

PR #45560 完整报告

vllm-project/vllm

[GPT-OSS] Strict tool call and constrained decoding for Harmony

合并时间: 2026-08-02 00:10

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45560>

执行摘要

- 一句话: Harmony 严格工具调用与全生成期约束解码
- 推荐动作: 值得精读。该 PR 展示了三个可借鉴的设计决策: 一是用 vLLM 自建 structural tag 而非 patch 上游 xgrammar 内置实现, 降低对上游 bug 的耦合; 二是把用户各类 response_format 统一规范化为单个 structural_tag 通道, 用 grammar admission 测试 (xgrammar.testing._is_grammar_accept_string) 从正反样本锁定接受集合; 三是把约束起点从 final 通道前移到全生成期并用 reasoning-aware 包装避免行为回退。阅读时建议重点对照 vllm/parser/harmony.py 的 _assemble_tag/get_harmony_structural_tag 与 tests/parser/test_harmony.py 的 TestAdjustRequest, 理解两处前端接线 (online_renderer.py、responses/serving.py) 的同步关系。

功能与动机

PR body 指出: 'vLLM's constrained decoding for Harmony delays enforcing response_format constraints until <lchannel>final...<lmessage> is detected. This works correctly for JSON formats but creates a mismatch for structural-tag tool calling, since those constraints need to apply during the commentary/tool-call phase rather than only in the final-message channel.' 即现有实现只在 final 通道触发约束, 工具调用阶段的 commentary/analysis 通道不受约束, 导致 GPT-OSS 小模型生成格式漂移。作者在 issue 评论中补充: 'The structural tags help gpt-oss-20b, which has trouble following the Harmony format, a lot. 20% uplift in BFCL multiturn-base if we decide to enable it by default', 说明 20b 模型是本次严格约束的主要受益者。

实现拆解

实现按以下 5 步展开:

1. 在 vllm/parser/harmony.py 定义 Harmony 专用 structural tag: 新增 _END_TAG、_FINAL_BEGIN、_TOOL_CALL_CHANNELS、_FUNCTION_CALL_BEGINS 等常量; _assemble_tag 把 analysis/commentary 可选通道 (内部放行任意文本) 与核心 content 拼接为完整 SequenceFormat; 通过 @register_vllm_structural_tag('harmony') 注册 get_harmony_structural_tag, 其中无内置工具时按工具 x 前缀 x 通道的笛卡尔积生成严格 JSONSchemaFormat 约束, 有内置工具时退化为仅保证调用某个工具的 fallback 路径, tool_choice == 'auto' 时额外放行两种 final 通道写法。

2. 新增请求规范化入口：_params_to_final_content 把 StructuredOutputsParams (json_object、json、regex、choice、grammar、structural_tag, 含 LegacyStructuralTagResponseFormat 与 StructuralTagResponseFormat 两种 tag 格式) 映射为 xgrammar Format; _adjust_output_format 将请求改写为只保留 structured_outputs.structural_tag 的 reasoning-aware tag, 其余约束字段清空; HarmonyParser.adjust_request 先执行 _adjust_output_format 再走父类逻辑。
3. 简化 GptOssReasoningParser: 删除约 130 行, 包括 no_func_reasoning_tag、from_built_in_tool_to_tag、tag_with_built_in_funcs、prepare_structured_tag 以及与 <|channell>final 前缀扫描相关的 token 级推理结束检测; is_reasoning_end 与 is_reasoning_end_streaming 直接返回 True, 理由是由 structural tag 全程接管约束、不再需要边界切换。
4. 接入两处前端调用点: vllm/renderers/online_renderer.py 的 render_chat 与 vllm/entrypoints/openai/responses/serving.py 的 _make_request_with_harmony 在进入 Harmony 渲染前调用 parser.adjust_request; 同时删除 serving 中 tool_choice 仅支持 auto/none 的 NotImplementedError 守卫, 使 required/named 也可用 (strict 逻辑本身已支持), 并同步删除集成测试里“required 必抛 InternalServerError”的旧断言。
5. 配套调整与测试: vllm/tool_parsers/structural_tag_registry.py 把 _get_function_parameters 公开重命名为 get_function_parameters 供 harmony.py 使用, 并从 XGRAMMAR_BUILTIN_STRUCTURAL_TAG_MODELS 移除 harmony; vllm/tool_parsers/gptoss_tool_parser.py 声明 structural_tag_model = 'harmony'。测试方面, tests/parser/test_harmony.py 新增 TestAdjustRequest, 用 xgrammar.testing._is_grammar_accept_string 对 COMMENTARY、TOOL_CALL_1/2、FINAL_JSON_SCHEMA 等样本断言 grammar 接受 / 拒绝集合; tests/reasoning/test_gptoss_reasoning_parser.py 从 351 行缩减为对恒 True 边界的断言; Responses 集成测试对 tool_choice 做 auto/required 参数化。

关键文件:

- vllm/parser/harmony.py (模块 解析器; 类别 source; 类型 core-logic; 符号 adjust_request, _assemble_tag, get_harmony_structural_tag, _params_to_final_content): 核心改动文件: 新增 Harmony 感知 structural tag builder、_assemble_tag、_params_to_final_content、_adjust_output_format 与 adjust_request, 把约束解码从 final 通道扩展到全生成期。
- vllm/reasoning/gptoss_reasoning_parser.py (模块 推理解析; 类别 source; 类型 core-logic; 符号 from_built_in_tool_to_tag, tag_with_built_in_funcs, prepare_structured_tag, is_reasoning_end): 删除了约 130 行的推理结束检测与手工 structural tag 构造逻辑, is_reasoning_end 恒为 True, 是整个行为转变的另一半。
- tests/parser/test_harmony.py (模块 解析测试; 类别 test; 类型 test-coverage; 符号 TestAdjustRequest, _build_request, _assert_format_cleared, _assert_structured_outputs_admission): 新增 337 行 TestAdjustRequest, 用 xgrammar Grammar 与 _is_grammar_accept_string 对多组 Harmony 生成样本断言接受 / 拒绝集合, 是锁定行为契约的关键测试。

- `vllm/tool_parsers/structural_tag_registry.py` (模块 工具解析; 类别 source; 类型 core-logic; 符号 `_get_function_parameters`, `get_function_parameters`) : 把 `_get_function_parameters` 公开为 `get_function_parameters` 供 `harmony.py` 复用, 并把 `harmony` 从 `XGRAMMAR_BUILTIN_STRUCTURAL_TAG_MODELS` 移除, 标志着 `harmony tag` 所有权从 `xgrammar` 移交 `vLLM`。
- `vllm/entrypoints/openai/responses/serving.py` (模块 响应服务; 类别 source; 类型 core-logic) : `Responses API` 的 `Harmony` 分支调用 `adjust_request`, 并删除 `tool_choice` 仅限 `auto/none` 的限制, 使 `required/named` 可用。
- `vllm/renderers/online_renderer.py` (模块 渲染器; 类别 source; 类型 core-logic) : `Chat Completions (online renderer)` 的 `Harmony` 分支同样接入 `adjust_request`, 保证 `CC` 与 `Responses` 两条前端路径行为一致。
- `vllm/tool_parsers/gptoss_tool_parser.py` (模块 工具解析; 类别 source; 类型 core-logic) : 声明 `structural_tag_model = 'harmony'`, 让工具解析器框架选择 `vLLM` 自建的 `harmony tag builder`, 是注册机制的接线点。
- `tests/reasoning/test_gptoss_reasoning_parser.py` (模块 推理测试; 类别 test; 类型 test-coverage; 符号 `test_gptoss_reasoning_ended_is_true`) : 测试从 351 行缩减为 12 行, 直接反映旧边界扫描逻辑整体作废, 改为断言 `is_reasoning_end` 恒 `True`。
- `tests/entrypoints/openai/responses/test_harmony.py` (模块 集成测试; 类别 test; 类型 test-coverage; 符号 `test_function_calling`, `test_function_calling_multi_turn`, `test_function_calling_required`, `test_function_calling_full_history`) : 工具调用集成测试按 `tool_choice (auto/required)` 参数化, 并删除旧有的“`required` 抛 `InternalServerError`”断言, 验证放开限制后的端到端行为。

关键符号: `adjust_request`, `_assemble_tag`, `get_harmony_structural_tag`, `_params_to_final_content`, `_adjust_output_format`, `get_function_parameters`, `is_reasoning_end`, `is_reasoning_end_streaming`

评论区精华

核心讨论集中在 4 个点上:

1. 自建 tag 还是 patch xgrammar: `sfeng33` 在 `structural_tag_registry.py` 的 `_patch_harmony` 处评论 'We can define the tags directly in `HarmonyParser`', `chaunceyjiang` 也建议 'implementing the complete structural tag directly in `vLLM` rather than relying on this kind of patch-based solution'。结论: `yzong-rh` 改为在 `HarmonyParser` 内通过 `@register_vllm_structural_tag('harmony')` 自建完整 tag, 绕开 `xgrammar` 内置实现的 bug。
2. 无条件 `is_reasoning_end=True` 的行为影响: `bbrowning` 警告 'if a client passes `structured_outputs` directly the new unconditional `is_reasoning_end` of `True` will cause that to start enforcing output immediately ... I suspect we'll break some in the wild usage'。结论: `yzong-rh` 追加 `commit` 把用户传入的 `request.structured_outputs` 也转换为 `reasoning-aware structural_tag`, 避免用户可见行为变化。

