PR #29519 完整报告

sgl-project/sglang

[diffusion] warmup: default to model sampling resolution (declare Z-Image default)

合并时间：2026-06-30 10:32

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29519

执行摘要

- 一句话：图像预热默认使用模型分辨率，声明 Z-Image 默认 1024x1024
- 推荐动作：本 PR 设计清晰，改动量小但成效明显，值得阅读 _resolve_default_warmup_resolution 的新逻辑和 ZImageTurboSamplingParams 的配置声明。对于团队，建议在新模型配置中统一声明 width/height，使预热自动受益。

功能与动机

PR body 指出服务器端图像预热默认使用面积上限的"代表性"分辨率（SERVER_WARMUP_IMAGE_MAX_AREA=768x768），对于更大真实请求（如1024x1024），第一次请求仍然支付首形状内核自动调优开销（H100 上约 0.1s），即使预热已经运行。Z-Image 之前没有声明默认分辨率，因此 fallback 到上限。本 PR 让预热使用模型采样默认值（最可能的真实请求形状），消除该残余冷启动。

实现拆解

1. 修改预热分辨率选择逻辑（warmup_request_builder.py::_resolve_default_warmup_resolution）：当 server_based_warmup=True 且任务类型为图像生成时，不再直接调用 _resolve_representative_warmup_resolution 做面积缩放，而是优先返回模型 SamplingParams 中的 width/height（即模型的采样默认分辨率）。若模型未声明默认分辨率或不是图像生成，则回退到代表性的分辨率选择（面积缩放）。
2. 声明 Z-Image 默认分辨率（zimage.py::ZImageTurboSamplingParams）：取消之前注释掉的 height/width 字段，显式声明 height: int = 1024 和 width: int = 1024，并添加注释"Z-Image officially recommends starting at 1024x1024"。注意 ZImageSamplingParams 未声明分辨率（保持无默认），因为非 Turbo 版本可能使用其他分辨率。
3. 更新单元测试（test_cfg_parallel_warmup.py）：修改三个测试用例名和期望值，反映预热现在使用模型默认分辨率（1024x1024）而不是缩小的面积上限（512x512 或 768x768）。新增 test_server_based_image_warmup_uses_model_default_over_supported 和 test_server_based_image_warmup_uses_full_model_default 等，验证在 supported_resolutions 存在或仅默认分辨率时，预热都选择模型默认值。
4. 调整性能基线（perf_baselines.json）：更新 flux_2_klein_image_t2i 和 fsdp-inference 的 denoise_step_ms，因为预热分辨率变化导致各步时间分布改变，需重新基线化。

关键文件：

- python/sglang/multimodal_gen/runtime/warmup_request_builder.py（模块 预热逻辑；类别 source；类型 core-logic；符号 _resolve_default_warmup_resolution, _resolve_representative_warmup_resolution）：核心逻辑所在，修改了 _resolve_default_warmup_resolution 函数，改变图像预热的分辨率选择策略：从一律面积缩放改为优先使用模型默认分辨率。
- python/sglang/multimodal_gen/configs/sample/zimage.py（模块 模型配置；类别 source；类型 configuration；符号 ZImageTurboSamplingParams）：声明 Z-Image 默认分辨率 1024x1024，使预热能正确获取模型默认值，解决之前未声明导致的 fallback 问题。
- 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_server_based_image_warmup_uses_model_default_over_supported, test_server_based_image_warmup_uses_full_model_default, test_server_based_image_warmup_diffusers_uses_model_default）：更新测试用例以验证新预热行为，确保图像预热使用模型默认分辨率而非面积上限。测试名称和期望值均反映变更。
- python/sglang/multimodal_gen/test/server/perf_baselines.json（模块 性能基线；类别 test；类型 test-coverage）：调整性能基线以匹配新预热分辨率导致的时序变化，保持测试与 CI 基准一致。

关键符号：_resolve_default_warmup_resolution, _resolve_representative_warmup_resolution

关键源码片段

python/sglang/multimodal_gen/runtime/warmup_request_builder.py

核心逻辑所在，修改了 _resolve_default_warmup_resolution 函数，改变图像预热的分辨率选择策略：从一律面积缩放改为优先使用模型默认分辨率。

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

def _resolve_default_warmup_resolution(
    server_args: ServerArgs,
    sampling_defaults: SamplingParams,
    *,
    server_based_warmup: bool,
) -> tuple[int, int]:
    """Return the default warmup resolution.

