



(12) 发明专利

(10) 授权公告号 CN 111045708 B

(45) 授权公告日 2022.03.11

(21) 申请号 201911330765.0

(22) 申请日 2019.12.20

(65) 同一申请的已公布的文献号  
申请公布号 CN 111045708 A

(43) 申请公布日 2020.04.21

(73) 专利权人 新华三大数据技术有限公司  
地址 450000 河南省郑州市郑州高新技术  
产业开发区杜英街166号总部大观B18  
号楼

(72) 发明人 王明辉

(74) 专利代理机构 北京超成律师事务所 11646  
代理人 孔默

(51) Int. Cl.  
G06F 8/65 (2018.01)  
G06F 9/455 (2006.01)

(56) 对比文件

CN 105573780 A, 2016.05.11

CN 108880898 A, 2018.11.23

US 2019188135 A1, 2019.06.20

US 10007509 B1, 2018.06.26

Sriram Srinivasan等.A Shared-Memory Parallel Algorithm for Updating Single-Source Shortest Paths in Large Dynamic Networks.《2018 IEEE 25th International Conference on High Performance Computing (HiPC)》. IEEE, 2019, 第245-254页.

黄辉.大型仪器共享平台中仪器共享系统的设计与实现.《中国优秀硕士学位论文全文数据库(电子期刊)》.2012, (第1期), 第1138-252页.

审查员 李中兴

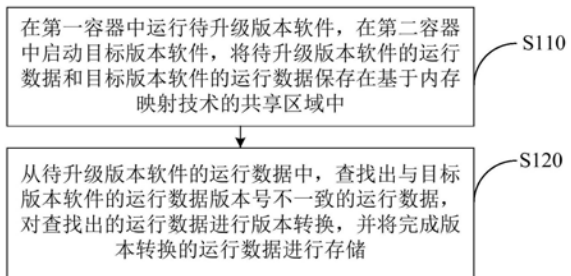
权利要求书2页 说明书9页 附图2页

(54) 发明名称

软件升级方法、电子设备和计算机可读存储介质

(57) 摘要

本发明实施例提出一种软件升级方法、电子设备和计算机可读存储介质,方法应用于电子设备,通过在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,以使待升级版本软件和目标版本软件能够访问到共享区域中的运行数据。从待升级版本软件的运行数据中查找出与目标版本软件的运行数据版本号不一致的运行数据,进而对查找出的运行数据进行版本转换,并将完成版本转换的运行数据进行存储即可实现软件升级,适用性较强,实现较为便捷。



1. 一种软件升级方法,应用于电子设备,其特征在于,所述电子设备包括用于存储待升级版本软件的第一容器和用于存储目标版本软件的第二容器,所述方法包括:

在所述第一容器中运行所述待升级版本软件,在所述第二容器中启动所述目标版本软件,将所述待升级版本软件的运行数据和所述目标版本软件的运行数据保存在基于内存映射技术的共享区域中,以使所述待升级版本软件和所述目标版本软件能够访问到所述共享区域中的运行数据;

从所述待升级版本软件的运行数据中,查找出与所述目标版本软件的运行数据版本号不一致的运行数据,对查找出的所述运行数据进行版本转换,并将完成版本转换的运行数据进行存储。

2. 根据权利要求1所述的软件升级方法,其特征在于,所述方法还包括:

检查是否存在被所述待升级版本软件进行有修改的运行数据,若存在,对所述目标版本软件中对应的运行数据进行修改转换,并将完成修改转换的运行数据进行存储。

3. 根据权利要求2所述的软件升级方法,其特征在于,在完成对所述待升级版本软件中版本号不一致的运行数据的版本转换,完成对所述目标版本软件中对应的运行数据的修改转换之后,所述方法还包括:

释放所述第一容器中的待升级版本软件,将所述第二容器中的目标版本软件切换至工作状态,并删除所述共享区域中进行版本转换之前的运行数据和进行修改转换之前的运行数据。

4. 根据权利要求2所述的软件升级方法,其特征在于,所述共享区域包括共享内存和内存映射数据库,所述将所述待升级版本软件的运行数据和所述目标版本软件的运行数据保存在基于内存映射技术的共享区域中,包括:

将所述待升级版本软件的运行数据和所述目标版本软件的运行数据中,具有固定长度的运行数据保存在基于内存映射技术的所述共享内存中;

将所述待升级版本软件的运行数据和所述目标版本软件的运行数据中,存在长度变化的运行数据保存在基于内存映射技术的所述内存映射数据库中。

5. 根据权利要求4所述的软件升级方法,其特征在于,将完成版本转换的运行数据进行存储的步骤,包括:

