



(12) 发明专利

(10) 授权公告号 CN 115629772 B

(45) 授权公告日 2023. 09. 19

(21) 申请号 202211095287.1

CN 106325965 A, 2017.01.11

(22) 申请日 2022.09.05

CN 106503570 A, 2017.03.15

(65) 同一申请的已公布的文献号

CN 111176669 A, 2020.05.19

申请公布号 CN 115629772 A

CN 112379828 A, 2021.02.19

(43) 申请公布日 2023.01.20

CN 112965761 A, 2021.06.15

(73) 专利权人 摩尔线程智能科技(北京)有限公司

CN 113703956 A, 2021.11.26

地址 100080 北京市海淀区海淀大街31号2层209

CN 113742716 A, 2021.12.03

CN 114237644 A, 2022.03.25

US 11075931 B1, 2021.07.27

US 2008139191 A1, 2008.06.12

US 2022109605 A1, 2022.04.07

(72) 发明人 杨上山

杨芳; 贺红卫; 谢鹏; 饶京宏. 嵌入式软件目标码仿真测试平台的研究与设计. 计算机工程与设计. 2009, (第19期), 全文.

(74) 专利代理机构 北京林达刘知识产权代理事务所(普通合伙) 11277

专利代理师 刘新宇

程仲汉 等. 一种基于Kubernetes的Web应用部署与配置系统. 《成都信息工程大学学报》. 2021, 第36卷(第05期), 全文.

(51) Int. Cl.

G06F 8/61 (2018.01)

G06F 9/455 (2006.01)

王政. 内网域名系统的安全保密风险研究. 《内网域名系统的安全保密风险研究》. 2020, 第2020年卷(第10期), 全文.

(56) 对比文件

CN 114047925 A, 2022.02.15

US 6448985 B1, 2002.09.10

审查员 王冠威

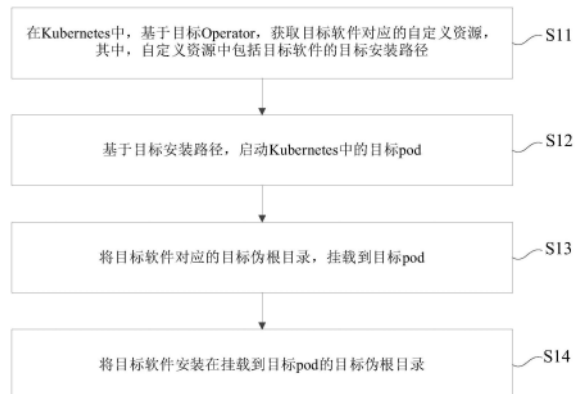
权利要求书1页 说明书10页 附图3页

(54) 发明名称

Kubernetes软件安装方法及装置、电子设备

(57) 摘要

本公开涉及一种Kubernetes软件安装方法及装置、电子设备,所述方法包括:在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,所述自定义资源中包括所述目标软件的目标安装路径;基于所述目标安装路径,启动所述Kubernetes中的目标pod;将所述目标软件对应的目标伪根目录,挂载到所述目标pod;将所述目标软件安装在挂载到所述目标pod的所述目标伪根目录。本公开实施例可以实现目标软件安装过程不会对Host OS进行任何侵入式修改,且无需依赖于任何Linux发行版的类型。



CN 115629772 B

1. 一种Kubernetes软件安装方法,其特征在于,包括:

在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,所述自定义资源中包括所述目标软件的目标安装路径;

基于所述目标安装路径,启动所述Kubernetes中的目标pod;

将所述目标软件对应的目标伪根目录,挂载到所述目标pod,其中,所述目标伪根目录是在Host上创建的,且区别于Host上的系统目录;

将所述目标软件安装在挂载到所述目标pod的所述目标伪根目录;

在所述目标pod中安装完成所述目标软件后,生成目标指示文件,其中,所述目标指示文件用于指示所述目标软件的安装状态为安装完成;

在所述目标Operator检测到所述目标指示文件的情况下,将所述目标伪根目录绑定挂载到对应的目标系统根目录。

2. 根据权利要求1所述的方法,其特征在于,所述目标软件包括:驱动软件、应用软件。

3. 根据权利要求1或2所述的方法,其特征在于,在所述目标软件是驱动软件的情况下,所述方法还包括:

根据所述目标伪根目录,通过执行目标ldconfig命令,将所述驱动软件缓存到ldcache;或,

根据所述目标伪根目录,通过执行insmod命令,将所述驱动软件加载到内核模块。

4. 根据权利要求1或2所述的方法,其特征在于,所述方法还包括:

通过执行umount命令或rmmod命令,卸载所述目标软件。

5. 根据权利要求1或2所述的方法,其特征在于,所述自定义资源包括:CRD或CR。

6. 一种Kubernetes软件安装装置,其特征在于,包括:

获取模块,用于在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,所述自定义资源中包括所述目标软件的目标安装路径;

启动模块,用于基于所述目标安装路径,启动所述Kubernetes中的目标pod;

第一挂载模块,用于将所述目标软件对应的目标伪根目录,挂载到所述目标pod,其中,所述目标伪根目录是在Host上创建的,且区别于Host上的系统目录;

安装模块,用于将所述目标软件安装在挂载到所述目标pod的所述目标伪根目录;

