PR #43872 完整报告

vllm-project/vllm

[Rust Frontend] Add `hy_v3` tool parser

合并时间：2026-05-28 22:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43872

执行摘要

本 PR 为 vLLM Rust 前端新增 hy_v3 工具解析器，支持腾讯 HY3 系列模型（如 tencent/Hy3-preview）的 XML 风格工具调用格式。实现遵循现有 MiniMax M2 风格的事件驱动架构，同时扩展了参数类型别名以兼容 HY3 的 schema 类型名称。PR 由 njhill 审核并合并，代码质量较高，测试覆盖充分。

功能与动机

需要为上游 HY3 模型提供 Rust 前端的工具调用支持。HY3 模型使用 <tool_calls> 包裹的 XML 标记进行工具调用，与已有解析器格式不同。此举填补了该模型在 Rust 前端的能力空白。

实现拆解

1. 解析器核心：在 rust/src/tool-parser/src/hy_v3.rs 中定义 HyV3ToolParser，通过有限状态机（HyV3Mode）管理文本 / 工具块 / 完成三种状态，利用 winnow 组合子解析 <tool_call> 内的参数键值对。事件驱动模型允许流式消费 token 输出。
2. 参数类型扩展：在 parameters.rs 的 from_type_name 中增加 double、map、list 的映射，使 JSON schema 转换能识别 HY3 的非标准类型名称。
3. 模块导出与注册：在 tool-parser/src/lib.rs 中声明模块并公开 HyV3ToolParser；在 chat/src/parser/tool/mod.rs 中注册 parser 名称 "hy_v3" 和模型匹配模式 "hy3" / "hy_v3"，自动关联 tencent/Hy3-preview 等模型。
4. 测试覆盖：继承上游 HY3 解析器的测试用例（零参数调用、多行参数、流式切片、不完整输入等），并在聊天示例中添加模型 ID 解析测试。

rust/src/tool-parser/src/parameters.rs

扩展参数类型别名映射，增加 double、map、list 以兼容 HY3 schema 类型名称。

```
// 在 from_type_name 函数中，本次 PR 增加了以下三行以支持 HY3 的类型别名
// ( 原 "number" | "float" 后添加 "double"，原 "object" 后添加 "dict" | "map"，原 "array" | "arr"
后添加 "list")
fn from_type_name(kind: &str) -> Option<Self> {
    let kind = kind.trim().to_ascii_lowercase();
    match kind.as_str() {
        "string" | "str" | "text" | "varchar" | "char" | "enum" => Some(Self::String),
        "integer" | "int" => Some(Self::Integer),
        // 增加 "double" 映射为 Number
        "number" | "float" | "double" => Some(Self::Number),
```

```
        "boolean" | "bool" | "binary" => Some(Self::Boolean),
        // 增加 "map" 映射为 Object
        "object" | "dict" | "map" => Some(Self::Object),
        // 增加 "list" 映射为 Array
        "array" | "arr" | "list" | "sequence" => Some(Self::Array),
        "null" => Some(Self::Null),
        _ if kind.starts_with("int")
            || kind.starts_with("uint")
            || kind.starts_with("long")
            || kind.starts_with("short")
            || kind.starts_with("unsigned") =>
        {
            Some(Self::Integer)
        }
        _ if kind.starts_with("num") || kind.starts_with("float") => Some(Self::Number),
        _ if kind.starts_with("dict") => Some(Self::Object),
        _ if kind.starts_with("list") => Some(Self::Array),
        _ => None,
    }
}

// 对应的测试：验证新增类型名称的转换正确性
mod tests {
    #[test]
    fn converts_supported_types() {
        let params = ToolSchema::from_schema(&json!({
            "type": "object",
            "properties": {
                // 原有类型 ...
                "ratio": { "type": "double" }, // 新增
                "mapping": { "type": "map" }, // 新增
                "names": { "type": "list" }, // 新增
            }
        }));
        assert_eq!(params.convert("ratio", "2.5"), json!(2.5));
        assert_eq!(params.convert("mapping", r#"{"k":1}"#), json!({"k":1}));
        assert_eq!(params.convert("names", r#"["a","b"]"#), json!(["a", "b"]));
    }
}
```

评论区精华

无 review 评论；PR 由 njhill 直接批准合并，说明实现符合预期且质量达标。

风险与影响

- 风险：新解析器仅服务 HY3 模型，不影响其他模型。主要风险在于 HY3 标记并非 tokenizer 的 special token，若模型输出不严格遵循格式可能导致解析失败（测试已覆盖 malformed input）。另外缺少端到端的 Rust 前端集成测试。

- 影响：对使用 tencent/Hy3-preview 的用户，工具调用功能正式可用。团队后续需维护与上游同步的兼容性。

关联脉络

本 PR 是 Rust 前端工具解析器系列的延续。此前已有 #43850（Gemma4 性能优化）和 #42879（DeepSeek DSML 流式修复）等 PR，共同构建了覆盖多模型的可扩展解析器生态。该 PR 进一步验证了架构的可复用性：通过复用 ToolSchema 和 parse_buffered_event，新增解析器只需约 550 行代码。