

PR #51426 完整报告

vllm-project/vllm

[Rust Frontend] Fix GLM-5.2 chat template rendering parity

合并时间: 2026-08-18 11:16

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51426>

执行摘要

本 PR 将 Rust 前端的 MiniJinja 从 2.22 升级到 2.24, 并给工具函数定义补上精确的 serde 序列化属性, 修复了 GLM-5.2 chat 模板无法编译、以及工具字段顺序和缺失语义与 Python 前端不一致的问题。改动集中在 rust/src/chat 渲染链路的 5 个文件, 最终用真实 GLM-5.2-FP8 模型的往返渲染验证 prompt 与 Python 完全一致 (980 字符、243 token、SHA-256 相同)。

功能与动机

PR 目标明确: “Fix GLM-5.2 chat-template rendering and align Rust tool serialization with the Python frontend.” MiniJinja 2.18 会拒绝 GLM-5.2 的模板 (mid-chain dotted integer lookup 报 unexpected float, expected identifier or integer); 2.22 能编译但工具字段顺序与缺失语义都和 Python 不一致。具体差异包括: 字段顺序变成 description、name、parameters, 以及缺失 description 时字段被整体丢弃, 而 Python 的 tool.model_dump() 保留 description=null。这些差异会让相同请求在两种前端渲染出不同的 prompt 与 token 序列, 破坏 Rust/Python 间的可互换性。

实现拆解

1. 依赖升级: rust/Cargo.toml 将 minijinja 与 minijinja-contrib 从 2.22 升到 2.24, 保留 preserve_order 特性; rust/Cargo.lock 同步版本与校验和。升级依据是上游 2.24 修复了 Serde struct 定义顺序在模板迭代中被保留的问题 (mitsuhiko/minijinja#920)。
2. 模板语法回归: rust/src/chat/src/renderer/hf/template.rs 的 tests 模块新增 test_midchain_dotted_integer_lookup, 用 {{ values.0.name }} 验证 mid-chain dotted integer lookup 的编译与求值, 把 GLM-5.2 模板的关键语法能力固化在版本上。
3. 字段序列化语义: rust/src/chat/src/renderer/hf/mod.rs 中 TemplateToolDefinition 保持 typed struct, 仅在 strict 字段上增加 #[serde(skip_serializing_if = "Option::is_none")]。由此实现: 缺失 description → null、parameters 为 null 保持 null、缺失 strict 省略、strict=false 保留, 逐字段对齐 Python FunctionDefinition 的 model_dump() 行为。
4. 字段级回归测试: 同一文件新增 chat_template_preserves_openai_tool_field_order (断言 nameldescriptionlparametersl 顺序) 与 chat_template_preserves_python_optional_tool_fields (断言空值语义) 两个测试, 防止后续改动悄悄改变 prompt。
5. 真实模型 roundtrip: rust/src/chat/tests/roundtrip.rs 新增 RoundtripCase::glm52 (zai-org/GLM-5.2-FP8), 注册 reasoning_and_content 与 tool_call_mix 两个 fixture, 加载真实 tokenizer 与模板, 完成渲染 → 解析 → 历史追加 → 再渲染的完整链路验证。

rust/src/chat/src/renderer/hf/mod.rs

核心实现文件：为 TemplateToolDefinition.strict 增加 skip_serializing_if 属性，使缺失 strict 被省略而 strict=false 保留；同时新增两个字段级回归测试锁定字段顺序与缺失语义。

```
// TemplateToolDefinition 是工具函数在 Jinja 模板上下文中的投影。
// 字段声明顺序即模板 items() 遍历顺序：name -> description -> parameters -> strict,
// 依赖 MiniJinja 2.24 的 preserve_order 特性与上游对 Serde struct 顺序的修复。
#[derive(Debug, Serialize)]
struct TemplateToolDefinition {
    name: String,
    // 缺失的 description 序列化为 null 而非丢字段，与 Python model_dump() 行为对齐
    description: Option<String>,
    // parameters 保留原始 JSON 值，null 参数维持 null
    parameters: JsonValue,
    // 缺失的 strict 直接省略；显式 strict: false 时保留，二者均与 Python 语义一致
    #[serde(skip_serializing_if = "Option::is_none")]
    strict: Option<bool>,
}

/// 回归测试：模板遍历工具字段时，迭代顺序必须严格为 nameldescriptionlparametersl
#[test]
fn chat_template_preserves_openai_tool_field_order() {
    let mut request = sample_request(vec![ChatMessage::text(ChatRole::User, "hello")]);
    let tools = vec![ChatTool {
        name: "get_weather".to_string(),
        description: Some("Get weather".to_string()),
        parameters: serde_json::json!({"type": "object"}),
        strict: None,
    }];
    request.tool_context = crate::request::ResolvedToolContext::new(
        &request.messages,
        tools,
        Some(ChatToolChoice::Auto),
        true,
    )
    .expect("tool context should resolve");

    let rendered = render(
        Some("{% for key, value in tools[0].function.items() %}{{ key }}|{% endfor %}"),
        &request,
    )
    .unwrap();

    assert_eq!(rendered, "nameldescriptionlparametersl");
}
```