生成模块,用于在所述目标pod中安装完成所述目标软件后,生成目标指示文件,其中,所述目标指示文件用于指示所述目标软件的安装状态为安装完成;

第二挂载模块,用于在所述目标Operator检测到所述目标指示文件的情况下,将所述目标伪根目录绑定挂载到对应的目标系统根目录。

7. 一种电子设备,其特征在于,包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器被配置为调用所述存储器存储的指令,以执行权利要求1至5中任意一项所述的方法。

8. 一种计算机可读存储介质,其上存储有计算机程序指令,其特征在于,所述计算机程序指令被处理器执行时实现权利要求1至5中任意一项所述的方法。

Kubernetes软件安装方法及装置、电子设备

技术领域

[0001] 本公开涉及计算机技术领域,尤其涉及一种Kubernetes软件安装方法及装置、电子设备。

背景技术

[0002] 在Kubernetes中安装软件时,Kubernetes集群中的主机节点首先需要通过apt、yum、apk等包管理工具,或者RUN文件来安装节点上所需要的驱动、软件等;之后才可以通过kubeadm、kubectl等工具将其转换为Kubernetes的节点。这种安装方式需要依赖于Linux发行版的类型,且还会对主机操作系统(Host OS)造成影响,因此,亟需一种无特定系统依赖且不会对Host OS进行任何侵入式修改的Kubernetes软件安装方法。

发明内容

[0003] 本公开提出了一种Kubernetes软件安装方法及装置、电子设备的技术方案。

[0004] 根据本公开的一方面,提供了一种Kubernetes软件安装方法,包括:在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,所述自定义资源中包括所述目标软件的目标安装路径;基于所述目标安装路径,启动所述Kubernetes中的目标pod;将所述目标软件对应的目标伪根目录,挂载到所述目标pod;将所述目标软件安装在挂载到所述目标pod的所述目标伪根目录。

[0005] 在一种可能的实现方式中,所述方法还包括:在所述目标pod中安装完成所述目标软件后,生成目标指示文件,其中,所述目标指示文件用于指示所述目标软件的安装状态为安装完成。

[0006] 在一种可能的实现方式中,所述方法还包括:在所述目标Operator检测到所述目标指示文件的情况下,将所述目标伪根目录绑定挂载到对应的目标系统根目录。

[0007] 在一种可能的实现方式中,所述目标软件包括:驱动软件、应用软件。

[0008] 在一种可能的实现方式中,在所述目标软件是驱动软件的情况下,所述方法还包括:根据所述目标伪根目录,通过执行目标ldconfig命令,将所述驱动软件缓存到ldcache;或,根据所述目标伪根目录,通过执行insmod命令,将所述驱动软件加载到内核模块。

[0009] 在一种可能的实现方式中,所述方法还包括:通过执行umount命令或rmmod命令,卸载所述目标软件。

[0010] 在一种可能的实现方式中,所述自定义资源包括:CRD或CR。

[0011] 根据本公开的一方面,提供了一种Kubernetes软件安装装置,包括:获取模块,用于在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,所述自定义资源中包括所述目标软件的目标安装路径;启动模块,用于基于所述目标安装路径,启动所述Kubernetes中的目标pod;第一挂载模块,用于将所述目标软件对应的目标伪根目录,挂载到所述目标pod;安装模块,用于将所述目标软件安装在挂载到所述目标pod的所述目标伪根目录。

[0012] 根据本公开的一方面,提供了一种电子设备,包括:处理器;用于存储处理器可执行指令的存储器;其中,所述处理器被配置为调用所述存储器存储的指令,以执行上述方法。

[0013] 根据本公开的一方面,提供了一种计算机可读存储介质,其上存储有计算机程序指令,所述计算机程序指令被处理器执行时实现上述方法。

[0014] 在本公开实施例中,在Kubernetes中,基于目标Operator获取目标软件对应的自定义资源,基于自定义资源中包括的目标软件的目标安装路径,启动Kubernetes中的目标pod,进而将目标软件对应的目标伪根目录挂载到目标pod,将目标软件安装在挂载到目标pod的目标伪根目录。由于目标伪根目录是在Host中创建出来的,而并非Host OS中的目录,因此,目标软件安装过程不会对Host OS进行任何侵入式修改;此外,由于云原生场景下Kubernetes中的目标pod已经预置有安装环境,以使得目标软件的目标安装包仅需适配目标pod的安装环境,即可在目标pod中实现目标软件的容器化安装,无需依赖于任何Linux发行版的类型。

[0015] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,而非限制本公开。根据下面参考附图对示例性实施例的详细说明,本公开的其它特征及方面将变得清楚。

附图说明

[0016] 此处的附图被并入说明书中并构成本说明书的一部分,这些附图示出了符合本公开的实施例,并与说明书一起用于说明本公开的技术方案。

[0017] 图1示出根据本公开实施例的一种Kubernetes软件安装方法的流程图;

[0018] 图2示出根据本公开实施例的一种Kubernetes软件安装装置的框图;

[0019] 图3示出根据本公开实施例的一种电子设备的框图;

[0020] 图4示出根据本公开实施例的一种电子设备的框图。

具体实施方式

[0021] 以下将参考附图详细说明本公开的各种示例性实施例、特征和方面。附图中相同的附图标记表示功能相同或相似的元件。尽管在附图中示出了实施例的各种方面,但是除非特别指出,不必按比例绘制附图。

[0022] 在这里专用的词“示例性”意为“用作例子、实施例或说明性”。这里作为“示例性”所说明的任何实施例不必解释为优于或好于其它实施例。

[0023] 本文中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本文中术语“至少一种”表示多种中的任意一种或多种中的至少两种的任意组合,例如,包括A、B、C中的至少一种,可以表示包括从A、B和C构成的集合中选择的任意一个或多个元素。

[0024] 另外,为了更好地说明本公开,在下文的具体实施方式中给出了众多的具体细节。本领域技术人员应当理解,没有某些具体细节,本公开同样可以实施。在一些实例中,对于本领域技术人员熟知的方法、手段、元件和电路未作详细描述,以便于凸显本公开的主旨。

[0025] 对于不同的Linux发行版来说,包管理工具有很多区别。例如,centos一般使用yum

作为包管理工具;debian based OS使用apt作为包管理工具。相关技术中进行驱动或应用程序等软件开发时,需要考虑提供多种类型的软件安装包,以确保能够适配不同的Linux发行版,导致软件开发效率较低。

[0026] 此外,相关技术中进行驱动或应用程序等软件安装时,会对Host OS进行侵入式修改,即软件安装包中的内容会直接写入到Host上的对应目录(例如,/etc/lib)。如果软件安装包中有与系统目录中相同文件名的文件时,默认会生成一些.origin的文件,覆盖掉系统目录中原来的文件,对Host OS产生影响。

[0027] 相关技术中在进行驱动或应用程序等软件的开发过程中,可能会存在软件安装包的打包漏洞,导致后续需要进行软件卸载时,出现无法卸载或者卸载不干净等问题,从而在下次再尝试安装该软件时会产生错误导致安装失败。

[0028] 此外,DevOps场景下会经常需要QA/Ops重装主机,以完成Kubernetes节点中的软件安装,相关技术中的软件安装方式,需要人工执行很多重复的安装操作,导致用户使用体检较差且安装效率较低。

[0029] 为了解决上述技术问题,本公开提供了一种Kubernetes软件安装方法,在云原生场景下,通过在Kubernetes的目标pod中进行目标软件的容器化安装,以使得软件安装过程无需依赖于任何特定Linux发行版,且安装过程不会对Host OS进行任何侵入式修改,以使得安装好的目标软件不会对Host造成影响。下文对本公开实施例提供的一种Kubernetes软件安装方法进行详细描述。

[0030] 图1示出根据本公开实施例的一种Kubernetes软件安装方法的流程图。该方法应用于云原生场景下的Kubernetes。如图1所示,该方法包括:

[0031] 在步骤S11中,在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,自定义资源中包括目标软件的目标安装路径。

[0032] 为了在Kubernetes进行软件安装,首先需要创建目标软件对应的自定义资源,其中,自定义资源中包括目标软件的软件安装包的基本信息,例如,目标软件的目标安装路径。

[0033] 将目标软件对应的自定义资源应用到Kubernetes之后,基于目标Operator,可以获取到自定义资源中包括的目标软件包的目标安装路径。

[0034] 在一种可能的实现方式中,目标软件包括:驱动软件、应用软件。

[0035] 本公开实施例提供的Kubernetes软件安装方法既可以应用于驱动软件的安装,还可以应用于应用软件的安装,本公开对此不作具体限定。

[0036] 一种可能的实现方式中,自定义资源包括:CRD (Custom Resource Define) 或CR (Custom Resource)。

[0037] 其中,CR是目标软件对应的自定义资源。由于CR是自定义资源,可能无法被Kubernetes识别,因此,可以进一步创建CR的描述文件CRD,以使得Kubernetes可以基于CRD识别到CR。

[0038] 在步骤S12中,基于目标安装路径,启动Kubernetes中的目标pod。

[0039] 在基于自定义资源获取到目标安装包的目标安装路径之后,可以基于目标安装路径,启动目标安装路径指示的目标pod,进而在目标pod中进行后续目标软件的容器化安装。

[0040] 在步骤S13中,将目标软件对应的目标伪根目录,挂载到目标pod。

[0041] 目标pod的描述文件spec (Specification) 中, 会指示Host上创建的目标伪根目录 (fakeroot) 例如, /app/1/lib, 作为目标软件对应的软件安装包的安装根目录。因此, 为了实现在目标pod中进行目标软件的容器化安装, 将目标软件对应的目标伪根目录, 挂载到目标pod。

[0042] 在步骤S14中, 将目标软件安装在挂载到目标pod的目标伪根目录。

[0043] 在目标软件对应的目标伪根目录挂载到目标pod之后, 目标pod通过安装工具将目标软件安装在目标伪根目录, 即将目标软件对应的软件安装包写入目标伪根目录。由于目标伪根目录虽然在Host上, 但是并非Host上的系统目录, 因此, 目标软件对应的软件安装包写入目标伪根目录, 不会对Host OS产生影响。在需要卸载目标软件时, 仅需删除目标伪根目录即可, 不会出现无法卸载或者卸载不干净的问题。