若进行版本转换的运行数据来自于所述共享内存,则将进行版本转换的运行数据保留在所述共享内存中,并映射新的共享内存,将完成版本转换后的运行数据存储在新的共享内存中;若进行版本转换的运行数据来自于所述内存映射数据库,则将进行版本转换的运行数据保留在所述内存映射数据库中,并将完成版本转换后的运行数据存储在新的内存映射数据库中;

所述将完成修改转换的运行数据进行存储的步骤,包括:

若进行修改转换的运行数据来自于所述共享内存,则将进行修改转换的运行数据保留在所述共享内存中,并映射新的共享内存,将完成修改转换后的运行数据存储在新的共享内存中;若进行修改转换的运行数据来自于所述内存映射数据库,则将进行修改转换的运行数据保留在所述内存映射数据库中,并将完成修改转换后的运行数据存储在新的内存映射数据库中。

6. 根据权利要求1所述的软件升级方法,其特征在于,在所述第一容器中运行所述待升

级版本软件,在所述第二容器中启动所述目标版本软件的步骤,包括:

允许所述第一容器中的所述待升级版本软件进行硬件设置和报文收发,禁止所述第二容器中的所述目标版本软件进行硬件设置和报文收发,以确保所述第一容器中的所述待升级版本软件的软硬件表项一致性。

7. 根据权利要求1所述的软件升级方法,其特征在于,所述运行数据中包括记录软件的版本号标识信息,所述从所述待升级版本软件的运行数据中,查找出与所述目标版本软件的运行数据版本号不一致的运行数据的步骤,包括:

从所述待升级版本软件的运行数据中,查找出与所述目标版本软件的运行数据的标识信息不一致的运行数据,作为版本号不一致的运行数据。

8. 根据权利要求2所述的软件升级方法,其特征在于,所述运行数据中包括序列号,所述序列号在运行数据被修改时发生变化,所述检查是否存在被所述待升级版本软件进行有修改的运行数据的步骤,包括:

检查所述待升级版本软件的运行数据中是否存在序列号发生变化的运行数据,若存在,将该序列号发生变化的运行数据作为被所述待升级版本软件进行有修改的运行数据。

9. 一种电子设备,其特征在于,包括处理器和存储器,所述存储器存储有能够被所述处理器执行的机器可执行指令,所述处理器可执行所述机器可执行指令以实现权利要求1至8任一所述的方法。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至8任一项所述的方法。

## 软件升级方法、电子设备和计算机可读存储介质

### 技术领域

[0001] 本发明涉及通信技术领域,具体而言,涉及一种软件升级方法、电子设备和计算机可读存储介质。

### 背景技术

[0002] 不中断业务升级(In-Service Software Upgrade,ISSU)是指在不停机也不用中断网络服务的情况下升级网络设备上的软件。ISSU允许网络管理员在不中断网络可用性的情况下处理软件错误或者给交换机和路由器添加新功能。

[0003] 通常ISSU需要有具备冗余控制面板的网络设备,如监控引擎(supervisor engines)或路由引擎(routing engines)。这样,网络管理员可以在一台机器上更新软件,而其它机器还能支持网络正常服务,使得网络中断的影响降到最低。但是对于不具备冗余控制面板的网络设备,如单机盒式设备或框式设备,在只有一块主控板时,无法通过上述方式实现ISSU。有鉴于此,如何提供一种适用性更强、实现更为便捷的升级方案,为所需改善的技术问题。

### 发明内容

[0004] 有鉴于此,本发明的目的之一在于提供一种软件升级方法、电子设备和计算机可读存储介质,以提供一种适用性较强,实现较为便捷的软件升级方案。

[0005] 为了实现上述目的,本发明实施例采用的技术方案如下:

[0006] 第一方面,实施例提供一种软件升级方法,应用于电子设备,所述电子设备包括用于存储待升级版本软件的第一容器和用于存储目标版本软件的第二容器,所述方法包括:

[0007] 在所述第一容器中运行所述待升级版本软件,在所述第二容器中启动所述目标版本软件,将所述待升级版本软件的运行数据和所述目标版本软件的运行数据保存在基于内存映射技术的共享区域中,以使所述待升级版本软件和所述目标版本软件能够访问到所述共享区域中的运行数据;

[0008] 从所述待升级版本软件的运行数据中,查找出与所述目标版本软件的运行数据版本号不一致的运行数据,对查找出的所述运行数据进行版本转换,并将完成版本转换的运行数据进行存储。

[0009] 第二方面,实施例提供一种电子设备,包括处理器和存储器,所述存储器存储有能够被所述处理器执行的机器可执行指令,所述处理器可执行所述机器可执行指令以实现前述实施方式任一所述的方法。

