

PR #56332 完整报告

vllm-project/vllm

[CI/Build] Add DSv4.1 auto-label rules and narrow DSv4

合并时间: 2026-09-13 00:50

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56332>

执行摘要

- 一句话: 新增 DSv4.1 自动打标规则, 并用负向断言收紧 DSv4
- 推荐动作: 若你是维护者或 CI 负责人, 值得精读 PR body 中“两个打标器如何保持互斥”的推理段落, 尤其是用负向先行断言做版本边界、以及“Flash/Pro 不可按文件条件区分所以不拆标签”的取舍; 普通工程师只需知道这是标签配置变更, 无需逐行看 YAML。建议跟进三件事: 一是合并后立刻确认 DSv4.1 标签存在并完成回填与 #56228 纠正; 二是把作者验证过的 25 条标题案例固化成测试 (作者已明确愿意提交), 补上 check-label-rules 不覆盖的标题正则盲区; 三是把 `(?![. _]?)` 这种一次性 lookahead 抽象成更通用的版本号边界策略, 否则 V4.2 出现时会重演同一问题。

功能与动机

PR body 明确给出两个具体缺陷: 其一, 「DeepSeek-V4.1 had no label of its own, and the existing DSv4 patterns were claiming all of its traffic」——旧正则 `deepseek[-\s]?v4` 在 v4 之后没有任何边界约束, 因此 `DeepSeek-V4.1-Flash` 被 DSv4 命中, #56228 这个纯 V4.1 PR 被错误标成 DSv4; 其二是 issue labeler 的 DSv4 匹配器是子串 `deepseek-v4` 加关键字 `DSv4`, 而 mergify 允许 `[-\s]?` 分隔符, 两个打标器语义不一致, #56297 (“DeepSeek V4.1 rejects Responses API text content types”) 只拿到 `deepseek` 标签。作者进一步论证架构差异: V4.1 有独立模型树 `vllm/models/deepseek_v4_1/`、独立 `tokenizer_mode` (`deepseek_v41`)、独立的 `parser`、`reasoning parser`、`tool parser` 与 `config` 类, 并新增 `common/engram.py`、`nvidia/model_state.py`、`common/ops/indexer_k_store.py` 等文件, 同时不含 V4 的 `mtp.py`、`cpu/`、`xpu/`、`fi_moe.py`。因此需要把打标边界从“DeepSeek V4 系列”收窄为“DeepSeek V4”与“DeepSeek V4.1”两条互斥标签。

实现拆解

本 PR 是纯配置变更, 没有源码、测试或部署产物改动, 实现可拆为四步:

1. 确立标签边界策略 (`.github/mergify.yml`)。核心手法是用负向先行断言 `(?![. _]?)` 表示“v4 之后若紧跟 `.1` 或 `_1` 则不算 V4”, 同时把分隔符从 `[-\s]?` 放宽为 `[-\s_]?` 以覆盖 `deepseek_v4` 写法。选择 lookahead 而不是精确枚举版本号, 是为了让规则仍然匹配所有非 4.1 的 V4 标题 (V4-Flash、V4-Pro 等)。
2. 新增 `label-dsv4-1` 规则 (`.github/mergify.yml`)。规则名用 `label-dsv4-1`, 标签值是 `DSv4.1`; 条件同样先排除 `stale`, 再用 5 条 `file` 条件加 1 条 `title` 条件做“或”组合, 并以 `title-only` 方式匹配标题, 避免正文里的模型名带来噪音标签。文件条件的枚举依据是 4.1