[0044] 在本公开实施例中, 在Kubernetes中, 基于目标Operator获取目标软件对应的自定义资源, 基于自定义资源中包括的目标软件的目标安装路径, 启动Kubernetes中的目标pod, 进而将目标软件对应的目标伪根目录挂载到目标pod, 将目标软件安装在挂载到目标pod的目标伪根目录。由于目标伪根目录是在Host中创建出来的, 而并非Host OS中的目录, 因此, 目标软件安装过程不会对Host OS进行任何侵入式修改; 此外, 由于云原生场景下Kubernetes中的目标pod已经预置有安装环境, 以使得目标软件的目标安装包仅需适配目标pod的安装环境, 即可在目标pod中实现目标软件的容器化安装, 无需依赖于任何Linux发行版的类型。

[0045] 其中, Kubernetes中目标pod的预置安装环境, 可以是任意一个Linux发行版, 本公开对此不作具体限定。

[0046] 由于目标软件在目标pod中进行了容器化安装, 容器化的目标软件后续可以应用于任何Linux发行版, 也就是说, 目标软件的安装过程不再受限于任何Linux发行版的类型, 但是安装之后仍然可以应用于任何Linux发行版。

[0047] 在一种可能的实现方式中, 该方法还包括: 在目标pod中安装完成目标软件后, 生成目标指示文件, 其中, 目标指示文件用于指示目标软件的安装状态为安装完成。

[0048] 在目标pod中安装完成目标软件后, 通过生成目标指示文件, 以指示目标软件的安装状态为安装完成。

[0049] 在一种可能的实现方式中, 该方法还包括: 在目标Operator检测到目标指示文件的情况下, 将目标伪根目录绑定挂载到对应的目标系统根目录。

[0050] 目标Operator侧存在一个DaemonSet, 用于对目标pod进行监控, 在检测到目标pod中生成指示目标软件的安装状态为安装完成的目标指示文件之后, 可以使用绑定挂载 (bind mount) 命令, 将目标伪根目录绑定挂载到对应的目标系统根目录。

[0051] 在一示例中, 目标软件安装在目标伪根目录/app/1/lib, 则绑定挂载过程为mount bind/app/1/lib/lib, 其中, /lib为目标系统根目录。

[0052] 在一种可能的实现方式中, 在目标软件是驱动软件的情况下, 该方法还包括: 根据目标伪根目录, 通过执行目标ldconfig命令, 将驱动软件缓存到ldcache; 或, 根据目标伪根目录, 通过执行insmod命令, 将驱动软件加载到内核模块。

[0053] 在目标pod中实现驱动软件的容器化安装之后, 为了使得驱动软件能够被主机系统所识别并共享, 可以通过执行目标ldconfig命令 (动态链接库管理命令), 刷新ldcache

(动态链接库缓存),或者,通过执行insmod命令,将驱动软件加载到内核模块(kernel module)。

[0054] 在一种可能的实现方式中,该方法还包括:通过执行umount命令或rmmod命令,卸载目标软件。

[0055] 在执行目标ldconfig命令刷新ldcache,将驱动软件缓存到ldcache之后,可以根据实际运行需要,通过执行umount命令,以使得主机系统无法识别到驱动软件,完成卸载。

[0056] 在执行insmod命令,将驱动软件加载到kernel module之后,可以根据实际运行需要,通过执行rmmod命令,完成驱动软件的卸载。

[0057] 本公开实施例提供的Kubernetes软件安装方法应用到DevOps场景下,可以基于实际情况需要,采用容器化安装方式快速实现软件安装,以确保业务正常运行。

[0058] 在本公开实施例中,在Kubernetes中,基于目标Operator获取目标软件对应的自定义资源,基于自定义资源中包括的目标软件的目标安装路径,启动Kubernetes中的目标pod,进而将目标软件对应的目标伪根目录挂载到目标pod,将目标软件安装在挂载到目标pod的目标伪根目录。由于目标伪根目录是在Host中创建出来的,而并非Host OS中的目录,因此,目标软件安装过程不会对Host OS进行任何侵入式修改;此外,由于云原生场景下Kubernetes中的目标pod已经预置有安装环境,以使得目标软件的目标安装包仅需适配目标pod的安装环境,即可在目标pod中实现目标软件的容器化安装,无需依赖于任何Linux发行版的类型。

[0059] 可以理解,本公开提及的上述各个方法实施例,在不违背原理逻辑的情况下,均可以彼此相互结合形成结合后的实施例,限于篇幅,本公开不再赘述。本领域技术人员可以理解,在具体实施方式的上述方法中,各步骤的具体执行顺序应当以其功能和可能的内在逻辑确定。

[0060] 此外,本公开还提供了一种Kubernetes软件安装装置、电子设备、计算机可读存储介质、程序,上述均可用来实现本公开提供的任一种Kubernetes软件安装方法,相应技术方案和描述和参见方法部分的相应记载,不再赘述。

[0061] 图2示出根据本公开实施例的一种Kubernetes软件安装装置的框图。如图2所示,装置20包括:

[0062] 获取模块21,用于在Kubernetes中,基于目标Operator,获取目标软件对应的自定义资源,其中,自定义资源中包括目标软件的目标安装路径;

[0063] 启动模块22,用于基于目标安装路径,启动Kubernetes中的目标pod;

[0064] 第一挂载模块23,用于将目标软件对应的目标伪根目录,挂载到目标pod;