    Prefer the model's sampling-default resolution — the most likely real
    request shape — so warmup specializes kernels for it. Server-based image
    warmup used to shrink this to an area cap (``SERVER_WARMUP_IMAGE_MAX_AREA``,
    768x768) to bound startup, but that left a residual first-request
    cold-start when the real request is larger (e.g. 1024x1024 paid ~0.1s of
    first-shape kernel autotuning, measured on H100).
    """
    width = sampling_defaults.width
```

```
    height = sampling_defaults.height
    # 只在图像生成时优先使用模型默认分辨率
    is_image_gen = server_args.pipeline_config.task_type.is_image_gen()
    if (
        width is not None
        and height is not None
        and (not server_based_warmup or is_image_gen) # 非服务器预热或图像生成时直接使用
    ):
        return width, height

    # 服务器预热且非图像生成（视频等），或模型未声明默认分辨率时走 representative 逻辑
    if server_based_warmup:
        return _resolve_representative_warmup_resolution(server_args, sampling_defaults)

    # 以下为客户端预热的 fallback
    supported_resolutions = sampling_defaults.supported_resolutions
    if supported_resolutions:
        return min(supported_resolutions, key=lambda size: size[0] * size[1])

    if server_args.pipeline_config.task_type.is_image_gen():
        return DEFAULT_LIGHTWEIGHT_IMAGE_RESOLUTION

    return (
        width or DEFAULT_LIGHTWEIGHT_IMAGE_RESOLUTION[0],
        height or DEFAULT_LIGHTWEIGHT_IMAGE_RESOLUTION[1],
    )
```

python/sglang/multimodal_gen/configs/sample/zimage.py

声明 Z-Image 默认分辨率 1024x1024，使预热能正确获取模型默认值，解决之前未声明导致的 fallback 问题。

```
# python/sglang/multimodal_gen/configs/sample/zimage.py
from dataclasses import dataclass, field
from sglang.multimodal_gen.configs.sample.sampling_params import SamplingParams
from sglang.multimodal_gen.configs.sample.teacache import TeaCacheParams


@dataclass
class ZImageTurboSamplingParams(SamplingParams):
    num_inference_steps: int = 9
    num_frames: int = 1
    negative_prompt: str = None
    # Z-Image officially recommends starting at 1024x1024
    height: int = 1024 # 之前被注释掉，现在显式声明以让预热使用正确分辨率
    width: int = 1024
    guidance_scale: float = 0.0
    cfg_normalization: float | bool = False

    teacache_params: TeaCacheParams = field(
```

```
        default_factory=lambda: TeaCacheParams(
            teacache_thresh=0.15,
            coefficients=[
                7.33226126e02,
                -4.01131952e02,
                6.75869174e01,
                -3.14987800e00,
                9.61237896e-02,
            ],
        )
    )


@dataclass
class ZImageSamplingParams(SamplingParams):
    num_inference_steps: int = 50
    num_frames: int = 1
    negative_prompt: str = " "
    guidance_scale: float = 5.0
    cfg_normalization: float | bool = True
    # 注意：非 Turbo 版本未声明默认分辨率，预热将回退到其他策略
```

评论区精华

Gemini Code Assist 在 warmup_request_builder.py 第 84 行评论：is_image_gen 变量在 server_based_warmup=False 时仍被无条件求值，可能引起属性错误（当 server_args 被 mock或部分初始化时）。建议利用Python短路求值，将is_image_gen的调用移到条件内部。作者未回复也未采纳建议，可能是由于当前所有测试路径中 server_args.pipeline_config.task_type 均存在，短期无问题。该建议虽合理，但实际影响有限。

- is_image_gen 无条件求值可能带来属性错误 (correctness): 作者未回应，PR 已合并。当前测试未触发该问题，但理论上存在风险。建议后续优化。

风险与影响

- 风险：
  1. 回归风险：对未声明默认分辨率的图像模型，预热分辨率选择行为从 area-cap fallback 变为 representative 选择（_resolve_representative_warmup_resolution），该函数内部也考虑模型默认值和 supported_resolutions，但最终可能回退到 DEFAULT_LIGHTWEIGHT_IMAGE_RESOLUTION，与旧行为略有差异。测试覆盖了多数场景，但边缘情况（如既无 width/height 又无 supported_resolutions）需额外关注。
  2. 兼容性风险：用户若依赖预热在特定面积下运行（如为了控制显存），现在预热可能使用更大分辨率（如 1024x1024），增加启动时显存占用。但预热本身旨在编译内核，显存增加是临时的，且用户可通过 --warmup-resolutions 显式指定。
  3. 视频预热不受影响：视频生成仍使用面积 / 帧数限制，风险隔离。
- 影响：

1. 用户影响：所有使用服务器端图像预热（serve --warmup 且未指定 --warmup-resolutions）的用户都会受益：首请求延迟降低约 0.1s（H100），预热后的第一次推理不再有内核编译冷启动。Z-Image 用户之前未声明默认分辨率导致预热带回退到 512x512，现在正确使用 1024x1024，影响显著。
2. 系统影响：预热分辨率变大会略微增加启动时显存和计算开销，但通常忽略不计，因为预热步骤数很少（2 步）。
3. 团队影响：明确了预热分辨率的选择策略，后续添加新模型时需确保其 SamplingParams 正确声明默认分辨率。 - 风险标记：核心预热逻辑变更，模型默认配置变更，潜在兼容性影响（显存占用）

关联脉络

- PR #29649 [diffusion] keep image-model auxiliary components resident under auto memory policy: 同为 diffusion 模块改进，涉及服务器启动时资源管理，与预热逻辑有间接关联（预热后辅助组件常驻可进一步减少冷启动）。