PR #29649 完整报告

sgl-project/sglang

[diffusion] keep image-model auxiliary components resident under auto memory policy

合并时间：2026-06-30 01:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29649

执行摘要

- 一句话：让扩散模型在内存充足时保持辅助组件常驻 GPU
- 推荐动作：值得精读，尤其是自动内存策略的阈值分层设计和组件常驻 / 卸载的权衡考量。Review 中的 opt-out 缺陷可作为后续改进方向。

功能与动机

来自 #26247 的自动内存策略让图像生成模型走通用 layerwise-offload 路径，但图像模型没有针对 modality 的 ' 内存充足时保持常驻 ' 阈值：其 ModelDeploymentConfig 的 auto_disable_component_offload_min_available_memory_gb 为 None，导致高层次内存过滤器提前返回，从不保留任何组件。此外默认 auto_disable_component_offload_components 未包含 VAE。这使得数据中心 GPU 即使有大量空闲内存也每次请求都要从 CPU 重载组件，尤其是 text encoder 加载占 dominant 开销。

实现拆解

1. 重命名并调整数据契约：在 ModelDeploymentConfig 中用 keep_resident_min_available_gb 和 keep_resident_components 替代旧的 auto_disable_component_offload_*，默认组件从 (dit, text_encoder, image_encoder) 变为 (vae,)，并移除 DiT 以保留 FSDP 控制权。
2. 新增三级回退阈值解析：在 ServerArgsAutoTuner 中添加 _resolve_keep_resident_min_available_gb()，优先级为显式模型配置 > 任务类型默认（图像 45GB，其他 120GB） > 全局默认 120GB。两个入口 maybe_adjust_auto_component_residency_after_offload 和 _filter_high_memory_resident_components 均改为调用该方法。
3. 更新各 pipeline 配置：在 wan.py、ltx_2.py、sana_wm.py、hunyuan.py 中同步字段改名；FastHunyuan 移除硬编码 150GB 改用视频默认；Wan 系列保留 60GB 阈值但改用新字段。
4. 配套测试覆盖：新增 7 个测试用例覆盖默认图像保持常驻、低阈值卸载、多 GPU Qwen 与 CFG/headroom 组合等场景，并验证 FastHunyuanConfig 和 QwenImagePipelineConfig 的默认配置。

关键文件：

- python/sglang/multimodal_gen/test/unit/test_server_args.py（模块 配置测试；类别 test；类型 test-coverage；符号 test_default_auto_keeps_image_vae_resident_when_memory_allows, test_auto_image_offloads_aux_below_resident_threshold, test_auto_multi_gpu_qwen_replaces_text_encoder_offload_with_cfg, test_auto_multi_gpu_qwen_keeps_vae_resident_with_cfg）：测试覆盖新增的所有自动常驻场景，是验证核心逻辑正确性的关键配套。
- python/sglang/multimodal_gen/runtime/server_args_auto_tune.py（模块 自动调优；类别 source；类型 core-logic；符号 _resolve_keep_resident_min_available_gb）：核心自动调优逻辑所在，新增三级阈值解析函数，控制组件常驻策略的决策。
- python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py（模块 部署配置；类别 source；类型 data-contract）：数据契约变更，定义新的常驻字段和默认值，影响所有模型配置。
- python/sglang/multimodal_gen/configs/pipeline_configs/wan.py（模块 Wan 配置；类别 source；类型 configuration）：更新两个模型配置类（TurboWanT2V1_3B480PConfig、FastWan2_1_T2V_480P_Config）使用新字段名，保留原有阈值。
- python/sglang/multimodal_gen/configs/pipeline_configs/ltx_2.py（模块 LTX 配置；类别 source；类型 configuration）：LTX2PipelineConfig 使用新字段，并设置 keep_resident_components=("dit",)，保持行为不变。
- python/sglang/multimodal_gen/configs/pipeline_configs/sana_wm.py（模块 Sana 配置；类别 source；类型 configuration）：SanaWMPipelineConfig 使用新字段，keep_resident_min_available_gb=120，keep_resident_components=("dit",)，且关闭 auto_enable_cfg_parallel。
- python/sglang/multimodal_gen/configs/pipeline_configs/hunyuan.py（模块 Hunyuan 配置；类别 source；类型 configuration）：移除 FastHunyuan 上的 150GB 硬编码，让其回退到全局视频默认 120GB，保持代码简洁。

关键符号：_resolve_keep_resident_min_available_gb, maybe_adjust_auto_component_residency_after_offload, _filter_high_memory_resident_components, get_model_deployment_config (in multiple configs)

关键源码片段

python/sglang/multimodal_gen/test/unit/test_server_args.py

测试覆盖新增的所有自动常驻场景，是验证核心逻辑正确性的关键配套。

```
# 测试：默认自动模式下，可用内存充足时 (80 GB > 图像阈值 45GB) 保持 VAE 常驻
class TestServerArgsAutoTune(unittest.TestCase):
    ...
    def test_default_auto_keeps_image_vae_resident_when_memory_allows(self):
        args = self._from_dict_with_pipeline_config(
            QwenImagePipelineConfig(),
            kwargs={"model_path": "Qwen/Qwen-Image"},
        )
        self.assertEqual(args.performance_mode, "auto")
```

```
        self.assertFalse(args.use_fsdp_inference)
        # VAE 不应该被卸载：vae_cpu_offload 应保持 False
        self.assertFalse(args.vae_cpu_offload)
        # text_encoder 和 image_encoder 仍然 layerwise-offload
        self.assertEqual(
            args.layerwise_offload_components,
            ["text_encoder", "image_encoder"],
        )

    # 测试：可用内存低于阈值时 (40GB)，辅助组件全部卸载
    def test_auto_image_offloads_aux_below_resident_threshold(self):
        args = self._from_dict_with_pipeline_config(
            QwenImagePipelineConfig(),
            memory_gb=40,
            kwargs={"model_path": "Qwen/Qwen-Image"},
        )
        self.assertEqual(args.performance_mode, "auto")
        self.assertTrue(args.dit_cpu_offload)
        self.assertTrue(args.vae_cpu_offload) # VAE 也被卸载
```

