

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2021 年 9 月 16 日 (16.09.2021)



(10) 国际公布号
WO 2021/179809 A1

(51) 国际专利分类号:
G06F 9/455 (2006.01)

(21) 国际申请号: PCT/CN2021/073557

(22) 国际申请日: 2021 年 1 月 25 日 (25.01.2021)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202010172887.8 2020 年 3 月 13 日 (13.03.2020) CN

(71) 申请人: 支付宝 (杭州) 信息技术有限公司 (ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) [CN/CN]; 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。

(72) 发明人: 郭学鹏(GUO, Xuepeng); 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。 姚忠孝(YAO, Zhongxiao); 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。 李东威(LI, Dongwei); 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。 林志平(LIN, Zhiping); 中国浙江省杭州市西湖区西溪路 556 号 8 层 B 段 801-11, Zhejiang 310000 (CN)。

(74) 代理人: 北京博思佳知识产权代理有限公司 (BEIJING BESTIPR INTELLECTUAL PROPERTY LAW CORPORATION); 中国北京市海淀区上地三街 9 号嘉华大厦 B 座 409, Beijing 100085 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

(54) Title: INTELLIGENT CONTRACT CACHING AND EXECUTING METHOD AND APPARATUS

(54) 发明名称: 缓存和执行智能合约的方法和装置

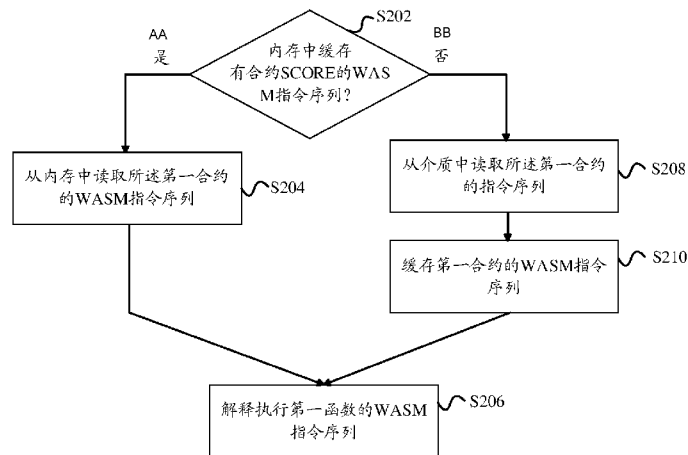


图 2

S202 WHETHER A WASM INSTRUCTION SEQUENCE OF A CONTRACT SCORE IS CACHED IN A MEMORY
S204 READ THE WASM INSTRUCTION SEQUENCE OF A FIRST CONTRACT FROM THE MEMORY
S206 INTERPRET AND EXECUTE A WASM INSTRUCTION SEQUENCE OF A FIRST FUNCTION
S208 READ THE INSTRUCTION SEQUENCE OF THE FIRST CONTRACT FROM A MEDIUM
S210 CACHE THE WASM INSTRUCTION SEQUENCE OF THE FIRST CONTRACT
AA YES
BB NO

(57) Abstract: The embodiments of the present description provide an intelligent contract caching method and apparatus. The method is executed when a virtual machine of a blockchain node executes a first transaction, wherein the first transaction is used for calling a first function of a first contract. The method comprises: determining whether a WASM instruction sequence of the first contract is cached in a memory; when it is determined that the WASM instruction sequence of the first contract is cached in the memory, reading the WASM instruction sequence of the first contract from the memory to obtain a WASM instruction sequence of the first function; and on the basis of a data field in the first transaction, interpreting and executing the WASM instruction sequence of the first function.

BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告 (条约第21条(3))。

(57) 摘要: 本说明书实施例提供了一种智能合约缓存方法和装置, 所述方法由区块链节点的虚拟机执行第一交易时执行, 所述第一交易用于调用第一合约的第一函数, 所述方法包括: 确定内存中是否缓存有所述第一合约的WASM指令序列; 在确定内存中缓存了所述第一合约的WASM指令序列的情况下, 从内存中读取所述第一合约的WASM指令序列, 以获取所述第一函数的WASM指令序列; 基于所述第一交易中的数据字段解释执行所述第一函数的WASM指令序列。

缓存和执行智能合约的方法和装置

技术领域

[01] 本说明书一个或多个实施例涉及区块链技术领域，更具体地，涉及在区块链中缓存和执行智能合约的方法和装置。

背景技术

[02] 虚拟机 (Virtual Machine) 是通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。由于虚拟机可以隔离底层硬件平台以及操作系统对上层应用的影响，因此非常有利于上层应用的开发。上层应用开发过程中无需关注底层平台的细节，只需要关注具体的业务逻辑。开发完成后，由虚拟机运行上层应用，负责将应用的代码转换为适于底层平台执行的代码。具体地，在许多场景中，上层应用由开发人员使用高级语言编写开发，之后通过编译器编译为字节码 (bytecode)。字节码是一种执行程序，由一序列操作码 (op 码)-数据对组成的二进制文件，是一种中间码。然后，虚拟机中的解释器对字节码代表的指令流进行解释和执行。

[03] 例如，在支持智能合约的区块链应用场景中，可以在区块链网络的每个节点中部署虚拟机。用户可以用高级语言编写智能合约，然后经由编译器编译为字节码之后，将该字节码包含在创建智能合约的交易中，发布到区块链网络中，也就是部署到区块链网络的各个节点中。在需要执行智能合约时，由各个节点中的虚拟机对该字节码进行解释执行。

[04] 随着 WASM 语言的出现，越来越多的区块链中使用 WASM 作为智能合约的编写语言，相对应地，使用 WASM 虚拟机来执行智能合约。WASM 即 WebAssembly，是由 W3C 社区组织推出的一种为栈式虚拟机设计的二进制指令形式，是一种新的平台无关的中间字节码格式。WASM 虚拟机为栈式虚拟机，其中，将所有指令的操作数都放到栈上，即，每次操作都从栈上获取操作数，并在指令执行结束后将结果压入栈中。

[05] 在通过区块链各个节点的 WASM 虚拟机执行智能合约的应用场景中，WASM 虚拟机解释器对合约的执行速度对于整个系统的性能都至关重要。因此，希望能有改进的方案，进一步提高 WASM 虚拟机对智能合约的执行效率。

发明内容

[06] 本说明书实施例旨在提供一种更有效的缓存和执行智能合约的方案，以解决现有技术中的不足。

[07] 为实现上述目的，本说明书一个方面提供一种智能合约执行方法，所述方法由区块链节点的虚拟机执行第一交易时执行，所述第一交易用于调用第一合约的第一函数，所述方法包括：确定内存中是否缓存有所述第一合约的 WASM 指令序列；在确定内存中缓存了所述第一合约的 WASM 指令序列的情况下，从内存中读取所述第一合约的 WASM 指令序列，以获取所述第一函数的 WASM 指令序列；基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[08] 在一种实施方式中，所述方法还包括，在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况下，从所述节点本地存储介质中读取所述第一合约的指令序列，以获取所述第一合约的 WASM 指令序列；将所述第一合约的 WASM 指令序列与所述第一

合约的标识关联地缓存到内存中。

[09] 在一种实施方式中，从所述节点本地存储介质中读取第一合约的指令序列，以获取所述第一合约的 WASM 指令序列包括，从节点本地存储介质中读取第一合约的经可变量编码的指令序列，通过对第一合约的经可变量编码的指令序列进行解码，获取所述第一合约的 WASM 指令序列。

[10] 在一种实施方式中，所述方法还包括，在获取第一合约的 WASM 指令序列之后，对第一合约的 WASM 指令序列进行预处理，以获取第一合约的经预处理的 WASM 指令序列，其中，将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中包括，将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中。

[11] 在一种实施方式中，对第一合约的 WASM 指令序列进行预处理包括，基于所述第一合约的 WASM 指令序列确定全局变量标识范围；对于所述第一合约的 WASM 指令序列中的每个指令，解析该指令是否为全局变量访问指令；在解析出所述指令为全局变量访问指令的情况下，检查该指令包括的全局变量标识是否在所述全局变量标识范围内；在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况下，获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

[12] 在一种实施方式中，对第一合约的 WASM 指令序列进行预处理包括，确定所述 WASM 指令序列中是否包括预定连续指令，所述预定连续指令被预设有相应的合并指令，并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果；在确定所述指令序列中包括至少一组预定连续指令的情况下，将所述指令序列中的各组预定连续指令分别合并为对应的合并指令，从而获取所述第一合约的经指令融合的 WASM 指令序列。

[13] 在一种实施方式中，所述方法还包括，在将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中之后，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[14] 在一种实施方式中，将所述第一合约的 WASM 指令序列与第一合约的标识关联地缓存到内存中包括，将所述第一合约的 WASM 指令序列在内存中的存储地址与第一合约的合约地址的哈希值关联地存储到内存中。

[15] 本说明书另一方面提供一种智能合约执行方法，所述方法由区块链节点的虚拟机执行第一交易时执行，所述第一交易用于调用第一合约的第一函数，所述方法包括：确定内存中是否缓存有所述第一函数的 WASM 指令序列；在确定内存中缓存了所述第一函数的 WASM 指令序列的情况下，从内存中读取所述第一函数的 WASM 指令序列；基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[16] 在一种实施方式中，所述方法还包括，在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况下，从所述节点本地存储介质中读取所述第一合约的指令序列，以获取所述第一函数的 WASM 指令序列；将所述第一函数的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中。

[17] 在一种实施方式中，将所述第一函数的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中包括，将所述第一函数的 WASM 指令序列在内存中的存储地址与第

一数据的哈希值关联地存储到内存中，其中，所述第一数据基于第一合约的合约地址和第一函数的名称获取。

[18] 本说明书另一方面提供一种智能合约执行装置，所述装置在区块链节点的虚拟机执行第一交易时部署，所述第一交易用于调用第一合约的第一函数，所述装置包括：确定单元，配置为，确定内存中是否缓存有所述第一合约的 WASM 指令序列；第一读取单元，配置为，在确定内存中缓存了所述第一合约的 WASM 指令序列的情况下，从内存中读取所述第一合约的 WASM 指令序列，以获取所述第一函数的 WASM 指令序列；第一执行单元，配置为，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[19] 在一种实施方式中，所述装置还包括，第二读取单元，配置为，在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况下，从所述节点本地存储介质中读取第一合约的指令序列，以获取所述第一合约的 WASM 指令序列；缓存单元，配置为，将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中。

[20] 在一种实施方式中，所述第二读取单元还配置为，从节点本地存储介质中读取第一合约的经可变长编码的指令序列，通过对第一合约的经可变长编码的指令序列进行解码，获取所述第一合约的 WASM 指令序列。

[21] 在一种实施方式中，所述装置还包括，预处理单元，配置为，在获取第一合约的 WASM 指令序列之后，对第一合约的 WASM 指令序列进行预处理，以获取第一合约的经预处理的 WASM 指令序列，其中，所述缓存单元还配置为，将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中。

[22] 在一种实施方式中，所述预处理单元包括，第一确定子单元，配置为，基于所述第一合约的 WASM 指令序列确定全局变量标识范围；解析子单元，配置为，对于所述第一合约的 WASM 指令序列中的每个指令，解析该指令是否为全局变量访问指令；检查子单元，配置为，在解析出所述指令为全局变量访问指令的情况下，检查该指令包括的全局变量标识是否在所述全局变量标识范围内；获取子单元，配置为，在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况下，获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

[23] 在一种实施方式中，所述预处理单元包括，第二确定子单元，配置为，确定所述 WASM 指令序列中是否包括预定连续指令，所述预定连续指令被预设有相应的合并指令，并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果；合并子单元，配置为，在确定所述指令序列中包括至少一组预定连续指令的情况下，将所述指令序列中的各组预定连续指令分别合并为对应的合并指令，从而获取所述第一合约的经指令融合的 WASM 指令序列。

[24] 在一种实施方式中，所述装置还包括，第二执行单元，配置为，在将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中之后，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[25] 在一种实施方式中，所述缓存单元还配置为，将所述第一合约的 WASM 指令序列在内存中的存储地址与第一合约的合约地址的哈希值关联地存储到内存中。

[26] 本说明书另一方面提供一种智能合约执行装置，所述装置在区块链节点的虚拟机

执行第一交易时部署,所述第一交易用于调用第一合约的第一函数,所述装置包括:确定单元,配置为,确定内存中是否缓存有所述第一函数的 WASM 指令序列;第一读取单元,配置为,在确定内存中缓存了所述第一函数的 WASM 指令序列的情况下,从内存中读取所述第一函数的 WASM 指令序列;执行单元,配置为,基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[27] 在一种实施方式中,所述装置还包括,第二读取单元,配置为,在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况下,从所述节点本地存储介质中读取所述第一合约的指令序列,以获取所述第一函数的 WASM 指令序列;缓存单元,配置为,将所述第一函数的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中。

[28] 在一种实施方式中,所述缓存单元还配置为,将所述第一函数的 WASM 指令序列在内存中的存储地址与第一数据的哈希值关联地存储到内存中,其中,所述第一数据基于第一合约的合约地址和第一函数的名称获取。

[29] 本说明书另一方面提供一种计算机可读存储介质,其上存储有计算机程序,当所述计算机程序在计算机中执行时,令计算机执行上述任一项方法。

[30] 本说明书另一方面提供一种计算设备,包括存储器和处理器,所述存储器中存储有可执行代码,所述处理器执行所述可执行代码时,实现上述任一项方法。

[31] 在根据本说明书实施例的智能合约缓存和执行方案,通过虚拟机在执行交易时将获取的合约的指令序列缓存到内存中,从而后续交易可直接从内存中读取该合约的指令序列,而不需要从节点存储介质中读取该指令序列,也不需要再对编码的指令序列重复解码,并且不需要对指令序列重复进行预处理,缩短了后续交易的执行时间,提升了系统性能。

附图说明

[32] 通过结合附图描述本说明书实施例,可以使得本说明书实施例更加清楚:

[33] 图 1 是在区块链网络中调用合约 SCORE 的过程示意图;

[34] 图 2 示出根据本说明书一个实施例的智能合约执行方法流程图;

[35] 图 3 示意示出了一种记录合约与地址键值对的表格示例;

[36] 图 4 示出在节点中缓存和执行合约 SCORE 的示意图;

[37] 图 5 示出根据本说明书另一实施例的智能合约缓存方法流程图;

[38] 图 6 示出根据本说明书另一实施例的智能合约缓存方法流程图;

[39] 图 7 示出根据本说明书另一实施例的一种智能合约执行方法流程图;

[40] 图 8 示出根据本说明书一个实施例的一种智能合约执行装置 800;

[41] 图 9 示出根据本说明书另一实施例的一种智能合约执行装置 900。

具体实施方式

[42] 下面将结合附图描述本说明书实施例。

[43] 如本领域技术人员所知,区块链网络中包含若干节点,各个节点之间可以进行通信。在例如联盟链的区块链中,用户可以通过客户端向区块链中任一节点发送交易,交易既可以是普通的转账交易,也可以用于创建智能合约或调用智能合约。假设区块链的用户 Bob 想要发布智能合约 Contract SCORE,该智能合约例如通过如下的公式(1)基于平台中用户的消费金额 x 和销售金额 y 计算用户的奖励分值 s :

$$S = (x + y * 2) * 3 \quad (1)$$

[44] Bob 例如可使用 C++ 等高级语言编写该智能合约，然后通过编译器将该 C++ 智能合约转换成如下所示的 WASM 指令序列：

```
Contract SCORE
    function calculation(param x i32) (param y i32){
        get_local y
        i32.const 2
        i32.mul
        get_local x
        i32.add
        i32.const 3
        i32.mul
    }
```

[45] 可以理解，为了便于阅读，上述 WASM 指令序列以文本的形式显示，在实际运行中，WASM 指令序列以二进制的格式表示，从而上述 function calculation 具有如下的形式（这里为了简化表示，以十六进制数示出二进制码）：

```
{20 y
41 02
6c
20 x
6a
41 03
6c }
```

[46] 然后，Bob 可通过其客户端向区块链中任一节点发布用于创建该合约 SCORE 的交易，从而区块链网络中的各个节点均可以获取到上述智能合约的字节码，从而区块链网络的各个节点可将该智能合约部署到本地介质中。

[47] 在现有技术中，在各个节点部署了合约 SCORE 之后，在后期每次执行调用合约 SCORE 的交易时，都要从本地介质中获取合约 SCORE 的上述指令序列，并顺序执行上述指令序列。在交易 1 中的指令序列为经 leb 128 编码的指令序列的情况中，还需要在从介质中获取合约的指令序列之后，先进行 leb 128 解码，然后再顺序执行合约的指令序列。每次执行调用合约 SCORE 的交易时从存储介质中读取合约并解码的过程大大增加了交易执行时间。

[48] 为了减少交易执行时间，在本说明书实施例中，在执行调用合约 SCORE 的交易的过程中，通过 WASM 虚拟机从本地存储介质中获取合约 SCORE 的经 leb 128 编码的指令序列，对该编码的指令序列进行解码，并将该解码的指令序列缓存入内存中，以供后续的合约调用。从而在后续进行合约调用时，一方面，可从内存直接获取合约 SCORE 的指令序列，而不需要从存储介质中获取指令序列，从而减少了读取时间，另一方面，省去了对编码的指令序列进行解码的过程，从而大大减少合约执行时间。

[49] 图 1 是在区块链网络 100 中调用合约 SCORE 中函数 calculation 的过程示意图，其中区块链网络 100 包括节点 A ~ D，区块链的用户可通过客户端连接节点 A ~ D 中任一节点。假定区块链的用户 Alice 想要调用合约 SCORE 中的函数 calculation，Alice 可以

通过其客户端向区块链中的节点 A 发送例如交易 1 (Tx 1)，进行合约调用。

[50] 具体而言，在交易 1 中，from 字段可以是调用者 Alice 的地址，to 字段即为上述 SCORE 合约的合约名称和合约地址，表示该交易所调用的智能合约。在 Data 字段，包含要调用的合约中的方法或函数名称 (calculation) 和传入的参数值 (Alice 的消费金额 $x=50$ 和销售金额 $y=100$)。

[51] 在 Alice 将交易 1 发布到区块链中去之后，各个节点都可以获取交易 1，通过 WASM 虚拟机执行合约 SCORE 中的函数 calculation，从而返回合约执行结果。在一种情况中，例如，节点 A 内存中没有缓存的合约 SCORE 的 WASM 指令序列，从而节点 A 从本地存储介质中获取合约 SCORE 的经编码的字节码序列，对该字节码序列进行解码，将经解码的指令序列缓存到内存中，并解释执行该经解码的指令序列，从而使得后续的调用该合约的交易可以直接执行该合约的经解码的指令序列。在另一种情况中，例如，例如节点 D 在先前已经执行了其它调用合约 SCORE 的交易，从而本地的内存中缓存有合约 SCORE 的 WASM 指令序列，从而，节点 D 在执行该调用合约 SCORE 的交易时，从内存中获取合约 SCORE 的指令序列，并解释执行其中的函数 calculation 的指令序列。

[52] 可以理解，图 1 及相应的描述仅仅是示意性的，而不是限制性的，例如，所述区块链网络不限于为联盟链网络，而也可以为公链网络，等等。

[53] 下文将详细描述上述智能合约缓存和执行方案。

[54] 图 2 示出根据本说明书一个实施例的智能合约执行方法流程图，所述方法由区块链任一节点的 WASM 虚拟机执行，包括以下步骤。

[55] 步骤 S202，确定内存中是否缓存有合约 SCORE 的 WASM 指令序列。

[56] 该方法例如由图 1 中的节点 A 或节点 D 中的虚拟机执行。节点在接收到包括上述交易的区块之后，相对于交易 1 实例化虚拟机来执行交易 1。虚拟机在执行交易 1 的过程中，需要获取到合约 SCORE 的 WASM 指令序列，以执行其中的函数 calculation。

[57] 可通过多种方式确定内存中是否缓存有合约 SCORE 的 WASM 指令序列。

[58] 在一种实施方式中，在内存中记录有缓存的合约 SCORE 与存储合约 SCORE 的 WASM 指令序列的内存地址的关联关系，例如通过表格的形式将合约 SCORE 的合约地址哈希值与合约 SCORE 的内存存储地址关联起来，从而，通过在该表格中查询是否存储有合约 SCORE 的合约地址哈希值，可确定内存中是否缓存有合约 SCORE 的 WASM 指令序列。

