

一种区块数据校验方法和装置

技术领域

本申请涉及计算机通信技术领域，尤其涉及一种区块数据校验方法和装置。

5

背景技术

区块链技术也称为分布式账本技术，是一种分布式互联网数据库技术，其特点是去中心化、公开透明、不可篡改、可信任。区块链中存储的每一条数据，将会广播存储至区块链中各个区块中，因此，区块链的每个区块中存储全量的、

10 一致的数据。

在实际应用中，区块链需要保持各个区块中存储数据一致。通常采用数据校验的方式校验各个区块中存储的数据是否一致。

具体地，在进行区块数据校验时，需要确定每一个区块的校验值，并将各个区块的校验值进行比对，进而判断区块链的各个区块中存储的数据是否一

15 致。

经研究发现，在确定每一个区块的校验值时，由于区块中存储的数据量比较大，那么在确定该区块的校验值时，需要将区块中存储的所有数据进行计算，计算得到该区块的校验值。但是随着区块中存储数据量的增加，将使得计算校验值的计算量增加，进而导致数据校验效率降低。

20 因此，如何提高区块数据校验的效率成为本领域亟待解决的问题。

背景技术部分公开的信息只是为了加强对本申请的一般背景的理解，不应视为承认或默许这种信息构成本领域技术人员已知的现有技术。

发明内容

25 有鉴于此，本申请实施例提供了一种区块数据校验方法和装置，以解决现有技术中区块数据校验效率较低的问题。

本申请实施例提供一种区块数据校验方法，包括：

从包含多个数据节点的区块中选择用于存储新增数据的数据节点；

针对所述数据节点，确定所述新增数据的校验值，并根据所述新增数据的校验值，计算所述数据节点的更新校验值；

根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，得到所述区块的校验值；

其中，所述区块的校验值用于校验所述区块中存储的数据是否与区块链中其他区块中存储的数据一致。

本申请实施例还提供一种区块数据校验装置，包括：

选择模块，从包含多个数据节点的区块中选择用于存储新增数据的数据节点；

确定模块，针对所述数据节点，确定所述新增数据的校验值，并根据所述新增数据的校验值，计算所述数据节点的更新校验值；

处理模块，根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，得到所述区块的校验值；

其中，所述区块的校验值用于校验所述区块中存储的数据是否与区块链中其他区块中存储的数据一致。

本申请实施例采用的上述至少一个技术方案能够达到以下有益效果：

在区块中新增数据时，利用新增数据确定用于存储该新增数据的数据节点的更新校验值，避免了现有技术中当区块中新增数据时利用区块中所有数据计算校验值所导致的时间消耗较长的问题，使在计算新增数据的区块的校验值时，所消耗时间与区块中数据量的多少无关，有效缩短了校验值的计算时间，从而提高了区块数据校验的效率。

附图说明

此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

图 1 为本申请实施例提供的一种区块数据校验方法的流程示意图；

图 2 为本申请实施例提供的 Merkle tree 结构示意图；

图 3 为本申请实施例提供的确定新增数据的校验值的流程示意图；

图 4 为本申请实施例提供的确定新增数据的校验值的流程示意图；

5 图 5 为本申请实施例的一种区块数据校验装置的结构示意图。

具体实施方式

为了实现本申请的目的，本申请实施例提供了一种区块数据校验方法及装置，从包含多个数据节点的区块中选择用于存储新增数据的数据节点；针对所
10 述数据节点，确定所述新增数据的校验值，并根据所述新增数据的校验值，计
算所述数据节点的更新校验值；根据所述数据节点的更新校验值与所述区块中
包含的其他数据节点的原始校验值，得到所述区块的校验值；其中，所述区块
的校验值用于校验所述区块中存储的数据是否与区块链中其他区块中存储的
数据一致。与处理所有历史数据的现有技术相比，本申请实施例着重考虑区块
15 的新增数据，对于新增数据的区块，提出了采用增量方式计算校验值，避免了
现有技术中当区块中新增数据时利用区块中所有数据计算校验值所导致的时
间消耗较长的问题，使在计算新增数据的区块的校验值时，所消耗时间与区块
中数据量的多少无关，有效缩短了校验值的计算时间，从而提高了区块数据校
验的效率。

