

Non-Atomic Blockchain Transaction Identification

测试报告

2025 年 12 月 22 日

1 概述

- 目标：验证非原子套利识别网站在**数据采集、算法计算、API 输出、前端呈现、部署可用性**上的正确性、健壮性与可维护性。
- 方法：黑盒功能测试 + 部署验证，辅以关键路径健壮性和性能抽查，覆盖正常流与异常流。

2 测试环境

- 运行：`docker-compose up --build` 启动 db (PostgreSQL 15)、backend (FastAPI)、frontend (React)、nginx。
- 配置：`.env` 含 `DATABASE_URL/POSTGRES_USER/POSTGRES_PASSWORD`；
前端 `REACT_APP_USE_BACKEND=true` 读取真实数据，否则回退模拟数据。
- 数据：卷 `postgres_data` 持久化；时区统一 UTC。

3 测试范围

- **数据采集与入库**：Binance/Etherscan 拉取、TLS/限流重试、去重逻辑 (哈希 + `log_index`)、schema 升级幂等、UTC 时间戳校验。
- **套利计算**：启发式过滤、时间窗匹配、利润/方向计算、清表再写、手续费/滑点参数一致性、空表健壮性。
- **后端 API**：

- 健康: `/api/health`, `/api/db-check`。
- 价格: `/api/price-data` (日期窗口、OHLC 聚合、空数据返回结构)。
- 套利: `/api/arbitrage/statistics`, `/api/arbitrage/behaviors` (分页/排序/过滤)、`/api/arbitrage/opportunities`(时间过滤/利润率过滤/ISO8601)。
- CORS: 本地/代理跨源。
- 前端:
 - Info: 内容完整、外链属性。
 - Navbar 搜索: 高亮、结果数量、跳转、外部点击关闭。
 - Price Dashboard: 后端/模拟切换、K 线缩放与拖拽、对比模式、日期定位、刷新回退提示、表/图 hover 同步。
 - Arbitrage Analysis: 统计卡、分页表格、排序/过滤、时间轴缩放与拖拽、tooltip/图例、离线/空数据提示。
- 部署/代理: Nginx 路由转发、端口映射、服务依赖 (`depends_on` + 健康检查)、卷持久化验证。

4 用例摘要

领域	关键验证	要点/输入	结果
采集/入库	限流/重试、去重、schema 升级、UTC	主/备 Binance API 切换; Etherscan 分块; 哈希 +log_index 防重; 出错回滚; 缺失.env 报错提示	轻微问题
套利计算	过滤与匹配、利润率、方向	300s 窗口; 手续费/滑点参数一致; 利润率百分比转换; 空表/单侧数据健壮性	通过
后端 API	校验与分页、空数据、时间格式	start > end 拒绝; 空数据返回空结构; ISO8601/Z 解析; 分页/排序边界	通过
前端交互	路由/搜索/图表/过滤	K 线缩放拖拽、对比模式、时间轴缩放拖拽、筛选器复位、hover 同步、回退提示	通过
部署/代理	反代路由、健康依赖	/api → backend, / → frontend; 卷持久化; 健康检查后启动	通过

5 主要测试步骤

- **采集脚本**: 清空数据库, 运行 `fetch_data.py`, 检查日志限流重试、数据完整性 (哈希 +log_index 唯一)、UTC 时间戳。
- **套利计算**: 清空目标表, 运行 `compute_arbitrage.py` 与 `compute_opportunities.py`, 检查利润率、方向、时间序列是否递减返回。
- **前后端通信延迟**: 模拟前端请求, 检查响应时间是否在合理范围内。
- **部署**: Nginx 路由转发检查; 卷持久化验证 (停服重启数据仍在); CORS 在本地/代理路径下可用。
- **Price Dashboard**: 滚轮缩放、拖拽平移、日期选择对齐; 后端断开时回退模拟数据并提示; 对比模式显示 spread 与双曲线。
- **Arbitrage Analysis**: 筛选 (利润率、时间范围)、重置; 时间轴缩放与拖拽; 表格分页与排序一致性; 空数据/加载/错误占位。

6 后端测试结果展示

- **采集脚本**：多次运行脚本 `reset_recompute.sh` 检查 log 信息, 发现受限于 API 访问频率, 会随机出现通过 API 获取数据返回为空的情况, 经查询为 API 非会员受限于访问频率限制
- **套利计算**：各项计算均符合预期, 无异常
- **前后端通信延迟**：脚本 `api_latency_test.sh` 模拟前端请求后端 API, 获取完整交易数据响应时间平均在 3s 以上, 不符合预期, 其余响应时间均在合理范围内, 未发现显著延迟问题。

```
已提交 1000 条 Swap 记录, 总计 37998 条
正在处理区块范围: 23454566 -> 23459565
请求 Uniswap 数据时发生错误: HTTPConnectionPool(host='api.etherscan.io', port=443): Max retries e
xceeded with url: /v2/api/modules/logs&action=getLogs&address=0x1b815ef88f5b1194ae79806624e3d814b/
697f66fromBlock=23454566&toBlock=23459565&topic=0x0c42079f94a6358d7e6235f29174924f928cc2ac818eb64f
ed8004e115fbcc678apikey=GCVZSUG4U1C228SSVF17YITHP64YVBNKCKchainId=1 (Caused by SSL:SSL:Error
(1, '[SSL] record layer failure ([ssl.c:2588]))')
正在处理区块范围: 23454566 -> 23459565
已提交 1000 条 Swap 记录, 总计 38998 条
```

