

PR #35340 完整报告

sgl-project/sglang

[AMD][bugfix] Add moe_ep_size/moe_tp_size to the allreduce-fusion gate test stub

合并时间: 2026-08-25 16:24

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35340>

执行摘要

- 一句话: 修复 AMD allreduce 融合测试桩缺失 moe 并行字段
- 推荐动作: 值得快速浏览。该 PR 展示了当生产代码新增字段读取时, 测试桩需要同步更新的典型场景, 提醒关注测试环境与生产环境的并行上下文一致性。

功能与动机

PR body 指出: #30700 在 `should_fuse_mlp_allreduce_with_next_layer` 顶部读取 `get_parallel().moe_ep_size` 和 `get_parallel().moe_tp_size`, 而 `TestAiterAllreduceFusionGate` 的测试桩仅包含 `tp_size`, 导致测试在评估 Aiter 分支前抛出 `AttributeError`。

实现拆解

这是一个纯测试修复, 步骤如下: 1. 识别根因: 在 `test/registered/ops/test_aiter_allreduce_fusion_amd.py` 的 `_evaluate_gate` 辅助方法中, mock 的 `get_parallel` 返回 `SimpleNamespace(tp_size=tp_world_size)`, 缺少 #30700 新增读取的 `moe_ep_size` 和 `moe_tp_size` 字段, 触发 `AttributeError`。2. 修正测试桩: 在 `get_parallel` 的 `SimpleNamespace` 中增加 `moe_ep_size=1` 和 `moe_tp_size=1`, 使混合 EP+TP 防护条件保持不激活, 从而隔离 Aiter 分支的测试意图。3. 验证: 运行 ROCm 7.2 目标分片, `test_aiter_allreduce_fusion_amd.py` 的 10 个测试全部通过, lint 通过。

关键文件:

- `test/registered/ops/test_aiter_allreduce_fusion_amd.py` (模块 AMD 模块; 类别 test; 类型 test-coverage; 符号 `_evaluate_gate`): 唯一修改文件, 为 allreduce 融合门控测试的 `get_parallel` 桩补充 `moe_ep_size` 和 `moe_tp_size` 字段, 修复因 #30700 引入的 `AttributeError`。

关键符号: `_evaluate_gate`, `should_fuse_mlp_allreduce_with_next_layer`

关键源码片段

`test/registered/ops/test_aiter_allreduce_fusion_amd.py`

唯一修改文件, 为 allreduce 融合门控测试的 `get_parallel` 桩补充 `moe_ep_size` 和 `moe_tp_size` 字段, 修复因 #30700 引入的 `AttributeError`。

```

# test/registered/ops/test_aiter_allreduce_fusion_amd.py
def _evaluate_gate(self, *, dp_attention, a2a_is_none, aiter_enabled=True,
                    use_aiter=True, tp_world_size=8,
                    mlp_mode=ScatterMode.TP_ATTN_FULL, is_last_layer=False,
                    tp_size=8):
    """Run the gate with the aiter branch isolated (flashinfer forced off)."""
    a2a_backend = types.SimpleNamespace(is_none=lambda: a2a_is_none)

    with ExitStack() as stack:
        # 关闭 MoE CP allgather 和 flashinfer 融合，只保留 Aiter 分支
        stack.enter_context(mock.patch.object(comm, "is_enable_moe_cp_allgather", lambda:
        False))
        stack.enter_context(mock.patch.object(comm, "get_attn_tp_context",
        lambda: types.SimpleNamespace(input_scattered=False)))
        stack.enter_context(mock.patch.object(comm, "apply_flashinfer_allreduce_fusion", lambda
        batch_size: False))
        stack.enter_context(mock.patch.object(comm, "_use_aiter", use_aiter))
        # moe_ep_size/moe_tp_size 用 1，保证混合 EP+TP 防护不触发，
        # 从而让 Aiter 分支成为唯一决策者
        stack.enter_context(mock.patch.object(comm, "get_parallel",
        lambda: types.SimpleNamespace(
            tp_size=tp_world_size,
            moe_ep_size=1,
            moe_tp_size=1)))
        # 覆盖运行时配置以启用 Aiter 融合
        stack.enter_context(get_context().override_server_args(
            enable_aiter_allreduce_fusion=aiter_enabled))
        stack.enter_context(get_flags().dp.override(enabled=dp_attention))
        stack.enter_context(mock.patch.object(comm, "get_moe_a2a_backend", lambda: a2a_
        backend))

    fake_self = _fake_self(mlp_mode=mlp_mode, is_last_layer=is_last_layer, tp_size=tp_size)
    return LayerCommunicator.should_fuse_mlp_allreduce_with_next_layer(
        fake_self, _fake_forward_batch())

```

评论区精华

无 review 评论。CI 评论中作者确认了目标分片成功，且 shard 3 是此前 main 分支持续失败的 shard。

- 目标分片测试结果 (testing): 目标分片通过，10 个测试全部成功。

风险与影响

- 风险：风险极低。仅修改测试桩，不涉及生产代码，不影响内核或模型前向逻辑。需注意测试桩的字段补齐是否与未来并行上下文扩展保持同步。
- 影响：影响仅限于 AMD allreduce 融合的单元测试稳定性，确保 CI 中相关测试不再因测试桩过时而失败。对用户和系统运行无影响。

- 风险标记：测试桩同步风险

关联脉络

- PR #30700 Add hybrid EP+TP guard to `should_fuse_mlp_allreduce_with_next_layer`:
本 PR 的根因来自该 PR 新增的 `moe_ep_size/moe_tp_size` 读取，测试桩因此需要同步更新。