(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106407430 B
(45)授权公告日 2019.09.03

(21)申请号 201610857670.4

(22)申请日 2016.09.27

(65)同一申请的已公布的文献号
申请公布号 CN 106407430 A

(43)申请公布日 2017.02.15

(73)专利权人 北京天德科技有限公司
地址 100089 北京市海淀区知春路113号1708-048

(72)发明人 邓恩艳

(51)Int.Cl.
G06F 16/27(2019.01)
G06F 16/23(2019.01)
G06Q 40/04(2012.01)

(56)对比文件
CN 105893042 A,2016.08.24,
CN 105809062 A,2016.07.27,
US 2016217436 A1,2016.07.28,

审查员 何华

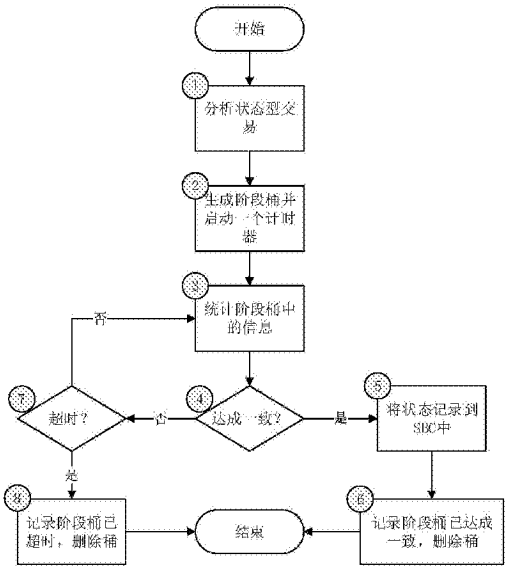
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种基于阶段桶的复杂智能合约状态同步方法

(57)摘要

本发明提供了一种基于阶段桶的复杂智能合约状态同步方法,包括如下步骤:(1)判断状态类型交易,确定需要更新状态的账户地址;(2)根据状态交易的信息生成阶段桶,然后为每个阶段桶设定一个计时器;(3)统计阶段桶内的状态信息,并分别统计每类信息的条数;(4)检查步骤(3)的结果,确定某阶段桶是否已达成一致,如果已达成一致,继续执行步骤(5),否则继续执行步骤(7);(5)将状态存储到状态区块链中;(6)标记该阶段桶为“已达成一致”,然后删除该阶段桶;(7)检查阶段桶的计时器是否已超时,如果未超时则跳转执行步骤(3),否则继续执行步骤(8);(8)标记该阶段桶为“已超时”,然后删除该阶段桶,此时的阶段桶称为“废桶”。



CN 106407430 B

1.一种基于阶段桶的复杂智能合约状态同步方法,其特征在于,包括如下步骤:

(1)判断状态类型交易,确定需要更新状态的账户地址;

(2)根据状态交易的信息生成阶段桶,然后为每个阶段桶设定一个计时器;

(3)统计阶段桶内的状态信息,并分别统计每类信息的条数;

(4)检查步骤(3)的结果,确定某阶段桶是否已达成一致,如果已达成一致,继续执行步骤(5),否则继续执行步骤(7);

(5)将状态存储到状态区块链中;

(6)标记该阶段桶为“已达成一致”,然后删除该阶段桶;

(7)检查阶段桶的计时器是否已超时,如果未超时则跳转执行步骤(3),否则继续执行步骤(8);

(8)标记该阶段桶为“已超时”,然后删除该阶段桶,此时的阶段桶称为“废桶”。

2.根据权利要求1所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:步骤(1)的所述交易是从交易区块链发来的。

3.根据权利要求1所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:步骤(2)中所述信息包括合约账户的地址、该合约账户总共被触发执行的次数以及需要更改的账户地址。

4.根据权利要求1所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:步骤(3)所述统计阶段桶内的状态信息是对阶段桶内的来自交易区块链不同节点的相同内容信息进行归类。

5.根据权利要求1所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:步骤(4)所述确定某阶段桶是否已达成一致是确定否有超过2/3的合约执行节点拥有相同的状态。

6.根据权利要求1所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:步骤(8)所述“废桶”是由某个阶段桶中的意见长期无法达成一致产生的。

