PR #38373 完整报告

vllm-project/vllm

[torch.compile]: Disable Sequence Parallelism (SP) for piecewise compilation

合并时间：2026-04-27 01:44

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/38373

执行摘要

- 一句话：禁用 piecewise 编译时的 Sequence Parallelism，仅保留 full-graph 支持
- 推荐动作：建议所有使用 vLLM 中 torch.compile 与 SP 的开发者和研究员阅读此 PR 的讨论，特别是关于配置降级策略和 pass 断言的设计，了解为何 piecewise 编译下的 SP 不被支持。对于希望开启 SP 的用户，文档应明确告知需要启用 inductor 分区或清空 splitting_ops。

功能与动机

关联的 RFC Issue #35771 指出：当使用 piecewise 编译（Dynamo 分区且 splitting_ops 非空）时，RMSNorm 残差张量会在子图间传递，而 SP 会沿 num_tokens 维度分割张量，导致残差大小在不同 TP rank 间不一致。已有的切片处理方式与 prompt_embeds 等多模态输入不兼容，且存在不安全的假设。因此提议仅在全图编译（Inductor 分区或空 splitting_ops）时支持 SP，以简化切片逻辑并提升正确性。

实现拆解

1. 配置冲突自动降级：在 vllm/config/compilation.py 的 set_splitting_ops_for_v1 方法中，当 enable_sp 为 True 且 use_inductor_graph_partition 为 False 且 splitting_ops 非空时，强制将 splitting_ops 清空并降级 cudagraph_mode 到 FULL，同时输出警告。
2. 主要配置文件调整：在 vllm/config/vllm.py 中，移除原有的 snapshot/reconcile 机制（_snapshot_user_compilation_inputs / _reconcile_sequence_parallelism_for_cudagraph_mode），改为在 set_splitting_ops_for_v1 之后直接调用 _finalize_sequence_parallelism_config，避免二次计算。同时调整 SP 初始化顺序，使用本地变量 pass_config 简化重复引用。
3. Pass 层强制执行：在 sequence_parallelism.py 的 SequenceParallelismPass.is_applicable_for_range 和 collective_fusion.py 的 AsyncTPPass.is_applicable_for_range 中，移除对 piecewise 模式的兼容逻辑，代之以明确的 assert，要求必须为 full-graph 模式。
4. 运行时函数简化：在 vllm/v1/worker/utils.py 的 is_residual_scattered_for_sp 中，移除对 compile_sizes 的查询，直接断言 SP 要求 full-graph 编译，并简化返回逻辑。
5. 测试配套：新增三个测试函数：test_sequence_parallelism_requires_full_graph_compilation（配置降级验证）、test_sequence_parallelism_pass_requires_full_graph_compilati

on（pass 断言触发验证）、test_async_tp_pass_requires_full_graph_compilation（异步 TP pass 断言验证）。

关键文件：

- vllm/config/vllm.py（模块 配置层；类别 source；类型 core-logic）：核心配置文件，调整了 SP 初始化顺序、移除了 snapshot/reconcile 机制、简化了条件判断。是功能行为变更的主要入口。
- vllm/config/compilation.py（模块 配置层；类别 source；类型 core-logic）：新增 SP 与 piecewise 编译冲突的自动降级逻辑，是核心行为变更之一。
- vllm/compilation/passes/fusion/sequence_parallelism.py（模块 编译层；类别 source；类型 core-logic）：SP pass 关键文件，is_applicable_for_range 方法被重构，移除 piecewise 支持并添加断言。
- vllm/compilation/passes/fusion/collective_fusion.py（模块 编译层；类别 source；类型 core-logic）：AsyncTPPass 也依赖 SP，添加了类似的全图断言。
- vllm/v1/worker/utils.py（模块 Worker；类别 source；类型 core-logic）：运行时函数 is_residual_scattered_for_sp 被简化，移除对 compile_sizes 的依赖，添加全图断言。
- tests/compile/test_config.py（模块 测试；类别 test；类型 test-coverage；符号 test_sequence_parallelism_requires_full_graph_compilation）：新增测试验证配置降级行为，确保 SP 与 piecewise 冲突时正确处理。
- tests/compile/passes/distributed/test_sequence_parallelism.py（模块 测试；类别 test；类型 test-coverage；符号 test_sequence_parallelism_pass_requires_full_graph_compilation）：新增测试直接验证 SequenceParallelismPass 在非全图时是否会触发断言。
- tests/compile/passes/distributed/test_async_tp.py（模块 测试；类别 test；类型 test-coverage；符号 test_async_tp_pass_requires_full_graph_compilation）：新增测试验证 AsyncTPPass 在非全图时是否会触发断言，与 SP pass 类似。
- tests/compile/correctness_e2e/test_sequence_parallel.py（模块 测试；类别 test；类型 test-coverage）：端到端测试，仅做了微小的适应性改动（增加对 full-graph 模式的配置）。

