



(12)发明专利

(10)授权公告号 CN 105095970 B

(45)授权公告日 2018.09.21

(21)申请号 201410172373.7

G06F 9/455(2006.01)

(22)申请日 2014.04.25

(56)对比文件

(65)同一申请的已公布的文献号

申请公布号 CN 105095970 A

CN 102214093 A ,2011.10.12,

CN 102239488 A ,2011.11.09,

(43)申请公布日 2015.11.25

(73)专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

王桥.《社会网络个性化隐私保护方法的研究与实现》.《中国优秀硕士学位论文全文数据库信息科技辑》.2013,(第1期),I139-137.

姜文广.《面向第三方平台的个性化隐私保护研究》.《中国优秀硕士学位论文全文数据库信息科技辑》.2013,(第11期),I139-98.

(72)发明人 张茂森

审查员 侯鹏

(74)专利代理机构 北京润泽恒知识产权代理有限公司 11319

代理人 苏培华

(51)Int.Cl.

G06F 8/20(2018.01)

G06F 8/60(2018.01)

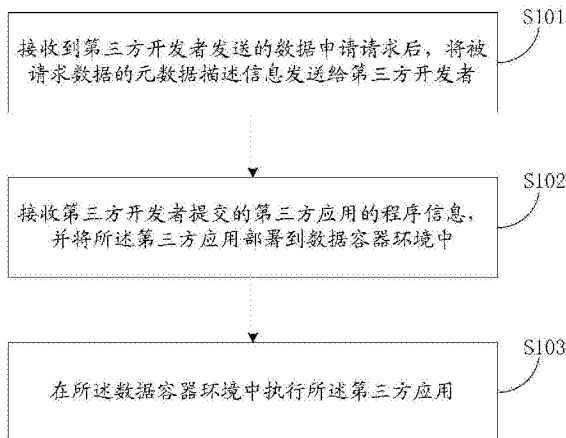
权利要求书2页 说明书9页 附图1页

(54)发明名称

第三方应用的执行方法及系统

(57)摘要

本申请公开了第三方应用的执行方法及系统,其中,所述方法包括:接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;接收第三方开发者提交的第三方应用的程序信息,并将所述第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;在所述数据容器环境中执行所述第三方应用。通过本申请,解决了数据安全风险和开放高价值数据的矛盾。



1. 一种第三方应用的执行方法,其特征在于,包括:

接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;

接收第三方开发者提交的第三方应用的程序信息,根据任务依赖关系和/或应用场景需求对第三方应用进行调度,并根据调度结果,将当前需要执行的第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

在所述数据容器环境中执行所述第三方应用。

2. 根据权利要求1所述的方法,其特征在于,还包括:

接收到第三方开发者提交的第三方应用的程序信息后,对第三方应用对数据的操作进行安全审核。

3. 根据权利要求1所述的方法,其特征在于,还包括:

接收到第三方开发者发送的数据申请请求后,向第三方开发者提供样本数据,以供第三方开发者对开发出的第三方应用的逻辑进行验证。

4. 根据权利要求1所述的方法,其特征在于,不同的应用类型对应不同的开发入口,所述方法还包括:

从某类型的开发入口接收到用户的开发请求时,为用户提供该类型应用的开发界面。

5. 根据权利要求4所述的方法,其特征在于,不同的应用类型对应不同的数据容器环境,所述方法还包括:

从某类型的开发入口接收到用户的开发请求后,将对应开发的第三方应用所属的类型进行标识;

所述将所述第三方应用部署到数据容器环境中,包括:

根据所述第三方应用所属的类型,将所述第三方应用部署到该类型对应的数据容器环境中。

6. 根据权利要求4或5所述的方法,所述应用类型包括计算类或算法类。

7. 根据权利要求1所述的方法,其特征在于,所述在所述数据容器环境中执行所述第三方应用,包括:

根据第三方应用需执行的任务对时间的要求,在所述数据容器环境中对所述第三方应用进行分布式或者同步处理。

8. 根据权利要求1至5、7任一项所述的方法,其特征在于,还包括:

所述第三方应用执行完成后,根据输出数据的标识封装成接口,以便第三方开发者根据所述接口获取第三方应用的执行结果。

9. 根据权利要求1至5、7任一项所述的方法,其特征在于,还包括:

接收第三方开发者提交的第三方应用的前台界面逻辑;

将所述前台界面逻辑部署到数据容器中,以便在所述第三方应用执行完成后,在前台界面中显示执行结果。

10. 一种第三方应用的执行系统,其特征在于,包括:

元数据描述信息发送单元,用于接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信

息进行第三方应用的开发；

应用部署单元,用于接收第三方开发者提交的第三方应用的程序信息,根据任务依赖关系和/或应用场景需求对第三方应用进行调度,并根据调度结果,将当前需要执行的第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

应用执行单元,用于在所述数据容器环境中执行所述第三方应用。

## 第三方应用的执行方法及系统

### 技术领域

[0001] 本申请涉及第三方应用开发技术领域,特别是涉及第三方应用的执行方法及系统。

### 背景技术

[0002] 平台型互联网应用(例如,某电子商务交易平台)为了给用户(例如,电子商务交易平台中的卖方用户)提供更细分、更垂直的服务,一般需要引入第三方开发者来完成。例如,第三方开发者可以对点击量、跨店铺点击、订单流转量、甚至相关即时通信工具中的聊天记录等信息,进行收集和分析,最终提供给卖方用户直观的建议。也就是说,对于某互联网应用的用户而言,在该互联网应用的网页中浏览到的一些数据分析结果等信息,通常是由第三方服务器提供的。

[0003] 现有技术中,互联网应用需要通过开放平台开放相关数据给第三方开发者,第三方开发者将获取到的数据存放入自有的存储系统,以使用这些数据来完成服务的开发以及后续的运行。

[0004] 但是,这种实现方式至少具有以下缺点:首先,数据被第三方开发者取走后,存在较大的数据泄露、滥用和不当使用等风险,因为无法对被取走的数据进行安全控制。其次,正是由于这样的安全缺陷,使得互联网应用一般不能开放高价值(隐私、安全、竞争等风险)的数据给第三方开发者,这就使得第三方开发者能够实现的功能也很有限。另外,第三方开发者自身的硬件处理能力比较有限,无法实现大数据量的计算处理。

[0005] 例如,某第三方开发者想要给某电子商务交易平台中的卖方用户提供商品补货预测服务,则现有技术中,第三方开发者就需要从开放平台将所需的数据取走,放入第三方开发者的环境中进行处理和计算。而补货预测需要对商品的销量进行预测,但由于数据安全的问题,开放平台不能直接提供预测模型需要的数据。例如行业商品销量(涉及到行业数据隐私)、同类商品成交的人群属性(涉及到行业和消费者数据隐私)等等。而对于卖方用户的补货渠道、补货策略等,这些也是比较核心的竞争力,直接提供给第三方开发者也存在一定信任问题。另外,预测模型需要考虑行业用户行为、商品交易行为等,数据量很大,第三方开发者可能不具备必要的软硬件开发环境等。总之,以上各种因素都可能导致该第三方服务器不能很好地为交易平台中的用户提供相关服务。

### 发明内容

[0006] 本申请提供了第三方应用的执行方法及系统,解决了数据安全风险和开放高价值数据的矛盾。

[0007] 本申请提供了如下方案:

[0008] 一种第三方应用的执行方法,包括:

[0009] 接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;

[0010] 接收第三方开发者提交的第三方应用的程序信息,并将所述第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

[0011] 在所述数据容器环境中执行所述第三方应用。

[0012] 一种第三方应用的执行系统,包括:

[0013] 元数据描述信息发送单元,用于接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;

[0014] 应用部署单元,用于接收第三方开发者提交的第三方应用的程序信息,并将所述第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

[0015] 应用执行单元,用于在所述数据容器环境中执行所述第三方应用。

[0016] 根据本申请提供的具体实施例,本申请公开了以下技术效果:

[0017] 通过本申请实施例,为第三方开发者提供了数据容器系统,将计算所需的数据按类别等存放在系统内部的数据容器中,而不是发送给第三方开发者,第三方开发者在开发完成第三方应用的计算逻辑或算法逻辑之后,将第三方应用也部署到该数据容器中,这样数据以及对数据的操作过程相当于都是在数据容器内部完成的,第三方开发者不用看到具体的数据内容,只要提供应用的计算逻辑或者算法就好,从而解决了数据安全风险和开放高价值数据的矛盾。

