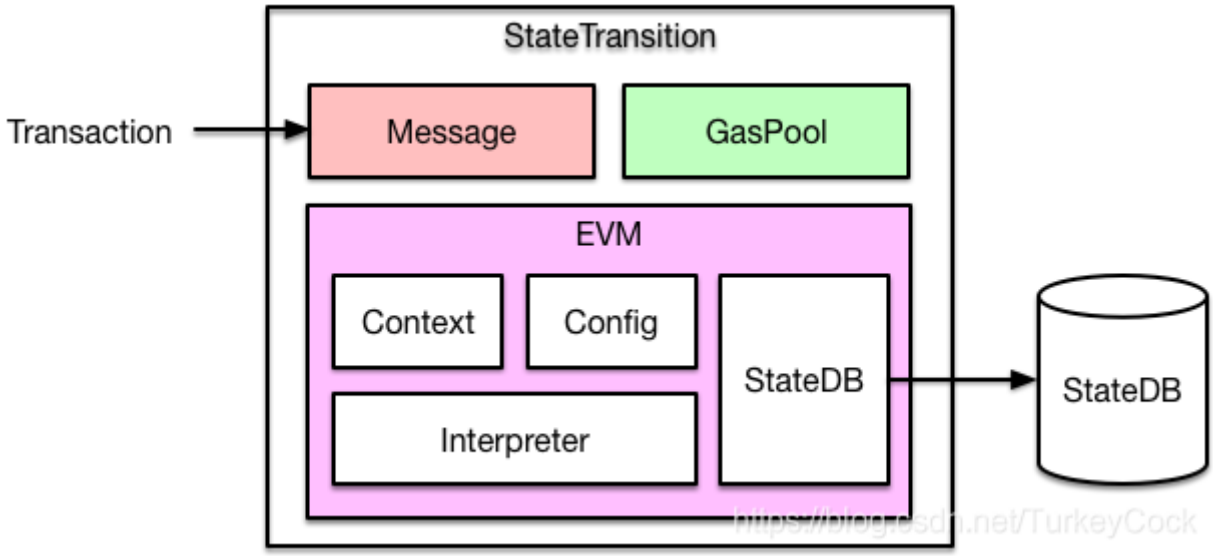
图解以太坊虚拟机EVM_飞久-程序员资料

技术标签： 虚拟机 (/searchArticle?qc=虚拟机&page=1) 以太坊 (/searchArticle?qc=以太坊&page=1) 以太坊源码 (/searchArticle?qc=以太坊源码&page=1) 以太坊源码分析 (/searchArticle?qc=以太坊源码分析&page=1) EVM (/searchArticle?qc=EVM&page=1) 图解 (/searchArticle?qc=图解&page=1)

