(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112445851 B
(45)授权公告日 2024.07.16

(21)申请号 201910823664.0
(22)申请日 2019.09.02
(65)同一申请的已公布的文献号
申请公布号 CN 112445851 A
(43)申请公布日 2021.03.05
(73)专利权人 北京神州泰岳软件股份有限公司
地址 100080 北京市海淀区海淀大街34号8层818室
(72)发明人 冯中强
(74)专利代理机构 北京市隆安律师事务所 11323
专利代理师 权鲜枝
(51)Int.Cl.
G06F 16/25(2019.01)
G06F 16/27(2019.01)

(56)对比文件
CN 106777375 A,2017.05.31

审查员 黄蓉冰

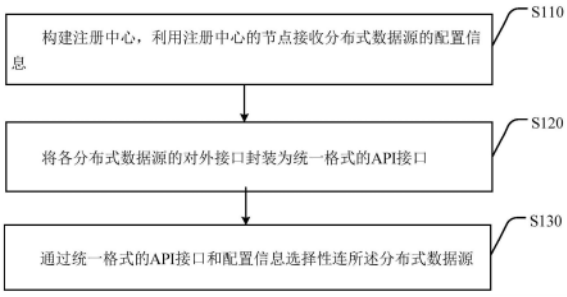
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种插拔式ORM框架实现方法、装置、电子设备和存储介质

(57)摘要

本发明公开了一种插拔式ORM框架实现方法、装置、电子设备和存储介质。所述方法包括：构建注册中心，利用所述注册中心的节点接收分布式数据源的配置信息；将各所述分布式数据源的对外接口封装为统一格式的API接口；通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。该方案通过注册中心的构建以及各数据源接口统一封装提高了对数据源底层操作的统一管理、配置、监控和排错的水平，数据源配置信息只需修改一次即可实现全局发布，大幅减少了工作量；并且可以解决团队成员水平参差不齐的问题，减少ORM框架维护和学习成本。



CN 112445851 B

1.一种插拔式ORM框架实现方法,其特征在于,所述方法包括:

构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息,其中,各机器应用程序通过JavaAPI获取注册中心中各个分布式数据源的配置信息,并且每个应用程序均在所述注册中心的节点注册监听器,以使得各机器获取最新的配置信息连接分布式数据源;

所述分布式数据源包括数据库和缓存库,所述构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息包括:

封装各所述分布式数据源中数据库API接口和缓存库API接口,并将各所述数据库API接口和缓存库API接口的信息提供给所述注册中心;

将各所述分布式数据源的对外接口封装为统一格式的API接口,形成各所述统一格式的API接口的命名空间,根据所述命名空间和所述API接口的地址配置数据源路由,所述统一格式的API接口以插拔式组件的形式部署在各机器应用程序和注册中心之间;

通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源,其中,各机器应用程序选择性接入一个或多个所述统一格式的API接口,并且通过路由寻址方式连接到注册中心,之后根据注册中心的配置信息连接到对应的分布式数据源。

2.如权利要求1所述的方法,其特征在于,所述统一格式的API接口至少包括如下的一项或几项信息:统一命名、统一参数、统一返回结果、统一对外输出、统一环境配置、统一jar依赖、统一日志。

3.如权利要求1-2任一项所述的方法,其特征在于,所述方法包括:

搭建Zookeeper服务,将所述Zookeeper服务确定为注册中心。

4.如权利要求1-2任一项所述的方法,其特征在于,所述方法包括:

搭建Springboot框架,利用所述Springboot框架的组件式开发实现各所述统一格式的API接口。

5.如权利要求4所述的方法,其特征在于,所述方法包括:

利用所述Springboot框架的Endpoint实现各所述分布式数据源的健康检查。

6.一种插拔式ORM框架实现装置,其特征在于,所述装置包括:

构建模块,适于构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息,其中,各机器应用程序通过JavaAPI获取注册中心中各个分布式数据源的配置信息,并且每个应用程序均在所述注册中心的节点注册监听器,以使得各机器获取最新的配置信息连接分布式数据源;

所述分布式数据源包括数据库和缓存库,所述构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息包括:

封装各所述分布式数据源中数据库API接口和缓存库API接口,并将各所述数据库API接口和缓存库API接口的信息提供给所述注册中心;

