

PR #29329 完整报告

sgl-project/sglang

[CI] Fix false spec accept length failure after profiling phase

合并时间: 2026-06-26 13:19

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29329>

执行摘要

- 一句话: 修复 CI 后 profiling 阶段误报 spec accept length
- 推荐动作: 值得精读: 展示了 CI 框架中一个典型的竞态 / 状态残留 bug, 修复优雅且低风险。

功能与动机

nightly speculative-decoding perf tests (如 `test_kimi_k25_nvfp4_eagle.py`) 会误报 'Spec accept length threshold set but no accept length reported', 即使 accept length 健康且远超阈值。根本原因是 profiling 阶段的 `flush_cache` 重置了 spec 计数器, 且 `profile_steps=5` 小于 `decode_log_interval=40`, 导致 `/server_info` 缺少 `avg_spec_accept_length`。

实现拆解

1. 识别缺失场景: 在 `run_performance_test` 中, 当 `avg_spec_accept_length` 为 `None` 时, 判断为 profiling 阶段后的计数器重置场景。
2. 回退到 benchmark 结果: 遍历 `results` 列表, 收集各 `BenchmarkResult.acc_length` (仅收集非 `None` 且 > 0 的值)。
3. 计算均值: 若有有效值, 计算算术平均值赋值给 `avg_spec_accept_length`。
4. 保持原逻辑: 若仍为 `None` 或低于阈值, 则继续报错; 否则正常通过。仅修改 `python/sglang/test/performance_test_runner.py` 文件, 新增约 14 行 fallback 逻辑。

关键文件:

- `python/sglang/test/performance_test_runner.py` (模块 性能测试; 类别 test; 类型 test-coverage): 唯一修改的文件, 在 `run_performance_test` 函数中新增回退逻辑, 从 `BenchmarkResult` 列表计算 `avg_spec_accept_length`。

关键符号: 未识别

关键源码片段

`python/sglang/test/performance_test_runner.py`

唯一修改的文件, 在 `run_performance_test` 函数中新增回退逻辑, 从 `BenchmarkResult` 列表计算 `avg_spec_accept_length`。

```
# 位于 run_performance_test 函数中, 在获得 results 和 avg_spec_accept_length 后
```

```

if success and results:
    perf_runner.add_report(results, variant=model.variant)
    print(f"✓ Performance test succeeded for {model.model_path}")

# /server_info 中的累计 accept length 会被 profiling 阶段的
# cache flush 重置，因此可能缺失。这里回退到 benchmark 阶段
# 已捕获的各 run 的 accept length 均值。
if avg_spec_accept_length is None:
    # 收集所有非 None 且大于 0 的 run accept length
    run_accept_lengths = [
        r.acc_length
        for r in results
        if r.acc_length is not None and r.acc_length > 0
    ]
    if run_accept_lengths:
        avg_spec_accept_length = sum(run_accept_lengths) / len(
            run_accept_lengths
        )

# 后续阈值验证逻辑不变
error_msg = None
passed = True
if spec_accept_length_threshold is not None:
    if avg_spec_accept_length is None:
        error_msg = "Spec accept length threshold set but no accept length reported"
        passed = False
        print(f"X {error_msg}")
    elif avg_spec_accept_length < spec_accept_length_threshold:
        error_msg = (
            f"Spec accept length {avg_spec_accept_length:.2f} < "
            f"threshold {spec_accept_length_threshold}"
        )
        passed = False
        print(f"X {error_msg}")
    else:
        print(
            f"✓ Spec accept length {avg_spec_accept_length:.2f} >= "
            f"threshold {spec_accept_length_threshold}"
        )

```

评论区精华

无 review 讨论，因为 PR 由作者自行批准合并，且评论仅涉及 CI 触发和状态确认。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：回退逻辑仅在所有 benchmark run 都未报 accept length 时才会失败（和原行为一致）；若某个 run 的 acc_length 为 None 但其他 run 有值，均值可能略不同于 server 累计值，但阈值验证仍为保守近似。
- 影响：影响范围仅限于 nightly CI 中的 speculative-decoding 性能测试；对生产代码无影响，不会改变用户可见行为。修复后，之前因此问题失败的 job 将正确通过。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR