PR #29631 完整报告

sgl-project/sglang

[diffusion][cache-dit] add cache-dit support for Ideogram 4

合并时间：2026-07-03 19:58

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29631

执行摘要

- 一句话：为 Ideogram 4 添加 Cache-DiT 加速支持
- 推荐动作：值得精读。该 PR 展示了如何通过基类抽象方法为不同模型扩展通用优化技术，评审中关于去重、SCM 修复和抽象化的讨论提供了实际工程权衡。

功能与动机

为 Ideogram 4 在 SGLang 中添加 Cache-DiT 支持。Ideogram 4 使用条件和无条件两个 DiT 模块。通过已有的 SGLANG_CACHE_DIT_ENABLED 标志可对两个 transformer 启用可选的 Cache-DiT 加速。

实现拆解

1. 引入双 transformer 抽象及执行模式：在 denoising.py 中添加 DualTransformerExecutionMode 枚举和一系列受保护方法（_cache_dit_dual_model_name, _dual_transformer_execution_mode 等），基类通过 transformer_2 参数统一管理两个 transformer 的 Cache-DiT 配置、SCM 掩码和 torch.compile 顺序。
2. 修改 Ideogram4DenoisingStage 重用通用路径：在 ideogram.py 中，将无条件 transformer 通过 transformer_2 传入基类，删除原有的独立 Cache-DiT 设置逻辑；覆盖 _cache_dit_dual_model_name 返回 'ideogram4'，_dual_transformer_execution_mode 返回 PAIRED_PER_STEP，_cache_dit_secondary_uses_primary_config 返回 True。
3. 扩展缓存集成层支持 Ideogram 4：在 cache_dit_integration.py 的 enable_cache_on_dual_transformer 中增加 'ideogram4' 分支，使用 layers 属性、ForwardPattern.Pattern_3、has_separate_cfg=False。
4. 更新渐进分辨率阶段的上下文刷新：在 progressive_resolution/ideogram.py 中，将单 transformer 刷新替换为 refresh_context_on_dual_transformer，传递显式的 steps_computation_mask 以避免 SCM bins 丢失。

关键文件：

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 去噪框架；类别 source；类型 dependency-wiring；符号 DualTransformerExecutionMode, _cache_dit_dual_model_name, _cache_dit_secondary_uses_primary_config, _dual_transformer_execution_mode）：引入 DualTransformerExecutionMode 和抽象方

法，是双 transformer Cache-DiT 支持的核心枢纽。

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/ideogram.py（模块 Ideogram 模型；类别 source；类型 data-contract；符号 component_uses, _cache_dit_dual_model_name, _maybe_enable_cache_dit_and_torch_compile, _dual_transformer_execution_mode）：具体实现 Ideogram 4 的 Cache-DiT 适配，覆盖基类抽象方法并重用通用路径。
- python/sglang/multimodal_gen/runtime/cache/cache_dit_integration.py（模块 缓存集成；类别 source；类型 core-logic）：扩展 enable_cache_on_dual_transformer 支持 Ideogram 4 的 blocks 属性、ForwardPattern 和 has_separate_cfg。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/progressive_resolution/ideogram.py（模块 渐进分辨率；类别 source；类型 core-logic）：更新渐进分辨率阶段的 Cache-DiT 上下文刷新以支持双 transformer。

关键符号：DualTransformerExecutionMode, DenoisingStage._cache_dit_dual_model_name, DenoisingStage._dual_transformer_execution_mode, DenoisingStage._cache_dit_secondary_uses_primary_config, DenoisingStage._cache_dit_step_counts, DenoisingStage._parse_cache_dit_scm_bins, DenoisingStage._cache_dit_scm_masks, Ideogram4DenoisingStage._cache_dit_dual_model_name, Ideogram4DenoisingStage._dual_transformer_execution_mode, Ideogram4DenoisingStage._cache_dit_secondary_uses_primary_config, Ideogram4DenoisingStage._maybe_enable_cache_dit, Ideogram4ProgressiveDenoisingStage._refresh_cache_dit_context