源码的三种拼写：`deepseek_v4_1`（模型树）、`deepseek_v41`（大多数位置）、`deepseekv41`（tool parsers）。

条件	覆盖范围	要点
<code>^vllm/models/deepseek_v4_1/</code>	V4.1 模型树	与 DSv4 的 <code>^vllm/models/deepseek_v4/</code> 不重叠，后者要求 <code>deepseek_v4</code> 后紧跟 <code>/</code>
<code>^vllm/(?:(?:parser tokenizers reasoning transformers_utils/configs)/deepseek_v41.*\.py</code>	解析器 / tokenizer / config	覆盖 <code>deepseek_v41</code> 拼写
<code>^vllm/tool_parsers/deepseekv41.*\.py</code>	tool parsers	覆盖 <code>deepseekv41</code> 拼写
<code>^rust/src/(?:(?:chat parser)/src/.*)deepseek_v41</code>	Rust 前端	只覆盖 <code>chat/parser</code> 子目录
<code>^tests/.*)deepseek_?v41</code>	测试	兼容带与不带下划线两种写法

1. 统一两个打标器的语义（`.github/workflows/issue_autolabel.yml`）。把 `DSv4` 条目从 `keywords: [{ term: "DSv4" }] + substrings: [{ term: "deepseek-v4" }]` 改为单条 `regexPatterns`，正则文本与 `mergify` 完全一致，并用 JS 的 `gi` 标志替代 Python 端的 `inline (?i)`；新增 `"DSv4.1"` 条目，正则同时接受 `DSv4.1/DSV4_1` 与 `DeepSeek-V4.1/DeepSeek V4_1` 等写法。两个条目的 `searchIn` 都保持 `title`，与 `mergify` 的标题匹配范围对齐。这样两个引擎在同一标题上给出相同结论，避免“同一 PR 在 `mergify` 与 `issue labeler` 下得到不同标签”的长期不一致。
2. 验证、前置依赖与后续动作。验证侧使用 `.venv/bin/pre-commit run --files .github/mergify.yml .github/workflows/issue_autolabel.yml 与 tools/pre_commit/check_label_rules.py`（含 `actionlint`、`check-label-rules`），作者还额外注入“故意失效的 `glob`”确认 `check-label-rules` 不是空跑；仓库外另做 25 条真实 V4/V4.1 标题在 Python re 与 JS RegExp 双引擎对拍（25/25 全部通过）、文件条件与代码树核对（44/44 个 V4.1 文件，无过匹配、无向 DSv4 泄漏）、以及两个标签互斥性检查。本 PR 未新增任何测试文件；作者仅表示“愿意提交标题正则用例作为长期守卫”，但没有落地。前置依赖是 `DSv4.1` 标签必须先存在，规则本身不会创建标签；后续需要回填约 14 个 V4.1 PR 和 2 个 issue，并修正 #56228 的 DSv4 标签。

关键文件：

- `.github/mergify.yml` (模块 标签自动化; 类别 `infra`; 类型 `configuration`; 符号 `label-dsv4`, `label-dsv4-1`): 打标边界的核心落点: 用负向先行断言收紧 `label-dsv4` 的标题正则, 并新增 `label-dsv4-1` 规则 (5 条文件条件 + 1 条标题条件), 决定 `DSv4` 与 `DSv4.1` 是否互斥。
- `.github/workflows/issue_autolabel.yml` (模块 问题打标; 类别 `infra`; 类型 `configuration`; 符号 `DSv4`, `DSv4.1`): `issue labeler` 侧配套: 新增 "`DSv4.1`" 条目, 并把 `DSv4` 从 `keywords + substrings` 子串匹配改成与 `mergify` 同源的正则, 消除两个打标器之间的语义差 (原先漏掉空格分隔标题)。

关键符号: `label-dsv4`, `label-dsv4-1`, `DSv4` (`issue labeler labelConfig` 条目), `DSv4.1` (`issue labeler labelConfig` 条目)

关键源码片段

`.github/workflows/issue_autolabel.yml`

`issue labeler` 侧配套: 新增 "`DSv4.1`" 条目, 并把 `DSv4` 从 `keywords + substrings` 子串匹配改成与 `mergify` 同源的正则, 消除两个打标器之间的语义差 (原先漏掉空格分隔标题)。

