PR #25318 完整报告

sgl-project/sglang

split test_dsa_models_mtp into 4 files

合并时间：2026-05-16 05:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25318

执行摘要

- 一句话：拆分 DSAMTP 测试为 4 文件，避免 CI 超时
- 推荐动作：如果你关注 CI 流水线优化或测试架构设计，这个 PR 值得精读。特别是通过多重继承（Mixin）组合测试逻辑的模式，在 SGLang 测试框架中已有大量使用，该 PR 是良好的实践案例。对于仅使用 SGLang 推理的用户，可快速浏览了解变更即可。

功能与动机

原始 test_dsa_models_mtp.py 单文件包含 4 个重型集成测试用例，每个用例需启动 8 GPU 服务器运行 GSM8K 评估和单请求基准测试，累计运行时间超过 1200 秒，触发了 CI 每文件超时硬限制。拆分为 4 个文件可使每个文件独立分配时间预算，同时通过提取公共 fixture 减少重复代码。

实现拆解

1. 删除原始单文件：移除 test/registered/8-gpu-models/test_dsa_models_mtp.py（-368 行）。
2. 创建共享 fixture：在 python/sglang/test/server_fixtures/dsa_mtp_fixture.py 中定义 DsaMtpServerBase（服务器生命周期管理）和 DsaMtpEvalConfigDefaults（共享评估阈值），所有变体通过多重继承复用。
3. 创建 4 个独立测试文件：置于 test/registered/dsa_models_e2e/ 下，分别对应 DeepSeek V3.2 DP/TP 和 GLM-5 DP/TP，每个文件只定义一个测试类，继承 fixture 和 Mixin，仅需指定模型路径和少数变体参数。
4. 增强 Mixin：修改 SpecDecodingMixin（新增重试机制和 flush_cache）及 GSM8KMixin（微小调整），使独立文件无需重复编写测试方法。
5. 调整 CI stage：将原 stage-c 中的部分测试移至 extra-b 阶段，实现负载均衡，并更新每个文件的 est_time 和 runner_config。

关键文件：

- test/registered/8-gpu-models/test_dsa_models_mtp.py（模块 MTP 测试；类别 test；类型 deletion；符号 TestDeepseekV32DPMTP, setUpClass, tearDownClass, test_a_gsm8k）：原始单文件被完全删除，是本次拆分的起点，包含 4 个测试类的完整实现。
- python/sglang/test/server_fixtures/dsa_mtp_fixture.py（模块 测试夹具；类别 test；类型 test-coverage；符号 TestDsv32DP, DsaMtpEvalConfigDefaults,

DsaMtpServerBase, get_server_args）：新创建的共享 fixture 文件，封装服务器启动 /停止和评估阈值，是提升复用性的核心。

- test/registered/dsa_models_e2e/test_dsa_dsv32_dp_mtp.py（模块 MTP 测试；类别 test；类型 test-coverage；符号 TestDeepseekV32DPMTP）：新创建的 DeepSeek V3.2 DP + MTP 测试文件，独立运行，使用 fixture 和 Mixin 组装。
- test/registered/dsa_models_e2e/test_dsa_glm5_dp_mtp.py（模块 MTP 测试；类别 test；类型 test-coverage；符号 TestGLM5DPMTP）：新创建的 GLM-5 DP + MTP 测试文件，独立运行，使用 fixture 和 Mixin 组装。
- test/registered/dsa_models_e2e/test_dsa_dsv32_tp_mtp.py（模块 MTP 测试；类别 test；类型 test-coverage；符号 TestDeepseekV32TPMTP）：新创建的 DeepSeek V3.2 TP + MTP 测试文件，独立运行，使用 fixture 和 Mixin 组装。
- test/registered/dsa_models_e2e/test_dsa_glm5_tp_mtp.py（模块 MTP 测试；类别 test；类型 test-coverage；符号 TestGLM5TPMTP）：新创建的 GLM-5 TP + MTP 测试文件，独立运行，使用 fixture 和 Mixin 组装。
- python/sglang/test/kits/spec_decoding_kit.py（模块 测试工具；类别 test；类型 test-coverage）：修改了 SpecDecodingMixin，增加了速度测试的重试机制和 flush_cache 调用，提升测试稳定性。
- python/sglang/test/kits/eval_accuracy_kit.py（模块 测试工具；类别 test；类型 test-coverage）：小幅度修改 GSM8KMixin，配合新测试结构。

