

PR #56597 完整报告

vllm-project/vllm

[CI/Build] Retry CPU image stream interruptions

合并时间: 2026-09-15 12:18

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56597>

执行摘要

- 一句话: 扩展 CPU 镜像构建重试正则, 覆盖下载流中断
- 推荐动作: 建议 5 分钟内快速浏览即可, 不必精读: 改动是一行正则, 实现意图清晰, 风险边界也已被作者在 body 中显式说明。值得关注的设计决策有三点: 一是「复用同一个有界重试框架、只扩模式不提上限」这一取舍, 把不确定性风险封在 4 次尝试 / 70 秒退避内; 二是签名集合来自真实失败日志 (8 个样本逐条反推), 而非凭空枚举; 三是它暴露了 vLLM CI 中「靠长正则做故障分类」这一模式的脆弱性——若后续再出现同类 PR, 建议推动把 BUILD_RETRY_PATTERN 拆成可测试的分类函数, 并补一份「样例行命中 / 编译错误不命中」的脚本级测试。此外, 读者应留意重试逻辑只能缓解 aicf-jf-cicd-01 这类宿主侧问题, 根因排查不应止步于本 PR。

功能与动机

PR body 明确说明这是对 #55317 引入的重试逻辑的扩展: 「This extends the retry logic introduced by #55317.」此前的分类器只认识 registry/DNS/TLS 层抖动, 认不出下载流被中断这一类失败。证据链非常具体: Main #88718 的 CPU Language shard 1 「later lost the same 1.8-GiB LLVM download with `BrokenPipeError`, but the classifier rejected it and exited after attempt 1」; 9 月 15 日 Main #88996 在 `aicf-jf-cicd-01` 上丢掉三个 CPU job (两次 `triton-cpu` clone 以 HTTP/2 cancellation/early EOF 结束, 一次 LLVM 下载以 `BrokenPipeError` 结束), Main #88997 在同一宿主同一时刻丢掉全部三个 CPU Language shard, Main #88999 又新增第八个失败。作者强调 「Each job reached the existing classifier, none matched, and each exited after attempt 1」, 并且这些失败都发生在测试之前、对应的 PR 构建此前已通过 CPU lane, 因此可归因于共享传输层而非某个源码 diff。

实现拆解

本 PR 只有 +1/-1, 全部改动集中在 `.buildkite/scripts/hardware_ci/run-cpu-test.sh` 第 59 行附近的 `BUILD_RETRY_PATTERN` 变量上, 属于「只扩模式、不动控制流」的最小改动。

1. 变更入口: `run-cpu-test.sh` 在 `docker pull ubuntu:25.04 || true` 与 `echo "--- :docker: Building Docker image"` 之后、`build_log="$(mktemp)"` 之前定义 `BUILD_RETRY_PATTERN`, 这是镜像构建重试循环唯一的判定依据, 因此改这一个字符串就能改变全部分类行为。
2. 模式扩展分三类: 新增 6 个备选项, 覆盖此前未识别的失败形态——
 - Python 下载器层: `The read operation timed out`;
 - 管道断裂层: `BrokenPipeError:.*Broken pipe` (对应

1.8 GiB LLVM 包拉取中断) ; - Git/HTTP 传输层: HTTP/[0-9.]+ stream [0-9]+ was not closed cleanly、fetch-pack: unexpected disconnect、fatal: early EOF、invalid index-pack output (对应 triton-cpu clone 被 HTTP/2 取消后的一串连带报错)。这些项被追加在原 11 项 (dial tcp、i/o timeout、failed to authorize、TLS/DNS/Connect 类错误、Failed to fetch: 等) 之后, 保持原有语义不变。

3. 控制流与预算完全保留: BUILD_MAX_ATTEMPTS=4 (1 次初始 + 3 次重试) 与 BUILD_RETRY_WAITS=(10 20 40) 未做任何改动。作者在 body 中明确「preserve the existing four-attempt limit and 10/20/40-second backoff」, 即用「有界重试」把新增模式的爆炸半径限制在最坏多等约 70 秒, 而不是让瞬时故障无限消耗 CI 资源。
4. 提交演进 (3 个 commit, 逐步加签名的过程): ce07d29 先加读超时; feddb517 追加断管签名; 3248f775 (head) 追加截断 Git 流签名。三个 commit 分别由 OpenAI Codex 与 Sherlock 协作署名, 呈现「从日志里发现一种失败 -> 补一条签名 -> 再发现下一种」的迭代式修复路径, 而不是一次性设计。
5. 测试与验证配套: 本 PR 没有源码 / 测试联动信号, 也没有新增自动化测试。作者在 body 中给出的验证手段是 bash -n、逐条比对捕获到的 timeout / broken-pipe / HTTP2 / early-EOF / index-pack 日志行确实命中 BUILD_RETRY_PATTERN、并验证一条代表性编译错误不命中该模式, 再加 shellcheck、pre-commit 与 git diff --check; 线上验证是 Buildkite #89005 (head 3248f7752f11 精确解析), bootstrap、Buildkite pre-commit、GitHub pre-commit bridge 与 CPU image gate 均通过。

