PR #36985 完整报告

sgl-project/sglang

test: re-enable FlashInfer per-token NVFP4 coverage

合并时间：2026-08-31 14:44

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36985

执行摘要

- 一句话：恢复 Blackwell NVFP4 MoE 两项 GSM8K 精度测试
- 推荐动作：值得快速浏览，尤其适合关注跨仓库依赖治理与测试恢复流程的读者。设计要点：① 严格区分「测试启用」与「依赖升级」职责，本 PR 刻意不碰包版本，将改动面压到最小；② 用完整 GSM8K 复测 + 明确环境快照（镜像、硬件、包版本、日志哈希）作为恢复测试的证据链，而非仅依赖上游声明修复；③ 保留原始阈值和工作负载不做放宽，确保召回质量不打折。

功能与动机

FlashInfer 0.6.16.post4 起，TRTLLM-Gen per-token NVFP4 MoE 在 SM100/SM103 上会产生 NaN（根因见 flashinfer-ai/flashinfer#4486），导致首个真实 prefill 触发采样器 NaN 断言、GSM8K 得分为 0，因此两个精度测试被临时 skip。FlashInfer 0.6.18 包含 #4563，该修复移除了 per-token FP4 缩放场景下的非法 tileN=192 tactic（同时保留其他受支持模式的该 tile），问题解决后即可恢复测试覆盖。本 PR 依赖 #36954 使注册 CI 镜像先安装 FlashInfer 0.6.18。

实现拆解

本 PR 全部改动为删除测试中的 @unittest.skip 装饰器及其注释，共 2 个文件、0 行新增、14 行删除，无任何运行时源码或配置改动。

1. 恢复在线 NVFP4 测试：在 test/registered/backends/test_flashinfer_nvfp4_online_moe_backend.py 中，删除 TestFlashinferTrtllmGenMoeBackendNvFp4Online 类前的 @unittest.skip 装饰器及 8 行说明注释。注释原本解释“该文件以 failfast 运行，保留 skip 会连带截断排序在后的类”，删除后其后缀的 TestFlashinferCuteDSLMoeBackendNvFp4Online 等类也会正常执行。
2. 恢复 per-token 路由测试：在 test/registered/backends/test_flashinfer_trtllm_gen_moe_backend.py 中，删除 TestFlashinferTrtllmGenMoeBackendNvFp4PerTokenActivationRouted 类前的 @unittest.skip 及 6 行注释。该类通过 extra_env = {"SGLANG_FLASHINFER_NVFP4_PER_TOKEN_ACTIVATION": "1"} 打开 per-token 激活路径，是 #4486 直接影响到的模式，是本次恢复的重点覆盖。
3. 保持既有断言不变：两个 test_gsm8k 方法的 200 条 GSM8K 样本、128 线程、max_tokens=512 工作负载和分数阈值（0.90/0.89）均保持不变。

4. 验证与依赖配套：作者在 4×B300（SM103）、TP4/EP4、FlashInfer 0.6.18 下完成完整 GSM8K 复测，routed 模式得分 0.920（阈值 >0.89）、在线模式得分 0.960（阈值 >0.90），均通过。PR body 明确当前镜像元数据仍 pin 0.6.17，验证期间使用显式包覆盖，正式的镜像升级由 #36954 负责。

关键文件：

- test/registered/backends/test_flashinfer_nvfp4_online_moe_backend.py（模块 NVFP4 测试；类别 test；类型 test-coverage；符号 TestFlashinferTrtllmGenMoeBackendNvFp4Online, FlashinferNvFp4OnlineMoeBackendBase, test_gsm8k）：删除 TestFlashinferTrtllmGenMoeBackendNvFp4Online 的 @unittest.skip，恢复 FlashInfer TRTLLM-Gen 后端在线 NVFP4 MoE 的 GSM8K 精度覆盖；注释中说明了 failfast 模式下 skip 会连带截断后续类，删除后整个文件覆盖恢复完整。
- test/registered/backends/test_flashinfer_trtllm_gen_moe_backend.py（模块 NVFP4 测试；类别 test；类型 test-coverage；符号 TestFlashinferTrtllmGenMoeBackendNvFp4PerTokenActivationRouted, FlashinferTrtllmGenMoeBackendNVFP4Base）：删除 TestFlashinferTrtllmGenMoeBackendNvFp4PerTokenActivationRouted 的 @unittest.skip 及 failfast 说明，恢复 #4486 直接影响的 per-token NVFP4 路由模式测试，是本次恢复的核心覆盖点。

