PR #23962 完整报告

sgl-project/sglang

[Spec] Split `accept_length` into `num_accepted_drafts` and `num_accepted_tokens`

合并时间：2026-04-29 15:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23962

执行摘要

- 一句话：拆分 accept_length 为 drafts 和 tokens 两个字段
- 推荐动作：值得精读，尤其是 EagleDraftInput 的字段设计、CUDA 图运行器的双缓冲策略、以及 eagle_info_v2.py 中 sample() 的变异解耦。这些设计决策可以在类似需要消除隐式语义的场景中复用。

功能与动机

accept_length 字段在推测解码中同时被用于两种语义：有时只计已接受 draft 数，有时包含 bonus token（drafts + 1）。这种模糊性导致代码中频繁出现 + 1 和 add_(1) 原地变异，易于引入 bug。此 PR 将其拆分为两个显式字段，让每个消费者直接读取语义匹配的值，提升可维护性和正确性。（见 PR body）

实现拆解

1. 数据契约拆分：在 EagleDraftInput 和 NgramVerifyInput 中添加 num_accepted_tokens 字段（torch.Tensor 和 List[int]），与原有的 num_accepted_drafts 并存。num_accepted_drafts 只计数通过验证的 draft token 数（不包含 bonus），num_accepted_tokens 包含 bonus token（即 num_accepted_drafts + 1）。
2. 生命周期解耦：在 eagle_info_v2.py 的 sample() 中，移除 .add_(1) 原地变异，改为在返回值中直接返回 num_accepted_drafts + 1。在 eagle_info.py 的 prepare_extend_after_decode() 中，不再修改 self.num_accepted_drafts，而是使用局部变量 extend_lens 传递给 Triton kernel。所有写站点（verify kernel 输出、状态重算、V2 worker 赋值、CUDA 图别名）同时设置两个字段。
3. 注意力后端适配：所有注意力后端（aiter, flashattention, trtllm_mha, nsa, nsa_backend_mtp_precompute, wave, triton）中，将原本 spec_info.accept_length + 1 读取替换为直接读取 spec_info.num_accepted_tokens（或 num_accepted_tokens_cpu）。
4. CUDA 图运行器更新：EagleDraftExtendInputBuffers 和 MultiLayerEagleDraftExtendInputBuffers 新增 num_accepted_tokens 张量作为并行缓冲区，在 replay 时同时拷贝两个字段。缓冲区的初始化、填充、后处理均调整为双字段操作。

5. 重命名与测试：将内部变量名 accept_length 统一改为 num_accepted_drafts，局部变量保留 accept_lens 表示含 bonus 的值。更新所有导入引用和测试文件（test_eagle_infer_a.py 等）。保留 Prometheus 指标 accept_length 名称以保持兼容性。

关键文件：

- python/sglang/srt/speculative/spec_utils.py（模块 推测解码工具；类别 source；类型 core-logic；符号 create_accept_length_filter, create_num_accepted_drafts_filter, get_target_cache_loc, filter_finished_cache_loc_kernel）：核心 Triton kernel 和工具函数，修改了 get_target_cache_loc、get_src_tgt_cache_loc、filter_finished_cache_loc_kernel 等函数的参数名和变量名，并新增 create_num_accepted_drafts_filter 替换原来的 filter。
- python/sglang/srt/speculative/eagle_info.py（模块 Eagle 解码；类别 source；类型 core-logic；符号 create_num_accepted_drafts_filter, num_accepted_drafts, num_accepted_drafts_per_req_cpu）：Eagle 验证模块的主要逻辑，修改了 verify() 方法中的变量名和返回字段名，并更新导入引用。
- python/sglang/srt/speculative/multi_layer_eagle_draft_extend_cuda_graph_runner.py（模块 CUDA 图运行器；类别 source；类型 core-logic；符号 num_accepted_drafts, num_accepted_tokens）：多层 Eagle 的 CUDA 图输入缓冲区结构体新增 num_accepted_drafts 和 num_accepted_tokens 字段，图重放逻辑相应调整。
- python/sglang/srt/speculative/eagle_draft_extend_cuda_graph_runner.py（模块 CUDA 图运行器；类别 source；类型 core-logic；符号 num_accepted_drafts, num_accepted_tokens）：单层 Eagle 的 CUDA 图输入缓冲区同样拆分字段，replay 方法中同步拷贝两个字段。
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 Eagle Worker；类别 source；类型 core-logic；符号 accept_lens, num_accepted_drafts, num_accepted_tokens）：Eagle V2 worker 的 verify 和 draft extend 路径，涉及字段赋值和参数传递的重命名。
- python/sglang/srt/model_executor/forward_batch_info.py（模块 ForwardBatch；类别 source；类型 data-contract；符号 num_accepted_drafts, num_accepted_tokens）：ForwardBatch 的 pad 和 post_forward 操作需要同时处理两个字段，保证张量正确截断。