[59] 在一种实施方式中，可通过在内存中搜索所述合约 SCORE 的合约地址，从而确定内存中是否缓存有合约 SCORE 的 WASM 指令序列。

[60] 步骤 S204，在确定内存中缓存了所述第一合约的 WASM 指令序列的情况中，从内存中读取所述第一合约的 WASM 指令序列，以获取所述第一函数的 WASM 指令序列。

[61] 例如图 1 中的节点 D 在确定内存中缓存有合约 SCORE 的 WASM 指令序列的情况中，在节点 D 中设置的虚拟机可直接从内存读取合约 SCORE 的 WASM 指令序列。例如，如在上文中所述，当在所述表格中查询到存储有合约 SCORE 的合约地址哈希值之后，根据该表格，可读取与该哈希值关联存储的合约 SCORE 在内存中的存储地址，从而可从该存储地址读取到合约 SCORE 的 WASM 指令序列。在另一种实施方式中，当在内存中搜索到合约 SCORE 的合约地址之后，例如，可预先将合约地址与合约代码关

联地存储在一起，从而在搜索到合约 SCORE 的合约地址之后，即找到了合约 SCORE 的 WASM 指令序列的存储地址，从而可读取合约 SCORE 的 WASM 指令序列。在获取到合约 SCORE 的 WASM 指令序列之后，可从该指令序列中读取函数 calculation 的指令序列，例如，通过搜索函数 calculation 的函数名称，从而可在合约 SCORE 的 WASM 指令序列中找到函数 calculation 的 WASM 指令序列。

[62] 步骤 S206, 基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[63] 在通过从内存中读取到函数 calculation 的 WASM 指令序列之后，基于交易 1 中的数据字段解释执行该合约中的函数 calculation 包括的指令序列，从而完成对交易 1 的执行。例如，如上文所述，交易 1 的数据字段中包含要调用的合约中的方法或函数名称（calculation）和传入的参数值（Alice 的消费金额 $x=50$ 和销售金额 $y=100$ ），从而将 $x=50$ 和 $y=100$ 作为传入参数执行函数 calculation 的 WASM 指令序列，从而函数返回的值即为分数值。

[64] 步骤 S208, 在确定内存中没有缓存所述合约 SCORE 的 WASM 指令序列的情况下，从所述节点本地存储介质中读取第一合约的指令序列，以获取所述第一合约的 WASM 指令序列。

[65] 例如，当上文中的节点 A 执行调用合约 SCORE 中某个函数的交易（例如交易 1）时，当确定内存中没有缓存合约 SCORE 的 WASM 指令序列的情况下，可从本地存储介质中找到预先部署的合约 SCORE 的指令序列。在节点的存储介质（例如硬盘）中以非结构化的键值对的形式存储了合约 SCORE 的指令序列，例如，可通过搜索所述合约 SCORE 的合约地址，而找到所述合约 SCORE 在介质中的存储地址。或者，与内存中的存储类似地，介质中存储有合约的合约地址哈希值与合约的存储地址的关联表格，通过从该表格中搜索合约 SCORE 的合约地址哈希值，从而可找到合约 SCORE 在介质中的存储地址，从而可从介质中读取到合约 SCORE 的指令序列。

[66] 在一种实施方式中，在存储介质中存储的合约 SCORE 的指令序列为 WASM 指令序列，从而可从存储介质中直接读取合约 SCORE 的 WASM 指令序列。

[67] 在一种实施方式中，在存储介质中存储的合约 SCORE 的指令序列采用 leb 128 可变量编码方式进行编码，以减小文件体积，更好利用存储空间。Leb 128 即"Little-Endian Base 128"，是对任意有符号或者无符号整型数的可变长度的编码，因此，用 Leb128 编码的整数，会根据数字的大小改变所占字节数。根据 leb 128 的编码方式，每个字节的 8 位（8 个 bits）中，较低的 7 位为有效数据部分，最高位用于指示当前字节是否为最后一个字节，1 表示最后一个字节，0 表示不是最后一个字节，也就是需要接续后续字节。如此，可以采用不同字节数编码不同大小的整数。

[68] 相应地，对于 leb 编码的指令序列，在解码该数据时，先将各个字节还原为 8 位，通过最高位判断是否要接续后面的字节，共同形成目标数据。因此，对该指令序列进行解析和解码，会得到不定长的指令。为了对这些指令进行存储，可以将可变量指令转换为内存对齐的定长指令。从而可获取合约 SCORE 的 WASM 指令序列。

[69] 步骤 S210, 将所述第一合约的 WASM 指令序列与第一合约的标识关联地缓存到内存中。

[70] 在一种实施方式中,节点 A 可在其内存中分配用于存储所述 WASM 指令序列的起始地址,从而在节点 A 中设置的虚拟机 1 将该指令序列从该起始地址开始存入内存中,并在内存中的预定表格中记录合约 SCORE 与该起始地址相对应。该表例如记录键值对,键例如为合约 SCORE 的合约地址哈希值,值为所述起始地址。可以理解,这里虽然描述了在预定表格中记录合约 SCORE 的合约地址哈希值用于标识合约 SCORE,本说明书实施例不限于此,在预定表格中只要记录可唯一标识合约 SCORE 的标识即可,例如,在预定表格中也可以记录合约 SCORE 的合约地址用于标识合约 SCORE。

[71] 图 3 示意示出了一种记录合约标识与地址键值对的表格示例。如图 3 中所示,该表格中上部一行为各个合约的合约地址哈希值,下部一行记录相应合约在内存中的存储起始地址。例如合约 SCORE 的合约地址哈希值为“21c...”,其对应的内存存储地址为 300。

[72] 从而通过进行该缓存,节点 A 后续在执行其它调用该合约 SCORE 的交易时,可从其内存中的所述预定表格中找到合约 SCORE 的合约地址哈希值“21c...”,从而获取合约 SCORE 在内存中的存储地址 300,通过在内存中访问该地址 300,可直接读取到合约 SCORE 的 WASM 指令序列,从而加速了合约执行时间。

[73] 在一种实施方式中,节点 A 可在其内存中分配用于存储所述 WASM 指令序列的起始地址,从而虚拟机 1 将该指令序列与合约 SCORE 的合约地址一起从所述起始地址开始存入内存中。从而在后续调用该合约 SCORE 时,可通过搜索合约 SCORE 的合约地址而找到合约 SCORE。

[74] 在节点 A 在进行上述缓存之后,可执行上述步骤 S206,即,节点 A 中的虚拟机 1 可基于交易 1 中的数据字段解释执行合约 SCORE 中的函数 calculation 包括的指令序列,从而完成对交易 1 的执行。

[75] 图 4 示出在节点 A 中缓存和执行合约 SCORE 的示意图。在节点 A 中,包括内存 41、CPU42 和存储介质 43。当执行内存 41 中的交易队列中的调用合约 SCORE 中函数的交易 1 时,通过 CPU42 实例化虚拟机 1 来执行交易 1。虚拟机 1 首先确定内存 41 中是否缓存有合约 SCORE 的 WASM 指令序列,当确定内存 41 中未缓存所述指令序列的情况中,虚拟机 1 从介质 43 中获取合约 SCORE 的经 leb 128 编码的指令序列,对该编码指令序列进行 leb 128 解码(未示出),并将解码后的指令序列缓存到内存 41 中,然后基于交易 1 的数据字段中的函数参数解释执行合约 SCORE 中的相应函数。在执行了交易 1 之后,当执行交易队列中的调用合约 SCORE 中某函数的交易 2 时,通过 CPU42 实例化虚拟机 2 来执行交易 2。类似地,虚拟机 2 首先确定内存 41 中是否缓存有合约 SCORE 的 WASM 指令序列,当确定内存中缓存有所述指令序列的情况中,读取该 WASM 指令序列,并基于交易 2 的数据字段中的函数参数执行相应函数的 WASM 指令序列。从而在执行交易 1 时进行对合约 SCORE 的 WASM 指令序列的缓存之后,后续在执行其它调用合约 SCORE 中函数的交易时不再需要从介质 43 中读取合约 SCORE,也不需要重复进行 leb 128 解码,从而缩短了后续交易的执行时间,提升了系统性能。

[76] 图 5 示出根据本说明书另一实施例的智能合约缓存方法流程图,所述方法由区块链中任一节点的虚拟机执行用于调用合约 SCORE 中函数的交易时执行,所述方法包括步骤 S502~S510。

[77] 步骤 S502, 确定内存中是否缓存有合约 SCORE 的 WASM 指令序列。

[78] 步骤 S504, 在确定内存中没有缓存合约 SCORE 的 WASM 指令序列的情况下, 从所述节点本地存储介质中读取合约 SCORE 的指令序列, 以获取合约 SCORE 的 WASM 指令序列。

[79] 步骤 S502 和 S504 可参考上文对步骤 S202 和 S204 的描述, 在此不再详述。

[80] 步骤 S506, 确定所述 WASM 指令序列中是否包括预定连续指令, 所述预定连续指令被预设有相应的合并指令, 并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果。

[81] 所述预定连续指令是预先确定的可以进行合并的若干个连续指令, 对于这样的预定连续指令, 已经预先设置了对应的合并指令, 以及与该合并指令相对应的机器码, 该机器码在执行之后与其对应的连续指令具有相同的执行结果。

[82] 例如, 对于上述函数 calculation 的指令序列, 可以相对于连续指令 “get_local y/i32.const 2/i32.mul” 预设一个合并指令 “i32.mul y, const 2”, 同样地, 该合并指令实际中为二进制码, 可选定一个在 WASM 指令集中未使用的二进制码来表示该合并指令。同时, 在虚拟机的解释器中, 预先设定了与该合并指令相对应的机器码(machine code), 该机器码对应的操作为, 获取参数 y 的值, 将参数 y 的值与 2 相乘, 并将乘积推入栈中, 其中, 可从调用合约 SCORE 的交易中的数据中获取参数 y 的值。可见, 该合并指令对应的机器码的操作结果与所述三个连续指令对应的机器码执行结果是相同的, 都是将参数 y 的值与 2 的乘积推入栈中。

[83] 另外, 可相对于连续指令 “get_local x/i32.add” 预设一个合并指令 “i32.add x”, 同时, 在虚拟机的解释器中, 预先设定了与该合并指令相对应的机器码, 该机器码对应的操作为, 从栈中弹出最新的值, 获取参数 x 的值, 将弹出的值与参数 x 的值相加, 并将相加的结果推入栈中, 其中, 可从调用合约 SCORE 的交易的数据中获取 x 的值。可见, 该合并指令对应的机器码的操作结果与所述两个连续指令对应的机器码执行结果是相同的, 都是将上述乘积与参数 x 的值的相加结果推入栈中。