[0010] 第三方面,实施例提供一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如前述实施方式任一项所述的方法。

[0011] 本发明实施例提供的软件升级方法、电子设备和计算机可读存储介质,通过设置两个容器,在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,

从而通过对版本号不一致的运行数据的转换,即可完成待升级版本软件的升级,适用性较强,实现较为便捷。

[0012] 为使本发明的上述目的、特征和优点能更明显易懂,下文特举较佳实施例,并配合所附附图,作详细说明如下。

### 附图说明

[0013] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,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相关的附图。

[0014] 图1示出了本发明实施例提供的一种应用场景示意图。

[0015] 图2示出了本发明实施例提供的软件升级方法的流程示意图。

[0016] 图3示出了本发明实施例提供的软件升级方法的另一流程示意图。

[0017] 图4示出了本发明实施例提供的软件升级装置的示例性结构框图。

[0018] 图标:100-电子设备;110-存储器;120-处理器;130-通信模块;140-软件升级装置;141-处理模块;142-版本转换模块。

### 具体实施方式

[0019] 下面将结合本发明实施例中附图,对本发明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发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。

[0020] 因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0021] 需要说明的是,术语“第一”和“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,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、物品或者设备不仅包括那些要素,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、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0022] 请参照图1,是本实施例提供的一种电子设备100的方框示意图。本实施例中的电子设备100可以泛指需要进行软件升级的各设备,如可以为需要在不停机也不用中断网络服务的情况下升级软件的网络设备。电子设备100包括存储器110、处理器120及通信模块130。存储器110、处理器120以及通信模块130各元件相互之间直接或间接地电性连接,以实现数据的传输或交互。例如,这些元件相互之间可通过一条或多条通讯总线或信号线实现电性连接。

[0023] 其中,存储器110用于存储程序或者数据。存储器110可以是,但不限于,随机存取存储器(Random Access Memory, RAM),只读存储器(Read Only Memory, ROM),可编程只读

存储器 (Programmable Read-Only Memory, PROM), 可擦除只读存储器 (Erasable Programmable Read-Only Memory, EPROM), 电可擦除只读存储器 (Electric Erasable Programmable Read-Only Memory, EEPROM) 等。

[0024] 处理器120用于读/写存储器110中存储的数据或程序,并执行相应地功能,从而实现本实施例中的软件升级方法。

[0025] 通信模块130用于通过网络建立电子设备100与其它通信终端之间的通信连接,并用于通过网络收发数据。

[0026] 应当理解的是,图1所示的结构仅为电子设备100的结构示意图,电子设备100还可包括比图1中所示更多或者更少的组件,或者具有与图1所示不同的配置。图1中所示的各组件可以采用硬件、软件或其组合实现。例如,本实施例中,电子设备100还可以包括用于存储待升级版本软件的第一容器和用于存储目标版本软件的第二容器。

[0027] 请结合参阅图2,为本发明实施例提供的一种软件升级方法的流程示意图,可以由图1所示电子设备100执行,例如可以由电子设备100中的处理器120执行,电子设备100包括用于存储待升级版本软件的第一容器和用于存储目标版本软件的第二容器。该软件升级方法包括S110和S120。

[0028] S110,在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,以使待升级版本软件和目标版本软件能够访问到共享区域中的运行数据。

[0029] S120,从待升级版本软件的运行数据中,查找出与目标版本软件的运行数据版本号不一致的运行数据,对查找出的运行数据进行版本转换,并将完成版本转换的运行数据进行存储。

[0030] 通过图2所示流程,在软件升级过程中,在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件,并将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,使得待升级版本软件和目标版本软件均能够访问到共享区域中的运行数据,从而仅需对版本号不一致的运行数据进行转换,即可完成软件升级,实现较为便捷。待升级版本软件和目标版本软件分别位于不同的容器中,无需电子设备100具备冗余控制面板即可实现不中断业务升级,适用性较强。

[0031] 为了确保软件升级过程中,待升级版本软件运行的可靠性,S110中,在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件的情况下,允许第一容器中的待升级版本软件进行硬件设置和报文收发,禁止第二容器中的目标版本软件进行硬件设置和报文收发,以确保第一容器中的待升级版本软件的软硬件表项一致性。

[0032] 鉴于软件升级过程中,第一容器中的待升级版本软件正常运行(允许正常进行硬件设置和报文收发),待升级版本软件具有完整的运行数据,第二容器中的目标版本软件新启动(禁止进行硬件设置和报文收发),目标版本软件的运行数据较少,因而,若执行将待升级版本软件的运行数据向目标版本软件同步的操作,同步过程会耗费较多的时间,严重影响软件升级效率。而本实施例中,通过将待升级版本软件的运行数据和目标版本软件的运行数据均保存在共享区域中,能够被待升级版本软件和目标版本软件同时访问到,从而无需执行上述同步操作,进而显著节省软件升级所需时间,提高软件升级便捷性。

