PR #33634 完整报告

sgl-project/sglang

[NPU] Add test for --dllm-fdfo

合并时间：2026-08-25 10:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33634

执行摘要

- 一句话：新增 NPU 端 FDFO 调度性能回归测试
- 推荐动作：值得快速浏览而非精读。关注三点：NPU 测试的封装方式（run_bench_serving、register_npu_ci）；长耗时测试如何通过 nightly 与 PR 套件划分来平衡 CI 开销与覆盖；以及用均值 TTFT、吞吐、P99 TTFT 三重指标交叉验证调度特性收益的断言写法。若要扩展，可考虑补充正确性对比和容差阈值。

功能与动机

PR body 说明：'This PR adds a test case to verify that the --dllm-fdfo (First-Done-First-Out) scheduling flag takes effect and improves performance for Diffusion LLM models on Ascend NPU backend. When enabled, FDFO prioritizes requests that complete their diffusion steps earlier, reducing waiting time and improving hardware utilization.' 即通过可重复的基准测试把这个调度特性的性能收益固化下来，避免后续改动破坏 NPU 端 Diffusion LLM 的调度效果。

实现拆解

实现拆解如下：

1. 新增测试文件与测试骨架：在 test/registered/npu/basic_function/dllm/ 下新增 test_npu_llada2_mini_fdfo.py，定义 TestLLaDA2MiniFDFO(CustomTestCase) 测试类及唯一测试方法 test_dlla_fdfo_vs_no_fdfo_performance。测试复用 sglang.test.ascend.test_ascend_utils 里的 run_bench_serving 封装和 LLaDA2_0_MINI_WEIGHTS_PATH 权重常量，避免在测试中硬编码模型路径。
2. 构造 A/B 对比参数：准备两组 common_args，除 FDFO 开关外完全一致——实验组不显式传开关（依赖默认开启），对照组显式传 --no-dllm-fdfo 关闭 FDFO。两组都固定 --tp-size 2、--mem-fraction-static 0.9、--max-running-requests 16、--dllm-algorithm LowConfidence，并在运行时附加 --attention-backend ascend --disable-cuda-graph。
3. 运行基准并断言三项指标：每组参数下以 128 条 prompt、输入 3584 token、输出 1024 token、request_rate=inf、max_concurrency=16 的负载各跑一轮 bench，收集 mean_ttft_ms、total_throughput、p99_ttft_ms。最后断言开启 FDFO 后三项指标更优：平均 TTFT 更低、总吞吐更高、P99 TTFT 更低，用多个相互印证的指标降低单指标抖动带来的误判。

4. CI 注册与配套演进：通过 register_npu_ci(est_time=400, suite="nightly-2-npu-a3", nightly=True) 注册到 NPU nightly 套件。提交历史显示最初还注册了 PR 运行套件 stage-b-test-2-npu-a3，经 review 后仅在最后的 commit（'Update NPU CI registration for test_npu_llada2_mini_fdfo'）中移除；其他配套调整还包括模型路径改用常量、调整 import 顺序以满足 lint。

关键文件：

- test/registered/npu/basic_function/dllm/test_npu_llada2_mini_fdfo.py（模块 NPU 测试；类别 test；类型 test-coverage；符号 TestLLaDA2MiniFDFO, test_dlla_fdfo_vs_no_fdfo_performance）：本 PR 唯一变更文件，新增 FDFO 性能回归测试：使用 LLaDA2.0-mini 跑两组 A/B 基准并断言开启 FDFO 后 TTFT 与吞吐更优，同时演示 NPU 测试的注册方式。

