PR #34146 完整报告

sgl-project/sglang

[CI] Pin the rust frontend parity test to eager prefill

合并时间：2026-08-09 11:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34146

执行摘要

- 一句话：rust 前端 parity 测试固定 *eager* prefill，修复 CI 偶发失败
- 推荐动作：值得快速阅读。虽然改动只有 8 行，但 PR body 对根因的推导过程质量很高：通过对比 #33352 自身 CI 与计划任务运行的两端 #running-req 状态，用表格证明了失败是时序依赖而非确定性 bug。它展示了 CI 稳定性维护中一个关键原则：bit-identical 类断言必须控制所有影响数值的变量，测试只应覆盖自己声称覆盖的东西。

功能与动机

CI 计划任务运行中 base-b-test-1-gpu-small 的 test_logprobs_have_zero_kl_against_python_frontend 失败，断言 rust_logprobs["token_logprobs"] 与 python_logprobs["token_logprobs"] 完全相等。根因是 #33352 移除 auto-prefill-graph 内存门槛后，该模型 launch 开始捕获 prefill CUDA graph（日志显示 avail mem=1.83 GB），graph 将 token 数 pad 到 bucket，数值取决于请求共享的 forward pass；失败时 python 服务器处于 #running-req: 0、rust 服务器仍带着 warmup 请求解码处于 #running-req: 1，属时序依赖的非确定失败（#33352 自身 CI 恰好两端都是 0 才通过）。作者认为该测试检查的是 rust 前端是否驱动与 python 前端相同的推理路径，prefill 是否跑在 CUDA graph 下不是测试目标，因此固定 *eager* 以消除 batch 组合变量。

实现拆解

1. 根因定位：作者通过对比 #33352 自身 CI 与计划任务运行中 python/rust 两端 #running-req 状态的差异，用表格证明失败是 batch 组合不同导致的时序依赖问题，而非确定性 bug。
2. 修改测试：在 test/registered/openai_server/basic/test_openai_completion_rust.py 的 _get_logprobs 中，为 python 与 rust 两个 server 的 other_args 统一追加 --disable-prefill-cuda-graph，并新增注释说明 prefill CUDA graph 会 padding batch、影响数值确定性的原因。
3. 验证：通过 /rerun-test 在 1-gpu-5090 上重跑 test_openai_completion_rust.py，测试通过，确认修复有效。

无配置、schema 或部署配套改动；该 PR 只涉及单一测试文件。

关键文件：

- test/registered/openai_server/basic/test_openai_completion_rust.py（模块 前端测试；类别 test；类型 test-coverage；符号 _get_logprobs）：唯一变更文件，通过为两个 server 统一追加 --disable-prefill-cuda-graph 固定 eager prefill，消除 prefill CUDA graph 的 batch padding 变量，修复 bit-identical parity 断言在 python/rust 前端 batch 状态不同时的偶发失败。

关键符号：_get_logprobs

关键源码片段

test/registered/openai_server/basic/test_openai_completion_rust.py

唯一变更文件，通过为两个 server 统一追加 --disable-prefill-cuda-graph 固定 eager prefill，消除 prefill CUDA graph 的 batch padding 变量，修复 bit-identical parity 断言在 python/rust 前端 batch 状态不同时的偶发失败。

```
def _get_logprobs(self, *, rust_frontend):
    # 背景：prefill CUDA graph 会把 batch padding 到 bucket 大小，
    # 数值结果因此取决于哪些请求共享同一个 forward pass；本测试
    # 断言两个前端输出 bit-identical，所以两个 server 都必须固定
    # eager prefill，排除 batch 组合变量（回归来源见 PR#33352）。
    process = popen_launch_server(
        self.model,
        DEFAULT_URL_FOR_TEST,
        timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
        api_key=self.api_key,
        env={"SGLANG_RUST_SERVER": "1" if rust_frontend else "0"},
        other_args=[
            "--random-seed",
            "42",
            "--disable-prefill-cuda-graph",
        ],
    )
    try:
        # 以固定 prompt 与 temperature=0 发起 completion 请求，
        # 返回 top-5 logprobs 供两端做 bit-identical 对比。
        response = requests.post(
            DEFAULT_URL_FOR_TEST + "/v1/completions",
            headers={"Authorization": f"Bearer {self.api_key}"},
            json={
                "model": self.model,
                "prompt": "The capital of France is",
                "temperature": 0,
                "max_tokens": 8,
                "logprobs": 5,
            },
            timeout=30,
        )
        response.raise_for_status()
        return response.json()["choices"][0]["logprobs"]
```

```
    finally:
        kill_process_tree(process.pid)
```

评论区精华

本 PR 没有 review 评论（review_comments_count = 0）。唯一的评论区交互是作者发起 /rerun-test test/registered/openai_server/basic/test_openai_completion_rust.py，CI bot 随即报告在 1-gpu-5090 上重跑通过。PR body 中关于测试边界的论证是核心观点：该测试检查的是 rust 前端是否驱动与 python 前端相同的推理路径，prefill 是否跑在 CUDA graph 下不是测试目标，graph 覆盖应留给专门测试。

- 通过 /rerun-test 验证修复 (test): 修复经重跑验证有效，测试在 1-gpu-5090 上通过。

风险与影响

- 风险：整体风险极低：变更仅涉及测试启动参数，不触碰产品代码。需关注的剩余风险包括：1）该测试自身不再覆盖 prefill CUDA graph 路径，若未来对相同模型做 graph 相关回归，需要依赖 test_prefill_cuda_graph_runner.py 等专门测试补位；2）同类 ' bit-identical 前端对比 ' 测试若继续在默认配置下运行，仍可能遇到相同的 batch padding 干扰，本次修复只覆盖了 test_openai_completion_rust.py 一个文件；3）测试通过的确定性现在依赖 *eager* prefill 的数值确定性，如果未来调度器改变 *eager* 路径的 batch 组包方式，该测试仍可能受影响。
- 影响：影响范围限定在 CI：修复了 base-b-test-1-gpu-small 阶段的偶发失败，消除了计划任务运行中的 flaky。对最终用户无行为影响，对团队而言恢复了 rust 前端 parity 测试作为回归防线的可信度。同时该 PR 的分析过程为其他模块提供了值得借鉴的方法论：当测试目标是验证 ' 两条路径驱动同一推理 ' 时，应显式固定与测试目标无关的变量（此处为 batch 组合）。
- 风险标记：测试覆盖盲区，CI 时序依赖回归，同类 parity 测试未一并加固

关联脉络

- PR #33352 fix: always capture default prefill CUDA graph: 直接因果：该 PR 移除 prefill CUDA graph 内存门槛后，rust parity 测试模型开始捕获 prefill graph，其 batch padding 引入数值不确定性，触发本 PR 修复的 CI 偶发失败。
- PR #32785 fix: avoid piecewise prefill graph for trtllm_mla: 同主题：同样是 prefill graph 行为改变导致测试 / 性能回归，说明 prefill graph 的副作用需要系统性关注。