PR #25039 完整报告

sgl-project/sglang

[AMD] Disable unittest fail-fast for deepseekv4 perf test

合并时间：2026-05-13 13:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25039

执行摘要

- 一句话：禁用 AMD DeepSeek-V4 测试的 unittest fail-fast
- 推荐动作：该 PR 展示了处理 CI 框架与测试用例需求冲突的巧妙方法（通过运行时过滤 sys.argv），值得测试维护人员注意。环境变量的同步更新也反映了对 AMD 平台最新优化配置的跟进。建议未来考虑将精度测试和性能测试拆分为独立文件，以更根本地避免此类问题。

功能与动机

The 4 DSv4 test files each define both an accuracy test (test_a_gsm8k) and a performance test (test_b_perf_8k_1k) that share a single very-expensive server launch in setUpClass. run_suite.py's run_one_file launches each test file with python3 <file> -f which enables unittest fail-fast, so an accuracy assertion failure would skip the perf measurement and lose the perf table from the GitHub step summary.

实现拆解

1. 定位问题：run_suite.py 通过 run_one_file 使用 python3 <file> -f 运行每个测试文件，开启 unittest 的 fail-fast 模式，导致 test_a_gsm8k（精度）失败后 test_b_perf_8k_1k（性能）被跳过。
2. 解决方式：在每个测试文件的 if __name__== "__main__": 中，导入 sys，从 sys.argv 中移除 -f 和 --failfast 参数，然后调用 unittest.main()，从而绕过 CI 框架的 fail-fast 设置。
3. 同步环境变量：将四个文件的 COMMON_ENV_VARS 中的多个配置项从 false 改为 true 或新增，以匹配最新的 AMD ROCm 7.2 推荐配置，包括：SGLANG_OPT_USE_FUSED_COMPRESS、SGLANG_OPT_USE_TRITON_SWA_PREPARE、SGLANG_OPT_USE_AITER_MHC_PRE、SGLANG_OPT_USE_AITER_MHC_POST、AITER_BF16_FP8_MOE_BOUND，并将 SGLANG_HACK_FLASHMLA_BACKEND 从 tilelang 改为 triton。
4. 统一修改：四个测试文件（test_deepseek_v4_flash_fp4.py、test_deepseek_v4_flash_fp8.py、test_deepseek_v4_pro_fp4.py、test_deepseek_v4_pro_fp8.py）均同步应用上述修改，保证一致性。

关键文件：

- test/registered/amd/test_deepseek_v4_flash_fp4.py（模块 测试脚本；类别 test；类型 test-coverage）：第一个被修改的测试文件，展示了核心的两处变更（环境变量和 fail-fast 过滤），影响其他三个文件同步修改。
- test/registered/amd/test_deepseek_v4_flash_fp8.py（模块 测试脚本；类别 test；类型 test-coverage）：与 flash_fp4 相同的修改，适用于 FP8 变体。
- test/registered/amd/test_deepseek_v4_pro_fp4.py（模块 测试脚本；类别 test；类型 test-coverage）：FP4 Pro 变体的测试文件，应用相同修改。
- test/registered/amd/test_deepseek_v4_pro_fp8.py（模块 测试脚本；类别 test；类型 test-coverage）：FP8 Pro 变体的测试文件，完成全部四个文件的修改。

关键符号：未识别

关键源码片段

test/registered/amd/test_deepseek_v4_flash_fp4.py

第一个被修改的测试文件，展示了核心的两处变更（环境变量和 fail-fast 过滤），影响其他三个文件同步修改。

```
# 文件：test/registered/amd/test_deepseek_v4_flash_fp4.py
# 关键修改 1：环境变量配置更新，适配 AMD ROCm 7.2 最佳实践
COMMON_ENV_VARS = {
    "SGLANG_OPT_USE_FUSED_COMPRESS": "true", # 从 false 改为 true，启用 fused 压缩
    "SGLANG_OPT_USE_OLD_COMPRESSOR": "true",
    "SGLANG_OPT_USE_TILELANG_SWA_PREPARE": "false",
    "SGLANG_OPT_USE_TRITON_SWA_PREPARE": "true", # 新增，使用 triton 实现
    "SGLANG_OPT_USE_JIT_KERNEL_FUSED_TOPK": "false",
    "SGLANG_OPT_USE_FUSED_HASH_TOPK": "false",
    "SGLANG_OPT_DEEPGEMM_HC_PRENORM": "false",
    "SGLANG_OPT_USE_TILELANG_MHC_PRE": "false",
    "SGLANG_OPT_USE_AITER_MHC_PRE": "true", # 新增，启用 AITER MHC pre
    "SGLANG_OPT_USE_TILELANG_MHC_POST": "false",
    "SGLANG_OPT_USE_AITER_MHC_POST": "true", # 新增，启用 AITER MHC post
    "SGLANG_ENABLE_THINKING": "1",
    "SGLANG_USE_AITER": "1",
    "AITER_BF16_FP8_MOE_BOUND": "1", # 新增，设置 AITER MoE 边界
    "SGLANG_USE_ROCM700A": "1",
    "SGLANG_FP8_PAGED_MQA_LOGITS_TORCH": "1",
    "SGLANG_OPT_DPSK_V4_RADIX": "0",
    "SGLANG_OPT_USE_OVERLAP_STORE_CACHE": "false",
    "SGLANG_OPT_USE_FUSED_STORE_CACHE": "false",
    "SGLANG_TOPK_TRANSFORM_512_TORCH": "1",
    "SGLANG_OPT_USE_TILELANG_INDEXER": "true",
    "SGLANG_HACK_FLASHMLA_BACKEND": "triton", # 从 tilelang 改为 triton
    "SGLANG_DSV4_REASONING_EFFORT": "max",
}

# FP4 variant: FP4 mixed-precision experts
FP4_ENV_VARS = {
```

```
    "SGLANG_DSV4_FP4_EXPERTS": "true",
    "SGLANG_FORCE_TRITON_MOE_FP8": "0",
}

# ... 测试方法（test_a_gsm8k, test_b_perf_8k_1k）保持不变 ...

if __name__ == "__main__":
    # 关键修改 2：run_suite.py 使用 -f 启动，开启 unittest fail-fast
    # 我们希望性能测试即使在精度测试失败时也能运行
    # 因此从 sys.argv 中移除 -f / --failfast 参数
    import sys
    sys.argv = [a for a in sys.argv if a not in ("-f", "--failfast")]
    unittest.main()
```

评论区精华

无，没有 review 评论或讨论。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。仅修改测试文件的环境变量和启动逻辑，不涉及生产代码。环境变量变更可能影响测试覆盖率，但已在 AMD 机器上验证（见 PR 附带的测试结果）。移除 fail-fast 可能导致一个测试方法失败后后续测试方法继续执行，但这是设计意图，不会引入雪崩风险因为测试方法彼此独立（共享一个 setUpClass 但各测试方法不依赖对方）。
- 影响：影响 AMD CI 中 DeepSeek-V4 nightly 测试的稳定性和数据完整性。性能测试将不再被精度测试失败阻塞，从而确保持续收集性能指标。对用户无直接影响。环境变量的更新有助于确保测试使用正确的硬件优化配置。
- 风险标记：环境变量变更，测试流程调整

关联脉络

- 暂无明显关联 PR