PR #25988 完整报告

sgl-project/sglang

[diffusion] feat: enable warmup for sglang serve by default

合并时间：2026-05-22 08:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25988

执行摘要

- 一句话：默认启用 diffusion 服务预热，优化首次请求延迟
- 推荐动作：值得合并。该 PR 有效地解决了 diffusion 服务冷启动问题，设计上优先缓存默认负面提示，并提供了合理的 fallback。建议在后续迭代中增加对预热失败的回退机制和更详细的日志。

功能与动机

为了降低 diffusion 模型首次请求的响应时间，避免用户遇到明显的冷启动延迟。PR 标题明确说明启用默认预热。

实现拆解

1. 在 CLI 入口 serve.py 中，如果用户未显式指定 --warmup，则默认设置为 True。
2. 在调度器 scheduler.py 中，新增 _logged_server_ready_after_warmup 标记，当所有预热请求完成后输出“服务器就绪”日志。
3. 核心变更位于 text_encoding.py：
   - 新增 get_model_default_negative_prompt 函数，从模型信息中获取默认负面提示。
   - 引入 _should_cache_negative_text_embedding、_get_cached_negative_text_embedding、_maybe_cache_negative_text_embedding 等方法来精细化控制缓存的写入时机。
   - 关键变化：预热请求也会缓存其编码结果，使得后续真实请求（使用相同默认负面提示时）可以命中缓存，避免重复计算。
4. 测试方面：更新了 test_text_encoding_cache.py 以验证预热缓存行为；调整了集成测试配置，移除旧的 enable_warmup 字段。
5. 删除了 testcase_configs.py 和 gpu_cases.py 中与 enable_warmup 相关的代码，简化测试参数。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/text_encoding.py（模块 文本编码；类别 source；类型 core-logic；符号 get_model_default_negative_prompt, _should_cache_negative_text_embedding, _get_cached_negative_text_embedding, _maybe_cache_negative_text_embedding）：重构负面文本编码缓存，支持在预热时缓存

默认负面提示，新增模型默认提示获取函数

- python/sglang/multimodal_gen/test/unit/test_text_encoding_cache.py（模块 单元测试；类别 test；类型 test-coverage；符号 make_server_args, get_negative_embedding_twice, test_negative_text_cache_keeps_default_warmup）：新增测试用例验证预热缓存行为，提取公共辅助函数
- python/sglang/multimodal_gen/runtime/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：添加 _logged_server_ready_after_warmup 状态和就绪日志
- python/sglang/multimodal_gen/runtime/entrypoints/cli/serve.py（模块 CLI 入口；类别 source；类型 configuration）：默认启用 warmup 选项
- python/sglang/multimodal_gen/test/server/test_server_common.py（模块 集成测试；类别 test；类型 test-coverage）：移除手动添加 --warmup 的代码，因为默认已启用
- python/sglang/multimodal_gen/test/server/testcase_configs.py（模块 测试配置；类别 test；类型 configuration）：移除 enable_warmup 字段及关闭预热的用例设置
- python/sglang/multimodal_gen/test/server/gpu_cases.py（模块 GPU 测试；类别 test；类型 configuration）：移除 enable_warmup=False 参数

关键符号：get_model_default_negative_prompt, _should_cache_negative_text_embedding, _get_cached_negative_text_embedding, _maybe_cache_negative_text_embedding, _uses_model_default_negative_prompt, _get_model_default_negative_prompt, _normalize_negative_prompt_for_default_match, _append_positive_text_outputs, execute_serve_cmd, _log_warmup_result

关键源码片段

python/sglang/multimodal_gen/runtime/pipelines_core/stages/text_encoding.py

重构负面文本编码缓存，支持在预热时缓存默认负面提示，新增模型默认提示获取函数

```
# 从模型注册表获取默认 negative_prompt，缓存结果避免重复查询
@lru_cache(maxsize=1)
def get_model_default_negative_prompt(
    model_path: str, backend: Any, model_id: str | None
):
    from sglang.multimodal_gen.registry import get_model_info
    model_info = get_model_info(model_path, backend=backend, model_id=model_id)
    if model_info is None:
        return None
    return model_info.sampling_param_cls().negative_prompt


class TextEncodingStage(PipelineStage):
    # ... 初始化 _negative_text_cache_key / _negative_text_cache_value 等 ...

    def get_or_compute_negative_text_embedding(
        self, batch: Req, server_args: ServerArgs, all_indices: list[int]
```

```
):
    """
    获取缓存的负面文本嵌入，若未命中则计算并视情况缓存。
    预热请求也会缓存，只要使用的 negative_prompt 是模型默认值。
    """
    negative_cache_key = self._build_negative_text_cache_key(
        batch, server_args, all_indices
    )
    cached_negative = self._get_cached_negative_text_embedding(negative_cache_key)
    if cached_negative is not None:
        return cached_negative

    negative_text_outputs = self.encode_text(
        batch.negative_prompt,
        server_args,
        encoder_index=all_indices,
        return_attention_mask=True,
    )
    self._maybe_cache_negative_text_embedding(
        negative_cache_key, negative_text_outputs
    )
    return negative_text_outputs

def _should_cache_negative_text_embedding(
    self, batch: Req, server_args: ServerArgs
) -> bool:
    # 非预热请求始终缓存；预热请求仅当使用模型默认 negative_prompt 时缓存
    if not batch.is_warmup:
        return True
    return self._uses_model_default_negative_prompt(batch, server_args)

def _get_cached_negative_text_embedding(self, negative_cache_key):
    if negative_cache_key is None:
        return None
    if self._negative_text_cache_key == negative_cache_key:
        return self._negative_text_cache_value
    return None

def _maybe_cache_negative_text_embedding(
    self, negative_cache_key, negative_text_outputs
):
    # 外部已决定是否缓存，此处写入缓存
    self._negative_text_cache_key = negative_cache_key
    self._negative_text_cache_value = negative_text_outputs
```

评论区精华

无 review 评论记录，但提交历史显示多次细化迭代，表明对预热缓存逻辑进行了逐步完善。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 默认启用预热可能增加服务启动时间，对于某些不希望预热的场景（如快速测试）可能需要显式禁用。
  2. 缓存逻辑依赖于 *negative_cache_key* 的正确性，若 key 构建不完整可能导致缓存误命中。
  3. 新增的 *get_model_default_negative_prompt* 函数依赖模型注册表，若模型信息缺失则返回 None，此时预热缓存将跳过，行为与未启用预热一致，但需确保日志提示清晰。
  4. 测试覆盖了主要缓存场景，但未覆盖多 *encoder* 或动态 batch 下的并发缓存行为。 - 影响：用户侧：使用 sglang *serve* 启动 diffusion 服务时，首次推理耗时显著降低（预计减少文本编码时间）。系统侧：启动阶段增加约一次文本编码的开销，但后续请求受益。团队侧：简化了配置（不再需要手动指定 --warmup），但默认行为变更可能影响现有单元测试脚本的预期。 - 风险标记：默认行为变更，缓存命中依赖，启动时间增加，模型注册表依赖

关联脉络

- 暂无明显关联 PR