今天聊一聊以太坊虚拟机的原理。

以太坊虚拟机，简称EVM，是用来执行以太坊上的交易的。业务流程参见下图：



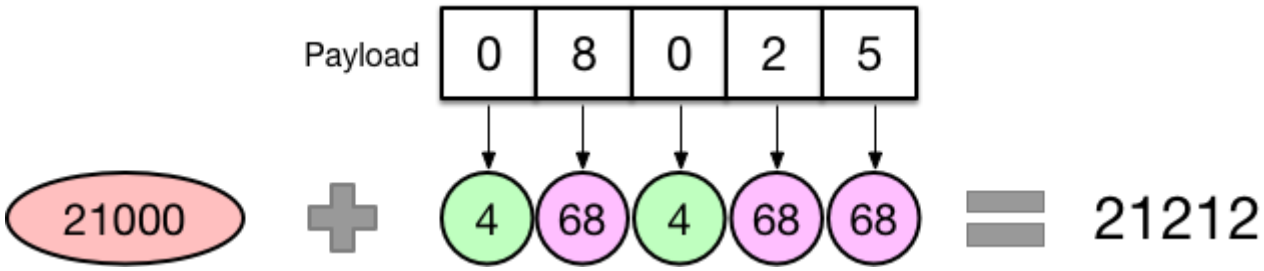
输入一笔交易，内部会转换成一个Message对象，传入EVM执行。

如果是一笔普通转账交易，那么直接修改StateDB中对应的账户余额即可。

如果是智能合约的创建或者调用，则通过EVM中的解释器加载和执行字节码，执行过程中可能会查询或者修改StateDB。

1.固定油费（Intrinsic Gas）

每笔交易过来，不管三七二十一先需要收取一笔固定油费，计算方法如下：

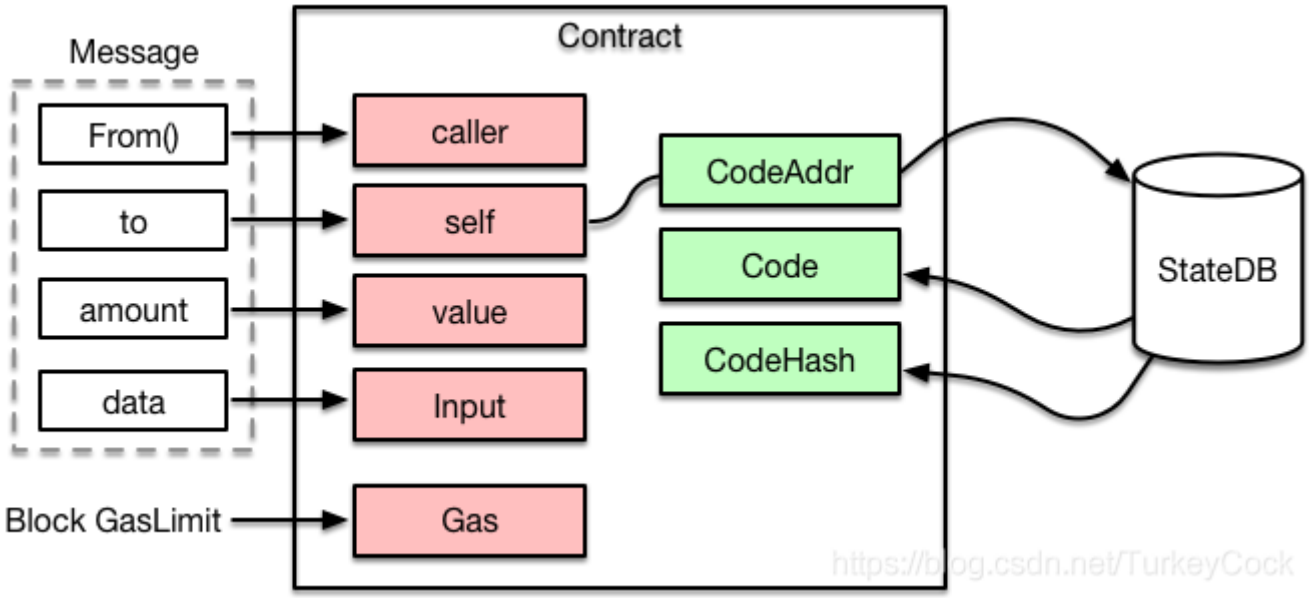


如果你的交易不带额外数据（Payload），比如普通转账，那么需要收取21000的油费。

如果你的交易携带额外数据，那么这部分数据也是需要收费的，具体来说是按字节收费：字节为0的收4块，字节不为0收68块，所以你会看到很多做合约优化的，目的就是减少数据中不为0的字节数量，从而降低油费消耗。

2.生成Contract对象

交易会被转换成一个Message对象传入EVM，而EVM则会根据Message生成一个Contract对象以便后续执行：



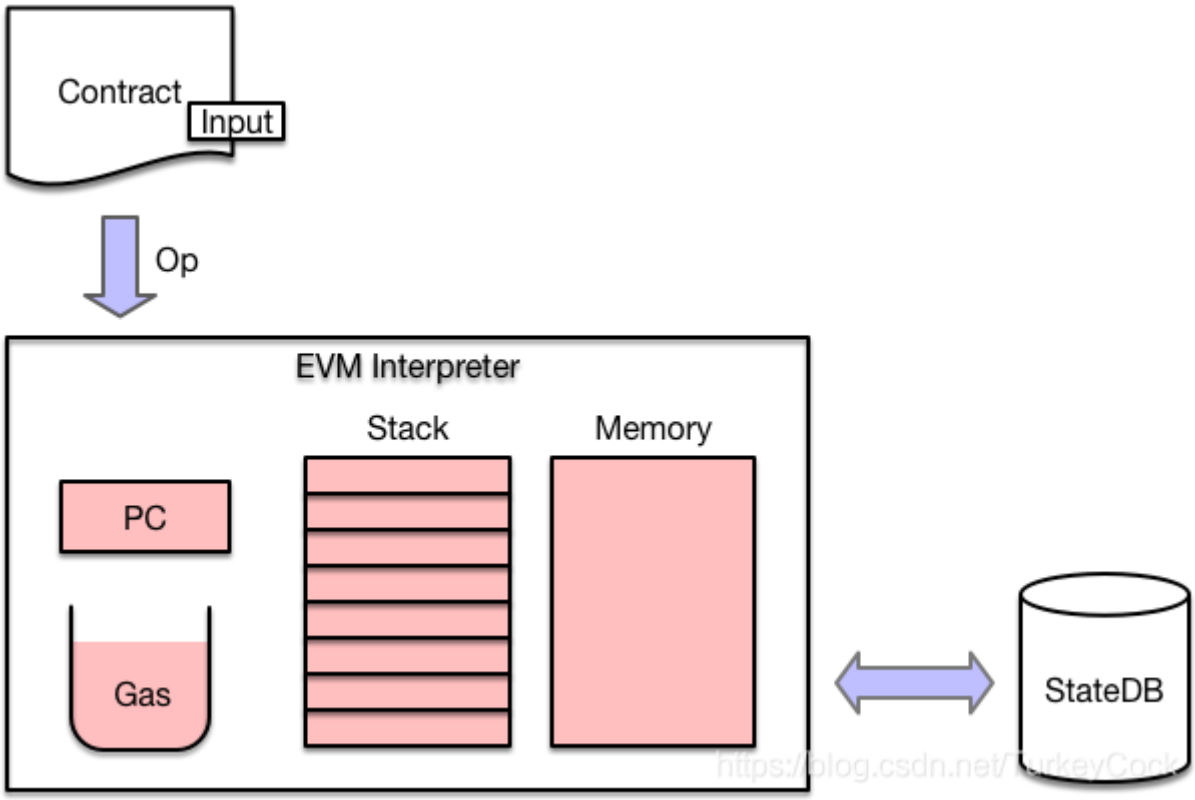
可以看到，Contract中会根据合约地址，从StateDB中加载对应的代码，后面就可以送入解释器执行了。

另外，执行合约能够消耗的油费有一个上限，就是节点配置的每个区块能够容纳的GasLimit。

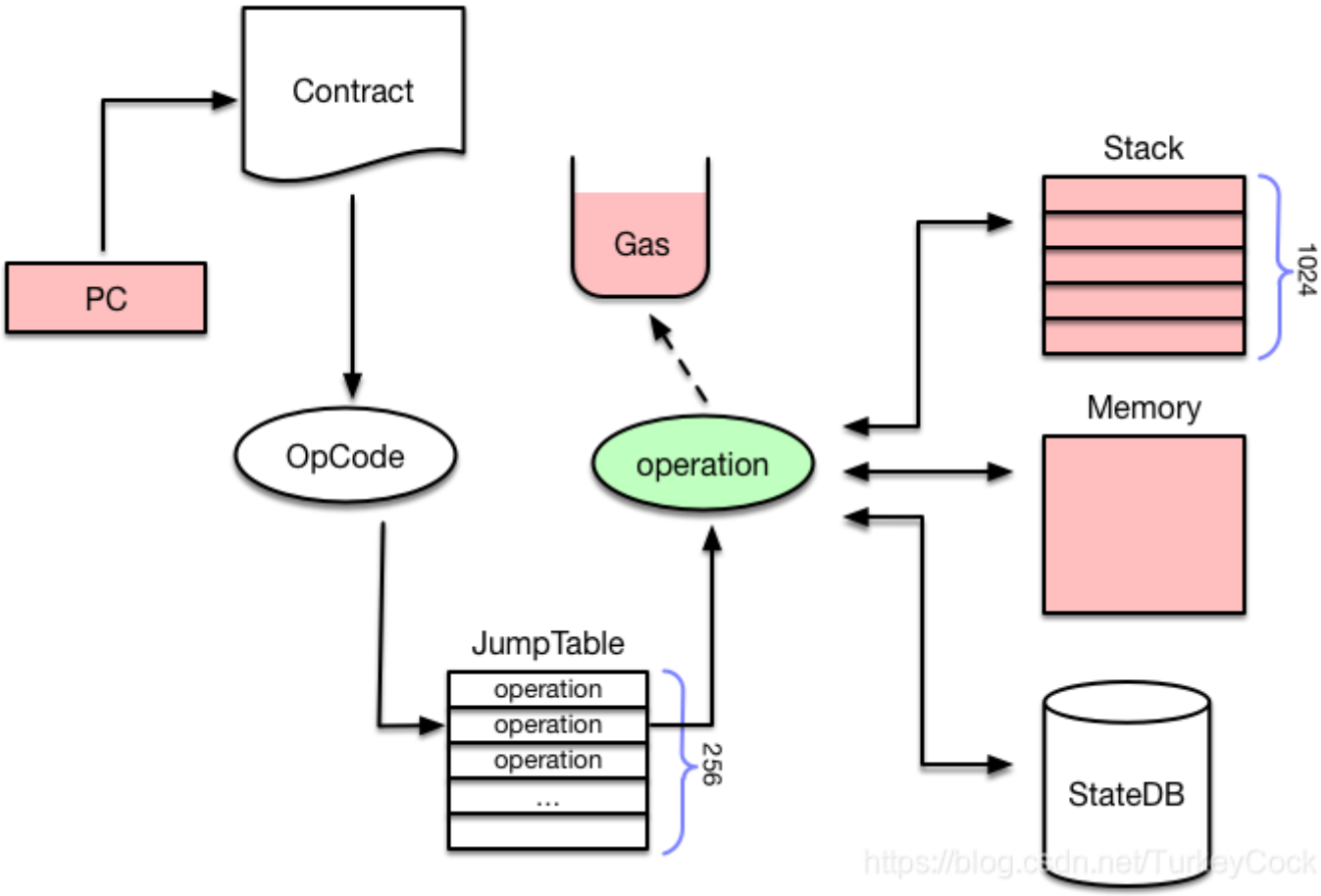
3.送入解释器执行

代码跟输入都有了，就可以送入解释器执行了。EVM是基于栈的虚拟机，解释器中需要操作四大组件：

- PC：类似于CPU中的PC寄存器，指向当前执行的指令
- Stack：执行堆栈，位宽为256 bits，最大深度为1024
- Memory：内存空间
- Gas：油费池，耗光邮费则交易执行失败



具体解释执行的流程参见下图：



EVM的每条指令称为一个OpCode，占用一个字节，所以指令集最多不超过256，具体描述参见：https://ethervm.io (https://ethervm.io)。比如下图就是一个示例（PUSH1=0x60, MSTORE=0x52）：

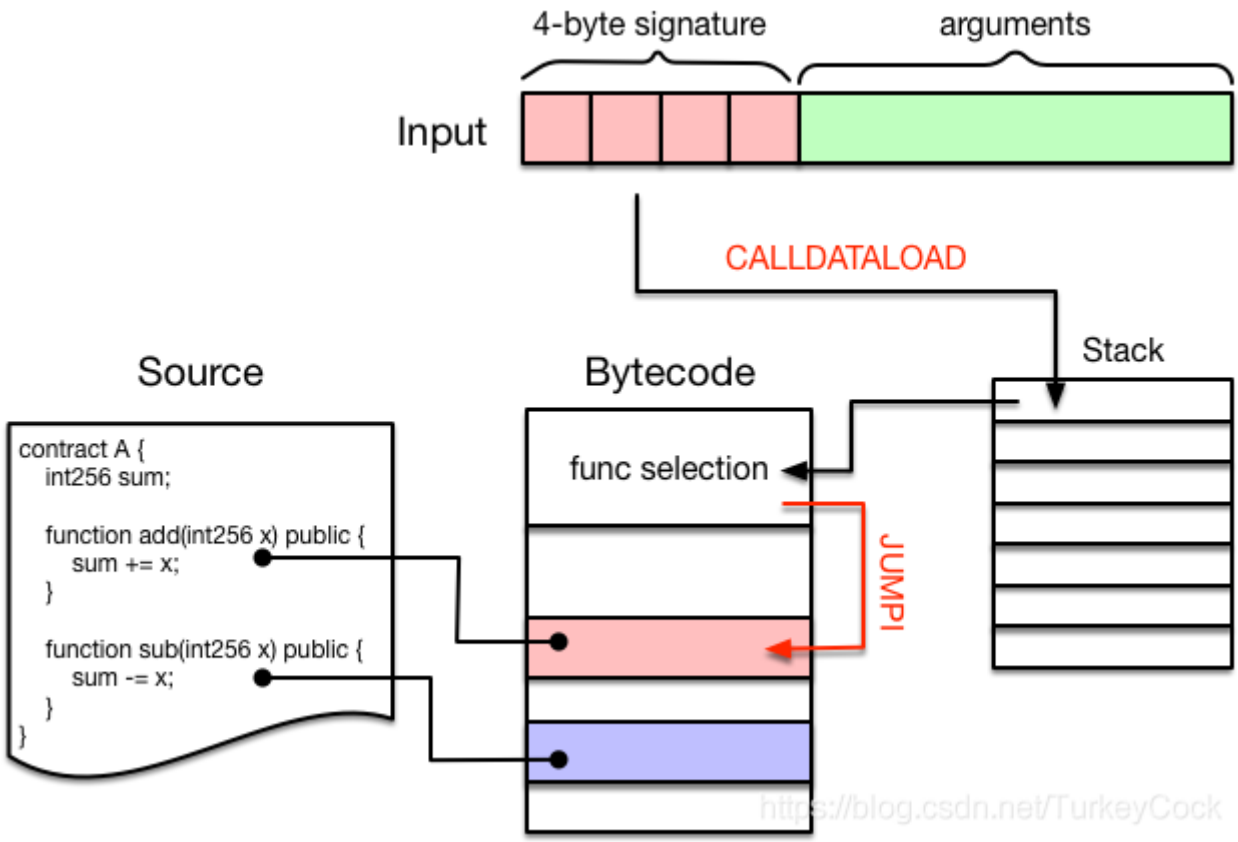
PUSH1 0x60 PUSH1 0x40 MSTORE ⇨ 60 60 60 40 52

