

区块链验证服务管理方法和系统

本申请要求在 2023 年 12 月 20 日提交中国国家知识产权局、申请号为 202311763697.3 的中国专利申请的优先权，发明名称为“区块链质押的方法及系统”的中国专利申请的优先权，在 2024 年 04 月 09 日提交中国国家知识产权局、申请号为 202410427496.4 的中国专利申请的优先权，发明名称为“区块链验证服务管理方法和系统”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及云计算领域，并且更具体地，涉及一种区块链验证服务管理方法和系统。

背景技术

以太坊（Ethereum）是一种基于区块链技术的智能合约平台，以太坊中的区块链节点可以参与质押行为，成为质押节点。具体地，以太坊的用户可以通过验证计算客户端（validator client）核实身份，成为验证用户，处理或者验证平台中的交易信息、向区块链中添加新区块，或者验证新添加区块的有效性，把验证计算应用得到的数据通过至少一个区块链节点广播给区块链中的其他节点。

以太坊用户通常会在云服务系统的用户节点池中部署区块链节点，即用户节点，以加快验证计算，增强广播覆盖速度，并提升多个节点的平均有效性。由于验证计算服务消耗的运算资源较多，用户通常在想要参与质押行为时临时创建用户节点，但是，创建用户节点所需的时间较长，用户节点在目标时段内可能处于不可用状态。此外，在以太坊中，每个节点的软件架构需要周期性维护升级，使它的功能和整个区块链中的协议保持一致。如果执行验证计算服务期间，遇到用户节点的软件架构维护升级等情况，也会导致用户节点在目标时段内处于不可用状态。

发明内容

本申请提供一种区块链验证服务管理方法和系统，以期提高用户节点在区块链中的有效性。

第一方面，本申请实施例提供一种区块链验证服务管理方法，区块链包括多个区块链节点；该方法用于区块链验证服务管理系统，区块链验证服务管理系统包括多个共享节点和多个专用节点，共享节点和专用节点用作区块链节点；共享节点存储有区块链的同步数据；该方法包括：接收来自第一用户的用户节点创建请求，根据多个共享节点中的第一共享节点的数据，创建至少一个专用节点，至少一个专用节点用作第一用户的用户节点；接收来自第一用户的验证计算应用的第一数据；根据第一数据生成第二数据，其中，第二数据包括第一数据、第一用户的标识符和用户节点的标识符；向多个区块链节点发送第二数据。

本申请实施例提供的区块链验证服务管理系统包括多个共享节点，共享节点存储有区块链的同步数据，可以提升用户节点的创建速度，从而提升用户执行验证任务的有效性。

在第一方面一种可能的实现方式中，根据第一数据生成第二数据，包括：在生成第二数据的过程中，若用户节点可用，用户节点根据第一数据生成第二数据。

在第一方面一种可能的实现方式中，根据第一数据生成第二数据，包括：在生成第二数据的过程中，若存在用户节点不可用的第一时段，在第一时段内，第一共享节点根据第一数据生成第二数据的一部分；在用户节点可用的第二时段内，用户节点根据第一数据生成第二数据的另一部分；其中，第二数据包括第一数据、第一用户的标识符、用户节点的标识符和第一共享节点的标识符。

在这样的实现方式中，第一共享节点可以临时代替不可用的用户节点执行验证计算服务，无论用户节点是否可用，整个区块链验证服务管理系统都可以持续接收第一数据，提升第一数据的广播效率和可靠性。

在第一方面一种可能的实现方式中，用户节点不可用，包括：用户节点处于创建过程中，或者，用户节点处于升级维护过程中。

本申请实施例的技术方案覆盖了多种用户节点不可用的情况，区块链验证服务管理系统稳定可靠。

在第一方面一种可能的实现方式中，区块链验证服务管理系统还包括验证计算应用接口，接收来自第一用户的验证计算应用的第一数据，包括：验证计算应用接口接收来自验证计算应用的第一数据；方法还包括：若用户节点可用，验证计算应用接口向用户节点发送第一数据；若用户节点不可用，验证计算应用接口向第一共享节点发送第一数据。

在区块链验证计算服务管理系统中部署验证计算应用接口，使得验证计算应用不需要进行复杂计算来确定用于接收第一数据的节点，第一共享节点也不需要进行复杂计算来确定每个时段是否用于接收第一数据，降低整个系统的复杂度。

在第一方面一种可能的实现方式中，多个共享节点部署于多个区域，向多个区块链节点发送第二数据，包括：向第二共享节点发送第二数据，其中，第二共享节点是与用户节点位于相同的区域的共享节点；第二共享节点向多个区块链节点发送第二数据。

在这样的实现方式中，用户节点可以通过相同区域的共享节点，在本区域广播第二数据。

在第一方面一种可能的实现方式中，第二共享节点向多个区块链节点发送第二数据，包括：第二共享节点向目标区块链节点所属区域的第三共享节点发送第二数据；第三共享节点向目标区块链节点发送第二数据。

在这样的实现方式中，共享节点用于向外部的区块链节点广播验证数据，增大广播覆盖面，减小广播时延，降低用户节点的流量消耗，可以显著提高用户节点执行验证计算任务的成功率。

在第一方面一种可能的实现方式中，区块链验证服务管理系统还包括验证计算应用接口，第二共享节点向多个区块链节点发送第二数据，包括：第二共享节点通过验证计算应用接口向多个区块链节点发送第二数据。

在这样的实现方式中，用户节点能够充分利用多个验证计算应用接口之间的网络性能，向多个共享节点广播第二数据，降低内部广播第二数据的时延。

在第一方面一种可能的实现方式中，区块链验证服务管理系统还包括人机交互界面，人机交互界面用于部署多个共享节点和多个专用节点；该方法包括：人机交互界面接收来自第一用户的用户节点创建请求；在用户节点创建完成的情况下，向第一用户输出创建成功响应。

