PR #45019 完整报告

vllm-project/vllm

[NIXL][Mamba] Add Mamba1 support to NIXL P/D disaggregation

合并时间：2026-06-25 20:50

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45019

执行摘要

- 一句话：NIXL 连接器支持 Mamba1 模型 P/D 分离
- 推荐动作：值得精读。该 PR 展示了如何通过泛化数据结构而非添加分支条件来扩展已有功能，设计简洁。对于维护 kv-connector 的工程师，理解 MambaConvSplitInfo 的偏移计算逻辑对后续修改至关重要。

功能与动机

PR body 明确目标是"Add Mamba1 hybrid-model support (e.g. Jamba) to the NIXL connector's conv-state transfer for prefill/decode disaggregation"。此前 NIXL 连接器仅支持 Mamba2/GDN 的 conv-state 传输，Mamba1 模型（如 Jamba）无法利用 P/D 分离能力。

实现拆解

1. 泛化数据结构：在 ssm_conv_transfer_utils.py 中将 MambaConvSplitInfo.local_proj_dims 从 tuple[int, int, int] 改为 tuple[int, ...]，proj_bytes 对应变为 tuple[int, ...]。
2. 重构偏移计算：local_conv_offsets 和 remote_conv_offsets 从硬编码三段解包改为基于 proj_bytes 的循环累加，支持任意子投影数。
3. 调整描述符 ID 计算：在 base_worker.py 的 _compute_desc_ids 中，将 num_ssm_regions 从固定值 len(self.block_len_per_layer) * 4 改为根据 len(self._conv_decomp.local_conv_offsets) + 1 动态计算。
4. 添加 Mamba1 测试：在单元测试 test_nixl_connector_hma.py 中增加 jamba_mini_tp1/4/8 参数化用例；在集成测试配置 config_sweep_accuracy_test.sh 中添加 ai21labs/AI21-Jamba2-3B 运行项；在 test_accuracy.py 中添加精度基线。
5. 回归验证：用 Qwen3.5-0.8B（GDN Mamba2）在 1P4D 和 4P1D 下验证，精度与预期一致（0.33±0.05），确保 Mamba2/GDN 不受影响。

关键文件：

- vllm/distributed/kv_transfer/kv_connector/v1/ssm_conv_transfer_utils.py（模块 SSM 传输工具；类别 source；类型 core-logic；符号 MambaConvSplitInfo, local_proj_dims, proj_bytes, local_conv_offsets）：核心变更文件，泛化 Mamba1 卷积状态子投影数据结构

- vllm/distributed/kv_transfer/kv_connector/v1/nixl/base_worker.py（模块 NIXL 工作器；类别 source；类型 core-logic；符号 _compute_desc_ids, _build_mamba_local, _build_mamba_remote）：调整描述符 ID 计算，动态计算 SSM 区域数
- tests/v1/kv_connector/unit/test_nixl_connector_hma.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_derive_mamba_conv_split, _make_mock_worker_for_desc_ids）：添加 Mamba1 参数化测试，验证卷积分裂
- tests/v1/kv_connector/nixl_integration/config_sweep_accuracy_test.sh（模块 集成测试；类别 test；类型 test-coverage）：集成测试新增 Mamba1 配置
- tests/v1/kv_connector/nixl_integration/test_accuracy.py（模块 精度测试；类别 test；类型 test-coverage）：添加 Jamba2-3B 精度基线

关键符号：MambaConvSplitInfo.init, MambaConvSplitInfo.proj_bytes, MambaConvSplitInfo.local_conv_offsets, MambaConvSplitInfo.remote_conv_offsets, NixlBaseConnectorWorker._compute_desc_ids, test_derive_mamba_conv_split