[0033] 基于内存映射技术的共享区域可以为多种,例如,共享区域可以为共享内存、内存

映射数据库等。在一种实现方式中,共享区域可以包括共享内存和内存映射数据库。考虑到共享内存中往往会根据运行数据的长度适应性分配存储空间,运行数据的长度发生变化后,原有的存储空间可能不适配,而内存映射数据库可以不受运行数据的长度变化灵活进行数据存储,S110中,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中可以包括:将待升级版本软件的运行数据和目标版本软件的运行数据中,具有固定长度的运行数据保存在基于内存映射技术的共享内存中。将待升级版本软件的运行数据和目标版本软件的运行数据中,存在长度变化的运行数据保存在基于内存映射技术的内存映射数据库。

[0034] 其中,具有固定长度的运行数据可以包括单个变量、数组等。存在长度变化的运行数据可以包括链表、树、Hash表等。

[0035] S120中,从待升级版本软件的运行数据中查找出与目标版本软件的运行数据版本号不一致的运行数据的方式有多种,例如,各运行数据中可以包括记录软件的版本号的标识信息,进而从待升级版本软件的运行数据中,查找出与目标版本软件的运行数据的标识信息不一致的运行数据,即可得到版本号不一致的运行数据。其中,标识信息可以灵活选择,示例性的,可以为用于标识版本号的字段。

[0036] 在基于内存映射技术的共享区域为共享内存、内存映射数据库等的情况下,S120中进行版本转换的运行数据可能存储在共享内存、内存映射数据库等位置。鉴于共享内存的存储空间与运行数据的长度适配,而内存映射数据库可以不受运行数据的长度变化灵活进行数据存储的特性,将完成版本转换的运行数据进行存储的操作可以为:若进行版本转换的运行数据来自于共享内存,则将进行版本转换的运行数据保留在共享内存中,并映射新的共享内存,将完成版本转换后的运行数据存储在新的共享内存中。若进行版本转换的运行数据来自于内存映射数据库,则将进行版本转换的运行数据保留在内存映射数据库中,并将完成版本转换后的运行数据存储在内存映射数据库中。

[0037] 本实施例中,在将完成版本转换后的运行数据进行存储的同时,将进行版本转换的运行数据(原运行数据)进行保留,从而确保在软件升级过程中,待升级版本软件能够基于原运行数据正常运行,而完成运行数据的版本转换、实现软件升级之后,目标版本软件能够使用完成版本转换后的运行数据正常运行。

[0038] 其中,实现运行数据版本转换的方式可以灵活选择,例如,可以通过版本转换函数、版本转换组件等实现。

[0039] 在软件升级过程中,待升级版本软件可能会修改某些运行数据,为了确保待升级版本软件和目标版本软件中运行数据的“一致性”,本实施例中的软件升级方法还包括:检查是否存在被待升级版本软件进行有修改的运行数据,若存在,对目标版本软件中对应的运行数据进行修改转换,并将完成修改转换的运行数据进行存储。

[0040] 检查是否存在被待升级版本软件进行有修改的运行数据的方式有多种,例如,各运行数据中可以包括序列号,序列号在运行数据被修改时发生变化,从而通过检查待升级版本软件的运行数据中是否存在序列号发生变化的运行数据,即可得出是否存在被待升级版本软件进行有修改的运行数据,如在存在序列号发生变化的运行数据时,该序列号发生变化的运行数据即为被待升级版本软件进行有修改的运行数据。其中,序列号可以灵活选择,示例性地,可以通过一字段记录序列号。

[0041] 在基于内存映射技术的共享区域为共享内存、内存映射数据库等的情况下,被待升级版本软件进行有修改的运行数据可能存储在共享内存、内存映射数据库等位置。鉴于共享内存的存储空间与运行数据的长度适配,而内存映射数据库可以不受运行数据的长度变化灵活进行数据存储的特性,将完成修改转换的运行数据进行存储的操作可以为:若进行修改转换的运行数据来自于共享内存,则将进行修改转换的运行数据保留在共享内存中,并映射新的共享内存,将完成修改转换后的运行数据存储在新的共享内存中。若进行修改转换的运行数据来自于内存映射数据库,则将进行修改转换的运行数据保留在内存映射数据库中,并将完成修改转换后的运行数据存储在内映射数据库中。