关键符号：TestLLaDA2MiniFDFO, test_dlla_fdfo_vs_no_fdfo_performance

关键源码片段

test/registered/npu/basic_function/dllm/test_npu_llada2_mini_fdfo.py

本 PR 唯一变更文件，新增 FDFO 性能回归测试：使用 LLaDA2.0-mini 跑两组 A/B 基准并断言开启 FDFO 后 TTFT 与吞吐更优，同时演示 NPU 测试的注册方式。

```
import unittest

from sglang.test.ascend.test_ascend_utils import (
    LLaDA2_0_MINI_WEIGHTS_PATH,
    run_bench_serving,
)
from sglang.test.ci.ci_register import register_npu_ci
from sglang.test.test_utils import CustomTestCase

# 该用例单次约需 400 秒，只注册到 nightly-2-npu-a3 套件，避免阻塞 PR 流水线
register_npu_ci(est_time=400, suite="nightly-2-npu-a3", nightly=True)


class TestLLaDA2MiniFDFO(CustomTestCase):
    """验证 LLaDA2.0-mini 在开启 --dllm-fdfo 后调度性能提升。

    [Test Category] Diffusion LLM
    [Test Target] --dllm-fdfo
    """

    def test_dlla_fdfo_vs_no_fdfo_performance(self):
        # 两组 server 参数除 FDFO 开关外完全一致，构成 A/B 对比；
        # 对照组显式传 --no-dllm-fdfo，避免依赖默认值变化
        common_args = [
            [
                "--trust-remote-code",
                "--tp-size", 2,
```

```
                "--mem-fraction-static", 0.9,
                "--disable-radix-cache",
                "--max-running-requests", 16,
                "--dllm-algorithm", "LowConfidence",
                # 实验组：不传开关，即默认启用 FDFO
            ],
            [
                "--trust-remote-code",
                "--tp-size", 2,
                "--disable-radix-cache",
                "--mem-fraction-static", 0.9,
                "--max-running-requests", 16,
                "--dllm-algorithm", "LowConfidence",
                "--no-dllm-fdfo", # 对照组：显式关闭 FDFO
            ],
        ]
        # 依次收集两组实验的 mean TTFT、总吞吐、P99 TTFT
        ttfts, throughputs, p99_ttfts = [], [], []
        for common_arg in common_args:
            # NPU 后端固定使用 ascend attention backend，并关闭 cuda graph
            other_args = common_arg + [
                "--attention-backend", "ascend", "--disable-cuda-graph",
            ]
            # 固定负载：128 条 prompt、输入 3584 token、输出 1024 token
            res = run_bench_serving(
                model=LLaDA2_0_MINI_WEIGHTS_PATH,
                num_prompts=128,
                random_input_len=3584,
                random_output_len=1024,
                request_rate=float("inf"),
                max_concurrency=16,
                other_server_args=other_args,
            )
            ttfts.append(res["mean_ttft_ms"])
            throughputs.append(res["total_throughput"])
            p99_ttfts.append(res["p99_ttft_ms"])

        # FDFO（First-Done-First-Out）优先调度先完成扩散步骤的请求，
        # 预期效果：更低的平均 TTFT、更高的总吞吐、更低的 P99 TTFT
        assert float(ttfts[0]) < float(ttfts[1])
        assert float(throughputs[0]) > float(throughputs[1])
        assert float(p99_ttfts[0]) < float(p99_ttfts[1])


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

评论区精华

review 里只有 cherryblo 的一条评论，聚焦 CI 套件取舍：

> cherryblo 在 test_npu_llada2_mini_fdfo.py 第 10 行评论：'Add this test case to the nightly pipeline, not to PR runs.'

核心争议是约 400 秒的长耗时测试不应进入 PR 运行流水线，否则会拖慢主流程 CI 的反馈速度。作者在最终 commit 中移除了 stage-b-test-2-npu-a3 注册、仅保留 nightly 注册，cherryblo 随后 APPROVED，讨论闭环。

- 测试是否应进入 PR 运行 CI (testing): 作者在最终 commit 中移除 stage-b-test-2-npu-a3 注册，仅保留 nightly-2-npu-a3，cherryblo 随后 APPROVED。

风险与影响

- 风险：风险点如下：
- 断言 flaky 风险：三个性能断言都是无容差阈值的严格大小比较（TTFT[0] < TTFT[1] 等），NPU 共享 CI 环境存在负载波动、散热降频等因素，可能导致 nightly 偶发失败，需要有合理重跑预期。
- 缺少正确性校验：测试只比较性能指标，没有校验 FDFO 开启后输出与关闭时是否一致，无法发现 ' 调度优化以牺牲输出质量换速度 ' 的回归。
- 覆盖范围有限：仅覆盖 LLaDA2.0-mini 单模型、单负载点（128 prompts、16 并发），FDFO 在其他 Diffusion LLM 模型或其他并发档位下的表现未覆盖。
- 生产代码零改动：仅新增测试文件，对线上服务无回归风险；但测试依赖 run_bench_serving 与 CustomTestCase 的接口稳定性，相关封装变更时需要同步维护。
- 影响：影响评估：
- 用户：无用户可见影响，不改变任何运行行为。
- 系统 /CI：NPU nightly 套件 nightly-2-npu-a3 新增一个约 400 秒的测试任务；不进入 PR 流水线，PR 反馈速度不受影响。
- 团队：为 FDFO 特性在 Ascend NPU 端的性能表现提供持续回归保护，后续调度器或 NPU 后端的改动若导致 Diffusion LLM 的 TTFT 或吞吐回退，可由该测试暴露；同时为 NPU 端 ' 性能特性配性能回归测试 ' 建立了可复用的写法与套件划分惯例。
- 风险标记：性能断言无容差易 flaky，缺少正确性校验，仅 nightly 覆盖、无 PR 级保护

关联脉络

- PR #35840 Add PD test for inkling with mxfp8 KV: 同为新增特定硬件 / 调度相关特性的回归测试，并涉及 CI 套件注册与调度路径状态处理，与本 PR' 为特性补性能测试 ' 的模式一致。
- PR #36222 [CP V1 Deprecation 1/5] Migrate tests to strategy-based prefill CP: 同为测试组织与 CI 套件调整方向，反映调度相关测试在演进中的注册与迁移惯例。