封装模块,适于将各所述分布式数据源的对外接口封装为统一格式的API接口,所述统一格式的API接口以插拔式组件的形式部署在各机器应用程序和注册中心之间;

所述封装模块还适于:

形成各所述统一格式的API接口的命名空间,根据所述命名空间和所述API接口的地址配置数据源路由;

连接模块,适于通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源,其中,各机器应用程序选择性接入一个或多个所述统一格式的API接口,并且通过路由寻址方式连接到注册中心,之后根据注册中心的配置信息连接到对应的分布式数据源。

7.一种电子设备,其中,该电子设备包括:处理器;以及被安排成存储计算机可执行指令的存储器,所述可执行指令在被执行时使所述处理器执行如权利要求1-5中任一项所述的方法。

8.一种计算机可读存储介质,其中,所述计算机可读存储介质存储一个或多个程序,所述一个或多个程序当被处理器执行时,实现如权利要求1-5中任一项所述的方法。

一种插拔式ORM框架实现方法、装置、电子设备和存储介质

技术领域

[0001] 本发明涉及ORM框架技术领域,具体涉及一种插拔式ORM框架实现方法、装置、电子设备和存储介质。

背景技术

[0002] ORM(Object Relational Mapping)框架采用元数据来描述对象_关系映射细节,元数据一般采用XML格式,并且存放在专门的对象_映射文件中,只要提供了持久化类与表的映射关系,ORM框架在运行时就能参照映射文件的信息,把对象持久化到数据库中。随着大数据时代的来临,数据关系的越来越复杂,对于业务数据,实时数据,缓存数据,都提出了不同的要求。采用ORM框架是实现上述数据集成是满足数据需求的有效方法。

[0003] 然而对于传统的ORM整合框架,使用起来比较笨重,各个项目无法根据自身的特点来有效的选择最合适的ORM整合框架。

[0004] 并且,分布式数据库大多部署在不同的机器上,用来保证数据批量的存储和查询的效率,对各数据库的连接和调用成为一个当前的难点。

[0005] 进一步地,由于各个厂家的数据库API的多样化,每个项目自己开发一个适合自己的ORM框架又不现实。

发明内容

[0006] 鉴于上述问题,提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的一种插拔式ORM框架实现方法、装置、电子设备和存储介质。

[0007] 依据本发明的一个方面,提供了一种插拔式ORM框架实现方法,所述方法包括:

[0008] 构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息;

[0009] 将各所述分布式数据源的对外接口封装为统一格式的API接口;

[0010] 通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。

[0011] 可选的,所述将各所述分布式数据源的对外接口封装为统一格式的API接口包括:

[0012] 形成各所述统一格式的API接口的命名空间,根据所述命名空间和所述API接口的地址配置数据源路由。

[0013] 可选的,所述统一格式的API接口至少包括如下的一项或几项信息:统一命名、统一参数、统一返回结果、统一对外输出、统一环境配置、统一jar依赖、统一日志。

[0014] 可选的,所述分布式数据源包括数据库和缓存库,所述构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息包括:

[0015] 封装各所述分布式数据源中数据库API接口和缓存库API接口,并将各所述数据库API接口和缓存库API接口的信息提供给所述注册中心。

[0016] 可选的,所述方法包括:搭建Zookeeper服务,将所述Zookeeper服务确定为注册中心。

[0017] 可选的,所述方法包括:搭建Springboot框架,利用所述Springboot框架的组件式

开发实现各所述统一格式的API接口。

[0018] 可选的,所述方法包括:利用所述Springboot框架的Endpoint实现各所述分布式数据源的健康检查。

[0019] 本发明的另一方面提供了一种插拔式ORM框架实现装置,所述装置包括:

[0020] 构建模块,适于构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息;

[0021] 封装模块,适于将各所述分布式数据源的对外接口封装为统一格式的API接口;

[0022] 连接模块,适于通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。

[0023] 依据本发明的又一方面,提供了一种电子设备,包括:处理器;以及被安排成存储计算机可执行指令的存储器,所述可执行指令在被执行时使所述处理器执行如上述任一所述的方法。

[0024] 依据本发明的再一方面,提供了一种计算机可读存储介质,其中,所述计算机可读存储介质存储一个或多个程序,所述一个或多个程序当被处理器执行时,实现如上述任一所述的方法。

