(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108351882 A
(43)申请公布日 2018.07.31

(21)申请号 201680061456.6

(22)申请日 2016.08.26

(30)优先权数据
62/211,411 2015.08.28 US
14/988,873 2016.01.06 US
15/153,011 2016.05.12 US
62/344,682 2016.06.02 US
15/205,688 2016.07.08 US

(85)PCT国际申请进入国家阶段日
2018.04.20

(86)PCT国际申请的申请数据
PCT/US2016/049067 2016.08.26

(87)PCT国际申请的公布数据
WO2017/040313 EN 2017.03.09

(71)申请人 斯沃尔德斯股份有限公司
地址 美国得克萨斯

(72)发明人 L·C·贝尔德三世

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 李晓芳

(51)Int.Cl.
G06F 17/30(2006.01)
G06F 9/54(2006.01)
G06F 3/06(2006.01)
G06F 7/00(2006.01)
G06F 12/08(2016.01)

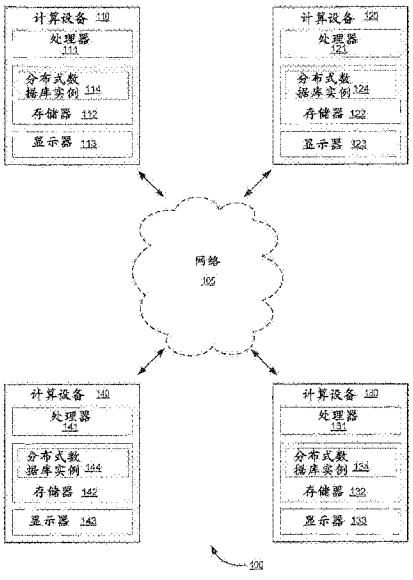
权利要求书10页 说明书41页 附图19页

(54)发明名称

用于网络内的分布式数据库的方法和装置

(57)摘要

在一些实施例中，一种装置包括在第一计算设备处的分布式数据库的实例，该第一计算设备被配置为包括在实现分布式数据库的计算设备集合内。该装置还包括被配置为定义链接到第一事件集合的第一事件的处理器。该处理器被配置为从计算设备集合中的第二计算设备接收表示(1)由第二计算设备定义的并且(2)链接到第二事件集合的第二事件的信号。该处理器被配置为至少基于协议的结果来识别与第三事件集合相关联的顺序。该处理器被配置为在分布式数据库的实例中存储与第三事件集合相关联的顺序。



CN 108351882 A

1.一种装置,包括:

在第一计算设备处的分布式数据库的实例,第一计算设备被配置为包括在多个计算设备内,所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库,第一计算设备被配置为在所述分布式数据库的所述实例中存储多个事务的指示;以及

可操作地耦合到所述分布式数据库的所述实例的第一计算设备的处理器,

所述处理器被配置为在第一时间定义链接到第一多个事件的第一事件,所述第一多个事件中的每个事件是字节序列并且与(1)多个事务集合中的事务集合以及(2)所述事务集合所关联的顺序相关联,所述事务集合中的每个事务来自所述多个事务,

所述处理器被配置为在第一时间之后的第二时间并且从所述多个计算设备中的第二计算设备接收(1)由第二计算设备定义的并且(2)链接到第二多个事件的第二事件,

所述处理器被配置为定义链接到第一事件和第二事件的第三事件,

所述处理器被配置为至少基于第一多个事件和第二多个事件来识别与第三多个事件相关联的顺序,第三多个事件中的每个事件来自第一多个事件或第二多个事件中的至少一个,

所述处理器被配置为至少基于(1)与第三多个事件相关联的顺序以及(2)与所述多个事务集合中的每个事务集合相关联的顺序来识别与所述多个事务相关联的顺序,

所述处理器被配置为在所述分布式数据库的所述实例中存储与所述多个事务相关联的顺序。

2.如权利要求1所述的装置,其中所述处理器被配置为至少基于(1)所述多个事务以及(2)与所述多个事务相关联的顺序来定义数据库状态变量。

3.如权利要求1所述的装置,其中所述多个事务中的每个事务与加密货币的转移、对银行账户余额的改变的指示、转移物品的所有权的顺序、或对多人游戏的状态的修改中的至少一个相关联。

4.如权利要求1所述的装置,其中与所述多个事务相关联的顺序与多个事务顺序值相关联,所述多个事务顺序值中的每个事务顺序值与来自所述多个事务集合中的至少一个事务集合的事务相关联。

5.如权利要求1所述的装置,其中所述处理器被配置为至少部分地基于在计算事件集合的加权计数中被用作权重的权益值来识别与第三多个事件相关联的顺序,所述处理器被配置为基于权益值集合的总和来计算加权计数,所述权益值集合中的每个权益值与定义所述事件集合中的事件的所述分布式数据库的实例相关联。

6.一种装置,包括:

在第一计算设备处的分布式数据库的实例,第一计算设备被配置为包括在多个计算设备内,所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库;以及

在第一计算设备的存储器或处理器中实现的数据库收敛模块,所述数据库收敛模块可操作地耦合到所述分布式数据库的所述实例,

所述数据库收敛模块被配置为在第一时间定义链接到第一多个事件的第一事件,所述第一多个事件中的每个事件是字节序列,

所述数据库收敛模块被配置为在第一时间之后的第二时间并且从所述多个计算设备中的第二计算设备接收(1)由第二计算设备定义的并且(2)链接到第二多个事件的第二事件,第二多个事件中的每个事件是字节序列,

所述数据库收敛模块被配置为定义链接到第一事件和第二事件的第三事件,

所述数据库收敛模块被配置为至少基于第一多个事件和第二多个事件来识别与第三多个事件相关联的顺序,第三多个事件中的每个事件来自第一多个事件或第二多个事件中的至少一个,

所述数据库收敛模块被配置为在所述分布式数据库的所述实例中存储与第三多个事件相关联的顺序。

7.如权利要求6所述的装置,其中:

第一多个事件中的每个事件与(1)第一多个事务集合中的事务集合以及(2)第一多个事务集合中的该事务集合所关联的顺序相关联,

第二多个事件中的每个事件与(1)第二多个事务集合中的事务集合以及(2)第二多个事务集合中的该事务集合所关联的顺序相关联,

所述数据库收敛模块被配置为至少基于第一多个事件和第二多个事件来识别与多个事务相关联的顺序,所述多个事务中的每个事务来自第一多个事务集合中的至少一个事务集合或第二多个事务集合中的至少一个事务集合,以及

所述数据库收敛模块被配置为在所述分布式数据库的所述实例中存储与所述多个事务相关联的顺序。

8.如权利要求6所述的装置,其中:

第一多个事件包括由所述数据库收敛模块先前定义的事件和从所述多个计算设备中的第三计算设备接收到的事件,以及

第一事件包括与由所述数据库收敛模块先前定义的事件相关联的标识符和与从所述多个计算设备中的第三计算设备接收到的事件相关联的标识符。

9.如权利要求6所述的装置,其中:

第二多个事件包括由第二计算设备先前定义的事件和由第二计算设备从所述多个计算设备中的第三计算设备接收到的事件,以及

第二事件包括与由第二计算设备先前定义的事件相关联的标识符和与由第二计算设备从第三计算设备接收到的事件相关联的标识符。

10.如权利要求6所述的装置,其中所述分布式数据库不包括领导者实体。

11.如权利要求6所述的装置,其中:

第一多个事件包括由所述数据库收敛模块先前定义的事件和从所述多个计算设备中的第三计算设备接收到的事件,以及

第一事件包括与由所述数据库收敛模块先前定义的事件相关联的散列值和与从所述多个计算设备中的第三计算设备接收到的事件相关联的散列值。

12.如权利要求6所述的装置,其中:

第一事件包括所述多个计算设备中的已被第一计算设备识别为(1)与无效事件相关联或(2)与第一时间之前的无效事务相关联的计算设备集合的指示,以及

第二事件包括所述多个计算设备中的已被第二计算设备识别为(1)与无效事件相关联

或(2)与第二时间之前的无效事务相关联的计算设备集合的指示。

13.如权利要求6所述的装置,其中与所述多个事件相关联的顺序至少部分地基于与所述多个计算设备中的每个计算设备相关联的权重。

14.如权利要求6所述的装置,其中所述数据库收敛模块被配置为在接收到第二多个事件中的每个事件之后接收第二事件。

15.如权利要求6所述的装置,其中所述数据库收敛模块被配置为从第二计算设备接收除了第二多个事件中的由第一计算设备定义的事件之外的第二多个事件中的每个事件。

16.如权利要求6所述的装置,其中第三事件包括与第一计算设备相关联的数字签名。

17.如权利要求6所述的装置,其中第三事件包括时间和日期,所述时间和所述日期与第三事件的定义相关联。

18.如权利要求6所述的装置,其中所述数据库收敛模块被配置为基于至少(1)第三多个事件以及(2)与第三多个事件相关联的顺序来定义数据库状态变量。

19.一种方法,包括:

在多个计算设备中的第一计算设备处接收与第一事务相关联的数据,所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现分布式数据库,所述多个计算设备中的每个计算设备具有所述分布式数据库的单独实例;

在第一时间定义与第一事务相关联的第一事务顺序值;

从所述多个计算设备中的第二计算设备接收与第二事务相关联的数据;

在第一计算设备处的所述分布式数据库的实例中存储多个事务的指示,所述多个事务至少包括第一事务和第二事务;

在第一时间之后的第二时间选择至少包括第一事务顺序值和第二事务顺序值的多个事务顺序值,第二事务顺序值与第二事务相关联;以及

至少基于所述多个事务和所述多个事务顺序值来定义数据库状态变量。

20.如权利要求19所述的方法,还包括:

在第一时间之后的第三时间定义事件,所述事件包括(1)所述数据库状态变量的散列,所述数据库状态变量的散列与第三时间之前的第四时间相关联,以及(2)在第四时间影响了所述数据库状态变量的事务集合,所述事务集合中的每个事务来自所述多个事务。

21.如权利要求19所述的方法,还包括:

在第一时间之后的第三时间定义事件,所述事件包括(1)与在第三时间之前的第四时间相关联的所述数据库状态变量的散列,(2)在第四时间影响了所述数据库状态变量的事务集合,以及(3)第四时间的状态变量的散列的阈值签名的份额,所述事务集合中的每个事务来自所述多个事务。

22.如权利要求19所述的方法,其中定义所述数据库状态变量响应于选择所述多个事务顺序值。

23.如权利要求18所述的方法,其中所述数据库状态变量被维护在快速克隆数组列表、快速克隆散列表、快速克隆关系数据库或快速克隆文件系统中的至少一个中。

24.一种装置,包括:

存储器,所述存储器包括第一计算设备处的分布式数据库的实例,第一计算设备被配置为包括在多个计算设备内,所述多个计算设备经由可操作地耦合到所述多个计算设备的

网络来实现所述分布式数据库;以及

可操作地耦合到所述分布式数据库的所述实例的处理器,

所述处理器被配置为在第一时间定义链接到第一多个事件的第一事件,第一多个事件中的每个事件是字节序列,

所述处理器被配置为在第一时间之后的第二时间并且从所述多个计算设备中的第二计算设备接收表示(1)由第二计算设备定义的并且(2)链接到第二多个事件的第二事件的信号,第二多个事件中的每个事件是字节序列,

所述处理器被配置为至少基于协议的结果来识别与第三多个事件相关联的顺序,第三多个事件中的每个事件来自第一多个事件或第二多个事件中的至少一个,

所述处理器被配置为在所述分布式数据库的所述实例中存储与第三多个事件相关联的顺序。

25.如权利要求24所述的装置,其中所述处理器被配置为至少部分地基于与所述多个计算设备中的每个计算设备相关联的权益值来识别与第三多个事件相关联的顺序。

26.如权利要求24所述的装置,其中所述处理器被配置为基于与第三多个事件相关联的顺序来更改与所述分布式数据库相关联的状态。

27.如权利要求24所述的装置,其中第三多个事件中的每个事件与属性集合相关联,所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值。

28.如权利要求24所述的装置,其中第三多个事件中的每个事件与属性集合相关联,所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值,

所述属性集合中的第一属性的值包括数值,以及

所述属性集合中的第二属性的值包括与所述数值相关联的二进制值。

29.如权利要求24所述的装置,其中第三多个事件中的每个事件与属性集合相关联,所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值,

所述属性集合中的第一属性的值包括数值,以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值,

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的事件集合之间的关系是否满足标准。

30.如权利要求24所述的装置,其中第三多个事件中的每个事件与属性集合相关联,所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值,

所述属性集合中的第一属性的值包括数值,以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值,

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的事件集合之间的关系是否满足标准,

所述事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先,并且(2)与作为所述事件集合中的剩余事件的公共属性相关联。

31.如权利要求24所述的装置,其中第三多个事件中的每个事件与属性集合相关联,所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值,

所述属性集合中的第一属性的值包括数值,以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值,

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的事件集合之间的关系是否满足标准，

所述事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为所述事件集合中的剩余事件的公共属性相关联，所述公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与特定值相关联。

32.如权利要求24所述的装置，其中第三多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括数值，以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值，

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的事件集合之间的关系是否满足标准，

所述事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为所述事件集合中的剩余事件的公共属性相关联，所述公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与特定值相关联，

所述事件集合是否满足标准是基于所述事件集合中的事件的数量与基于所述多个计算设备中的计算设备的数量的阈值的比较。

33.如权利要求24所述的装置，其中第三多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括数值，以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值，

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的事件集合之间的关系是否满足标准，

所述事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为所述事件集合中的剩余事件的公共属性相关联，所述公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与特定值相关联，

所述事件集合是否满足标准是基于与所述事件集合中的每个事件相关联的加权值的组合与基于与所述多个计算设备中的每个计算设备相关联的加权值的组合定义的阈值的比较。

34.如权利要求24所述的装置，其中第三多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括第一数值，以及所述属性集合中的第二属性的值包括与第一数值相关联的二进制值，

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为所述第一事件集合中的剩余事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联。

35.如权利要求24所述的装置，其中第三多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括第一数值，以及所述属性集合中的第二属性的值包括与第一数值相关联的二进制值，

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为所述第一事件集合中的剩余事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与第一特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联，

第二公共属性(1)与第三公共属性相关联，其中第三公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联，并且(2)与基于指示集合的结果相关联，所述指示集合中的每个指示与第三事件集合中的事件相关联，第三事件集合中的每个事件与第四公共属性相关联，第四公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第三事件与不同于第一特定值和第二特定值的第三特定值相关联。

36.如权利要求24所述的装置，其中第三多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括第一数值，以及所述属性集合中的第二属性的值包括与第一数值相关联的二进制值，

第三多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是第三多个事件中的所述事件的祖先，并且(2)与作为第一事件集合中的剩余事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的事件与第一特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联，

第二公共属性(1)与第三公共属性相关联，其中第三公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联，并且(2)与基于指示集合的结果相关联，所述指示集合中的每个指示与第三事件集合中的事件相关联，第三事件集合中的每个事件与第四公共属性相关联，第四公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第三事件与不同于第一特定值和第二

特定值的第三特定值相关联，

第一特定值是第一整数，

第二特定值是大于第一整数的第二整数，

第三特定值是大于第二整数的第三整数。

37.一种装置，包括：

存储器，包括第一计算设备处的分布式数据库的实例，第一计算设备被配置为包括在多个计算设备内，所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库；以及

可操作地耦合到所述分布式数据库的所述实例的处理器，

所述处理器被配置为接收表示链接到多个事件的事件的信号，

所述处理器被配置为至少基于协议的结果来识别与所述多个事件相关联的顺序，

所述处理器被配置为在所述分布式数据库的所述实例中存储与所述多个事件相关联的顺序。

38.如权利要求37所述的装置，其中所述处理器被配置为至少部分地基于与所述分布式数据库的所述实例相关联的权益值来识别与所述多个事件相关联的顺序。

39.如权利要求37所述的装置，其中所述多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括数值，以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值，

所述多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是所述多个事件中的所述事件的祖先，并且(2)与作为第一事件集合中的其它事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第一事件与第一特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联，

第二公共属性(1)与第三公共属性相关联，其中第三公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联，并且(2)与基于指示集合的结果相关联，所述指示集合中的每个指示与第二事件集合中的事件和第三事件集合中的事件相关联，第三事件集合中的每个事件与第四公共属性相关联，第四公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第三事件与不同于第一特定值和第二特定值的第三特定值相关联，

所述指示集合中的每个指示是基于第三事件集合中的事件是否是第二事件集合中的事件的后代的二进制值。

40.如权利要求37所述的装置，其中所述多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括数值，以及所述属性集合中的第二属性的值包括

与所述数值相关联的二进制值，

所述多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是所述多个事件中的所述事件的祖先，并且(2)与作为所述事件集合中的其它事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第一事件与第一特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联，

第二公共属性(1)与第三公共属性相关联，其中第三公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联，并且(2)与基于指示集合的结果相关联，所述指示集合中的每个指示与第二事件集合中的事件和第三事件集合中的事件相关联，第三事件集合中的每个事件与第四公共属性相关联，第四公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第三事件与不同于第一特定值和第二特定值的第三特定值相关联，所述结果基于第三事件集合与不同于第一特定值、第二特定值和第三特定值的第四特定值所关联的第四事件之间的关系。

41.如权利要求37所述的装置，其中所述多个事件中的每个事件与属性集合相关联，所述协议的结果包括第三多个事件中的每个事件的属性集合中的每个属性的值，

所述属性集合中的第一属性的值包括数值，以及所述属性集合中的第二属性的值包括与所述数值相关联的二进制值，

所述多个事件中的事件的第二属性的二进制值基于该事件与链接到该事件的第一事件集合之间的关系是否满足标准，

第一事件集合中的每个事件(1)是所述多个事件中的所述事件的祖先，并且(2)与作为第一事件集合中的其它事件的第一公共属性相关联，第一公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第一事件与第一特定值相关联，

所述属性集合中的第三属性的值包括基于所述事件与链接到所述事件的第二事件集合之间的关系的第二数值，

第二事件集合中的每个事件是所述事件的后代并且与作为第二事件集合中的剩余事件的第二公共属性相关联，

第二公共属性(1)与第三公共属性相关联，其中第三公共属性指示第一实例由所述多个计算设备中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联，并且(2)与基于指示集合的结果相关联，所述指示集合中的每个指示与第二事件集合中的事件和第三事件集合中的事件相关联，第三事件集合中的每个事件与第四相同属性相关联，第四相同属性指示第一实例由所述多个计算设备中的每个计算设备定义的第三事件与不同于第一特定值和第二特定值的第三特定值相关联，所述结果基于第三事件集合和第四事件之间的关系，第四事件与被配置为基于所述结果而改变的二进制值相关联。

42.一种存储表示要由处理器执行的指令的代码的非暂态处理器可读介质，所述代码包括使所述处理器执行以下操作的代码：

接收表示链接到多个事件的事件的信号；

基于与所述多个事件中的每个事件相关联的轮次以及对于何时递增与每个事件相关联的轮次的指示来识别与所述多个事件相关联的顺序；以及

在第一计算设备处的分布式数据库的实例中，存储与所述多个事件相关联的顺序，第一计算设备被配置为包括在多个计算设备内，所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库，所述分布式数据库的所述实例可操作地耦合到所述处理器。

43.如权利要求42所述的非暂态处理器可读介质，其中用于识别的代码包括用于基于与所述分布式数据库的多个实例相关联的权益值来识别与所述多个事件相关联的顺序的代码。

44.如权利要求42所述的非暂态处理器可读介质，其中用于识别的代码包括通过以下操作来识别与所述多个事件相关联的顺序的代码：

将所述多个事件中的每个事件与多个事件集合中的事件集合相关联，所述多个事件集合中的每个事件集合与公共轮次相关联；

对于所述多个事件集合中的每个事件集合，识别该事件集合中的事件的子集，所述事件的子集中的每个事件是第一实例，由所述多个计算设备中的每个计算设备定义的事件与所述公共轮次相关联；

基于所述事件的子集中的每个事件与所述多个事件中的剩余事件的关系来识别所述事件的子集中的该事件的二进制属性；

对于所述事件的子集中的事件，基于该事件与具有二进制属性的正值的事件集合之间的关系来识别接收到的轮次值；以及

至少基于该事件的接收到的轮次值来识别与所述多个事件相关联的顺序。

45.一种方法，包括：

从多个计算设备中的第一计算设备处的分布式数据库的实例接收第一事件，所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库；

基于第一事件和第二事件来定义第三事件；

至少部分地基于第三事件来确定第一事件集合，第一事件集合中的每个事件：

a)由第二事件集合来识别，与第二事件集合相关联的集体权益值满足第一权益值标准，第二事件集合中的每个事件(1)

由所述分布式数据库的不同实例定义，以及(2)由第三事件来识别，以及

b)与第一轮次号相关联；

基于确定与第一事件集合中的每个事件相关联的权益值的总和满足第二权益值标准来计算第三事件的轮次号，第一事件的轮次号对应于大于第一轮次号的第二轮次号；

基于第三事件确定第三事件集合，第三事件集合中的每个事件：

a)由包括第三事件的第四事件集合来识别，第四事件集合中的每个事件由所述分布式数据库的不同实例来定义，与第四事件集合相关联的集体权益值满足第三权益值标准，以及

b)来自第一事件集合；

基于与第三事件集合相关联的集体权益值满足第四权益值标准来定义第四事件的顺

序值;以及

将所述顺序值存储在所述多个计算设备中的第二计算设备处的所述分布式数据库的实例中。

46.如权利要求45所述的方法,其中权益值集合包括与定义第二事件集合中的事件的所述分布式数据库的每个实例相关联的权益值,与第二事件集合相关联的集体权益值基于所述权益值集合中的权益值的总和。

47.如权利要求45所述的方法,其中权益值集合包括(1)与定义第二事件集合中的事件的所述分布式数据库的每个实例相关联、并且(2)与所述分布式数据库的该实例所关联的加密货币的量成比例的权益值,与第二事件集合相关联的集体权益值基于所述权益值集合中的权益值的总和。

48.如权利要求45所述的方法,其中第二事件来自第二计算设备。

49.如权利要求45所述的方法,所述顺序值被存储为唯一地识别所述分布式数据库的不可撤销状态的散列值的一部分。

50.如权利要求45所述的方法,其中:

第一事件集合中的每个事件由所述分布式数据库的不同实例定义,以及

第一事件集合中的每个事件是由定义该事件的所述分布式数据库的实例定义的事件集合中的具有第一轮次号的最早事件。

51.如权利要求45所述的方法,其中第一权益值标准、第二权益值标准、第三权益值标准或第四权益值标准中的至少一个基于所述分布式数据库的集体权益值来定义。

52.如权利要求45所述的方法,其中在第一时间实现所述分布式数据库的多个计算设备与可信实体集合相关联,在第一时间之后的第二时间实现所述分布式数据库的多个计算设备与包括不是来自所述可信实体集合的实体的实体集合相关联。

53.一种方法,包括:

从多个计算设备中的第一计算设备处的分布式数据库的实例接收第一事件,所述多个计算设备经由可操作地耦合到所述多个计算设备的网络来实现所述分布式数据库;

基于第一事件和第二事件来定义第三事件,第三事件链接到事件集合;

至少部分地基于与事件集合相关联的集体权益值满足权益值标准来定义第四事件的顺序值;以及

将所述顺序值存储在所述多个计算设备中的第二计算设备处的所述分布式数据库的实例中。

54.如权利要求53所述的方法,还包括:

基于权益值集合的总和来计算所述集体权益值,所述权益值集合中的每个权益值与定义所述事件集合中的事件的所述分布式数据库的实例相关联。

用于网络内的分布式数据库的方法和装置

[0001] 对相关申请的交叉引用

[0002] 本申请是于2016年7月8日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国专利申请No.15/205,688的部分继续申请，美国专利申请No.15/205,688是于2016年1月6日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国专利申请No.14/988,873的继续申请，美国专利申请No.14/988,873要求于2015年8月28日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国临时专利申请No.62/211,411的优先权和权益，这些申请中的每一个都通过引用被整体结合于此。

[0003] 本申请还是于2016年5月12日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国专利申请No.15/153,011的部分继续申请，美国专利申请No.15/153,011是于2016年1月6日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国专利申请No.14/988,873的部分继续申请，美国专利申请No.14/988,873要求于2015年8月28日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国临时专利申请No.62/211,411的优先权和权益，这些申请中的每一个都通过引用被整体结合于此。

[0004] 本申请还要求于2015年8月28日提交的标题为“Methods and Apparatus for a Distributed Database within a Network”的美国临时专利申请No.62/211,411的优先权和权益，该申请通过引用被整体结合于此。

[0005] 本申请还要求于2016年6月2日提交的标题为“Methods and Apparatus for a Distributed Database with Consensus Determined Based on Weighted Stakes”的美国临时专利申请No.62/344,682的优先权和权益，该申请通过引用被整体结合于此。

背景技术

[0006] 本文描述的实施例一般而言涉及数据库系统，并且更具体地涉及用于实现跨网络中的多个设备的数据库系统的方法和装置。

[0007] 一些已知的分布式数据库系统试图对分布式数据库系统内的(例如，关于事务发生的顺序的)值达成共识(consensus)。例如，在线多人游戏可能具有用户可以访问以玩游戏的许多计算机服务器。如果两个用户试图同时拾取游戏中的特定物品，那么分布式数据库系统内的服务器就这两个用户中的哪个用户首先拾取该物品最终达成一致是重要的。

[0008] 这种分布式共识可以通过诸如Paxos算法或其变体之类的方法和/或过程来处理。依据这样的方法和/或过程，数据库系统的一个服务器被设置为“领导者(leader)”，并且领导者决定事件的顺序。(例如，多人游戏内的)事件被转发给领导者，领导者选择事件的排序，并且领导者向数据库系统的其它服务器广播该排序。

[0009] 但是，这样的已知方法使用由数据库系统的用户(例如，游戏玩家)信任的一方(例

如,中央管理服务器)操作的服务器。因此,存在对于用于分布式数据库系统的方法和装置的需要,该分布式数据库系统不需要领导者或可信的第三方来操作数据库系统。

[0010] 其它分布式数据库被设计为没有领导者,但是效率低。例如,一种这样的分布式数据库是基于可以达成共识的"区块链"数据结构。但是,这样的系统可以被限制为对于所有参与者加起来总共每秒较少数量的事务(例如,每秒7个事务),这对于大规模游戏或对于许多传统的数据库应用是不够的。因此,需要一种分布式数据库系统,该分布式数据库系统在没有领导者的情况下达成共识并且是高效的。

发明内容

[0011] 在一些实施例中,一种装置包括在第一计算设备处的分布式数据库的实例,该第一计算设备被配置为包括在实现分布式数据库的计算设备集合内。该装置还包括被配置为定义链接到(linked to)第一事件集合的第一事件的处理器。该处理器被配置为从来自计算设备集合的第二计算设备接收表示(1)由第二计算设备定义并且(2)链接到第二事件集合的第二事件的信号。该处理器被配置为至少基于协议的结果来识别与第三事件集合相关联的顺序。该处理器被配置为在分布式数据库的实例中存储与第三事件集合相关联的顺序。

附图说明

[0012] 图1是示出根据实施例的分布式数据库系统的高级框图。

[0013] 图2是示出根据实施例的分布式数据库系统的计算设备的框图。

[0014] 图3-图6示出根据实施例的hashDAG的示例。

[0015] 图7是示出根据实施例的第一计算设备和第二计算设备之间的通信流程的流程图。

[0016] 图8是示出根据实施例的第一计算设备和第二计算设备之间的通信流程的流程图。

[0017] 图9a-图9c是示出值的向量的示例的向量图。

[0018] 图10a-图10d是示出被更新以包括新值的值的向量的示例的向量图。

[0019] 图11是示出根据实施例的分布式数据库系统的操作的流程图。

[0020] 图12是示出根据实施例的分布式数据库系统的操作的流程图。

[0021] 图13是示出根据实施例的分布式数据库系统的操作的流程图。

[0022] 图14是根据实施例的hashDAG的示例。

[0023] 图15是根据实施例的hashDAG的示例。

[0024] 图16a-图16b示出根据实施例的用于与hashDAG一起使用的示例共识方法。

[0025] 图17a-17b示出根据另一个实施例的用于与hashDAG一起使用的示例共识方法。

具体实施方式

[0026] 在一些实施例中,一种装置包括在第一计算设备处的分布式数据库的实例,该第一计算设备被配置为包括在计算设备集合内,该计算设备集合经由可操作地耦合到计算设备集合的网络来实现分布式数据库。该装置还包括可操作地耦合到存储分布式数据库的实

例的存储器的处理器。处理器被配置为在第一时间定义链接到第一事件集合的第一事件。处理器被配置为在第一时间之后的第二时间并且从计算设备集合中的第二计算设备接收表示(1)由第二计算设备定义并且(2)链接到第二事件集合的第二事件的信号。处理器被配置为至少基于协议的结果来识别与第三事件集合相关联的顺序。第三事件集合中的每个事件来自第一事件集合或第二事件集合中的至少一个。处理器被配置为在分布式数据库的实例中存储与第三事件集合相关联的顺序。

[0027] 在一些情况下，第三事件集合中的每个事件与属性集合(例如，序列号、世代(generation)号、轮次号、接收号和/或时间戳等)相关联。协议的结果可以包括第三事件集合中的每个事件的属性集合中的每个属性的值。属性集合中的第一属性的值可以包括第一数值，并且属性集合中的第二属性的值可以包括与第一数值相关联的二进制值。第三事件集合中的事件的第二属性的二进制值(例如，轮次增量值)可以基于该事件与链接到该事件的第四事件集合之间的关系是否满足标准(例如，由该事件强烈(strongly)识别出的事件数量)。第四事件集合中的每个事件(1)是第三事件集合中的事件的祖先，并且(2)与第四事件集合中的剩余事件的第一公共属性(例如，公共轮次号、作为R轮第一事件的指示等)相关联。第一公共属性可以指示如下第一实例：由计算设备集合中的每个计算设备定义的事件与第一特定值(例如，作为R轮第一事件的指示等)相关联。

[0028] 属性集合中的第三属性(例如，接收轮次号)的值可以包括基于事件与链接到该事件的第五事件集合之间的关系的第二数值。第五事件集合中的每个事件是该事件的后代并且与作为第五事件集合中的剩余事件的第二公共属性(例如，是著名的)相关联。第二公共属性可以(1)与第三公共属性(例如，是R轮第一事件或证人)相关联，其中第三公共属性指示由计算设备集合中的每个计算设备定义的第二事件与不同于第一特定值的第二特定值相关联的第一实例，以及(2)与基于指示集合的结果相关联。指示集合中的每个指示可以与第六事件集合中的事件相关联。第六事件集合中的每个事件可以与第四公共属性相关联，第四公共属性指示由计算设备集合中的每个计算设备定义的第三事件与不同于第一特定值和第二特定值的第三特定值相关联的第一实例。在一些情况下，第一特定值是第一整数(例如，第一轮次号R)，第二特定值是大于第一整数的第二整数(例如，第二轮次号R+n)并且第三特定值是大于第二整数的第三整数(例如，第三轮次号R+n+m)。

[0029] 在一些实施例中，一种装置包括存储器和处理器。存储器包括在第一计算设备处的分布式数据库的实例，该第一计算设备被配置为包括在经由可操作地耦合到计算设备集合的网络来实现分布式数据库的计算设备集合内。处理器可操作地耦合到存储分布式数据库的实例的存储器，并且被配置为接收表示链接到事件集合的事件的信号。处理器被配置为至少基于协议的结果来识别与事件集合相关联的顺序。处理器被配置为在分布式数据库的实例中存储与事件集合相关联的顺序。

[0030] 在一些实施例中，非暂态处理器可读介质存储表示指令的代码，该指令要由处理器执行，以接收表示链接到事件集合的事件的信号，以及基于与事件集合中的每个事件相关联的轮次和对于何时递增与每个事件相关联的轮次的指示来识别与事件集合相关联的顺序。代码还包括使处理器在第一计算设备处的分布式数据库的实例中存储与事件集合相关联的顺序的代码，该第一计算设备被配置为包括在经由可操作地耦合到计算设备集合的网络来实现分布式数据库的计算设备集合内。分布式数据库的实例可操作地耦合到处理

器。

[0031] 在一些实施例中，在第一计算设备处的分布式数据库的实例可以被配置为包括在经由可操作地耦合到计算设备集合的网络来实现分布式数据库的计算设备集合内。第一计算设备在分布式数据库的实例中存储多个事务。数据库收敛模块可以在第一计算设备的存储器或处理器中实现。数据库收敛模块可以与分布式数据库的实例可操作地耦合。数据库收敛模块可以被配置为在第一时间定义链接到第一事件集合的第一事件。第一事件集合中的每个事件都是字节序列，并且(1)与多个事务集合中的事务集合相关联，以及(b)与该事务集合所关联的顺序相关联。事务集合中的每个事务都来自该多个事务。数据库收敛模块可以被配置为在第一时间之后的第二时间并且从计算设备集合中的第二计算设备接收(1)由第二计算设备定义的并且(2)链接到第二事件集合的第二事件。数据库收敛模块可以被配置为定义链接到第一事件和第二事件的第三事件。数据库收敛模块可以被配置为至少基于第一事件集合和第二事件集合来识别与第三事件集合相关联的顺序。第三事件集合中的每个事件来自第一事件集合或第二事件集合中的至少一个。数据库收敛模块可以被配置为至少基于(1)与第三事件集合相关联的顺序和(2)与多个事务集合中的每个事务集合相关联的顺序来识别与多个事务相关联的顺序。数据库收敛模块可以被配置为在分布式数据库的实例中存储与被存储在第一计算设备中的多个事务相关联的顺序。

[0032] 在一些实施例中，第一计算设备处的分布式数据库的实例可以被配置为被包括在经由可操作地耦合到计算设备集合的网络来实现分布式数据库的计算设备集合内。数据库收敛模块可以在第一计算设备的存储器或处理器中实现。数据库收敛模块可以被配置为在第一时间定义链接到第一事件集合的第一事件。第一事件集合中的每个事件都是字节序列。数据库收敛模块可以被配置为在第一时间之后的第二时间并且从计算设备集合中的第二计算设备接收(1)由第二计算设备定义的并且(2)链接到第二事件集合的第二事件。第二事件集合中的每个事件都是字节序列。数据库收敛模块可以被配置为定义链接到第一事件和第二事件的第三事件。数据库收敛模块可以被配置为至少基于第一事件集合和第二事件集合来识别与第三事件集合相关联的顺序。第三事件集合中的每个事件来自第一事件集合或第二事件集合中的至少一个。数据库收敛模块可以被配置为在分布式数据库的实例中存储与第三事件集合相关联的顺序。

[0033] 在一些实施例中，可以在计算设备集合中的第一计算设备处接收与第一事务相关联的数据，其中计算设备集合经由可操作地耦合到计算设备集合的网络来实现分布式数据库。计算设备集合中的每个计算设备都具有分布式数据库的单独实例。可以在第一时间定义与第一事务相关联的第一事务顺序值。可以从计算设备集合中的第二计算设备接收与第二事务相关联的数据。事务集合可以被存储在第一计算设备处的分布式数据库的实例中。事务集合可以至少包括第一事务和第二事务。可以在第一时间之后的第二时间选择至少包括第一事务顺序值和第二事务顺序值的事务顺序值集合。第二事务顺序值可以与第二事务相关联。数据库状态变量可以至少基于事务集合和事务顺序值集合来定义。

[0034] 在一些实施例中，一种方法包括从计算设备集合中的第一计算设备处的分布式数据库的实例接收第一事件，该计算设备集合经由可操作地耦合到计算设备集合的网络来实现分布式数据库。该方法还包括基于第一事件和第二事件来定义第三事件。第三事件链接到事件集合。可以至少部分地基于满足权益值标准的、与事件集合相关联的集体权益值来

为第四事件定义顺序值。顺序值可以被存储在计算设备集合中的第二计算设备处的分布式数据库的实例中。在一些实施例中，该方法还包括基于权益值集合的总和来计算集体权益值。权益值集合中的每个权益值都与定义事件集合中的事件的分布式数据库的实例相关联。

[0035] 在一些实施例中，一种方法包括从计算设备集合中的第一计算设备处的分布式数据库的实例接收第一事件，该计算设备集合经由可操作地耦合到计算设备集合的网络来实现分布式数据库。该方法还包括基于第一事件和第二事件来定义第三事件，并且至少部分地基于第三事件来确定第一事件集合。第一事件集合中的每个事件a)由第二事件集合来识别，并且b)与第一轮次号相关联。与第二事件集合相关联的集体权益值满足第一权益值标准，并且第二事件集合中的每个事件(1)由分布式数据库的不同实例来定义，并且(2)由第三事件来识别。可以基于确定与第一事件集合中的每个事件相关联的权益值的总和满足第二权益值标准来计算第三事件的轮次号。第一事件的轮次号对应于大于第一轮次号的第二轮次号。该方法还包括基于第三事件确定第三事件集合。第三事件集合中的每个事件a)由包括第三事件的第四事件集合来识别，并且b)来自第一事件集合。第四事件集合中的每个事件由分布式数据库的不同实例定义，并且与第四事件集合相关联的集体权益值满足第三权益值标准。然后基于满足第四权益值标准的、与第三事件集合相关联的集体权益值来为第四事件定义顺序值，并且可以将该顺序值存储在第二计算设备处的分布式数据库的实例中。

[0036] 在一些实施例中，权益值集合包括(1)与定义第二事件集合中的事件的分布式数据库的每个实例相关联并且(2)与分布式数据库的该实例所关联的加密货币的量成比例的权益值。与第二事件集合相关联的集体权益值基于权益值集合中的权益值的总和。

[0037] 在一些实施例中，基于分布式数据库的集体权益值来定义第一权益值标准、第二权益值标准、第三权益值标准或第四权益值标准中的至少一个。此外，在一些实施例中，在第一时间实现分布式数据库的计算设备集合与可信实体集合相关联，并且在第一时间之后的第二时间实现分布式数据库的计算设备集合与包括不是来自可信实体集合的实体的实体集合相关联。

[0038] 如本文所使用的，模块可以是例如与执行特定功能相关联的任何组件和/或可操作地耦合的电部件集合，并且可以包括例如存储器、处理器、电迹线、光学连接器、(在硬件中执行的)软件等等。

[0039] 如本说明书中所使用的，除非上下文另外明确指出，否则单数形式“一”、“一个”和“该”包括复数指示物。因此，例如，术语“模块”旨在表示单个模块或模块的组合。例如，“网络”旨在表示单个网络或网络的组合。

[0040] 图1是示出根据实施例的分布式数据库系统100的高级框图。图1示出了跨四个计算设备(计算设备110、计算设备120、计算设备130和计算设备140)实现的分布式数据库100，但是应当理解的是，分布式数据库100可以使用包括图1中未示出的计算设备的任何数量的计算设备集合。网络105可以是被实现为有线网络和/或无线网络并且用于可操作地耦合计算设备110、120、130、140的任何类型的网络(例如，局域网(LAN)、广域网(WAN)、虚拟网络、电信网络)。如本文进一步详细描述的，在一些实施例中，例如，计算设备是经由互联网服务提供商(ISP)和互联网(例如，网络105)相互连接的个人计算机。在一些实施例中，可以

经由网络105在任何两个计算设备110、120、130、140之间定义连接。如图1所示，例如，可以在计算设备110与计算设备120、计算设备130或计算设备140中的任何一个之间定义连接。

[0041] 在一些实施例中，计算设备110、120、130、140可以经由中间网络和/或可替代网络(图1中未示出)彼此通信(例如，向彼此发送数据和/或从彼此接收数据)和与网络通信。这样的中间网络和/或可替代网络可以是与网络105类型相同和/或类型不同的网络。

[0042] 每个计算设备110、120、130、140可以是被配置为通过网络105发送数据和/或从其它计算设备中的一个或多个接收数据的任何类型的设备。计算设备的示例在图1中示出。计算设备110包括存储器112、处理器111和输出设备113。存储器112可以是例如随机存取存储器(RAM)、存储器缓冲器、硬盘驱动器、数据库、可擦除可编程只读存储器(EPROM)、电可擦除只读存储器(EEPROM)、只读存储器(ROM)等。在一些实施例中，计算设备110的存储器112包括与分布式数据库的实例(例如，分布式数据库实例114)相关联的数据。在一些实施例中，存储器112存储指令，该指令用于使处理器执行与以下操作相关联的模块、过程和/功能：向分布式数据库的另一个实例(例如，在计算设备120处的分布式数据库实例124)发送和/或从分布式数据库的另一个实例接收同步事件的记录、与其它计算设备的先前同步事件的记录、同步事件的顺序、参数(例如，量化事务的数据库字段、量化事件发生的顺序的数据库字段和/或值可以针对其被存储在数据库中的任何其它合适的字段)的值。

[0043] 分布式数据库实例114可以例如被配置为操纵数据，包括存储、修改和/或删除数据。在一些实施例中，分布式数据库实例114可以是关系数据库、对象数据库、后关系数据库和/或任何其它合适类型的数据库。例如，分布式数据库实例114可以存储与任何特定功能和/或行业有关的数据。例如，分布式数据库实例114可以存储(例如，计算设备110的用户的)金融事务，该金融事务包括与特定金融工具的所有权历史有关的值和/或值的向量。通常，向量可以是参数的值的任何集合，并且参数可以是能够取不同值的任何数据对象和/或数据库字段。因此，分布式数据库实例114可以具有多个参数和/或字段，该多个参数和/或字段中的每一个与值的向量相关联。值的向量用于确定该数据库实例114内的参数和/或字段的实际值。

[0044] 在一些情况下，分布式数据库实例114还可以用于实现其它数据结构，诸如(键，值)对集合。由分布式数据库实例114记录的事务可以是例如添加、删除或修改(键，值)对集合中的(键，值)对。

[0045] 在一些情况下，可以查询分布式数据库系统100或分布式数据库实例114、124、134、144中的任何分布式数据库实例。例如，查询可以包括键，并且从分布式数据库系统100或分布式数据库实例114、124、134、144返回的结果可以是与该键相关联的值。在一些情况下，分布式数据库系统100或分布式数据库实例114、124、134、144中的任何分布式数据库实例还可以通过事务而被修改。例如，修改数据库的事务可以包含授权修改事务的一方的数字签名。

[0046] 分布式数据库系统100可以用于许多目的，诸如例如将与各种用户相关联的属性存储在分布式身份系统中。例如，这样的系统可以使用用户的身份作为“键”，并且使用与用户相关联的属性的列表作为“值”。在一些情况下，身份可以是密码公钥，并且对应的私钥是该用户已知的。每个属性可以例如由有权声明(assert)该属性的授权方进行数字签名。每个属性还可以例如用与有权读取属性的个体或个体的组相关联的公钥进行加密。一些键或

值还可以具有附加到它们的被授权修改或删除键或值的各方的公钥列表。

[0047] 在另一个示例中，分布式数据库实例114可以存储与大型多玩家游戏(MMG)有关的数据，诸如游戏物品的当前状态和所有权。在一些情况下，分布式数据库实例114可以在计算设备110内实现，如图1所示。在其它情况下，分布式数据库的实例可由计算设备(例如，经由网络)访问，但是不在该计算设备中实现(图1中未示出)。

[0048] 计算设备110的处理器111可以是被配置为运行和/或执行分布式数据库实例114的任何合适的处理设备。例如，处理器111可以被配置为响应于从计算设备120接收到信号而更新分布式数据库实例114，和/或使信号被发送到计算设备120，如本文进一步详细描述的。更具体而言，如本文进一步详细描述的，处理器111可以被配置为响应于接收到与来自另一个计算设备的事务相关联的同步事件、与同步事件的顺序相关联的记录等而执行用于更新分布式数据库实例114的模块、功能和/或过程。在其它实施例中，处理器111可以被配置为响应于接收到存储在分布式数据库的另一个实例(例如，计算设备120处的分布式数据库实例124)中的参数的值而执行模块、功能和/或过程以更新分布式数据库实例114，和/或使存储在计算设备110处的分布式数据库实例114中的参数的值被发送到计算设备120。在一些实施例中，处理器111可以是通用处理器、现场可编程门阵列(FPGA)、专用集成电路(ASIC)、数字信号处理器(DSP)等等。

[0049] 显示器113可以是任何合适的显示器，诸如例如液晶显示器(LCD)、阴极射线管显示器(CRT)等。在其它实施例中，作为显示器113、123、133、143的替代或除了显示器113、123、133、143之外，计算设备110、120、130、140中的任何一个还包括另一种输出设备。例如，计算设备110、120、130、140中的任何一个可以包括音频输出设备(例如，扬声器)、触感输出设备等。在另外的其它实施例中，作为显示器113、123、133、143的替代或除了显示器113、123、133、143之外，计算设备110、120、130、140中的任何一个还包括输入设备。例如，计算设备110、120、130、140中的任何一个可以包括键盘、鼠标等等。

[0050] 计算设备120具有处理器121、存储器122和显示器123，它们可以在结构上和/或功能上分别类似于处理器111、存储器112和显示器113。此外，分布式数据库实例124可以在结构上和/或功能上类似于分布式数据库实例114。

[0051] 计算设备130具有处理器131、存储器132和显示器133，它们可以在结构上和/或功能上分别类似于处理器111、存储器112和显示器113。此外，分布式数据库实例134可以在结构上和/或功能上类似于分布式数据库实例114。

[0052] 计算设备140具有处理器141、存储器142和显示器143，它们可以在结构上和/或功能上分别类似于处理器111、存储器112和显示器113。此外，分布式数据库实例144可以在结构上和/或功能上类似于分布式数据库实例114。

[0053] 虽然计算设备110、120、130、140被示出为彼此相似，但是分布式数据库系统100中的每个计算设备可以与其它计算设备不同。分布式数据库系统100的每个计算设备110、120、130、140可以是例如计算实体(例如，个人计算设备，诸如台式计算机，膝上型计算机等)、移动电话、个人数字助理(PDA)等中的任何一个。例如，计算设备110可以是台式计算机，计算设备120可以是智能电话，并且计算设备130可以是服务器。

[0054] 在一些实施例中，计算设备110、120、130、140的一个或多个部分可以包括基于硬件的模块(例如，数字信号处理器(DSP)、现场可编程门阵列(FPGA))和/或基于软件的模块

(例如,存储在存储器中和/或在处理器处执行的计算机代码的模块)。在一些实施例中,与计算设备110、120、130、140相关联的功能中的一个或多个功能(例如,与处理器111、121、131、141相关联的功能)可以被包括在一个或多个模块中(参见例如图2)。

[0055] 分布式数据库系统100的属性,包括计算设备(例如,计算设备110、120、130、140)的属性、计算设备的数量和网络105可以以任何数量的方式来选择。在一些情况下,分布式数据库系统100的属性可以由分布式数据库系统100的管理员来选择。在其它情况下,分布式数据库系统100的属性可以由分布式数据库系统100的用户集体选择。

[0056] 由于使用了分布式数据库系统100,因此在计算设备110、120、130和140之间不指定领导者。具体而言,计算设备110、120、130或140都不被识别和/或选择为解决计算设备110、120、130、140的分布式数据库实例111、12、131、141中存储的值之间的争议的领导者。替代地,通过使用本文描述的事件同步过程、投票过程和/或方法,计算设备110、120、130、140可以共同地关于参数的值收敛。

[0057] 在分布式数据库系统中没有领导者会增加分布式数据库系统的安全性。具体而言,在具有领导者的情况下,存在单一的攻击点和/或故障点。如果恶意软件感染领导者和/或领导者的分布式数据库实例处的参数的值被恶意更改,则故障和/或不正确的值被传播到遍布其它分布式数据库实例。但是,在无领导者系统中,不存在单一的攻击点和/或故障点。具体而言,如果无领导者系统的分布式数据库实例中的参数包含值,则该值将在分布式数据库实例与系统中的其它分布式数据库实例交换值之后改变,如本文进一步详细描述的。此外,本文描述的无领导者分布式数据库系统增加了收敛的速度,同时减少了设备之间发送的数据量,如本文进一步详细描述的。

[0058] 图2示出根据实施例的分布式数据库系统(例如,分布式数据库系统100)的计算设备200。在一些实施例中,计算设备200可以类似于关于图1示出和描述的计算设备110、120、130、140。计算设备200包括处理器210和存储器220。处理器210和存储器220可操作地耦合到彼此。在一些实施例中,处理器210和存储器220可以分别类似于关于图1详细描述的处理器111和存储器112。如图2所示,处理器210包括数据库收敛(convergence)模块211和通信模块210,并且存储器220包括分布式数据库实例221。通信模块212使得计算设备200能够与其它计算设备通信(例如,向其它计算设备发送数据和/或从其它计算设备接收数据)。在一些实施例中,通信模块212(图1中未示出)使得计算设备110能够与计算设备120、130、140通信。通信模块210可以包括和/或启用例如网络接口控制器(NIC)、无线连接、有线端口等等。如此,通信模块210可以建立和/或维持计算设备200与另一个设备之间(例如,经由诸如图1的网络105或互联网(未示出)之类的网络)的通信会话。类似地,通信模块210可以使得计算设备200能够向另一个设备发送数据和/或从另一个设备接收数据。

[0059] 在一些情况下,数据库收敛模块211可以与其它计算设备交换事件和/或事务,存储数据库收敛模块211接收的事件和/或事务,以及基于由事件之间的引用模式定义的部分顺序来计算事件和/或事务的排序。每个事件可以是包含两个较早事件的密码散列(将事件链接到该两个较早的事件及其祖先事件,反之亦然)、有效载荷数据(诸如要记录的事务)、诸如当前时间之类的其它信息、其创建者声明是事件第一次被定义的时间的时间戳(例如,日期和UTC时间)等的记录。在一些情况下,由成员定义的第一事件仅包含由另一个成员定义的单个事件的散列。在这种情况下,成员还没有先前的自散列(例如,由该成员先前定义

的事件的散列)。在一些情况下,分布式数据库中的第一事件不包含任何先前事件的散列(因为不存在该分布式数据库的先前事件)。

[0060] 在一些实施例中,两个较早事件的这种密码散列可以是基于使用事件作为输入的密码散列函数而定义的散列值。具体而言,在这样的实施例中,事件包括字节的特定序列或串(其表示该事件的信息)。事件的散列可以是从使用该事件的字节序列作为输入的散列函数返回的值。在其它实施例中,与事件相关联的任何其它合适的数据(例如,表示事件的特定部分的字节、标识符、序列号等)可以被用作散列函数的输入来计算该事件的散列。任何合适的散列函数都可以用于定义散列。在一些实施例中,每个成员使用相同的散列函数,使得针对给定事件在每个成员处生成相同的散列。然后事件可以由定义和/或创建该事件的成员进行数字签名。

[0061] 在一些情况下,事件集合以及它们的互连可以形成有向无环图(DAG)。在一些情况下,DAG中的每个事件引用两个较早的事件(将该事件链接到该两个较早的事件及其祖先事件,反之亦然),并且每个引用严格限定于较早的事件,使得不存在循环。在一些实施例中,由于DAG基于密码散列,因此数据结构可以被称为hashDAG。hashDAG直接对部分顺序进行编码,这意味着如果Y包含X的散列,或者如果Y包含包括X的散列的事件的散列或者对于任意长度的这样的路径,则已知事件X在事件Y之前到来。但是,如果从X到Y或从Y到X没有路径,则部分顺序不定义哪个事件先到来。因此,数据库收敛模块可以基于部分顺序计算总顺序。这可以通过由计算设备使用的任何合适的确定性函数来完成,使得计算设备计算相同的顺序。在一些实施例中,每个成员可以在每次同步之后重新计算该顺序,并且最终这些顺序可以收敛,以使得出现共识。

[0062] 可以使用共识算法来确定hashDAG中事件的顺序和/或事件内存储的事务的顺序。作为根据顺序执行这些事务的结果,事务的顺序又可以定义数据库的状态。所定义的数据库的状态可以被存储为数据库状态变量。

[0063] 在一些情况下,数据库收敛模块可以使用以下函数来依据hashDAG中的部分顺序计算总顺序。对于其它计算设备(称为“成员”)中的每个计算设备,数据库收敛模块可以检查hashDAG以发现由该成员接收到的事件(和/或这些事件的指示)的顺序。然后,数据库收敛模块可以进行计算,就好像该成员向每个事件指派了数字“排名”一样,其中对于成员接收到的第一事件的排名为1,对于成员接收到的第二事件的排名为2,等等。数据库收敛模块可以针对hashDAG中的每个成员执行这个操作。然后,对于每个事件,数据库收敛模块可以计算所指派的排名的中值(median),并且可以按它们的中值对事件排序。这种排序可以以确定性的方式打破平局(ties),诸如按照它们的散列的数字顺序或者通过一些其它方法对两个平局事件进行排序,其中每个成员的数据库收敛模块使用相同的方法。这种排序的结果是总顺序。

[0064] 图6示出了用于确定总顺序的一个示例的hashDAG 640。HashDAG 640示出了两个事件(最低的带条纹的圆和最低的带虚线的圆),以及每个成员接收到这些事件(其它带条纹的圆和带虚线的圆)的指示的第一时间。顶部的每个成员的名称都是按照在它们的慢顺序中哪个事件是第一个来进行着色的。带条纹的初始投票比带虚线的初始投票更多,因此每个成员的共识投票被加条纹。换句话说,这些成员最终会收敛到协议,即,带条纹的事件在带虚线的事件之前发生。

[0065] 在这个示例中，成员(标记为Alice、Bob、Carol、Dave和Ed的计算设备)将工作以定义事件642还是事件644首先发生的共识。每个带条纹的圆指示其中成员第一次接收到事件644(和/或该事件644的指示)的事件。类似地，每个带虚线的圆指示其中成员第一次接收到事件642(和/或该事件642的指示)的事件。如hashDAG 640中所示，Alice、Bob和Carol各自在事件642之前接收到事件644(和/或事件644的指示)。Dave和Ed两者在事件644(和/或事件644的指示)之前接收到事件642(和/或事件642的指示)。因此，由于更多数量的成员在事件642之前接收到事件644，因此总顺序可以由每个成员来确定为指示事件644在事件642之前发生。

[0066] 在其它实例中，数据库收敛模块可以使用不同的函数来依据hashDAG中的部分顺序计算总顺序。在这样的实施例中，例如，数据库收敛模块可以使用以下函数来计算总顺序，其中正整数Q是由成员共享的参数。

[0067] creator(x)＝创建事件x的成员

[0068] anc(x)＝作为x的祖先的事件集合(包括x本身)

[0069] other(x)＝由刚好在x被创建之前同步的成员创建的事件

[0070] self(x)＝具有相同创建者的在x之前的最后一个事件

[0071] self(x,0)＝self(x)

[0072] self(x,n)＝self(self(x),n-1)

[0073] order(x,y)＝k，其中y是creator(x)获知的第k个事件

[0074]

$$last(x) = \{y | y \in anc(x) \wedge \neg \exists z \in anc(x), (y \in anc(z) \wedge creator(y) = creator(z))\}$$

[0075]

$$slow(x,y) = \begin{cases} \infty & if\ y \notin anc(x) \\ order(x,y) & if\ y \in anc(x) \wedge y \notin anc(self(x)) \\ fast(x,y) & if\ \forall i \in \{1,\ldots,Q\},\ fast(x,y) = fast(self(x,i),y) \\ slow(self(x),y) & 否则 \end{cases}$$

[0076] fast(x,y)＝在排序列表中y的位置，其中元素z∈anc(x)通过 $\underset{w \in last(x)}{median}\ slow(w,z)$ 进行排序并且平局由每个事件的散列打破。

[0077] 在这个实施例中，基本上紧接在x被创建和/或被定义之后，fast(x,y)给出基于creator(x)的观点的y在事件的总顺序中的位置。如果Q是无穷大，则以上计算出与前述实施例中相同的总顺序。如果Q是有限的，并且所有成员都在线，则以上计算出与前述实施例中相同的总顺序。如果Q是有限的，并且在给定的时间少数成员在线，则这个函数允许在线成员在它们之间达成共识，该共识随着新成员缓慢地一个接一个地上线将保持不变。但是，如果存在网络的分区，那么每个分区的成员可以达成它们自己的共识。然后，当分区复原时，较小分区的成员将采用较大分区的共识。

[0078] 在还有的其它实例中，如关于图14-图17b所描述的，数据库收敛模块可以使用不同函数来依据hashDAG中的部分顺序计算总顺序。如图14-图15所示，每个成员(Alice、Bob、Carol、Dave和Ed)创建和/或定义事件(如图14中所示的1401-1413；如图15中所示的1501-1506)。通过使用关于图14-图17b描述的函数和子函数，事件的总顺序可以通过按照它们的

接收轮次(本文中也称为顺序值)对事件进行排序、通过它们的接收时间戳打破平局、以及通过它们的签名打破这些平局来计算,如本文进一步详细描述的。在其它情况下,事件的总顺序可以通过按照它们的接收轮次对事件进行排序、通过它们的接收世代(而不是它们被接收的时间戳)打破平局、以及通过它们的签名打破这些平局来计算。以下段落指定用于计算和/或定义事件的接收轮次和接收世代以确定事件的顺序的函数。以下术语结合图14-图17b被使用和示出。

[0079] “父亲”:如果事件Y包含事件X的散列,则事件X是事件Y的父亲。例如,在图14中,事件1412的父亲包括事件1406和事件1408。

[0080] “祖先”:事件X的祖先是X、X的父亲、X的父亲的父亲,等等。例如,在图14中,事件1412的祖先是事件1401、1402、1403、1406、1408和1412。事件的祖先可以被称为链接到该事件,反之亦然。

[0081] “后代”:事件X的后代是X、X的孩子、X的孩子的孩子,等等。例如,在图14中,事件1401的后代是图中示出的每个事件。又例如,事件1403的后代是事件1403、1404、1406、1407、1409、1410、1411、1412和1413。事件的后代可以被称为链接到该事件,反之亦然。

