PR #25689 完整报告

sgl-project/sglang

Add spec_verify_calls_total metric for speculative decoding

合并时间：2026-05-19 09:35

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25689

执行摘要

- 一句话：新增 spec_verify_calls_total 指标
- 推荐动作：值得精读，展示如何为系统增加可观测性指标，可参考此模式添加其他监控。

功能与动机

监控 speculative decoding 的验证调用次数，便于性能分析和调优。

实现拆解

1. 注册 Prometheus 计数器：在 TokenzierMetricsCollector.__init__ 中新增 Counter 指标 sglang:spec_verify_calls_total。
2. 修改收集点：在 observe_one_finished_request 方法签名中增加可选参数 spec_verify_ct，并在方法体内部根据 spec_verify_ct > 0 递增计数器。
3. 提取数据源：在 tokenizer_manager.py 的 collect_metrics 方法中，从 recv_obj 中安全提取 spec_verify_ct 字段并传递给 metrics collector。

关键文件：

- python/sglang/srt/observability/metrics_collector.py（模块 可观测性；类别 source；类型 core-logic；符号 TokenizerMetricsCollector.init, TokenizerMetricsCollector.observe_one_finished_request）：注册并实现新的 Prometheus 计数器。
- python/sglang/srt/managers/tokenizer_manager.py（模块 调度器；类别 source；类型 core-logic；符号 TokenizerManager.collect_metrics）：提取 spec_verify_ct 数据并传递给 metrics collector。

关键符号：TokenizerMetricsCollector.init, TokenizerMetricsCollector.observe_one_finished_request, TokenizerManager.collect_metrics

关键源码片段

python/sglang/srt/observability/metrics_collector.py

注册并实现新的 Prometheus 计数器。

```
# 在 __init__ 中注册新的 Prometheus 计数器
self.spec_verify_calls_total = Counter(
```

```
        name="sglang:spec_verify_calls_total",
        documentation="Number of speculative decoding verification calls.",
        labelnames=labels.keys(),
    )

    # 修改 observe_one_finished_request 签名，增加 spec_verify_ct 参数
    # 默认值为 0，确保向后兼容
    def observe_one_finished_request(
        self,
        labels: Dict[str, str],
        prompt_tokens: int,
        generation_tokens: int,
        cached_tokens: int,
        e2e_latency: float,
        has_grammar: bool,
        cached_tokens_details: Optional[Dict[str, Any]] = None,
        spec_verify_ct: int = 0, # 新增参数
    ):
        self.prompt_tokens_total.labels(**labels).inc(prompt_tokens)
        self.generation_tokens_total.labels(**labels).inc(generation_tokens)
        # 只在 spec_verify_ct > 0 时递增计数器，避免无 spec 时增加开销
        if spec_verify_ct > 0:
            self.spec_verify_calls_total.labels(**labels).inc(spec_verify_ct)
        # ... 其余代码保持不变
```

python/sglang/srt/managers/tokenizer_manager.py

提取 spec_verify_ct 数据并传递给 metrics collector。

```
    # 在 collect_metrics 方法中，在处理完 cached_tokens_details 之后
    # 安全地从 recv_obj 提取 spec_verify_ct 字段
    spec_verify_ct = (
        recv_obj.spec_verify_ct[i] # 尝试按 index 取值
        if hasattr(recv_obj, "spec_verify_ct") # 确保 recv_obj 有该属性
        and recv_obj.spec_verify_ct # 确保属性不为 None
        and len(recv_obj.spec_verify_ct) > i # 确保索引不越界
        else 0 # 安全 fallback
    )

    # 将提取的值传递给 observe_one_finished_request
    self.metrics_collector.observe_one_finished_request(
        labels,
        recv_obj.prompt_tokens[i],
        completion_tokens,
        recv_obj.cached_tokens[i],
        state.time_stats.get_e2e_latency(),
        self._request_has_grammar(state.obj),
        cached_tokens_details,
        spec_verify_ct=spec_verify_ct, # 新增参数
    )
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：变更极小，无风险。新增计数器和可选参数均向后兼容。
- 影响：影响范围小，仅增加一个 Prometheus 指标，对已有流程无影响。
- 风险标记：无

关联脉络

- PR #25566 [Spec] fold can_run_cuda_graph into EagleVerifyOutput; drop dead extend-after-decode check: 同属 speculative decoding 模块，涉及 verify 相关逻辑。
- PR #25489 Support draft extend cuda graph for tokenspeed_mla attention backend: 同为 speculative decoding 功能扩展。