PR #23294 完整报告

sgl-project/sglang

[BugFix] UniPC device placement + FlowUniPC sigma_min crash fix

合并时间：2026-05-18 09:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23294

执行摘要

- 一句话：修复 UniPC 和 FlowUniPC 调度器崩溃
- 推荐动作：值得合入，修复明确，改动量小，风险低。建议阅读 PR body 中的手动测试脚本以理解验证方法，并考虑后续补充自动化测试。

功能与动机

UniPCMultistepScheduler.set_timesteps 无条件将 self.sigmas 移至 CPU，当 pipeline 在 CUDA/MPS 上运行时，sigma 用作非标量张量（如 shape[1]）会导致设备不匹配崩溃。FlowUniPCMultistepScheduler.set_timesteps 在 final_sigmas_type='sigma_min' 时引用了 self.alphas_cumprod[0]，但该调度器未定义 alphas_cumprod，导致 AttributeError。

实现拆解

1. UniPCMultistepScheduler.set_timesteps 设备放置修复：在创建 sigmas 张量后，如果传入了 device 参数且非 CPU，则将其移动到目标设备；原有的无条件移动到 CPU 的操作改为仅在 device 为 None 或 CPU 时执行，避免 GPU/MPS 推理时的设备不匹配。
2. FlowUniPCMultistepScheduler sigma_min 修复：将 final_sigmas_type='sigma_min' 分支的 sigma_last 计算从使用不存在的 self.alphas_cumprod[0] 改为直接取计算好的 sigmas 数组的最后一个元素 sigmas[-1]，因为该调度器直接在 sigma 空间构建调度。
3. 回归验证：PR body 中提供了手动回归测试脚本，验证 UniPC 加载器 + CUDA sigma 路径工作正常，以及 FlowUniPC 的 sigma_min 分支不再抛出 AttributeError。

关键文件：

- python/sglang/multimodal_gen/runtime/models/schedulers/scheduling_unipc_multistep.py（模块 调度器；类别 source；类型 data-contract）：修复 sigmas 设备放置逻辑，避免 GPU/MPS 推理时 device mismatch 崩溃。
- python/sglang/multimodal_gen/runtime/models/schedulers/scheduling_flow_unipc_multistep.py（模块 调度器；类别 source；类型 data-contract）：修复 final_sigmas_type='sigma_min' 分支中引用不存在的 alphas_cumprod 导致的 AttributeError。

关键符号：set_timesteps

关键源码片段

python/sglang/multimodal_gen/runtime/models/schedulers/scheduling_unipc_multistep.py

修复 sigmas 设备放置逻辑，避免 GPU/MPS 推理时 device mismatch 崩溃。

```
# 在 set_timesteps 方法中，创建 sigmas 张量后：
# 之前无条件移动到 CPU，现在根据 device 决定放置位置
self.sigmas = torch.from_numpy(sigmas)
if device is not None:
    self.sigmas = self.sigmas.to(device=device) # 移动到目标设备

# 原有无条件移动到 CPU 的代码改为条件执行：
# 仅在 device 为 None 或目标为 CPU 时才移动到 CPU
if device is None or torch.device(device).type == "cpu":
    self.sigmas = self.sigmas.to("cpu")
```

python/sglang/multimodal_gen/runtime/models/schedulers/scheduling_flow_unipc_multistep.py

修复 final_sigmas_type='sigma_min' 分支中引用不存在的 alphas_cumprod 导致的 AttributeError。

```
# 在 set_timesteps 方法中，final_sigmas_type == "sigma_min" 分支：
# 之前：
# sigma_last = ((1 - self.alphas_cumprod[0]) / self.alphas_cumprod[0]) ** 0.5
# FlowUniPCMultistepScheduler 没有 alphas_cumprod 属性，导致 AttributeError
# 修复后：直接使用调度中计算出的最后一个 sigma 值
if self.config.final_sigmas_type == "sigma_min":
    sigma_last = sigmas[-1] # 使用计算好的 sigma 序列的最后一个值
```

评论区精华

没有实质性的 review 讨论。代码变更简洁，Mick Qian 直接批准了 PR。

- 暂无高价值评论线程

风险与影响

- 风险：回归风险低：两个 bugfix 都是针对特定崩溃场景的窄范围修复，不改变调度算法的数值行为。UniPC 修改仅影响 sigmas 张量的设备放置，FlowUniPC 修改仅影响 sigma_min 分支，且 sigma_last 值在数值上应与原意图一致（终端 sigma）。缺少直接的单元测试覆盖，但 PR 提供了手动验证脚本。
- 影响：影响范围小：仅影响使用 UniPCMultistepScheduler 或 FlowUniPCMultistepScheduler 且满足触发条件的用户（GPU/MPS 上使用 UniPC，或使用 FlowUniPC 且 final_sigmas_type='sigma_min'）。修复后这些用户不会再遭遇崩溃。
- 风险标记：缺少自动化测试

关联脉络

- 暂无明显关联 PR