(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
2020 年 10 月 22 日 (22.10.2020)

WIPO | PCT

(10) 国际公布号
WO 2020/211554 A1

(51) 国际专利分类号:
G06F 16/2455 (2019.01)

(21) 国际申请号: PCT/CN2020/077614

(22) 国际申请日: 2020 年 3 月 3 日 (03.03.2020)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910321455.6 2019年4月19日 (19.04.2019) CN

(71) 申请人: 深圳前海微众银行股份有限公司 (**WEBANK CO., LTD**) [CN/CN]; 中国广东省深圳市南山区沙河西路1819号深圳湾科技生态园7栋A座, Guangdong 518052 (CN)。

(72) 发明人: 刘建波 (**LIU, Jianbo**); 中国广东省深圳市南山区沙河西路1819号深圳湾科技生态园7栋A座, Guangdong 518052 (CN)。

(74) 代理人: 深圳市世纪恒程知识产权代理事务所 (**CENFO INTELLECTUAL PROPERTY AGENCY**); 中国广东省深圳市南山区西丽街道松坪山社区松坪山路3号奥特迅电力大厦201, Guangdong 518052 (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, 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, 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。

WO 2020/211554 A1

(54) **Title:** CACHE PROCESSING METHOD, APPARATUS AND DEVICE, AND COMPUTER READABLE STORAGE MEDIUM

(54) 发明名称: 缓存处理方法、装置、设备及计算机可读存储介质

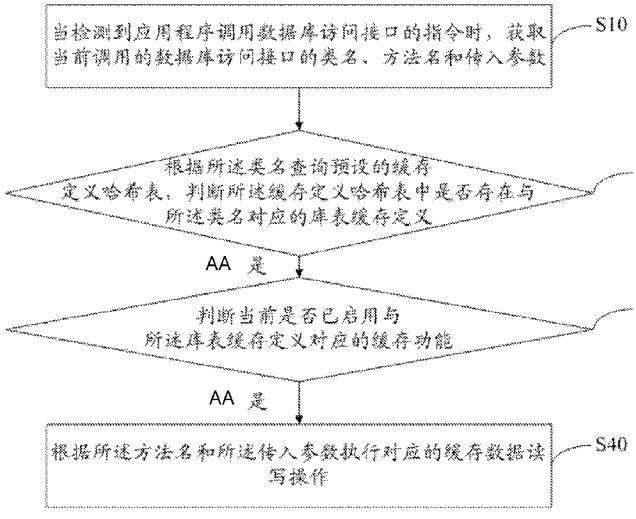

图 2

S10 When an instruction of an application calling a database access interface is detected, obtain a class name, a method name, and an incoming parameter of the currently called database access interface

S20 Query a preset cache definition hash table according to the class name, and determine whether a library table cache definition corresponding to the class name exists in the cache definition hash table

S30 Determine whether a cache function corresponding to the library table cache definition is currently enabled

S40 Execute a corresponding cache data read-write operation according to the method name and the incoming parameter

AA Yes

(57) **Abstract:** A cache processing method, apparatus and device, and a computer readable storage medium, improving the flexibility of a JVM local thread level cache scheme. The method comprises: when an instruction of an application calling a database access interface is detected, obtaining a class name, a method name, and an incoming parameter of the currently called database access interface (S10); querying a preset cache definition hash table according to the class name, and determining whether a library table cache definition corresponding to the class name exists in the cache definition hash table (S20); if yes, determining whether a thread level cache function corresponding to the library table cache definition is currently enabled (S30); and if yes, executing a corresponding thread level cache data read-write operation according to the method name and the incoming parameter (S40).

[见续页]

(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：一种缓存处理方法、缓存处理装置、设备和计算机可读存储介质，提高了JVM本地线程级缓存方案的灵活性。该方法包括：当检测到应用程序调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数（S10）；根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义（S20）；是，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能（S30）；是，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作（S40）。

发明名称：缓存处理方法、装置、设备及计算机可读存储介质

[0001] 本申请要求于2019年4月19日提交中国专利局、申请号为201910321455.6、发明名称为“缓存处理方法、装置、设备及计算机可读存储介质”的中国专利申请的优先权，其全部内容通过引用结合在申请中。

技术领域

[0002] 本申请涉及金融科技（Fintech）技术领域，尤其涉及缓存处理方法、装置、设备及计算机可读存储介质。

背景技术

[0003] 近年来，随着互联网技术，尤其是互联网金融科技（Fintech）的飞速发展，越来越多的技术（大数据、分布式、区块链Blockchain、人工智能等）应用在金融领域，金融数据也呈几何级增长，以大型商业银行为例，通常它们拥有成百上千个业务系统以及上亿用户的海量数据，这种情况下，对数据存储的数据量，并发性和响应速度都提出了更高要求。目前在金融领域，在使用JVM（Java Virtual Machine，Java虚拟机）访问金融数据库表时会用到缓存技术，即，将要操作的数据库表记录加载到本地JVM内存，以提高程序访问数据的速度，并降低数据库压力。

[0004] 在现有的基于Java程序语言的数据持久化框架（如Mybatis 、Hibernate）中，线程级缓存无法做到细粒度控制，即一旦缓存启用之后，会对所有库表的查询结果做缓存，这会导致占用过多的内存空间，影响系统性能。因而，现有的JVM本地线程级缓存方案的灵活性还有待提高。

