# PR #36485 完整报告

sgl-project/sglang

[diffusion] align video BCG warmup frame count

合并时间：2026-08-27 20:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36485

## 执行摘要

- 一句话：新增 --warmup-num-frames，对齐视频 BCG 预热与 serving 帧数
- 推荐动作：值得短读。重点关注三处设计决策：_resolve_warmup_num_frames 中显式覆写与 MagicMock 兼容的写法；_adjust_warmup 将帧数纳入“显式形状”推断的联动逻辑；build_sglang_cmd 中 preset 参数到服务参数的透传模式。这些模式对后续增加更多 warmup 形状配置有参考价值。

## 功能与动机

PR body 明确指出问题根因：Breakable CUDA graphs replay only exact latent shapes。SANA-Video 模型默认 81 帧，而入库的 benchmark preset 请求 17 帧，因此 warmup 捕获了 (1, 16, 21, 60, 104)，serving 请求却是 (1, 16, 5, 60, 104)，BCG 日志出现 serving-signature miss 并回退 eager 执行，性能收益完全丢失。修复方式是提供显式帧数覆写，使 warmup 与 serving 使用完全相同的帧数。

## 实现拆解

1. 配置入口（server_args.py）：在 ServerArgs 中新增 warmup_num_frames: int | None = None 字段，并在 add_cli_args 注册 --warmup-num-frames 参数。_adjust_warmup 增加正数校验（<= 0 抛 ValueError），并把“显式 warmup 形状”的判定从仅 warmup_resolutions 扩展为分辨率或帧数任一非空，从而在 warmup_mode 为 off 时自动提升为 request 模式。
2. 预热请求构建（warmup_request_builder.py）：_resolve_warmup_num_frames 优先读取 server_args.warmup_num_frames，且只接受具体 int 实例（规避 MagicMock 缺失属性被误判为配置）；随后仍走既有分支——BCG 或非 server warmup 使用完整帧数，普通 server warmup 按 SERVER_WARMUP_MAX_VIDEO_FRAMES 帧预算封顶——最后统一经过 pipeline_config.adjust_num_frames 做模型专属时序对齐。
3. 基准脚本透传（bench_diffusion_denoise.py）：build_sglang_cmd 在用户未显式传 --warmup-num-frames 且 preset 带 num-frames 时自动追加该参数，保证 BCG 基准跑出的 warmup 与 serving 帧数一致。
4. 测试与文档配套：test_server_args.py 新增 test_num_frames_forces_warmup_on 与 test_num_frames_must_be_positive；test_cfg_parallel_warmup.py 新增 test_breakable_cuda_graph_uses_explicit_warmup_num_frames（验证显式 17 覆盖模型默认 81，且 adjust_num_frames 只调用一次）；test_diffusion_benchmark_skill.py

