PR #25940 完整报告

sgl-project/sglang

[SPEC] feat: add adaptive speculative decoding metrics

合并时间：2026-06-02 04:53

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25940

执行摘要

- 一句话：为自适应推测解码添加 Prometheus 指标
- 推荐动作：该 PR 设计简洁清晰，适合作为如何为动态配置添加可观测性的参考。推荐在以下场景精读：需要为自适应或动态调整的参数添加指标暴露时；理解 metrics_reporter 与 MetricsCollector 如何协作时。

功能与动机

自适应推测解码在运行时根据当前负载动态调整推测步数和草稿 token 数，但此前缺乏对应的监控指标，用户无法直观了解当前使用的推测配置。PR body 明确说明目标为「add metrics for adaptive speculative decoding」。

实现拆解

1. 在 metrics_reporter.py 中添加 _active_spec_config_snapshot() 方法，通过 draft_worker 对象获取当前活跃的 speculative_num_steps 和 speculative_num_draft_tokens；若 draft_worker 为 None 则返回 0。
2. 在 report_decode_stats() 中，当 current_scheduler_metrics_enabled 为真时调用 snapshot，将结果写入 self.stats（SchedulerStats 实例）。
3. 在 SchedulerStats dataclass 中新增 spec_num_steps: int = 0 和 spec_num_draft_tokens: int = 0 字段，用于承载指标值。
4. 在 MetricsCollector 的 __init__ 中创建 sglang:spec_num_steps 和 sglang:spec_num_draft_tokens 两个 Gauge，并在 log_stats() 中推送数值。
5. 在 test_adaptive_speculative.py 中新增 test_adaptive_metrics_exposed 测试，启动服务器时添加 --enable-metrics，驱动自适应 upshift 后通过 /metrics 端点 scrape 新指标并断言值正确。
6. 在 production_metrics.mdx 中补充两个新指标的输出示例。

关键文件：

- python/sglang/srt/managers/scheduler_components/metrics_reporter.py（模块 调度器；类别 source；类型 core-logic；符号 _active_spec_config_snapshot）：核心实现文件，新增 _active_spec_config_snapshot 方法并在 report_decode_stats 中调用，完成配置读取与统计注入。

- test/registered/spec/eagle/test_adaptive_speculative.py（模块 推测测试；类别 test；类型 test-coverage；符号 _scrape_metric, test_adaptive_metrics_exposed）：新增测试方法 test_adaptive_metrics_exposed 和辅助方法 _scrape_metric，验证指标可被正确暴露和 scrape。同时在 setUpClass 中添加 --enable-metrics 标志。
- python/sglang/srt/observability/metrics_collector.py（模块 可观测性；类别 source；类型 core-logic）：在 SchedulerStats dataclass 中增加 spec_num_steps 和 spec_num_draft_tokens 字段，并在 MetricsCollector 中注册对应的 Prometheus Gauge。
- docs_new/docs/references/production_metrics.mdx（模块 文档；类别 other；类型 documentation）：文档更新，添加新指标的 Prometheus 文本输出示例。

关键符号：_active_spec_config_snapshot, report_decode_stats, log_stats, _scrape_metric, test_adaptive_metrics_exposed

关键源码片段

python/sglang/srt/managers/scheduler_components/metrics_reporter.py

核心实现文件，新增 _active_spec_config_snapshot 方法并在 report_decode_stats 中调用，完成配置读取与统计注入。

```
def _active_spec_config_snapshot(self) -> dict[str, int]:
    '''读取当前活跃的推测解码配置（步数和草稿 token 数）'''
    draft_worker = self.scheduler.draft_worker
    if draft_worker is None:
        return {'num_steps': 0, 'num_draft_tokens': 0}
    # 若 draft_worker 未直接暴露属性，则回退到 server_args
    server_args = self.scheduler.server_args
    num_steps = getattr(draft_worker, 'speculative_num_steps',
                        server_args.speculative_num_steps)
    num_draft_tokens = getattr(draft_worker, 'speculative_num_draft_tokens',
                               server_args.speculative_num_draft_tokens)
    return {'num_steps': num_steps or 0,
            'num_draft_tokens': num_draft_tokens or 0}

# 在 report_decode_stats() 中：
if self.current_scheduler_metrics_enabled:
    spec_snapshot = self._active_spec_config_snapshot()
    spec_num_steps = spec_snapshot['num_steps']
    spec_num_draft_tokens = spec_snapshot['num_draft_tokens']

# 随后赋值给 stats
self.stats.spec_num_steps = spec_num_steps
self.stats.spec_num_draft_tokens = spec_num_draft_tokens
```

评论区精华

reviewer Qiaolin-Yu 提出 nit：使用 current_scheduler_metrics_enabled 替代 enable_metrics 条件判断更合适。作者 alphabetc1 回复 'good idea, updated'。最终代码采

用了 current_scheduler_metrics_enabled，确保只在开启 metrics 时才执行 snapshot 抓取，避免无谓开销。

- 使用 current_scheduler_metrics_enabled 替代 enable_metrics (design): 作者 alphabetc1 采纳建议，将条件从句改为 current_scheduler_metrics_enabled。

风险与影响

- 风险：该 PR 新增的指标收集逻辑对核心推测解码路径无影响，风险较低。潜在风险：
  - _active_spec_config_snapshot 通过 getattr 回退到 server_args，保证了兼容性，但若 draft_worker 对象的属性名未来变化需同步更新。
  - 每次 decode 统计周期（由 decode_log_interval 控制）调用一次 snapshot，额外开销可忽略。
  - 指标通过 Prometheus 多进程模式暴露，需确保 multiprocess_mode='mostrecent' 在子进程中正确工作；现有架构已为此设计。
  - 影响：对用户：新增两个 Prometheus 指标，可监控自适应推测解码的当前 num_steps 和 num_draft_tokens。对系统：无性能回归，仅增加极轻量的指标采集。对团队：新增的测试和文档降低了维护成本。影响范围限定于自适应推测解码场景。
  - 风险标记：新增 Prometheus 指标，依赖 draft_worker 动态属性

关联脉络

- PR #26866 Support spec v2 tree drafting (eagle topk>1) with page_size==1: 同属 speculative decoding 功能线，本次 PR 的指标可以监控该 PR 引入的 draft tokens 变化。
- PR #26424 [Perf][Spec Decoding] Skip cat/topk/sort/gather in draft_forward for topk=1: 同属 speculative decoding 性能优化，本次 PR 的指标与 draft 配置紧密相关。