关键符号：test_gsm8k

关键源码片段

test/registered/backends/test_flashinfer_nvfp4_online_moe_backend.py

删除 TestFlashinferTrtllmGenMoeBackendNvFp4Online 的 @unittest.skip，恢复 FlashInfer TRTLLM-Gen 后端在线 NVFP4 MoE 的 GSM8K 精度覆盖；注释中说明了 failfast 模式下 skip 会连带截断后续类，删除后整个文件覆盖恢复完整。

```
# 基类：启动 flashinfer_trtllm 后端的 sglang serve（TP4/EP4、nvfp4_online 量化）
class FlashinferNvFp4OnlineMoeBackendBase(CustomTestCase):
    # ... setUpClass 中组装 serve 命令并拉起进程 ...

    @classmethod
    def tearDownClass(cls):
        kill_process_tree(cls.process.pid)

    def test_gsm8k(self):
        # 完整 GSM8K 精度入口：200 条样本、128 线程、completion API、max_tokens=512
        args = SimpleNamespace(
            base_url=self.base_url,
            model=self.model,
            eval_name="gsm8k",
            num_examples=200,
            num_threads=128,
            **self.eval_args,
        )
        metrics = run_eval(args)
```

```
        print(f"{metrics=}")
        # 阈值保持 0.90 不变，不因重新启用而放宽
        self.assertGreater(metrics["score"], 0.90)
        if self.spec_accept_length_threshold is not None:
            server_info = requests.get(self.base_url + "/server_info").json()
            avg_spec_accept_length = server_info["internal_states"][0][
                "avg_spec_accept_length"
            ]
            print(f"{avg_spec_accept_length=}")
            self.assertGreater(
                avg_spec_accept_length, self.spec_accept_length_threshold
            )


# 重新启用：FlashInfer 0.6.18（#4563）已移除 per-token FP4 下非法的 tileN=192 tactic
class TestFlashinferTrtllmGenMoeBackendNvFp4Online(
    FlashinferNvFp4OnlineMoeBackendBase, CustomTestCase
):
    backend = "flashinfer_trtllm"
    model = "Qwen/Qwen3-30B-A3B-Instruct-2507-FP8"
    eval_args = {"api": "completion", "max_tokens": 512}
    # NVFP4 4-over-6 转换相关环境变量，覆盖 default 与 E4M3 变体
    extra_env = {
        "FLASHINFER_NVFP4_4OVER6": "1",
        "FLASHINFER_NVFP4_4OVER6_ERR_MODE": "MSE",
        "FLASHINFER_NVFP4_4OVER6_ERR_USE_FAST_MATH": "1",
        "FLASHINFER_NVFP4_4OVER6_E4M3_USE_256": "1",
        # 与 #4486 排查时保持一致：shared_expert 与第 40-47 层不参与 NVFP4
        "SGLANG_FP4_IGNORED_LAYERS": ",".join(
            ["shared_expert"]
            + [f"model.layers.{layer_id}" for layer_id in range(40, 48)]
        ),
    }
```

