PR #20910 完整报告

sgl-project/sglang

Add SGLang CUDA crash API logging inspired by FlashInfer

合并时间：2026-03-22 16:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/20910

执行摘要

本 PR 为 SGLang 添加了 CUDA 崩溃 API 日志记录功能，灵感来自 FlashInfer，旨在解决 LLM 和扩散内核调用边界在 CUDA 错误（如非法内存访问）时调试困难的问题。通过核心 logging 模块、装饰器工具和环境变量控制，实现崩溃前输入张量捕获，覆盖关键路径，并附详细技能文档。该功能仅在调试时启用，不影响生产性能，已通过 review 优化命名和设计。

功能与动机

为什么做：根据 PR body，CUDA 错误（如 illegal memory access, device-side assert）常导致程序崩溃，难以捕获崩溃前数据。本 PR 添加 SGLang-native API-level logging，在崩溃前记录输入张量形状、类型和值，便于调试数据问题。灵感来自 FlashInfer 的 API logging utility，但专注于崩溃调试和 level-10 dump capture，不包括 replay 代码以保持实现简洁。

实现拆解

核心模块：

- python/sglang/kernel_api_logging.py：新增 470 行代码，实现 debug_kernel_api 装饰器，支持环境变量 SGLANG_KERNEL_API_LOGLEVEL（0-10 级别）控制日志输出和 dump 文件生成。
- python/sglang/jit_kernel/debug_utils.py：45 行代码，提供 maybe_wrap_jit_kernel_debug 装饰器，自动推断 op 名称，减少手动指定。

集成点：

1. JIT Kernel：修改 46 个文件，如 norm.py、flash_attention_v4.py，添加 @maybe_wrap_jit_kernel_debug 装饰器到关键函数（如 fused_inplace_qknorm）。
2. 扩散模块：更新注意力层（layer.py）、线性层（linear.py），使用 wrap_method_with_debug_kernel_once 包装 forward 方法。
3. LLM 模块：修改 custom_op.py、bitsandbytes.py 等，用 debug_torch_op 替换 torch.ops.sglang.* 调用。
4. sgl-kernel：在 __init__.py 中自动包装导出函数，添加 maybe_wrap_debug_kernel 装饰器。

文档与环境：新增技能文档 .claude/skills/debug-cuda-crash/SKILL.md（657 行），详细教程；更新环境变量文档记录新变量。

评论区精华

review 讨论聚焦设计改进：

- 命名约定：merrymercy 指出“SGLANG API”太宽泛，建议重命名为“SGLANG_KERNEL_API_LOGLEVEL”，BBuf 修改。
- 自动推断：merrymercy 多次强调装饰器应自动推断 op 名称，而非手动指定，BBuf 在 commit 中重构实现。
- 类型友好性：DarkSharpness 建议装饰器保持签名信息，BBuf 优化了类型提示。

> merrymercy 原话：“It is not very precise to call them 'api' of sglang. The API of sglang is 'generate', 'chat'.” 讨论结论显示团队注重代码质量和用户体验，问题均被解决。

风险与影响

风险：

- 性能开销：启用 logging 增加调用开销，但默认禁用，风险低。
- 兼容性：新环境变量需文档说明，避免配置冲突。
- 代码复杂度：装饰器侵入多个核心文件，可能增加维护负担。 影响：
- 用户：提升调试效率，缩短问题定位时间。
- 系统：添加调试基础设施，不影响生产性能。
- 团队：改善开发体验，促进内核稳定性。

关联脉络

与历史 PR 关联：

- PR #21122：清理扩散 Triton 内核，与本 PR 修改的文件重叠，反映团队对内核代码的持续优化。
- PR #21130：添加测试技能指南，与本 PR 的技能文档相辅相成，显示文档和测试的协同演进。 从近期 PR 看，SGLang 在性能优化、调试工具和文档方面并行发展，本 PR 是调试基础设施的重要补充。