关键文件:

- .buildkite/scripts/hardware_ci/run-cpu-test.sh (模块 镜像构建; 类别 infra; 类型 configuration; 符号 BUILD_RETRY_PATTERN, BUILD_MAX_ATTEMPTS, BUILD_RETRY_WAITS): 本 PR 唯一改动文件, 也是 CPU 镜像构建重试逻辑的全部实现位置: BUILD_RETRY_PATTERN 决定哪些失败允许重试, BUILD_MAX_ATTEMPTS / BUILD_RETRY_WAITS 决定重试预算, 直接决定 CPU lane 在宿主网络抖动下的存活率。

关键符号: BUILD_RETRY_PATTERN, BUILD_MAX_ATTEMPTS, BUILD_RETRY_WAITS

关键源码片段

.buildkite/scripts/hardware_ci/run-cpu-test.sh

本 PR 唯一改动文件, 也是 CPU 镜像构建重试逻辑的全部实现位置:

BUILD_RETRY_PATTERN 决定哪些失败允许重试, BUILD_MAX_ATTEMPTS / BUILD_RETRY_WAITS 决定重试预算, 直接决定 CPU lane 在宿主网络抖动下的存活率。

```
# .buildkite/scripts/hardware_ci/run-cpu-test.sh
```

```
# CPU 镜像构建阶段: 本 PR 唯一改动点, 就在下面这一条模式字符串上。
```

```
docker pull ubuntu:25.04 || true
```

```
# building the docker image
```

```
echo "--- :docker: Building Docker image"
```

```
# 以 | 分隔的瞬时故障签名表, 日志命中才允许重试。
```

```
# 前 11 项 (dial tcp ... Failed to fetch:) 由 #55317 引入, 覆盖 registry / DNS / TLS 层抖动, 本 PR 未改动;
# 后 6 项为本 PR 新增, 覆盖此前被漏判的“下载流被中断”一类失败:
# 1) The read operation timed out -> Python 下载器读超时
# 2) BrokenPipeError:.*Broken pipe -> 1.8 GiB LLVM 包拉取时管道断裂
# 3) HTTP/<ver> stream <id> was not closed cleanly -> Git / HTTP2 流被取消
# 4) fetch-pack: unexpected disconnect -> Git 传输层断连
# 5) fatal: early EOF -> triton-cpu clone 提前结束
# 6) invalid index-pack output -> 上述截断的连带报错
BUILD_RETRY_PATTERN='dial tcp|/o timeout|failed to authorize|TLS handshake
timeout|connection reset|PROTOCOL_ERROR|Could not resolve host|Temporary failure in name
resolution|dns error: failed to lookup address information|client error |(Connect)|error sending
request for url|Failed to fetch:|The read operation timed out|BrokenPipeError:.*Broken
pipe|HTTP/[0-9.]+ stream [0-9]+ was not closed cleanly|fetch-pack: unexpected disconnect|fatal:
early EOF|invalid index-pack output'

# 重试预算与退避节奏保持不变, 仍然是“有界重试”:
# 1 次初始构建 + 3 次重试, 重试前分别等待 10 / 20 / 40 秒。
BUILD_MAX_ATTEMPTS=4 # 1 initial + 3 retries
BUILD_RETRY_WAITS=(10 20 40) # seconds to wait before retry 1/2/3

build_log="$(mktemp)"

# 说明: 其后的重试循环 (读取 build_log、按 BUILD_RETRY_PATTERN 判定、命中则按
# BUILD_RETRY_WAITS 退避后重试、超出 BUILD_MAX_ATTEMPTS 即失败) 在本 PR
# 中未做任何改动,
# 因此“瞬时故障 vs 确定性故障”的判别责任全部压在上面那一条正则上。
```

评论区精华

本 PR 的讨论量很小 (3 条 issue 评论、3 条 review、0 条行级 review 评论), 有效信息集中在作者自述的验证边界与维护者的批准动作上:

claude[bot] (两次 COMMENTED): 「This pull request is from a fork — automated review is disabled. A repository maintainer can comment [@claude review](#) to run a one-time review.」——即自动 review 未真正执行, 模式边界的把关完全落在人工侧。

tlrmchlsmth: [APPROVED](#) (无正文、无行级评论)——最终以维护者批准合入, 未留下任何反对或需要修改的意见。

khluu (issue 评论): 「Human line review remains required before readiness.」以及「the external failure paths cannot be forced deterministically」——作者自己承认新增行必须人工逐行确认, 且无法在本地确定性复现这些外部网络失败路径。

khluu (issue 评论): 「This head covers the eight pre-test failures now observed on physical [aicf-jf-cicd-01](#) across unrelated mains [#88996/#88997/#88999](#): seven truncated Git/HTTP clone streams plus one Triton LLVM [BrokenPipeError](#).」——这是本 PR 最有价值的一条证据归纳, 说明签名集合是从真实失败样本反推出来的, 而非凭经验穷举。

