PR #25405 完整报告

sgl-project/sglang

[XPU] Add registry mechanism for XPU CI tests

合并时间：2026-05-27 08:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25405

执行摘要

- 一句话：XPU CI 测试迁移至注册架构，实现分阶段流水线
- 推荐动作：值得相关维护者精读，了解如何将 CI 测试对接注册体系。对于后续 XPU 测试的添加，应参照此模式。设计上烟雾测试与主测试分离的思路值得借鉴。

功能与动机

PR 说明指出：XPU CI 使用传统的硬编码测试列表（suite_xpu 字典），此 PR 旨在与 AMD 和 Nvidia 的注册架构对齐，提高可维护性和可扩展性。

实现拆解

1. 注册后端支持：在 python/sglang/test/ci/ci_register.py 中添加 HWBackend.XPU 枚举值和 register_xpu_ci() 函数，并在 REGISTER_MAPPING 中注册。
2. 创建烟雾测试：新增 test/registered/xpu/test_xpu_basic.py，注册到 stage-a 测试集，作为轻量级门控。
3. 迁移现有测试：将 test/srt/xpu/ 下的三个测试文件移至 test/registered/xpu/，并在文件头添加 register_xpu_ci() 调用，分配至 stage-b 测试集。
4. 更新测试套件配置：在 test/srt/run_suite.py 中清空 suite_xpu 字典；在 test/run_suite.py 中添加 XPU 的硬件映射和 per-commit 套件定义。
5. 重构 CI 工作流：在 .github/workflows/pr-test-xpu.yml 中将单 job 拆分为 stage-a、wait-for-stage-a、stage-b 三个阶段，并修正 finish job 的依赖关系，确保 stage-a 失败时正确报告。此外，还将 test/registered/attention/test_chunk_gated_delta_rule.py 的 XPU 注册从暂禁状态改为启用。

关键文件：

- test/registered/xpu/test_xpu_basic.py（模块 基础测试；类别 test；类型 test-coverage；符号 TestXPUBasic, test_basic_generation）：新增的 stage-a 烟雾测试，是 XPU CI 门控的关键组成部分，验证基本解码功能。
- .github/workflows/pr-test-xpu.yml（模块 CI 工作流；类别 infra；类型 infrastructure）：核心 CI 工作流变更，实现多阶段流水线，是此次重构的部署体现。
- python/sglang/test/ci/ci_register.py（模块 注册中心；类别 test；类型 test-coverage；符号 HWBackend.XPU, register_xpu_ci）：注册中心新增 XPU 后端支持，是注册架构的核

心入口。

- test/srt/run_suite.py（模块 测试套件；类别 test；类型 test-coverage）：清理旧的硬编码 XPU 测试列表，确保不再使用旧方式。
- test/registered/xpu/test_intel_xpu_backend.py（模块 XPU 后端；类别 test；类型 rename-or-move）：重命名并迁移的测试文件，代表现有测试迁移到注册架构。

关键符号：register_xpu_ci, test_basic_generation

关键源码片段

test/registered/xpu/test_xpu_basic.py

新增的 stage-a 烟雾测试，是 XPU CI 门控的关键组成部分，验证基本解码功能。

```
# Basic XPU test: verifies the server starts and produces a non-empty
# response on Intel XPU with the default attention backend.
# Assigned to stage-a so it gates stage-b before the heavier tests run.
import unittest
from sglang.test.ci.ci_register import register_xpu_ci
from sglang.test.test_utils import (
    DEFAULT_SMALL_MODEL_NAME_FOR_TEST_QWEN,
    CustomTestCase,
    is_in_ci,
    run_bench_one_batch,
)
register_xpu_ci(est_time=300, suite="stage-a-test-1-gpu-xpu")

class TestXPUBasic(CustomTestCase):
    def test_basic_generation(self):
        args = ["--device", "xpu", "--disable-radix-cache",
                "--mem-fraction-static", "0.6", "--batch-size", "1"]
        if is_in_ci():
            args += ["--input", "64", "--output", "4"]
        _, decode_throughput, _ = run_bench_one_batch(
            DEFAULT_SMALL_MODEL_NAME_FOR_TEST_QWEN, args
        )
        self.assertGreater(decode_throughput, 0, "XPU decode throughput must be > 0")
```

评论区精华

讨论中主要关注点包括：

- gemini-code-assist[bot] 建议将部分测试移至 stage-a 以实现烟雾门控，作者解释已有专门的轻量级测试，且 stage-b 测试较重，设计合理。
- mingfeima 指出 finish job 仅依赖 stage-b 可能导致 stage-a 失败时流水线仍然通过，作者修复了 finish job 的依赖，加入 stage-a 检查。
- Xia-Weiwen 怀疑 triton 安装变动导致 CI 失败，作者确认未涉及 triton 包，故障另有原因；随后 Xia-Weiwen 指出 triton 索引 URL 已在其他 PR 中更新。

- Xia-Weiwen 质疑禁用 GDN 测试的决定，作者重新检查并启用，测试通过。
- 烟雾测试空置建议 (*design*): 接受现有设计，无需调整。
- finish job 依赖缺陷 (*correctness*): 修复 finish job，同时依赖 *stage-a* 和 *stage-b*。
- triton 安装导致 CI 失败 (*performance*): 确认非本 PR 导致，无关。
- GDN 测试禁用决策 (*testing*): 重新启用测试，确认 Pass。

风险与影响

- 风险：主要风险包括：新 CI 流水线的 *stage-a*/*stage-b* 依赖关系可能导致门控失效（已修复）；测试文件迁移可能遗漏旧的 import 或依赖；新增的 *tabulate* 等依赖可能引入版本冲突；GDN 测试的启用需确保在 XPU 上稳定。但整体风险可控，因为与 AMD 方案一致，且经过 CI 验证。
- 影响：对 XPU 开发流程影响较大：测试编写者需遵循注册模式，不再手动修改 run_suite.py；CI 流水线从单 job 变为多阶段，总执行时间可能略有增加但门控更精细。对其他硬件后端无影响。团队需知晓新的测试注册规范。
- 风险标记：门控依赖风险，测试迁移风险

关联脉络

- 暂无明显关联 PR