

基于区块链的数据处理方法、装置、设备及存储介质

本申请要求于 2021 年 10 月 09 日提交中国专利局，申请号为 202111176926.2，申请名称为“基于区块链的数据处理方法、装置、设备及可读存储介质”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

本申请涉及计算机技术领域，尤其涉及一种基于区块链的数据处理方法、装置、计算机设备、计算机可读存储介质和计算机程序产品。

背景技术

区块链中的数据存储于硬盘中，因为硬盘中的数据可以持久化的保存，而内存中的数据在区块链节点关机后消失。但是，区块链节点对硬盘的读/写效率远低于对内存的读/写效率，因此，为了提高区块链中数据的读取效率，区块链节点在工作时，会为区块链分配一块内存，作为区块链的缓存，用于缓存一些热点数据，即需要频繁访问的数据，例如最近的交易、热点合约的代码等。区块链的缓存有限，当缓存不足时，区块链节点通常会基于最近最少使用的淘汰机制清除热度相对较低的数据。

但是，当最近上线的一个合约成为短期热点合约时，其大量的数据的短期访问频率可能会比一些在整个区块链节点运行过程中会频繁使用的长期热点合约的数据的短期访问频率高。当缓存不足时，根据最近最少淘汰机制，长期热点合约的数据会被淘汰。当后续区块链节点需要访问长期热点合约的数据时，就不得不从硬盘中的数据库获取，数据的访问效率就变低了。

发明内容

一种基于区块链的数据处理方法，包括：

获取针对目标定制合约的申请缓存指令，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；总缓存区域中的每个缓存区域的缓存淘汰机制相互独立；

当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；

若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；新的可用缓存容量大于待缓存容量；

根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。

一种基于区块链的数据处理装置，包括：

指令获取模块，用于获取针对目标定制合约的申请缓存指令；

缓存创建模块，用于根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；总缓存区域中的每个缓存区域的缓存淘汰机制相互独立；

缓存查找模块，用于当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；

数据淘汰模块，用于若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；新的可用缓存容量大于待缓存容量；

数据存储模块，用于根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。

一种计算机设备，包括：处理器、存储器、网络接口；

所述处理器与所述存储器、所述网络接口相连，存储器存储有计算机可读指令，所述计算机可读指令被处理器执行时，使得所述处理器执行本申请实施例中的方法。

一种非易失性的计算机可读存储介质，所述计算机可读存储介质中存储有计算机可读指令，所述计算机可读指令适于由处理器加载并执行本申请实施例中的方法。

一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机可读指令，该计算机可读指令存储在计算机可读存储介质中，计算机设备的处理器从计算机可读存储介质读取该计算机可读指令，处理器执行该计算机可读指令，使得该计算机设备执行本申请实施例中的方法。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图1是本申请实施例提供的一种区块链节点系统的结构示意图；

图2a是本申请实施例提供的一种基于区块链的数据处理的场景示意图；

图2b是本申请实施例提供的一种基于区块链的数据处理的场景示意图；

图3是本申请实施例提供的一种基于区块链的数据处理方法的流程示意图；

图4是本申请实施例提供的一种定制缓存审批的场景示意图；

图5是本申请实施例提供的一种基于区块链的数据处理方法的流程示意图；

图6是本申请实施例提供的一种区块链多缓存实现的框架示意图；

图7是本申请实施例提供的一种基于区块链的数据处理装置的结构示意图；

图8是本申请实施例提供的一种计算机设备的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行

清楚、完整地描述，显然，所描述的实施例仅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作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。

请参见图 1，图 1 是本申请实施例提供的一种区块链节点系统的结构示意图。区块链是一种分布式数据存储、点对点传输、共识机制以及加密算法等计算机技术的新型应用模式，主要用于对数据按时间顺序进行整理，并加密成账本，使其不可被篡改和伪造，同时可进行数据的验证、存储和更新。区块链本质上是一个去中心化的数据库，该数据库中的每个节点均存储一条相同的区块链，区块链网络可以将节点区分为共识节点和业务节点，其中共识节点负责区块链全网的共识。区块链网络中交易数据被写入账本的过程可以为：客户端发送交易数据至业务节点，随后该交易数据以接力棒的方式在区块链网络中的业务节点之间传递，直到共识节点收到该交易数据，共识节点再将该交易数据打包进区块，与其他共识节点之间进行共识，在共识通过后，将携带该交易数据的区块写入账本。

区块 (Block) 是在区块链网络上承载交易数据 (即交易业务) 的数据包，是一种被标记上时间戳和之前一个区块的哈希值的数据结构，区块经过网络的共识机制验证并确定区块中的交易。哈希 (hash) 值，也称作信息特征值或特征值，哈希值是通过哈希算法将任意长度的输入数据转换为密码并进行固定输出而生成的，不能通过解密哈希值来检索原始输入数据，它是一个单向的加密函数。在区块链中，每个区块 (除了初始区块) 都包含前继区块的哈希值，前继区块被称为当前区块的父区块。哈希值是区块链技术中的潜力核心基础和最重要的方面，它保留了记录和查看数据的真实性，以及区块链作为一个整体的完整性。

区块链系统中可以包括有智能合约，该智能合约在区块链系统中可以理解为是一种区块链各节点 (包括共识节点) 可以理解并执行的代码，可以执行任意逻辑并得到结果。用户可以通过客户端发起一个交易业务请求的方式，调用区块链上已经部署的智能合约，随后，区块链上的业务节点可以将该交易业务请求发送至共识节点，而区块链上的各个共识节点可以分别运行该智能合约。区块链中可以包括一个或多个智能合约，这些智能合约可以通过标识号 (Identity document, ID) 或名称来进行区分，而客户端发起的交易业务请求中，也可以携带智能合约的标识号或名称，以此指定区块链需要运行的智能合约。而若客户端所指定的智能合约为需要读取数据的合约，则各个共识节点会访问本地账本来进行数据的读取，最后各个共识节点会互相验证执行结果是否一致 (也就是进行共识)，若一致则可以将执行结果存入各自的本地账本中，并将执行结果返回至客户端。

智能合约的管理可以分为部署、升级以及删除等管理操作。区块链节点可以通过发起一笔交易 (请求) 的方式，来创建或修改区块链系统中指定合约 (即智能合约，下文叙述的合约均指代智能合约) 的代码，从而改变合约的执行逻辑；智能合约的执行是指通过发起一笔交易的方式，调用区块链上已

经部署的合约。区块链系统中的各个节点分别运行同一个合约。对于需要读取数据的合约，会访问节点自己的账本。最终各节点会互相验证执行结果是否一致(共识)，若执行结果一致各节点将必要结果存入各自账本中，并返回结果。

如图 1 所示的区块链节点系统可以对应于区块链网络，该区块链网络可以包括但不限于联盟链所对应的区块链网络。区块链节点系统是指用于进行区块链节点与区块链节点之间数据共享的系统，该区块链节点系统中可以包括多个区块链节点，多个区块链节点具体可以包括区块链节点 10a、区块链节点 10b、区块链节点 10c、区块链节点 10d、…、区块链节点 10n。其中，每个节点在进行正常工作时可以接收到外界发送的数据，并基于接收到的数据进行区块上链处理，也可以向外界发送数据。为了保证各个节点之间的数据互通，每个节点之间可以存在数据连接，例如区块链节点 10a 和区块链节点 10b 之间存在数据连接，区块链节点 10a 和区块链节点 10c 之间存在数据连接，区块链节点 10b 和区块链节点 10c 之间存在数据连接。

可以理解的是，区块链节点之间可以通过上述数据连接进行数据或者区块传输。区块链网络可以基于节点标识实现区块链节点之间的数据连接，对于区块链网络中的每个区块链节点，均具有与其对应的节点标识，而且上述每个区块链节点均可以存储与自身有相连关系的其他区块链节点的节点标识，以便后续根据其他区块链节点的节点标识，将获取到的数据或生成的区块广播至其他区块链节点，例如区块链节点 10a 中可以维护一个如表 1 所示的节点标识列表，该节点标识列表保存着其他节点的节点名称和节点标识：

表 1

节点名称	节点标识
区块链节点 10a	117.114.151.174
区块链节点 10b	117.116.189.145
区块链节点 10b	117.114.151.183
区块链节点 10b	117.117.125.169
...	...
区块链节点 10b	117.116.189.125

其中，节点标识可为网络之间互联的协议 (Internet Protocol, IP) 地址以及其他任意一种能够用于标识区块链网络中节点的信息，表 1 中仅以 IP 地址为例进行说明。例如，区块链节点 10a 可以通过节点标识 117.116.189.145 向区块链节点 10b 发送信息 (例如，区块)，且区块链节点 10b 可以通过节点标识 117.114.151.174 确定该信息是由区块链节点 10a 所发送的。

在区块链中，在将一个区块进行上链之前，该区块必须经过区块链网络中的共识节点进行共识，在共识通过后才能将该区块添加到区块链上。可以

理解的是，当区块链被用于政府或者商业机构的一些场景中时，并非区块链中的所有参与节点（即上述区块链节点系统中的区块链节点）都有足够的资源和必要性成为区块链的共识节点。例如，在图 1 所示的区块链节点系统中，可以将区块链节点 10a、区块链节点 10b、区块链节点 10c 和区块链节点 10d 作为该区块链节点系统中的共识节点。区块链节点系统中的共识节点参与共识，也就是对区块（包含一批交易）进行共识，包括生成区块，对区块进行投票；而非共识节点不参与共识，但会帮助传播区块和投票消息，以及相互同步状态等。

