

PR #56324 完整报告

vllm-project/vllm

[CI] Shard V1 KV Connectors 1→4

合并时间: 2026-09-15 17:58

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56324>

执行摘要

- 一句话: V1 KV Connectors 作业 1→4 分片, 消除 45 分钟超时取消
- 推荐动作: 值得精读, 尤其对 CI 工程团队。虽然只是一份 9 行 YAML 变更, 但它的方法论可复用: 先挖掘历史 CI 轨迹得到逐用例权重, 再复现 pytest-shard 的哈希分配函数做模拟选 N, 最后用 scoped build 在真实硬件上验证——整个决策链有数据、有模型、有实测闭环, 而不是靠拍脑袋定分片数。另一个值得借鉴的点是先把问题定性为超时边界失败 (而非测试不稳定) 再动手, 避免误修。可关注后续是否有将 timeout 从 45 分钟下调至约 25 分钟的 follow-up PR。

功能与动机

PR body 明确指出, 这是 #52346 拆分旧组合作业 V1 Core + KV + Metrics (p90 约 1h7m) 后唯一仍超预算的继任作业: V1 KV Connectors 的 pytest 耗时约 37.5 min (v1/kv_connector/unit) 加约 4 min (v1/ec_connector/unit), p90 作业 44.2 min (Sep 8-10 测量), 距离 45 分钟上限几乎没有余量。khluu 在主分支复现了问题的真实形态: 在 teardown 阶段测试已全绿却因越界被取消 / 超时 (#88841、#88854 等 5 次运行), 并触发 dashboard alert 1181, 说明这确实是一个需要提升并行度而非修复测试的问题。

实现拆解

1. 变更入口: 唯一改动文件是 .buildkite/test_areas/misc.yaml 中的 :nvidia: (H200 MIG 35GB) V1 KV Connectors 步骤块 (key 为 v1-kv-connectors), 该步骤负责运行 v1/kv_connector/unit 与 v1/ec_connector/unit 两类单元测试。
2. 核心改动: 为该步骤增加 parallelism: 4, Buildkite 会把步骤展开为 4 个并发实例; label 改为 V1 KV Connectors Shard %N 以便区分; 两条 pytest 命令分别追加 --num-shards=\$\$BUILDKITE_PARALLEL_JOB_COUNT 与 --shard-id=\$\$BUILDKITE_PARALLEL_JOB, 由 pytest-shard 按 sha256(nodeid) % N 在同一个 shard-id 下同时切分两个测试目录。AMD mirror 继承同样的命令与标签模式。
3. 分片数选取方法: 作者从 13 次已通过 CI 轨迹中挖掘出全部 1444 个用例 (1165 + 279) 的逐用例运行时间, 并复现 pytest-shard 的精确分配函数 (rootdir 相对 nodeid、保留 tests/ 前缀、little-endian 哈希), 以 collect-only 精确复现 4 分片分解为 292/309/301/263 项验证了建模正确性; 对比 N=2..6 模拟耗时后, N=4 是既能满足中位数权重、也能容忍最坏观测权重 (最慢分片约 13.8 分钟) 的最小值。

4. 验证闭环与配套：先后在旧 head (#88879) 与 rebase 后的当前 head (#88937) 上跑只选中 v1-kv-connectors 的 scoped build，4 分片全部一次通过，墙时间 11.8–20.4m；与主分支未分片对照 45m28s 形成明确对比。45 分钟超时上限刻意保持不变，待真实分片数据积累后再下调至约 25 分钟；无测试文件、命令行或覆盖率改动。
5. 固定开销评估：install-kv-connectors.sh 步骤仅约 0.3 分钟（不同于 language 作业的 mamba 安装），因此每个 shard 的固定开销可忽略；AMD/XPU 无硬件可用，misc.yaml 的 AMD mirror 只继承了命令而未经实测。

关键文件：

- .buildkite/test_areas/misc.yaml（模块 CI 配置；类别 config；类型 configuration）：唯一的变更文件，定义了 V1 KV Connectors 作业及其 AMD 镜像的分片策略：parallelism: 4、label 增加 Shard %N、两条 pytest 命令追加 --num-shards/--shard-id。该文件同时承载了 #52346 拆分后的 5 个 V1 misc 继任作业，本 PR 只动其中一个步骤，diff 范围极小但决策依据充分。

