PR #27440 完整报告

sgl-project/sglang

[Diffusion] Avoid GPU syncs in UniPC scheduler

合并时间：2026-06-06 22:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27440

执行摘要

- 一句话：用 torch.stack 替换 torch.tensor 避免 GPU 同步
- 推荐动作：值得精读，尤其关注如何在推理框架中通过简单代码替换消除隐式 GPU-CPU 同步。建议后续考虑采纳 reviewer 的 torch.ones_like 简化建议。

功能与动机

Cosmos3 H200 的 denoising profile 显示 aten::item / _local_scalar_dense / cudaStreamSynchronize 在 UniPC 调度器中重复出现（torch.tensor 构造时从 GPU 标量创建张量会触发CPU同步）。保持张量在设备上构建可以消除这些同步开销，同时保持数值等价。

实现拆解

1. 替换标量张量创建（所有 4 处）：在 multistep_uni_p_bh_update 和 multistep_uni_c_bh_update 两个方法中，将 rks.append(1.0) 改为 rks.append(torch.ones(() , dtype=h.dtype, device=h.device))，避免从 Python 浮点数构造时触发 CPU 同步。
2. 替换列表到张量的转换（所有 2 处）：将 torch.tensor(rks, device=device) 改为 torch.stack(rks).to(device=device)，因为 rks 中的元素已经是 GPU 上的标量张量，用 stack 直接拼接而不需要隐式传递数据到 CPU。
3. 替换系数列表 b 的构建（所有 2 处）：将 torch.tensor(b, device=device) 改为 torch.stack(b).to(device=device)，理由同上，b 中的元素也是 GPU 张量。
4. 测试验证：通过 test/unit/test_cosmos3.py（37 passed），确保数值正确性。

关键文件：

- python/sglang/multimodal_gen/runtime/models/schedulers/scheduling_unipc_multistep.py（模块 扩散调度器；类别 source；类型 performance；符号 multistep_uni_p_bh_update, multistep_uni_c_bh_update）：包含所有 6 处修改：用 torch.stack 替代 torch.tensor，并预创建设备上的标量张量以避免 GPU 同步。

关键符号：multistep_uni_p_bh_update, multistep_uni_c_bh_update

关键源码片段

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

包含所有 6 处修改：用 torch.stack 替代 torch.tensor，并预创建设备上的标量张量以避免 GPU 同步。

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

# 修改 1: multistep_uni_p_bh_update 中 rks 的构建
# 原：rks.append(1.0) + torch.tensor(rks, device=device) -> 触发 CPU 同步
# 改：直接 append 一个 GPU 标量张量，然后用 torch.stack 拼接
rks.append(torch.ones((), dtype=h.dtype, device=h.device))
rks = torch.stack(rks).to(device=device)

# 修改 2: multistep_uni_p_bh_update 中系数 b 的构建
# 原：b = torch.tensor(b, device=device)
# 改：由于 b 中元素已是 GPU 张量，用 stack 避免隐式数据传输
b = torch.stack(b).to(device=device)

# 修改 3 & 4: multistep_uni_c_bh_update 中完全相同模式的替换
rks.append(torch.ones((), dtype=h.dtype, device=h.device))
rks = torch.stack(rks).to(device=device)
b = torch.stack(b).to(device=device)
```

评论区精华

Gemini Code Assist Bot 建议将 torch.ones((), dtype=h.dtype, device=h.device) 简化为 torch.ones_like(h)，认为这样更简洁且自动推断 shape/dtype/device。该建议未被采纳但属于风格优化，不影响性能。

- 使用 torch.ones_like 简化标量创建 (style): 建议未被采用，但属于可选风格优化。

风险与影响

- 风险：低风险：变更仅涉及将 Python 标量和 torch.tensor 替换为设备上已有张量的 stack 操作，逻辑等价。若 h 是标量张量（shape 为 ()），torch.ones_like(h) 和 torch.ones((), ...) 行为一致。但需注意 rks 列表中元素数据类型：原 1.0 是 float，若 h 为 float 则无问题；若 h 为 half 或其他 dtype，需确保 torch.ones_like(h) 与之匹配。
- 影响：直接影响 UniPC 调度器（扩散模型 denoising 过程），Cosmos3 模型去噪阶段加速约 1.8%（536ms 节省）。不影响 API 接口、数据格式或模型精度，仅内部实现优化。
- 风险标记：微小边界情况风险（dtype 匹配）

关联脉络

- 暂无明显关联 PR