[0025] 由上述可知,本发明的技术方案,提供了一种插拔式ORM框架实现方法,所述方法包括:构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息;将各所述分布式数据源的对外接口封装为统一格式的API接口;通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。该方案通过注册中心的构建以及各数据源接口统一封装提高了对数据源底层操作的统一管理、配置、监控和排错的水平,数据源配置信息只需修改一次即可实现全局发布,减少了修改工作量;还可以解决团队成员水平参差不齐的问题,减少ORM框架维护、学习的成本。

[0026] 上述说明仅是本发明技术方案的概述,为了能够更清楚了解本发明的技术手段,而可依照说明书的内容予以实施,并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂,以下特举本发明的具体实施方式。

附图说明

[0027] 通过阅读下文优选实施方式的详细描述,各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的,而并不认为是对本发明的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

[0028] 图1示出了根据本发明一个实施例的一种插拔式ORM框架实现方法的流程示意图;

[0029] 图2示出了根据本发明一个实施例的一种插拔式ORM框架实现装置的结构示意图;

[0030] 图3示出了根据本发明一个实施例的电子设备的结构示意图;

[0031] 图4示出了根据本发明一个实施例的计算机可读存储介质的结构示意图。

具体实施方式

[0032] 下面将参照附图更详细地描述本发明的示例性实施例。虽然附图中显示了本发明的示例性实施例,然而应当理解,可以以各种形式实现本发明而不应被这里阐述的实施例所限制。相反,提供这些实施例是为了能够更透彻地理解本发明,并且能够将本发明的范围

完整的传达给本领域的技术人员。

[0033] 图1示出了根据本发明一个实施例的一种插拔式ORM框架实现方法的流程示意图;所述方法包括:

[0034] 步骤S110,构建注册中心,利用注册中心的节点接收分布式数据源的配置信息。

[0035] 该实施例中的插拔式引入了即插即用的概念,只要将相应接口模块或接口组件引用到程序中,通过上述接口模块或组件提供的统一API接口即可实现与数据库的连接。在该实施例公开的所述方法中,首先构建各分布式数据源的注册中心,将各分布式数据源的配置信息保存在注册中心的节点上,从而建立起各分布式数据源与注册中心的通信连接。

[0036] 在实际应用中,各台机器上的应用程序可以使用JavaAPI去获取注册中心中各数据源的配置信息。每一个应用都在注册中心节点注册监听器,一旦节点中配置信息改变,各台机器就获取该配置信息,使用最新的信息连接数据库,这样,一是方便了管理,只需配置一份数据在注册中心,没必要放到多个机器上去;二是一旦配置信息改变了,在注册中心做一个发布的动作即可。

[0037] 步骤S120,将各所述分布式数据源的对外接口封装为统一格式的API接口。

[0038] 该步骤是在各机器应用程序和注册中心之间设计实现插拔式组件,该组件由各数据源对外统一格式的API接口子模块组成,根据各数据源的类型不同,可以有Mysql-API、Hbase-API、Redisl-API以及各种缓存库提供的API模块等,从而为应用程序连接各所述分布式数据源提供统一的API接口,其中,接口的统一格式可以包括统一的调用参数、统一的返回类型、统一的对外输出结果等内容。

[0039] 步骤S130,通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。

[0040] 各应用程序根据数据请求的需要选择性接入一个或多个所述统一格式的API接口,进而可以通过路由寻址等方式连接到注册中心,然后根据注册中心的配置信息连接到对应的分布式数据源。

[0041] 综上,该实施例公开的方法,首先通过构建注册中心,方便了各分布式数据源的管理,减少了配置信息的工作量,且仅需要配置一份配置信息即可实现全局发布;并且通过设置统一的API接口,减少了开发工作量,提高了开发效率,为数据库健康检查以及诊断提供了条件;还可以解决团队成员水平参差不齐的问题,减少ORM框架维护、学习的成本。

[0042] 在一个实施例中,所述将各所述分布式数据源的对外接口封装为统一格式的API接口包括:形成各所述统一格式的API接口的命名空间,根据所述命名空间和所述API接口地址配置数据源路由。

