

PR #38641 完整报告

vllm-project/vllm

[UX] Log worker exit code when process dies unexpectedly

合并时间: 2026-07-07 20:36

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/38641>

执行摘要

- 一句话: 日志增加 worker 进程退出码
- 推荐动作: 可快速合入, 提升调试体验。

功能与动机

当 worker 进程意外退出时, 仅记录进程名无法区分失败类型 (如 OOM 被 SIGKILL、段错误 SIGSEGV 或 Python 异常退出码 1), 增加退出码可让用户和开发者直接从日志判断故障类别。

实现拆解

修改 `vllm/v1/executor/multiproc_executor.py` 中 `monitor_workers` 函数:

1. 将 `next(h.proc.name ...)` 改为获取完整进程对象 `proc = next(h.proc ...)`。
2. 在日志消息中添加 exit code 占位符 `"(exit code: %s)"`, 并传入 `proc.name` 和 `proc.exitcode`。

关键文件:

- `vllm/v1/executor/multiproc_executor.py` (模块 执行器; 类别 `source`; 类型 `core-logic`): 单文件修改, 在 worker 监控日志中增加进程退出码输出。

关键符号: `monitor_workers`

关键源码片段

`vllm/v1/executor/multiproc_executor.py`

单文件修改, 在 worker 监控日志中增加进程退出码输出。

```
# vllm/v1/executor/multiproc_executor.py
# 修改前: 仅记录进程名
# 修改后: 记录进程名和退出码, 便于区分 OOM、段错误等

def monitor_workers():
    sentinels = [h.proc.sentinel for h in workers]
    died = multiprocessing.connection.wait(sentinels)
    _self = self_ref()
    if not _self or getattr(_self, "shutting_down", False):
        logger.debug("MultiprocWorkerMonitor: shutdown already initiated")
```

```
        return
    _self.is_failed = True
    # 获取进程对象本身而非仅进程名
    proc = next(h.proc for h in workers if h.proc.sentinel == died[0])
    logger.error(
        "Worker proc %s died unexpectedly (exit code: %s), "
        "shutting down executor.",
        proc.name,
        proc.exitcode, # 新增: 输出退出码
    )
    _self.shutdown()
    callback = _self.failure_callback
    if callback is not None:
        _self.failure_callback = None
        callback()
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：纯日志格式变更，不改变控制流或异常处理逻辑，不会引发回归。
- 影响：影响范围极小，仅改进 MultiprocWorkerMonitor 的错误日志可读性，对功能无影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR