PR #26268 完整报告

sgl-project/sglang

[CI] Align score threshold in dsv4 disaggregation test

合并时间：2026-05-25 17:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26268

执行摘要

- 一句话：对齐 DSv4 分解测试阈值与测试工具
- 推荐动作：该 PR 为小范围维护优化，建议合并。值得关注的是 GSM8KMixin 的使用方式，若后续其他测试也采用类似模式，可进一步统一测试基础设施。

功能与动机

DSv4 分解 CI 测试（test_disaggregation_dsv4.py）存在两个问题：硬编码的精度阈值 0.95 过于严格，导致 CI 间歇性失败（根本原因将在其他 PR 修复）；测试内联实现了 test_gsm8k 方法，而不是复用共享的 GSM8KMixin 工具。本 PR 旨在与通用评估基础设施对齐并放松阈值以提升 CI 稳定性。

实现拆解

1. 复用 GSM8KMixin：删除内联的 test_gsm8k 方法，使 TestDisaggregationDSV4 同时继承 PDDisaggregationServerBase 和 GSM8KMixin，从而自动获得标准化评估逻辑。
2. 降低精度阈值：通过类属性 gsm8k_accuracy_thres = 0.93 将阈值从 0.95 下调至 0.93，减少因精度波动导致的 CI 失败。
3. 调整服务器启动参数：在 prefill 和 decode 服务器启动参数中，将 --max-running-requests 从 256 降至 128，移除 --mem-fraction-static 0.7，增加 --watchdog-timeout 900 防止超时杀死进程。

关键文件：

- test/registered/disaggregation/test_disaggregation_dsv4.py（模块 DSv4 测试；类别 test；类型 test-coverage；符号 TestDisaggregationDSV4, test_gsm8k, gsm8k_accuracy_thres）：唯一变更文件：复用 GSM8KMixin、降低阈值、调整服务器参数。

关键符号：test_gsm8k (deleted), start_prefill, start_decode

关键源码片段

test/registered/disaggregation/test_disaggregation_dsv4.py

唯一变更文件：复用 GSM8KMixin、降低阈值、调整服务器参数。

```
# test/registered/disaggregation/test_disaggregation_dsv4.py
```

```
import unittest
from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.kits.eval_accuracy_kit import GSM8KMixin # 复用标准评估工具
from sglang.test.server_fixtures.disaggregation_fixture import PDDisaggregationServerBase
from sglang.test.test_utils import (
    DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
    popen_launch_pd_server,
    try_cached_model,
)

register_cuda_ci(est_time=250, stage="base-c", runner_config="dsv4-8-gpu-h200")

DSV4_FLASH_MODEL = "sgl-project/DeepSeek-V4-Flash-FP8"
DEEPEP_CONFIG = '{"normal_dispatch":{"num_sms":96},"normal_combine":{"num_sms":96}}'
DSV4_FLASH_ENV = {
    "SGLANG_DSV4_FP4_EXPERTS": "0",
    "SGLANG_DEEPEP_NUM_MAX_DISPATCH_TOKENS_PER_RANK": "256",
}

_EAGLE_SPEC_ARGS = [
    "--speculative-algorithm", "EAGLE",
    "--speculative-num-steps", "1",
    "--speculative-eagle-topk", "1",
    "--speculative-num-draft-tokens", "2",
]

class TestDisaggregationDSV4(PDDisaggregationServerBase, GSM8KMixin):
    # 覆盖 GSM8KMixin 的默认阈值，从 0.95 降低到 0.93 以减少 CI 不稳定
    gsm8k_accuracy_thres = 0.93

    @classmethod
    def setUpClass(cls):
        super().setUpClass()
        cls.model = try_cached_model(DSV4_FLASH_MODEL)
        cls.start_prefill()
        cls.start_decode()
        cls.wait_server_ready(cls.prefill_url + "/health", process=cls.process_prefill)
        cls.wait_server_ready(cls.decode_url + "/health", process=cls.process_decode)
        cls.launch_lb()

    @classmethod
    def start_prefill(cls):
        prefill_args = [
            "--trust-remote-code",
            "--disaggregation-mode", "prefill",
            "--disaggregation-bootstrap-port", cls.bootstrap_port,
            "--tp", 4, "--dp", 4,
            "--enable-dp-attention",
            "--moe-a2a-backend", "deepep",
```

```
            "--deepep-config", DEEPEP_CONFIG,
            "--cuda-graph-max-bs", "128",
            "--max-running-requests", "128", # 从 256 降低到 128
            *_EAGLE_SPEC_ARGS,
            "--watchdog-timeout", "900", # 增加 watchdog 超时防止进程被误杀
        ]
        prefill_args += cls.transfer_backend + cls.rdma_devices
        cls.process_prefill = popen_launch_pd_server(
            cls.model, cls.prefill_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=prefill_args, env=DSV4_FLASH_ENV,
        )

    @classmethod
    def start_decode(cls):
        decode_args = [
            "--trust-remote-code",
            "--disaggregation-mode", "decode",
            "--disaggregation-bootstrap-port", cls.bootstrap_port,
            "--tp", 4, "--dp", 4,
            "--enable-dp-attention",
            "--base-gpu-id", 4,
            "--moe-a2a-backend", "deepep",
            "--deepep-config", DEEPEP_CONFIG,
            "--cuda-graph-max-bs", "128",
            "--max-running-requests", "128",
            *_EAGLE_SPEC_ARGS,
            "--watchdog-timeout", "900",
        ]
        decode_args += cls.transfer_backend + cls.rdma_devices
        cls.process_decode = popen_launch_pd_server(
            cls.model, cls.decode_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=decode_args, env=DSV4_FLASH_ENV,
        )

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

评论区精华

PR 作者在 body 中明确指出了两个动机：降低阈值缓解 flaky 失败，以及消除代码重复。review 评论仅来自 gemini-code-assist[bot]，表示对降低阈值没有额外反馈。实际 CI 运行（8-gpu-h200）一次即通过，作者确认成功。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。主要变更在测试文件，不涉及生产代码。降低阈值可能掩盖真正的精度退化，但作者明确声明根本原因将在其他 PR 修复，因此临时放松是合理的。服务器参数调整（max-running-requests 减半）可能影响吞吐，但测试环境压力较小，风险可控。
- 影响：仅影响 DSv4 分解 CI 测试。降低阈值后测试更稳定，减少了误报；复用 GSM8KMixin 使测试更易维护。对其他测试或生产部署无影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR