PR #34401 完整报告

sgl-project/sglang

Fix model-driven DiT layerwise offload auto policy

合并时间：2026-08-12 09:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34401

执行摘要

- 一句话：重构模型驱动的 DiT 分层卸载自动策略
- 推荐动作：值得精读。核心价值在于将“模型配置声明”和“运行策略决策”解耦，用显式模式列表替代类名 / 模块名分发，是可复用的设计模式。重点关注 auto_tune.py 的 _should_auto_enable_dit_layerwise_offload 重写和 model_deployment_config.py 的数据契约变化，以及 MOVA 复用 keep_resident_components 的机制。

功能与动机

PR body 指出：ModelDeploymentConfig.auto_dit_layerwise_offload 被多个非 Wan 模型声明，但自动调优器只接受 Wan 模块，导致配置误导。MOVA 的高内存阈值也没有运行时消费者，使预期的内存策略失效。PR 目标是让策略显式化，不依赖 family 名称分发，同时让 MOVA 在 auto 高内存模式下同时禁用 layerwise 交换和粗粒度 DiT CPU offload。

实现拆解

本次变更的核心是移除基于类名 / 模块名的 Wan 大家庭分发，改为让每个 pipeline config 显式声明允许的模式。具体拆解如下：

1. 数据契约改造（model_deployment_config.py）：将 auto_dit_layerwise_offload: bool 和从未被消费的 auto_dit_layerwise_offload_high_memory_disable_gb 替换为 dit_layerwise_offload_modes: tuple[Literal["auto","memory"], ...] 与 auto_dit_offload_prefetch_size。这消除了误导性声明，使策略由模型自身声明而不是由自动调优器推断。
2. 自动调优逻辑重写（auto_tune.py）：_should_auto_enable_dit_layerwise_offload 不再调用 _is_wan_pipeline_config() 和 _is_wan2_2_a14b_pipeline_config()，改为直接检查 args.performance_mode in deployment_config.dit_layerwise_offload_modes，并调用平台钩子 enable_dit_layerwise_offload_by_default()。_set_default_wan_dit_offload_prefetch_size 泛化为 _set_default_dit_offload_prefetch_size，prefetch size 从 deployment config 读取。删除两个以类名和模块名判断的辅助函数。
3. 各模型配置迁移：wan.py 中普通 Wan 模型配置（WanT2V480PConfig、WanI2V480PConfig 等）改为 ("memory",) 模式，而 Wan2_2_T2V_A14B_Config 和 Wan2_2_I2V_A14B_Config 显式声明 ("auto", "memory") 且

auto_dit_offload_prefetch_size=2，保留之前已验证的默认 prefetch 行为。mova.py 将原本的高内存阈值改到共享的 keep_resident_min_available_gb=130 与 keep_resident_components=("dit","vae")。lingbot_world.py 新增加 get_model_deployment_config 声明 memory 模式。sana_wm.py、lingbot_video_moe.py 同步调整。

4. 平台能力钩子泛化：enable_dit_layerwise_offload_for_wan_by_default 在所有 platform（interface、cpu、npu、rocm）中更名为 enable_dit_layerwise_offload_by_default，语义从“专用于 Wan”扩展为通用平台能力声明。
5. 测试覆盖：test_server_args.py 中新增 MOVA 低于 130 GiB 时加入 dit、高于阈值时保持 dit 驻留的用例；memory 模式下 SanaWM 也加入 dit；test_pipeline_configs_declare_auto_tune_hints 断言模式元组。sana_wm/test_pipeline_config.py 与 h100.json 性能基线同步刷新。

