

PR #49440 完整报告

vllm-project/vllm

[Bugfix][KV Offload] Namespace persistent cache by model runner

合并时间: 2026-07-26 13:21

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49440>

执行摘要

- 一句话: 为 KV 卸载持久缓存命名空间添加模型运行器标识
- 推荐动作: 建议详细阅读该 PR 的 issue 讨论和测试代码, 了解 KV 卸载缓存布局的差异及处理方式。核心设计决策 (将 `parallel_agnostic` 纳入哈希字段而非引入独立标识) 是值得借鉴的模式。

功能与动机

V1 和 V2 模型运行器的序列化缓存布局不兼容——V1 使用 (`head, layer, token, value`) 的跨层紧凑表示, 而 V2 使用每层独立的张量 (12 个张量, 每个形状为 (`blocks, heads, tokens, values`))。尽管两种布局的字节数相同 (均为 589824 bytes/row), 但实际数据顺序不同, 导致尺寸检查无法捕获不匹配, V2 加载 V1 缓存后会输出错误结果 (例如从 `Paris...` 变成 `,,,,, the the,,,, by a`)。该 PR 通过让非并行无关的配置文件路径不同来隔离 V1 和 V2 缓存。

实现拆解

1. 在 `FileMapper.__init__` 中增加字段记录: 当 `parallel_agnostic` 为 `False` 时, 在 `self.fields` 字典中添加 "`parallel_agnostic`": `False` (之前只有 `parallel_agnostic=True` 时才会对 `fields` 产生影响——将 `tp/pp/pcp/dcp` 置为 1 和 `rank` 置为 0)。由于 `fields` 用于计算持久化路径的哈希, 这一变更使得非并行无关的配置产生与之前不同的路径哈希, 从而自动与旧路径隔离。
2. 调整测试用例断言: 更新 `test_get_run_config_fields` 的预期输出, 使其包含 "`parallel_agnostic`": `False`。更新 `test_parallel_agnostic_collapses_namespace_when_config_allows` 断言, 确保 `parallel_agnostic=True` 且配置允许并行无关时, `fields` 中不包含 "`parallel_agnostic`" 键。更新 `test_parallel_agnostic_ignored_when_config_disallows` 和 `test_namespace_kept_without_parallel_agnostic_opt_in`, 断言 `fields["parallel_agnostic"]` 为 `False`。
3. 新增回归测试 `test_parallel_agnostic_separates_persistent_layouts`: 创建一个并行无关的 `mapper` 和一个非并行无关的 `mapper`, 验证它们的 `base_path` 不同, 且 `fields` 中并行无关的 `mapper` 不包含 "`parallel_agnostic`" 键, 而非并行无关的 `mapper` 包含 "`parallel_agnostic`": `False`。

关键文件:

- vllm/v1/kv_offload/file_mapper.py (模块 文件映射器; 类别 source; 类型 core-logic; 符号 FileMapper.init) : 核心修复文件: 在 FileMapper.__init__ 中添加了条件判断 if not parallel_agnostic: self.fields["parallel_agnostic"] = False, 使得非并行无关的配置在 fields 中包含该字段, 从而改变持久化路径的哈希值, 隔离 V1 和 V2 缓存。
- tests/v1/kv_offload/test_file_mapper.py (模块 测试; 类别 test; 类型 test-coverage; 符号 test_parallel_agnostic_separates_persistent_layouts) : 测试覆盖文件: 更新已有测试断言以匹配新行为, 并新增回归测试 test_parallel_agnostic_separates_persistent_layouts, 验证并行无关和非并行无关的 mapper 产生不同 base_path。