[0082] “N”:群体中成员的总数。例如,在图14中,成员是被标记为Alice、Bob、Carol、Dave和Ed的计算设备,并且N等于五。

[0083] “M”:大于N的某个百分比(例如,大于N的2/3)的最小整数。例如,在图14中,如果百分比定义为2/3,则M等于四。在其它情况下,M可以被定义为例如N的不同百分比(例如,1/3、1/2等)、特定的预定义的数和/或以任何其它合适的方式被定义。

[0084] “自父亲(self-parent)”:事件X的自父亲是由同一成员创建和/或定义的X的父亲事件Y。例如,在图14中,事件1405的自父亲是1401。

[0085] “自祖先(self-ancestor)”:事件X的自祖先是X、X的自父亲、X的自父亲的自父亲,等等。

[0086] “序列号”(或“SN”):事件的整数属性,被定义为事件的自父亲的序列号加1。例如,在图14中,事件1405的自父亲为1401。由于事件1401的序列号是1,因此事件1405的序列号是2(即,1加1)。

[0087] “世代号”(或“GN”):事件的整数属性,被定义为事件的父亲的世代号的最大值加1。例如,在图14中,事件1412具有两个父亲:事件1406和1408,它们分别具有世代号4和2。因此,事件1412的世代号是5(即,4加1)。

[0088] “轮次增量”(或“RI”):可以是零或一的事件的属性。

[0089] “轮次号”(或“RN”):事件的整数属性。在一些情况下,轮次号可以被定义为事件的父亲的轮次号的最大值加上事件的轮次增量。例如,在图14中,事件1412具有两个父亲:事件1406和1408,这两者都具有轮次号1。事件1412还具有轮次增量1。因此,事件1412的轮次号是2(即,1加1)。在其它实例中,如果R是最小整数,则事件可以具有轮次号R,使得该事件可以强烈地看到(如本文所述)由不同成员定义和/或创建的至少M个事件,该至少M个事件都具有轮次号R-1。如果不存在这样的整数,则事件的轮次号可以是默认值(例如,0、1等)。在这种情况下,可以不使用轮次增量来计算事件的轮次号。例如,在图14中,如果M被定义为大于N的1/2倍的最小整数,则M是三。然后,事件1412强烈地看到M个事件1401、1402和1408,这些事件中的每个事件由不同的成员定义并且具有轮次号1。事件1412不能强烈地看到由

不同的成员定义的具有轮次号2的至少M个事件。因此，事件1412的轮次号是2。在一些情况下，分布式数据库中的第一事件包括轮次号1。在其它情况下，分布式数据库中的第一事件可以包括轮次号0或任何其它合适的数。

[0090] “分叉(forking)”：如果事件X和事件Y由同一成员定义和/或创建，并且事件X和事件Y中的任一个都不是另一个的自祖先，则事件X是与事件Y的分叉。例如，在图15中，成员Dave通过创建和/或定义具有同一自父亲(即，事件1501)的事件1503和1504来分叉，使得事件1503不是事件1504的自祖先，并且事件1504不是事件1503的自祖先。

[0091] 分叉的“识别”：分叉可以由在彼此分叉的两个事件之后创建和/或定义的第三事件“识别”，如果这两个事件都是第三事件的祖先的话。例如，在图15中，成员Dave通过创建事件1503和1504来分叉，这两个事件中的任一个都不是另一个的自祖先。由于事件1503和1504两者都是事件1506的祖先，因此该分叉可以由后来的事件1506来识别。在一些情况下，分叉的识别可以指示特定的成员(例如，Dave)已经作弊。

[0092] 事件的“识别”：如果X没有与Y分叉的祖先事件Z，则事件X“识别”或“看到”祖先事件Y。例如，在图14中，事件1412识别(也称为“看到”)事件1403，因为事件1403是事件1412的祖先，并且事件1412没有与事件1403分叉的祖先事件。在一些情况下，如果X在事件Y之前未识别分叉，则事件X可以识别事件Y。在这种情况下，即使事件X识别出由定义事件Y的成员在事件Y之后进行的分叉，事件X也可以看到事件Y。事件X不识别该成员在分支之后的事件。此外，如果成员定义了两个不同的事件，这两个事件都是该成员的历史中的第一事件，则事件X可以识别出分叉并且不识别该成员的任何事件。

[0093] “强烈识别”(本文中也称为“强烈地看到”)事件：如果X识别Y，则事件X“强烈地识别”(或“强烈地看到”)由与X相同的成员创建和/或定义的祖先事件Y。如果存在(1)包括X和Y两者并且(2)是事件X的祖先并且(3)是祖先事件Y的后代并且(4)由X识别并且(5)可以各自识别Y并且(6)由至少M个不同的成员创建和/或定义的事件集合S，则事件X“强烈地识别”不是由与X相同的成员创建和/或定义的祖先事件Y。例如，在图14中，如果M被定义为大于N的2/3的最小整数(即，M＝1+floor(2N/3)，其在这个示例中将是四)，则事件1412强烈地识别祖先事件1401，因为事件集合1401、1402、1406和1412是作为事件1412的祖先和事件1401的后代的至少四个事件的集合，并且它们由四个成员Dave、Carol、Bob和Ed分别创建和/或定义，并且事件1412识别事件1401、1402、1406和1412中的每一个，并且事件1401、1402、1406和1412中的每一个识别事件1401。类似地，如果X可以看到由不同成员创建或定义的至少M个事件(例如，事件1401、1402、1406和1412)，其中该至少M个事件中的每一个可以看到Y，则事件X(例如，事件1412)可以“强烈地看到”事件Y(例如，事件1401)。

[0094] “R轮第一”事件(在本文中也称为“证人”)：如果事件(1)具有轮次号R，并且(2)具有轮次号小于R的自父亲或者没有自父亲，则事件是“R轮第一”事件(或“证人”)。例如，在图14中，事件1412是“2轮第一”事件，因为它具有轮次号2，并且它的自父亲是事件1408，事件1408具有轮次号1(即，小于二)。

[0095] 在一些情况下，当且仅当X“强烈地识别”至少M个“R轮第一”事件时，事件X的轮次增量被定义为1，其中R是其父亲的最大轮次号。例如，在图14中，如果M被定义为大于N的1/2倍的最小整数，则M是三。然后，事件1412强烈地识别M个事件1401、1402和1408，所有这些事件都是1轮第一事件。1412的两个父亲都是1轮，并且1412强烈地识别至少M个1轮第一事件，

因此用于1412的轮次增量为一。图中标记有“RI＝0”的事件每个都不能强烈地识别至少M个1轮第一事件，因此它们的轮次增量为0。

[0096] 在一些情况下，可以使用以下方法来确定事件X是否可以强烈地识别祖先事件Y。对于每个R轮第一祖先事件Y，维护整数数组A1，其中每成员一个整数，从而给出其中该成员创建和/或定义事件X并且X可以识别Y的最低序列号。对于每个事件Z，维护整数数组A2，其中每成员一个整数，从而给出由该成员创建和/或定义的事件W的最高序列号，使得Z可以识别W。为了确定Z是否可以强烈地识别祖先事件Y，对使得A1[E]<＝A2[E]的元素位置E的数量计数。当且仅当该计数大于M时，事件Z可以强烈地识别Y。例如，在图14中，成员Alice、Bob、Carol、Dave和Ed可以各自识别事件1401，其中可以这样做的最早事件分别是它们的事件{1404,1403,1402,1401,1408}。这些事件具有序列号A1＝{1,1,1,1,1,1}。类似地，由事件1412识别的它们中的每一个的最近事件是事件{NONE(没有),1406,1402,1401,1412}，其中Alice被列为“NONE”，因为1412不能识别Alice的任何事件。这些事件分别具有序列号A2＝{0,2,1,1,2}，其中所有事件都具有正序列号，因此0意味着Alice没有由1412识别的事件。将列表A1与列表A2进行比较给出结果{1<＝0,1<＝2,1<＝1,1<＝1,1<＝2}，其相当于{假，真，真，真，真}，这具有四个值为真。因此，存在作为1412的祖先和1401的后代的四个事件的集合S。4是至少M，因此1412强烈地识别1401。

[0097] 实现用A1和A2确定事件X是否可以强烈地识别祖先事件Y的方法的另一个变型如下。如果两个数组中的整数元素小于128，则有可能将每个元素存储在单个字节中，并将8个这样的元素打包到单个64位字中，并让A1和A2成为这些字的数组。A1中每个字节的最高有效位可以被设置为0，并且A2中每个字节的最高有效位可以被设置为1。两个对应的字相减，然后用掩码执行按位与(bitwise AND)以将除了最高有效位之外的其它位清零，然后右移7位的位置，以得到用C编程语言表达如下的值：((A2[i]-A1[i])&0x8080808080808080)>>7)。这可以被添加到被初始化为零的正在运行的累加器S。在多次这样做之后，通过移位和添加字节来将累加器转换为计数，以得到：((S&0xff)+((S>>8)&0xff)+((S>>16)&0xff)+((S>>24)&0xff)+((S>>32)&0xff)+((S>>40)&0xff)+((S>>48)&0xff)+((S>>56))&0xff))。在一些情况下，这些计算可以用诸如C、Java等之类的编程语言来执行。在其它情况下，计算可以使用特定于处理器的指令来执行，特定于处理器的指令诸如由Intel和AMD提供的高级向量扩展(AVX)指令或者在图形处理单元(GPU)或通用图形处理单元(GPGPU)中的等效指令。在一些体系架构中，可以通过使用大于64位(诸如128位、256位、512位或更多位)的字来更快地执行计算。

[0098] “著名”事件：如果(1)事件X是“R轮第一”事件(或“证人”)并且(2)经由执行如下所述的拜占庭(Byzantine)一致性协议达到“是”的决定，则R轮事件X是“著名的”。在一些实施例中，拜占庭一致性协议可以由分布式数据库的实例(例如，分布式数据库实例114)和/或数据库收敛模块(例如，数据库收敛模块211)来执行。例如，在图14中，示出了五个1轮第一事件：1401、1402、1403、1404和1408。如果M被定义为大于N的1/2倍的最小整数(其是3)，那么1412是2轮第一。如果协议运行时间更长，那么hashDAG将向上生长，并且最终其它四个成员也将在这个图的顶部之上具有2轮第一。每个2轮第一将具有关于1轮第一中的每一个是否“著名”的“投票”。事件1412将对于1401、1402和1403是著名的投票为“是”(YES)，因为这些是它可以识别的1轮第一。事件1412将对于1404是著名的投票为“否”(NO)，因为1412不能

识别1404。对于给定的1轮第一，诸如1402，它是否“著名”的状态将通过计算对于它是否“著名”的每个2轮第一的投票来决定。然后，这些投票将传播到3轮第一，然后传播到4轮第一，等等，直到最终关于1402是否著名达成一致。对于其它第一重复相同的过程。

[0099] 拜占庭一致性协议可以收集和使用“R轮第一”事件的投票和/或决定来识别“著名”事件。例如，如果“R+1轮第一”Y可以“识别”事件X，则Y将投票“是”，否则它投票“否”。然后对于每轮G，对于G＝R+2、R+3、R+4等计算投票，直到由任何成员做出决定为止。对于每轮G计算投票，直到做出决定为止。这些轮(round)中的一些轮可以是“多数(majority)”轮，而一些其它轮可以是“硬币(coin)”轮。在一些情况下，例如，轮R+2是多数轮，并且未来的轮次(例如，根据预定义的时间表)被指定为多数轮或者硬币轮。例如，在一些情况下，未来的轮次是多数轮还是硬币轮可以被任意确定，但条件是不能存在两个连续的硬币轮。例如，可能预定义的是，将存在五个多数轮，然后是一个硬币轮，然后是五个多数轮，然后是一个硬币轮，这一过程重复直到达成一致。

[0100] 在一些情况下，如果轮G是多数轮，则可以如下计算投票。如果存在强烈地识别至少M个G-1轮第一的G轮事件投票V(其中V为“是”或者“否”)，则共识决定为V，并且拜占庭一致性协议结束。否则，每个G轮第一事件计算作为每个G轮第一事件可以强烈识别的G-1轮第一的多数票的新的投票。在存在平局而非多数票的情况下，投票可以被指定为“是”。

[0101] 类似地，如果X是R轮证人(或R轮第一)，那么可以计算轮R+1、R+2等中的投票结果，其中每轮中的证人都对于X是否著名进行投票。在轮R+1中，可以看到X的每个证人投票“是”，并且其它证人投票“否”。在轮R+2中，每个证人根据它可以强烈地看到的R+1轮证人的投票的多数票进行投票。类似地，在轮R+3中，每个证人根据它可以强烈地看到的R+2轮证人的投票的多数票进行投票。这可以持续多轮。在平局的情况下，投票可以被设置为“是”。在其它情况下，平局可以被设置为“否”或者可以被随机设置。如果任何轮次具有证人中的至少M个证人投票“否”，则选举结束，并且X不是著名的。如果任何轮次具有证人中的至少M个证人投票“是”，则选举结束，并且X是著名的。如果“是”和“否”都没有至少M个投票，则选举继续到下一轮。

[0102] 作为示例，在图14中，考虑所示图下方的某轮第一事件X。然后，每个1轮第一将关于X是否著名进行投票。事件1412可以强烈地识别1轮第一事件1401、1402和1408。因此，事件1412的投票将基于它们的投票。如果这是多数轮，则1412将检查{1401,1402,1408}中的至少M个是否具有“是”的投票。如果它们确实具有，则决定为“是”，并且一致性已经实现。如果它们中的至少M个投票“否”，则决定为“否”，并且一致性已经实现。如果投票在任一方向上都不具有至少M个，则1412被给予作为1401、1402和1408的投票的多数票的投票(如果存在平局，则将通过投票“是”来打破平局)。该投票然后将在下一轮中被使用，继续直到达成一致为止。

[0103] 在一些情况下，如果轮G是硬币轮，则可以如下计算投票。如果事件X可以识别至少M个G-1轮第一投票V(其中V为“是”或者“否”)，则事件X将把它的投票改变为V。否则，如果轮G是硬币轮，则每个G轮第一事件X将它的投票改变为伪随机确定的结果(类似于在一些情况下的掷硬币)，该结果被定义为事件X的签名的最低有效位。

[0104] 类似地，在这种情况下，如果选举达到轮R+K(硬币轮)，其中K是指定因数(例如，诸如3、6、7、8、16、32之类的数字的倍数，或任何其它合适的数字)，则选举在那一轮不结束。如

果选举达到这一轮，则它可以继续至少再一轮。在这一轮中，如果事件Y是R+K轮证人，则如果它可以强烈地看到轮R+K-1中至少M个证人投票V，则Y将投票V。否则，Y将根据随机值(例如，根据事件Y的签名的位(例如，最低有效位、最高有效位、随机选择的位)，其中1＝“是”且0＝“否”，或者反之亦然，根据事件Y的时间戳、使用密码“共享币”协议和/或任何其它随机确定)进行投票。在Y被创建之前，这种随机确定是不可预测的，并且因此可以提高事件和共识协议的安全性。

[0105] 例如，在图14中，如果轮2是硬币轮，并且投票是关于轮1之前的某个事件是否是著名的，则事件1412将首先检查{1401,1402,1408}中的至少M个投票为“是”，还是它们中的至少M个投票为“否”。如果是这种情况，则1412将以相同的方式投票。如果在任一方向上都没有至少M个投票，则1412将具有随机投票或伪随机投票(例如，基于在Ed创建和/或定义事件1412时对事件1412签名时，Ed为事件1412创建的数字签名的最低有效位)。

[0106] 在一些情况下，伪随机确定的结果可以是密码共享币协议的结果，该结果可以例如被实现为轮次号的阈值签名的最低有效位。

[0107] 可以依据上文描述的用于计算伪随机确定的结果的方法中的任何一种方法来构建系统。在一些情况下，系统按某种顺序循环使用不同的方法。在其它情况下，系统可以根据预定义的模式在不同的方法之间进行选择。

[0108] “接收轮次”：如果R是使得具有轮次号R的著名R轮第一事件(或著名证人)中的至少一半是X的后代和/或可以看到X的最小整数，则事件X具有“接收轮次”R。在其它情况下，可以使用任何其它合适的百分比。例如，在另一个实例中，如果R是使得具有轮次号R的著名R轮第一事件(或著名证人)的至少预定百分比(例如，40％、60％、80％等)是X的后代和/或可以看到X的最小整数，则事件X具有“接收轮次”R。

[0109] 在一些情况下，事件X的“接收世代”可以如下计算。找出哪个成员创建和/或定义了可以识别事件X的每个R轮第一事件。然后确定该成员的可以识别X的最早事件的世代号。然后将X的“接收世代”定义为该列表的中值。

[0110] 在一些情况下，事件X的“接收时间戳”T可以是包括每个成员的识别和/或看到X的第一事件的事件中的时间戳的中值。例如，事件1401的接收时间戳可以是事件1402、1403、1403和1408的时间戳的值的中值。在一些情况下，事件1401的时间戳可以被包括在中值计算中。在其它情况下，X的接收时间戳可以是作为每个成员的识别或看到X的第一事件的事件中的时间戳的值中的任何其它值或组合。例如，X的接收时间戳可以基于时间戳的平均值、时间戳的标准差、修改后的平均值(例如，通过从计算中去除最早的和最新近的时间戳)，等等。在其它情况下，可以使用扩展的中值。

[0111] 在一些情况下，事件的总顺序和/或共识顺序通过按它们的接收轮次(本文中也称为顺序值)对事件进行排序、通过它们的接收时间戳打破平局、以及通过它们的签名打破这些平局来计算。在其它情况下，事件的总顺序可以通过按它们的接收轮次对事件进行排序、通过它们的接收世代打破平局、以及通过它们的签名打破这些平局来计算。以上段落指定了用于计算和/或定义事件的接收轮次、接收时间戳和/或接收世代的函数。

[0112] 在其它情况下，作为使用每个事件的签名的替代，可以使用该事件的签名与该轮次中具有相同的接收轮次和/或接收世代的著名事件或著名证人的签名进行异或(XOR)。在其它情况下，可以使用任何其它合适的事件签名的组合来打破平局以定义事件的共识顺

序。

[0113] 在还有的其它情况下，作为将“接收世代”定义为列表的中值的替代，可以将“接收世代”定义为列表本身。然后，当按接收世代进行排序时，两个接收世代可以通过它们的列表的中间元素进行比较，从而通过紧接中间之前的元素打破平局，通过紧接中间之后的元素打破平局，以及通过在到目前为止使用的元素之前的元素和之后的元素之间进行交替来继续，直到平局被打破为止。

[0114] 在一些情况下，中值时间戳可以用“扩展中值”来代替。在这种情况下，可以为每个事件定义时间戳列表，而不是单个接收时间戳。事件X的时间戳列表可以包括每个成员的识别和/或看到X的第一事件。例如，在图14中，事件1401的时间戳列表可以包括事件1402、1403、1403和1408的时间戳。在一些情况下，还可以包括事件1401的时间戳。当用时间戳列表打破平局(即，两个事件具有相同的接收轮次)时，每个事件的列表的中间时间戳(或者两个中间时间戳中的第一个或第二个中的预定的一个，如果具有偶数长度的话)可以进行比较。如果这些时间戳相同，则可以比较紧接在中间时间戳之后的时间戳。如果这些时间戳相同，则可以比较紧接在中间时间戳之前的时间戳。如果这些时间戳也相同，则比较三个已比较的时间戳之后的时间戳。这可以继续交替，直到平局被打破为止。与以上讨论类似，如果两个列表完全相同，则可以通过两个元素的签名来打破平局。

[0115] 在还有的其它情况下，可以使用“截断的扩展中值”而不是“扩展中值”。在这种情况下，不存储每个事件的时间戳的完整列表。相反，靠近列表的中间的值中的仅一些值被存储并用于比较。

[0116] 除了计算事件的总顺序之外，接收到的中值时间戳还可以潜在地用于其它目的。例如，Bob可能签署合约，该合约声明当且仅当存在包含其中Alice签署同一合约的事务的事件X时，他同意受合约约束，其中X的接收时间戳在某个截止日期时或在某个截止日期之前。在这种情况下，如以上所述，如果如“接收中值时间戳”所指示的，Alice在截止日期之后签署合约，则Bob将不受合约约束。

[0117] 在一些情况下，可以在达成共识之后定义分布式数据库的状态。例如，如果S(R)是R轮中著名证人可以看到的事件集合，则最终S(R)中的所有事件都将具有已知的接收轮次和接收时间戳。此时，S(R)中的事件的共识顺序是已知的并且将不改变。一旦达到这一点，成员就可以计算和/或定义事件和它们的顺序的表示。例如，成员可以按照事件的共识顺序来计算S(R)中的事件的散列值。然后，成员可以对散列值进行数字签名，并将散列值包含在成员定义的下一个事件中。这可以用于向其它成员通知该成员已经确定S(R)中的事件具有将不会改变的给定顺序。在成员中的至少M个成员(或任何其它合适数量或百分比的成员)已签署S(R)的散列值(并且因此同意由散列值表示的顺序)之后，事件的该共识列表以及成员的签名列表可以形成可以用于证明共识顺序为如S(R)中的事件声明的那样的单个文件(或其它数据结构)。在其它情况下，如果事件包含更新分布式数据库系统的状态的事务(如本文所述)，则散列值可以具有在以共识顺序应用S(R)中的事件的事务之后分布式数据库系统的状态。

[0118] 在一些情况下，M(如上所述)可以基于指派给每个成员的权重值(在本文中也称为权益(stake)值)，而不仅仅是总成员数的一部分、百分比和/或值。在这种情况下，每个成员具有与它在分布式数据库系统中的利益和/或影响相关联的权益。这样的权益可以是权重

值和/或权益值。该成员定义的每个事件都可以被称为具有它的定义成员的权重值。然后，M可以是所有成员的总权益的一部分，并且可以被称为权益值标准和/或阈值。当具有至少为M的权益总和的成员集合同意(即，满足权益值标准)时，上文描述为取决于M的事件将发生。因此，基于它们的权益，某些成员可以对系统以及如何得出共识顺序具有更大的影响。在一些情况下，事件中的事务可以改变一个或多个成员的权益、添加新成员和/或删除成员。如果这样的事务具有接收轮次R，则在接收轮次已被计算出之后，R轮证人之后的事件将使用修改后的权益和修改后的成员列表来重新计算它们的轮次号和其它信息。在一些情况下，关于R轮事件是否著名的投票可以使用旧的权益和成员列表，但是关于R之后的轮次的投票可以使用新的权益和成员列表。

[0119] 在还有的一些情况下，可以向分布式数据库系统的每个成员指派预定的权重或权益值。因此，经由拜占庭一致性协议达成的共识可以用相关联的安全级别来实现，以保护成员组或群体免受潜在的Sybil(女巫)攻击。在一些情况下，这种安全级别可以在数学上得到保证。例如，攻击者可能想要通过重组在hashDAG中注册的事件的部分顺序来影响一个或多个事件的结果。可以通过在分布式数据库的成员之间达成共识和/或最终一致之前重组一个或多个hashDAG部分顺序来进行攻击。在一些情况下，关于多个竞争事件发生的时机可能会引起争议。如以上所讨论的，与事件相关联的结果可以取决于M的值。因此，在一些情况下，当达成一致的投票成员的投票数或权益总数大于或等于M的值时，可以作出关于是Alice还是Bob首先针对事件(和/或事件内的事务)采取动作的确定。

[0120] 一些类型的事件顺序重组攻击要求攻击者控制N的至少一部分或百分比(例如，1/10、1/4、1/3等)，这取决于M的值。在一些情况下，M的值可以被配置为是例如组或群体N的2/3。在这种情况下，只要该组或群体的成员的2/3以上不是攻击的参与者，就可以由不是攻击的一部分的成员达成一致，并且分布式数据库将继续达成共识并按预期操作。此外，在这种情况下，攻击者在攻击时间段期间将必须控制至少组或群体的至少N减M(N-M)个成员(在这个示例中为成员的1/3)，以阻止数据库收敛，从而使分布式数据库以有利于攻击者的方式收敛(例如，使数据库以不公平的顺序收敛)、收敛到两个不同的状态(例如，使得成员正式同意两个矛盾的状态)或伪造货币(当分布式数据库系统与加密货币一起操作时)。

[0121] 在一些实现中，可以向组或群体的成员中的每个成员指派权重或权益，并且N将是所有它们的权重或权益的总和。因此，可以基于信任度或可靠性将较高的权重或权益值指派给群体或成员的组的子集。例如，可以将较高的权重或权益值指派给不太可能参与攻击或者具有示出它们没有参与不诚实行为的倾向的一些指标的成员。

[0122] 在一些情况下，可以通过选择M为N的较大部分来增加分布式数据库系统安全级别。例如，当M对应于大于群体或组的成员的数量N的2/3的最小整数，并且所有成员具有相等的投票权时，攻击者将需要控制或影响N的至少1/3来防止在非攻击者成员之间达成一致并且以使得分布式数据库不能达成共识。类似地，在这种情况下，攻击者将需要控制或影响超过N的至少1/3来使分布式数据库系统以有利于攻击者的方式收敛和/或达成一致(例如，使数据库以不公平的顺序收敛)、收敛到两个不同的状态(例如，使得成员正式同意两个矛盾的状态)或伪造货币(当分布式数据库系统与加密货币一起操作时)。

[0123] 在一些情况下，例如，不诚实的成员可以以两种不同的方式投票来使分布式数据库收敛在两种不同的状态上。例如，如果N＝300并且100个成员不诚实，则存在200个诚实成

员，其中例如100个为事务投票“是”，100个投票“否”。如果100个不诚实的成员向100个诚实的“是”投票者发送100个不诚实的成员投票“是”的消息(或事件)，则100个诚实的“是”投票者将认为最终的共识为“是”，因为它们将相信成员的2/3投票为“是”。类似地，如果100个不诚实的成员向100个诚实的“否”投票者发送100个不诚实的成员投票“否”的消息(或事件)，则100个诚实的“否”投票者将认为最终的共识为“否”，因为它们将相信成员的2/3投票为“否”。因此，在这种情况下，一些诚实的成员将认为共识为“是”，而其它诚实的成员将认为共识为“否”，从而导致分布式数据库收敛到两个不同的状态。但是，如果不诚实成员的数量少于100个，那么诚实成员将最终收敛到单个值(“是”或者“否”)，因为不诚实的成员将不能推动“是”和“否”的投票两者都超过200(即，N的2/3)。根据分布式数据库系统的应用的规范和/或具体要求，可以使用其它合适的M值。

[0124] 在一些其它情况下，当成员具有不相等的投票权时，例如，当最可靠或可信的投票者具有一个单位的投票权(例如，权重值或权益值)，而其余成员具有一个单位的一部分的投票权时，当权益或权重总和达到值M时可以达成一致。因此，在一些情况下，即使当多数成员与最终决定不一致，但是多数可靠的或可信的成员一致时，有时也可以达成一致。换句话说，不受信任的成员的投票权可以被削弱以防止或减轻可能的攻击。因此，在一些情况下，可以通过要求具有M的总权益的成员的共识，而不是仅仅M个成员的计数来增加安全级别。较高的M值意味着较大部分的权益(例如，未加权系统中的较多成员)必须同意以使得分布式数据库系统收敛。

[0125] 在一些情况下，分布式数据库系统可以支持多个参与度安全协议，包括但不限于表1中示出的协议及其任何组合。表1中示出的协议描述了用于向组或群体的成员指派权益或权重的多种技术。表1中的协议可以用加密货币来实现，加密货币诸如例如，比特币、本地加密货币的衍生物、在分布式数据库系统内定义的加密货币或任何其它合适类型的加密货币。虽然关于加密货币进行描述，但是在其它情况下，表1中示出的协议可以用在任何其它合适的分布式数据库系统中，以及与用于指派权益的任何其它方法一起使用。

[0126] 表1

[0127]

参与度安全协议的示例	
协议名称	*描述*
权益证明	每个成员可以将自己自身与它们拥有的一个或多个加密货币钱包相关联，并且它们的投票权益被设置为与这些一个或多个加密货币钱包的总余额相关联的值。
燃烧证明	类似于权益证明，但是成员证明它们销毁或消费了所讨论的（at issue）加密货币。换句话说，费用被支付以加入投票群体，并且投票权益与支付的金额成比例。
工作证明	成员可以通过执行计算任务或解决难题来获得投票权益。与常规的加密货币不同，计算成本可以产生以获得投票权，而不是挖掘区块。 在这个协议中，可能需要群体的成员继续执行计算任务或解决难题以跟上系统需求，并且不失去它们保持系统安全的能力。这个协议还可以被配置为使投票权益随时间衰减，以鼓励成员持续工作。
许可的	每个成员获得恰好一个单位的投票权益。成员只有在它们具有许可时才被允许成为成员。 加入群体的许可可以受到现有成员的投票、某个现有组织中的成员资格证明或任何其它合适的条件的制约。
混合的	群体的创始成员可以以相等的投票权益开始。创始成员可以邀请其它成员加入群体，因此成员资格可以像病毒一样传播。每个成员将与它们所邀请的成员分割它们自己的投票权益。以这种方式，如果成员邀请 1000 个马甲成为成员，那么它们所有 1001 个一起将仍然具有与该成员原来所具有的相同的总

[0128]