[84] 另外, 可相对于连续指令 “i32.const 3/i32.mul” 预设一个合并指令 “i32.mul const 3”, 同时, 在虚拟机的解释器中, 预先设定了与该合并指令相对应的机器码, 该机器码对应的操作为, 从栈中弹出最新的值, 并将该弹出的值与 3 相乘, 并将乘积推入栈中。可见, 该合并指令对应的机器码的操作结果与所述两个连续指令对应的机器码执行结果是相同的, 都是将上述相加的结果与 3 的乘积推入栈中, 该最后推入栈中的值即为该函数的返回值。

[85] 从上面的示例中可以看出, 在预定连续指令中, 各个指令具有一定的相关性。例如, 对于连续指令 “get_local y/i32.const 2/i32.mul”, 最后一个指令 i32.mul 会用到前面两个指令的获取值(y)或立即数(即常数 2)。

[86] 步骤 S508, 在确定所述指令序列中包括至少一组预定连续指令的情况下, 将所述指令序列中的各组预定连续指令分别合并为对应的合并指令, 从而获取所述合约 SCORE 的经指令融合的 WASM 指令序列。

[87] 例如, 在上述步骤 S506 中, 已经确定所述指令序列中包括三组预定连续指令, 从而, 可将这三组指令分别合并为对应的合并指令, 即, 经合并之后的 WASM 指令序列

为:

```
{ i32. mul y const 2
  i32.add x
  i32.mul const 3
}
```

[88] 步骤 S510, 将所述合约 SCORE 的经指令融合的 WASM 指令序列与合约 SCORE 的标识关联地缓存到内存中。

[89] 该步骤可参考上文对步骤 S206 的描述, 在此不再赘述。

[90] 在执行交易 1 时进行对合约 SCORE 的经指令融合的指令序列的缓存之后, 后续在执行调用合约 SCORE 中函数的交易时不再需要从存储介质中读取合约 SCORE, 也不需要重复进行 leb 128 解码, 也不需要重复进行指令融合, 从而缩短了后续交易的执行时间, 提升了系统性能, 而通过执行指令融合的 WASM 指令序列, 可进一步减少交易的执行时间。

[91] 图 6 示出根据本说明书另一实施例的智能合约缓存方法流程图, 所述方法由区块链中任一节点的虚拟机执行用于调用合约 SCORE2 中函数的交易时执行, 所述方法包括步骤 S602~S612。

[92] 步骤 S602, 确定内存中是否缓存有所述合约 SCORE2 的 WASM 指令序列。

[93] 步骤 S604, 在确定内存中没有缓存所述合约 SCORE2 的 WASM 指令序列的情况下, 从所述节点本地存储介质中读取合约 SCORE2 的指令序列, 以获取合约 SCORE2 的 WASM 指令序列。

[94] 步骤 S602 和 S604 可参考上文对步骤 S202 和 S204 的描述, 在此不再详述。

[95] 假设获取的合约 SCORE2 的 WASM 指令序列为:

```
(Contract SCORE2
  (type (;0;) (func (result i32)))
  (func (;0;) (type 0) (result i32)
    get_global 0)
  (global (;0;) i32 (i32.const -2))
  (global (;1;) f32 (f32.const -0x1.8p+1 (=-3;)))
  (global (;2;) f64 (f64.const -0x1p+2 (=-4;)))
  (global (;3;) i64 (i64.const -5))
  (global (;4;) (mut i32) (i32.const -12))
  (global (;5;) (mut f32) (f32.const -0x1.ap+3 (=-13;)))
  (global (;6;) (mut f64) (f64.const -0x1.cp+3 (=-14;)))
  (global (;7;) (mut i64) (i64.const -15))
  (export "get-a" (func 0))).
```

[96] 步骤 S606, 基于所述合约 SCORE2 的 WASM 指令序列确定全局变量标识范围。

[97] 参考上述合约 SCORE2 的 WASM 指令序列, 虚拟机在获取与该指令序列之后, 通过逐条读取每条指令, 可获知该指令序列中共包括多少全局变量, 即全局变量 0~7, 共 8 个全局变量, 从而为这些全局变量分配的变量下标分别为 0~7, 即全局变量标识范围为 0~7。

[98] 步骤 S608, 对于所述合约 SCORE2 的 WASM 指令序列中的每个指令, 解析该指令是否为全局变量访问指令。

[99] 例如, 当解析到合约 SCORE2 的第 4 行的 “get_global 0”, 可解析出该指令为全局变量读指令。全局变量访问指令不限于为全局变量读指令, 还包括全局变量写指令, 例如, “set_global 1”。

[100] 步骤 S610, 在解析出所述指令为全局变量访问指令的情况中, 检查该指令包括的全局变量标识是否在所述全局变量标识范围内。

[101] 例如, 对于合约 SCORE2 中的全局变量读指令 get_global 0, 即检查其中的变量下标 “0” 是否在上述全局变量标识范围 0~7 之内, 显然, 0 在范围 0~7 之内。

[102] 假设合约 SCORE2 中除了上述全局变量读指令, 还包括全局变量写指令 “set_global 10”, 很显然, 该指令中的变量下标 “10” 超出了上述虚拟机获取的全局变量标识范围 0~7, 从而, 该全局变量写指令的变量下标是越界的, 即, 合约 SCORE2 中包括错误指令。在该情况中, 将退出对该交易的执行。

[103] 步骤 S612, 在检查所述合约 SCORE2 中不包括全局变量访问指令的情况中、或者在检查所述合约 SCORE2 的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况中, 将所述合约 SCORE2 的经越界检查的 WASM 指令序列与合约 SCORE2 的标识关联地缓存到内存中。

[104] 在执行该交易时进行对合约 SCORE2 的经越界检查的指令序列的缓存之后, 后续在执行其它调用合约 SCORE2 中函数的交易时不再需要从存储介质中读取合约 SCORE2, 也不需要重复进行 leb 128 解码, 也不需要重复进行越界检查, 从而缩短了后续交易的执行时间, 提升了系统性能, 而通过在虚拟机中在静态解析阶段对 WASM 指令序列进行越界检查, 从而在解释执行该 WASM 指令序列时, 可大大减少程序执行时间, 加快程序执行速度。

[105] 图 5 和图 6 所示方法仅仅是在缓存合约之前对合约的 WASM 指令序列进行预处理的两个示例, 实际中, 在缓存之前可根据具体的场景对合约的 WASM 指令序列进行多种需要的预处理, 在此不一一列出。

[106] 图 7 示出根据本说明书另一实施例的一种智能合约执行方法流程图, 所述方法由区块链节点的虚拟机执行第一交易时执行, 所述第一交易用于调用第一合约的第一函数, 所述方法包括步骤 S702~S706。

[107] 步骤 S702, 确定内存中是否缓存有所述第一函数的 WASM 指令序列。

[108] 在一种实施方式中, 在内存中记录有缓存的合约 SCORE 的函数 calculation 与存储函数 calculation 的 WASM 指令序列的内存存储地址的关联关系, 例如通过表格的形式将包括合约 SCORE 的合约地址和函数 calculation 的函数名称的数据的哈希值与函数 calculation 的内存存储地址关联起来, 从而, 通过在该表格中查询是否存储有与函数 calculation 对应的哈希值, 可确定内存中是否缓存有函数 calculation 的 WASM 指令序列。

[109] 在一种实施方式中, 可通过在内存中搜索所述合约 SCORE 的合约地址, 从而确定内存中是否缓存有合约 SCORE 的 WASM 指令序列, 从而可确定是否缓存有函数 calculation 的 WASM 指令序列。

[110] 步骤 S704, 在确定内存中缓存了所述第一函数的 WASM 指令序列的情况中, 从内

存中读取所述第一函数的 WASM 指令序列。

[111] 在一种实施方式中，如在上文中所述，当在所述表格中查询到与函数 calculation 对应的哈希值之后，根据该表格，可读取与该哈希值关联存储的函数 calculation 在内存中的存储地址，从而可从该存储地址读取到函数 calculation 的 WASM 指令序列。

[112] 在另一种实施方式中，当在内存中搜索到合约 SCORE 的合约地址之后，例如，可预先将合约地址与合约代码关联地存储在一起，从而在搜索到合约 SCORE 的合约地址之后，即找到了合约 SCORE 的 WASM 指令序列的存储地址，通过从该存储地址开始搜索函数 calculation 的名称，从而可读取函数 calculation 的 WASM 指令序列。

[113] 步骤 S706，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[114] 该步骤可参考上文中对步骤 S206 的描述，在此不再详述。

[115] 步骤 S708，在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况中，从所述节点本地存储介质中读取所述第一合约的指令序列，以获取所述第一函数的 WASM 指令序列。

[116] 从所述节点本地存储介质中读取所述第一合约的指令序列可参考上文中对步骤 S208 的描述，在此不再详述。在获取到第一合约的指令序列之后，可获取到该合约中的各个函数的 WASM 指令序列。例如，第一合约中不仅包括上述函数 calculation 的 WASM 指令序列，可能还包括函数 calculation2 的 WASM 指令序列。例如，函数 calculation 是用于计算平台中用户的分数，函数 calculation2 用于计算平台中商户的分数，它们可能具有不同的计算公式，从而具有不同的 WASM 指令序列。在执行交易的过程中，部分交易调用函数 calculation，用于计算用户的分数，部分交易调用函数 calculation2，用于计算商户的分数。因此，在获取合约 SCORE 的 WASM 指令序列之后，可获取其中包括的各个函数的 WASM 指令序列，以分别将这些函数缓存到内存中，以供交易进行调用。

[117] 步骤 S710，将所述第一函数的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中。

[118] 在一种实施方式中，将所述第一函数的 WASM 指令序列在内存中的存储地址与第一数据的哈希值关联地存储到内存中，其中，所述第一数据基于第一合约的合约地址和第一函数的名称获取。例如，所述第一数据通过将第一合约的合约地址和第一函数的名称拼接到一起获取，或者基于预定公式对第一合约的合约地址和第一函数的名称进行计算而获取，等等。具体地，可通过与图 3 所示表格类似的表格来存储上述关联关系，该表格例如记录键值对，键例如为第一数据的哈希值，值为所述起始地址。

