# PR #35862 完整报告

sgl-project/sglang

[diffusion] keep a CPU-started VAE's weights on the checkpoint mapping

合并时间：2026-08-22 09:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35862

## 执行摘要

- 一句话：CPU 启动的 VAE 权重保留文件映射，显著释放主机内存
- 推荐动作：值得精读。核心看点：一是 file-backed 映射与匿名内存的取舍——匿名页无法被 page cache 回收，而映射首次使用要付缺页，门控以“整个部署权重 vs 可用主机内存”为决策基准而非单个组件；二是 _match_checkpoint_dtypes 把“必须拷贝的张量”精确限定为 dtype 不匹配者；三是 CI 性能基线如何推动设计修正。若要借鉴此模式，可关注瞬时内存测量与统计口径的边界情况。

## 功能与动机

PR body 指出：VAE loader 除 MPS 外总是把每个权重拷贝进预分配参数（assign=bool(cpu_offload_flag and current_platform.is_mps())），即使 safetensors 映射已持有这些字节。匿名页（anonymous pages）无法像文件映射页那样被 page cache 丢弃后重新 fetch，而小内存主机缺的恰恰是匿名内存——MiniMax-H3 在 32 GiB 主机预算下 video VAE 的拷贝占 9.7 GB；改为赋值映射张量后，offload 配置时刻可用主机内存从 8.8 GiB 提升到 18.5 GiB，这些预算正是 DiT 的固定预取（pinned prefetch）能实际使用的部分。

## 实现拆解

1. 新增部署级内存决策工具（python/sglang/multimodal_gen/runtime/loader/utils.py）：新增 checkpoint_bytes(model_path)，递归统计路径下所有 .safetensors 的磁盘字节数，只读文件大小、不加载张量，可在任何文件被读入前完成估算；新增 keep_checkpoint_mapped(*, weight_bytes, component)，延迟导入 host_memory_budget 模块的 host_copies_would_not_fit 与 host_memory_available_bytes，以整个部署的权重字节数对比可用主机内存，判断是否保留文件映射。延迟导入用于避免与 memory_managers 形成循环依赖。
2. VAE 加载路径接入门控（python/sglang/multimodal_gen/runtime/loader/component_loaders/vae_loader.py）：VAELoader.load_customized 中原先的 assign=bool(cpu_offload_flag and current_platform.is_mps()) 改为计算 keep_mapping = component_starts_on_cpu and (current_platform.is_mps() or keep_checkpoint_mapped(...))，其中权重字节数使用 checkpoint_bytes(server_args.model_path)——统计整个部署路径而非单个 VAE 组件，与“按部署整体决策”的口径一致。MPS 行为保持不变（统一内存，始终 assign）。

3. dtype 匹配保护（同上文件）：新增 _match_checkpoint_dtypes(loaded, target_state)，在 assign 前把 dtype 与模块参数不一致的 checkpoint 张量转换为参数 dtype。因为 assign 是替换参数而非写入参数，dtype 不一致会静默改变模块 dtype；转换即拷贝，恰好做到“只有无法留在文件映射上的张量才占用匿名内存”，多余键不处理、留给 strict 检查。
4. 测试配套（python/sglang/multimodal_gen/test/unit/test_vae_loader.py）：新增 TestKeepCheckpointMapped 与 TestMatchCheckpointDtypes 两组测试，通过 mock host_memory_available_bytes 覆盖两类门控决策（3 GiB 权重对 64 GiB 主机走拷贝、117 GiB 权重对 19 GiB 主机保留映射）与三类 dtype 场景（匹配不动、不匹配转换、模块不需要的键不动）。
5. 演进修正：第一版提交为无条件 assign（只要 CPU 启动且非 MPS），CI 在内存充裕的 runner 上抓到 joyai image VAE encode 从 68 ms 退化到 279 ms（映射首次使用缺页 vs 匿名拷贝页常驻），第二版提交加入上述门控并新增门控测试，形成最终形态。