参与度安全协议的示例	
协议名称	*描述*
	投票权益。因此马甲将不会有助于发起 Sybil 攻击。
平常的（trivial）	每个成员被提供与一个单位对应的投票权益。任何成员都可以邀请新成员加入群体。新成员被提供与一个单位对应的投票权。

[0129] 在参与度安全协议之间的选择可以取决于具体应用。例如，混合协议可以适用于实现其中安全性和最小计算费用之间的折衷倾向于后者的临时的低价值事务、商业协作应用、计算机游戏和其它类似类型的应用。混合协议可以有效地防止单个不满的成员干扰或攻击群体或成员组。

[0130] 对于另一个示例，当安全性要求是最高优先级并且群体成员不是完全的陌生人或未知的时，许可协议可能是期望的。许可协议可以用于实现涉及例如银行和类似类型的金融实体或绑定在联合体(consortium)中的实体的应用。在这种情况下，联合体中的银行可以成为群体的成员，并且每个银行可以被限制为作为单个成员参与。因此，M可以被设置为超过群体的三分之二的最小整数。作为单独的实体，银行可能不相互彼此信任，但是可以依赖于由分布式数据库系统提供的安全级别，该安全级别在这个示例中将不诚实成员的数量限制为不超过群体中银行的三分之一。

[0131] 对于又一个示例，当群体包括大量陌生人或未知成员时，可以实施燃烧证明协议。在这种情况下，攻击者可能能够获得对超过给予M的值的总权益的一部分的控制。但是，入场费可以被设置得足够高，以使得攻击的成本超过任何预期的收益或利润。

[0132] 对于另一个示例，权益证明协议可以适用于较大的组。当存在拥有以大体相等的部分的大量加密货币的大成员组，并且无法预见或有可能有破坏性成员将获得比由该大成员组集体拥有的金额更高的加密货币金额时，权益证明协议可能是最佳的或理想的解决方案。

[0133] 在其它情况下，可以从表1中示出的协议或协议的组合得到其它进一步的或更复杂的协议。例如，分布式数据库系统可以被配置为实现混合协议，该混合协议遵循许可协议达预定的时间段，并最终允许成员向彼此出售投票权益。又例如，分布式数据库系统可以被配置为实施燃烧证明协议，并且一旦事件或所涉及的事务的值达到阈值或预定的加密货币值，就最终转变为权益证明协议。

[0134] 前面的术语、定义和算法用于示出图14-图17b中描述的实施例和概念。图16a和图16b示出以数学形式示出的共识方法和/或过程的第一示例应用。图17a和图17b示出以数学形式示出的共识方法和/或过程的第二示例应用。

[0135] 在其它情况下并且如本文进一步详细描述的，数据库收敛模块211可以初始地定义参数的值的向量，并且可以在它从其它计算设备接收到参数的附加值时更新值的向量。例如，数据库收敛模块211可以经由通信模块212从其它计算设备接收参数的附加值。在一

些情况下，数据库收敛模块可以基于参数的所定义的和/或更新后的值的向量来选择参数的值，如本文进一步详细描述的。在一些实施例中，数据库收敛模块211还可以经由通信模块212将参数的值发送到其它计算设备，如本文进一步详细描述的。

[0136] 在一些实施例中，数据库收敛模块211可以向存储器220发送信号以使得在存储器220中存储(1)参数的所定义的和/或更新后的值的向量，和/或(2)基于参数的所定义的和/或更新后的值的向量来选择的参数的值。例如，(1)参数的所定义的和/或更新后的值的向量和/或(2)基于参数的所定义的和/或更新后的值的向量来选择的参数的值可以被存储在存储器220中实现的分布式数据库实例221中。在一些实施例中，分布式数据库实例221可以类似于图1所示的分布式数据库系统100的分布式数据库实例114、124、134、144。

[0137] 在图2中，数据库收敛模块211和通信模块212在图2中被示为在处理器210中实现。在其它实施例中，数据库收敛模块211和/或通信模块212可以在存储器220中实现。在还有的其它实施例中，数据库收敛模块211和/或通信模块212可以是基于硬件的(例如，ASIC、FPGA等)。

[0138] 图7示出了根据实施例的两个计算设备同步事件的信号流程图。具体而言，在一些实施例中，分布式数据库实例703和803可以交换事件以获得收敛。计算设备700可以基于与计算设备700的关系、基于与计算设备700的接近度、基于与计算设备700相关联的有序列表等随机地选择与计算设备800同步。在一些实施例中，由于计算设备800可以由计算设备700从属于分布式数据库系统的计算设备集合中选择，因此计算设备700可以连续多次选择计算设备800或者可以在一段时间内不选择计算设备800。在其它实施例中，先前选择的计算设备的指示可以存储在计算设备700处。在这样的实施例中，计算设备700在能够再次选择计算设备800之前可以等待预定数量的选择。如上所述，分布式数据库实例703和803可以分别在计算设备700的存储器和计算设备800的存储器中实现。

[0139] 图3-图6示出了根据实施例的hashDAG的示例。存在五个成员，这五个成员中的每个成员由黑色的垂直线表示。每个圆圈表示事件。来自事件的两条向下的线表示两个先前事件的散列。在这个示例中，除了每个成员的第一事件之外，每个事件具有两条向下的线(一条粗线(dark line)到同一成员，并且一条细线到另一个成员)。时间向上进展。在图3-图6中，分布式数据库的计算设备被指示为Alice、Bob、Carol、Dave和Ed。应当理解的是，这样的指示是指结构上和功能上类似于关于图1示出和描述的计算设备110、120、130和140的计算设备。

[0140] 示例系统1：如果计算设备700被称为Alice，并且计算设备800被称为Bob，则它们之间的同步可以如图7中所示。Alice和Bob之间的同步可以如下：

[0141] -Alice向Bob发送存储在分布式数据库703中的事件。

[0142] -Bob创建和/或定义新事件，该新事件包含：

[0143] --Bob创建和/或定义的最后事件的散列

[0144] --Alice创建和/或定义的最后事件的散列

[0145] --Bob对上述内容的数字签名

[0146] -Bob向Alice发送存储在分布式数据库803中的事件。

[0147] -Alice创建和/或定义新事件。

[0148] -Alice向Bob发送该事件。

[0149] -Alice根据hashDAG计算事件的总顺序

[0150] -Bob根据hashDAG计算事件的总顺序

[0151] 在任何给定时间，成员可以存储到目前为止接收到的事件以及与创建和/或定义每个事件的计算设备和/或分布式数据库实例相关联的标识符。除了初始事件(其没有父亲散列)和每个新成员的第一事件(其具有单个父亲事件散列，该单个父亲事件散列表示邀请它们加入的现有成员的事件)之外，每个事件包含两个较早事件的散列。可以绘制表示这个事件集合的图。它可以示出对于每个成员的垂直线，并对于由该成员创建和/或定义的每个事件示出该线上的点。每当事件(较高的点)包含较早事件(较低的点)的散列时，就在两个点之间绘制一条斜线。如果事件可以经由该事件的散列(直接地或者通过中介事件)引用另一个事件，则该事件可以被称为链接到该另一个事件。

[0152] 例如，图3示出了hashDAG 600的示例。作为Bob与Carol同步的结果并且在与Carol同步之后，事件602由Bob创建和/或定义。事件602包括事件604(由Bob创建和/或定义的先前事件)的散列和事件606(由Carol创建和/或定义的先前事件)的散列。在一些实施例中，例如，包括在事件602内的事件604的散列包括指向它的直接祖先事件，即事件608和610的指针。由此，Bob可以使用事件602来引用事件608和610，并且使用指向先前事件的指针来重新构建hashDAG。在一些实例中，事件602可以被称为链接到hashDAG600中的其它事件，因为事件602可以经由较早的祖先事件而引用hashDAG 600中的每个事件。例如，事件602经由事件604链接到事件608。又例如，事件602经由事件606和事件612链接到事件616。

[0153] 示例系统2：来自示例系统1的系统，其中事件还包括要记录的事务的“有效载荷”或其它信息。这样的有效载荷可以用于利用自从计算设备的直接先前事件以来发生和/或被定义的任何事务和/或信息来更新事件。例如，事件602可以包括自从事件604被创建和/或定义以来由Bob执行的任何事务。因此，当将事件602与其它计算设备同步时，Bob可以共享该信息。相应地，由Bob执行的事务可以与事件相关联并且使用事件与其它成员共享。

[0154] 示例系统3：来自示例系统1的系统，其中事件还包括对于调试、诊断和/或其它目的有用的当前时间和/或日期。时间和/或日期可以是当计算设备(例如，Bob)创建和/或定义事件时的本地时间和/或日期。在这样的实施例中，这样的本地时间和/或日期不与其余设备同步。在其它实施例中，时间和/或日期可以(例如，在交换事件时)跨设备同步。在还有的其它实施例中，可以使用全局计时器来确定时间和/或日期。

[0155] 示例系统4：来自示例系统1的系统，其中Alice既不向Bob发送由Bob创建和/或定义的事件，也不向Bob发送这种事件的祖先事件。如果y包含x的散列，或者y包含作为x的祖先的事件的散列，则事件x是事件y的祖先。类似地，在这样的实施例中，Bob向Alice发送尚未由Alice存储的事件，并且不发送已经由Alice存储的事件。

[0156] 例如，图4示出了示例hashDAG 620，其示出事件622(黑色圆圈)的祖先事件(带虚线的圆圈)和后代事件(带条纹的圆圈)。线段建立事件的部分顺序，其中祖先出现在黑色事件之前，并且后代出现在黑色事件之后。部分顺序不指示白色事件是在黑色事件之前还是之后，因此总顺序用于决定它们的序列。对于另一示例，图5示出示例hashDAG，其示出一个特定事件(实心圆圈)以及每个成员接收该事件的指示的第一时间(带条纹的圆圈)。当Carol与Dave同步以创建和/或定义事件624时，Dave不向Carol发送事件622的祖先事件，因为Carol已经知道并且已经接收到这样的事件。相反，Dave向Carol发送Carol尚未接收和/

或存储在Carol的分布式数据库实例中的事件。在一些实施例中，Dave可以基于Dave的hashDAG揭示的关于Carol先前接收到哪些事件的内容来识别要发送给Carol的事件。事件622是事件626的祖先。因此，在事件626时，Dave已经接收到事件622。图4示出了Dave从Ed接收到事件622，Ed从Bob接收到事件622，Bob从Carol接收到事件622。此外，在事件624时，事件622是Dave接收到的、由Carol创建和/或定义的最后一个事件。因此，Dave可以向Carol发送除了事件622及其祖先之外的Dave已存储的事件。此外，在接收到来自Dave的事件626之后，Carol可以基于存储在Carol的分布式数据库实例中的事件中的指针来重新构建hashDAG。在其它实施例中，基于Carol向Dave发送事件622(图4中未示出)以及Dave使用事件622(以及其中的引用)来识别Carol已经接收到的事件，Dave可以识别要发送给Carol的事件。

[0157] 示例系统5：来自示例系统1的系统，其中两个成员以一定顺序向另一个成员发送事件，使得直到接收者接收和/或存储该事件的祖先之后才发送该事件。因此，发送者从最旧到最新来发送事件，使得接收者可以在接收到事件时通过将两个散列与已经接收到的两个祖先事件进行比较来检查每个事件的两个散列。发送者可以基于发送者的hashDAG的当前状态(例如，由发送者定义的数据库状态变量)以及该hashDAG指示接收者已经接收到的内容来识别要发送给接收者的事件。参考图3，例如，当Bob正在与Carol同步以定义事件602时，Carol可以识别事件619是Carol接收到的、由Bob创建和/或定义的最后一个事件。因此，Carol可以确定Bob知道该事件及其祖先。因此，Carol可以首先向Bob发送事件618和事件616(即，Carol接收到的、Bob尚未接收到的最旧事件)。Carol然后可以向Bob发送事件612并且然后发送事件606。这允许Bob容易地链接事件并重新构建Bob的hashDAG。使用Carol的hashDAG来识别Bob尚未接收到的事件可以提高同步的效率，并且可以减少网络流量，因为Bob不从Carol请求事件。

[0158] 在其它实施例中，可以首先发送最新近的事件。如果接收者(基于最新近事件中的两个先前事件的散列和/或最新近事件中的指向先前事件的指针)确定它们还没有接收到两个先前事件中的一个事件，则接收者可以请求发送者发送这样的事件。这可以一直发生，直到接收者已接收到和/或存储最新近事件的祖先为止。参考图3，在这样的实施例中，例如，当Bob从Carol接收到事件606时，Bob可以识别事件606中的事件612和事件614的散列。Bob在创建和/或定义事件604时可以确定事件614是先前从Alice接收到的。因此，Bob不需要从Carol请求事件614。Bob还可以确定事件612尚未被接收到。Bob然后可以从Carol请求事件612。然后Bob可以基于事件612内的散列确定Bob尚未接收到事件616或618并且可以相应地从Carol请求这些事件。基于事件616和618，Bob然后将能够确定他已接收到事件606的祖先。

[0159] 示例系统6：来自示例系统5的系统，其具有附加的限制，即，当成员可以在接下来要发送的几个事件之间进行选择时，选择使由该成员创建和/或定义的到目前为止发送的字节总数最小化的事件。例如，如果Alice只剩下两个事件要发送给Bob，并且一个是100个字节并且由Carol创建和/或定义，一个是10个字节并且由Dave创建和/或定义，并且到目前为止在这个同步中Alice已经发送了Carol的事件的200个字节和Dave的事件的210个字节，那么Alice应当首先发送Dave事件，然后随后发送Carol事件。因为210+10<100+200。这可以用于解决其中单个成员发送出单个巨大事件或者大量微小事件的攻击。在流量超过大多数

成员的字节限制的情况下(如关于示例系统7所讨论的),示例系统6的方法可以确保攻击者的事件被忽略而不是合法用户的事件被忽略。类似地,通过在较大事件之前发送较小的事件(以防止一个巨大事件占用连接),可以减少攻击。此外,如果成员不能在单个同步中发送事件中的每个事件(例如,由于网络限制、成员字节限制等),那么该成员可以发送来自每个成员的若干事件,而不是仅仅发送由攻击者定义和/或创建的事件而不发送由其它成员创建和/或定义的(很少的)事件。

[0160] 示例系统7:来自示例系统1的系统,其具有附加的第一步骤,在该第一步骤中Bob向Alice发送指示他愿意在该同步期间接收的最大字节数的数字,并且Alice用她的限制进行回复。当下一个事件将超过这个限制时,然后Alice停止发送。Bob也这样做。在这样的实施例中,这限制了传送的字节的数量。这可能会增加收敛时间,但是将减少每次同步的网络流量的量。

[0161] 示例系统8:来自示例系统1的系统,其中在同步过程开始时添加以下步骤:

[0162] -Alice识别S,即,她已经接收到和/或存储的事件集合,从而跳过由Bob创建和/或定义的事件或者作为由Bob创建和/或定义的事件的祖先的事件。

[0163] -Alice识别创建和/或定义S中的每个事件的成员,并向Bob发送成员的ID号列表。Alice还发送她已经收到和/或存储的、由每个成员创建和/或定义的若干事件。

[0164] -Bob利用他已经接收到的、由其它成员创建和/或定义的事件的数量的列表进行回复。

[0165] -Alice然后只向Bob发送他尚未接收到的事件。例如,如果Alice向Bob指示她已接收到由Carol创建和/或定义的100个事件,并且Bob回复他已经接收到由Carol创建和/或定义的95个事件,则Alice将仅发送由Carol创建和/或定义的最近5个事件。

[0166] 示例系统9:来自示例系统1的系统,具有用于识别和/或处理作弊者的附加机制。每个事件包含两个散列,一个散列来自由该成员创建和/或定义的最后一个事件(“自散列”),另一个散列来自由另一个成员创建和/或定义的最后一个事件(“外散列”)。如果成员创建和/或定义具有相同自散列的两个不同事件,那么该成员是“作弊者”。如果Alice通过接收由Dave创建和/或定义的具有相同自散列的两个不同事件而发现Dave是作弊者,则她存储他是作弊者的指示符,并且避免将来与他同步。如果她发现他是作弊者但仍然再次与他同步并创建和/或定义记录该事实的新事件,那么Alice也成为作弊者,并且知道Alice还与Dave同步的其它成员停止与Alice同步。在一些实施例中,这仅在某种程度上影响同步。例如,当Alice发送她接收到的针对每个成员的事件数量以及标识符列表时,她不发送作弊者的ID或计数,因此Bob将不会利用任何对应的数字进行回复。然后,Alice向Bob发送她接收到的并且她没有接收到Bob已接收到这种事件的指示的作弊者的事件。在同步完成之后,Bob也将能够确定Dave是作弊者(如果他还没有将Dave识别为作弊者),并且Bob也将拒绝与作弊者同步。

[0167] 示例系统10:示例系统9中的系统,其中添加Alice通过向Bob发送她已识别出的并且她仍存储其事件的作弊者列表来开始同步过程,并且Bob利用除了Alice识别出的作弊者之外的、他已识别出的任何作弊者进行回复。然后他们照常继续,但是在彼此同步时不考虑作弊者。

[0168] 示例系统11:示例系统1中的系统,其具有基于在同步期间接收到的任何新事件内

的事务来重复地更新(例如,由系统的成员定义的数据库状态变量所捕获的)当前状态的过程。这还可以包括第二过程,该第二过程每当事件的序列改变时,通过返回到先前状态的副本并且通过按新顺序处理事件来重新计算目前状态从而重复地再建该状态(例如,事件的顺序)。在一些实施例中,当前状态是与事务的结果相关联的状态、余额、条件等。类似地,状态可以包括由事务修改的数据结构和/或变量。例如,如果事务是银行账户之间的汇款,那么当前状态可以是账户的当前余额。又例如,如果事务与多人游戏相关联,则当前状态可以是与游戏相关联的位置、生命的数量、获得的物品、游戏的状态等等。

[0169] 示例系统12:示例系统11中的系统,通过使用"快速克隆"arrayList(数组列表)来维持状态(例如,银行账户余额、游戏状态等)以使该系统更快。快速克隆arrayList是一种数据结构,它像数组一样工作,但具有一个附加特征:它支持"克隆"操作,该"克隆"操作看起来像是创建和/或定义作为原件的副本的新对象。克隆物就好像它是真正的副本一样,因为对克隆的改变不会影响原件。但是,克隆操作比创建真正副本更快,因为创建克隆物(clone)实际上不涉及将一个arrayList的全部内容复制和/或更新到另一个arrayList。可以使用各自具有散列表和指向原始列表的指针的两个小对象,而不是具有原始列表的两个克隆物和/或副本。当对克隆物进行写入时,散列表记住哪个元素被修改以及新的值。在对某个位置执行读取时,首先检查散列表,并且如果该元素被修改,则返回散列表中的新值。否则,返回原始arrayList中的该元素。以这种方式,两个"克隆物"初始地仅仅是指向原始arrayList的指针。但是,随着每个都被重复地修改,它长成为具有存储自身与原始列表之间的差异的大的散列表。克隆物本身可以被克隆,从而导致数据结构扩展为对象树,其中每个对象具有它自己的散列表和指向它的父亲的指针。因此,读取会导致在树上向上行走,直到找到具有所请求的数据的顶点或到达根为止。如果顶点变得太大或太复杂,那么可以用父亲的真实副本来替代该顶点,散列表中的变化可以对副本进行,并且散列表被丢弃。此外,如果不再需要克隆物,那么在垃圾收集期间,它可以从树中去除,并且树可以被收缩。

[0170] 示例系统13:示例系统11中的系统,通过使用"快速克隆"散列表来维持状态(例如,银行账户余额、游戏状态等)使该系统更快。除了树的根是散列表而不是arrayList之外,这与系统12相同。

[0171] 示例系统14:示例系统11中的系统,通过使用"快速克隆"关系数据库来维持状态(例如,银行账户余额、游戏状态等)使该系统更快。这是充当现有关系数据库管理系统(RDBMS)的包装器的对象。每个明显的"克隆物"实际上是具有ID号的对象和指向包含数据库的对象的指针。当用户的代码尝试对数据库执行结构查询语言(SQL)查询时,该查询首先被修改,然后被发送到真正的数据库。除了每个表具有用于克隆ID的一个附加字段之外,真正的数据库与客户端代码看到的数据库相同。例如,假设存在具有克隆ID 1的原始数据库,并且然后做出数据库的两个克隆物,具有ID 2和3。每个表中的每一行将在克隆ID字段中具有1、2或3。当来自用户代码的查询针对克隆物2时,查询被修改,使得查询将只能从该字段中具有2或1的行中读取。类似地,对3的读取查找具有3或1ID的行。如果结构化查询语言(SQL)命令转到克隆物2并且声称要删除一行,并且该行具有1,那么该命令应当仅将1改变为3,这将把该行标记为不再由克隆物2和3共享,并且现在只能由3看到。如果在操作中存在若干克隆物,那么可以插入行的若干副本,并且每个副本可以被改变为不同克隆物的ID,使得新行对除了刚刚"删除"该行的克隆物以外的其它克隆物可见。类似地,如果向克隆物2添

加一行，则该行被添加到具有ID为2的表。行的修改相当于删除然后插入。和之前一样，如果若干克隆物被垃圾收集，那么树可以被简化。该树的结构将被存储在克隆物不能访问的附加表中，但是纯粹在内部使用。

[0172] 示例系统15：示例系统11中的系统，通过使用“快速克隆”文件系统来维持状态而使该系统更快。这是充当文件系统的包装器的对象。文件系统建立在现有文件系统之上，使用快速克隆关系数据库来管理不同版本的文件系统。底层文件系统将大量文件存储在一个目录中，或者根据文件名进行划分(以保持目录小)。目录树可以存储在数据库中，并且不提供给主机文件系统。当文件或目录被克隆时，“克隆物”只是带有ID号的对象，并且数据库被修改以反映该克隆物现在存在。如果快速克隆文件系统被克隆，则对于用户看起来像是整个新的硬盘驱动器已被创建和/或定义，并利用现有硬盘驱动器的副本进行了初始化。对一个副本的改变可以对其它副本没有影响。实际上，仅存在每个文件或目录的一个副本，并且当通过一个克隆物修改文件时，会发生复制。

[0173] 示例系统16：示例系统15中的系统，其中为快速克隆文件系统中的文件的每个N字节部分在主机操作系统上创建和/或定义单独的文件。N可以是某个合适的大小，诸如例如4096或1024。以这种方式，如果一个字节在大文件中被改变，那么该大文件的仅一个块被复制和修改。这还提高了当在驱动器上存储仅有几个字节不同的许多文件时的效率。

[0174] 示例系统17：示例系统11中的系统，其中每个成员在他们创建和/或定义的事件中的一些或全部中包括在某个先前时间的状态的散列，以及直到该点发生的事件的数量，从而指示该成员认识到和/或识别出现在存在关于事件的顺序的共识。在成员收集到包含来自多数用户的针对给定状态的这种散列的已签名事件之后，该成员然后可以将其存储为在该点时的共识状态证明，并从存储器中删除在该点之前的事件和事务。

[0175] 示例系统18：示例系统1中的系统，其中计算中值或多数票的操作被替换为加权中值或加权多数票，其中成员通过它们的“权益”进行加权。权益是指示该成员的投票计数多少的数字。权益可以是以加密货币形式的持有股份，或只是当成员首次被邀请加入时被指派的并且然后在成员邀请加入的新成员之间进行划分的任意数字。当足够的成员已经同意共识状态时，旧的事件可以被丢弃，使得它们的总权益是现有权益的多数。如果总顺序是使用成员贡献的排名的中值计算的，那么结果是其中一半的成员具有较高的排名并且一半的成员具有较低的排名的数字。另一方面，如果总顺序是使用加权中值计算的，那么结果是其中大约一半的总权益与低于该加权中值的排名相关联并且大约一半的总权益与高于该加权中值的排名相关联的数字。加权投票和中值在防止Sybil攻击时可以是有用的，在Sybil攻击中，一个成员邀请大量的“马甲”用户加入，这些“马甲”用户中的每一个只是由邀请成员控制的假名。如果邀请成员被迫与受邀者划分其权益，那么马甲将对企图控制共识结果的攻击者而言是没有用的。因此，在一些情况下，权益证明可能是有用的。

[0176] 示例系统19：示例系统1中的系统，其中作为单个分布式数据库的代替，在层次结构中存在多个数据库。例如，可能存在用户是其成员的单个数据库，然后是若干较小的数据库或“块”，这些较小的数据库或“块”中的每个都具有成员的子集。当事件在块中发生时，它们在该块的成员之间同步，并且不在该块之外的成员之间同步。然后，不时地，在块内已经决定共识顺序之后，可以与大数据库的全部成员共享结果得到的状态(或具有其共识总顺序的事件)。

[0177] 示例系统20:示例系统11中的系统,其有能力具有更新用于更新(例如,如由系统的成员定义的数据库状态变量所捕获的)状态的软件的事件。例如,根据读取这些事件内的事务的软件代码,事件X和Y可以包含修改状态的事务,并且然后适当地更新状态。然后,事件Z可以包含软件的新版本现在可用的通知。如果总顺序声称事件以顺序X、Z、Y发生,那么可以通过用旧软件处理X中的事务,然后用新软件处理Y中的事务来更新状态。但是如果共识顺序是X、Y、Z,那么X和Y两者都可以用旧软件来更新,这可能会给出不同的最终状态。因此,在这样的实施例中,升级代码的通知可以在事件内发生,使得社区可以关于何时从旧版本切换到新版本达成共识。这确保了成员将维持同步的状态。它还确保了系统即使在升级期间也可以保持运行,而无需重新引导或重新启动过程。

[0178] 示例系统21:来自示例系统1的系统,其中权益证明协议被实现以达成共识并且每个成员的投票权与成员拥有的加密货币的量成比例。这个示例的加密货币将在下文中被称为StakeCoin(权益币)。组或群体的成员资格是开放的,不是许可的,因此,成员之间可能没有信任。

[0179] 与其它协议(例如,工作证明协议)相比,权益证明协议可以在计算上比较便宜。在这个示例中,M(如上所述)可以是由成员集体拥有的StakeCoin量的2/3。因此,如果攻击者不能获得由参与成员加在一起拥有的总StakeCoin的1/3,则分布式数据库系统可以是安全的(并且可以按预期收敛)。只要超过2/3的StakeCoin由诚实的活跃成员拥有,分布式数据库系统就可以在数学上得到保证的安全级别下保持运转。这允许数据库正确地收敛。

[0180] 攻击者获得对分布式数据库系统的控制的方式可以通过与分布式数据库系统内的StakeCoin拥有者单独地协商以购买他们的StakeCoin来实现。例如,Alice可以通过购买由Bob、Carol和Dave所拥有的StakeCoin来获得大多数StakeCoin,从而使Ed处于弱势地位。这类似于垄断商品的市场,或试图购买一家公司的足够股份以进行恶意收购。所描述的场景不仅表示对使用StakeCoin的分布式数据库系统的攻击,还表示它是对StakeCoin本身的攻击。如果成员对于加密货币获得近似垄断,那么这个成员就可以操纵加密货币的市场价值,并进行反复地高卖低买。这在短期内可能是有收益的,但最终会破坏对加密货币的信任,并可能导致它被普遍抛弃。货币市场价值可以独立于用来传达货币的技术。例如,如果个人或实体获得世界上的大多数美元的所有权,或世界上大多数玉米期货的所有权,那么这样的个人或实体可能破坏市场进行牟利。

[0181] 如果加密货币既有价值又普遍,那么通过获取近乎垄断的加密货币进行攻击较难。如果加密货币是有价值的,那么购买大部分StakeCoin货币供应将花费巨大。如果加密货币是普遍的,并且许多不同的人拥有StakeCoin,那么试图垄断StakeCoin市场将在早期就被看到,这将自然地提高StakeCoin的价格,从而使得更难以获得剩余的货币供应。

[0182] 可以通过获得与跨多个分布式数据库系统的StakeCoin的集体量相比可能较小、但与由参与特定分布式数据库系统的成员所拥有的StakeCoin的量相比较大的StakeCoin的量来进行第二类型的攻击。当加密货币被专门定义为用在分布式数据库系统的应用中时,可以避免这种类型的攻击。换句话说,StakeCoin和分布式数据库系统的实现可以被同时定义为链接到彼此,并且每个都有助于提高另一个的价值。类似地,不存在交易StakeCoins的分布式数据库的附加实现。

[0183] 可能期望从一开始当分布式数据库系统的实现被新定义时具有有价值的加密货

币。虽然加密货币可以随着时间的推移增加其价值，但是在系统的早期阶段，有价值的加密货币可能是有益的。在一些情况下，参与实体的联合体可以发起加密货币及其相关联的分布式数据库系统。例如，作为创始者的十个知名的大公司或组织可以被给予大量的StakeCoin，以发起StakeCoin加密货币和分布式数据库系统。系统可以被配置为使得加密货币的供应将不会迅速增长，并且将具有一些最终的大小限制。每个创始实体可以具有作为成员参与分布式数据库系统和StakeCoin的实现(例如，实现为可以利用共识算法被构造为hashDAG的分布式数据库系统)的激励。由于没有工作证明，因此成为运行节点的参与成员可能并不昂贵。创始实体可以足够可靠，以至于它们中的任何大部分将共谋破坏系统是不可能的，尤其是因为这可以破坏他们拥有的StakeCoin和实现的分布式数据库系统的价值。

