PR #26487 完整报告

sgl-project/sglang

feat: convert mm_hashes to str in encode_server for Mooncake key compat

合并时间：2026-05-28 14:16

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26487

执行摘要

- 一句话：修复 hash 类型不匹配导致 Mooncake 缓存失效
- 推荐动作：该 PR 是典型的接口类型不匹配 bugfix，逻辑简单直接，适合快速合入。建议阅读以了解多模态缓存与 Mooncake 的集成点。

功能与动机

PR body 明确指出：EmbeddingCacheController 的 batch_is_exist、prefetch、get_embeddings、insert_batch 方法中 image_hashes 参数均声明为 List[str]，但实际 image_hashes 由 data_hash 或 tensor_hash 计算得出，类型为 int。这导致 L2 缓存找不到有效 key，缓存命中率下降。

实现拆解

仅修改 python/sglang/srt/disaggregation/encode_server.py 一个文件，分两步：

1. 修正返回类型声明：将 _calculate_hashes_from_features 的返回类型标注从 List[str] 改为 List[int]，与实现一致。
2. 统一转换并传递 str：在 encode_with_global_cache 中，将 mm_hashes（List[int]）通过 [str(h) for h in mm_hashes] 转换为 str_mm_hashes，然后所有调用缓存方法的地方——batch_is_exist、prefetch、get_embeddings、insert_batch——全部改用 str_mm_hashes 而非原始的 mm_hashes。 无测试、配置或部署配套改动。

关键文件：

- python/sglang/srt/disaggregation/encode_server.py（模块 缓存层；类别 source；类型 core-logic；符号 _calculate_hashes_from_features, encode_with_global_cache）：本次变更的唯一文件，包含了所有改动：类型声明修正和调用处统一转换为 str。

关键符号：_calculate_hashes_from_features, encode_with_global_cache

关键源码片段

python/sglang/srt/disaggregation/encode_server.py

本次变更的唯一文件，包含了所有改动：类型声明修正和调用处统一转换为 str。

```
# From _calculate_hashes_from_features (line 682-703)
def_calculate_hashes_from_features(     self, mm_feature, grid_thw, modality ) ->
List[int]: # 修正类型：之前错误标注为 List[str] """CPU Task: Compute hashes based on
processed feature patches."""     hashes = []     if modality == Modality.AUDIO and
isinstance(mm_feature, list):         for feature in mm_feature:             tmp_item =
MultimodalDataItem(modality=modality, feature=feature)
tmp_item.set_pad_value()             hashes.append(tmp_item.hash) # tmp_item.hash 是
int         return hashes     offset = 0     for grid in grid_thw:         num_patches =
self.get_num_patches(grid, modality)         feature_slice = mm_feature[offset : offset +
num_patches]         tmp_item = MultimodalDataItem(modality=modality,
feature=feature_slice)         tmp_item.set_pad_value()
hashes.append(tmp_item.hash) # int         offset += num_patches     return hashes #
Inside encode_with_global_cache (line 798 onward) # 转换 int hash 为 str hash，确保与
EmbeddingCacheController 接口兼容 str_mm_hashes = [str(h) for h in mm_hashes]
exist_mask = await self.mm_global_cache.batch_is_exist(str_mm_hashes) # ... 后续所有
使用 hash 的地方都用 str_mm_hashes # 例如 prefetch、get_embeddings、insert_batch
等
```

评论区精华

讨论非常简短：审核者 ShangmingCai 指出了类型声明错误（should return -> List[int]），作者 QiuMike 回复“fixed”。此外 gemini-code-assist 的自动 review 无实质反馈。最终 ShangmingCai 和 liusy58 均 approve，未遗留未解决的疑虑。

- 类型声明错误 (correctness): 作者 QiuMike 已修复，返回类型修正为 List[int]。

风险与影响

- 风险：风险极低：变更集中在单个函数的内部逻辑，不涉及公共 API 或对外接口。唯一需要关注的是所有 mm_hashes 的使用点均已正确替换为 str_mm_hashes，经过 patch 验证已覆盖。
- 影响：直接影响 Mooncake 多模态缓存路径的 cache key 匹配。修复后 L2 缓存能正确识别 hash，提升缓存命中率，减少不必要的 ViT 重计算。对不启用 Mooncake 或 L2 缓存的场景无影响。影响范围仅限于 encode_server.py 中调用全局缓存的逻辑。
- 风险标记：低风险

关联脉络

- 暂无明显关联 PR