(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114385218 A
(43)申请公布日 2022.04.22

(21)申请号 202111593592.9

(22)申请日 2021.12.22

(71)申请人 东莞理工学院
地址 523808 广东省东莞市松山湖科技产业园区大学路1号

(72)发明人 丁烨 何智杨 廖清

(74)专利代理机构 广州三环专利商标代理有限公司 44202
代理人 郭浩辉 许羽冬

(51)Int.Cl.
G06F 8/70(2018.01)
G06F 8/61(2018.01)
G06F 9/455(2006.01)

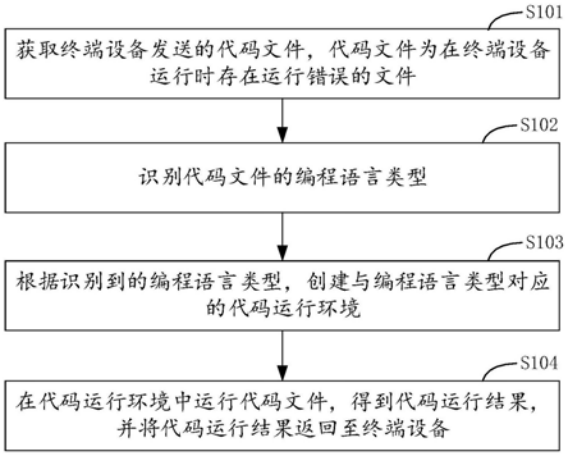
权利要求书2页 说明书8页 附图3页

(54)发明名称

代码运行方法、装置、设备及存储介质

(57)摘要

本申请公开了一种代码运行方法、装置、设备及存储介质，通过在边缘端设备获取所述终端设备发送的代码文件，并识别所述代码文件的编程语言类型，以能够针对在终端设备无法运行代码文件时，将代码文件转移到边缘端设备上运行，从而解决在本地需要配置多个环境而存在操作繁琐的问题；再根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境，以自动化创建适于运行代码文件的代码运行环境；最后在所述代码运行环境中运行所述代码文件，得到代码运行结果，并将所述代码运行结果返回至所述终端设备，从而实现代码文件能够根据代码所需的实际运行环境选择在终端设备或边缘端设备上运行，代码运行方式更加全面、便捷和效率。



CN 114385218 A

1.一种代码运行方法,其特征在于,应用于边缘端设备,所述边缘端设备与终端设备通信连接,所述方法包括:

获取所述终端设备发送的代码文件,所述代码文件为在所述终端设备运行时存在运行错误的文件;

识别所述代码文件的编程语言类型;

根据识别到的编程语言类型,创建与所述编程语言类型对应的代码运行环境;

在所述代码运行环境中运行所述代码文件,得到代码运行结果,并将所述代码运行结果返回至所述终端设备。

2.如权利要求1所述的代码运行方法,其特征在于,所述识别所述代码文件的编程语言类型,包括:

利用预设的机器学习分类模型,提取所述代码文件的代码特征,并根据所述代码特征,对所述代码文件进行识别,得到所述代码文件的编程语言类型。

3.如权利要求2所述的代码运行方法,其特征在于,所述代码特征包括所述代码文件的目标字符、目标令牌和文件扩展名。

4.如权利要求1所述的代码运行方法,其特征在于,所述根据识别到的编程语言类型,创建与所述编程语言类型对应的代码运行环境,包括:

向云端设备发送文件获取请求,所述文件获取请求用于获取与所述编程语言类型对应的镜像文件;

基于所述镜像文件,创建所述与所述编程语言类型对应的代码运行环境,所述代码运行环境包括通信容器和语言环境容器,所述通信容器用于将所述代码文件传输给所述语言环境容器,以及将所述代码运行结果返回至所述终端设备,所述语言环境容器用于运行所述代码文件。

5.如权利要求1至4任一项所述的代码运行方法,其特征在于,所述运行错误包括所述终端设备的代码运行环境配置错误和所述终端设备未配置代码运行环境。

6.一种代码运行方法,其特征在于,应用于终端设备,所述终端设备与边缘端设备通信连接,所述方法包括:

运行代码文件,并监听所述代码文件的运行状态;

若所述代码文件的运行状态为存在运行错误,则显示用于选择是否将所述代码文件发送至所述边缘端设备的用户选项,所述用户选择包括确认选项和取消选项;

若接收到用户触发的确认选项,则将所述代码文件发送至所述边缘端设备,所述代码文件能够被所述边缘端设备识别到的编程语言类型,并创建与所述编程语言类型对应的代码运行环境,以及在所述代码运行环境中运行所述代码文件,得到代码运行结果;