[0184] 在一些实现中，其它成员可以加入分布式数据库系统，并且其它个人或实体可以直接从创始实体或者在交易所购买StakeCoin。分布式数据库系统可以被配置为通过对于参与支付少量的StakeCoin来激励成员参与。随着时间的推移，系统可以变得更加分布式，权益最终扩散开，以使得任何个人或实体，甚至即使是创始实体合谋进行攻击，也难以垄断市场。那时，加密货币可以达到独立的价值；分布式数据库系统可以具有独立的安全级别；并且系统可以是开放的而没有许可要求(例如，不需要必须被创始成员邀请才能加入)。因此，节省了用替代协议实现的系统(例如，用工作证明协议实现的系统)所需的经常性的计算成本。

[0185] 虽然以上关于使用hashDAG分布式数据库进行了描述，但是可以使用任何其它合适的分布式数据库协议来实现示例系统21。例如，虽然具体示例数字和权益可能改变，但是示例系统21可以用于增加任何合适的分布式数据库系统的安全性。

[0186] 预期上述系统将创建和/或实现用于分布式共识并具有最终共识的高效收敛机制。若干个定理可以证明这一点，如以下所示。

[0187] 示例定理1：如果事件x在部分顺序中在事件y之前，那么在给定时间在给定成员对其它成员的认识中，其它成员中的每个成员或者将接收到x在y之前的指示，或者将还没有接收到y的指示。

[0188] 证明：如果事件x在部分顺序中在事件y之前，那么x是y的祖先。当成员第一次接收到y的指示时，该成员或者已经在之前接收到x的指示(在这种情况下，他们知道x在y之前)，或者将是这种情况，即，同步为该成员提供x和y两者(在这种情况下，他们将在该同步期间知道x在y之前，因为在单次同步期间接收到的事件被认为是按照关于示例系统5所描述的、与祖先关系一致的顺序接收到的)。QED

[0189] 示例定理2：对于任何给定的hashDAG，如果x在部分顺序中在y之前，则在针对该hashDAG计算出的总顺序中x将在y之前。

[0190] 证明：如果x在部分顺序中在y之前，则通过定理1：

[0191] 对于所有的i，rank(i,x)<rank(i,y)

[0192] 其中rank(i,x)是由成员i指派给事件x的排名，如果x是成员i接收到的第一事件，则rank(i,x)为1，如果x是第二事件则rank(i,x)为2，以此类推。令med(x)为所有i上的rank(i,x)的中值，并且对于med(y)也类似。

[0193] 对于给定的k，选择i1和i2，使得rank(i1,x)是第k小(kth-smallest)的x排名，并

且rank(i2,y)是第k小的y排名。那么：

[0194] rank(i1,x)<rank(i2,y)

[0195] 这是因为rank(i2,y)大于或等于y排名中的k个，其中每一个严格地大于对应的x排名。因此，rank(i2,y)严格地大于x排名中的至少k个，因此严格大于第k小的x排名。这个论点适用于任何k。

[0196] 令n是成员的数量(这是i值的数量)。那么n必须是奇数或偶数。如果n是奇数，则令k＝(n+1)/2，并且第k小的排名将是中值。因此med(x)<med(y)。如果n是偶数，那么当k＝n/2时，第k小的x排名将严格小于第k小的y排名，并且第(k+1)小的x排名将严格小于第(k+1)小的y排名。因此两个x排名的平均值将小于两个y排名的平均值。因此，med(x)<med(y)。因此在这两种情况下，x排名的中值都严格小于y排名的中值。因此，如果总顺序通过按中值排名对动作进行排序来定义，则在总顺序中x将在y之前。QED

[0197] 示例定理3：如果“流言(gossip)时间段”是现有事件通过同步传播到所有成员的时间量，则：

[0198] 在1个流言时间段之后：所有成员都接收到事件

[0199] 在2个流言时间段之后：所有成员都同意这些事件的顺序

[0200] 在3个流言时间段之后：所有成员都知道已达成一致

[0201] 在4个流言时间段之后：所有成员都从所有其它成员获得数字签名，从而认可这个共识顺序。

[0202] 证明：令S0是由给定时间T0创建和/或定义的事件的集合。如果每个成员将最终无限经常地与每个其它成员同步，那么以概率1将最终存在时间T1，在时间T1时S0中的事件已经传播到每个成员，使得每个成员都知道所有事件。这是第一流言时间段的结束。令S1是在时间T1时存在并且在T0时尚不存在的事件集合。那么将以概率1最终存在时间T2，在时间T2处每个成员已接收到集合S1中的每个事件，其是在时间T1时存在的事件。这是第二流言时间段的结束。类似地，T3是S2中的所有事件，即到T2时存在但在T1之前不存在的事件，已传播到所有成员的时间。要注意的是，每个流言时间段最终以概率1结束。平均而言，如果存在n个成员，则每个流言时间段将持续执行log2(n)个同步所花费的时间。

[0203] 到时间T1时，每个成员将已经接收到S0中的每个事件。

[0204] 到时间T2时，给定的成员Alice将已经接收到其它成员中的每个成员接收到S0中的每个事件的记录。因此，Alice可以针对每个成员在S0中的每个动作计算排名(这是该成员接收到该动作的顺序)，并且然后按照排名的中值对事件进行排序。对于S0中的事件，得到的总顺序不会改变。这是因为得到的顺序基于每个成员第一次接收到这些事件中的每个事件的指示的顺序，其不会改变。有可能Alice计算出的顺序将把来自S1的一些事件穿插在S0事件之间。这些S1事件可能仍然改变它们落在S0事件的序列内的位置。但S0中的事件的相对顺序将不会改变。

[0205] 到时间T3时，Alice将已经知道S0和S1的联合的总顺序，并且该联合中的事件的相对顺序将不会改变。此外，她可以在这个序列内找到来自S1的最早事件，并且可以推断出S1之前的事件的序列将不会改变，甚至在S0外部插入新事件也将不会改变。因此，到时间T3时，Alice可以确定，对于在第一S1事件之前的历史中的事件的顺序已经达成共识。她可以对由以这个顺序发生的这些事件产生的(例如，由Alice定义的数据库状态变量所捕获的)

状态的散列进行数字签名，并将签名作为她创建和/或定义的下一个事件的一部分发出。

[0206] 到时间T4时，Alice将已经从其它成员接收到类似的签名。那时，她可以仅保留签名列表连同它们证实的状态，并且她可以丢弃在第一S1事件之前她已经存储的事件。QED

[0207] 本文描述的系统描述了快速且安全地实现共识的分布式数据库。这对许多应用来说可能是有用的构建块。例如，如果事务描述从一个加密货币钱包向另一个加密货币钱包转移加密货币，并且如果状态仅仅是每个钱包中的当前金额的声明，那么该系统将构成避免现有系统中昂贵的工作证明的加密货币系统。自动规则实施允许这样以添加在当前加密货币中不常见的特征。例如，可以恢复丢失的币以避免通货紧缩，这是通过实施规则：即，如果钱包在某个时间段内既没有发送也没有接收加密货币，那么该钱包被删除，并且它的值被分配给其他现有的钱包并且分配的值与他们当前包含的金额成比例。以这种方式，即使钱包的私钥丢失，货币供应量也不会增长或减少。

[0208] 另一个示例是分布式游戏，其就像在服务器上玩的大型多人在线(MMO)游戏，但是在不使用中央服务器的情况下实现该游戏。可以在没有任何中央服务器进行控制的情况下达成共识。

[0209] 另一个示例是建立在这样的数据库之上的用于社交媒体的系统。由于事务是被数字签名的，并且成员接收关于其它成员的信息，因此与当前系统相比，这提供了安全性和便利性优势。例如，由于电子邮件不能具有伪造的返回地址，因此可以实现具有强大的反垃圾邮件策略的电子邮件系统。这样的系统还可以成为统一的社交系统，在单个分布式数据库中组合当前通过电子邮件、推文、文本、论坛、维基和/或其它社交媒体完成的功能。

[0210] 其它应用可以包括更复杂的密码功能，诸如组数字签名，其中组作为整体合作以签署合约或文档。这种形式以及其它形式的多方计算可以使用这样的分布式共识系统来有效地实现。

[0211] 另一个示例是公共记帐系统。任何人都可以支付以在系统中存储一些信息，从而每年每字节支付少量的加密货币(或现实世界货币)以在系统中存储信息。然后，这些资金可以被自动分发给存储该数据的成员，以及重复同步以工作来达成共识的成员。它可以在他们每次同步时自动向成员转移少量的加密货币。

[0212] 这些示例示出分布式共识数据库作为许多应用的部件是有用的。由于数据库不使用昂贵的工作证明，而是可能使用较便宜的权益证明，因此数据库可以以运行在较小计算机或甚至移动设备和嵌入式设备上的全节点来运行。

[0213] 虽然在上文被描述为包含两个先前事件的散列(一个自散列和一个外散列)的事件，但是在其它实施例中，成员可以与两个其它成员同步以创建和/或定义包含三个先前事件的散列(一个自散列和两个外散列)的事件。在还有的其它实施例中，来自任何数量的成员的先前事件的任何数量的事件散列可以被包括在事件内。在一些实施例中，不同的事件可以包括先前事件的不同数量的散列。例如，第一事件可以包括两个事件散列，并且第二事件可以包括三个事件散列。

[0214] 虽然上文将事件描述为包括先前事件的散列(或密码散列值)，但是在其它实施例中，事件可以被创建和/或定义为包括对先前事件的指针、标识符和/或任何其它合适的引用。例如，事件可以被创建和/或定义为包括与先前事件相关联的并且用于识别先前事件的序列号，从而链接事件。在一些实施例中，这样的序列号可以包括例如与创建和/或定义事

件的成员相关联的标识符(例如,介质访问控制(MAC)地址、互联网协议(IP)地址、指派的地址等)和由该成员定义的事件的顺序。例如,具有标识符10并且事件是由其创建和/或定义的第15个事件的成员可以向该事件指派标识符1015。在其它实施例中,可以使用任何其它合适的格式来为事件指派标识符。

[0215] 在其它实施例中,事件可以包含完整的密码散列,但是在同步期间仅传送这些散列的部分。例如,如果Alice向Bob发送包含散列H的事件,并且J是H的前3个字节,并且Alice确定在她已存储的事件和散列中,H是以J开头的唯一散列,则她可以在同步期间发送J而不是H。如果Bob然后确定他具有另一个以J开头的散列,则他可以回复Alice以请求完整的H。以这种方式,散列可以在传输期间被压缩。

[0216] 虽然上文示出和描述的示例系统是参考其它系统描述的,但是在其它实施例中,可以实现示例系统及其相关联的功能的任何组合以创建和/或定义分布式数据库。例如,示例系统1、示例系统2和示例系统3可以被组合以创建和/或定义分布式数据库。又例如,在一些实施例中,示例系统10可以用示例系统1实现而不用示例系统9来实现。还例如,示例系统7可以与示例系统6组合并且用示例系统6实现。在还有的其它实施例中,可以实现示例系统的任何其它合适的组合。

[0217] 虽然在上文被描述为交换事件以获得收敛,但是在其它实施例中,分布式数据库实例可以交换值和/或值的向量以获得如关于图8-图13所描述的收敛。具体而言,例如,图8示出根据实施例的来自分布式数据库系统(例如,分布式数据库系统100)的第一计算设备400与来自分布式数据库系统(例如,分布式数据库系统100)的第二计算设备500之间的通信流。在一些实施例中,计算设备400、500可以在结构上和/或功能上类似于图2中所示的计算设备200。在一些实施例中,计算设备400和计算设备500以与关于图1所示和描述的计算设备110、120、130、140在分布式数据库系统100内彼此通信的方式类似的方式彼此通信。

[0218] 类似于关于图2描述的计算设备200,计算设备400、500可以各自初始地定义参数的值的向量,更新值的向量,基于所定义的和/或更新后的参数的值的向量来选择参数的值、以及存储(1)所定义的和/或更新后的参数的值的向量,和/或(2)基于所定义的和/或更新后的参数的值的向量而选择的参数的值。计算设备400、500中的每一个可以以任何数量的方式来初始地定义参数的值的向量。例如,计算设备400、500中的每一个可以通过将值的向量中的每个值分别设置为等于初始存储在分布式数据库实例403、503中的值来初始地定义参数的值的向量。又例如,计算设备400、500中的每一个可以通过将值的向量中的每个值设置为等于随机值来初始地定义参数的值的向量。如何初始地定义参数的值的向量可以例如由计算设备400、500所属的分布式数据库系统的管理员选择,或者由分布式数据库系统的计算设备(例如,计算设备400、500)的用户单独地或共同地选择。

[0219] 计算设备400、500还可以各自分别在分布式数据库实例403、503中存储参数的值的向量和/或参数的选定值。分布式数据库实例403、503中的每一个可以在类似于图2所示的存储器220的存储器(图8中未示出)中实现。

[0220] 在步骤1中,计算设备400从计算设备500请求存储在计算设备500的分布式数据库实例503中的参数的值(例如,存储在分布式数据库实例503的特定字段中的值)。在一些实施例中,计算设备500可以由计算设备400从属于分布式数据库系统的计算设备集合中选择。计算设备500可以被随机选择、基于与计算设备400的关系来选择、基于与计算设备400

的接近度来选择、基于与计算设备400相关联的有序列表来选择，等等。在一些实施例中，由于计算设备500可以由计算设备400从属于分布式数据库系统的计算设备集合中选择，因此计算设备400可以连续多次选择计算设备500或者可以在一段时间内不选择计算设备500。在其它实施例中，先前选择的计算设备的指示可以被存储在计算设备400处。在这样的实施例中，计算设备400可以在能够再次选择计算设备500之前等待预定数量的选择。如上所述，分布式数据库实例503可以在计算设备500的存储器中实现。

[0221] 在一些实施例中，来自计算设备400的请求可以是由计算设备400的通信模块(图8中未示出)发送的信号。该信号可以由网络(诸如网络105(图1中示出))承载并由计算设备500的通信模块接收。在一些实施例中，计算设备400、500的通信模块中的每一个可以在处理器或存储器内实现。例如，计算设备400、500的通信模块可以类似于图2所示的通信模块212。

[0222] 在从计算设备400接收到对存储在分布式数据库实例503中的参数的值的请求之后，计算设备500在步骤2中将存储在分布式数据库实例503中的参数的值发送到计算设备400。在一些实施例中，计算设备500可以从存储器中检索参数的值，并且通过计算设备500的通信模块(图8中未示出)将该值作为信号发送。在一些情况下，如果分布式数据库实例503尚未包括参数的值(例如，事务尚未在分布式数据库实例503中被定义)，则分布式数据库实例503可以从计算设备403请求参数的值(如果该参数的值尚未在步骤1中被提供)并且将该参数的值存储在分布式数据库实例503中。在一些实施例中，计算设备400然后将使用这个值作为在分布式数据库实例503中的参数的值。

[0223] 在步骤3中，计算设备400向计算设备500发送存储在分布式数据库实例403中的参数的值。在其它实施例中，存储在分布式数据库实例403中的参数的值(步骤1)和对存储在分布式数据库实例503中的相同参数的值的请求(步骤3)可以作为单个信号被发送。在其它实施例中，存储在分布式数据库实例403中的参数的值可以在与用于对于存储在分布式数据库实例503中的参数的值的请求的信号不同的信号中发送。在存储在分布式数据库实例403中的参数的值在与用于对于存储在分布式数据库实例503中的参数的值的请求的信号不同的信号中发送的实施例中，存储在分布式数据库实例403中的参数的值，这两个信号可以以任何顺序发送。换句话说，任一个信号都可以在另一个信号之前发送。

[0224] 在计算设备400接收到从计算设备500发送的参数的值和/或计算设备500接收到从计算设备400发送的参数的值之后，在一些实施例中，计算设备400和/或计算设备500可以分别更新存储在分布式数据库实例403中的值的向量和/或存储在分布式数据库实例503中的值的向量。例如，计算设备400、500可以更新存储在分布式数据库实例403、503中的值的向量，以分别包括由计算设备400、500接收到的参数的值。计算设备400、500还可以基于分别存储在分布式数据库实例403中的更新后的值的向量和/或存储在分布式数据库实例503中的更新后的值的向量来分别更新存储在分布式数据库实例403中的参数的值和/或存储在分布式数据库实例503中的参数的值。

[0225] 虽然这些步骤在图8中以及在以上讨论中被标记为1、2和3，但是应当理解的是，步骤1、2和3可以以任何顺序执行。例如，步骤3可以在步骤1和步骤2之前执行。此外，如本文详细描述的，计算设备400和500之间的通信不限于图3所示的步骤1、2和3。此外，在步骤1、2和3完成之后，计算设备400可以从分布式数据库系统内的计算设备集合中选择与其交换值

(类似于步骤1、2和3)的另一个计算设备。

[0226] 在一些实施例中,在计算设备400、500之间传送的数据可以包括压缩数据、加密数据、数字签名、密码校验和,等等。此外,计算设备400、500中的每一个可以将数据发送到另一个计算设备以确认由该另一个设备先前发送的数据的接收。计算设备400、500中的每一个还可以忽略已被另一个设备重复发送的数据。

[0227] 计算设备400、500中的每一个可以初始地定义参数的值的向量,并且将该参数的值的向量分别存储在分布式数据库实例403、503中。图9a-图9c示出了参数的值的向量的示例。向量可以是参数的任何值集合(例如,参数的值的一维数组、各自具有多个部分的值的数组等)。为了说明的目的,在图9a-图9c中提供了向量的三个示例。如图所示,向量410、420、430中的每一个具有特定参数的五个值。但是,应当理解的是,值的向量可以具有任意数量的值。在一些情况下,包括在值的向量中的值的数量可以由用户设置、根据情况设置、随机地设置,等等。

[0228] 参数可以是能够取不同值的任何数据对象。例如,参数可以是二进制投票,其中投票值可以为“是”或者“否”(或者二进制“1”或“0”)。如图9a所示,值的向量410是具有五个二进制投票的向量,其中值411、412、413、414、415分别为“是”、“否”、“否”、“是”和“是”。又例如,参数可以是数据元素集合。图9b示出了参数是字母集合的示例。如图所示,值的向量420具有五个四字母集合,其中值421、422、423、424、425分别为{A,B,C,D}、{A,B,C,E}、{A,B,C,F}、{A,B,F,G}和{A,B,G,H}。还例如,参数可以是数据元素的经排名的和/或有序的集合。图9c示出了参数是人的排名集合的示例。如图所示,值的向量430具有五个六人的排名集合,其中值431、432、433、434、435分别为:

[0229] (1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)、

[0230] (1.Bob,2.Alice,3.Carol,4.Dave,5.Ed,6.Frank)、

[0231] (1.Bob,2.Alice,3.Carol,4.Dave,5.Frank,6.Ed)、

[0232] (1.Alice,2.Bob,3.Carol,4.Ed,5.Dave,6.Frank)以及

[0233] (1.Alice,2.Bob,3.Ed,4.Carol,5.Dave,6.Frank)。

[0234] 在定义参数的值的向量之后,计算设备400、500中的每一个可以基于参数的值的向量来选择参数的值。该选择可以根据任何方法和/或过程(例如,规则或规则集合)来执行。例如,选择可以根据“多数规则”来执行,其中参数的值被选择为在向量中包括的值的50%以上中出现的值。举例说明,(图9a中所示的)值的向量410包括三个“是”值和两个“否”值。在“多数规则”下,基于值的向量为参数选择的值将为“是”,因为“是”出现在(值的向量410的)值411、412、413、414、415的50%以上中。

[0235] 又例如,可以根据“多数出现”来执行选择,其中参数的值被选择为数据元素集合,其中每个数据元素在向量中包括的值的50%以上中出现。使用图9b举例说明,数据元素“A”、“B”和“C”在值的向量420的值421、422、423、424、425的50%以上中出现。在“多数出现”规则下,基于值的向量为参数选择的值将是{A,B,

[0236] C},因为只有这些数据元素(即,“A”、“B”和“C”)在值的向量420的五个值中的三个中出现。

[0237] 还例如,可以根据“按中值排名”来执行选择,其中参数的值被选择为数据元素的排名集合(例如,值的向量的值内的不同数据值),每个数据元素的排名等于该数据元素跨

向量中包括的所有值的中值排名。举例来说，图9c中的每个数据元素的中值排名如下计算：

[0238] Alice： (1,2,2,1,1)； 中值排名＝1；

[0239] Bob： (2,1,1,2,2)； 中值排名＝2；

[0240] Carol： (3,3,3,3,4)； 中值排名＝3；

[0241] Dave： (4,4,4,5,5)； 中值排名＝4；

[0242] Ed： (5,5,6,4,3)； 中值排名＝5；

[0243] Frank： (6,6,5,6,6)； 中值排名＝6。

[0244] 因此，在“按中值排名”的规则下，基于值的向量430计算出的数据元素的排名集合的值将是(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)。在一些实施例中，如果两个或更多个数据元素具有相同的中值(例如，平局)，则顺序可以通过任何合适的方法(例如，随机地、根据排名的第一指示、排名的最后指示、按字母和/或数字等)来确定。

[0245] 对于附加的示例，可以根据“Kemeny Young投票”来执行选择，其中参数的值被选择为数据元素的排名集合，该排名被计算为使得成本值最小化。例如，对于五个值的向量中的总共三个，在值的向量431、434、435中，Alice排名在Bob之前。对于五个值的向量中的总共两个，在值的向量432和433中，Bob排名在Alice之前。对于将Alice排名在Bob之前的成本值是2/5，并且对于将Bob排名在Alice之前的成本值是3/5。因此，对于Alice在Bob之前的成本值较低，并且依据“Kemeny Young投票”，Alice将排名在Bob之前。

[0246] 应当理解的是，“多数规则”、“多数出现”、“按中值排名”和“Kemeny Young投票”是作为可以用于基于参数的值的向量来选择参数的值的方法和/或过程的示例来讨论的。还可以使用任何其它方法和/或过程。例如，参数的值可以被选择为在向量中包括的值的x％以上中出现的值，其中x％可以是任何百分比(即，不限于“多数规则”中使用的50％)。百分比(即，x％)还可以因在不同时间执行的选择，例如，相对于(本文详细讨论的)置信度值而变化。

[0247] 在一些实施例中，由于计算设备可以随机选择与其交换值的其它计算设备，因此计算设备的值的向量可以在任何一个时间包括来自另一个单个计算设备的多个值。例如，如果向量大小为五，则计算设备可能在最后五次值交换迭代中两次随机选择了另一个计算设备。因此，存储在另一个计算设备的分布式数据库实例中的值将被包括在用于请求计算设备的五个值的向量中两次。

[0248] 作为示例，图10a-图10d一起示出了当一个计算设备与另一个计算设备通信时可以如何更新值的向量。例如，计算设备400可以初始地定义值的向量510。在一些实施例中，可以基于存储在计算设备400处的分布式数据库实例403中的参数的值来定义值的向量510。例如，当第一次定义值的向量510时，值的向量510中的每个值(即，值511、512、513、514、515中的每一个)可以被设置为等于存储在分布式数据库实例403中的参数的值。举例来说，如果在定义值的向量510时存储在分布式数据库实例403中的参数的值为“是”，则来自值的向量510中的每个值(即，值511、512、513、514、515中的每一个)将被设置为“是”，如图10a所示。当计算设备400接收到存储在另一个计算设备的分布式数据库的实例(例如，计算设备500的分布式数据库实例504)中的参数的值时，计算设备400可以更新值的向量510以包括存储在分布式数据库实例504中的参数的值。在一些情况下，值的向量510可以根据先入先出(FIFO)来更新。例如，如果计算设备400接收到值516(“是”)，则计算设备400可以

将值516添加到值的向量510并从值的向量510中删除值511，以定义值的向量520，如图10b所示。例如，如果在以后的时间计算设备接收到值517、518，则计算设备400可以分别将值517、518添加到值的向量510并从值的向量510中删除值512、513以分别定义值的向量530、540。在其它情况下，值的向量510可以根据除了先入先出之外的其它方案(诸如，后进先出(LIFO))来更新。

[0249] 在计算设备400更新值的向量510以定义值的向量520、530和/或540之后，计算设备400可以基于值的向量520、530和/或540来选择参数的值。该选择可以根据如以上关于图9a-图9c所讨论的任何方法和/或过程(例如，规则或规则集合)来执行。

[0250] 在一些情况下，计算设备400、500可以属于存储与涉及金融工具的事务有关的信息的分布式数据库系统。例如，计算设备400、500中的每一个可以存储关于特定股票是否可用于购买(“参数”的示例)的二进制投票(“值”的示例)。例如，计算设备400的分布式数据库实例403可以存储值“是”，指示特定股票确实可用于购买。另一方面，计算设备500的分布式数据库实例503可以存储值“否”，指示特定股票不可用于购买。在一些情况下，计算设备400可以基于存储在分布式数据库实例403中的二进制投票来初始地定义二进制投票的向量。例如，计算设备400可以将二进制投票的向量内的每个二进制投票设置为等于存储在分布式数据库实例403中的二进制投票。在这种情况下，计算设备400可以定义类似于值的向量510的二进制投票的向量。在某个以后的时间，计算设备400可以与计算设备500通信，从而请求计算设备500发送它关于特定股票是否可用于购买的二进制投票。一旦计算设备400接收到计算设备500的二进制投票(在这个示例中为“否”，其指示特定股票不可用于购买)，计算设备400就可以更新它的二进制投票的向量。例如，更新后的二进制投票的向量可以类似于值的向量520。这可以无限次地发生直到置信度值满足(本文进一步详细描述的)预定标准为止、周期性地发生，等等。

[0251] 图11示出了根据实施例的示出由分布式数据库系统100内的计算设备110执行的步骤的流程图10。在步骤11中，计算设备110基于存储在分布式数据库实例113中的参数的值来定义参数的值的向量。在一些实施例中，计算设备110可以基于存储在分布式数据库实例113中的参数的值来定义参数的值的向量。在步骤12中，计算设备110选择分布式数据库系统110内的另一个计算设备，并从所选择的计算设备请求存储在所选择的计算设备的分布式数据库实例中的参数的值。例如，计算设备110可以从计算设备120、130、140中随机选择计算设备120，并且从计算设备120请求存储在分布式数据库实例123中的参数的值。在步骤13中，计算设备110(1)从所选择的计算设备(例如，计算设备120)接收存储在所选择的计算设备的分布式数据库实例(例如，分布式数据库实例123)中的参数的值，并且(2)向所选择的计算设备(例如，计算设备120)发送存储在分布式数据库实例113中的参数的值。在步骤14中，计算设备110在参数的值的向量中存储从所选择的计算设备(例如，计算设备120)接收到的参数的值。在步骤15中，计算设备110基于参数的值的向量来选择参数的值。如以上关于图9a-图9c所讨论的，该选择可以根据任何方法和/或过程(例如，规则或规则集合)来执行。在一些实施例中，计算设备110可以在不同时间重复对参数的值的选择。计算设备110还可以在参数的值的每次选择之间重复地循环步骤12至14。

[0252] 在一些情况下，分布式数据库系统100可以存储与大型多人游戏(MMG)内的事务有关的信息。例如，属于分布式数据库系统100的每个计算设备可以按特定物品被拥有(“参

数”的示例)的顺序来存储玩家的排名集合(“值”的示例)。例如,计算设备110的分布式数据库实例114可以存储类似于值431的玩家的排名集合(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank),其指示特定物品的所有权从Alice开始,然后传给Bob,然后传给Carol,然后传给Dave,然后传给Ed,并且最后传给Frank。计算设备120的分布式数据库实例124可以存储与值432类似的玩家的排名集合的值:(1.Bob,2.Alice,3.Carol,4.Dave,5.Ed,6.Frank);计算设备130的分布式数据库实例134可以存储类似于值433的玩家的排名集合的值:(1.Bob,2.Alice,3.Carol,4.Dave,5.Frank,6.Ed);计算设备140的分布式数据库实例144可以存储类似于值434的玩家的排名集合的值:(1.Alice,2.Bob,3.Carol,4.Ed,5.Dave,6.Frank);第五计算设备(图1中未示出)的分布式数据库实例可以存储类似于值435的玩家的排名集合的值:(1.Alice,2.Bob,3.Ed,4.Carol,5.Dave,6.Frank)。

[0253] 在计算设备110定义玩家的排名集合的向量之后,计算设备可以从分布式数据库系统100的其它计算设备接收玩家的排名集合的值。例如,计算设备110可以从计算设备120接收(1.Bob,2.Alice,3.Carol,4.Dave,5.Ed,6.Frank);从计算设备130接收(1.Bob,2.Alice,3.Carol,4.Dave,5.Frank,6.Ed);从计算设备140接收(1.Alice,2.Bob,3.Carol,4.Ed,5.Dave,6.Frank);以及从第五计算设备(图1中未示出)接收(1.Alice,2.Bob,3.Ed,4.Collar,5.Dave,6.Frank)。当计算设备110从其它计算设备接收到玩家的排名集合的值时,计算设备110可以更新它的玩家的排名集合的向量以包括从其它计算设备接收到的玩家的排名集合的值。例如,在接收到以上列出的排名集合的值之后,存储在计算设备110的分布式数据库实例114中的玩家的排名集合的向量可以被更新为与值的向量430类似。在玩家的排名集合的向量已被更新为与值的向量430类似之后,计算设备110可以基于玩家的排名集合的向量来选择玩家的排名集合。例如,如以上关于图9a-图9c所讨论的,该选择可以根据“按中值排名”来执行。依据“按中值排名”,计算设备110将基于类似于值的向量430的玩家的排名集合的向量而选择(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)。

