

PR #47987 完整报告

vllm-project/vllm

Make tiering offload region DP-replica aware

合并时间: 2026-07-12 18:10

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47987>

执行摘要

- 一句话: 使卸载区域 DP 副本感知, 修复 multiprocessing 路径下 mmap 冲突
- 推荐动作: 值得合并, 修复了 DP 下 KV 卸载的关键冲突问题, 设计决策清晰 (复用已有 DP 感知的 `engine_id`)。建议关注 `create_worker` 中 rank 计算的假设是否普适, 以及在非标准设备映射下的表现。

功能与动机

PR body 指出每个数据并行副本是一个独立引擎, 拥有自己的 KV 块, 因此 `/dev/shm` 卸载区域必须不同。Issue 评论中 orozery 指出 multiprocessing 路径 (`CoreEngineProcManager`) 下的 `instance_id` 未按 DP 副本唯一化, 而 Ray 路径已通过 `_apply_dp_identity_suffix` 正确添加后缀, 因此需要在 multiprocessing 路径 (`run_engine_core`) 中补上 `instance_id` 的 DP 后缀, 或直接使用已 DP 感知的 `engine_id`。

实现拆解

1. 在层级卸载规格模块 (`vllm/v1/kv_offload/tiering/spec.py`) 中引入 DP 感知标识: 在 `TieringOffloadingSpec.__init__` 中新增断言, 确保 `vllm_config.kv_transfer_config.engine_id` 不为空, 并将其保存为实例变量 `self._engine_id`。该 `engine_id` 在 Ray 路径 (`_apply_dp_identity_suffix`) 和 multiprocessing 路径 (`run_engine_core`) 中均已自动附加 `_dp{rank}` 后缀, 确保每个 DP 副本拥有唯一值。
2. 将 `SharedOffloadRegion` 构造参数从 `instance_id` 替换为 `engine_id`: 在 `get_manager` 中创建调度器端 mmap 时, 以及在 `create_worker` 中创建工作进程 mmap 时, 均使用 `self._engine_id` 替代原来的 `self.vllm_config.instance_id`。文件名随之变为 `/dev/shm/vllm_offload_{engine_id}.mmap`。
3. 修复工作进程 slot rank 计算: `create_worker` 原来直接使用 `torch.accelerator.current_device_index()` 作为 slot 索引, 这会返回全局物理设备编号。在 DP 下, 每个副本只拥有连续设备子集, 因此改为 `current_device_index() % world_size`, 使 rank 落在 `[0, world_size)` 范围内。
4. 更新 `SharedOffloadRegion` 类 (`vllm/v1/kv_offload/cpu/shared_offload_region.py`): 将构造函数参数名从 `instance_id` 改为 `engine_id`, 并更新 docstring 中的文件路径示例。

5. 同步更新测试文件：test_shared_offload_region.py 和 test_gpu_worker.py 中的所有辅助函数和测试用例将参数名从 instance_id 改为 engine_id，确保与新的 API 签名兼容。

关键文件：

- vllm/v1/kv_offload/tiering/spec.py（模块 层级卸载规格；类别 source；类型 core-logic；符号 TieringOffloadingSpec.init, TieringOffloadingSpec.get_manager, TieringOffloadingSpec.create_worker）：核心逻辑变更：引入 DP 感知的 engine_id 替代 instance_id，并调整 worker rank 计算。
- vllm/v1/kv_offload/cpu/shared_offload_region.py（模块 卸载区域；类别 source；类型 core-logic；符号 SharedOffloadRegion.init）：SharedOffloadRegion 类参数重命名，反映标识含义的变更。
- tests/v1/kv_offload/cpu/test_shared_offload_region.py（模块 测试；类别 test；类型 test-coverage；符号 _make_region, _region, _multi_region, _race_construct）：测试辅助函数和所有测试用例适配 engine_id 参数名。
- tests/v1/kv_offload/cpu/test_gpu_worker.py（模块 测试；类别 test；类型 test-coverage）：测试中创建 SharedOffloadRegion 时使用 engine_id 参数。

关键符号：TieringOffloadingSpec.init, TieringOffloadingSpec.get_manager, TieringOffloadingSpec.create_worker, SharedOffloadRegion.init

