PR #35001 完整报告

sgl-project/sglang

[Frontend] Apply request header overrides to chat completions

合并时间：2026-08-17 06:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35001

执行摘要

- 一句话：chat 补全接入请求头覆盖机制
- 推荐动作：值得精读，尤其是设计取舍：覆盖目标选内部 GenerateReqInput 而非 OpenAI 请求对象，保持了 API 请求的可审计性；用 env 开关显式门控，避免对既有用户产生隐式行为变化；测试覆盖 8 类 header 且断言原始请求不被污染，是入口一致性修复加高覆盖单测的小而美范本。关注点在于 header 覆盖与 DP 路由计算结果的优先级，需在分布式部署中进一步验证。

功能与动机

PR body 明确指出："Request header overrides are applied by the native generation endpoint but were not propagated when chat completions converted an OpenAI request into the internal generation request." 即同一个覆盖机制在原生生成端点生效，却在 chat 补全入口丢失，需要补齐这一行为分叉。

实现拆解

步骤 1：变更入口与依赖接入

- 在 python/sglang/srt/entrypoints/openai/serving_chat.py 中新增 from sglang.srt.entrypoints.request_headers import apply_header_overrides，把既有的覆盖逻辑引入 chat 入口。

步骤 2：核心逻辑改动

- 在 OpenAIServingChat._convert_to_internal_request 构造完 GenerateReqInput（字段包括 bootstrap_host / bootstrap_port / bootstrap_room、routed_dp_rank、disagg_prefill_dp_rank、rid、session_id、priority、routing_key 等）之后、return 之前新增分支：当 raw_request 非空且 envs.SGLANG_ENABLE_REQUEST_HEADER_OVERRIDES.get() 为真时，调用 apply_header_overrides(adapted_request, raw_request.headers)。
- 设计要点：覆盖目标是内部请求 adapted_request 而非原始 ChatCompletionRequest，避免污染 OpenAI 请求对象；测试中专门断言 request.model_dump() 与初始 body 相等，且 request 上没有被新增 conversation_id 属性。
- 行为影响：header 覆盖值（如 x-override-rid、x-override-routed-dp-rank、x-override-priority）优先级高于 body 字段。

步骤 3：测试配套

- test/registered/unit/entrypoints/openai/test_serving_chat.py 新增 test_chat_applies_pd_header_overrides，用 envs.SGLANG_ENABLE_REQUEST_HEADER_OVERRIDES.override(True) 显式开启开关，覆盖 8 类头部（rid、bootstrap host/port/room、conversation-id、routed/disagg rank、priority）并逐项断言内部请求字段。
- 同步补齐 Mock 的 weight_version 元数据与 request_logger，适配新处理路径；该测试经 register_cpu_ci 注册进 CPU CI。

步骤 4：配套说明

- 无配置、部署、schema 改动；作者在 body 中说明这是既有 opt-in 行为，无需更新文档。

关键文件：

- python/sglang/srt/entrypoints/openai/serving_chat.py（模块 聊天入口；类别 source；类型 dependency-wiring；符号 _convert_to_internal_request, apply_header_overrides）：核心源码改动：在 _convert_to_internal_request 构造 GenerateReqInput 之后、返回之前，按 env 开关与 raw_request 条件调用 apply_header_overrides，把 header 覆盖能力接入 chat 补全入口。
- test/registered/unit/entrypoints/openai/test_serving_chat.py（模块 聊天入口；类别 test；类型 test-coverage；符号 test_chat_applies_pd_header_overrides）：新增 test_chat_applies_pd_header_overrides，覆盖 8 类覆盖头并验证内部请求字段取值与原始 OpenAI 请求对象不被污染；同时补齐 Mock 的 weight_version 与 request_logger。

关键符号：_convert_to_internal_request, apply_header_overrides, test_chat_applies_pd_header_overrides

关键源码片段

python/sglang/srt/entrypoints/openai/serving_chat.py

核心源码改动：在 _convert_to_internal_request 构造 GenerateReqInput 之后、返回之前，按 env 开关与 raw_request 条件调用 apply_header_overrides，把 header 覆盖能力接入 chat 补全入口。

```
# 构造内部生成请求（GenerateReqInput），字段与 OpenAI 请求一一映射
adapted_request = GenerateReqInput(
    top_logprobs_num=request.top_logprobs or 0,
    return_sampling_mask=request.return_sampling_mask,
    stream=request.stream,
    return_text_in_logprobs=True,
    modalities=processed_messages.modalities,
    lora_path=lora_path,
    bootstrap_host=request.bootstrap_host,
    bootstrap_port=request.bootstrap_port,
    bootstrap_room=request.bootstrap_room,
    routed_dp_rank=effective_routed_dp_rank,
    disagg_prefill_dp_rank=request.disagg_prefill_dp_rank,
```

```
            return_hidden_states=request.return_hidden_states,
            return_routed_experts=request.return_routed_experts,
            routed_experts_start_len=request.routed_experts_start_len,
            rid=request.rid,
            session_id=request.session_id,
            extra_key=request.extra_key,
            cache_salt=request.cache_salt,
            require_reasoning=processed_messages.require_reasoning,
            priority=request.priority,
            routing_key=self.extract_routing_key(raw_request),
            custom_labels=custom_labels,
            custom_logit_processor=request.custom_logit_processor,
            images_config=getattr(request, "images_config", None),
            image_max_dynamic_patch=img_max_dynamic_patch,
            video_max_dynamic_patch=vid_max_dynamic_patch,
            max_dynamic_patch=getattr(request, "max_dynamic_patch", None),
            use_audio_in_video=getattr(request, "use_audio_in_video", False),
            return_prompt_token_ids=request.return_prompt_token_ids
            or request.return_token_ids,
        )

        # 关键新逻辑：仅当显式开启 header 覆盖开关且存在原始 HTTP 请求时，
        # 把 header 里的覆盖值应用到内部请求上；覆盖目标是 adapted_request 而非
        # 原始 ChatCompletionRequest，因此 body 内容保持原样，可供后续日志与审计复用
        if (
            raw_request is not None
            and envs.SGLANG_ENABLE_REQUEST_HEADER_OVERRIDES.get()
        ):
            apply_header_overrides(adapted_request, raw_request.headers)

        return adapted_request, request
```

test/registered/unit/entrypoints/openai/test_serving_chat.py

新增 test_chat_applies_pd_header_overrides，覆盖 8 类覆盖头并验证内部请求字段取值与原始 OpenAI 请求对象不被污染；同时补齐 Mock 的 weight_version 与 request_logger。

```
    def test_chat_applies_pd_header_overrides(self):
        # 请求 body 中的字段会被 header 覆盖，且原始请求对象保持不变
        request = ChatCompletionRequest(
            model="x",
            messages=[{"role": "user", "content": "Hi?"}],
            rid="body-rid",
            routed_dp_rank=3,
            disagg_prefill_dp_rank=4,
            priority=5,
        )
        self.fastapi_request.headers = {
            "x-override-rid": "header-rid",
            "x-override-bootstrap-host": "header-host",
```

```
        "x-override-bootstrap-port": "8998",
        "x-override-bootstrap-room": "456",
        "x-override-conversation-id": "conversation-1",
        "x-override-routed-dp-rank": "6",
        "x-override-disagg-prefill-dp-rank": "7",
        "x-override-priority": "8",
    }
    body = request.model_dump()

    processed_messages = MessageProcessingResult(
        "Test prompt", [1, 2, 3], None, None, [], [], None
    )
    with (
        envs.SGLANG_ENABLE_REQUEST_HEADER_OVERRIDES.override(True),
        patch.object(self.chat, "_process_messages", return_value=processed_messages),
    ):
        response = get_or_create_event_loop().run_until_complete(
            self.chat.handle_request(request, self.fastapi_request)
        )

    # 内部请求拿到 header 中的覆盖值，body 中的同名字段被覆盖
    self.assertEqual(response.choices[0].message.content, "Test response")
    adapted_request = self.tm.generate_request.call_args.args[0]
    self.assertEqual(adapted_request.bootstrap_room, 456)
    self.assertEqual(adapted_request.bootstrap_host, "header-host")
    self.assertEqual(adapted_request.bootstrap_port, 8998)
    self.assertEqual(adapted_request.rid, "header-rid")
    self.assertEqual(adapted_request.conversation_id, "conversation-1")
    self.assertEqual(adapted_request.routed_dp_rank, 6)
    self.assertEqual(adapted_request.disagg_prefill_dp_rank, 7)
    self.assertEqual(adapted_request.priority, 8)
    # 原始 OpenAI 请求对象保持原样，不被覆盖逻辑污染
    self.assertEqual(request.model_dump(), body)
    self.assertFalse(hasattr(request, "conversation_id"))
```

评论区精华

本 PR 没有收到成文 review 评论，唯一的 PR 活动是作者自己的 CI 重跑指令：

- 作者评论 /tag-and-rerun-ci，在 PR Test (Extra) 失败后主动触发重跑，最终 PR Test (Base)（Run #31927954410）通过后合并。
- 作者在 body 中明确表示不做性能测试：this is request metadata propagation and does not alter model execution，界定了改动风险范围。
- CI 状态与重跑 (other): 没有产生代码层面的 review 交锋；Base CI 通过后 PR 由作者直接合并。

风险与影响

- 风险：
  - 热路径分支：改动落在每次 chat 补全请求都会经过的 _convert_to_internal_request，新增一次条件判断与 header 扫描；apply_header_overrides 目前仅做字段赋值，开销可忽略，但未来若覆盖逻辑变重需重新评估。
  - 覆盖优先级语义：开关开启后 header 覆盖 body 字段（如 body-rid 被 header-rid 替代），可能静默改写用户显式传入的参数，属于该 opt-in 特性的固有语义，需要在部署文档中明示。
  - 与 DP 路由逻辑的交互：effective_routed_dp_rank 在转换过程中已按调度逻辑计算，随后又被 header 覆盖（测试中覆盖为 6），跨 rank 或 prefill/decode 分离场景下覆盖值可能绕过调度决策，值得在真实分布式环境验证。
  - 覆盖一致性：disagg_prefill_dp_rank、bootstrap_* 被独立覆盖，若用户只覆盖部分字段，可能产生不一致的组合。
  - 测试局限：全部基于 Mock，未做端到端请求级验证；extra CI（Run #31927954363）失败，虽未必与代码正确性相关，但合并前并未完全绿。
- 影响：
  - 用户侧：启用 SGLANG_ENABLE_REQUEST_HEADER_OVERRIDES 的部署中，chat completions 请求与原生生成端点行为对齐，可用 header 控制路由、优先级、bootstrap 等；未启用者完全无感。
  - 系统侧：缩小 OpenAI 兼容入口与原生入口的行为分叉，为网关层统一注入控制信号（request ID、优先级、disaggregation 路由）补齐最后一块拼图。
  - 团队侧：后续维护 header override 只需维护 apply_header_overrides 一个机制，无需在多个入口重复实现。
  - 风险标记：OpenAI 入口热路径新增分支，覆盖优先级 header 高于 body，功能由 env 开关门控，extra CI 未通过

关联脉络

- 暂无明显关联 PR