PR #27439 完整报告

sgl-project/sglang

[Diffusion] Enable Cosmos3 denoising profiling

合并时间：2026-06-07 10:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27439

执行摘要

- 一句话：Cosmos3 去噪循环集成 profiler 步进
- 推荐动作：值得快速合并的针对性修复，改动简洁且语义清晰。如果想了解 SGLang diffusion 模型的 profiling 机制，可以阅读 SGLDiffusionProfiler 的实现。

功能与动机

Cosmos3 使用模型特定的去噪循环代替通用去噪阶段。因此 --profile --num-profiled-timesteps ... 可以启动 profiler 但从不推进去噪调度，产生的 trace 缺少请求的时间步覆盖。

实现拆解

1. 导入 profiler 类: 在 cosmos3.py 中添加 from sglang.multimodal_gen.runtime.utils.profiler import SGLDiffusionProfiler。
2. 新增 step_profile 方法: 在 Cosmos3DenoisingStage 类中定义 step_profile 方法，通过单例获取 SGLDiffusionProfiler 实例并调用其 step_denoising_step() 方法。
3. 在 forward 末尾调用: 在 forward 方法中 if image_latent is not None: 赋值之后、batch.latents = latents 之前，添加条件判断：当 batch.profile 为 True 且 batch.is_warmup 为 False 时调用 self.step_profile()。这样每个去噪步骤（包括 CFG 的两步）都会使 profiler 前进，同时保证预热阶段不触发 profiling。
4. 测试验证: 通过运行 pytest -q python/sglang/multimodal_gen/test/unit/test_cosmos3.py，37 个测试通过，无回归。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py（模块 扩散模型；类别 source；类型 data-contract；符号 step_profile）：核心变更文件：新增 step_profile 方法并在 forward 中条件调用，使 Cosmos3 自定义去噪循环支持 profiler 步进。

关键符号：step_profile, forward

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py

核心变更文件：新增 step_profile 方法并在 forward 中条件调用，使 Cosmos3 自定义去噪循环支持 profiler 步进。

```
# python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3.py

from sglang.multimodal_gen.runtime.utils.profiler import SGLDiffusionProfiler

class Cosmos3DenoisingStage(PipelineStage):
    # ... 其他方法 ...

    def step_profile(self):
        """Advance profiler by one denoising step."""
        profiler = SGLDiffusionProfiler.get_instance()
        if profiler:
            profiler.step_denoising_step()

    def _run_transformer(self, latents, timestep, text_ids, text_mask,
                         video_shape, fps, cache_key="default",
                         noisy_frame_mask=None, max_text_seq_len=None,
                         current_timestep=None) -> torch.Tensor:
        # ... 原有逻辑 ...
        pass

    def forward(self, batch: Req, server_args: ServerArgs) -> Req:
        # ... 前面的去噪循环 ...
        if image_latent is not None:
            latents[:, :, 0:1, :, :] = image_latent

        # 仅在 profile 开启且非预热时调用 profiler 步进
        if batch.profile and not batch.is_warmup:
            self.step_profile()

        batch.latents = latents
        self.log_info("Denoising complete")
        return batch
```

评论区精华

review 中 gemini-code-assist[bot] 建议使用 getattr(batch, "profile", False) 替代直接访问 batch.profile，以防止 batch 对象缺少 profile 属性时抛出 AttributeError。但该建议未被采纳，因为 batch 对象在此上下文中始终会设置 profile 属性，且现有代码风格一致。最终 PR 获得 mickqian 的批准。

- 使用 getattr 避免 AttributeError (correctness): 建议未被采纳，因为 batch 对象在此上下文中始终有 profile 属性，且现有代码风格一致。

风险与影响

- 风险：风险很低：改动仅 9 行，条件 not batch.is_warmup 确保预热阶段不触发 profiler，SGLDiffusionProfiler.get_instance() 返回 None 时不会调用 step。唯一潜在风险是 batch.profile 属性类型或存在性假设，但 batch 对象在 profiling 开启时必定设置该属性。未对非 Cosmos3 模型产生影响。
- 影响：对用户：Cosmos3 用户现可正常使用 --profile --num-profiled-timesteps 参数进行逐时间步 profiling，获得正确的 trace 覆盖。对系统：新增的 step_profile 调用仅在 profile 且非预热时执行，对常规推理无性能影响。对团队：无负面影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR