PR #26675 完整报告

sgl-project/sglang

[mem_cache][1/N] refactor: split allocator.py into allocator/ subpackage

合并时间：2026-05-31 20:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26675

执行摘要

- 一句话：将 allocator.py 拆分为 allocator/ 子包
- 推荐动作：该 PR 是模块拆分的最佳实践，适合关注代码组织和重构策略的开发者学习。它展示了如何在不破坏现有 API 的情况下逐步重构，并保留 Git 历史。推荐精读以了解子包拆分和向后兼容的导入模式。

功能与动机

PR 是 Issue #25371（父 Issue #24335）「allocator-promotion」系列的第一部分。目标是将 allocator.py 拆分为 allocator/ 子包，每个文件对应一种分配策略，以遵循父 Issue 的设计规则，并为后续将 SWATokenToKVPoolAllocator、HiSparse 分配器和 Mamba 分配器迁入同一包做好准备。

实现拆解

1. 通过 git mv 将 allocator.py 重命名为 allocator/paged.py，Git 自动检测为 R073 重命名，保留 PagedTokenToKVPoolAllocator 及其辅助函数的完整历史。
2. 新建 allocator/base.py，将 BaseTokenToKVPoolAllocator 抽象基类从原文件搬入此处，仅调整导入（删除 triton 等不必要依赖，添加 from sglang.srt.mem_cache.allocator.base import BaseTokenToKVPoolAllocator）。
3. 新建 allocator/token.py，将 TokenToKVPoolAllocator （page_size=1 的单页分配器）搬入，继承自 base.py 中基类。
4. 新建 allocator/__init__.py，重新导出所有公共类和辅助函数，确保 from sglang.srt.mem_cache.allocator import X 的旧导入路径继续生效。同时更新 paged.py 的导入以引用 base.py。
5. 验证：运行相关单元测试（934 passed）和 gsm8k 精度 / 速度基准，结果与 base 一致，确认无行为变化。

关键文件：

- python/sglang/srt/mem_cache/allocator/paged.py（模块 内存分配器；类别 source；类型 rename-or-move；符号 PagedTokenToKVPoolAllocator, alloc_extend_naive, alloc_extend_kernel, alloc_decode_kernel）：核心来源文件，通过 git mv 从原 allocator.py 重命名得来，保留了 PagedTokenToKVPoolAllocator 及其辅助函数的完整历史。

- python/sglang/srt/mem_cache/allocator/base.py（模块 内存分配器；类别 source；类型 dependency-wiring；符号 BaseTokenToKVPoolAllocator）：新建立的抽象基类文件，定义了所有分配器的公共接口。
- python/sglang/srt/mem_cache/allocator/token.py（模块 内存分配器；类别 source；类型 dependency-wiring；符号 TokenToKVPoolAllocator）：新建立的单页分配器文件，管理 page_size=1 的 KV 缓存槽位。
- python/sglang/srt/mem_cache/allocator/__init__.py（模块 内存分配器；类别 source；类型 dependency-wiring）：关键子包入口，重新导出所有公共符号，确保向后兼容。

关键符号：BaseTokenToKVPoolAllocator, TokenToKVPoolAllocator, PagedTokenToKVPoolAllocator, alloc_extend_naive, alloc_extend_kernel, alloc_decode_kernel

关键源码片段

python/sglang/srt/mem_cache/allocator/paged.py

核心来源文件，通过 git mv 从原 allocator.py 重命名得来，保留了 PagedTokenToKVPoolAllocator 及其辅助函数的完整历史。

```
"""Page-aligned memory pool."""

from __future__ import annotations

from typing import TYPE_CHECKING

import torch
import triton
import triton.language as tl

# 导入新的基类，而非在同文件中定义
from sglang.srt.mem_cache.allocator.base import BaseTokenToKVPoolAllocator
from sglang.srt.utils import get_bool_env_var, get_num_new_pages, next_power_of_2

if TYPE_CHECKING:
    from sglang.srt.mem_cache.memory_pool import KVCache


# 以下辅助函数和类在原文件中原封不动保留
def alloc_extend_naive(prefix_lens, seq_lens, last_loc, free_pages, out_indices, page_size,
device):
    """..."""
    # ... 实现略


class PagedTokenToKVPoolAllocator(BaseTokenToKVPoolAllocator):
    """Page-size >= 1 的分页分配器。"""
    # ... 实现略
```

python/sglang/srt/mem_cache/allocator/base.py

新建立的抽象基类文件，定义了所有分配器的公共接口。

```
"""Token-to-KV-slot 分配器的抽象基类。"""

from __future__ import annotations

import abc
from typing import TYPE_CHECKING

import torch

if TYPE_CHECKING:
    from sglang.srt.mem_cache.memory_pool import KVCache


class BaseTokenToKVPoolAllocator(abc.ABC):
    @abc.abstractmethod
    def __init__(self, size: int, page_size: int, dtype: torch.dtype, device: str, kvcache: KVCache,
    need_sort: bool):
        self.size = size
        self.page_size = page_size
        # ... 省略其他属性

    @property
    def size_full(self):
        return self.size

    def available_size(self):
        return (len(self.free_pages) + len(self.release_pages)) * self.page_size

    # ... 其他方法
```

评论区精华

Review 中仅有一个讨论线程：ispobock 建议调整 base.py 中 from __future__import annotations 与注释的顺序，作者已调整。无其他争议。

- 代码风格：import 与注释的顺序 (style): 已调整 import 顺序，将 from __future__import annotations 放在模块文档字符串之后。

风险与影响

- 风险：变更风险低，因为这是纯机械代码搬迁，类体完全未修改。但需注意： 1) 若未来有人误修改了子包文件且未保持与原始文件同步，可能引入不一致； 2) git mv 检测到的重命名保留了 paged.py 的历史，但 base.py 和 token.py 需要 -C 选项追溯历史； 3) 如果存在未发现的隐式导入（如动态导入路径），可能中断。但作者通过 __init__.py 的 re-export 和 grep 验证了所有外部导入，风险已控制。

- 影响：对用户：无影响，API 完全向后兼容。对开发者：代码结构更清晰，每个文件职责单一，降低认知负担；新增分配器只需在子包中添加文件并更新 __init__.py。对系统：无性能影响，因为纯导入变更未改变运行时行为。
- 风险标记：纯机械搬迁无逻辑变化，向后兼容导入，git blame 需 -C 选项用于新文件

关联脉络

- 暂无明显关联 PR