PR #23820 完整报告

sgl-project/sglang

Update XPU Docker runtime stack & hf_home config

合并时间：2026-04-29 10:03

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23820

执行摘要

- 一句话：更新 XPU Docker 栈与 CI 配置以修复挂起
- 推荐动作：该 PR 是典型的基础设施维护，对于关注 XPU 后端和 CI 可靠性的团队成员值得精读。Dockerfile 中驱动安装步骤可作为 XPU 环境的参考。测试清理部分展示了如何消除冗余代码，但需注意显存清理的删除是否会影响测试隔离性。

功能与动机

PR body 指出：'Update XPU Docker runtime stack to resolve the xpu ci hang issue'，即更新 Docker 栈以解决 XPU CI 挂起。

实现拆解

1. 更新 Docker 运行时栈：在 docker/xpu.Dockerfile 中，通过安装 libze-intel-gpu1 等 Intel UMD 驱动包，确保 GPU 运行时兼容性。
2. 调整 CI 工作流：在 .github/workflows/pr-test-xpu.yml 中，移除原先通过环境变量 HF_HOME 设置的缓存路径，改为直接挂载宿主机的 HuggingFace 缓存目录；同时添加 /dev/dri/by-path 设备卷挂载，改善 GPU 设备识别。
3. 重构测试代码：在 test_intel_xpu_backend.py 和 test_deepseek_ocr.py 中，删除了重复的 _cleanup_xpu_memory 函数，将内存清理委托给更通用的测试框架；参数化 mem_fraction_static；将进程终止方式从直接 kill_process_tree 改为先 terminate 再等待超时后 kill。
4. 临时跳过不兼容测试：在 test_deepseek_ocr_triton.py 中，为 TestDeepSeekOCRTriton 类添加 @unittest.skip 装饰器，并加上 TODO 注释，等待 Triton-XPU 升级后重新启用。
5. 配套调整：更新 test_deepseek_ocr_triton.py 的导入方式，从模块整体导入改为直接导入基类，避免 pytest 重复收集。

关键文件：

- test/srt/xpu/test_intel_xpu_backend.py（模块 XPU 后端；类别 test；类型 test-coverage；符号 _cleanup_xpu_memory, intel_xpu_benchmark）：移除了冗余的显存清理函数，参数化内存比例，简化测试流程。
- test/srt/xpu/test_deepseek_ocr.py（模块 OCR 测试；类别 test；类型 test-coverage；符号 _cleanup_xpu_memory）：删除了 _cleanup_xpu_memory 方法，调整进程终止逻辑。

- test/srt/xpu/test_deepseek_ocr_triton.py（模块 Triton 测试；类别 test；类型 test-coverage；符号 TestDeepSeekOCRTriton）：导入方式改为直接导入，添加 @unittest.skip 临时跳过。
- docker/xpu.Dockerfile（模块 Docker 构建；类别 infra；类型 infrastructure）：安装最新 Intel UMD 驱动，解决运行时兼容性。
- .github/workflows/pr-test-xpu.yml（模块 CI 配置；类别 infra；类型 infrastructure）：调整 HF_HOME 配置和卷挂载，解决 CI 挂起。

关键符号：intel_xpu_benchmark, TestDeepSeekOCR.tearDownClass, TestDeepSeekOCR.setUpClass, TestDeepSeekOCRTriton.setUpClass

关键源码片段

test/srt/xpu/test_intel_xpu_backend.py

移除了冗余的显存清理函数，参数化内存比例，简化测试流程。

```
# intel_xpu_benchmark 装饰器：为测试函数配置基准测试参数。
# 参数：
# extra_args - 额外命令行参数列表（可选）
# min_throughput - CI 中最低吞吐量断言阈值（可选）
# mem_fraction_static - 内存比例，默认 "0.4"，可覆盖（新增参数）
def intel_xpu_benchmark(
    extra_args=None, min_throughput=None, mem_fraction_static="0.4"
):
    def decorator(test_func):
        @wraps(test_func)
        def wrapper(self):
            # 移除了 try-finally 和显式 _cleanup_xpu_memory 调用
            common_args = [
                "--disable-radix",
                "--trust-remote-code",
                "--mem-fraction-static",
                str(mem_fraction_static), # 使用参数化值而非固定 "0.4"
                "--batch-size",
                "1",
                "--device",
                "xpu",
            ]
            ci_args = ["--input", "64", "--output", "4"] if is_in_ci() else []
            full_args = common_args + ci_args + (extra_args or [])
            model = test_func(self)
            prefill_latency, decode_throughput, decode_latency = run_bench_one_batch(
                model, full_args
            )
            # 打印结果，CI 中还会检查吞吐量
            print(f"{model=}")
            print(f"{prefill_latency=}")
            print(f"{decode_throughput=}")
```

```
            print(f"{decode_latency=}")
            if is_in_ci() and min_throughput is not None:
                self.assertGreater(decode_throughput, min_throughput)
        return wrapper
    return decorator
```

