



(12) 发明专利

(10) 授权公告号 CN 111522561 B

(45) 授权公告日 2023. 06. 06

(21) 申请号 202010152017.4

(22) 申请日 2020.03.06

(65) 同一申请的已公布的文献号  
申请公布号 CN 111522561 A

(43) 申请公布日 2020.08.11

(73) 专利权人 杜晓楠  
地址 新加坡大牌233碧山22街门牌05-132,  
57023

(72) 发明人 杜晓楠

(74) 专利代理机构 深圳市顺天达专利商标代理  
有限公司 44217  
专利代理师 邹秋菊

(51) Int. Cl.  
G06F 8/65 (2018.01)  
G06F 11/14 (2006.01)

(56) 对比文件

- CN 109814905 A, 2019.05.28
- US 2006195836 A1, 2006.08.31
- CN 109992285 A, 2019.07.09
- CN 109166037 A, 2019.01.08
- WO 2017028375 A1, 2017.02.23
- WO 2019232789 A1, 2019.12.12
- US 2019303353 A1, 2019.10.03
- CN 109669709 A, 2019.04.23
- EP 2083354 A1, 2009.07.29
- US 2019058581 A1, 2019.02.21

审查员 王田园

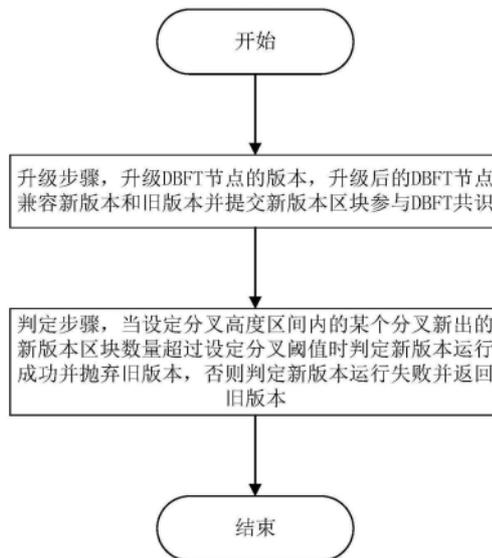
权利要求书1页 说明书6页 附图3页

(54) 发明名称

DBFT分布式网络中平滑向后兼容升级的方法、计算机可读存储介质和DBFT网络

(57) 摘要

本发明涉及一种DBFT分布式网络中平滑向后兼容升级的方法,包括:升级步骤,升级DBFT节点的版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识;判定步骤,当设定分叉高度区间内的某个分叉出的新版本区块数量超过设定分叉阈值时判定新版本运行成功并抛弃旧版本,否则判定新版本运行失败并返回旧版本。本发明还涉及一种计算机可读存储介质和DBFT分布式网络。实施本发明的DBFT分布式网络中平滑向后兼容升级的方法、计算机可读存储介质和DBFT分布式网络,可以同时运行两个版本的共识,其中新版本向后兼容旧版本,从而平滑向后兼容升级。



1. 一种DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,包括:

升级步骤,升级DBFT节点的版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识;

判定步骤,当设定分叉高度区间内的某个分叉值新出的新版本区块数量超过设定分叉阈值时判定新版本运行成功并抛弃旧版本,否则判定新版本运行失败并返回旧版本;

所述升级步骤进一步包括:

S1、判定DBFT节点是否拒绝任何新版本,如果是直接判定新版本运行失败;否则执行步骤S2;

S2、判定DBFT节点是否配置有节点分叉阈值,如果是则执行步骤S3,否则执行步骤S4;

S3、判定所述DBFT节点接收到的区块的阈值是否大于或等于所述节点分叉阈值,如果是执行步骤S4,否则直接判定新版本运行失败;

S4、升级DBFT节点的版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识。

2. 根据权利要求1所述的DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,在所述步骤S2中,所述节点分叉阈值包括最小分叉执行高度区间和/或最小分叉激活阈值,所述区块的阈值包括区块分叉执行高度区间和/或区块分叉激活阈值。

3. 根据权利要求2所述的DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,所述步骤S3进一步包括:

S31、判定所述DBFT节点接收到的所述区块分叉执行高度区间是否大于所述最小分叉执行高度区间,如果是则执行步骤S4,否则直接判定新版本运行失败。

4. 根据权利要求2所述的DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,所述步骤S3进一步包括:

S32、判定所述DBFT节点接收到的所述区块分叉激活阈值是否大于所述最小分叉激活阈值,如果是则执行步骤S4,否则直接判定新版本运行失败。

5. 根据权利要求1-4中任意一项所述的DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,所述判定步骤进一步包括:

Sa、当位于分叉执行开始高度和分叉执行结束高度之间的分叉高度区间内的某个分叉值新出的新版本区块的个数大于或等于设定分叉区块阈值时判定新版本运行成功并抛弃旧版本。

6. 根据权利要求5所述的DBFT分布式网络中平滑向后兼容升级的方法,其特征在于,所述判定步骤进一步包括:

Sb、当前区块高度高于分叉执行结束高度,但新出的新版本区块的个数小于设定分叉区块阈值时,判定新版本运行失败,且切换回旧版本。

7. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现根据权利要求1-6中任意一项权利要求所述的DBFT分布式网络中平滑向后兼容升级的方法。

8. 一种DBFT网络,包括多个DBFT节点,所述DBFT节点上存储有计算机程序,其特征在于,所述程序被处理器执行时实现根据权利要求1-6中任意一项权利要求所述的DBFT分布式网络中平滑向后兼容升级的方法。

## DBFT分布式网络中平滑向后兼容升级的方法、计算机可读存储介质和DBFT网络

### 技术领域

[0001] 本发明涉及分布式网络领域,更具体地说,涉及一种DBFT分布式网络中平滑向后兼容升级的方法、计算机可读存储介质和DBFT网络。

### 背景技术

[0002] DBFT (Delegated Byzantine Fault Tolerance) 是一种通用的共识机制模块,提出了一种改进的拜占庭容错算法,使其能够适用于区块链系统。这种共识机制是在Castro和Liskov提出的“实用拜占庭容错算法”(Practical Byzantine Fault Tolerance)的基础上,经过改进后使其能够适用于区块链系统。拜占庭容错技术被广泛应用在分布式系统中,比如分布式文件系统、分布式协作系统、云计算等。DBFT主要做了以下改进:1) 将C/S架构的请求响应模式,改进为适合P2P网络的对等节点模式;2) 将静态的共识参与节点改进为可动态进入、退出的动态共识参与节点;3) 为共识参与节点的产生设计了一套基于持有权益比例的投票机制,通过投票决定共识参与节点(记账节点);4) 在区块链中引入数字证书,解决了投票中对记账节点真实身份的认证问题。

[0003] 由于DBFT机制是由权益来选出记账人,然后记账人之间通过拜占庭容错算法来达成共识,这种方式的优点是:1) 专业化的记账人;2) 可以容忍任何类型的错误;3) 记账由多人协同完成,每一个区块都有最终性,不会分叉;4) 算法的可靠性有严格的数学证明。

[0004] 但是DBFT机制的缺点在于,新旧版本之间兼容性差,导致升级非常困难,影响用户使用体验。

### 发明内容

[0005] 本发明要解决的技术问题在于,针对现有技术的上述缺陷,提供一种可以同时运行两个版本的共识,其中新版本向后兼容旧版本,从而平滑向后兼容升级的方法、计算机可读存储介质和DBFT分布式网络。

[0006] 本发明解决其技术问题所采用的技术方案是:构造一种DBFT分布式网络中平滑向后兼容升级的方法,包括:

[0007] 升级步骤,升级DBFT节点版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识;

[0008] 判定步骤,当设定分叉高度区间内的某个分叉值新出的新版本区块数量超过设定分叉阈值时判定新版本运行成功并抛弃旧版本,否则判定新版本运行失败并返回旧版本。