20 需要说明的是，本申请实施例中所记载的用于计算校验值的设定算法可以
是各种校验算法，还可以是各种不可逆算法，具体算法类型在本申请实施例中
不做具体限定。

例如：可以使用哈希算法计算得到需要的校验值。哈希（Hash）算法，即
散列函数，又可以称之为哈希函数。它是一种单向密码体制，即它是一个从明
25 文到密文的不可逆的映射，即只有加密过程，没有解密过程。同时，哈希函数
可以将任意长度的输入经过变化以后得到固定长度的输出。典型的哈希算法包
括 MD2、MD4、MD5 和 SHA-1 等。

本领域技术人员应该理解，本申请实施例的说明书中提到的哈希算法仅是示例，并不构成对本发明的限制。本领域技术人员能够根据实际应用和需要采用其他用于校验的算法，这里不做具体限定。

为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的
5 实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

以下结合附图，详细说明本申请各实施例提供的技术方案。

10 图 1 为本申请实施例提供的一种区块数据校验方法的流程示意图。所述区块数据校验方法包括以下步骤。

步骤 S110：从包含多个数据节点的区块中选择用于存储新增数据的数据节点。

区块链由若干区块采用链式数据结构连接形成。每个区块中包括多个数据
15 节点。区块中包含的数据节点可以构建为梅克尔树（Merkle tree）的形式，即包含根节点和多个叶子节点。

在 Merkle tree 结构中，区块中包含的数据节点作为 Merkle tree 结构中的底层叶子节点，如图 2 所示。

从图 2 中可以看出，在 Merkle tree 结构中，A、B、C、D 和 E 对应于区块
20 中的数据节点。

本领域技术人员应该理解，在区块中基于数据节点构建梅克尔树，可以根据需要确定梅克尔树的深度。

需要说明的是，为了保证区块链中各个区块中存储数据的一致性，在区块链的各个区块中构建的梅克尔树的深度相同，此外，各个区块中包含的数据节
25 点的个数也相同。此外，还应该理解，这里列出的梅克尔树只是一种举例说明，本领域技术人员可以采用其他适当的数据构建形式，这里不做具体限定。

对于待存储的数据（这里的数据可以称之为新增数据），首先，确定用于

存储该新增数据的区块；其次，确定在该区块中用于存储该新增数据的数据节点。

那么在接收到新增数据时，首先，确定用于存储所述新增数据的区块信息，所述区块信息中包含区块标识和数据节点标识。

5 具体地，对所述新增数据对应的数据内容进行计算，得到确定的区块信息。

根据本申请实施例，新增数据对应的数据内容可以根据新增数据的类型确定，也可以根据新增数据中包含的数据确定，这里不做具体限定。

例如：该新增数据为文本数据，那么新增数据对应的数据内容可以理解为该文本数据的内容摘要；也可以理解为该文本数据；那么可以按照设定算法，
10 对该新增数据对应的数据内容（例如：内容摘要，或者文本数据）进行计算，得到计算结果，此时根据计算结果确定区块信息。

再例如：该新增数据为一条交易数据，那么可以按照设定算法，对该交易数据进行计算，得到计算结果，此时根据计算结果确定区块信息。

本申请实施例中所记载的设定算法：例如采用求余的计算方式，那么将数据内容转换为整数数据，以区块链中所包含的区块个数作为除数，利用该除数
15 对整数数据进行计算，得到的商值和余数。这里的商值可以视为区块标识，余数可以视为区块中数据节点标识；或者商值可以视为数据节点标识，余数可以视为区块标识。

当然，为了根据需要将新增数据分配至适当的区块和数据节点，本领域技术人员也可以设想对数据内容进行计算的其他方式，例如对数据内容进行随机
20 计算等等，这里不做限定。

其次，根据所述区块标识，从区块链中查找与所述区块标识对应的区块；以及从查找到的区块中确定与所述数据节点标识对应的数据节点。

在形成区块链的过程中，为每一个区块确定一个区块标识，那么在得到区块信息时，可以根据区块信息中的区块标识，从区块链中查找到与该区块标识
25 对应的区块；

由于每一个区块中包含多个数据节点，可以为每一个数据节点确定数据节

点标识，那么在得到区块信息时，可以根据区块信息中的数据节点标识，从查找到的区块中确定与该数据节点标识对应的数据节点。

最后，将查找到的数据节点作为用于存储所述新增数据的数据节点。

步骤 S120：针对所述数据节点，确定所述新增数据的校验值。

5 在本申请实施例中，由于新增数据在被存储至数据节点之前，数据节点中有可能已经存储了数据，而这些数据中有可能有些数据与新增数据之间存在某种关联关系，那么与新增数据之间存在关联关系的数据可以被称之为关联数据；也可能数据节点中存储的数据与该新增数据之间不具备关联关系，即在数据节点中不存在与该新增数据关联的关联数据。