[0043] 在该实施例中,通过配置数据源路由实现寻址与数据库的正确连接。为了能够实现统一的管理,对各分布式数据源的API接口进行命名,形成命名空间,然后建立各命名的接口和IP地址的映射关系,最终通过路由的方式建立应用程序与数据源之间准确的连接关系。

[0044] 在一个实施例中,所述统一格式的API接口至少包括如下的一项或几项信息:统一命名、统一参数、统一返回结果、统一对外输出、统一环境配置、统一jar依赖、统一日志。

[0045] 其中,统一命名包括类的命名,方法的命名,实现统一管理,方便阅读和维护。具体命名包括:类名(Mysql*),方法名,查询(select*),新增(insert*),删除(delete*),修改

(update*)等。统一参数包括条件查询,可以封装规则,批量插入更新参数定义为List等。统一返回结果,可以统一定义所有的查询返还List,所有的新增、删除、更新返回boolean等。统一对外输出,可以封装为ResultMessage等。统一环境配置,每个产品组都有各自的配置文件,非常有必要统一,从而实现多个产品的整合。统一jar依赖,对于分布式的开发,每个产品组都有各自的依赖,通过统一依赖实现多个产品的整合。统一日志,产品对于日志的存储和存储的格式要求也非常高,可以统一定义日志格式,存入到比如分布式全文检索框架ElasticSearch中。

[0046] 在一个实施例中,分布式数据源包括数据库和缓存库,构建注册中心,利用所述注册中心的节点接收分布式数据源的配置信息包括:封装各所述分布式数据源中数据库和缓存库API接口,并将各所述数据库和缓存库API接口的信息提供给所述注册中心。

[0047] 为了进一步实现数据库接口的统一,对各数据库自身API进行封装,并且将封装后的API信息作为配置信息注册到注册中心的相应节点上,从而方便后续的连接调用。

[0048] 在一个实施例中,所述方法包括:搭建Zookeeper服务,将所述Zookeeper服务确定为注册中心。

[0049] Zookeeper是一个分布式的,开放源码的分布式应用程序协调服务,为分布式应用提供一致性服务的软件,提供的功能包括:配置维护、域名服务、分布式同步、组服务等。将数据源配置信息注册到Zookeeper中,避免了将配置信息分别注册到不同的机器上,提高了注册和连接的效率。

[0050] 在一个实施例中,所述方法包括:搭建Springboot框架,利用所述Springboot框架的组件式开发实现各所述统一格式的API接口。

[0051] 选择Springboot实现组件式开发,从而减少了比如各API接口组件之间的相互联系,可以实现各API接口模块的独立开发,提高了开发效率,当然也可以利用上述框架实现其他组件的开发。

[0052] 在一个实施例中,所述方法包括:利用Springboot框架的Endpoint实现各所述分布式数据源的健康检查。

[0053] Springboot框架的Spring-boot-actuator可以帮助监控和管理Springboot应用,比如健康检查、审计、统计和HTTP追踪等,在实现上可以定义统一接口和模板类与各数据源建立监控连接,健康检查主要包括检测该数据源的端口状态是否正常。

[0054] 图2示出了根据本发明一个实施例的一种插拔式ORM框架实现装置的结构示意图;其中所述插拔式ORM框架实现装置200包括:

[0055] 构建模块210,适于构建注册中心,利用注册中心的节点接收分布式数据源的配置信息。

[0056] 该实施例中的插拔式引入了即插即用的概念,只要将相应接口模块或接口组件引用到程序中,通过上述接口模块或组件提供的统一API接口即可实现与数据库的连接。在该实施例公开的装置中,构建模块适于构建各分布式数据源的注册中心,将各分布式数据源的配置信息保存在注册中心的节点上,从而建立起各分布式数据源与注册中心的通信连接。

[0057] 在实际应用中,各台机器上的应用程序可以使用JavaAPI去获取注册中心中各数据源的配置信息。每一个应用都在注册中心节点注册监听器,一旦节点中配置信息改变,各

台机器就获取该配置信息,使用最新的信息连接数据库,这样,一是方便了管理,只需配置一份配置信息在注册中心,不必将配置信息放到多个机器上去;二是一旦配置信息改变了,在注册中心做一个发布的动作即可。

