

PR #32977 完整报告

sgl-project/sglang

fix: preserve priority for batched embedding requests

合并时间: 2026-08-07 10:26

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32977>

执行摘要

- 一句话: 修复批量 embedding 请求拆分时丢失 priority
- 推荐动作: 值得快速精读: 可作为“沿数据链路定位单点丢失”的范例, 测试设计针对关键字参数易漏传的回归类型。建议后续在 `__getitem__` 的构造调用上考虑集中封装字段透传或加入基于 dataclass 的字段完整性校验, 避免同类问题再发。

功能与动机

Issue #32844 指出: `EmbeddingReqInput.__getitem__` 拆分批请求时遗漏 `priority`, 而 `_create_tokenized_object` 与 `handle_embedding_request` 均正确传递 / 读取, 值只在 batch split 处丢失; 同时 `_set_default_priority` 写入的默认配置也被一并丢弃。PR body 说明 "GenerateReqInput.__getitem__ already forwards priority; this makes the embedding path match", 即让 embedding 路径与生成路径的优先级语义对齐。

实现拆解

1. 定位单点丢失: 在 `python/sglang/srt/managers/io_struct.py` 的 `EmbeddingReqInput.__getitem__` 中确认两条分支 (`is_cross_encoder_request=True` 与普通分支) 构造子 `EmbeddingReqInput` 时均未传 `priority`; `_create_tokenized_object` 与 `handle_embedding_request` 已正确使用该字段, 因此修改点唯一。
2. 补字段透传: 在两个分支的构造调用中各加一个关键字参数 `priority=self.priority`, 共 2 行。由于 `self.priority` 在 `normalize_batch_and_arguments()` 中已被 `_set_default_priority` 填好默认值, 显式配置与默认配置都会正确传递。
3. 新增回归测试: 在 `test/registered/unit/managers/test_io_struct.py` 中新增 `TestEmbeddingReqInputGetItem.test_priority_is_preserved`, 同时覆盖普通批量 (期望 `[7, 7]`) 与 cross-encoder 批量 (期望 `[3, 3]`); 该测试文件已有 CUDA/AMD/CPU 的 CI 注册标记, 无需新增注册。Revert 修复后运行新测试即可复现 `[None, None] != [7, 7]` 的失败。
4. 验证与合入: rebase main 后定向重跑 `test_io_struct.py` 通过 (1-gpu-h100), pre-commit 通过, 维护者 APPROVE 后合并。

关键文件:

- `python/sglang/srt/managers/io_struct.py` (模块 请求结构; 类别 source; 类型 core-logic; 符号 `EmbeddingReqInput.getitem`): 修复主路径: 在

EmbeddingReqInput.__getitem__两个分支补 priority 透传，是丢失字段的唯一修正点。

- test/registered/unit/managers/test_io_struct.py (模块 回归测试; 类别 test; 类型 test-coverage; 符号 TestEmbeddingReqInputGetItem, test_priority_is_preserved) : 新增 TestEmbeddingReqInputGetItem 回归测试, 覆盖两个 __getitem__分支, 并复用现有 CI 注册。