[0065] 安装模块24,用于将目标软件安装在挂载到目标pod的目标伪根目录。

[0066] 在一种可能的实现方式中,装置20还包括:

[0067] 生成模块,用于在目标pod中安装完成目标软件后,生成目标指示文件,其中,目标指示文件用于指示目标软件的安装状态为安装完成。

[0068] 在一种可能的实现方式中,装置20还包括:

[0069] 第二挂载模块,用于在目标Operator检测到目标指示文件的情况下,将目标伪根目录绑定挂载到对应的目标系统根目录。

[0070] 在一种可能的实现方式中,目标软件包括:驱动软件、应用软件。

- [0071] 在一种可能的实现方式中,在目标软件是驱动软件的情况下,装置20还包括:
- [0072] 缓存更新模块,用于根据目标伪根目录,通过执行目标ldconfig命令,将驱动软件缓存到ldcache;或,
- [0073] 加载模块,用于根据目标伪根目录,通过执行insmod命令,将驱动软件加载到内核模块。
- [0074] 在一种可能的实现方式中,装置20还包括:
- [0075] 卸载模块,用于通过执行umount命令或rmmod命令,卸载目标软件。
- [0076] 在一种可能的实现方式中,自定义资源包括:CRD或CR。
- [0077] 该方法与计算机系统的内部结构存在特定技术关联,且能够解决如何提升硬件运算效率或执行效果的技术问题(包括减少数据存储量、减少数据传输量、提高硬件处理速度等),从而获得符合自然规律的计算机系统内部性能改进的技术效果。
- [0078] 在一些实施例中,本公开实施例提供的装置具有的功能或包含的模块可以用于执行上文方法实施例描述的方法,其具体实现可以参照上文方法实施例的描述,为了简洁,这里不再赘述。
- [0079] 本公开实施例还提出一种计算机可读存储介质,其上存储有计算机程序指令,所述计算机程序指令被处理器执行时实现上述方法。计算机可读存储介质可以是易失性或非易失性计算机可读存储介质。
- [0080] 本公开实施例还提出一种电子设备,包括:处理器;用于存储处理器可执行指令的存储器;其中,所述处理器被配置为调用所述存储器存储的指令,以执行上述方法。
- [0081] 本公开实施例还提供了一种计算机程序产品,包括计算机可读代码,或者承载有计算机可读代码的非易失性计算机可读存储介质,当所述计算机可读代码在电子设备的处理器中运行时,所述电子设备中的处理器执行上述方法。
- [0082] 电子设备可以被提供为终端、服务器或其它形态的设备。
- [0083] 图3示出根据本公开实施例的一种电子设备的框图。参照图3,电子设备800可以是用户设备(User Equipment,UE)、移动设备、用户终端、终端、蜂窝电话、无绳电话、个人数字处理(Personal Digital Assistant,PDA)、手持设备、计算设备、车载设备、可穿戴设备等终端设备。
- [0084] 参照图3,电子设备800可以包括以下一个或多个组件:处理组件802,存储器804,电源组件806,多媒体组件808,音频组件810,输入/输出(I/O)接口812,传感器组件814,以及通信组件816。
- [0085] 处理组件802通常控制电子设备800的整体操作,诸如与显示,电话呼叫,数据通信,相机操作和记录操作相关联的操作。处理组件802可以包括一个或多个处理器820来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件802可以包括一个或多个模块,便于处理组件802和其他组件之间的交互。例如,处理组件802可以包括多媒体模块,以方便多媒体组件808和处理组件802之间的交互。
- [0086] 存储器804被配置为存储各种类型的数据以支持在电子设备800的操作。这些数据的示例包括用于在电子设备800上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器804可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦

除可编程只读存储器 (EPROM), 可编程只读存储器 (PROM), 只读存储器 (ROM), 磁存储器, 快闪存储器, 磁盘或光盘。

[0087] 电源组件806为电子设备800的各种组件提供电力。电源组件806可以包括电源管理系统, 一个或多个电源, 及其他与为电子设备800生成、管理和分配电力相关联的组件。

[0088] 多媒体组件808包括在所述电子设备800和用户之间的提供一个输出接口的屏幕。在一些实施例中, 屏幕可以包括液晶显示器 (LCD) 和触摸面板 (TP)。如果屏幕包括触摸面板, 屏幕可以被实现为触摸屏, 以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界, 而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中, 多媒体组件808包括一个前置摄像头和/或后置摄像头。当电子设备800处于操作模式, 如拍摄模式或视频模式时, 前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0089] 音频组件810被配置为输出和/或输入音频信号。例如, 音频组件810包括一个麦克风 (MIC), 当电子设备800处于操作模式, 如呼叫模式、记录模式和语音识别模式时, 麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器804或经由通信组件816发送。在一些实施例中, 音频组件810还包括一个扬声器, 用于输出音频信号。

[0090] I/O接口812为处理组件802和外围接口模块之间提供接口, 上述外围接口模块可以是键盘, 点击轮, 按钮等。这些按钮可包括但不限于: 主页按钮、音量按钮、启动按钮和锁定按钮。