7.根据权利要求6所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:所述废桶是由于桶内已经收集了所有执行合约的节点的意见,但相同的状态信息总数小于总节点数的2/3而没有达成一致所产生的。

8.根据权利要求6所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:所述废桶是由于执行智能合约的个别节点执行速度慢,当它将自己的状态型交易(sTx,State Transaction,明确目前合约的执行状态)发送到状态区块链时,状态区块链已经完成了对该阶段的状态统计,超过2/3的节点已经对该阶段的状态改变达成了一致,并已经销毁了对应的阶段桶而产生的,状态区块链会为该条sTx新建一个阶段桶并等待永远无法完成的状态统计。

9.根据权利要求6所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:所述废桶是由于某些sTx未能到达状态区块链,或者某些节点未能成功发送sTx导致对某阶段的状态统计达不到执行合约总节点数的2/3以上所产生的。

10.根据权利要求7所述的一种基于阶段桶的复杂智能合约状态同步方法,其特征在于:所述废桶产生后,对于该合约的后续状态无论是否一致都不再统计,记录本次合约的源账户地址和标记了本次执行是合约第几次被激活执行的本次的执行序号,如果后续sTx是

对同一合约的相同执行序号的状态信息，则直接丢弃。

11.根据权利要求8所述的一种基于阶段桶的复杂智能合约状态同步方法，其特征在于：执行智能合约的个别节点执行速度慢时，对于最近一段时间内已经确认达成一致的阶段桶，系统动态维护一张“状态表”，销毁该阶段桶时在表中记录了桶的标记名，以及销毁时桶内累计的意见总数，如果随后的sTx对应标记名与表中记录的标记名匹配，直接丢弃该sTx并将累计在该标记名的意见总数加1，当一条阶段桶缓存记录上所累积的意见总数达到执行合约的所有节点的节点点数，即执行合约的所有节点已经执行完这个阶段，或者该缓存记录超过了一定时间限制时，将记录写入日志，然后在“状态表”中清除它。

12.根据权利要求9所述的一种基于阶段桶的复杂智能合约状态同步方法，其特征在于：某些sTx未能到达状态区块链，或者某些节点未能成功发送sTx时，系统在每次新建阶段桶的时候都会启动一个计时器，当一个阶段桶经过一段时间后还没有达成状态一致时，系统将销毁阶段桶。

一种基于阶段桶的复杂智能合约状态同步方法

技术领域

[0001] 本发明涉及一种区块链技术，特别是一种基于阶段桶的复杂智能合约状态同步方法。

背景技术

[0002] 所谓复杂智能合约，就是执行时间长，逻辑较复杂的合约，通常具有多个阶段。在区块链系统中，智能合约需要部署在所有节点上，并且智能合约在每次执行时也需要系统中所有节点来同时执行，以使得所有的节点状态保持一致。然而，在实际中，区块链系统中的各个节点的环境可能各有不同，智能合约在各个节点运行的速度可能各不相同，加之逻辑复杂，智能合约可能会在运行中的任何一个阶段改变其所在节点的状态，而现有技术的方案并没有针对只能合约节点环境存在差异情况下的智能合约执行方法，因此会产生合约状态异步，区块链系统对于支持复杂智能合约执行的能力低的技术缺陷，同时，多个独立节点同时执行合约时数据一致性，结果统一性，数据的完整性以及数据的隔离性难以保证，数据同步会相互干扰。

发明内容

[0003] 本发明的目的在于提供一种基于阶段桶的复杂智能合约状态同步方法，允许执行智能合约的各节点环境存在差异，并且在执行同一个智能合约时的速度存在差异，包括如下步骤：(1)判断状态类型交易，确定需要更新状态的账户地址；(2)根据状态交易的信息生成阶段桶，然后为每个阶段桶设定一个计时器；(3)统计阶段桶内的状态信息，并分别统计每类信息的条数；(4)检查步骤(3)的结果，确定某阶段桶是否已达成一致，如果已达成一致，继续执行步骤(5)，否则继续执行步骤(7)；(5)将状态存储到状态区块链中；(6)标记该阶段桶为“已达成一致”，然后删除该阶段桶；(7)检查阶段桶的计时器是否已超时，如果未超时则跳转执行步骤(3)，否则继续执行步骤(8)；(8)标记该阶段桶为“已超时”，然后删除该阶段桶，此时的阶段桶称为“废桶”。