首先PC会从合约代码中读取一个OpCode，然后从一个JumpTable中检索出对应的operation，也就是与其相关

联的函数集合。接下来会计算该操作需要消耗的油费，如果油费耗光则执行失败，返回ErrOutOfGas错误。如果油费充足，则调用execute()执行该指令，根据指令类型的不同，会分别对Stack、Memory或者StateDB进行读写操作。

4.调用合约函数

前面分析完了EVM解释执行的主要流程，可能有些同学会问：那么EVM怎么知道交易想调用的是合约里的哪个函数呢？别急，前面提到跟合约代码一起送到解释器里的还有一个Input，而这个Input数据是由交易提供的。



Input数据通常分为两个部分：

前面4个字节被称为“4-byte signature”，是某个函数签名的Keccak哈希值的前4个字节，作为该函数的唯一标识。（可以在该网站查询目前所有的函数签名：https://www.4byte.directory (https://www.4byte.directory)）

后面跟的就是调用该函数需要提供的参数了，长度不定。

举个例子：我在部署完A合约后，调用add(1)对应的Input数据是 0x87db03b7000000000000000000000000000000000000000000000000000000000000000000000001

而在我们编译智能合约的时候，编译器会自动在生成的字节码的最前面增加一段函数选择逻辑：

首先通过CALLDATALOAD指令将“4-byte signature”压入堆栈中，然后依次跟该合约中包含的函数进行比对，如果匹配则调用JUMPI指令跳入该段代码继续执行。

这么讲可能有点抽象，我们可以看一看上图中的合约对应的反汇编代码就一目了然了：

FUNCTIONHASHES

```
{
    "87db03b7": "add(int256)",
    "fa3bd6c5": "sub(int256)"
}
```



```
ontract A {\n    int256 sum;
...
    PUSH 0                    contract A {\n    int256 sum;
...
    CALLDATALOAD                      contract A {\n    int256 sum;
...
    DIV                       contract A {\n    int256 sum;
...
    AND                       contract A {\n    int256 sum;
...
    PUSH 87DB03B7                     contract A {\n    int256 sum;
...
    DUP2                      contract A {\n    int256 sum;
...
    EQ                        contract A {\n    int256 sum;
...
    PUSH [tag] 2                      contract A {\n    int256 sum;
...
    JUMPI                     contract A {\n    int256 sum;
...
    DUP1                      contract A {\n    int256 sum;
...
    PUSH FA3BD6C5                     contract A {\n    int256 sum;
...
    EQ                        contract A {\n    int256 sum;
...
    PUSH [tag] 3                      contract A {\n    int256 sum;
...
    JUMPI                     contract A {\n    int256 sum;
```

load input (func hash)

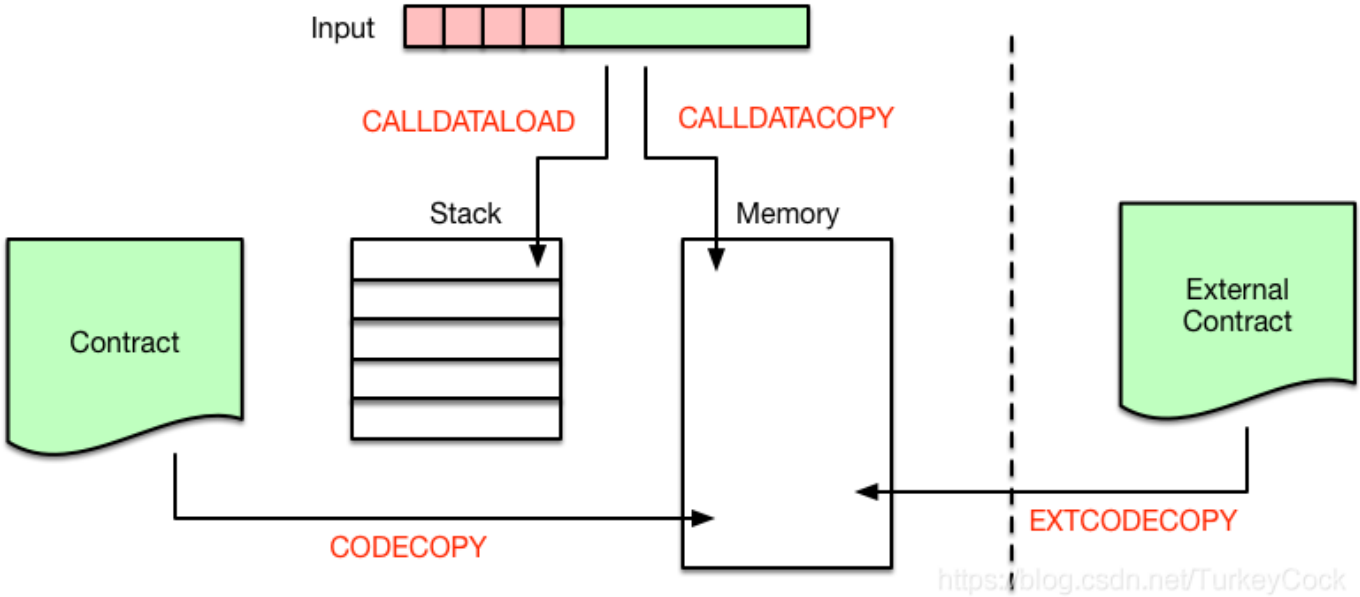
is add()?

is sub()?