10 需要说明的是，本申请实施例中所记载的关联关系可以是在用户账户、支付活动、交易时间、交易地点等因素方面产生的关系，例如：同一个用户的两个交易数据，那么这两个交易数据因为用户账户相同而产生关联关系；也可以是交易行为产生的关系，例如：一个交易数据所记载的内容是用户 A 向用户 B 转账 100 元；另一条交易数据所记载的内容是用户 B 向用户转账 100 元；那么
15 这两条交易数据因为转账行为中所包含的对象和金额相同而产生关联关系；这里对于关联关系的具体内容不做具体限定。

此外，新增数据在被存储至数据节点之前，数据节点中可能尚未存储数据，此时，可以视为在数据节点中不存在与该新增数据关联的关联数据。

那么在确定所述新增数据的校验值时，可以但不限于分为以下两种情形：

20 第一种情形：在数据节点中不存在与该新增数据关联的关联数据。

那么确定所述新增数据的校验值可以参考图 3，图 3 为本申请实施例提供的确定新增数据的校验值的流程示意图。

S310：确定所述新增数据对应的字符串。

在本申请实施例中，确定新增数据对应的字符串可以采用将数据转换为字符串的任何适当的技术手段，例如通过相应的转换函数将新增数据转换为字符串等，这里不做具体限定。
25

S320：按照设定算法，对所述字符串进行计算得到所述新增数据的校验值。

具体地，利用哈希算法，对所述字符串进行计算得到所述新增数据的哈希值，将该哈希值作为该新增数据的校验值。

本领域技术人员应该理解，这里提到的哈希算法仅是示例，并不构成对本发明的限制。

5 第二种情形：在数据节点中存在与该新增数据关联的关联数据。

那么确定所述新增数据的校验值可以参考图 4，图 4 为本申请实施例提供的确定新增数据的校验值的流程示意图。

S410：分别确定所述关联数据对应的第一字符串和所述新增数据对应的第二字符串。

10 同样地，确定关联数据对应的第一字符串和新增数据对应的第二字符串可以采用将数据转换为字符串的任何适当的技术手段，例如通过相应的转换函数将数据转换为字符串等，这里不做具体限定。

S420：按照设定方式对所述第一字符串和所述第二字符串进行排序。

需要说明的是，设定方式，可以根据实际需要确定，也可以根据实验结果
15 确定，在本申请实施例中所采用的排序方式不限定。例如：将第一字符串和第二字符串顺序排列，其目的是将第一字符串和第二字符串进行组合，形成用于计算的字符串组，是第一字符串在前还是第二字符串在前，在本申请实施例中不限定。本领域技术人员也可以采用其他排序方式。

S430：按照设定算法，对排序后得到的字符串进行计算得到所述新增数据
20 的校验值。

具体地，利用哈希算法，对排序后得到的字符串进行计算得到所述新增数据的哈希值，将该哈希值作为该新增数据的校验值。

步骤 S130：根据所述新增数据的校验值，计算所述数据节点的更新校验值。

具体地，确定所述数据节点在未存储所述新增数据之前的原始校验值；根据
25 所述原始校验值和所述新增数据的校验值，计算得到所述数据节点的更新校验值。

在本申请实施例中，由于数据节点在存储该新增数据之前如果已经存储了

其他数据，那么在存储其他数据时，将为该数据节点计算一个校验值，这个校验值相对于新增数据来说，可以称之为原始校验值。

例如：对于数据节点 A，每存储一个数据，将计算一个校验值，也就是说数据节点的校验值将随着数据节点中数据的更新发生变化。假设在存储新增数据之前，该数据节点 A 中已存储了数据 11 和数据 12，那么数据节点在存储数据 11 时，计算得到校验值 a；数据节点在存储数据 12 时，a 可以称之为原始校验值，此时利用原始校验值计算得到校验值 b；那么 b 可以称之为更新校验值。

假设新增数据为数据 13，那么根据步骤 S120 得到数据 13 的校验值 c，此时需要确定该数据节点在未存储数据 13 之前的校验值，即校验值 b，那么此时校验值 b 被称之为原始校验值。利用校验值 c 和校验值 b，按照设定算法，计算得到该数据节点的更新校验值。

