

PR #31960 完整报告

sgl-project/sglang

[feat] Optional base64 encoding for the flat prompt top logprob arrays

合并时间: 2026-07-30 03:15

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/31960>

执行摘要

- 一句话: 可选 base64 编码 prompt top logprob flat 数组, 加速 4 倍
- 推荐动作: 值得精读。实现干净, 设计上预见了未来 dtype 扩展 (通过 `_b64_dtype` 标记字段), 且与 #22244 的独立输出侧 base64 互不干扰。可作为自定义序列化格式优化的参考。

功能与动机

PR #32078 已实现 flat raw top logprobs 格式, 但 JSON number 的浮点序列化 / 反序列化仍是主要瓶颈。PR 作者给出的微基准显示: 32768 位置、k=2 时, flat lists 的服务器 assembly + json.dumps 需 35.4 ms, 客户端 json.loads 需 19.7 ms, round trip 共 55.1 ms。Base64 二进制编码可将两者替换为 memcpy, round trip 降至 13.2 ms, payload 从 1,827,898 字节降至 699,289 字节。

实现拆解

1. 新增请求标志 (python/sglang/srt/managers/io_struct.py): 在 GenerateReqInput 中增加 `return_flat_raw_top_logprobs_b64: bool = False`, 并在 `_validate_inputs` 中添加校验: 该标志必须与 `return_flat_raw_top_logprobs` 同时使用。
2. 核心编码逻辑 (python/sglang/srt/managers/tokenizer_manager.py): 修改 `_build_flat_input_top_logprobs_fields` 函数, 新增可选的 `return_b64` 参数。当为 True 时, 将 val 和 idx 的二维数组 (省略 null prefix 后) 转换为 `np.float32 / np.int32` 的 flat ndarray, 通过 `ndarray.tobytes()` 获取连续二进制, 再用 `pybase64.b64encode` 编码为 base64 字符串, 并添加 `_b64_dtype` 字段标记类型。
3. 调用处适配 (tokenizer_manager.py): 在 `add_logprob_to_meta_info` 中调用 `_build_flat_input_top_logprobs_fields` 时, 传递 `return_b64=state.obj.return_flat_raw_top_logprobs_b64`。
4. 标志传播 (io_struct.py): `GenerateReqInput.__getitem__` 中将 `return_flat_raw_top_logprobs_b64` 传递给 batch item。
5. 测试覆盖 (test/registered/unit/managers/test_flat_raw_top_logprobs.py): 新增 5 个测试用例: `test_b64_requires_flat_flag` (校验二者必须同时开启)、`test_b64_flag_propagates_to_batch_items` (batch 传播)、`test_b64_roundtrip` (b64 解码后与原始数组一致)、`TestB64MetaInfo` (meta_info 中 b64 键存在且 flat 键不存在, 并验证 chunked 与 one-shot 等价)。

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 GenerateReqInput.return_flat_raw_top_logprobs_b64, GenerateReqInput._validate_inputs, GenerateReqInput.getitem) : 新增请求标志 return_flat_raw_top_logprobs_b64 及验证逻辑
- python/sglang/srt/managers/tokenizer_manager.py (模块 管理器; 类别 source; 类型 dependency-wiring; 符号 _build_flat_input_top_logprobs_fields, add_logprob_to_meta_info) : 实现 base64 编码的核心函数 _build_flat_input_top_logprobs_fields 修改
- test/registered/unit/managers/test_flat_raw_top_logprobs.py (模块 测试; 类别 test; 类型 test-coverage; 符号 test_b64_requires_flat_flag, test_b64_flag_propagates_to_batch_items, test_b64_roundtrip, TestB64MetaInfo) : 新增 5 个测试用例覆盖新标志

关键符号: _build_flat_input_top_logprobs_fields, add_logprob_to_meta_info, GenerateReqInput._validate_inputs, GenerateReqInput.getitem, test_b64_roundtrip, test_b64_requires_flat_flag

关键源码片段

python/sglang/srt/managers/io_struct.py

新增请求标志 return_flat_raw_top_logprobs_b64 及验证逻辑

```
class GenerateReqInput:
    # ... 其他字段 ...
    return_flat_raw_top_logprobs: bool = False
    # Base64-encode the flat arrays. Requires return_flat_raw_top_logprobs.
    return_flat_raw_top_logprobs_b64: bool = False

    def _validate_inputs(self):
        # ... 其他验证 ...
        # 新标志必 require flat 标志
        if (
            self.return_flat_raw_top_logprobs_b64
            and not self.return_flat_raw_top_logprobs
        ):
            raise ValueError(
                "return_flat_raw_top_logprobs_b64 requires return_flat_raw_top_logprobs."
            )

    def __getitem__(self, i):
        # ... 原有逻辑 ...
        return GenerateReqInput(
            # ... 其他字段 ...
            return_flat_raw_top_logprobs=self.return_flat_raw_top_logprobs,
            return_flat_raw_top_logprobs_b64=self.return_flat_raw_top_logprobs_b64,
        )
```