[0091] 传感器组件814包括一个或多个传感器, 用于为电子设备800提供各个方面的状态评估。例如, 传感器组件814可以检测到电子设备800的打开/关闭状态, 组件的相对定位, 例如所述组件为电子设备800的显示器和小键盘, 传感器组件814还可以检测电子设备800或电子设备800一个组件的位置改变, 用户与电子设备800接触的存在或不存在, 电子设备800方位或加速/减速和电子设备800的温度变化。传感器组件814可以包括接近传感器, 被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件814还可以包括光传感器, 如互补金属氧化物半导体 (CMOS) 或电荷耦合装置 (CCD) 图像传感器, 用于在成像应用中使用。在一些实施例中, 该传感器组件814还可以包括加速度传感器, 陀螺仪传感器, 磁传感器, 压力传感器或温度传感器。

[0092] 通信组件816被配置为便于电子设备800和其他设备之间有线或无线方式的通信。电子设备800可以接入基于通信标准的无线网络, 如无线网络 (Wi-Fi)、第二代移动通信技术 (2G)、第三代移动通信技术 (3G)、第四代移动通信技术 (4G)、通用移动通信技术的长期演进 (LTE)、第五代移动通信技术 (5G) 或它们的组合。在一个示例性实施例中, 通信组件816经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中, 所述通信组件816还包括近场通信 (NFC) 模块, 以促进短程通信。例如, 在NFC模块可基于射频识别 (RFID) 技术, 红外数据协会 (IrDA) 技术, 超宽带 (UWB) 技术, 蓝牙 (BT) 技术和其他技术来实现。

[0093] 在示例性实施例中, 电子设备800可以被一个或多个应用专用集成电路 (ASIC)、数字信号处理器 (DSP)、数字信号处理设备 (DSPD)、可编程逻辑器件 (PLD)、现场可编程门阵列 (FPGA)、控制器、微控制器、微处理器或其他电子元件实现, 用于执行上述方法。

[0094] 在示例性实施例中,还提供了一种非易失性计算机可读存储介质,例如包括计算机程序指令的存储器804,上述计算机程序指令可由电子设备800的处理器820执行以完成上述方法。

[0095] 图4示出根据本公开实施例的一种电子设备的框图。参照图4,电子设备1900可以被提供为一服务器或终端设备。参照图4,电子设备1900包括处理组件1922,其进一步包括一个或多个处理器,以及由存储器1932所代表的存储器资源,用于存储可由处理组件1922的执行的指令,例如应用程序。存储器1932中存储的应用程序可以包括一个或一个以上的每一个对应于一组指令的模块。此外,处理组件1922被配置为执行指令,以执行上述方法。

[0096] 电子设备1900还可以包括一个电源组件1926被配置为执行电子设备1900的电源管理,一个有线或无线网络接口1950被配置为将电子设备1900连接到网络,和一个输入输出(I/O)接口1958。电子设备1900可以操作基于存储在存储器1932的操作系统,例如微软服务器操作系统(Windows Server™),苹果公司推出的基于图形用户界面操作系统(Mac OS X™),多用户多进程的计算机操作系统(Unix™),自由和开放原代码的类Unix操作系统(Linux™),开放原代码的类Unix操作系统(FreeBSD™)或类似。

[0097] 在示例性实施例中,还提供了一种非易失性计算机可读存储介质,例如包括计算机程序指令的存储器1932,上述计算机程序指令可由电子设备1900的处理组件1922执行以完成上述方法。

[0098] 本公开可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质,其上载有用于使处理器实现本公开的各个方面的计算机可读程序指令。

[0099] 计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是(但不限于)电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身,诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0100] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备,或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令,并转发该计算机可读程序指令,以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0101] 用于执行本公开操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等,以及常规的过程式编程语言—诸如“C”语言或类似的编程语言。计算机

可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中,通过利用计算机可读程序指令的状态信息来个性化定制电子电路,例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA),该电子电路可以执行计算机可读程序指令,从而实现本公开的各个方面。

[0102] 这里参照根据本公开实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本公开的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0103] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作,从而,存储有指令的计算机可读介质则包括一个制品,其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0104] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0105] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令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。在有些作为替换的实现中,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。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0106] 该计算机程序产品可以具体通过硬件、软件或其结合的方式实现。在一个可选实施例中,所述计算机程序产品具体体现为计算机存储介质,在另一个可选实施例中,计算机程序产品具体体现为软件产品,例如软件开发包(Software Development Kit, SDK)等等。

[0107] 上文对各个实施例的描述倾向于强调各个实施例之间的不同之处,其相同或相似之处可以互相参考,为了简洁,本文不再赘述。

[0108] 本领域技术人员可以理解,在具体实施方式的上述方法中,各步骤的撰写顺序并不意味着严格的执行顺序而对实施过程构成任何限定,各步骤的具体执行顺序应当以其功能和可能的内在逻辑确定。

[0109] 若本申请技术方案涉及个人信息,应用本申请技术方案的产品在处理个人信息

前,已明确告知个人信息处理规则,并取得个人自主同意。若本申请技术方案涉及敏感个人信息,应用本申请技术方案的产品在处理敏感个人信息前,已取得个人单独同意,并且同时满足“明示同意”的要求。例如,在摄像头等个人信息采集装置处,设置明确显著的标识告知已进入个人信息采集范围,将会对个人信息进行采集,若个人自愿进入采集范围即视为同意对其个人信息进行采集;或者在个人信息处理的装置上,利用明显的标识/信息告知个人信息处理规则的情况下,通过弹窗信息或请个人自行上传其个人信息等方式获得个人授权;其中,个人信息处理规则可包括个人信息处理者、个人信息处理目的、处理方式以及处理的个人信息种类等信息。

