PR #29207 完整报告

sgl-project/sglang

Add scheduler metrics extension hooks

合并时间：2026-06-25 06:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29207

执行摘要

- 一句话：添加调度器指标扩展钩子，支持模型特定 SoL 统计
- 推荐动作：该 PR 值得精读，它展示了如何通过钩子模式为内部系统添加可扩展性，同时注意保留必要数据供异步处理器使用。关注点在 MetricsReporter 中的覆写方法和 ScheduleBatch.copy 的数据保留策略。

功能与动机

为了更精确地报告预填充阶段的性能指标，避免壁钟时间与 GPU 实际执行时间不一致导致的偏差，需要为模型子类提供扩展点，使其能根据批次中的请求级别数据（如 extend/prefix 长度）计算精确的 attention 操作对数量，并输出 SoL 百分比。同时，需要区分 target_verify 和 extend 阶段的定时器范围，以正确归因性能数据。

实现拆解

1. 在 ScheduleBatch.copy 方法中，添加 extend_lens 和 prefix_lens 的深拷贝（schedule_batch.py），使得在异步指标报告时保留原始请求的 extend/prefix 长度。
2. 修改 MetricsReporter._estimate_prefill_perf 接口，改为接收 batch 对象并从 batch.extend_lens 计算 token 总数（metrics_reporter.py），从而利用拷贝时保存的精确扩展长度。
3. 在 MetricsReporter 中添加 _prefill_sol_suffix 和 _decode_sol_suffix 钩子方法，默认返回空字符串（metrics_reporter.py），子类可覆写以提供模型特定的 SoL 百分比。
4. 在 report_prefill_stats 中，如果 _prefill_sol_suffix 返回非空字符串，则直接附加到日志行，并跳过估算 TFLOPS 的计算；否则仍使用原有估算逻辑。在 report_decode_stats 中类似调用 _decode_sol_suffix。
5. 在 model_runner.py 和 eager_runner.py 的 device_timer 包装中，根据 forward_batch.forward_mode.is_target_verify() 区分 category 为 "target_verify" 或 "extend"，使得定时器统计能正确区分目标验证和常规扩展阶段。

关键文件：

- python/sglang/srt/managers/scheduler_components/metrics_reporter.py（模块 指标报告；类别 source；类型 core-logic；符号 _estimate_prefill_perf, _prefill_sol_suffix, _decode_sol_suffix）：核心变更文件：修改 _estimate_prefill_perf 接口，添加 _prefill_sol_suffix 和 _decode_sol_suffix 钩子，并在报告函数中使用它们

- python/sglang/srt/managers/schedule_batch.py（模块 调度器；类别 source；类型 data-contract；符号 copy）：在 ScheduleBatch.copy 中添加 extend_lens 和 prefix_lens 的快照，确保延迟指标报告能获取原始请求的扩展 / 前缀长度
- python/sglang/srt/model_executor/model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 _forward_raw）：在 prefill cuda graph 路径中将 device_timer category 从固定的 "extend" 改为根据 forward_mode 判断 "target_verify" 或 "extend"
- python/sglang/srt/model_executor/runner/eager_runner.py（模块 运行器；类别 source；类型 data-contract；符号 _execute_extend）：在 eager 模式中类似修改 device_timer category，以区分 target_verify 和 extend

关键符号：_estimate_prefill_perf, _prefill_sol_suffix, _decode_sol_suffix, copy (ScheduleBatch), _forward_raw, _execute_extend

关键源码片段

python/sglang/srt/managers/schedule_batch.py

在 ScheduleBatch.copy 中添加 extend_lens 和 prefix_lens 的快照，确保延迟指标报告能获取原始请求的扩展 / 前缀长度

```
# 在 ScheduleBatch.copy 方法中
return ScheduleBatch(
    reqs=self.reqs[:],
    # 保留 extend_lens 和 prefix_lens 的快照
    # 使得在原始 batch 被修改后，延迟指标报告仍可读取这些值
    extend_lens=self.extend_lens[:] if self.extend_lens is not None else None,
    prefix_lens=self.prefix_lens[:] if self.prefix_lens is not None else None,
    # 其他字段不变 ...
    req_to_token_pool=self.req_to_token_pool,
    # ...
)
```

python/sglang/srt/model_executor/model_runner.py

在 prefill cuda graph 路径中将 device_timer category 从固定的 "extend" 改为根据 forward_mode 判断 "target_verify" 或 "extend"

```
# 在 ModelRunner._forward_raw 中
elif (
    forward_batch.forward_mode.is_extend(include_draft_extend_v2=True)
    and not isinstance(self.prefill_cuda_graph_runner, EagerRunner)
    and self.prefill_cuda_graph_runner is not None
    and self.prefill_cuda_graph_runner.can_run_graph(forward_batch)
    and get_cp_strategy() is None
):
    # 区分 target_verify 和 extend
    category = (
        "target_verify"
        if forward_batch.forward_mode.is_target_verify()
```

```
        else "extend"
    )
    # ...
    ctx = (
        self.device_timer.wrap(metadata={"category": category})
        if self.device_timer
        else contextlib.nullcontext()
    )
    with ctx:
        ret = self.prefill_cuda_graph_runner.execute(
            forward_batch, **kwargs
        )
```

python/sglang/srt/model_executor/runner/eager_runner.py

在 eager 模式中类似修改 device_timer category，以区分 target_verify 和 extend

```
# 在 EagerRunner._execute_extend 中
category = (
    "target_verify"
    if forward_batch.forward_mode.is_target_verify()
    else "extend"
)
ctx = (
    model_runner.device_timer.wrap(metadata={"category": category})
    if model_runner.device_timer
    else contextlib.nullcontext()
)
with ctx:
    # ... 继续执行 model.forward
```

评论区精华

该 PR 没有 review 评论，讨论可能集中在内部。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险包括：1）_estimate_prefill_perf 的参数类型从 int 改为 batch 对象，任何直接调用此方法的子类或外部代码需要更新；2）新增的 extend_lens 和 prefix_lens 拷贝增加了 batch copy 的内存开销，但仅为两个 list 的浅拷贝，影响很小；3）device-timer category 的区分可能影响现有监控面板，如果已有基于 "extend" 类别的汇总统计，需要确认 target_verify 的分离是否匹配预期。
- 影响：对用户透明：不改变任何外部 API 或行为。对系统：扩展点的添加使得未来可以更灵活地添加模型特定指标。对团队：指标报告更精确，有助于性能分析和优化。影响范围：仅限于指标报告模块，不影响核心推理路径。
- 风险标记：接口向后兼容，无专用测试覆盖，批次拷贝开销小

关联脉络

- 暂无明显关联 PR