PR #52374 完整报告

vllm-project/vllm

[MRV2] Support attention-free models

合并时间：2026-08-15 07:54

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52374

执行摘要

- 一句话：MRV2 支持 attention-free 模型，补 CI 兼容小改动
- 推荐动作：值得精读。虽然只有 3 个文件、9 行净变更，但它示范了两件事：一是如何把一个大型迁移 PR 拆成语义单一、可独立合入的小步（关注 #46646 的拆分策略）；二是 MambaHybridModelState 复用于 attention-free 模型的 ModelState 路由决策，以及测试中 envs.disable_envs_cache() 处理环境变量缓存的细节。若你正在跟踪 MRV2 默认切换或 mamba 状态管理相关代码，建议阅读 init_model_state 的完整分支顺序与 shutdown 的资源释放序列。

功能与动机

PR body 明确说明："These use MambaHybridModelState And other minor changes for CI compatibility when switching to MRV2 by default. These are split out from https://github.com/vllm-project/vllm/pull/46646." 也就是说，随着 MRV2 即将默认启用，attention-free 模型需要在新的 ModelState 路由下正确获得 mamba 状态管理（此前没有 attention 层的模型会落入 DefaultModelState，无法管理 SSM 状态）；另外由于 Qwen3-Next 等 MoE/hybrid 模型在 MRV2 下默认走 V2 runner，依赖 V1 runner 内部 patch 的 Mamba 前缀缓存测试必须显式固定 V1。

实现拆解

整个变更可按三步拆解：

1. 扩展 ModelState 路由条件：vllm/v1/worker/gpu/model_states/__init__.py 的 init_model_state 中，将原本仅判断 vllm_config.model_config.is_hybrid 的分支扩为 is_hybrid or is_attention_free，让纯 Mamba/SSM 等 attention-free 模型与混合模型一样使用 MambaHybridModelState。该状态类负责 mamba 状态缓存（mamba_cache）的申请、对齐与管理，因此是否包含 attention 层并不影响复用。路由顺序保持不变：模型自定义 get_model_state_cls 优先，其次是 CrossAttention、EncoderOnlyAttention，最后才落到 hybrid/attention-free 与默认分支。
2. ModelRunner 生命周期与日志微调：vllm/v1/worker/gpu/model_runner.py 的 shutdown 中新增 self.cudagraph_manager = None，在清空 kv_caches 与 attn_groups 之前显式解除对 CUDA Graph 管理器的引用，使 graph 占用的显存能随随后的 gc.collect() 和 empty_cache() 一并回收；同时把模型加载日志从 "Model loading took %s GiB and %.6f seconds" 调整为 "Model loading took %s GiB memory and %.6f seconds"，让文案更清

晰。

3. 测试固定 V1 runner：tests/v1/e2e/general/test_mamba_prefix_cache.py 的 _run_mamba_prefix_cache_mrv1 新增 monkeypatch.setenv("VLLM_USE_V2_MODEL_RUNNER", "0") 和 envs.disable_envs_cache()。由于该测试会 patch V1 model runner 内部逻辑，而 MRV2 默认开启后 Qwen3-Next 会默认走 V2 runner，必须显式钉住 V1；disable_envs_cache() 用于使 vllm/envs.py 的进程级环境变量缓存失效，否则 setenv 不生效。测试内部仍按原逻辑执行 mamba 前缀缓存的 step 级状态对齐验证，未改变既有断言。

关键文件：

- vllm/v1/worker/gpu/model_states/__init__.py（模块 模型状态；类别 source；类型 data-contract；符号 init_model_state）：核心变更文件：init_model_state 路由条件新增 is_attention_free，直接决定 attention-free 模型在 MRV2 下使用哪个 ModelState 类。
- vllm/v1/worker/gpu/model_runner.py（模块 模型运行器；类别 source；类型 core-logic；符号 ModelRunner.shutdown, ModelRunner.load_model）：ModelRunner 生命周期补强：shutdown 时显式解除 cudagraph_manager 引用，保证 CUDA graph 显存可回收；同时调整模型加载日志文案。
- tests/v1/e2e/general/test_mamba_prefix_cache.py（模块 前缀缓存；类别 test；类型 test-coverage；符号 _run_mamba_prefix_cache_mrv1）：CI 兼容关键改动：该测试会 patch V1 model runner 内部逻辑，而 Qwen3-Next 在 MRV2 默认切换后会走 V2 runner，因此必须显式钉住 V1。

关键符号：init_model_state, ModelRunner.shutdown, ModelRunner.load_model, _run_mamba_prefix_cache_mrv1

