PR #22913 完整报告

sgl-project/sglang

test(4-gpu-b200): split test_qwen35_models.py + bump partitions 5→6

合并时间：2026-04-17 09:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22913

执行摘要

- 一句话：拆分 Qwen3.5 FP4 模型测试文件并增加 CI 分区，避免超时失败。
- 推荐动作：该 PR 是基础设施优化，值得技术管理者关注 CI 配置变更以调整测试策略；工程师可参考测试分割策略，在类似场景下优化测试套件执行时间。

功能与动机

PR body 指出，原文件合并运行在 B200 主机上经常超过 30 分钟步超时（示例失败链接：https://github.com/sgl-project/sglang/actions/runs/24454292569/job/71542805109），因此拆分文件以便 CI 自动分区器分散负载。此外，根据 reviewer 反馈移除了 v1 MTP 测试，因为 v2 MTP 将成为默认。

实现拆解

1. 拆分原测试文件：删除 test/registered/4-gpu-models/test_qwen35_models.py，创建两个新文件：
   - test_qwen35_fp4_triton.py：包含 TestQwen35FP4 类，专注于 Triton 后端准确性测试，设置 est_time=720s。
   - test_qwen35_fp4_mtp_v2.py：包含 TestQwen35FP4MTPV2 类，用于测试 v2 MTP 推测解码，设置 est_time=540s，并通过 envs.SGLANG_ENABLE_SPEC_V2.set(True) 启用 v2 特性。
2. 移除冗余测试：根据 reviewer 反馈，删除 TestQwen35FP4MTP（v1 MTP 测试），因为 v2 MTP 将默认启用，避免测试冗余。
3. 更新 CI 配置：修改 .github/workflows/pr-test.yml，将 stage-c-test-4-gpu-b200 的分区数量从 5 增加到 6，并更新 auto-partition-size 参数，确保测试负载均匀分布，防止超时。
4. 测试配套验证：保持准确性阈值（如 gsm8k ⩾ 0.95）和性能指标（如 avg_spec_accept_length > 3.3）不变，通过 CI 运行验证所有分区在超时内完成。

关键文件：

- test/registered/4-gpu-models/test_qwen35_models.py（模块 测试套件；类别 test；类型 deletion；符号 TestQwen35FP4, test_gsm8k, TestQwen35FP4MTP, setUpClass）：原测试文件，包含 TestQwen35FP4、TestQwen35FP4MTP 和 TestQwen35FP4MTPV2 三个类，因拆分而被删除。

- test/registered/4-gpu-models/test_qwen35_fp4_triton.py（模块 测试套件；类别 test；类型 test-coverage；符号 TestQwen35FP4, test_gsm8k）：新增文件，包含 TestQwen35FP4 类，专注于 Qwen3.5 FP4 模型的 Triton 后端准确性测试，是拆分后的核心测试文件之一。
- test/registered/4-gpu-models/test_qwen35_fp4_mtp_v2.py（模块 测试套件；类别 test；类型 test-coverage；符号 TestQwen35FP4MTPV2, setUpClass, tearDownClass, test_gsm8k）：新增文件，包含 TestQwen35FP4MTPV2 类，用于测试 Qwen3.5 FP4 模型的 v2 MTP 推测解码功能，是拆分后的另一核心测试文件。
- .github/workflows/pr-test.yml（模块 CI 配置；类别 infra；类型 infrastructure）：修改 CI 工作流配置，将 stage-c-test-4-gpu-b200 的分区数量从 5 增加到 6，以适配测试文件拆分后的负载分布。

关键符号：TestQwen35FP4.test_gsm8k, TestQwen35FP4MTPV2.setUpClass, TestQwen35FP4MTPV2.tearDownClass, TestQwen35FP4MTPV2.test_gsm8k

关键源码片段

test/registered/4-gpu-models/test_qwen35_fp4_triton.py

新增文件，包含 TestQwen35FP4 类，专注于 Qwen3.5 FP4 模型的 Triton 后端准确性测试，是拆分后的核心测试文件之一。

```
import unittest
from sglang.test.accuracy_test_runner import AccuracyTestParams
from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.run_combined_tests import run_combined_tests
from sglang.test.test_utils import CustomTestCase, ModelLaunchSettings

# 注册 CI 测试，设置估计时间为 720 秒，以匹配拆分后的负载
register_cuda_ci(est_time=720, suite="stage-c-test-4-gpu-b200")

QWEN35_FP4_MODEL = "nvidia/Qwen3.5-397B-A17B-NVFP4"
ACC_THRESHOLDS = {QWEN35_FP4_MODEL: {"gsm8k": 0.95}} # 定义准确性阈值

class TestQwen35FP4(CustomTestCase):
    def test_gsm8k(self):
        # 基础服务器启动参数，针对 4-GPU B200 配置，包括 TP 大小、Mamba 调度等
        base_args = [
            "--tp-size", "4",
            "--chunked-prefill-size", "2048",
            "--mamba-scheduler-strategy", "extra_buffer",
            "--mamba-track-interval", "128",
            "--mamba-ssm-dtype", "bfloat16",
            "--max-running-requests", "128",
            "--reasoning-parser", "qwen3",
            "--attention-backend", "trtllm_mha",
            "--quantization", "modelopt_fp4",
            "--model-loader-extra-config", '{"enable_multithread_load": true,"num_threads": 64}',
```

```
]

# 定义测试变体，目前只启用 Triton 后端，FlashInfer 后端待修复
variants = [
    ModelLaunchSettings(
        QWEN35_FP4_MODEL,
        extra_args=base_args,
        variant="Triton",
    ),
    # TODO: 修复 FlashInfer 后端后重新启用
    # ModelLaunchSettings(
    #     QWEN35_FP4_MODEL,
    #     extra_args=base_args + ["--linear-attn-decode-backend", "flashinfer"],
    #     variant="FlashInfer",
    # ),
]

# 运行组合测试，验证 gsm8k 数据集的准确性，确保不低于阈值
run_combined_tests(
    models=variants,
    test_name="Qwen3.5-397B-A17B-NVFP4",
    accuracy_params=AccuracyTestParams(
        dataset="gsm8k",
        baseline_accuracy=ACC_THRESHOLDS[QWEN35_FP4_MODEL]["gsm8k"],
        num_examples=200,
        num_threads=128,
        max_tokens=16000,
        thinking_mode="qwen3",
        temperature=0.6,
        top_p=0.95,
        top_k=20,
    ),
)
```

