

PR #28098 完整报告

sgl-project/sglang

Add DeepSeek V4 MTP acceptance length checks

合并时间: 2026-06-13 09:53

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28098>

执行摘要

- 一句话: 为 DeepSeek V4 添加 MTP 接受长度检查
- 推荐动作: 值得精读的文件是 `sglang/test/kits/spec_decoding_kit.py` (如果其定义可见), 以理解 Mixin 的实现方式。本 PR 的批量混入模式可作为后续为其他模型添加性能断言的样板。建议维护一份阈值基线文档, 定期验证阈值的有效性。

功能与动机

确保 DeepSeek V4 的多步推测解码 (MTP) 在 CI 运行中持续维持预期性能, 避免因模型、算子或配置变更导致的性能退化被忽略。PR body 明确说明: `add a reusable MTP acceptance-length/speed smoke test mixin` 并 `set initial acc_length thresholds`。

实现拆解

1. 导入可复用混入类: 在 6 个测试文件中添加 `from sglang.test.kits.spec_decoding_kit import SpecDecodingMixin` 导入语句。
2. 混入测试类: 在每个测试类的继承列表中将 `SpecDecodingMixin` 置于第一位 (或适当位置), 例如 `class TestDSV4FlashFP4B200(SpecDecodingMixin, BasicDecodeCorrectnessMixin, ...)`。
3. 设置阈值属性: 在每个测试类中定义 `accept_length_thres` (接受长度阈值) 和 `bs_1_speed_thres` (batch=1 速度阈值) 两个类属性, 根据推测步数调整——3-step 配置 (如 LowLatency 配方) 阈值设为 2.6~2.8, 1-step 配置 (Balanced 配方) 阈值设为 1.8。
4. 调整解聚测试: 在 `test/registered/disaggregation/test_disaggregation_dsv4.py` 中, 除了混入 `SpecDecodingMixin` 并设置阈值外, 还移除了父类已定义的冗余属性 `gsm8k_accuracy_thres`, 并重新对齐继承顺序。
5. CI 注册确认: 所有受影响的测试文件均已通过 `register_cuda_ci` 注册到 CI 流水线, 本次变更不改变其 `stage` 和 `runner_config`, 确保新断言在现有 CI 框架中生效。

关键文件:

- `test/registered/disaggregation/test_disaggregation_dsv4.py` (模块 解聚测试; 类别 `test`; 类型 `test-coverage`; 符号 `TestDisaggregationDSV4`): 该文件新增了 `SpecDecodingMixin` 的使用, 调整了测试类继承结构, 并删除了冗余属性, 是变更最综合的文件。

- test/registered/models_e2e/test_deepseek_v4_flash_fp4_b200.py (模块 B200 FP4; 类别 test; 类型 test-coverage; 符号 TestDSV4FlashFP4B200, TestDSV4FlashFP4B200Balanced) : 为两个测试类 (LowLatency 和 Balanced) 添加了 SpecDecodingMixin 及对应阈值。
- test/registered/models_e2e/test_deepseek_v4_flash_fp4_h200.py (模块 H200 FP4; 类别 test; 类型 test-coverage; 符号 TestDSV4FlashFP4H200, TestDSV4FlashFP4H200FlashInferCutlass) : 为两个测试类 (Marlin 和 FlashInferCutlass) 添加了 SpecDecodingMixin 及对应阈值。
- test/registered/models_e2e/test_deepseek_v4_flash_fp4_megamoe_b200.py (模块 MegaMoE B200; 类别 test; 类型 test-coverage; 符号 TestDSV4FlashFP4B200W4A8MegaMoE, TestDSV4FlashFP4B200W4A4MegaMoE) : 为两个 MegaMoE 测试类添加了 SpecDecodingMixin 及对应阈值。
- test/registered/models_e2e/test_deepseek_v4_flash_fp8_h200.py (模块 H200 FP8; 类别 test; 类型 test-coverage; 符号 TestDSV4FlashFP8H200) : 为单个测试类添加了 SpecDecodingMixin 及阈值。
- test/registered/cp/test_deepseek_v4_flash_fp4_b200_cp.py (模块 B200 CP; 类别 test; 类型 test-coverage; 符号 TestDSV4FlashFP4B200Balanced_CP, TestDSV4FlashFP4B200Balanced_CP_NonDeepEP) : 为两个 Context Parallel 测试类添加了 SpecDecodingMixin 及阈值。

关键符号: TestDisaggregationDSV4

关键源码片段

test/registered/disaggregation/test_disaggregation_dsv4.py

该文件新增了 SpecDecodingMixin 的使用, 调整了测试类继承结构, 并删除了冗余属性, 是变更最综合的文件。

```
import unittest

from sglang.test.ci.ci_register import register_cuda_ci
from sglang.test.kits.eval_accuracy_kit import GSM8KMixin
from sglang.test.kits.spec_decoding_kit import SpecDecodingMixin # 新增: 接受长度与速度混入
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=500, stage="base-c", runner_config="deepep-8-gpu-h200")

# ... 常量定义省略 ...
```

```
# 变更后：混入 SpecDecodingMixin，设置阈值
class TestDisaggregationDSV4(SpecDecodingMixin, PDDisaggregationServerBase, GSM8KMixin):
    gsm8k_accuracy_thres = 0.93
    accept_length_thres = 1.8 # 1-step MTP 接受长度阈值（原 suggestion 从 1.6 上调整至 1.8）
    bs_1_speed_thres = 140 # batch=1 速度阈值（tok/s）

    @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()
```

评论区精华

Review 中无外部争议。PR 作者（Fridge003）自行在 6 处 Review 评论中建议将 1-step 测试的 `accept_length_thres` 从初始值 1.6 上调至 1.8，并主动提交 suggestion。最终合并时所有 1-step 测试均采用 1.8 阈值，表明作者基于初步运行结果进行了细粒度校准。

- 调整 1-step MTP 接受长度阈值 (testing): 作者自行采纳建议，合并时所有 1-step 测试阈值均为 1.8。

风险与影响

- 风险：风险较低，但阈值设定存在两个潜在问题：1) 阈值可能过于严格或宽松，导致 CI 频繁失败（假阳性）或掩盖真实回归（假阴性）；2) 当前阈值基于特定硬件（H200/B200）和软件版本测得，环境升级后可能需要重新校准。此外，测试覆盖率不包括所有可能的推测参数组合（如不同 topk 或 draft token 数），因此阈值可能无法反映全貌。
- 影响：直接影响：对 DeepSeek V4 相关的 6 个 CI 测试增加了性能断言，确保 MTP 接受长度和单 batch 速度不退化。间接影响：为项目中其他模型或模拟场景提供了 SpecDecodingMixin 可复用模式，降低后续添加类似检查的成本。对用户无感知，对内部 CI 稳定性有正面作用。
- 风险标记：阈值需长期校准，测试覆盖不全所有配置

关联脉络

- PR #28026 [Bugfix][Spec] Fix multi-layer EAGLE DRAFT_EXTEND_V2 attn-TP logprob metadata capture: 同为 DeepSeek V4 推测解码 bug 修复，本 PR 为其提供性能回归检测。
- PR #28093 [Spec] Move draft-extend prep to EagleDraftWorkerBase; unify prepare_for_* names: 重构推测解码基类，本 PR 可验证重构后性能是否满足阈值。
- PR #28096 [Spec] Fix EagleDraftWorker draft-extend attn backend assignment: 修复注意力后端问题，本 PR 的 smoke test 可以捕获类似回归。