

以太坊 智能合约 代码 数据空间 存储在哪儿 智能合约和存储空间在哪里?

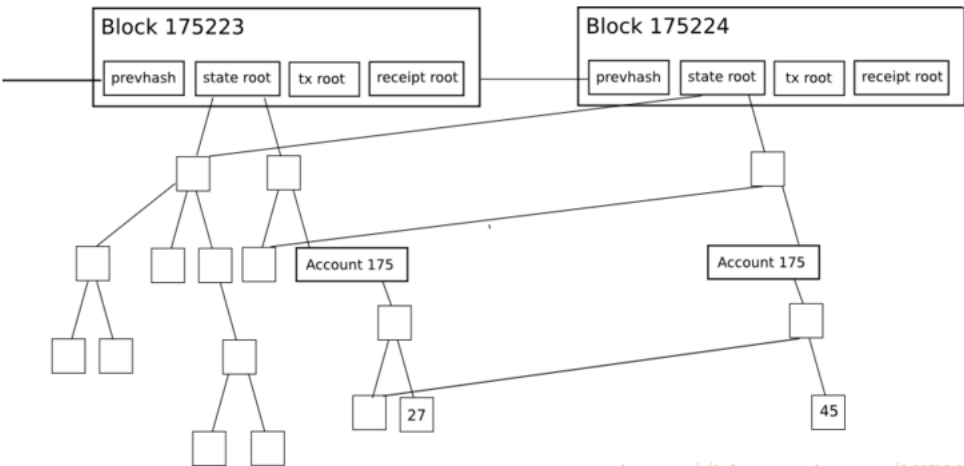
原创

软件工程小施同学 2021-06-17 11:44:03

文章标签 社会时事 文章分类 IT职场 其它 阅读数 763

©著作权

一、以太坊结构



<http://blog.csdn.net/51cto博客>

1. 版本号;
2. 前一区块哈希;
3. 时间戳;
4. 随机数;
5. 目标哈希;
6. **MPT三棵树**

交易树

最底层是交易;

状态树

包含账户,账户允许有空间和代码(外部账户没有代码,有代码的是合约账户.);

收据树

保存的是交易的收据,虽然资料不足,但从架构思路它也不会需要代码和空间,只是完善SPV快速索引.
树结构本身是key-value对,实际也只是哈希值;

合约部署后会创建一个**合约账户**。
合约账户里保存着智能合约的**可执行字节码**(编译之后的,在EVM里执行),并且有存储空间,合约能对合约账户的空间写入信息。
而abi是一个接口结构,利用abiDefinition可以创建调用该合约的结构。
abi应该由合约所有方自己保存和提供。
其次,EVM只是用来执行字节码的,也不会保存智能合约或者存储空间,那么只能是在区块链中。

文章目录

- 交易树
- 状态树
- 收据树
- 1. 智能合约
- 2. 以太坊智能合约
- 3. 触发执行合约
- 4. 以太坊智能合约的执行
- 5. 智能合约的局限性
- 6. 消耗以太币
- 7. 智能合约的代码存储在哪儿
- 8. 智能合约产生的数据,存储在哪儿
- 9. 以太坊的智能合约是并发的,如果调用智能合约,会阻塞其他智能合约的执行

- 2.git 查看冲突文件 unmerg
- 3.git 重命名文件
- 4.区块链 hyperledger fabri
- 5.Hyperledger Sawtooth区

2022年

01月

88篇

2021年 1175篇

热门文章

- cmake ..是什么意思
- fisco bcos transaction交
- TBB concurrent_unorder
- 软件学报 参考文献著录格
- C++ 项目 如何更改依赖

七日热门

- 职场
- 职场
- IT职场
- 职场
- 【职场规划】职场规则
- 职场表达
- 职场定律
- 职场箴言
- 职场箴言
- 职场法则

以太坊 智能合约 代码 数据空间 存储在哪儿 智能合约和存储空间在哪里?