发明概述

技术问题

问题的解决方案

技术解决方案

[0005] 本申请的主要目的在于提出一种缓存处理方法、装置、设备及计算机可读存储介质，旨在提高JVM本地线程级缓存方案的灵活性。

[0006] 为实现上述目的，本申请提供一种缓存处理方法，所述缓存处理方法包括如下步骤：

[0007] 当检测到应用程序调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

[0008] 根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

[0009] 若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

[0010] 若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[0011] 此外，为实现上述目的，本申请还提供一种缓存处理装置，所述缓存处理装置包括：

[0012] 获取模块，用于当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

[0013] 第一判断模块，用于根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

[0014] 第二判断模块，用于若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

[0015] 执行模块，用于若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[0016] 此外，为实现上述目的，本申请还提供一种缓存处理设备，所述缓存处理设备包括：存储器、处理器及存储在所述存储器上并可在所述处理器上运行的缓存处理程序，所述缓存处理程序被所述处理器执行时实现如上所述的缓存处理方法的步骤。

[0017] 此外，为实现上述目的，本申请还提供一种计算机可读存储介质，所述计算机可读存储介质上存储有缓存处理程序，所述缓存处理程序被处理器执行时实现如上所述的缓存处理方法的步骤。

[0018] 本申请在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当

前调用的数据库访问接口的类名、方法名和传入参数；根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；若存在，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。这种方式相比于现有技术中对所有库表的查询结果做缓存的方式，实现了线程级缓存的细粒度控制，即，只有当预设的缓存定义哈希表中存在与当前调用的数据库访问接口的类名对应的库表缓存定义，且当前已启用与所述库表缓存定义对应的线程级缓存功能时，才会执行对应的线程级缓存数据读写操作，从而本申请提高了JVM本地线程级缓存方案的灵活性。

发明的有益效果

对附图的简要说明

附图说明

[0019] 图1是本申请实施例方案涉及的硬件运行环境的设备结构示意图；

[0020] 图2为本申请缓存处理方法第一实施例的流程示意图。

[0021] 本申请目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

发明实施例

本发明的实施方式

[0022] 应当理解，此处所描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

[0023] 如图1所示，图1是本申请实施例方案涉及的硬件运行环境的设备结构示意图。

[0024] 本申请实施例缓存处理设备可以是PC机或服务器设备，其上运行有Java虚拟机。

[0025] 如图1所示，该缓存处理设备可以包括：处理器1001，例如CPU，网络接口1004，用户接口1003，存储器1005，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。用户接口1003可以包括显示屏（Display）、输入单元比如键盘（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1005可以是高速RAM存储器，也可以是稳定的存储器（non-volatile memory），例如磁盘存储器。存储器1005可选的还可以是独立于前述处理器1001的存储装置。

[0026] 本领域技术人员可以理解，图1中示出的设备结构并不构成对设备的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件布置。

[0027] 如图1所示，作为一种计算机存储介质的存储器1005中可以包括操作系统、网络通信模块、用户接口模块以及缓存处理程序。

[0028] 在图1所示的设备中，网络接口1004主要用于连接后台服务器，与后台服务器进行数据通信；用户接口1003主要用于连接客户端（用户端），与客户端进行数据通信；而处理器1001可以用于调用存储器1005中存储的缓存处理程序，并执行下述缓存处理方法中的操作。

[0029] 基于上述硬件结构，提出本申请缓存处理方法实施例。

[0030] 参照图2，图2为本申请缓存处理方法第一实施例的流程示意图，所述方法包括：

[0031] 步骤S10，当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

[0032] 本实施例缓存处理方法应用于缓存处理设备，该缓存处理设备上运行有JVM（Java Virtual Machine，Java虚拟机），JVM采用Mybatis或Hibernate等基于Java程序语言的数据持久化框架实现对数据库进行操作。比如在金融领域中，经常需要使用JVM访问金融数据库表，如账务信息表，开户信息表，金融交易信息表等，这一过程中会用到缓存技术以满足金融业务的并发性和访问速度要求。

[0033] 在本实施例中，以采用Mybatis对数据库进行操作为例进行说明。 Mybatis访问数据库的接口称为Mapper，当JVM在应用程序的预设线程中检测到调用数据库访问接口Mapper的指令时，获取当前调用的Mapper类名、Mapper方法名和传入参数，其中，传入参数用于表示要进行读写的域（Domain）对象，一个域对象

对应于一条数据库表记录。

[0034] 具体地，可以预先在所有Mapper上定义拦截器，比如：

[0035] @Around("execution(public * com.xxx..mapper..*.*(..)) && bean(*Mapper)")，表示对所有Mapper中的所有方法都执行拦截。系统运行时，每次对Mapper的调用，都会被拦截器拦截，拦截器获取到当次调用的Mapper类名、Mapper方法名和传入参数。

[0036] 需要说明的是，在本实施例中，可以采用Mybatis代码生成器生成访问数据库的代码，以提高开发效率，并使得所有Mapper和其方法的命名规则统一，从而保证拦截器能够拦截访问数据库的代码；此外也可以不采用Mybatis代码生成器，只要Mybatis Mapper命名和其方法命名规则统一，拦截器也可对其进行拦截。