python/sglang/multimodal_gen/runtime/server_args_auto_tune.py

核心自动调优逻辑所在，新增三级阈值解析函数，控制组件常驻策略的决策。

```
class ServerArgsAutoTuner:
    ...
    # 三级回退阈值：显式模型配置 > 任务类型默认 > 全局默认
    def _resolve_keep_resident_min_available_gb(
        self, deployment_config: ModelDeploymentConfig
    ) -> float | None:
        explicit = deployment_config.keep_resident_min_available_gb
        if explicit is not None:
            return explicit
        if self.server_args.pipeline_config.task_type.is_image_gen():
            return IMAGE_GEN_KEEP_RESIDENT_MIN_AVAILABLE_GB # 45 GB
        return DEFAULT_KEEP_RESIDENT_MIN_AVAILABLE_GB # 120 GB

    # 自动调整常驻：合并到 maybe_adjust_auto_component_residency_after_offload
    def maybe_adjust_auto_component_residency_after_offload(self) -> None:
        ...
        disable_threshold_gb = self._resolve_keep_resident_min_available_gb(
            deployment_config
        )
        if (
            min_available_gb is not None
            and disable_threshold_gb is not None
            and min_available_gb >= disable_threshold_gb
        ):
            components = deployment_config.keep_resident_components
            # 从 layerwise_offload_components 中移除这些组件（例外情况）
```

...

python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py

数据契约变更，定义新的常驻字段和默认值，影响所有模型配置。

```
@dataclass(frozen=True)
class ModelDeploymentConfig:
    auto_dit_layerwise_offload: bool = False
    auto_dit_layerwise_offload_high_memory_disable_gb: float | None = None
    # 新增：内存阈值（GB），高于此值时尝试保持组件常驻
    keep_resident_min_available_gb: float | None = None
    # 仅在内存充足时保留 VAE（体积小效率高），
    # 大 encoder 保持卸载，DiT 由 FSDP/dit-layerwise 策略管理
    keep_resident_components: tuple[OffloadComponentName, ...] = ("vae",)
    fsdp_auto_min_available_memory_gb: float | None = None
    fsdp_auto_requires_cfg: bool = True
    fsdp_auto_requires_default_parallelism: bool = True
    auto_enable_cfg_parallel: bool = True
    ...
```

评论区精华

Review 中 gemini-code-assist[bot] 指出 _resolve_auto_disable_component_offload_threshold_gb 的 docstring 声称可以返回 None 来 opt out，但因 fallback 永远返回非 None，导致模型无法显式退出自动常驻策略。该函数在最终版本已重命名为 _resolve_keep_resident_min_available_gb，但 docstring 依然存在类似矛盾，建议明确 opt-out 机制。此问题在 PR 合并时未被完全解决。

- 自动常驻策略无法显式 opt-out (correctness): PR 合并时未明确解决此问题。后续可能需要修复 docstring 或增加显式 opt-out 机制（如设置 -1）。

风险与影响

- 风险：
  - 默认值选择：图像阈值 45GB 基于典型数据中心 GPU，但若可用内存刚好在边界（44GB）仍会卸载，影响一致体验；视频默认 120GB 对部分高显存 GPU（如 H100 80GB）永远无法达到，意味着视频辅助组件始终卸载，这与原先某些模型（如 FastHunyuan 曾设 150GB 门槛）意图相反，需确认。
  - 重命名兼容性：直接使用 ModelDeploymentConfig 字段的外部代码可能失效，需确保内部已全部替换。
  - DiT 排除：默认常驻集合不再包含 DiT，确保 FSDP 层间卸载不被覆盖；但对于没有使用 FSDP 的模型，可能误使 DiT 仍然卸载（虽然 auto_dit_layerwise_offload 控制）。
  - 测试覆盖：多 GPU 视频场景的阈值行为未在单元测试覆盖，依赖 nightly CI。
- 影响：

- 用户：图像模型在内存充足时首 token 延迟降低（消除 text encoder 重载）；视频模型几乎无变化。显式设置 performance_mode 或手动控制组件的用户不受影响。
- 系统：GPU 常驻 VAE（~1GB）增加少量内存占用，但节省 CPU→GPU 传输带宽和 CPU 开销。整体吞吐受益。
- 团队：需要更新相关文档和最佳实践；配置字段重命名需在 release notes 中标注。
- 风险标记：重命名可能导致兼容性问题，视频模型阈值 120GB 可能过高，opt-out 机制缺失，测试未覆盖视频低内存场景

关联脉络

- PR #26247 auto memory policy for diffusion models: 本 PR 修复了该 PR 引入的自动内存策略中图像模型无条件 offload 的问题，并扩展了 keep-resident 机制。