PR #35997 完整报告

sgl-project/sglang

[diffusion] fix: stabilize LTX-2.3 two-stage cold requests

合并时间：2026-08-23 10:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35997

执行摘要

- 一句话：LTX-2.3 多 GPU 预热帧数升至 25，稳定冷启动首请求
- 推荐动作：值得精读 warmup_request_builder.py 中的 _resolve_warmup_num_frames：它演示了“让 warmup 覆盖 serving 真实 shape”这一通用原则，以及如何用 pipeline 名称与 num_gpus 做差异化帧数预算。缺点是逻辑目前只对 LTX2 两阶段多 GPU 生效，未来类似的多阶段管线（upsample/refine）可考虑提取更通用的规则；建议结合 PR body 的性能表理解冷启动开销的量级。

功能与动机

PR body 指出：多 GPU LTX-2.3 两阶段 CI 用例在服务器 warmup 后，首次请求间歇性承担 shape setup 开销；通用 17 帧 warmup 产生 temporal latent size 3，而常见的 1 秒双 GPU 请求从 24 帧对齐到 25 帧、产生 temporal latent size 4。作者明确排除了 denoising 吞吐回归（失败尝试 8.01s vs 成功重试 8.12s），并说明额外延迟出现在两阶段转换处，根因是 warmup 没有覆盖 serving 请求的真实 latent shape。

实现拆解

1. 新增专用帧数预算：在 python/sglang/multimodal_gen/runtime/warmup_request_builder.py 顶部新增常量 SERVER_WARMUP_LTX2_TWO_STAGE_MAX_VIDEO_FRAMES = 25，与通用 SERVER_WARMUP_MAX_VIDEO_FRAMES = 17 并列，作为 LTX2 两阶段多 GPU 的 warmup 帧数上限。
2. 修改帧数选择逻辑：在 _resolve_warmup_num_frames 的 server_based_warmup 分支内，通过 is_ltx2_two_stage_pipeline_name(server_args.pipeline_class_name) and server_args.num_gpus > 1 判断使用 25 还是 17 作为 frame_budget，随后统一走 min(num_frames, frame_budget) 和 pipeline_config.adjust_num_frames(...) 对齐。这样 warmup 请求的 temporal latent shape 与首个双 GPU 服务请求一致，避免 CUDA graph / kernel 在首次请求时重新 specialize。
3. 单元测试同步加固：python/sglang/multimodal_gen/test/unit/test_cfg_parallel_warmup.py 中为 test_ltx2_two_stage_warmup_uses_pipeline_alignment 补充 num_gpus = 2 与 adjust_num_frames.return_value = 25，断言 reqs[0].num_frames == 25；新增 test_ltx2_two_stage_single_gpu_keeps_generic_frame_cap，用 121 帧采样默认值验证单 GPU 仍回落 17；同时给 test_video_warmup_preserves_model_frame_alignment 补 pipeline_class_name = None，防止 SimpleNamespace 缺属性导致误判。

4. 刷新性能基线：python/sglang/multimodal_gen/test/server/perf_baselines/h100.json 中仅把 ltx_2_3_two_stage_ti2v_2gpus 的 LTX2UpsampleStage 从 2.31ms 更新为 172.53ms。作者用冷缓存测量了 unmodified main（174.51ms）与 patched（170.40-173.56ms），证明旧基线失真而非性能回归；E2E、denoise、VRAM、一致性阈值均未改动。

关键文件：

- python/sglang/multimodal_gen/runtime/warmup_request_builder.py（模块 预热请求；类别 source；类型 core-logic；符号 _resolve_warmup_num_frames）：核心源码变更，直接决定 warmup 请求的帧数上限与 serving latent shape 是否一致，是本次修复的关键路径。
- python/sglang/multimodal_gen/test/unit/test_cfg_parallel_warmup.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_ltx2_two_stage_warmup_uses_pipeline_alignment, test_ltx2_two_stage_single_gpu_keeps_generic_frame_cap, test_video_warmup_preserves_model_frame_alignment）：覆盖多 GPU 25 帧与单 GPU 17 帧两条分支，防止 warmup 策略回退。
- python/sglang/multimodal_gen/test/server/perf_baselines/h100.json（模块 性能基线；类别 test；类型 baseline-update）：刷新过时的阶段耗时基线，避免 CI 性能用例误报回归。

