

PR #29623 完整报告

sgl-project/sglang

fix test_weight_checker_comparator assertion and ue8m0 scale unpack

合并时间: 2026-06-29 14:36

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29623>

执行摘要

- 一句话: 修复 weight checker comparator 中 ue8m0 填充块截断和测试断言
- 推荐动作: 此 PR 是常规 bugfix, 变更小且目标明确, 建议合入。可关注 ue8m0 截断逻辑在其他使用场景中是否一致。

功能与动机

PR 标题和提交信息表明, 主要动机是修复 weight checker comparator 中的两个 bug: ue8m0 打包的 scale 在反变换后未截断填充块, 导致 chunked 比较时 block 数量与真实值不匹配; 以及 chunked 与 unchunked 的 mean_err 因浮点运算顺序不同而产生微小差异, 导致断言失败。

实现拆解

1. 截断 ue8m0 打包 scale 的填充块: 在 python/sglang/srt/utils/weight_checker_comparator.py 中, Fp8BlockComparable._normalize_scale 方法对 int32 类型的 scale 调用 inverse_transform_scale_ue8m0 后, 新增 `w_s = w_s[..., : -(w_q.shape[-1] // 128)]` 用于截断 ue8m0 打包时因 k 对齐到 4 的倍数而引入的填充块。
2. 松弛 chunked 比较的测试断言: 在测试文件 test/registered/unit/utils/test_weight_checker_comparator.py 中, 将 test_chunked_result_matches_unchunked 方法中的 `self.assertEqual(chunked, reference)` 替换为分别比较各字段, 其中 mean_err 使用 `assertAlmostEqual` 并指定 `places=7`, 容忍浮点运算顺序带来的微小差异。

关键文件:

- python/sglang/srt/utils/weight_checker_comparator.py (模块 权重检查器; 类别 source; 类型 core-logic): 核心修复: 在 ue8m0 scale 反变换后截断填充块, 确保 block 数量与真实 k 维度匹配。
- test/registered/unit/utils/test_weight_checker_comparator.py (模块 测试; 类别 test; 类型 test-coverage): 测试修复: 将 chunked 与 unchunked 的严格相等断言改为逐字段比较, 容忍浮点运算顺序导致的微小差异。

关键符号: 未识别

关键源码片段

python/sglang/srt/utils/weight_checker_comparator.py

核心修复：在 ue8m0 scale 反变换后截断填充块，确保 block 数量与真实 k 维度匹配。

```
# python/sglang/srt/utils/weight_checker_comparator.py
# Fp8BlockComparable._normalize_scale 中新增填充截断
@staticmethod
def _normalize_scale(w_q: torch.Tensor, w_s: torch.Tensor) -> torch.Tensor:
    if w_s.dtype == torch.int32:
        # 反变换：将 ue8m0 打包的 int32 scale 解码为 float32
        w_s = inverse_transform_scale_ue8m0(w_s, mn=w_q.shape[-2])
        # ue8m0 打包时会把 k 对齐到 4 的倍数（128 的倍数），
        # 此处截断最后的填充块，使得 s_k = ceil(k / 128)
        w_s = w_s[..., : -(w_q.shape[-1] // 128)]
    return w_s.to(torch.float32)
```

test/registered/unit/utils/test_weight_checker_comparator.py

测试修复：将 chunked 与 unchunked 的严格相等断言改为逐字段比较，容忍浮点运算顺序导致的微小差异。

```
# test/registered/unit/utils/test_weight_checker_comparator.py
# test_chunked_result_matches_unchunked 中断言修复
def test_chunked_result_matches_unchunked(self):
    reference = _compare_quant_pair(self.e_q, self.e_s, self.a_q, self.a_s)
    with patch("sglang.srt.utils.weight_checker_comparator.CHUNK_NUMEL", 128 * 128):
        chunked = _compare_quant_pair(self.e_q, self.e_s, self.a_q, self.a_s)
    # 分解元组，分别比较各字段
    eq_c, max_c, mean_c, ex_c = chunked
    eq_r, max_r, mean_r, ex_r = reference
    # equal、max_err、num_exceed 仍要求严格相等
    self.assertEqual((eq_c, max_c, ex_c), (eq_r, max_r, ex_r))
    # mean_err 因浮点运算顺序不同可能有微小差异，使用 assertAlmostEqual
    self.assertAlmostEqual(mean_c, mean_r, places=7)
```

评论区精华

该 PR 的 review 评论为空，issue 评论主要是作者触发 CI 重跑测试的指令，未记录设计讨论。

- 暂无高价值评论线程

风险与影响

- 风险：变更集中在 weight checker comparator 模块，该模块主要用于测试和调试场景，不涉及模型推理主路径。补丁较小（源码 +2/-0，测试 +4/-1），风险较低。但 ue8m0 填充截断的索引计算若出错，可能导致 scale 截取错误位置，需要在测试中覆盖非 128 整数倍的 k 维度。
- 影响：影响范围限于 weight checker 流，仅影响 FP8 量化权重比较的正确性。对用户透明，但可提高 FP8 模型权重检查的准确性。测试修复后，CI 中 test_weight_checker_comparator 测试可通过。

- 风险标记：逻辑变更缺少测试覆盖（ue8m0 截断边界条件）

关联脉络

- PR #28974 [weight checker] refactor: add precision branch; allow ULP quant err; used chunked compare: 与该 PR 直接相关，是 weight checker comparator 模块的首次引入，该 PR 修复了引入后的两个问题。