[0058] 封装模块220,适于将各所述分布式数据源的对外接口封装为统一格式的API接口。

[0059] 在各机器应用程序和注册中心之间设计实现插拔式组件,该组件由各数据源对外统一格式的API接口子模块组成,根据各数据源的类型不同,可以有Mysql-API、Hbase-API、Redisl-API以及各种缓存库提供的API模块等,从而为应用程序连接各所述分布式数据源提供统一的API接口,其中,接口的统一格式可以包括统一的调用参数、统一的返回类型、统一的对外输出结果等内容。

[0060] 连接模块230,适于通过所述统一格式的API接口和所述配置信息选择性连接所述分布式数据源。

[0061] 各应用程序根据数据请求的需要选择性接入一个或多个所述统一格式的API接口,进而可以通过路由寻址等方式连接到注册中心,然后根据注册中心的配置信息连接到对应的分布式数据源。

[0062] 在其他实施例中对上述公开的插拔式ORM框架实现装置作进一步的限定,分别与上述方法对应,比如还包括路由配置模块、健康检查模块等,这里不再赘述。

[0063] 综上所述,本发明公开的上述技术方案,通过注册中心的构建以及各数据源接口统一封装提高了对数据源底层操作的统一管理、配置、监控和排错的水平,数据源配置信息只需修改一次即可实现全局发布,减少了修改工作量;还可以解决团队成员水平参差不齐的问题,减少ORM框架维护、团队成员学习的成本。

[0064] 需要说明的是:

[0065] 在此提供的算法和显示不与任何特定计算机、虚拟装置或者其它设备固有相关。各种通用装置也可以与基于在此的示教一起使用。根据上面的描述,构造这类装置所要求的结构是显而易见的。此外,本发明也不针对任何特定编程语言。应当明白,可以利用各种编程语言实现在此描述的本发明的内容,并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0066] 在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

[0067] 类似地,应当理解,为了精简本发明并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而,并不应将该公开的方法解释成反映如下意图:即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说,如下面的权利要求书所反映的那样,发明方面在于少于前面公开的单个实施例的所有特征。因此,遵循具体实施方式的权利要求书由此明确地并入该具体实施方式,其中每个权利要求本身都作为本发明的单独实施例。

[0068] 本领域那些技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单

元或组件组合成一个模块或单元或组件,以及此外可以把它们分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外,可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述,本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0069] 此外,本领域的技术人员能够理解,尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征,但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如,在下面的权利要求书中,所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0070] 本发明的各个部件实施例可以以硬件实现,或者以在一个或者多个处理器上运行的软件模块实现,或者以它们的组合实现。本领域的技术人员应当理解,可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的插拔式ORM框架实现装置中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如,计算机程序和计算机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上,或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到,或者在载体信号上提供,或者以任何其他形式提供。

[0071] 例如,图3示出了根据本发明一个实施例的电子设备的结构示意图。该电子设备300包括处理器310和被安排成存储计算机可执行指令(计算机可读程序代码)的存储器320。存储器320可以是诸如闪存、EEPROM(电可擦除可编程只读存储器)、EPROM、硬盘或者ROM之类的电子存储器。存储器320具有存储用于执行上述方法中的任何方法步骤的计算机可读程序代码331的存储空间330。例如,用于存储计算机可读程序代码的存储空间330可以包括分别用于实现上面的方法中的各种步骤的各个计算机可读程序代码331。计算机可读程序代码331可以从一个或者多个计算机程序产品中读出或者写入到这一个或者多个计算机程序产品中。这些计算机程序产品包括诸如硬盘,紧致盘(CD)、存储卡或者软盘之类的程序代码载体。这样的计算机程序产品通常为例如图4所述的计算机可读存储介质。图4示出了根据本发明一个实施例的一种计算机可读存储介质的结构示意图。该计算机可读存储介质400存储有用于执行根据本发明的方法步骤的计算机可读程序代码331,可以被电子设备300的处理器310读取,当计算机可读程序代码331由电子设备300运行时,导致该电子设备300执行上面所描述的方法中的各个步骤,具体来说,该计算机可读存储介质存储的计算机可读程序代码331可以执行上述任一实施例中示出的方法。计算机可读程序代码331可以以适当形式进行压缩。

[0072] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制,并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项

