

PR #2389 完整报告

radixark/miles

fix: avoid Mooncake metrics port collisions

合并时间: 2026-08-13 15:32

原文链接: <http://prhub.com.cn/radixark/miles/pull/2389>

执行摘要

- 一句话: 默认 metrics 端口改为 0, 避免 Mooncake 启动端口碰撞
- 推荐动作: 值得快速浏览: 这是一个教科书式的“固定端口 → 端口 0 原子分配”修复, 改动极小但解决了多进程环境下的真实启动故障。值得关注的设计点是让就绪判定与可选项端口解耦、只等待核心 RPC 端口, 以及用测试锁定默认命令格式。若团队有依赖 50052 拉取指标的自动化, 需要同步排查适配。

功能与动机

PR body 描述了完整症状与复现路径: Mooncake 启动时 RPC 端口 50051 正常, 但 admin 端口 50052 绑定失败即退出, launcher 报 RPC readiness timeout; 复现方式是先 hold 住 127.0.0.1:50052 再启动 mooncake_master。根因是 MOONCAKE_MASTER_METRICS_PORT 把每次启动都固定到 50052, 任何其他监听者都能抢在 Mooncake 绑定之前占有该端口。修复思路是让 OS 分配端口, 而不是在固定端口上等待重试。

实现拆解

1. 修改常量 MOONCAKE_MASTER_METRICS_PORT 的默认值, 从 50052 改为 0 (miles/utils/external_utils/command_utils.py), 使 start_mooncake_master 默认把 --metrics_port 0 传给 mooncake_master, 由内核在 bind 时原子分配空闲 admin 端口, 消除固定端口的竞争窗口。
2. 保持 RPC 端口 50051 固定: _is_tcp_server_ready 与 wait_for_server_ready 的就绪判定仍只看 RPC 端口, metrics 端口动态化不会影响启动等待逻辑, 调用方拿到的“就绪”语义不变。
3. 新增回归测试 test_lets_the_os_choose_the_metrics_port (tests/fast/utils/test_command_utils.py), 验证默认启动命令包含 --metrics_port 0, 且等待目标仍是 ("127.0.0.1", 50051); 原有显式传端口测试 test_restarts_and_waits_until_ready 保持 50152 显式传递, 确保覆盖行为不被破坏。
4. 提交历史显示测试曾一度被移除 (第二个 commit) 后又恢复 (第三个 commit), 最终确认该测试与 GPU W&B 认证失败无关, 保留为回归保护。
5. 验证: pytest -q tests/fast/utils/test_command_utils.py 共 71 个用例通过; 人工端口占用 smoke 测试中, 50052 被占用时 Mooncake 绑定到 41935, /metrics 返回 HTTP 200。

关键文件：

- miles/utils/external_utils/command_utils.py（模块 命令工具；类别 source；类型 core-logic；符号 MOONCAKE_MASTER_METRICS_PORT, start_mooncake_master）：核心修复点：MOONCAKE_MASTER_METRICS_PORT 默认值从固定 50052 改为 0，让 OS 原子分配 admin 端口，start_mooncake_master 的就绪判定仍只依赖固定 RPC 端口。
- tests/fast/utils/test_command_utils.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_lets_the_os_choose_the_metrics_port）：新增回归测试 test_lets_the_os_choose_the_metrics_port，锁定默认启动命令携带 --metrics_port 0 且仍等待固定 RPC 端口，防止后续改动回退。

关键符号：start_mooncake_master, test_lets_the_os_choose_the_metrics_port

关键源码片段

miles/utils/external_utils/command_utils.py

核心修复点：MOONCAKE_MASTER_METRICS_PORT 默认值从固定 50052 改为 0，让 OS 原子分配 admin 端口，start_mooncake_master 的就绪判定仍只依赖固定 RPC 端口。

```
# 固定 RPC 端口保持不变，metrics/admin 端口默认交由 OS 分配
MOONCAKE_MASTER_PORT = 50051
MOONCAKE_MASTER_METRICS_PORT = 0 # 0 表示由内核在 bind 时原子分配空闲端口
MOONCAKE_MASTER_LOG_PATH = Path("/tmp/mooncake_master.log")
```

```
def start_mooncake_master(
    rpc_port: int = MOONCAKE_MASTER_PORT,
    metrics_port: int = MOONCAKE_MASTER_METRICS_PORT,
    timeout: float = 30,
    log_path: str | Path = MOONCAKE_MASTER_LOG_PATH,
) -> None:
    host = "127.0.0.1"
    # 若 RPC 端口已在监听，说明 master 已就绪，直接复用，避免重复拉起
    if _is_tcp_server_ready(host, rpc_port):
        print(f"Mooncake master is already ready at {host}:{rpc_port}", flush=True)
        return

    # 先清理可能残留的旧进程，再异步启动；metrics_port 传 0 时由内核原子分配空闲端口
    cmd = (
        f"pkill -x mooncake_master; "
        f"mooncake_master --rpc_port {rpc_port} --metrics_port {metrics_port} "
        f"> {shlex.quote(str(log_path))} 2>&1 &"
    )
    exec_command_cpu(cmd)
    # 阻塞等待 RPC 端口就绪；metrics 端口不再参与就绪判定，与启动参数解耦
    wait_for_server_ready(host, rpc_port, timeout=timeout)
```