可以理解的是，上述的数据连接不限定连接方式，可以通过有线通信方式进行直接或间接地连接，也可以通过无线通信方式进行直接或间接地连接，还可以通过其他连接方式，本申请在此不做限制。

可以理解的是，本申请实施例所提供的基于区块链的数据处理方法可以由计算机设备执行，计算机设备包括但不限于上述区块链节点（可以为终端或服务器）。上述服务器可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。上述终端可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表、车载终端等，但并不局限于此。

可以理解的是，本申请实施例可应用于各种场景，包括但不限于智慧交通、辅助驾驶等场景。区块链节点系统可以对车载终端所发送的一些驾驶行为数据、道路轨迹数据等等进行共识，共识通过后对其进行上链存储。

本申请实施例提供的基于区块链的数据处理方法，可以应用于如图 1 所示的区块链节点系统中，如图 1 所示，区块链节点系统中的区块链节点（例如，区块链节点 10a）在运行过程中，数据会存储在硬盘的数据库中，但是区块链节点 10a 对于内存的访问速度远大于对硬盘的访问速度，因为区块链节点 10a 可以分配一块内存区域，作为区块链的总缓存区域，用于缓存一些区块链中需要经常访问的数据。区块链节点 10a 可以对总缓存区域进行进一步地精细化管理，将总缓存区域划分为至少两个缓存区域，每个缓存区域存储不同类型的数据，且每个缓存区域的缓存淘汰机制相互独立。其中，缓存淘汰机制用于在缓存区域对应的可用缓存容量不足时，将该缓存区域中的部分数据淘汰，释放这部分数据占用的缓存区域，即对该缓存区域中的部分数据进行淘汰时，不会影响到其他的缓存区域中的数据。区块链节点 10a 可以为区块链中的目标定制合约分配缓存区域，具体的，区块链节点 10a 获取针对目标定制合约的申请缓存指令后，就可以根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与该目标定制合约关联的缓存区域。其中，与该目标定制合约关联的缓存区域仅存储该目标定制合约对应的缓存合约数据。

当区块链节点 10a 接收到目标定制合约对应的待缓存合约数据时，会在

至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域。如果目标缓存区域的可用缓存容量大于待缓存合约数据所需的待缓存容量，则区块链节点 10a 会将待缓存合约数据存储进目标缓存区域；如果目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量，最后根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。可以理解，新的可用缓存容量应该大于待缓存容量。

进一步地，请参见图 2a-图 2b，是本申请实施例提供的一种基于区块链的数据处理的场景示意图。如图 2a 所示的区块链节点 20 可以为上述图 1 所对应实施例中区块链节点系统中的任一区块链节点，例如区块链节点 10b。区块链节点 20 的内存中有一块总缓存区域 22，用于存储区块链的相关数据。如图 2a 所示，总缓存区域 22 中可以包括至少两个缓存区域，其中，不同缓存区域可用于存储不同类型的数据。例如，总缓存区域 22 包含缓存区域 221 和缓存区域 222。缓存区域 221 可以是定制合约关联的缓存区域，其中，定制合约是指单独为其分配一块缓存区域的合约，缓存区域 221 可以映射有合约标识 1，代表缓存区域 221 与合约标识为合约标识 1 的定制合约 A 相关联，该缓存区域 221 用于存储定制合约 A 的缓存合约数据，合约标识可以为合约名称；缓存区域 222 可以是通用合约关联的缓存区域，其中，区块链中没有关联指定的缓存区域的合约均为通用合约，缓存区域 222 用于存储通用合约的缓存合约数据；通用合约可以有多个，因此缓存区域 222 中可以存储多个通用合约的合约标识分别对应的缓存合约数据。

如图 2a 所示，区块链节点 20 获取到针对定制合约 B 的申请缓存指令 21，其中，申请缓存指令 21 包含有定制缓存创建参数，该定制缓存创建参数包含有定制合约 B 的合约标识，假设为合约标识 2，还包含有定制合约 B 的预分配缓存容量。其中，申请缓存指令 21 可以是区块链节点 20 根据申请缓存请求执行申请缓存合约得到的。申请缓存请求可以由区块链节点系统中的任一区块链节点（例如，上述图 1 所对应实施例中的区块链节点系统中的区块链节点 10c）发起的。需要说明的是，区块链节点先基于申请缓存请求和申请缓存合约，生成待上链的申请缓存指令 21，随后，区块链节点系统会对待上链的申请缓存指令 21 进行共识，共识通过后，区块链节点才继续确定是否执行该申请缓存指令 21。可以理解，申请缓存指令 21 用于指示区块链节点 20 在总缓存区域 22 中为定制合约 B 分配预分配缓存容量对应的缓存区域。不过，在一些情况下，并不是所有区块链节点中的合约均需要为其分配一个缓存区域，因此，在一个实施例中，可以设置缓存申请条件，区块链节点 20 需要先确认申请缓存指令 21 满足缓存申请条件后，再执行申请缓存指令 21。其中，缓存申请条件可以根据实际情况来设定，比如，需要经过 N 个区块链节点系统开发者的审批，N 为整数，这里不作限制。缓存申请条件用于避免恶意创建大量合约生成巨量缓存，从而导致系统崩溃的情况。如图 2a 所示，当确定申请缓存指令 21 满足缓存申请条件时，区块链节点 20 可以在总缓存区域 22

中创建与定制合约 B 关联的缓存区域 223，该缓存区域 223 映射有定制合约 B 的合约标识 2。当确定申请缓存指令 21 不满足缓存申请条件时，区块链节点 20 不执行申请缓存指令 21，即不在总缓存区域 22 中创建与定制合约 B 关联的缓存区域 223。

进一步地，请参见图 2b，如图 2b 所示，当区块链节点 20 接收到定制合约 B 对应的待缓存合约数据 23 时，需要先从总缓存区域 22 中找到定制合约 B 关联的缓存区域 223。区块链节点 22 可以根据待缓存合约数据 23 中的合约标识 22 与缓存区域 223 的映射关系，精确查找到缓存区域 223。随后，区块链节点 20 需要判断缓存区域 223 中的可用缓存是否足够，即缓存区域 223 的可用缓存容量是否小于待缓存合约数据 23 的待缓存容量。如果缓存区域 223 的可用缓存容量不小于待缓存合约数据 23 的待缓存容量，则区块链节点 20 可以直接将待缓存合约数据 23 存储进缓存区域 223。如果，缓存区域 223 的可用缓存容量小于待缓存合约数据 23 的待缓存容量，则区块链节点会先根据缓存区域 223 的缓存淘汰机制对缓存区域 223 中的缓存合约数据进行淘汰处理，即释放掉部分缓存合约数据，得到缓存区域 223 的新的可用缓存容量。可以理解，新的可用缓存容量大于待缓存容量。最后，区块链节点 20 就可以根据新的可用缓存容量将待缓存合约数据 23 存储进缓存区域 223。

本申请实施例将总缓存区域扩展为多个缓存区域，并对这些缓存区域做精细化管理，即可以为每个合约单独指定一个缓存区域，且每个缓存区域的大小可以根据需求进行配置，这样区块链中长期热点合约及短期热点合约的数据可以并存，不同合约的缓存互不影响，因此区块链节点常用的数据都可以在缓存中找到，避免因缓存淘汰机制从硬盘的数据库中访问长期热点合约的数据而导致的数据访问效率较低的问题，从而提升了访问长期热点合约以及短期热点合约的数据的效率，提升区块链的整体性能。

进一步地，请参见图 3，图 3 是本申请实施例提供的一种基于区块链的数据处理方法的流程示意图。其中，该数据处理方法可以由区块链节点（例如，上述图 1 所示的区块链节点系统中的任一区块链节点）执行。以下将以该基于区块链的数据处理方法由区块链节点执行为例进行说明。其中，该基于区块链的数据处理方法至少可以包括以下步骤 S101-步骤 S104：

S101，获取针对目标定制合约的申请缓存指令，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；总缓存区域中的每个缓存区域的缓存淘汰机制相互独立。

具体的，申请缓存指令用于指示区块链节点在总缓存区域中为目标定制合约分配对应的缓存区域。总缓存区域是指区块链节点从内存区域中划分的一块内存区域，用于暂时存储区块链对应的热点数据，其中，热点数据是指区块链节点工作时需要频繁访问的数据，例如系统配置数据、合约数据、最近的交易数据等等。为了减少不同热点数据之间的相互影响，区块链节点可以将总缓存区域划分为至少两个缓存区域，不同的缓存区域用于存储不同的

