

PR #52325 完整报告

vllm-project/vllm

[CI] Shard Humming A100 eval

合并时间: 2026-08-17 17:03

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/52325>

执行摘要

- 一句话: Humming A100 评估拆 3 路并行, CI 耗时降为约三分之一
- 推荐动作: 值得快速浏览, 作为 Buildkite step 并行分片的低成本范例。核心可借鉴点有三: 一是按实测耗时 (而非配置数量) 做均衡分片; 二是用 BUILDKITE_PARALLEL_JOB 静态选择清单文件, 避免在命令行里做复杂动态解析; 三是 PR body 中完整的验证方法论 (14/14 唯一性与覆盖率检查、执行集合与 shard 清单逐项比对、墙钟与 GPU 消耗双维度评估)。建议后续用脚本从单一 config.txt 生成 shard 文件, 或增加 CI 检查保证 4 份清单内容始终一致, 消除手工同步漂移。无需精读源码, 因为本次没有源码变更。

功能与动机

PR body 明确指出瓶颈与目标: `lm-eval-humming-f16-a100` took 64.3 minutes in full CI build 83851. Its 14 model configurations ran serially even though each evaluation is independent. 即 14 个模型配置互相独立, 却被放在同一个 job 里串行执行, 白白浪费并行资源。PR 同时声明实施前已按 step key 与 Humming/A100 分片关键词搜索过开放 PR, 避免重复工作。拆分的量化目标是让每个 shard 的测试负载落在 30 分钟以内, 实测基线为 20.20、21.47、21.60 分钟, 留有约 8 分钟余量。

实现拆解

1. 变更入口: .buildkite/test_areas/lm_eval.yaml 中的 lm-eval-humming-f16-a100 步骤。为该步骤增加 parallelism: 3, label 追加 %N 占位符, 命令从固定 config.txt 改为 config-a100-shard-\$\$BUILDKITE_PARALLEL_JOB.txt。这样 Buildkite 会并发启动 3 个 job, BUILDKITE_PARALLEL_JOB 依次为 0、1、2, 各 job 据此读取对应分片清单; timeout_in_minutes (75)、device (a100)、optional、num_devices、source_file_dependencies 等属性保持不变。
2. 配置清单拆分: 在 tests/evals/gsm8k/configs/humming/ 下新增 config-a100-shard-0/1/2.txt 三个清单, 把原 14 个配置按构建 83851 的逐配置耗时均衡分配为 4、5、5 个, 每个配置只出现一次。这些文件被 pytest 的 --config-list-file 直接消费, 属于机器读取的 CI 配置; 原 config.txt 保留, 继续给 H100 步骤使用。
3. 验证与配套: 未改动产品代码和测试代码。PR body 给出了完整验证: 目标构建 83894 三个 shard 全部通过, 执行集合与清单逐项比对、14/14 配置全覆盖且无重复无遗漏; 墙钟 24.09、22.07、24.44 分钟, 关键路径 24.44 分钟对比基线 64.27 分钟提速 2.63 倍; 总 GPU 时间 70.59 vs 64.27 GPU 分钟 (+9.8%); uvx pre-commit 与 git diff --check 通

