PR #33654 完整报告

sgl-project/sglang

[CI] Move CPU-only unit tests to the CPU suite and trim dead 5090 registrations

合并时间：2026-08-05 16:43

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33654

执行摘要

- 一句话：CPU 单测迁至 CPU 池并压缩 5090 队列
- 推荐动作：值得 CI 维护者和测试贡献者精读。重点关注三处：一是注册级 disabled= 取代 pytestmark.skip 的分片治理思路；二是 run_bench_serving_multi 让多个 benchmark 共享 server 的聚合测试模式；三是“CUDA 屏蔽后跑通 + 保留一个设备路径注册（如 AMD）”的 CPU-only 判定模板。该 PR 不涉及产品逻辑，对业务研发可略读。

功能与动机

PR 标题和正文明确指出这是一次“registration-level trims on the 1-gpu-5090 pool (~675s per commit off base-b, no test logic touched)”。核心动机有三：

1) 大量纯 CPU 单元测试占用了稀缺的 1-gpu-5090 队列资源； 2) 模块级 pytestmark.skip 对分片不可见，文件仍计入预算并产生 import 开销，还阻塞 owner 在本地直接运行测试；3) VLM 性能测试的硬阈值放在队列竞争最激烈的池子里容易误报，更适合 label 门控。正文还明确排除了两个不能迁移的文件，说明作者对真实设备依赖做了逐一核实。

实现拆解

以“CUDA 屏蔽后仍能跑通”作为判定标准，将 19 个文件从 base-b / 1-gpu-small 迁移到 base-a-test-cpu：

- 典型文件包括 test/registered/unit/models/test_llava.py、test/registered/unit/managers/test_profile_merger_http_api.py、test/registered/unit/managers/test_prefill_adder.py、test/registered/unit/models/test_deepseek_mla_dispatch.py 等；
- 原本指向休眠 base-c-test-cpu（无派发 workflow）的注册统一改指 base-a-test-cpu；
- 已存在 live CPU 注册的文件（如 test_eagle_worker_v2_topk1_fastpath.py、test_vit_pos_embed_interpolate.py）删除重复的 CUDA 注册；
- 同步移除这些文件的 AMD 1-gpu 注册，但 test_vit_pos_embed_interpolate.py 因在有 GPU 时真实走设备路径而保留 AMD 注册。

把位于功能目录下的 CPU 迁移文件搬到 test/registered/unit/ 对应子目录，符合单测放置约定：

- disaggregation/test_specv2_kvcache_offloading.py → unit/disaggregation/；

- vlm/test_evs.py → unit/multimodal/；
- prefill_only/test_embed_overrides.py → unit/managers/；
- models/test_vit_pos_embed_interpolate.py → unit/models/。

8 个 unit/lora 文件将模块级 pytestmark = pytest.mark.skip(...) 替换为注册级 register_cuda_ci(..., disabled="reason")，例如 test_experimental_sgl_marlin_multi_prefill.py、test_inkling_linearized_lora_unit.py、test_inkling_moe_lora_overlap_unit.py。这样分片时能感知禁用状态，不再占用预算，同时移除 pytestmark 行后 owner 可临时去掉 disabled 在本地稳定测试。

test/registered/perf/test_vlm_perf_5090.py 将 test_vlm_offline_throughput 与 test_vlm_online_latency 两个用例合并为 test_vlm_perf：

- 通过 run_bench_serving_multi 让离线与在线两个 benchmark 共享一个 server；
- online 阶段从 250 个 prompt 降到 50 个（1 req/s），保留全部四个阈值断言；
- 新增 _local_tokenizer_path() 优先使用本地模型快照，避免 bench 客户端在 CI 中长时间阻塞于 HF Hub API；
- 注册从 base-b 移到 extra-a，est_time 从 406 降到 180。

未被迁移的两类文件有明确依据：test_radix_cache_unit.py 的 test_memory_allocated 读取 torch.get_device_module().memory_allocated() 依赖设备模块；test_hicache_nixl_storage.py 依赖只在 CUDA runner 上安装的 nixl。PR 经过 /tag-and-rerun-ci 与 /rerun-test test_vlm_perf_5090.py 验证，重跑在 1-gpu-5090 上通过。