[0004] 优选的，步骤(1)的所述交易是从交易区块链发来的。

[0005] 优选的，步骤(2)中所述信息包括合约账户的地址、该合约账户总共被触发执行的次数以及需要更改的账户地址，

[0006] 优选的，步骤(3)所述统计阶段桶内的状态信息是对阶段桶内的来自交易区块链不同节点的相同内容信息进行归类。

[0007] 优选的，步骤(4)所述确定某阶段桶是否已达成一致是确定否有超过2/3的合约执行节点拥有相同的状态。

[0008] 优选的，步骤(8)所述“废桶”是由某个阶段桶中的意见长期无法达成一致产生的。

[0009] 优选的，废桶是由于桶内已经收集了所有执行合约的节点的意见，但相同的状态信息总数小于总节点数的2/3而没有达成一致所产生的。

[0010] 优选的，废桶是由于执行智能合约的个别节点执行速度慢，当它将自己的状态型

交易(sTx,State Transaction,明确目前合约的执行状态)发送到状态区块链时,状态区块链已经完成了对该阶段的状态统计,超过2/3的节点已经对该阶段的状态改变达成了一致,并已经销毁了对应的阶段桶而产生的,状态区块链会为该条sTx新建一个阶段桶并等待永远无法完成的状态统计。

[0011] 优选的,废桶是由于某些sTx未能到达状态区块链,或者某些节点未能成功发送sTx导致对某阶段的状态统计达不到执行合约总节点数的2/3以上所产生的。

[0012] 优选的,废桶产生后,对于该合约的后续状态无论是否一致都不再统计,记录本次合约的源账户地址和标记了本次执行是合约第几次被激活执行的本次的执行序号,如果后续sTx是对同一合约的相同执行序号的状态信息,则直接丢弃。

[0013] 优选的,执行智能合约的个别节点执行速度慢时,对于最近一段时间内已经确认达成一致的阶段桶,系统动态维护一张“状态表”,销毁该阶段桶时在表中记录了桶的标记名,以及销毁时桶内累计的意见总数,如果随后的sTx对应标记名与表中记录的标记名匹配,直接丢弃该sTx并将累计在该标记名的意见总数加1,当一条阶段桶缓存记录上所累积的意见总数达到执行合约的所有节点的节点点数,即执行合约的所有节点已经执行完这个阶段,或者该缓存记录超过了一定时间限制时,将记录写入日志,然后在“状态表”中清除它。

[0014] 优选的,某些sTx未能到达状态区块链,或者某些节点未能成功发送sTx时,系统在每次新建阶段桶的时候都会启动一个计时器,当一个阶段桶经过一段时间后还没有达成状态一致时,系统将销毁阶段桶。

[0015] 本发明通过在单独存储状态的区块链上使用“阶段桶”的办法来保证执行智能合约的各节点状态的一致,区块链系统对于支持复杂智能合约执行的能力提高,同时,多个独立节点同时执行合约时数据一致性,结果统一性,数据的完整性以及数据的隔离性的以保证,数据同步不会相互干扰。

[0016] 根据下文结合附图对本发明具体实施例的详细描述,本领域技术人员将会更加明了本发明的上述以及其他目的、优点和特征。

附图说明

[0017] 后文将参照附图以示例性而非限制性的方式详细描述本发明的一些具体实施例。附图中相同的附图标记标示了相同或类似的部件或部分。本领域技术人员应该理解,这些附图未必是按比例绘制的。本发明的目标及特征考虑到如下结合附图的描述将更加明显,附图中:

[0018] 图1是根据本发明实施例的简单账户结构示意图;

[0019] 图2是根据本发明实施例的复杂合约账户结构示意图;

