

# 链上基金

Ether-Hedge

项目说明文档

买链上基金，从此恋上基金

2018 年 6 月 26 日

## 一、概览

本项目是一个基于区块链技术的线上封闭式基金 DApp，依托智能合约和专有代币（Ether-Hedge Token），实现了线上基金、策略投票两个核心功能。

### 1.1 线上基金

#### 1.1.1 募集

### Ether-Hedge 演示

当前策略: Multi\_Factor

Phase 1 募集阶段

购买账户

投资额 (Hedge)

同意用户协议

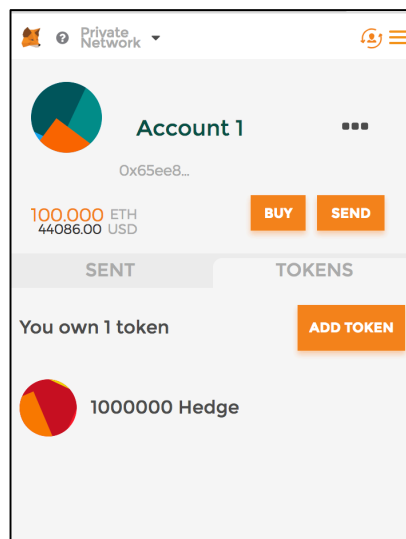
☐ 勾选同意

提交

进入下一阶段

从上图可以看到当前选出的策略是多因子策略（对应 Muti\_Factor.py）。

输入购买账户及投资额（以 Ether-Hedge Token 计），勾选同意后即可提交购买请求。点击进入下一阶段即可开始基金交易。使用 MetaMask 接入本地区块链后可以查看 Token 余额。



1.1.2 交易

Ether-Hedge 演示		
Phase 2 封闭阶段		
项目	总额	盈亏
资金池	550000	0.0
0x65ee808b534191Dae0c99f3E886Ab850A9bB3355	100000	0
0x53b89B649628640EAB6cC21DB42d82Ec430b9521	150000	0
0x017e328c496504bb25eA32CD1d84eA265674B634	300000	0
<div>清算</div>		

此页面会根据后台量化策略的交易结果实时显示浮动盈亏。最上方一行显示的是整个资金池的盈亏，下面是各明细账户盈亏。

点击清算后，进入结算阶段。

1.1.3 结算

Ether-Hedge 演示		
Phase 3 结算		
项目	余额	盈亏
资金池	550000.0	0.0
0x65ee808b534191Dae0c99f3E886Ab850A9bB3355	100000.0	0.0
0x53b89B649628640EAB6cC21DB42d82Ec430b9521	150000.0	0.0
0x017e328c496504bb25eA32CD1d84eA265674B634	300000.0	0.0