关键符号：_resolve_warmup_num_frames

关键源码片段

python/sglang/multimodal_gen/runtime/warmup_request_builder.py

核心源码变更，直接决定 warmup 请求的帧数上限与 serving latent shape 是否一致，是本次修复的关键路径。

```
# 通用视频 warmup 帧数上限：保持启动开销有界
SERVER_WARMUP_MAX_VIDEO_FRAMES = 17
# LTX2 两阶段多 GPU 专用上限：覆盖 1 秒请求对齐后的 25 帧形态
SERVER_WARMUP_LTX2_TWO_STAGE_MAX_VIDEO_FRAMES = 25


def _resolve_warmup_num_frames(
    server_args: ServerArgs,
    sampling_defaults: SamplingParams,
    *,
    server_based_warmup: bool,
) -> int:
    num_frames = getattr(sampling_defaults, "num_frames", 1)
    if not _is_video_warmup_task(server_args) or num_frames is None:
        return num_frames

    # breakable CUDA graph 只按精确 latent shape 回放：
    # warmup 必须跑完整服务帧数，否则捕获的 graph 签名与 serving 不匹配
    #  （这与 _resolve_warmup_steps 中 steps 不设上限的规则同理）。
    if (
```

```
            not server_based_warmup
            or getattr(server_args, "enable_breakable_cuda_graph", False) is True
        ):
            warmup_num_frames = num_frames
        else:
            # 多 GPU LTX2 两阶段会把 1 秒请求从 24 帧对齐到 25 帧（temporal latent size 4）；
            # 若 warmup 仍用通用 17 帧（latent size 3），首个请求会在两阶段切换处触发额外 setup。
            frame_budget = (
                SERVER_WARMUP_LTX2_TWO_STAGE_MAX_VIDEO_FRAMES
                if is_ltx2_two_stage_pipeline_name(server_args.pipeline_class_name)
                and server_args.num_gpus > 1
                else SERVER_WARMUP_MAX_VIDEO_FRAMES
            )
            # 采样默认帧数可能远大于预算，取较小值以控制启动开销
            warmup_num_frames = min(num_frames, frame_budget)

        return server_args.pipeline_config.adjust_num_frames(warmup_num_frames)
```

评论区精华

本 PR 没有实质 review 评论，也没有挂关联 Issue。唯一的 issue 评论是作者 mickqian 触发的 /tag-and-rerun-ci，用于重跑 CI。技术论证全部体现在 PR body 的实测数据中：失败尝试 8.01s vs 成功重试 8.12s 排除了 denoising 吞吐回归；unmodified main 冷缓存实测 LTX2UpsampleStage 为 174.51ms，而旧基线是 2.31ms，说明基线本身已失真。没有未解决的公开疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 行为变更范围：仅影响 server_based_warmup=True、enable_breakable_cuda_graph=False 且 num_gpus > 1 的 LTX2 两阶段管线；单 GPU 及其他视频管线全部保持 17 帧，且有单测兜底。
  - 显存与启动时间：25 帧 warmup 会让峰值 VRAM 从 60,272MiB 微升到 60,728-60,772MiB（约 0.8%），warmup 本身也更长；显存余量紧张的多卡部署需要留意。
  - 基线更新风险：LTX2UpsampleStage 从 2.31ms 跳到 172.53ms 数值跨度大，但 PR 用 unmodified-main 冷缓存实测证明是旧基线失真；若其他 GPU 型号（如 A100、5090）也有同类基线且存在相同的首请求 shape setup 问题，需要相应同步刷新，否则相关性能用例可能误报回归。
  - 假设耦合：25 帧绑定“1 秒请求对齐到 25 帧”的 serving 形态；如果未来 LTX2 的 adjust_num_frames 对齐规则或默认帧数变化，该常量需要重新评估。
  - 测试边界：num_gpus > 1 的判定只被 num_gpus = 2 的用例覆盖，4/8 卡等更大规模没有专门测试，但逻辑上一致。
- 影响：

- 用户侧：多 GPU LTX-2.3 两阶段首次请求延迟显著下降（E2E 从 10.73s 降至 9.01-9.52s），首次 denoise/refinement 步骤耗时减半以上，冷启动体验明显改善。
- 系统 /CI 侧：消除了冷启动后首请求 shape setup 的间歇性抖动，CI 稳定性提升；刷新后的 h100 基线避免性能用例误报。
- 团队侧：源码仅 11 行改动加测试 / 基线配套，回归面小；同时为后续新增视频管线提供了一种“按管线类型与 GPU 数差异化 warmup 预算”的范式。
- 风险标记：warmup 帧数策略变更，多 GPU 专属分支，性能基线更新 2.31→172.53ms，峰值 VRAM 微升

关联脉络

- PR #36032 test: the 5090 consumer case runs two warm requests on the full recipe: 同为 diffusion 服务端 warmup/ 冷请求稳定性方向，5090 用例连发两次 warm 请求以消除首请求抖动，与本 PR 目标一致。
- PR #36051 [diffusion] CI: guard the anonymous-host budget alongside peak VRAM: 同为 diffusion CI 稳定性加固，涉及峰值 VRAM 预算守卫，与本 PR 的冷启动与显存影响评估相关。