[0254] 在一些情况下,计算设备110不从另一个计算设备接收整个值。在一些情况下,计算设备110可以接收与整个值的部分(也称为复合值)相关联的标识符(诸如密码散列值),而不是部分本身。举例来说,计算设备110在一些情况下不从计算设备140接收(1.Alice,2.Bob,3.Carol,4.Ed,5.Dave,6.Frank),即整个值434,而是从计算设备140仅接收(4.Ed,5.Dave,6.Frank)。换句话说,计算设备110不从计算设备140接收(1.Alice,2.Bob,3.Carol),即,值434的某些部分。替代地,计算设备110可以从计算设备140接收与值434的这些部分(即,(1.Alice,2.Bob,3.Carol))相关联的密码散列值。

[0255] 密码散列值唯一地表示它所关联的值的部分。例如,表示(1.Alice,2.Bob,3.Carol)的密码散列将与表示以下各项的密码散列不同:

[0256] (1.Alice);

[0257] (2.Bob);

[0258] (3.Carol);

[0259] (1.Alice,2.Bob);

[0260] (2.Bob,3.Carol);

[0261] (1.Bob,2.Alice,3.Carol);

[0262] (1.Carol,2.Bob,3.Alice);

[0263] 等等。

[0264] 在计算设备110从计算设备140接收到与值434的某些部分相关联的密码散列值之后，计算设备110可以(1)使用存储在分布式数据库实例113中的值431的相同部分来生成密码散列值，以及(2)将生成的密码散列值与接收到的密码散列值进行比较。

[0265] 例如，计算设备110可以从计算设备140接收与如下由斜体指示的值434的某些部分相关联的密码散列值：(1.Alice,2.Bob,3.Colol,4.Ed,5.Dave,6.Frank)。然后计算设备可以使用如下由斜体指示的(存储在分布式数据库实例113中的)值431的相同部分来生成密码散列值：(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)。由于值434的斜体部分和值431的斜体部分是相同的，因此(与值434的斜体部分相关联的)接收到的密码散列值也将与(与值431的斜体部分相关联的)生成的密码散列值相同。

[0266] 通过将生成的密码散列值与接收到的密码散列值进行比较，计算设备110可以确定是否从计算设备140请求与接收到的密码散列值相关联的实际部分。如果生成的密码散列值与接收到的密码散列值相同，则计算设备110可以确定与接收到的密码散列值所关联的实际部分相同的副本已经存储在分布式数据库实例113中，并且因此不需要来自计算设备140的、与接收到的密码散列值相关联的实际部分。另一方面，如果生成的密码散列值与接收到的密码散列值不相同，则计算设备110可以从计算设备140请求与接收到的密码散列值相关联的实际部分。

[0267] 虽然以上讨论的密码散列值与单个值的部分相关联，但是应当理解的是，密码散列值可以与整个单个值和/或多个值相关联。例如，在一些实施例中，计算设备(例如，计算设备140)可以在它的分布式数据库实例(例如，分布式数据库实例144)中存储值集合。在这样的实施例中，在自从值已在数据库实例中被更新以来的预定时间段之后、在值的置信度值(关于图13讨论)满足预定标准(例如，达到预定阈值)之后、在自从事务发起以来的指定时间量之后和/或基于任何其它合适的因素，当从另一个数据库实例请求数据以及向另一个数据库实例发送数据时，该值可以与其它值一起被包括在密码散列值中。这减少了在数据库实例之间发送的特定值的数量。

[0268] 在一些情况下，例如，数据库中的值集合可以包括：包括在2000年和2010年之间的事务的第一值集合；包括在2010年和2013年之间的事务的第二值集合；包括在2013年和2014年之间的事务的第三值集合；以及包括在2014年和现在之间的事务的第四值集合。使用这个示例，如果计算设备110从计算设备140请求存储在计算设备140的分布式数据库实例144中的数据，则在一些实施例中，计算设备140可以向计算设备110发送(1)与第一值集合相关联的第一密码散列值，(2)与第二值集合相关联的第二密码散列值，(3)与第三值集合相关联的第三密码散列值；以及(4)来自第四值集合的每个值。关于何时将值添加到密码散列的标准可以由管理员、各个用户、基于数据库实例中已有的许多值等来设置。发送密码散列值而不是每个单独的值减少了在数据库实例之间交换值时提供的单独的值的数量。

[0269] 当接收计算设备(例如，图8的步骤2中的计算设备400)接收到(例如，由计算设备500基于分布式数据库实例503中的值生成的)密码散列值时，该计算设备使用相同的方法和/或过程以及其数据库实例(例如，分布式数据库实例403)中的用于生成接收到的密码散列值的参数(例如，在指定时间段期间的事务)的值来生成密码散列值。接收计算设备然后可以将接收到的密码散列值与生成的密码散列值进行比较。如果这些值不匹配，则接收计

算设备可以从发送计算设备(例如,图8中的计算设备500)请求用于生成接收到的密码散列的各个值,并将来自发送数据库实例(例如,分布式数据库实例503)的各个值与接收数据库实例(例如,分布式数据库实例403)中的那些事务的各个值进行比较。

[0270] 例如,如果接收计算设备接收到与2000年和2010年之间的事务相关联的密码散列值,则接收计算设备可以使用存储在其数据库实例中的2000年和2010年之间的事务的值来生成密码散列。如果接收到的密码散列值与本地生成的密码散列值匹配,则接收计算设备可以假设用于2000年和2010年之间的事务的值在这两个数据库中是相同的,并且不请求附加信息。但是,如果接收到的密码散列值与本地生成的密码散列值不匹配,则接收计算设备可以向发送计算设备请求用于生成接收到的密码散列值的单独的值。接收计算设备然后可以识别差异并更新用于该单独的值的值的向量。

[0271] 密码散列值可以依赖于任何合适的过程和/或散列函数来将多个值和/或值的部分组合成单个标识符。例如,可以使用任何合适数量的值(例如,时间段内的事务)作为散列函数的输入,并且可以基于散列函数来生成散列值。

[0272] 虽然以上讨论使用密码散列值作为与值和/或值的部分相关联的标识符,但是应当理解的是,可以使用用于表示多个值和/或值的部分的其它标识符。其它标识符的示例包括数字指纹、校验和、常规散列值等。

[0273] 图12示出了根据实施例的示出由分布式数据库系统100内的计算设备110执行的步骤的流程图(流程图20)。在图12所示的实施例中,基于预定义的概率来重置值的向量。类似地,值的向量中的每个值可以不时地并且基于概率被重置为某个值。在步骤21中,计算设备110基于参数的值的向量来选择参数的值,类似于图11中所示和以上讨论的步骤15。在步骤22中,计算设备110从其它计算设备(例如计算设备120、130、140)接收参数的值,并将存储在分布式数据库实例113中的参数的值发送给其它计算设备(例如,计算设备120、130、140)。例如,步骤22可以包括针对其它计算设备中的每一个执行图11中所示和以上讨论的步骤12和13。在步骤23中,计算设备110将从其它计算设备(例如,计算设备120、130、140)接收到的参数的值存储在参数的值的向量中,类似于图11中所示和以上讨论的步骤14。在步骤24中,计算设备110基于预定义的重置值的向量的概率来确定是否重置值的向量。在一些情况下,例如,在计算设备110每次更新存储在分布式数据库实例114中的参数的值的向量之后,计算设备110将重置参数的值的向量的概率为10%。在这种情况下,在步骤24,计算设备110将基于10%的概率来确定是否重置。在一些情况下,该确定可以由计算设备110的处理器111执行。

[0274] 如果计算设备110基于预定义的概率确定重置值的向量,则计算设备110在步骤25处重置值的向量。在一些实施例中,计算设备110可以将参数的值的向量中的每个值重置为等于在重置时存储在分布式数据库实例113中的参数的值。例如,如果就在重置之前,值的向量是值的向量430,并且存储在分布式数据库实例113中的参数的值是(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)(例如,依据“按中值排名”),则值的向量中的每个值将被重置为等于(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)。换句话说,值的向量430中的值431、432、433、434、435中的每一个将被重置为等于值431。不时地并且基于概率将参数的值的向量中的每个值重置为等于在重置时存储在分布式数据库实例中的参数的值有助于(计算设备所属的)分布式数据库系统达成共识。类似地,重置促进在分布式数据库系统

的计算设备之间就参数的值达成一致。

[0275] 例如，计算设备110的分布式数据库实例114可以存储类似于值431的玩家的排名集合(1.Alice,2.Bob,3.Carol,4.Dave,5.Ed,6.Frank)，其指示特定物品的所有权从Alice开始，然后传给Bob，然后传给Carol，然后传给Dave，然后传给Ed，并且最后传给Frank。

[0276] 图13示出了根据实施例的示出由分布式数据库系统100内的计算设备110执行的步骤的流程图(流程图30)。在图13所示的实施例中，当与分布式数据库的实例相关联的置信度值为零时，基于参数的值的向量来选择参数的值发生。置信度值可以指示存储在计算设备110中的参数的值与存储在分布式数据库系统100的其它计算设备(例如，计算设备120、130、140)中的参数的值之间的“共识”或一致性的级别。在一些实施例中，如本文详细描述的，每当由计算设备110从另一个计算设备接收到的参数的值等于存储在计算设备110中的参数的值时，置信度值就递增(例如，增加1)，并且每当由计算设备110从另一个计算设备接收到的参数的值不等于存储在计算设备110中的参数的值时，如果置信度值在零以上，那么置信度值就递减(即，减小1)。

[0277] 在步骤31中，计算设备110从另一个计算设备(例如，计算设备120)接收参数的值，并将存储在分布式数据库实例113中的参数的值发送到该另一个计算设备(例如，计算设备120)。例如，步骤31可以包括执行图11中所示和以上讨论的步骤12和13。在步骤32中，计算设备110将从该另一个计算设备(例如，计算设备120)接收到的参数的值存储在参数的值的向量中，类似于图11中所示和以上所讨论的步骤14。在步骤33中，计算设备110确定从该另一个计算设备(例如，计算设备120)接收到的参数的值是否等于存储在分布式数据库实例113中的参数的值。如果从该另一个计算设备(例如，计算设备120)接收到的参数的值等于存储在分布式数据库实例113中的参数的值，则计算设备110在步骤34处将与分布式数据库实例113相关联的置信度值递增1，并且流程图30所示的过程循环回到步骤31。如果从该另一个计算设备(例如，计算设备120)接收到的参数的值不等于存储在分布式数据库实例113中的参数的值，那么如果置信度值大于零，则计算设备110在步骤35处将与分布式数据库实例113相关联的置信度值递减1。

[0278] 在步骤36处，计算设备110确定与分布式数据库实例113相关联的置信度值是否等于零。如果置信度值等于零，则计算设备在步骤37处基于参数的值的向量来选择参数的值。如以上所讨论的，该选择可以根据任何方法和/或过程(例如，规则或规则集合)来执行。如果置信度值不等于零，则流程图30所示的过程循环回到步骤31。

[0279] 如以上所讨论的，置信度值与分布式数据库实例相关联。但是，应当理解的是，作为与分布式数据库实例相关联的替代，或者除了与分布式数据库实例相关联之外，置信值还可以与存储在分布式数据库实例和/或(例如，在其分布式数据库实例内)存储向量的值的计算设备中的向量的值相关联。

[0280] 关于图13使用的与置信度值有关的值(例如，阈值、增量值和减量值)仅用于说明性目的。应当理解的是，可以使用与置信度值有关的其它值(例如，阈值、增量值和减量值)。例如，分别在步骤34和35中使用的对置信度值的增加和/或减少可以是任何值。又例如，在步骤35和36中使用的置信度阈值零还可以是任何值。此外，与置信度值相关的值(例如，阈值、增量值和减量值)可以在操作过程期间改变，即，如流程图30所示的过程那样循环。

[0281] 在一些实施例中，置信度值可以影响分布式数据库系统中的第一计算设备与分布

式数据库系统中的第二计算设备之间的通信流，如以上关于图8所描述的。例如，如果第一计算设备(例如，计算设备110)具有与其分布式数据库实例(例如，分布式数据库实例114)相关联的高置信度值，则第一计算设备可以从第二计算设备请求比第一计算设备否则将从第二计算设备请求(例如，如果第一计算设备具有与其分布式数据库实例相关联的低置信度值的话)的较小的参数的值的部分(和与参数的值的较大部分相关联的密码散列值)。高置信度值可以指示存储在第一计算设备中的参数的值可能与存储在分布式数据库系统中的其它计算设备中的参数的值一致，并且因此，密码散列值被用于验证一致性。

[0282] 在一些情况下，第一计算设备的置信度值可以增加以达到如下阈值，在该阈值处第一计算设备确定它不再应当从分布式数据库系统中的其它计算设备请求特定值、值的特定部分和/或与特定值和/或值的特定部分相关联的密码散列值。例如，如果值的置信度值满足特定标准(例如，达到阈值)，则第一计算设备可以确定值已经收敛，并且不进一步请求与其它设备交换该值。又例如，可以基于其置信度值满足标准来将值添加到密码散列值。在这种情况下，如以上详细讨论的，可以在置信度值满足标准之后发送该值集合的密码散列值，而不是单独的值。利用与(值的)其余部分相关联的密码散列值而交换较少的值和/或较小的(值的)实际部分可以促进分布式数据库系统的计算设备之间的高效通信。

[0283] 在一些情况下，随着分布式数据库实例的参数的特定值的置信度值增加，与该分布式数据库实例相关联的计算设备可以较不频繁地请求与其它计算设备交换该参数的值。类似地，在一些情况下，随着分布式数据库实例的参数的特定值的置信度值降低，与该分布式数据库实例相关联的计算设备可以较频繁地请求与其它计算设备交换该参数的值。因此，置信度值可以用于减少计算设备之间交换的值的数量。

[0284] 虽然以上已经描述了各种实施例，但是应当理解的是，它们仅作为示例给出，而不是限制。在上述方法指示某些事件按某种顺序发生的情况下，某些事件的排序可以被修改。此外，如果可能的话，事件中的某些事件可以在并行过程中同时执行，以及如上文所描述的那样顺序地执行。

[0285] 本文描述的一些实施例涉及具有非暂态计算机可读介质(也可称为非暂态处理器可读介质)的计算机存储产品，该非暂态计算机可读介质在其上具有用于执行各种计算机实现的操作的指令或计算机代码。计算机可读介质(或处理器可读介质)在它本身不包括暂态传播信号(例如，在诸如空间或线缆之类的传输介质上承载信息的传播电磁波)的意义上是非暂态的。介质和计算机代码(也可以称为代码)可以是为特定的一个或多个目的设计和构造的介质和计算机代码。非暂态计算机可读介质的示例包括但不限于：磁存储介质，诸如硬盘、软盘和磁带；光存储介质，诸如紧凑盘/数字视频盘(CD/DVD)、紧凑盘只读存储器(CD-ROM)和全息设备；磁光存储介质，诸如光盘；载波信号处理模块；专门被配置为存储和执行程序代码的硬件设备，诸如专用集成电路(ASIC)、可编程逻辑器件(PLD)、只读存储器(ROM)和随机存取存储器(RAM)设备。本文描述的其它实施例涉及计算机程序产品，其可以包括例如本文讨论的指令和/或计算机代码。

[0286] 计算机代码的示例包括但不限于微代码或微指令、诸如由编译器产生的机器指令、用于产生web服务的代码、以及包含由计算机使用解释器执行的较高级指令的文件。例如，可以使用命令式编程语言(例如，C、Fortran等)、函数式编程语言(Haskell、Erlang等)、逻辑编程语言(例如，Prolog)、面向对象的编程语言(例如，Java、C++等)或其它合适的编程

语言和/或开发工具来实现实施例。计算机代码的附加示例包括但不限于控制信号、加密代码和压缩代码。

[0287] 虽然以上已经描述了各种实施例，但是应当理解的是，它们仅作为示例而不是限制给出，并且可以进行形式和细节上的各种改变。本文描述的装置和/或方法的任何部分可以以除了互相排斥的组合之外的任何组合进行组合。本文描述的实施例可以包括所描述的不同实施例的功能、部件和/或特征的各种组合和/或子组合。

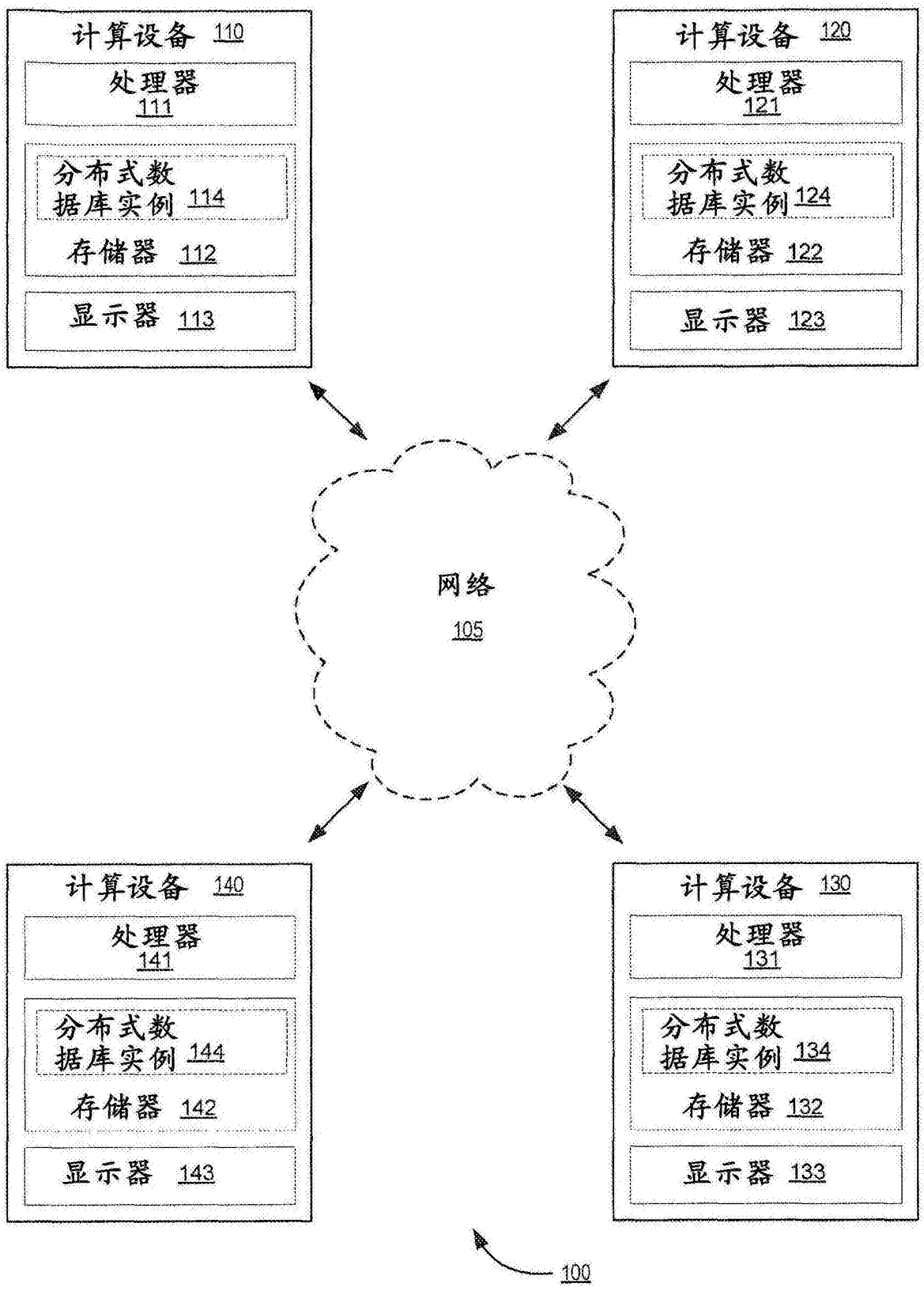

计算设备 110
处理器 111
分布式数据库实例 114
存储器 112
显示器 113
计算设备 120
处理器 121
分布式数据库实例 124
存储器 122
显示器 123
网络 105
计算设备 140
处理器 141
分布式数据库实例 144
存储器 142
显示器 143
计算设备 130
处理器 131
分布式数据库实例 134
存储器 132
显示器 133
100