3. 控制 token 泄漏: ankrovv 实测发现 gpt-oss-120b 的响应中 mcp_call 的 name/server_label 变成 '<lconstrainl>json', 即 harmony 控制 token 泄漏进 recipient。结论: yzong-rh 确认是 openai_harmony StreamableParser 的处理问题, 由 #45657 与 #47185 修复, 并额外禁止 <lchannel>final json<lmessage> 形态。
4. 死代码: bbrowning 在 approval 中指出本 PR 删除了 prepare_structured_tag 的唯一真实实现, 剩余调用点变成 no-op, 但该方法 ReasoningParser 接口的一部分, 删除需要更广泛考虑, 同意推迟。
- 自建 harmony structural tag 还是 patch xgrammar 内置实现 (design): yzong-rh 改为在 HarmonyParser 内通过 @register_vllm_structural_tag('harmony') 自建完整 tag, 绕开 xgrammar 内置实现的 bug, 且不占用 registry 的 patch 路径。
- 无条件 is_reasoning_end=True 对 structured_outputs 存量用户的影响 (correctness): yzong-rh 追加 commit 将用户传入的 request.structured_outputs 也转换为 reasoning-aware structural_tag, 避免用户可见行为变化。
- <lconstrainl>json 控制 token 泄漏导致 MCP 调用 (correctness): 依赖上游修复, 本 PR 通过 _normalize_recipient 与禁止危险前缀规避。
- reasoning 参数是否应传给 structural tag builder (design): 保持 reasoning=False, 未改动 abstract_parser 行为。
- prepare_structured_tag 成为死代码 (other): 推迟清理, 保留接口, 后续单独处理。
- Responses API 放开 tool_choice 限制 (design): 已放开限制, 集成测试按 auto/required 参数化覆盖。

风险与影响

- 风险:
 1. 核心生成路径行为变更: is_reasoning_end 从 token 扫描改为恒 True, 所有依赖该边界判断的逻辑 (推理分段、约束施加时机) 都会改变; bbrowning 已提示这可能破坏线上直传 structured_outputs 的存量用户, 虽然本 PR 做了转换兜底, 仍需关注边缘路径。
 2. 特殊 token 泄漏: yzong-rh 自测发现模型可能以子词方式逐步生成特殊 token (如 '<l e n d l >') 而非单个 <lendl>, 导致 tag 边界泄漏进输出; 这不是 Harmony 特有, 但在全生成期约束下更易暴露, 属于未解决的已知问题。
 3. xgrammar API 依赖: 自建 tag 大量依赖 xgrammar 的 OptionalFormat、SequenceFormat、TriggeredTagsFormat 等 API, 且 _END_TAG 里的 '' 是绕开默认停止符的 workaround, xgrammar 版本升级可能引入兼容性风险。
 4. 前端双调用点: online_renderer.py 与 responses/serving.py 各自独立调用 adjust_request 且都带 TODO 注明未来统一, 后续重构时容易遗漏一侧。
 5. 遗留死代码: ReasoningParser 抽象上的 prepare_structured_tag 已无实现, 后续调用者可能误用。- 影响: 用户影响: GPT-OSS/Harmony 系列 (20b/120b) 的工具调用行为显著变化——Responses API 的 tool_choice=required/named 从 NotImplementedError 变为可用; gpt-oss-20b 的 BFCL multi_turn 四类指标从 baseline 的 23%-35.5% 提升到 guided 的 32.5%-58%; 120b 变化不大 (部分指标小幅波动)。系统影响: 约束解码从 xgrammar 内置 harmony tag 切换为 vLLM 自建 tag

, benchmark 显示 guided 与 baseline 吞吐几乎一致 (78.55 vs 80.37 req/s, 输出 token 吞吐 7563.75 vs 7773.67 tok/s), 无明显性能回退。团队影响: parser、reasoning、tool_parsers、frontend 多模块联动, get_function_parameters 公开与 structural_tag_model 声明为后续其他模型复用 vLLM 自有 tag 提供了模式; 测试报告 429 passed、6 skipped、1 xfailed、1 xpassed。 - 风险标记: 核心解析路径行为变更, structured_outputs 语义变化, 特殊 token 泄漏风险, 两处前端调用点需同步维护, prepare_structured_tag 遗留死代码

关联脉络

- PR #45003 (标题未在材料中给出): PR body 声明本 PR 是 #45003 与 #45396 的后续, 延续 Harmony 工具调用约束解码功能线。
- PR #45396 (标题未在材料中给出): 同上, Harmony 早期实现的前序 PR。
- PR #45657 (标题未在材料中给出, 修复 openai_harmony 对 <lconstraint>json 的处理): review 中 yzong-rh 指出它修复 ankrovv 发现的控制 token 泄漏问题。
- PR #47185 (标题未在材料中给出, 将 _normalize_recipient 接入 Responses API): review 中 yzong-rh 指出需要它配合修复 recipient 泄漏。
- PR #44664 (标题未在材料中给出, 通过 prefill final channel 实现严格工具调用的替代方案): yzong-rh 在评论中对比该方案与本 PR 的 structural tag 方案, 邀请 ankrovv 验证。
- PR #49227 (标题未在材料中给出): yzong-rh 回复 hanahhh 时表示希望先让 #49227 落地, 两者存在先后依赖关系。