这里提到了CALLDATALOAD，就顺便讲一下数据加载相关的指令，一共有4种：

- CALLDATALOAD：把输入数据加载到Stack中
- CALLDATACOPY：把输入数据加载到Memory中
- CODECOPY：把当前合约代码拷贝到Memory中
- EXTCODECOPY：把外部合约代码拷贝到Memory中

最后一个EXTCODECOPY不太常用，一般是为了审计第三方合约的字节码是否符合规范，消耗的gas一般也比较多。这些指令对应的操作如下图所示：

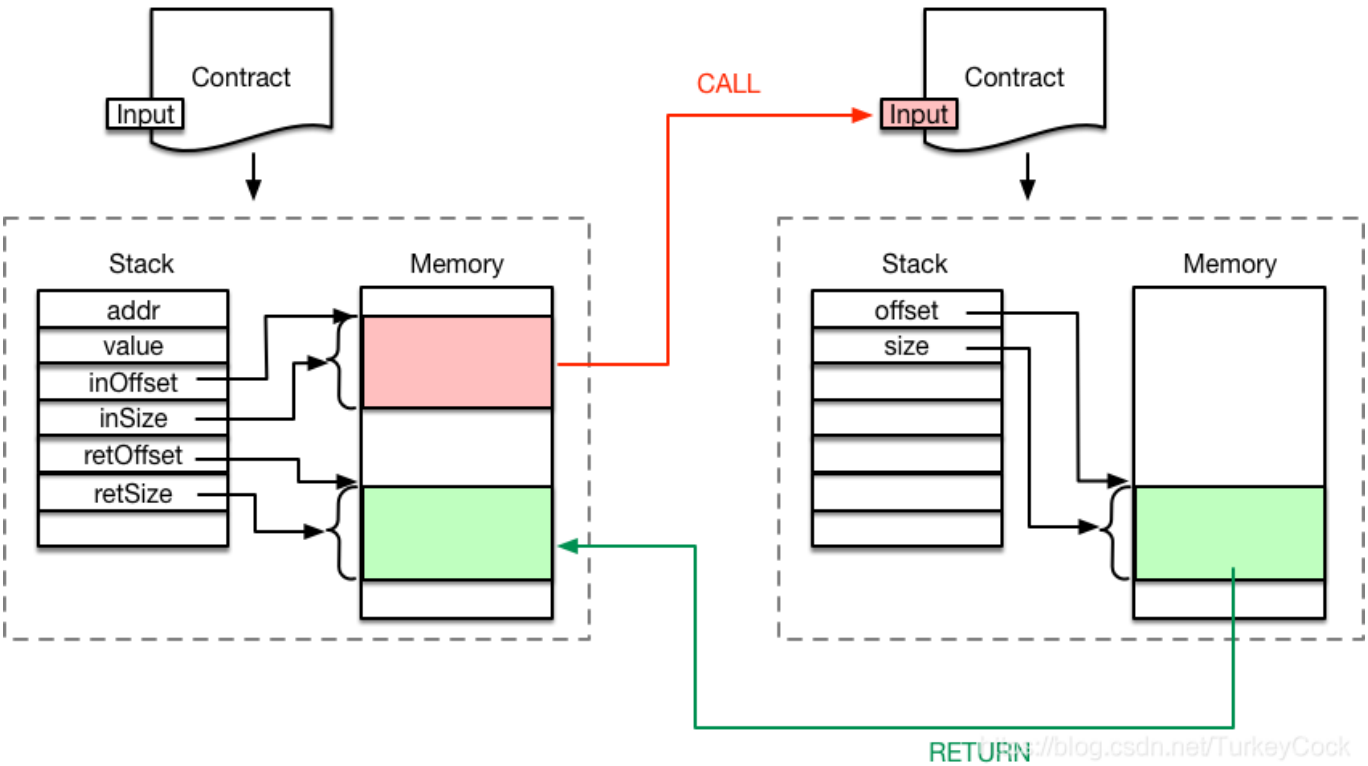


5.合约调用合约

合约内部调用另外一个合约，有4种调用方式：

- CALL
- CALLCODE
- DELEGATECALL
- STATICALL

后面会专门写篇文章比较它们的异同，这里先以最简单的CALL为例，调用流程如下图所示：



可以看到，调用者把调用参数存储在内存中，然后执行CALL指令。

CALL指令执行时会创建新的Contract对象，并以内存中的调用参数作为其Input。

解释器会为新合约的执行创建新的Stack和Memory，从而不会破坏原合约的执行环境。

新合约执行完成后，通过RETURN指令把执行结果写入之前指定的内存地址，然后原合约继续向后执行。

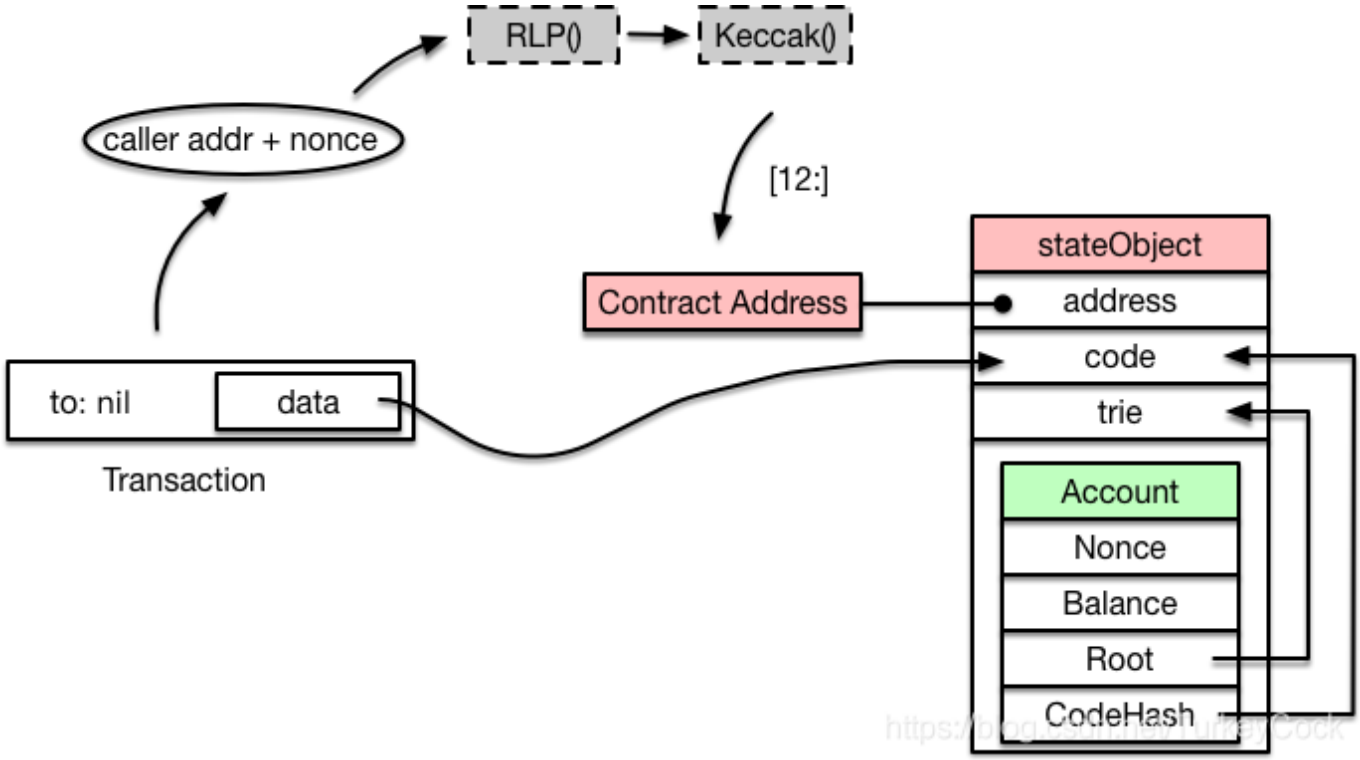
6.创建合约

前面都是讨论的合约调用，那么创建合约的流程时怎么样的呢?

如果某一笔交易的to地址为nil，则表明该交易是用于创建智能合约的。

首先需要创建合约地址，采用下面的计算公式：Keccak(RLP(call_addr, nonce))[:12]。也就是说，对交易发起人的地址和nonce进行RLP编码，再算出Keccak哈希值，取后20个字节作为该合约的地址。

下一步就是根据合约地址创建对应的stateObject，然后存储交易中包含的合约代码。该合约的所有状态变化会存储在一个storage trie中，最终以Key-Value的形式存储到StateDB中。代码一经存储则无法改变，而storage trie中的内容则是可以通过调用合约进行修改的，比如通过SSTORE指令。



7.油费计算

最后啰嗦一下油费的计算，计算公式基本上是根据以太坊黄皮书中的定义：http://gavwood.com/paper.pdf (http://gavwood.com/paper.pdf)

(220)

$$C(\boldsymbol{\sigma},\boldsymbol{\mu},I) \equiv C_{mem}(\boldsymbol{\mu}_i') - C_{mem}(\boldsymbol{\mu}_i) + \begin{cases} C_{\text{SSTORE}}(\boldsymbol{\sigma},\boldsymbol{\mu}) & \text{if} \quad w = \text{SSTORE} \\ G_{exp} & \text{if} \quad w = \text{EXP} \wedge \boldsymbol{\mu}_{\mathbf{s}}[1] = 0 \\ G_{exp} + G_{expbyte} \times (1 + \lfloor \log_{256}(\boldsymbol{\mu}_{\mathbf{s}}[1]) \rfloor) & \text{if} \quad w = \text{EXP} \wedge \boldsymbol{\mu}_{\mathbf{s}}[1] > 0 \\ G_{verylow} + G_{copy} \times \lceil \boldsymbol{\mu}_{\mathbf{s}}[2] \div 32 \rceil & \text{if} \quad