关键文件：

- 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 _is_wan2_2_a14b_pipeline_config, _set_default_wan_dit_offload_prefetch_size, _set_default_dit_offload_prefetch_size, _is_wan_pipeline_config）：自动调优核心逻辑：移除 Wan 类名 / 模块名分发，改为基于部署配置的显式模式列表判断，泛化 prefetch size 设置。
- python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py（模块 部署配置；类别 source；类型 data-contract）：数据契约核心变更：bool 标志替换为模式元组 + prefetch 值，移除从未被消费的高内存阈值字段。
- python/sglang/multimodal_gen/configs/pipeline_configs/wan.py（模块 模型配置；类别 source；类型 core-logic；符号 get_model_deployment_config）：Wan 家族模型配置迁移：普通型号收敛为 memory-only，A14B 保留 auto 模式并显式声明 prefetch=2。
- python/sglang/multimodal_gen/configs/pipeline_configs/mova.py（模块 模型配置；类别 source；类型 core-logic）：MOVA 恢复 auto 分层卸载：将无效的高内存阈值替换为共享的 keep-resident 机制，使高内存时 d: 在低内存下自动启用 dit，高内存下禁用粗粒度 offload。
- python/sglang/multimodal_gen/test/unit/test_server_args.py（模块 参数解析；类别 test；类型 test-coverage；符号 test_auto_mova_layerwise_offload_adds_dit_below_memory_threshold, test_auto_mova_keeps_dit_resident_at_memory_threshold, test_memory_sana_wm_layerwise_offload_adds_dit）：测试配套：新增 MOVA 阈值上下限行为和 SanaWM memory 模式行为，mock 改为新的平台钩子名。
- python/sglang/multimodal_gen/runtime/platforms/npu.py（模块 平台层；类别 source；类型 core-logic；符号 enable_dit_layerwise_offload_by_default）：平台能力钩子重命名与语义扩展，NPU 仍默认关闭。
- python/sglang/multimodal_gen/runtime/platforms/interface.py（模块 平台层；类别 source；类型 core-logic；符号 enable_dit_layerwise_offload_by_default）：平台接口默认实现仍返回 True，语义泛化为所有模型。
- python/sglang/multimodal_gen/runtime/platforms/rocm.py（模块 平台层；类别 source；类型 core-logic；符号 enable_dit_layerwise_offload_by_default）：ROCm 保持默认关

闭，避免在 RoCm 上自动启用未验证的 DiT 分层卸载。

- python/sglang/multimodal_gen/configs/pipeline_configs/lingbot_world.py（模块 模型配置；类别 source；类型 core-logic；符号 get_model_deployment_config）：新增加部署配置声明，避免继承 A14B 的 auto 模式而隐式启用未经验证的卸载策略。
- python/sglang/multimodal_gen/configs/pipeline_configs/sana_wm.py（模块 模型配置；类别 source；类型 core-logic）：SanaWM 配置同步到 memory-only 模式。

关键符号：_should_auto_enable_dit_layerwise_offload, _set_default_dit_offload_prefetch_size, get_model_deployment_config, enable_dit_layerwise_offload_by_default

关键源码片段

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

自动调优核心逻辑：移除 Wan 类名 / 模块名分发，改为基于部署配置的显式模式列表判断，泛化 prefetch size 设置。

```
# python/sglang/multimodal_gen/runtime/server_args/auto_tune.py
# 变更后：策略完全由 deployment config 声明驱动，不再按类名 / 模块名分发

def _should_auto_enable_dit_layerwise_offload(self) -> bool:
    args = self.server_args
    deployment_config = self._deployment_config()

    # 判断当前 performance_mode（如 "auto" / "memory"）是否在该模型声明
    # 的允许模式列表中；如 MOVA 可声明 ("auto", "memory")，普通 Wan 仅 ("memory",)
    if args.performance_mode not in deployment_config.dit_layerwise_offload_modes:
        return False

    # 与旧实现一致的守卫：DMD 步数、平台默认开关、缓存 DiT、FSDP、
    # 显式 dit_cpu_offload 任一命中都禁止自动启用
    if (
        args.pipeline_config.dmd_denoising_steps is not None
        or not current_platform.enable_dit_layerwise_offload_by_default()
        or envs.SGLANG_CACHE_DIT_ENABLED
        or args.use_fsdp_inference
        or args.is_arg_explicitly_set("dit_cpu_offload")
    ):
        return False

    return True


def _set_default_dit_offload_prefetch_size(self) -> None:
    args = self.server_args
    # prefetch 值由模型配置声明，而非在代码里硬编码 Wan2.2 A14B 专属常量
    prefetch_size = self._deployment_config().auto_dit_offload_prefetch_size
    if (
```

```
            args.performance_mode == "auto"
            and prefetch_size is not None
            and not args.is_arg_explicitly_set("dit_offload_prefetch_size")
        ):
            args.dit_offload_prefetch_size = prefetch_size
```