关键符号：set_splitting_ops_for_v1, SequenceParallelismPass.is_applicable_for_range, AsyncTPPass.is_applicable_for_range, is_residual_scattered_for_sp, test_sequence_parallelism_requires_full_graph_compilation, test_sequence_parallelism_pass_requires_full_graph_compilation, test_async_tp_pass_requires_full_graph_compilation

关键源码片段

vllm/config/vllm.py

核心配置文件，调整了 SP 初始化顺序、移除了 snapshot/reconcile 机制、简化了条件判断。是功能行为变更的主要入口。

```
# vllm/config/vllm.py 中 SP 初始化部分（调整后）

# ... 省略前后文 ...
# async tp 建立在 seq parallelism 之上，需要它先启用
```

```
    pass_config = self.compilation_config.pass_config
    if pass_config.fuse_gemm_comms:
        pass_config.enable_sp = True

    if pass_config.enable_sp:
        if self.parallel_config.tensor_parallel_size == 1:
            logger.warning("Sequence Parallelism requires TP>1, disabling")
            pass_config.enable_sp = False
            pass_config.fuse_gemm_comms = False
        else:
            # 若未设置 min_token_num 阈值，则自动计算
            if pass_config.sp_min_token_num is None:
                from vllm.compilation.passes.fusion.sequence_parallelism import (
                    get_sequence_parallelism_threshold,
                )
                tp_size = self.parallel_config.tensor_parallel_size
                hidden_size = self.model_config.get_hidden_size()
                element_size = self.model_config.dtype.itemsize
                pass_config.sp_min_token_num = get_sequence_parallelism_threshold(
                    hidden_size, tp_size, element_size
                )

            if pass_config.sp_min_token_num is None:
                logger.warning(
                    "Model hidden_size too small for the SP threshold heuristic, "
                    "disabling. To force SP, set pass_config.sp_min_token_num manually."
                )
                pass_config.enable_sp = False
                pass_config.fuse_gemm_comms = False
    # 随后在 set_splitting_ops_for_v1 之后调用 _finalize_sequence_parallelism_config
    # 该函数会再次检查冲突并可能强制 full-graph 编译
    # ... 省略后文 ...
```

vllm/config/compilation.py

新增 SP 与 piecewise 编译冲突的自动降级逻辑，是核心行为变更之一。

```
    # vllm/config/compilation.py 中 set_splitting_ops_for_v1 方法片段

    # ... 原有逻辑 ...
    # 当启用 SP 且未使用 Inductor 分区时，piecewise 编译与 SP 不兼容
    if (
        not self.use_inductor_graph_partition
        and (self.pass_config.enable_sp or self.pass_config.fuse_gemm_comms)
        and self.splitting_ops
    ):
        logger.warning_once(
            "Sequence parallelism requires full-graph compilation when "
            "use_inductor_graph_partition is off. Setting splitting_ops "
            "to an empty list to preserve SP and async TP."
```

```
    )
    self.splitting_ops = [] # 强制全图编译
    if self.cudagraph_mode.has_piecewise_cudagraphs():
        logger.warning_once(
            "Sequence parallelism is incompatible with piecewise "
            "cudagraph when use_inductor_graph_partition is off. "
            "Setting cudagraph_mode to FULL."
        )
        self.cudagraph_mode = CUDAGraphMode.FULL # 降级 CUDA graph 模式
# ... 后续逻辑 ...
```