断言 sana-video BCG 命令包含 --warmup-num-frames 17；两份 benchmark 技能文档同步更新。

关键文件：

- python/sglang/multimodal_gen/runtime/server_args/server_args.py（模块 服务参数；类别 source；类型 core-logic；符号 ServerArgs, warmup_num_frames, _adjust_warmup, add_cli_args）：配置入口：新增 warmup_num_frames 字段、CLI 参数与正数校验，并让显式帧数与 warmup_resolutions 一样能自动开启 request 模式 warmup。
- python/sglang/multimodal_gen/runtime/warmup_request_builder.py（模块 启动预热；类别 source；类型 core-logic；符号 _resolve_warmup_num_frames）：核心逻辑：_resolve_warmup_num_frames 支持显式帧数覆写，仍经 adjust_num_frames 保持模型时序对齐；只接受具体 int 以避免 MagicMock 误判。
- python/sglang/multimodal_gen/test/unit/test_server_args.py（模块 参数测试；类别 test；类型 test-coverage；符号 test_num_frames_forces_warmup_on, test_num_frames_must_be_positive）：新增两个单测，覆盖“帧数强制开启 warmup”和“非正数报错”两条新行为。
- python/sglang/multimodal_gen/test/unit/test_cfg_parallel_warmup.py（模块 预热测试；类别 test；类型 test-coverage；符号 test_breakable_cuda_graph_uses_explicit_warmup_num_frames）：验证 BCG 场景下显式 warmup_num_frames=17 覆盖模型默认 81 帧，并只调用一次 adjust_num_frames(17)。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/scripts/bench_diffusion_denoise.py（模块 基准脚本；类别 source；类型 core-logic；符号 build_sglang_cmd）：基准脚本在 BCG 模式下自动把 preset 的 num-frames 透传为 --warmup-num-frames，避免人工遗漏导致 warmup 与 serving 帧数不一致。
- python/sglang/multimodal_gen/test/unit/test_diffusion_benchmark_skill.py（模块 基准测试；类别 test；类型 test-coverage；符号 test_quality_and_bcg_comparators_are_explicit_and_exclusive）：为 sana-video BCG 命令断言 --warmup-num-frames 17 已正确注入。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/benchmark-and-profile.md（模块 基准文档；类别 docs；类型 documentation）：基准与性能分析文档同步更新，说明 BCG 模式下 warmup 帧数会自动沿用视频 preset。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：Claude 技能文档同步更新，避免技能使用者按旧命令手动指定帧数。

关键符号：_resolve_warmup_num_frames, _adjust_warmup, add_cli_args, build_sglang_cmd

## 关键源码片段

### python/sglang/multimodal_gen/runtime/server_args/server_args.py

配置入口：新增 warmup_num_frames 字段、CLI 参数与正数校验，并让显式帧数与 warmup_resolutions 一样能自动开启 request 模式 warmup。

## server_args.py：帧数校验与 warmup mode 推断

```
def _adjust_warmup(self):
    # 校验 canonical warmup mode 取值。
    if self.warmup_mode is not None and self.warmup_mode not in WARMUP_MODES:
        raise ValueError(
            f'Invalid --warmup-mode {self.warmup_mode!r}; '
            f'expected one of {WARMUP_MODES}.'
        )
    # 新配置项：帧数必须是正整数，否则直接拒绝启动。
    if self.warmup_num_frames is not None and self.warmup_num_frames <= 0:
        raise ValueError('--warmup-num-frames must be a positive integer.')

    # torch.compile 首次请求会付编译延迟，未显式指定时自动开启
    # server warmup，让第一个真实请求不被打断。
    if self.enable_torch_compile and self.warmup_mode is None:
        self.warmup_mode = 'server'
        logger.info(
            'Automatically enabled server warmup for torch.compile so first '
            'real requests do not pay compile latency. Set --warmup-mode off '
            'to disable this behavior.'
        )

    # 显式 warmup 形状（分辨率或帧数）需要一个请求路径，除非已有
    # server 默认的合成启动请求；否则在 off 时提升为 request 模式。
    if (
        self.warmup_resolutions is not None or self.warmup_num_frames is not None
    ) and self.warmup_mode in (None, 'off'):
        self.warmup_mode = 'request'

    # 后续 BCG 强制 server warmup、disagg 角色互斥等逻辑保持不变，
    # 本 PR 只扩展了“显式形状”的判定条件。
```

## python/sglang/multimodal_gen/runtime/warmup_request_builder.py

核心逻辑：_resolve_warmup_num_frames 支持显式帧数覆写，仍经 adjust_num_frames 保持模型时序对齐；只接受具体 int 以避免 MagicMock 误判。

