# PR #36100 完整报告

sgl-project/sglang

[ci] xpu: trigger pr-test-xpu on multimodal_gen changes

合并时间：2026-08-27 14:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36100

## 执行摘要

- 一句话：XPU CI 新增 multimodal_gen 触发与 24 GiB 适配扩散套件
- 推荐动作：值得精读，重点看 gpu_cases.py 中 " 平台能力差异驱动的测试分层 "：如何按显存预算从共享用例池中裁剪平台子集、如何用 _select_xpu_cases 避免用例定义漂移、为何必须原地 mutate ONE_GPU_CASES 才能让 pytest 参数化生效。对 CI 维护者而言，路径过滤器与 pr-gate 的 changes_exist 兜底设计、perf baseline 播种流程都可直接借鉴。若关注 XPU 扩散推理，denoising.py 的 None vs [] 修复也是 diffusers 集成的一个典型坑。

## 功能与动机

此前 .github/workflows/pr-test-xpu.yml 的 main_package 路径过滤器使用 !(multimodal_gen) 否定，导致 python/sglang/multimodal_gen/**下的变更完全不会触发 XPU CI。PR body 明确指出：既有 1-gpu 套件为 1x H100 (80 GiB) 编写，包含 FLUX.1-dev、FLUX.2-dev、Qwen-Image 等大 checkpoint，在 24 GiB 的 Battlemage B580 上必然 OOM（实测 FLUX.2-dev 已分配 23.67 GiB / 23.91 GiB），且 "the CUDA-only FP8 / NVFP4 fast paths cannot rescue those models on XPU"。因此需要一套按显存预算裁剪、带有独立性能基线的 XPU 专用扩散套件。

## 实现拆解

### 第 1 步：重构 pr-test-xpu.yml 的过滤与门控

- check-changes job 新增 multimodal_gen 路径过滤器（python/sglang/multimodal_gen/**/!(*.md|*.ipynb)、python/pyproject_xpu.toml、工作流自身、docker/xpu.Dockerfile），并保留 main_package 上的 !(multimodal_gen) 否定，与 NPU/MUSA/AMD 工作流形状对齐。
- 输出三个布尔值：changes_exist（两者取或，含 run-all 模式）、main_package、multimodal_gen；pr-gate 条件从 main_package 改为 changes_exist，避免 multimodal_gen 变更绕过 PR 门禁。

### 第 2 步：新增 multimodal-gen-test-1-gpu-xpu job

- 运行在 bmg-multigen-models runner 上，if 条件为 multimodal_gen == 'true'。
- 流程：重置 workspace 所有权、checkout、启动 XPU 容器（xpu_ci_start_container.sh）、安装 pytest/expecttest/ray 等依赖、以 pyproject_xpu.toml 覆盖本地安装、fetch tags 让 setuptools_scm 解析真实版本，最后 docker exec 运行 run_suite.py --suite 1-gpu-xpu。

### 第 3 步：定义 1-gpu-xpu 测试套件（gpu_cases.py）

- 新增 ONE_GPU_XPU_CASE_IDS：zimage_image_t2i、flux_2_klein_image_t2i、flux_2_klein_base_image_t2i、wan2_1_t2v_1.3b（均为 ~5B 以下、BF16 全驻留 24 GiB 的 checkpoint）。
- 新增 _select_xpu_cases()：按 id 从 ONE_GPU_CASES 精确挑选并校验缺失 id，避免复制用例定义造成两处漂移。
- 注册 PARAMETRIZED_CASE_GROUPS["1-gpu-xpu"]，复用 test_server_1_gpu.py 做参数化。
- 在 current_platform.is_xpu() 时原地 mutate ONE_GPU_CASES 关闭 run_consistency_check（因为 test_server_1_gpu.py 直接参数化自 ONE_GPU_CASES，只覆盖 ONE_GPU_XPU_CASES 会被 pytest 忽略——这是提交 c141c217 返工后确认的）。

### 第 4 步：性能基线平台注册（testcase_configs.py + xpu_b60.json）

- PERF_BASELINE_FILE_BY_PLATFORM 增加 xpu_b60，别名表只保留 xpu（家族）与 bmg（代号，与 runner label 一致）。
- get_perf_baseline_platform() 在 XPU 平台优先返回 xpu_b60，避免套用快约 10 倍的 h100.json 导致误报。
- 新增 xpu_b60.json：含 long_term / pr_test 两套容差、按 stage 与 denoise step 的耗时分布，数据来自 XPU CI run 32736878259（PR #36100 自身）播种。

### 第 5 步：附带运行时修复（denoising.py）

- _prepare_denoising_loop() 中 encoder_hidden_states_image 由直接传 image_embeds 改为 image_embeds if image_embeds else None：T2V 请求下 image_embeds 为 []，而 diffusers 的 Wan transformer 只按 is not None 判断，[] 会误入不存在的图像分支。该修复影响所有平台的 T2V 路径。