关键源码片段

vllm/v1/worker/gpu/model_states/__init__.py

核心变更文件：init_model_state 路由条件新增 is_attention_free，直接决定 attention-free 模型在 MRV2 下使用哪个 ModelState 类。

```
from vllm.config import VllmConfig
from vllm.model_executor.layers.attention import CrossAttention, EncoderOnlyAttention
from vllm.v1.worker.gpu.mm.encoder_cache import EncoderCache


def init_model_state(
    vllm_config: VllmConfig,
    model: nn.Module,
    encoder_cache: EncoderCache | None,
    device: torch.device,
):
    # 优先让模型自己声明 ModelState，这是最灵活的扩展点。
    if hasattr(model, "get_model_state_cls"):
        cls = model.get_model_state_cls()
        return cls(vllm_config, model, encoder_cache, device)

    # 交叉注意力编码器 - 解码器模型（Whisper、CohereASR、NemotronParse 等）
```

```
    # 需要额外的编码器缓存管理。
    if any(isinstance(m, CrossAttention) for m in model.modules()):
        from vllm.v1.worker.gpu.model_states.encoder_decoder import (
            EncoderDecoderModelState,
        )
        return EncoderDecoderModelState(vllm_config, model, encoder_cache, device)

    # 仅有编码器的模型（BERT/RoBERTa）：非因果自注意力，无 KV cache。
    if any(isinstance(m, EncoderOnlyAttention) for m in model.modules()):
        from vllm.v1.worker.gpu.model_states.encoder_only import EncoderOnlyModelState
        return EncoderOnlyModelState(vllm_config, model, encoder_cache, device)

    # 关键变更：attention-free（纯 Mamba/SSM 类）模型同样复用
    # MambaHybridModelState。该状态类负责 mamba 状态缓存的管理，
    # 是否包含 attention 层并不影响这一点。
    if vllm_config.model_config.is_hybrid or vllm_config.model_config.is_attention_free:
        from vllm.v1.worker.gpu.model_states.mamba_hybrid import MambaHybridModelState
        return MambaHybridModelState(vllm_config, model, encoder_cache, device)

    # 其余常规 Transformer 模型走默认状态管理。
    from vllm.v1.worker.gpu.model_states.default import DefaultModelState
    return DefaultModelState(vllm_config, model, encoder_cache, device)
```

vllm/v1/worker/gpu/model_runner.py

ModelRunner 生命周期补强：shutdown 时显式解除 cudagraph_manager 引用，保证 CUDA graph 显存可回收；同时调整模型加载日志文案。

```
  def shutdown(self) -> None:
    """Release GPU tensors (model weights, KV caches, workspace) so that
    memory is reclaimable when running in the same process."""
    torch.accelerator.synchronize()
    # MRV2 下显式解除对 CUDA Graph 管理器的引用，确保其占用的
    # graph 内存能随之后的 gc.collect() 与 empty_cache() 一并回收。
    self.cudagraph_manager = None
    if hasattr(self, "kv_caches"):
        self.kv_caches.clear()
    if hasattr(self, "attn_groups"):
        self.attn_groups.clear()
    if hasattr(self, "kv_cache_config"):
        del self.kv_cache_config
    if hasattr(self, "model_state") and self.model_state.supports_mm_inputs:
        self.model_state.encoder_runner.clear()
    free_before_shutdown(self.vllm_config)
    if hasattr(self, "model_state"):
        del self.model_state
    if getattr(self, "speculator", None) is not None:
        self.speculator = None
    if hasattr(self, "model"):
        del self.model
```

```
    gc.collect()
    torch.accelerator.empty_cache()
    logger.debug("Cleaned up model weights, KV caches, and workspace")
```

tests/v1/e2e/general/test_mamba_prefix_cache.py

CI 兼容关键改动：该测试会 patch V1 model runner 内部逻辑，而 Qwen3-Next 在 MRV2 默认切换后会走 V2 runner，因此必须显式钉住 V1。

```
def _run_mamba_prefix_cache_mrv1(
    monkeypatch: pytest.MonkeyPatch, async_scheduling: bool
):
    # 本测试通过 patch 直接侵入 V1 model runner 内部执行逻辑，
    # 而 Qwen3-Next 这类 MoE/hybrid 模型在 MRV2 默认开启后会
    # 默认走 V2 runner，因此必须显式钉住 V1：
    monkeypatch.setenv("VLLM_USE_V2_MODEL_RUNNER", "0")
    # vllm 的环境变量存在进程级缓存，setenv 后必须使缓存失效才会生效。
    envs.disable_envs_cache()
    global async_scheduling_mode
    async_scheduling_mode = async_scheduling
    run_ref_mamba_state_in_subprocess()
    apply_patch(monkeypatch)
    prompt_dataset = datasets.load_dataset("heheda/a_long_article")
    full_prompt = prompt_dataset["train"][0]["text"]
    tests = get_mamba_prefix_cache_step_configs(async_scheduling)

    engine = LLM(
        model=MODEL,
        load_format="dummy",
        enable_prefix_caching=True,
        block_size=BLOCK_SIZE,
        mamba_cache_mode="align",
        speculative_config={
            "method": "qwen3_next_mtp",
            "num_speculative_tokens": num_speculative_tokens,
        },
        max_num_batched_tokens=3072,
        hf_overrides={"num_hidden_layers": NUM_HIDDEN_LAYERS},
        async_scheduling=async_scheduling,
        seed=42,
    )
```