[0037] 步骤S20，根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

[0038] 该步骤中，根据上述获取到的Mapper类名查询预设的缓存定义哈希表，判断该缓存定义哈希表中是否存在与Mapper类名对应的库表缓存定义。

[0039] 具体地，可以预先在系统中设置一个JVM级的缓存定义区，该缓存定义区用于保存缓存定义哈希表，该缓存定义哈希表表现为一个HashMap，该HashMap实现了“Key-Value键值对”接口，结构如下：

[] [表1]

Key	Value
mapperName1	sysTransCacheDef1
mapperName2	sysTransCacheDef2
……	……

[0040] 其中，Key为SysTransCacheDef域对象的mapperName（Mapper类名），value为SysTransCacheDef域对象本身。

[0041] 如果缓存定义哈希表中存在当前调用的Mapper类名，则可以判定该缓存定义哈希表中存在与该Mapper类名对应的库表缓存定义，反之，则判定该缓存定义哈

希表中不存在与该Mapper类名对应的库表缓存定义。

[0042] 若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则执行步骤S30，判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

[0043] 该步骤中，若缓存定义哈希表中存在与当前调用的Mapper类名对应的库表缓存定义，则进一步判断当前是否启用与该库表缓存定义对应的线程级缓存功能。

[0044] 具体实施时，可以在SysTransCacheDef域对象中用一个缓存功能使能位enabled来控制是否启用与该SysTransCacheDef域对象对应的线程级缓存功能，通过获取enabled值，即可判断当前是否启用与该库表缓存定义对应的线程级缓存功能，比如，若检测到enabled=1，则判定当前已启用与库表缓存定义对应的线程级缓存功能，若检测到enabled=0，则判定当前未启用与库表缓存定义对应的线程级缓存功能。

[0045] 若当前已启用与所述库表缓存定义对应的线程级缓存功能，则执行步骤S40，根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[0046] 该步骤中，若当前已启用与库表缓存定义对应的线程级缓存功能，则根据Mapper方法名和传入参数执行对应的线程级缓存数据读写操作，其中，Mapper方法名用于标识所执行的线程级缓存数据读写操作的类型，其包括数据查询、数据插入、数据更新、数据删除等操作。比如，当Mapper方法名标识数据查询操作时，先根据传入参数从当前线程的缓存数据区中查询数据，若缓存数据区中存在对应的数据，则返回查询结果，若缓存数据区中不存在对应的数据，则再从数据库中查询，并返回查询结果，并将查询结果保存在缓存数据区中；当Mapper方法名标识数据插入操作时，先在数据库中插入对应的数据，然后根据传入参数将插入的数据保存在当前线程的缓存数据区中；当Mapper方法名标识数据更新操作时，先对数据库中的对应的数据进行更新，然后根据传入参数将更新后的数据保存在当前线程的缓存数据区中。由此，实现了对缓存数据进行不同类型的读写操作。

[0047] 需要说明的是，本实施例中所提及的缓存数据区的缓存均为线程级的缓存，缓存数据区中的缓存数据仅对当前线程可用，线程结束后缓存自动销毁，该线程级的缓存尤其适用于在同一个线程中需要对相同记录做多次操作的库表。

[0048] 本实施例当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；若存在，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。这种方式相比于现有技术中对所有库表的查询结果做缓存的方式，实现了线程级缓存的细粒度控制，即，只有当预设的缓存定义哈希表中存在与当前调用的数据库访问接口的类名对应的库表缓存定义，且当前已启用与所述库表缓存定义对应的线程级缓存功能时，才会执行对应的线程级缓存数据读写操作，从而本实施例提高了JVM本地线程级缓存方案的灵活性。

[0049] 进一步地，基于本申请缓存处理方法第一实施例，提出本申请缓存处理方法第二实施例。

[0050] 在本实施例中，上述步骤S10之前，还可以包括：在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

[0051] 在本实施例中，在应用程序启动时需加载缓存定义到JVM内存中。具体实施时，首先读取预设的缓存配置表，该缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息，该缓存配置信息可以包括Mybatis访问数据库的接口的类名、Mybatis访问数据库的Domain域名、Domain中的主键属性字段、缓存功能使能位以及描述信息。具体地，缓存配置表sys_trans_cache_def的结构如下表所示：

[]

[表2]

列名	类型	可否为空	主键	备注
mapper_name	varchar(50)	NO	PRI	Mybatis访问数据库的接口的类名
domain_name	varchar(50)	NO		Mybatis访问数据库的Domain域名
primary_Key	varchar(100)	NO		Domain中的主键属性字段
enabled	char(1)	NO		缓存是否开启
description	varchar(100)	YES		描述信息

[0052] 比如，有一个银行客户账信息主表，表名为ca_acct_mast，其主键为entity_code + acct_no，需要将其配置为线程级缓存，则做配置如下：

[0053] mapper_name: CaAcctMastMapper

[0054] domain_name: CaAcctMast

[0055] primary_Key: entityCode,acctNo

[0056] enabled: 1

[0057] description: 客户账信息主表

[0058] 在读取到缓存配置表后，获取该缓存配置表中的所有记录，得到一个由SysTransCacheDef域对象组成的链表，将该链表重新组装为一个缓存定义哈希表HashMap；在该缓存定义哈希表中，Key为SysTransCacheDef域对象的mapperName（Mapper类名），value为SysTransCacheDef域对象本身；之后，将该HashMap保存至预设的JVM级的缓存定义区中，以供后续查询。

[0059] 上述方式通过将在应用程序启动时加载缓存定义到JVM内存中，为后续根据当前调用的数据库访问接口的类名查询预设的缓存定义哈希表提供了前提保证。

[0060] 进一步地，所述缓存处理方法还可以包括：接收基于所述缓存配置表的修改指

令，根据所述修改指令修改所述缓存配置表中的缓存配置信息；根据修改后的所述缓存配置信息生成一个新的缓存定义哈希表；将所述java虚拟机级缓存定义区中保存的所述缓存定义哈希表更新为所述新的缓存定义哈希表。

[0061] 在本实施例中，用户可以触发修改指令以修改上述缓存配置表。当系统接收到基于缓存配置表的修改指令时，即根据该修改指令修改缓存配置表中的缓存配置信息，其中修改包括增加、删除和改动。比如，当用户想要新增缓存定义时，只需在缓存配置表中插入一条记录；当用户想要禁用某一条缓存时，只需将缓存配置表中对应的记录删除，或将enabled位改为0。之后，系统根据修改后的缓存配置信息生成一个新的缓存定义哈希表，并将java虚拟机级缓存定义区中保存的缓存定义哈希表更新为新的缓存定义哈希表。

[0062] 需要说明的是，在一实施方式中，用户在修改缓存配置表中的缓存配置信息后，可以手动触发更新指令以更新缓存定义哈希表；在另一实施方式中，也可以通过创建一个轮询线程，通过该轮询线程定时触发（比如每隔30秒）读取缓存配置表中的全部数据，并根据读取到的数据对缓存定义哈希表进行更新。

[0063] 本实施例通过提供对缓存配置表的修改功能，不需要修改代码或配置文件，不需要重新部署，也不需要重启系统，只需要对缓存配置表做增、删、改，即可达到启用缓存、禁用缓存的目的，灵活程度非常高。

[0064] 进一步地，基于本申请缓存处理方法第一实施例，提出本申请缓存处理方法第三实施例。

[0065] 在本实施例中，上述步骤S40可以包括：当所述方法名标识未加记录锁的数据查询操作时，根据所述传入参数查询预设的第一线程级缓存数据哈希表，得到第一查询结果；判断所述第一查询结果是否为空；若所述第一查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表中。

[0066] 当Mapper方法名标识未加记录锁的数据查询操作时，先根据传入参数获得访问数据库的域名domain_name，再根据domain_name和传入参数从预设的第一线程级缓存数据哈希表中查询，得到第一查询结果。该第一线程级缓存数据哈希表保存在预设的第一线程级缓存数据区中，其结构如下表所示：

[] [表3]

Key	Value
domainName1 + primaryKey1	domain1
domainName1 + primaryKey2	domain2
domainName2 + primaryKey3	domain3
… …	… …

[0067] 其中，Key为域名+主键名，value为域对象，一个域对象对应数据库表中的一条记录。

[0068] 若该第一查询结果不为空，说明第一线程级缓存数据区中存在对应的数据，此时返回查询结果即可；若该第一查询结果为空，说明第一线程级缓存数据区中不存在对应的数据，此时执行调用数据库访问接口的操作，以从数据库中查询对应的数据，并将对应的调用结果保存至第一线程级缓存数据哈希表中，以供下次查询相同的数据时直接从缓存中读取。

[0069] 进一步地，上述步骤S40还可以包括：当所述方法名标识加记录锁的数据查询操作时，根据所述传入参数查询预设的第二线程级缓存数据哈希表，得到第二查询结果；判断所述第二查询结果是否为空；若所述第二查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中。

[0070] 在本实施例中，记录锁即当一个进程正在读或修改文件的某个部分时，它可以阻止其他进程修改同一文件区。当Mapper方法名标识加记录锁的数据查询操作（一般在更新数据时用到）时，先根据传入参数获得访问数据库的域名domain_name，再根据domain_name和传入参数从预设的第二线程级缓存数据哈希表中查询，得到第二查询结果。其中，第二线程级缓存数据哈希表保存在预设的第二线程级缓存数据区中，其结构与上述第一线程级缓存数据哈希表类似，此处不作赘述。

[0071] 若该第二查询结果不为空，说明第二线程级缓存数据区中存在对应的数据，此时返回查询结果即可；若该第二查询结果为空，说明第二线程级缓存数据区中

不存在对应的数据，此时执行调用数据库访问接口的操作，以从数据库中加锁查询对应的数据，并将对应的调用结果保存至第一线程级缓存数据哈希表和第二线程级缓存数据哈希表中，以供下次查询相同的数据时直接从缓存中读取。

[0072] 需要说明的是，设置第一线程级缓存数据哈希表和第二线程级缓存数据哈希表的目的在于区分不加记录锁的数据查询操作和加记录锁的数据查询操作，通过这种设置，如果先执行不加记录锁的数据查询操作，再对相同的记录执行加记录锁的数据查询操作，仍然需要操作数据库，如果先执行加记录锁的数据查询操作，再对相同的记录执行不加记录锁的数据查询操作，则不需要再查询数据库，从而提高了数据查询效率。