热点数据,例如,至少两个缓存区域可以包括定制合约缓存区域,其中,定制合约缓存区域仅用于存储与该定制合约缓存区域相关联的定制合约对应的缓存合约数据;至少两个缓存区域可以包括通用合约缓存区域,其中,通用合约缓存区域用于存储一个或多个通用合约分别对应的缓存合约数据,通用合约是指没有关联定制合约缓存区域的合约;至少两个缓存区域还可以包括区块缓存区域,其中,区块缓存区域用于存储未写入区块链账本中的待上链区块对应的区块缓存数据;至少两个缓存区域还可以包括交易缓存区域,其中,交易缓存区域用于存储未写入区块链账本中的待上链交易对应的交易缓存数据;至少两个缓存区域还可以包括系统缓存区域,其中,系统缓存区域用于存储一个或多个系统合约分别对应的缓存合约数据,例如,区块链节点系统的配置参数。缓存区域的类型可以根据区块链节点系统的实际功能情况来设置,这里不作限制。

具体的,针对区块链节点系统使用的合约中,部分合约可能成为热点合约,会被区块链节点系统中的区块链节点频繁访问使用,为了加强区块链节点的数据访问效率以及提供区块链节点系统的性能,可以将这部分合约作为定制合约,在总缓存区域中为定制合约关联特定的缓存区域,定制合约关联的缓存区域将只用于存储定制合约的合约数据。

在一个实施例中,区块链节点获取针对目标定制合约的申请缓存指令的具体实现过程,可以为:区块链节点获取申请缓存请求,其中,申请缓存请求包括申请缓存合约标识、与目标定制合约相关联的定制缓存创建参数。然后,区块链节点调用申请缓存请求中的申请缓存合约标识所指示的申请缓存合约,在合约虚拟机中,根据定制缓存创建参数执行申请缓存合约,得到针对目标定制合约的申请缓存指令,其中,申请缓存指令包含定制缓存创建参数。其中,定制缓存创建参数可以包括目标定制合约的合约标识和预分配缓存容量。因此,根据申请缓存指令中的定制缓存创建参数,在总缓存区域中创建与目标定制合约关联的缓存区域的具体实现过程,可以为:在总缓存区域中获取与预分配缓存容量对应的预分配缓存区域;将预分配缓存区域和目标定制合约的合约标识进行映射处理,得到与目标定制合约关联的缓存区域。

具体的,为了避免为大量合约申请缓存区域的恶意行为而造成系统崩溃,区块链节点在执行申请缓存指令之前,需要先确定该申请缓存指令满足缓存申请条件。在一个实施例中,当在共识网络中对申请缓存指令共识通过时,区块链节点可以通过缓存审批组件获取针对申请缓存指令的一个或多个缓存审批结果,其中,每个缓存审批结果均为经过共识网络共识通过后的数据;然后,当一个或多个缓存审批结果中,属于缓存审批通过的缓存审批结果的数量超过审批通过阈值时,确定申请缓存指令满足缓存申请条件。其中,缓存审批结果可以由区块链节点系统的开发者决定。

S102,当接收到目标定制合约对应的待缓存合约数据时,在至少两个缓存区域中查找与目标定制合约关联的缓存区域,作为目标缓存区域。

具体的,上述可知,区块链节点在建立与目标定制合约关联的缓存区域

时,可以将该缓存区域和目标定制合约的合约标识进行映射处理,因此,区块链节点可以根据目标定制合约的合约标识与目标定制合约关联的缓存区域的映射关系,来进行目标缓存区域的查找。

S103,若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量,则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理,得到新的可用缓存容量;新的可用缓存容量大于待缓存容量。

具体的,区块链节点中的每个缓存区域对应的缓存容量均有限,当目标缓存区域的可用缓存容量不足时,区块链节点就会根据目标缓存区域的缓存淘汰机制来释放部分存储于该目标缓存区域的缓存合约数据,得到新的可用缓存容量,可以理解,新的可用缓存容量应该大于待缓存合约数据对应的待缓存容量。

在一个实施例中,根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理,得到新的可用缓存容量的具体实现过程,可以为:区块链节点可以获取目标缓存区域对应的第一访问记录列表;然后,区块链节点将数据访问最近时间最早的缓存合约数据,作为目标缓存区域中的目标淘汰合约数据;最后,区块链节点释放目标淘汰合约数据,将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量,作为新的可用缓存容量。其中,第一访问记录列表包含至少两个缓存合约数据分别对应的数据访问最近时间,至少两个缓存合约数据均存储于目标缓存区域,即均为目标定制合约对应的缓存合约数据。例如,区块链节点A可以为与定制合约m关联的缓存区域B建立一张访问记录列表C(即第一访问记录列表),缓存区域B中存储有缓存合约数据b1、缓存合约数据b2以及缓存合约数据b3,则访问记录列表C中可以存储区块链节点A对缓存合约数据b1的数据访问最近时间,即区块链节点A上一次访问缓存合约数据b1的时间,访问记录列表C中还可以存储区块链节点A对缓存合约数据b2的数据访问最近时间和区块链节点A对缓存合约数据b3的数据访问最近时间。假设区块链节点A对缓存合约数据b1的数据访问最近时间为9:21,区块链节点A对缓存合约数据b2的数据访问最近时间为9:19,区块链节点A对缓存合约数据b3的数据访问最近时间为9:22。此时,倘若区块链节点A接收到定制合约m对应的待缓存合约数据b4,此时缓存区域B为目标缓存区域,假设缓存区域B中的可用缓存容量不足,此时区块链节点A会对访问记录列表C进行查找,确定出数据访问最近时间最早的缓存合约数据,即缓存合约数据b2,区块链节点A会将缓存合约数据b2作为目标淘汰合约数据进行释放。不过,可以理解的是,假设区块链节点A释放掉缓存合约数据b2后,缓存区域B中仍然无法存储待缓存合约数据b4,区块链节点A会在缓存区域B中剩余的缓存合约数据中,继续获取数据访问最近时间最早的缓存合约数据,作为新的目标淘汰合约数据,继续释放新的目标淘汰合约数据,直到缓存区域B的新的可用缓存容量足够存储待缓存合约数据b4。

在一个实施例中,根据目标缓存区域的缓存淘汰机制对目标缓存区域中

的缓存合约数据进行淘汰处理，得到新的可用缓存容量的具体实现过程，可以为：区块链节点可以获取目标缓存区域对应的第二访问记录列表；然后，将数据访问次数最少的缓存合约数据，作为目标缓存区域中的目标淘汰合约数据；最后，区块链节点释放目标淘汰合约数据，将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。其中，第二访问记录列表包含至少两个缓存合约数据在目标时间段内，分别对应的数据访问次数；至少两个缓存合约数据均存储于目标缓存区域，即均为目标定制合约对应的缓存合约数据。其中，目标时间段内通常是指最近一段时间内，例如最近五分钟，最近十分钟，最近一小时等等，仅记录目标时间段内缓存合约数据的数据访问次数，可以反映缓存合约数据的最近被访问频次高低的情况。例如，区块链节点D可以为与定制合约n关联的缓存区域E建立一张访问记录列表F(即第二访问记录列表)，缓存区域E中存储有缓存合约数据e1、缓存合约数据e2以及缓存合约数据e3，则访问记录列表F中可以存储区块链节点D在目标时间段内对缓存合约数据e1的数据访问次数，区块链节点D在目标时间段内对缓存合约数据e2的数据访问次数和区块链节点D在目标时间段内对缓存合约数据e3的数据访问次数。假设区块链节点D在最近一小时内对缓存合约数据e1的数据访问次数为10，区块链节点D在最近一小时内对缓存合约数据e2的数据访问次数为50，区块链节点D在最近一小时内对缓存合约数据e3的数据访问次数为1。此时，倘若区块链节点D接收到定制合约n对应的待缓存合约数据e4，此时缓存区域E为目标缓存区域，假设缓存区域E中的可用缓存容量不足，此时区块链节点D会对访问记录列表F进行查找，确定出最近一小时内被访问次数最少的缓存合约数据，即缓存合约数据e3，区块链节点A会将缓存合约数据e3作为目标淘汰合约数据进行释放。不过，可以理解的是，假设区块链节点D释放掉缓存合约数据e3后，缓存区域E中仍然无法存储待缓存合约数据e4，区块链节点D会在缓存区域E中剩余的缓存合约数据中，继续获取在目标时间段内数据访问次数最少的缓存合约数据，作为新的目标淘汰合约数据，继续释放新的目标淘汰合约数据，直到缓存区域E的新的可用缓存容量足够存储待缓存合约数据e4。

在一个实施例中，根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量的具体实现过程，可以为：区块链节点可以获取目标缓存区域对应的数据存储记录列表；然后，将数据缓存时间最早的缓存合约数据，作为目标缓存区域中的目标淘汰合约数据；最后，区块链节点释放目标淘汰合约数据，将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。其中，数据存储记录列表包含至少两个缓存合约数据分别对应的数据缓存时间；至少两个缓存合约数据均存储于目标缓存区域，即均为目标定制合约对应的缓存合约数据。例如，区块链节点G可以为与定制合约l关联的缓存区域H建立一张数据存储记录列表I，缓存区域H中存储有缓存合约数据h1、缓存合约数据h2以及缓存合约数据h3，则访问记录列表I中可以存储有缓存合约数据h1、

缓存合约数据 h2 以及缓存合约数据 h3 分别存储进缓存区域 H 的数据缓存时间。假设区块链节点 G 将缓存合约数据 h1 存储进缓存区域 H 的数据缓存时间为 9:21, 区块链节点 G 将缓存合约数据 h2 存储进缓存区域 H 的数据缓存时间为 9:19, 区块链节点 G 将缓存合约数据 h3 存储进缓存区域 H 的数据缓存时间为 9:22。此时, 倘若区块链节点 G 接收到定制合约 1 对应的待缓存合约数据 h4, 此时缓存区域 H 为目标缓存区域, 假设缓存区域 H 中的缓存可用容量不足, 此时区块链节点 G 会对访问记录列表 I 进行查找, 确定出数据缓存时间最早的缓存合约数据, 即缓存合约数据 h2, 区块链节点 G 会将缓存合约数据 h2 作为目标淘汰合约数据进行释放。不过, 可以理解的是, 假设区块链节点 G 释放掉缓存合约数据 h2 后, 缓存区域 H 中仍然无法存储待缓存合约数据 h4, 区块链节点 G 会在缓存区域 H 中剩余的缓存合约数据中, 继续获取数据缓存时间最早的缓存合约数据, 作为新的目标淘汰合约数据, 继续释放新的目标淘汰合约数据, 直到缓存区域 H 的新的可用缓存容量足够存储待缓存合约数据 h4。

S104, 根据新的可用缓存容量将所述待缓存合约数据存储进目标缓存区域。

具体的, 区块链节点将待缓存合约数据存储进对应的目标缓存区域。

采用本申请实施例提供的方法, 可以获取针对目标定制合约的申请缓存指令, 根据申请缓存指令中的定制缓存创建参数, 在总缓存区域中创建与目标定制合约关联的缓存区域; 然后, 当接收到目标定制合约对应的待缓存合约数据时, 在至少两个缓存区域中查找与目标定制合约关联的缓存区域, 作为目标缓存区域; 若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量, 则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理, 得到新的可用缓存容量, 最后根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。其中, 总缓存区域包括至少两个缓存区域, 且每个缓存区域的缓存淘汰机制相互独立; 新的可用缓存容量大于待缓存容量。通过本申请实施例提供的方法, 可以为区块链系统中的合约分配专属的缓存区域, 该缓存区域仅用于存储该合约的缓存合约数据, 倘若缓存不足, 也仅根据淘汰机制在该缓存区域中淘汰缓存合约数据, 不会影响其他合约的缓存合约数据, 因此, 可以为区块链中需要频繁访问的合约分配专属的缓存区域, 避免该合约对应的缓存合约数据被其余合约的缓存合约数据所影响而被淘汰, 从而避免区块链节点频繁去硬盘中获取该合约对应的数据, 从而提高了区块链节点的数据的访问效率。

进一步地, 请参见图 4, 图 4 是本申请一个实施例中定制缓存审批的场景示意图。其中, 如图 4 所示的区块链节点系统 40 可以为上述图 1 所对应实施例中的区块链节点系统; 如图 4 所示的合约开发者 A 对应的区块链节点 40a 可以为上述图 1 所对应实施例中区块链节点系统中的任一区块链节点, 如, 区块链节点 40a 可以为图 1 中的区块链节点 10a; 如图 4 所示的链开发者 B 对应的区块链节点 40b、链开发者 C 对应的区块链节点 40c 以及链开发者 N

对应的区块链节点 40n 均可以为上述图 1 所对应实施例中区块链节点系统中的任一区块链节点, 如, 区块链节点 40b 可以为图 1 中的区块链节点 10b, 区块链节点 40c 可以为图 1 中的区块链节点 10c, 区块链节点 40n 可以为图 1 中的区块链节点 10d。可以理解, 区块链节点 40a、区块链节点 40b、区块链节点 40c 以及区块链节点 40d 均是区块链节点系统 40 中的区块链节点, 同区块链节点系统中的其余区块链节点一样, 均需要完成区块链的共识上链处理过程。其中, 合约开发者可以发布区块链中各个区块链节点可以执行的合约; 链开发者可以对区块链的底层架构与逻辑进行管理和修改。

如图 4 所示, 合约开发者 A 可以通过区块链节点 40a 编写合约 X, 随后部署合约 X, 并将部署合约 X 的交易请求 41 发送至区块链节点系统 40, 区块链节点系统 40 中的区块链节点会对该部署合约 X 的交易请求 41 进行共识上链处理, 如果区块链节点系统 40 对该部署合约 X 的交易请求 41 共识通过, 区块链节点系统 40 中的每个区块链节点均会将合约 X 写入区块链。假设合约 X 为通用合约, 区块链节点系统 40 中的任一区块链节点在工作时, 倘若需要将合约 X 对应的合约数据写进缓存中, 只会将合约 X 对应的合约数据和其他通用合约一起, 写进通用合约缓存区域。假设合约 X 成为区块链上的热点合约, 区块链节点需要频繁访问合约 X 对应的合约数据, 通用合约缓存区域中其他通用合约对应的合约数据新增时, 容易触发缓存淘汰机制, 将合约 X 对应的合约数据挤出去, 区块链节点又需要重新从数据库中获取合约 X 对应的合约数据, 导致区块链节点的数据访问效率很低。因此, 合约开发者 A 可以通过区块链节点 40a 发起为合约 X 申请缓存的交易请求 42 (即上述图 3 所对应实施例中所述的申请缓存请求), 区块链节点系统 40 接收到为合约 X 申请缓存的交易请求 42 后, 会对该为合约 X 申请缓存的交易请求 42 进行共识上链处理, 如果区块链节点系统 40 对该为合约 X 申请缓存的交易请求 42 共识通过, 则区块链节点系统 40 中的每个区块链节点会执行该交易请求 42, 得到申请缓存指令。

在一个实施例中, 在区块链节点执行申请缓存指令之前, 需要先确定该申请缓存指令满足缓存申请条件, 即有超过 M 个链开发者同意为合约 X 分配定制合约缓存区域, 假设一共有 L 个链开发者, M、L 皆为正整数, 且 M 应该小于或者等于 L。如图 4 所示, 链开发者 B 同意为合约 X 分配定制合约缓存区域, 因此通过区块链节点 40b 向区块链节点系统 40 发送缓存审批的交易请求 43, 区块链节点系统 40 会对该缓存审批的交易请求 43 进行共识上链处理, 如果区块链节点系统 40 对该缓存审批的交易请求 43 共识通过, 则区块链节点系统 40 会执行该缓存审批的交易请求 43, 确定属于缓存审批通过的缓存审批结果的数量为 1。同理可知, 链开发者 C、……、链开发者 N 均可以通过对应的区块链节点发送各自的缓存审批的交易请求, 每当区块链节点 40 完成缓存审批的交易请求的共识后, 会执行缓存审批的交易请求, 得到对应的缓存审批结果。当区块链节点系统 40 确定属于缓存审批通过的缓存审批结果的数量超过 M 时, 则区块链节点系统 40 中的每个区块链节点均可以根

据申请缓存指令为合约 X 分配定制合约缓存区域。可选的，为合约 X 申请缓存的交易请求 42 可以包含在部署合约 X 的交易请求 41 中，即合约开发者可以在部署合约 X 的同时，就申请为合约 X 分配缓存了，无需二次发起交易请求。

采用本申请实施例提供的方法，对于区块链中的任一合约，合约开发者可以为其申请分配定制合约缓存区域，该定制合约缓存区域仅用于存储该合约对应的缓存合约数据，因此，区块链中其余合约对应的缓存合约数据就无法影响该合约对应的缓存合约数据，为区块链中需要频繁访问的合约分配定制合约缓存区域，可以避免需要频繁访问的合约对应的缓存合约数据被淘汰，从而避免区块链节点频繁通过访问硬盘来获取需要的数据，可以提高区块链节点的数据访问效率。

进一步地，请参见图 5，图 5 是本申请实施例提供的一种基于区块链的数据处理方法的流程示意图。其中，该基于区块链的数据处理方法可以由区块链节点（例如，上述图 1 所对应实施例中区块链节点系统中的任一区块链节点）执行。以下将以该基于区块链的数据处理方法由区块链节点执行为例进行说明。其中，该基于区块链的数据处理方法至少可以包括以下步骤 S201-步骤 S202：

步骤 S201，通过缓存路由组件接收数据查询请求；数据查询请求包含待执行合约的合约标识。

具体的，区块链中的区块链节点可以通过发起交易请求的形式，来调用区块链上已经部署的合约（即智能合约）。由于区块链中可以包括一个或多个合约，可以通过合约标识来区分不同的合约，其中，合约标识可以为标识号或者合约名称等等。区块链节点发起的交易请求中，可以携带合约的合约标识，以此指定区块链需要运行的智能合约。当区块链中的区块链节点需要执行交易请求时，需要先获取对应的数据，因此可以根据交易请求中的合约标识来生成数据查询请求，区块链节点中的缓存路由组件接收到数据查询请求后，就可以根据其中的合约标识确定需要获取哪个待执行合约的数据。

步骤 S202，通过缓存路由组件，在总缓存区域的至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据。

具体的，在区块链节点的总缓存区域中，可以包括至少两个缓存区域，不同缓存区域用于存储不同的数据，每个缓存区域的大小可以不同，配置的缓存淘汰机制也可以不同。上述可知，区块链中的合约均对应有唯一的合约标识，因此在缓存路由组件中可以记录合约的合约标识与存储该合约的缓存区域之间的映射关系，例如，合约 O 存储于缓存区域 P 中，则合约 O 的合约标识可以与缓存区域 P 具有映射关系，区块链节点可以根据合约 O 的合约标识与缓存区域 P 的映射关系，快速确定存储合约 O 的缓存区域为缓存区域 P。当缓存路由组件接收到路由查询请求时，可以根据待执行合约的合约标识，来查找与待执行合约的合约标识相关联的查询缓存数据。

为便于理解，请一并参见图 6，图 6 是本申请实施例提供的一种区块链

多缓存实现的框架示意图。区块链节点可以将区块链缓存（即上述总缓存区域）细分为多个缓存（即上述缓存区域），并通过缓存管理单元对多个缓存进行管理。如图 6 所示，缓存管理单元包含缓存路由单元（即上述图 3 所对应实施例中的缓存路由组件）和缓存审批单元（即上述图 3 所对应实施例中的缓存审批组件）。其中，缓存路由单元用于根据查询请求（即上述图 5 所对应实施例中的数据查询请求）中的合约名（即上述合约标识）找到对应的缓存，并从缓存中查找目标数据（即上述图 5 所对应实施例中的查询缓存数据）。其中，缓存审批单元用于管理合约的缓存分配，避免通过恶意创建大量合约生成巨量缓存，使系统崩溃。本申请在区块链中新增申请缓存合约及 API（Application Programming Interface，应用程序接口），目标合约所有者（即上述图 4 所对应实施例中的合约开发者）可调用此接口，申请缓存合约，申请时以目标合约名、缓存大小为接口调用参数，经由 N 个系统开发者（即上述图 4 所对应实施例中的链开发者）审批，可为此目标合约生成定制缓存。

在一个实施例中，至少两个缓存区域可以包括通用合约缓存区域和 N 个定制合约缓存区域。其中，N 为大于或等于 0 的整数；通用合约缓存区域中包括一个或多个通用合约分别对应的缓存合约数据；一个定制合约缓存区域中存储的缓存合约数据和一个定制合约缓存区域相关联的定制合约对应。定制合约缓存区域的创建可以参见上述图 3 所对应实施例中步骤 S101 的描述，也可以参见上述图 4 所对应实施例中的场景描述。此时，通过缓存路由组件，在至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据的具体实现过程，可以为：通过缓存路由组件，获取与 N 个定制合约缓存区域相关联的定制合约对应的定制合约标识；在定制合约标识中查找待执行合约的合约标识；若在定制合约标识中查找到与待执行合约的合约标识相同的定制合约标识，则通过缓存路由组件在与待执行合约的合约标识对应的定制合约关联的定制合约缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据；若在定制合约标识中未查找到与待执行合约的合约标识相同的定制合约标识，则通过缓存路由组件在通用合约缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据。请再参见图 6，如图 6 所示，区块链缓存（即总缓存区域）中包含有通用合约缓存（即通用合约缓存区域）和定制合约缓存（即定制合约缓存区域），其中，定制合约缓存包含一个或多个定制合约缓存。假设定制合约缓存中包含有两个定制合约缓存，合约 A 关联的定制合约缓存 1 和合约 B 关联的定制合约缓存 2，此时，区块链节点中，合约 A 对应的缓存合约数据将存储在定制合约缓存 1，合约 B 对应的缓存合约数据将存储在定制合约缓存 2，而区块链上的其他合约，比如合约 C、合约 D 等对应的缓存合约数据均会存储在通用合约缓存。此时，在缓存路由单元中，会记录合约 A 的合约标识与定制合约缓存 1 的映射关系，合约 B 的合约标识与定制合约缓存 2 的映射关系。当缓存路由单元从数据查询请求中获取到待执行合约的合约标识后，可以获取定制合约对应的定制合约标识，即合约 A 的合约标识和合约 B 的合约标识，缓存路由单元会在定制合约标识中查

找待执行合约的合约标识，如果待执行合约的合约标识与某个定制合约标识相同，假设待执行合约的合约标识和合约 A 的合约标识相同，则缓存路由单元会确定在定制合约缓存 1 中查找与待执行合约的合约标识相关联的查询缓存数据；如果待执行合约的合约标识与定制合约标识均不相同，说明待执行合约属于通用合约，缓存路由单元会在通用合约缓存中查找与待执行合约的合约标识相关联的查询缓存数据。

需要说明的是，区块链节点允许为指定合约定制缓存，包括缓存大小，淘汰机制（即上述所说的缓存淘汰机制）等。其中，淘汰机制可以为 LRU（Least Recently Used，最近最久未使用）缓存，LFU（Least Frequently Used，最近最少未使用）缓存，FIFO（Least Frequently Used，先进先出）缓存等。其中，LRU 缓存是指如果一个数据在最近一段时间没有被访问到，那么在将来它被访问的可能性也很小；也就是说，当限定的空间已存满数据时，应当把最久没有被访问到的数据淘汰；LFU 缓存是指如果一个数据在最近一段时间内使用次数很少，那么在将来一段时间内被使用的可能性也很小；FIFO 缓存是指如果一个数据最先进入缓存中，则应该最早淘汰掉。

在一个实施例中，数据查询请求还包含调用待执行合约的待执行交易；至少两个缓存区域还包括区块缓存区域和交易缓存区域。其中，区块缓存区域用于存储未写入区块链账本中的待上链区块对应的区块缓存数据；交易缓存区域用于存储未写入区块链账本中的待上链交易对应的交易缓存数据。此时，通过缓存路由组件，在至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据的具体实现过程，可以为：在缓存路由组件中，根据与待执行合约的合约标识和待执行交易对交易缓存区域中的交易缓存数据进行遍历；若在交易缓存区域中遍历到与待执行合约的合约标识和待执行交易相关联的交易缓存数据，则将所遍历到的交易缓存数据，作为查询缓存数据；若在交易缓存区域中未遍历到与待执行合约的合约标识和待执行交易相关联的交易缓存数据，则在缓存路由组件中，根据与待执行合约的合约标识和待执行交易对区块缓存区域中的区块缓存数据进行遍历；若在区块缓存区域中遍历到与待执行合约的合约标识和待执行交易相关联的区块缓存数据，则将所遍历到的区块缓存数据，作为查询缓存数据；若在区块缓存区域中未遍历到与待执行合约的合约标识和待执行交易相关联的区块缓存数据，则在区块链数据库中获取与待执行合约的合约标识相关联的查询缓存数据。

请再参见图 6，区块链缓存中还可以包含有区块缓存（即区块缓存区域）和交易缓存（交易缓存区域）。其中，区块缓存用于缓存最近的区块，以提升区块链节点同步区块的速度。其中，交易缓存用于缓存最近的交易，用于提升交易访问效率。在区块链节点工作的过程中，对于某些交易对应的数据查询请求，区块链节点获取的查询缓存数据中，除了包含待执行合约对应的合约数据，还需要包含调用该待执行合约的待执行交易的交易数据。例如，区块链节点接收到一个转账交易，假设该转账交易为 A 转账 100 元给 B。该转账交易对应的数据查询请求中将包含转账交易所需的转账合约的合约标识，

区块链节点通过缓存管理单元获取到转账合约的合约标识，就可以获取到该转账合约对应的缓存合约数据，可以理解，仅有转账合约对应的缓存合约数据，该转账交易无法实现，因为区块链节点不知道该转账交易的交易数据，即 A 的余额和 B 的余额。因此，区块链节点还需要先从交易缓存中，根据转账合约的合约标识和该转账交易对交易缓存中的交易缓存数据进行遍历，以确定 A 的余额和 B 的余额；如果区块链节点在交易缓存中未查询到 A 的余额和 B 的余额，区块链节点将对区块缓存中的区块缓存数据进行遍历；如果区块链节点在区块缓存中仍然未查询到 A 的余额和 B 的余额，区块链节点就将从数据库中进行查询。当区块链节点查询到 A 的余额和 B 的余额，就会将其和转账合约的缓存合约数据一起作为交易查询数据，发送给合约虚拟机，合约虚拟机就可以根据交易查询数据完成该转账交易。

可选的，待上链区块中包含一个或多个待上链交易。当待上链区块写入区块链账本中时，从区块缓存区域中，释放待上链区块对应的区块缓存数据；从交易缓存区域中，释放待上链区块中的一个或多个待上链交易对应的交易缓存数据。

在一个实施例中，至少两个缓存区域还包括系统缓存区域；系统缓存区域中包括一个或多个系统合约分别对应的缓存合约数据。此时，通过缓存路由组件，在至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据的具体实现过程，可以为：获取一个或多个系统合约对应的系统合约标识；在系统合约标识中查找待执行合约的合约标识；若在系统合约标识中查找到与待执行合约的合约标识相同的系统合约标识，则通过缓存路由组件在与待执行合约的合约标识对应的系统合约关联的系统缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据。请再参见图 6，如图 6 所示，区块链缓存中还可以包含有系统缓存（即系统缓存区域），用于保存系统参数，如 gaslimit（消耗上限），gasprice（消耗单价），出块时间，区块最大交易数量限制等。