评论区精华

Review 中 reviewer（wangxingran222 和 ProExpertProg）提出了以下核心意见：

- 保留 pass 断言：尽管配置层已强制降级，但仍建议在 SequenceParallelismPass 和 AsyncTPPass 的 is_applicable_for_range 中添加显式断言，以确保即使通过非标准路径实例化也能保持一致性。
- 配置冲突处理位置：wangxingran222 建议将冲突处理直接放在 set_splitting_ops_for_v1 内部，而非通过 snapshot/reconcile，这样更简洁且符合现有模式（如 fuse_attn_quant）。作者采纳并重写了这部分逻辑。
- 测试覆盖：ProExpertProg 指出需要更新 SP 相关的单元测试，包括 pass 测试和 e2e 测试，并且不能直接传入 MagicMock 参数。最终测试调整为使用默认 VllmConfig 或 object.__new__ 方式构造 pass 实例。
- CUDAGraphMode 与 torch.compile 正交性：wangxingran222 指出不应在 SP 条件中依赖 CUDAGraphMode，因为 piecewise 编译可以独立于 cudagraph 模式启用。作者修正了该处逻辑。
- E2E 验证：ProExpertProg 要求提供 lm_eval 结果以确认端到端正确性，作者附上了 Qwen3-14B 在 gsm8k 上的评测结果（准确率约 0.78），表明功能正常。
  - 是否需要在 pass 中保留显式断言 (correctness): 作者同意并添加了 assert。
  - 配置冲突处理位置选择 (design): 作者重构为在 set_splitting_ops_for_v1 中直接处理冲突，并移除 snapshot/reconcile。
  - 测试覆盖度的讨论 (testing): 作者调整了测试实现，使用 default_vllm_config 或 object.__new__构造，并更新了 e2e 测试的配置。
  - CUDAGraphMode 与 torch.compile 的正交性 (correctness): 作者去除对 CUDAGraphMode 的依赖，仅基于 use_inductor_graph_partition 和 splitting_ops 判断。

风险与影响

- 风险：
  1. 配置静默覆盖风险：如果用户显式设置了 splitting_ops 并启用 SP，系统会自动清空 splitting_ops 并降级 cudagraph 模式。虽然日志中包含 warning，但用户可能未注意，导致与预期编译行为不一致（例如编译时间增加或 CUDA graph 行为改变）。

2. 断言失败风险：新增的 assert 在非全图编译且 SP 仍启用的场景下会直接触发异常。虽然配置层已保证不会发生，但若存在绕过配置层的代码路径（如直接构造 Pass 实例），可能引发程序崩溃。
3. 性能退化风险：强制全图编译可能增大编译时间或峰值内存，对首次启动延迟敏感的应用有不利影响。
4. 兼容性风险：此前依赖 piecewise+SP 的用户（如有特殊编译需求）将受到影响，可能找不到替代方案。 - 影响：影响范围：使用 torch.compile 并启用 Sequence Parallelism（enable_sp=True）的用户，尤其是显式设置 splitting_ops 或未启用 inductor 分区的用户。 用户层面：这些用户将不再获得 piecewise 编译下的 SP 优化（该优化本身可能不安全），而是被静默切换为 full-graph 编译。对于模型较小或 token 数较少的场景，SP 本身也可能因达不到阈值而被禁用。 系统层面：配置初始化流程被简化，移除了 snapshot/reconcile 二次计算逻辑，有利于维护。 团队层面：消除了一个已知的正确性 issue，降低了后续多模态模型引入时的兼容性风险。

- 风险标记：配置静默覆盖，断言可能触发，性能退化（编译时间），缺少用户通知文档

关联脉络

- PR #35771 [RFC][torch.compile]: Disable Sequence Parallelism (SP) for piecewise compilation: 该 PR 是此 RFC 的具体实现，直接关联作为设计文档和动机来源。
- PR #27126 related to native rms_norm support for SP piecewise: PR #27126 引入了不安全的切片方式以支持 native rms_norm，本 PR 移除了对该方式的依赖，是关联的延续。
- PR #33322 related to prompt_embeds fix breaking piecewise SP: PR #33322 修复了 prompt_embeds 多模态输入问题，但暴露了 piecewise SP 的兼容性问题，本 PR 从根本上解决该问题。