PR #26648 完整报告

sgl-project/sglang

[CI] Split PP tests into base and extra suites

合并时间：2026-05-29 12:15

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26648

执行摘要

- 一句话：拆分 PP 测试为基础和扩展两套
- 推荐动作：值得关注这类 CI 拆分模式，可作为 SGLang 项目中其他大型测试文件（如 attention 测试、multimodal 测试）拆分的参考。推荐阅读 test/registered/pp/test_pp_single_node_extra.py 的注册方式和类结构，了解如何将测试与 CI 阶段映射。

功能与动机

原始测试文件 test_pp_single_node.py 达到 681 行、12 个测试类，在 CI 中作为单一阶段运行耗时较长且难以并行。PR 作者在描述中指出『For better CI experience』，通过拆分文件使基础测试和扩展测试可以分属不同 CI 阶段（base-c 和 extra-b），从而降低单次 CI 的等待时间，提高反馈效率。

实现拆解

1. 创建新文件：新增 test/registered/pp/test_pp_single_node_extra.py，将原文件中的所有扩展测试类（涵盖 Qwen 系列、GLM-41V 的 PP 精度测试）完整移入，并为每个类注册独立的 CI 阶段（CUDA 下 stage="extra-b"，AMD 下 suite="stage-c-test-4-gpu-amd"）。
2. 精简原文件：从 test_pp_single_node.py 中删除上述 6 个扩展测试类和相关的导入（如 DEFAULT_MODEL_NAME_FOR_TEST_GLM_41V_PP），保留 6 个核心测试类，并调整 CI 注册的估计时间从 554 秒降至 500 秒。
3. 更新文档字符串与注释：同步修改两个文件的 Usage 部分，准确列出各文件包含的测试命令；更新 TestGemma4PPAccuracy 的 skip 理由和类注释，使其与实际运行条件一致。
4. 调整 CI 配置：通过 register_cuda_ci 和 register_amd_ci 为新文件定义独立的 CI 阶段，确保 extra 阶段的 runner 配置与 base 阶段一致（均为 4-gpu-h100），并设置 350 秒的预估时间。
5. 验证与重跑：PR 作者通过 comment /rerun-test 手动触发了两个新文件的独立运行，确认均能通过。

关键文件：

- test/registered/pp/test_pp_single_node.py（模块 PP 测试；类别 test；类型 test-coverage；符号 TestPPAccuracy, TestDPAttentionDP2PP2, TestGemma4PPAccuracy, TestGemma4PLEPPAccuracy）：原文件，被精简，删除了 6

个扩展测试类并调整 CI 注册

- test/registered/pp/test_pp_single_node_extra.py（模块 扩展测试；类别 test；类型 test-coverage；符号 TestQwenVLPPAccuracy, TestQwenPPAccuracy, TestQwenPPTieWeightsAccuracy, TestQwenMoePPAccuracy）：新文件，包含移出的 6 个扩展测试类，独立注册 CI 阶段

关键符号：test_gsm8k, test_mmmu, test_pp_consistency, run_gsm8k_test, setUpClass, tearDownClass

关键源码片段

test/registered/pp/test_pp_single_node.py

原文件，被精简，删除了 6 个扩展测试类并调整 CI 注册

```
"""
Usage:
python3 -m unittest test_pp_single_node.TestPPAccuracy.test_gsm8k
python3 -m unittest test_pp_single_node.TestDPAttentionDP2PP2.test_gsm8k
python3 -m unittest test_pp_single_node.TestGemma4PPAccuracy.test_gsm8k
...
"""

# 注册到 base-c 阶段，预估时间 500 秒
register_cuda_ci(est_time=500, stage="base-c", runner_config="4-gpu-h100")
register_amd_ci(est_time=500, suite="stage-c-test-4-gpu-amd")

@unittest.skipIf(
    is_in_amd_ci(),
    "Gemma4 PP not yet validated on AMD",
)
class TestGemma4PPAccuracy(unittest.TestCase):
    """End-to-end PP=2 accuracy gate for Gemma4 multimodal.

    Gemma4 has full-attention layers with head_dim=512 (FA's max is 256), so
    sglang auto-selects the triton attention backend; no manual flag needed.
    """
    @classmethod
    def setUpClass(cls):
        cls.model = DEFAULT_MODEL_NAME_FOR_TEST_GEMMA4_PP
        cls.base_url = "http://127.0.0.1:23333"
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
                "--tp-size", 1,
                "--pp-size", 2,
                "--trust-remote-code",
```

```
                "--enable-multimodal",
            ],
        )
    # ... 测试方法略
```

