

PR #45300 完整报告

vllm-project/vllm

[Rust Frontend][Bugfix] Forward `--shutdown-timeout` and `--disable-log-stats` to the managed Python engine

合并时间: 2026-06-12 15:39

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45300>

执行摘要

PR #45300 修复了 `vllm-rs serve` 托管模式下两个被遗漏的标志: `--shutdown-timeout` 和 `--disable-log-stats`, 确保它们从 Rust 前端正确传递到管理的 Python 引擎。此前引擎始终使用默认值, 导致优雅关闭失效和日志统计无法禁用。

功能与动机

在 `vllm-rs serve` 的托管模式 (managed mode) 中, Rust CLI splitter 将识别的前端标志保留在 Rust 侧, 因此独立启动的 Python 引擎需要显式接收部分标志。PR 描述中提到两个典型场景:

- `--shutdown-timeout`: 引擎始终以默认 `shutdown_timeout=0` (abort 模式) 运行, 收到 `SIGTERM` 后立即杀死所有进行中的请求, 即使 Rust 前端仍在保持连接以排空请求;
- `--disable-log-stats`: 无头引擎通过 `run_headless` 独立控制 `log_stats` 参数, 但该参数未被传递, 导致调度器统计信息持续输出。

因此需要将这两个标志像 `--max-model-len` 和 `--language-model-only` 一样, 显式转发到引擎的 `python_args` 中。

实现拆解

1. 修改 `ManagedEngineArgs::into_config` 签名 (`rust/src/managed-engine/src/cli.rs`): 新增 `disable_log_stats: bool` 和 `shutdown_timeout: u64` 参数。该方法是构建托管引擎配置的核心。
2. 添加转发逻辑: 在 `into_config` 中, 当 `disable_log_stats` 为 `true` 时, 向 `python_args` 追加 `--disable-log-stats`; 当 `shutdown_timeout > 0` 时, 追加 `--shutdown-timeout` 及其值。注意: `timeout=0` 时不转发, 因为 Python 引擎默认也是 0。
3. 更新调用方 (`rust/src/cmd/src/cli.rs`): `ServeArgs::to_managed_engine_config` 方法从 `self.runtime` 读取 `disable_log_stats` 和 `shutdown_timeout` 并传递给 `into_config`。
4. 添加回归测试 (`rust/src/cmd/src/cli/tests.rs`): 新增了两个测试函数, 验证参数被正确解析并出现在 `ManagedEngineConfig.python_args` 中。

`rust/src/managed-engine/src/cli.rs`

核心逻辑修改: 在 `into_config` 中新增 `disable_log_stats` 和 `shutdown_timeout` 转发。

```
// rust/src/managed-engine/src/cli.rs (head) implManagedEngineArgs{ /// Build the
managed Python - engine spawn configuration. pubfninto_config( self, model:String,
max_model_len:Option<u32>, language_model_only:bool, disable_log_stats:bool, // <-- 新增：是否禁用日志统计 shutdown_timeout:u64, // <-- 新增：优雅关闭超时秒数
handshake_port:u16, )->ManagedEngineConfig{ letmutpython_args=self.python_args; //
手动转发一些参数给 Python 引擎 ifletSome(max_model_len)=max_model_len{
python_args.push("--max-model-len".to_string());
python_args.push(max_model_len.to_string()); } iflanguage_model_only{
python_args.push("--language-model-only".to_string()); } // 如果指定了 disable_log_stats
, 则转发给引擎 ifdisable_log_stats{ python_args.push("--disable-log-stats".to_string()); }
// 如果 shutdown_timeout > 0, 则转发时间值; 默认 0 表示 abort 模式, 不转发
ifshutdown_timeout>0{ python_args.push("--shutdown-timeout".to_string());
python_args.push(shutdown_timeout.to_string()); }
ifletSome(data_parallel_size_local)=self.data_parallel_size_local{
python_args.push("--data-parallel-size-local".to_string());
python_args.push(data_parallel_size_local.to_string()); } ManagedEngineConfig{
python:self.python, model, handshake_host:self.handshake_host, handshake_port,
data_parallel_size:self.data_parallel_size, python_args, } } } 注意: shutdown_timeout =
0 时不转发, 因为 Python 引擎默认也是 0 (abort 模式), 避免冗余参数。
```

rust/src/cmd/src/cli/tests.rs

新增两个回归测试, 验证参数转发正确性。

```
// rust/src/cmd/src/cli/tests.rs (head, 新增的两个测试 ) #[test]
fnserve_args_forward_shutdown_timeout_to_managed_engine(){ // 模拟用户传入
--shutdown-timeout 60 letcli=Cli::try_parse_from([ "vllm-rs", "serve", "
Qwen/Qwen3-0.6B", "--shutdown-timeout", "60", ]) .unwrap();
letCommand::Serve(args)=cli.commandelse{ panic!("expected serve args"); }; // 验证
Rust 端解析正确 assert_eq!(args.runtime.shutdown_timeout,60); // 验证生成的托管引擎配
置包含该参数 letconfig=args.to_managed_engine_config(5555);
assert_eq!(config.python_args,vec!["--shutdown-timeout","60"]); } #[test]
fnserve_args_forward_disable_log_stats_to_managed_engine(){ // 模拟用户传入
--disable-log-stats letcli=Cli::try_parse_from([ "vllm-rs", "serve", "Qwen/Qwen3-0.6B", "--disa
ble-log-stats"]) .unwrap(); letCommand::Serve(args)=cli.commandelse{ panic!("expected
serve args"); }; // 验证 Rust 端解析正确 assert!(args.runtime.disable_log_stats); // 验证生
成的托管引擎配置包含该参数 letconfig=args.to_managed_engine_config(5555);
assert_eq!(config.python_args,vec!["--disable-log-stats"]); } 测试确保参数经过
try_parse_from 和 to_managed_engine_config 后正确出现在 python_args 中。
```

评论区精华

该 PR 没有 review 评论讨论。审核者 BugenZhao 直接批准。

风险与影响

- 低风险：变更仅涉及托管模式下两个标志的显式转发，逻辑清晰且新增代码量少。测试覆盖了正向路径，但缺少 `shutdown_timeout=0` 时不转发的测试断言（不过该行为与文档一致）。
- 影响范围：仅影响使用 `vllm-rs serve` 托管模式的用户，其他路径不受影响。合入后：
 - `--shutdown-timeout` 能正常优雅排空进行中的请求；
 - `--disable-log-stats` 能真正禁用调度器统计输出。

关联脉络

无直接关联的其他 PR。该 PR 是 Rust 前端托管模式增强的一部分，属于渐进式修复。