需要说明的是，如果该数据节点之前尚未存储数据，该新增数据作为该数据节点的第一数据，那么在本申请实施例中，该新增数据的校验值可以看做是该数据节点的校验值。

步骤 S140：根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，得到所述区块的校验值。

在本申请实施例中，确定所述区块中包含的多个数据节点对应的树结构，以及确定所述区块中包含的其他数据节点的原始校验值；根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，遍历所述树结构，得到所述树结构的根节点的校验值，并将所述根节点的校验值作为所述区块的校验值。

以图 2 中所示的内容为例。假设在尚未有新增数据存储至图 2 中所示的区块时，图 2 中所示的数据节点 A 对应的校验值为 a、数据节点 B 对应的校验值为 b、数据节点 C 对应的校验值为 c、数据节点 D 对应的校验值为 d 以及数据节点 E 对应的校验值为 e。

此时，新增数据需要存储至数据节点 A，那么校验值 a、校验值 b、校验

值 c、校验值 d 和校验值 e 此时被称之为原始校验值。

当新增数据存储至数据节点 A 之后，数据节点 A 的校验值将发生变化，假设数据节点 A 的校验值由校验值 a 变化为校验值 g，那么校验值 g 可以称之为数据节点 A 的更新校验值。

5 此时，基于数据节点 A 的校验值 g、数据节点 B 的校验值 b、数据节点 C 的校验值 c、数据节点 D 的校验值 d 以及数据节点 E 的校验值 e，按照设定算法，计算得到区块的校验值。

例如：根据数据节点 A、数据节点 B、数据节点 C、数据节点 D 和数据节点 E 在图 2 中所示的树结构，利用数据节点 A 的校验值 g、数据节点 B 的校验值 b、数据节点 C 的校验值 c、数据节点 D 的校验值 d 以及数据节点 E 的校验值 e，计算得到区块的校验值。

10

在获得该区块的校验值之后，可以将其与区块链中其他区块的校验值进行比较。如果比较的结果一致，则说明该区块中存储的数据与其他区块中存储的数据一致，校验通过。如果比较的结果不一致，则说明该区块中存储的数据与其他区块中存储的数据不一致，例如区块数据可能已经丢失或者已经被篡改，不能通过校验。

15

在本申请实施例中，通过增量计算的方式，快速确定区块因新增数据而发生变化的变化后的校验值，缩短了校验值计算的时间，提升了系统计算性能。

通过本申请实施例所提供的技术方案，从包含多个数据节点的区块中选择用于存储新增数据的数据节点；针对所述数据节点，确定所述新增数据的校验值，并根据所述新增数据的校验值，计算所述数据节点的更新校验值；根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，得到所述区块的校验值；其中，所述区块的校验值用于校验所述区块中存储的数据是否与区块链中其他区块中存储的数据一致。与处理所有历史数据的现有技术相比，本申请实施例着重考虑区块的新增数据，对于新增数据的区块，提出了采用增量方式计算校验值，避免了现有技术中当区块中新增数据时利用区块中所有数据计算校验值所导致的时间消耗较长的问题，使在计算新增数据的区

20

25

块的校验值时，所消耗时间与区块中数据量的多少无关，有效缩短了校验值的计算时间，从而提高了区块数据校验的效率。

另一方面，对于未处理的原始数据，如果被篡改，即使根据本申请实施例
5 在计算新增数据的校验值时未发现这种篡改，但是由于区块链的构造和性质，
在区块链的各个区块中执行数据更新时肯定能发现这样的篡改；从而，不会影
响区块数据的最终校验。