来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

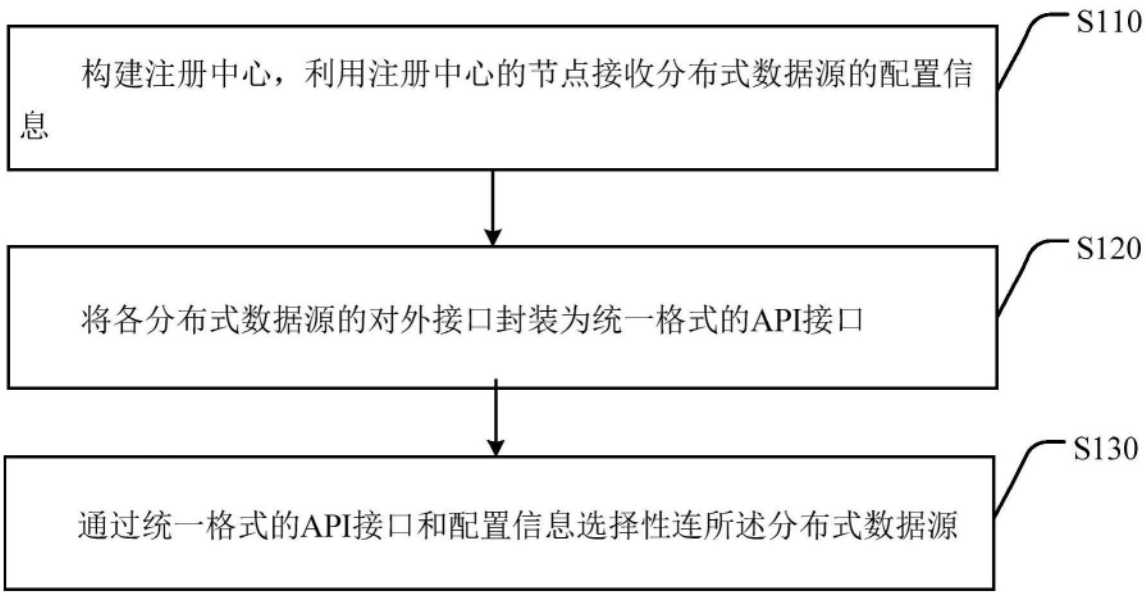

构建注册中心，利用注册中心的节点接收分布式数据源的配置信息
S110
将各分布式数据源的对外接口封装为统一格式的API接口
S120
通过统一格式的API接口和配置信息选择性连所述分布式数据源
S130