[0009] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中,所述升级步骤进一步包括:

[0010] S1、判定DBFT节点是否拒绝任何新版本,如果是直接判定新版本运行失败;否则执行步骤S2;

[0011] S2、判定DBFT节点是否配置有节点分叉阈值,如果是则执行步骤S3,否则执行步骤

S4;

[0012] S3、判定所述DBFT节点接收到的区块的阈值是否大于或等于所述节点分叉阈值，如果是执行步骤S4，否则直接判定新版本运行失败；

[0013] S4、升级DBFT节点的版本，升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识。

[0014] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中，在所述步骤S2中，所述节点分叉阈值包括最小分叉执行高度区间和/或最小分叉激活阈值，所述区块的阈值包括区块分叉执行高度区间和/或区块分叉激活阈值。

[0015] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中，所述步骤S3进一步包括：

[0016] S31、判定所述DBFT节点接收到的所述区块分叉执行高度区间是否大于所述最小分叉执行高度区间，如果是则执行步骤S4，否则直接判定新版本运行失败。

[0017] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中，所述步骤S3进一步包括：

[0018] S32、判定所述DBFT节点接收到的所述区块分叉激活阈值是否大于所述最小分叉激活阈值，如果是则执行步骤S4，否则直接判定新版本运行失败。

[0019] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中，所述判定步骤进一步包括：

[0020] Sa、当位于分叉执行开始高度和分叉执行结束高度之间的分叉高度区间内的某个分叉值新出的新版本区块的个数大于或等于设定分叉区块阈值时判定新版本运行成功并抛弃旧版本。

[0021] 在本发明所述的DBFT分布式网络中平滑向后兼容升级的方法中，所述判定步骤进一步包括：

[0022] Sb、当当前区块高度高于分叉执行结束高度，但新出的新版本区块的个数小于设定分叉区块阈值时，判定新版本运行失败，且切换回旧版本。

[0023] 本发明解决其技术问题采用的另一技术方案是，构造一种计算机可读存储介质，其上存储有计算机程序，所述程序被处理器执行时实现所述的DBFT分布式网络中平滑向后兼容升级的方法。

[0024] 本发明解决其技术问题采用的再一技术方案是，构造一种DBFT分布式网络，包括多个DBFT节点，所述DBFT节点上存储有计算机程序，其特征在于，所述程序被处理器执行时实现根据所述的DBFT分布式网络中平滑向后兼容升级的方法。

[0025] 实施本发明的DBFT分布式网络中平滑向后兼容升级的方法、计算机可读存储介质和DBFT分布式网络，可以同时运行两个版本的共识，其中新版本向后兼容旧版本，从而平滑向后兼容升级。

## 附图说明

[0026] 下面将结合附图及实施例对本发明作进一步说明，附图中：

[0027] 图1是本发明的DBFT分布式网络中平滑向后兼容升级的方法的第一优选实施例的流程示意图；

[0028] 图2是本发明的DBFT分布式网络中平滑向后兼容升级的方法的升级步骤的第二优选实施例的流程示意图；

[0029] 图3是本发明的DBFT分布式网络的状态机运行示意图。

### 具体实施方式

[0030] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0031] 在本发明中，通过改变DBFT区块的少部分保留字段内容，写入软分叉版本的一些信息，然后让共识节点在共识产生过程当中逐渐接受新版本，最后达到绝大多数共识节点承认某个分叉版本，并废弃旧版本的过程。实施本发明，可以同时运行两个版本的共识，其中新版本向后兼容旧版本，从而平滑向后兼容升级。

[0032] 图1是本发明的DBFT分布式网络中平滑向后兼容升级的方法的第一优选实施例的流程示意图。如图1所示，本发明的DBFT分布式网络中平滑向后兼容升级的方法包括升级步骤和判定步骤。在升级步骤中，升级DBFT节点的版本，升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识。在本发明的优选实施例中，升级范围包括DBFT的产生区块的数据类型，交易的数据类型，以及DBFT的共识规则。