关键符号：TestDeepseekV32DPMTP, TestDeepseekV32TPMTP, TestGLM5DPMTP, TestGLM5TPMTP, DsaMtpServerBase, DsaMtpEvalConfigDefaults, GSM8KMixin, SpecDecodingMixin

关键源码片段

python/sglang/test/server_fixtures/dsa_mtp_fixture.py

新创建的共享 fixture 文件，封装服务器启动 / 停止和评估阈值，是提升复用性的核心。

```
"""
DSA model + MTP (EAGLE) speculative-decoding server fixture.

Variants combine `DsaMtpServerBase` (server lifecycle) with
`DsaMtpEvalConfigDefaults` (shared eval thresholds/params),
`GSM8KMixin` and `SpecDecodingMixin`, then set `model` and per-variant
overrides (`enable_dp_attention`, `mem_fraction_static`, `bs_1_speed_thres`).
"""

from sglang.srt.utils import kill_process_tree
from sglang.test.test_utils import (
    DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
    DEFAULT_URL_FOR_TEST,
    CustomTestCase,
    popen_launch_server,
)
```

```
class DsaMtpEvalConfigDefaults:
    """共享的评估阈值和参数，所有 MTP 变体复用."""

    # GSM8K 精度测试默认阈值
    gsm8k_accuracy_thres = 0.94 # 准确率下限
    gsm8k_accept_length_thres = 2.7 # 平均接受长度下限
    gsm8k_num_questions = 500 # 测试题目数量
    gsm8k_num_threads = 500 # 并发线程数
    gsm8k_num_shots = 20 # few-shot 示例数

    # 单请求速度测试默认阈值，子类通过 `bs_1_speed_thres` 覆盖
    accept_length_thres = 2.7


class DsaMtpServerBase(CustomTestCase):
    """
    管理 8-GPU 服务器的生命周期。
    子类只需设置 `model` 等类属性，无需重复编写启动/停止逻辑。
    """
    base_url = DEFAULT_URL_FOR_TEST

    # 以下属性由子类覆盖
    model: str = ""
    mem_fraction_static: float = 0.7
    enable_dp_attention: bool = False

    # EAGLE MTP 参数（所有变体一致）
    speculative_algorithm: str = "EAGLE"
    speculative_num_steps: int = 3
    speculative_eagle_topk: int = 1
    speculative_num_draft_tokens: int = 4

    @classmethod
    def get_server_args(cls):
        """根据子类属性动态生成服务器启动参数."""
        assert cls.model, f"{cls.__name__} 必须设置 `model`"
        args = ["--trust-remote-code", "--tp", "8"]
        if cls.enable_dp_attention:
            args += ["--dp", "8", "--enable-dp-attention"]
        args += [
            "--speculative-algorithm", cls.speculative_algorithm,
            "--speculative-num-steps", str(cls.speculative_num_steps),
            "--speculative-eagle-topk", str(cls.speculative_eagle_topk),
            "--speculative-num-draft-tokens", str(cls.speculative_num_draft_tokens),
            "--mem-frac", str(cls.mem_fraction_static),
            "--model-loader-extra-config",
            '{"enable_multithread_load": true, "num_threads": 64}',
        ]
        return args
```

```
    @classmethod
    def setUpClass(cls):
        """启动目标模型服务器."""
        cls.process = popen_launch_server(
            cls.model,
            cls.base_url,
            timeout=DEFAULT_TIMEOUT_FOR_SERVER_LAUNCH,
            other_args=cls.get_server_args(),
        )

    @classmethod
    def tearDownClass(cls):
        """强制终止服务器进程."""
        kill_process_tree(cls.process.pid)
```

评论区精华

无实质技术讨论，PR 由作者独立完成并合并。仅有的评论是作者 /rerun-test 命令和 CI 报告，未涉及设计争议。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险：拆分后测试覆盖可能遗漏（如某些配置组合未在独立文件中体现）；但通过再次运行全部 4 个测试验证通过。另外，fixture 提取假设所有变体共享相同服务器参数，若有新增变体需确保不会破坏该模式。总体风险较低，因为测试逻辑未改变，仅重构结构。
- 影响：影响范围限于测试组织、CI 调度和开发维护。对使用 SGLang 推理的最终用户无影响。对团队而言，DSAMTP 回归测试从硬编码单文件变为模块化设计，添加新模型变体只需新建一个文件并指定参数，降低了维护成本。CI 方面，每个文件独立超时，避免了因一个变体超时而阻塞其他变体的评估。
- 风险标记：测试覆盖完整性，CI 配置同步

关联脉络

- 暂无明显关联 PR