[119] 在一种实施方式中，节点可将合约 SCORE 的 WASM 指令序列与合约 SCORE 的合约地址一起从所述起始地址开始存入内存中。从而在后续调用函数 calculation 时，可通过搜索合约 SCORE 的合约地址和函数 calculation 的函数名称而找到函数 calculation 的 WASM 指令序列。

[120] 在合约 SCORE 包括多个函数的情况中，例如还包括函数 calculation2，对于函数 calculation2，也进行与函数 calculation 相同的处理，从而使得后续调用函数 calculation2 的交易可直接从内存中读取到函数 calculation2 的 WASM 指令序列。

[121] 图 8 示出根据本说明书一个实施例的一种智能合约执行装置 800, 所述装置在区块链节点的虚拟机执行第一交易时部署, 所述第一交易用于调用第一合约的第一函数, 所述装置包括以下单元。

[122] 确定单元 81, 配置为, 确定内存中是否缓存有所述第一合约的 WASM 指令序列。

[123] 第一读取单元 82, 配置为, 在确定内存中缓存了所述第一合约的 WASM 指令序列的情况下, 从内存中读取所述第一合约的 WASM 指令序列, 以获取所述第一函数的 WASM 指令序列。

[124] 第一执行单元 83, 配置为, 基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[125] 在一种实施方式中, 所述装置还包括, 第二读取单元 84, 配置为, 在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况下, 从所述节点本地存储介质中读取第一合约的指令序列, 以获取所述第一合约的 WASM 指令序列。

[126] 缓存单元 85, 配置为, 将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中。

[127] 在一种实施方式中, 所述第二读取单元 84 还配置为, 从节点本地存储介质中读取第一合约的经可变长编码的指令序列, 通过对第一合约的经可变长编码的指令序列进行解码, 获取所述第一合约的 WASM 指令序列。

[128] 在一种实施方式中, 所述装置 800 还包括, 预处理单元 86, 配置为, 在获取第一合约的 WASM 指令序列之后, 对第一合约的 WASM 指令序列进行预处理, 以获取第一合约的经预处理的 WASM 指令序列, 其中, 所述缓存单元 85 还配置为, 将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中。

[129] 在一种实施方式中, 所述预处理单元 86 包括, 第一确定子单元 861, 配置为, 基于所述第一合约的 WASM 指令序列确定全局变量标识范围; 解析子单元 862, 配置为, 对于所述第一合约的 WASM 指令序列中的每个指令, 解析该指令是否为全局变量访问指令; 检查子单元 863, 配置为, 在解析出所述指令为全局变量访问指令的情况下, 检查该指令包括的全局变量标识是否在所述全局变量标识范围内; 获取子单元 864, 配置为, 在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况下, 获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

[130] 在一种实施方式中, 所述预处理单元包括, 第二确定子单元 865, 配置为, 确定所述 WASM 指令序列中是否包括预定连续指令, 所述预定连续指令被预设有相应的合并指令, 并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果; 合并子单元 866, 配置为, 在确定所述指令序列中包括至少一组预定连续指令的情况下, 将所述指令序列中的各组预定连续指令分别合并为对应的合并指令, 从而获取所述第一合约的经指令融合的 WASM 指令序列。

[131] 在一种实施方式中, 所述装置 800 还包括, 第二执行单元 87, 配置为, 在将所述第一合约的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中之后, 基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[132] 在一种实施方式中, 所述缓存单元 85 还配置为, 将所述第一合约的 WASM 指令

序列在内存中的存储地址与第一合约的合约地址的哈希值关联地存储到内存中。

[133] 图 9 示出根据本说明书另一实施例的一种智能合约执行装置 900, 所述装置在区块链节点的虚拟机执行第一交易时部署, 所述第一交易用于调用第一合约的第一函数, 所述装置包括: 确定单元 91, 配置为, 确定内存中是否缓存有所述第一函数的 WASM 指令序列; 第一读取单元 92, 配置为, 在确定内存中缓存了所述第一函数的 WASM 指令序列的情况下, 从内存中读取所述第一函数的 WASM 指令序列; 执行单元 93, 配置为, 基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

[134] 在一种实施方式中, 所述装置还包括, 第二读取单元 94, 配置为, 在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况下, 从所述节点本地存储介质中读取所述第一合约的指令序列, 以获取所述第一函数的 WASM 指令序列; 缓存单元 95, 配置为, 将所述第一函数的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中。

[135] 在一种实施方式中, 所述缓存单元 95 还配置为, 将所述第一函数的 WASM 指令序列在内存中的存储地址与第一数据的哈希值关联地存储到内存中, 其中, 所述第一数据基于第一合约的合约地址和第一函数的名称获取。

[136] 本说明书另一方面提供一种计算机可读存储介质, 其上存储有计算机程序, 当所述计算机程序在计算机中执行时, 令计算机执行上述任一项方法。

[137] 本说明书另一方面提供一种计算设备, 包括存储器和处理器, 所述存储器中存储有可执行代码, 所述处理器执行所述可执行代码时, 实现上述任一项方法。

[138] 在根据本说明书实施例的智能合约缓存和执行方案, 通过虚拟机在执行交易时将获取的合约的指令序列缓存到内存中, 从而后续交易可直接从内存中读取该合约的指令序列, 而不需要从存储介质中读取该指令序列, 也不需要再对编码的指令序列重复解码, 另外, 也不需要再对在执行调用合约的交易时对合约的指令序列重复进行预处理, 缩短了后续交易的执行时间, 提升了系统性能。

[139] 需要理解, 本文中的“第一”, “第二”等描述, 仅仅为了描述的简单而对相似概念进行区分, 并不具有其他限定作用。

[140] 本说明书中的各个实施例均采用递进的方式描述, 各个实施例之间相同相似的部分互相参见即可, 每个实施例重点说明的都是与其他实施例的不同之处。尤其, 对于系统实施例而言, 由于其基本相似于方法实施例, 所以描述的比较简单, 相关之处参见方法实施例的部分说明即可。

[141] 上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下, 在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外, 在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中, 多任务处理和并行处理也是可以的或者可能是有利的。

[142] 本领域普通技术人员应该还可以进一步意识到, 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功能一般性地描述了各示例的组成及步骤。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。本领域普通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。其中，软件模块可以置于随机存储器（RAM）、内存、只读存储器（ROM）、电可编程 ROM、电可擦除可编程 ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[143] 以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

权利要求书

1、一种智能合约执行方法,所述方法由区块链节点的虚拟机执行第一交易时执行,所述第一交易用于调用第一合约的第一函数,所述方法包括:

确定内存中是否缓存有所述第一合约的 WASM 指令序列;

在确定内存中缓存了所述第一合约的 WASM 指令序列的情况中,从内存中读取所述第一合约的 WASM 指令序列,以获取所述第一函数的 WASM 指令序列;基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况中,从所述节点本地存储介质中读取第一合约的指令序列,以获取所述第一合约的 WASM 指令序列;在获取第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取所述第一合约的经预处理的 WASM 指令序列;将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中;

其中,对所述第一合约的 WASM 指令序列进行预处理包括:

基于所述第一合约的 WASM 指令序列确定全局变量标识范围;

对于所述第一合约的 WASM 指令序列中的每个指令,解析该指令是否为全局变量访问指令;

在解析出所述指令为全局变量访问指令的情况中,检查该指令包括的全局变量标识是否在所述全局变量标识范围内;

在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况中,获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

2、一种智能合约执行方法,所述方法由区块链节点的虚拟机执行第一交易时执行,所述第一交易用于调用第一合约的第一函数,所述方法包括:

确定内存中是否缓存有所述第一合约的 WASM 指令序列;

在确定内存中缓存了所述第一合约的 WASM 指令序列的情况中,从内存中读取所述第一合约的 WASM 指令序列,以获取所述第一函数的 WASM 指令序列;基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况中,从所述节点本地存储介质中读取第一合约的指令序列,以获取所述第一合约的 WASM 指令序列;在获取第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取所述第一合约的经预处理的 WASM 指令序列;将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中;

其中,对所述第一合约的 WASM 指令序列进行预处理包括:

确定所述 WASM 指令序列中是否包括预定连续指令,所述预定连续指令被预设有相应的合并指令,并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果;

在确定所述指令序列中包括至少一组预定连续指令的情况中,将所述指令序列中的各组预定连续指令分别合并为对应的合并指令,从而获取所述第一合约的经指令融合的 WASM 指令序列。

3、根据权利要求 1 或 2 所述的方法,其中,从所述节点本地存储介质中读取第一合约的指令序列,以获取所述第一合约的 WASM 指令序列包括,从节点本地存储介质中读取第一合约的经可变长编码的指令序列,通过对第一合约的经可变长编码的指令序列进行解码,获取所述第一合约的 WASM 指令序列。

4、根据权利要求 1 或 2 所述的方法,还包括,在将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中之后,基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

5、根据权利要求 1 或 2 所述的方法,其中,将所述第一合约的经预处理的 WASM

指令序列与所述第一合约的标识关联地缓存到内存中包括,将所述第一合约的经预处理的 WASM 指令序列在内存中的存储地址与第一合约的合约地址的哈希值关联地存储到内存中。

6、一种智能合约执行方法,所述方法由区块链节点的虚拟机执行第一交易时执行,所述第一交易用于调用第一合约的第一函数,所述方法包括:

确定内存中是否缓存有所述第一函数的 WASM 指令序列;

在确定内存中缓存了所述第一函数的 WASM 指令序列的情况中,从内存中读取所述第一函数的 WASM 指令序列;基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况中,从所述节点本地存储介质中读取所述第一合约的指令序列,以获取所述第一函数的 WASM 指令序列;在获取第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取所述第一函数的经预处理的 WASM 指令序列;将所述第一函数的经预处理的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中;

其中,对所述第一合约的 WASM 指令序列进行预处理包括:

基于所述第一合约的 WASM 指令序列确定全局变量标识范围;

对于所述第一合约的 WASM 指令序列中的每个指令,解析该指令是否为全局变量访问指令;

在解析出所述指令为全局变量访问指令的情况中,检查该指令包括的全局变量标识是否在所述全局变量标识范围内;

在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况中,获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

7、一种智能合约执行方法,所述方法由区块链节点的虚拟机执行第一交易时执行,所述第一交易用于调用第一合约的第一函数,所述方法包括:

确定内存中是否缓存有所述第一函数的 WASM 指令序列;