[0073] 进一步地，上述步骤S40还可以包括：当所述方法名标识数据更新操作时，执行调用数据库访问接口的操作，以在数据库中更新对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中；判断所述第二线程级缓存数据哈希表中是否存在与所述传入参数对应的更新对象，若存在，则根据所述传入参数对所述更新对象进行更新。

[0074] 当Mapper方法名标识数据更新操作时，执行调用数据库访问接口的操作，以在数据库中更新对应的数据，并将传入参数保存至第一线程级缓存数据哈希表中，然后判断第二线程级缓存数据哈希表中是否存在与该传入参数对应的更新对象，若存在，则根据该传入参数对该更新对象进行更新，以保证第二线程级缓存数据哈希表中保存的是最新的数据；若不存在，则不作任何处理。

[0075] 进一步地，上述步骤S40还可以包括：当所述方法名标识数据插入操作时，执行调用数据库访问接口的操作，以在数据库中插入对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中。

[0076] 当Mapper方法名标识当所述方法名标识数据插入操作时，执行调用数据库访问接口的操作，以在数据库中插入对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中，以使第一线程级缓存数据哈希表中的数据和数据库保持一致性。

[0077] 进一步地，上述步骤S40还可以包括：当所述方法名标识数据删除操作时，执行调用数据库访问接口的操作，以在数据库中删除对应的数据；根据所述传入

参数在所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中删除对应的数据。

[0078] 当Mapper方法名标识当所述方法名标识数据删除操作时，执行调用数据库访问接口的操作，以在数据库中删除对应的数据；然后，根据传入参数在第一线程级缓存数据哈希表和第二线程级缓存数据哈希表中删除对应的数据，以使第一线程级缓存数据哈希表和第二线程级缓存数据哈希表中的数据和数据库保持一致性。

[0079] 通过上述方式，实现了对数据查询、插入和更新操作进行缓存处理，相比于现有技术仅能对数据查询操作做缓存处理，提高了JVM缓存功能的全面性，从而有利于提高相应的数据读写效率；此外，上述线程级读写缓存的机制与业务逻辑完全解耦，业务层开发人员无需感知缓存的存在，只需要调用数据库访问接口读写数据库即可，相比于现有的在业务逻辑处理的工程中执行缓存读写的机制，降低了编码复杂度，提高了编码效率，且具有较高的灵活性。

[0080] 本申请还提供一种缓存处理装置。所述缓存处理装置包括：

[0081] 获取模块，用于当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

[0082] 第一判断模块，用于根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

[0083] 第二判断模块，用于若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

[0084] 执行模块，用于若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[0085] 进一步地，所述缓存处理装置还包括：

[0086] 读取模块，用于在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；

[0087] 生成模块，用于根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

[0088] 进一步地，所述缓存处理装置还包括：

[0089] 修改模块，用于接收基于所述缓存配置表的修改指令，根据所述修改指令修改所述缓存配置表中的缓存配置信息；

[0090] 所述生成模块，还用于根据修改后的所述缓存配置信息生成一个新的缓存定义哈希表；

[0091] 更新模块，用于将所述java虚拟机级缓存定义区中保存的所述缓存定义哈希表更新为所述新的缓存定义哈希表。

[0092] 进一步地，所述执行模块还用于：

[0093] 当所述方法名标识未加记录锁的数据查询操作时，根据所述传入参数查询预设的第一线程级缓存数据哈希表，得到第一查询结果；

[0094] 判断所述第一查询结果是否为空；

[0095] 若所述第一查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表中。

[0096] 进一步地，所述执行模块还用于：

[0097] 当所述方法名标识加记录锁的数据查询操作时，根据所述传入参数查询预设的第二线程级缓存数据哈希表，得到第二查询结果；

[0098] 判断所述第二查询结果是否为空；

[0099] 若所述第二查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中。

[0100] 进一步地，所述执行模块还用于：

[0101] 当所述方法名标识数据更新操作时，执行调用数据库访问接口的操作，以在数据库中更新对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中；

[0102] 判断所述第二线程级缓存数据哈希表中是否存在与所述传入参数对应的更新对象，若存在，则根据所述传入参数对所述更新对象进行更新。

[0103] 进一步地，所述执行模块还用于：

[0104] 当所述方法名标识数据插入操作时，执行调用数据库访问接口的操作，以在数据库中插入对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈

希表中。

[0105] 进一步地，所述执行模块还用于：

[0106] 当所述方法名标识数据删除操作时，执行调用数据库访问接口的操作，以在数据库中删除对应的数据；

[0107] 根据所述传入参数在所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中删除对应的数据。

[0108] 上述各程序模块所执行的方法可参照本申请缓存处理方法各个实施例，此处不再赘述。

[0109] 本申请还提供一种计算机可读存储介质。

[0110] 本申请计算机可读存储介质上存储有缓存处理程序，所述缓存处理程序被处理器执行时实现如上所述的缓存处理方法的步骤。

[0111] 其中，在所述处理器上运行的缓存处理程序被执行时所实现的方法可参照本申请缓存处理方法各个实施例，此处不再赘述。

[0112] 需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者系统不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者系统所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者系统中还存在另外的相同要素。

[0113] 上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

[0114] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。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如上所述的一个存储介质(如ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端设备(可以是手机，计算机，服务器，空调器，或者网络设备等)执行本申请各个实施例所述的方法。

