

PR #27599 完整报告

sgl-project/sglang

[Spec] Naming cleanup: contiguous draft-loc kernel + `accepted`->`accept`

合并时间: 2026-06-09 06:04

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27599>

执行摘要

- 一句话: 规范 Spec V2 代码命名, `accepted`→`accept`, `page_size_1`→`contiguous`
- 推荐动作: 此 PR 是代码规范化的良好实践, 值得精读以理解团队命名约定。对于关注 speculative decoding 的实现细节的工程师, 重命名后的函数名更易理解其语义 (`accept` 而非 `accepted`)。

功能与动机

PR 标题及 body 明确说明是 'Spec naming cleanup (behavior-preserving renames)', 堆叠在已合并的 #27552 之上。通过消除 'accepted' (过去分词) 与 'accept' (动词原形) 混用、'page_size_1' 与 'contiguous' 不匹配的命名不一致, 提升代码可读性和维护性。

实现拆解

1. Eagle Worker V2 核心逻辑重命名: 在 `eagle_worker_v2.py` 中将三个内部方法重命名, 并更新所有调用点 (包括 `verify` 和 `_mamba_verify_update` 中的引用)。这些方法用于在验证后压缩被接受的树路径、移动 KV cache 等。
2. Triton Kernel 重命名: 在 `triton_ops/eagle.py` 中将 `fill_accepted_out_cache_loc` 重命名为 `fill_accept_out_cache_loc`, 并在 `triton_ops/cache_locs.py` 中将 `assign_draft_cache_locs_page_size_1` 重命名为 `assign_draft_cache_locs_contiguous`, 同时更新函数内参数名。
3. 导入与调用点同步: 在 `eagle_info_v2.py` 中更新 import 语句, 将旧的 kernel 名替换为新名; 在 `eagle_worker_v2.py` 中同样更新 import。
4. 工具函数辅助重命名: 在 `spec_utils.py` 中将 `traverse_tree` 内部变量 `curr_token_id` 改为 `current_token`, 与代码库风格一致。
5. 测试文档更新: 在 `test_spec_eagle_stress.py` 中更新类 docstring, 将引用的函数名改为新名称 (`move_accept_tokens_to_target_kvcache`)。

关键文件:

- `python/sglang/srt/speculative/eagle_worker_v2.py` (模块 调度器; 类别 source; 类型 core-logic; 符号 `_finalize_accepted_tree_path`, `_finalize_accept_tree_path`, `move_accepted_tokens_to_target_kvcache`, `move_accept_tokens_to_target_kvcache`) : 核心 Worker 文件, 包含三个关键方法的重命名 (`_finalize_accepted_tree_path` → `_finalize_accept_tree_path`, `move_accepted_tokens_to_target_kvcache` →

move_accept_tokens_to_target_kvcache, _compact_accepted_to_front
→_compact_accept_to_front), 以及内部所有调用点和 import 的同步。

- python/sglang/srt/speculative/spec_utils.py (模块 工具库; 类别 source; 类型 core-logic) : 在 traverse_tree 函数中将局部变量 curr_token_id 重命名为 current_token, 统一命名风格。
- python/sglang/srt/speculative/eagle_info_v2.py (模块 调度器; 类别 source; 类型 core-logic) : 更新 import 语句以匹配新 kernel 名称 (fill_accepted_out_cache_loc、assign_draft_cache_locs_contiguous) 。
- python/sglang/srt/speculative/triton_ops/eagle.py (模块 Kernel; 类别 infra; 类型 infrastructure; 符号 fill_accepted_out_cache_loc, fill_accept_out_cache_loc) : Triton kernel 函数名重命名 fill_accepted_out_cache_loc → fill_accept_out_cache_loc, 参数名同步更新。
- python/sglang/srt/speculative/triton_ops/cache_locs.py (模块 Kernel; 类别 infra; 类型 infrastructure; 符号 assign_draft_cache_locs_page_size_1, assign_draft_cache_locs_contiguous) : Triton kernel 函数名重命名 assign_draft_cache_locs_page_size_1 → assign_draft_cache_locs_contiguous, 反映实际功能 (连续分配, 非 page_size=1 专用) 。
- test/registered/spec/eagle/test_spec_eagle_stress.py (模块 测试; 类别 test; 类型 test-coverage) : 更新测试类 docstring 中引用的函数名 (move_accepted_tokens_to_target_kvcache → move_accept_tokens_to_target_kvcache), 保持测试文档准确。

关键符号: _finalize_accept_tree_path, move_accept_tokens_to_target_kvcache, _compact_accept_to_front, fill_accept_out_cache_loc, assign_draft_cache_locs_contiguous

关键源码片段

python/sglang/srt/speculative/triton_ops/eagle.py

Triton kernel 函数名重命名 fill_accepted_out_cache_loc → fill_accept_out_cache_loc, 参数名同步更新。

```
@triton.jit
def fill_accept_out_cache_loc(
    accept_index,
    out_cache_loc,
    accept_out_cache_loc, # 原参数名 accepted_out_cache_loc
    size_upper: tl.constexpr,
):
    """
    根据 accept_index 从 out_cache_loc 中收集被接受位置的 KV cache 索引。
    # 原函数名 fill_accepted_out_cache_loc
    """
    pid = tl.program_id(axis=0)
    src = tl.load(accept_index + pid)
```

```
if src > -1:
    value = tl.load(out_cache_loc + src)
    tl.store(accept_out_cache_loc + dst, value) # 原参数名 accepted_out_cache_loc
```

评论区精华

评论主要围绕 CI 触发：作者通过 `/rerun-test registered/spec/eagle/test_spec_eagle_stress.py` 指令重跑单测，CI 结果显示通过。没有出现架构或设计层面的讨论，说明重命名无争议。

- 暂无高价值评论线程

风险与影响

- 风险：该 PR 为纯重命名，不影响运行时行为，回归风险极低。风险主要集中在可能遗漏的调用点或外部引用（如动态符号查找），但经审查，所有 6 个文件中引用均同步更新，且 CI 单测通过。无性能、安全或兼容性风险。
- 影响：对用户透明，无功能行为变化。对开发团队而言，代码命名更加一致，降低了未来维护时的认知负担。影响范围局限于 speculative decoding V2 模块，不涉及其它子系统。
- 风险标记：无功能变更，纯重命名

关联脉络

- PR #27552 [Spec] Rename token resolver to `_resolve_spec_v2_tokens`; remove dead V1 helpers: 该 PR 直接堆叠在 #27552 之上，延续命名清理工作，移除 V1 辅助代码。
- PR #25464 [Spec] Deprecate Spec V1: 废弃 Spec V1 的系列工作之一，本 PR 进一步清理 V1 遗留下来的命名不一致。