本 PR 不涉及产品源码、配置或 schema 变更，全部为测试注册与文件位置调整。

关键文件：

- test/registered/perf/test_vlm_perf_5090.py（模块 性能测试；类别 test；类型 test-coverage；符号 _local_tokenizer_path, test_vlm_perf, test_vlm_offline_throughput, test_vlm_online_latency）：VLM 性能测试从 base-b 移至 extra-a，并通过 run_bench_serving_multi 将离线 / 在线两个阶段合并到同一个 server，预估耗时从 406 秒降至 180 秒；是本次改动最实质的测试工程变更。
- test/registered/unit/lora/test_experimental_sgl_marlin_multi_prefill.py（模块 LoRA 测试；类别 test；类型 configuration）：作为 8 个 LoRA 测试的代表，展示从模块级 pytestmark.skip 到注册级 disabled= 的规范转换，是本次分片预算治理的核心模式。
- test/registered/unit/models/test_vit_pos_embed_interpolate.py（模块 位置嵌入；类别 test；类型 rename-or-move）：从 test/registered/models 迁到 unit/models，移除重复的 CUDA 注册并保留 AMD 设备路径，体现 CPU-only 判定的典型边界。
- test/registered/unit/disaggregation/test_specv2_kvcache_offloading.py（模块 KV 缓存；类别 test；类型 rename-or-move）：从功能目录迁到 unit 子树，并把 CUDA/AMD 注册替换为 CPU 注册；mock 化的 KV offload 卸载逻辑在 CPU 上即可验证。
- test/registered/unit/multimodal/test_evs.py（模块 多模态；类别 test；类型 rename-or-move）：EVS 配置解析测试为纯 dataclass 逻辑，迁至 unit/multimodal 并接入 base-a-test-cpu，同时移除曾指向休眠 base-c-test-cpu 的注册。

关键符号：test_vlm_perf, _local_tokenizer_path, test_vlm_offline_throughput, test_vlm_online_latency

关键源码片段

test/registered/perf/test_vlm_perf_5090.py

VLM 性能测试从 base-b 移至 extra-a，并通过 run_bench_serving_multi 将离线 / 在线两个阶段合并到同一个 server，预估耗时从 406 秒降至 180 秒；是本次改动最实质的测试工程变更。

```
# test/registered/perf/test_vlm_perf_5090.py
# 将两个独立的性能测试合并为共享 server 的流程：离线吞吐与在线延迟
# 复用同一个 SGLang 实例，避免重复拉起引擎的固定开销；
# 注册从 base-b 移到 extra-a，预估耗时从 406 秒压到 180 秒。

import os
import unittest

from sglang.test.ci.ci_register import register_amd_ci, register_cuda_ci
from sglang.test.test_utils import (
    DEFAULT_SMALL_VLM_MODEL_NAME_FOR_TEST,
    DEFAULT_URL_FOR_TEST,
    CustomTestCase,
    auto_config_device,
    get_benchmark_args,
    is_in_ci,
    run_bench_serving_multi,
    write_github_step_summary,
)

register_cuda_ci(est_time=180, stage="extra-a", runner_config="1-gpu-small")
register_amd_ci(est_time=300, suite="stage-b-test-1-gpu-small-amd")


def _local_tokenizer_path():
    # 优先使用本地快照中的 tokenizer，避免 bench 客户端在 CI 中触发
    # HF Hub API 调用（可能阻塞数分钟），找不到时回退到 None
    try:
        from sglang.srt.utils import find_local_repo_dir

        local_dir = find_local_repo_dir(
            DEFAULT_SMALL_VLM_MODEL_NAME_FOR_TEST, revision=None
        )
        if local_dir and os.path.isdir(local_dir):
            return local_dir
    except Exception:
        pass
    return None
```

```
class TestVLMPerf5090(CustomTestCase):
    def test_vlm_perf(self):
        # 构建统一的 benchmark 参数字典，离线与在线两个场景共用一组配置
        common = dict(
            base_url=DEFAULT_URL_FOR_TEST,
            dataset_name="mmmu",
            dataset_path="",
            tokenizer=_local_tokenizer_path(),
            random_input_len=4096,
            random_output_len=2048,
            seed=0,
            device=auto_config_device(),
        )
        offline = get_benchmark_args(
            num_prompts=200, request_rate=float("inf"), **common
        )
        # 在线阶段从 250 个 prompt 降到 50 个（1 req/s），约 1 分钟跑完；
        # 中位数在该样本量下稳定，且断言使用的都是宽松上限，不会误报
        online = get_benchmark_args(num_prompts=50, request_rate=1, **common)

        (_, res_offline), (_, res_online) = run_bench_serving_multi(
            DEFAULT_SMALL_VLM_MODEL_NAME_FOR_TEST,
            DEFAULT_URL_FOR_TEST,
            other_server_args=["--mem-fraction-static", "0.7"],
            benchmark_args=[offline, online],
        )

        if is_in_ci():
            write_github_step_summary(
                f"### test_vlm_perf (5090)\n"
                f"Output throughput: {res_offline['output_throughput']:.2f} token/s\n"
                f"median_e2e_latency_ms: {res_online['median_e2e_latency_ms']:.2f} ms\n"
            )
            # 四个关键阈值断言全部保留，只合并了执行路径
            self.assertGreater(res_offline["output_throughput"], 2000)
            self.assertLess(res_online["median_e2e_latency_ms"], 16500)
            self.assertLess(res_online["median_ttft_ms"], 150)
            self.assertLess(res_online["median_itl_ms"], 8)


if __name__ == "__main__":
    unittest.main()
```

