

PR #28551 完整报告

sgl-project/sglang

Add opt-in CUDA-graph capture-trace export

合并时间: 2026-06-19 09:51

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28551>

执行摘要

- 一句话: 新增 CUDA-Graph 捕获跟踪导出功能
- 推荐动作: 简单实用的调试增强, 值得快速合入。设计上保持了最小侵入性 (默认关闭), 值得学习。

功能与动机

现有的 `--enable-profile-cuda-graph` 只能输出 `key_averages` 摘要表格, 缺少 kernel 级别的 shape/identity 记录, 不便于离线分析每次捕获的细节。通过导出台面 trace, 可以更好地排查 CUDA-Graph 捕获阶段的性能瓶颈。

实现拆解

1. 在 `environ.py` 新增环境变量: 定义 `SGLANG_ENABLE_CUDA_GRAPH_CAPTURE_TRACE`, 默认 `False`, 使用 `EnvBool` 封装。
2. 在 `profile_utils.py` 添加导出函数: 新增 `export_cuda_graph_capture_trace(profile_context, *, runner_name, tp_rank)`, 检查环境变量开关, 若开启则将 `profile_context` 的 chrome trace 导出为 `<SGLANG_TORCH_PROFILER_DIR>/graph_capture_profile/cuda_graph_capture-{runner_name}-TP-{tp_rank}.json.gz`, 确保多 runner 和多 TP rank 不互相覆盖。
3. 在 `decode_cuda_graph_runner.py` 的捕获后处理中调用: 在 `_post_process_after_profile` 方法末尾 (保持原有摘要日志不变) 调用 `export_cuda_graph_capture_trace`。注意首次提交中 `import` 在函数内部, 经 review 后调整为模块顶部全局 `import`。
4. 不涉及测试、配置或部署配套变更: 该功能为调试辅助, 不添加单元测试。

关键文件:

- `python/sglang/srt/utils/profile_utils.py` (模块 profile 工具; 类别 source; 类型 core-logic; 符号 `export_cuda_graph_capture_trace`): 新增核心导出函数 `export_cuda_graph_capture_trace`
- `python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py` (模块 解码器 runner; 类别 source; 类型 data-contract): 在 CUDA-Graph 捕获后处理中调用导出函数
- `python/sglang/srt/envron.py` (模块 环境配置; 类别 source; 类型 core-logic): 新增 `SGLANG_ENABLE_CUDA_GRAPH_CAPTURE_TRACE` 环境变量

关键符号: `export_cuda_graph_capture_trace`, `_post_process_after_profile`

关键源码片段

`python/sglang/srt/utils/profile_utils.py`

新增核心导出函数 `export_cuda_graph_capture_trace`

```
# python/sglang/srt/utils/profile_utils.py

def export_cuda_graph_capture_trace(prof_context, *, runner_name: str, tp_rank: int):
    """Persist a CUDA-graph capture profiler trace (chrome trace) to disk.

    Opt-in via ``SGLANG_ENABLE_CUDA_GRAPH_CAPTURE_TRACE`` (no-op otherwise). The
    capture profiler must have run with ``record_shapes=True`` so the trace can
    be inspected offline as a per-kernel shape/identity record. The file lands in
    ``<SGLANG_TORCH_PROFILER_DIR>/graph_capture_profile/`` and is namespaced by
    runner class and TP rank so concurrent capture passes (e.g. EAGLE3
    target/draft/draft-extend) and ranks don't overwrite each other.
    """
    if not envs.SGLANG_ENABLE_CUDA_GRAPH_CAPTURE_TRACE.get():
        return
    # 使用已有的 `SGLANG_TORCH_PROFILER_DIR` 作为基准目录
    output_dir = os.path.join(
        envs.SGLANG_TORCH_PROFILER_DIR.get(), "graph_capture_profile"
    )
    os.makedirs(output_dir, exist_ok=True)
    # 用 runner 名称 + TP rank 命名，避免并发覆盖
    path = os.path.join(
        output_dir, f"cuda_graph_capture-{runner_name}-TP-{tp_rank}.json.gz"
    )
    prof_context.export_chrome_trace(path)
    logger.info(f"CUDA graph capture trace saved to: {path}")
```

评论区精华

review 中 merrymercy 建议将 `import` 移到模块顶部 ("move imports to the top whenever possible")，luccafong 已采纳并提交修复。

- `import` 位置应放模块顶部 (style): luccafong 已修改为模块顶部全局 `import`

风险与影响

- 风险：风险极低：导出仅在 warmup 结束时一次性写入，不影响稳态服务性能。
`os.makedirs` 和 `export_chrome_trace` 可能因权限或磁盘问题抛出异常，但异常会自然冒泡，不会静默失败。
- 影响：仅影响 CUDA-Graph 捕获调试流程，对普通用户完全透明。开发者可通过设置 `SGLANG_ENABLE_CUDA_GRAPH_CAPTURE_TRACE=1` 获取 trace 文件。
- 风险标记：磁盘写入可能失败，权限问题

关联脉络

- 暂无明显关联 PR