PR #33281 完整报告

sgl-project/sglang

[CI] Add MiniMax-H3 2-GPU consistency coverage

合并时间：2026-08-03 22:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33281

执行摘要

- 一句话：MiniMax-H3 新增 2 卡 H100 一致性 CI 用例
- 推荐动作：建议 CI 与 diffusion 模块负责人精读：本 PR 展示了模型一致性测试在硬件拓扑变化下的迁移与闭环完整样板，包括 memory 模式 layerwise offload 的 2 卡配置、缺失 GT 自动产物化与发布路径、按 case 过滤的分区调度。一般开发者可跳过。

功能与动机

4 卡 H100 runner 当前不可用（原 job 注释 'Temporarily disabled while the 4-gpu-h100 runner is unstable'），而 MiniMax-H3 是 diffuser 套件中的新模型。PR body 说明目标是在可用的 2 卡 H100 拓扑上保留 PR 阻塞级端到端模型一致性覆盖，并让首次缺失 GT 的运行产出可发布到固定 ci-data 仓库的 artifact。

实现拆解

1. 新增 2 卡用例：python/sglang/multimodal_gen/test/server/gpu_cases.py 的 TWO_GPU_CASES 列表头部插入 minimax_h3_t2va_2gpu_h100，模型 MiniMaxAI/MiniMax-H3，tp_size=2，通过 --performance-mode memory 与 --layerwise-offload-components dit,text_encoder,vae 等参数在 2 卡上装载完整 T2VA pipeline，并开启 run_perf_check 与 run_consistency_check。
2. 缺失 GT 产物化：test_utils.py 新增 save_missing_consistency_gt_artifact，按官方 GT 命名规则把输出帧写入 missing_consistency_gt 目录；test_server_common.py 的 _validate_consistency 在 gt_exists 失败分支先提取关键帧（视频优先 pop_realtime_key_frames，回退 extract_key_frames_from_video）再保存，使首次运行的输出可被 GT 生成工作流直接发布。
3. CI 拓扑调整：.github/workflows/pr-test-multimodal-gen.yml 删除 multimodal-gen-test-4-h100 job（原已 if: false），2 卡 job 注入 HF_TOKEN / HUGGING_FACE_HUB_TOKEN（优先 MINIMAX_H3_HF_TOKEN 以认证私有模型）；.github/workflows/diffusion-ci-gt-gen.yml 的 2gpu 生成 job 同样注入 token。
4. GT 生成按需调度：scripts/ci/utils/diffusion/compute_diffusion_partitions.py 新增 --case-ids 参数，只调度包含指定 case 的分区，未知 ID 直接报错退出；gt-gen workflow 把 inputs.case_ids 透传给该参数。

5. 基线与 pin 更新：test_utils.py 将 SGL_TEST_FILES_CI_DATA_REVISION 更新（CUDA 739c6c9…，NPU d180ad3…），perf_baselines/h100.json 写入 minimax_h3_t2va_2gpu_h100 的期望耗时（E2E 约 47.5 秒、denoise 步约 2.3 秒）。

关键文件：

- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 测试用例；类别 test；类型 test-coverage；符号 minimax_h3_t2va_2gpu_h100）：新增 minimax_h3_t2va_2gpu_h100 用例，是本次一致性覆盖迁移的核心载体，定义了 2 卡 memory 模式 + layerwise offload 的完整配置样板。
- python/sglang/multimodal_gen/test/test_utils.py（模块 测试工具；类别 test；类型 test-coverage；符号 save_missing_consistency_gt_artifact）：新增 save_missing_consistency_gt_artifact 并在缺失 GT 时保存产物，同时更新 CUDA/NPU 两条 GT revision pin，是闭环发布机制的关键。
- python/sglang/multimodal_gen/test/server/test_server_common.py（模块 服务测试；类别 test；类型 test-coverage；符号 _validate_consistency）：在 _validate_consistency 的 GT 缺失分支中提取关键帧并调用 artifact 保存逻辑，使首次运行即可产出可发布的 GT 素材。
- .github/workflows/pr-test-multimodal-gen.yml（模块 工作流；类别 infra；类型 infrastructure）：删除不可用的 4 卡 H100 job 并为 2 卡 job 注入私有模型 token，是本次 CI 拓扑调整的主入口。
- scripts/ci/utils/diffusion/compute_diffusion_partitions.py（模块 分区脚本；类别 infra；类型 infrastructure；符号 main）：新增 --case-ids 参数，GT 生成工作流可按需只调度包含目标 case 的分区，显著收敛生成耗时。
- .github/workflows/diffusion-ci-gt-gen.yml（模块 工作流；类别 infra；类型 infrastructure）：GT 生成工作流透传 inputs.case_ids 并注入 HF token，实现按需生成私有模型 GT。
- python/sglang/multimodal_gen/test/server/perf_baselines/h100.json（模块 性能基线；类别 test；类型 test-coverage）：写入 minimax_h3_t2va_2gpu_h100 的期望性能基线，支撑用例的 run_perf_check。