[0115] 以上仅为本申请的优选实施例，并非因此限制本申请的专利范围，凡是利用本

申请说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本申请的专利保护范围内。

权利要求书

[权利要求 1] 一种缓存处理方法，其中，所述缓存处理方法包括如下步骤：

当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[权利要求 2] 如权利要求1所述的缓存处理方法，其中，所述当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数的步骤之前，还包括：

在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；

根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

[权利要求 3] 如权利要求2所述的缓存处理方法，其中，所述缓存处理方法还包括：

接收基于所述缓存配置表的修改指令，根据所述修改指令修改所述缓存配置表中的缓存配置信息；

根据修改后的所述缓存配置信息生成一个新的缓存定义哈希表；

将所述java虚拟机级缓存定义区中保存的所述缓存定义哈希表更新为所述新的缓存定义哈希表。

[权利要求 4] 如权利要求1所述的缓存处理方法，其中，所述根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作的步骤包括：

当所述方法名标识未加记录锁的数据查询操作时，根据所述传入参数查询预设的第一线程级缓存数据哈希表，得到第一查询结果；

判断所述第一查询结果是否为空；

若所述第一查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表中。

[权利要求 5] 如权利要求4所述的缓存处理方法，其中，所述根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作的步骤包括：

当所述方法名标识加记录锁的数据查询操作时，根据所述传入参数查询预设的第二线程级缓存数据哈希表，得到第二查询结果；

判断所述第二查询结果是否为空；

若所述第二查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中。

[权利要求 6] 如权利要求5所述的缓存处理方法，其中，所述根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作的步骤包括：

当所述方法名标识数据更新操作时，执行调用数据库访问接口的操作，以在数据库中更新对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中；

判断所述第二线程级缓存数据哈希表中是否存在与所述传入参数对应的更新对象，若存在，则根据所述传入参数对所述更新对象进行更新。

[权利要求 7] 如权利要求6所述的缓存处理方法，其中，所述根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作的步骤包括：

当所述方法名标识数据插入操作时，执行调用数据库访问接口的操作，以在数据库中插入对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中。

[权利要求 8] 如权利要求7所述的缓存处理方法，其中，所述根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作的步骤包括：

当所述方法名标识数据删除操作时，执行调用数据库访问接口的操作，以在数据库中删除对应的数据；

根据所述传入参数在所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中删除对应的数据。

[权利要求 9] 一种缓存处理装置，其中，所述缓存处理装置包括：

获取模块，用于当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

第一判断模块，用于根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

第二判断模块，用于若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

执行模块，用于若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[权利要求 10] 如权利要求9所述的缓存处理装置，其中，所述缓存处理装置还包括：

读取模块，用于在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；

生成模块，用于根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

[权利要求 11] 如权利要求10所述的缓存处理装置，其中，所述缓存处理装置还包括：

修改模块，用于接收基于所述缓存配置表的修改指令，根据所述修改指令修改所述缓存配置表中的缓存配置信息；

所述生成模块，还用于根据修改后的所述缓存配置信息生成一个新的缓存定义哈希表；

更新模块，用于将所述java虚拟机级缓存定义区中保存的所述缓存定

义哈希表更新为所述新的缓存定义哈希表。

[权利要求 12] 如权利要求9所述的缓存处理装置，其中，所述执行模块还用于：

当所述方法名标识未加记录锁的数据查询操作时，根据所述传入参数查询预设的第一线程级缓存数据哈希表，得到第一查询结果；

判断所述第一查询结果是否为空；

若所述第一查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表中。

[权利要求 13] 如权利要求12所述的缓存处理装置，其中，所述执行模块还用于：

当所述方法名标识加记录锁的数据查询操作时，根据所述传入参数查询预设的第二线程级缓存数据哈希表，得到第二查询结果；

判断所述第二查询结果是否为空；

若所述第二查询结果为空，则执行调用数据库访问接口的操作，并将对应的调用结果保存至所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中。

[权利要求 14] 如权利要求13所述的缓存处理装置，其中，所述执行模块还用于：

当所述方法名标识数据更新操作时，执行调用数据库访问接口的操作，以在数据库中更新对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中；

判断所述第二线程级缓存数据哈希表中是否存在与所述传入参数对应的更新对象，若存在，则根据所述传入参数对所述更新对象进行更新。

[权利要求 15] 如权利要求14所述的缓存处理装置，其中，所述执行模块还用于：

当所述方法名标识数据插入操作时，执行调用数据库访问接口的操作，以在数据库中插入对应的数据，并将所述传入参数保存至所述第一线程级缓存数据哈希表中。

[权利要求 16] 如权利要求15所述的缓存处理装置，其中，所述执行模块还用于：

当所述方法名标识数据删除操作时，执行调用数据库访问接口的操作，以在数据库中删除对应的数据；

根据所述传入参数在所述第一线程级缓存数据哈希表和所述第二线程级缓存数据哈希表中删除对应的数据。

[权利要求 17] 一种缓存处理设备，其中，所述缓存处理设备包括：存储器、处理器及存储在所述存储器上并可在所述处理器上运行的缓存处理程序，所述缓存处理程序被所述处理器执行时实现如下步骤：

当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[权利要求 18] 如权利要求17所述的缓存处理设备，其中，所述缓存处理程序被所述处理器执行时还实现如下步骤：