二、以太坊的智能合约

chainmaker

1. 智能合约

一个智能合约可以简单的理解为一段可执行的程序片段，由很多方法和变量组成。
具体的代码由发布人使用特定的编程语言来编写（以太坊使用的是Solidity编程语言，类似JavaScript）。
在本地编译成功后可以发布（部署）到区块链上（相当于发送一个交易）。

2. 以太坊智能合约

以太坊的智能合约可以理解作为一种特殊的交易（包括可执行代码的），被发送出去后会被矿工打包记录在某一个区块中，当需要调用这个智能合约的方法时只需要向这个智能合约的地址发送一笔交易即可。

3. 触发执行合约

智能合约的执行有两种方式：

- 一种是需要更改合约的状态（即存储数据）那么会触发一个交易（或者说是以交易的方式来触发）
- 一种是不更改合约状态的本地调用（如只读取数据等）。一般来说通过ABI的方法调用时会自动处理，如果声明了constant返回那么就是一个call操作，其他则会触发一个sendTransaction操作。

智能合约的调用是告诉EVM说我想执行某个智能合约的某个方法，这种告诉方式是通过发送一个transaction的方式（改变状态的调用）来实现的，而执行智能合约的方法是需要消耗计算资源和存储资源的，这些资源消耗是需要付出代价的，而代价在以太坊的经济模型中就是gas了

4. 以太坊智能合约的执行环境

每个节点电脑都需要安装以太坊客户端，而这个以太坊客户端自带了EVM，一个以太坊智能合约的执行环境，类似JVM一样。
知道智能合约的地址才能访问智能合约，同时还需要知道智能合约的ABI接口才能和智能合约互动（包括访问智能合约的方法和数据），通过交易触发智能合约后智能合约的代码就会在EVM中执行了。
这种方式相当于把程序部署到了非常非常多的电脑上（只要这个电脑安装了以太坊客户端并接入了以太坊网络），随时都可以通过交易来触发这些智能合约的执行，也从而完成了分布式程序的部署和调用。

5. 智能合约的局限性

目前智能合约的功能还非常有限，不能完成所有的现在编程可以解决的事情（比如访问网络下载文件，这个是目前EVM不允许和不支持的）。
但这并不妨碍以太坊和智能合约技术的发展，以及有可能成为下一代互联网基础设施和协议的可能！

6. 消耗以太币

- 以太坊上只要涉及到状态的变更都会消耗以太币，比如一笔交易的产生会更改账户的状态、一个智能合约的存储或更改智能合约的状态等等，消耗掉计算资源和存储资源的都需要支付费用。
- 智能合约是一个程序片段，由很多方法和变量组成，部署智能合约到链上（相当于发送一个交易）需要消耗以太币，通过智能合约改变智能合约的内部状态也需要消耗以太币，但也有在本地读取而不改变状态的是不需要消耗以太币的。

文章目录

交易树
状态树
收据树
1. 智能合约
2. 以太坊智能合约
3. 触发执行合约
4. 以太坊智能合约的执行
5. 智能合约的局限性
6. 消耗以太币
7. 智能合约的代码存储在
8. 智能合约产生的数据，
9. 以太坊的智能合约是第
如果想调用智能合约，作



以太坊 智能合约 代码 数据空间 存储在哪儿 智能合约和存储空间在哪里?



7. 智能合约的代码存储在哪儿

合约代码是存储在区块链中的，并且是在accountstate中的codehash指向的一块存储区域中。

智能合约是在发布的时候，是通过提交一个交易，包括合约的代码和接口，然后被包含在一个区块里，最后被链接到区块链上。
智能合约的代码和接口信息是被放在区块的一个名字叫做状态树的数据结构里（默克尔树）。调用的时候是通过向以太坊网络发一个交易，其中包括智能合约的地址（交易id）和其他调用参数。
矿工在打包这个交易的时候，会通过合约地址找到合约，并且通过调用参数来运行合约。