采用本申请实施例提供的方法，可以在区块链节点中将总缓存区域划分至少两个缓存区域，每个缓存区域用于存储不同的数据，区块链节点可以根据需要配置每个缓存区域的大小，不同缓存区域中缓存数据的淘汰均不会影响到其他缓存区域。区块链节点可以通过缓存路由组件接收数据查询请求，其中，数据查询请求包含待执行合约的合约标识；然后，区块链节点可以通过缓存路由组件，在至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据。因此，为区块链中需要频繁访问的合约分配专属的缓存区域，避免该合约对应的缓存合约数据被其余合约的缓存合约数据所影响而被淘汰，从而避免区块链节点频繁去硬盘中获取该合约对应的数据，从而提高了区块链节点的数据的访问效率。

请参见图 7，是本申请实施例提供的一种基于区块链的数据处理装置的结构示意图。该基于区块链的数据处理装置可以是运行于计算机设备的一个计算机程序（包括程序代码），例如该基于区块链的数据处理装置为一个应

用软件；该装置可以用于执行本申请实施例提供的基于区块链的数据处理方法中的相应步骤。如图 7 所示，该基于区块链的数据处理装置 1 可以包括：指令获取模块 11、缓存创建模块 12、缓存查找模块 13、数据淘汰模块 14 以及数据存储模块 15。

指令获取模块 11，用于获取针对目标定制合约的申请缓存指令；

缓存创建模块 12，用于根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；总缓存区域中的每个缓存区域的缓存淘汰机制相互独立；

缓存查找模块 13，用于当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；

数据淘汰模块 14，用于若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；新的可用缓存容量大于待缓存容量；

数据存储模块 15，用于根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。

其中，指令获取模块 11、缓存创建模块 12、缓存查找模块 13、数据淘汰模块 14 以及数据存储模块 15 的具体功能实现方式可以参见图 3 对应实施例中的步骤 S101 到步骤 S104 的具体描述。

请再参见图 7，指令获取模块 11 可以包括：请求获取单元 111、合约调用单元 112 以及指令生成单元 113。

请求获取单元 111，用于获取申请缓存请求；申请缓存请求包括申请缓存合约标识、与目标定制合约相关联的定制缓存创建参数；

合约调用单元 112，用于调用申请缓存请求中的申请缓存合约标识所指示的申请缓存合约；

指令生成单元 113，用于在合约虚拟机中，根据定制缓存创建参数执行申请缓存合约，得到针对目标定制合约的申请缓存指令；申请缓存指令包含定制缓存创建参数。

其中，请求获取单元 111、合约调用单元 112 以及指令生成单元 113 的具体功能实现方式可以参见图 3 对应实施例中的步骤 S101 的具体描述。

请再参见图 7，上述基于区块链的数据处理装置 1，还可以包括：审批模块 16。

审批模块 16，用于当在共识网络中对申请缓存指令共识通过时，通过缓存审批组件获取针对申请缓存指令的一个或多个缓存审批结果；每个缓存审批结果均为经过共识网络共识通过后的数据；

审批模块 16，还用于当一个或多个缓存审批结果中，属于缓存审批通过的缓存审批结果的数量超过审批通过阈值时，确定申请缓存指令满足缓存申请条件；

缓存创建模块 12，用于在验证出申请缓存指令满足缓存申请条件时，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域。

其中，审批模块 16 的具体功能实现方式可以参见图 3 对应实施例中的步骤 S101 的可选描述。

其中，定制缓存创建参数包括目标定制合约的合约标识和预分配缓存容量；

请再参见图 7，缓存创建模块 12 可以包括：预分配单元 121 以及映射单元 122。

预分配单元 121，用于在总缓存区域中获取与预分配缓存容量对应的预分配缓存区域；

映射单元 122，用于将预分配缓存区域和目标定制合约的合约标识进行映射处理，得到与目标定制合约关联的缓存区域。

其中，预分配单元 121 以及映射单元 122 的具体功能实现方式可以参见图 3 对应实施例中的步骤 S101 的具体描述。

请再参见图 7，数据淘汰模块 14 可以包括：第一列表获取单元 141、第一数据确定单元 142 以及第一数据淘汰单元 143。

第一列表获取单元 141，用于获取目标缓存区域对应的第一访问记录列表；第一访问记录列表包含至少两个缓存合约数据分别对应的数据访问最近时间；目标缓存区域包含至少两个缓存合约数据；

第一数据确定单元 142，用于将数据访问最近时间最早的缓存合约数据，作为目标缓存区域中的目标淘汰合约数据；

第一数据淘汰单元 143，用于释放目标淘汰合约数据，将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

其中，第一列表获取单元 141、第一数据确定单元 142 以及第一数据淘汰单元 143 的具体功能实现方式可以参见图 3 对应实施例中的 S103 的具体描述。

请再参见图 7，数据淘汰模块 14 可以包括：第二列表获取单元 144、第二数据确定单元 145 以及第二数据淘汰单元 146。

第二列表获取单元 144，用于获取目标缓存区域对应的第二访问记录列表；第二访问记录列表包含至少两个缓存合约数据在目标时间段内，分别对应的数据访问次数；目标缓存区域包含至少两个缓存合约数据；

第二数据确定单元 145，用于将数据访问次数最少的缓存合约数据，作为目标缓存区域中的目标淘汰合约数据；

第二数据淘汰单元 146，用于释放目标淘汰合约数据，将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

其中，第二列表获取单元 144、第二数据确定单元 145 以及第二数据淘汰单元 146 的具体功能实现方式可以参见图 3 对应实施例中的 S103 的具体描述。

请再参见图 7，数据淘汰模块 14 可以包括：第三列表获取单元 147、第三数据确定单元 148 以及第三数据淘汰单元 149。

第三列表获取单元 147，用于获取目标缓存区域对应的数据存储记录列表；数据存储记录列表包含至少两个缓存合约数据分别对应的数据缓存时间；目标缓存区域包含至少两个缓存合约数据；

第三数据确定单元 148，用于将数据缓存时间最早的缓存合约数据，作为目标缓存区域中的目标淘汰合约数据；

第三数据淘汰单元 149，用于释放目标淘汰合约数据，将释放目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

其中，第三列表获取单元 147、第三数据确定单元 148 以及第三数据淘汰单元 149 的具体功能实现方式可以参见图 3 对应实施例中的 S103 的具体描述。

请再参见图 7，上述基于区块链的数据处理装置 1，还可以包括：查询接收模块 17 以及数据查询模块 18。

查询接收模块 17，用于通过缓存路由组件接收数据查询请求；数据查询请求包含待执行合约的合约标识；

数据查询模块 18，用于通过缓存路由组件，在至少两个缓存区域中查找与待执行合约的合约标识相关联的查询缓存数据。

其中，查询接收模块 17 以及数据查询模块 18 的具体功能实现方式可以参见图 5 对应实施例中的步骤 S201-步骤 S202 的具体描述。

其中，至少两个缓存区域包括通用合约缓存区域和 N 个定制合约缓存区域；N 个定制合约缓存区域包括目标缓存区域；N 为大于或等于 0 的整数；通用合约缓存区域中包括一个或多个通用合约分别对应的缓存合约数据；一个定制合约缓存区域中存储的缓存合约数据和一个定制合约缓存区域相关联的定制合约对应；

请再参见图 7，数据查询模块 18 可以包括：定制标识获取单元 181、定制标识查找单元 182 以及第一数据查询单元 183。

定制标识获取单元 181，用于通过缓存路由组件，获取与 N 个定制合约缓存区域相关联的定制合约对应的定制合约标识；

定制标识查找单元 182，用于在定制合约标识中查找待执行合约的合约标识；

第一数据查询单元 183，用于若在定制合约标识中查找到与待执行合约的合约标识相同的定制合约标识，则通过缓存路由组件在与待执行合约的合约标识对应的定制合约关联的定制合约缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据；

第一数据查询单元 183，还用于若在定制合约标识中未查找到与待执行合约的合约标识相同的定制合约标识，则通过缓存路由组件在通用合约缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据。

其中，定制标识获取单元 181、定制标识查找单元 182 以及第一数据查

询单元 183 的具体功能实现方式可以参见图 5 对应实施例中的步骤 S202 的具体描述。

其中，数据查询请求还包含调用待执行合约的待执行交易；至少两个缓存区域还包括区块缓存区域和交易缓存区域；区块缓存区域用于存储未写入区块链账本中的待上链区块对应的区块缓存数据；交易缓存区域用于存储未写入区块链账本中的待上链交易对应的交易缓存数据；

请再参见图 7，数据查询模块 18 可以包括：第二数据查询单元 184。

第二数据查询单元 184，用于在缓存路由组件中，根据与待执行合约的合约标识和待执行交易对交易缓存区域中的交易缓存数据进行遍历；

第二数据查询单元 184，还用于若在交易缓存区域中遍历到与待执行合约的合约标识和待执行交易相关联的交易缓存数据，则将所遍历到的交易缓存数据，作为查询缓存数据；

第二数据查询单元 184，还用于若在交易缓存区域中未遍历到与待执行合约的合约标识和待执行交易相关联的交易缓存数据，则在缓存路由组件中，根据与待执行合约的合约标识和待执行交易对区块缓存区域中的区块缓存数据进行遍历；

第二数据查询单元 184，还用于若在区块缓存区域中遍历到与待执行合约的合约标识和待执行交易相关联的区块缓存数据，则将所遍历到的区块缓存数据，作为查询缓存数据；

第二数据查询单元 184，还用于若在区块缓存区域中未遍历到与待执行合约的合约标识和待执行交易相关联的区块缓存数据，则在区块链数据库中获取与待执行合约的合约标识相关联的查询缓存数据。

