## 股票打板策略实战手册

Tushare、QMT获取数据，QMT实盘，BackTrader线下回测。

代码位置：~/pwork/qmt/limitup

策略目录：D:\银河证券QMT实盘 - 交易终端-北京\python

策略运行目录：D:\银河证券QMT实盘 - 交易终端-北京\bin.x64

Redis位置：C:\Program Files\Redis

1. 简单打板策略（SimpleHit）

根据banlist.h5建立一个1800支左右的股票池，预留10～40%资金，早盘买入股票，盘中自动或手动卖出股票，触发风控则暂停自动交易。期望大盘涨的时候赚取跟随指数的绝对收益，大盘跌的时候做好风控减少损失，大盘盘整的时候能够捕捉到10%的封板股票、当日80%股票正收益、全周期60%股票正收益。以赚取绝对收益为主，相对收益期望年化10%。

产品管理策略：

1. 普通池：根据banlist建立一个1800支左右的股票池（banlist.txt）；
2. 优先池：根据题材热点选取的300支左右的股票池（prolist.txt）；
3. 优先池的触发参数可适度优化，比方说，A类60%加成（上涨4.8%、快速3%、急速1.8%），B类80%加成（上涨6.4%、快速4%、急速2.4%）。

SimpleHit策略用分钟线作为判断依据，用tick行情作为触发条件，只做早盘（9:30～10:00）。有以下买入触发条件：

1. 价格上涨到8%；
2. 价格快速上涨+5%且涨幅>5%（3分钟内）；
3. 价格急速上涨+3%且涨幅>3%（1分钟内）。

买入订单管理策略：

1. 分为I类单（优先单）、II类单（普通单）；
2. I类单在资金充裕时直接下单，II类单先加入订单队列，在周期末统一排序下单，如果资金不足则进行撤单、卖单，下一周期资金恢复再下单；
3. 使用Tick行情High价格下单，5秒内未成交则撤单再下单，改用市价下单。

卖出触发条件：

1. 分为I类单（止损单）、II类单（调仓单）；
2. I类：相比建仓买入价，亏损5%；
3. I类：相比昨收，下跌5%；
4. I类：连续涨之后，跌破5日均线；
5. II类：相比建仓买入价，亏损2%；
6. II类：相比昨收，下跌2%；
7. II类单按照跌幅排序，在资金不足时卖出以恢复资金，使用Tick行情High价格下单，5秒内未成交则撤单再下单，改用市价下单。

记录买卖成交（transaction.log），用来计算收益，核对持仓。

记录买卖订单（order.log），用来分析下单失败情况，优化下单策略。

日中控制文件（control.json），用来人工暂停策略执行，通常不在QMT中终止策略，让QMT策略持续执行。

1. 量化脚本

程序：simplehit.py

初始化函数：init

处理函数：handlebar，交易时段期间，每3秒被调用一次

1. Redis风控

使用redis作为风控信号传递工具

启动redis服务端：

redis-server.exe redis.windows.conf

启动redis客户端：

redis-cli

查看风控信号：

> keys \*

> get halt\_flag

盘中设置暂停信号：

> set halt\_flag True

盘中恢复交易：

> set halt\_flag False

Handlebar处理函数：

从redis获取风控信号，假如设置为True，函数不作处理直接返回

1. 账户持仓

init初始化函数：

调用get\_trade\_detail\_data获得账户及持仓

1. 股票池

普通股票池为banlist.csv，使用banlist.py（~/pwork/tushare/limitup）生成，手工拷贝到策略运行目录下

init初始化函数：

加载普通股票池，扣掉停牌池、持仓池（已有股票不要重复买入）

1. 取日线历史行情

init初始化函数：

使用2025days.csv，计算上一交易日

调用download\_history\_data下载昨日行情数据，调用get\_market\_data\_ex获取日线行情，获得昨收盘价格，写入day\_YYYYMMDD.pickle。假如day\_YYYYMMDD.pickle文件存在，则可快速加载，跳过调用API的时间（约两分钟）

1. 取tick最新行情

Handlebar处理函数：

调用get\_full\_tick获取最新tick行情

根据tick行情复现分钟线行情，用于快速规则判断

1. 买入规则判断

Handlebar处理函数：

价格规则：

* 昨收上涨8%
* 一分钟上涨3%
* 三分钟上涨5%

时段保护：仅在9:30～11:30、13:00～15:00之间执行

价格保护：不买涨停板

边界条件：考虑历史分钟线行情无数据的情况

1. 订单执行

order\_entry函数

调用passorder下达买入订单，使用最新价

已下单股票记录进A.buylist，避免重复购买

设置每日订单阀值A.quota，控制资金使用

资金使用实时更新A.cash，减少无效计算

暂未考虑不成交、撤单、手工下单、手工卖出等情况，建议后续增加一个refresh信号，或者最简单就是停掉策略、重新执行

1. 卖出规则判断

Handlebar处理函数：

价格规则：

* 相比买入价下跌2%
* 昨收下跌2%
* 相比5日均线下跌2%

1. 远程风控

程序：riskcontrol.py

提供API接口（基于fastapi）实现远程访问， QMT将交易数据放于redis供程序读取，程序可通过redis间接操控QMT

接口遵循最小可用原则，股票名称、开收盘价等通过tushare获取

实现以下功能：

* /clean，由数据采集客户端调用，清除redis数据
* /collect，由数据采集客户端调用，上传redis数据至NC5主机，包括账户、持仓、订单、成交数据
* /holdings，查看持仓
* /orders，查看订单
* 查看策略所下订单（买单、卖单）
* /trades，查看成交
* /account，查看账户
* 暂停/恢复策略执行
* 发送重置指令（更新资金、重置阀值、重取可卖出持仓）