关键文件：

- python/sglang/multimodal_gen/runtime/loader/utils.py（模块 加载工具；类别 source；类型 core-logic；符号 checkpoint_bytes, keep_checkpoint_mapped）：新增 checkpoint_bytes 与 keep_checkpoint_mapped 两个部署级内存决策工具，是门控逻辑的核心：递归统计 safetensors 磁盘字节数，并对比部署整体权重与可用主机内存决定是否保留文件映射。
- python/sglang/multimodal_gen/runtime/loader/component_loaders/vae_loader.py（模块 VAE 加载；类别 source；类型 core-logic；符号 _match_checkpoint_dtypes, load_customized）：核心逻辑变更点：load_customized 中把无条件按 MPS 判断的 assign 改为带部署级门控的 keep_mapping，并新增 _match_checkpoint_dtypes 保证 assign 前后 dtype 一致。
- python/sglang/multimodal_gen/test/unit/test_vae_loader.py（模块 VAE 测试；类别 test；类型 test-coverage；符号 TestKeepCheckpointMapped, test_a_small_deployment_on_a_roomy_host_copies, test_a_deployment_larger_than_the_host_stays_mapped, TestMatchCheckpointDtypes）：测试配套：新增 TestKeepCheckpointMapped 与 TestMatchCheckpointDtypes 两组单元测试，用 mock 隔离宿主机内存，覆盖门控决策与 dtype 匹配助手的核心行为。

关键符号：checkpoint_bytes, keep_checkpoint_mapped, _match_checkpoint_dtypes, load_customized

## 关键源码片段

### python/sglang/multimodal_gen/runtime/loader/utils.py

新增 checkpoint_bytes 与 keep_checkpoint_mapped 两个部署级内存决策工具，是门控逻辑的核心：递归统计 safetensors 磁盘字节数，并对比部署整体权重与可用主机内存决定是否保留文件映射。

```
def checkpoint_bytes(model_path: str) -> int:
    """On-disk size of every safetensors under a path, readable before any is."""
    # 递归收集路径下所有 .safetensors 文件，累加磁盘字节数；
    # 统计的是“整个部署路径”而非单个组件 —— 门控决策需要对比
```

```
    # 部署整体权重与可用主机内存，而不是只看当前 VAE 组件。
    total = 0
    for path in glob.glob(
        os.path.join(str(model_path), "**", "*.safetensors"), recursive=True
    ):
        try:
            total += os.path.getsize(path)
        except OSError:
            # 单个文件读取失败时跳过，不因个别坏文件阻断加载
            continue
    return total


def keep_checkpoint_mapped(*, weight_bytes: int, component: str) -> bool:
    """Whether a component's weights should stay on their file mapping."""
    # 延迟导入 host_memory_budget，避免与 memory_managers 模块循环依赖
    from sglang.multimodal_gen.runtime.managers.memory_managers.host_memory_budget
    import (
        host_copies_would_not_fit,
        host_memory_available_bytes,
    )

    # 主机放得下整个部署的拷贝时返回 False，走旧的匿名拷贝路径：
    # 拷贝页常驻内存，而文件映射的首次使用要付一次缺页开销。
    # 只有“拷贝放不进主机内存”时才保留映射 —— 文件页可以被
    # page cache 丢弃后重新 fetch，匿名页一旦落下就无法回收。
    if not host_copies_would_not_fit(weight_bytes):
        return False
    logger.info(
        "%s stays on its checkpoint mapping: the deployment is %.2f GiB of "
        "weights against %.2f GiB of host memory, so copies are host memory "
        "the streamed components need more.",
        component,
        weight_bytes / 1024**3,
        host_memory_available_bytes() / 1024**3,
    )
    return True
```

## python/sglang/multimodal_gen/runtime/loader/component_loaders/vae_loader.py

核心逻辑变更点：load_customized 中把无条件按 MPS 判断的 assign 改为带部署级门控的 keep_mapping，并新增 _match_checkpoint_dtypes 保证 assign 前后 dtype 一致。

```
def _match_checkpoint_dtypes(loaded: dict, target_state: dict) -> dict:
    """Convert checkpoint tensors whose dtype differs from their parameter's."""
    # assign 是“替换参数”而不是“写入参数”，dtype 不一致会静默改变
    # 模块的 dtype；因此只在 dtype 不匹配时做 .to() 转换。
    # 转换即产生新拷贝 —— 恰好让“无法留在文件映射上的张量”
    # 才占用匿名内存，其余张量保持 file-backed。
```

```
    for name, tensor in list(loaded.items()):
        param = target_state.get(name)
        if param is not None and param.dtype != tensor.dtype:
            loaded[name] = tensor.to(dtype=param.dtype)
    return loaded


# 以下为 VAELoader.load_customized 中的决策块。组件从 CPU 启动且主机
# 放不下整个部署的拷贝时，用 assign=True 让权重保持文件映射 ——
# MiniMax-H3 的 video VAE 占 32 GiB 预算中的 9.70 GiB；主机内存
# 充裕时仍走拷贝，因为拷贝页常驻，而映射的首次使用要付缺页
# （CI 抓到的 joyai image VAE encode 从 68 ms 涨到 279 ms）。
keep_mapping = component_starts_on_cpu and (
    current_platform.is_mps()
    or keep_checkpoint_mapped(
        weight_bytes=checkpoint_bytes(server_args.model_path),
        component=f"{component_name or 'vae'} (VAE)",
    )
)
if keep_mapping:
    _match_checkpoint_dtypes(loaded, vae.state_dict())
vae.load_state_dict(
    loaded,
    strict=strict_load,
    assign=keep_mapping,
)
```