[0042] 本实施例中,在将完成修改转换后的运行数据进行存储的同时,将进行修改转换的运行数据(原运行数据)进行保留,从而确保在软件升级过程中,待升级版本软件能够基于原运行数据正常运行,而完成运行数据的修改转换之后,目标版本软件能够使用完成修改转换后的运行数据,确保待升级版本软件和目标版本软件中运行数据的“一致性”。

[0043] 通过上述流程,将待升级版本软件和目标版本软件分别设置于不同的容器中,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,仅需对版本号不一致的运行数据进行版本转换,根据被待升级版本软件进行有修改的运行数据进行对应的修改转换,即可实现待升级版本软件和目标版本软件的运行数据“同步”。从而在完成对待升级版本软件中版本号不一致的运行数据的版本转换,完成对目标版本软件中对应的运行数据的修改转换之后,可以释放第一容器中的待升级版本软件,将第二容器中的目标版本软件切换至工作状态,从而完成不中断业务升级。

[0044] 在完成软件升级之后,可以删除共享区域中进行版本转换之前的运行数据和进行修改转换之前的运行数据,以释放存储空间。例如,在共享区域包括共享内存和内存映射数据库时,将共享内存中进行版本转换之前的运行数据和进行修改转换之前的运行数据删除(释放存储进行版本转换之前的运行数据和进行修改转换之前的运行数据的共享内存),将内存映射数据库中进行版本转换之前的运行数据和进行修改转换之前的运行数据删除。

[0045] 为了更为清楚地阐述本发明实施例的实现原理,现以下述场景为例进行举例说明。

[0046] 请参阅图3,以待升级版本软件为OLD APP(旧软件),目标版本软件为NEW APP(新软件),第一容器为C1(亦称active容器),第二容器为C2(亦称standby容器)为例,对本发明实施例提供的软件升级方法的总体实现流程进行举例说明。

[0047] 请参阅图3(a),通常情况下,如无需进行软件升级的情况下,只需要运行一个容器即可,图3(a)示出了在无需进行软件升级的情况下,只运行C1(active容器)的示意图。在ISSU期间,需要另起一个容器,图3(b)示出了在ISSU期间,运行C1的同时,启动C2(standby容器)的示意图。

[0048] 在ISSU期间,旧软件在active容器中正常工作,可以对硬件进行各种设置,可以进行报文收发等操作。新软件在standby容器中启动,不允许对硬件进行设置,不允许进行报文收发等操作。例如,新软件的各应用进程不允许设置驱动,不允许向驱动发包,驱动将硬件的线程(如一些定时器线程、硬件检测线程、收包线程等)设置为空转模式,对收包队列中的报文直接做丢弃处理。从而保障active容器中的新软件的软件、硬件表项的一致性不受到损害。

[0049] 其中,软件分为用户态软件和内核态软件,内核态软件相对较小,与业务应用没用直接关联,一般无需对内核态软件进行升级,因而,在ISSU期间,可以只考虑对用户态软件的升级。所有用户态的进程组成用户态软件,这些用户态的进程存在于容器,如docker(一个开源的应用容器引擎)中。相应地,本实施例中,在ISSU期间,旧软件的用户态的进程存在于active容器,新软件的用户态的进程存在于standby容器,active容器中的旧软件和standby容器中的新软件互不干扰。

[0050] 用户态软件可以划分为许多不同的软件特性,每个软件特性具有各自的运行数据,这些运行数据与硬件相匹配。例如,若路由模块下发了一条路由给驱动,驱动将这条路由下发到芯片中,并返回分配的资源索引给路由模块,以后删除该路由时需要携带此资源索引。这样路由模块(可看为一个软件特性)和驱动路由模块(可看为另一个软件特性)都需要记录这些运行数据。

[0051] Active容器中的旧软件的软件特性具有完整的运行数据,而standby容器中的新软件的软件特性是新启动的,运行数据基本为空。因而,若将旧软件的运行数据向新软件进行同步,同步过程会花费较多时间,进而严重影响ISSU的完成速度。

[0052] 为了提高ISSU的便捷性和完成速度,本实施例中,将旧软件的运行数据和新软件的运行数据均保存在基于内存映射技术的共享区域中,从而使得旧软件和新软件可以同时访问共享区域中的运行数据。内存映射技术可以灵活选择,例如,可以选择将文件或者其它对象映射进内存。

[0053] 本实施例中,以共享区域包括共享内存和内存映射数据库,内存映射数据库为LMDB(Lightning Memory-Mapped Database Manager,轻量级内存映射数据库)为例,进行说明。旧软件的运行数据和新软件的运行数据均保存在共享内存、LMDB中,可以被旧软件和新软件同时访问。其中,旧软件的运行数据和新软件的运行数据中,具有固定长度的运行数据,如单个变量、数组等保存在共享内存中,存在长度变化的运行数据,如链表、树、Hash表等保存在LMDB中。通过将旧软件的运行数据和新软件的运行数据均保存在共享内存、LMDB中,对于旧软件和新软件中数据结构未发生变化的运行数据,无需进行上述同步操作,仅需对旧软件和新软件中数据结构发生变化的部分运行数据进行转换操作,即可完成软件升级,从而可以显著节省升级时间,提升ISSU的完成速度。