rust/src/chat/src/renderer/hf/template.rs

模板编译链路所在文件，新增 mid-chain dotted integer lookup 回归测试，固化 MiniJinja 2.24 对该语法的支持，防止 GLM-5.2 模板编译期回退。

```
/// 回归测试：模板中段出现的点号后整数索引（values.0.name）语法。
/// MiniJinja 2.18 会在此处报 unexpected float, expected identifier or integer,
/// 导致 GLM-5.2 自带模板编译失败；该测试把语法能力固化在 2.24 版本上。
#[test]
fn test_midchain_dotted_integer_lookup() {
    let template = CompiledChatTemplate::new(
        "{{ values.0.name }}".to_string(),
        ChatTemplateContentFormatOption::Auto,
    )
    .unwrap();
    let mut kwargs = HashMap::new();
    kwargs.insert("values".to_string(), serde_json::json!([{"name": "first"}]));

    let result = template
        .apply(TemplateContext {
            template_kwargs: Some(&kwargs),
            ..Default::default()
        })
        .unwrap();

    assert_eq!(result, "first");
}
```

rust/src/chat/tests/roundtrip.rs

集成测试：新增 GLM-5.2 真实模型往返用例，覆盖 reasoning+content 与混合工具调用两种 fixture，验证渲染、解析、历史追加的完整链路与 Python 一致。

```
/// GLM-5.2 XML-like 工具参数格式，带 `thinking` 推理标签。
/// 与 GLM-4.5/4.7 同族，但内嵌模板依赖 mid-chain dotted integer lookup 语法。
fn glm52() -> Self {
    Self {
        model_id: "zai-org/GLM-5.2-FP8",
        assistant_stop_suffix: "",
        tool_call_parser: ParserSelection::Auto,
        reasoning_parser: ParserSelection::Auto,
        thinking_behavior: ThinkingBehavior::Toggleable { default: true },
        json_fmt: compact_json_fmt(),
        sort_json_keys: false,
    }
}

// 注册到宏生成的 roundtrip_glm52 测试：覆盖纯文本 + 推理内容、
// 以及混合工具调用两种 fixture，渲染固定对话、解析输出、追加历史再渲染下一轮。
roundtrip_tests! {
    // ...
    glm52 => [reasoning_and_content, tool_call_mix],
}
```

```
// ...  
}
```

评论区精华

BugenZhao: “serializing the Rust tool structs changed the function-field order to description, name, parameters, while the OpenAI request and Python frontend preserve name, description, parameters.”

BugenZhao: “It turns out the most recent release of minijinja v0.24 has resolved this issue (mitsuhiko/minijinja#920) so we can stick to structs.”

Codex review (P1) : “The Python frontend uses tool.model_dump() ... while FunctionDefinition retains description=None ... so Python supplies description: null. Templates that serialize tools or distinguish undefined from null ... will therefore render different prompts and token sequences.”

WoosukKwon: “@codex review This was written by Codex btw.”

BugenZhao: “My last @codex review was also sent by Codex 🤖.”

风险与影响

- 依赖升级面: MiniJinja 2.22 → 2.24 影响 rust/src/chat 下全部模板编译与渲染路径。HF renderer 套件 76/76 通过, 但其他模型模板若依赖旧版本边界行为, 可能在升级后产生细微差异。
- 字段顺序隐式依赖 struct 定义: preserve_order 按 Serde struct 声明顺序输出, 后续调整 TemplateToolDefinition 字段位置会静默改变所有工具类模板的 prompt, 需依赖新增回归测试兜底。
- 序列化语义变更: 缺失 description 从“丢弃字段”变为“输出 null”, 对已依赖旧行为的模板是 prompt 变化, 接入方需回归线上模板。
- CI 稳定性: roundtrip_glm52 需从 HF 下载真实 tokenizer 与模板, 网络或缓存问题可能导致不稳定 (与 kimi_k3 用例类似)。
- 影响范围: 仅 Rust 前端渲染链, 不涉及 Python 侧与核心调度路径; 对用户而言, GLM-5.2 在 Rust 前端可正常使用且 prompt 与 Python 完全一致。

关联脉络

本 PR 是 MiniJinja 升级链路 (#51235 先行升到 2.22, 本 PR 再升到 2.24) 的延续, 目标是消除 Rust 前端与 Python 前端的渲染差异。从近期 Rust 前端系列 PR (如 #52031 Rust 前端 gRPC 能力通告、#52573 结构化输出跳过 reasoner) 可以看出, Rust 前端正在系统性地对齐 Python 能力与协议。本 PR 建立的“真实模型 roundtrip + 字段级回归 + 哈希比对”验证范式, 可复用于后续新模型接入与依赖升级, 是 Rust 前端走向生产就绪的重要一环。