在确定内存中缓存了所述第一函数的 WASM 指令序列的情况中,从内存中读取所述第一函数的 WASM 指令序列;基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况中,从所述节点本地存储介质中读取所述第一合约的指令序列,以获取所述第一函数的 WASM 指令序列;在获取第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取所述第一函数的经预处理的 WASM 指令序列;将所述第一函数的经预处理的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中;

其中,对所述第一合约的 WASM 指令序列进行预处理包括:

确定所述 WASM 指令序列中是否包括预定连续指令,所述预定连续指令被预设有相应的合并指令,并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果;

在确定所述指令序列中包括至少一组预定连续指令的情况中,将所述指令序列中的各组预定连续指令分别合并为对应的合并指令,从而获取所述第一合约的经指令融合的 WASM 指令序列。

8、根据权利要求 6 或 7 所述的方法,其中,将所述第一函数的经预处理的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中包括,将所述第一函数的经预处理的 WASM 指令序列在内存中的存储地址与第一数据的哈希值关联地存储到内存中,其中,所述第一数据基于第一合约的合约地址和第一函数的名称获取。

9、一种智能合约执行装置,所述装置在区块链节点的虚拟机执行第一交易时部署,所述第一交易用于调用第一合约的第一函数,所述装置包括:

确定单元,配置为,确定内存中是否缓存有所述第一合约的 WASM 指令序列;

第一读取单元,配置为,在确定内存中缓存了所述第一合约的 WASM 指令序列的情况下,从内存中读取所述第一合约的 WASM 指令序列,以获取所述第一函数的 WASM 指令序列;

第一执行单元,配置为,基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

第二读取单元,配置为,在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况下,从所述节点本地存储介质中读取第一合约的指令序列,以获取所述第一合约的 WASM 指令序列;

预处理单元,配置为,在获取所述第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取第一合约的经预处理的 WASM 指令序列;

缓存单元,配置为,将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中;

其中,所述预处理单元包括:

第一确定子单元,配置为,基于所述第一合约的 WASM 指令序列确定全局变量标识范围;

解析子单元,配置为,对于所述第一合约的 WASM 指令序列中的每个指令,解析该指令是否为全局变量访问指令;

检查子单元,配置为,在解析出所述指令为全局变量访问指令的情况下,检查该指令包括的全局变量标识是否在所述全局变量标识范围内;

获取子单元,配置为,在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况下,获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

10、一种智能合约执行装置,所述装置在区块链节点的虚拟机执行第一交易时部署,所述第一交易用于调用第一合约的第一函数,所述装置包括:

确定单元,配置为,确定内存中是否缓存有所述第一合约的 WASM 指令序列;

第一读取单元,配置为,在确定内存中缓存了所述第一合约的 WASM 指令序列的情况下,从内存中读取所述第一合约的 WASM 指令序列,以获取所述第一函数的 WASM 指令序列;

第一执行单元,配置为,基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列;

第二读取单元,配置为,在确定内存中没有缓存所述第一合约的 WASM 指令序列的情况下,从所述节点本地存储介质中读取第一合约的指令序列,以获取所述第一合约的 WASM 指令序列;

预处理单元,配置为,在获取所述第一合约的 WASM 指令序列之后,对所述第一合约的 WASM 指令序列进行预处理,以获取第一合约的经预处理的 WASM 指令序列;

缓存单元,配置为,将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中;

其中,所述预处理单元包括:

第二确定子单元,配置为,确定所述 WASM 指令序列中是否包括预定连续指令,所述预定连续指令被预设有相应的合并指令,并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果;

合并子单元,配置为,在确定所述指令序列中包括至少一组预定连续指令的情况下,将所述指令序列中的各组预定连续指令分别合并为对应的合并指令,从而获取所述第一合约的经指令融合的 WASM 指令序列。

11、根据权利要求 9 或 10 所述的装置,其中,所述第二读取单元还配置为,从节

点本地存储介质中读取第一合约的经可变长编码的指令序列，通过对第一合约的经可变长编码的指令序列进行解码，获取所述第一合约的 WASM 指令序列。

12、根据权利要求 9 或 10 所述的装置，还包括，第二执行单元，配置为，在将所述第一合约的经预处理的 WASM 指令序列与所述第一合约的标识关联地缓存到内存中之后，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列。

13、根据权利要求 9 或 10 所述的装置，其中，所述缓存单元还配置为，将所述第一合约的经预处理的 WASM 指令序列在内存中的存储地址与第一合约的合约地址的哈希值关联地存储到内存中。

14、一种智能合约执行装置，所述装置在区块链节点的虚拟机执行第一交易时部署，所述第一交易用于调用第一合约的第一函数，所述装置包括：

确定单元，配置为，确定内存中是否缓存有所述第一函数的 WASM 指令序列；

第一读取单元，配置为，在确定内存中缓存了所述第一函数的 WASM 指令序列的情况下，从内存中读取所述第一函数的 WASM 指令序列；

执行单元，配置为，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列；

第二读取单元，配置为，在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况下，从所述节点本地存储介质中读取所述第一合约的指令序列，以获取所述第一函数的 WASM 指令序列；

预处理单元，配置为，在获取第一合约的 WASM 指令序列之后，对所述第一合约的 WASM 指令序列进行预处理，以获取所述第一函数的经预处理的 WASM 指令序列；

缓存单元，配置为，将所述第一函数的经预处理的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中；

其中，所述预处理单元包括：

第一确定子单元，配置为，基于所述第一合约的 WASM 指令序列确定全局变量标识范围；

解析子单元，配置为，对于所述第一合约的 WASM 指令序列中的每个指令，解析该指令是否为全局变量访问指令；

检查子单元，配置为，在解析出所述指令为全局变量访问指令的情况下，检查该指令包括的全局变量标识是否在所述全局变量标识范围内；

获取子单元，配置为，在检查所述第一合约的各个全局变量访问指令中包括的全局变量标识都在所述全局变量标识范围内的情况下，获取所述第一合约的经全局变量越界检查的 WASM 指令序列。

15、一种智能合约执行装置，所述装置在区块链节点的虚拟机执行第一交易时部署，所述第一交易用于调用第一合约的第一函数，所述装置包括：

确定单元，配置为，确定内存中是否缓存有所述第一函数的 WASM 指令序列；

第一读取单元，配置为，在确定内存中缓存了所述第一函数的 WASM 指令序列的情况下，从内存中读取所述第一函数的 WASM 指令序列；

执行单元，配置为，基于所述第一交易中的数据字段解释执行所述第一函数的 WASM 指令序列；

第二读取单元，配置为，在确定内存中没有缓存所述第一函数的 WASM 指令序列的情况下，从所述节点本地存储介质中读取所述第一合约的指令序列，以获取所述第一函数的 WASM 指令序列；

预处理单元，配置为，在获取第一合约的 WASM 指令序列之后，对所述第一合约的 WASM 指令序列进行预处理，以获取所述第一函数的经预处理的 WASM 指令序列；

缓存单元，配置为，将所述第一函数的经预处理的 WASM 指令序列与所述第一函数的标识关联地缓存到内存中；

其中，所述预处理单元包括：

第二确定子单元，配置为，确定所述 WASM 指令序列中是否包括预定连续指令，所述预定连续指令被预设有相应的合并指令，并且所述预定连续指令与其对应的合并指令具有相同的解释执行结果；

合并子单元，配置为，在确定所述指令序列中包括至少一组预定连续指令的情况下，将所述指令序列中的各组预定连续指令分别合并为对应的合并指令，从而获取所述第一合约的经指令融合的 WASM 指令序列。

16、根据权利要求 14 或 15 所述的装置，其中，所述缓存单元还配置为，将所述第一函数的经预处理的 WASM 指令序列在内存中的存储地址与第一数据的哈希值关联地存储到内存中，其中，所述第一数据基于第一合约的合约地址和第一函数的名称获取。

17、一种计算机可读存储介质，其上存储有计算机程序，当所述计算机程序在计算机中执行时，令计算机执行权利要求 1-8 中任一项的所述的方法。

18、一种计算设备，包括存储器和处理器，所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现权利要求 1-8 中任一项所述的方法。

