PR #29925 完整报告

sgl-project/sglang

ci: make multi-GPU jit test hangs attributable from the CI log

合并时间：2026-07-08 10:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29925

执行摘要

- 一句话：为多 GPU 测试添加超时可归因诊断
- 推荐动作：该 PR 是基础设施层面的改进，值得相关测试维护者关注。建议后续跟进 review 中提到的 timeout=None 和整数除法问题，考虑采用更安全的条件判断。

功能与动机

CI nightly 测试中 test_custom_all_reduce.py 在 nproc=2 时挂起，600 秒后被杀，但日志中没有任何证据表明哪个测试挂起或在哪里挂起。pytest 的块缓冲进度点混乱地刷入下一个测试的输出，被杀前也没有捕获堆栈。

实现拆解

1. 在 mp.py 中设置子进程无缓冲输出：在 multigpu_launch 函数中，在环境变量设置阶段添加 os.environ.setdefault("PYTHONUNBUFFERED", "1")，确保 torchrun 子进程的 pytest 进度输出实时到达 CI 日志，避免超时被杀时输出丢失或乱序。
2. 在 utils.py 中启用 faulthandler_timeout：在 multigpu_pytest_main 的 inner 函数中，在调用 pytest.main 时传递 -o faulthandler_timeout=<dump_after>，其中 dump_after = timeout // 2（如果 timeout 为 None 则默认 300 秒）。当单个测试超过该阈值时，pytest 的 faulthandler 会 dump 所有线程的堆栈（stderr 不会被重定向），而不杀死运行，从而在外部超时前显示挂起的测试名和完整堆栈。

关键文件：

- python/sglang/jit_kernel/mp.py（模块 JIT 启动；类别 source；类型 core-logic）：核心变更：在多 GPU 启动函数中设置 PYTHONUNBUFFERED=1，确保子进程标准输出无缓冲，避免日志乱序。
- python/sglang/jit_kernel/tests/utils.py（模块 测试工具；类别 test；类型 test-coverage）：测试框架入口：添加 pytest faulthandler_timeout 配置，在测试挂起时自动 dump 堆栈。

关键符号：multigpu_launch, multigpu_pytest_main

关键源码片段

python/sglang/jit_kernel/mp.py

核心变更：在多 GPU 启动函数中设置 PYTHONUNBUFFERED=1，确保子进程标准输出无缓冲，避免日志乱序。

```
# python/sglang/jit_kernel/mp.py ( 片段 )
    os.environ[env_key] = "1"
    os.environ[pid_key] = str(os.getpid())
    os.environ.setdefault("OMP_NUM_THREADS", "1")
    os.environ.setdefault("GLOO_SOCKET_IFNAME", "lo") # single-machine setup
    # Unbuffered child stdout: when a worker is killed on timeout, pytest's
    # block-buffered progress output is otherwise lost or flushed out of
    # order into the CI log, making it impossible to tell which test hung.
    os.environ.setdefault("PYTHONUNBUFFERED", "1")
    signal.signal(signal.SIGINT, signal.default_int_handler)
```

python/sglang/jit_kernel/tests/utils.py

测试框架入口：添加 pytest faulthandler_timeout 配置，在测试挂起时自动 dump 堆栈。

```
# python/sglang/jit_kernel/tests/utils.py ( 片段 )
    def inner() -> int:
        # CI's run_unittest_files invokes `python3 <file> -f` (legacy
        # unittest failfast). Translate to pytest's `-x` so it survives.
        pytest_args = ["-x" if a == "-f" else a for a in sys.argv[1:]]
        # Dump all thread stacks (every rank; stderr is not redirected) if a
        # single test exceeds half the harness budget, so a hung collective
        # is attributable from the CI log before the outer timeout kills the
        # process group. Non-fatal: the test keeps running after the dump.
        dump_after = (timeout // 2) if timeout else 300
        return pytest.main(
            [file, "-o", f"faulthandler_timeout={dump_after}"] + pytest_args
        )
```

评论区精华

gemini-code-assist[bot] 指出：如果 timeout 显式设为 None（禁用超时），dump_after 仍会回退为 300 秒，导致即使不需要超时也会在 5 分钟后 dump 回溯。另外，整数除法（//）在很小的超时值（如 1）下会使 dump_after 为 0，从而禁用 faulthandler 超时。建议改用浮点数除法（/）并在 timeout is None 时跳过 faulthandler。注意：该评论在最终代码中 未采纳，当前实现仍使用整数除法和 None 回退。

- timeout=None 时 faulthandler 仍被启用 (correctness): 未采纳。当前实现仍使用整数除法和 None 回退。

风险与影响

- 风险：
  - 当 timeout=None 时，dump_after 默认 300 秒，可能在不期望超时的场景中意外触发 faulthandler dump，产生过多日志。
  - 整数除法在 timeout=1 时导致 dump_after=0，禁用 faulthandler，但此类场景极少。

- PYTHONUNBUFFERED 可能带来少量性能开销，但在测试场景中可接受。
- 该改动仅影响 JIT kernel 测试框架，不影响生产路径。
- 影响：影响范围限于 sglang/jit_kernel 的多 GPU 测试。在 CI 中，超时测试将自动 dump 堆栈，极大降低排查挂起问题的难度。对用户无影响。
- 风险标记：未解决 review 意见：timeout=None 行为，整数除法边缘情况

关联脉络

- PR #30255 Fix DSV4 prefill large Triton recompilation idle across context lengths: 同属 JIT kernel 测试基础设施，共享测试框架文件
- PR #30303 [spec decoding] support rejection sampling in multi layer eagle: 同属 JIT kernel 模块，可能涉及相同的多 GPU 测试工具