关键源码片段

vllm/distributed/kv_transfer/kv_connector/v1/ssm_conv_transfer_utils.py

核心变更文件，泛化 Mamba1 卷积状态子投影数据结构

```
# SPDX-License-Identifier: Apache-2.0
# SPDX-FileCopyrightText: Copyright contributors to the vLLM project
"""Mamba conv-state 子投影分解工具，用于 NIXL 传输。
   支持 Mamba1（单子投影）、Mamba2（x/B/C）和 GDN（Q/K/V）。
"""

import math
from dataclasses import dataclass
import torch
from vllm.model_executor.layers.mamba.mamba_utils import is_conv_state_dim_first
from vllm.v1.attention.backends.registry import MambaAttentionBackendEnum
from vllm.v1.kv_cache_interface import MambaSpec


@dataclass(frozen=True)
class MambaConvSplitInfo:
    """每个 rank 的卷积状态子投影字节大小。
       使用 DS 连续布局（dim, state_len），子投影在内存中连续。

       内存布局在一个 page 内：
         Mamba1: |---- x ----|  (单个子投影，无分解)
         Mamba2: |-- x --|- B -|- C -|  (B == C)
         GDN:    |- Q -|- K -|-- V --|  (dim(Q)==dim(K), V 可能不同)
    """
    conv_rows: int # conv_kernel - 1（通常为 3）
    # 每个 rank 各子投影的列数，
    # Mamba1 为 1 个值，Mamba2/GDN 为 3 个值
```

```
local_proj_dims: tuple[int, ...]
conv_dtype_size: int # 每元素字节数（如 float16 为 2）
ssm_sizes: tuple[int, int] # (conv_state_bytes, ssm_state_bytes)

@property
def local_conv_dim(self) -> int:
    """当前 rank 的总卷积列数"""
    return sum(self.local_proj_dims)

@property
def proj_bytes(self) -> tuple[int, ...]:
    """返回各子投影的字节大小（本地 rank）"""
    row_bytes = self.conv_rows * self.conv_dtype_size
    return tuple(d * row_bytes for d in self.local_proj_dims)

@property
def local_conv_offsets(self) -> list[tuple[int, int]]:
    """返回 (byte_offset, byte_size) 列表，供本地描述符注册"""
    offsets: list[tuple[int, int]] = []
    offset = 0
    for size in self.proj_bytes:
        offsets.append((offset, size))
        offset += size
    return offsets

def remote_conv_offsets(
    self, local_rank_offset: int, tp_ratio: int
) -> list[tuple[int, int]]:
    """返回本 D rank 在 P page 内的子投影切片 (byte_offset, byte_size)。
      tp_ratio 标识 P/D 的 tensor 并行大小关系。
    """
    offsets: list[tuple[int, int]] = []
    if tp_ratio >= 1:
        # D_TP >= P_TP；P page 更大，D 读取其切片
        remote_base = 0
        for size in self.proj_bytes:
            offsets.append((remote_base + local_rank_offset * size, size))
            remote_base += size * tp_ratio
    else:
        # P_TP > D_TP；P page 更小，D 需要按比例缩放
        abs_ratio = -tp_ratio
        remote_base = 0
        for size in self.proj_bytes:
            remote_size = size // abs_ratio
            offsets.append((remote_base, remote_size))
            remote_base += remote_size
    return offsets
```

评论区精华

- 模型选择：NickLucche 建议使用更小的 Mamba1 模型进行 CI 测试，从 ai21labs/AI21-Jamba2-Mini（52B）改为 ai21labs/AI21-Jamba2-3B。
- 内联建议：ZhanqiuHu 建议将 _ssm_regions_per_layer 属性内联到 _compute_desc_ids 中，作者接受并实施。
- 回归验证：ZhanqiuHu 要求检查 Mamba2/GDN 是否仍工作，作者用 Qwen3.5-0.8B 验证 1P4D 和 4P1D，结果在预期范围内。
  - 选择更小的 Mamba1 测试模型 (design): 使用 ai21labs/AI21-Jamba2-3B 作为 CI 测试模型。
  - 内联 _ssm_regions_per_layer (design): 作者采纳，最终版本中直接计算 ssm_regions_per_layer 局部变量。
  - 回归验证 Mamba2/GDN (correctness): 回归通过。

风险与影响

- 风险：核心数据结构从固定三投影改为可变长度，若 local_proj_dims 长度不符合预期（如 Mamba2/GDN 意外传入 1 个维度），可能导致偏移计算错误。但测试覆盖了三种类型且回归验证通过。性能影响可忽略，循环只多走 1-3 次。兼容性要求设置 VLLM_SSM_CONV_STATE_LAYOUT=DS 环境变量，未设置时新功能不会启用。
- 影响：用户侧：支持 Mamba1 混合模型（如 Jamba）的 P/D 分离，降低显存占用。系统侧：改动集中在两个核心源文件，影响范围有限。团队侧：后续添加更多 SSM 变体（如 Mamba3）时，泛化结构可直接复用。
- 风险标记：核心路径变更，环境变量依赖

关联脉络

- PR #46394 [SimpleCPUOffloadConnector] Fix remaining global→block conversions under PCP/DCP: 同为 kv-connector 模块的修复 PR，与本 PR 属于同一功能域。