关键符号: FileMapper.init, test_parallel_agnostic_separates_persistent_layouts

关键源码片段

vllm/v1/kv_offload/file_mapper.py

核心修复文件: 在 FileMapper.__init__ 中添加了条件判断 if not parallel_agnostic: self.fields["parallel_agnostic"] = False, 使得非并行无关的配置在 fields 中包含该字段, 从而改变持久化路径的哈希值, 隔离 V1 和 V2 缓存。

```
# vllm/v1/kv_offload/file_mapper.py
```

```
class FileMapper:
    def __init__(
        self,
        root_dir: str,
        model_name: str,
        tokens_per_hash: int,
        blocks_per_file: int,
        tp_size: int,
        pp_size: int,
        pcp_size: int,
        dcp_size: int,
        rank: int,
        dtype: str,
        kv_cache_groups: list[dict] | None = None,
        inference_engine: str = "vllm",
        parallel_agnostic: bool = False,
    ):
        """
        Initialize the file mapper. Each worker constructs its own, but
        `config.json` is shared across workers since rank lives outside the hash.
        When `parallel_agnostic=True`, tp/pp/pcp/dcp are forced to 1 and rank
        to 0 so multiple parallelism layouts collapse into the same folder.
        """
        # -- 并行无关时归一化并行参数到 1/0, 避免路径差异 --
        if parallel_agnostic:
            tp_size = pp_size = pcp_size = dcp_size = 1
            rank = 0
        self.rank: int = rank
```

```

# 核心变更：当非并行无关时，显式记录 `parallel_agnostic: False` 到 fields 中
# 这会改变 `_compute_base_path` 的哈希结果，使 V2 的持久化路径
# 自动与 V1 的旧路径隔离（V1 使用跨层紧凑布局，V2 使用逐层张量）
self.fields: dict = {
    "model_name": model_name,
    "tokens_per_hash": tokens_per_hash,
    "blocks_per_file": blocks_per_file,
    "tp_size": tp_size,
    "pp_size": pp_size,
    "pcp_size": pcp_size,
    "dcp_size": dcp_size,
    "dtype": str(dtype),
    "kv_cache_groups": kv_cache_groups or [],
    "inference_engine": inference_engine,
}
# 新增行：只有显式声明非并行无关时，才加入该字段
# 若为并行无关，该字段不出现，保持向后兼容
if not parallel_agnostic:
    self.fields["parallel_agnostic"] = False
self.base_path: str = self._compute_base_path(root_dir, self.fields)

```

tests/v1/kv_offload/test_file_mapper.py

测试覆盖文件：更新已有测试断言以匹配新行为，并新增回归测试 `test_parallel_agnostic_separates_persistent_layouts`，验证并行无关和非并行无关的 mapper 产生不同 `base_path`。

tests/v1/kv_offload/test_file_mapper.py (新增测试)

```

def test_parallel_agnostic_separates_persistent_layouts():
    # 创建并行无关的 mapper（模拟 V1：不使用 `parallel_agnostic: False`）
    agnostic = make_mapper_from_offloading_spec(
        is_parallelism_agnostic=True,
        parallel_agnostic=True,
    )
    # 创建非并行无关的 mapper（模拟 V2：会添加 `parallel_agnostic: False`）
    specific = make_mapper_from_offloading_spec(
        is_parallelism_agnostic=False,
        parallel_agnostic=True,
    )

    # 核心断言：两种布局应具有不同的 base_path
    assert agnostic.base_path != specific.base_path
    # 并行无关的 fields 中不应包含 `parallel_agnostic` 键
    assert "parallel_agnostic" not in agnostic.fields
    # 非并行无关的 fields 中应包含 `parallel_agnostic: False`
    assert specific.fields["parallel_agnostic"] is False

```

评论区精华

Reviewer orozery 最初建议不要在 `FileMapper` 中引入 `model_runner` 字段，而是将 `parallel_agnostic` 纳入哈希字段中，具体方式为在 `fields` 中添加 `"parallel_agnostic": False`。Author jongukc 采纳该建议并修改了实现。

- 是否引入 `model_runner` 字段还是使用 `parallel_agnostic` 哈希 (design): Author jongukc 采纳建议，修改实现为当 `not parallel_agnostic` 时在 `fields` 中添加 `"parallel_agnostic": False`。

风险与影响

- 风险：低风险。核心逻辑仅增加一个条件判断（if `not parallel_agnostic`: `self.fields["parallel_agnostic"] = False`），不改变现有并行无关配置的行为。测试覆盖了主要路径和回归场景。潜在风险：任何依赖旧路径哈希的持久化缓存将在升级后失效，但这是预期行为——旧缓存本就是不安全的。
- 影响：直接影响：修复了 V1/V2 模型运行器共享持久化 KV 卸载缓存时的静默数据损坏 bug。用户无需手动操作，升级后 V2 将自动使用隔离后的缓存路径。影响范围：所有使用 V1 KR 卸载引擎且持久化缓存已包含 V1 数据的用户，但 V2 运行器在升级后不会再误加载这些数据。
- 风险标记：缓存路径变更，存量缓存失效

关联脉络

- PR #44733 [KV offload] Parallel-agnostic fs-tier cache for single full-attention group: 引入并行无关缓存共享机制，本 PR 在其基础上修复了 V1/V2 布局兼容性问题。
- PR #48906 [KV Offload] Deduplicate replicated MLA KV in the shared CPU region: 近期对 KV 卸载的修改，涉及共享 CPU 区域，与本 PR 的缓存命名空间相关。
- PR #49671 [Bugfix][KV Offloading] Defer request finalization until final store: 近期对 KV 卸载调度器的 bugfix，与本 PR 属于同一功能链。