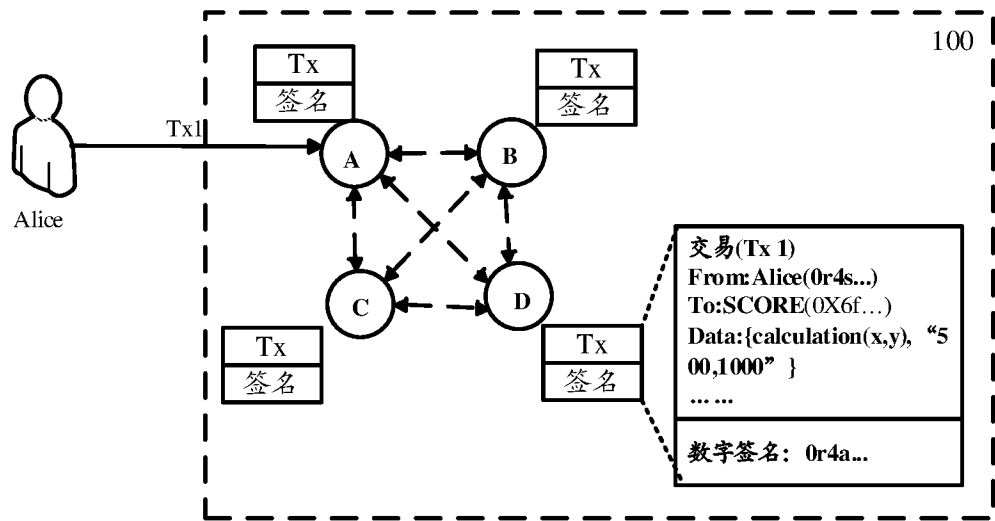


图 1

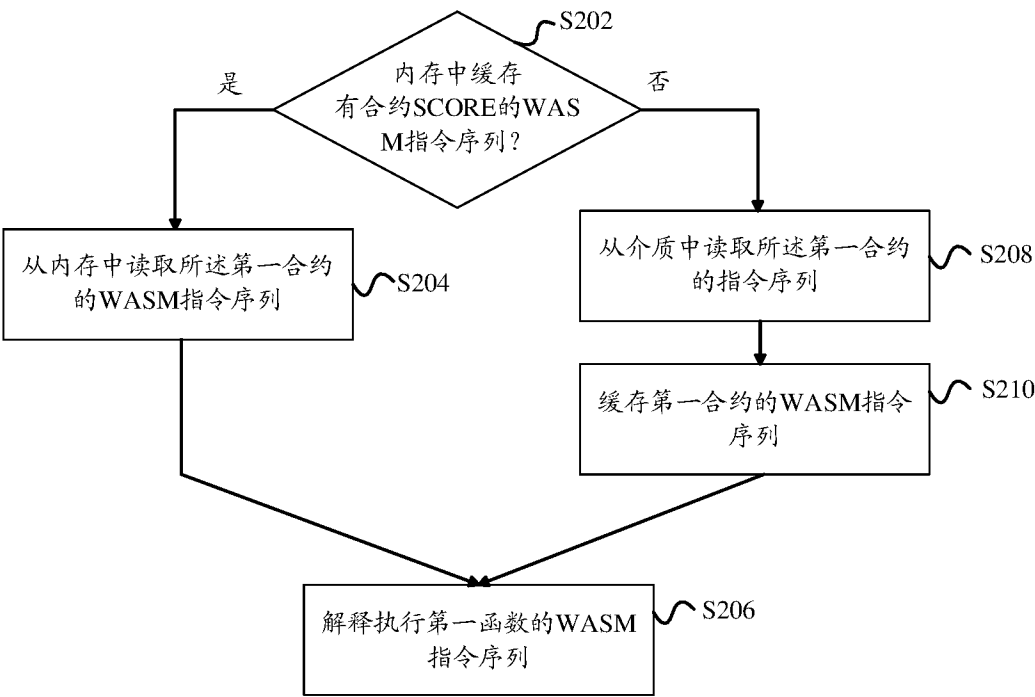


图 2

哈希值	15a...	...	21c...	...		
地址	100	...	300	...		