另外 body 中提到一份短暂开启的重复草稿 #56938 在验证前关闭，其独有的流签名被合并进本 PR，体现了同类 CI 修复的去重与收敛。

- 新增正则的误匹配边界需人工逐行确认，但自动 review 不可用 (correctness): 改动最终由 tlrnchlsmith 批准合入，没有反对意见，但也没有任何行级讨论记录；模式边界的确认完全依赖作者自测 (bash -n、shellcheck、逐条样例行命中、代表性编译错误不命中)。
- 外部失败路径无法确定性复现，验证只能依赖线上 CI (testing): 以线上 CI 通过加历史日志抽样作为等价验证；未新增可离线复现的回归测试，假阳性（误把确定性失败当瞬时故障）与假阴性（日志换行导致漏匹配）仍无自动化守护。
- 只扩模式不改预算：复用 #55317 的有界重试框架 (design): 设计上选择以「有界重试」控制爆炸半径，代价是分类器变成一条 17 个备选项的超长正则，维护性与可测试性下降。

风险与影响

- 风险：
 1. 正则误匹配、掩盖确定性失败：fatal: early EOF 与 invalid index-pack output 既可能来自瞬时断流，也可能来自仓库侧确定性故障（例如 pin 的 commit 不可达、镜像源被改写）。此时重试不会成功，只会把失败延迟约 70 秒并重复拉取大体积产物（LLVM 包约 1.8 GiB），排障时容易误判为「网络抖动」。
 2. 缺少自动化守护：整个分类器是一条横跨 17 个 | 备选项的超长单行字符串，没有任何测试断言「样例行命中 / 编译错误不命中」。后续任何人格式化或改写该行（例如被 shellcheck 提示拆分、误改转义）都可能静默破坏分类能力，而 CI 只会在下次真实断流时才暴露问题。
 3. 正则细节脆弱性：BrokenPipeError:.*Broken pipe 依赖两个片段出现在同一行，日志换行输出时会漏匹配；HTTP/[0-9.]+ stream [0-9.]+ was not closed cleanly 中的 [0-9.]+ 语义偏宽（也会匹配 1.2.3），虽然实际日志中无害，但说明模式是从样本行直接抄录而非按语义构造。
 4. 治标不治本：失败集中在 aicf-jf-cicd-01 这一台宿主，属于宿主网络 / 传输层质量问题，重试只是提高容错概率，未解决根因；若宿主持续劣化，重试会放大 CI 总耗时。
 5. 安全 / 兼容性风险可忽略：改动不涉及凭据、镜像来源或产物校验逻辑，不改变构建内容，也不影响推理路径。- 影响：影响面限定在 CI 的 CPU 镜像构建阶段（.buildkite/scripts/hardware_ci/run-cpu-test.sh 一处），不涉及 vLLM 运行时、模型加载或服务接口。
- 对开发者：直接收益是 CPU Language 等 lane 因下载流中断而产生的假失败会显著减少（作者已用 8 个真实失败样本验证覆盖），减少重跑 CI 与人工排查成本；代价是新增签名被命中时会多消耗 10/20/40 秒退避与重新拉取时间。
- 对 CI 系统：有界重试（4 次上限）意味着最坏情况下单个 job 额外耗时约 1 分钟级别，不会出现无限重试耗尽构建机资源的情况；但宿主 aicf-jf-cicd-01 仍可能成为整批 CPU job 同时失败的瓶颈点，本 PR 未消除该单点依赖。
- 对团队：这是一次典型的「把运维经验固化成分类器签名」的动作，会形成一条持续维护成本——后续每出现一类新的传输中断签名都需要再加一项，长期看更适合改造成带样例的测试化分类器。

- 风险标记：缺少自动化回归测试，长正则误匹配风险，可能掩盖确定性失败，仅缓解宿主网络问题

关联脉络

- PR #56735 [CI] Extend DSv4 engine readiness timeout on main: 同属「CI 基础设施抖动容忍度调整」这一条线：一个放宽超时阈值，一个放宽可重试的错误签名集合，动机都是把宿主 / 环境波动与真实代码回归区分开。
- PR #56921 [ROCm][CI] Prepare TheRock image for CI: 同样是容器镜像构建相关的 CI 修复 (docker/Dockerfile.rock*)，与本 PR 的 CPU 镜像构建重试属于同一类「镜像构建链路可靠性」工作。
- PR #56169 [CI] Check target branch freshness before starting CI: 同在 .buildkite / GitHub workflow 脚本层做 CI 流程健壮性改造，体现近期 CI 侧「把失败提前拦截或自动恢复」的持续投入。
- PR #56819 [CI] Pass the scale-out endpoint flag in EC E2E: 同样是单文件 CI 脚本的小修小补，与本 PR 的 +1/-1 改动形态和「让 CI 稳定跑通」的目标一致。