人机交互界面可以接收来自用户的用户节点创建请求，或者接收管理员设置的配置信息，使区块链验证服务管理系统易于维护。

第二方面，本申请实施例提供一种区块链验证服务管理系统，区块链包括多个区块链节点；区块链验证服务管理系统包括多个共享节点和多个专用节点，共享节点和专用节点用作区块链节点；共享节点存储有区块链的同步数据；区块链验证服务管理系统用于：接收来自第一用户的用户节点创建请求，根据多个共享节点中的第一共享节点的数据，创建至少一个专用节点，至少一个专用节点用作第一用户的用户节点；接收来自第一用户的验证计算应用的第一数据；根据第一数据生成第二数据，其中，第二数据包括第一数据、第一用户的标识符和用户节点的标识符；向多个区块链节点发送第二数据。

在第二方面一种可能的实现方式中，在生成第二数据的过程中，若用户节点可用，用户节点用于根据第一数据生成第二数据。

在第二方面一种可能的实现方式中，在生成第二数据的过程中，若存在用户节点不可用的第一时段，在第一时段内，第一共享节点用于根据第一数据生成第二数据的一部分；在用户节点可用的第二时段内，用户节点用于根据第一数据生成第二数据的另一部分；其中，第二数据包括第一数据、第一用户的标识符、用户节点的标识符和第一共享节点的标识符。

在第二方面一种可能的实现方式中，用户节点不可用，包括：用户节点处于创建过程中，或者，用户节点处于升级维护过程中。

在第二方面一种可能的实现方式中，区块链验证服务管理系统还包括验证计算应用接口，验证计算应用接口用于：接收来自验证计算应用的第一数据；若用户节点可用，向用户节点发送第一数据；若用户节点不可用，向第一共享节点发送第一数据。

在第二方面一种可能的实现方式中，多个共享节点部署于多个区域，用户节点和/或第一共享节点用于向第二共享节点发送第二数据，其中，第二共享节点是与用户节点位于相同的区域的共享节点；第二共享节点用于向多个区块链节点发送第二数据。

在第二方面一种可能的实现方式中，第二共享节点用于向目标区块链节点所属区域的第三共享节点发送第二数据；第三共享节点用于向目标区块链节点发送第二数据。

在第二方面一种可能的实现方式中，区块链验证服务管理系统还包括验证计算应用接口，第二共享节点用于通过验证计算应用接口向多个区块链节点发送第二数据。

在第二方面一种可能的实现方式中，区块链验证服务管理系统还包括人机交互界面，人机交互界面用于：部署多个共享节点和多个专用节点；在用户节点创建完成的情况下，向第一用户输出创建成功响应。

第三方面，本申请实施例提供一种计算设备集群，包括至少一个计算设备，每个计算设备包括处理器和存储器；所述至少一个计算设备的处理器用于执行所述至少一个计算设备的存储器中存储的指令，以使得所述计算设备集群执行如第一方面任一实现方式所述的方法。

第四方面，本申请实施例提供一种计算机可读存储介质，包括计算机程序指令，当所述计算机程序指令由区块链验证服务管理系统执行时，所述区块链验证服务管理系统执行如第一方面任一实现方式所述的方法。

第五方面，本申请实施例提供一种计算机程序产品，当所述计算机程序产品在区块链验证服务管理系统上运行时，使得所述区块链验证服务管理系统执行如第一方面任一实现方式所述的方法。

附图说明

图1是本申请实施例的一种应用场景示意图。

图2是本申请实施例的一种区块链验证服务管理系统示意性框图。

图3是本申请实施例的一种区块链验证服务管理系统示意性框图。

图4是本申请实施例的一种区块链验证服务管理系统的人机交互界面的示意性框图。

图5是本申请实施例的一种区块链验证服务管理方法示意性流程图。

图6是本申请实施例的一种接收验证数据的示意性流程图。

图7是本申请实施例的一种验证数据应用接口的示意性框图。

图8是本申请实施例中用户节点向第二共享节点发送第二数据的示意性流程图。

图9是本申请实施例的一种区块链质押装置的示意性框图。

图10是本申请实施例的一种控制器的示意性框图。

具体实施方式

本申请将围绕包括多个设备、组件、模块等的系统来呈现各个方面、实施例或特征。应当理解和明白的是，各个系统可以包括另外的设备、组件、模块等，并且/或者可以并不包括结合附图讨论的所有设备、组件、模块等。此外，还可以使用这些方案的组合。

另外，在本申请实施例中，“示例的”、“例如”等词用于表示作例子、例证或说明。本申请中被描述为“示例”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言，使用示例一词旨在以具体方式呈现概念。

本申请实施例中，“相应的(corresponding, relevant)”和“对应的(corresponding)”有时可以混用，应当指出的是，在不强调其区别时，其所要表达的含义是一致的。

本实施例涉及一些新的、按照功能或逻辑划分的功能模块或单元，如验证计算应用接口(validating hub, VH)以及其子模块，本领域普通技术人员可知，随着网络架构的演变和新业务场景的出现，具有类似功能和定义的模块或者单元可以具有不同的名称，名称的不同不应视为对本申请技术方案的限制。

本申请实施例描述的业务场景是为了更加清楚地说明本申请实施例的技术方案，并不构成对于本申请实施例提供的技术方案的限定，本领域普通技术人员可知，随着网络架构的演变和新业务场景的出现，本申请实施例提供的技术方案对于类似的技术问题，同样适用。

下面对本领域常用技术术语作简要介绍。

云计算(cloud computing)是一种通过网络以服务的方式提供动态可伸缩的虚拟化的资源的计算模式。平台即服务(platform as a service, PaaS)是云计算多租户的主要模式之一。PaaS平台上会注册不同的租户，使用PaaS平台的资源和服务。PaaS平台的共有云服务可以为公有云租户提供区块链即服务(blockchain as a service, BaaS)。

在云计算领域，区域(region)和可用区(availability zone, AZ)是用来划分数据中心和服务器资源的概念。区域是指一个地理区域，通常包含多个数据中心，而每个数据中心又包含多个AZ。每个AZ通