在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；

根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

[权利要求 19] 一种计算机可读存储介质，其中，所述计算机可读存储介质上存储有缓存处理程序，所述缓存处理程序被处理器执行时实现如下步骤：

当在应用程序的预设线程中检测到调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数；

根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义；

若所述缓存定义哈希表中存在与所述类名对应的库表缓存定义，则判断当前是否已启用与所述库表缓存定义对应的线程级缓存功能；

若当前已启用与所述库表缓存定义对应的线程级缓存功能，则根据所

述方法名和所述传入参数执行对应的线程级缓存数据读写操作。

[权利要求 20] 如权利要求19所述的计算机可读存储介质，其中，所述缓存处理程序被处理器执行时还实现如下步骤：

在应用程序启动时，读取预设的缓存配置表，所述缓存配置表中记录有基于不同的数据库访问接口的类名设置的缓存配置信息；

根据所述缓存配置信息生成一个缓存定义哈希表，将所述缓存定义哈希表保存至预设的java虚拟机级缓存定义区中。

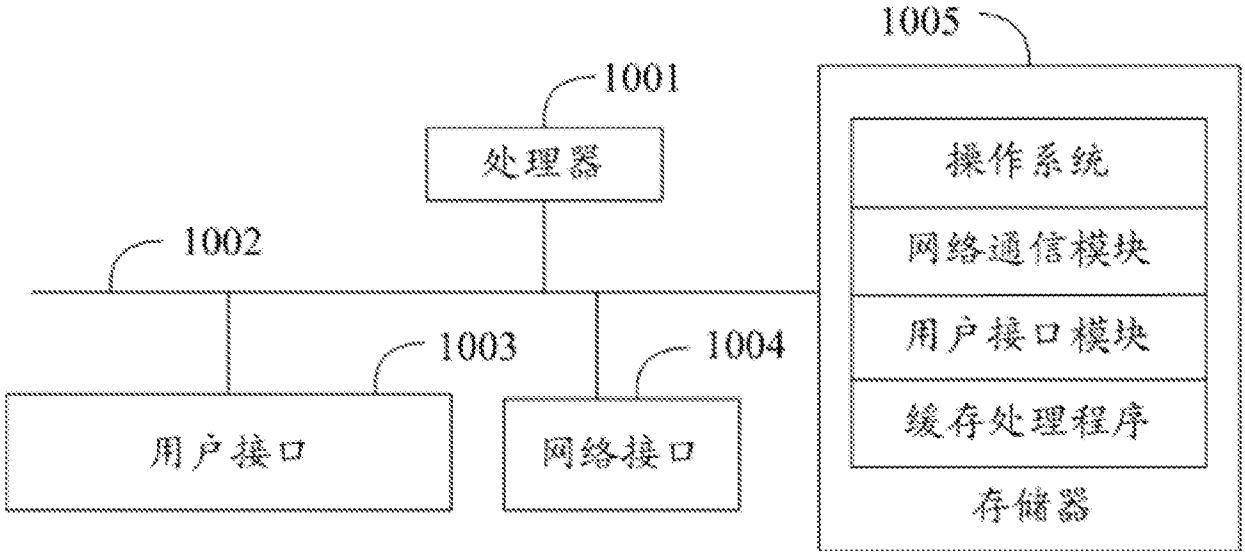
1005
1001
处理器
操作系统
网络通信模块
1002
用户接口模块
1003
1004
缓存处理程序
用户接口
网络接口
存储器