接收所述边缘端设备返回的所述代码运行结果。

7.一种代码运行装置,其特征在于,包括:

获取模块,用于获取终端设备发送的代码文件,所述代码文件为在所述终端设备运行时存在运行错误的文件;

识别模块,用于识别所述代码文件的编程语言类型;

创建模块,用于根据识别到的编程语言类型,创建与所述编程语言类型对应的代码运行环境;

运行模块，用于在所述代码运行环境中运行所述代码文件，得到代码运行结果，并将所述代码运行结果返回至所述终端设备。

8.一种代码运行装置，其特征在于，包括：

监听模块，用于运行代码文件，并监听所述代码文件的运行状态；

显示模块，用于若所述代码文件的运行状态为存在运行错误，则显示用于选择是否将所述代码文件发送至所述边缘端设备的用户选项，所述用户选择包括确认选项和取消选项；

发送模块，用于若接收到用户触发的确认选项，则将所述代码文件发送至所述边缘端设备，所述代码文件能够被所述边缘端设备识别到的编程语言类型，并创建与所述编程语言类型对应的代码运行环境，以及在所述代码运行环境中运行所述代码文件，得到代码运行结果；

接收模块，用于接收所述边缘端设备返回的所述代码运行结果。

9.一种计算机设备，其特征在于，包括处理器和存储器，所述存储器用于存储计算机程序，所述计算机程序被所述处理器执行时实现如权利要求1至5任一项所述的代码运行方法的步骤，或权利要求6所述的代码运行方法的步骤。

10.一种计算机可读存储介质，其特征在于，其存储有计算机程序，所述计算机程序被处理器执行时实现如权利要求1至5任一项所述的代码运行方法的步骤，或权利要求6所述的代码运行方法的步骤。

代码运行方法、装置、设备及存储介质

技术领域

[0001] 本申请涉及计算机技术领域，尤其涉及一种代码运行方法、装置、设备及存储介质。

背景技术

[0002] 随着计算机技术的不断发展，计算机知识的不断普及，计算机的广泛运用，以及许多实用软件使编程学习的门槛降低，其中编程好的代码需要在计算机上运行。

[0003] 目前，运行代码的技术主要有本地运行和云端运行。本地运行是在本地部署好语言环境后，在命令窗口或本地软件中运行代码，例如PyCharm、IntelliJ IDEA等。但是在本地运行代码，需要花费大量时间配置代码运行环境，步骤过于繁琐，同时在更换机器之后需要反复地配置环境，需要运行多种语言的代码时需要在本地配置多个环境。云端运行是完全在云端上运行代码，只要在有网络的环境下，无需配置本地环境，打开云端平台即可对编写好的代码运行，与本地环境无关。但是在云端运行代码，虽然提升编程的便捷性，但过于依赖网络环境，当需要运行对本地文件进行操作的代码，或者存在与其他代码文件相联系的代码时，无法在云端上完成运行。可见，当前代码运行方式只能基于环境在本地运行代码，或者是完全忽略本地环境在云端运行代码。

发明内容

[0004] 本申请提供了一种代码运行方法、装置、设备及存储介质，以解决当前代码运行方式不够便捷等局限性的技术问题。

[0005] 为了解决上述技术问题，本申请实施例提供了一种代码运行方法，应用于边缘端设备，所述边缘端设备与终端设备通信连接，所述方法包括：

[0006] 获取所述终端设备发送的代码文件，所述代码文件为在所述终端设备运行时存在运行错误的文件；

[0007] 识别所述代码文件的编程语言类型；

[0008] 根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境；

[0009] 在所述代码运行环境中运行所述代码文件，得到代码运行结果，并将所述代码运行结果返回至所述终端设备。

[0010] 本实施例通过在边缘端设备获取所述终端设备发送的代码文件，并识别所述代码文件的编程语言类型，以能够针对在终端设备无法运行代码文件时，将代码文件转移到边缘端设备上运行，从而解决在本地需要配置多个环境而存在操作繁琐的问题；再根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境，以自动化创建适于运行代码文件的代码运行环境，无需用户操作，提高便捷性；最后在所述代码运行环境中运行所述代码文件，得到代码运行结果，并将所述代码运行结果返回至所述终端设备，从而实现代码文件能够根据代码所需的实际运行环境选择在终端设备或边缘端设备上运行，代码运行方式更加全面、便捷和效率。