常是一个独立的数据中心，具有自己的电力、网络和物理设备。例如，全球区域可以分为亚洲区域、美洲区域、欧洲区域，国内区域可以分为华东区域、华北区域、华中区域等。

区块链（blockchain）是指通过基于密码学技术设计的共识机制方式，在对等网络中多个节点共同维护一个持续增长，由时间戳和有序记录数据块所构建的链式列表账本的分布式数据库技术。该技术方案中，参与区块链的任意多个区块链节点把一段时间系统内全部信息交流的数据通过密码学算法计算和记录到一个数据块（block），并且生成该数据块的指纹，用于链接（chain）和校验下个数据块，系统中的所有节点共同验证数据的真实性和有效性。

块高（block height）是指区块链网络上已经确认的最新版块的高度。每当新的区块被添加到区块链上时，块高就会增加。因此，块高可以用来表示区块链网络的当前状态和进展。当一个节点正常运行并与区块链网络同步时，它会不断地接收新的区块并更新自己的块高。因此，通过检查节点的块高可以判断节点是否在正常运行并与网络同步。如果一个节点长时间没有更新块高，或者其块高与其他节点的块高存在明显的偏差，可能表示该节点出现了问题，无法正常运行或者与网络失去了连接，即处于不可用状态。

通用类 PoS 链是指一种基于权益证明（proof of stake, PoS）共识机制的区块链网络，其主要特点是支持多种类型的应用和智能合约。以太坊（Ethereum）是一种基于通用类 PoS 链，在以太坊中，每个时段会产生一个新区块，以太坊中的区块链节点可以参与质押行为，成为质押节点。具体地，以太坊的用户拥有数字签名、验证计算客户端和多个验证节点，可以通过验证计算客户端（validator client）核实身份，成为验证用户。随后验证用户通过验证计算客户端中的多个验证计算应用（validator）处理或者验证平台中的交易信息、向区块链中添加新区块，或者验证新添加区块的有效性，把验证计算应用得到的数据通过至少一个区块链节点广播给区块链中的其他节点。

在本申请实施例中，参与质押行为的用户可称为验证用户或者质押用户，参与质押行为可称为执行验证服务，处理或者验证得到的数据可称为质押数据或者验证数据，执行验证服务的节点可称为质押节点或用户节点，验证计算应用可称为验证者。具体名称的混用不应被视为对本申请技术方案的限制。后续实施例用以太坊为例说明应用于通用类 PoS 链的方法、装置、系统等，不应理解为对本申请技术方案的限制。

具体地，以太坊中质押节点的选取规则是通过随机性来确定的。以太坊使用了 RANDAO 算法来实现随机性。这个算法将来自区块提议者的一个哈希值与一个随每个区块更新的种子（seed）混合起来，生成一个随机数，用于在所有质押节点中选择一个特定的质押节点执行验证服务。因此，区块链系统中的活动通常涉及全球区域。

具体地，区块链节点可以是硬件、虚拟硬件或者容器，每个区块链节点具有 IP 地址和数字签名。如图 1 所示，多个验证计算客户端可以从区块链中的至少一个任意节点获取当前区块链的最新数据，执行具体的验证计算，并把验证数据发送至多个用户节点，多个用户节点分别向区块链中的大部分节点广播验证数据，以证明质押行为的有效性。其中，一个验证用户可以为每个用户节点部署一个验证计算客户端，每个验证计算客户端包括多个可以并行计算的验证计算应用，以加快验证计算的速度。

以太坊用户通常会在云服务系统的用户节点池中部署区块链节点，即用户节点，以加快验证计算，增强广播覆盖速度，并提升多个节点的平均有效性。由于验证计算服务消耗的运算资源较多，用户通常在想要参与质押行为时临时创建用户节点，但是，创建用户节点所需的时间较长，用户节点在目标时段内可能处于不可用状态。此外，在以太坊中，每个节点的软件架构需要周期性维护升级，使它的功能和整个区块链中的协议保持一致。如果执行验证计算服务期间，遇到用户节点的软件架构维护升级等情况，也会导致用户节点在目标时段内处于不可用状态。

进一步地，如果用户节点都位于同一个区域，可能导致向其他区域广播数据的网络延时高，数据传输流量低，其他区域的多数节点在当前时段发生消息丢失（miss），引起多次验证任务失败，节点在区块链中的有效性低。与此同时，用户节点在每个时段都要向区块链中的所有节点广播验证数据，流量消耗大。

有鉴于此，本申请实施例提供一种区块链验证服务管理方法和系统，以期提高用户节点在区块链中的有效性。

下面结合图 2 说明本申请提供的一种区块链验证服务管理系统。如图 2 所示，区块链包括多个区块链节点，区块链验证服务管理方法用于所述系统，区块链验证服务管理系统的节点池包括多个共享节点

和多个用户节点，共享节点和用户节点用作区块链节点。其中，每个共享节点池分别周期性地同步外部区块链节点中的区块链数据，并存储在自身的存储设备中。

在一些实施例中，验证计算客户端不直接向用户节点发送验证数据，而是向验证计算应用接口发送验证数据。当用户节点可用时，验证计算应用接口向用户节点发送验证数据；当用户节点不可用时，例如用户节点处于创建过程中，或者处于升级维护过程中，则验证计算应用接口向共享节点发送验证数据，共享节点用作用户节点。此外，可以直接从共享节点获取最新的区块链数据用于初始化用户节点，提升用户节点的创建速度。

本申请实施例提供的区块链验证服务管理系统包括多个共享节点，共享节点存储有区块链的同步数据，可以提升用户节点的创建速度，从而提升用户执行验证任务的有效性。

上述方案适用于在用户节点所属区域建立单个节点池的场景，该方案也可以进一步地扩展为在多个区域建立多个节点池，在多个区域提升用户节点的可用性，并更精确地管理用户节点和共享节点。下面结合图 3 所示的区块链验证服务管理系统示意性框图，说明本实施例的一个区块链验证服务管理系统中部分实体的逻辑关系。