[0033] 在判定步骤中，当设定分叉高度区间内的某个分叉值新出的新版本区块数量超过设定分叉阈值时判定新版本运行成功并抛弃旧版本，否则判定新版本运行失败并返回旧版本。在本发明中，依旧将DBFT网络的运行复杂度限制在多项式级别，不会变成指数级别。通过DBFT的区块当中的保留字段添加少量的变量，让DBFT网络当中的共识节点可以同时运行两个版本的共识，其中新版本向后兼容就版本。当共识节点的收到的新版本区块到达某个阈值的时候，则彻底不兼容就版本。到达阈值的算法依旧严格遵守DBFT的一致性和正确性原则。

[0034] 实施本发明的DBFT分布式网络中平滑向后兼容升级的方法，可以同时运行两个版本的共识，其中新版本向后兼容旧版本，从而平滑向后兼容升级。

[0035] 图2是本发明的DBFT分布式网络中平滑向后兼容升级的方法的升级步骤的第二优选实施例的流程示意图。在步骤S1中，判定DBFT节点是否拒绝任何新版本，如果是则执行步骤S4直接判定新版本运行失败。否则执行步骤S2。在步骤S2中，判定DBFT节点是否配置有节点分叉阈值，如果是则执行步骤S3，否则执行步骤S5。在本发明的优选实施例中，所述节点分叉阈值包括最小分叉执行高度区间和/或最小分叉激活阈值。在步骤S3中，判定所述DBFT节点接收到的区块的阈值是否大于或等于所述节点分叉阈值，如果是执行步骤S5，否则则执行步骤S4直接判定新版本运行失败。对应地，所述区块的阈值包括区块分叉执行高度区间和/或区块分叉激活阈值。在步骤S5中，升级DBFT节点的版本，升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识。

[0036] 在本发明的进一步的优选实施例中，所述步骤S3进一步包括：判定所述DBFT节点接收到的所述区块分叉执行高度区间是否大于所述最小分叉执行高度区间，如果是则执行步骤S5，升级DBFT节点的版本，升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识，否则则执行步骤S4直接判定新版本运行失败。在本发明的进一步的优选实

施例中,所述步骤S3进一步包括:判定所述DBFT节点接收到的所述区块分叉激活阈值是否大于所述最小分叉激活阈值,如果是则执行步骤S5,升级DBFT节点的版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识,否则则执行步骤S4直接判定新版本运行失败。

[0037] 在本发明的优选实施例中,所述节点分叉阈值可以包括最小分叉执行高度区间和最小分叉激活阈值中的至少一者,或者两者,可以只执行最小分叉执行高度区间判定或最小分叉激活阈值判定,也可以执行两种判定,并且可以以任何顺序执行,或者同时执行。

[0038] 在本发明的进一步的优选实施例中,所述判定步骤进一步包括:当位于分叉执行开始高度和分叉执行结束高度之间的分叉高度区间内的某个分叉值新出的新版本区块的个数大于或等于设定分叉区块阈值时判定新版本运行成功并抛弃旧版本,且当当前区块高度高于分叉执行结束高度,但新出的新版本区块的个数小于设定分叉区块阈值时,判定新版本运行失败,且切换回旧版本。

[0039] 下面,本发明以具体实施方式进行说明,在本实施例中,本发明通过改变DBFT区块的少部分保留字段内容,写入软分叉版本的一些信息,然后让共识节点在共识产生过程当中逐渐接受新版本,最后达到绝大多数共识节点承认某个分叉版本,并废弃旧版本的过程。过程如下:

[0040] 首先,定义新版本的数据结构,该新版本的数据结构不同于现行的旧版本的数据结构,与此同时,在区块的保留字段当中设置新版本号,即分叉版本号,并进一步定义如下参数:

[0041] 分叉执行开始高度  $h_{start}^{execute}$

[0042] 分叉执行结束高度  $h_{end}^{execute}$

[0043] 分叉区块阈值  $n_{threshold}^{fork}$

[0044] 分叉值新出的新版本区块的个数,即检查分叉信号值,其中,值  $n_{signal}^{fork}$ ,其中

$$n_{signal}^{fork} \geq 2 * n_{threshold}^{fork}, n_{signal}^{fork} \leq \frac{1}{5} * (h_{end}^{execute} - h_{start}^{execute})。$$

[0045] 在本优选实施例中,升级DBFT节点的版本,升级后的DBFT节点兼容新版本和旧版本并提交新版本区块参与DBFT共识。新版本区块可能会被其他DBFT节点拒绝,但是随着DBFT的网络运行,越来越多的节点开始升级新版本,并且在每次DBFT当中都会通过新版本的区块共识,虽然有可能依旧无法最终通过。但是旧版本节点打包的区块依旧会被新版本认同。当超过一定数量,例如2/3的DBFT节点运行新版本时,网络当中会同时存在两种版本的区块,此时剩余的旧版本的节点依旧可以出块。

[0046] 在本实施例中,当在分叉执行开始高度  $h_{start}^{execute}$  和分叉执行结束高度  $h_{end}^{execute}$  之间的分叉高度区间  $[h_{start}^{execute}, h_{end}^{execute}]$  的某一个检查分叉信号值(第  $i$  个  $n_{signal}^{fork}$ ) 出的新版本区块超过分叉区块阈值  $n_{threshold}^{fork}$  时,则认为新版本运行成功,并抛弃旧版本。如果当前区块高度超过分叉执行开始高度  $h_{start}^{execute}$ ,但是最新的  $n_{signal}^{fork}$  出的新版本区块没有达

到分叉区块阈值 $n_{threshold}^{fork}$ 时,则认为软分叉失败,即新版本运行失败,且切换回旧版本。

[0047] 图3是本发明的DBFT分布式网络的状态机运行示意图。下面结合图3对本发明的进一步的优选实施例说明如下。

[0048] 如图3所示,对应DBFT节点,在其收到新版本 $v_{fork}$ ,为该版本创建一个新的上下文,并将上下文状态置为D(Defined)状态。如果DBFT节点愿意执行该分叉,则在当前区块高度达到分叉执行开始高度 $h_{start}^{execute}$ 之后将分叉上下文状态置为S(Start)状态,并开始观察分叉执行的结果。

[0049] 如果分叉版本(即新版本)被执行,此时将分叉上下文状态置为A(Active)状态,并且在分叉执行结束高度 $h_{end}^{execute}$ 之前的最后一个分叉检查分叉信号值 $n_{signal}^{fork}$ 出新版本区块超过分叉区块阈值 $n_{threshold}^{fork}$ 时则认为新版本 $v_{fork}$ 分叉成功,并且此时DBFT节点依旧可以接受新版本 $v_{fork}$ 之前的版本。

[0050] 如果在分叉执行结束高度 $h_{end}^{execute}$ 之后的任意一个分叉执行检查区间,新版本区块超过分叉区块阈值 $n_{threshold}^{fork}$ 时则认为新版本运行成功,则DBFT节点拒绝之前的旧版本,此时将分叉上下文状态置为FI(Final)状态。

[0051] 在本发明的一个优选实施例中,如果DBFT节点配置了拒绝任何分叉版本,则直接进入FA(Failed)状态,即DBFT节点拒绝任何更新,新版本更新失败。

[0052] 在本发明的一个优选实施例中,DBFT节点可以自行配置最小分叉执行高度区间 $range_{min}^{fork}$ ,那么判定所述DBFT节点接收到的所述区块分叉执行高度区间是否大于所述最小分叉执行高度区间 $range_{min}^{fork}$ ,如果是,则DBFT节点愿意执行该分叉,则在当前区块高度达到分叉执行开始高度 $h_{start}^{execute}$ 之后将分叉上下文状态置为S(Start)状态,并开始观察分叉执行的结果,如果不是,则拒绝执行该分叉版本并进入FA(Failed)状态。

