PR #27607 完整报告

sgl-project/sglang

Support spec v2 for Frozen-KV MTP; remove v1 worker

合并时间：2026-06-10 06:30

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27607

执行摘要

- 一句话：Frozen-KV MTP 迁移到 spec v2 并删除 v1 worker
- 推荐动作：此 PR 值得精读，特别是学习如何将一种推测解码算法从 v1 迁移到 v2 orchestrator 架构。关键设计决策包括：直接将 FrozenKVMTPDraftWorker 继承 BaseDraftWorker 而非 EAGLEWorkerV2，从而复用 EAGLE 的 verify 契约；通过 AST 移动等价性脚本验证重构正确性。对于代码库维护者，建议关注被删除测试的替代覆盖；对于 spec v2 新用户，可参考此 PR 将其他算法也迁移到 v2。

功能与动机

让 Frozen-KV MTP 能利用 spec v2 的 orchestrator 架构，从而支持 overlap 调度（提升吞吐）并统一与 EAGLE 等其他推测解码算法的代码路径。PR body 明确说明："Run Frozen-KV MTP on the spec v2 worker — FrozenKVMTPWorkerV2 (orchestrator, subclasses EAGLEWorkerV2) + FrozenKVMTPDraftWorker (draft layer) — supporting both overlap and non-overlap scheduling"。此外，关联 issue #23802 和 #25980 修复了 spec v2 下的 stop 字符串检测边界问题，使 v2 精度能与 v1 匹配。

实现拆解

1. 实现 FrozenKVMTPDraftWorker：在 frozen_kv_mtp_worker_v2.py 中新增该类，继承 BaseDraftWorker 和 TpModelWorker，将 v1 worker 中大部分方法（如 draft_forward、forward_batch_generation 等）按 spec v2 契约重写，使其可作为 orchestrator 的 draft layer 被调用。
2. 修改调度路由：在 spec_info.py 的 SpeculativeAlgorithm.create_worker 方法中，将 FROZEN_KV_MTP 的 worker 创建从 v1 的 FrozenKVMTPWorker 改为 FrozenKVMTPWorkerV2（即 v2 orchestrator），并移除原来的 overlap 禁用限制。
3. 清理依赖与数据类：删除 v1 专属的 frozen_kv_mtp_worker.py 整个文件；从 frozen_kv_mtp_utils.py 中移除不再使用的 select_last_verified_seed 和 capture_for_decode 函数；从 frozen_kv_mtp_info.py 中删除 FrozenKVMTPVerifyOutput 别名及相关的 _to_frozen_kv_mtp_draft_extend_input 转换逻辑，因为 v2 的 EagleVerifyOutput 可直接使用。
4. 调整测试配套：删除针对 v1 worker 的单元测试 test_frozen_kv_mtp_all_reqs_finish_in_verify.py；修改 3 个集成测试文件（test_frozen_kv_mtp.py、test_gemma4_mtp_26b_a4b_extra.py、test_gemma4_mtp_31b_extra.py），移除

--disable-overlap-schedule 参数（因为 v2 默认支持 overlap），并添加 --speculative-algorithm FROZEN-KV-MTP 显式指定算法。

5. 其他配套：在 cuda_graph_runner 和 scheduler 中做微小调整，适应 v2 worker 的接口变化；在 arg_groups/speculative_hook.py 中更新算法检查，确保 FROZEN_KV_MTP 能在 overlap 开启时正常启动。

