PR #21709 完整报告

sgl-project/sglang

Fix draft extend cuda graph when spec_step=1

合并时间：2026-04-01 09:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21709

执行摘要

- 一句话：修复 spec_step=1 时 CUDA 图支持判断错误，确保草稿扩展使用正确后端。
- 推荐动作：该 PR 值得精读，展示了草稿扩展中后端选择与 CUDA 图支持的耦合关系。关注点：1. draft_attn_backend 与 draft_extend_attn_backend 的职责分离；2. HIP 代码块未修复的潜在问题；3. 后端类名重构的上下文。

功能与动机

根据 PR body 描述，当 spec_step = 1 时，draft_attn_backend 为 None。如果使用 draft_attn_backend 来判断是否应使用 CUDA 图，会导致判断错误。这影响了草稿扩展步骤的 CUDA 图支持，可能导致性能下降或功能异常。

实现拆解

修改了 python/sglang/srt/speculative/eagle_worker_v2.py 文件中的 init_cuda_graphs 函数。关键改动包括：1. 将判断 CUDA 图支持的后端变量从 draft_attn_backend 改为 draft_extend_attn_backend；2. 更新了后端类名引用，从 TritonMultiStepDraftBackend 改为 TritonAttnBackend，从 TRTLLMMLAMultiStepDraftBackend 改为 TRTLLMMLABackend。

关键文件：

- python/sglang/srt/speculative/eagle_2.py（模块 speculative）：修复草稿扩展 CUDA 图支持的核心逻辑，影响 speculative decoding 性能

关键符号：init_cuda_graphs

评论区精华

review 中 gemini-code-assist[bot] 指出，HIP 支持代码块（第 292-293 行）仍使用 self.draft_attn_backend，存在相同问题。建议对 HIP 代码应用类似修改以确保一致性。但 PR 作者未回应此建议，最终合并时 HIP 代码块未修改。

- HIP 代码块未同步修复 (correctness): PR 作者未回应，合并时 HIP 代码块未修改

风险与影响

- 风险：主要风险：1. HIP 代码块未同步修改，当 spec_step=1 且使用 HIP 后端时，可能仍存在 CUDA 图支持判断错误。2. 后端类名变更可能影响其他依赖这些类名的代码，但变更

范围小，风险较低。3. 缺少测试验证 spec_step=1 场景下的 CUDA 图行为。

- 影响：影响范围：仅影响使用草稿扩展且 spec_step=1 的 CUDA 图路径。对用户透明，但可能提升该场景下的推理性能。对系统影响小，仅修改单个文件中的条件判断逻辑。对团队影响：提醒需注意 HIP 后端的一致性修复。
- 风险标记：HIP 路径未修复，缺少 spec_step=1 测试

关联脉络

- PR #21783 [DSA] Support trtllm sparse mla kernel for prefill batches: 涉及 TRT-LLM 后端相关修改，可能共享类似的后端类名重构
- PR #21233 [refactor] Clean up duplicate flashinfer trtllm moe code: 同为代码清理和重构，涉及后端类名统一