PR #35496 完整报告

sgl-project/sglang

[Spec] Support quantized target lm_head in the DFlash2 selector

合并时间：2026-08-20 09:06

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35496

执行摘要

- 一句话：DFlash2 选择器支持量化目标 lm_head，mask 尾部保连续
- 推荐动作：值得精读。核心看点是“mask padding 尾部而非 crop”如何同时解决 flashinfer 连续性约束与 TP 挂起问题，以及“借目标模型既有 CUDA graph 路径证明量化 kernel 可捕获”的推理方式；测试用 fake quant method 和 top-k 契约桩做 mutation-verified 覆盖，方法论也可借鉴。建议重点关注后续 #35580/#35581 对 degenerate-config 的补强。

功能与动机

PR body 明确说明：DFlash2's selector matmuls draft hiddens against the target lm_head.weight directly, so quantized targets (ModelOpt NVFP4, compressed-tensors FP8) fail with RuntimeError: DFlash2 selector requires a dense FP16/BF32/FP32 target lm_head — on 32GB GPUs those checkpoints are the only way to serve Qwen3.8-27B。#34859 已通过 quant_method.apply 修复 DSpark 的同类问题，本 PR 为 DFlash2 selector 补齐同样能力，并额外处理 padded vocab 下 crop 视图非连续导致的 flashinfer top-k 失败和 TP>1 挂起问题。

实现拆解

1. 统一投影入口（python/sglang/srt/models/dflash.py）：新增 _project_candidate_logits(hidden, lm_head, num_org, use_quant_head)，dense 分支保持原有 matmul(hidden, weight[:num_org].T)；量化分支调用 lm_head.quant_method.apply(lm_head, hidden, None) 后 .contiguous()，并将 padding 尾部 logits[:, num_org:] 置为 -inf。选择 mask 而非 crop 是关键设计：packed weight 无法按 org vocab 行切片，而 crop 产生非连续视图，会被 flashinfer radix top-k 的 CHECK_INPUT 拒绝。
2. 变更准入判定（dflash.py compute_candidates）：用 should_apply_lm_head_quant_method(lm_head, quant_method) 判定是否走量化路径；门控通过则不再要求 dense head，错误信息同步放宽为“dense 或受支持的 quant_method”。tp_size == 1 与 tp > 1 两个分支都改调 _project_candidate_logits，TP 分支保留 org_vocab_start_index 偏移和 all-gather 前的 fp32 转换，保证跨 shard 排序精度。
3. worker侧CUDAgraph折叠准入（python/sglang/srt/speculative/dflash_worker_v2.py）：_maybe_build_draft_sampler 的 selector 分支把原来的“必须有 dense weight”检查改为

“dense 或 quant gate 通过即可准入”，量化与 dense head 走同一 _SelectorDraftSampler graph-folded 路径；理由是被准入的量化 head 所执行的 kernel 与目标模型自身 logits 路径在 CUDA graph 下的一致。非 selector 的 static-matmulsampler 分支仍保留原 dense-only 检查。

4. 测试配套（test/registered/unit/spec/test_dflash_logits.py）：新增 _flashinfer_contract_topk 桩（断言输入连续、sorted/deterministic 为真）和 _FakeQuantMethod（故意返回 strided 视图、制造占主导的 padding 垃圾），新增三个测试分别钉住：量化 head 经 quant_method 投影且 mask 生效、TP=2 下 per-shard org-vocab 限制与全局 id 偏移及 fp32 转换、worker 对 gate 通过的 packed head 准入并折叠、对不支持的 head 保持 eager。

关键文件：

- python/sglang/srt/models/dflash.py（模块 选择器；类别 source；类型 core-logic；符号 _project_candidate_logits, compute_candidates, _radix_topk）：核心实现文件：新增 _project_candidate_logits 统一 dense/quant 两条投影路径，compute_candidates 两个 TP 分支均接入，并通过 should_apply_lm_head_quant_method 放宽准入条件。
- python/sglang/srt/speculative/dflash_worker_v2.py（模块 草稿 worker；类别 source；类型 dependency-wiring；符号 _maybe_build_draft_sampler）：worker 侧 CUDA graph 折叠准入逻辑：selector 分支从仅接受 dense head 放宽为 dense 或 quant gate 通过即准入，使量化 head 与 dense head 走同一 graph-folded _SelectorDraftSampler 路径。
- test/registered/unit/spec/test_dflash_logits.py（模块 单元测试；类别 test；类型 test-coverage；符号 _flashinfer_contract_topk, _FakeQuantMethod, test_selector_projects_a_quantized_target_lm_head_through_its_quant_method, test_selector_gathers_global_candidates_across_vocab_shards）：测试配套：新增 flashinfer 契约桩与 fake quant method，三个新测试分别钉住量化投影 +mask、TP 聚合契约（org-vocab 限制 / 全局 id 偏移 /fp32 转换）和 worker 准入折叠行为，且通过 mutation 验证每个关键步骤都有测试覆盖。