test/registered/pp/test_pp_single_node_extra.py

新文件，包含移出的 6 个扩展测试类，独立注册 CI 阶段

```
"""
Usage:
python3 -m unittest test_pp_single_node_extra.TestQwenVLPPAccuracy.test_gsm8k
...
"""
import time
import unittest
from types import SimpleNamespace

from sglang.srt.utils import kill_process_tree
from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
from sglang.test.run_eval import run_eval
from sglang.test.test_utils import (
    DEFAULT_MODEL_NAME_FOR_TEST_VL_PP,
    DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
    is_in_amd_ci,
    is_in_ci,
    popen_launch_server,
)

# 注册该文件到 CI 的 extra-b 阶段
register_cuda_ci(est_time=350, stage="extra-b", runner_config="4-gpu-h100")
register_amd_ci(est_time=350, suite="stage-c-test-4-gpu-amd")


@unittest.skipIf(
    is_in_amd_ci(),
    "VLM PP accuracy too low on AMD (0.48-0.50 with both aiter and triton)",
)
class TestQwenVLPPAccuracy(unittest.TestCase):
    """测试 Qwen VL 模型在 PP 下的精度，需 1TP+4PP 配置"""

    @classmethod
    def setUpClass(cls):
        cls.model = DEFAULT_MODEL_NAME_FOR_TEST_VL_PP
        cls.base_url = "http://127.0.0.1:23333"
        cls.process = popen_launch_server(
            DEFAULT_MODEL_NAME_FOR_TEST_VL_PP,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
```

```
            "--tp-size", 1,
            "--pp-size", 4,
            "--chunked-prefill-size", 8192,
            "--enable-multimodal",
        ],
    )

def test_gsm8k(self):
    """GSM8K 精度测试，阈值 0.65"""
    args = SimpleNamespace(
        base_url=self.base_url,
        model=self.model,
        eval_name="gsm8k",
        api="completion",
        max_tokens=512,
        num_examples=200,
        num_threads=128,
    )
    metrics = run_eval(args)
    print(f"{metrics=}")
    self.assertGreaterEqual(metrics["score"], 0.65)
    time.sleep(4) # 等待内存检查

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

@unittest.skipIf(is_in_ci(), "To reduce the CI execution time.")
def test_mmmu(self):
    """MMMU 精度测试，仅在本地运行"""
    args = SimpleNamespace(
        base_url=self.base_url,
        model=self.model,
        eval_name="mmmu",
        num_examples=None,
        num_threads=32,
    )
    metrics = run_eval(args)
    print(f"{metrics=}")
    self.assertGreater(metrics["score"], 0.26)
```

评论区精华

无 review 讨论，PR 由作者自行合并。但存在一条机器人的每日配额提示和一次手动 rerun 命令，确认了拆分后的两个文件可单独运行。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。主要风险包括：（1）新文件导入路径或依赖遗漏导致测试失败；（2）CI 阶段注册重复或遗漏导致测试被跳过或重复执行；（3）后续开发者不适应新文件命名，可能需要额外 documentation 指引。但这些风险已通过 rerun 验证得到缓解。
- 影响：该 PR 仅影响测试组织方式，不影响任何生产代码。对 CI 的影响：base-c 阶段测试数减少，extra-b 阶段测试数增加，总并行度提升，预计可缩短 PP 测试的整体 CI 耗时。对开发者：提交涉及 PP 测试的 PR 时，需要将新测试类放入对应文件，或调整注册阶段。对系统：无运行时影响。
- 风险标记：CI 阶段配置变更，测试文件新增，依赖导入调整

关联脉络

- 暂无明显关联 PR