PR #25880 完整报告

sgl-project/sglang

Update MooncakeStore batch tests to use v1 APIs

合并时间：2026-05-29 15:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25880

执行摘要

- 一句话：更新 MooncakeStore 批处理测试以使用 v1 API
- 推荐动作：建议 MooncakeStore 相关开发者阅读，了解 v1 批处理 API 的正确用法和测试模式，可作为后续类似测试的参考。

功能与动机

来自 PR body："The existing MooncakeStore batch test was written against an older interface and no longer matched the current HiCache storage workflow. This PR refreshes the test so it covers the current MooncakeStore batch API behavior for both MHA and MLA configurations."

实现拆解

1. 重构辅助函数：新增 make_hicache_storage_config 工厂函数，统一构造 HiCacheStorageConfig；简化 generate_batch_query_keys，不再拼接 key 后缀，直接返回 UUID 列表。
2. 更新模拟 HostKVCache：create_mock_host_kv_cache 新增 entries_per_page 和 page_elements 参数，MockHostKVCache.get_page_buffer_meta 按分页逐条计算指针和大小；新增 get_ksize_per_token 方法。
3. 更新单操作测试：test_single_operation 改用新配置工厂，构造非 MLA 配置，并通过新模拟缓存注册。
4. 更新批操作测试：test_batch_operation 根据配置的 is_mla_model 设置 entries_per_page（MLA 为 1，非 MLA 为 2），直接使用 v1 版本 batch_set / batch_get 传递无后缀 key，并通过 batch_exists 校验存在性。

关键文件：

- python/sglang/srt/mem_cache/storage/mooncake_store/test_mooncake_store.py（模块 Mooncake 存储；类别 test；类型 test-coverage；符号 generate_batch_query_keys, create_mock_host_kv_cache, make_hicache_storage_config, init）：本 PR 唯一修改的文件，更新了所有测试用例以使用 v1 batch API 并支持 MHA/MLA 配置。

关键符号：make_hicache_storage_config, generate_batch_query_keys, create_mock_host_kv_cache, get_ksize_per_token, test_single_operation, test_batch_operation

关键源码片段

python/sglang/srt/mem_cache/storage/mooncake_store/test_mooncake_store.py

本 PR 唯一修改的文件，更新了所有测试用例以使用 v1 batch API 并支持 MHA/MLA 配置。

```
# 构造测试用的 HiCacheStorageConfig，方便设置 MHA / MLA 及 TP 配置
def make_hicache_storage_config(
    *,
    is_mla_model: bool,
    tp_rank: int,
    tp_size: int,
) -> HiCacheStorageConfig:
    return HiCacheStorageConfig(
        tp_rank=tp_rank,
        tp_size=tp_size,
        pp_rank=0,
        pp_size=1,
        attn_cp_rank=0,
        attn_cp_size=1,
        is_mla_model=is_mla_model,
        enable_storage_metrics=False,
        is_page_first_layout=True,
        model_name=None,
    )


# 创建模拟的 HostKVCache，支持条目分页配置
def create_mock_host_kv_cache(
    buffer_size,
    entries_per_page=2, # 每页 KV 条目数，MHA=2, MLA=1
    page_elements=1,
    dtype=torch.float32,
):
    buffer = torch.randn(buffer_size, dtype=dtype)

    class MockHostKVCache:
        def __init__(self, buffer, entries_per_page, page_elements):
            self.kv_buffer = buffer
            self.layout = "page_first"
            self.page_size = 1
            self.entries_per_page = entries_per_page
            self.page_elements = page_elements

        def get_page_buffer_meta(self, indices):
            # 根据分页索引生成指针和大小列表，每页包含 entries_per_page 个条目
            ptr_list = []
            element_size_list = []
```

```
            for idx in indices:
                page_idx = int(idx)
                page_offset = page_idx * self.entries_per_page * self.page_elements
                for entry_idx in range(self.entries_per_page):
                    offset = page_offset + entry_idx * self.page_elements
                    ptr_list.append(self.kv_buffer[offset:].data_ptr())
                    element_size_list.append(
                        self.page_elements * self.kv_buffer.element_size()
                    )
            return ptr_list, element_size_list

        def get_ksize_per_token(self):
            # 返回每个 token 的 KV 缓存大小（字节）
            return (
                self.entries_per_page
                * self.page_elements
                * self.kv_buffer.element_size()
            )

    return MockHostKVCache(buffer, entries_per_page, page_elements), buffer

def test_batch_operation(config: HiCacheStorageConfig):
    """使用 v1 batch API 测试多 KV 的 set/get/exists。"""
    buffer_size = 1024 * 1024 * 16 # 16MB
    value_elements = 256
    kv_num = 13
    # 根据模型类型设置每页条目数：MLA=1，其他 =2
    entries_per_page = 1 if config.is_mla_model else 2
    store = MooncakeStore(config)
    mock_host_kv_cache, buffer = create_mock_host_kv_cache(
        buffer_size,
        entries_per_page=entries_per_page,
        page_elements=value_elements,
    )
    store.register_mem_pool_host(mock_host_kv_cache)

    keys = generate_batch_query_keys(kv_num) # 直接生成 key，无需后缀
    set_slices = [
        buffer[i * value_elements : (i + 1) * value_elements]
        for i in range(len(keys))
    ]
    set_locations = [s.data_ptr() for s in set_slices]
    target_sizes = [value_elements * buffer.element_size()] * len(keys)

    # v1 batch_set 直接使用 key 列表
    assert store.batch_set(keys, target_locations=set_locations, target_sizes=target_sizes)
    # v1 batch_exists 同样直接使用 key 列表
    assert store.batch_exists(keys)
    # v1 batch_get 返回成功键数
```

```
get_locations = [buffer[(len(keys) + i) * value_elements:].data_ptr() for i in range(len(keys))]
assert store.batch_get(keys, target_locations=get_locations, target_sizes=target_sizes) == kv_
num
```

评论区精华

此 PR 没有实质性的 reviewer 讨论。机器人 gemini-code-assist[bot] 自动评论无具体反馈，随后由 huangtingwei9988 直接批准。

- 自动化代码审查 (other): 无实际讨论。

风险与影响

- 风险：此变更仅涉及测试文件，未修改任何生产代码，风险极低。主要风险是测试用例可能无法完全覆盖真实工作负载的边缘情况，但 mock 层已按分页模式重构，足以验证 v1 API 的基本流程。
- 影响：影响范围限定在 mooncake_store 测试模块，提升了测试与当前 HiCache 存储工作流的匹配度，有助于 CI 对 MooncakeStore 批处理功能的正确性验证。不影响任何生产逻辑或其他模块。
- 风险标记：测试更新，无生产代码变更，API 适配，mock 重构

关联脉络

- PR #25959 Ensure multi-node MM embedding cache consistency in insert_batch: 同样涉及 mooncake_store 的嵌入缓存一致性修复，与当前 PR 属于同一模块。
- PR #22587 [EPD] Optimize the Mooncake backend: 对 Mooncake 后端进行性能优化，涉及存储层，与本测试更新相关联。