PR #51733 完整报告

vllm-project/vllm

[Attention] Fix MLA prefill workspace allocation size

合并时间：2026-08-11 13:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51733

执行摘要

- 一句话：修复 MLA prefill workspace 按 max_num_seqs 放大的显存回归
- 推荐动作：值得精读。该 PR 展示了 per-request 调度对内存分配契约的影响，是 #50613 与 #50484 拉锯后的最终回归修复，逻辑简单但涉及 MLA 核心路径。关注点：workspace 容量与调度器的契约如何随调度模型演进，以及 DCP 对齐计算的边界。

功能与动机

PR body 明确说明："With #50613（[Attention][MLA] Per-request scheduling for MLA chunked context），MLA prefill workspace no longer requires to be at least max-num-seqs * block_size. This requirement is added back in https://github.com/vllm-project/vllm/pull/50484. This PR reverts it." 即 per-request 调度引入后，workspace 容量只需覆盖单个 chunk step，#50484 却把 max_num_seqs * block_size 的下限加回，导致长上下文、大并发配置下 prefill 显存被过度预留。本 PR 的目的是还原该回归。

实现拆解

1. mla_attention.py：修改 align_mla_chunked_context_workspace_size，删除 max(workspace_size, vllm_config.scheduler_config.max_num_seqs * alignment) 下限，改为 round_up(max(workspace_size, alignment), alignment)。DCP 场景下 alignment 仍取 lcm(block_size, decode_context_parallel_size * cp_kv_cache_interleave_size)，保证 all-gather 分片对齐。
2. sparse_mla_attention.py：修改 determine_chunked_prefill_workspace_size，将下限从 scheduler_config.max_num_seqs * cache_config.block_size 降为 cache_config.block_size，与 per-request chunk 工作区契约一致。
3. 测试文件 tests/distributed/test_dcp_direct_a2a_lse_reduce.py：更新 test_dcp_chunk_workspace_alignment_covers_interleave，移除 max_num_seqs = 3 的配置，断言从单一 192 调整为 128（输入 100）与 64（输入 8），并在注释中说明 workspace 只须容纳单个对齐 chunk step、与 max_num_seqs 无关。
4. 验证：跑通 test_mla_backends.py、test_sparse_mla_backends.py，并在 Kimi K3 DCP 8 B300 上完成 GSM8k 精度验证（exact_match 0.9568）。

关键文件：

- vllm/model_executor/layers/attention/mla_attention.py（模块 注意力层；类别 source；类型 data-contract；符号 align_mla_chunked_context_workspace_size）：核心修改文件，align_mla_chunked_context_workspace_size 移除 max_num_seqs 线性下限，决定 MLA dense prefill workspace 的分配大小。
- vllm/model_executor/layers/attention/sparse_mla_attention.py（模块 注意力层；类别 source；类型 data-contract；符号 determine_chunked_prefill_workspace_size）：稀疏 MLA 的 workspace 计算同步修正，determine_chunked_prefill_workspace_size 下限从 max_num_seqs * block_size 降为 block_size。
- tests/distributed/test_dcp_direct_a2a_lse_reduce.py（模块 测试用例；类别 test；类型 test-coverage；符号 test_dcp_chunk_workspace_alignment_covers_interleave）：DCP 对齐测试同步更新，验证新语义（workspace 与 max_num_seqs 无关），防止回归。

关键符号：align_mla_chunked_context_workspace_size, determine_chunked_prefill_workspace_size

关键源码片段

vllm/model_executor/layers/attention/mla_attention.py

核心修改文件，align_mla_chunked_context_workspace_size 移除 max_num_seqs 线性下限，决定 MLA dense prefill workspace 的分配大小。

```
def align_mla_chunked_context_workspace_size(
    vllm_config: VllmConfig,
    workspace_size: int,
) -> int:
    """计算 MLA chunked context workspace 对齐后的大小。

    在 per-request 调度（PR#50613）之后，每个请求按序使用 workspace，
    因此容量只需容纳单个对齐的 chunk step，不再需要按最大并发请求数
    max_num_seqs 线性放大（该下限是 PR#50484 加回的回归）。
    """
    parallel_config = vllm_config.parallel_config
    # 基础对齐单位是 KV cache 的 page 大小
    alignment = vllm_config.cache_config.block_size
    if parallel_config.decode_context_parallel_size > 1:
        # DCP 场景下，所有 rank 的分片需要经过 all-gather 合并，
        # 因此对齐单位取 block_size 与 dcp 分片大小的最小公倍数（lcm）
        alignment = lcm(
            alignment,
            parallel_config.decode_context_parallel_size
            * parallel_config.cp_kv_cache_interleave_size,
        )
    # 只保证 workspace 不小于单个对齐单位，避免为 max_num_seqs 预留过多显存
    return round_up(max(workspace_size, alignment), alignment)
```