### 测试与部署配套

- 无新增独立单元测试文件，通过 suite 注册 + CI job 提供端到端覆盖；PR 自身触发 PR Test (XPU) 自引用验证，测试计划中列出 4 个参数化用例的通过标准。

关键文件：

- .github/workflows/pr-test-xpu.yml（模块 XPU 流水线；类别 infra；类型 infrastructure）：核心 CI 变更：新增 multimodal_gen 路径过滤器与 multimodal-gen-test-1-gpu-xpu 专用 job，pr-gate 改用 changes_exist 兜底，是解锁 XPU 扩散测试通道的入口。
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 测试用例；类别 test；类型 test-coverage；符号 _select_xpu_cases）：测试套件的设计核心：定义 ONE_GPU_XPU_CASE_IDS、_select_xpu_cases 与 1-gpu-xpu 注册，并处理 XPU 平台一致性校验禁用；承载了显存预算裁剪与平台分层测试的主要决策。
- python/sglang/multimodal_gen/test/server/perf_baselines/xpu_b60.json（模块 性能基线；类别 test；类型 test-coverage）：新增 Arc Pro B60 平台的扩散性能基线（+222 行），避免套用快约 10 倍的 h100.json 造成性能门禁误报，是 XPU 扩散 job 可运行的关键配

套。

- python/sglang/multimodal_gen/test/server/testcase_configs.py（模块 用例配置；类别 test；类型 test-coverage）：性能基线平台注册：把 xpu_b60 纳入 PERF_BASELINE_FILE_BY_PLATFORM 与别名表，get_perf_baseline_platform 在 XPU 平台优先返回 xpu_b60。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 扩散流水线；类别 source；类型 core-logic）：共享运行时小修复：T2V 请求下 encoder_hidden_states_image 传 [] 会骗过 diffusers 的 is not None 守卫误入图像分支，改为 None 后影响所有平台 T2V 路径。

关键符号：_select_xpu_cases, get_perf_baseline_platform, _prepare_denoising_loop, _normalize_perf_baseline_platform

## 关键源码片段

### python/sglang/multimodal_gen/test/server/gpu_cases.py

测试套件的设计核心：定义 ONE_GPU_XPU_CASE_IDS、_select_xpu_cases 与 1-gpu-xpu 注册，并处理 XPU 平台一致性校验禁用；承载了显存预算裁剪与平台分层测试的主要决策。

```
# Intel Arc Pro B60 只有 24 GiB 显存，仅 ~5B 以下 checkpoint 能全量驻留。
# ONE_GPU_CASES 里更大的 FLUX.1-dev / FLUX.2-dev / Qwen-Image / Hunyuan3D /
# SANA-Video / image-edit 系列会在 24 GiB 上 OOM，且 FP8/NVFP4 量化仅限 CUDA。
ONE_GPU_XPU_CASE_IDS = (
    "zimage_image_t2i",
    "flux_2_klein_image_t2i",
    "flux_2_klein_base_image_t2i",
    "wan2_1_t2v_1.3b",
)


def _select_xpu_cases(case_ids: tuple[str, ...]) -> list[DiffusionTestCase]:
    # 从 ONE_GPU_CASES 按 id 精确挑选，避免复制用例定义造成两处漂移
    cases_by_id = {case.id: case for case in ONE_GPU_CASES}
    missing = [case_id for case_id in case_ids if case_id not in cases_by_id]
    if missing:
        raise RuntimeError(f"Unknown XPU diffusion case(s): {missing}")
    return [cases_by_id[case_id] for case_id in case_ids]


# 一致性 GT 图是 H100 生成的；XPU 在 Xe2 上使用不同 attention 内核与 fp 归约，
# 像素级输出必然偏离 H100，SSIM/PSNR 对照 golden 永远失败。
# test_server_1_gpu.py 直接基于 ONE_GPU_CASES 做参数化，因此只能原地修改
# 原列表条目 —— 仅通过 ONE_GPU_XPU_CASES 覆盖会被 pytest 静默忽略。
if current_platform.is_xpu():
    _xpu_ids = set(ONE_GPU_XPU_CASE_IDS)
    for _i, _case in enumerate(ONE_GPU_CASES):
        if _case.id in _xpu_ids and _case.run_consistency_check:
            ONE_GPU_CASES[_i] = replace(_case, run_consistency_check=False)
```

```
ONE_GPU_XPU_CASES = _select_xpu_cases(ONE_GPU_XPU_CASE_IDS)
```