[0018] 当然,实施本申请的任一产品并不一定需要同时达到以上所述的所有优点。

## 附图说明

[0019] 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。

[0020] 图1是本申请实施例提供的方法的流程图;

[0021] 图2是本申请实施例提供的系统的示意图。

## 具体实施方式

[0022] 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所获得的所有其他实施例,都属于本申请保护的

范围。  
[0023] 在本申请实施例中,为了保证数据的安全性,并且使得第三方开发者能够提供更丰富的服务,提供了数据容器系统,将计算所需的数据按类别等存放在系统内部的数据容器中,而不是发送给第三方开发者,第三方开发者在开发完成第三方应用的计算逻辑或算法逻辑之后,将第三方应用也部署到该数据容器中,这样数据以及对数据的操作过程相当于都是在数据容器内部完成的,第三方开发者不用看到具体的数据内容,只要提供应用的

计算逻辑或者算法就好,从而解决了数据安全风险和开放高价值数据的矛盾。下面对具体的实现过程进行详细地介绍。

[0024] 首先需要说明的是,由于并不会直接将数据内容提供给第三方开发者,因此,在整个实现过程中,应用平台在第三方服务器进行服务开发的过程也要提供相应的支持。开发完成之后,应用平台还需要进行后续的部署等方面的支持。参见图1,本申请实施例提供的第三方应用的执行方法可以包括以下步骤:

[0025] S101:接收到第三方开发者发送的数据申请请求后,将被请求数据的元数据描述信息发送给第三方开发者,以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;

[0026] 为第三方开发者提供用于后台开发的界面,第三方开发者可以在其中编写代码、调试等等。例如,一般可以直接以网页的形式,也即第三方开发者直接登录应用平台的系统,就可以进入后台开发页面进行具体的开发操作。

[0027] 其中,由于数据容器中的数据众多,而第三方开发者一般只会提供某一方面的服务,因此,一般只会使用到其中的一部分数据,因此,可以预先对数据容器中的数据划分成多个类别,每个类别的数据可以对应一个数据主题。例如,在某电子商务交易平台中,数据主题可能包括发货、收货、订单管理、客服等主题。第三方开发者进入开发平台后,可以首先将各种可选的数据主题显示给第三方开发者,第三方服务器可以根据自己需要开发的服务,选择对应的数据主题。例如,某第三方开发者想要开发订单管理方面的服务,则可以选择订单管理这一数据主题,等等。

[0028] 在第三方开发者选择了某一数据主题之后,可以将该数据主题中可能会使用到的元数据描述信息提供给第三方开发者,以供第三方开发者进行应用的开发。需要说明的是,开放平台中的数据一般以数据表(关系型数据库中的表)的格式进行保存,因此,具体在向第三方开发者提供元数据描述信息时,可以包括数据表的名称,以及数据表中的字段信息。也就是说,可以使第三方开发者知晓数据表中有哪些字段,例如,对于某保存有用户信息的数据表A而言,其中的字段包括年龄、性别等字段,此时,就可以将该数据表A的名称,以及年龄、性别等字段名称提供给第三方开发者。这样,第三方开发者就可以基于这些字段名称,结合数据表的基本操作方法,来开发第三方应用中具体的逻辑。其中,对数据表的基本操作可以是支持多种编程语言,例如,JAVA、SQL等用于存取数据以及查询、更新和管理关系数据库系统的语言,来实现对数据表中各字段数据的操作。

[0029] 也就是说,在本申请实施例中,不需要将数据表中的具体数据提供给第三方开发者,而只需要将数据表中有哪些字段等信息提供给第三方开发者。当然,在实际应用中,为了使得第三方开发者在开发出具体的应用之后,对应用的计算逻辑等进行验证操作,开放平台可以为第三方开发者提供一些样本数据。这种样本数据可以是一些示例性的数据,例如,可以是将用户几年前的行为数据拼接在一起组合而成。由于已经是几年前的数据,并且是由一些数据片段组合而成的,因此,一般不会存在信息泄漏等问题。

[0030] 对于一些大型的第三方应用而言,需要的代码量往往非常大,编写代码的过程比较复杂,且容易出错。在本申请实施例中,为了简化开发过程,允许第三方开发者将一个应用拆分成多个子应用,在开发环境中分别开发各个子应用的代码,最后再由开放平台根据各个子应用之间的依赖关系等,将各个子应用关联起来,共同实现对应的第三方应用。

[0031] 另外,还可以将第三方应用划分为不同的类型,分别为各种类型的应用提供对应的开发界面,并且分别提供各种类型的开发入口。如果第三方开发者想要开发某类型的应用,则从该类型的开发入口进入,开放平台在通过该开发入口接收到用户的请求后,就可以显示出对应的开发界面,第三方开发者就可以在该界面中进行编写代码、调试等工作。例如,可以将第三方应用分为计算类应用以及算法类应用。同时,开放平台可以根据具体进入的界面,将对应开发的第三方应用所属的类型进行标识,以便后续在进行调度时,发送给相匹配的数据容器。

[0032] 需要说明的是,第三方开发者开发的同一个第三方应用可能由多个子应用组成,各子应用可以是不同类型的应用,并且,相互之间也可以进行数据传递。

[0033] 另外需要说明的是,在实际应用中,在接收到某第三方开发者发送的数据申请请求时,还可以对这种数据申请进行审核,例如,包括对第三方开发者的资质等进行的审核。再者,如果涉及到用户数据,还可以建立用户授权。也即,询问用户是否允许第三方开发者使用其数据,如果用户允许,再向第三方开发者返回相应的元数据描述信息、样本数据等。否则,如果没有获取到用户授权,则不会向第三方开发者返回相应的信息。

[0034] S102:接收第三方开发者提交的第三方应用的程序信息,并将所述第三方应用部署到数据容器环境中,其中,所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

[0035] 第三方开发者在完成了某第三方应用的开发之后,就可以将其提交到开发平台,相应的,开发平台就可以将第三方应用的逻辑部署到开放平台内部的数据容器环境中。由于该数据容器环境中保存有第三方应用在执行过程中所需的具体数据,因此,就可以在该数据容器环境中执行第三方应用。这样,具体的数据处理的过程都是在数据容器内部完成的,不需要将待处理的数据公开给第三方开发者。

[0036] 在实际应用中,在接收到第三方开发者提交的第三方应用之后,还可以对应用进行安全审核。例如,判断是否存在敏感数据,是否存在对数据的敏感操作等,进而,如果存在对敏感数据的操作,或者存在敏感操作,则可以在数据输出时进行拦截,进一步提高数据的安全性。具体在进行安全审核时可以有多种方式,例如,在一种简单的方式下,可以直接利用关键字等进行过滤;或者,判断第三方应用中是否存在关于某些特殊商品对象的交易信息;或者,对于可以量化的数据,还可以判断第三方应用使用的某行业数据的数据量所占总数据量的百分比,如果该百分比达到某数值,则可能会由此推导出行业的其他信息,导致该行业数据的泄漏,因此认为存在安全性问题。又或者,还可以在第三方应用中包含一些用户ID等信息的情况下,判断是否能够反推获知用户的真实身份等敏感信息,如果是,则也认为存在安全性问题,等等。总之,在进行安全性审核时,可以预先设置多个判断条件,对于某个第三方应用而言,可以利用这些判断条件进行判断,如果命中其中某个或者某几个条件,则认为存在安全性问题。

[0037] 如前文所述,不同的第三方应用之间可能存在一定的依赖关系。例如,某应用需要依赖于另一应用的执行结果,则该应用需要在另一应用执行完成结束之后执行。另外,还有一些第三方应用需要在特定的应用场景下执行,等等。因此,在向数据容器提交部署之前,可以根据依赖关系、应用场景等信息对提交上来的各个第三方应用进行调度,将当前需要执行的第三方应用提交到数据容器。

[0038] 向数据容器提交需要执行的的应用的同时,还可以提交一些控制命令,例如,该应用的优先级、资源分配情况(例如,分配多少的CPU、多少的内存给该应用)等,以便数据容器按照这些控制命令来执行。其中,关于资源分配情况,可以预估数据的输入输出量,并根据一定的算法计算出需要占用的资源,当然实际应用中,还可以根据第三方开发者订购的资源数量等信息来综合确定。

