

PR #31042 完整报告

sgl-project/sglang

[CI] Fix SGLANG_JIT_KERNEL_RUN_FULL_TESTS never activating the nightly full jit-kernel sweep

合并时间: 2026-07-14 17:32

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31042>

执行摘要

- 一句话: 修复夜间 JIT kernel 全量测试从未生效
- 推荐动作: 可快速合并并跟进夜间 CI 执行情况。值得关注的是该 PR 体现了 CI 配置与代码行为的不一致性问题, 建议未来通过环境变量类型系统强制统一布尔值的解析语义。

功能与动机

PR body 中指出: `nightly-test-nvidia.yml` 设置了

`SGLANG_JIT_KERNEL_RUN_FULL_TESTS: '1'`, 但 `should_run_full_tests()` 只将其与 `'true'` 比较, 导致 `'1'` 被判定为 `False`, 因此夜间全量测试从未实际激活。作者强调 `'it just never fired'`。

实现拆解

1. 注册 EnvBool 配置项: 在 `python/sglang/srt/envron.py` 的 # SGLang CI 区域新增 `SGLANG_JIT_KERNEL_RUN_FULL_TESTS = EnvBool(False)`, 使环境变量支持 `1/true/yes` 等常见真值表示, 并自动处理非法值。
2. 替换字符串比较逻辑: 删除 `python/sglang/jit_kernel/utils.py` 中已放弃的 `_FULL_TEST_ENV_VAR` 常量和基于 `os.getenv` 的显式小写转换比较, 改为通过 `envs.SGLANG_JIT_KERNEL_RUN_FULL_TESTS.get()` 获取该布尔值。
3. 调整 CI 超时配置: 在 `.github/workflows/nightly-test-nvidia.yml` 中为 8 GPU 的 `'JIT kernel full unit tests'` 作业显式指定 `--timeout-per-file 3600`, 并附注释说明全量网格参数化扩展后执行时间增长约为 6 倍, 原有默认 1200 秒不足以覆盖。

关键文件:

- `python/sglang/jit_kernel/utils.py` (模块 JIT Kernel; 类别 source; 类型 dependency-wiring): 修复了根本 bug: 将原始的字符串比较逻辑替换为通过 `envs` 读取 `EnvBool` 的方式, 确保 `should_run_full_tests()` 能正确识别 `'1'` 等真值。
- `python/sglang/srt/envron.py` (模块 环境配置; 类别 source; 类型 core-logic): 在 `Envs` 类中注册了新的 `EnvBool` 配置项, 使环境变量解析行为统一, 是修复的核心基础设施变更。
- `.github/workflows/nightly-test-nvidia.yml` (模块 CI 配置; 类别 infra; 类型 infrastructure): 为 8 GPU 全量测试 job 增加 `--timeout-per-file 3600`, 确保首次真正激

活的全量测试不会因超时而失败，并附有说明和测量数据注释。

关键符号：should_run_full_tests, get_ci_test_range

关键源码片段

python/sglang/jit_kernel/utils.py

修复了根本 bug：将原始的字符串比较逻辑替换为通过 `envs` 读取 `EnvBool` 的方式，确保 `should_run_full_tests()` 能正确识别 '1' 等真值。

```
# python/sglang/jit_kernel/utils.py
# 将原始字符串比较替换为通过 EnvBool 类型读取，
# 现在可以正确识别值为 '1'、'true'、'yes' 等常见真值。
from sglang.srt.environ import envs

def should_run_full_tests() -> bool:
    # 原先的逻辑是：os.getenv(_FULL_TEST_ENV_VAR, "false").lower() == "true"
    # 但 CI 里设置的是 SGLANG_JIT_KERNEL_RUN_FULL_TESTS="1"，比较 "1" == "true" 永远为
    # False。
    # 通过 EnvBool.get() 统一解析，自动处理各种真值格式。
    return envs.SGLANG_JIT_KERNEL_RUN_FULL_TESTS.get()

def get_ci_test_range(full_range: List[Any], ci_range: List[Any]) -> List[Any]:
    if should_run_full_tests():
        return full_range # 夜间全量扩展：执行完整的参数网格
    return ci_range if is_in_ci() else full_range
```

python/sglang/srt/envron.py

在 `Envs` 类中注册了新的 `EnvBool` 配置项，使环境变量解析行为统一，是修复的核心基础设施变更。

```
# python/sglang/srt/envron.py
# 在 Envs 类的 '# SGLang CI' 区域新增一行，
# EnvBool 的构造函数接收默认值并内部实现 get() 方法，
# 兼容 '1'/'true'/'yes' 等常见真值表示。
class Envs:
    ...
    # SGLang CI
    SGLANG_IS_IN_CI = EnvBool(False)
    SGLANG_IS_IN_CI_AMD = EnvBool(False)
    SGLANG_CUDA_COREDUMP = EnvBool(False)
    SGLANG_CUDA_COREDUMP_DIR = EnvStr(None)
    SGLANG_TEST_MAX_RETRY = EnvInt(None)
    # 新增：展开 jit_kernel 测试网格到完整参数范围（夜间使用）
    SGLANG_JIT_KERNEL_RUN_FULL_TESTS = EnvBool(False)
    ...
```

评论区精华

无 review 讨论线程。PR body 中作者注明该修改安全且将合并后监视夜间 CI。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 回归风险：低。仅修改了环境变量解析路径和 CI 超时配置，不涉及任何运行时推理逻辑或核心 kernel 行为。
 - 性能风险：中等。夜间全量测试将首次真正执行，可能导致 CI 运行时间显著增加，但已在 PR 中通过增加超时参数和 body 中的测量数据做了预期。
 - 若 EnvBool 的解析因某些意外配置格式引发异常，可能导致相关 job 失败；但 EnvBool 已在多个其他配置项中广泛使用，风险极低。
- 影响：
 - CI 用户：夜间测试将按预期输出全量 grids，有助于更早发现自定义 all-reduce 等 kernel 在更多参数配置下的回归问题。
 - 系统影响：仅限定于夜间 CI 流水线，不影响运行时稳定性或用户 API。
 - 团队影响：小规模修改，无跨部门协调需求。
 - 风险标记：CI 运行时间增加

关联脉络

- PR #30999 [Kernel] test_custom_all_reduce nightly timeout investigation: 此 PR 的动机源自对 #30999 夜间超时问题的调查，并提及会在此基础上为 test_custom_all_reduce 增加单独的 per-torchrun budget。