Solidity Ethereum Uniswap

Uniswap v3 详解（六）：闪电贷

Posted on April 5, 2021

Flash swap

Uniswap v2 版本中，就已经提供了被称作 flash swap 的闪电贷功能。即可以向一个交易对借贷 x token，但在还贷时使用 y token.

一个普通的 Uniswap v2 交易的执行顺序为：

1. 交易之前，Uniswap Pair 合约中 x, y token 余额满足 $x \cdot y = k$
2. 用户支付 x token 到合约中
3. 调用合约的交易接口
4. 合约计算出用户可以得到的 y token，并发送给用户
5. 交易完成后，Uniswap Pair 合约中 x, y token 余额满足 $x' \cdot y' \geq k$

以上过程都发生在同一个交易中。

falsh swap 的实现原理是：

1. 借贷方可以先向合约借贷 x, y token 中某一个（或者两个都借贷）
2. 借贷方指定借贷的数量，以及回调函数的参数，调用 flashswap
3. 合约会先将用户请求借贷的 token 按指定数量发送给借贷方
4. 发送完毕后，Uniswap Pair 合约会向借贷方指定的合约的地址调用指定的回调函数，并将回调函数的参数传入
5. 调用完成后，Uniswap Pair 合约检查 x, y token 余额满足 $x' \cdot y' \geq k$

以上过程都发生在同一个交易中。

在 flash swap 中，用户可以不需要预先支付 token 就可以得到想要的 token，这部分需要支付的 token 只需要在回调函数中转回给合约即可。在 flashswap 完成后 AMM 池中的价格会发生改变（如果使用同币种还债则价格不会改变）。flash swap 可以用来进行 AMM 之间套利，借贷平台清算等操作。

flash swap 类似于一个功能更强的闪电贷，一个接口即可完成借贷和交易的操作。关于 flash swap 的更多内容，可以参考官方文档。

Uniswap v3 的 flash swap

Uniswap v3 版本中，和 v2 一样也有两种闪电贷的方式，但是是通过不同的函数接口来完成的。

- 第一种是普通的闪电贷，即借入 token 和还贷 token 相同，通过 `UniswapV3Pool.flash()` 完成
- 第二种是类似 v2 的 `flash swap`，即借入 token 和还贷 token 不同，这个是通过 `UniswapV3Pool.swap()` 来完成的。

flash

普通闪电贷的接口为交易池合约的 `UniswapV3Pool.flash()` 函数，它的实现也比较简单：

```
function flash(
    address recipient,  // 借贷方地址，用于调用回调函数
    uint256 amount0,    // 借贷的 token0 的数量
    uint256 amount1,    // 借贷的 token1 的数量
    bytes calldata data  // 回调函数的参数
) external override lock noDelegateCall {
    uint128 _liquidity = liquidity;
    require(_liquidity > 0, 'L');

    // 计算借贷所需要扣除的手续费
    uint256 fee0 = FullMath.mulDivRoundingUp(amount0, fee, 1e6);
    uint256 fee1 = FullMath.mulDivRoundingUp(amount1, fee, 1e6);
    // 记录下当前的余额
    uint256 balance0Before = balance0();
    uint256 balance1Before = balance1();

    // 将所需 token 发送给借贷方
    if (amount0 > 0) TransferHelper.safeTransfer(token0, recipient, amount0);
    if (amount1 > 0) TransferHelper.safeTransfer(token1, recipient, amount1);

    // 调用借贷方地址的回调函数，将函数带内传入的 data 参数传给这个回调函数
    IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0, fee1, data);

    // 记录调用完成后的余额
    uint256 balance0After = balance0();
    uint256 balance1After = balance1();

    // 比对借出代币和归还回调函数调用完成后余额的数量，对于每个 token，余额只能多不能少
    require(balance0Before.add(fee0) <= balance0After, 'F0');
    require(balance1Before.add(fee1) <= balance1After, 'F1');

    // 手续费相关的计算
    uint256 paid0 = balance0After - balance0Before;
    uint256 paid1 = balance1After - balance1Before;

    if (paid0 > 0) {
        uint8 feeProtocol0 = slot0.feeProtocol % 16;
        uint256 fees0 = feeProtocol0 == 0 ? 0 : paid0 / feeProtocol0;
        if (uint128(fees0) > 0) protocolFees.token0 += uint128(fees0);
        feeGrowthGlobal0X128 += FullMath.mulDiv(paid0 - fees0, FixedPoint128.Q128, _liquidity);
    }
    if (paid1 > 0) {
        uint8 feeProtocol1 = slot0.feeProtocol >> 4;
        uint256 fees1 = feeProtocol1 == 0 ? 0 : paid1 / feeProtocol1;
        if (uint128(fees1) > 0) protocolFees.token1 += uint128(fees1);
        feeGrowthGlobal1X128 += FullMath.mulDiv(paid1 - fees1, FixedPoint128.Q128, _liquidity);
    }

    emit Flash(msg.sender, recipient, amount0, amount1, paid0, paid1);
}
```

flash swap

通过 `UniswapV3Pool.swap()` 函数，可以完成 `flashswap` 的功能，这个函数在Uniswap v3 详解（三）：交易过程已经有过详细的描述。

在使用 `flashswap` 时，需要实现其 `IUniswapV3SwapCallback` 接口，完成闪电贷的还贷即可，这里不再聚述具体实现。

理解闪电贷

理解闪电贷，你才能理解 DeFi. 虽然 DeFi 领域一直有着大大小小的创新，号称颠覆传统金融。但是在我看来，只有闪电贷才是真正的颠覆者，它是 DeFi 的精髓。它区块链和智能合约的特性发挥到了极致，使得借贷资金的使用效率在短时间内提升到了前所未有的高度。引用 DODO 文档里一段话：

> *Once you have a deep understanding of flash swap, you will realize the superiority of the DeFi world over the centralized world. The composability of smart contracts has elevated the fund utilization of DeFi to an unprecedented level. Thanks to trustlessness, the cost of credit in DeFi is incredibly low. Once this financial system is integrated into the real world, its potential for improving our society and productivity will be truly boundless. The DODO team hopes that flash swap serves as a primer for DeFi builders and beginners alike to gain an appreciation for the power of DeFi.*

至此，关于 Uniswap v3 的所有内容就介绍完毕了。

Uniswap v3 详解系列

本系列所有文章：

- Uniswap v3 详解（一）：设计原理
- Uniswap v3 详解（二）：创建交易对/提供流动性
- Uniswap v3 详解（三）：交易过程
- Uniswap v3 详解（四）：交易手续费
- Uniswap v3 详解（五）：Oracle 预言机
- Uniswap v3 详解（六）：闪电贷

PREVIOUS
UNISWAP V3 详解（五）：ORACLE 预言机

NEXT
UNISWAP V3 的小彩蛋：NFT 和 LP 迁移工具

FEATURED TAGS

Life Basic Performance Tuning Operating System OpenStack Python Web Solidity Ethereum Uniswap DeFi