其中，第二数据查询单元 184 的具体功能实现方式可以参见图 5 对应实施例中的步骤 S202 的具体描述。

其中，待上链区块中包含一个或多个待上链交易；

请再参见图 7，上述基于区块链的数据处理装置 1，还可以包括：数据释放模块 19。

数据释放模块 19，用于当待上链区块写入区块链账本中时，从区块缓存区域中，释放待上链区块对应的区块缓存数据；

数据释放模块 19，还用于从交易缓存区域中，释放待上链区块中的一个或多个待上链交易对应的交易缓存数据。

其中，数据释放模块 19 的具体功能实现方式可以参见图 5 对应实施例中的步骤 S202 的可选描述。

其中，至少两个缓存区域还包括系统缓存区域；系统缓存区域中包括一个或多个系统合约分别对应的缓存合约数据；

请再参见图 7，数据查询模块 18 可以包括：系统标识获取单元 185、系统标识查找单元 186 以及第三数据查询单元 187。

系统标识获取单元 185，用于获取一个或多个系统合约对应的系统合约标识；

系统标识查找单元 186，用于在系统合约标识中查找待执行合约的合约标识；

第三数据查询单元 187，用于若在系统合约标识中查找到与待执行合约的合约标识相同的系统合约标识，则通过缓存路由组件在与待执行合约的合约标识对应的系统合约关联的系统缓存区域中，查找与待执行合约的合约标识相关联的查询缓存数据。

其中，系统标识获取单元 185、系统标识查找单元 186 以及第三数据查询单元 187 的具体功能实现方式可以参见图 5 对应实施例中的步骤 S202 的具体描述。

本申请实施例可以中，获取针对目标定制合约的申请缓存指令，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；然后，当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量，最后根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。其中，总缓存区域包括至少两个缓存区域，且每个缓存区域的缓存淘汰机制相互独立；新的可用缓存容量大于待缓存容量。通过本申请实施例提供的方法，可以为区块链系统中的合约分配专属的缓存区域，该缓存区域仅用于存储该合约的缓存合约数据，倘若缓存不足，也仅根据淘汰机制在该缓存区域中淘汰缓存合约数据，不会影响其他合约的缓存合约数据，因此，可以为区块链中需要频繁访问的合约分配专属的缓存区域，避免该合约对应的缓存合约数据被其余合约的缓存合约数据所影响而被淘汰，从而避免区块链节点频繁去硬盘中获取该合约对应的数据，从而提高了区块链节点的数据的访问效率。

请参见图 8，图 8 是本申请实施例提供的一种计算机设备的结构示意图。该计算机设备可以是执行本申请实施例提供的任一基于区块链的数据处理方法的设备。如图 8 所示，该计算机设备 1000 可以包括：处理器 1001，网络接口 1004 和存储器 1005，此外，上述计算机设备 1000 还可以包括：用户接口 1003，和至少一个通信总线 1002。其中，通信总线 1002 用于实现这些组件之间的连接通信。其中，用户接口 1003 可以包括显示屏（Display）、键盘（Keyboard），可选用户接口 1003 还可以包括标准的有线接口、无线接口。网络接口 1004 可选的可以包括标准的有线接口、无线接口（如 WI-FI 接口）。存储器 1004 可以是高速 RAM 存储器，也可以是非不稳定的存储器（non-volatile memory），例如至少一个磁盘存储器。存储器 1005 可选的还可以是至少一个位于远离前述处理器 1001 的存储装置。如图 8 所示，作为一种计算机可读存储介质的存储器 1005 中可以包括操作系统、网络通信模块、用户接口模块以及计算机可读指令。

在如图 8 所示的计算机设备 1000 中，网络接口 1004 可提供网络通讯网

元；而用户接口 1003 主要用于为用户提供输入的接口；而处理器 1001 可以用于调用存储器 1005 中存储的计算机可读指令，以实现：

获取针对目标定制合约的申请缓存指令，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；每个缓存区域的缓存淘汰机制相互独立；

当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；

若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；新的可用缓存容量大于待缓存容量；

根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。

应当理解，本申请实施例中所描述的计算机设备 1000 可执行前文任一个所对应实施例中对该基于区块链的数据处理方法的描述。另外，对采用相同方法的有益效果描述，也不再进行赘述。

此外，这里需要指出的是：本申请实施例还提供了一种非易失性的计算机可读存储介质，且上述计算机可读存储介质中存储有前文提及的基于区块链的数据处理装置 1 所执行的计算机可读指令，当上述处理器执行上述计算机可读指令时，能够执行前文任一个所对应实施例中对上述基于区块链的数据处理方法的描述。另外，对采用相同方法的有益效果描述，也不再进行赘述。对于本申请所涉及的计算机可读存储介质实施例中未披露的技术细节，请参照本申请方法实施例的描述。

上述非易失性的计算机可读存储介质可以是前述任一实施例提供的基于区块链的数据处理装置或者上述计算机设备的内部存储单元，例如计算机设备的硬盘或内存。该非易失性的计算机可读存储介质也可以是该计算机设备的外部存储设备，例如该计算机设备上配备的插接式硬盘，智能存储卡（smart media card，SMC），安全数字（secure digital，SD）卡，闪存卡（flash card）等。进一步地，该计算机可读存储介质还可以既包括该计算机设备的内部存储单元也包括外部存储设备。该计算机可读存储介质用于存储该计算机可读指令以及该计算机设备所需的其他程序和数据。该计算机可读存储介质还可以用于暂时地存储已经输出或者将要输出的数据。

此外，这里需要指出的是：本申请实施例还提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机可读指令，该计算机可读指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机可读指令，处理器执行该计算机可读指令，使得该计算机设备执行前文任一个所对应实施例提供的方法。

本申请实施例的说明书和权利要求书及附图中的术语“第一”、“第二”等是用于区别不同对象，而非用于描述特定顺序。此外，术语“包括”以及它们任何变形，意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、装置、产品或设备没有限定于已列出的步骤或模块，而是可

选地还包括没有列出的步骤或模块，或可选地还包括对于这些过程、方法、装置、产品或设备固有的其他步骤单元。

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、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照网元一般性地描述了各示例的组成及步骤。这些网元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网元，但是这种实现不应认为超出本申请的范围。

以上所揭露的仅为本申请较佳实施例而已，当然不能以此来限定本申请之权利范围，因此依本申请权利要求所作的等同变化，仍属本申请所涵盖的范围。

权利要求书

1、一种基于区块链的数据处理方法，由计算机设备执行，所述方法包括：

获取针对目标定制合约的申请缓存指令，根据所述申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与所述目标定制合约关联的缓存区域；所述总缓存区域包括至少两个缓存区域；所述总缓存区域中的每个缓存区域的缓存淘汰机制相互独立；

当接收到所述目标定制合约对应的待缓存合约数据时，在所述至少两个缓存区域中查找与所述目标定制合约关联的缓存区域，作为目标缓存区域；

若所述目标缓存区域的可用缓存容量小于所述待缓存合约数据的待缓存容量，则根据所述目标缓存区域的缓存淘汰机制对所述目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；所述新的可用缓存容量大于所述待缓存容量；

根据所述新的可用缓存容量将所述待缓存合约数据存储进所述目标缓存区域。

2、根据权利要求1所述的方法，其特征在于，所述获取针对目标定制合约的申请缓存指令，包括：

获取申请缓存请求；所述申请缓存请求包括申请缓存合约标识、与所述目标定制合约相关联的定制缓存创建参数；

调用所述申请缓存请求中的所述申请缓存合约标识所指示的申请缓存合约；

在合约虚拟机中，根据所述定制缓存创建参数执行所述申请缓存合约，得到针对所述目标定制合约的申请缓存指令；所述申请缓存指令包含所述定制缓存创建参数。

3、根据权利要求1所述的方法，其特征在于，所述方法还包括：

在验证出所述申请缓存指令满足缓存申请条件时，执行所述根据所述申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与所述目标定制合约关联的缓存区域的步骤；

其中，验证所述申请缓存指令是否满足缓存申请条件的步骤包括：

当在共识网络中对所述申请缓存指令共识通过时，通过缓存审批组件获取针对所述申请缓存指令的一个或多个缓存审批结果；每个缓存审批结果均为经过所述共识网络共识通过后的数据；

当所述一个或多个缓存审批结果中，属于缓存审批通过的缓存审批结果的数量超过审批通过阈值时，确定所述申请缓存指令满足缓存申请条件。

4、根据权利要求1所述的方法，其特征在于，所述定制缓存创建参数包括所述目标定制合约的合约标识和预分配缓存容量；

所述根据所述申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与所述目标定制合约关联的缓存区域，包括：

在总缓存区域中获取与所述预分配缓存容量对应的预分配缓存区域；

将所述预分配缓存区域和所述目标定制合约的合约标识进行映射处理，得到与所述目标定制合约关联的缓存区域。

5、根据权利要求1所述的方法，其特征在于，所述根据所述目标缓存区域的缓存淘汰机制对所述目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量，包括：

获取所述目标缓存区域对应的第一访问记录列表；所述第一访问记录列表包含至少两个缓存合约数据分别对应的数据访问最近时间；所述目标缓存区域包含所述至少两个缓存合约数据；