test/srt/xpu/test_deepseek_ocr.py

删除了 _cleanup_xpu_memory 方法，调整进程终止逻辑。

```
    @classmethod
    def tearDownClass(cls):
        """Fixture that is run once after all tests in the class."""
        if hasattr(cls, "process") and cls.process:
            # 先尝试优雅终止
            cls.process.terminate()
            try:
                # 等待 30 秒让进程自动退出
                cls.process.wait(timeout=30)
            except Exception:
                # 超时后强制杀死进程树
                kill_process_tree(cls.process.pid)
        # 移除了原先的 _cleanup_xpu_memory() 调用
```

docker/xpu.Dockerfile

安装最新 Intel UMD 驱动，解决运行时兼容性。

```
# 切换到 root 用户（虽然 Dockerfile 默认以 root 运行，但明确切换以确保后续指令权限）
USER root

# 安装最新 Intel UMD 驱动及相关工具：包括 Level Zero 运行时、Intel 图形计算、
# OpenCL ICD、Intel 媒体驱动、libva 等，以支持 SYCL-TLA 和硬件加速。
RUN apt-get install -y software-properties-common && \
    add-apt-repository -y ppa:kobuk-team/intel-graphics && \
    apt-get update && \
    apt-get install -y libze-intel-gpu1 libze1 intel-metrics-discovery intel-opencl-icd clinfo intel-gsc
    && \
    apt-get install -y intel-media-va-driver-non-free libmfx-gen1 libvpl2 libvpl-tools libva-glx2 va-
    driver-all vainfo && \
    apt-get install -y libze-dev intel-ocloc

# 切换回非 root 用户
USER sdp
```

评论区精华

Review 中 gemini-code-assist[bot] 提出三个意见：

1. Critical: test_deepseek_ocr.py 的 tearDownClass 仍调用已删除的 _cleanup_xpu_memory，会导致 AttributeError。该问题在后续提交中已修复。

2. Medium: Dockerfile 中存在冗余的 USER root 指令。
3. Medium: 建议合并 apt-get 命令并清理缓存以减小镜像体积。这些优化建议未被采纳。mingfeima 最终批准了合并。

- test_deepseek_ocr.py 中 tearDownClass 引用已删除方法 (correctness): 该问题在后续提交中已修复，tearDownClass 不再调用 _cleanup_xpu_memory。
- Dockerfile 中冗余 USER root 指令 (style): 未被采纳，PR 合并时该指令仍存在。
- 建议合并 apt-get 命令并清理缓存 (performance): 未被采纳，最终版本未做修改。

风险与影响

- 风险： 1) Docker 驱动版本更新可能引入新的兼容性问题，需关注 XPU CI 其他测试是否受影响。 2) 移除显式内存清理可能导致测试间显存泄漏，但测试使用独立进程，风险较低。 3) 跳过 Triton 测试导致相关回归风险增加，需在 Triton-XPU 升级后及时恢复。 4) 进程终止方式的改变可能在极端情况下不生效，但相比原先的强行杀掉更为安全。
  - 影响：直接影响 XPU 后端的 CI 管道，解决挂起问题后 CI 稳定性提升。对普通用户无影响。团队需记录 TODO，在 Triton-XPU 升级后重新启用相关测试。测试代码简化降低了维护成本。
  - 风险标记：Docker 驱动更新，测试隔离性风险，Triton 测试被跳过

关联脉络

- PR #12771 Add intel_xpu as backend for GptOssForCausalLM, enabled for bf16 models: 该 PR 首次引入 Intel XPU 后端支持，本 PR 在此基础上解决 XPU CI 稳定性问题，属于同一功能线的后续维护。