智能合约部署完后,你可以获得这个合约的地址,可以利用这个合约地址进行查找。
由于存在prune,因此历史的修改记录很可能会被prune掉,关于定义的变量值,你可以在编写合约的时候就写上get方法,这样就可以直接查看了。或者你在wallet上利用合约地址和abi信息,就可以看到public的成员变量。

文章目录

- 交易树
- 状态树
- 收据树
- 1. 智能合约
- 2. 以太坊智能合约
- 3. 触发执行合约
- 4. 以太坊智能合约的执行
- 5. 智能合约的局限性
 - 6. 消耗以太币
- 7. 智能合约的代码存储在哪里
- 8. 智能合约产生的数据，保存到哪儿
- 9. 以太坊的智能合约是带“销毁”的，但是区块链上的数据都是不可篡改的，只能不断的增加长度，那么这个“销毁”是如何做到的呢？
- 如果想调用智能合约，作

8. 智能合约产生的数据，保存到哪儿

每个合约账号下都有一个存储空间，用来存储合约产生的数据，我理解的最底层应该是一个kv数据库，查询普通账户的时候对应的是账户余额，查询合约账户对应的是合约的abi，再通过查询指令操作在evm中运行abi。
通过客户端查询的是键值对的关系，所有的数据都保存在最底层的数据库，而这个数据库按照时间顺序被拆成了一个连接的区块，同时每一个区块里都包含前面所有区块的由mekle树生成的一个hash值，我觉得要是我前面理解的这些应该都没错的话，区块链的设计架构应该还有一个计算层，用来计算每个key对应的value的存储位置，不然查询响应速度应该没那么快才对，那TiDb的架构跟这就差不多了。
之前一直没想明白，为毛区块链上创建那么多账户不会崩，现在想想，每个新账户就是创建了一个kv关系，只有智能合约是消耗存储的。

9. 以太坊的智能合约是带“销毁”的，但是区块链上的数据都是不可篡改的，只能不断的增加长度，那么这个“销毁”是如何做到的呢？

如何执行合约自毁操作呢？调用selfdestruct(recipient);

只有在某个地址上的合约执行自毁操作时，合约代码才会从区块链上移除。合约地址上剩余的以太币会发送给指定的目标，然后其存储和代码被移除。
请问合约自毁后，合约代码真的会物理上从区块链上移除吗？这样不是会造成历史区块中的状态默克尔树的根哈希的改变吗？谢谢！

合约执行自毁操作后（selfdestruct），合约代码不会从区块链上移除。
合约只是从以太坊的底层存储数据结构，三个默克尔树之一的状态树中被去掉了。历史区块不会受到影响，只是未来的区块不会再有对这个合约的引用了。这样对全节点的存储空间是没有影响，但对某些轻节点是会起到减少存储的目的。



<https://www.zhihu.com/question/268916053/answer/345524709>

如果想调用智能合约，但是合约在之前的链上，是如何根据地址找到并执行的呢？

总感觉这么找的效率不高啊

以太坊的智能合约存储在哪里？

<https://www.zhihu.com/question/57056818/answer/187414731>

智能合约和存储空间在哪里？

<https://blog.csdn.net/LNZ001/article/details/72842753>

文章目录

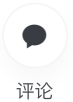
- 交易树
- 状态树
- 收据树
- 1. 智能合约
- 2. 以太坊智能合约
- 3. 触发执行合约
- 4. 以太坊智能合约的执行
- 5. 智能合约的局限性
 - 6. 消耗以太币
- 7. 智能合约的代码存储在
- 8. 智能合约产生的数据，
- 9. 以太坊的智能合约是并
- 如果想调用智能合约，作



赞



收藏



评论



分享



举报

上一篇：kubernetes视频教程笔记（38）-高可用的K8S...

下一篇：python调用shell命令 批量执行python程序



提问和评论都可以，用心的回复会被更多人看到

评论

相关文章

以太坊：入门智能合约

入门智能合约简单的智能合约让我们先看一下最基本的例子。现在就算你都不理解也不要紧，后面我们会更有深入的讲解。存储...

