

# PR #37249 完整报告

sgl-project/sglang

fix(gateway): bump wfaas to 1.0.2 so ContinueNextStep unblocks dependents

合并时间: 2026-09-01 02:30

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/37249>

## 执行摘要

- 一句话: wfaas 钉版升至 1.0.2, 修复 ContinueNextStep 依赖死锁
- 推荐动作: 值得精读, 但重点不在代码而在方法: PR body 对 wfaas 调度器 bug 的根因追踪 (tracker 记录与调度器信号两个状态源的错位、workflow 形状分析、生产影响还原) 是教科书级的依赖 bug 分析; 审核人的独立验证和“触发条件比描述更窄”的纠正也展示了良好的协作模式。可关注的设计决策包括: 在精确钉版纪律内选择最小修复版本、用判别性测试矩阵双向验证 pin 版本、以及如何为钉版依赖建立定期回访机制 (如 Dependabot 或专门巡检)。建议后续跟进把判别测试入库, 并评估 1.1.0 与 #32322 的落地。

## 功能与动机

PR body 指出 sgl-model-gateway 将 wfaas 钉在 =1.0.0, 而 wfaas 1.0.0/1.0.1 存在调度 bug: ContinueNextStep 失败的步骤被 tracker 记为 skipped, 但发给调度器的信号是 StepResult::Failure, 调度器只在 Success/Skip 时入队下游步骤, 最终走进死锁分支把整个 workflow 置为 Failed。作者在生产环境 fork 中遇到“rolled pod 从未被拾取, 路由器一直服务陈旧 worker 集合直到重启”。根因是关联 Issue #19224 为 CI 稳定性把 11 个内部 crate 全部改钉 =1.0.0, 恰好发生在 wfaas 1.0.2 发布后一天, 把上游修复挡在门外; 作者原话称“那是 ContinueNextStep 声明在这个步骤上要达成的效果的反面”。

## 实现拆解

1. 变更入口: 唯一改动是 sgl-model-gateway/Cargo.toml 第 87 行的 wfaas 依赖, 从 =1.0.0 改为 =1.0.2, 保持 #19224 引入的精确钉版纪律不变; 没有修改任何 Rust 源码、测试或 CI 配置, 仓库也没有提交 Cargo.lock 需要同步。
2. 版本选择依据: 1.0.2 是携带修复的最小版本。作者逐一列出 1.0.0→1.0.2 的公开 delta: src/engine.rs 包含核心修复 (ContinueNextStep 分支改为发送 (StepResult::Skip, true))、pending\_check 索引去重 (避免 depends\_on\_any 步骤在多个依赖完成时重复启动) 以及若干 clippy 属性与内联 format 参数; src/event.rs 仅 clippy 属性; 其余为版本号和 README (1.0.1 加入)。没有任何公开 API 变化。1.1.0 是当前最新版本但 delta 更大, 作者选择最小修复集。
3. 验证方式: 作者写了三个判别测试, 针对未修改的 wfaas 依赖、仅变动 pin 版本运行: 上游自带控制用例 (无依赖步骤在 ContinueNextStep 后仍运行)、扩展后带 dependent 的判别用例、以及 gateway AddWorker 形状 (detect\_connection\_mode → discover\_metadata(ContinueNextStep) → create\_worker → register\_workers) 的用例。

结果矩阵显示 1.0.0/1.0.1 在两个判别用例上 FAIL、1.0.2/1.1.0 通过，双向判别有效；  
cargo check --all-targets 在工具链 1.90 下通过。

4. 测试与配套：判别测试未入库，作者明确表示可按需补到 sgl-model-gateway/tests/；本变更不涉及模型推理路径，accuracy 与 speed 测试均不适用。

关键文件：

- sgl-model-gateway/Cargo.toml（模块 网关依赖；类别 config；类型 configuration）：唯一变更文件：将 wfaas 精确钉版从 =1.0.0 提升到 =1.0.2，解锁上游 ContinueNextStep 调度死锁修复，同时保持 #19224 引入的精确钉版纪律。

关键符号：未识别

## 关键源码片段

### sgl-model-gateway/Cargo.toml

唯一变更文件：将 wfaas 精确钉版从 =1.0.0 提升到 =1.0.2，解锁上游 ContinueNextStep 调度死锁修复，同时保持 #19224 引入的精确钉版纪律。

```
# ===== sgl-model-gateway/Cargo.toml • 内部 crate 精确钉版段（节选） =====
```

```
# 内部 crate 一律使用「精确钉版 =」而非「兼容范围 ~」：
# 这是 #19224 的决定，因为 Cargo.lock 被 gitignore，CI 每次全新解析，
# 宽松范围会拉到带破坏性 API 变化的 patch 版本（如 smg-wasm 的
# WasmModuleManager::new() 签名变更），导致 Docker CI 编译失败。
```

```
reasoning-parser = "=1.0.0"
openai-protocol = { version = "=1.0.0", features = ["axum"] }
tool-parser = "=1.0.0"
llm-tokenizer = "=1.3.2"
smg-auth = "=1.0.0"
```

```
# wfaas: 1.0.0 是 #19224 当时钉住的版本，但 1.0.0/1.0.1 存在调度死锁——
# 失败动作声明为 ContinueNextStep 的步骤，tracker 记为 skipped，
# 发往调度器的信号却是 StepResult::Failure，导致 depends_on 它的步骤
# 永不入队，最终整条 AddWorker  workflow 被死锁分支打成 Failed，
# worker 永远无法注册。
# 1.0.2（本 PR 目标版本）把该分支改为发送 (StepResult::Skip, true)，
# 依赖步骤得以被调度；该升级无任何公开 API 变化，兼容风险极低。
```

```
wfaas = "=1.0.2"
```

```
data-connector = "=1.0.0"
smg-mcp = "=1.0.0"
smg-wasm = "=1.0.0"
smg-mesh = "=1.0.0"
```