关键文件：

- python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py（模块 推测解码；类别 source；类型 dependency-wiring；符号 FrozenKVMTPDraftWorker, init, draft_model_runner, draft_runner）：核心实现文件：从占位符（raise NotImplementedError）重写为完整的 v2 draft worker（FrozenKVMTPDraftWorker），包含 init、draft forward、verify 等所有关键方法，与 EAGLEWorkerV2 构成 orchestrator 层。
- python/sglang/srt/speculative/frozen_kv_mtp_worker.py（模块 推测解码；类别 source；类型 deletion；符号 FrozenKVMTPWorker, init, draft_model_runner, get_attn_backend）：被删除的 v1 worker 核心文件，包含 FrozenKVMTPWorker 类及所有辅助方法。彻底移除意味着不再支持 v1 架构。
- python/sglang/srt/speculative/spec_info.py（模块 推测解码；类别 source；类型 dependency-wiring；符号 supports_spec_v2, create_worker）：修改调度路由：允许 FROZEN_KV_MTP 算法使用 V2 worker，移除了之前对 overlap 的报错限制。这是启用 v2 的关键开关。
- python/sglang/srt/speculative/frozen_kv_mtp_utils.py（模块 推测解码；类别 source；类型 core-logic；符号 select_last_verified_seed, capture_for_decode）：移除两个不再需要的辅助函数（select_last_verified_seed 和 capture_for_decode），简化工具层。
- test/registered/unit/spec/test_frozen_kv_mtp_all_reqs_finish_in_verify.py（模块 测试；类别 test；类型 deletion；符号 _stale_verify_input, _make_prefill_draft_input, _FakeVerifyOutput, iter）：删除的 v1 worker 专用测试，测试 verify 完成所有请求后安装 idle draft 的逻辑。v2 通过基类 mixin 处理，但缺少直接替代测试。
- python/sglang/srt/speculative/frozen_kv_mtp_info.py（模块 推测解码；类别 source；类型 core-logic；符号 verify, _to_frozen_kv_mtp_draft_extend_input）：删除 FrozenKVMTPVerifyOutput 别名和转换函数 _to_frozen_kv_mtp_draft_extend_input，简化数据类层次。

关键符号：FrozenKVMTPDraftWorker.init, frozen_kv_target_view, select_last_verified_seed（已删除）, capture_for_decode（已删除）, SpeculativeAlgorithm.supports_spec_v2, SpeculativeAlgorithm.create_worker

关键源码片段

python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py

核心实现文件：从占位符（raise NotImplementedError）重写为完整的 v2 draft worker（FrozenKVMTPDraftWorker），包含 init、draft forward、verify 等所有关键方法，与 EAGLEWorkerV2 构成 orchestrator 层。

```
class FrozenKVMTPDraftWorker(BaseDraftWorker, TpModelWorker):
    """Frozen-KV MTP draft worker.

    The assistant reads target KV only. It reuses EAGLE's verify input/output
    contract, but owns the seed and recurrent draft loop because there is no
    assistant-side KV extension.
    """

    def __init__(
        self,
        server_args: ServerArgs,
        gpu_id: int,
        tp_rank: int,
        dp_rank: Optional[int],
        moe_ep_rank: int,
        attn_cp_rank: int,
        moe_dp_rank: int,
        nccl_port: int,
        target_worker: TpModelWorker,
    ):
        # 保存配置，与 v1 相同的初始化逻辑
        self.server_args = server_args
        self.topk = server_args.speculative_eagle_topk
        self.speculative_num_steps = server_args.speculative_num_steps
        self.speculative_num_draft_tokens = server_args.speculative_num_draft_tokens
        self.gpu_id = gpu_id
        self.device = server_args.device
        self.target_worker = target_worker
        self.page_size = server_args.page_size
        self.speculative_algorithm = SpeculativeAlgorithm.from_string(
            server_args.speculative_algorithm
        )
        assert self.speculative_algorithm.is_frozen_kv_mtp(), (
            "FrozenKVMTPDraftWorker should only be instantiated for "
            "SpeculativeAlgorithm.FROZEN_KV_MTP, got "
            f"{self.speculative_algorithm.name}."
        )
        # 禁用 CUDA graph 自动捕获，由 worker 内部管理
        backup_disable_cuda_graph = server_args.disable_cuda_graph
        server_args.disable_cuda_graph = True

        # 共享 target 的 memory pool（只读）
        self.req_to_token_pool, self.token_to_kv_pool_allocator = (
            target_worker.get_memory_pool()
        )

        target_cfg = target_worker.model_runner.memory_pool_config
        draft_pool_config = MemoryPoolConfig(
            max_total_num_tokens=64, # 占位值
```

```
            max_running_requests=target_cfg.max_running_requests,
        )

        self.hot_token_id = None

        # 初始化 attention backend
        with (
            empty_context()
        ), speculative_moe_backend_context(), speculative_moe_a2a_backend_context():
            super().__init__(
                server_args=server_args,
                gpu_id=gpu_id,
                tp_rank=tp_rank,
                dp_rank=dp_rank,
                moe_ep_rank=moe_ep_rank,
                attn_cp_rank=attn_cp_rank,
                moe_dp_rank=moe_dp_rank,
                nccl_port=nccl_port,
            )
        server_args.disable_cuda_graph = backup_disable_cuda_graph

        self.draft_runner = self.model_runner # 别名
```