关键符号：未识别

评论区精华

讨论以验证闭环为主，无实质性技术争论：(1) khluu 先后提供主分支 5 次复现证据 (#88841、#88854、#88912、#88913、#88916)，核心观察是所有 287 个测试全部通过后才在 teardown 阶段越过 45 分钟边界被取消，没有任何测试断言失败，并触发 dashboard alert 1181，从而把问题性质钉死为作业超时预算不足；(2) 作者与 khluu 合作完成两轮 scoped build 实测——旧 head #88879 四分片 16.9–20.4m 全绿，rebase 后 #88937 四分为片 11.8–15.4m 全绿，对比未分片主分支 #88854 的 45m28s 才勉强越界；(3) 作者在 body 中承认 N=4 的取值依赖测试集合稳定，新增或重命名测试会经哈希重排分片成员，需要周期性重新推导。唯一未解决的顾虑是 AMD/XPU 无硬件实测。

- 主分支复现：超时发生在 teardown 阶段而非测试失败 (testing)：把问题定性为作业超时边界问题，分片是唯一的窄修复；khluu 明确表示不会另开超时类 PR 与其竞争。
- 分片实测验证：4 分片墙时间对比未分片 45m28s (performance)：1→4 拆分在真实硬件上验证通过，最慢分片也远低于 45 分钟边界；据此保留 45 分钟超时，留待后续下调至约 25 分钟。
- 分片数选择与可维护性：为什么是 4 而不是 3 或 5 (design)：接受 N=4 与随测试演变定期重推的代价；该结论在后续实测中被验证。

风险与影响

- 风险：
 1. 分片成员漂移：pytest-shard 按 nodeid 的 sha256 哈希分配，任何新增、重命名或删除测试都会重排分片成员，可能重新出现单分片倾斜甚至再次逼近超时；作者在 body 中已承认需要周期性重推分片数。
 2. AMD/XPU 侧未验证：misc.yaml 中 AMD mirror 只是继承命令与标签，没有 MI300 实测数据；虽然测试集合与 NVIDIA 相同，但不排除个别重用例在 AMD 上耗时差异导致的倾斜。

3. CI 资源占用上升：parallelism 4 意味着同一时刻占用 4 台 H200 运行同一作业，资源成本和调度压力约为原来的 4 倍。
4. 超时上限未同步下调：45 分钟 timeout 保持不变，实测最慢分片仅 20.4 分钟，余量充足但偏低的上限会掩盖未来测试膨胀造成的回归；body 计划待数据稳定后下调至约 25 分钟，但尚未形成后续 PR。
5. 回归面：纯 YAML 配置变更，不触及任何产品代码或测试断言，风险面很小。- 影响：对用户无任何直接感知，属于纯 CI 基础设施优化。对 CI 流程：该作业 p90 从 44.2 分钟降到约 20 分钟以内，且不再出现测试全绿却被超时取消的假失败，主分支稳定性和开发者信心都有提升。对团队：v1/kv_connector 相关功能线（NIXL、KV offload、EC connector）近期迭代密集（如 #56104、#56640、#56955 等），分片后每次提交的 CI 反馈周期显著缩短；同时也为后续把作业超时从 45 分钟下调到约 25 分钟留出了空间。AMD/XPU 流程不受影响，仍是旧合并范围。- 风险标记：分片成员随测试增删漂移，AMD/XPU 无实测数据，超时上限暂未下调，CI 资源占用 4 倍

关联脉络

- PR #52346（标题不可用）：PR body 说明旧组合作业 V1 Core + KV + Metrics（p90 约 1h7m）在 #52346 中按命令拆分，本 PR 的 V1 KV Connectors 步骤正是其继任作业之一；当时 KV Connectors 仍为单分片约 44 分钟，本 PR 是紧随其后的分片 follow-up。
- PR #56313（标题不可用）：PR body 将 #56313 列为语言类作业的 companion sharding 工作，本 PR 复用其已建立的 Shard %N 标签与 --num-shards/--shard-id 模式。
- PR #56316（标题不可用）：PR body 将 #56316 列为多模态作业的 companion sharding 工作，与本 PR 共同构成 CI 分片系列。
- PR #56291（标题不可用）：PR body 提到 #56291–#56294 是哈希分片之外的结构性替代方案（按目录整拆），与本 PR 的哈希分片方案互为对比和备选。