关键符号：_project_candidate_logits, compute_candidates, _maybe_build_draft_sampler

关键源码片段

python/sglang/srt/models/dflash.py

核心实现文件：新增 _project_candidate_logits 统一 dense/quant 两条投影路径，compute_candidates 两个 TP 分支均接入，并通过 should_apply_lm_head_quant_method 放宽准入条件。

```
# python/sglang/srt/models/dflash.py 中的核心新增逻辑
def _project_candidate_logits(
    hidden: torch.Tensor, lm_head: nn.Module, *, num_org: int, use_quant_head: bool
) -> torch.Tensor:
    """把 draft hidden 经过 target head 投影，限制在 org vocab 范围内。"""
    if not use_quant_head:
        # dense 路径保持原行为：按 org vocab 行切片后直接 matmul。
```

```
            weight = lm_head.weight
            return torch.matmul(hidden.to(weight.dtype), weight[:num_org].T)
        # packed 权重无法像 dense 路径那样按 org vocab 行切片；
        # 且 flashinfer radix top-k 的 CHECK_INPUT 要求输入连续，crop 出来的视图
        # 非连续会导致 TP=1 报错、TP>1 只有持 tail 的 rank 报错而其余 rank 进入
        # all-gather，最终挂起。所以这里改为把 padding 尾部 mask 成 -inf。
        logits = lm_head.quant_method.apply(lm_head, hidden, None).contiguous()
        if logits.shape[-1] > num_org:
            logits[:, num_org:] = float("-inf")
        return logits

    # compute_candidates 中两个 TP 分支统一走 _project_candidate_logits：
    # tp_size == 1 时直接对本地 org vocab 做 top-k 并返回；
    # tp > 1 时每个 shard 先在本地 org vocab 上做 top-k，再以
    # org_vocab_start_index 修正全局 id，并对 vals 先转 fp32 再 all-gather，
    # 避免 bf16 低精度影响跨 shard 的全局排序。
```

python/sglang/srt/speculative/dflash_worker_v2.py

worker 侧 CUDA graph 折叠准入逻辑：selector 分支从仅接受 dense head 放宽为 dense 或 quant gate 通过即准入，使量化 head 与 dense head 走同一 graph-folded _SelectorDraftSampler 路径。

```
    # python/sglang/srt/speculative/dflash_worker_v2.py 中 _maybe_build_draft_sampler 的 selector
    分支
    def _maybe_build_draft_sampler(self):
        def _eager(reason):
            # 返回 None 表示保持 eager，不折叠进 draft CUDA graph。
            if self.ps.tp_rank == 0:
                logger.info("DFLASH draft greedy head kept eager (reason=%s).", reason)
            return None

        # ... 前面的 eager 条件（环境变量、block_size 等）省略 ...
        target_model = self._target_worker.model_runner.model
        lm_head = getattr(target_model, "lm_head", None)
        if lm_head is None:
            return _eager("no target lm_head")

        if self.selector is not None:
            # compute_candidates 需要在 capture 前把 target lm_head 挂到 draft 模型上。
            # 通过 quant gate 的量化 head 也可以安全折叠：目标模型自身的 logits 路径
            # 在 CUDA graph 下已经运行过同一个 kernel，capture 行为一致。
            if not is_dense_head_weight(
                getattr(lm_head, "weight", None)
            ) and not should_apply_lm_head_quant_method(
                lm_head, getattr(lm_head, "quant_method", None)
            ):
                return _eager("unsupported quantized lm_head")
            self.draft_model.lm_head = lm_head
            if self.ps.tp_rank == 0:
```

```
            logger.info(
                "DFLASH selector decode (greedy + sampling) folded into the "
                "draft cuda graph."
            )
        return _SelectorDraftSampler(
            draft_model=self.draft_model,
            block_size=self.block_size,
            max_bs=max(get_exec().graph.cuda_graph_config.decode.bs),
            device=self.device,
        )

    # 非 selector 分支（static matmul sampler）仍然只接受 dense head：
    # 量化 head (FP8/INT) 会破坏静态 matmul，所以继续走原有 dense-only 检查。
    if not hasattr(lm_head, "weight"):
        return _eager("quantized lm_head has no dense weight")
    if not is_dense_head_weight(lm_head.weight):
        return _eager("quantized lm_head")
    # ... 后续 tp_group shard_indices 处理省略 ...
```

