PR #24685 完整报告

sgl-project/sglang

[NPU] fix profiler on npu

合并时间：2026-05-09 17:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24685

执行摘要

- 一句话：修复 NPU 上 torch profiler 算子形状信息缺失
- 推荐动作：本 PR 是 NPU 平台 profiling 功能的关键修复，建议合并。但需确认非 NPU 平台不会因 experimental_config=None 而报错，并考虑后续使用字典解包的更安全模式。

功能与动机

当在 NPU 上收集 profiling 数据时，算子形状信息即使设置了环境变量也缺失。本 PR 修复该问题，同时收集算子统计及其他 profiling 信息。

实现拆解

1. 定位问题：在 python/sglang/srt/managers/scheduler_profiler_mixin.py 的 start_profile 方法中，当 _is_npu 为真时，torch.profiler.profile 创建时缺少 experimental_config 参数，导致 NPU 端无法正确导出算子形状等信息。
2. 添加 NPU 专属配置：在 torch_npu.profiler.profile 的调用中，新增 experimental_config 参数，仅当 _is_npu 为真时传入一个 torch_npu.profiler._ExperimentalConfig 实例，配置了 ExportType.Text、ProfilerLevel.Level1 等选项。
3. 保持非 NPU 兼容：通过三元表达式 None if not _is_npu else ... 确保 CUDA/CPU 后端不受影响，避免 TypeError。

关键文件：

- python/sglang/srt/managers/scheduler_profiler_mixin.py（模块 分析器；类别 source；类型 core-logic）：唯一的修改文件，在 torch profiler 创建时添加了 NPU 特定的 experimental_config，是修复的核心。

关键符号：未识别

关键源码片段

python/sglang/srt/managers/scheduler_profiler_mixin.py

唯一的修改文件，在 torch profiler 创建时添加了 NPU 特定的 experimental_config，是修复的核心。

```
# 在 start_profile 方法中，构造 torch.profiler.profile 时
```

```
# 为 NPU 添加 experimental_config，以正确记录算子形状信息
self.torch_profiler = torch.profiler.profile(
    activities=torchprof_activities,
    with_stack=with_stack if with_stack is not None else True,
    record_shapes=record_shapes if record_shapes is not None else False,
    on_trace_ready=(
        None
        if not _is_npu
        else torch_npu.profiler.tensorboard_trace_handler(
            str(self.torch_profiler_output_dir)
        )
    ),
    # 新增：仅在 NPU 下传入 experimental_config，
    # 配置导出文本格式、分析级别 Level1，并关闭不必要的 AI Core 指标
    experimental_config=(
        None
        if not _is_npu
        else torch_npu.profiler._ExperimentalConfig(
            export_type=torch_npu.profiler.ExportType.Text,
            profiler_level=torch_npu.profiler.ProfilerLevel.Level1,
            msprof_tx=False,
            aic_metrics=torch_npu.profiler.AiCMetrics.AiCoreNone,
            l2_cache=False,
            op_attr=False,
            data_simplification=False,
            record_op_args=False,
            gc_detect_threshold=None,
        )
    ),
)
```

评论区精华

gemini-code-assist[bot] 指出：直接将 experimental_config 作为 keyword argument 传入会导致 CUDA/CPU 平台抛出 TypeError，因为标准 PyTorch profiler 不支持该参数。建议使用字典解包（dictionary unpacking）来条件性传递参数。但最终的实现仍采用了三元表达式方式，在非 NPU 时传入 None，这依赖于 torch_npu.profiler.profile 的 monkey-patch 或兼容性处理。审查者未进一步讨论，但需注意 None 是否被正确忽略。

- *experimental_config 参数兼容性问题 (correctness)*: 作者使用三元表达式 None if not _is_npu else ... 的方式，在非 NPU 时传入 None。可能依赖于 torch_npu 的 monkey-patch 兼容，但未进一步讨论。

风险与影响

- 风险：
  1. 兼容性风险：experimental_config=None 在标准 PyTorch profiler 中可能不被接受，如果 torch_npu 的 monkey-patch 未覆盖所有版本，可能导致 CUDA 端 profiler 调用

失败。需要在非 NPU 环境下充分测试。

2. 功能风险：配置中 *aic_metrics* 设为 *AiCoreNone* 会禁用部分 AI Core 指标，可能影响某些性能分析的完整性。
3. 回归风险：该修改仅影响 profiling 路径，不影响推理主逻辑，回归风险较低。 - 影响：影响范围：仅影响 NPU 后端使用 profiler 的场景，用户可通过 --profile 参数和 SGLANG_PROFILE_RECORD_SHAPES 环境变量启用。影响程度：提高 NPU 上 profiling 数据质量，使算子形状信息正确记录，便于性能分析和优化。

- 风险标记：平台兼容性风险，依赖 torch_npu 补丁

关联脉络

- PR #24815 Revert "[NPU] fix profiler on npu": 后续回滚 PR，说明本 PR 的修改可能引入了问题或需要重新评估。