过。

关键文件：

- `.buildkite/test_areas/lm_eval.yaml`（模块 CI 流水线；类别 config；类型 configuration）：核心变更文件。为 `lm-eval-humming-f16-a100` 步骤增加 `parallelism: 3`，并让命令按 `BUILDKITE_PARALLEL_JOB` 序号读取对应 shard 清单，是本次 CI 分片的控制点。
- `tests/evals/gsm8k/configs/humming/config-a100-shard-0.txt`（模块 评估配置；类别 config；类型 configuration）：新增的 0 号分片清单，包含 4 个模型评估配置（`experts-int8`、`gpt-oss-20b`、`GPTQ-Int4`、`FP8-block`），是三个 shard 中配置数最少的一份，用于平衡耗时较长的专家量化变体。虽按扩展名可视为文本文件，但被 `pytest` 的 `--config-list-file` 直接消费，本质是 CI 配置清单。
- `tests/evals/gsm8k/configs/humming/config-a100-shard-1.txt`（模块 评估配置；类别 config；类型 configuration）：新增的 1 号分片清单，包含 Qwen3 系列多种量化格式（`NVFP4`、`w8a8`、`FP8`、`AWQ`、`MXFP8`）的 5 个配置，与 shard-2 配置数相同以均衡负载。
- `tests/evals/gsm8k/configs/humming/config-a100-shard-2.txt`（模块 评估配置；类别 config；类型 configuration）：新增的 2 号分片清单，包含 Qwen3.5-35B-FP8、Nemotron 30B、MXFP4A16 等 5 个配置，与 shard-1 同为 5 个配置，保证三路并行负载接近。

关键符号：未识别

评论区精华

本 PR 没有人工 review 讨论：`comments_count` 与 `review_comments_count` 均为 0。唯一记录是 `claude[bot]` 的自动回复，说明该 PR 来自 fork、自动审查被禁用，维护者可手动触发 `@claude review` 进行一次性评审。由于最终由 `vllm-bot` 合入，可以推断已通过维护者确认。PR body 反而承担了主要的「讨论」角色，以大量实测数据论证了分片方案的有效性，包括配置唯一性检查、执行集合比对、墙钟与 GPU 消耗对比等。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 配置漂移风险：A100 的三个 shard 清单是从 `config.txt` 手工拆出的，而 `config.txt` 仍被 H100 步骤使用，形成 4 份需要同步的清单；将来新增或删除 Humming 配置时若只改一份，会出现 A100 漏配置或 H100 与 A100 不一致，且 CI 不一定能及时发现（步骤为 `optional: true`）。
 2. 资源占用上升：`parallelism: 3` 使该步骤同时占用 3 台 A100，总 GPU 时间增加约 9.8%；在 GPU 队列拥挤时，并行 job 可能互相等待，实际收益会打折扣。
 3. 分片平衡依赖历史耗时：当前 1.11x 不平衡度与 2.37 分钟 max/min 差异均基于构建 83851 的观测，模型更新或数据变化后耗时分布会漂移，最大 shard 将成为新的关键路径。

4. 排障复杂度: 3 个 job 共享同一 step key 与 label 前缀, 失败时需通过 BUILDKITE_PARALLEL_JOB 序号定位具体 shard。 - 影响: 影响范围集中在 CI 基础设施: lm_eval.yaml 的步骤定义与 tests/evals/gsm8k/configs/humming/ 下的清单文件, 不涉及任何产品代码、模型行为或用户可见接口。对团队的主要收益是 Humming 量化回归的反馈周期从约 64 分钟缩短到约 24 分钟, 且每次触发时并行消耗 3 个 A100 job; 该模式可低成本复制到 lm-eval-humming-act-a100 和 H100 步骤。对仓库外部用户无感知。 - 风险标记: 分片清单手工维护易漂移, 并行 job 增加 GPU 总占用约 9.8%, 关键路径受最大 shard 制约

关联脉络

- PR #52570 [CI/Build] Reduce more duplicate runner startup in tests: 同一批 CI 提速工作: 合并池化测试用例以减少 runner 启动, 与本 PR 用并行分片缩短关键路径方向一致, 共同反映团队对 CI 时间的持续优化。
- PR #52417 [CI/Build] Avoid duplicate runner startup for multimodal test: 同为 CI 测试时间优化, 通过合并 VLM 尺寸用例降低 runner 启动开销; 与本 PR 的并行化思路互补。
- PR #52496 [CI] Fit small KV-offload evals within shared memory: 同为 evals 的 CI 基础设施调整, 说明 lm_eval/gsm8k 评估步骤是近期 CI 治理的重点区域。