# PR #34145 完整报告

sgl-project/sglang

[CI] Gate Kimi-K3 acceptance length on the GSM8K average

合并时间：2026-08-09 11:28

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34145

## 执行摘要

- 一句话：K3 接受长度阈值改为 GSM8K 平均，放宽单提示阈值
- 推荐动作：该 PR 值得精读，尤其是对负责 CI 测试阈值设计和维护的工程师。它展示了如何识别并修复由单样本方差导致的测试不稳定，以及如何将阈值建立在更稳定的统计量（GSM8K 平均）上。关键设计决策包括：区分“稳态吞吐”与“端到端速度”、避免让正确性修复被不相关指标绑架、以及用历史数据支撑安全边际。

## 功能与动机

PR body 中说明：定时运行的 *base-c-test-8-gpu-b300* 任务失败，*bs_1_speed* 断言 5.308 不大于 6.6。作者分析原因是单条贪婪提示在 775 token 处提前到达 EOS，而 *accept_length = completion_tokens / spec_verify_ct*，*speed = completion_tokens / e2e_latency*，都极度依赖生成长度。与 #33764 之前完整运行 2048 token 相比，同一 token 范围内的接受长度实际上持平甚至略高。GSM8K 在同一 job 中通过（0.98），说明投机解码本身没有退化，只是数值扰动改变了 EOS 位置。因此需要将阈值建立在稳定的 200 题 GSM8K 平均之上，而不是单条提示的瞬间读数。

## 实现拆解

本 PR 仅修改一个文件：test/registered/models_e2e/test_kimi_k3_b300.py。

1. 新增 GSM8K 接受长度阈值：引入类属性 gsm8k_accept_length_thres = 4.5，利用 GSM8KMixin 中现有的 200 题 GSM8K 评测流程，在评测完成后通过 avg_spec_accept_length 指标（投机接受长度平均值）进行门控。作者给出的历史数据为 4.75 - 4.86，4.5 留下约 6% 的安全边际。
2. 放宽单提示阈值：将 accept_length_thres 从 6.6 降到 4.0，bs_1_speed_thres 从 440 降到 300。理由是这两个指标都随单条提示生成长度变化，而 bs_1_speed 是端到端速度（包含 launch、TTFT），与服务器稳态解码速率（约 428-452 token/s）存在系统性差距。作者实测当前分支上 *acc_length* 为 5.31、*speed* 为 343.99 / 346.20，因此新阈值作为粗粒度回归守卫。
3. 增加注释说明：在阈值前添加多行注释，解释为什么基于 GSM8K 平均而非单提示，以及速度指标与稳态速率的关系，便于后续维护者理解。
4. 验证：PR 提交后通过 /rerun-test 触发测试，第二次运行即通过（workflow run 31291945395 显示 ）。

关键文件：

- test/registered/models_e2e/test_kimi_k3_b300.py（模块 K3 测试；类别 test；类型 test-coverage）：Kimi-K3 B300 低延迟 CI 测试的阈值定义，通过将接受长度门控迁移到 GSM8K 平均并放宽单提示阈值，修复了偶发失败。

关键符号：TestKimiK3B300LowLatency

## 关键源码片段

### test/registered/models_e2e/test_kimi_k3_b300.py

Kimi-K3 B300 低延迟 CI 测试的阈值定义，通过将接受长度门控迁移到 GSM8K 平均并放宽单提示阈值，修复了偶发失败。

```
class TestKimiK3B300LowLatency(GSM8KMixin, SpecDecodingMixin, CustomTestCase):
    """TP8 Low Latency recipe with DSPARK linear ReplaySSM speculation."""

    gsm8k_score_threshold = 0.95
    gsm8k_num_examples = 200
    # 接受长度门控放在 GSM8K 平均上，而不是 test_bs_1_speed 的单条提示上：
    # 一个 200 题的均值在数值扰动导致单条贪婪提示提前到达 EOS 时依然保持稳定。
    gsm8k_accept_length_thres = 4.5
    # 以下两个单提示阈值仅作粗粒度回归守卫：它们都随该条提示的生成长度变化，
    # 而且 speed 是端到端指标（包含 launch 与
    TTFT），会系统性低于服务器日志中的稳态解码速率。
    accept_length_thres = 4.0
    bs_1_speed_thres = 300
```