test/registered/backends/test_flashinfer_trtllm_gen_moe_backend.py

删除 TestFlashinferTrtllmGenMoeBackendNvFp4PerTokenActivationRouted 的 @unittest.skip 及 failfast 说明，恢复 #4486 直接影响的 per-token NVFP4 路由模式测试，是本次恢复的核心覆盖点。

```
# 该文件按后端模式组织多个 GSM8K 精度测试类，均继承各自 Base
class TestFlashinferTrtllmGenMoeBackendFP8(
    FlashinferTrtllmGenMoeBackendFP8Base, CustomTestCase
):
    backend = "flashinfer_trtllm"


class TestFlashinferTrtllmGenMoeBackendNVFP4(
    FlashinferTrtllmGenMoeBackendNVFP4Base, CustomTestCase
):
```

```
        backend = "flashinfer_trtllm"


    # 重新启用：此模式即 flashinfer-ai/flashinfer#4486 的 NaN 复现路径。
    # FlashInfer 0.6.18 通过 #4563 在 per-token scaling 下过滤 tileN=192 后恢复。
    class TestFlashinferTrtllmGenMoeBackendNvFp4PerTokenActivationRouted(
        FlashinferTrtllmGenMoeBackendNVFP4Base, CustomTestCase
    ):
        # 关键开关：打开 FlashInfer per-token NVFP4 激活量化路径
        extra_env = {"SGLANG_FLASHINFER_NVFP4_PER_TOKEN_ACTIVATION": "1"}
        # 使用 routed 变体（对应 flashinfer_trtllm_routed 后端）
        backend = "flashinfer_trtllm_routed"


    if __name__ == "__main__":
        unittest.main()
```

评论区精华

本 PR 的 review 阶段非常简洁：两位审核者 mmangkad 和 Fridge003 均在无评论的情况下直接 APPROVED，没有产生任何 inline review 评论。

唯一的互动发生在 PR 时间线上：作者请求合并（"Hi @mmangkad, can we merge this?"），mmangkad 随即触发 /rerun-test 对两个测试文件在 4-gpu-b200 上重跑，github-actions 返回两个测试全部通过。这说明变更意图明确、风险面窄，审核焦点全部落在 CI 复测结果上。

- CI 重跑验证两个 NVFP4 测试 (testing): 在 FlashInfer 0.6.18 环境下两个 GSM8K 测试均通过，验证了上游 #4563 修复有效，可以合入。

风险与影响

- 风险：
  - 依赖时序耦合：本 PR 的测试能否通过强依赖 #36954 将 CI 镜像升级到 FlashInfer 0.6.18。PR body 已注明当前镜像仍 pin 0.6.17，若 #36954 未先合入或镜像未重建，两个测试会因 NaN 问题直接失败。
  - 硬件可达性：测试目标平台是 SM100/SM103（B300），CI 若落到其他架构 runner（如 b200 已确认可用）可能 skip 或行为不一致。
  - 上游修复范围局限：#4563 只对 per-token NVFP4 过滤 tileN=192，其他 FP4 模式仍保留该 tile。若上游后续调整 tactic 表或引入新的非法组合，测试可能再次暴露问题——这正是本 PR 的价值所在，但也意味着测试对上游变更敏感。
  - 无源码回归风险：仅删除测试 skip，不触碰任何运行时路径，SGLang 推理逻辑零风险。
- 影响：
  - CI 覆盖：恢复两个关键的 Blackwell NVFP4 MoE 精度测试，为 FlashInfer 上游回归提供守门能力，防止 0.6.16.post4/0.6.17 的 NaN 问题再次悄无声息地进入 CI。
  - 依赖升级闭环：与 #36954 构成「依赖升级 + 功能验证」的完整协作链路，是跨仓库（SGLang + FlashInfer）联合排查的典型闭合案例。

- 团队参考价值：PR body 记录了完整的版本 bisect 结论、验证环境、日志哈希和复现命令，可为后续同类依赖问题提供可复用的排查模板。
- 用户影响：无直接运行体验影响；间接上，NVFP4 MoE 推理质量此后受 CI 持续守护。
- 风险标记：依赖上游镜像升级，跨仓库时序耦合，测试硬件受限 SM100/SM103，上游修复范围局限

关联脉络

- PR #36954 [Deps] Bump FlashInfer to 0.6.18: 本 PR 的前置依赖：CI 镜像需先升级到 FlashInfer 0.6.18，才能保证两个恢复的 NVFP4 测试通过；PR body 明确当前镜像仍 pin 0.6.17，验证期间使用显式包覆盖。
- PR #35547 Add Laguna-XS-2.1 / S-2.1 NVFP4 nightly gsm8k tests: 同为 NVFP4 + GSM8K 精度测试主题，反映仓库对 NVFP4 量化路径精度持续建立 CI 守护的演进方向。