# PR #43797 完整报告

vllm-project/vllm

[kv_offload] Skip decode-phase blocks in CPU offload

合并时间：2026-05-29 11:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43797

## 执行摘要

- 一句话：跳过 decode 阶段 KV block 的 CPU 卸载
- 推荐动作：值得精读，特别是如何通过 clamp 操作实现大幅性能提升，以及 Review 过程中设计演进（默认值、命名）的决策思路。

## 功能与动机

Reasoning models strip prior <think> across turns. A completed turn's decode KV is therefore reuse-dead: the thinking is dropped, and the answer is re-prefilled at a shifted position next turn (new block hashes). Offloading those blocks GPU→CPU burns PCIe bandwidth and evicts genuinely reusable prefill from a capacity-limited CPU tier.

## 实现拆解

1. 配置入口：在 vllm/v1/kv_offload/base.py 的 OffloadingSpec.__init__ 中从 extra_config 读取 offload_prompt_only，默认 True，并存储为实例属性。
2. 配置传递：在 vllm/distributed/kv_transfer/kv_connector/v1/offloading/scheduler.py 的 SchedulerOffloadConfig.from_spec 中将该属性值复制到 SchedulerOffloadConfig 的 offload_prompt_only 字段。
3. 核心逻辑：在 Scheduler._build_store_jobs 中，当 self.config.offload_prompt_only 为 True 时，将 num_offloadable_tokens 限制为 min(num_offloadable_tokens, req.num_prompt_tokens)，从而只允许 prompt 块进入卸载队列，decode 块被跳过。
4. 测试配套：在 tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py 新增 test_offload_prompt_only，验证仅 prompt 块被卸载；同时修改 tests/v1/kv_connector/unit/offloading_connector/utils.py 中的 RequestRunner 和 fixture 以支持 extra_config_overrides 参数，允许测试覆盖配置。

关键文件：

- vllm/v1/kv_offload/base.py（模块 卸载配置；类别 source；类型 configuration；符号 OffloadingSpec.init）：配置入口，读取 offload_prompt_only 并存储为属性。
- vllm/distributed/kv_transfer/kv_connector/v1/offloading/scheduler.py（模块 卸载调度；类别 source；类型 core-logic；符号 SchedulerOffloadConfig.from_spec, Scheduler._build_store_jobs）：核心逻辑改动：将配置传递到调度器，并在

_build_store_jobs 中实现 clamp 限制。

- tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py（模块 卸载测试；类别 test；类型 test-coverage；符号 test_offload_prompt_only）：新增 test_offload_prompt_only 单元测试，验证配置正确生效。
- tests/v1/kv_connector/unit/offloading_connector/utils.py（模块 测试工具；类别 test；类型 test-coverage；符号 RequestRunner.init, runner_factory）：修改 RequestRunner 和 fixture 以支持 extra_config_overrides，使测试能自定义配置。

关键符号：OffloadingSpec.init, SchedulerOffloadConfig.from_spec, Scheduler._build_store_jobs, test_offload_prompt_only

## 关键源码片段

### vllm/v1/kv_offload/base.py

配置入口，读取 offload_prompt_only 并存储为属性。

```
# vllm/v1/kv_offload/base.py
class OffloadingSpec(ABC):
    def __init__(self, vllm_config: "VllmConfig", kv_cache_config: "KVCacheConfig"):
        logger.warning(
            "Initializing OffloadingSpec. This API is experimental and "
            "subject to change in the future as we iterate the design."
        )
        self.vllm_config = vllm_config
        self.kv_cache_config = kv_cache_config

        kv_transfer_config = vllm_config.kv_transfer_config
        assert kv_transfer_config is not None
        self.extra_config = kv_transfer_config.kv_connector_extra_config

        # 当 offload_prompt_only 为 True 时，仅 prompt（prefill）块会被
        # 卸载到 CPU；decode 阶段生成的块被跳过。默认为 True，因为
        # 推理模型多轮对话中 decode KV 无法复用。
        self.offload_prompt_only: bool = bool(
            self.extra_config.get("offload_prompt_only", True)
        )
        # ... 后续初始化代码
```