vllm/model_executor/layers/attention/sparse_mla_attention.py

稀疏 MLA 的 workspace 计算同步修正，determine_chunked_prefill_workspace_size 下限从 max_num_seqs * block_size 降为 block_size。

```
@staticmethod
def determine_chunked_prefill_workspace_size(vllm_config: "VllmConfig") -> int:
    scheduler_config = vllm_config.scheduler_config
    cache_config = vllm_config.cache_config
    model_config = vllm_config.model_config
    topk_tokens = model_config.hf_config.index_topk

    # 稀疏 MLA 的 workspace 用于 gather 每请求的 topk token 分块:
    # 取若干启发式上限的较小值，保证不超 64 KiB 单块容量
    workspace_size = min(
        max(
            8 * model_config.max_model_len,
            4 * scheduler_config.max_num_seqs * cache_config.block_size,
        ),
        64 * 1024,
        scheduler_config.max_num_seqs * topk_tokens,
    )
    # 下限从 max_num_seqs * block_size 降为单个 block_size:
    # per-request 调度后同一时刻只处理一个 chunk，不随并发数线性放大
    workspace_size = max(workspace_size, cache_config.block_size)
    if vllm_config.parallel_config.decode_context_parallel_size > 1:
        # DCP 场景由公共工具函数负责额外的对齐计算
        return align_mla_chunked_context_workspace_size(vllm_config, workspace_size)
    return workspace_size
```

评论区精华

本次 PR 无实质 review 讨论。三位维护者均直接 approve：zyongye 触发 CI 并 approve；LucasWilkinson 评论 "LGTM thanks for fixing!"；合并者 MatthewBonanni 评论 "LGTM thanks!"。claude[bot] 因 fork 来源自动 review 被禁用，仅提示可手动触发。未留下未解决疑虑。

- 改动范围与回归风险确认 (other): 改动被接受并合并，未留下未解决疑虑。

风险与影响

- 风险:
  1. workspace 容量边界：per-request 调度下 workspace 从按 max_num_seqs 线性放大降为单 chunk 容量，若某个请求的 chunk 超过单个 workspace 行容量，理论上存在越界风险。现有测试只覆盖对齐计算，未覆盖超长请求边界，建议关注超长上下文场景。
  2. DCP 对齐改动：align_mla_chunked_context_workspace_size 中 alignment 计算保留，但最低容量从 max_num_seqs * alignment 降为单 alignment，多卡 all-gather 的边界依赖调度契约，需要在大并行度（如 8 卡以上）配置下实测。
  3. 测试对应关系：测试文件 tests/distributed/test_dcp_direct_a2a_lse_reduce.py 与 mla_attention.py 的关联较弱，未直接覆盖 sparse_mla_attention.py 的 workspace 计

算分支（DCP 分支除外）。 - 影响：该改动直接影响 MLA 架构模型（DeepSeek、Kimi 等）的 prefill 显存分配。对长上下文、大并发、DCP 场景收益明显：workspace 不再随 max_num_seqs 线性放大，避免显存浪费。对用户透明，无需修改配置；精度验证无回退。团队侧，这是一个 7 行小改动，但修正了 #50484 引入的回归，影响面覆盖所有 MLA prefill 路径。 - 风险标记：核心路径变更，workspace 容量契约，DCP 对齐边界

关联脉络

- PR #50613 [Attention][MLA] Per-request scheduling for MLA chunked context: 本 PR 的动机来源：引入 per-request 调度后 workspace 不再需要 max_num_seqs 线性放大，本 PR 修复 #50484 加回该下限的回归。
- PR #50484 （标题未在材料中提供）: PR body 指出该 PR 将 max_num_seqs * block_size 的下限加回，本 PR 将其还原。
- PR #51739 [Kernel] Optimize long-context MLA cache gathers: 同文件 vllm/model_executor/layers/attention/mla_attention.py 的 MLA 路径优化，属于同一模块近期演进。