## 评论区精华

该 PR 没有正式的 review 评论，作者在 PR body 中完成了主要论证：

- 作者论证了单提示指标不稳定的根源：acc_length 和 speed 都是基于单条贪婪提示的比值，当数值变化（如 #33764 的 fp32 路由 logits）使 EOS 提前时，即使投机解码行为完全不变，指标也会大幅下降。对比同一 token 范围内的接受长度（5.22 vs 5.38）即可证明这一点。
- 作者明确区分了“服务器稳态解码速率”与“端到端 speed”两个量，指出 bs_1_speed_thres = 440 与测试实际测量的端到端值（约 346 token/s）不是同一个量，是先前阈值设置的根本性错误。
- 关于是否回退 #33764，作者明确拒绝：那是正确性修复（路由 logits 精度提升），GSM8K 分数与 GSM8K 上的接受长度均未改变，不应因单提示的 EOS 位置变化而回退。

Issue 评论中仅包含 /rerun-test 的自动化结果，无实质讨论。

- 暂无高价值评论线程

## 风险与影响

- 风险：主要风险集中在阈值放宽可能导致的回归漏检：

1. accept_length_thres = 4.0：相比 6.6 降低了约 40%，如果未来出现真正的投机解码回归（例如接受长度掉到 4 以下），仍能捕捉，但 4.0 - 6.6 之间的性能退化会被忽略。不过 GSM8K 平均阈值 4.5 能弥补部分敏感性，因为 GSM8K 平均对单提示波动不敏感。
2. bs_1_speed_thres = 300：实测约 344-346，留了约 13-15% 余量。若未来出现启动延迟或首 token 延迟大幅增加，可能导致测试通过但实际性能下降。由于该指标是端到端速度，稳态解码速率下降 20% 左右可能仍然超过 300。
3. 依赖 GSM8K 平均：gsm8k_accept_length_thres = 4.5 依赖 GSM8K 评测的稳定性。历史数据显示该值在 4.75 - 4.86 之间波动，但若未来模型或推理路径发生数值变化导致 GSM8K 上的接受长度整体下降（但 GSM8K 分数仍 ⩾ 0.95），可能出现误报。不过 4.5 与历史值差距约 6%，风险可控。
4. 本 PR 不改动共享测试 kit（SpecDecodingMixin、GSM8KMixin），因此不会影响其他使用这些 mixin 的测试类。 - 影响：对 CI 的影响：修复了 base-c-test-8-gpu-b300 定时任务的偶发失败，消除因单提示 EOS 位置变化导致的误报，提高 CI 稳定性。对开发者的影响：降低了对追求正确的数值改动（如 #33764）的敏感度，让开发者不必担心正确的精度修复会破坏无关的性能测试。对性能回归检测的影响：新阈值仍能捕捉显著退化（如接受长度低于 4.0 或端到端速度低于 300），但灵敏度有所下降。该 PR 仅影响 Kimi-K3 B300 测试，其他测试类不受影响。 - 风险标记：测试阈值放宽可能降低回归灵敏度，依赖 GSM8K 平均稳定性，单提示阈值失去精细监控能力

## 关联脉络

- PR #33764 Fix the router GEMM inaccuracy when using _front_w in Kimi-K3: 该 PR 将 Kimi-K3 路由 GEMM 输出提升到 fp32，是本 PR 所面对的单提示 EOS 位置变化的直接原因。
- PR #34089 [CI] Add Kimi-K3 low-latency performance check: 该 PR 创建了 test_kimi_k3_b300.py 测试文件，本 PR 是对其阈值策略的修正。