## 评论区精华

PR 无 review 评论（review_comments_count: 0），两条 issue 评论均由作者 mickqian 说明 CI 状态。其一，AMD ROCm 720 的失败是 pre-existing：minimax_h3_ref2va_video_audio_2gpu_h100 死于 #33880 引入的平台门控 RuntimeError("MiniMax H3 full-loop denoise requires CUDA or MPS")，任何 partition 分配含 H3 server case 的 PR 都会在该 runner 上同样失败，调度修复将单独提交。其二，multimodal-gen-test-1-gpu (2) 的失败是真实回归并已修复：无条件 assign 让内存充裕的主机付出首次使用缺页代价（joyai ImageVAEEncodingStage 68 ms → 279 ms），门控改为对比整个部署 checkpoint 字节数与可用主机内存后恢复；同时 wan2_1_lora 的 miss 是 177.1 ms 对 175.8 ms 限制（0.7% 超标），属于重试级噪声。

- AMD ROCm 720 CI 失败为 pre-existing，与本 PR 无关 (other): 确认为既有问题，调度修复将单独提交，不影响本 PR 合入。
- 无条件 assign 导致内存充裕主机性能回退（CI 真实失败） (performance): 已修复：新增 keep_checkpoint_mapped 门控与两个门控测试，形成最终合入形态。

## 风险与影响

- 风险：

1. 缺页性能回退：保留映射的组件首次使用时每个权重页都要付一次 page fault；门控依赖 host_memory_available_bytes 的瞬时测量，若测量值与实际运行时的空闲内存有偏差，可能误判并让内存并不紧张的主机也走映射路径。
2. assign 语义：assign 替换参数而非写入，dtype 不一致会静默改变模块 dtype，已由 _match_checkpoint_dtypes 覆盖；但 assign 之后 _convert_conv3d_weights_to_channels_last_3d 与 current_platform.optimize_vae 的原地改写行为没有专门测试。
3. 统计口径：checkpoint_bytes(server_args.model_path) 统计整个部署路径，若路径下混有非本组件或临时权重文件，会高估部署字节数、更倾向于保留映射。
4. 兼容性：非 CPU 启动、非 MPS、内存充裕场景行为不变；MPS 行为不变；影响范围限于 diffusion 的 VAE 加载路径。 - 影响：对用户：小内存主机（如 32 GiB）上部署 MiniMax-H3 等大型 diffusion 模型的能力显著提升，video VAE 的 9.7 GB 匿名拷贝被消除，offload 配置时可用主机内存从 8.8 GiB 提升到 18.5 GiB；内存充裕主机与 MPS 平台行为不变。对系统：门控决策进入 VAE 加载核心路径，加载日志从 host pageable: 9.7 变为 host mmap: 9.7，后续任何 diffusion 组件加载策略调整都需考虑该门控。对团队：CI 性能基线（joyai 68 ms → 279 ms）推动设计从不加区分的 assign 收敛到带门控的最终形态，并沉淀了两组单元测试保护该决策。 - 风险标记：缺页延迟风险，assign 改变模块 dtype，瞬时内存测量误差，部署字节统计口径

## 关联脉络

- PR #36085 [Diffusion] Support VAE weight-file overrides: 同改 vae_loader.py 与 test_vae_loader.py，同属 VAE 组件权重加载能力演进线。
- PR #36078 [Diffusion] Add composable component weight path CLI: 组件权重路径选择（select_vae_weight_files）与本 PR 的映射门控同处 VAE 加载链路，权重文件覆盖能力叠加。
- PR #36051 [diffusion] CI: guard the anonymous-host budget alongside peak VRAM: 同为守卫匿名主机内存预算，与本 PR 消除 VAE 匿名拷贝的目标一致，同属小内存主机部署能力收紧方向。