test/registered/unit/lora/test_experimental_sgl_marlin_multi_prefill.py

作为 8 个 LoRA 测试的代表，展示从模块级 pytestmark.skip 到注册级 disabled= 的规范转换，是本次分片预算治理的核心模式。

```
# test/registered/unit/lora/test_experimental_sgl_marlin_multi_prefill.py
# 原实现使用模块级 pytestmark = pytest.mark.skip 跳过 CI，
```

```
# 该方式对分片不可见：文件仍计入分片预算并产生一次 import 开销；
# 改为注册级 disabled 后，调度器能看到这是禁用项，owner 也可在本地直接运行。
from pathlib import Path

from sglang.test.ci.ci_register import register_cuda_ci

register_cuda_ci(
    est_time=45,
    stage="base-b",
    runner_config="1-gpu-small",
    disabled="fused MoE LoRA-add kernel needs more opt-in shared memory than the ",
)
# 背景：该 fused kernel 的共享内存占用超过小 GPU CI runner 的 opt-in 上限
# （L4 上约 99 KiB，rank=128 时会 OOM）；选择在 CI 禁用而不是裁剪生产 kernel。
```

test/registered/unit/models/test_vit_pos_embed_interpolate.py

从 test/registered/models 迁到 unit/models，移除重复的 CUDA 注册并保留 AMD 设备路径，体现 CPU-only 判定的典型边界。

```
# test/registered/unit/models/test_vit_pos_embed_interpolate.py
# 纯 stub 级位精确单测：无模型权重、无分布式初始化，
# 在 CPU 上即可完成插值算术的 bit-exact 验证，因此迁到 CPU 队列；
# 原有 base-a-test-cpu 注册已生效，故删除重复的 CUDA 注册。
import torch

from sglang.test.ci.ci_register import register_amd_ci, register_cpu_ci
from sglang.test.test_utils import CustomTestCase

register_cpu_ci(est_time=20, suite="base-a-test-cpu")
# AMD 注册保留：当 GPU 存在时该测试仍然走设备路径，覆盖 ROCm 行为
register_amd_ci(est_time=20, stage="stage-a", runner_config="1-gpu-small-amd")

NUM_POS = 2304 # Qwen3-VL 的位置编码数，对应 48x48 网格
HIDDEN = 64 # 小 hidden dim 保持单测快速
MERGE = 2

# 覆盖单图 / 大上采样 / 混合 batch / 视频 / 视频 + 图片 / 多重复等场景
GRID_CASES = {
    "single": [[1, 16, 16]],
    "single_large": [[1, 64, 98]], # h, w 可超过网格边长（上采样）
    "multi_mixed": [[1, 16, 24], [1, 32, 12], [1, 8, 40]],
    "video": [[4, 16, 20]],
    "video_plus_image": [[3, 12, 16], [1, 20, 28], [2, 8, 8]],
    "many": [[1, 24, 24]] * 8,
}


def _devices():
    # CPU 上必须跑；CUDA 可用时追加 GPU 以覆盖设备路径
```

```
devs = [torch.device("cpu")]
if torch.cuda.is_available():
    devs.append(torch.device("cuda"))
return devs
```