[0039] S103:在所述数据容器环境中执行所述第三方应用。

[0040] 在将第三方应用部署到数据容器之后,就可以在数据容器中执行第三方应用。具体在执行时,可以根据第三方应用任务的性质,进行同步或者异步的处理。相应的,数据容器可以分为“分布式处理容器”或“同步处理容器”。其中,前者相当于是将任务分配到多个容器分别执行,最后再进行汇总,一般用于处理对时间要求不高的任务;后者就是接收到执行任务后马上进行处理,一般用于处理对时间要求比较高的任务。因此,前述任务的性质可以是指对时间要求的高低。其中,关于任务的性质可以是由第三方开发者在开发时进行指定。

[0041] 在第三方应用执行完成之后,就可以将执行结果提供给用户使用。其中,具体在向用户提供执行结果时,可以有多种方式,其中一种方式是:可以由第三方开发者将执行结果取走,然后在其界面中展现执行结果。为了使得第三方开发者能够获取到执行结果,可以如下进行:对于第三方开发者开发完成的第三方应用代码,可以根据输出数据的标识,封装成接口(一般是一个URL),数据容器完成执行之后,可以记录执行结果与该URL的对应关系。第三方开发者请求执行结果时,可以根据该URL返回执行结果。具体实现时,可以有多种方式。例如,其中一种方式可以是,向各个第三方开发者提供统一的获取执行结果的入口,也即,每个第三方开发者在每次请求执行结果时,可向同一个URL发送请求,但是,同时要携带上第三方开发者的ID等标识信息,以及此次请求的输出数据的标识,开发平台在接收到该请求之后,就可以根据携带的ID以及输出数据的标识确定出所需的执行结果实际对应的URL,然后向第三方开发者返回执行结果即可。或者,在另一种实现方式下,也可以直接将执行结果实际对应的URL提供给第三方开发者,这样,第三方开发者就可以根据实际对应的URL请求获取对应的执行结果。

[0042] 另外,在实际应用中,还可能存在以下情况:对于在数据容器中执行得到的执行结果仍然存在一些敏感数据,此时,如果直接由第三方开发者将执行结果取走,则仍然有可能造成对用户数据的泄漏等危险。例如,某第三方应用用于筛选出一些用户进行某种信息的推广,则该第三方应用的执行结果中可能会包含用户的联系方式等信息,如果将这些信息提供给第三方开发者显然会存在风险。因此,对于这种情况,在具体实现时,还可以要求第三方开发者将其前台界面展现逻辑也部署到数据容器中,这样,在获取到执行结果之后,就可以直接根据数据容器中的前台界面展现逻辑,将执行结果直接提供给用户。

[0043] 为了便于理解,下面通过一个实际的例子进行详细的介绍。

[0044] 具体实现时,可以包括以下步骤:

[0045] 步骤1:第三方开发者登录到开放平台的后台;此时,界面中可以展现出各个可选择的数据主题;

[0046] 步骤2:第三方开发者申请某数据主题;

[0047] 步骤3:开放平台对第三方开发者的申请进行审核,主要用于确定该第三方开发者



是否具有申请该数据主题的资质；

[0048] 步骤4:判断数据申请审核是否通过;如果未通过,则返回失败等通知消息;如果通过,则进入步骤5;

[0049] 步骤5:第三方开发者申请样本数据;

[0050] 步骤6:开放平台向用户发送授权审核消息;

[0051] 步骤7:判断授权审核是否通过,如果未通过,则返回失败等通知消息;如果通过,则进入步骤8;

[0052] 步骤8:第三方开发者进入开发界面进行第三方应用的开发;

[0053] 步骤9:开发完毕之后,将第三方应用提交到开放平台进行审核;

[0054] 步骤10:开放平台对第三方应用进行数据安全审核;

[0055] 步骤11:判断数据安全审核是否通过,如果未通过,则可以返回审核失败等通知消息,如果通过,则继续进入步骤12;

[0056] 步骤12:第三方开发者将第三方应用提交到开放平台进行部署;

[0057] 步骤13:开放平台将第三方应用部署到数据容器进行执行;

[0058] 步骤14:将执行结果提供给用户进行使用。

[0059] 为了更好的理解本申请实施例的技术方案,下面将开发平台划分为多个模块,分别介绍各个模块的功能以及模块之间的关联关系。具体的,可以包括以下模块:

[0060] 登录验证模块:第三方开发者使用预先注册的账户信息登录到开放平台后,开放平台可以对第三方开发者的身份进行认证,完成身份认证后,将授予第三方开发者令牌,第三方开发者可以带上该令牌进入系统;

[0061] 授权验证模块:开放平台还可以对第三方开发者拥有的数据权限和操作权限进行认证,完成授权认证后,将授予第三方开发者对应的数据权限和应用开发权限;

[0062] 数据应用开发模块:集成数据应用开发环境,为第三方开发者提供开发界面,可以在其中编写代码、调试等等。具体实现时,数据应用开发模块可以由两部分组成,分别为通用计算类数据应用开发子模块和算法类数据应用开发子模块,第三方开发者可以根据待开发的子程序所属的类别选择进入对应的子模块。在开发界面中,可以提供进入两个子模块的入口,进入之后,两个子模块具有各自的开发界面。如果某第三方开发者想要开发计算类的子程序,则可以从通用计算类数据应用开发子模块的入口进入,进而打开通用计算类数据应用开发子模块对应的开发界面;类似的,第三方开发者想要开发算法类的子程序,则可以从通用算法类数据应用开发子模块的入口进入,进而打开算法类数据应用开发子模块对应的开发界面。

[0063] 其中,通用计算类数据应用开发模块的主要功能可以包括:

[0064] a) 应用项目管理:包括对应用项目的增删改查、分享、授权、协作等;

[0065] b) 代码开发环境界面:

[0066] i. 代码开发编辑器;

[0067] ii. 代码高亮、语法检查等效率辅助工具;

[0068] iii. 元数据管理:元数据详细描述(包括表标识、字段名等)、数据依赖、交互管理;

[0069] c) 代码沙箱环境管理界面:

[0070] i. 代码模拟运行(可以在开发完代码后,进行测试,对计算逻辑进行验证)

[0071] ii .数据集创建

[0072] d) 代码管理:代码版本管理、目录管理。当代码开发完成之后,开放平台可以将代码保存到代码版本管理中,后续如果需要对代码进行修改,则可以生成新的版本。在保存一份代码时,开放平台可以将其封装成对应的程序包,对于同一应用中的不同子程序对应的程序包而言,还可以标识各程序包的应用场景需求、启动条件(例如,需要在某时间点启动,或者需要在某指定程序运行完毕之后启动,等等),以及各程序包之间的依赖关系等信息。其中,关于程序包的启动条件可以是由第三方开发者指定的,而程序包之间的依赖关系,可以由第三方开发者指定,也可以由开放平台自动获取。例如,开放平台通过对各程序包的代码进行分析得知,程序包1的逻辑是对表a进行处理生成表b、c,而程序包2的逻辑是对表b处理生成表d,因此,可以判断出程序包2依赖程序包1,进而就可以自动标识该依赖关系。

[0073] e) 应用管理:应用发布、应用下架、应用状态查询。

[0074] f) 费用管理:交费、续费、结算、清算、余额管理。

[0075] 算法类数据应用开发子模块的主要功能可以包括:

[0076] a) 算法模型管理:增删改查、分享、授权、协作。

[0077] b) 基础模型库。

[0078] c) 数据特征分析环境:统计分布验证、特征提取、验证。

[0079] d) 模型训练管理:模型训练、结果验证。

[0080] 沙箱数据容器环境模块,提供代码、数据集验证功能的计算环境,包括:

[0081] a) 代码运行环境:

[0082] i .分布式代码运行环境(Map/Reduce程序、SQL程序)

[0083] ii .算法运行环境

[0084] b) 数据环境:提供代码运行的数据支撑环境,必要的数据库,数据集,数据集创建工具等,这些数据可以是开放平台提供的样本数据,例如,将用户几年前的产生的数据拼接在一起等等,生成具有相同的数据结构,但并不真是的数据,这种数据提供给第三方开发者,用于对其开发的代码进行验证。

[0085] 任务 workflow 调度模块:负责将第三方开发者提交的应用根据任务依赖关系、应用场景需求执行调度,并将需要执行的应用提交给数据容器环境(数据容器通过预置的接口监听任务 workflow 调度模块的提交请求,因此,当任务 workflow 调度模块发现有需要执行的应用时,就可以通过数据容器的接口进行发送)。其中,如果开发模块包括计算类数据应用开发子模块以及算法类数据应用开发子模块,则在调度时,可以根据开发过程中标识出的计算类应用或者算法类应用,将需要执行的应用发送到“通用计算容器环境”或者“算法容器环境”执行。另外,任务 workflow 调度模块在向数据容器提交需要执行的申请的同时,还可以提交一些控制命令,以便数据容器按照这些控制命令来执行。