[0053] 在本发明的一个优选实施例中,DBFT节点可以自行配置最小分叉激活阈值 $n_{min}^{thresold}$ ,那么判定所述DBFT节点接收到的所述区块分叉激活阈值大于所述最小分叉激活阈值 $n_{min}^{thresold}$ ,如果是,则DBFT节点愿意执行该分叉,则在当前区块高度达到分叉执行开始高度 $h_{start}^{execute}$ 之后将分叉上下文状态置为S(Start)状态,并开始观察分叉执行的结果,如果不是,则拒绝执行该分叉版本并进入FA(Failed)状态。优选地,最小分叉激活阈值 $n_{min}^{thresold}$ 也可以是一个阈值期间。

[0054] 本领域技术人员知悉,可以根据DBFT网络的节点数量,网络环境要求,安全要求等各种实际条件设置上述分叉执行开始高度 $h_{start}^{execute}$ ,分叉执行结束高度 $h_{end}^{execute}$ ,分叉区块阈值 $n_{threshold}^{fork}$ ,最小分叉执行高度区间 $range_{min}^{fork}$ ,最小分叉激活阈值 $n_{min}^{thresold}$ ,其具体设置方法为本领域技术人员已知,在此就不再累述了。

[0055] 实施本发明的DBFT分布式网络中平滑向后兼容升级的方法,可以同时运行两个版本的共识,其中新版本向后兼容旧版本,从而平滑向后兼容升级。

[0056] 本发明还涉及一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现所述的DBFT分布式网络中平滑向后兼容升级的方法。

[0057] 本发明还涉及一种DBFT分布式网络,包括多个DBFT节点,所述DBFT节点上存储有计算机程序,所述程序被处理器执行时实现根据所述的DBFT分布式网络中平滑向后兼容升级的方法。

[0058] 实施本发明的DBFT分布式网络中平滑向后兼容升级的计算机可读存储介质和DBFT分布式网络,可以同时运行两个版本的共识,其中新版本向后兼容旧版本,从而平滑向后兼容升级。

[0059] 因此,本发明可以通过硬件、软件或者软、硬件结合来实现。本发明可以在至少一个计算机系统中以集中方式实现,或者由分布在几个互连的计算机系统中的不同部分以分散方式实现。任何可以实现本发明方法的计算机系统或其它设备都是可适用的。常用软件件的结合可以是安装有计算机程序的通用计算机系统,通过安装和执行程序控制计算机系统,使其按本发明方法运行。

[0060] 本发明还可以通过计算机程序产品进行实施,程序包含能够实现本发明方法的全部特征,当其安装到计算机系统中时,可以实现本发明的方法。本文件中的计算机程序所指的是:可以采用任何程序语言、代码或符号编写的一组指令的任何表达式,该指令组使系统具有信息处理能力,以直接实现特定功能,或在进行下述一个或两个步骤之后实现特定功能:a)转换成其它语言、编码或符号;b)以不同的格式再现。

[0061] 虽然本发明是通过具体实施例进行说明的,本领域技术人员应当明白,在不脱离本发明范围的情况下,还可以对本发明进行各种变换及等同替代。另外,针对特定情形或材料,可以对本发明做各种修改,而不脱离本发明的范围。因此,本发明不局限于所公开的具体实施例,而应当包括落入本发明权利要求范围内的全部实施方式。