图 1

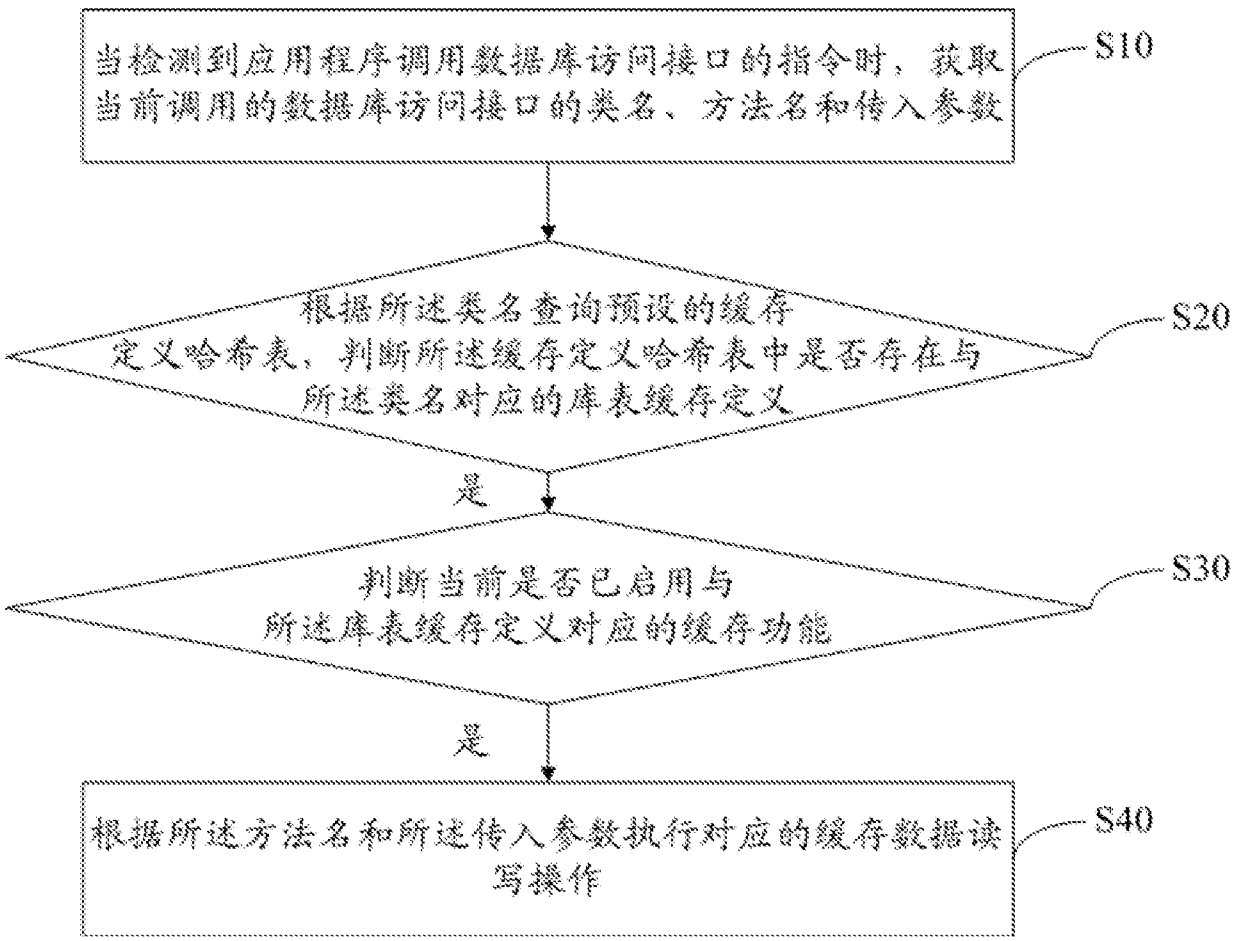
当检测到应用程序调用数据库访问接口的指令时，获取当前调用的数据库访问接口的类名、方法名和传入参数
S10
根据所述类名查询预设的缓存定义哈希表，判断所述缓存定义哈希表中是否存在与所述类名对应的库表缓存定义
S20
是
判断当前是否已启用与所述库表缓存定义对应的缓存功能
S30
是
根据所述方法名和所述传入参数执行对应的缓存数据读写操作
S40

图 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/077614

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/2455(2019.01)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)

CNPAT, WPI, EPODOC, CNKI, IEEE, GOOGLE: 缓存, 数据库, 线程, 名, 参数, 哈希, 定义, 读写, cache, database, thread, name, parameter, hash, definition, read, write

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 110109958 A (SHENZHEN QIANHAI WEIZHONG BANK CO., LTD.) 09 August 2019 (2019-08-09) claims 1-18	1-20
A	CN 105426238 A (ZHEJIANG WEIRONG ELECTRONIC CO., LTD.) 23 March 2016 (2016-03-23) description, paragraphs [0033]-[0046]	1-20
A	CN 101075241 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 21 November 2007 (2007-11-21) entire document	1-20
A	CN 104750720 A (CHINA UNIONPAY CO., LTD.) 01 July 2015 (2015-07-01) entire document	1-20
A	US 2016364315 A1 (ARIZONA BOARD OF REGENTS ON BEHALF OF ARIZONA STATE UNIVERSITY) 15 December 2016 (2016-12-15) entire document	1-20

☐ 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	Date of mailing of the international search report
19 May 2020	**27 May 2020**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/077614

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 110109958 A	09 August 2019	None	
CN 105426238 A	23 March 2016	None	
CN 101075241 A	21 November 2007	None	
CN 104750720 A	01 July 2015	None	
US 2016364315 A1	15 December 2016	None	

国际检索报告

国际申请号
PCT/CN2020/077614

A. 主题的分类

G06F 16/2455(2019.01)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））

CNPAT, WPI, EPODOC, CNKI, IEEE, GOOGLE：缓存，数据库，线程，名，参数，哈希，定义，读写，cache, database, thread, name, parameter, hash, definition, read, write

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 110109958 A (深圳前海微众银行股份有限公司) 2019年 8月 9日 (2019-08-09) 权利要求1-18	1-20
A	CN 105426238 A (浙江维融电子科技股份有限公司) 2016年 3月 23日 (2016-03-23) 说明书第[0033]-[0046]段	1-20
A	CN 101075241 A (腾讯科技深圳有限公司) 2007年 11月 21日 (2007-11-21) 全文	1-20
A	CN 104750720 A (中国银联股份有限公司) 2015年 7月 1日 (2015-07-01) 全文	1-20
A	US 2016364315 A1 (ARIZONA BOARD OF REGENTS ON BEHALF OF ARIZONA STATE UNIVERSITY) 2016年 12月 15日 (2016-12-15) 全文	1-20

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 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
2020年 5月 19日	2020年 5月 27日
ISA/CN的名称和邮寄地址 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	受权官员 胡百乐 电话号码 86-(10)-53961519

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2020/077614

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 110109958 A	2019年 8月 9日	无	
CN 105426238 A	2016年 3月 23日	无	
CN 101075241 A	2007年 11月 21日	无	
CN 104750720 A	2015年 7月 1日	无	
US 2016364315 A1	2016年 12月 15日	无	