

PR #25161 完整报告

sgl-project/sglang

[Refactor] Remove dead key_convert_fn / convert_to_bigram_key

合并时间: 2026-05-14 04:54

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/25161>

执行摘要

- 一句话: 移除死代码 `convert_to_bigram_key`
- 推荐动作: 可直接合并。该 PR 是常规清理, 但值得注意的是此类删除最好附带 `grep` 验证结果 (已包含)。可作为代码整洁的范例。

功能与动机

PR #23106 ("Make EAGLE bigram key an O(1) view on RadixKey") 已将 EAGLE 的 bigram key 转换逻辑迁移到 `RadixKey.maybe_to_bigram_view` 和构造参数 `is_bigram`, 但旧的 `self.key_convert_fn` 赋值和 `convert_to_bigram_key` 函数未被移除, 成为死代码。清理它们以消除代码膨胀和潜在混淆。

实现拆解

1. 删除辅助函数: 在 `python/sglang/srt/mem_cache/utils.py` 中删除 `convert_to_bigram_key` 函数 (10 行) 及其 docstring。该函数将 token id 列表转换为二元组列表, 已不再使用。
2. 清理 `SWARadixCache`: 在 `python/sglang/srt/mem_cache/swa_radix_cache.py` 中:
 - 移除 `from sglang.srt.mem_cache.utils import convert_to_bigram_key` 导入 (改为只导入 `split_node_hash_value`)。
 - 在 `__init__` 中删除根据 `self.is_eagle` 设置 `self.key_convert_fn` 的条件分支 (6 行)。
3. 清理 `UnifiedRadixCache`: 在 `python/sglang/srt/mem_cache/unified_radix_cache.py` 中:
 - 移除相同的导入。
 - 在 `__init__` 中删除相同的条件分支 (5 行)。
4. 验证: 提交者通过全局 `grep` 确认 `convert_to_bigram_key` 和 `key_convert_fn` 无任何引用。

关键文件:

- `python/sglang/srt/mem_cache/utils.py` (模块 缓存活; 类别 source; 类型 core-logic; 符号 `convert_to_bigram_key`): 删除核心辅助函数 `convert_to_bigram_key`, 10 行删除, 影响最大。
- `python/sglang/srt/mem_cache/swa_radix_cache.py` (模块 缓存活; 类别 source; 类型 dependency-wiring): 移除 `import` 和 `dead key_convert_fn` 赋值, +1/-6。

- python/sclang/srt/mem_cache/unified_radix_cache.py (模块 缓存活; 类别 source; 类型 dependency-wiring) : 移除 import 和 dead key_convert_fn 赋值, +0/-5。

关键符号: convert_to_bigram_key

关键源码片段

python/sclang/srt/mem_cache/utils.py

删除核心辅助函数 convert_to_bigram_key, 10 行删除, 影响最大。

```
# python/sclang/srt/mem_cache/utils.py
# 已删除的函数:
# def convert_to_bigram_key(tokens: List[int]) -> List[Tuple[int, int]]:
# """
# 将 token id 列表转换为二元组列表, 用于 EAGLE 的 bigram key。
# 例如 [1,2,3,4] -> [(1,2), (2,3), (3,4)]
# 现在已通过 RadixKey 的 O(1) view 实现, 此函数不再需要。
# """
# if len(tokens) and isinstance(tokens[0], tuple):
# return tokens
# if len(tokens) < 2:
# return []
# return [(tokens[i], tokens[i + 1]) for i in range(len(tokens) - 1)]
```

python/sclang/srt/mem_cache/swa_radix_cache.py

移除 import 和 dead key_convert_fn 赋值, +1/-6。

```
# python/sclang/srt/mem_cache/swa_radix_cache.py (head)
# 第 47 行: 导入只保留 split_node_hash_value
from sclang.srt.mem_cache.utils import split_node_hash_value

# __init__ 方法中删除了以下代码块 (第 356-359 行 base) :
# if self.is_eagle:
# self.key_convert_fn = convert_to_bigram_key
# else:
# self.key_convert_fn = lambda key: key
```

python/sclang/srt/mem_cache/unified_radix_cache.py

移除 import 和 dead key_convert_fn 赋值, +0/-5。

```
# python/sclang/srt/mem_cache/unified_radix_cache.py (head)
# 第 37 行: import 已删除
# from sclang.srt.mem_cache.utils import convert_to_bigram_key (已移除)

# __init__ 方法中删除了以下代码块 (第 228-231 行 base) :
# if self.is_eagle:
# self.key_convert_fn = convert_to_bigram_key
# else:
# self.key_convert_fn = lambda key: key
```

评论区精华

无实质性讨论。gemini-code-assist 机器人仅给出形式化评论，无反馈意见。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅删除死代码：
 1. 全局 grep 确认无任何调用点。
 2. 删除的函数和属性在功能上已被 RadixKey 的 $O(1)$ view 替代。
 3. 删减行数少（21 行），影响范围局限在三个文件内的导入和赋值。
 4. 无测试配套变更，因为被删代码已无可达路径。 - 影响：对用户无影响；对系统无运行时影响；对团队降低了维护负担，清理了遗留代码。 - 风险标记：暂无

关联脉络

- PR #23106 Make EAGLE bigram key an $O(1)$ view on RadixKey: 该 PR 是所有死代码产生的根源，迁移了 bigram 逻辑到 RadixKey，本 PR 是其后清理。