// `.github/workflows/issue_autolabel.yml` 中的 `labelConfig` (JS 匹配逻辑内嵌片段)
 // 目标: `issue labeler` 与 `mergify` 使用同一套标题正则, 两个打标器结论保持一致。

```
deepseek: { keywords: [{ term: "DeepSeek", searchIn: "title" }], },

// DSv4 与 DSv4.1 是两条独立架构线, 负向断言保证两个标签不会同时命中。
DSv4: {
  // 旧实现是 keywords: ["DSv4"] + substrings: ["deepseek-v4"]:
  // 子串匹配既漏掉 "DeepSeek V4.1" 这类空格分隔标题, 也无法把 4.1 排除,
  // 换成与 mergify 相同的正则后, 两边语义完全对齐。
  // 注意 YAML 双引号标量需要写成 \\b, 交给 RegExp 构造的才是 \b;
  // 大小写不敏感由 gi 标志承担 (mergify 侧用的是 inline (?i))。
  regexPatterns: [{ pattern: "(?:\\bDSv4\\bldeepseek[-\\s]?v4)(?![. _]?1)", flags: "gi",
    searchIn: "title" }],
},
"DSv4.1": {
  // 同时接受 DSv4.1 / DSV4_1 与 DeepSeek-V4.1 / DeepSeek V4_1 等写法,
  // searchIn 固定为 title: 只看标题、不看正文, 避免正文里的模型名带来噪音标签。
  regexPatterns: [{ pattern: "(?:\\bDSv4[. _]?1\\bldeepseek[-\\s]?v4[. _]?1)", flags: "gi",
    searchIn: "title" }],
},

llama: { keywords: [{ term: "Llama", searchIn: "title" }], },
```

评论区精华

本 PR 的 review 记录非常轻: `claude[bot]` 以「This pull request is from a fork — automated review is disabled.」表示 fork 来的 PR 不自动审查, `njhill` 未留文字直接 `APPROVED`, `issue` 侧只有 `njhill` 的 `/ci run` 与 `Buildkite` 机器人回复。真正的技术讨论发生在 PR body 的自证段落里, 有三点值得记录:

1. 为何 Flash 与 Pro 共用同一代标签：「Flash and Pro share one label per generation. They are the same code path in vLLM: one arch string, one config class, no variant branching anywhere in vllm/models/deepseek_v4/」——因此按变体拆标签不可能做出文件条件，只能靠标题路由，作者选择不做。
 2. 前置依赖被显式声明：「The DSv4.1 label must exist before this merges; a rule naming a nonexistent label does not create it.」这是一条容易被忽略的部署顺序约束。
 3. 测试守卫的悬置提议：「Happy to commit the title-regex cases as a test if reviewers want a permanent guard; there is no existing suite covering .github/ matcher behavior, and check-label-rules covers only the file conditions.」维护者未回应，PR 最终以单 commit 合并，标题正则仍无回归保护。
- 用负向先行断言做版本边界，并让两个打标器语义一致 (design): 方案被接受并合并；作者用 25 条真实标题在 Python re 与 JS RegExp 双引擎验证 25/25 通过，并确认没有标题会同时拿到两个标签。
 - Flash 与 Pro 是否要按变体拆分标签 (design): 决定每一代只用一个标签，不做变体级拆分；这是一条明确的“不过度设计”取舍。
 - 标题正则用例是否固化为测试 (testing): 未落地；PR 以单 commit 合并，标题正则行为仍无自动化回归保护，属于可跟进的遗留项。
 - DSv4.1 标签的前置依赖与流量回填 (other): 作为上线顺序约束记录在 PR body 中，合并后的回填由作者后续处理。

