

PR #21331 完整报告

sgl-project/sglang

[CI] Rewrite `killall_sglang` as Python with CI/local dual mode

合并时间: 2026-03-25 14:54

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/21331>

执行摘要

本次 PR 将 `killall_sglang` 脚本从 shell 重写为 Python 版本, 引入 CI 模式以限制在 `CUDA_VISIBLE_DEVICES` 指定的 GPU 上进行进程清理, 解决共享主机容器中跨容器进程误杀导致的 CI 失败问题。同时保留原脚本供本地使用, 标记为弃用。该变更提升了 CI 测试的隔离性和稳定性, 但依赖于 NVIDIA 工具和特定环境配置。

功能与动机

动机: 在共享主机容器 (如 H100 RadixArk) 中, 使用 `--pid=host` 导致 `pgrep` 看到所有容器的进程。这使 `killall_sglang.sh` 在 CI 启动和测试清理时, 误杀其他容器的 `sglang` 进程, 产生 `exit code -9` 失败。因此, 需要将 kill 操作限制在当前容器分配的 GPU 上, 避免跨容器干扰。

功能: 新增 Python 脚本 `killall.py`, 在 CI 模式下通过 `nvidia-smi` 查询 GPU 进程, 仅杀死 `CUDA_VISIBLE_DEVICES` 中的 GPU 相关进程, 并在清理后检查内存使用率 (超过 10% 则退出)。原 shell 脚本保留为本地使用, 但标记为弃用。

实现拆解

模块	关键改动	描述
Python CLI 脚本	新增 <code>python/sglang/cli/killall.py</code>	核心实现: 使用 <code>_run_smi</code> 查询 GPU 信息, <code>_get_target_gpus</code> 解析 <code>CUDA_VISIBLE_DEVICES</code> , <code>_kill_pids</code> 杀死进程, 并添加内存阈值检查。代码片段:
<code>```python</code>		
<code>def _get_target_gpus():</code>		
<code> cvd = os.environ.get("CUDA_VISIBLE_DEVICES")</code>		
<code> if cvd is not None and cvd.strip():</code>		

模块	关键改动	描述
CI 工作流	修改 <code>.github/workflows/pr-test-rust.yml</code>	添加环境变量 <code>SGLANG_IS_IN_CI: true</code> , 确保 CI 模式生效, 并将调用从 bash 脚本改为 Python 脚本。
CI 安装脚本	修改 <code>scripts/ci/cuda/ci_install_dependencies.sh</code>	替换 <code>killall</code> 调用为 <code>python3 python/sglang/cli/killall.py</code> , 并检查退出码, 若内存异常则提前退出 CI。
包配置	修改 <code>python/pyproject.toml</code>	添加 <code>killall_sglang</code> 命令行入口点, 便于用户直接调用。
原脚本保留	修改 <code>scripts/killall_sglang.sh</code>	添加弃用注释和 TODO, 保留默认 <code>/rocm/all/gpus</code> 模式供本地使用, 确保向后兼容。

评论区精华

review 评论区为空, 表明无公开讨论或直接合并。但从提交历史 (14 次提交) 可见多次演进, 如合并、简化代码、移除冗余路径, 这反映了内部迭代优化, 但缺乏外部技术交锋。

风险与影响

技术风险:

- 依赖 `nvidia-smi` 工具, 在非 NVIDIA 环境 (如 AMD 或 CPU) 中可能失败, 需确保环境兼容。
- `CUDA_VISIBLE_DEVICES` 仅支持数字索引 (如 `'0,1'`), 不支持 UUID 格式 (如 `'GPU-xxx'`), 限制使用场景。
- 内存阈值检查 (10%) 可能导致误报, 例如正常残留内存触发 CI 退出, 影响测试连续性。
- 环境变量 `SGLANG_IS_IN_CI` 若设置不当, 可能导致 CI 模式不生效, 引发行为不一致。

影响评估:

- 对 CI 系统: 显著提升测试稳定性和资源隔离性, 减少跨容器干扰导致的失败, 影响程度为中度。

- 对用户：本地用户无直接影响，但未来迁移需适应新脚本；团队需维护双模式脚本，增加一定复杂度和依赖。

关联脉络

近期历史 PR 显示 CI 优化是持续趋势：

- PR #21330 默认启用 failfast 优化测试执行，与本 PR 的清理功能互补，共同提升 CI 效率。
- PR #21341 添加跨作业健康检查，与本 PR 的内存检查机制类似，都旨在增强 CI 稳定性和资源管理。
- PR #21364 修复 CI 工作流程标签，展示了对基础设施的持续维护。这些 PR 共享 'ci' 和 'infra' 标签，反映了团队在改进 CI 流程、减少 flakiness 方面的系统化努力，本 PR 是这一脉络中的关键一环，专注于 GPU 资源隔离。