[0062] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

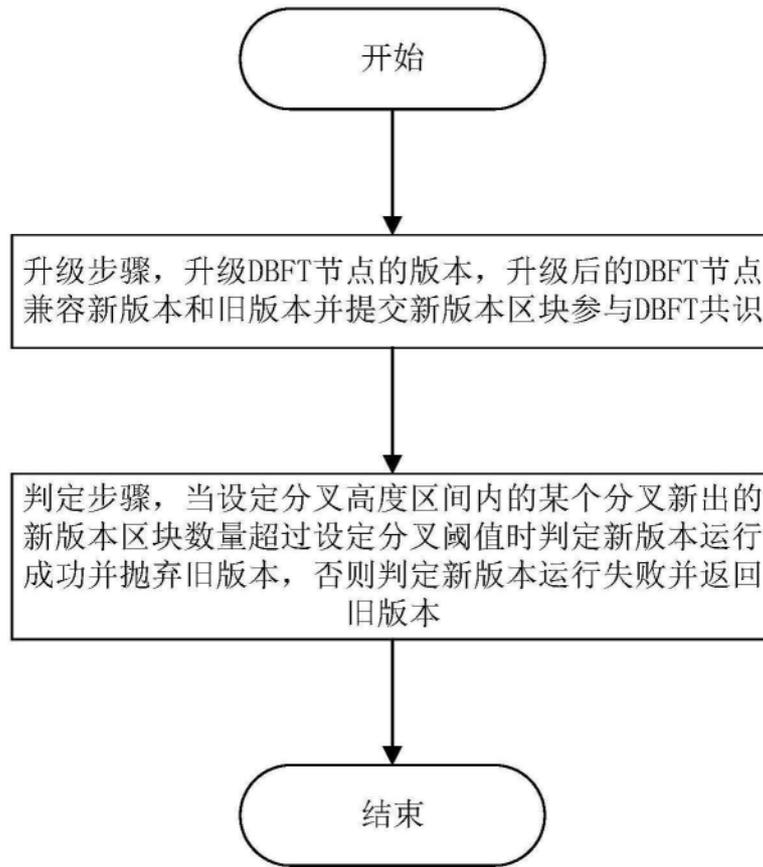