本实施例提供一种区块链验证服务管理系统，在多个区域设有数据中心。应理解，本申请的区域主要是指全球范围内的区域。区块链验证服务管理系统的管理员可以在系统中进行配置，在各区域的多个主机/虚拟机/容器中分别创建多个区块链节点，区块链节点可以是虚拟机/容器，部署在数据中心中。具体地，本实施例所述的区块链节点可以分为共享节点和专用节点，每个区域的数据中心都部署有多个共享节点，至少一个区域的数据中心部署有多个专用节点。其中，共享节点和专用节点都包括软件层和存储设备，专用节点包括软件层和动态容量存储设备。

进一步地，每个区域中的所有共享节点分别集合在一起，在每个区域分别形成一个共享节点池；每个区域的所有专用节点集合在一起，在至少一个区域分别形成专用节点池。

在一个实施例中，如图 4 所示，区块链验证服务管理系统包括人机交互界面，人机交互界面包括管理员界面，区块链验证服务管理系统的管理员可以通过管理员界面，部署共享节点池和专用节点池，或者管理部署在各个区域的共享节点池和专用节点池。具体地，可以改变区域的划分方式，调节共享节点或者专用节点的数量，设置节点的内存资源、存储容量或网络带宽等。

下面结合图 5 所示的示意性流程图，说明区块链验证服务管理方法的流程。

S510，接收来自用户的用户节点创建请求，根据多个共享节点中的第一共享节点的数据，创建至少一个专用节点用作用户节点。

当用户想要创建用户节点并参与区块链的质押行为时，在区块链验证服务管理系统中请求创建用户节点。区块链验证服务管理系统在至少一个区域的专用节点池中创建专用节点作为用户节点，也就是说，专用节点池可以包括多个不同用户的用户节点。

应理解，区块链中的每个节点会周期性地从其他节点获取当前时刻的最新区块链数据，并存储有区块链的同步数据，以保证各节点之间数据的一致性。这种同步数据的周期可以用时段（period）和时隙（slot）来描述。具体地，一个时段包括多个时隙，根据区块链的协议规定，一部分数据在每个时隙都需要进行同步，而另一部分数据只需要在每个时段进行同步，每个节点至少需要在每个时段内进行一次同步。

为便于描述，本申请将区块链验证服务管理系统中的共享节点或者专用节点称为内部节点，将区块链验证服务管理系统之外的其他区块链节点称为外部节点。此外，后续实施例中以一个时段的时长为 384 秒、一个时隙的时长为 12 秒为例进行说明。

在区块链验证服务管理系统中，每个共享节点池分别周期性同步外部节点中的区块链数据，其中，一个共享节点池中的每个节点同步数据的周期相同，但是同步数据的时刻不同。在一个实施例中，一个共享节点池的同步周期为一个时段的时长，即 384 秒，部分共享节点在第 $384K+12$ 秒进行同步，部分共享节点在第 $384K+24$ 秒进行同步，部分共享节点在第 $384K+36$ 秒进行同步，其中 K 是自然数，以此类推。当该共享节点池中的共享节点的数量大于或等于 32 个，该共享节点池中的至少一个共享节点的数据与当前时隙中外部节点的最新区块链数据保持一致，这些共享节点存储有区块链的最新同步数据，称为第一共享节点。

可选地，共享节点每次进行数据同步后，可以把成功获取的区块链数据存入备份数据存储设备，进一步提升数据的安全性。

在一个实施例中，区块链验证服务管理系统在第 384 K+15 秒创建一个用户节点，该用户节点可以采用当前区域的共享节点池中的第 384 K+12 秒的区块链数据进行初始化。

共享节点池中的多个共享节点在区块链时段的不同时段同步区块链数据，区块链验证服务管理系统可以根据当前区域的共享节点池中的最新区块链数据创建用户节点，初始化操作的时延低，有效性高。

在一个实施例中，如图 4 所示，区块链验证服务管理系统的人机交互界面包括用户界面，用户可以通过用户界面发起用户节点创建请求，对用户节点进行升级维护、查看日志，或者接入用户节点以查看区块链中的各种信息。每个用户分别管理自己的用户节点，例如，用户 411 可以管理用户节点 421，用户 412 可以管理用户节点 422。

进一步地，区块链验证服务管理系统在用户节点创建完成的情况下向用户输出创建成功响应。

人机交互界面可以接收来自用户的用户节点创建请求，或者接收管理员设置的配置信息，使区块链验证服务管理系统易于维护。

S520，接收来自验证计算应用的第一数据。

在一个实施例中，区块链验证服务管理系统中部署有验证计算应用接口（validating hub，VH）。VH 可以用于接收来自验证计算应用的第一数据，向用户节点和/或第一共享节点发送第一数据，接收来自一个验证计算应用接口的数据，或者向一个验证计算应用接口发送数据。

具体地，VH 是具有高性能网络通信组件的模块，可以进行大流量数据的转发，例如同时接收来自多个验证计算客户端的第一数据，并转发给指定的内部节点。应理解，上述组件和模块仅是一种逻辑上的划分，可以是软件、硬件或者软硬件的结合。

应理解，验证计算客户端中的验证计算应用需要获取当前最新的区块链数据以进行验证计算，在一个实施例中，验证计算客户端可以通过 VH，从该 VH 所属区域的内部节点获取区块链数据；在另一个实施例中，验证计算客户端可以通过一个已经建立连接的外部节点获取区块链数据。为便于描述，后续实施例中不再区分验证计算应用和验证计算客户端，统一用验证计算应用进行描述。

本申请实施例中，验证计算应用和 VH 建立长连接，VH 为长连接提供高可靠性。长连接是指在通信双方建立连接后，保持连接状态一段时间，并进行多次数据传输。在长连接中，一旦连接建立，通信双方可以随时发送和接收数据，而不需要每次通信都重新建立连接。

为便于描述，称验证计算应用和 VH 之间的连接为外部连接，称内部节点和 VH 之间的连接为内部连接。

S521，在用户节点可用的情况下，用户节点接收来自验证计算应用的第一数据。

前述实施例中，创建用户节点的过程需要数分钟。若第一用户节点处于尚未创建完成的状态，则认为第一用户节点处于不可用状态。此外，如果第一用户节点的软件层正在进行维护升级，也认为第一用户节点处于不可用状态。

为方便描述，称与验证计算应用位于相同区域的 VH 为第一验证计算应用接口，称与用户节点位于相同区域的 VH 为第二验证计算应用接口。下面结合图 6 所示的示意性流程图对接收第一数据进行详细说明。

在一个实施例中，如图 6 中的(a)所示，第二验证计算应用接口接收来自验证计算应用的第一数据，第二验证计算应用接口向第一用户节点发送第一数据。

在一个实施例中，如图 6 中的(b)所示，第一验证计算应用接口接收来自验证计算应用的第一数据，第二验证计算应用接口接收来自第一验证计算应用接口的第一数据，第二验证计算应用接口向第一用户节点发送第一数据。

这样的实现方式中，验证计算应用和第一验证计算应用接口之间的网络质量高，能够充分利用多个验证计算应用接口之间的网络性能，提高对内发送第一数据的速率。

在上述实施例中，在第一用户节点可用的情况下，VH 向第一用户节点发送第一数据。然而，在一些实施例中，当第一用户节点处于可用状态时，第一用户节点接收验证计算应用直接向第一用户节点发送的第一数据，区块链验证服务管理系统中的 VH 是可选的。

S522，在用户节点不可用的情况下，第一共享节点接收来自验证计算应用的第一数据。

在一些实施例中，每个区域的共享节点池包括至少一个第一共享节点，第一共享节点可以用作第二用户节点。具体地，当第二用户节点处于不可用状态时，与第二用户节点位于相同区域的 VH 对本区域的所有共享节点的工作状态进行分析，按照特定的规则选择一个第一共享节点，把传输第一数据的网络

流量从不可用的第二用户节点切换至该第一共享节点，使第一共享节点代替第二用户节点接收第一数据，并执行第二用户节点的其他功能，如同步区块链数据、为用户接入区块链浏览器等。

具体地，按照特定的规则选择第一共享节点，可以是在当前区域的所有第一共享节点中选择当前工作负载最低的第一共享节点，工作负载包括内存使用率、网络流量使用率等参数；也可以是选择与用户节点位于相同 AZ 的第一共享节点；还可以选择当前区域的、在区块链中有效性较高的第一共享节点，还可以根据多个共享节点的块高选取第一共享节点等，本实施例对此不作限定。

在上述实施例中，在第二用户节点不可用的情况下，VH 向第一共享节点发送第一数据。然而，在一些实施例中，当第二用户节点处于不可用状态时，第一共享节点接收验证计算应用直接向第一共享节点发送的第一数据，区块链验证服务管理系统中的 VH 是可选的。

进一步地，可以通过检测块高的方法确认第二用户节点是否可用。

进一步地，VH 可以通过消息队列（message queue）、心跳检测（heartbeat detection）、服务注册与发现（service registration and discovery）等方法把传输第一数据的网络流量从不可用的用户节点切换至第一共享节点。

应理解，验证计算应用向第一共享节点发送第一数据的过程与向第一用户节点发送第一数据的过程类似。

应理解，当第二用户节点恢复可用，VH 可以把传输第一数据的网络流量从第一共享节点切换至第二用户节点，使第二用户节点继续接收第一数据。

在一些实施例中，区块链验证服务管理系统中没有设置 VH，则选择第一共享节点、判断第二用户节点是否可用、切换网络流量等步骤可以由用户部署的其他网络管理系统实现。

在一些实施例中，在第二用户节点可用的情况下，第一共享节点也可以接收来自验证计算应用的第一数据，例如第二用户节点的网络带宽较小，或者计算资源偏少，独立完成验证计算任务耗时较长，而空闲的第一共享节点数量较多，则第一共享节点也可以和第二用户节点共同执行验证计算任务。

在区块链验证计算服务管理系统中部署验证计算应用接口，使得验证计算应用不需要进行复杂计算来确定用于接收第一数据的节点，第一共享节点也不需要进行复杂计算来确定每个时段是否用于接收第一数据，降低整个系统的复杂度。

在这样的实现方式中，第一共享节点可以临时代替不可用的用户节点执行验证计算服务，无论用户节点是否可用，整个区块链验证服务管理系统都可以持续接收第一数据，提升第一数据的广播效率和可靠性。

S530，根据第一数据生成第二数据。

第一数据是用户的验证计算应用得到的数据，包括用户的数字签名，不包括区块链节点的数字签名。用户节点需要把第一数据发送给用户节点进行进一步的转换和处理，生成第二数据，例如追加用户节点的数字签名，修改时间戳等，然后把第二数据广播给区块链中的大部分节点。

具体地，第一数据可以是区块链系统中产生的待验证数据，例如智能合约的输入参数、交易请求或其他相关信息。第二数据可以是经过计算、验证或处理后的结果数据，例如智能合约的执行结果、交易的确认信息或其他相关数据，用于更新区块链网络的状态或执行智能合约的逻辑。

S531，若用户节点可用，用户节点根据第一数据生成第二数据。

应理解，每个区块链节点都有对应的数字签名，若生成第二数据的整个过程中用户节点可用，最终得到的第二数据包括用户的数字签名和用户节点的数字签名。应理解，数字签名是标识符的一种。

S532，若存在用户节点不可用的时段，用户节点和第一共享节点分别根据第一数据生成第二数据的一部分。

具体地，在生成第二数据的过程中，若存在用户节点不可用的第一时段，在第一时段内，第一共享节点根据第一数据生成第二数据的一部分；在用户节点可用的第二时段内，用户节点根据第一数据生成第二数据的另一部分。其中，第二数据包括用户节点的数字签名和第一共享节点的数字签名。

S540，向外部节点发送第二数据。

本实施例以用户节点为例，说明向外部节点发送第二数据的过程，应理解，本实施例中的用户节点执行的功能也可以由第一共享节点完成。在后续实施例中，“广播”和“发送”的含义类似，可以混合使用，不构成对本申请技术方案的限制。

S541，获取外部节点的信息。

在一个实施例中，每个区域的共享节点池周期性地获取本区域的外部节点信息，本申请的外部节点是指该周期性获取得到的外部节点。

具体地，可以通过区块链浏览器、节点公开信息或者节点发现协议周期性地获取本区域的外部节点信息。一些区块链浏览器提供了节点的 IP 地址信息。通过查看区块链浏览器上的节点信息，可以获取一部分节点的 IP 地址信息；有些节点可能会通过官方网站或社交媒体账号公开其 IP 地址信息；一些区块链网络使用节点发现协议来帮助节点互相发现和连接，通过参与节点发现协议，可以获取一部分节点的 IP 地址信息。

S542，用户节点和/或第一共享节点向第二共享节点发送第二数据。

在一个实施例中，在每个共享节点池中分别选择多个共享节点用于向外部节点发送第二数据。其中，第二共享节点是与用户节点位于相同的区域的共享节点，第二共享节点用于向外部节点发送第二数据。

在另一个实施例中，如图 7 所示，验证计算应用接口包括对外连接模块、共享节点路由模块和专用节点路由模块。具体地，对外连接模块用于接收来自验证计算应用的第一数据，接收来自一个 VH 的数据，或者向一个 VH 发送数据；共享节点路由模块用于向共享节点发送第一数据或者第二数据，专用节点路由模块用于向用户节点发送第一数据，或者接收用户节点发送的第二数据。

进一步地，区块链验证服务管理系统的管理员可以通过人机交互界面对上述各模块的参数进行配置。

用户节点需要向第二共享节点发送第二数据，从而使第二共享节点向外部节点广播第二数据。首先结合图 8 所示的示意性流程图说明用户节点向第二共享节点发送第二数据的流程。

在一些实施例中，如图 8 中的(a)所示，用户节点向第二共享节点发送第二数据，第二共享节点向外部节点发送第二数据。

在这样的实现方式中，用户节点可以通过相同区域的共享节点，在本区域广播第二数据。

在一些实施例中，如图 8 中的(b)所示，第二验证计算应用接口接收来自用户节点的第二数据，向多个第二共享节点发送第二数据。

其次，结合图 8 说明用户节点向第三共享节点发送第二数据的流程，其中，第三共享节点是与用户节点位于不同区域的共享节点，第三共享节点用于向外部节点发送第二数据。在一些实施例中，如图 8 中的(a)所示，第三验证计算应用接口接收来自第二验证计算应用接口的第二数据，分别向目标区块链节点所属区域的第三共享节点发送第二数据。其中，第三验证计算应用接口是与第二验证计算应用接口位于不同区域的验证计算应用接口。

在一些实施例中，部分区域不存在第三共享节点，则第二共享节点向这些区域的外部节点发送第二数据。

这样的实现方式中，用户节点能够充分利用多个验证计算应用接口之间的网络性能，向多个共享节点广播第二数据，降低内部广播第二数据的时延。

在一些实施例中，区块链验证服务管理系统中没有设置 VH，用户节点可以直接向所有第二共享节点广播第二数据。

S543，第二共享节点向外部节点发送第二数据。

在一些实施例中，如图 8 中的(a)所示，一个第二共享节点向与该第二共享节点位于相同区域的多个外部节点发送第二数据。在另一些实施例中，一个第二共享节点向位于任意区域的多个外部节点发送第二数据。

这样的实现方式中，共享节点用于向外部的区块链节点广播验证数据，增大广播覆盖面，减小广播时延，降低用户节点的流量消耗，可以显著提高用户节点执行验证计算任务的成功率。

S544，用户节点向外部节点发送第二数据。

在一些实施例中，用户节点直接向外部节点发送第二数据。当因为故障或者受到网络攻击等原因，导致部分区域的大量共享节点都处于不可用状态时，用户节点仍然可以根据已获取的外部节点信息，向这些区域的外部节点发送第二数据。

本申请实施例提供的区块链验证服务管理系统，在多个区域部署预创建包括多个共享节点的共享节点池，共享节点用于加快用户节点的创建，以及临时代替不可用的用户节点执行验证计算服务；共享节点还用于向外部的区块链节点广播验证数据，增大广播覆盖面，减小广播时延，降低用户节点的流量消耗。这两种功能可以显著提高用户节点在区块链中的有效性。

图 9 是本申请实施例的区块链验证服务管理的装置的示意性框图。图 9 所示的装置 900 可以用于执行

本申请区块链验证服务管理的方法。

如图 9 所示, 该装置包括获取单元 910、处理单元 920 和输出单元 930。这里的术语“单元”可以通过软件和/或硬件形式实现, 对此不作具体限定。

例如, “单元”可以是实现上述功能的软件程序、硬件电路或二者结合, 可以包括运行在计算实例上的代码。示例性的, 接下来以处理单元为例, 介绍处理单元的实现方式。类似的, 获取单元以及输出单元的实现方式可以参考处理单元的实现方式。

具体地, 处理单元可以包括运行在多个主机/虚拟机/容器上的代码。需要说明的是, 用于运行该代码的多个主机/虚拟机/容器可以分布在相同的区域(region)中, 也可以分布在不同的 region 中。进一步地, 用于运行该代码的多个主机/虚拟机/容器可以分布在相同的可用区(availability zone, AZ)中, 也可以分布在不同的 AZ 中, 每个 AZ 包括一个数据中心或多个地理位置相近的数据中心。其中, 通常一个 region 可以包括多个 AZ。

同样, 用于运行该代码的多个主机/虚拟机/容器可以分布在同一个虚拟私有云(virtual private cloud, VPC)中, 也可以分布在多个 VPC 中。其中, 通常一个 VPC 设置在一个 region 内, 同一 region 内两个 VPC 之间, 以及不同 region 的 VPC 之间跨区通信需在每个 VPC 内设置通信网关, 经通信网关实现 VPC 之间的互连。

处理单元作为硬件功能单元的一种举例, 处理单元可以包括至少一个计算设备, 如服务器等。或者, 处理单元也可以是利用专用集成电路(application-specific integrated circuit, ASIC)实现、或可编程逻辑器件(programmable logic device, PLD)实现的设备等。其中, 上述 PLD 可以是复杂程序逻辑器件(complex programmable logical device, CPLD)、现场可编程门阵列(field-programmable gate array, FPGA)、通用阵列逻辑(generic array logic, GAL)或其任意组合实现。

处理单元包括的多个计算设备可以分布在相同的 region 中, 也可以分布在不同的 region 中。处理单元包括的多个计算设备可以分布在相同的 AZ 中, 也可以分布在不同的 AZ 中。同样, 处理单元包括的多个计算设备可以分布在同一个 VPC 中, 也可以分布在多个 VPC 中。其中, 所述多个计算设备可以是服务器、ASIC、PLD、CPLD、FPGA 和 GAL 等计算设备的任意组合。

因此, 在本申请的实施例中描述的各示例的模块, 能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行, 取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能, 但是这种实现不应认为超出本申请的范围。

本申请还提供一种控制器 1000。如图 10 所示, 控制器 1000 包括: 处理器 1004 和通信接口 1008。进一步地, 控制器 1000 还可以包括总线 1002 和存储器 1006。应理解, 总线 1002 和存储器 1006 为可选项。处理器 1004、存储器 1006 和通信接口 1008 之间通过总线 1002 通信。示例性地, 该控制器 1000 可以为计算设备或者计算设备中用于实现本申请实施例的方法的装置。控制器 1000 可以是服务器或终端设备。应理解, 本申请不限定控制器 1000 中的处理器、存储器的个数。

总线 1002 可以是外设部件互连标准(peripheral component interconnect, PCI)总线或扩展工业标准结构(extended industry standard architecture, EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示, 图 10 中仅用一条线表示, 但并不表示仅有一根总线或一种类型的总线。总线 1002 可包括在控制器 1000 各个部件(例如, 存储器 1006、处理器 1004、通信接口 1008)之间传送信息的通路。

处理器 1004 可以包括中央处理器(central processing unit, CPU)、图形处理器(graphics processing unit, GPU)、微处理器(micro processor, MP)或者数字信号处理器(digital signal processor, DSP)等处理器中的任意一种或多种。

存储器 1006 可以包括易失性存储器(volatile memory), 例如随机存取存储器(random access memory, RAM)。存储器 1006 还可以包括非易失性存储器(non-volatile memory), 例如只读存储器(read-only memory, ROM), 快闪存储器, 机械硬盘(hard disk drive, HDD)或固态硬盘(solid state drive, SSD)。

存储器 1006 中存储有可执行的程序代码, 处理器 1004 执行该可执行的程序代码以分别实现前述获取单元和处理单元的功能, 从而实现区块链验证服务管理的方法。也即, 存储器 1006 上存有用于区块链验证服务管理的方法的指令。

通信接口 1008 使用例如但不限于网络接口卡、收发器一类的收发模块, 来实现控制器 1000 与其他

设备或通信网络（例如，多个服务器）之间的通信。通信接口也可以称为接口电路。

本申请实施例还提供了一种计算设备集群。该计算设备集群包括控制器 1000 和多台计算设备。计算设备可以是服务器，例如是中心服务器、边缘服务器，或者是本地数据中心中的本地服务器。在一些实施例中，计算设备也可以是台式机、笔记本电脑或者智能手机等终端设备。该多台计算设备可以为前文中的多个服务器。该计算设备集群可用于实现区块链验证服务管理系统。

在一种可能的实现方式中，该控制器 1000 为该多台计算设备中的一个计算设备或者为该计算设备中的用于实现前文的方法的装置。

在另一种可能的实现方式中，该控制器 1000 为该多台计算设备以外的其他计算设备或者为其他计算设备中的用于实现前文的方法的装置。

本申请实施例还提供了一种包含指令的计算机程序产品。所述计算机程序产品可以是包含指令的，能够运行在计算设备上或被储存在任何可用介质中的软件或程序产品。当所述计算机程序产品在至少一个计算设备上运行时，使得至少一个计算设备执行本申请实施例中的方法。

本申请实施例还提供了一种计算机可读存储介质。所述计算机可读存储介质可以是计算设备能够存储的任何可用介质或者是包含一个或多个可用介质的数据中心等数据存储设备。所述可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘）等。该计算机可读存储介质包括指令，所述指令指示计算设备执行本申请实施例中的方法，或指示计算设备执行本申请实施例中的方法。

应理解，在本申请的各种实施例中，上述各过程的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不应对本申请实施例的实施过程构成任何限定。

本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统、装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（read-only memory, ROM）、随机存取存储器（random access memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到的变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以所述权利要求的保护范围为准。

权 利 要 求 书

1. 一种区块链验证服务管理方法，其特征在于，所述区块链包括多个区块链节点；所述方法用于区块链验证服务管理系统，所述区块链验证服务管理系统包括多个共享节点和多个专用节点，所述共享节点和所述专用节点用作区块链节点；所述共享节点存储有所述区块链的同步数据；

所述方法包括：

接收来自第一用户的用户节点创建请求，根据所述多个共享节点中的第一共享节点的数据，创建至少一个专用节点，所述至少一个专用节点用作所述第一用户的用户节点；

接收来自所述第一用户的验证计算应用的第一数据；

根据所述第一数据生成第二数据，其中，所述第二数据包括所述第一数据、所述第一用户的标识符和所述用户节点的标识符；

向所述多个区块链节点发送所述第二数据。

2. 根据权利要求1所述的方法，其特征在于，所述根据所述第一数据生成第二数据，包括：

在生成所述第二数据的过程中，若所述用户节点可用，所述用户节点根据所述第一数据生成所述第二数据。

3. 根据权利要求1或2所述的方法，其特征在于，所述根据所述第一数据生成第二数据，包括：

在生成所述第二数据的过程中，若存在所述用户节点不可用的第一时段，在所述第一时段内，所述第一共享节点根据所述第一数据生成所述第二数据的一部分；

在所述用户节点可用的第二时段内，所述用户节点根据所述第一数据生成所述第二数据的另一部分；其中，所述第二数据包括所述第一数据、所述第一用户的标识符、所述用户节点的标识符和所述第一共享节点的标识符。