[0110] 以上已经描述了本公开的各实施例,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施例的原理、实际应用或对市场中的技术的改进,或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

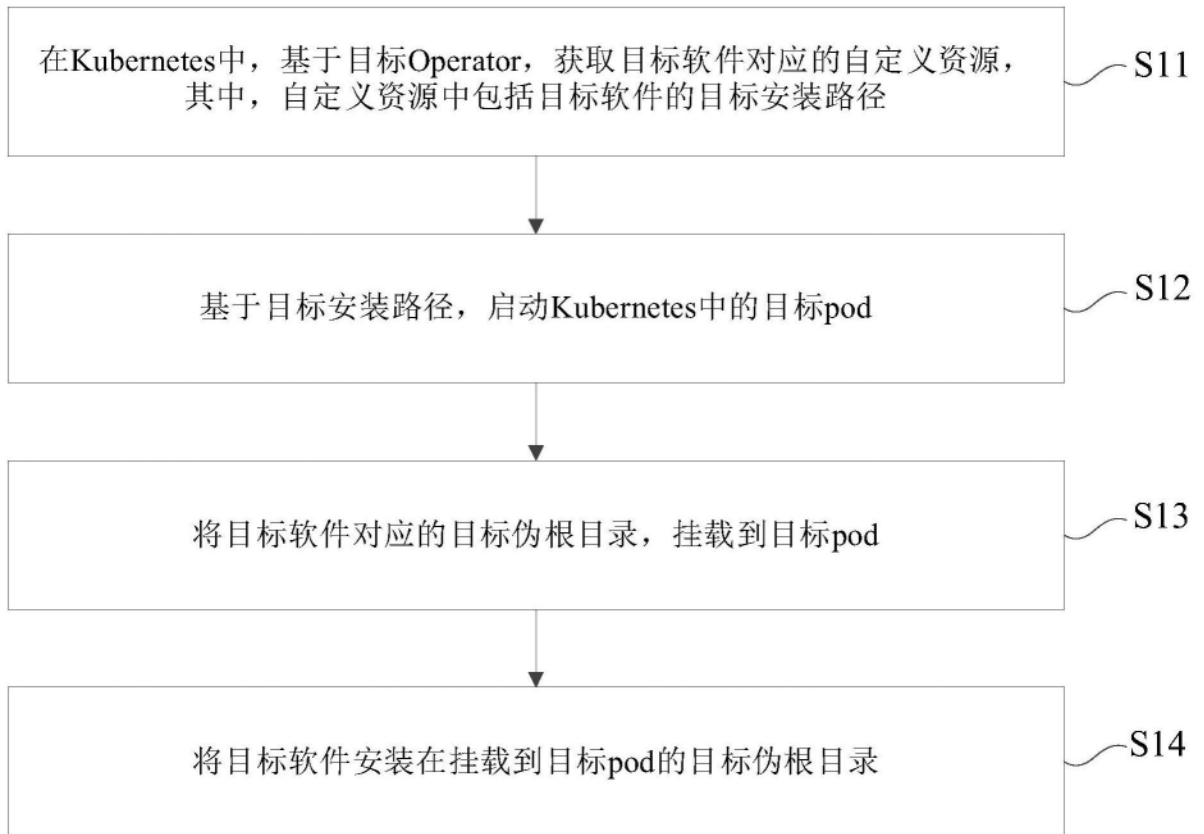


图1

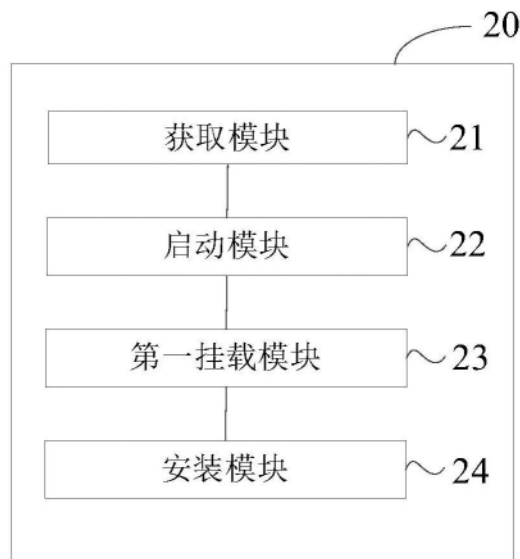


图2

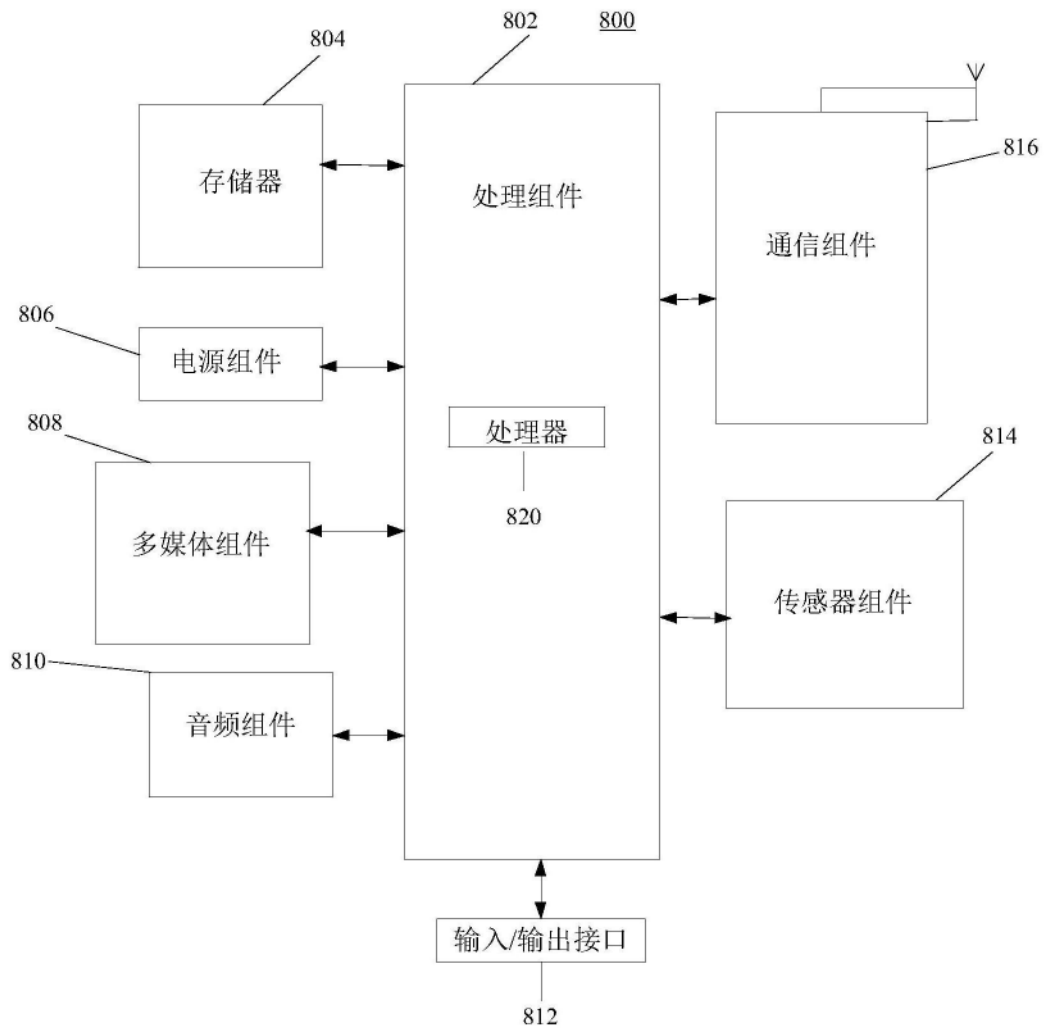


图3

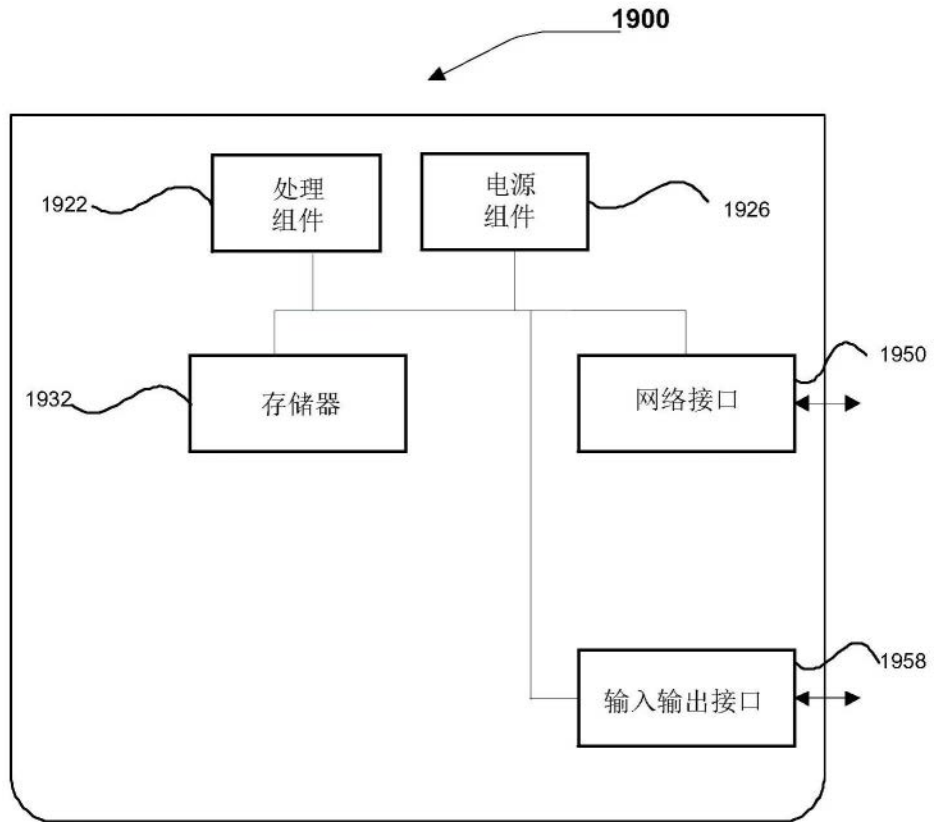


图4