[0054] ISSU升级时,一些运行数据的数据结构可能发生变化,导致旧软件的运行数据不能被新软件直接使用,需要进行转换处理。其中,旧软件和新软件中数据结构发生变化的运行数据可能为版本号发生变化的运行数据。

[0055] 为了从旧软件的运行数据中便捷地识别出与新软件的运行数据版本号不一致的运行数据,可以为各数据结构运行数据分别添加用于记录软件的版本号的标识信息,如版本号(version)字段。当运行数据的数据结构发生变化时,版本号字段取值发生变化,当运行数据的数据结构未发生变化时,版本号字段取值不变。

[0056] 基于版本号字段的设置,通过检查保存在共享内存、LMDB中的各运行数据的版本号字段的取值,即可得出哪些运行数据的数据结构发生了变化,需要做版本转换处理。对应地,为各数据结构运行数据分别提供版本转换函数,当从旧软件的运行数据中查找出与新软件的运行数据版本号不一致的、需要进行版本转换的运行数据时,执行版本转换函数,对需要进行版本转换的运行数据进行版本转换。

[0057] 鉴于各运行数据均保存在共享内存、LMDB中,对于保存在LMDB中的运行数据,通过遍历LMDB中的每个运行数据,若查找出需要进行版本转换的运行数据(简称“旧的运行数据”),执行版本转换函数,将“旧的运行数据”转换为能够被新软件使用的“新的运行数据”,并将“新的运行数据”添加到同一个LMDB中。虽然“新的运行数据”和“旧的运行数据”存放在同一个LMDB中,旧软件仅使用、处理LMDB中“旧的运行数据”,新软件只对“旧的运行数据”做版本转换处理,得到“新的运行数据”,新软件不对“旧的运行数据”作进一步处理,因此不会修改版本转换后的“新的运行数据”,旧软件亦可以基于“旧的运行数据”继续稳定运行。对于保存在共享内存中的运行数据,通过遍历共享内存中的每个运行数据,若查找出需要进行版本转换的“旧的运行数据”,则映射一块新的共享内存,执行版本转换函数,将“旧的运行数据”转换为能够被新软件使用的“新的运行数据”,并将“新的运行数据”添加到新的共享内存中,如此设置,旧软件可以使用原有的共享内存中的运行数据继续稳定运行,新软件可以使用新的共享内存中进行版本转换后的运行数据,二者互不影响。

[0058] 旧软件和新软件中数据结构发生变化的运行数据也可能为被旧软件进行修改的运行数据。例如,ISSU升级过程中,新软件检测得出与旧软件中的运行数据的数据结构的版本号不一致时,执行版本转换函数完成运行数据的数据结构的转换,该过程中,旧软件可能同时在修改相关的运行数据。为了保障新软件和旧软件中运行数据的“一致性”,在旧软件的运行数据因修改而发生变化时,新软件需及时对发生变化的运行数据做修改转换处理。

[0059] 为了便捷地识别出因修改而发生变化的运行数据,可以为各运行数据分别添加用于记录是否被修改的序列号字段,旧软件每次修改运行数据,运行数据的序列号字段发生变化(如每次加1),从而通过检查序列号字段即可查找出被旧软件进行了修改的运行数据,进而对新软件的运行数据中对应的运行数据进行修改转换。

[0060] 本实施例中,用于存储旧软件和新软件的运行数据的共享内存和LMDB的数量不限,例如,共享内存和LMDB均可以为多个,各共享内存和LMDB在软件升级过程中的工作原理类似,因而在此不作赘述。

[0061] 基于上述方式,完成对发生变化(如版本号发生变化和因修改而发生变化)的所有运行数据的转换之后,即可释放旧软件,将新软件切换到工作状态,并删除LMDB中进行数据转换之前的运行数据,释放共享内存中不再使用的、进行转换之前的运行数据,完成ISSU过程。请结合参阅图3(c),为终止旧软件,删除进行版本转换之前的“旧的运行数据”,保留未进行版本转换的运行数据和经过修改转换之后的运行数据的示意图。