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

数据契约核心变更：bool 标志替换为模式元组 + prefetch 值，移除从未被消费的高内存阈值字段。

```
# python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py
@dataclass(frozen=True)
class ModelDeploymentConfig:
    # 允许自动启用 DiT 分层 offload 的 performance_mode 列表；
    # 空元组 () 表示该模型从不自动启用
    dit_layerwise_offload_modes: tuple[Literal["auto", "memory"], ...] = ()

    # auto 模式下 DiT offload 的默认 prefetch_size，由模型配置声明；
    # None 表示不自动设置
    auto_dit_offload_prefetch_size: float | None = None

    keep_resident_min_available_gb: float | None = None
    # 仅 vae 常驻 — 它很小，常驻几乎不影响内存；大编码器保持 offload，
    # dit 的放置由 FSDP / dit-layerwise 路径决定
    keep_resident_components: tuple[str, ...] = ("vae",)
```

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

Wan 家族模型配置迁移：普通型号收敛为 memory-only，A14B 保留 auto 模式并显式声明 prefetch=2。

```
# python/sglang/multimodal_gen/configs/pipeline_configs/wan.py
@dataclass
class WanT2V480PConfig(WanT2V480PConfig):
    # 普通 Wan 模型：仅 memory 模式自动启用 DiT 分层卸载，
    # auto 模式由部署配置显式声明是否参与
    def get_model_deployment_config(self) -> ModelDeploymentConfig:
        return ModelDeploymentConfig(
            dit_layerwise_offload_modes=("memory",),
        )


@dataclass
class Wan2_2_T2V_A14B_Config(WanT2V480PConfig):
    # A14B 保留经过 sweep 验证的 auto 策略：
    # auto 模式下自动启用层间 offload，并默认 prefetch size=2
    def get_model_deployment_config(self) -> ModelDeploymentConfig:
        return ModelDeploymentConfig(
            dit_layerwise_offload_modes=("auto", "memory"),
```

```
        auto_dit_offload_prefetch_size=2,
    )
```

评论区精华

该 PR 没有 review 评论和 review 线程，讨论主要沉淀在 PR body 中：核心争议是旧实现把 auto_dit_layerwise_offload 声明在多个模型上但自动调优器只接受 Wan 模块，造成配置与行为不一致；结论是采用显式模式列表消除 family 名称分发。另一个决策点是 MOVA 的高内存阈值没有运行时消费者，改为复用共享的 component-residency 机制，确保高内存 auto 模式同时禁用 layerwise 交换和粗粒度 DiT CPU offload。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 行为变更范围：Wan 家族配置的 dit_layerwise_offload_modes 从统一 auto_dit_layerwise_offload=True 改为仅 memory 模式（除 A14B 外），auto 模式下 Wan 非 A14B 可能不再自动启用 DiT 分层 offload，对依赖旧行为的用户是隐性变更。
  2. 平台语义扩大：enable_dit_layerwise_offload_by_default 在 CUDA 接口默认 True，但此前仅 Wan 使用；现在任何声明了 auto 模式的模型都会受该平台开关影响。ROCm/NPU 仍默认 False，影响有限。
  3. 测试可靠性：MOVA 测试依赖 memory_gb=140 模拟可用内存，若测试环境内存报告不稳定可能 flaky；prefetch size 改为从部署配置读取后，若模型未声明则默认 None，行为需验证。
  4. 基线刷新：h100.json 的 Wan TI2V stage timing 基线刷新，可能掩盖真实性能差异。 - 影响：影响范围集中在 sglang.multimodal_gen 模块：所有声明 layerwise offload 的模型配置（Wan、MOVA、SanaWM、LingBotWorld）都会改变自动策略行为。对用户来说，MOVA 在 auto 模式下低内存场景会重新启用 DiT 分层 offload，高内存场景获得完整驻留；对 Wan 普通模型，auto 模式不再隐式加入 dit，memory 模式行为不变。对团队来说，消除了误导性配置，为后续模型扩展提供更清晰的数据契约。 - 风险标记：行为变更范围广，缺 review 评审，平台钩子语义扩大，测试依赖内存阈值

关联脉络

- PR #34249 [diffusion] move DiT execution capabilities to runtime models: 同一 diffusion 模块的架构演进，将 DiT 执行能力迁移到 runtime 模型，与本 PR 的模型驱动策略方向一致。
- PR #34315 [diffusion] LTX-2: mount the bit-exact fused modulate at the 8 bare adaLN sites: 同为 diffusion 性能优化系列，涉及模型配置与运行时的解耦。