评论区精华

本 PR 的 review 讨论非常克制，没有出现技术争论：

- claude[bot] 指出该 PR 来自 fork，自动化 review 默认关闭，需维护者评论 @claude review 才能触发一次性审查。最终未触发额外审查。
- WoosukKwon 直接 APPROVED 且未留评论，说明维护者认可这一从大 PR 拆出的最小子集。

- PR body 本身表述了拆分策略：把 MRV2 默认切换所需的 attention-free 支持与 CI 兼容改动从 #46646 中拆出，保持每次变更聚焦、可独立审查。
- 从大 PR #46646 拆出小步子集的审查策略 (design): WoosukKwon 直接 approve，改动被合入，说明拆分粒度获得维护者认可。
- fork PR 的自动化审查限制 (other): 未触发额外自动化审查，由维护者 WoosukKwon 手工 approve 完成。

风险与影响

- 风险：风险整体偏低，但有几个点值得关注：

1. ModelState 路由语义扩展（model_states/__init__.py）：is_attention_free 模型现在强制走 MambaHybridModelState，该状态类依赖 mamba 状态缓存相关假设。当前纯 SSM/attention-free 模型与 mamba 状态管理语义匹配，但若未来出现既无 attention 又无 SSM 状态的模型，会被误路由。另外检查顺序上 attention-free 分支位于 EncoderOnlyAttention 之后，若某模型同时带编码器注意力与 attention-free 标记，会先命中 EncoderOnlyModelState；当前无此类模型。
2. shutdown 解引用（model_runner.py）：self.cudagraph_manager = None 是显式置空而非删除。若 MRV2 后续代码在 shutdown 后仍访问该属性，可能触发 AttributeError。当前生命周期内无可感知影响，但依赖模型运行器 shutdown 后不再使用该属性的隐式约定。
3. 环境变量全局副作用（测试文件）：VLLM_USE_V2_MODEL_RUNNER=0 是进程级全局设置，虽由 pytest monkeypatch 自动恢复，但 disable_envs_cache() 会让同进程其他测试的环境变量读取暂时失效；若该测试与其他用例共享进程，极端情况下可能影响并发读取。
4. MRV2 默认切换的联动风险：本 PR 只是拼图之一，单看无法验证 MRV2 全局行为；日志文案与 shutdown 改动属配套清理，无独立回归面。 - 影响：影响范围集中在 MRV2 模型状态分发与相关 e2e 测试：

- 用户侧：attention-free 模型（纯 Mamba/SSM 类）在启用 MRV2（或 MRV2 默认切换后）可正常运行并获得正确的 mamba 状态管理，不再错误落入默认状态类。
- 系统侧：模型 shutdown 时 CUDA Graph 管理器引用被显式解除，显存回收更彻底；ModelState 路由条件语义扩展为 hybrid 与 attention-free 两类。
- 团队侧：为后续 MRV2 默认切换扫清一类模型兼容问题；测试中 "setenv + disable_envs_cache 钉住老 runner" 的模式可被其他依赖 V1 内部实现的测试复用。
- 影响程度：低到中。改动量小且全部为向前兼容的增量判断，但处于 MRV2 迁移的关键路径上。
- 风险标记：MRV2 默认切换前置依赖，模型状态路由语义扩展，测试环境变量全局副作用（已缓解），shutdown 后属性访问隐式约定

关联脉络

- PR #46646 MRV2 大型迁移 PR（本 PR 的直接来源，未在本仓库近期列表中）：PR body 明确声明本 PR 是从 #46646 拆出的子集，聚焦 attention-free 模型支持与 CI 兼容。
- PR #50174 [3/N][Feat][Perf] Add new warmup infrastructure for JITs: 同为 MRV2 工作线的一部分，改造 v1/worker/gpu/model_runner.py 等文件，与本 PR 的 MRV2 默认切

换目标一致。

- PR #52329 [Performance][MRV2] Cache logits-processing request state: MRV2 默认切换前的另一块拼图，同样改动 vllm/v1/worker/gpu 路径，反映 MRV2 逐步补齐模型执行链路的演进方向。