tests/fast/utils/test_command_utils.py

新增回归测试 `test_lets_the_os_choose_the_metrics_port`，锁定默认启动命令携带 `--metrics_port 0` 且仍等待固定 RPC 端口，防止后续改动回退。

```
def test_lets_the_os_choose_the_metrics_port(self, monkeypatch):
    """Binding port zero atomically avoids collisions with other listeners."""
    commands = []
    waits = []
    # 强制认为 RPC 端口不可用，才会真正执行启动命令
    monkeypatch.setattr(command_utils, "_is_tcp_server_ready", lambda host, port: False)
    monkeypatch.setattr(command_utils, "exec_command_cpu", commands.append)
    monkeypatch.setattr(
        command_utils, "wait_for_server_ready", lambda *args, **kwargs: waits.append((args,
        kwargs))
    )

    command_utils.start_mooncake_master()

    # 只拉起一次，且命令里必须带 --metrics_port 0
    assert len(commands) == 1
    assert "mooncake_master --rpc_port 50051 --metrics_port 0" in commands[0]
    # 就绪等待仍只看固定的 RPC 端口，与 admin 端口无耦合
    assert waits == [(("127.0.0.1", 50051), {"timeout": 30})]
```

评论区精华

PR 没有线上 review 评论，Shi-Dong 直接 APPROVED。真正有价值的取舍体现在三个 commit 的演进中：第二个 commit 移除了新增测试，理由是当时 CPU CI 失败与 GPU W&B 认证问题混在一起；第三个 commit 恢复测试，并明确说明该测试在 CPU CI 已通过、与 W&B 认证失败无关。最终保留了这条回归保护，说明作者经历了“先缩小嫌疑面、再还原保护”的判断过程。

- 新增测试该不该保留 (testing): 保留测试，作为默认启动命令的回归保护。

风险与影响

- 风险：
 1. 默认端口行为变化：metrics 端口不再固定为 50052，任何依赖固定端口抓取 /metrics 的外部监控或脚本需要改为读取实际分配端口，或显式传入 `--metrics_port`。
 2. 就绪判定不等待 metrics 端口：start_mooncake_master 只等待 RPC 端口，不等待 admin 端口就绪；由于端口 0 在 bind 时已分配，实际可用性风险很低，但调用方在 RPC ready 后立即访问 metrics 仍可能遇到瞬时未监听窗口。
 3. 测试覆盖局限：新增测试只断言命令字符串与等待参数，没有做真实端口绑定集成测试；端口 0 依赖 OS/ 内核语义，跨平台行为可能不同，不过 Mooncake 目标环境以 Linux 为主。
 4. 显式覆盖路径不受影响：test_restarts_and_waits_until_ready 仍验证 50152 显式传递的场景，兼容性有保障。- 影响：影响范围集中在通过 start_mooncake_master 默认启动 Mooncake master 的训练任务与 launcher，尤其是多实例或残留进程场景下 50052

冲突导致的启动失败；RPC 端口语义不变，客户端寻址不受影响。对用户而言启动更健壮，但使用固定 metrics 端口的监控需要适配；对团队而言是单行常量加单条测试的小改动，回归风险低。 - 风险标记：默认端口行为变更，依赖固定 50052 的监控需适配，就绪判定不等待 metrics 端口，缺少真实绑定集成测试

关联脉络

- PR #2356 Replace all the .sh launch scripts with .py launch script: 同改 miles/utils/external_utils/command_utils.py，启动命令生成逻辑集中在此文件，本次端口修复是该启动器演进线的延续。
- PR #2355 [fix] fix the bugs/outdated commands in .sh scripts and the corresponding snapshots: 同样涉及 command_utils.py 的启动命令生成与测试 harness，属于同一模块的健壮性修复脉络。