评论区精华

本 PR 没有任何 review 评论（review_comments_count 为 0），关键设计决策均由作者在 PR body 中给出论证：

- pytest skip 与注册级 disabled 的取舍：作者明确指出 pytest-level skip 对分片不可见，文件仍在消耗预算并产生 import 开销，同时阻塞 owner 在稳定化阶段直接运行测试，因此注册标志是 canonical CI gate。
- VLM perf 缩减依据：作者称 medians are stable at that sample size and the thresholds are loose ceilings，因此只降 prompt 数而不放松断言；同时也说明"两个 bench 共用一个 server"是主要耗时来源。
- 排除迁移的判定：对未迁移文件分别给出了具体理由（memory_allocated 设备依赖、nixl 仅 CUDA runner 安装），说明并非一刀切迁移。
  - 暂无高价值评论线程

风险与影响

- 风险：
  1. 测试覆盖转移风险：CPU-only 判定基于"屏蔽 CUDA 后跑通"，可能掩盖真实 GPU 路径上的回归。test_vit_pos_embed_interpolate.py 保留 AMD 注册作为设备路径兜底，但 NVIDIA/CUDA 侧的注册被移除，后续 CUDA 特有行为失去回归覆盖。
  2. 注册语义变更风险：8 个 LoRA 文件从 pytestmark.skip 改为 registration 的 disabled= 参数，行为依赖 sglang.test.ci.ci_register 的实现；若 disabled 的收集语义与 pytest skip 不一致，可能出现"看似禁用实则仍被分片"或相反的情况，需要观察首个全量 CI 结果。
  3. 基准采样量缩减：VLM perf 的在线阶段从 250 个 prompt 减到 50 个，虽作者称中位数稳定，但未来模型或 runtime 变化导致方差增大时，可能更容易出现误报或漏报。
  4. 文件迁移的历史追踪成本：30 个文件中多个发生 rename/move，git blame 与跨版本追溯会暂时变弱。
  5. CPU 套件承载能力：迁移后 base-a-test-cpu 的测试增多，若某些测试隐式依赖 GPU 相关环境变量或路径，可能在托管 runner 上出现偶发失败。
- 影响：对普通用户无影响；对 CI 与开发流程影响显著：
  - CI 吞吐：每个 commit 在 base-b 上约节省 675 秒 GPU 队列时间，1-gpu-5090 池竞争明显缓解，真实 GPU 测试的排队等待减少；
  - 分片预算可观测性：LoRA 测试因改为注册级 disabled，分片预算不再被无效文件吞噬，CI 状态展示更准确；
  - 开发者体验：LoRA 相关 owner 可在本地直接运行这些测试文件来做稳定性修复；VLM perf 迁到 extra-a 后可通过 label 精准触发；

- 团队规范：unit/ 镜像树成为纯单测的规范位置，后续新增单测放置更清晰，也为更大规模的“仅 CPU 回归层”铺路。
- 风险标记：测试覆盖转移风险，基准采样缩减，注册语义变更，无 review 讨论

关联脉络

- PR #33637 [CI] Skip sglang-kernel and sgl-deep-gemm reinstall on version match: 同一 CI 提速主题，按版本匹配跳过不必要的重装以节省 runner 时间。
- PR #33644 [CI] Free hosted-runner disk space only when it is low: 同为 1-gpu 托管 runner 资源管理优化，减少每个 commit 的队列占用。
- PR #33619 [CI] Speed up dependency install: dual-ABI Rust ext cache and prevalidation pruning: CI 安装链路提速，与本 PR 的注册裁剪共同压缩 CI 总时长。
- PR #33596 [Test] Replace GEMM backend e2e matrices with layer-level unit tests: 用更便宜的单测替代昂贵 e2e 矩阵，与本 PR 把 CPU 测试移出 GPU 池是同一方向。