[0011] 在一实施例中，所述识别所述代码文件的编程语言类型，包括：

[0012] 利用预设的机器学习分类模型，提取所述代码文件的代码特征，并根据所述代码特征，对所述代码文件进行识别，得到所述代码文件的编程语言类型。

[0013] 在一实施例中，所述代码特征包括所述代码文件的目标字符、目标令牌和文件扩展名。

[0014] 在一实施例中，所述根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境，包括：

[0015] 向云端设备发送文件获取请求，所述文件获取请求用于获取与所述编程语言类型对应的镜像文件；

[0016] 基于所述镜像文件，创建所述与所述编程语言类型对应的代码运行环境，所述代码运行环境包括通信容器和语言环境容器，所述通信容器用于将所述代码文件传输给所述语言环境容器，以及将所述代码运行结果返回至所述终端设备，所述语言环境容器用于运行所述代码文件。

[0017] 在一实施例中，所述运行错误包括所述终端设备的代码运行环境配置错误和所述终端设备未配置代码运行环境。

[0018] 第二方面，本申请实施例提供另一种代码运行方法，应用于终端设备，所述终端设备与边缘端设备通信连接，所述方法包括：

[0019] 运行代码文件，并监听所述代码文件的运行状态；

[0020] 若所述代码文件的运行状态为存在运行错误，则显示用于选择是否将所述代码文件发送至所述边缘端设备的用户选项，所述用户选择包括确认选项和取消选项；

[0021] 若接收到用户触发的确认选项，则将所述代码文件发送至所述边缘端设备，所述代码文件能够被所述边缘端设备识别到的编程语言类型，并创建与所述编程语言类型对应的代码运行环境，以及在所述代码运行环境中运行所述代码文件，得到代码运行结果；

[0022] 接收所述边缘端设备返回的所述代码运行结果。

[0023] 第三方面，本申请实施例提供一种代码运行装置，包括：

[0024] 获取模块，用于获取终端设备发送的代码文件，所述代码文件为在所述终端设备运行时存在运行错误的文件；

[0025] 识别模块，用于识别所述代码文件的编程语言类型；

[0026] 创建模块，用于根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境；

[0027] 运行模块，用于在所述代码运行环境中运行所述代码文件，得到代码运行结果，并将所述代码运行结果返回至所述终端设备。

[0028] 第四方面，本申请实施例提供另一种代码运行装置，包括：

[0029] 监听模块，用于运行代码文件，并监听所述代码文件的运行状态；

[0030] 显示模块，用于若所述代码文件的运行状态为存在运行错误，则显示用于选择是否将所述代码文件发送至所述边缘端设备的用户选项，所述用户选择包括确认选项和取消选项；

[0031] 发送模块，用于若接收到用户触发的确认选项，则将所述代码文件发送至所述边缘端设备，所述代码文件能够被所述边缘端设备识别到的编程语言类型，并创建与所述编

程语言类型对应的代码运行环境，以及在所述代码运行环境中运行所述代码文件，得到代码运行结果；

[0032] 接收模块，用于接收所述边缘端设备返回的所述代码运行结果。

[0033] 第五方面，本申请实施例提供一种计算机设备，包括处理器和存储器，所述存储器用于存储计算机程序，所述计算机程序被所述处理器执行时实现如第一方面所述的代码运行方法的步骤，或第二方面所述的代码运行方法的步骤。

[0034] 第六方面，本申请实施例提供一种计算机可读存储介质，其存储有计算机程序，所述计算机程序被处理器执行时实现如第一方面所述的代码运行方法的步骤，或第二方面所述的代码运行方法的步骤。

[0035] 需要说明的是，上述第二方面至第四方面的有益效果请参见上述第一方面的相关描述，在此不再赘述。

附图说明

[0036] 图1为本申请一实施例提供的代码运行方法的流程示意图；

[0037] 图2为本申请另一实施例提供的代码运行方法的流程示意图；

[0038] 图3为本申请一实施例提供的代码运行装置的结构示意图；

[0039] 图4为本申请另一实施例提供的代码运行装置的结构示意图；

[0040] 图5为本申请实施例提供的计算机设备的结构示意图。

具体实施方式

[0041] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0042] 如相关技术记载，本地运行是在本地部署好语言环境后，在命令窗口或本地软件中运行代码，例如PyCharm、IntelliJ IDEA等。但是在本地运行代码，需要花费大量时间配置代码运行环境，步骤过于繁琐，同时在更换机器之后需要反复地配置环境，需要运行多种语言的代码时需要在本地配置多个环境。云端运行是完全在云端上运行代码，只要在有网络的环境下，无需配置本地环境，打开云端平台即可对编写好的代码运行，与本地环境无关。但是在云端运行代码，虽然提升编程的便捷性，但过于依赖网络环境，当需要运行对本地文件进行操作的代码，或者存在与其他代码文件相联系的代码时，无法在云端上完成运行。可见，当前代码运行方式只能基于环境在本地运行代码，或者是完全忽略本地环境在云端运行代码。