图1

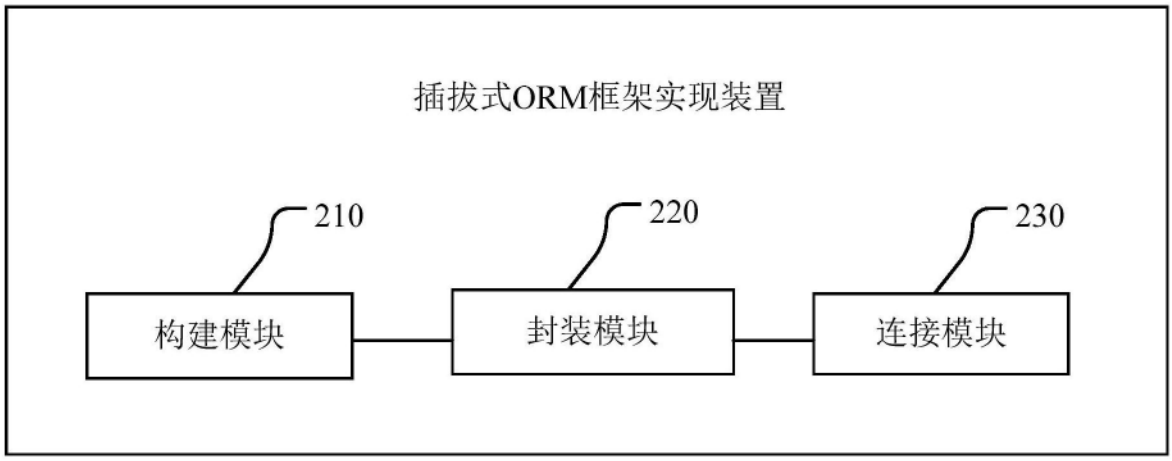

插拔式ORM框架实现装置
210
构建模块
220
封装模块
230
连接模块

图2

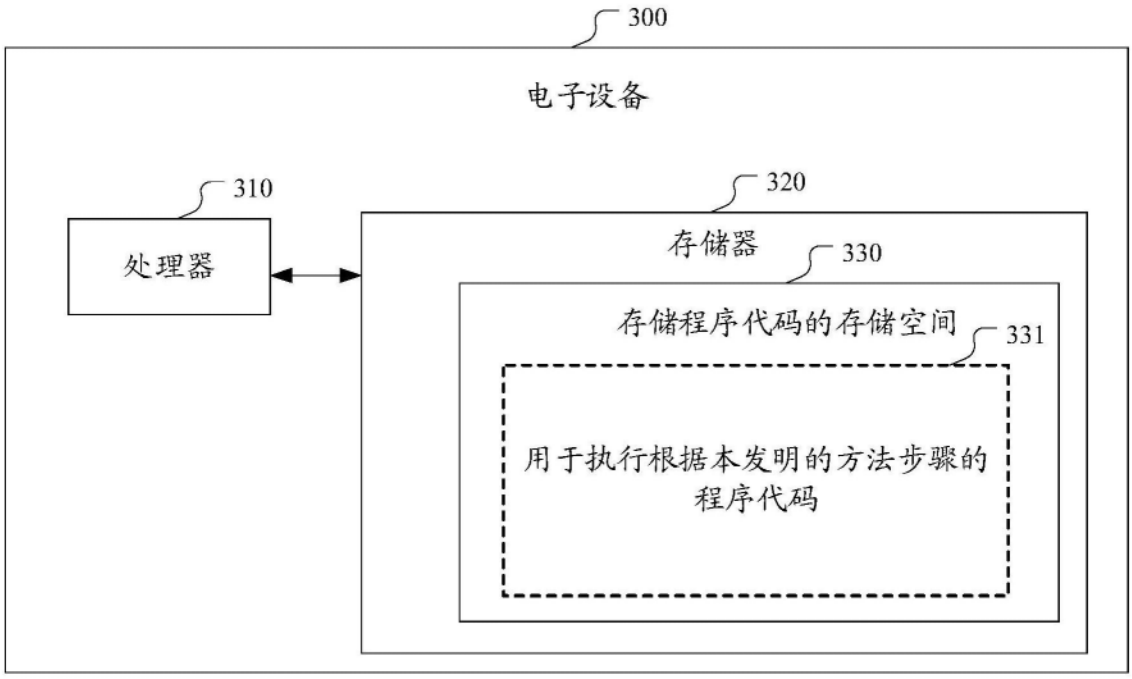

300
电子设备
310
处理器
320
存储器
330
存储程序代码的存储空间
331
用于执行根据本发明的方法步骤的
程序代码

图3

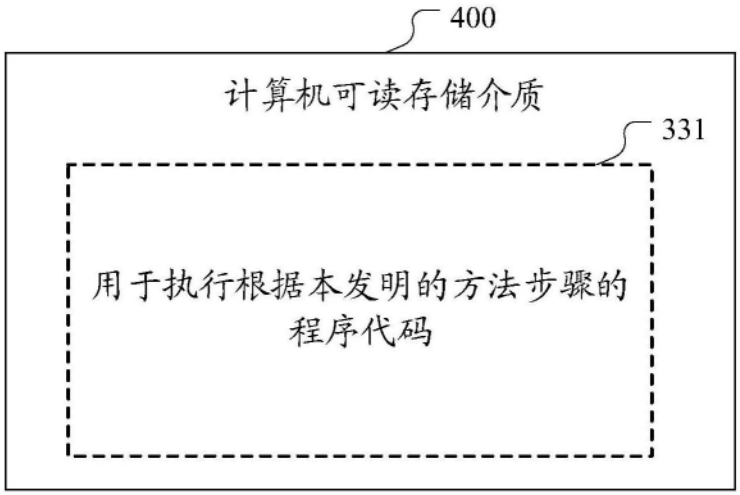

400
计算机可读存储介质
331
用于执行根据本发明的方法步骤的
程序代码

图4