python/sglang/srt/speculative/spec_info.py

修改调度路由：允许 FROZEN_KV_MTP 算法使用 V2 worker，移除了之前对 overlap 的报错限制。这是启用 v2 的关键开关。

```
    def supports_spec_v2(self) -> bool:
        # 之前返回 self.is_eagle() and not self.is_frozen_kv_mtp() or self.is_standalone()
        # 现在允许 frozen-KV MTP 也支持 v2
        return self.is_eagle() or self.is_standalone()

    def create_worker(
        self,
        ...
    ):
        ...
        if self.is_frozen_kv_mtp():
            # 之前：若 enable_overlap 则 raise ValueError
            # 现在：始终返回 V2 worker（scheduler 根据 overlap 配置决定同步或异步执行）
            from sglang.srt.speculative.frozen_kv_mtp_worker_v2 import (
                FrozenKVMTPWorkerV2,
            )
            return FrozenKVMTPWorkerV2
```

评论区精华

Review 中没有实质性讨论，只有一个 approval。Issue 评论中主要是作者触发的 CI 重跑命令（/rerun-test），涉及 test_frozen_kv_mtp.py 和 test_gemma4_mtp_31b_extra.py 等测试

的失败重试，最终全部通过。无未解决争议。

- 测试失败与重跑 (testing): 测试通过，无需代码修改。

风险与影响

- 风险：
  1. 精度回归风险：v2 架构下 draft 与 verify 的交互模式与 v1 不同，若存在未覆盖的 corner case（如 stop 字符串出现在多 token 接受中间），可能产生精度下降。作者已通过 GSM8K 评测和确定性测试验证 v2 与 v1 精度一致（~0.730），且依赖的 stop 修复 issue #25980 已合入 main，风险较低。
  2. 性能风险：v2 orchestrator 引入额外上下文切换，可能增加延迟；但 overlap 调度可带来吞吐提升。评测显示 top-1 接受长度约 2.83，与 v1 一致，表明无显著性能退化。
  3. 测试覆盖损失：删除了 test_frozen_kv_mtp_all_reqs_finish_in_verify.py，该测试专门验证 v1 的"verify 完成所有请求后安装 idle draft"逻辑。新代码通过 FrozenKVMTPDraftWorker 继承的 EagleDraftInputV2Mixin 来处理类似场景，但缺少针对性单元测试，存在回归隐患。
  4. 兼容性风险：删除了 frozen_kv_mtp_worker.py，任何外部代码若直接引用 FrozenKVMTPWorker 将断裂。仓库内所有引用已通过本 PR 更新，但外部插件或定制代码需同步迁移。
  5. CUDA 图兼容性：v2 worker 使用不同的 draft 输入类型，若 CUDA 图捕获未正确处理，可能导致运行时错误。PR 中已调整 frozen_kv_mtp_cuda_graph_runner.py，风险可控。 - 影响：影响范围：中等。影响所有使用 --speculative-algorithm FROZEN-KV-MTP 的用户。用户无需修改命令行参数即可获得 v2 架构支持（overlap 默认启用）。删除 v1 后，代码库减少约 800 行，维护负担降低。影响程度：对于开启 overlap 调度的场景，吞吐可能提升（与 EAGLE 一致）；对于关闭 overlap 的场景，行为透明。测试和评测均显示精度不变。团队后续可复用 EAGLE v2 的 infrastructure 改进（如 piecewise CUDA graph 支持）于 Frozen-KV MTP。 - 风险标记：核心路径变更，测试覆盖变化，算法迁移

关联脉络

- PR #23802 fix: stop-string check misses early matches during speculative decoding: 本 PR 依赖此修复来确保 spec v2 下 stop 字符串被正确检测，否则 v2 会多生成 token 导致精度下降。
- PR #25980 Fix spec v2 stop output boundary: 进一步增强 stop 边界处理，确保多 token 接受时输出被正确截断，本 PR 的精度验证依赖此修复已合入 main。