关键源码片段

vllm/v1/kv_offload/tiering/spec.py

核心逻辑变更：引入 DP 感知的 engine_id 替代 instance_id，并调整 worker rank 计算。

```
# vllm/v1/kv_offload/tiering/spec.py
class TieringOffloadingSpec:
    def __init__(self, vllm_config: VllmConfig, kv_cache_config: KVCacheConfig):
        super().__init__(vllm_config, kv_cache_config)
        self._manager: OffloadingManager | None = None
        # ...
        # engine_id is unique per DP replica (suffixed with _dp{rank} in both
        # the Ray and multiprocessing paths), so it names a per-replica offload
        # region. Non-None is guaranteed by OffloadingSpec.__init__.
        assert vllm_config.kv_transfer_config is not None
        assert vllm_config.kv_transfer_config.engine_id is not None
        self._engine_id: str = vllm_config.kv_transfer_config.engine_id

    def get_manager(self) -> OffloadingManager:
        if not self._manager:
            scheduler_mmap = SharedOffloadRegion(
                engine_id=self._engine_id, # 使用 DP 感知标识
                num_blocks=self.num_blocks,
                rank=None,
                kv_bytes_per_block=self.kv_bytes_per_offloaded_block,
                cpu_page_size=self.cpu_page_size_per_worker,
            )
```

```

        # ...
    return self._manager

@override
def create_worker(self, kv_caches: CanonicalKVCaches) -> CPUOffloadingWorker:
    # Fold the global physical device index into the replica-local
    # [0, world_size) slot range.
    world_size = self.vllm_config.parallel_config.world_size
    rank = torch.accelerator.current_device_index() % world_size
    worker_mmap = SharedOffloadRegion(
        engine_id=self._engine_id, # 使用相同的副本唯一标识
        num_blocks=self.num_blocks,
        rank=rank,
        kv_bytes_per_block=self.kv_bytes_per_offloaded_block,
        cpu_page_size=self.cpu_page_size_per_worker,
    )
    # ...

```

vllm/v1/kv_offload/cpu/shared_offload_region.py

SharedOffloadRegion 类参数重命名，反映标识含义的变更。

vllm/v1/kv_offload/cpu/shared_offload_region.py

class SharedOffloadRegion:

"""

Single mmap-backed memory region shared across all workers for a vLLM instance. Workers coordinate via the filesystem: the first worker to open the file with O_EXCL becomes the creator and calls ftruncate; the rest open the existing file and wait until it reaches the expected size. Each worker then mmap()s the full file.

File path: /dev/shm/vllm_offload_{engine_id}.mmap # 更新为 engine_id

"""

```

def __init__(
    self,
    engine_id: str, # 从 instance_id 更名
    num_blocks: int,
    rank: int | None,
    kv_bytes_per_block: int,
    cpu_page_size: int,
) -> None:
    self.page_size = mmap.PAGESIZE
    assert kv_bytes_per_block % self.page_size == 0
    self.num_blocks = num_blocks
    self._row_stride = kv_bytes_per_block
    self.total_size_bytes = self.num_blocks * self._row_stride
    self.mmap_path = f"/dev/shm/vllm_offload_{engine_id}.mmap" # 文件名使用 engine_id
    # ...

```

评论区精华

orozery 建议直接使用 `kv_transfer_config.engine_id` (该 ID 已在两种部署路径中附加 DP 后缀)，并重构 `SharedOffloadRegion` 的参数名从 `instance_id` 为 `engine_id`。liranschour 采纳建议后，orozery 批准合并。讨论中还对 `create_worker` 中的 rank 计算提出了修复：将全局物理设备索引 `torch.accelerator.current_device_index() % world_size` 以适配 DP 下每个副本只拥有连续设备子集。

- 使用 `engine_id` 替代 `instance_id` 并重命名 (design): liranschour 采纳建议并实施，随后 orozery 批准合并。

风险与影响

- 风险：变更集中于卸载区域的标识和 rank 计算，影响范围限于 DP 模式下的 KV 卸载功能。主要风险包括：1) 如果 `kv_transfer_config.engine_id` 在某些配置下为 `None`，则引入的 `assert` 会导致启动失败（但根据注释应是 `Guaranteed`）；2) 将 rank 取模 `world_size` 后，假设每个 DP 副本的设备索引连续且从 `rank * world_size` 开始，若实际设备分配不连续（例如通过 `CUDA_VISIBLE_DEVICES` 自定义），则可能出错。测试覆盖了基本场景，但未覆盖复杂的设备映射场景。
- 影响：影响使用数据并行（DP）且启用 KV 卸载的功能的用户。修复后，多个 DP 副本可以独立使用 `/dev/shm` 卸载区域，不会因文件名冲突或索引越界导致失败。变更向后兼容，以 `engine_id` 替代 `instance_id`，两者在非 DP 模式下值相同，因此对非 DP 用户无影响。测试文件同步更新，确保功能和回归覆盖。
- 风险标记：假设设备索引连续，`engine_id` 必须非空

关联脉络

- 暂无明显关联 PR