图 3

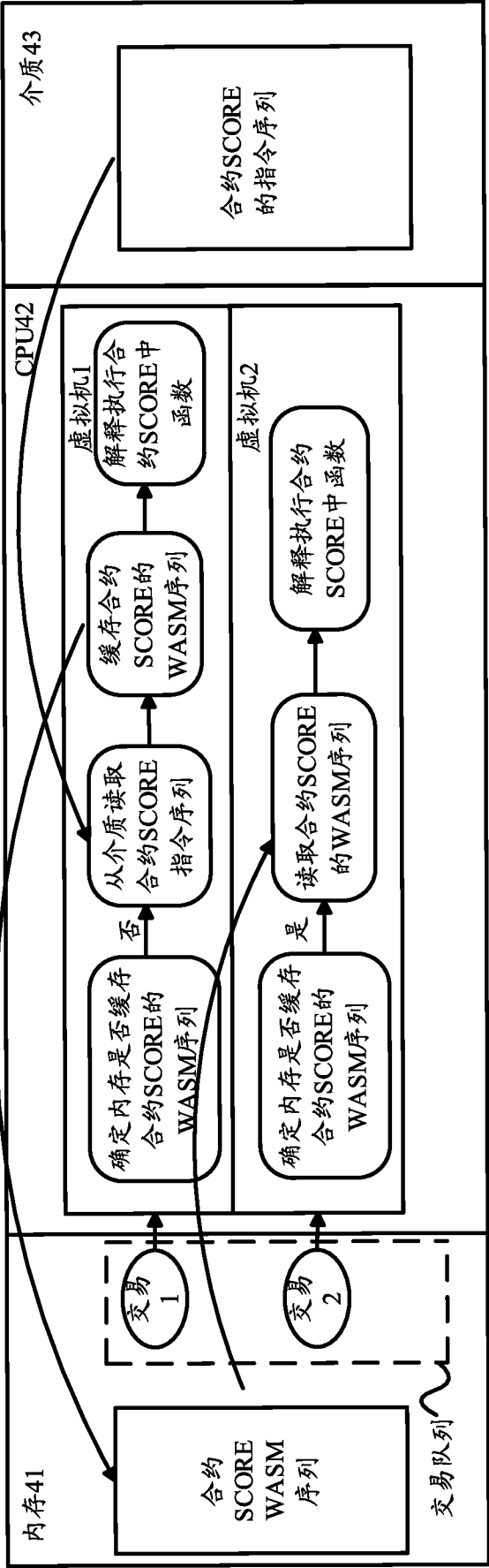


图 4

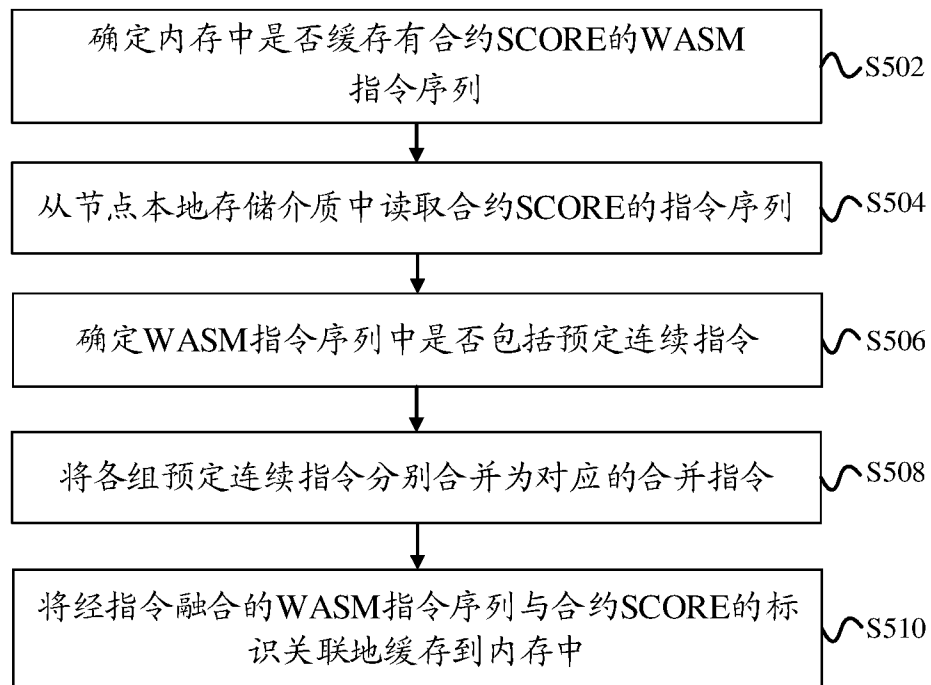


图 5

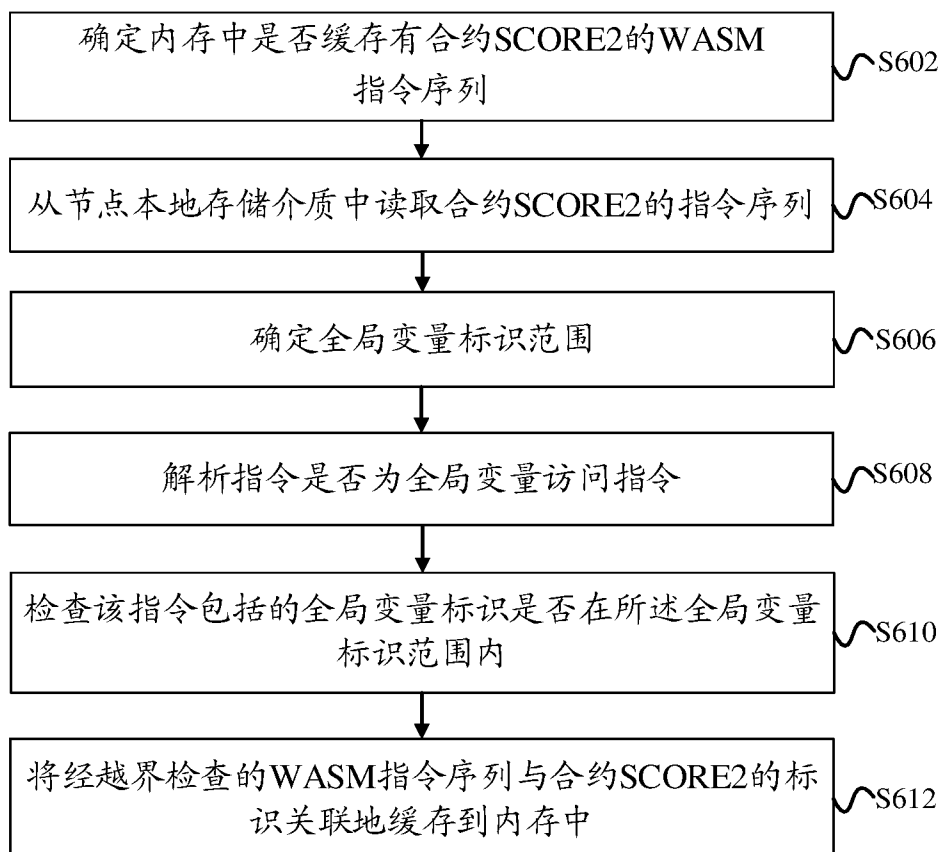


图 6

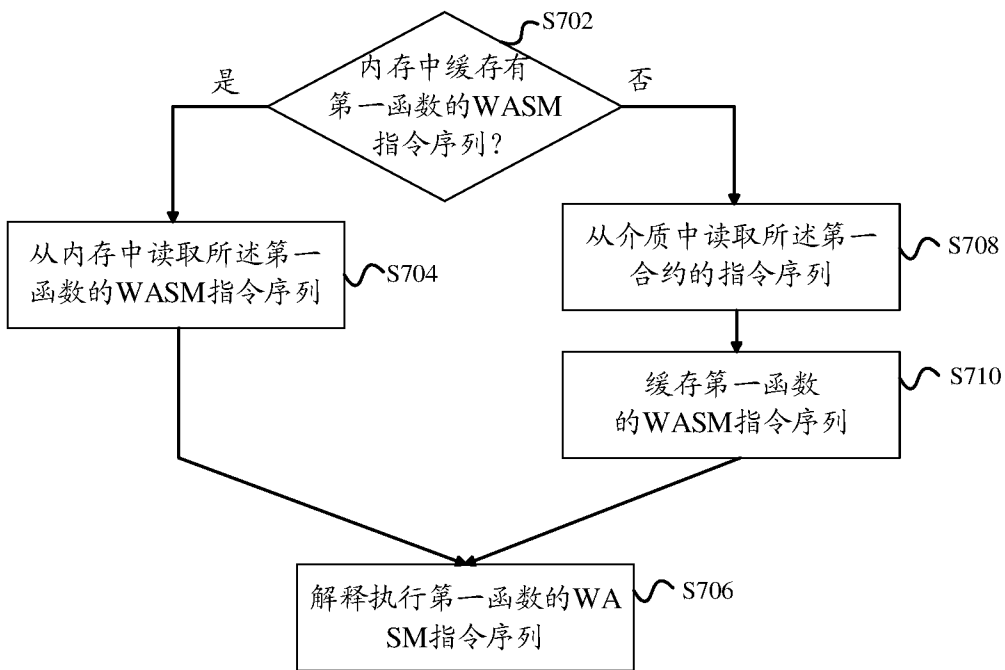


图 7

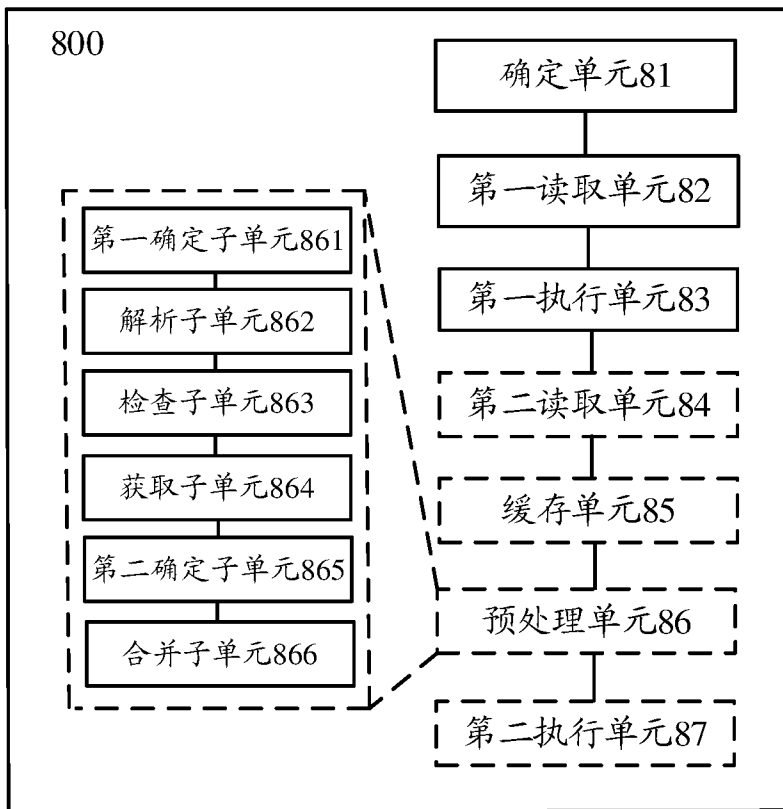


图 8

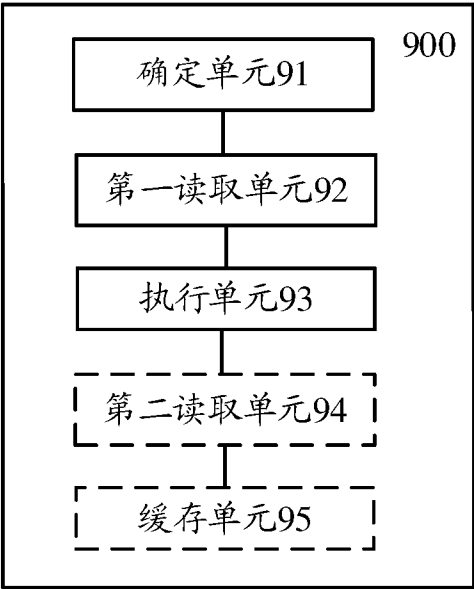


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/073557

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/455(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; CNKI; SIPOABS; DWPI; USTXT; WOTXT; EPTXT: 提高, 提升, 性能, 效率, 本地, 内存, 缓存, 加载, 解码, 指令, 合并, WASM, webassembly, improve, performance, efficiency, local, cache, memory, decode, instruction, merge

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 111045792 A (ALIPAY (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) 21 April 2020 (2020-04-21) description, paragraphs [0006]-[0121], and figures 1-9	1-18
Y	CN 109949156 A (JUZIX TECHNOLOGY CO., LTD.) 28 June 2019 (2019-06-28) description, paragraphs [0015]-[0086], and figures 1-5	1-18
Y	CN 110675256 A (ALIBABA GROUP HOLDING LIMITED) 10 January 2020 (2020-01-10) description, paragraphs [0047]-[0128], and figures 1-7	1-18
Y	CN 103365626 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 23 October 2013 (2013-10-23) description, paragraphs [0018]-[0085], and figures 1-10	2, 7, 10, 15
A	CN 110704063 A (ZHIFUBAO (HANGZHOU) INFORMATION TECHNOLOGY CO., LTD.) 17 January 2020 (2020-01-17) entire document	1-18



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

29 March 2021

Date of mailing of the international search report

15 April 2021

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/073557

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	111045792	A	21 April 2020	None			
CN	109949156	A	28 June 2019	HK	40009454	A0	26 June 2020
CN	110675256	A	10 January 2020	CN	110675256	B	21 August 2020
				US	10783082	B2	22 September 2020
				US	2020364149	A1	19 November 2020
				US	2020174937	A1	04 June 2020
CN	103365626	A	23 October 2013	US	2013262822	A1	03 October 2013
				DE	102013204420	A1	02 October 2013
				US	9384000	B2	05 July 2016
				CN	103365626	B	01 June 2016
				US	9323530	B2	26 April 2016
				US	2014089636	A1	27 March 2014
CN	110704063	A	17 January 2020	HK	40021004	A0	30 October 2020

国际检索报告

国际申请号

PCT/CN2021/073557

A. 主题的分类 G06F 9/455 (2006.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类																				
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNABS; CNTXT; CNKI; SIPOABS; DWPI; USTXT; WOTXT; EPTXT: 提高, 提升, 性能, 效率, 本地, 内存, 缓存, 加载, 解码, 指令, 合并, WASM, webassembly, improve, performance, efficiency, local, cache, memory, decode, instruction, merge																				
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>PX</td> <td>CN 111045792 A (支付宝杭州信息技术有限公司) 2020年 4月 21日 (2020 - 04 - 21) 说明书第[0006]-[0121]段, 图1-9</td> <td>1-18</td> </tr> <tr> <td>Y</td> <td>CN 109949156 A (矩阵元技术深圳有限公司) 2019年 6月 28日 (2019 - 06 - 28) 说明书第[0015]-[0086]段, 图1-5</td> <td>1-18</td> </tr> <tr> <td>Y</td> <td>CN 110675256 A (阿里巴巴集团控股有限公司) 2020年 1月 10日 (2020 - 01 - 10) 说明书第[0047]-[0128]段, 图1-7</td> <td>1-18</td> </tr> <tr> <td>Y</td> <td>CN 103365626 A (国际商业机器公司) 2013年 10月 23日 (2013 - 10 - 23) 说明书第[0018]-[0085]段, 图1-10</td> <td>2、7、10、15</td> </tr> <tr> <td>A</td> <td>CN 110704063 A (支付宝杭州信息技术有限公司) 2020年 1月 17日 (2020 - 01 - 17) 全文</td> <td>1-18</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	PX	CN 111045792 A (支付宝杭州信息技术有限公司) 2020年 4月 21日 (2020 - 04 - 21) 说明书第[0006]-[0121]段, 图1-9	1-18	Y	CN 109949156 A (矩阵元技术深圳有限公司) 2019年 6月 28日 (2019 - 06 - 28) 说明书第[0015]-[0086]段, 图1-5	1-18	Y	CN 110675256 A (阿里巴巴集团控股有限公司) 2020年 1月 10日 (2020 - 01 - 10) 说明书第[0047]-[0128]段, 图1-7	1-18	Y	CN 103365626 A (国际商业机器公司) 2013年 10月 23日 (2013 - 10 - 23) 说明书第[0018]-[0085]段, 图1-10	2、7、10、15	A	CN 110704063 A (支付宝杭州信息技术有限公司) 2020年 1月 17日 (2020 - 01 - 17) 全文	1-18
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求																		
PX	CN 111045792 A (支付宝杭州信息技术有限公司) 2020年 4月 21日 (2020 - 04 - 21) 说明书第[0006]-[0121]段, 图1-9	1-18																		
Y	CN 109949156 A (矩阵元技术深圳有限公司) 2019年 6月 28日 (2019 - 06 - 28) 说明书第[0015]-[0086]段, 图1-5	1-18																		
Y	CN 110675256 A (阿里巴巴集团控股有限公司) 2020年 1月 10日 (2020 - 01 - 10) 说明书第[0047]-[0128]段, 图1-7	1-18																		
Y	CN 103365626 A (国际商业机器公司) 2013年 10月 23日 (2013 - 10 - 23) 说明书第[0018]-[0085]段, 图1-10	2、7、10、15																		
A	CN 110704063 A (支付宝杭州信息技术有限公司) 2020年 1月 17日 (2020 - 01 - 17) 全文	1-18																		
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。																				
<table border="0"> <tr> <td> * 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 </td> <td> “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件 </td> </tr> </table>			* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																			
国际检索实际完成的日期		国际检索报告邮寄日期																		
2021年 3月 29日		2021年 4月 15日																		
ISA/CN的名称和邮寄地址		受权官员																		
中国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451		任洪潮 电话号码 (86-512) 88995644																		

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/073557

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	111045792	A	2020年 4月 21日	无			
CN	109949156	A	2019年 6月 28日	HK	40009454	A0	2020年 6月 26日
CN	110675256	A	2020年 1月 10日	CN	110675256	B	2020年 8月 21日
				US	10783082	B2	2020年 9月 22日
				US	2020364149	A1	2020年 11月 19日
				US	2020174937	A1	2020年 6月 4日
CN	103365626	A	2013年 10月 23日	US	2013262822	A1	2013年 10月 3日
				DE	102013204420	A1	2013年 10月 2日
				US	9384000	B2	2016年 7月 5日
				CN	103365626	B	2016年 6月 1日
				US	9323530	B2	2016年 4月 26日
				US	2014089636	A1	2014年 3月 27日
CN	110704063	A	2020年 1月 17日	HK	40021004	A0	2020年 10月 30日