[0043] 为此，本申请实施例提供一种代码运行方法、装置、设备及存储介质，本实施例通过在边缘端设备获取所述终端设备发送的代码文件，并识别所述代码文件的编程语言类型，以能够针对在终端设备无法运行代码文件时，将代码文件转移到边缘端设备上运行，从而解决在本地需要配置多个环境而存在操作繁琐的问题；再根据识别到的编程语言类型，创建与所述编程语言类型对应的代码运行环境，以自动化创建适于运行代码文件的代码运行环境，无需用户操作，提高便捷性；最后在所述代码运行环境中运行所述代码文件，得到

代码运行结果，并将所述代码运行结果返回至所述终端设备，从而实现代码文件能够根据代码所需的实际运行环境选择在终端设备或边缘端设备上运行，代码运行方式更加全面、便捷和效率。

[0044] 请参照图1，图1为本申请实施例提供的一种代码运行方法的流程示意图。本申请实施例的代码运行方法可应用于边缘端设备，该边缘端设备与终端设备通信连接，该边缘端设备包括但不限于智能手机、平板电脑、笔记本电脑和桌上型计算机等设备，终端设备包括但不限于不同于上述边缘端设备的智能手机、平板电脑、笔记本电脑和桌上型计算机等设备。如图1所示，代码运行方法包括步骤S101至步骤S104，详述如下：

[0045] 步骤S101，获取终端设备发送的代码文件，代码文件为在终端设备运行时存在运行错误的文件。

[0046] 在本步骤中，运行错误包括终端设备的代码运行环境配置错误和终端设备未配置代码运行环境。作为示例而非限定，在一种实施情况下，终端设备配置的是Java语言的代码运行环境，但用户所输入的是C++语言的代码，则在该终端设备上无法运行C++代码，因此将对应的代码文件上传至边缘端设备进行运行。在另一种实施情况下，由于用户属于初学者，用户不能正确配置代码运行环境，导致代码文件的运行结果报错，或者用户配置的代码运行环境崩溃，所以将代码文件上传至边缘端设备进行运行。应理解，上述实施情况仅用作示例，在其他实施方式中可能存在运行错误的其他实施情况，例如终端设备上根本没有配置代码运行环境的情况，在此不再赘述。

[0047] 步骤S102，识别代码文件的编程语言类型。

[0048] 在本步骤中，编程语言类型包括但不限于Java、python、C、C++、C#、rust、go、JavaScript、PHP和Ruby。可选地，通过代码文件的目标位置上特殊字符或者文件扩展名等特征，识别编程语言类型。

[0049] 步骤S103，根据识别到的编程语言类型，创建与编程语言类型对应的代码运行环境。

[0050] 在本步骤中，根据编程语言类型获取对应的环境配置文件，并利用容器创建方法根据环境配置文件，创建代码运行环境。可以理解的是，每种编程语言类型均对应有环境配置文件，该环境配置文件可以是脚本文件，其可能在不需要用户干预的情况下，完成代码运行环境的创建。

[0051] 步骤S104，在代码运行环境中运行代码文件，得到代码运行结果，并将代码运行结果返回至终端设备。

[0052] 在本步骤中，边缘端设备创建代码运行环境后，执行代码文件中的代码，得到代码运行结果，并将代码运行结果返回至终端设备供用户查阅或进一步处理。

[0053] 在一实施例中，在图1所示实施例的基础上，上述步骤S102，包括：

[0054] 利用预设的机器学习分类模型，提取代码文件的代码特征，并根据代码特征，对代码文件进行识别，得到代码文件的编程语言类型。

[0055] 在本实施例中，可选地，代码特征包括代码文件的目标字符、目标令牌和文件扩展名。其中目标字符可以是代码文件的前5个特殊字符以及其他常用特殊字符，例如冒号、花括号和分号等；目标令牌可以是代码文件的前20个令牌。

[0056] 示例性地，采用GitHub平台的OctoLingua机器学习分类模型，该模型能够对

GitHub托管的多种语言进行分类,OctoLingua的训练数据包含不同编程语言类型所表示的相同任务的源码,例如,生成Fibonacci序列的任务可以用C、C++、CoffeeScript、D、Java或Julia等编程语言类型表示。同时以编程方式从GitHub上的公共仓库收集源码,以满足最低资格标准的仓库,例如具有最小数量的分支,以及涵盖目标编程语言类型和涵盖特定文件扩展名,以增加一些额外来源的训练集,从而提高语言覆盖率和性能。对于训练数据,通过表格形式提取以下特征:每个文件的前五个特殊字符、每个文件前20个令牌、文件扩展名以及存在代码文件中常用的某些特殊字符如冒号、花括号和分号,放入分类器中进行训练。

