PR #23530 完整报告

sgl-project/sglang

[Spec] Fix `spec_accept_rate` and unify `accept`/`draft` naming

合并时间：2026-04-29 05:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23530

执行摘要

- 一句话：修复 spec_accept_rate 偏差并统一命名约定
- 推荐动作：建议阅读此 PR，它修复了 spec metrics 的系统性偏差并建立了清晰的命名约定，有助于理解 sglang speculative decoding 的度量设计。但需关注测试覆盖是否充分，建议在后续 PR 中补充针对偏差修复的单元测试。

功能与动机

PR body 指出旧公式 (num_accepted_drafts + bs) / (bs * num_draft_tokens) 存在两个错误：分子误加了 bonus token，分母计数偏大，导致中间值向上偏差；同时内部命名混杂 accept 与 draft 语义，容易误将 bonus token 计入 draft-only 计数。修复后严格区分 draft-only 与包含 bonus 的指标，并统一命名约定。

实现拆解

1. 核心公式修复（scheduler_metrics_mixin.py）：update_spec_metrics 参数从 num_accepted_tokens 改为 num_accepted_drafts，累加时不再额外加 bs；report_decode_stats 中 acceptance rate 计算改为 num_accepted_drafts / (spec_num_forward_ct * draft_per_round)，消除偏差。
2. 内部字段重命名：GenerationBatchResult.num_accepted_tokens → num_accepted_drafts，Req.spec_accepted_tokens → spec_accepted_drafts，BatchTokenIDOutput.spec_accepted_tokens → spec_accepted_drafts，以及各类 worker 和 info 类中的对应参数。
3. 外部 meta_info 键重命名（tokenizer_manager.py）：spec_accept_token_num → spec_accepted_drafts，spec_draft_token_num → spec_proposed_drafts，保持 spec_accept_rate 和 spec_accept_length 不变。
4. 数据传输对象更新（io_struct.py）：SpeculativeDecodingMetricsMixin 中字段名变更，同时改进 SpeculativeMetrics.accept_length 的文档描述。
5. 各 speculative worker 对齐：eagle_worker.py、ngram_worker.py、dflash_worker.py、multi_layer_eagle_worker.py、ngram_info.py 同步更新局部变量和函数参数名。
6. Prometheus 指标描述更新（metrics_collector.py）：sglang:spec_accept_length 指标描述澄清为包含 bonus token。

7. 测试：无新增测试文件，但 PR body 标记了需要运行 test_eagle_infer.py 和 test_ngram_infer.py，CI 已通过相关 MTP 测试。

关键文件：

- python/sglang/srt/observability/scheduler_metrics_mixin.py（模块 调度监控；类别 source；类型 core-logic；符号 update_spec_metrics, report_decode_stats）：核心变更文件：修复 spec_accept_rate 计算公式（update_spec_metrics 和 report_decode_stats），并重构接受率与接受长度的计算逻辑。
- python/sglang/srt/managers/tokenizer_manager.py（模块 令牌管理；类别 source；类型 core-logic；符号 _calculate_spec_decoding_metrics）：更新 meta_info 键名，将 spec_accept_token_num 和 spec_draft_token_num 重命名为 spec_accepted_drafts 和 spec_proposed_drafts，影响所有请求结束后返回的指标字典。
- python/sglang/srt/managers/scheduler_output_processor_mixin.py（模块 输出处理；类别 source；类型 core-logic；符号 _resolve_spec_overlap_token_ids, process_batch_result_decode, stream_output_generation）：调整 result 字段引用（num_accepted_tokens → num_accepted_drafts）以及 stream_output_generation 中 spec 字段组装，是连接 worker 输出与 scheduler 度量的关键桥梁。
- python/sglang/srt/managers/io_struct.py（模块 数据结构；类别 source；类型 core-logic；符号 SpeculativeDecodingMetricsMixin, SpeculativeMetrics）：定义数据传输结构：SpeculativeDecodingMetricsMixin 和 SpeculativeMetrics 等 dataclass 字段名和描述更新，确保 IPC 通信层使用新命名。

关键符号：update_spec_metrics, report_decode_stats, _calculate_spec_decoding_metrics, _resolve_spec_overlap_token_ids, process_batch_result_decode