w = \text{CALLDATACOPY} \vee \text{CODECOPY} \\ G_{extcode} + G_{copy} \times \lceil \boldsymbol{\mu}_{\mathbf{s}}[3] \div 32 \rceil & \text{if} \quad w = \text{EXTCODECOPY} \\ G_{log} + G_{logdata} \times \boldsymbol{\mu}_{\mathbf{s}}[1] & \text{if} \quad w = \text{LOG0} \\ G_{log} + G_{logdata} \times \boldsymbol{\mu}_{\mathbf{s}}[1] + G_{logtopic} & \text{if} \quad w = \text{LOG1} \\ G_{log} + G_{logdata} \times \boldsymbol{\mu}_{\mathbf{s}}[1] + 2G_{logtopic} & \text{if} \quad w = \text{LOG2} \\ G_{log} + G_{logdata} \times \boldsymbol{\mu}_{\mathbf{s}}[1] + 3G_{logtopic} & \text{if} \quad w = \text{LOG3} \\ G_{log} + G_{logdata} \times \boldsymbol{\mu}_{\mathbf{s}}[1] + 4G_{logtopic} & \text{if} \quad w = \text{LOG4} \\ C_{\text{CALL}}(\boldsymbol{\sigma},\boldsymbol{\mu}) & \text{if} \quad w = \text{CALL} \vee \text{CALLCODE} \vee \text{DELEGATECALL} \\ C_{\text{SUICIDE}}(\boldsymbol{\sigma},\boldsymbol{\mu}) & \text{if} \quad w = \text{SUICIDE} \\ G_{create} & \text{if} \quad w = \text{CREATE} \\ G_{sha3} + G_{sha3word} \lceil \mathbf{s}[1] \div 32 \rceil & \text{if} \quad w = \text{SHA3} \\ G_{jumpdest} & \text{if} \quad w = \text{JUMPDEST} \\ G_{sload} & \text{if} \quad w = \text{SLOAD} \\ G_{zero} & \text{if} \quad w \in W_{zero} \\ G_{base} & \text{if} \quad w \in W_{base} \\ G_{verylow} & \text{if} \quad w \in W_{verylow} \\ G_{low} & \text{if} \quad w \in W_{low} \\ G_{mid} & \text{if} \quad w \in W_{mid} \\ G_{high} & \text{if} \quad w \in W_{high} \\ G_{extcode} & \text{if} \quad w \in W_{extcode} \\ G_{balance} & \text{if} \quad w = \text{BALANCE} \\ G_{blockhash} & \text{if} \quad w = \text{BLOCKHASH} \end{cases}$$

