

PR #29729 完整报告

sgl-project/sglang

Add opt-in SGLANG_ROPE_CACHE_FP32 to keep RoPE cache in fp32 on non-CUDA

合并时间: 2026-07-09 17:10

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29729>

执行摘要

- 一句话: 新增 RoPE 缓存 FP32 保留开关
- 推荐动作: 建议关注 review 中指出的 forward 路径 dtype 覆盖问题, 补充对 `forward_cuda` 和 `_match_cos_sin_cache_dtype` 的修改, 并添加测试用例验证 FP32 缓存在整个生命周期中保持。可作为非 CUDA 数值稳定性改进的起点。

功能与动机

RoPE 的 `cos/sin` 缓存仅在 CUDA 后端保持 FP32 精度, 其他后端 (如 ROCm) 会将其转换为模型 dtype, 导致精度损失和数值漂移。PR body 明确引用了代码中的注释和这一现象。

实现拆解

1. 新增环境变量声明 (`python/sglang/srt/envron.py`): 在 RoPE 缓存配置块中添加 `SGLANG_ROPE_CACHE_FP32 = EnvBool(False)`, 默认关闭, 不影响现有行为。
2. 导入提升 (`python/sglang/srt/layers/rotary_embedding/base.py`): 将 `from sglang.srt.envron import envs` 从 `_ensure_cos_sin_cache_length` 方法内部提升到模块顶层, 并移除该方法内的重复导入。
3. 条件判断修改 (`python/sglang/srt/layers/rotary_embedding/base.py`): 将初始化缓存时的降精度条件由 `if not _is_cuda:` 改为 `if not (_is_cuda or envs.SGLANG_ROPE_CACHE_FP32.get())`, 使得当环境变量开启时, 非 CUDA 后端也保持 FP32 缓存。

关键文件:

- `python/sglang/srt/layers/rotary_embedding/base.py` (模块 旋转编码; 类别 source; 类型 dependency-wiring; 符号 `RotaryEmbedding.init`): 核心变更文件, 修改了初始化时的降精度条件并提升导入, 但 review 指出 forward 路径仍需补充。
- `python/sglang/srt/envron.py` (模块 环境配置; 类别 source; 类型 core-logic): 新增环境变量声明, 是功能的开关入口。

关键符号: `RotaryEmbedding.init`

关键源码片段

`python/sglang/srt/layers/rotary_embedding/base.py`

核心变更文件，修改了初始化时的降精度条件并提升导入，但 review 指出 forward 路径仍需补充。

```
# python/sglang/srt/layers/rotary_embedding/base.py (head)

from sglang.srt.environ import envs # 导入提升到模块级别

class RotaryEmbedding(MultiPlatformOp):
    def __init__(self, head_size, rotary_dim, max_position_embeddings, base,
                 is_neox_style, dtype):
        # ...
        cache = self._compute_cos_sin_cache()
        # NOTE(ByronHsu): cache needs to be in FP32 for numerical stability.
        if not (_is_cuda or envs.SGLANG_ROPE_CACHE_FP32.get()):
            cache = cache.to(dtype) # 仅在非 CUDA 且未设置环境变量时降精度
        # ... 后续 forward 中仍可能覆盖 cache dtype
```

python/sglang/srt/envron.py

新增环境变量声明，是功能的开关入口。

```
# python/sglang/srt/envron.py (head)

# RoPE cache configuration
SGLANG_SPEC_EXPANSION_SAFETY_FACTOR = EnvInt(2)
SGLANG_ROPE_CACHE_FP32 = EnvBool(False) # 新增：非 CUDA 后端保留 FP32 精度
SGLANG_ROPE_CACHE_SAFETY_MARGIN = EnvInt(256)
SGLANG_ROPE_CACHE_ALIGN = EnvInt(128)
```

评论区精华

Gemini Code Assist bot 在 review 中指出，当前实现仅在初始化时保留了 FP32 缓存，但 forward 过程中（如 `forward_cuda` 和 `_match_cos_sin_cache_dtype`）仍会将 `self.cos_sin_cache` 原地转换为 query 的 dtype，导致 FP32 精度被覆盖。由于是 bot 评论且作者未回复，该问题未解决。

- forward 路径中缓存 dtype 被覆盖 (correctness): 作者未回复，问题未解决。

风险与影响

- 风险：主风险：当前实现不完整，forward 路径中仍会覆盖缓存 dtype，导致环境变量实际失效（在 ROCm 等非 CUDA 后端，`forward_cuda` 中的 fallback 路径会重新转换缓存）。此外，缺少单元测试验证环境变量生效。
- 影响：影响范围小：仅修改两个文件，增加 4 行、删除 4 行。对默认用户无影响（开关关闭）。开启后可在非 CUDA 后端消除 RoPE 数值漂移，但受限于 forward 路径覆盖，实际效果可能有限。
- 风险标记：实现不完整，缺少测试覆盖

关联脉络

- 暂无明显关联 PR