[0057] 在一实施例中,在图1所示实施例的基础上,上述步骤S103,包括:

[0058] 向云端设备发送文件获取请求,文件获取请求用于获取与编程语言类型对应的镜像文件;

[0059] 基于镜像文件,创建与编程语言类型对应的代码运行环境,代码运行环境包括通信容器和语言环境容器,通信容器用于将代码文件传输给语言环境容器,以及将代码运行结果返回至终端设备,语言环境容器用于运行代码文件。

[0060] 在本实施例中,边缘端设备与云端设备通信连接,使得终端设备与云端设备之间无需直接交互,而边缘端设备处理信息的路由距离短,与云端交互速度更快,降低延迟,提高用户编程过程中的体验。

[0061] 示例性地,在识别出代码文件所采用的编程语言类型后,查看边缘端设备是否已经存在该编程语言类型的代码运行环境。若没有此环境,则边缘端设备初次建立代码运行环境时与云端设备进行通信,其中云端设备有镜像仓库和备用文件仓库,镜像仓库中存储了文件索引,边缘端设备从云端设备的镜像仓库中拉取镜像获得文件索引,并通过文件索引在云端设备的备用文件仓库中拉取环境配置文件,根据该环境配置文件在边缘端设备建立通信容器和语言环境容器,通信容器将终端设备本地上传的代码文件发送到对应的语言环境容器中。

[0062] 可选地,通过将本申请实施例的代码运行方法整合为功能插件,将功能插件安装到终端设备或边缘端设备即可,以能够实现在不同设备上的功能复用,进一步提高便捷性。

[0063] 请参照图2,图2为本申请实施例提供的另一种代码运行方法的流程示意图。本申请实施例的代码运行方法可应用于上述终端设备,该终端设备与上述边缘端设备通信连接。如图2所示,代码运行方法包括步骤S201至步骤S204,详述如下:

[0064] 步骤S201,运行代码文件,并监听代码文件的运行状态;

[0065] 步骤S202,若代码文件的运行状态为存在运行错误,则显示用于选择是否将代码文件发送至边缘端设备的用户选项,用户选择包括确认选项和取消选项;

[0066] 步骤S203,若接收到用户触发的确认选项,则将代码文件发送至边缘端设备,代码文件能够被边缘端设备识别到的编程语言类型,并创建与编程语言类型对应的代码运行环境,以及在代码运行环境中运行代码文件,得到代码运行结果;

[0067] 步骤S204,接收边缘端设备返回的代码运行结果。

[0068] 在上述步骤S201至S204中,当终端设备本地运行代码文件时,持续捕获代码控制台的内容,当识别到控制台有错误信息出现时,判断错误类型,若错误类型为环境相关类型(即运行错误),则弹出提示框询问用户是否将代码我呢见上传到边缘端设备。

[0069] 需要说明的是,当运行状态不为运行错误时,即终端设备在代码运行环境配置成

功的情况下，能够在vscode中正常编辑和运行代码，则整个过程都是在终端设备本地完成，而无需使用边缘端设备的代码运行环境。相应地，当需要运行对本地文件进行操作的代码，或者存在与其他代码文件相联系的代码时，可以保留在终端设备本地运行。

[0070] 可以理解的是，图2所示方法实施例的步骤解释可参见上述图1所示方法实施例的相关描述，在此不再赘述。

[0071] 为了执行上述图1实施例对应的代码运行方法，以实现相应的功能和技术效果。参见图3，图3示出了本申请实施例提供的一种代码运行装置的结构框图。为了便于说明，仅示出了与本实施例相关的部分，本申请实施例提供的代码运行装置，包括：

[0072] 获取模块301，用于获取终端设备发送的代码文件，代码文件为在终端设备运行时存在运行错误的文件；

[0073] 识别模块302，用于识别代码文件的编程语言类型；

[0074] 创建模块303，用于根据识别到的编程语言类型，创建与编程语言类型对应的代码运行环境；

[0075] 运行模块304，用于在代码运行环境中运行代码文件，得到代码运行结果，并将代码运行结果返回至终端设备。

[0076] 在一实施例中，识别模块302，具体用于：

[0077] 利用预设的机器学习分类模型，提取代码文件的代码特征，并根据代码特征，对代码文件进行识别，得到代码文件的编程语言类型。

[0078] 在一实施例中，代码特征包括代码文件的目标字符、目标令牌和文件扩展名。

[0079] 在一实施例中，创建模块303，包括：

[0080] 获取单元，用于向云端设备发送文件获取请求，文件获取请求用于获取与编程语言类型对应的镜像文件；

[0081] 创建单元，用于基于镜像文件，创建与编程语言类型对应的代码运行环境，代码运行环境包括通信容器和语言环境容器，通信容器用于将代码文件传输给语言环境容器，以及将代码运行结果返回至终端设备，语言环境容器用于运行代码文件。