将数据访问最近时间最早的缓存合约数据，作为所述目标缓存区域中的目标淘汰合约数据；

释放所述目标淘汰合约数据，将释放所述目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

6、根据权利要求1所述的方法，其特征在于，所述根据所述目标缓存区域的缓存淘汰机制对所述目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量，包括：

获取所述目标缓存区域对应的第二访问记录列表；所述第二访问记录列表包含至少两个缓存合约数据在目标时间段内，分别对应的数据访问次数；所述目标缓存区域包含所述至少两个缓存合约数据；

将数据访问次数最少的缓存合约数据，作为所述目标缓存区域中的目标淘汰合约数据；

释放所述目标淘汰合约数据，将释放所述目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

7、根据权利要求1所述的方法，其特征在于，所述根据所述目标缓存区域的缓存淘汰机制对所述目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量，包括：

获取所述目标缓存区域对应的数据存储记录列表；所述数据存储记录列表包含至少两个缓存合约数据分别对应的数据缓存时间；所述目标缓存区域包含所述至少两个缓存合约数据；

将数据缓存时间最早的缓存合约数据，作为所述目标缓存区域中的目标淘汰合约数据；

释放所述目标淘汰合约数据，将释放所述目标淘汰合约数据后的目标缓存区域的可用缓存容量，作为新的可用缓存容量。

8、根据权利要求1所述的方法，其特征在于，所述方法还包括：

通过缓存路由组件接收数据查询请求；所述数据查询请求包含待执行合约的合约标识；

通过所述缓存路由组件，在所述至少两个缓存区域中查找与所述待执行合约的合约标识相关联的查询缓存数据。

9、根据权利要求8所述的方法，其特征在于，所述至少两个缓存区域包括通用合约缓存区域和N个定制合约缓存区域；所述N个定制合约缓存区域包括所述目标缓存区域；N为大于或等于0的整数；所述通用合约缓存区域中包括一个或多个通用合约分别对应的缓存合约数据；一个定制合约缓存区域中存储的缓存合约数据和所述一个定制合约缓存区域相关联的定制合约对应；

所述通过所述缓存路由组件，在所述至少两个缓存区域中查找与所述待执行合约的合约标识相关联的查询缓存数据，包括：

通过所述缓存路由组件，获取与所述 N 个定制合约缓存区域相关联的定制合约对应的定制合约标识；

在所述定制合约标识中查找所述待执行合约的合约标识；

若在所述定制合约标识中查找到与所述待执行合约的合约标识相同的定制合约标识，则通过所述缓存路由组件在与所述待执行合约的合约标识对应的定制合约关联的定制合约缓存区域中，查找与所述待执行合约的合约标识相关联的查询缓存数据；

若在所述定制合约标识中未查找到与所述待执行合约的合约标识相同的定制合约标识，则通过所述缓存路由组件在所述通用合约缓存区域中，查找与所述待执行合约的合约标识相关联的查询缓存数据。

10、根据权利要求 8 所述的方法，其特征在于，所述数据查询请求还包含调用所述待执行合约的待执行交易；所述至少两个缓存区域还包括区块缓存区域和交易缓存区域；所述区块缓存区域用于存储未写入区块链账本中的待上链区块对应的区块缓存数据；所述交易缓存区域用于存储未写入区块链账本中的待上链交易对应的交易缓存数据；

所述通过所述缓存路由组件，在所述至少两个缓存区域中查找与所述待执行合约的合约标识相关联的查询缓存数据，包括：

在所述缓存路由组件中，根据与所述待执行合约的合约标识和所述待执行交易对所述交易缓存区域中的交易缓存数据进行遍历；

若在所述交易缓存区域中遍历到与所述待执行合约的合约标识和所述待执行交易相关联的交易缓存数据，则将所遍历到的交易缓存数据，作为查询缓存数据；

若在所述交易缓存区域中未遍历到与所述待执行合约的合约标识和所述待执行交易相关联的交易缓存数据，则在所述缓存路由组件中，根据与所述待执行合约的合约标识和所述待执行交易对所述区块缓存区域中的区块缓存数据进行遍历；

若在所述区块缓存区域中遍历到与所述待执行合约的合约标识和所述待执行交易相关联的区块缓存数据，则将所遍历到的区块缓存数据，作为查询缓存数据；

若在所述区块缓存区域中未遍历到与所述待执行合约的合约标识和所述待执行交易相关联的区块缓存数据，则在区块链数据库中获取与所述待执行合约的合约标识相关联的查询缓存数据。

11、根据权利要求 10 所述的方法，其特征在于，所述待上链区块中包含一个或多个待上链交易；

所述方法还包括：

当所述待上链区块写入区块链账本中时，从所述区块缓存区域中，释放所述待上链区块对应的区块缓存数据；

从所述交易缓存区域中，释放所述待上链区块中的所述一个或多个待上链

交易对应的交易缓存数据。

12、根据权利要求 8 所述的方法，其特征在于，所述至少两个缓存区域还包括系统缓存区域；所述系统缓存区域中包括一个或多个系统合约分别对应的缓存合约数据；

所述通过所述缓存路由组件，在所述至少两个缓存区域中查找与所述待执行合约的合约标识相关联的查询缓存数据，包括：

获取所述一个或多个系统合约对应的系统合约标识；

在所述系统合约标识中查找所述待执行合约的合约标识；

若在所述系统合约标识中查找到与所述待执行合约的合约标识相同的系统合约标识，则通过所述缓存路由组件在与所述待执行合约的合约标识对应的系统合约关联的系统缓存区域中，查找与所述待执行合约的合约标识相关联的查询缓存数据。

13、一种基于区块链的数据处理装置，所述装置包括：

指令获取模块，用于获取针对目标定制合约的申请缓存指令；

缓存创建模块，用于根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域；总缓存区域包括至少两个缓存区域；所述总缓存区域中的每个缓存区域的缓存淘汰机制相互独立；

缓存查找模块，用于当接收到目标定制合约对应的待缓存合约数据时，在至少两个缓存区域中查找与目标定制合约关联的缓存区域，作为目标缓存区域；

数据淘汰模块，用于若目标缓存区域的可用缓存容量小于待缓存合约数据的待缓存容量，则根据目标缓存区域的缓存淘汰机制对目标缓存区域中的缓存合约数据进行淘汰处理，得到新的可用缓存容量；新的可用缓存容量大于待缓存容量；

数据存储模块，用于根据新的可用缓存容量将待缓存合约数据存储进目标缓存区域。

14、根据权利要求 13 所述的装置，其特征在于，所述指令获取模块，包括：

请求获取单元，用于获取申请缓存请求；所述申请缓存请求包括申请缓存合约标识、与所述目标定制合约相关联的定制缓存创建参数；

合约调用单元，用于调用所述申请缓存请求中的所述申请缓存合约标识所指示的申请缓存合约；

指令生成单元，用于在合约虚拟机中，根据所述定制缓存创建参数执行所述申请缓存合约，得到针对所述目标定制合约的申请缓存指令；所述申请缓存指令包含所述定制缓存创建参数。

15、根据权利要求 13 所述的装置，其特征在于，所述装置还包括：

审批模块，用于当在共识网络中对所述申请缓存指令共识通过时，通过缓存审批组件获取针对所述申请缓存指令的一个或多个缓存审批结果；每个缓存审批结果均为经过所述共识网络共识通过后的数据；

所述审批模块，还用于当所述一个或多个缓存审批结果中，属于缓存审批通过的缓存审批结果的数量超过审批通过阈值时，确定所述申请缓存指令满足缓存申请条件；

所述缓存创建模块，还用于在所述申请缓存指令满足缓存申请条件时，根据申请缓存指令中的定制缓存创建参数，在总缓存区域中创建与目标定制合约关联的缓存区域。

16、根据权利要求13所述的装置，其特征在于，所述定制缓存创建参数包括所述目标定制合约的合约标识和预分配缓存容量；

所述缓存创建模块，包括：

预分配单元，用于在总缓存区域中获取与所述预分配缓存容量对应的预分配缓存区域；

映射单元，用于将所述预分配缓存区域和所述目标定制合约的合约标识进行映射处理，得到与所述目标定制合约关联的缓存区域。

17、根据权利要求13所述的装置，其特征在于，所述装置还包括：

查询接收模块，用于通过缓存路由组件接收数据查询请求；所述数据查询请求包含待执行合约的合约标识；

数据查询模块，用于通过所述缓存路由组件，在所述至少两个缓存区域中查找与所述待执行合约的合约标识相关联的查询缓存数据。

18、一种计算机设备，包括：处理器、存储器以及网络接口；

所述处理器与所述存储器、所述网络接口相连，存储器存储有计算机可读指令，所述计算机可读指令被处理器执行时，使得所述处理器执行如权利要求1-12任一项所述的方法。

19、一种非易失性的计算机可读存储介质，所述计算机可读存储介质中存储有计算机可读指令，所述计算机可读指令适于由处理器加载并执行权利要求1-12任一项所述的方法。

20、一种计算机程序产品，所述计算机程序产品包括计算机可读指令，所述计算机可读指令存储在计算机可读存储介质中，计算机设备的处理器从所述计算机可读存储介质读取所述计算机可读指令，所述处理器执行所述计算机可读指令，实现权利要求1-12任一项所述的方法。