(221)

$$w \equiv \begin{cases} I_{\mathbf{b}}[\boldsymbol{\mu}_{pc}] & \text{if} \quad \boldsymbol{\mu}_{pc} < \|I_{\mathbf{b}}\| \\ \text{STOP} & \text{otherwise} \end{cases}$$

where:

(222)

$$C_{mem}(a) \equiv G_{memory} \cdot a + \left\lfloor \frac{a^2}{512} \right\rfloor$$

with C_{CALL}, C_{SUICIDE} and C_{SSTORE} as specified in the appropriate section below. We define the following subsets of instructions:

$W_{zero} = \{\text{STOP, RETURN}\}$
$W_{base} = \{\text{ADDRESS, ORIGIN, CALLER, CALLVALUE, CALLDATASIZE, CODESIZE, GASPRICE, COINBASE, TIMESTAMP, NUMBER, DIFFICULTY, GASLIMIT, POP, PC, MSIZE, GAS}\}$
$W_{verylow} = \{\text{ADD, SUB, NOT, LT, GT, SLT, SGT, EQ, ISZERO, AND, OR, XOR, BYTE, CALLDATALOAD, MLOAD, MSTORE, MSTORE8, PUSH*, DUP*, SWAP*}\}$
$W_{low} = \{\text{MUL, DIV, SDIV, MOD, SMOD, SIGNEXTEND}\}$
$W_{mid} = \{\text{ADDMOD, MULMOD, JUMP}\}$
$W_{high} = \{\text{JUMPI}\}$
$W_{extcode} = \{\text{EXTCODESIZE}\}$

当然你可以直接read the fucking code，代码位于core/vm/gas.go和core/vm/gas_table.go中。

好，今天就聊到这里吧。

更多文章欢迎关注“鑫鑫点灯”专栏：https://blog.csdn.net/turkeycock (https://blog.csdn.net/turkeycock)

或关注飞久微信公众号：

智能推荐

搜索框测试用例_m0_52622766的博客-程序员资料_搜索框测试用例 (/article/m0_52622766/5135094)

界面测试：1、查看显示是否正确，布局是否合理2、查看是否有错别字功能测试：1、在搜索框中输入列表中存在的内容2、中输入列表中不存在的内容3、不输入内容4、输入空格5、输入超长字符串6、输入搜索框限制长度的字符串N个字符7、输入限制长度的字符串N-1个字符8、输入限制长度的字符串N+1个字符9、在输入的关键字后面，加空格10、在输入的关键字前面，加空格11、在输入的关键字中间，加空格12、输入全角符号的中文13、输入半角符号的中文14、输入全角符号的英文(大、小写字母).1

ajax怎样跨域发送请求数据库,ajax 如何跨域 post请求数据库_换个宇宙的博客-程序员资料 (/article/weixin_32487557/5135082)

ajax 如何跨域 post请求数据库 内容精选换一换该API用于新增数据库。您可以在API Explorer中调试该接口。URI格式：POST /v1.0/{project_id}/databasesPOST /v1.0/{project_id}/databases参数说明URI参数参数名称是否必选参数类型说明project_id是String项目编号，用于资源隔离。获取方式请参考获取项目aja...

全面精通Web 2.0，做互联网潮头人_weixin_30367169的博客-程序员资料 (/article/weixin_30367169/5135079)

来源于：www.toptalk.com.cn Web 2.0? Web 2.0 正在让互联网逐渐找回Internet的真正含义：平等、交互，去中心化。你不应该只是互联网的读者，你也应该是互联网的作者； 你不该只是在互联网上冲浪，你本身就是波浪制造者。Web 2.0 之于Web 1.0 ，如同分布式计算之于集中式计算，网格之于大型主机。 WEB2.0概念诠释 Web2.0，是相对Web1.0...

命令注入新玩法：巧借环境渗透测试目标_bylfsj的博客-程序员资料 (/article/bylfsj/5135078)

*本文作者：yangyangwithgnu，本文属 FreeBuf 原创奖励计划，未经许可禁止转载。在一次漏洞赏金活动中，挖掘到一个不标准的命令注入漏洞，我无法用命令分隔符、命令替换符注入新命令让系统执行，所以，从”型态”上讲，它不算是命令注入漏洞；但我又可以借助目标环境让载荷到达系统命令行，实现读写文件、执行新命令，所以，”神态”来看，...

samba不允许一个用户使用一个以上用户名与一个服务器或共享资源的多重_weixin_34150503的博客-程序员资料 (/article/weixin_34150503/5135074)

今天在虚拟机Redhat enterprise as 4.0中架设了samba服务器，samba服务器安装成功后。期初修改smb.conf文件中的security=share 时，共享文件public在客户端访问均正常，但后期修改security= user，设置valid users =@sales后，客户端访问public共享文件时，却出现下图异常：...