## warmup_request_builder.py：显式帧数如何进入 warmup 请求

```
def _resolve_warmup_num_frames(
    server_args: ServerArgs,
    sampling_defaults: SamplingParams,
    *,
    server_based_warmup: bool,
) -> int:
    # 先取模型默认帧数；非视频任务直接返回，不进入 warmup 特殊逻辑。
    default_num_frames = getattr(sampling_defaults, 'num_frames', 1)
```

```
    if not _is_video_warmup_task(server_args):
        return default_num_frames

    # 显式覆写只接受具体整数：测试或轻量集成常使用 MagicMock
    # 构造 server args，缺失属性会解析成另一个 mock，不能当作配置值。
    explicit_num_frames = getattr(server_args, 'warmup_num_frames', None)
    num_frames = (
        explicit_num_frames
        if isinstance(explicit_num_frames, int)
        else default_num_frames
    )
    if num_frames is None:
        return num_frames

    # BCG 只重放完全一致的 latent shape：warmup 请求必须携带 serving
    # 的真实帧数，capture 出的图才能命中 serving 签名（与
    # _resolve_warmup_steps 不封顶步数的规则同理）。
    if (
        not server_based_warmup
        or getattr(server_args, 'enable_breakable_cuda_graph', False) is True
    ):
        warmup_num_frames = num_frames
    else:
        # 普通 server warmup 按帧预算封顶，例如多卡 LTX 两阶段
        # 会把一秒请求对齐到 25 帧，只需覆盖其 latent shape。
        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
        )
        warmup_num_frames = min(num_frames, frame_budget)

    # 最后交给 pipeline 做模型专属时间维对齐（像素帧到 latent 帧换算）。
    return server_args.pipeline_config.adjust_num_frames(warmup_num_frames)
```

## 评论区精华

本 PR 没有公开 review 评论线程（review_comments_count = 0），设计取舍主要体现在代码注释中：一是显式覆写只接受具体 int，避免 MagicMock 缺失属性被当作配置，这是面向测试友好的兼容设计；二是帧数覆写仍经过各 pipeline 的 adjust_num_frames，保证模型专属的时间维对齐不被破坏；三是 quality=high + BCG 仍是 SANA-Video 不支持组合，由既有运行时 guard 拒绝，本 PR 不改变该边界。

- 暂无高价值评论线程

## 风险与影响

- 风险：

1. 显存峰值上升：RTX PRO 6000 验证中 reserved memory 从 20,470 MB 升至 21,540 MB（约 +5.2%），warmup 使用与 serving 一致的 latent shape 后显存占用可能变化，显存紧张环境需实测确认。
2. 平台验证缺口：AMD ROCm 7.2 CI 失败（Run #33051917814），大概率与本改动无关（未触碰 ROCm 路径），但该平台尚未得到绿验证。
3. 静默回退风险：_resolve_warmup_num_frames 只认 Python int 作为显式覆写；若未来配置源传入其他数值类型（如 numpy 标量），覆写会被静默忽略并回退模型默认，可能再次产生签名 miss。
4. 行为兼容性：未设置新参数时行为与之前一致；普通 server warmup 仍受 SERVER_WARMUP_MAX_VIDEO_FRAMES 帧预算封顶，显式传大帧数时需确认该封顶逻辑符合预期。 - 影响：影响范围集中于 multimodal_gen 视频生成 warmup 路径：用户侧，SANA-Video 等视频模型在 BCG 模式下可手动指定 warmup 帧数，规避 eager 回退；系统侧，server args 校验与 warmup 请求构建新增一个可选配置，未设置时保持模型默认行为，向后兼容；团队侧，diffusion benchmark 技能自动透传帧数，减少人工遗漏，文档同步更新。整体影响程度中等，不涉及核心推理算子或跨模块重构。 - 风险标记：BCG warmup 显存峰值上升约 5%, AMD ROCm CI 未通过待确认，显式覆写仅认 int，其他数值类型会静默回退，普通 server warmup 仍受帧预算封顶

# 关联脉络

- PR #36553 [diffusion] Accept mesh benchmark artifacts: 与本 PR 改动同一批文件（bench_diffusion_denoise.py、test_diffusion_benchmark_skill.py 及 benchmark 技能文档），共同演进 diffusion 基准工具链。
- PR #36571 [diffusion] Fuse Cosmos3 Nano T2I attention on Hopper: 同为 diffusion 视频模型在 Hopper 上的 BCG/ 图融合加速方向，验证 BCG 捕获与 serving 签名对齐的重要性。
- PR #36592 [Diffusion][Kernel] Fuse Wan FFN GELU epilogue: 同为 diffusion 性能优化系列（BCG/ 内核融合），体现该方向持续追求降低 denoise 延迟。