图 5 为根据本申请实施例的一种区块数据校验装置的结构示意图。所述区
块数据校验装置包括选择模块 510、确定模块 520 和处理模块 530。

选择模块 510，从包含多个数据节点的区块中选择用于存储新增数据的数
10 据节点；

确定模块 520，针对所述数据节点，确定所述新增数据的校验值，并根据
所述新增数据的校验值，计算所述数据节点的更新校验值；

处理模块 530，根据所述数据节点的更新校验值与所述区块中包含的其他
数据节点的原始校验值，得到所述区块的校验值；

15 其中，所述区块的校验值用于校验所述区块中存储的数据是否与区块链中
其他区块中存储的数据一致。

在本申请的另一个实施例中，所述选择模块 510 从包含多个数据节点的区
块中选择用于存储新增数据的数据节点，包括：

在接收到新增数据时，确定用于存储所述新增数据的区块信息，所述区块
20 信息中包含区块标识和数据节点标识；

根据所述区块标识，从区块链中查找与所述区块标识对应的区块；以及从
查找到的区块中确定与所述数据节点标识对应的数据节点；

将查找到的数据节点作为用于存储所述新增数据的数据节点。

在本申请的另一个实施例中，所述选择模块 510 确定用于存储所述新增数
25 据的区块信息，包括：

对所述新增数据对应的数据内容进行计算，得到确定的区块信息。

在本申请的另一个实施例中，所述确定模块 520 确定所述新增数据的校验

值，包括：

在确定所述数据节点中不存在与所述新增数据的关联数据时，确定所述新增数据对应的字符串；

按照设定算法，对所述字符串进行计算得到所述新增数据的校验值。

- 5 在本申请的另一个实施例中，所述确定模块 520 确定所述新增数据的校验值，包括：

在确定所述数据节点中存在与所述新增数据的关联数据时，分别确定所述关联数据对应的第一字符串和所述新增数据对应的第二字符串；

按照设定方式对所述第一字符串和所述第二字符串进行排序；

- 10 按照设定算法，对排序后得到的字符串进行计算得到所述新增数据的校验值。

在本申请的另一个实施例中，所述确定模块 520 根据所述新增数据的校验值，计算所述数据节点的更新校验值，包括：

确定所述数据节点在未存储所述新增数据之前的原始校验值；

- 15 根据所述原始校验值和所述新增数据的校验值，计算得到所述数据节点的更新校验值。

在本申请的另一个实施例中，所述处理模块 530 根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，得到所述区块的校验值，包括：

- 20 确定所述区块中包含的多个数据节点对应的树结构，以及确定所述区块中包含的其他数据节点的原始校验值；

根据所述数据节点的更新校验值与所述区块中包含的其他数据节点的原始校验值，遍历所述树结构，得到所述树结构的根节点的校验值，并将所述根节点的校验值作为所述区块的校验值。

- 25 需要说明的是，本申请实施例中所记载的区块数据校验装置可以通过软件方式实现，也可以通过硬件方式实现，这里不做具体限定。区块数据校验装置对于新增数据的区块，提出了采用增量方式计算校验值，避免了现有技术中当

区块中新增数据时利用区块中所有数据计算校验值所导致的时间消耗较长的问题，使在计算新增数据的区块的校验值时，所消耗时间与区块中数据量的多少无关，有效缩短了校验值的计算时间，从而提高了区块数据校验的效率。

在 20 世纪 90 年代，对于一个技术的改进可以很明显地区分是硬件上的改进
5 进（例如，对二极管、晶体管、开关等电路结构的改进）还是软件上的改进（对于方法流程的改进）。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件（Programmable
10 Logic Device, PLD）（例如现场可编程门阵列（Field Programmable Gate Array, FPGA））就是这样一种集成电路，其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片 PLD 上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器（logic compiler）”软件来实现，它
15 与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言（Hardware Description Language, HDL），而 HDL 也并非仅有一种，而是有许多种，如 ABEL（Advanced Boolean Expression Language）、AHDL（Altera Hardware Description Language）、Confluence、CUPL（Cornell University Programming Language）、HDCal、JHDL
20 （Java Hardware Description Language）、Lava、Lola、MyHDL、PALASM、RHDL（Ruby Hardware Description Language）等，目前最普遍使用的是 VHDL（Very-High-Speed Integrated Circuit Hardware Description Language）与 Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件
25 电路。

控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该（微）处理器执行的计算机可读程序代码（例如软件

或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit, ASIC)、可编程逻辑控制器和嵌入微控制器的形式,控制器的例子包括但不限于以下微控制器: ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及 Silicone Labs C8051F320, 存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道,除了以纯计算机可读程序代码方式实现控制器以外,完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件,而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至,可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

上述实施例阐明的系统、装置、模块或单元,具体可以由计算机芯片或实体实现,或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的,计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

为了描述的方便,描述以上装置时以功能分为各种单元分别描述。当然,在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入

式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

5 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

10 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器（CPU）、输入/输出接口、网络接口和内存。

15 内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器（RAM）和/或非易失性内存等形式，如只读存储器（ROM）或闪存（flash RAM）。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存（PRAM）、静态随机存取存储器（SRAM）、动态随机存取存储器（DRAM）、
20 其他类型的随机存取存储器（RAM）、只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、快闪记忆体或其他内存技术、只读光盘只读存储器（CD-ROM）、数字多功能光盘（DVD）或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计
25 算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体（transitory media），如调制的数据信号和载波。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排

他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

以上所述仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。