### python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py

共享运行时小修复：T2V 请求下 encoder_hidden_states_image 传 [] 会骗过 diffusers 的 is not None 守卫误入图像分支，改为 None 后影响所有平台 T2V 路径。

```
def _prepare_denoising_loop(self, batch: Req, server_args: ServerArgs):
    # ... 前序逻辑省略（SP latents 预处理、guidance 构建、pos/neg kwargs 组装）
    image_kwargs = self.prepare_extra_func_kwargs(
        getattr(self.transformer, "forward", self.transformer),
        {
            # 传 None（而不是 []）：T2V 请求没有图像条件，batch.image_embeds 为 []。
            # diffusers 的 Wan transformer 只按 `is not None` 判断是否走图像分支，
            # [] 会骗过守卫进入不存在的图像分支；None 才会正确跳过。
            # TODO: make sure on-device
            "encoder_hidden_states_image": image_embeds if image_embeds else None,
        },
    )
```

## 评论区精华

review 共 3 条评论线程，全部由 mingfeima 提出、作者采纳解决：

1. 过滤器形状对齐 NPU/MUSA/AMD：mingfeima 在 pr-test-xpu.yml 的 diff 上指出不应直接删除 !(multimodal_gen) 否定，而是 "prefer to match NPU/MUSA/AMD: keep !(multimodal_gen) on main_package, add a multimodal_gen: paths filter, and a dedicated XPU job that actually runs a diffusion suite"，并给出 pr-test-npu.yml 的代码引用。作者在提交 d3852d2 中改为独立过滤器 + 专用 job 方案。
2. 性能基线别名过多：mingfeima 对 5 个别名（xpu / xpub60 / arcprob60 / bmg / battlemage）提问 "do we need to use so many aliases? use 1 or 2 should be fine"，作者回复 "done and updated" 并在提交 391537a 中精简为 xpu + bmg 两个。
3. 提交历史还记录了作者自查返工：c141c217 明确说明此前用 replace() 副本覆盖 run_consistency_check 是无效的，因为 test_server_1_gpu.py 直接引用 ONE_GPU_CASES，这是有价值的工程教训。

- main_package 过滤器是否应保留 !(multimodal_gen) 否定 (design): 作者采纳，提交 d3852d2 改为独立 multimodal_gen 过滤器 + multimodal-gen-test-1-gpu-xpu 专用 job，pr-gate 条件改用 changes_exist 兜底。
- 性能基线平台别名是否过多 (design): 提交 391537a 精简为 xpu（家族）+ bmg（代号，与 runner label bmg-multigen-models 一致）两个别名。

## 风险与影响

- 风险：
  1. 共享推理代码变更：denoising.py 的 image_embeds if image_embeds else None 影响所有平台的 T2V 路径（不只 XPU），若某些 diffusers 后端对 None 与 [] 的处理有差异，可能引入回归，但概率较低且已通过 XPU CI 验证。

2. 性能基线单次采样：xpu_b60.json 由单次 CI run 播种，非 denoise stage 容差高达 0.90、显存容差仅 0.05，B580 显存分配抖动可能导致误报，需要更多运行样本校准。
3. 一致性校验全局关闭：在 is_xpu() 下原地 mutate ONE_GPU_CASES 关闭一致性检查，未来若有开发者以为 ONE_GPU_XPU_CASES 是独立副本而直接修改它，改动会被 pytest 静默忽略，是隐性维护陷阱。
4. 过滤器组合风险：changes_exist / main_package / multimodal_gen 三个输出的布尔组合依赖 paths-filter 的 glob 准确性，若匹配失误可能出现专用 job 不触发但 pr-gate 放行的空窗。
5. CI 状态观察：merged 前 CI States 显示 PR Test (Extra) 与 AMD ROCm 7.2 失败，作者发起 /rerun-failed-ci，需确认失败与本次变更无关。 - 影响：对用户：XPU 平台上的 diffusion/multimodal_gen 服务变更从此有了 CI 守护，B580 用户可预期回归被提前拦截。对系统：XPU CI 新增 bmg-multigen-models runner 通道、1 个专用 job、1 份性能基线与 1 个 suite 注册，NVIDIA/AMD/ROCm 各套件保持不变。对团队：维护者需要同步维护 1-gpu（NVIDIA）与 1-gpu-xpu 两套用例集合，并关注 B580 显存预算变化；本 PR 也确立了 " 按平台显存与量化能力分层测试 " 的模式，后续 XPU 量化路径（INT4/INT8 weight-only via IPEX）落地后可把大模型用例重新启用。 - 风险标记：新 CI 门禁通道，共享推理代码变更，性能基线单次采样，一致性校验全局关闭

# 关联脉络

- PR #36636 [AMD][CI] Add targeted Mori test labels: 同类平台 CI 门禁建设：为 AMD Mori 流水线增加专属测试标签，与本 PR 为 XPU 增加 diffusion 测试通道的模式一致，体现了多硬件平台 CI 分层覆盖的演进方向。
- PR #34492 XPU: remove SGLANG_USE_SGL_XPU flag: XPU 平台基础设施演进：默认启用 sgl-kernel MoE。本 PR 是其后的测试配套，说明 XPU 平台从基础推理走向 diffusion 全功能验证。