关键源码片段

python/sglang/srt/observability/scheduler_metrics_mixin.py

核心变更文件：修复 spec_accept_rate 计算公式（update_spec_metrics 和 report_decode_stats），并重构接受率与接受长度的计算逻辑。

```
# scheduler_metrics_mixin.py (head)
def update_spec_metrics(self: Scheduler, bs: int, num_accepted_drafts: int):
    # 每批次的 spec 计数器累加（不含 bonus token）
    self.spec_num_accepted_tokens += num_accepted_drafts + bs # 包含 bonus 用于 accept_
    length
    self.spec_num_forward_ct += bs
    # Bonus tokens 在其他地方更新（process_batch_result_decode 中加 bs）
    self.num_generated_tokens += num_accepted_drafts

def report_decode_stats(self: Scheduler, ..., num_accepted_drafts: int = 0):
    ...
    # 实时 token 计数：包括 bonus（即 num_accepted_drafts + 一个 bonus/req）
    decode_tokens = batch.batch_size() + num_accepted_drafts
    self.metrics_collector.increment_realtime_tokens(decode_tokens=decode_tokens)
```

```
    ...
    # log interval 结束时计算并重置
    if self.spec_num_forward_ct > 0:
        # 平均接受长度（含 bonus）
        spec_accept_length = self.spec_num_accepted_tokens / self.spec_num_forward_ct
        # 纯 draft 接受率
        draft_per_round = (self.server_args.speculative_num_draft_tokens - 1)
        total_draft_tokens = self.spec_num_forward_ct * draft_per_round
        # 旧公式曾用 total_accepted / total_draft，现在用专属 drafts-only 计数
        num_accepted_drafts = self.spec_num_accepted_tokens - self.spec_num_forward_ct
        if total_draft_tokens > 0:
            spec_accept_rate = num_accepted_drafts / total_draft_tokens
        ...
        self.logger.info(f"spec_accept_rate={spec_accept_rate:.3f}, ...")
        # 重置计数器
        self.spec_num_accepted_tokens = 0
        self.spec_num_forward_ct = 0
```

python/sglang/srt/managers/tokenizer_manager.py

更新 meta_info 键名，将 spec_accept_token_num 和 spec_draft_token_num 重命名为 spec_accepted_drafts 和 spec_proposed_drafts，影响所有请求结束后返回的指标字典。

```
# tokenizer_manager.py (head)
def _calculate_spec_decoding_metrics(self, meta_info, recv_obj, i):
    if (
        hasattr(recv_obj, "spec_verify_ct")
        and recv_obj.spec_verify_ct[i] > 0
        and hasattr(recv_obj, "spec_accepted_drafts")
        and len(recv_obj.spec_accepted_drafts) > i
    ):
        # 每请求的提议 draft 总数：steps × ( 每步 draft 数 )
        all_drafts = recv_obj.spec_verify_ct[i] * (
            self.server_args.speculative_num_draft_tokens - 1
        )
        accepted_drafts = recv_obj.spec_accepted_drafts[i]

        if all_drafts > 0:
            # 纯 draft 接受率（无 bonus）
            meta_info["spec_accept_rate"] = accepted_drafts / all_drafts
            # 平均接受长度：包含 bonus（completion_tokens / steps）
            meta_info["spec_accept_length"] = (
                recv_obj.completion_tokens[i] / recv_obj.spec_verify_ct[i]
            )
            # 新键名
            meta_info["spec_accepted_drafts"] = accepted_drafts
            meta_info["spec_proposed_drafts"] = all_drafts
            meta_info["spec_verify_ct"] = recv_obj.spec_verify_ct[i]
```

评论区精华

未收到外部 review 评论。作者自行触发 CI 运行 test_deepseek_v3_mtp.py、test_step3p5_flash_chain_mtp.py 等 MTP 测试，所有目标测试通过。

- CI 测试覆盖 (testing)：目标测试通过，未暴露回归问题。

风险与影响

- 风险：
  1. Prometheus 指标不兼容：sglang:spec_accept_rate 的中间值变化，依赖该指标的告警或 dashboard 需重新校准。
  2. meta_info 键重命名：外部代码若使用 spec_accept_token_num 或 spec_draft_token_num 将失效，需迁移到新键名。
  3. 测试覆盖不足：无新增测试验证公式正确性，现有测试可能未覆盖非全接受场景。
  4. 无外部 review：作者自行 merge，可能遗漏边缘情况。主要风险集中在 scheduler_metrics_mixin.py 和 tokenizer_manager.py 的公式与键变更。 - 影响：影响所有使用 speculative decoding 的用户（Eagle、NGram、DFlash、MultiLayerEagle）。影响中等：修复了度量偏差，使 spec_accept_rate 更真实地反映 draft 接受率；命名统一降低了未来混淆风险；但消费 meta_info 或 Prometheus 指标的外部系统需要适配新键名。 - 风险标记：核心度量公式变更，meta_info 键不兼容，缺少测试覆盖，无外部 review

关联脉络

- PR #21694 fix: resolve tensor file overwrite between target and draft models: 同一 speculative decoding 模块的先前 bugfix，修改了 eagle_worker 等文件，本次命名统一可能与其中的变量命名有隐含交互。