4. 根据权利要求3所述的方法，其特征在于，所述用户节点不可用，包括：

所述用户节点处于创建过程中，或者，所述用户节点处于升级维护过程中。

5. 根据权利要求1至4中任一项所述的方法，其特征在于，所述区块链验证服务管理系统还包括验证计算应用接口，所述接收来自所述第一用户的验证计算应用的第一数据，包括：

所述验证计算应用接口接收来自所述验证计算应用的所述第一数据；

所述方法还包括：

若所述用户节点可用，所述验证计算应用接口向所述用户节点发送所述第一数据；

若所述用户节点不可用，所述验证计算应用接口向所述第一共享节点发送所述第一数据。

6. 根据权利要求1至5中任一项所述的方法，其特征在于，所述多个共享节点部署于多个区域，所述向所述多个区块链节点发送所述第二数据，包括：

向第二共享节点发送所述第二数据，其中，所述第二共享节点是与所述用户节点位于相同的区域的共享节点；

所述第二共享节点向所述多个区块链节点发送所述第二数据。

7. 根据权利要求6所述的方法，其特征在于，所述第二共享节点向所述多个区块链节点发送所述第二数据，包括：

所述第二共享节点向目标区块链节点所属区域的第三共享节点发送所述第二数据；

所述第三共享节点向所述目标区块链节点发送所述第二数据。

8. 根据权利要求6或7所述的方法，其特征在于，所述区块链验证服务管理系统还包括验证计算应用接口，所述第二共享节点向所述多个区块链节点发送所述第二数据，包括：

所述第二共享节点通过所述验证计算应用接口向所述多个区块链节点发送所述第二数据。

9. 根据权利要求1至8中任一项所述的方法，其特征在于，所述区块链验证服务管理系统还包括人机交互界面，所述人机交互界面用于部署所述多个共享节点和所述多个专用节点；所述方法包括：

所述人机交互界面接收来自所述第一用户的用户节点创建请求；

在所述用户节点创建完成的情况下，向所述第一用户输出创建成功响应。

10. 一种区块链验证服务管理系统，其特征在于，所述区块链包括多个区块链节点；所述区块链验证服务管理系统包括多个共享节点和多个专用节点，所述共享节点和所述专用节点用作区块链节点；所述

共享节点存储有所述区块链的同步数据；

所述区块链验证服务管理系统用于：

接收来自第一用户的用户节点创建请求，根据所述多个共享节点中的第一共享节点的数据，创建至少一个专用节点，所述至少一个专用节点用作所述第一用户的用户节点；

接收来自所述第一用户的验证计算应用的第一数据；

根据所述第一数据生成第二数据，其中，所述第二数据包括所述第一数据、所述第一用户的标识符、所述第一用户的标识符和所述用户节点的标识符；

向所述多个区块链节点发送所述第二数据。

11. 根据权利要求 10 所述的区块链验证服务管理系统，其特征在于：

在生成所述第二数据的过程中，若所述用户节点可用，所述用户节点用于根据所述第一数据生成所述第二数据。

12. 根据权利要求 10 或 11 所述的区块链验证服务管理系统，其特征在于：

在生成所述第二数据的过程中，若存在所述用户节点不可用的第一时段，在所述第一时段内，所述第一共享节点用于根据所述第一数据生成所述第二数据的一部分；

在所述用户节点可用的第二时段内，所述用户节点用于根据所述第一数据生成所述第二数据的另一部分；其中，所述第二数据包括所述第一数据、所述用户节点的标识符和所述第一共享节点的标识符。

13. 根据权利要求 12 所述的区块链验证服务管理系统，其特征在于，所述用户节点不可用，包括：

所述用户节点处于创建过程中，或者，所述用户节点处于升级维护过程中。

14. 根据权利要求 10 至 13 中任一项所述的区块链验证服务管理系统，其特征在于，还包括验证计算应用接口，所述验证计算应用接口用于：

接收来自所述验证计算应用的所述第一数据；

若所述用户节点可用，向所述用户节点发送所述第一数据；

若所述用户节点不可用，向所述第一共享节点发送所述第一数据。

15. 根据权利要求 14 所述的区块链验证服务管理系统，其特征在于，所述多个共享节点部署于多个区域；

所述用户节点和/或所述第一共享节点用于向第二共享节点发送所述第二数据，其中，所述第二共享节点是与所述用户节点位于相同的区域的共享节点；

所述第二共享节点用于向所述多个区块链节点发送所述第二数据。

16. 根据权利要求 14 或 15 所述的区块链验证服务管理系统，其特征在于：

所述第二共享节点用于向目标区块链节点所属区域的第三共享节点发送所述第二数据；

所述第三共享节点用于向所述目标区块链节点发送所述第二数据。

17. 根据权利要求 14 至 16 中任一项所述的区块链验证服务管理系统，其特征在于，还包括验证计算应用接口，所述第二共享节点用于通过所述验证计算应用接口向所述多个区块链节点发送所述第二数据。

18. 根据权利要求 10 至 17 中任一项所述的区块链验证服务管理系统，其特征在于，还包括人机交互界面，所述人机交互界面用于：

部署所述多个共享节点和所述多个专用节点；

在所述用户节点创建完成的情况下，向所述第一用户输出创建成功响应。

19. 一种计算设备集群，其特征在于，包括至少一个计算设备，每个计算设备包括处理器和存储器；

所述至少一个计算设备的处理器用于执行所述至少一个计算设备的存储器中存储的指令，以使得所述计算设备集群执行如权利要求 1 至 9 中任一项所述的方法。

20. 一种计算机可读存储介质，其特征在于，包括计算机程序指令，当所述计算机程序指令由区块链验证服务管理系统执行时，所述区块链验证服务管理系统执行如权利要求 1 至 9 中任一项所述的方法。

21. 一种计算机程序产品，其特征在于，当所述计算机程序产品在区块链验证服务管理系统上运行时，使得所述区块链验证服务管理系统执行如权利要求 1 至 9 中任一项所述的方法。