图1

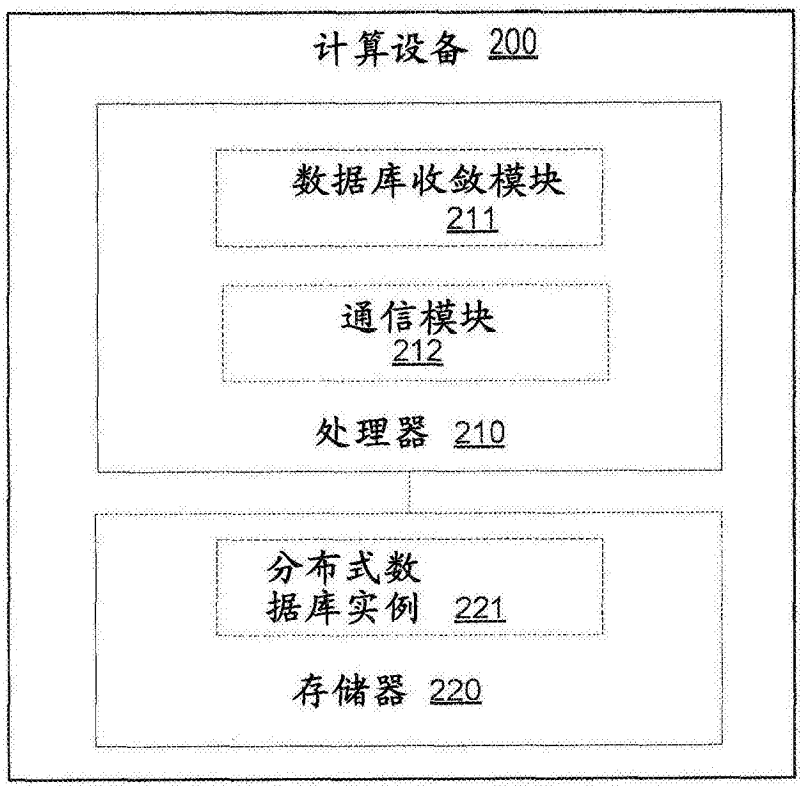


图2

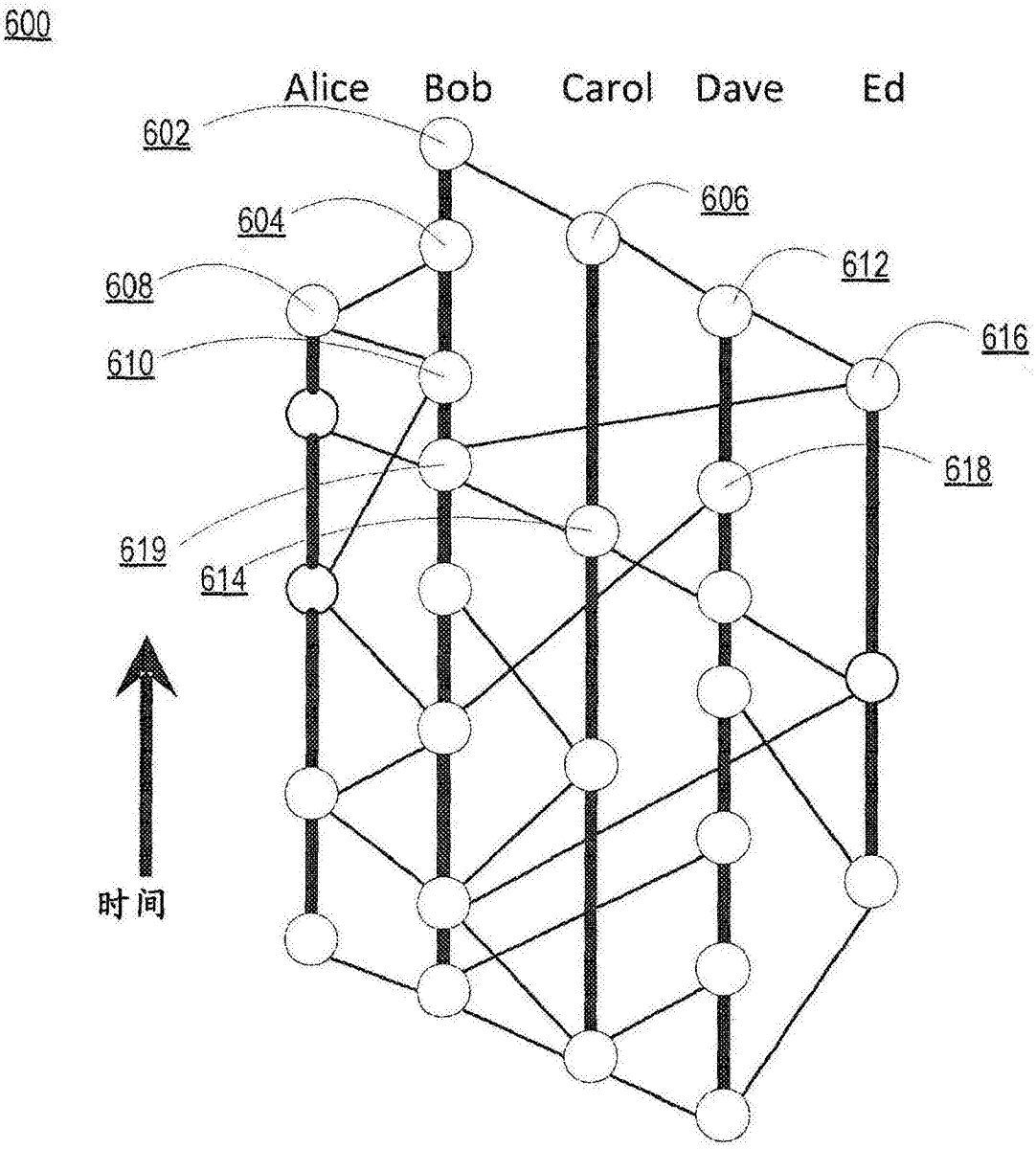

600
Alice
Bob
Carol
Dave
Ed
602
604
606
608
610
612
616
618
619
614
时间

图3

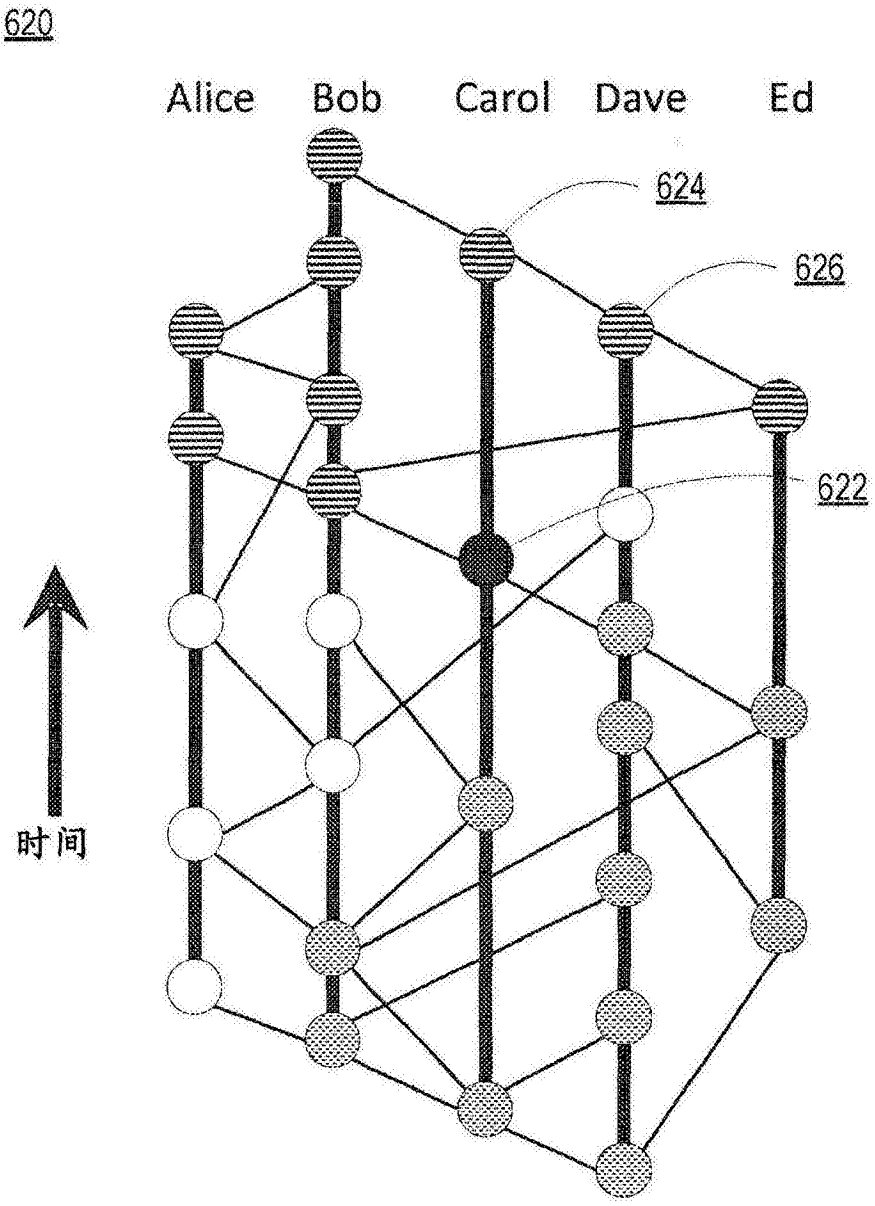


图4

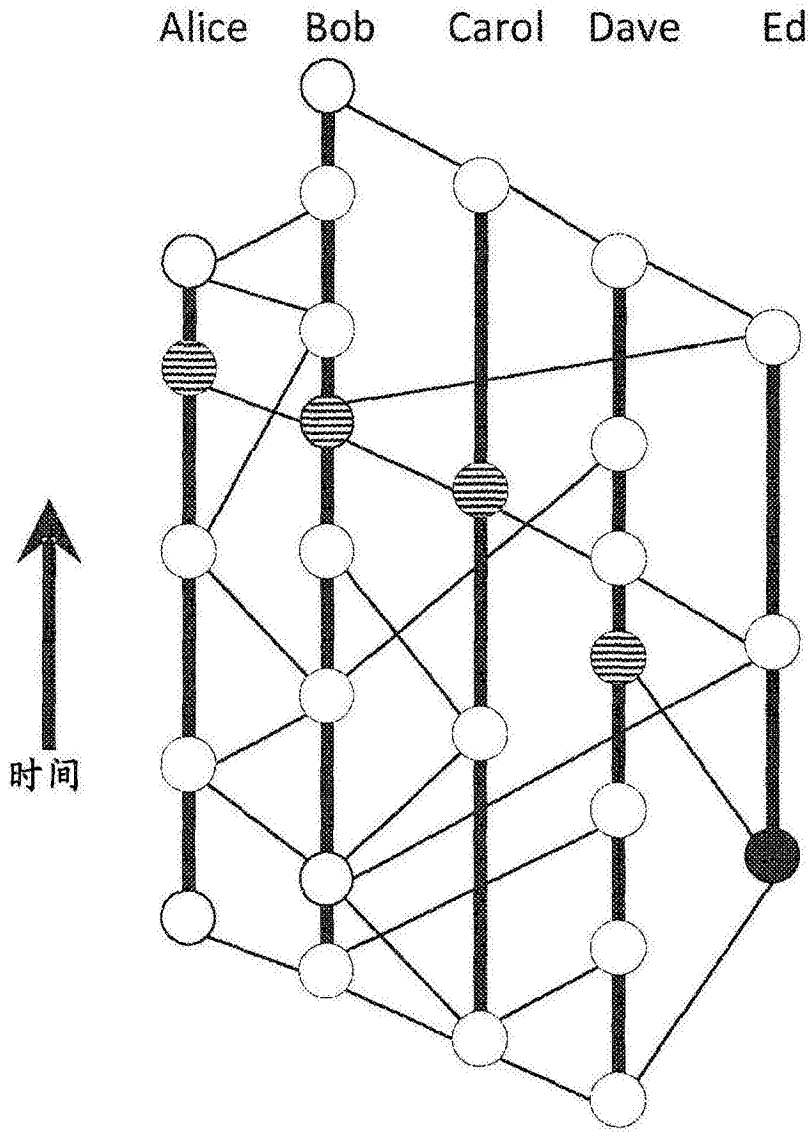


图5

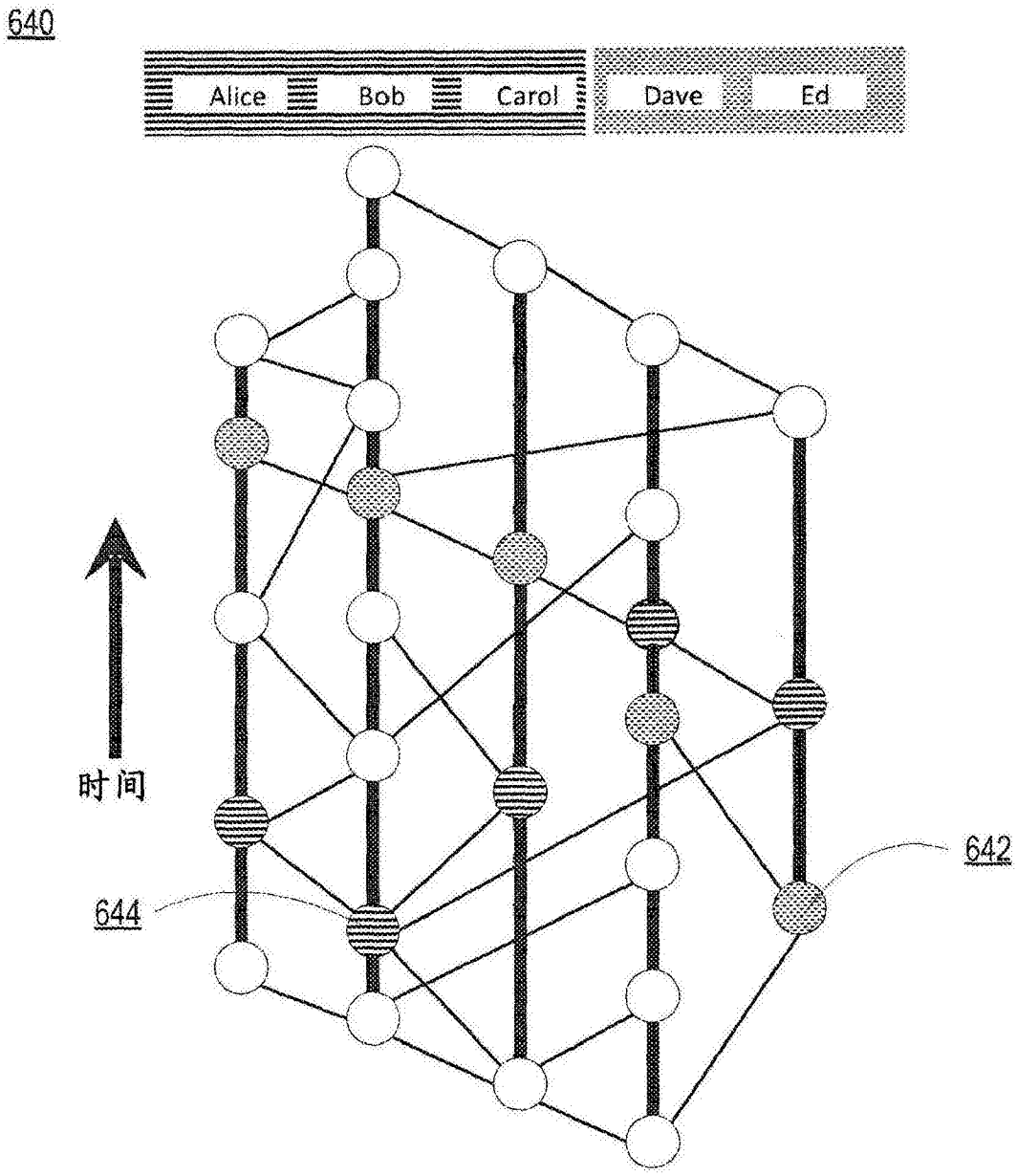


图6

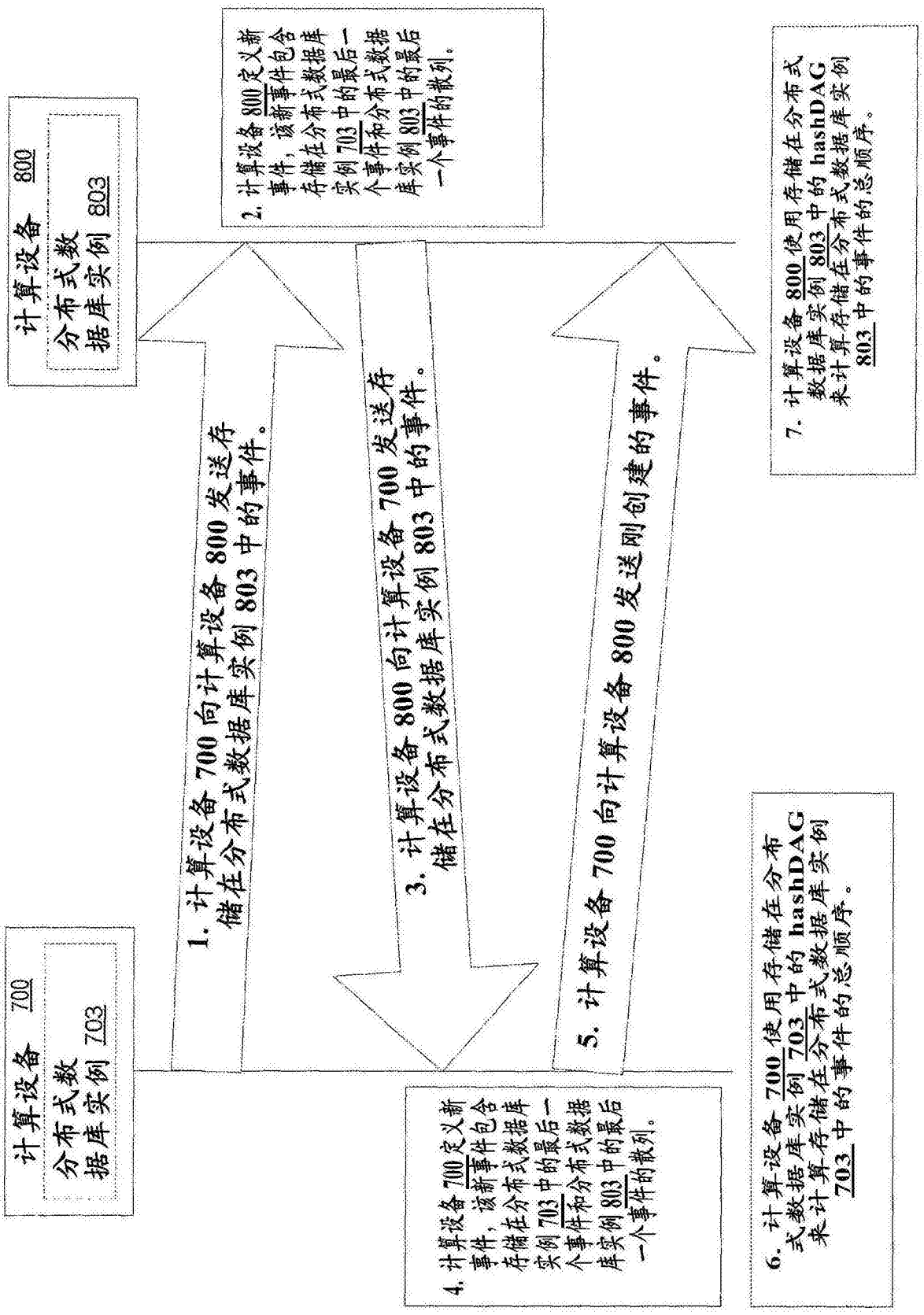

计算设备 800
分布式数据库实例 803
计算设备 700
分布式数据库实例 703
1. 计算设备 700 向计算设备 800 发送存储在分布式数据库实例 803 中的事件。
2. 计算设备 800 定义新事件，该新事件包含存储在分布式数据库实例 703 中的最后一个事件和分布式数据库实例 803 中的最后一个事件的散列。
3. 计算设备 800 向计算设备 700 发送存储在分布式数据库实例 803 中的事件。
4. 计算设备 700 定义新事件，该新事件包含存储在分布式数据库实例 703 中的最后一个事件和分布式数据库实例 803 中的最后一个事件的散列。
5. 计算设备 700 向计算设备 800 发送刚创建的事件。
6. 计算设备 700 使用存储在分布式数据库实例 703 中的 hashDAG 来计算存储在分布式数据库实例 703 中的事件的总顺序。
7. 计算设备 800 使用存储在分布式数据库实例 803 中的 hashDAG 来计算存储在分布式数据库实例 803 中的事件的总顺序。

图7

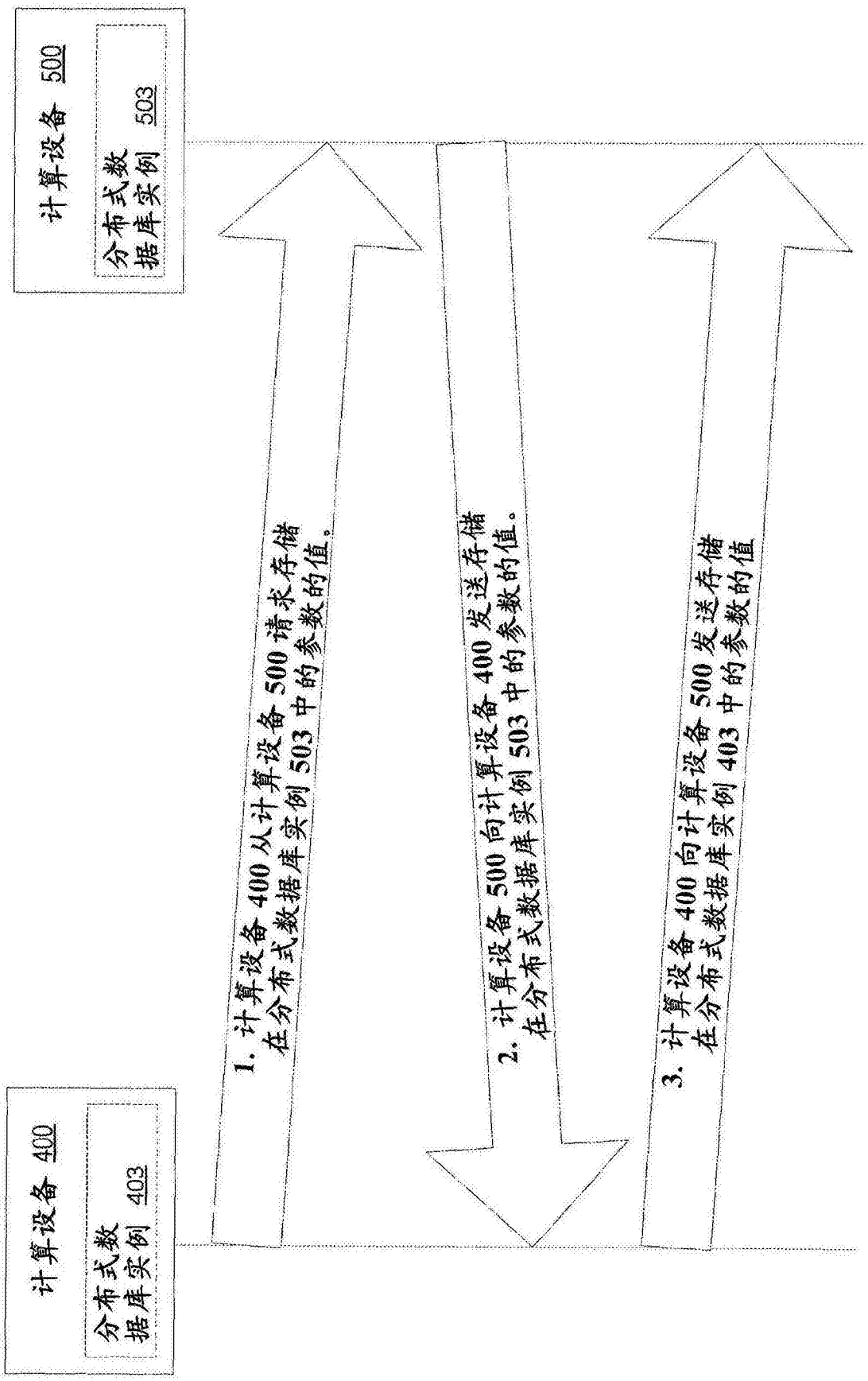

计算设备 500
分布式数据库实例 503
计算设备 400
分布式数据库实例 403
1. 计算设备 400 从计算设备 500 请求存储在分布式数据库实例 503 中的参数的值。
2. 计算设备 500 向计算设备 400 发送存储在分布式数据库实例 503 中的参数的值。
3. 计算设备 400 向计算设备 500 发送存储在分布式数据库实例 403 中的参数的值

图8

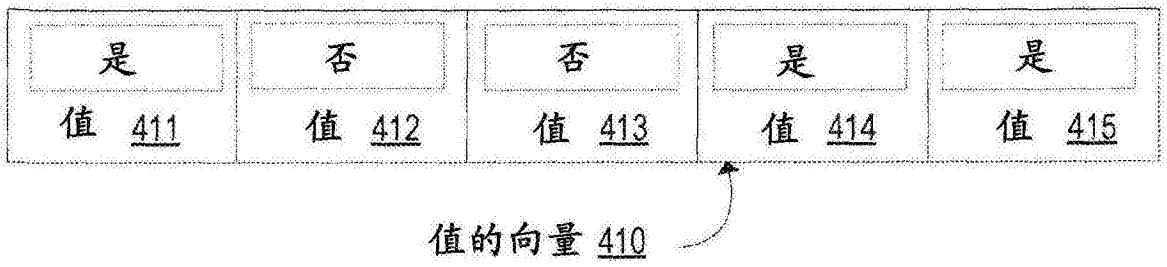

是
否
否
是
是
值 411
值 412
值 413
值 414
值 415
值的向量 410

图9a

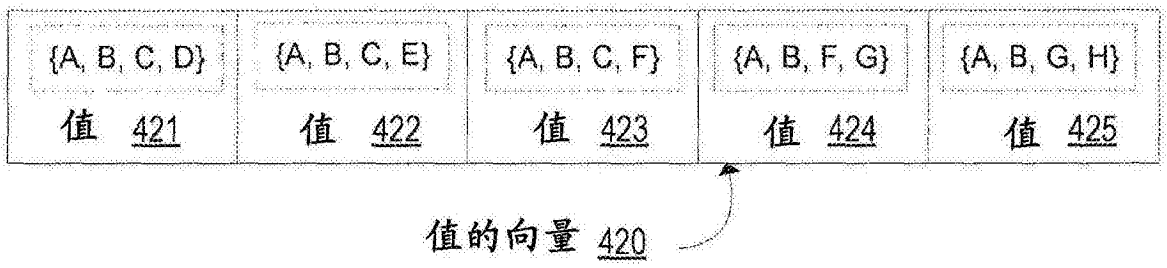

{A, B, C, D}
{A, B, C, E}
{A, B, C, F}
{A, B, F, G}
{A, B, G, H}
值 421
值 422
值 423
值 424
值 425
值的向量 420

图9b

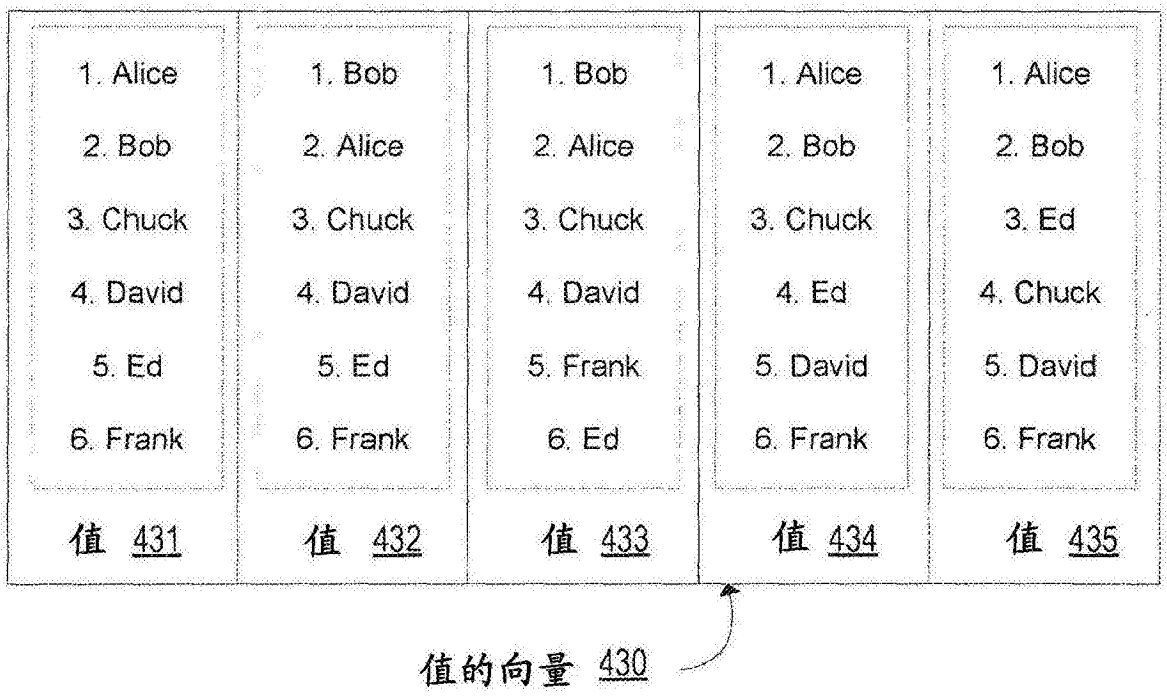

1. Alice
2. Bob
3. Chuck
4. David
5. Ed
6. Frank
1. Bob
2. Alice
3. Chuck
4. David
5. Ed
6. Frank
1. Bob
2. Alice
3. Chuck
4. David
5. Frank
6. Ed
1. Alice
2. Bob
3. Chuck
4. Ed
5. David
6. Frank
1. Alice
2. Bob
3. Ed
4. Chuck
5. David
6. Frank
值 431
值 432
值 433
值 434
值 435
值的向量 430

图9c

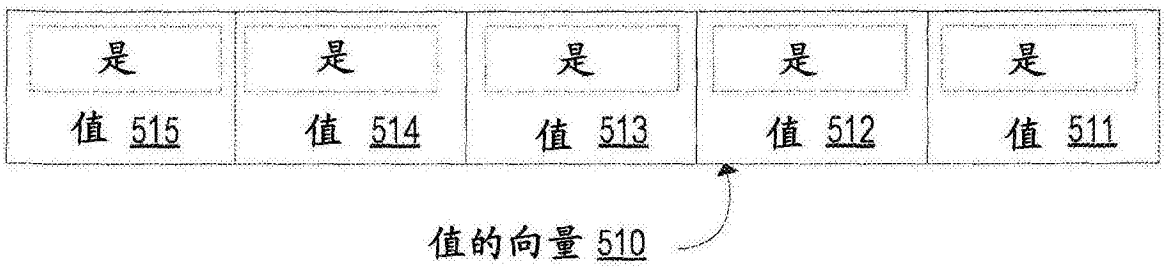


图10a

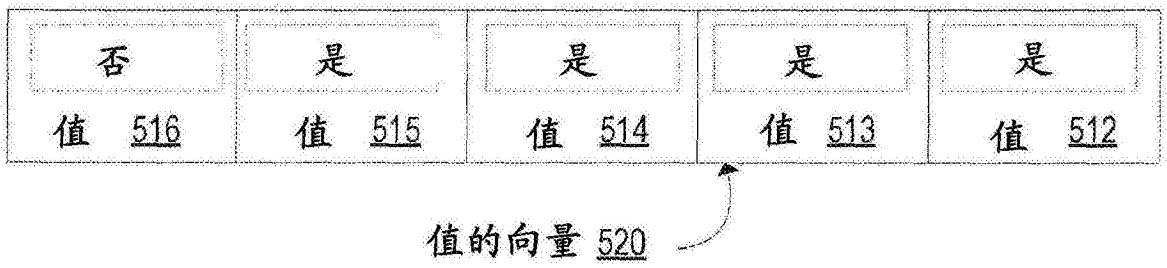


图10b

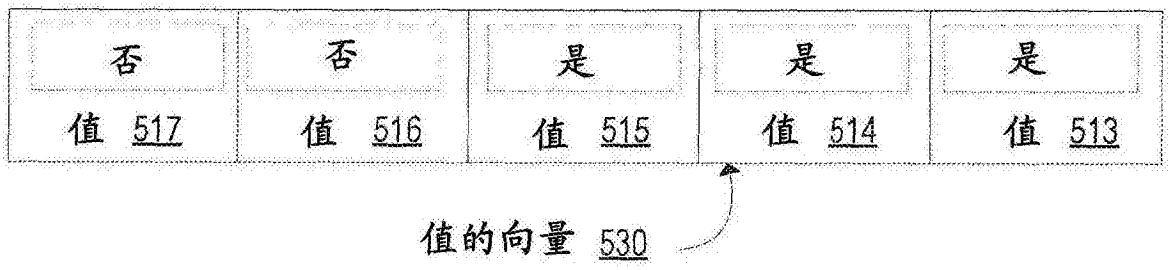


图10c

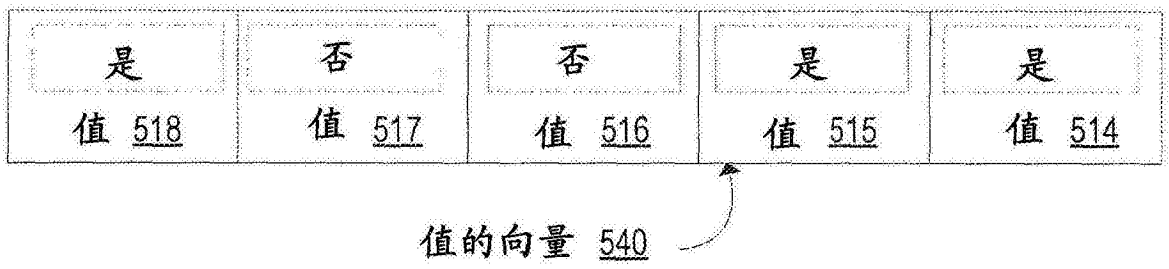


图10d

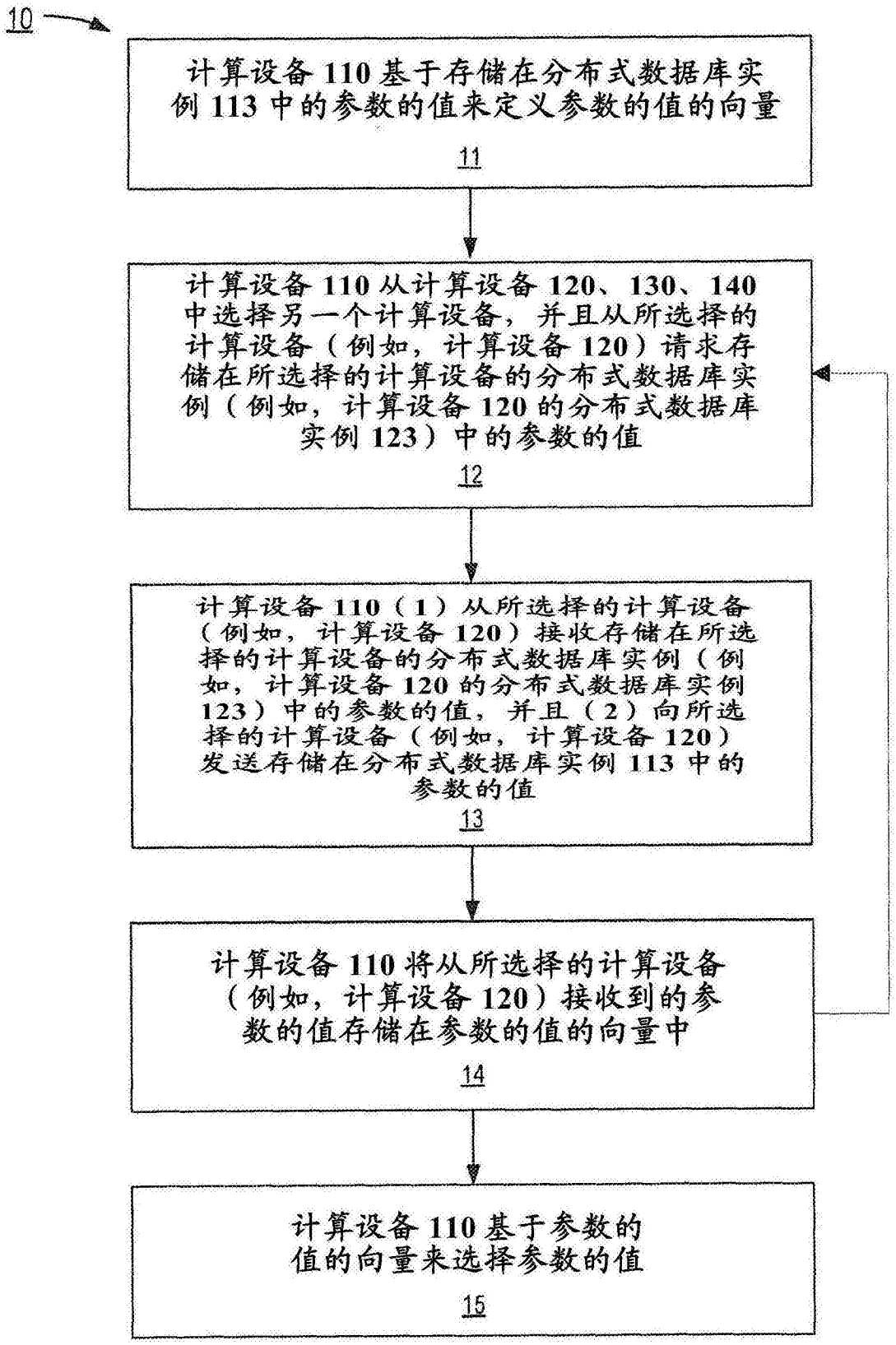

10
计算设备 110 基于存储在分布式数据库实例 113 中的参数的值来定义参数的值的向量
11
计算设备 110 从计算设备 120、130、140 中选择另一个计算设备，并且从所选择的计算设备（例如，计算设备 120）请求存储在所选择的计算设备的分布式数据库实例（例如，计算设备 120 的分布式数据库实例 123）中的参数的值
12
计算设备 110（1）从所选择的计算设备（例如，计算设备 120）接收存储在所选择的计算设备的分布式数据库实例（例如，计算设备 120 的分布式数据库实例 123）中的参数的值，并且（2）向所选择的计算设备（例如，计算设备 120）发送存储在分布式数据库实例 113 中的参数的值
13
计算设备 110 将从所选择的计算设备（例如，计算设备 120）接收到的参数的值存储在参数的值的向量中
14
计算设备 110 基于参数的值的向量来选择参数的值
15