[0082] 在一实施例中，运行错误包括终端设备的代码运行环境配置错误和终端设备未配置代码运行环境。

[0083] 为了执行上述图2实施例对应的代码运行方法，以实现相应的功能和技术效果。参见图4，图4示出了本申请实施例提供的一种代码运行装置的结构框图。为了便于说明，仅示出了与本实施例相关的部分，本申请实施例提供的代码运行装置，包括：

[0084] 监听模块401，用于运行代码文件，并监听代码文件的运行状态；

[0085] 显示模块402，用于若代码文件的运行状态为存在运行错误，则显示用于选择是否将代码文件发送至边缘端设备的用户选项，用户选择包括确认选项和取消选项；

[0086] 发送模块403，用于若接收到用户触发的确认选项，则将代码文件发送至边缘端设备，代码文件能够被边缘端设备识别到的编程语言类型，并创建与编程语言类型对应的代码运行环境，以及在代码运行环境中运行代码文件，得到代码运行结果；

[0087] 接收模块404，用于接收边缘端设备返回的代码运行结果。

[0088] 上述的代码运行装置可实施上述方法实施例的代码运行方法。上述方法实施例中的可选项也适用于本实施例，这里不再详述。本申请实施例的其余内容可参照上述方法实

施例的内容，在本实施例中，不再进行赘述。

[0089] 图5为本申请一实施例提供的计算机设备的结构示意图。如图5所示，该实施例的计算机设备5包括：至少一个处理器50(图5中仅示出一个)处理器、存储器51以及存储在所述存储器51中并可在所述至少一个处理器50上运行的计算机程序52，所述处理器50执行所述计算机程序52时实现上述任意方法实施例中的步骤。

[0090] 可以理解的是，当应用于图1方法实施例时，计算机设备为边缘端设备；当应用于图2方法实施例时，计算机设备为终端设备。

[0091] 所述计算机设备5可以是智能手机、平板电脑、桌上型计算机和云端服务器等计算设备。该计算机设备可包括但不仅限于处理器50、存储器51。本领域技术人员可以理解，图5仅仅是计算机设备5的举例，并不构成对计算机设备5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，例如还可以包括输入输出设备、网络接入设备等。

[0092] 所称处理器50可以是中央处理单元(Central Processing Unit，CPU)，该处理器50还可以是其他通用处理器、数字信号处理器(Digital Signal Processor，DSP)、专用集成电路(Application Specific Integrated Circuit，ASIC)、现成可编程门阵列(Field－Programmable Gate Array，FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0093] 所述存储器51在一些实施例中可以是所述计算机设备5的内部存储单元，例如计算机设备5的硬盘或内存。所述存储器51在另一些实施例中也可以是所述计算机设备5的外部存储设备，例如所述计算机设备5上配备的插接式硬盘，智能存储卡(Smart Media Card，SMC)，安全数字(Secure Digital，SD)卡，闪存卡(Flash Card)等。进一步地，所述存储器51还可以既包括所述计算机设备5的内部存储单元也包括外部存储设备。所述存储器51用于存储操作系统、应用程序、引导装载程序(BootLoader)、数据以及其他程序等，例如所述计算机程序的程序代码等。所述存储器51还可以用于暂时地存储已经输出或者将要输出的数据。

[0094] 另外，本申请实施例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存储有计算机程序，所述计算机程序被处理器执行时实现上述任意方法实施例中的步骤。

[0095] 本申请实施例提供了一种计算机程序产品，当计算机程序产品在终端设备上运行时，使得终端设备执行时实现上述各个方法实施例中的步骤。

[0096] 在本申请所提供的几个实施例中，可以理解的是，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意的是，在有些作为替换的实现方式中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。

[0097] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。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或者该技术方案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一个存储介质中，包括若干指令用以使得一台终端设备执行本申请各

个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read－Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0098] 以上所述的具体实施例,对本申请的目的、技术方案和有益效果进行了进一步的详细说明,应当理解,以上所述仅为本申请的具体实施例而已,并不用于限定本申请的保护范围。特别指出,对于本领域技术人员来说,凡在本申请的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