图1

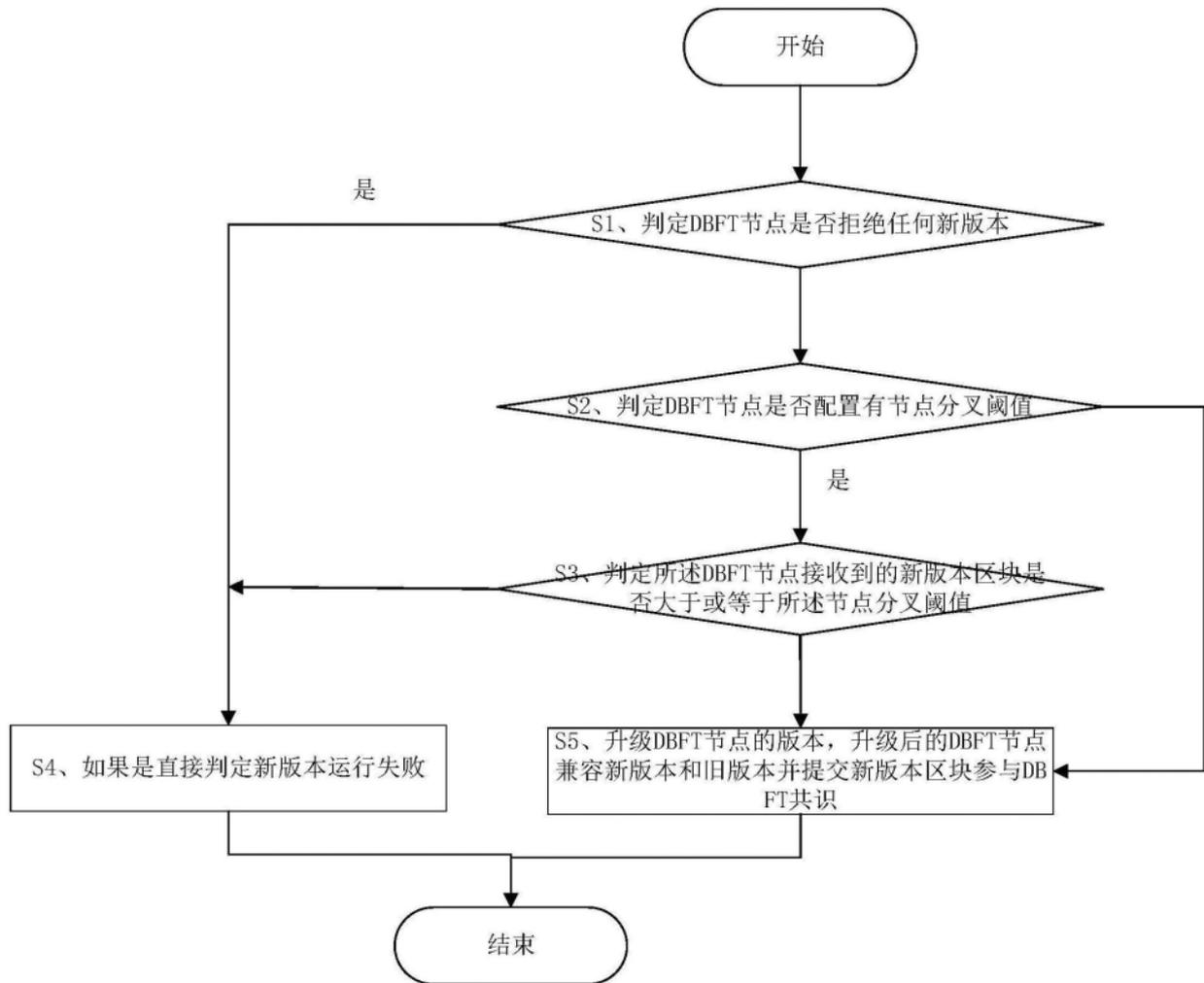


图2

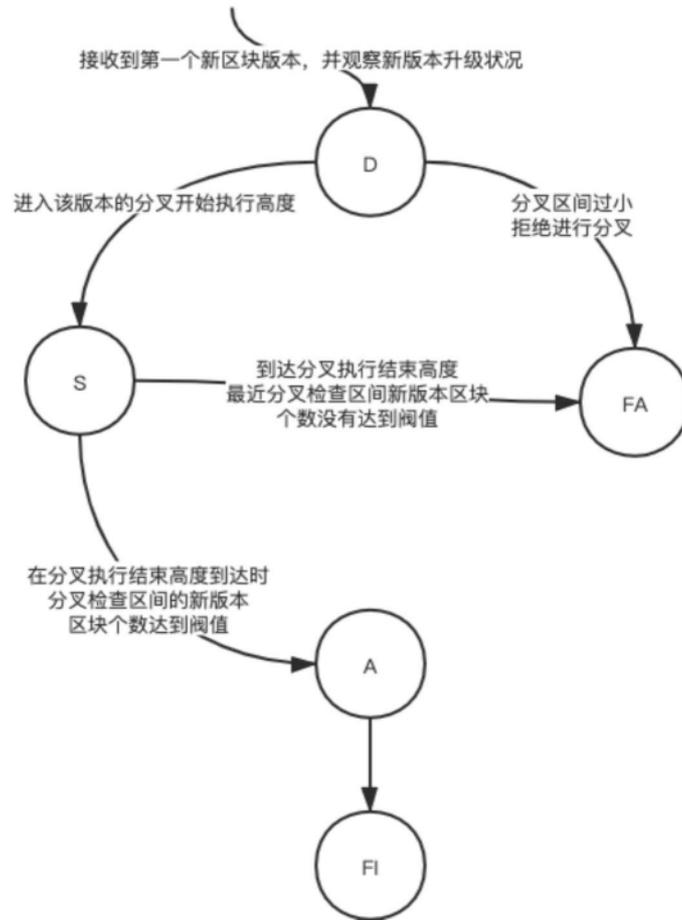


图3