PR #7141 完整报告

PaddlePaddle/FastDeploy

[BugFix] prevent requests from entering running state without a slot

合并时间：2026-04-03 14:07

原文链接：http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7141

执行摘要

- 一句话：修复调度器在请求状态转换时槽位计数不一致的 bug，防止请求无槽位进入运行状态。
- 推荐动作：该 PR 值得精读，特别是对于负责调度器模块的工程师。重点关注 resource_manager_v1.py 中新的槽位计数逻辑设计，这是调度器正确性的关键保障。建议结合近期调度器相关的 PR（如 #7152、#7129）一起阅读，理解调度器状态的完整演进。

功能与动机

根据 PR body 描述，修复调度器槽位计数不一致的问题。当请求处于运行、待中止、待重新调度或已抢占回等待队列等状态时，调度器可能过度接纳请求，即使有效占用槽位已达到 max_num_seqs，仍将新请求移入 running 状态。这会导致调度器在接近最大并发数时出现状态不一致。

实现拆解

修改集中在两个核心调度文件：

1. fastdeploy/engine/sched/resource_manager_v1.py：在 _allocate_decode_and_extend 函数中，将槽位检查条件从仅检查 running 队列长度，扩展为同时统计 running 队列、待重新调度请求集合 (to_be_rescheduled_request_id_set)、待中止请求集合 (to_be_aborted_req_id_set) 以及等待队列中状态为 PREEMPTED 的请求数量。只有当这些总数小于 max_num_seqs 时，才允许新请求从 waiting 进入 running。
2. fastdeploy/engine/common_engine.py：在 _fetch_request 函数中，移除对 RuntimeError("cannot schedule new futures after shutdown") 的静默处理，改为直接重新抛出异常，确保调度器关闭时的错误能被显式暴露。

关键文件：

- fastdeploy/engine/sched/resource_manager_v1.py（模块 Scheduler）：调度器资源管理的核心文件，修改了请求从 waiting 进入 running 的准入条件，修复了槽位计数不一致的关键 bug。
- fastdeploy/engine/common_engine.py（模块 Engine）：通用引擎文件，修改了异常处理逻辑，确保调度器关闭时的错误能被正确暴露，避免静默失败。

关键符号：_allocate_decode_and_extend, _fetch_request

评论区精华

Review 评论中未提供具体讨论内容。从提交历史看，作者通过三次提交逐步完善了修复：第一次提交添加了基本防护逻辑；第二次提交增加了对 to_be_aborted_req_id_set 的计数；第三次提交增加了对 waiting 队列中 PREEMPTED 状态请求的计数。这表明实现过程中考虑了不同状态请求对槽位占用的影响。

- 调度器槽位计数逻辑的完整性 (correctness): 最终实现同时统计 running 队列、待重新调度集合、待中止集合以及等待队列中 PREEMPTED 状态的请求，确保所有占用槽位的请求都被计入。
- 调度器关闭时的错误处理 (correctness): 移除对 "cannot schedule new futures after shutdown" 异常的捕获和忽略，改为重新抛出，确保调度器关闭时的错误能被显式暴露。

风险与影响

- 风险：1. 回归风险：修改了调度器的核心准入逻辑，如果新的槽位计数逻辑有误，可能导致调度器过度保守（拒绝本可调度的请求）或过度激进（仍允许超限调度）。 2. 性能影响：新增了 sum([req.status == RequestStatus.PREEMPTED for req in self.waiting]) 计算，在 waiting 队列较大时可能增加少量开销。 3. 兼容性：不涉及 API 变更，对用户透明。 4. 测试覆盖：PR body 中提到未添加专用单元测试，仅依赖现有测试。Codecov 报告显示有 1 行代码缺少覆盖，需要关注。
- 影响：1. 对系统：修复了调度器在高并发、频繁状态转换场景下的槽位计数 bug，提升了调度准确性和系统稳定性。 2. 对用户：在请求频繁被重新调度、中止或抢占的场景下，调度器行为更可预测，避免因过度调度导致的性能下降或错误。 3. 对团队：修改涉及调度器核心逻辑，需要团队成员理解新的槽位计数规则，并在相关测试中验证。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #7152 [Feature] Support chunk prefill disabled in scheduler v1: 同样修改了 resource_manager_v1.py 文件，涉及调度器 V1 的功能扩展，可结合理解调度器模块的演进。
- PR #7129 [Feature] Fix mixed cache-aware: 同样涉及调度器修复，虽然修改文件不同（golang_router），但都关注调度逻辑的正确性，可对比学习。
- PR #7127 [Others]add unit test: 恢复了 V1 版本缓存管理和资源调度的单元测试，与本次调度器修复相关，建议关注测试覆盖情况。