风险与影响

- 风险：
 1. 前置依赖风险：DSv4.1 标签必须已存在于仓库，否则新增的 label-dsv4-1 规则和 issue labeler 条目会指向不存在的标签（作者明确写了“a rule naming a nonexistent label does not create it”）。部署顺序搞错会让规则空转甚至报错，属于典型的配置上线顺序问题。
 2. 跨引擎正则漂移：同一份正则要同时被 mergify 的 Python re (inline (?i)、单引号 / 裸标量) 和 GitHub Actions 内的 JS RegExp (gi 标志、YAML 双引号标量里必须写成 \\b) 消费。本次已对齐，但转义层级与标志机制不同，后续任一侧单独修改就可能重新分叉，而 CI 里没有任何自动比对。
 3. 版本边界覆盖不全：(?![.]?1) 只能排除 v4 后紧跟 .1/_1 的写法，DeepSeek-V4-1（连字符分隔）仍会落到 DSv4，且该标题也不会命中 DSv4.1 的 [.]?1，属于“两边都不匹配 / 只匹配 DSv4”的盲区；未来出现 V4.2 时同样会被 DSv4 捕获，因为 lookahead 只针对“1”。
 4. 文件条件枚举式覆盖：5 条路径模式只覆盖 models、parser/tokenizers/reasoning/configs、tool_parsers、rust chatparser、tests。新增目录（如 rust/src/server/examples/、benchmark 或新子模块）下的 V4.1 改动不会被文件条件命中，只能依赖标题里带版本号，纯文档 / 示例类 PR 容易漏标。
 5. 缺少回归测试：check-label-rules 只校验文件条件，标题正则行为没有任何测试覆盖（作者已声明可补但未补），未来修 regex 时无守卫。

6. 误标的下游传导：若 DSv4/DSv4.1 后续被用于 release notes 归类或选择性 CI 触发，错误分类会随之传导到版本说明和测试调度（当前证据只表明它用于检索与 triage，此项为条件性风险）。- 影响：对推理运行时与用户完全无影响：不涉及 engine、kernel、model 或 serving 代码，改动仅在 GitHub 自动化层。对维护者与团队的影响是直接且正向的：DSv4/DSv4.1 的 PR 与 issue 流量从此可分，triage、检索与后续 release notes 归类更准确；同时需要一次性投入回填成本——约 14 个已开 V4.1 PR 与 2 个 issue 要补标签，#56228 的 DSv4 标签要纠正。对 CI 系统的影响是标签规则即时生效于合并后的新 PR/issue，本 PR 自身在合并前仍按旧规则被打上 DSv4，属于预期行为。对 DSv4.1 这条模型线的长期影响在于可见性：独立标签让 4.1 的 kernel/ 模型演进（如 #56217 跟踪 issue）能在看板上被单独统计，而不是被 V4 的流量淹没。- 风险标记：依赖标签先存在，双引擎正则易漂移，版本边界只覆盖 .1/_1，文件条件枚举式覆盖，缺少测试守卫，标题 only 匹配

关联脉络

- PR #56463 [XPU][CI] skip DeepSeek-V4.1-Flash in test_tensor_schema.py: 典型的 DSv4.1 流量样例：标题含 DeepSeek-V4.1-Flash、涉及 V4.1 模型在 CI 中的用例，正是本 PR 希望从 DSv4 中区分出来的对象。
- PR #56554 [DSV4.1] Remove compressor-aware image sentinel token padding: 同一条 DSv4.1 工作线：标题用 [DSV4.1] 前缀、文件位于 vllm/models/deepseek_v4_1/, 会被本 PR 新增的文件条件 (^vllm/models/deepseek_v4_1/) 与标题正则同时命中，可作为新规则正确性的现实校验样本。
- PR #56329 [CI] Give every Buildkite step an explicit key, and a hook to enforce it: 同一类“CI 配置 + pre-commit 守卫”治理模式：新增 tools/pre_commit/check_buildkite_step_keys.py 约束配置，而本 PR 依赖 check_label_rules.py 只覆盖文件条件、不覆盖标题正则，两者对照可见配置守卫的边界。
- PR #56501 Add @arpera to CODEOWNERS of Structured Output: 同属 .github/ 目录下的仓库治理元数据变更，说明该目录的改动通常走轻量 review + 快速合并路径。