## 评论区精华

审核人 ShangmingCai 给了两次 APPROVED，第二次附带了独立验证笔记，核心交锋如下：

- 根因独立核验：审核人在 vendored 的 wfaas-1.0.0/src/engine.rs 中确认，失败信号在 ContinueNextStep 分支把步骤映射为 skipped 之前就已按 result 计算为 StepResult::Failure；调度器只在 Success | Skip 时入队依赖步骤，随后走进死锁分支置 WorkflowStatus::Failed。create\_local\_worker\_workflow() (steps/worker/local/mod.rs:120) 正是该形状，discover\_metadata 失败后 register\_workers 不可达。
- 触发条件修正：PR 描述称“新启动 worker 的 /get\_server\_info 慢到 10s 超时即可触发”，审核人指出 HTTP 路径中 DiscoverMetadataStep 用 .ok() 加 unwrap\_or\_else 吞掉了 HTTP/gRPC fetch 错误、返回 Ok(Success) 空标签，普通失败 / 拒绝的 /server\_info 并不会触发；真正触发的是三轮 10s 步超时全部耗尽（冷启动 pod 的挂起连接）或 connection\_mode 上下文错误——真实但不罕见，只是比描述更窄。
- 行为变化面：审核人顺带检查了唯一的行为变化面 create\_mcp\_registration\_workflow，它也是一串 ContinueNextStep 步骤，bump 后其下游步骤的调度行为会从“永不执行”变为“按声明执行”（该段评论在材料中截断，完整结论未呈现）。
  - ContinueNextStep 死锁根因的独立源码核验 (correctness): 认可 PR 的根因判断，1.0.2 的修复 ((StepResult::Skip, true)) 确实解决问题，bump 方向正确。
  - 触发条件比 PR 描述更窄：HTTP fetch 错误其实被吞掉 (correctness): 故障真实但不罕见，比“一次抖动 HTTP 调用”更窄；不影响 bump 正确性，仅修正描述框架。
  - bump 的行为变化面：create\_mcp\_registration\_workflow (design): 未完整呈现，作为后续观察项；建议升级后对该 workflow 做一次端到端验证。

## 风险与影响

- 风险：兼容性风险低：审核人验证 1.0.0→1.0.2 的依赖要求字节级一致，[lints] workspace = true 这类 lint 配置对 registry 依赖编译没有影响。行为变化面主要有三点：一是 1.0.2 对 pending\_check 索引的去重会减少 depends\_on\_any 步骤的重复启动次数，若 gateway 中有步骤隐式依赖旧的重复启动行为需要观察；二是 create\_mcp\_registration\_workflow 这条 ContinueNextStep 链的下游步骤将开始真正被调度，属于正向解锁但未充分验证；三是本 PR 暴露的工程风险——精确钉版 + 被 gitignore 的 Cargo.lock + 无依赖回访机制，会让上游 patch 级修复被静默挡住一整年，1.1.0 是否值得跟进也尚未评估。另外，作者编制的判别性回归测试没有入库，后续再次升级或回退 wfaas 时没有自动防线。
- 影响：影响范围集中在 sgl-model-gateway 的 worker 注册路径：AddWorker workflow 不再因可选的 discover\_metadata 超时整体失败，滚动发布或冷启动的 worker 能被可靠注册，直接消除生产环境服务陈旧 worker 集合的问题；create\_mcp\_registration\_workflow 的行为也会随之解锁。对 SGLang 推理内核、调度器、模型精度均无影响。对团队而言，该案例提醒维护者：精确钉版策略需要配套依赖回访或自动化升级机制，否则上游修复会被长期阻断；#32322 落地后，恢复机制会更完整。总体影响程度中等，但直接命中一个真实生产故障。
- 风险标记：精确钉版阻隔上游补丁修复，周边 workflow 行为变化未评估，无 in-tree 回归测试

## 关联脉络

- PR #19224 fix(gateway): pin internal crate dependencies to exact versions: 直接前因：该 PR 把 wfaas 等 11 个内部 crate 从 ~1.0.0 改钉 =1.0.0 以稳定 Docker CI，恰好挡掉了次日发布的 1.0.2 上游修复；本 PR 在其钉版纪律内做定点升级。
- PR #32322 [k8s discovery] periodic LIST reconcile for k8s service discovery: PR body 中明确提到的互补 PR（作者自己的开放 PR）：其 forget\_lost\_registration 正是为“AddWorker 提交后 workflow 失败”的状态设计，与本 PR 可独立落地、互为补充；确切标题未在本次材料中完整给出。