[0020] 图3是根据本发明实施例的状态同步流程示意图。

具体实施方式

[0021] 在进行具体实施方式的说明之前,为了更为清楚的表达所论述的内容,首先说明本发明所涉及的一些重要概念。

[0022] 1、账户

[0023] 智能合约附属于某个账户，账户的结构描述如下：

[0024] (1)激活状态：账户是否已被激活。0代表未被激活，1代表已激活。如果该字段为0，则账户不能发送交易，并且忽略除激活信息外一切发送到它的交易信息，直到重新被激活；

[0025] (2)创建时间和更新时间：记录了此账户被创建的时间和上一次更新的时间；

[0026] (3)合约哈希值：对合约账户而言，这个字段记录了与该账户关联的合约代码的哈希值，账户一旦生成，这个值就不再改变；

[0027] (4)nonce：一个整数，记录从该账户地址共发出了多少交易；

[0028] (5)number：记录有多少交易发送到了该账户地址，记录了合约被激活执行的次数；

[0029] (6)账户类型：分为两种类型，1表示普通合约账户，2表示复杂合约账户；

[0030] (7)账户地址：一个20位长的字符串，在系统中唯一标识一个账户采用的是非对称加密技术生成的公钥和私钥，然后对公钥进行md5哈希，取最后20个字符作为地址；

[0031] (8)账户缓存：暂时记录合约执行过程中产生的临时数据。

[0032] 对于简单的合约，完全可以将编译完成后的二进制数据直接存储在区块链上，以提高加载速度。但是对于复杂智能合约，如果将合约执行相关的文件直接存储在区块链上将可能造成区块体积过大，因此复杂合约只在账户中保存了相关文件的地址以节省空间，真正的文件保存在区块链外。账户的结构如附图1和图2所示。

[0033] 2、交易区块链和状态区块链

[0034] 交易区块链接收来自区块链系统外部的信息，并将这些信息存储至区块中，智能合约的执行在交易区块链的节点中完成。

[0035] 状态区块链专门存储状态信息，维护账户信息，它负责智能合约执行过程中和执行完毕后的状态同步及存储。

[0036] 3、状态型交易

[0037] 在智能合约执行的过程中，每当有状态改变的操作时，各执行合约节点都会向专门存储状态的区块链发送状态型交易(sTx)来进行状态同步。sTx的内容如下：

[0038] (1)源地址：发出该状态交易的账户地址；

[0039] (2)源账户的公钥：对这个公钥进行哈希后取其最后20个字符，应该与源地址相同；

[0040] (3)执行信息：本次合约的执行信息，包括记录本次是合约的第几次执行；

[0041] (4)对象账户地址：改变状态的账户地址；

[0042] (5)状态信息：状态改变信息，与具体的领域有关；

[0043] (6)签名：使用源地址的私钥对对象账户地址、执行信息、状态信息生成的摘要字段的签名；

[0044] (7)时间戳：交易创建的时间

[0045] 4、阶段桶

[0046] 智能合约是在交易区块链的所有节点同时执行，因此执行过程中在进行状态改变操作时，每个节点都会产生一条状态型交易，为了对各个节点的状态改变结果进行汇总统一，维护状态区块链中的状态一致性，系统为智能合约的每个阶段设立一个阶段桶，每个阶段可能有一次或者多次状态改变。

[0047] 复杂类型的智能合约通常拥有较长的执行周期，并且具有多个阶段，当在多个独立的节点同时部署执行时，由于实际中不同节点的环境不同，智能合约执行的速度存在差别，使得对合约执行状态的同步变得困难。比如，一个智能合约分为三个阶段执行，分别为a，b，c，其中每个阶段都可能涉及一次或多次的状态改变。假设当node(i)的b阶段执行完毕时，node(i+1)由于某种原因，其a阶段刚刚执行完毕，那么此时的本地状态无法直接同步，而是需要等待node(i+1)的b阶段执行结束才可以进行，这时使用阶段桶从而减少处理等待时间就显得很有必要性了。

[0048] 阶段桶的实现原理是将阶段桶看成是一种结果统计器。状态区块链的每个节点都会接收到来自交易区块链所有参与合约执行的节点发来的状态交易。对合约执行的每个阶段，交易区块链都会以唯一的标注作为阶段桶的名称(比如以“合约账户名+该合约被触发的次数[即该账户合约执行的总次数]+合约执行阶段的名称或标号”来命名)。

