PR #27205 完整报告

sgl-project/sglang

Fix customized_info incremental streaming

合并时间：2026-06-05 21:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/27205

执行摘要

- 一句话：修复 incremental streaming 下 customized_info 切分与累积逻辑
- 推荐动作：值得精读。该 PR 展示了如何将动态键纳入现有流式框架，以及通过继承 Engine 进行集成测试的模式。对于需要自定义采样器并依赖流式正确性的开发者有直接参考价值。

功能与动机

Custom sampler backends can attach per-token metadata through customized_info. With incremental streaming enabled, that metadata should follow the same chunking semantics as output_ids: streaming chunks should contain only newly emitted values, while non-streaming responses should contain the accumulated final sequence.

实现拆解

实现分为四步：

1. 在 python/sglang/srt/managers/tokenizer_manager.py 的 StreamingOutput 数据类中添加 customized_info_accumulated 字段，用于累积每个请求的定制信息。
2. 修改 _slice_streaming_output_meta_info 函数，增加 customized_info_keys 参数，将定制信息键纳入切片集合，确保流式块中只包含新产生的数据。
3. 修改 _coalesce_streaming_chunks 函数，接受 customized_info_keys 参数，在合并批量通知时同时合并定制信息键的值。
4. 在 _wait_one_response 中传入 customized_info_accumulated.keys() 给合并函数，在 _handle_batch_output 中将调度器返回的定制信息累积到状态中，并在 meta_info 中设置当前累积值。
5. 新增测试文件 test/registered/unit/managers/test_customized_info_streaming.py，包含自定义采样器、引擎子类和多个测试场景。

关键文件：

- test/registered/unit/managers/test_customized_info_streaming.py（模块 流式测试；类别 test；类型 test-coverage；符号 CustomizedInfoSampler, forward, install_customized_info_sampler, run_scheduler_process_with_customized_info_sampler）：新增集成测试，验证 customized_info 在流式、非流式和 stream_interval 场景下与

output_ids 的行为一致性

- python/sglang/srt/managers/tokenizer_manager.py（模块 流式输出；类别 source；类型 core-logic；符号 customized_info_accumulated, _slice_streaming_output_meta_info, _coalesce_streaming_chunks, _wait_one_response）：核心修复文件，修改流式输出切片与合并逻辑，添加累积字段支持动态 customized_info 键

关键符号：_slice_streaming_output_meta_info, _coalesce_streaming_chunks, CustomizedInfoSampler.forward, install_customized_info_sampler, run_scheduler_process_with_customized_info_sampler

关键源码片段

test/registered/unit/managers/test_customized_info_streaming.py

新增集成测试，验证 customized_info 在流式、非流式和 stream_interval 场景下与 output_ids 的行为一致性

```
# test/registered/unit/managers/test_customized_info_streaming.py
# 自定义采样器：将每个 step 的 batch_next_token_ids 镜像到 customized_info
class CustomizedInfoSampler(Sampler):
    def forward(
        self,
        logits_output: "LogitsProcessorOutput",
        sampling_info: "SamplingBatchInfo",
        return_logprob: bool,
        top_logprobs_nums: List[int],
        token_ids_logprobs: List[List[int]],
        positions: torch.Tensor,
    ) -> torch.Tensor:
        # 先通过父类获取采样 token id
        batch_next_token_ids = super().forward(
            logits_output,
            sampling_info,
            return_logprob,
            top_logprobs_nums,
            token_ids_logprobs,
            positions,
        )
        # 初始化 customized_info dict
        if logits_output.customized_info is None:
            logits_output.customized_info = {}
        # 将 token id list 存入自定义字段
        logits_output.customized_info[CUSTOMIZED_INFO_FIELD] = (
            batch_next_token_ids.detach().cpu().tolist()
        )
        return batch_next_token_ids

def install_customized_info_sampler() -> None:
    """在 sampler 构造前注册自定义后端。"""
```

```
    register_sampler_backend(CUSTOMIZED_INFO_SAMPLER_BACKEND,
    CustomizedInfoSampler)
```

python/sglang/srt/managers/tokenizer_manager.py

核心修复文件，修改流式输出切片与合并逻辑，添加累积字段支持动态 customized_info 键

```
# python/sglang/srt/managers/tokenizer_manager.py
# 修改 _slice_streaming_output_meta_info 支持动态定制键

def _slice_streaming_output_meta_info(
    meta_info: Dict[Any, Any],
    last_output_offset: int,
    customized_info_keys: Optional[Iterable[str]] = None,
) -> None:
    """按 last_output_offset 对元数据进行切片。"""
    # 合并内置键与定制键
    streaming_meta_info_keys = set(_INCREMENTAL_STREAMING_META_INFO_KEYS)
    if customized_info_keys is not None:
        streaming_meta_info_keys.update(customized_info_keys)
    # 切片操作
    for key in meta_info.keys() & streaming_meta_info_keys:
        meta_info[key] = meta_info[key][last_output_offset:]


# 修改 _coalesce_streaming_chunks 合并定制信息

def _coalesce_streaming_chunks(
    self,
    out_list: list,
    rid: str,
    customized_info_keys: Optional[Iterable[str]] = None,
) -> dict:
    """合并多条流式块为一条，并正确拼接定制信息列表。"""
    if "meta_info" in out:
        meta_info_list = [chunk["meta_info"] for chunk in out_list]
        meta_info = dict(meta_info_list[-1])
        incremental_streaming_keys = set(_INCREMENTAL_STREAMING_META_INFO_KEYS)
        if customized_info_keys is not None:
            incremental_streaming_keys.update(customized_info_keys)
        for key in incremental_streaming_keys:
            if any(key in m for m in meta_info_list):
                meta_info[key] = [
                    item
                    for m in meta_info_list
                    for item in m.get(key, [])
                ]
        out["meta_info"] = meta_info
    return out
```

评论区精华

Reviewer ispobock 建议将测试文件移动到 test/registered/unit/managers 目录，作者随后执行了移动（commit f692b4c5b3a5）。此外无其他有意义的讨论。

- 测试文件位置 (question): 作者执行移动并提交

风险与影响

- 风险：核心风险在于 tokenizer_manager.py 的流式输出切分与合并逻辑，如果 customized_info_keys 处理不当，可能导致流式块中数据错乱或非流式响应中丢失累积数据。但已有集成测试覆盖流式、非流式和 stream_interval 场景，风险可控。
- 影响：影响范围：使用自定义采样器后端并通过 customized_info 传递每 token 元数据的用户。影响程度：修复了 incremental streaming 下的正确性问题，使 customized_info 行为与 output_ids 一致，属于重要 bugfix，但 API 无变化，兼容性良好。
- 风险标记：增量流式关键路径 , 动态键偶发性问题

关联脉络

- 暂无明显关联 PR