显示基金的盈亏结果，并将盈亏自动分配到各明细账户。

## 1.2 策略投票

策略投票

54

69

2333

多因子策略

一种应用十分广泛的选股策略，其基本思想就是找到某些和收益率最相关的指标，并根据该指标，建一个股票组合。



ARTIFICIAL INTELLIGENCE

机器学习策略

传统策略是通过建立数学模型找出市场的规律，往往会接近某一个局部的最优。真正的全局最优或许在我们的经验认知之外。



海龟交易策略

当收盘价大于过去20个交易日中的最高价时以收盘价买入，当收盘价小于过去10个交易日中的最低价时以收盘价卖出。

账户

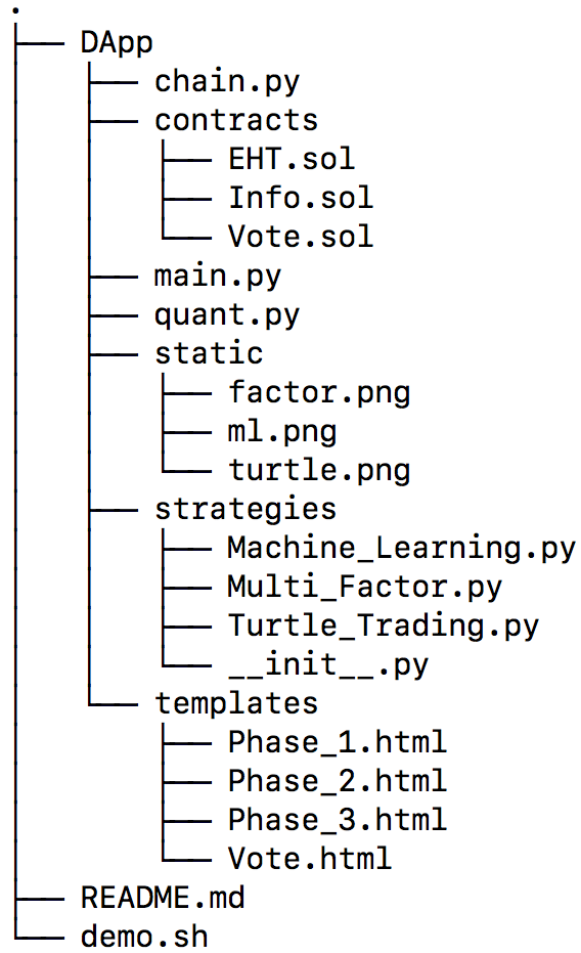
意向策略

提交

显示结果

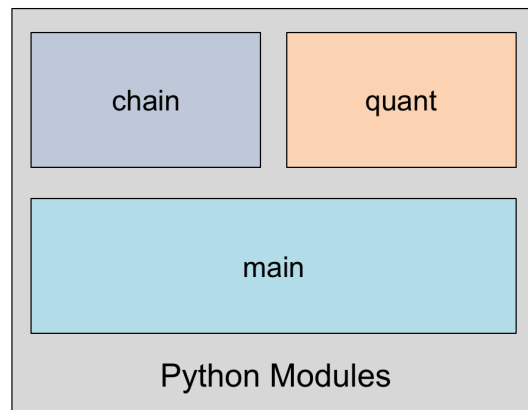
输入账户以及意向策略可以根据账户的份额权重对策略进行投票，投票选中的策略会用于下一期的基金交易。

## 二、模块说明



项目文件结构

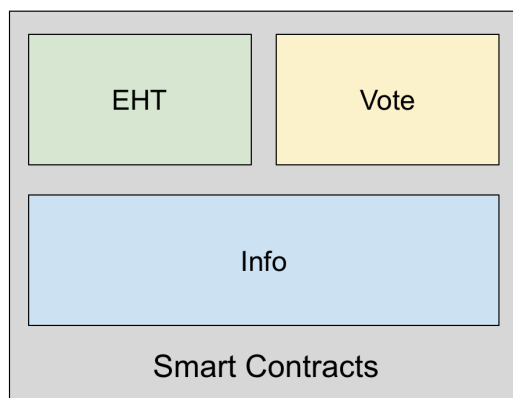
### 2.1 主模块



文件列表：main.py, chain.py, quant.py

编译并初始化 contracts 目录下的智能合约并登记上链, 启动 Flask 框架并响应请求, 与本地区块链进行交互, 与后台交易进程交互。

## 2.2 智能合约



文件列表：contracts 目录下所有 sol 文件

- EHT.sol 定义了 Ether-Hedge Token 的发行以及转账。
- Info.sol 实现了当前投票选取的量化策略的记录、更新, 实现了当前所有认购用户的份额记录的更新及查询。
- Vote.sol 则基于 Info.sol 上的份额记录实现了基于份额权重的策略投票, 并将投票结果记录在区块链上。

## 2.3 前端网页

文件列表：templates 目录下所有 html 文件

- 基于 Bootstrap 框架的网页, Phase1-3 对应募集、交易、结算三个阶段, Vote 对应策略投票。

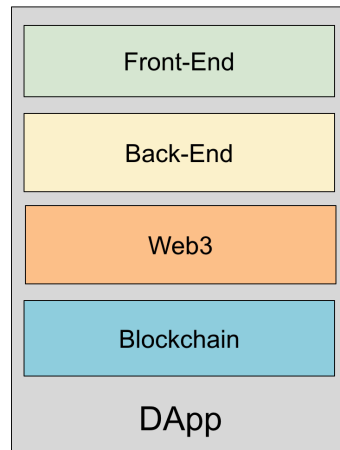
## 2.4 量化策略

文件列表：strategies 目录下所有 py 文件

- 根据项目提供的策略模板实现的经典量化策略。
- \_\_init\_\_.py 用于根据链上投票选出的策略实现策略的自动切换。

## 三、技术原理

使用语言：Python + Solidity + JavaScript



DApp 结构图

### 3.1 前端

前端网页基于 Bootstrap 框架搭建，使用 Jinja2 模板引擎渲染，配合 Python Flask 后端实现交互。

### 3.2 后端（网页）

与前端交互使用 Python Flask 后端框架搭建，根据前端网页表单提交的 Get/Post 请求动态处理。

### 3.3 后端（Web3）

与区块链交互使用的是以太坊官方提供的 Web3.py 库，与本地初始化的区块链进行交互。

### 3.4 区块链

使用 Ganache 一键部署区块链，在本机初始化一个局部区块链网络，并初始化 5 个区块链账户。

### 3.5 策略自动交易

使用多线程+消息队列的方法启动交易进程，并动态轮询当前交易结果。

此外，项目还提供了一个策略模板，任何基于此模板的量化交易策略都可以应用

于链上基金。

## 四、Demo 运行方法

### 4.1 线上基金

安装 Ganache、Python 及相应的 Python 依赖包之后，首先启动 Ganache，待区块链初始化完毕之后，将终端（Terminal）切换到 DApp 目录下，运行

```
Python3 main.py
```

即可进入基金募集页面。

### 4.2 策略投票

在 4.1 运行之后，在浏览器中访问

<http://127.0.0.1:5000/vote>

即可。

## 五、附录

### 5.1 项目开源地址

<https://github.com/yrluo/Ether-Hedge>

### 5.2 Ganache（一键区块链）项目网站

<https://truffleframework.com/ganache>