test/registered/4-gpu-models/test_qwen35_fp4_mtp_v2.py

新增文件，包含 TestQwen35FP4MTPV2 类，用于测试 Qwen3.5 FP4 模型的 v2 MTP 推测解码功能，是拆分后的另一核心测试文件。

```
import unittest
from types import SimpleNamespace
import requests
from sglang.srt.environ import envs
from sglang.srt.utils import kill_process_tree
from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.kits.reasoning_kit import ReasoningTokenUsageMixin
from sglang.test.run_eval import run_eval
from sglang.test.test_utils import (
    DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
    DEFAULT_URL_FOR_TEST,
```

```
    CustomTestCase,
    popen_launch_server,
)

# 注册 CI 测试，设置估计时间为 540 秒，以适配拆分后的分区负载
register_cuda_ci(est_time=540, suite="stage-c-test-4-gpu-b200")

QWEN35_FP4_MODEL = "nvidia/Qwen3.5-397B-A17B-NVFP4"
ACC_THRESHOLDS = {QWEN35_FP4_MODEL: {"gsm8k": 0.95}} # 准确性阈值定义

class TestQwen35FP4MTPV2(ReasoningTokenUsageMixin, CustomTestCase):
    reasoning_parser_name = "qwen3" # 设置推理解析器

    @classmethod
    def setUpClass(cls):
        cls.model = QWEN35_FP4_MODEL
        cls.base_url = DEFAULT_URL_FOR_TEST
        cls.init_reasoning_token_verifier() # 初始化推理 token 验证器
        envs.SGLANG_ENABLE_SPEC_V2.set(True) # 启用 v2 推测解码特性
        # 启动服务器，使用 4-GPU 配置和推测解码参数
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=[
                "--tp-size", "4",
                "--chunked-prefill-size", "2048",
                "--mamba-scheduler-strategy", "extra_buffer",
                "--mamba-track-interval", "128",
                "--mamba-ssm-dtype", "bfloat16",
                "--max-running-requests", "128",
                "--reasoning-parser", "qwen3",
                "--attention-backend", "trtllm_mha",
                "--quantization", "modelopt_fp4",
                "--speculative-algorithm", "NEXTN", # v2 MTP 相关参数
                "--speculative-num-steps", "3",
                "--speculative-eagle-topk", "1",
                "--speculative-num-draft-tokens", "4",
                "--mem-fraction-static", "0.8",
                "--model-loader-extra-config", '{"enable_multithread_load": true,"num_threads": 64}',
            ],
        )

    @classmethod
    def tearDownClass(cls):
        envs.SGLANG_ENABLE_SPEC_V2.set(False) # 测试结束后禁用 v2 特性
        kill_process_tree(cls.process.pid) # 清理服务器进程

    def test_gsm8k(self):
```

```
# 设置评估参数，运行 gsm8k 数据集测试
args = SimpleNamespace(
    model=self.model,
    eval_name="gsm8k",
    num_shots=5,
    num_examples=200,
    max_tokens=16000,
    num_threads=128,
    repeat=1,
    temperature=0.6,
    top_p=0.95,
    top_k=20,
    base_url=self.base_url,
    host="http://127.0.0.1",
    port=int(self.base_url.split(":")[-1]),
)
metrics = run_eval(args) # 执行评估
print(f"{metrics=}")
self.assertGreaterEqual(metrics["score"], ACC_THRESHOLDS[self.model]["gsm8k"]) #
验证准确性

# 获取服务器信息，检查推测解码性能指标
server_info = requests.get(self.base_url + "/server_info")
avg_spec_accept_length = server_info.json()["internal_states"][0]["avg_spec_accept_
length"]
print(f"{avg_spec_accept_length=}")
self.assertGreater(avg_spec_accept_length, 3.3) # 确保推测解码性能达标
```

评论区精华

review 中仅显示批准，无详细评论；但 PR body 提及“per reviewer feedback — v2 MTP will be the default”，表明有隐含的设计决策：移除v1 MTP测试以避免冗余，聚焦于v2版本。

- 移除 v1 MTP 测试的决策 (design): 决定删除 v1 测试以简化测试套件，聚焦于 v2 版本。

风险与影响

- 风险：风险较低：文件分割可能略微增加维护复杂性，但测试逻辑未变，回归风险小；CI 分区调整需验证不会引入新的失败或资源冲突，例如分区数量增加可能导致资源分配不均，但 PR body 已确认所有分区在超时内完成。
- 影响：对用户无直接影响；系统测试执行更可靠，减少超时失败，提升 CI 稳定性；团队测试代码更模块化，便于未来扩展和维护，但需注意文件分割后的依赖管理。
- 风险标记：CI 配置变更风险，测试维护负担

关联脉络

- 暂无明显关联 PR