Max Retry Exceeded

```
[5/5] 计算套利候选对...
[pair_candidates] processed 8038/80385 Uniswap swaps
[pair_candidates] processed 16076/80385 Uniswap swaps
[pair_candidates] processed 24114/80385 Uniswap swaps
[pair_candidates] processed 32152/80385 Uniswap swaps
[pair_candidates] processed 40190/80385 Uniswap swaps
[pair_candidates] processed 48228/80385 Uniswap swaps
[pair_candidates] processed 56266/80385 Uniswap swaps
[pair_candidates] processed 64304/80385 Uniswap swaps
[pair_candidates] processed 72342/80385 Uniswap swaps
[pair_candidates] processed 80380/80385 Uniswap swaps
[pair_candidates] processed 80385/80385 Uniswap swaps
找到 501 个套利候选对
已写入 501 条套利候选记录。
```

Arbitrage compute

```
./api_latency_test.sh
target base URL: http://127.0.0.1:8000
iterations per endpoint: 3

==> /api/health
[#01] http_code=200 time_total=0.005405s time_connect=0.000205s time_starttransfer=0.005325s
[#02] http_code=200 time_total=0.002075s time_connect=0.000215s time_starttransfer=0.002004s
[#03] http_code=200 time_total=0.001611s time_connect=0.000208s time_starttransfer=0.001572s

==> /api/db-check
[#01] http_code=200 time_total=0.004037s time_connect=0.000187s time_starttransfer=0.003988s
[#02] http_code=200 time_total=0.003878s time_connect=0.000222s time_starttransfer=0.003840s
[#03] http_code=200 time_total=0.004236s time_connect=0.000208s time_starttransfer=0.004166s

==> /api/price-data
[#01] http_code=200 time_total=3.204665s time_connect=0.000206s time_starttransfer=3.204566s
[#02] http_code=200 time_total=3.266181s time_connect=0.000304s time_starttransfer=3.266031s
[#03] http_code=200 time_total=3.004208s time_connect=0.000274s time_starttransfer=3.004117s

==> /api/arbitrage/statistics
[#01] http_code=200 time_total=0.002316s time_connect=0.000290s time_starttransfer=0.002279s
[#02] http_code=200 time_total=0.002850s time_connect=0.000308s time_starttransfer=0.002797s
[#03] http_code=200 time_total=0.002332s time_connect=0.000235s time_starttransfer=0.002292s

==> /api/arbitrage/behaviors?page=16page_size=5
[#01] http_code=200 time_total=0.002969s time_connect=0.000216s time_starttransfer=0.002925s
[#02] http_code=200 time_total=0.002899s time_connect=0.000228s time_starttransfer=0.002850s
[#03] http_code=200 time_total=0.002438s time_connect=0.000226s time_starttransfer=0.002402s

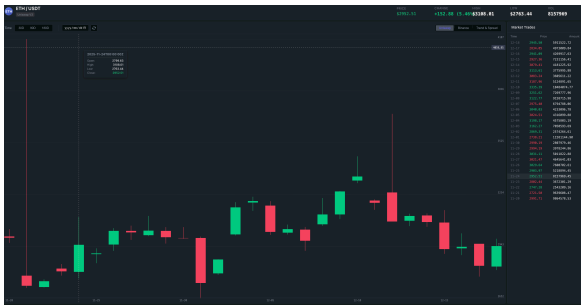
==> /api/arbitrage/opportunities?min_profit_rate=0
[#01] http_code=200 time_total=0.004222s time_connect=0.000220s time_starttransfer=0.004176s
[#02] http_code=200 time_total=0.003789s time_connect=0.000236s time_starttransfer=0.003732s
[#03] http_code=200 time_total=0.003933s time_connect=0.000243s time_starttransfer=0.003892s
```

Backend API Latency

图 1: 后端测试结果展示

7 前端效果展示

- **前后端打通效果**：Price Dashboard（前端）读取后端真实数据, Arbitrage Opportunities 列表分页/过滤正常。
- **交互验证**：K 线图缩放拖拽、对比模式、时间轴缩放拖拽、筛选器复位、hover 同步、回退提示均符合预期。



Price Dashboard



Arbitrage Opportunities

图 2: 前后端接口打通效果：仪表盘与套利机会并排展示

关键发现

- 1. 采集限流场景缺少**最大退避/总重试上限**，长时间 429/451 会持续尝试；建议增加上限并输出报警（中）。
- 2. 由于爬取数据 API 并非会员，由于爬取速度过快而被 API 禁止访问一段时间稳定可复现。
- 3. 前端 Dashboard 请求数据时，由于数据量较大，导致响应时间过长

8 改进与后续

- 采集：增加最大退避和失败告警；分段重试上限。
- CI：加入轻量 API 合约测试（健康、价格、套利列表）防回归。
- 后端响应速度较慢，可以对数据进行分页或按需加载以提升响应速度。
- 前端：离线模式提示，明确数据来源（后端/模拟）。

9 附录：覆盖清单

- 后端接口：/api/health, /api/db-check, /api/price-data, /api/arbitrage/statistics, /api/arbitrage/behaviors, /api/arbitrage/opportunities

- 主要脚本: `fetch_data.py`, `compute_arbitrage.py`, `compute_opportunities.py`, `reset_recompute.sh`, `api_latency_test.sh`
- 前端页面: `/info`, `/price-dashboard`, `/arbitrage-analysis`