关键源码片段

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

引入 DualTransformerExecutionMode 和抽象方法，是双 transformer Cache-DiT 支持的核心枢纽。

```
class DualTransformerExecutionMode(str, Enum):
    '''双 Transformer 执行模式

    BOUNDARY_EXPERTS: 每个时间步选择一个 transformer（如 Wan2.2）
    PAIRED_PER_STEP: 每步两个 transformer 均参与（如 Ideogram 4）
    '''
    BOUNDARY_EXPERTS = 'boundary_experts'
    PAIRED_PER_STEP = 'paired_per_step'


class DenoisingStage(PipelineStage, RolloutDenoisingMixin):
    # ...

    def _cache_dit_dual_model_name(self) -> str:
        # 派生类覆盖以返回模型名称，如 'ideogram4'
        return 'wan2.2'

    def _dual_transformer_execution_mode(self) -> DualTransformerExecutionMode | None:
```

```
        # 返回执行模式；若无第二 transformer 则返回 None
        if self.transformer_2 is None:
            return None
        return DualTransformerExecutionMode.BOUNDARY_EXPERTS

    def _cache_dit_secondary_uses_primary_config(self) -> bool:
        # 是否第二 transformer 复用主配置（Ideogram 需要 True）
        return False

    def _maybe_enable_cache_dit_and_torch_compile(self, num_inference_steps, batch):
        self._maybe_enable_cache_dit(num_inference_steps, batch)
        for transformer in filter(None, [self.transformer, self.transformer_2]):
            self._maybe_torch_compile(transformer)
```

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/ideogram.py

具体实现 Ideogram 4 的 Cache-DiT 适配，覆盖基类抽象方法并重用通用路径。

```
class Ideogram4DenoisingStage(DenoisingStage):
    def __init__(self, transformer, unconditional_transformer, pipeline=None) -> None:
        # 将无条件 transformer 传递给基类的 transformer_2 参数
        super().__init__(
            transformer=transformer,
            transformer_2=unconditional_transformer, # 基类统一管理两个 transformer
            scheduler=Ideogram4Scheduler(),
            pipeline=pipeline,
        )
        # 保留别名以兼容模型内部对 unconditional_transformer 的引用
        self.unconditional_transformer = self.transformer_2

    def _cache_dit_dual_model_name(self) -> str:
        return 'ideogram4'

    def _dual_transformer_execution_mode(self) -> DualTransformerExecutionMode | None:
        # Ideogram 4 条件与无条件 transformer 每一步都需参与
        return DualTransformerExecutionMode.PAIRED_PER_STEP

    def _cache_dit_secondary_uses_primary_config(self) -> bool:
        # 两个 transformer 共享相同的 Cache-DiT 配置
        return True

    def _maybe_enable_cache_dit(self, *args, **kwargs) -> None:
        # 调用基类方法后重新同步别名
        super()._maybe_enable_cache_dit(*args, **kwargs)
        self.unconditional_transformer = self.transformer_2
```

评论区精华

- 去重建议：AgainstEntropy 指出 Ideogram 的 Cache-DiT 代码大量重复 DenoisingStage，建议复用。作者采纳，通过 transformer_2 统一管理两个 transformer。
- SCM bins 丢失：gemini-code-assist 指出仅传递 scm_preset 会使自定义 bins 在后续请求中丢失。作者修复，扩展 refresh_context_on_dual_transformer 接口接受显式 steps_computation_mask。
- 模型名分派抽象：AgainstEntropy 建议创建更好的抽象替代 if/else 分支。作者同意并计划在后续 PR 引入适配器注册表。
- 移除不相关变更：mickqian 要求 PR 中移除渐进 API 字段。作者拆分到独立分支。
  - 复用基类 Cache-DiT 路径 (design): 作者采纳，将无条件 transformer 作为 transformer_2 传入基类，利用基类统一管理 Cache-DiT 设置、SCM 和编译。
  - SCM bins 在上下文刷新时丢失 (correctness): 作者新增 _cache_dit_scm_masks 解析并传递显式的 steps_computation_mask 给 refresh_context_on_dual_transformer。
  - 模型名称分派抽象 (design): 作者同意，计划在后续 PR 中引入双 transformer 适配器注册表，将分支替换为数据结构。
  - 移除渐进 API 不相关变更 (other): 作者将渐进 API 变更拆分到单独分支 /PR，此 PR 仅保留渐进阶段必要的 wiring。

风险与影响

- 风险：
  - 默认禁用，不引入回归。
  - 近似缓存特性不保证像素级精确。
  - 模型名分派采用 if-elif，每增加模型分支增多，但已计划重构。
  - 无新增单元测试，仅人工冒烟。
- 影响：
  - 用户：可通过环境变量 SGLANG_CACHE_DIT_ENABLED 启用加速，延迟降低约 2.1 倍；未启用无影响。
  - 系统：为 Diffusion 模型的 Cache-DiT 框架增加扩展性，后续模型可快速添加。
  - 团队：需关注后续适配器重构的合并。
  - 风险标记：缺少测试覆盖，模型名分派待重构，近似特性像素级不匹配

关联脉络

- 暂无明显关联 PR