以太坊智能合约部署

一、部署智能合约合约部署需要挖矿才能成功，我们新开一个窗口用与挖矿，新开一个控制台，输入命令：geth attach 连接到控...

solidity智能合约[37]-以太坊虚拟机数据存储

EVM当调用solidity函数时，都会在以太坊虚拟机当中来执行。因此了解以太坊虚拟机的细节，了解其数据的存储...

以太坊 智能合约编译分析

本章主要梳理一下，以太坊智能合约的编译过程以及如何运行的；虽然eWASM可能在未来代替EVM，不过了...



返回顶部

以太坊 智能合约 代码 数据空间 存储在哪儿 智能合约和存储空间在哪里?



这个例子相对复杂一些，用到了以太坊编程语言Solidity的很多特性。例子实现了一个投票智能合约。电子投票系统的一个主要问题...

以太坊智能合约安全入门

Ethernaut记录Fallback题目描述Look carefully at the contract's code below.You will beat this level ifyou claim ow...



以太坊智能合约升级（一）

我们都知道以太坊的智能合约是没办法升级的，然而一个程序出现bug是不可避免的，而通常解决的办法就是升...



以太坊智能合约项目-Token合约开发与部署

| 修订日期 | 姓名 | 邮箱 || :-----: | :-----: | :-----: || 2018-09-05 | brucefeng | brucefeng@brucefeng.com | public pure retur...

以太坊智能合约介绍，Solidity介绍

以太坊智能合约介绍，Solidity介绍 一个简单的智能合约 先从一个非常基础的例子开始，不用担心你现在还一点都不了解，我们...

防止对以太坊的智能合约攻击——代码分析

防止对以太坊的智能合约攻击——代码分析



以太坊智能合约ERC20 Token

pragma solidity ^0.4.8;contract Token{ // token总量，默认为public变量生成一个getter函数接口，名称为totalSupply(). uint25...

以太坊公开拍卖智能合约案例

这篇文章实现一个简单的公开拍卖智能合约，所有人都可以出价，把以太币发送到指定地址，出价B高于出价A时，那么A的资金...

以太坊私有链下智能合约部署

上一篇文章实现了搭建私有链，以下进行智能合约的部署 一、编写合约 简单的乘法例子： pragma solidity ^0.4.2; contract test...

以太坊 智能合约IDE 在线 Solidity IDE

<http://remix.hubwiz.com/> 使用，或者，独立部署 <https://blog.csdn.net/jerry81333/article/details/78118972>



以太坊智能合约在线开发环境

1.<http://remix.ethereum.org> 2.<https://ethereum.github.io/browser-solidity/#optimize=false&version=soljson-v0...>



以太坊智能合约交易最多包含多少数据?

...



神器在哪儿

&n...



如何在以太坊上验证你的智能合约代码

我们知道一般来说你如果你要用你在以太坊上发布的 ERC20 代币进行 ICO, 或者需要发布一款如 FOMO3D 这样的...



文章目录

- 交易树
- 状态树
- 收据树
- 1. 智能合约
- 2. 以太坊智能合约
- 3. 触发执行合约
- 4. 以太坊智能合约的执行
- 5. 智能合约的局限性
 - 6. 消耗以太币
- 7. 智能合约的代码存储在
- 8. 智能合约产生的数据，
- 9. 以太坊的智能合约是第
- 如果想调用智能合约，作



文章目录

- 交易树
- 状态树
- 收据树
- 1. 智能合约
- 2. 以太坊智能合约
- 3. 触发执行合约
- 4. 以太坊智能合约的执行
- 5. 智能合约的局限性
 - 6. 消耗以太币
- 7. 智能合约的代码存储在
- 8. 智能合约产生的数据，
- 9. 以太坊的智能合约是并
- 如果想调用智能合约，作



返回顶部