关键符号：save_missing_consistency_gt_artifact, _validate_consistency, main

关键源码片段

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

新增 minimax_h3_t2va_2gpu_h100 用例，是本次一致性覆盖迁移的核心载体，定义了 2 卡 memory 模式 + layerwise offload 的完整配置样板。

```
    # minimax_h3_t2va_2gpu_h100：2 卡 H100 上的 MiniMax-H3 T2VA 一致性用例。
    # 原 4 卡 runner 不可用，改用 memory 模式 + layerwise offload，让完整
    # 原生 pipeline（dit / text_encoder / vae）在 2 卡上放下，并开启
    # perf 与 consistency 双重校验。
    DiffusionTestCase(
        "minimax_h3_t2va_2gpu_h100",
        DiffusionServerArgs(
```

```
            model_path="MiniMaxAI/MiniMax-H3",
            modality="video",
            tp_size=2,
            ulysses_degree=1,
            extras=[
                "--model-variant", "fl2va",
                "--performance-mode", "memory",
                "--layerwise-offload-components", "dit,text_encoder,vae",
                "--dit-offload-prefetch-size", "1",
                "--dit-layerwise-resident-layers", "20",
                "--enable-torch-compile", "false",
            ],
        ),
        DiffusionSamplingParams(
            prompt=(
                "A static night view of a narrow London alley in soft rain, wet "
                "pavement reflecting a yellow streetlamp, the blue K. West sign "
                "glowing above a doorway, cardboard boxes near the wall, a pale "
                "parked car in the distance, and a slender glam-rock figure "
                "holding a guitar under the lamp, brick storefronts, muted teal "
                "and amber colors, subtle rain shimmer only."
            ),
            output_size="1344x768",
            seconds=4,
            output_format="mp4",
            num_outputs_per_prompt=1,
            extras={
                "task": "t2va",
                "conditions": [],
                "target": {
                    "short_edge": 768,
                    "aspect_ratio": "16:9",
                    "duration_seconds": 4.0,
                },
                "num_inference_steps": 8,
                "flow_shift": 12.0,
                "audio_flow_shift": 3.0,
                "seed": 42,
            },
        ),
        run_perf_check=True,
        run_consistency_check=True,
        run_component_accuracy_check=False,
        run_models_api_check=False,
        run_t2v_input_reference_check=False,
    ),
```

python/sglang/multimodal_gen/test/test_utils.py

新增 save_missing_consistency_gt_artifact 并在缺失 GT 时保存产物，同时更新 CUDA/NPU 两条 GT revision pin，是闭环发布机制的关键。

```
def save_missing_consistency_gt_artifact(
    artifact_dir: str | Path | None,
    case_id: str,
    num_gpus: int,
    output_frames: list[np.ndarray],
    is_video: bool,
    output_format: str | None = None,
) -> Path | None:
    # 未配置 artifact 目录时直接返回 None，调用方按无产物处理
    if not artifact_dir:
        return None

    # 统一落在 missing_consistency_gt 子目录，GT 生成工作流可直接发布该目录
    out_dir = Path(artifact_dir) / "missing_consistency_gt"
    out_dir.mkdir(parents=True, exist_ok=True)

    # 复用官方 GT 的命名规则，保证产物可无改动地发布到 ci-data-diffusion
    filenames = _consistency_gt_filenames(
        case_id,
        num_gpus,
        is_video=is_video,
        output_format=output_format,
    )
    # 视频用例按首帧 / 中帧 / 末帧分别落盘，图片用例仅保存单帧
    for frame, filename in zip(output_frames, filenames):
        Image.fromarray(_ensure_rgb_uint8_image(frame)).save(out_dir / filename)
    return out_dir
```