1. 查询转接

程序：wrapper.py，在QMT中运行

Redis只记录当天账户、持仓、订单、成交，每tick（3秒）更新数据。初始化时清除所有数据

账户：account

{asset:, value:, cash:, profit:, date:}

持仓：holding-600390.SH

{"volume": , " volume\_a": , "initprice":, "last":, "date:", "tradeid":}

订单：order-6190559075712501398

{"stock": "", direction: , price:, qty:, price\_x:, qty\_x:, orderno:, status:, info:, time:, remark: }

成交：trade-

{"stock": "", direction: , price:, qty:, txn:, time:, commission:}

1. Redis数据同步

程序：redis\_client.py，数据采集客户端

作为客户端连接riskcontrol，持续（每秒一次）将数据上传到NC5主机

调用接口：/clean, /collect

数据采集路径：

warpper@QMT => Redis@ND => redis\_client@ND => /collect/ => riskcontrol@NC5 => Redis@NC5 => redis2file@NC5 => eodYYYYMMDD.pickle => closebatch@NC5 => trade.txt

1. 卖出清单

每日新买入的股票，以及当日selllist未卖出的股票，合在一起成为次日的selllist

程序：gen\_selllist.py

通过messager将卖出清单传给QMT策略

1. NC5主机定时任务

早上开盘前清除redis订单成交数据

0 9 \* \* 1-5 /home/hifi/pwork/qmt/limitup/clean.sh

开盘后计算涨幅排名股票清单

31 9 \* \* 1-5 /home/hifi/pwork/tushare/morningphone.sh

闭市后生成当日订单成交数据、次日可卖股票清单

20 15 \* \* 1-5 /home/hifi/pwork/qmt/limitup/closebatch.sh

开盘操作步骤：

1. ND台式机clean、检查
2. NC5主机clean（自动）、检查
3. QMT启动warpper
4. NC5主机启动riskcontrol（建议隔夜重启）
5. ND台式机启动redis\_client（建议隔夜重启）
6. QMT启动simplehit（通常10点即可终止）
7. NC5主机跑morningphone（自动）

收盘操作步骤：

1. NC5主机跑closebatch（自动；如果不成功，需要检查并重走开盘操作步骤）
2. QMT停止simplehit
3. ND台式机停止redis-client
4. NC5主机停止riskcontrol
5. QMT停止warpper

信息输出：

开盘涨幅排名清单：http://n5.hifiax.com:8081/stocks/

收盘当日成交记录：http://n5.hifiax.com:8081/trades/

1. 历史订单成交管理

每天收盘将订单、成交、持仓、账户数据写入mysql数据库（qmt库）

程序：file2sql.py

要避免重复写入；要提供去重工具

1. 交易绩效评估

对每个订单（股票+日期）评估盈亏、天数、基本面、技术面

订单remark字段，记录虚拟户、策略，用于区分不同客户、不同策略的绩效

策略：SIMH（简单打板）、CBAS（可转债高频）

客户：HIFI（本人）

1. 打板策略（2024年）

预期建立1000～2000支的股票池，每日抓买10～30支股票，其中1/3～1/2达到涨停，期望当日收益达到0.1%，当日回撤不超过1%。次日预计继续持有1/3股票，卖出2/3股票并继续抓买股票。期望1～2支连板，带来1%收益，其余股票小涨，带来1%收益，次日回撤不超过1%。假如触发风控，即暂停打板。

买策略

一．早盘根据股价涨势，预判可能涨停的股票提前买进

二．追买封板或连板的股票，希望在开板时买进

三．根据集合竞价的虚拟成交价，追买可能开盘涨停的股票

四．根据公司公告、行业新闻和宏观经济数据，预判利好的股票提前买进

五．盘中根据股价涨势，预判可能涨停的股票提前买进

六．跟随进入涨势的板块或题材，提前买进相关板块或题材的股票

持有/卖策略

1. 判断有机会连板，继续持有
2. 判断连板结束，可以卖出
3. 判断转入跌势，立刻止损
4. 符合股票T0交易条件的，可以转入T0股票池

风控策略

1. 买入后未能涨停，反而触达止损线，立刻卖出单支股票
2. 当天整体回撤超过3%，立刻中止打板策略，视情况继续持有、转T0交易、当日清仓、整体清仓
3. 数据获取（2024年）

基于Tushare数据源

代码：getstock.py，获取日线数据（前复权）

代码：getdaily.py，获取日线数据（普通）

代码：banlist.py，只做沪深主板股票，排除黑名单，建立股票池（总计1878支）

黑名单规则：

1. ST股票
2. 接近退市股票（价格低于5元）
3. 流通市值小于20亿元
4. 市盈率（PE、PE\_TTM）为null，即亏损

代码：showstock.py，计算涨停股票

代码：filter\_stock.py，对股票池，计算涨停股票

代码：filter\_analysis.py，对股票池，作连板统计

代码：daily\_profit.py，统计日线盈亏

1. 分钟线行情（2024年）

基于QMT数据源

代码：getminute.py，下载分钟线数据，5108个文件

代码：showminute.py，统计所有涨停股票的形态

* bStraight：一字涨停，从开盘就是涨停价
* bOneHit：从未开板，触达涨停价后一直保持
* nStage：首次触达涨停时段，0～7，每半小时算一个时段

bOneHit：封板后一直未开板