[0049] 在首次收到交易区块链某结点发来的一个新的阶段的状态交易时建立阶段桶，接下来，每次状态区块链节点在收到交易区块链其余节点关于某阶段的状态交易时，就放入对应的阶段桶。由此可见，阶段桶中存放的是来自交易区块链不同节点的相同内容信息。当某个阶段桶内相同的内容信息数量在规定的时限内超过交易区块链节点总数的2/3时，阶段桶的信息内容达成一致，否则认为阶段桶的信息内容无法达成一致，即产生了废桶。

[0050] 引入阶段桶后，解决了在多个节点同时执行合约的状态同步问题，提高了区块链系统对于支持复杂智能合约执行的能力。首先，阶段桶保证了数据的一致性，使得在多个独立节点同时执行合约时，可以得到统一的结果。另外，阶段桶保证了数据的完整性，由于阶段桶的存在，当少数节点出现故障时并不会影响数据的准确性和可靠性。再次，阶段桶保证了数据的隔离性，如前文所述，复杂型合约通常具有较长的实行周期，并且具有多个阶段，阶段桶使得相同合约的不同阶段之间，相同合约在不同执行轮次的相同阶段之间(系统中可能同时有多个复杂型合约在执行，也可能是不同的交易触发了相同合约的多次执行，但他们属于不同的执行轮次)，不同合约之间的数据同步不会相互干扰。

[0051] 5、废桶

[0052] 如果某个阶段桶中的意见长期无法达成一致，就会产生废桶。废桶无法对状态同步做出贡献，应该及时的销毁以避免浪费资源。产生废桶的原因可能有多种：

[0053] 第一，桶内已经收集了所有执行合约的节点的意见，但没有达成一致，即相同的状态信息总数小于总节点数的2/3；

[0054] 第二，执行智能合约的个别节点执行速度慢，当它将自己的sTx发送到状态区块链时，状态区块链已经完成了对该阶段的状态统计，即超过2/3的节点已经对该阶段的状态改变达成了一致，并销毁的对应的阶段桶，按照本发明的策略，状态区块链会为该条sTx新建一个阶段桶并等待永远无法完成的状态统计；

[0055] 第三，由于未知原因，某些sTx未能到达状态区块链，或者某些节点未能成功发送sTx导致对某阶段的状态统计达不到执行合约总节点数的2/3以上。

[0056] 第一种情况属于错误，这说明执行合约的所有节点在执行合约时有总数超过1/3的节点与其他节点的状态不一致，此时对于该合约的后续状态无论是否一致都不再统计，应该记录本次合约的源账户地址和本次的执行序号(标记了本次执行是合约第几次被激活执行)，如果后续sTx是对同一合约的相同执行序号的状态信息，则直接丢弃。

[0057] 第二种情况，对于最近一段时间内已经确认达成一致的阶段桶，系统动态维护一张“状态表”，销毁该阶段桶时在表中记录了桶的标记名，以及销毁时桶内累计的意见总数。如果随后的sTx对应标记名与表中记录的标记名匹配，直接丢弃该sTx并将累计在该标记名的意见总数加1。当一条阶段桶缓存记录上所累积的意见总数达到执行合约的所有节点的总节点数，即执行合约的所有节点已经执行完这个阶段，或者该缓存记录超过了一定时间限制时，将记录写入日志，然后在“状态表”中清除它。

[0058] 第三种情况，系统在每次新建阶段桶的时候都会启动一个计时器，当一个阶段桶经过一段时间后还没有达成状态一致时，系统将销毁阶段桶。理由是，在正常的情况下，执行合约的所有节点各节点对同一个合约的执行速度不应该相差太大，因此大多数节点对同一阶段的状态确认时间间隔不会太久。如果经过很长时间还没有达成一致，那么很可能这个阶段桶以后也无法达成一致。

[0059] 由于状态区块链并不知道一个具体的合约在执行过程中会涉及到多少次的状态改变操作，所以按照前文所述，状态区块链会给每个新出现的阶段创建一个新的阶段桶，并对其进行关于状态一致性的意见统计。但如果阶段桶内的信息长期无法达成一致(即产生了废桶)，并且得不到适当的处理就会造成极大的资源浪费，造成系统性能下降，甚至影响系统的稳定性。因此以上的处理机制是必不可少的。