python/sglang/multimodal_gen/test/server/test_server_common.py

在 _validate_consistency 的 GT 缺失分支中提取关键帧并调用 artifact 保存逻辑，使首次运行即可产出可发布的 GT 素材。

```
        # 本地无 GT 时：把本次生成帧落盘为 artifact，供后续发布为正式 GT。
        # 视频走实时关键帧（优先），回退到从视频字节中抽取；图片直接转 numpy。
        if not gt_exists(
            case.id, num_gpus, is_video=is_video, output_format=output_format
        ):
            if is_video:
                output_frames = pop_realtime_key_frames(case.id)
                if output_frames is None:
                    output_frames = extract_key_frames_from_video(content)
            else:
                output_frames = [image_bytes_to_numpy(content)]
            artifact_path = save_missing_consistency_gt_artifact(
                artifact_dir=os.environ.get("SGLANG_DIFFUSION_ARTIFACT_DIR"),
                case_id=case.id,
```

```
            num_gpus=num_gpus,
            output_frames=output_frames,
            is_video=is_video,
            output_format=output_format,
        )
        if artifact_path is not None:
            logger.info("[Artifact] Saved missing consistency GT: %s", artifact_path)
```

评论区精华

本 PR 没有实质性 review 讨论（review comments 与审核记录均为空；2 条 issue 评论来自 gemini-code-assist bot 的 sunset 通告）。验证证据集中在 PR body：GT 生成 workflow 只运行了 minimax_h3_t2va_2gpu_h100，三帧质量门通过，发布 commit d180ad3…；标准 PR 一致性 job 对固定 revision 通过，三帧 CLIP 1.0000、SSIM 1.0000、PSNR inf、mean absolute difference 0.0000。可留意的潜在维护点：CI 中 CUDA/NPU 的 GT pin 需随发布同步（代码注释也强调 REPO 与 REVISION 必须一起更新）。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 覆盖缺口：移除 4 卡 job 后 MiniMax-H3 4 卡路径无 CI 保护，runner 恢复后需重建。
  2. 凭据依赖：MiniMax-H3 是私有模型，2 卡 job 强依赖 MINIMAX_H3_HF_TOKEN 等 secret，token 失效会导致 CI 假失败。
  3. GT pin 双线：CUDA 与 NPU 各维护一个 SGL_TEST_FILES_CI_DATA_REVISION，若 ci-data-diffusion 分支后续发布未同步更新此处，一致性检查会误报 MISSING GT。
  4. perf 基线首建：h100.json 中 minimax_h3_t2va_2gpu_h100 基线为首次写入且 per_frame_generation 为 null，机器差异可能触发 perf 失败。
  5. artifact 累积：缺失 GT 时每次失败都会写文件到 SGLANG_DIFFUSION_ARTIFACT_DIR，无保留策略，长跑会累积磁盘占用。 - 影响：影响面集中在 diffusion 多模态 CI 链路：PR 阻塞的一致性检查从 4 卡迁移到 2 卡继续生效；GT 生成 workflow 支持按需调度，减少无效算力消耗；对维护者新增了私有模型 token 与 GT pin 两个需要同步维护的配置点。对用户运行时无影响，本 PR 无 srt 运行时源码改动。 - 风险标记：4 卡覆盖缺口，私有模型凭据依赖，GT pin 需同步维护，perf 基线首建，artifact 无清理策略

关联脉络

- PR #33282 docs(diffusion): update skills for MiniMax-H3: 同一 MiniMax-H3 T2VA 基准线与测试技能链，本 PR 落地的 2 卡一致性用例与 Docs 中的 benchmark preset 配套。
- PR #33345 [Docs] Fix overlapping quality-profile table headers on MiniMax-H3 page: 同属 MiniMax-H3 在 diffusion 套件中的持续集成与文档建设。
- PR #33329 [CI] Size the CPU stage from the live partition model: 同属 CI 分区计算与工作流容量治理方向，compute_diffusion_partitions.py 与分区输出机制持续演进。