

PR #34612 完整报告

sgl-project/sglang

[Diffusion] Use current_platform instead of hardcoded "cuda" in cosmos3 guardrails

合并时间: 2026-08-19 20:26

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34612>

执行摘要

- 一句话: 修复 Cosmos3 NPU 守卫默认设备硬编码问题
- 推荐动作: 该 PR 值得快速阅读, 主要价值在于展示了如何利用 SGLang 已有的平台抽象来消除硬编码假设, 对后续在非 CUDA 硬件上适配其他模块有参考意义。关注点应放在 `current_platform` 的使用模式上, 可考虑为其补充单元测试以覆盖不同平台。

功能与动机

PR 描述中指出, `cosmos3_guardrails.py` 中默认设备硬编码为 `"cuda"`, 且 `offload_to_cpu` 始终为 `False`, 这假设 CUDA 总是活跃加速器, 导致在 Ascend NPU 等非 CUDA 硬件上无法正常工作。SGLang 的 `multimodal_gen` 运行时已通过 `sglang.multimodal_gen.runtime.platforms.current_platform` 暴露硬件抽象, 可通过 `device_type` 获取当前加速器类型。因此需要改用该抽象来消除硬编码假设。

实现拆解

本 PR 实现分为两部分:

1. 在 `python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3_guardrails.py` 中, 从 `sglang.multimodal_gen.runtime.platforms` 导入 `current_platform`, 并将 `_init_guardrails` 函数中 `idle_device` 的赋值由 `"cpu" if offload_to_cpu else "cuda"` 改为 `"cpu" if offload_to_cpu else current_platform.device_type`。这样当 `offload_to_cpu` 为 `False` 时, 模型会被放到当前活跃加速器上 (如 NPU), 而非固定 CUDA。
2. 在 `docs/cookbook/diffusion/Cosmos/Cosmos3.mdx` 中新增文档段落, 说明在 Ascend NPU 上加载 Cosmos-1.0-Guardrail 权重时可能遇到 `pickle.UnpicklingError` 问题, 并给出修改 `cosmos_guardrail/cosmos_utils.py` 将 `weight_only=True` 改为 `weights_only=False` 的 `workaround` 命令。测试方面, 本 PR 没有新增单元测试, 但改动逻辑简单直接, 且文档补充了部署注意事项。

关键文件:

- `python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3_guardrails.py` (模块守卫模块; 类别 `source`; 类型 `data-contract`; 符号 `_init_guardrails`): 核心改动文件, 将默认设备从硬编码 `'cuda'` 改为使用 `current_platform.device_type`, 以支持非 CUDA 硬件。

- docs/cookbook/diffusion/Cosmos/Cosmos3.mdx (模块 文档; 类别 other; 类型 core-logic) : 文档补充了 NPU 上加载守卫权重的 workaround 说明, 辅助用户部署。

关键符号: `_init_guardrails`

关键源码片段

[python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3_guardrails.py](#)

核心改动文件, 将默认设备从硬编码 'cuda' 改为使用 `current_platform.device_type`, 以支持非 CUDA 硬件。

```
# 文件 : python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/cosmos3_guardrails.py
```

```
# 导入硬件抽象, 用于获取当前活跃加速器类型
```

```
from sglang.multimodal_gen.runtime.platforms import current_platform
```

```
def _init_guardrails(offload_to_cpu: bool = False) -> None:
    global _checker
    if _checker is not None:
        return
    try:
        from cosmos_guardrail import CosmosSafetyChecker
    except ImportError:
        raise ImportError(
            "cosmos_guardrail is required for Cosmos3 safety checks. "
            "Install it with: pip install cosmos-guardrail==0.3.1"
        )
    logger.info(
        "Initializing Cosmos3 guardrails (offload_to_cpu=%s) ...", offload_to_cpu
    )
    _checker = CosmosSafetyChecker()
    # 关键修复: 不再硬编码 "cuda", 而是使用运行时平台抽象的设备类型,
    # 以支持 Ascend NPU 等非 CUDA 硬件。
    idle_device = "cpu" if offload_to_cpu else current_platform.device_type
    for runner in (_checker.text_guardrail, _checker.video_guardrail):
        if runner is None or not hasattr(runner, "models"):
            continue
        for m in runner.models:
            if isinstance(m, torch.nn.Module):
                m.to(idle_device)
    logger.info("Cosmos3 guardrails initialized.")
```

评论区精华

本 PR 的 Review 均为直接批准 (Approved), 没有形成深入讨论。评论只有维护者触发的 CI 命令 [/tag-and-rerun-ci](#)。因此没有实质性的技术交锋或疑虑记录。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险在于 `current_platform.device_type` 的可用性和返回值在不同硬件平台（如 CPU、不同版本的 NPU）上的表现未被测试覆盖。由于 `_init_guardrails` 在启动时被调用，若 `current_platform` 未正确初始化或返回空值，可能导致模型被放到错误设备（如 CPU）而影响性能。但该改动本身是朝着消除硬编码的正确方向，风险可控。
- 影响：本次变更影响 Cosmos3 系列模型的用户，特别是使用 Ascend NPU 等非 CUDA 硬件的用户，使他们能够正常启用安全守卫功能。对现有 CUDA 用户行为无影响（`current_platform.device_type` 在 CUDA 环境下通常返回 "cuda"，与之前一致）。文档的补充有助于 NPU 用户规避权重加载问题。影响范围较小，属于修复性改动。
- 风险标记：缺少测试覆盖，平台兼容性

关联脉络

- PR #34485 [AMD] Let the diffusion AITer backend take grouped-query K/V (fix Cosmos3-Nano startup): 两者都涉及 diffusion 在非默认硬件（AMD/NPU）上的运行修复，可能共享类似的平台适配模式。