[0060] 实施例

[0061] 本具体实施例所涉及的系统有多个节点，每个节点都部署相同的智能合约，智能合约作为合约账户的一部分而存在，由于区块链系统各节点信息需要保持一致，因此系统中各节点所部署的智能合约均相同。

[0062] 本具体实施例所涉及的智能合约为复杂类型，通常拥有较长的执行周期，具有多个阶段，并且可能多次进行状态的改变，如果有状态改变，则系统在每个阶段结束时进行一次状态同步。

[0063] 假设区块链系统中的交易区块链有A,B,C,D四个节点，状态区块链有E,F,G,H四个节点，一个智能合约M分为三个阶段执行，分别为a,b,c，其中每个阶段都可能涉及一次或多次的状态改变，智能合约在ABCD四个节点被同时触发执行。对于ABCD每个节点，每一个阶段执行结束，如果此阶段涉及到状态的改变，则改变后的状态需要同步至状态区块链，因此节点ABCD都需要向状态区块链发送sTx来进行状态同步。

[0064] 但是在实际中，不同节点之间不仅运行环境存在差别，执行速度也可能会不一致，即节点ABCD在执行智能合约M时，虽然同时被触发执行，但不一定可以每次都同时完成a,b,c三个阶段。

[0065] 假设在智能合约M被触发执行一段时间后的t1时刻，ABCD四个节点的执行状态如下：

[0066] A节点:阶段a执行完毕，正在执行阶段b，已经向状态区块链发送了sTx(a)；

[0067] B节点:正在执行阶段a，还未向状态区块链发送状态交易信息；

[0068] C节点:阶段a,b执行完毕，正在执行阶段c，已经向状态区块链发送了sTx(a)，sTx(b)；

[0069] D节点:阶段a执行完毕，正在执行阶段b，已经向状态区块链发送了sTx(a)。

[0070] 状态交易以广播方式发向状态区块链的每个节点，因此节点EFGH均可收到由交易

区块链节点ABCD发来的状态交易信息。节点EFGH每个节点执行的操作都相同，以节点E为例，由于t1时刻接到的不同阶段的状态信息共有两类，分别为阶段a和阶段b，所以共有阶段桶2个，t1时刻节点E的阶段桶情况如表1所示：

[0071] 表1 t1时刻节点E的阶段桶情况

[0072]

合约阶段 a 对应的阶段桶	合约阶段 b 对应的阶段桶
A 节点发送的 sTx(a) C 节点发送的 sTx(a) D 节点发送的 sTx(a)	C 节点发送的 sTx(b)

[0073] 可见t1时刻，合约阶段a对应的阶段桶中收集的相同意见已经超过了交易区块链总结点数(本例中是4个节点)的2/3，已经达成一致，sTx(a)将会被写入状态区块链，按照本文前述策略，该阶段桶将被删除。

[0074] 经过一段时间，到达t2时刻后，ABCD四个节点的执行状态如下：

[0075] A节点：阶段a，b执行完毕，正在执行阶段c，t1时刻后，又向状态区块链发送了sTx(b)；

[0076] B节点：阶段a，b，c执行完毕，t1时刻后，又向状态区块链发送了sTx(a)，sTx(b)，sTx(c)；

[0077] C节点：阶段a，b，c执行完毕，t1时刻后，又向状态区块链发送了sTx(c)；

[0078] D节点：阶段a，b执行完毕，正在执行阶段c，t1时刻后，又向状态区块链发送了sTx(b)；

[0079] t2时刻节点E的阶段桶情况如表2所示：

[0080] 表2 t2时刻节点E阶段桶情况

[0081]

合约阶段 b 对应的阶段桶	合约阶段 c 对应的阶段桶
C 节点发送的 sTx(b) A 节点发送的 sTx(b) B 节点发送的 sTx(b) D 节点发送的 sTx(b)	B 节点发送的 sTx(c) C 节点发送的 sTx(c)

