# PR #29373 完整报告

sgl-project/sglang

[AMD] [GLM5] Guard cuda_runtime.h for ROCm in fused_metadata_copy

合并时间：2026-06-27 14:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29373

## 执行摘要

此 PR 通过一行条件编译修复了 ROCm 平台下 fused_metadata_copy 内核因缺少 <hip/hip_runtime.h> 包含而编译失败的问题，从而避免 EAGLE 推测解码在高 draft 深度时回退到慢速逐元素拷贝，恢复约 84 倍推理吞吐量。

## 功能与动机

在 AMD MI300X/MI325X/MI355X 等 ROCm GPU 上，JIT 编译的 fused_metadata_copy 内核无条件包含 <cuda_runtime.h> 导致编译失败。这迫使 DSA 后端回退到 per-element 拷贝循环，其开销随 draft 长度线性增长，当 --speculative-num-steps >= 4 时，单步延迟从 ~10 ms 飙升到近 1 秒，输出吞吐从 227 tok/s 跌至 6.9 tok/s。

## 实现拆解

1. 定位问题文件：python/sglang/jit_kernel/csrc/elementwise/fused_metadata_copy.cuh 中第 38 行附近的 #include <cuda_runtime.h> 未加平台判断。
2. 添加 USE_ROCM 条件编译：使用与 sgl_kernel/utils.cuh 一致的 USE_ROCM 宏，在非 ROCm 时保留 cuda_runtime.h，在 ROCm 时替换为 hip/hip_runtime.h。
3. 构建与验证：仅 4 行新增，CUDA 构建路径字节不变；ROCm 下该内核正常编译，深度 4 EAGLE 的 median ITL 从 963 ms 降至 6.94 ms，输出吞吐从 6.9 tok/s 恢复至 578.3 tok/s。

### python/sglang/jit_kernel/csrc/elementwise/fused_metadata_copy.cuh

修复的核心文件，通过条件编译解决 ROCm 编译失败问题，恢复 EAGLE 深度推测编码性能。

```
// ... 文件头部
#include <tvm/ffi/container/tensor.h>
#include <algorithm> // for std::min

// 平台兼容头文件：ROCm 使用 hip/hip_runtime.h，CUDA 使用 cuda_runtime.h
#ifndef USE_ROCM
#include <cuda_runtime.h>
#else
#include <hip/hip_runtime.h>
#endif

// Forward mode enum (must match Python ForwardMode in sglang/srt/layers/attention/dsa_
```

```
backend.py)
enum ForwardModeEnum {
  DECODE = 0,
  TARGET_VERIFY = 1,
  DRAFT_EXTEND = 2
};
```

## 评论区精华

无 review 评论，提交者直接获得批准。

## 风险与影响

- 风险：极低。只在 ROCm 构建时添加头文件替换，CUDA 路径完全不碰。
- 影响：所有在 AMD GPU 上使用 EAGLE 推测解码、--speculative-num-steps >= 4 的用户将获得与深度 3 相当的正常吞吐，而非性能灾难。

## 关联脉络

此 PR 与同仓库近期 AMD/EAGLE 相关的优化（如 PR#30333 修复 DSV4 MTP 精度）互为补充，共同完善 AMD 平台上的推测解码体验。