[0062] 旧软件和新软件的运行数据“同步”结束后,新软件的运行数据已经和旧软件的运行数据“一致”,此时,可以将旧软件的进程杀掉,释放旧软件,并将新软件切换为工作状态,相应地,原有用于存储新软件的standby容器切换为新的Active容器。新软件中将之前空转的线程放开,可以实际工作,允许新软件对硬件进行读取和设置,允许新软件收发报文。请结合参阅图3(d),示出了在完成软件升级之后,仅运行存储新软件的新的Active容器(C2)。

[0063] 为了执行上述实施例及各个可能的方式中的相应步骤,下面给出一种软件升级装置的实现方式。请参阅图4,图4为本发明实施例提供的一种软件升级装置140的功能模块图,该软件升级装置140可以应用于图1所示电子设备100,该电子设备100包括用于存储待升级版本软件的第一容器和用于存储目标版本软件的第二容器。需要说明的是,本实施例所提供的软件升级装置140,其基本原理及产生的技术效果和上述实施例相同,为简要描

述,本实施例部分未提及之处,可参考上述的实施例中相应内容。该软件升级装置140包括处理模块141和版本转换模块142。

[0064] 其中,处理模块141用于在第一容器中运行待升级版本软件,在第二容器中启动目标版本软件,将待升级版本软件的运行数据和目标版本软件的运行数据保存在基于内存映射技术的共享区域中,以使待升级版本软件和目标版本软件能够访问到共享区域中的运行数据。

[0065] 版本转换模块142,用于从待升级版本软件的运行数据中,查找出与目标版本软件的运行数据版本号不一致的运行数据,对查找出的运行数据进行版本转换,并将完成版本转换的运行数据进行存储。

[0066] 上述模块可以软件或固件(Firmware)的形式存储于图1所示的存储器110中或固化于电子设备100的操作系统(Operating System,OS)中,并可由图1中的处理器120执行。同时,执行上述模块所需的数据、程序的代码等可以存储在存储器110中。

[0067] 在上述基础上,本发明实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如前述实施方式所述的方法。

[0068] 本发明实施例提供的软件升级方法、电子设备和计算机可读存储介质,实现了一种新的、基于内存映射技术的双容器ISSU实现方法,将待升级版本软件和目标版本软件分别以不同的容器形式提供,升级过程中待升级版本软件在一个容器中正常工作,目标版本软件在另一个容器中启动,待升级版本软件和目标版本软件将运行数据保存在基于内存映射技术的共享内存、LMDB中,使得待升级版本软件和目标版本软件都可以同时访问。各运行数据都具有一个字段,以记录当前软件的版本号,为各数据结构的运行数据提供版本转换函数,当待升级版本软件和目标版本软件的运行数据的版本号不一致时,提供版本转换方法,实现版本转换。各运行数据还具有记录序列号的字段,待升级版本软件对运行数据的每一次修改会导致序列号变化,目标版本软件检查序列号并对变化的运行数据及时进行转换处理,即可确保待升级版本软件和目标版本软件中运行数据的一致性。所有运行数据转换结束后,就可以释放待升级版本软件,将目标版本软件切到工作状态,完成ISSU过程。通过这种方式,使得在软件升级过程中只需对发生变化的运行数据作转换处理,处理效率大大提高,实现较为便捷。不受限于需要进行软件升级的设备是否具备冗余控制面板,适用性较强。

[0069] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,附图中的流程图和框图显示了根据本发明的多个实施例的装置、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,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。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。

[0070] 另外,在本发明各个实施例中的各功能模块可以集成在一起形成一个独立的部

分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0071] 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,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,本发明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)执行本发明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。

