

PR #29044 完整报告

sgl-project/sglang

Fix KV event publisher bind conflict under PP

合并时间: 2026-06-26 12:50

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29044>

执行摘要

- 一句话: 修复 PP 下 KV 事件发布者绑定冲突
- 推荐动作: 建议精读此 PR, 理解分布式系统中初始化资源的竞态条件及修复模式。代码变更简洁、正确, 是典型的多进程协调问题。

功能与动机

当 KV cache events 与 pipeline parallelism 同时启用时, 多个 PP 阶段会尝试在同一端点上启动 KV event publisher, 导致服务器启动失败 (端口绑定冲突)。

实现拆解

1. `kv_events_publisher.py`: 在 `init_kv_events` 方法的条件中增加 `self.ps.pp_rank == 0`, 确保只有 PP rank 0 才会启用 KV 事件发布者。
2. `scheduler.py`: 在 `build_kv_cache` 的 `enable_kv_cache_events` 参数中增加 `self.ps.pp_rank == 0`, 保持与发布者一致。
3. `metrics_collector.py`: 在 `init_new` 方法的 `enable_kv_cache_events` 条件中增加 `ps.pp_rank == 0`, 确保非零 PP 阶段不会收集 KV 事件指标。

关键文件:

- `python/sglang/srt/managers/scheduler_components/kv_events_publisher.py` (模块 KV 事件; 类别 source; 类型 core-logic): 核心修复文件: 在 `init_kv_events` 中增加 `pp_rank == 0` 检查, 确保仅有 PP rank 0 创建 `EventPublisher` 实例。
- `python/sglang/srt/managers/scheduler.py` (模块 调度器; 类别 source; 类型 core-logic): 同步修改: 在 `build_kv_cache` 的 `enable_kv_cache_events` 参数中增加 `self.ps.pp_rank == 0`, 与发布者条件保持一致, 防止内存泄漏。
- `python/sglang/srt/observability/metrics_collector.py` (模块 可观测性; 类别 source; 类型 core-logic): 同步修改: 在 `init_new` 的 `enable_kv_cache_events` 条件中增加 `ps.pp_rank == 0`, 确保指标收集与发布者一致。

关键符号: 未识别

关键源码片段

`python/sglang/srt/managers/scheduler_components/kv_events_publisher.py`

核心修复文件：在 `init_kv_events` 中增加 `pp_rank == 0` 检查，确保仅有 PP rank 0 创建 EventPublisher 实例。

```
# python/sglang/srt/managers/scheduler_components/kv_events_publisher.py
# 核心修复：在 KV 事件发布者初始化时增加 PP rank 0 检查
from sglang.srt.distributed import ParallelState
from sglang.srt.observability.kv_event import EventPublisherFactory

def init_kv_events(self, kv_events_config: Optional[str]):
    # 只有 PP rank 0、attn_tp_rank 0 和 attn_cp_rank 0 才启用 KV 缓存事件
    # 防止多个 PP 阶段绑定同一端口导致冲突
    self.enable_kv_cache_events = bool(
        kv_events_config
        and self.ps.pp_rank == 0
        and self.ps.attn_tp_rank == 0
        and self.ps.attn_cp_rank == 0
    )

    if self.enable_kv_cache_events:
        self.kv_event_publisher = EventPublisherFactory.create(
            kv_events_config, self.ps.attn_dp_rank
        )
```

`python/sglang/srt/managers/scheduler.py`

同步修改：在 `build_kv_cache` 的 `enable_kv_cache_events` 参数中增加 `self.ps.pp_rank == 0`，与发布者条件保持一致，防止内存泄漏。

```
# python/sglang/srt/managers/scheduler.py
# 同步检查：确保 KV cache 构建时也遵循 PP rank 0 约束
result = kv_cache_builder.build_kv_cache(
    server_args=self.server_args,
    model_config=self.model_config,
    tp_worker=self.tp_worker,
    page_size=self.page_size,
    spec_algorithm=self.spec_algorithm,
    attn_tp_cpu_group=self.attn_tp_cpu_group,
    tp_cpu_group=self.tp_cpu_group,
    attn_cp_cpu_group=self.attn_cp_cpu_group,
    enable_metrics=self.server_args.enable_metrics,
    enable_kv_cache_events=bool(
        self.server_args.kv_events_config
        and self.ps.pp_rank == 0 # 新增：确保只有 PP rank 0 启用
        and self.ps.attn_tp_rank == 0
        and self.ps.attn_cp_rank == 0
    ),
    ps=self.ps,
    tp_group=self.tp_group,
    pp_group=self.pp_group,
    enable_hierarchical_cache=self.enable_hierarchical_cache,
```

)

python/sglang/srt/observability/metrics_collector.py

同步修改：在 `init_new` 的 `enable_kv_cache_events` 条件中增加 `ps.pp_rank == 0`，确保指标收集与发布者一致。

```
# python/sglang/srt/observability/metrics_collector.py
# 指标收集器也同步检查 PP rank，确保三方条件一致
enable_kv_cache_events = bool(
    server_args.kv_events_config
    and ps.pp_rank == 0 # 新增：防止非零 PP 阶段收集指标
    and ps.attn_tp_rank == 0
    and ps.attn_cp_rank == 0
)
```

评论区精华

Review 过程中，gemini-code-assist[bot] 指出 `scheduler.py` 中尚未添加 `pp_rank == 0` 检查，可能导致非零 PP 阶段仍启用 KV 事件跟踪，造成内存泄漏。作者 ziang663 随后在 `scheduler.py` 和 `metrics_collector.py` 中同步添加了检查。ShangmingCai 也提示需要修改 `metrics_collector.py`。最终所有三点修改完成，审核通过。

- `scheduler.py` 中缺少 `pp_rank` 检查可能导致内存泄漏 (correctness): 作者 ziang663 在 `scheduler.py` 中补充了 `self.ps.pp_rank == 0` 检查，问题已解决。
- `metrics_collector.py` 同步修改 (correctness): 作者补充了 `metrics_collector.py` 中的 `pp_rank == 0` 检查，保持三方一致。

风险与影响

- 风险：风险较低。变更仅在条件判断中增加 `pp_rank == 0`，不会影响模型前向逻辑、logits、采样、内核或 KV 缓存内容。但需注意 PP rank 0 的发布者若发生故障，KV 事件将完全缺失；不过这种情况概率极低，且之前也只有一个发布者。
- 影响：影响范围仅限启用 KV 缓存事件和 PP 的部署场景。测试和文档未涉及，CI 中已有相关测试覆盖（PP 模式）。对非 PP 或无 KV 事件场景无影响。
- 风险标记：多进程竞态条件，内存泄漏风险

关联脉络

- PR #14194 [feature] implement dcp for deepseek_v2: 引入了 PP 相关的并行状态和 KV 缓存事件机制，与本 PR 的修复直接相关。