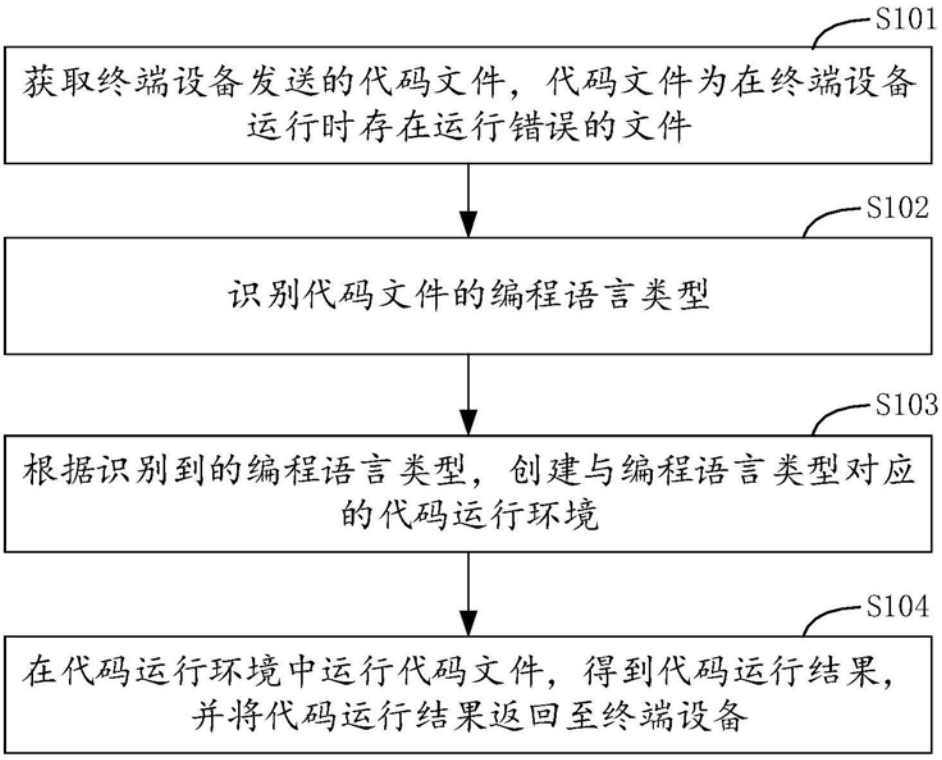

S101
获取终端设备发送的代码文件，代码文件为在终端设备运行时存在运行错误的文件
S102
识别代码文件的编程语言类型
S103
根据识别到的编程语言类型，创建与编程语言类型对应的代码运行环境
S104
在代码运行环境中运行代码文件，得到代码运行结果，并将代码运行结果返回至终端设备