python/slang/srt/managers/tokenizer_manager.py

实现 base64 编码的核心函数 _build_flat_input_top_logprobs_fields 修改

```
import numpy as np

def _build_flat_input_top_logprobs_fields(
    input_top_logprobs_val: List[Optional[List[float]]],
    input_top_logprobs_idx: List[Optional[List[int]]],
    top_logprobs_num: int,
    return_b64: bool = False,
) -> Dict[str, Any]:
    # ... 计算 null_prefix 和 k ...
    val_rows = input_top_logprobs_val[null_prefix:]
    idx_rows = input_top_logprobs_idx[null_prefix:]

    fields: Dict[str, Any] = {}
    if return_b64:
        # 将二维列表转为连续 numpy 数组, 再做 base64
        val_arr = np.asarray(val_rows, dtype=np.float32)
        idx_arr = np.asarray(idx_rows, dtype=np.int32)
        fields["input_top_logprobs_val_flat_b64"] = pybase64.b64encode(
            val_arr.tobytes()
        ).decode("utf-8")
        fields["input_top_logprobs_idx_flat_b64"] = pybase64.b64encode(
            idx_arr.tobytes()
        ).decode("utf-8")
        fields["input_top_logprobs_val_flat_b64_dtype"] = "float32"
        fields["input_top_logprobs_idx_flat_b64_dtype"] = "int32"
    else:
        # 原有逻辑: 直接输出 Python list
        fields["input_top_logprobs_val_flat"] = [v for row in val_rows for v in row]
        fields["input_top_logprobs_idx_flat"] = [i for row in idx_rows for i in row]
    fields["input_top_logprobs_shape"] = [len(val_rows), k]
    fields["input_top_logprobs_null_prefix"] = null_prefix
    return fields

# 调用处
meta_info.update(
    _build_flat_input_top_logprobs_fields(
        state.input_top_logprobs_val,
        state.input_top_logprobs_idx,
        top_logprobs_num,
        return_b64=state.obj.return_flat_raw_top_logprobs_b64,
    )
)
```

评论区精华

Engine API 透传讨论: chatgpt-codex-connector 提出 `return_flat_raw_top_logprobs_b64` 未通过 Python 端 `Engine.generate`、`HttpServerEngine.generate` 等公共 API 暴露, 可能导致 in-process 用户无法使用。作者 sshleifer 回应这是有意为之, 遵循已有模式 (如 `return_sampling_mask` 也仅限 HTTP), 强调主要消费者是 HTTP 客户端; 并愿意根据 maintainer 意见扩展。该讨论未进一步发酵, PR 最终经 Qiaolin-Yu 批准合并。

- Engine API 透传新标志 (design): sshleifer 回应这是有意限定范围, 遵循已有模式 (如 `return_sampling_mask` 也仅 HTTP-native), 主要消费者是 HTTP 客户端; 并表示若 maintainer 偏好可扩展。

风险与影响

- 风险:
 1. 兼容性风险 (低): 新标志默认关闭, 不影响现有请求。但客户端若设置该标志, 需处理新增的 `_b64` 和 `_b64_dtype` 键, 旧客户端忽略即可。
 2. 数据精度风险 (低): 当前固定 float32/int32, 若未来缩小宽度 (如 fp16), dtype 标记可保证向前兼容。
 3. 性能风险 (低): 新增代码路径仅当标志开启时执行, 且 numpy 转换和 base64 编码本身比 JSON number 序列化快, 不会负优化。
 4. 依赖风险 (极低): pybase64 已是现有依赖, numpy 由 `import numpy as np` 新增导入, 但 numpy 已是隐式依赖, 无新增安装负担。
 - 影响: 用户影响: 对普通用户无影响 (默认关闭)。对需要高频拉取 prompt top-k logprobs 的 RL 训练等场景, 可大幅降低延迟和带宽。系统影响: 服务器端在编码阶段减少 CPU 开销, 减少网络传输量。团队影响: 需维护两套输出格式和对应的客户端解析逻辑。新标志与 #22244 的输出侧 base64 模式可能并存, 需注意客户端行为冲突 (PR 作者已提及可择一或互斥)。
- 风险标记: 默认关闭低风险, 新键兼容性

关联脉络

- PR #32078 flat raw top logprobs support: 本 PR 依赖此 PR 实现的 flat 数组格式, 是在其基础上的性能优化。
- PR #22244 return_logprobs_in_base64 (output-side): 同仓库另一个 base64 编码 logprobs 的实现, 但覆盖输出侧。两者独立可共存, 仅在客户端同时设置两个标志时可能需要协调。