图11

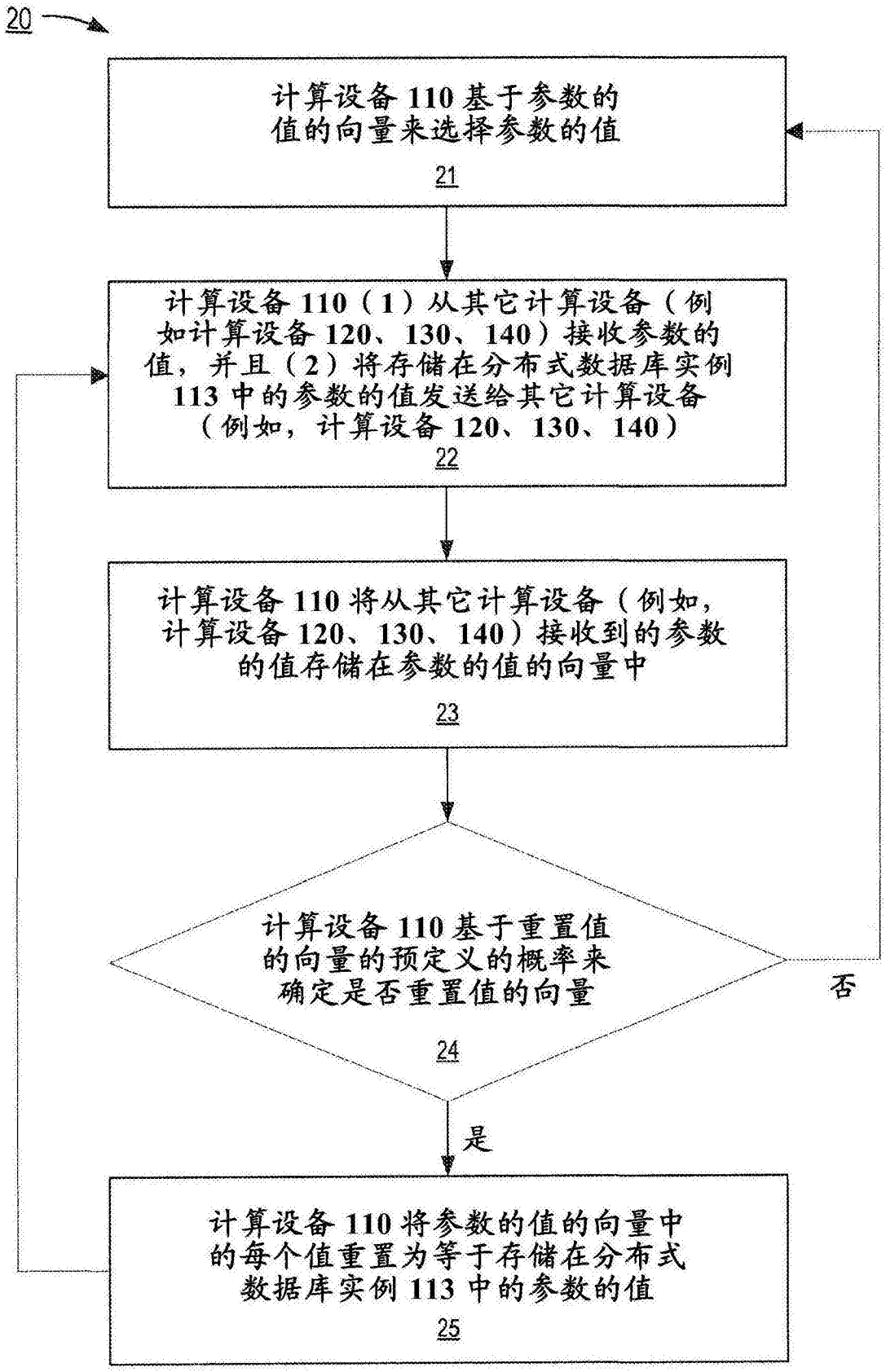

20
计算设备 110 基于参数的值的向量来选择参数的值
21
计算设备 110（1）从其它计算设备（例如计算设备 120、130、140）接收参数的值，并且（2）将存储在分布式数据库实例 113 中的参数的值发送给其它计算设备（例如，计算设备 120、130、140）
22
计算设备 110 将从其它计算设备（例如，计算设备 120、130、140）接收到的参数的值存储在参数的值的向量中
23
计算设备 110 基于重置值的向量的预定义的概率来确定是否重置值的向量
24
否
是
计算设备 110 将参数的值的向量中的每个值重置为等于存储在分布式数据库实例 113 中的参数的值
25

图12

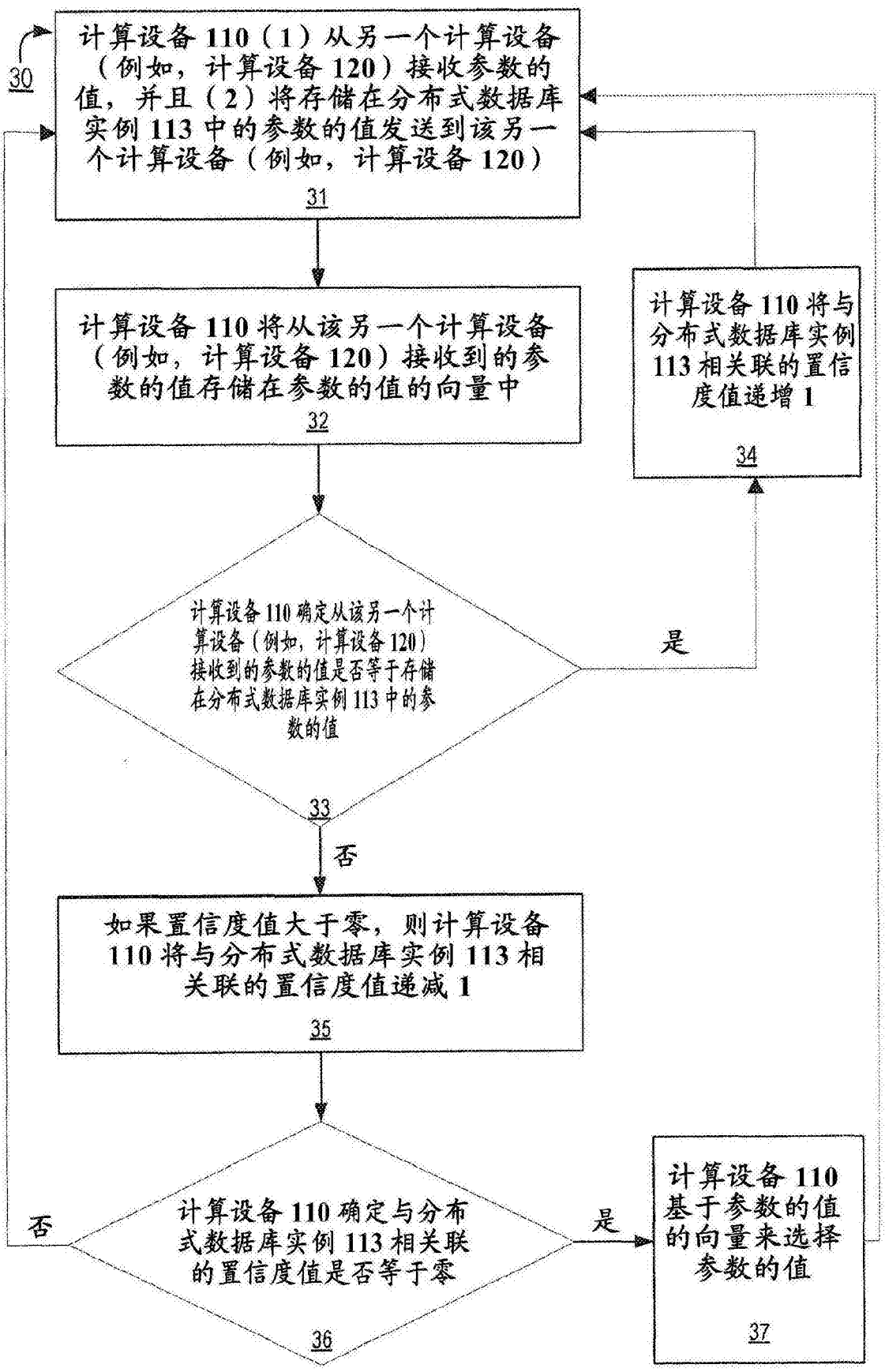

30
计算设备 110（1）从另一个计算设备（例如，计算设备 120）接收参数的值，并且（2）将存储在分布式数据库实例 113 中的参数的值发送到该另一个计算设备（例如，计算设备 120）
31
计算设备 110 将从该另一个计算设备（例如，计算设备 120）接收到的参数的值存储在参数的值的向量中
32
计算设备 110 将与分布式数据库实例 113 相关联的置信度值递增 1
34
计算设备 110 确定从该另一个计算设备（例如，计算设备 120）接收到的参数的值是否等于存储在分布式数据库实例 113 中的参数的值
33
是
否
如果置信度值大于零，则计算设备 110 将与分布式数据库实例 113 相关联的置信度值递减 1
35
计算设备 110 确定与分布式数据库实例 113 相关联的置信度值是否等于零
36
否
是
计算设备 110 基于参数的值的向量来选择参数的值
37

图13

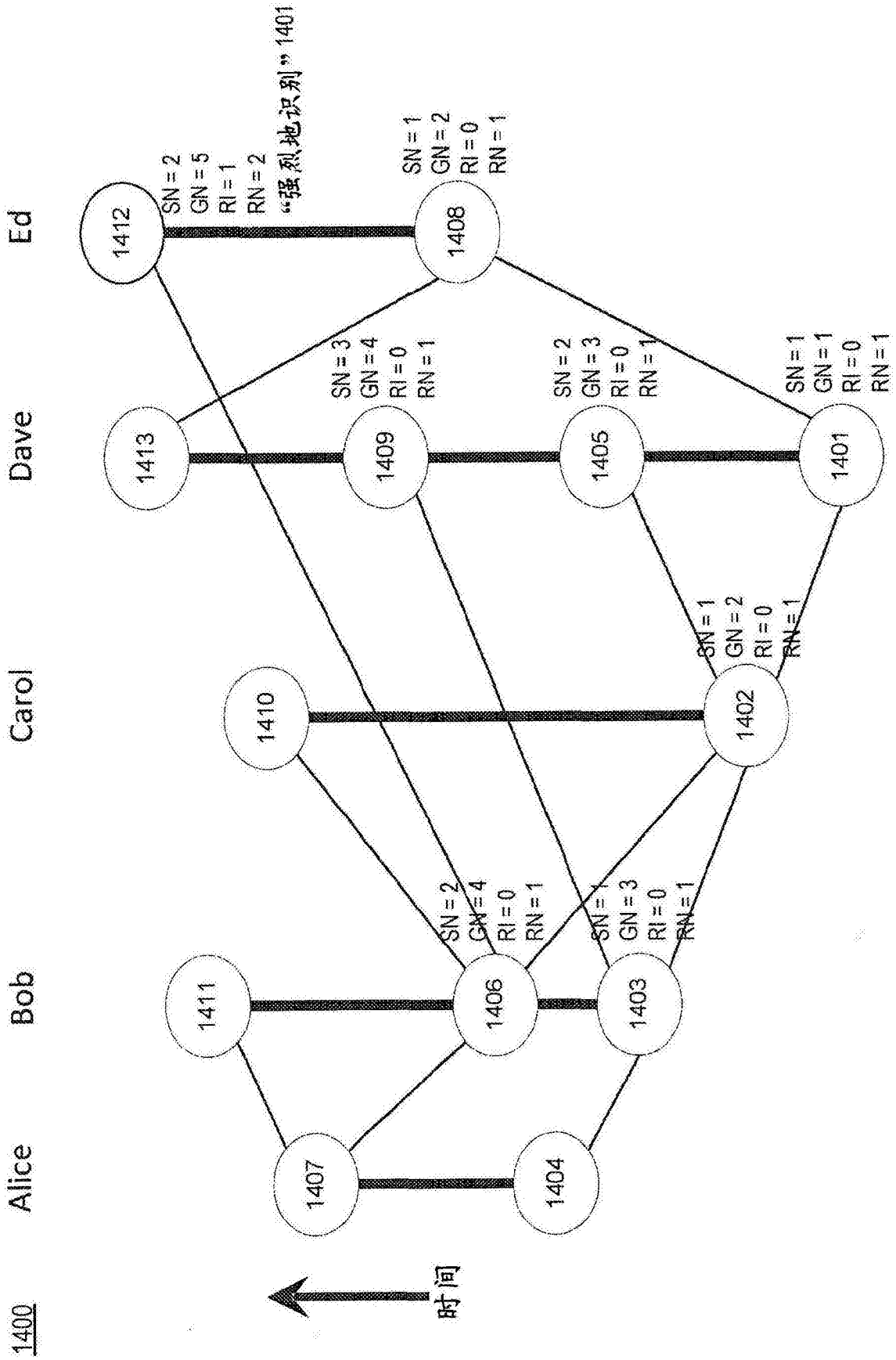

1400
Alice
Bob
Carol
Dave
Ed
时间
1412
SN = 2
GN = 5
RI = 1
RN = 2
"强烈地识别" 1401
1408
SN = 1
GN = 2
RI = 0
RN = 1
1413
1409
SN = 3
GN = 4
RI = 0
RN = 1
1405
SN = 2
GN = 3
RI = 0
RN = 1
1401
SN = 1
GN = 1
RI = 0
RN = 1
1410
1402
SN = 1
GN = 2
RI = 0
RN = 1
1411
1406
SN = 2
GN = 4
RI = 0
RN = 1
1403
SN = 1
GN = 3
RI = 0
RN = 1
1407
1404

图14

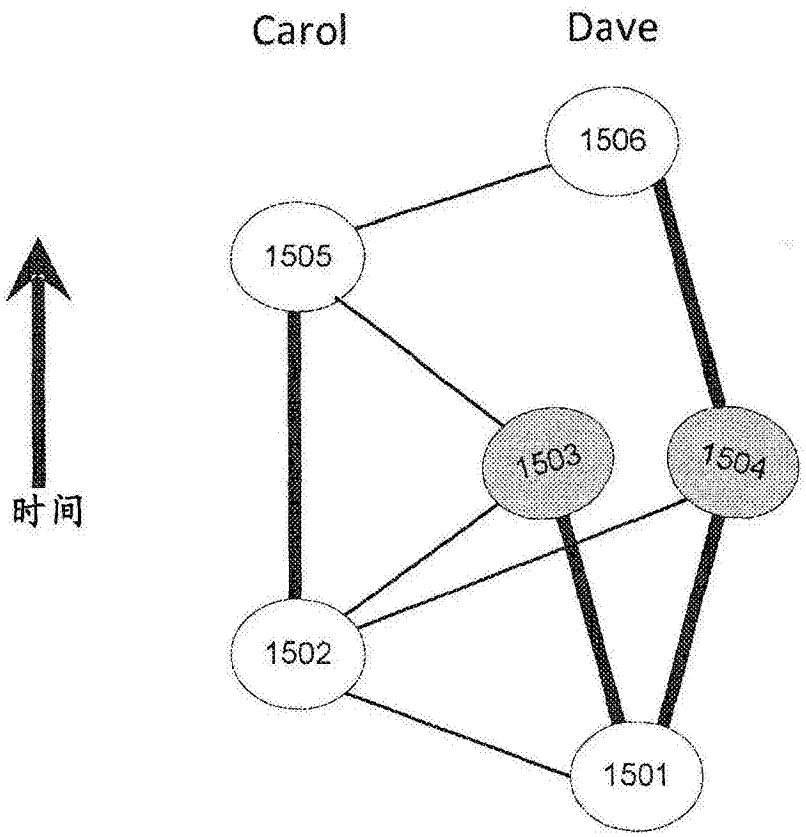


图15

事件为元组 $e = \{d, h, t, c, s\}$，其中：

$d = data(e)$ = “有效载荷”数据，其可以包括事务
$h = hashes(e)$ = 事件的父亲的散列的列表，首先是自父亲
$t = time(e)$ = 创建者声明的事件创建日期和时间
$c = creator(e)$ = 创建者的 ID 号
$s = sig(e)$ = 创建者对$\{d, h, t, c\}$的数字签名
n = 群体中的成员数量
$m = 1 + \lfloor 2n/3 \rfloor$
$first$ = 没有父亲的独特事件
E = 所有事件的集合
$\mathbb{T}$ = 所有可能的（时间，日期）对的集合
$\mathbb{B}$ = {真，假}
$\mathbb{N} = \{0, 1, 2, ...\}$

$$
\begin{aligned}
\text{ancestor} &: E \times E \to B \\
\text{selfAncestor} &: E \times E \to B \\
\text{see} &: E \times E \to B \\
\text{stronglySee} &: E \times E \to B \\
\text{parentRound} &: E \to \mathbb{N} \\
\text{witness} &: E \to \mathbb{B} \\
\text{round} &: E \to \mathbb{N} \\
\text{roundDiff} &: E \times E \to \mathbb{I} \\
\text{votes} &: E \times E \times \mathbb{B} \to \mathbb{N} \\
\text{voteFraction} &: E \times E \to \mathbb{R} \\
\text{vote} &: E \times E \to \mathbb{B} \\
\text{decide} &: E \times E \to \mathbb{B} \\
\text{allFamous} &: \mathbb{I} \to 2^E \\
\text{famous} &: E \to \mathbb{B} \\
\text{roundReceived} &: E \to \mathbb{N} \\
\text{timeReceived} &: E \to \mathbb{T}
\end{aligned}
$$

图16a

$$
\begin{array}{lcl}
\text{ancestor}(x,y) & = & (x=y)\vee(\exists z\in\text{parents}(x):\text{ancestor}(z,y)) \\
\text{selfAncestor}(x,y) & = & \text{ancestor}(x,y)\wedge((\text{selfParent}(x)=y)\vee\text{selfAncestor}(\text{selfParent}(x),y)) \\
\text{see}(x,y) & = & \text{ancestor}(x,y)\wedge\neg(\exists a,b,c\in E: \\
 & & (\text{ancestor}(y,a)\wedge\text{ancestor}(y,b)\wedge c\in\text{parents}(x)\wedge c\in\text{parents}(b)))\wedge \\
 & & \text{creator}(a)=\text{creator}(b)=\text{creator}(c) \\
\text{stronglySee}(x,y) & = & \text{see}(x,y)\wedge(\exists S\in 2^{E}:(|S|=m)\wedge(z\in S\iff(\text{see}(x,z)\wedge\text{see}(z,y)))) \\
\text{parentRound}(x) & = & \begin{cases}0 & \text{if } x=first\\ \max_{y\in parents(x)}round(y) & \text{otherwise}\end{cases} \\
\text{witness}(x) & = & \exists S\in 2^{E}:(|S|=m\wedge \\
 & & (\forall y\in S:(\text{round}(y)=\text{parentRound}(x)\wedge\text{stronglySee}(x,y)))) \\
\text{round}(x) & = & \begin{cases}1+\text{parentRound}(x) & \text{if witness}(x)\\ \text{parentRound}(x) & \text{otherwise}\end{cases} \\
\text{roundDiff}(x,y) & = & \text{round}(x)-\text{round}(y) \\
\text{votes}(x,y,v) & = & |\{z\in E\mid\text{see}(x,y)\wedge\text{roundDiff}(x,z)=1\wedge \\
 & & \text{stronglySee}(x,z)\wedge\text{vote}(z,y)=v\}| \\
\text{voteFraction}(x,y) & = & \text{votes}(x,true)/(\text{votes}(x,true)+\text{votes}(x,false)) \\
\text{vote}(x,y) & = & \begin{cases}\text{see}(x,y) & \text{if roundDiff}(x,y)=1\\ (\text{voteFraction}(x,y)>=1/2) & \text{if }(\text{roundDiff}(x,y)\bmod 5\neq 1)\vee \\ & |\text{voteFraction}(x,y)-1/2|>1/6\\ (1=\text{LSB}(\text{signature}(x))) & \text{otherwise}\end{cases} \\
\text{decide}(x,y) & = & \text{vote}(x,y)\wedge(\text{roundDiff}(x,y)\bmod 5\neq 1)\wedge(\text{voteFraction}(x,y)>2/3) \\
\text{allFamous}(r) & = & \{x\in E\mid\text{famous}(x)\wedge\text{round}(x)=4\} \\
\text{famous}(x) & = & \text{witness}(x)\wedge\exists y\in E:\text{decide}(y,x) \\
\text{roundReceived}(x) & = & \min_{r\in\mathbb{N}}(|\{y\in E|\ \text{round}(y)=r\wedge\ \text{famous}(y)\wedge\ \text{see}(y,x)\}|\ / \\
 & & |\{y\in E|\ \text{round}(y)=r\wedge\ \text{famous}(y)\}|>=1/2) \\
\text{timeReceived}(x) & = & \text{median}(\{\text{time}(y)|y\in E\wedge\text{see}(y,x)\wedge \\
 & & (\exists z\in E:\text{round}(z)=\text{roundReceived}(x)\wedge\text{selfAncestor}(z,y\})\wedge \\
 & & \neg(\exists w\in E:\text{selfAncestor}(y,w)\wedge\text{see}(w,x))\})
\end{array}
$$

图16b

事件为元组 $e = \{d, h, t, c, s\}$，其中：

$d = data(e)$ = “有效载荷”数据，其可以包括事务
$h = hashes(e)$ = 事件的父亲的散列的列表，首先是自父亲
$t = time(e)$ = 创建者声明的事件创建日期和时间
$i = creator(e)$ = 创建者的 ID 号
$s = sig(e)$ = 创建者对$\{d, h, t, c\}$的数字签名
n = 群体中的成员数量
c = 硬币轮的频率（例如，$c = 6$）
E = (所有事件的集合)∪{∅}
$\mathbb{T}$ = 所有可能的（时间，日期）对的集合
$\mathbb{B}$ = {真，假}
$\mathbb{N}$ = $\{0, 1, 2, ...\}$

$$
\begin{aligned}
\text{parents} &: E \to 2^E \\
\text{selfParent} &: E \to E \\
\text{ancestor} &: E \times E \to \mathbb{B} \\
\text{selfAncestor} &: E \times E \to \mathbb{B} \\
\text{see} &: E \times E \to \mathbb{B} \\
\text{stronglySee} &: E \times E \to \mathbb{B} \\
\text{parentRound} &: E \to \mathbb{N} \\
\text{roundInc} &: E \to \mathbb{B} \\
\text{round} &: E \to \mathbb{N} \\
\text{witness} &: E \to \mathbb{B} \\
\text{roundDiff} &: E \times E \to \mathbb{I} \\
\text{votes} &: E \times E \times \mathbb{B} \to \mathbb{N} \\
\text{fractTrue} &: E \times E \to \mathbb{R} \\
\text{decide} &: E \times E \to \mathbb{B} \\
\text{vote} &: E \times E \to \mathbb{B} \\
\text{famous} &: E \to \mathbb{B} \\
\text{roundReceived} &: E \to \mathbb{N} \\
\text{timeReceived} &: E \to \mathbb{T}
\end{aligned}
$$

图17a

$$
\begin{aligned}
\text{parents}(x) &= \textbf{事件 x 的父亲的集合} \\
\text{selfParent}(x) &= \textbf{事件 x 的自父亲，或者如果没有则为}\varnothing \\
\text{ancestor}(x,y) &= (x \neq \varnothing) \wedge ((x = y) \vee (\exists z \in \text{parents}(x) : \text{ancestor}(z,y))) \\
\text{selfAncestor}(x,y) &= (x \neq \varnothing) \wedge ((x = y) \vee \text{selfAncestor}(\text{selfParent}(x),y)) \\
\text{see}(x,y) &= \text{ancestor}(x,y) \wedge \neg(\exists a,b \in E : \text{creator}(y) = \text{creator}(a) = \text{creator}(b) \wedge \\
&\quad \text{ancestor}(x,a) \wedge \text{ancestor}(x,b) \wedge \neg\text{selfAncestor}(a,b) \wedge \neg\text{selfAncestor}(b,a)) \\
\text{stronglySee}(x,y) &= \text{see}(x,y) \wedge (\exists S \in 2^E : (|S| > 2n/3) \wedge (z \in S \iff (\text{see}(x,z) \wedge \text{see}(z,y)))) \\
\text{parentRound}(x) &= \max(\{0\} \cup \{\text{round}(y) | y \in \text{parents}(x)\}) \\
\text{roundInc}(x) &= \exists S \in 2^E : (|S| > 2n/3 \wedge \\
&\quad (\forall y \in S : (\text{round}(y) = \text{parentRound}(x) \wedge \text{stronglySee}(x,y)))) \\
\text{round}(x) &= \text{parentRound}(x) + \begin{cases} 1 & \text{if roundInc}(x) \\ 0 & \text{otherwise} \end{cases} \\
\text{witness}(x) &= (\text{selfParent}(x) = \varnothing) \vee (\text{round}(x) > \text{round}(\text{selfParent}(x))) \\
\text{roundDiff}(x,y) &= \text{round}(x) - \text{round}(y) \\
\text{votes}(x,y,v) &= |\{z \in E \mid \text{roundDiff}(x,z) = 1 \wedge \text{stronglySee}(x,z) \wedge \text{vote}(z,y) = v\}| \\
\text{fractTrue}(x,y) &= \frac{\text{votes}(x,y,true)}{(\text{votes}(x,y,true)+\text{votes}(x,y,false))} \\
\text{decide}(x,y) &= (x \neq \varnothing) \wedge (\text{roundDiff}(x,y) > 1) \wedge (\text{decide}(\text{selfParent}(x),y) \vee \\
&\quad (\text{witness}(x) \wedge (\text{roundDiff}(x,y) \mod c \neq 1) \wedge \neg(\tfrac{1}{3} \leq \text{fractTrue}(x,y) \leq \tfrac{2n}{3}))) \\
\text{vote}(x,y) &= \begin{cases} \text{vote}(\text{selfParent}(x),y) & \text{if } (\neg\,\text{witness}(x)) \vee \text{decide}(\text{selfParent}(x),y) \\ 1 = \text{middleBit}(\text{signature}(x)) & \text{if witness}(x) \\ & \quad \wedge\ \neg\,\text{decide}(\text{selfParent}(x),y) \\ & \quad \wedge\ (\text{roundDiff}(x,y) \neq 1) \\ & \quad \wedge\ (\text{roundDiff}(x,y) \mod c = 1) \\ \text{fractTrue}(x,y) \geq \frac{1}{2} & \text{otherwise} \end{cases} \\
\text{famous}(x) &= \text{witness}(x) \wedge \exists y \in E : \text{decide}(y,x) \wedge \text{vote}(y,x) \\
\text{roundReceived}(x) &= \min_{r \in \mathbb{N}} \frac{|\{y \in E | (\text{round}(y) = r) \wedge \text{famous}(y) \wedge \text{see}(y,x)\}|}{|\{y \in E | (\text{round}(y) = r) \wedge \text{famous}(y)\}|} \geq 1/2) \\
\text{timeReceived}(x) &= \text{median}(\{\text{time}(y) | y \in E \wedge \text{see}(y,x) \wedge \\
&\quad (\exists z \in E : \text{round}(z) = \text{roundReceived}(x) \wedge \text{selfAncestor}(z,y)) \wedge \\
&\quad \neg(\exists w \in E : \text{selfAncestor}(y,w) \wedge \text{see}(w,x))\})
\end{aligned}
$$

图17b