图1

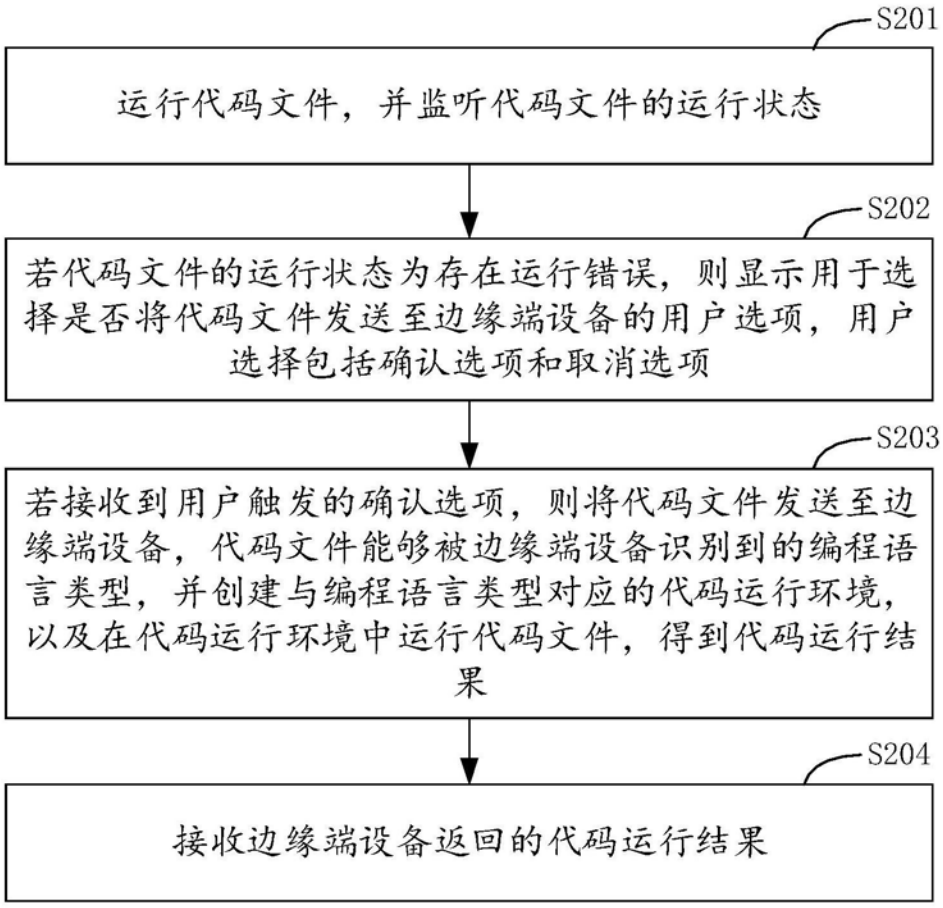

S201
运行代码文件，并监听代码文件的运行状态
S202
若代码文件的运行状态为存在运行错误，则显示用于选择是否将代码文件发送至边缘端设备的用户选项，用户选择包括确认选项和取消选项
S203
若接收到用户触发的确认选项，则将代码文件发送至边缘端设备，代码文件能够被边缘端设备识别到的编程语言类型，并创建与编程语言类型对应的代码运行环境，以及在代码运行环境中运行代码文件，得到代码运行结果
S204
接收边缘端设备返回的代码运行结果

图2

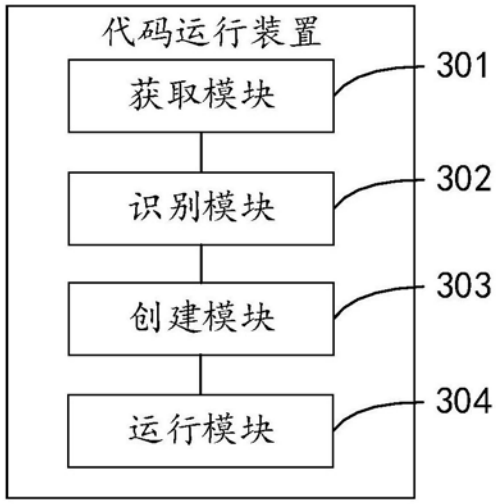

代码运行装置
获取模块
301
识别模块
302
创建模块
303
运行模块
304

图3

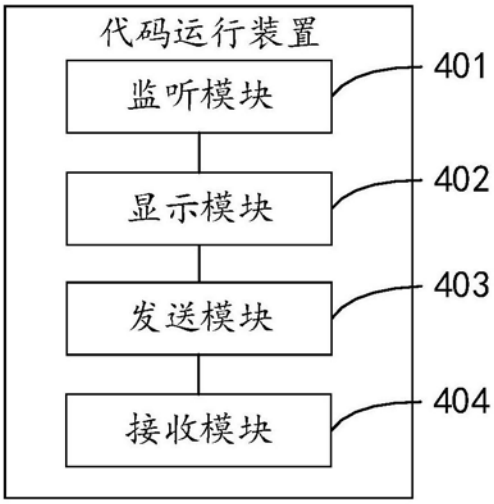

代码运行装置
监听模块
401
显示模块
402
发送模块
403
接收模块
404

图4

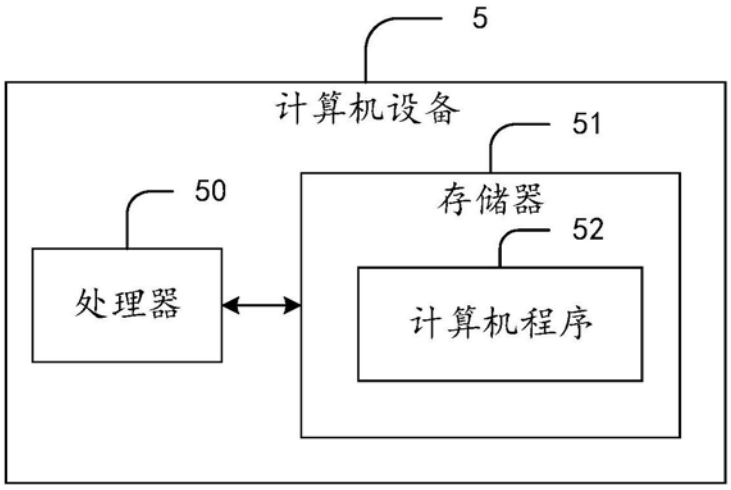

5
计算机设备
51
存储器
50
52
处理器
计算机程序

图5