### vllm/distributed/kv_transfer/kv_connector/v1/offloading/scheduler.py

核心逻辑改动：将配置传递到调度器，并在 _build_store_jobs 中实现 clamp 限制。

```
# vllm/distributed/kv_transfer/kv_connector/v1/offloading/scheduler.py
class Scheduler:
    # ...
    def _build_store_jobs(self, req: Request, ...) -> list[Job]:
        # ...
        if max_offload_tokens is not None:
            num_offloadable_tokens = min(num_offloadable_tokens, max_offload_tokens)
```

```
        # 当 offload_prompt_only 为 True 时，将可卸载 token 数限制为
        # prompt token 数，从而跳过 decode 阶段产生的块。
        if self.config.offload_prompt_only:
            num_offloadable_tokens = min(
                num_offloadable_tokens, req.num_prompt_tokens
            )
        # ... 后续过滤逻辑
```

## tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py

新增 test_offload_prompt_only 单元测试，验证配置正确生效。

```
# tests/v1/kv_connector/unit/offloading_connector/test_scheduler.py
@pytest.mark.parametrize("async_scheduling", [True, False])
def test_offload_prompt_only(request_runner, async_scheduling: bool):
    """验证 offload_prompt_only=True 时仅 offload prompt 块。

    配置：2 个 offloaded-block 的 prompt，然后生成足够的 decode token
    填充 4 个 offloaded block。标志将可卸载 token 数限制为 prompt 长度，
    因此只有 prompt 块（GPU offset 0-5）进入存储，decode 块（>=6）被跳过。
    请求故意不结束以避免 flush 时序干扰。
    """
    gpu_block_size = 4
    block_size_factor = 3
    offloaded_block_size = gpu_block_size * block_size_factor # 12
    num_prompt_blocks = 2
    num_decode_blocks = 4
    prompt_offsets = (0, 1, 2, 3, 4, 5)

    runner = request_runner(
        block_size=gpu_block_size,
        num_gpu_blocks=100,
        async_scheduling=async_scheduling,
        block_size_factor=block_size_factor,
        extra_config_overrides={"offload_prompt_only": True},
    )

    runner.manager.prepare_store.side_effect = (
        lambda keys, req_context: generate_store_output(keys)
    )

    runner.new_request(token_ids=[0] * offloaded_block_size * num_prompt_blocks)
    runner.run(
        decoded_tokens=[0] * (offloaded_block_size * num_decode_blocks),
        expected_stored=prompt_offsets,
    )

    # 额外检查：仅 prompt 的 key 出现在 prepare_store 调用中
    offered_keys = {
        key
```

```
        for call in runner.manager.prepare_store.call_args_list
        for key in call.args[0]
    }
    assert len(offered_keys) == num_prompt_blocks
```

## 评论区精华

Reviewer @orozery 提出三点关键意见：

1) 默认值应改为 False（即默认跳过 decode 卸载）； 2) 配置名从 offload_decode_blocks 改为 offload_prompt_only 更清晰； 3) 注释中移除 GPU→CPU 表述以支持不同后端。作者 @Etelis 采纳建议，翻转默认值为 True（跳过 decode），重命名配置，更新注释并添加单元测试。最终获得批准。

- 配置默认值与命名 (design): 接受 reviewer 建议，翻转默认值为 True，重命名为 offload_prompt_only，注释中立化。
- 添加单元测试 (testing): 测试已添加，覆盖异步和同步调度模式。

## 风险与影响

- 风险：默认行为从卸载所有块变为仅卸载 prompt 块，影响使用 KV offload 但依赖 decode 块卸载的用户（如开启 prefix caching 且 decode 块仍有复用可能的场景）。但多数情况下新版更优，且可通过设置 offload_prompt_only=False 恢复旧行为，风险较低。改动集中在调度器内部，不涉及其它模块。
- 影响：对使用 KV offload 的用户：默认减少 82% 的 GPU→CPU 写入，提升 CPU 缓存命中率，显著降低多轮对话 TTFT。对未使用 offload 的用户无影响。对开发：新增配置需文档说明，但易于理解。
- 风险标记：默认行为变更，核心路径变更

## 关联脉络

- PR #43205 [KV Offload] Add per-request offloading policy via on_new_request lifecycle hook: 同为 kv_offload 模块，引入 per-request 策略框架，本 PR 基于该框架扩展了配置项。