill 置后 " 变为 "decode 优先 + req_idx + batch slice 升序", 这是所有 PCP 场景的公共路径, 不只 PCP+DCP。若其他 attention backend 对行序有隐性假设, 可能回归。新测试覆盖了部分组合, 但 backend 矩阵未穷尽。
3. 字段契约风险: dcp_local_seq_lens_cpu_upper_bound 目前只被 sparse backend 消费, dense MLA 的 PCP+DCP 已被显式拒绝兜底; 但未来若在非 PCP 场景误用该字段, 会因无断言保护而静默出错。
4. 复制短 prefill 的计算冗余: replicated_requests 在 DCP>1 时把短 prefill 复制到每个 PCP rank, 牺牲算力换取 collective 一致性; 大量短 prompt 场景下 prefill 效率会下降。
5. 平台/后端限制: ROCmsparse-indexer与SparseAttnIndexerKpool (GLM-5.3相关) 均显式拒绝 PCP+DCP, 验证仅覆盖 NVIDIA GB200 (GLM-5.2-NVFP4-TP1-PCP4-DCP4-EP), 跨平台推广前需补验证; build_pcp_global_chunk_plan 对空 context 请求直接 assert, 调度层必须保证不会下发空 context 的 prefill。- 影响: 对用户: DeepSeek V3.2+ / GLM 等 sparse-MLA 模型的部署新增了 TP1+PCP4+DCP4 形态, prefill TTFT 相比纯 TP4 提升约 1.66x (3278ms vs 5512ms) 且 GSM8K 准确率无退化 (0.832 vs 0.830), 适合 " 长 prompt 拆分 + decode KV 分片 " 的场景; 但合并版在 decode 端 (PIECEWISE 路径) 暂时有明显开销, 需要搭配后续 fused all-reduce 优化才能达到生产级。对系统: PCP 管理器与 sparse attention 核心路径的排序、chunk 计划、metadata 契约均被修改, supports_pcp_dcp ClassVar 成为注意力层的新契约, 后续后端 (dense MLA、HiSparse、kpool) 接入 PCP+DCP 都要显式声明并实现 gather 语义。对团队: 该 PR 是 31 commits 的多人协作样本, LucasWilkinson 以 " 减法式 "review 分支显著缩小了最终 diff, 确立了 " 减少全局状态、数据流显式化、构造期失败 " 的分布式路径设计风格。- 风险标记: collective 一致性不变式依赖测试守护, decode 性能退化 (PIECEWISE 下 0.22x TP4), 核心排序策略影响纯 PCP 路径, 仅 NVIDIA Blackwell 验证, ROCm 与 kpool 路径显式拒绝

关联脉络

- PR #56562 [Perf] Fuse DSV4.1 input metadata preparation with Triton: 同属 DeepSeek 稀疏 MLA 索引器路径, 两 PR 都修改了 tests/v1/attention/test_indexer_deepseek_v4_slot_mapping.py, 且都在打磨 sparse indexer 在 v1 上的元数据构建与 kernel

契约。

- PR #53566 [5/N][warmup][DSv4] Migrate NVIDIA CuTeDSL attention kernels: 同属 DeepSeek NVIDIA 路径的 kernel 契约迁移；本 PR 修改的 `sparse_attn_indexer` 依赖 DeepGEMM warmup 契约，两者在预热覆盖与 workspace 契约上有直接交集。
- PR #53781 [3/N] HiSparse: host-resident sparse-MLA decode hot-buffering: 同属 sparse-MLA 后端演进方向；HiSparse 与 PCP+DCP 的 gathered KV workspace 都作用于稀疏 MLA 注意力路径，本 PR 在 `flashmla_sparse.py` 中显式处理了 `use_hispase` 与 `pcp_dcp_kv_gather` 的互斥关系。