[0072] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

100

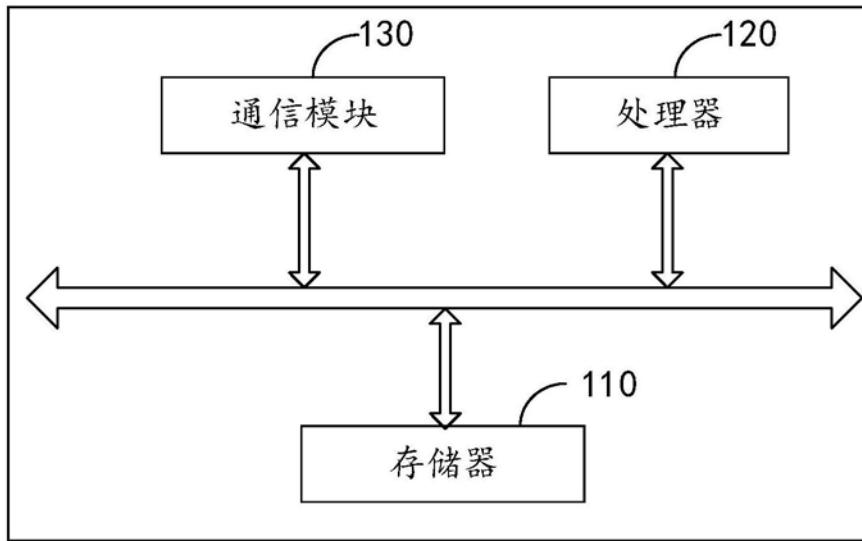


图1

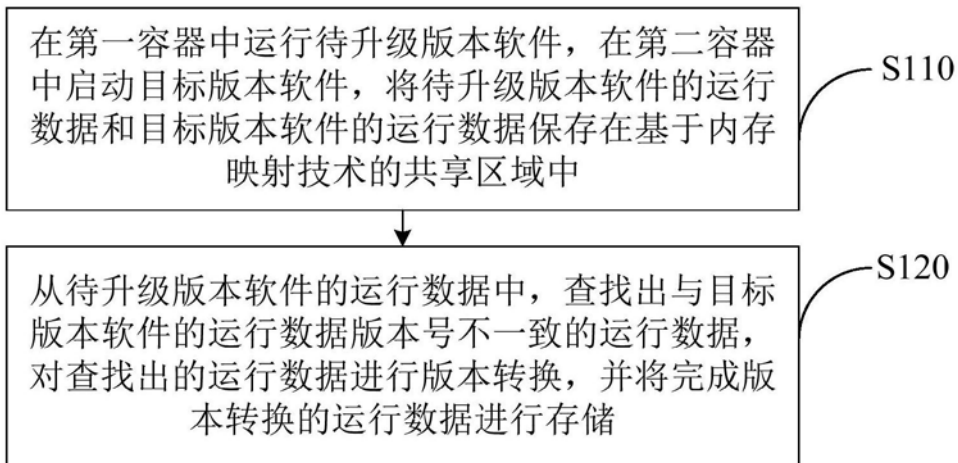


图2

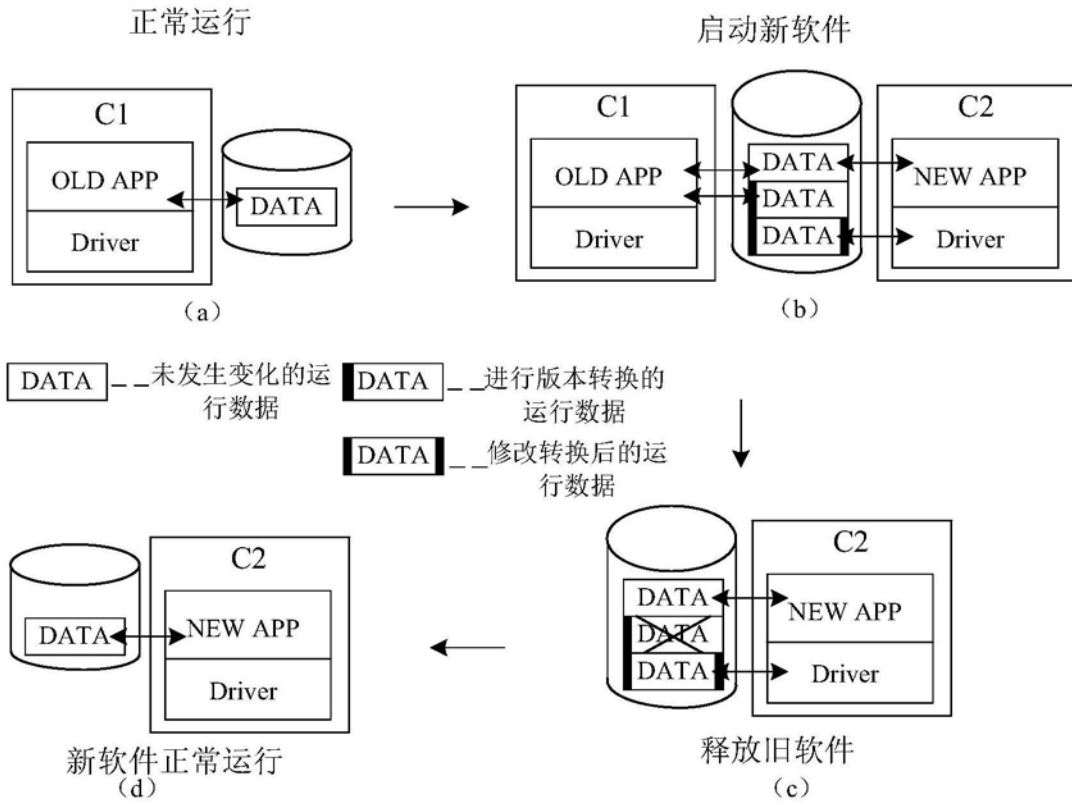


图3



图4