[0086] 通用计算容器环境:

[0087] a) 负责接收任务 workflow 调度模块提交的通用计算任务,将计算任务根据性质不同提交给对应的“分布式处理容器”或“同步处理容器”。

[0088] b) 对任务进行监控、跟踪、数据采集

[0089] c) 和“数据查询中间层”打通,完成数据查询功能

[0090] 算法容器环境:

[0091] a) 负责接收任务 workflow 调度模块提交的算法任务, 将算法任务根据性质不同提交给对应的: “分布式处理容器”或“同步处理容器”。

[0092] b) “基础算法包”子模块。第三方开发者在算法类数据应用开发子模块中开发时, 可以根据自己的需求编写算法代码, 或者, 开放平台也可以为第三方开发者提供一些基础算法, 这样, 第三方开发者只要在基础算法的基础上进行选择需要的特征、调整算法的参数等操作即可, 以此提高开发效率。相应的, 在算法容器环境中, 还可以提供基础算法可执行调用模块, 其中, 基础算法可以包括:

[0093] i. 分类算法

[0094] ii. 聚类算法

[0095] iii. 回归算法

[0096] 数据安全审核引擎: 当第三方开发者完成代码的开发之后, 在部署到数据容器之前, 还可以进行安全审核。

[0097] 与本申请实施例提供的第三方应用的执行方法相对应, 本申请实施例还提供了一种第三方应用的执行系统, 参见图2, 该系统可以包括:

[0098] 元数据描述信息发送单元201, 用于接收到第三方开发者发送的数据申请请求后, 将被请求数据的元数据描述信息发送给第三方开发者, 以供第三方开发者基于所述元数据描述信息进行第三方应用的开发;

[0099] 应用部署单元202, 用于接收第三方开发者提交的第三方应用的程序信息, 并将所述第三方应用部署到数据容器环境中, 其中, 所述数据容器环境中还保存有第三方应用在执行过程中所需的具体数据;

[0100] 应用执行单元203, 用于在所述数据容器环境中执行所述第三方应用。

[0101] 具体实现时, 该系统还可以包括:

[0102] 调度单元, 用于根据任务依赖关系和/或应用场景需求对第三方应用进行调度;

[0103] 所述应用部署单元202具体用于:

[0104] 根据调度结果, 将当前需要执行的第三方应用部署到数据容器环境中。

[0105] 另外, 该系统还可以包括:

[0106] 安全审核单元, 用于接收到第三方开发者提交的第三方应用后, 对第三方应用对数据的操作进行安全审核。

[0107] 为了便于第三方进行开发, 该系统还可以包括:

[0108] 样本数据提供单元, 用于接收到第三方开发者发送的数据申请请求后, 向第三方开发者提供样本数据, 以供第三方开发者对开发出的第三方应用的逻辑进行验证。

[0109] 其中, 不同的应用类型对应不同的开发入口, 所述系统还可以包括:

[0110] 界面提供单元, 用于从某类型的开发入口接收到用户的开发请求时, 为用户提供该类型应用的开发界面。

[0111] 不同的应用类型对应不同的数据容器环境, 所述系统还可以包括:

[0112] 标识单元, 用于从某类型的开发入口接收到用户的开发请求后, 将对应开发的第三方应用所属的类型进行标识;

[0113] 所述应用部署单元202具体可以用于:

[0114] 根据所述第三方应用所属的类型, 将所述第三方应用部署到该类型对应的数据容

器环境中。

[0115] 所述应用类型包括计算类或算法类。

[0116] 具体的,应用执行单元203具体可以用于:

[0117] 根据第三方应用需执行的任务对时间的要求,在所述数据容器环境中对所述第三方应用进行分布式或者同步处理。

[0118] 另外,该系统还可以包括:

[0119] 封装单元,用于所述第三方应用执行完成后,根据输出数据的标识封装成接口,以便第三方开发者根据所述接口获取第三方应用的执行结果。

[0120] 还可以包括:

[0121] 前台界面逻辑接收单元,用于接收第三方开发者提交的第三方应用的前台界面逻辑;

[0122] 界面逻辑部署单元,用于将所述前台界面逻辑部署到数据容器中,以便在所述第三方应用执行完成后,在前台界面中显示执行结果。

[0123] 总之,通过本申请实施例,为第三方开发者提供了数据容器系统,将计算所需的数据按类别等存放在系统内部的数据容器中,而不是发送给第三方开发者,第三方开发者在开发完成第三方应用的计算逻辑或算法逻辑之后,将第三方应用也部署到该数据容器中,这样数据以及对数据的操作过程相当于都是在数据容器内部完成的,第三方开发者不用看到具体的数据内容,只要提供应用的计算逻辑或者算法就好,从而解决了数据安全风险和开放高价值数据的矛盾。

[0124] 通过以上的实施方式的描述可知,本领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现。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,该计算机软件产品可以存储在存储介质中,如ROM/RAM、磁碟、光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例或者实施例的某些部分所述的方法。

[0125] 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,对于系统或系统实施例而言,由于其基本相似于方法实施例,所以描述得比较简单,相关之处参见方法实施例的部分说明即可。以上所描述的系统及系统实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0126] 以上对本申请所提供的第三方应用的执行方法及系统,进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处。综上所述,本说明书内容不应理解为对本申请的限制。

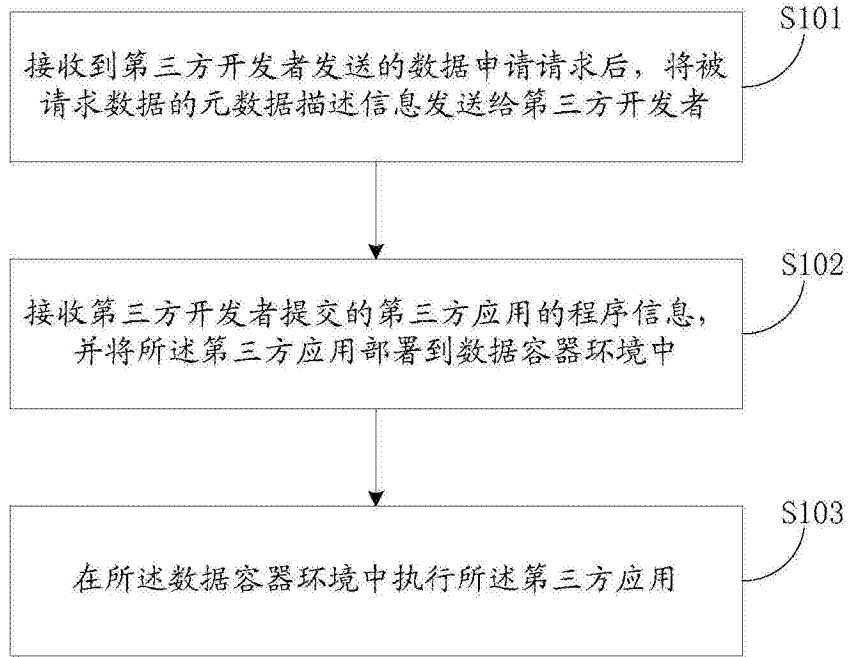


图1

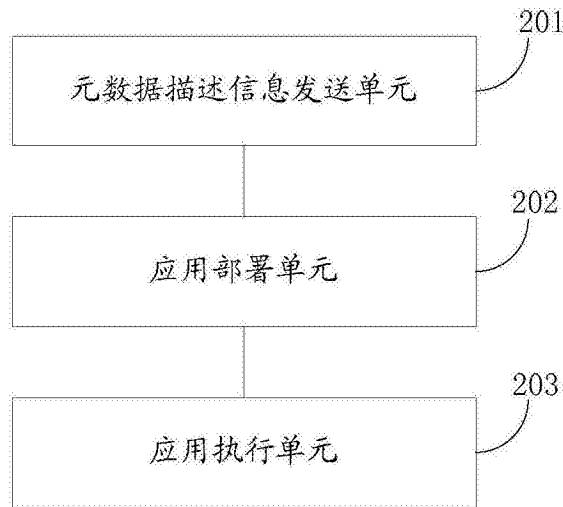


图2