学习hadoop的好文章_nxcjh321的专栏-程序员资料 (/article/nxcjh321/5135073)

http://sishuok.com/forum/blogCategory/showByCategory.html?categories_id=103&user_id=8636

随便推点

数据结构——单链表-不带头结点尾插法_末世灯光的博客-程序员资料_不带头结点的单链表尾插法 (/article/qq_25368751/108025458)

#include<stdio.h>#include <malloc.h>typedef struct LNode{ int data; struct LNode *next;}LNode, *LinkList;void creatL(LinkList &L,int n){ LNode *r; r = NULL; L = NULL; for(int i =1;i<=n;i++){ LNode *p = (LinkList)malloc(s...

nginx启动后，访问报403错误_我弟是个程序员-程序员资料_nginx报403错误 (/article/She_lock/79536857)

出现这种错有很多原因，文件缺失，比如在相应的配置位置没有找到 index.html 文件，也有权限的问题Permission deniedPermission denied错误查看nginx日志，路径为 /var/log/nginx/error.log，发现日志报错Permission denied。切换到目录下cd /var/log/nginx/ 查看错...

反编译apk_编程人生的博客-程序员资料 (/article/Z865785437029/86029731)

1.查看xml文件使用apktool工具，使用此 apktool d *.apk来编译如果报如下错 Microsoft Windows [版本 6.1.7601] 版权所有 (c) 2009 Microsoft Corporation。保留所有权利。C:\Users\Administrator\Desktop\apktool>apktool d 信手书....

利用EditPlus可视化编辑Linux系统上的文件_我弟是个程序员-程序员资料 (/article/She_lock/79925650)

你们一定也这么想过，linux上的黑框操作，一切的编辑工作，都是分外繁琐，有什么工具能不能帮我们更方便的编辑Linux上的文件，就像是在可视化的windows操作系统一样操作。还真有，用EditPlus就可以实现这一需求。【File】—>【FTP】—>【FTP Settings】 【Add】,填写远程服务器ip，登录名以及密码：点击【Advanced Opti...

Ubuntu18.04修复grub引导_Ice_Jeffrey的博客-程序员资料_ubuntu修复grub引导 (/article/Ice_Jeffrey/107509425)

Ubuntu 18.04修复grub引导制作Ubuntu启动盘设置BIOS修复前的准备正式对grub进行修复查看自己电脑的分区根据个人的分区进行挂载插入链接与图片如何插入一段漂亮的代码片生成一个适合你的列表创建一个表格设定内容居中、居左、居右SmartyPants创建一个自定义列表如何创建一个注脚注释也是必不可少的KaTeX数学公式新的甘特图功能，丰富你的文章UML 图表FLowchart流程图导出与导入导出导入之前不小心在喝水时把水撒进了电脑里面，之后返厂维修。在维修了主板后，发现开机直接进入了Wind

---

pdf书籍添加目录_少爷想养猫的博客-程序员资料 (/article/weixin_44180216/89300363)

下载一个软件，freepic2pdf,点击更改pdf。选择你要添加书签的书籍点击取出书签，得到如下两个文件，itf文件里面的Basepage决定书签中的页码在pdf的页数的起始点。例如：设定12，则书签里的页码1，代表的是pdf文件的第12页。txt文件则是储存书签的格式找到书籍的目录信息，例如在豆瓣读书里获取目录可以看出，这里直接拷贝的格式是无法直接使用的使用notepad+...

推荐文章

比较字符串s1和s2,若s1>s2，输出一个正数，若s1=s2,输出0,若s1<s2,输出一个负数。不用strcpy函数_YHY的专栏-程序员资料 (/article/ZX_YHY/44520579)

OMPL 安装与使用_zgh-程序员资料_ompl (/article/zghforever/106688410)

spark1.1.0学习路线_mmicky的hadoop、Spark世界-程序员资料 (/article/book_mmicky/40425541)

程序员装B指南v1.0_110818_weixin_33862514的博客-程序员资料 (/article/weixin_33862514/91606413)

轴承故障诊断matlab实现,基于SVM的齿轮箱轴承故障诊断(含matlab程序)_勤小墨的博客-程序员资料 (/article/weixin_31440211/115820265)

【稀疏矩阵转置】线性时间复杂度实现稀疏矩阵转置_BlessingXRY的博客-程序员资料_稀疏矩阵转置的时间复杂度 (/article/BlessingXRY/78297096)

"这是上级规定"的奥秘_weixin_33674976的博客-程序员资料 (/article/weixin_33674976/85208234)

Macbook pro2020配置maven、IDEA配置maven_PANDA博客-程序员资料_macbook配置idea (/article/qq_38685503/118468593)

热门文章

Python 装饰器计算函数或方法执行时间_ssjdoudou的博客-程序员资料_装饰器计算函数执行时间 (/article/ssjdoudou/104123129)

attention机制__天明_的博客-程序员资料 (/article/qq_32256033/89889135)

oracle 字段加密解密方法,oracle 字段加密解密方法_Ssiya的博客-程序员资料 (/article/weixin_30607173/116472162)

idea修改maven默认仓库不生效_王凯冲的博客-程序员资料 (/article/weixin_43154932/122002476)

维纳滤波及其简单实现_游戏里的编程游戏-程序员资料_维纳滤波实现 (/article/billy145533/102833469)

手把手教你用Pycharm连接远程Python环境_pdcfighting的博客-程序员资料 (/article/pdcfighting/113577959)

curl_easy_setopt参数详细介绍！_YearnWang的专栏-程序员资料_curl_easy_setopt (/article/doubleuto/8859687)

基于Socket的TCP长连接（服务端Java+客户端Android），Service配合AIDL实现_Lone_Star斌 的博客-程序员资料 (/article/qq_29405933/68489174)

相关标签

虚拟机 (/searchArticle?qc=虚拟机&page=1)

以太坊 (/searchArticle?qc=以太坊&page=1)

以太坊源码 (/searchArticle?qc=以太坊源码&page=1)

以太坊源码分析 (/searchArticle?qc=以太坊源码分析&page=1)

EVM (/searchArticle?qc=EVM&page=1)

图解 (/searchArticle?qc=图解&page=1)