评论区精华

该 PR 没有正式 review 评论（review_comments_count = 0），但 PR body 详细阐述了三个关键设计结论：

- mask 尾部代替 crop：padded local vocab 若用 crop 会得到非连续视图，flashinfer 的 radix top-k 在 TP=1 时 CHECK_INPUT 失败、在 TP>1 时只有持有 tail 的 rank 报错而其他 rank 进入 all-gather，从而挂起；mask 成 -inf 则零拷贝保持连续。
- 与 #35462 重叠：作者明确承认与 @ryou315 的 #35462 重叠并 credit 为 co-author，本 PR 额外贡献了 padding tail mask 与连续性处理。
- 拆出 follow-up：#35580（degenerate-config guard）与 #35581（optional cleanup）从本 PR 拆分，保持每个 PR 可独立 review。
  - 暂无高价值评论线程

风险与影响

- 风险：
  1. 核心推理路径变更：compute_candidates 是 DFlash2 选择器候选生成主路径，所有 DFlash2 用户都会经过；dense head 走原分支逻辑不变，但量化分支的新增依赖（quant_method.apply、org_vocab_size、padding 语义）需要各量化实现正确支持。
  2. TP>1 挂起风险：mask 方案依赖 logits.shape[-1] > num_org 判断与 org_vocab_size/shard_indices 的正确性；若某量化实现输出宽度与 org vocab 关系不符，仍可能出现跨 rank 行为不一致。测试已用 fake all-gather 钉住契约，但真实 NVFP4/FP8 TP 场景仍需回归验证。
  3. CUDA graph 捕获风险：worker 准入的前提是“被 gate 通过的量化 head 的 kernel 与目标自身 logits 路径在 CUDA graph 下已运行过”；如果某个 quant_method.apply 内部存在不可捕获分支，graph capture 阶段会失败。当前测试未覆盖真实

quant_method 的 capture。

4. 数值精度差异：作者实测 spec 与 target-only 在 near-tie token 处分叉，GSM8K 200q 精度 95.5% vs 98.0%，是稳定数值偏移而非损坏，但量化 head 下 spec 精度仍有可感知下降。 - 影响：对用户：DFlash2 + 量化 target（ModelOpt NVFP4、compressed-tensors FP8）从完全不可用变为可服务，尤其惠及 32GB GPU 上服务 Qwen3.8-27B 的场景；dense head 用户无感知。对系统：选择器路径新增量化分支，CUDA graph 折叠范围扩大；对团队：与 #35462 协作合并，后续 #35580/#35581 继续演进相关防御性检查，说明该功能线仍处活跃迭代期。 - 风险标记：核心路径变更（selector 候选生成），TP>1 潜在挂起，CUDA graph 捕获依赖 quant_method 实现，量化 head 下 spec 精度存在数值差异

关联脉络

- PR #35462 [Spec] Overlapping PR (co-author ryou315): PR body 明确说明本 PR 与 #35462 重叠并 credit @ryou315 为 co-author；本 PR 额外包含 padding tail mask 的连续性处理。
- PR #34859 [Spark/DSpark] Fix quantized target lm_head via quant_method.apply: PR body 引用 #34859 为同类问题的先例：通过 quant_method.apply 修复 DSpark 的量化目标 lm_head，本 PR 为 DFlash2 selector 做同样处理。
- PR #35580 [Spec] Degenerate-config guard (follow-up): PR body 说明 #35580 是从本 PR 拆出的 follow-up，用于补充 degenerate-config 防御，与本 PR 的准入判定直接相关。
- PR #35581 [Spec] Optional cleanup (follow-up): PR body 说明 #35581 是从本 PR 拆出的 follow-up，负责可选清理工作，保持本 PR 可独立 review。
- PR #35228 [Quant] Load compressed-tensors quantized lm_head instead of value-casting it: 同为 quantized lm_head 加载与量化路径修复，compressed-tensors FP8 是本文档中提到的目标 head 量化方案之一，相关代码路径可互相验证。
- PR #35397 Support custom draft worker classes in DSpark: 同为 speculative decoding 框架中 draft worker 的定制与准入逻辑演进，与本 PR 的 dflash_worker_v2 准入改动处于同一功能线。