[0082] t2时刻，合约阶段b对应的阶段桶已经集齐了交易区块链节点ABCD发来的状态交易，若桶中相同内容的信息数量超过交易区块链节点总数的2/3，该桶就会被删除，sTx(b)则会被写入状态区块链。合约阶段c对应的阶段桶还在等待节点A和D的状态交易，如果超过一定时长，状态区块链将不再等待sTx(c)的同步，则sTx(c)将不会被写入状态区块链，即sTx(c)没有达成一致。

[0083] 虽然本发明已经参考特定的说明性实施例进行了描述，但是不会受到这些实施例的限定而仅仅受到附加权利要求的限定。本领域技术人员应当理解可以在不偏离本发明的保护范围和精神的情况下对本发明的实施例能够进行改动和修改。

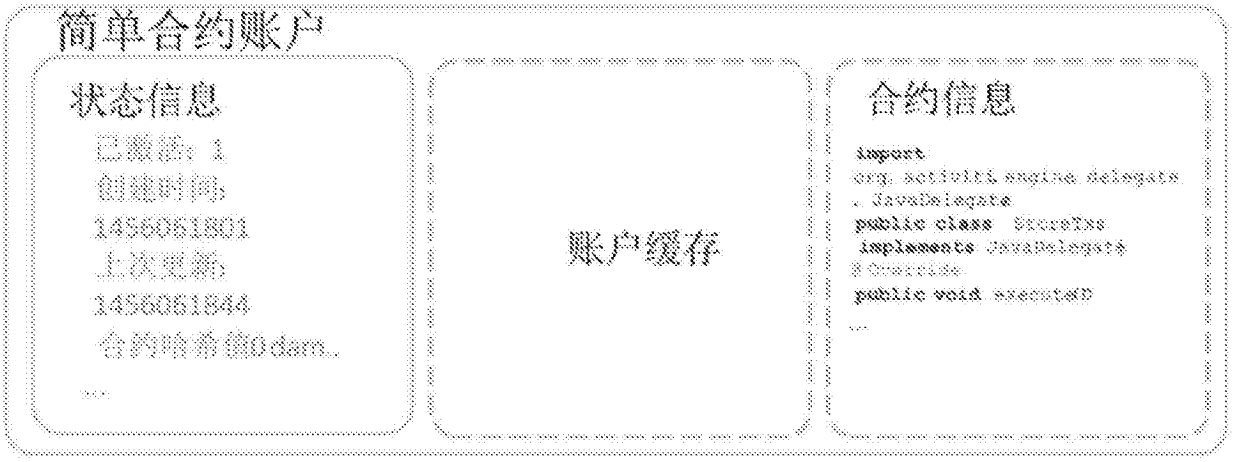

简单合约账户
状态信息
账户缓存
合约信息
import
public class
implements
public void

图1

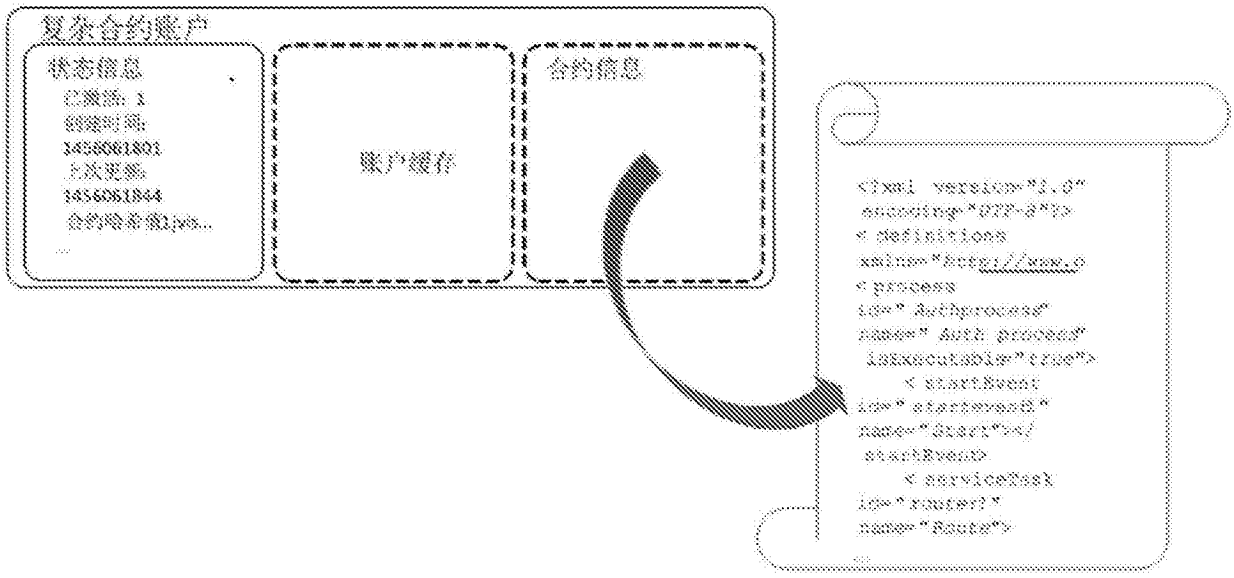

复杂合约账户
状态信息
账户缓存
合约信息

图2

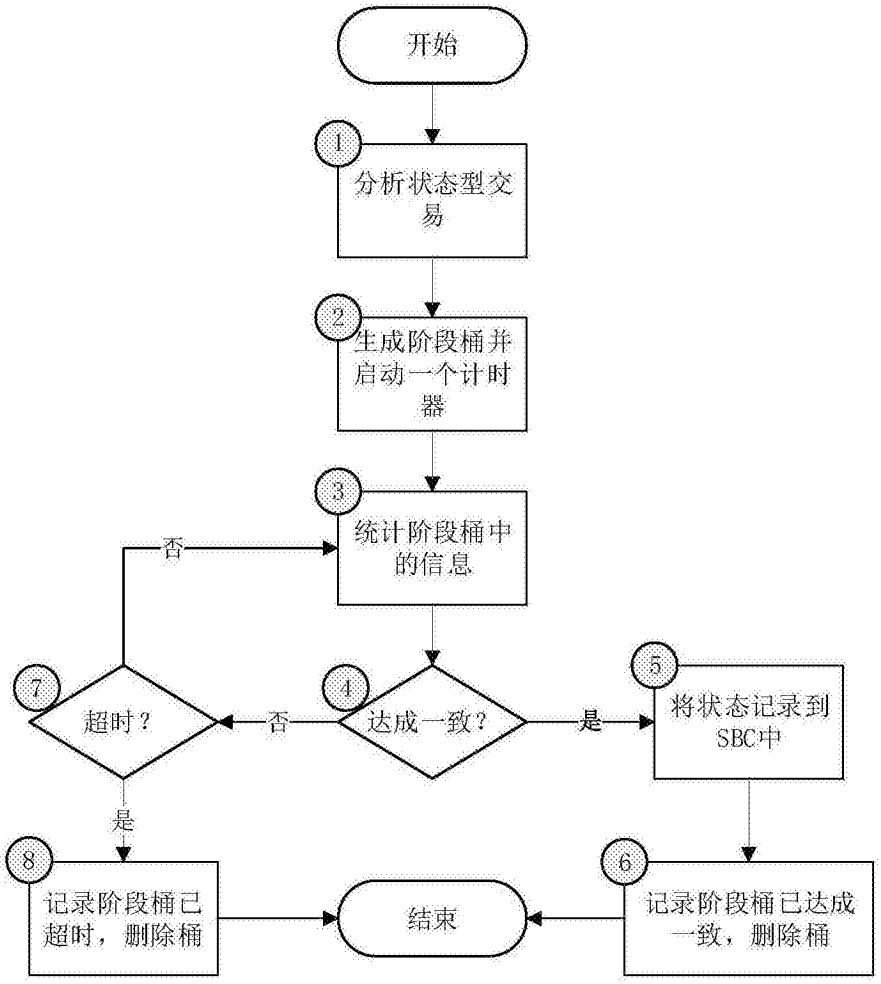

开始
1
分析状态型交易
2
生成阶段桶并启动一个计时器
3
统计阶段桶中的信息
否
4
达成一致？
是
5
将状态记录到SBC中
否
7
超时？
是
8
记录阶段桶已超时，删除桶
结束
6
记录阶段桶已达成一致，删除桶

图3