关键符号：create_num_accepted_drafts_filter, get_target_cache_loc, get_src_tgt_cache_loc, filter_finished_cache_loc_kernel, EagleVerifyInput.verify, eagle_info_v2.sample, EagleDraftInput.prepare_extend_after_decode, EagleDraftWorker._draft_extend_for_decode, EagleDraftWorker.move_accepted_tokens_to_target_kvcache, MultiLayerEagleDraftExtendCudaGraphRunner.init_buffers_and_capture, EAGLEDraftExtendCudaGraphRunner.replay, ForwardBatch.post_forward_mlp_sync_batch

关键源码片段

python/sglang/srt/speculative/spec_utils.py

核心 Triton kernel 和工具函数，修改了 get_target_cache_loc、get_src_tgt_cache_loc、filter_finished_cache_loc_kernel 等函数的参数名和变量名，并新增 create_num_accepted_drafts_filter 替换原来的 filter。

```
@triton.jit
def get_target_cache_loc(
    tgt_cache_loc,
    to_free_slots,
    num_accepted_drafts, # 原为 accept_length ，现语义为 drafts-only（不含 bonus）
    to_free_num_slots,
    out_cache_loc,
    num_verify_tokens: tl.constexpr,
    num_verify_tokens_upper: tl.constexpr,
    bs_upper: tl.constexpr,
):
    bid = tl.program_id(axis=0)
    offset = tl.arange(0, num_verify_tokens_upper)
    bs_offset = tl.arange(0, bs_upper)

    # 写入第一部分：将已接受的 token 复制到 tgt_cache_loc
    accept_len_all = tl.load(num_accepted_drafts + bs_offset, mask=bs_offset < bid)
    tgt_cache_loc_start = tl.sum(accept_len_all) + bid
    copy_len = tl.load(num_accepted_drafts + bid) + 1 # 需 +1 包含 bonus token
    out_cache_loc_row = tl.load(
        out_cache_loc + bid * num_verify_tokens + offset, mask=offset < copy_len
    )
    tl.store(
        tgt_cache_loc + tgt_cache_loc_start + offset,
        out_cache_loc_row,
        mask=offset < copy_len,
    )

    # 写入第二部分：处理 to_free_slots（省略）
    ...
```

评论区精华

无 review 评论。PR 作者独立完成 19 个 commit，含多次合并 main 和修复一个因拆分暴露的 bug（NSA 后端 extend_seq_lens 缺失 +1）。

- 整体设计无争议 (design): 设计已实施并合并。

风险与影响

- 风险：该 PR 修改了 30 个文件，涉及整个推测解码栈（Eagle v1/v2、Ngram、多层 Eagle、DP attention、多种注意力后端）。主要风险是字段赋值不一致导致运行时崩溃或静默错误。CI 已计划覆盖关键路径（eagle、ngram、dflash、standalone），但非回归测试（如 DeepSeek V3.2 + NSA）可能未覆盖。另外，所有注意力后端均需正确读取新字段，遗漏一处可能导致隐式错误。commit 0ca242f 修复了一个因拆分暴露的 bug，说明此类风险存

在。

- 影响：对最终用户无影响：所有 CLI 参数、Prometheus 指标名称（SpeculativeMetrics.accept_length 保留）和 meta_info 键不变。对内部开发者有正面影响：字段语义显式化，降低了代码阅读和维护成本，避免了 + 1 心智负担。性能影响可忽略：每个请求多一个 int32 tensor（约几 KB）。团队需要学习新命名约定。
- 风险标记：逻辑语义变更，跨 30 个文件联动，多个注意力后端需验证

关联脉络

- PR #23890 [spec decoding] add extra attribute 'spec_hidden_size': 同样涉及 speculative decoding 核心数据结构的字段添加，修改了 eagle_worker_v2.py 和 forward_batch_info.py 等重叠文件。
- PR #23631 [HiCache][SPEC] fix: normalize storage prefetch key: 与 speculative decoding 相关，修改了 spec_utils.py 等文件。