关键符号: EmbeddingReqInput.getitem, test_priority_is_preserved

关键源码片段

python/sglang/srt/managers/io_struct.py

修复主路径: 在 EmbeddingReqInput.__getitem__两个分支补 priority 透传，是丢失字段的唯一修正点。

```
# python/sglang/srt/managers/io_struct.py
```

```
# 批量 embedding 请求拆分: 为每个子请求构造独立的 EmbeddingReqInput
```

```
def __getitem__(self, i):
    # 缓存子对象, 确保重复的 obj[i] 调用返回同一实例
    cache = self.__dict__.setdefault("_sub_obj_cache", {})
    if i in cache:
        return cache[i]

    if self.is_cross_encoder_request:
        # cross-encoder 分支: 每对 query/doc 拆成第 i 个子请求
        sub = EmbeddingReqInput(
            rid=self.rid[i],
            text=[self.text[i]] if self.text is not None else None,
            sampling_params=self.sampling_params[i],
            is_cross_encoder_request=True,
            lora_path=self.lora_path[i] if self.lora_path is not None else None,
            lora_id=self.lora_id[i] if self.lora_id is not None else None,
            positional_embed_overrides=self._get_positional_embed_overrides_item(i),
            http_worker_ipc=self.http_worker_ipc,
            # 修复点: 此前漏掉该参数, 导致批内子请求 priority 全部为 None,
            # 开启 --enable-priority-scheduling 时 embedding 批请求被当作无优先级调度
            priority=self.priority,
            return_pooled_hidden_states=self.return_pooled_hidden_states,
            return_prompt_token_ids=self.return_prompt_token_ids,
            multi_item_delimiter_indices=(
                self.multi_item_delimiter_indices[i]
                if self.multi_item_delimiter_indices is not None
                else None
            ),
        )
    else:
        # 普通 embedding 分支: 同样透传 priority
        # (self.priority 的默认值已由 _set_default_priority 在 normalize 阶段写入)
```

```

sub = EmbeddingReqInput(
    rid=self.rid[i],
    text=self.text[i] if self.text is not None else None,
    input_ids=self.input_ids[i] if self.input_ids is not None else None,
    image_data=self.image_data[i] if self.image_data is not None else None,
    video_data=self.video_data[i] if self.video_data is not None else None,
    audio_data=self.audio_data[i] if self.audio_data is not None else None,
    embed_override_token_id=self.embed_override_token_id,
    embed_overrides=(
        self.embed_overrides[i] if self.embed_overrides is not None else None
    ),
    sampling_params=self.sampling_params[i],
    lora_path=self.lora_path[i] if self.lora_path is not None else None,
    lora_id=self.lora_id[i] if self.lora_id is not None else None,
    positional_embed_overrides=self._get_positional_embed_overrides_item(i),
    http_worker_ipc=self.http_worker_ipc,
    # 修复点：普通分支补上同样的透传
    priority=self.priority,
    dimensions=self.dimensions,
    return_pooled_hidden_states=self.return_pooled_hidden_states,
    return_prompt_token_ids=self.return_prompt_token_ids,
    external_trace_header=self.external_trace_header,
    received_time=self.received_time,
    multi_item_delimiter_indices=(
        self.multi_item_delimiter_indices[i]
        if self.multi_item_delimiter_indices is not None
        else None
    ),
)
cache[i] = sub
return sub

```

test/registered/unit/managers/test_io_struct.py

新增 TestEmbeddingReqInputGetItem 回归测试，覆盖两个 __getitem__ 分支，并复用现有 CI 注册。

test/registered/unit/managers/test_io_struct.py

回归测试：priority 必须在批处理拆分后保留，普通与 cross-encoder 两个分支都要覆盖

```

class TestEmbeddingReqInputGetItem(CustomTestCase):
    """验证 EmbeddingReqInput.__getitem__ 的批处理拆分为行。"""

    def test_priority_is_preserved(self):
        # 普通 embedding 分支：批内每个子请求都应保留优先级 7
        req = EmbeddingReqInput(text=["Hello", "World"], priority=7)
        req.normalize_batch_and_arguments()
        self.assertEqual([req[0].priority, req[1].priority], [7, 7])

        # cross-encoder 分支：query/doc 对拆出的子请求优先级应为 3

```

```
cross_encoder_req = EmbeddingReqInput(
    text=[["query 1", "doc 1"], ["query 2", "doc 2"]],
    is_cross_encoder_request=True,
    priority=3,
)
cross_encoder_req.normalize_batch_and_arguments()
self.assertEqual(
    [cross_encoder_req[0].priority, cross_encoder_req[1].priority], [3, 3]
)
```

评论区精华

评论互动集中在 CI 流程，没有技术方案上的分歧：维护者 hnyls2002 发起 [/rerun-test test_io_struct.py](#)；github-actions 机器人提示分支相对 required base commit [cdf33d](#) 已 diverged，要求先 rebase main；合并 main（commit [f170ec2](#)）后定向重跑在 1-gpu-h100 通过，随后触发 [/tag-and-rerun-ci](#) 完整 CI，最终 APPROVE。

- CI 重跑与 rebase 要求 (other): 通过 rebase 解决分歧后重跑通过，无技术争议，最终 APPROVE。

风险与影响

- 风险：风险极低：改动只在 EmbeddingReqInput.__getitem__ 两个分支的构造调用中各增加一个关键字参数，不触碰字段解析、多模态数据或 kernel 路径。行为变化是修复前 priority 静默丢失导致批量 embedding 被错误地按无优先级排队；修复后显式优先级与 _set_default_priority 的默认值都会正确生效。回归防线：priority 只是约 20 个参数构造调用里的一个关键字参数，删掉它不会报错，只能靠测试兜底；新测试覆盖两个分支且测试文件已有 CUDA/AMD/CPU CI 注册，能持续回归。无性能、安全与兼容性影响。
- 影响：用户侧：使用批量 embedding（含 cross-encoder）并开启 --enable-priority-scheduling 的部署，批内请求的优先级恢复生效，队列调度按预期。系统侧：仅影响请求入口数据结构，无协议 /API 变更。团队侧：回归测试已接入既有 CI 注册，后续改动可持续被守护。
- 风险标记：请求入口结构改动，调度优先级语义修复，回归测试守护

关联脉络

- 暂无明显关联 PR