



(12) 发明专利申请

(10) 申请公布号 CN 113312592 A

(43) 申请公布日 2021.08.27

(21) 申请号 202110662243.1

(22) 申请日 2021.06.15

(71) 申请人 曙光信息产业(北京)有限公司
地址 100193 北京市海淀区东北旺西路8号
院36号楼

(72) 发明人 郝文静 张涛 胡梦龙 吕灼恒
原帅 张磊

(74) 专利代理机构 北京品源专利代理有限公司
11332
代理人 孟金喆

(51) Int. Cl.
G06F 21/12 (2013.01)
G06F 9/48 (2006.01)

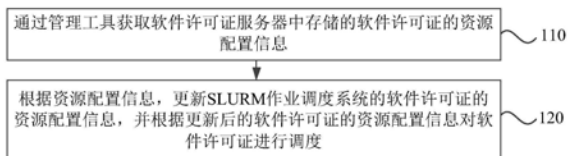
权利要求书2页 说明书11页 附图5页

(54) 发明名称

软件许可证的调度方法、装置、设备及存储介质

(57) 摘要

本发明实施例公开了一种软件许可证的调度方法、装置、设备及存储介质,应用于SLURM作业调度系统中,SLURM作业调度系统与软件许可证的管理工具通信连接,管理工具与至少一个软件许可证服务器通信连接;其中,方法包括:通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息;根据资源配置信息,更新SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。本发明实施例的方案,解决了现有技术中同时将同一软件许可证调度至不同的应用程序,无法对软件许可证进行准确调度的问题,SLURM作业调度系统可以实时获取到管理工具中的软件许可证的资源配置信息,实现了对软件许可证的准确调度。



1. 一种软件许可证的调度方法,应用于SLURM作业调度系统中,所述SLURM作业调度系统与软件许可证的管理工具通信连接,所述管理工具与至少一个软件许可证服务器通信连接,其特征在于,所述方法包括:

通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;

根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

2. 根据权利要求1所述的方法,其特征在于,通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息,包括:

解析用于获取所述资源配置信息的配置文件,得到配置信息;其中,所述配置信息包括目标软件许可证服务器的地址以及待获取的软件许可证信息;

执行第一预设脚本文件,基于所述地址确定所述目标软件许可证服务器,并从所述目标软件许可证服务器中获取与所述待获取的软件许可证信息对应的目标软件许可证的资源配置信息。

3. 根据权利要求2所述的方法,其特征在于,通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息,还包括:

响应于所述第一预设脚本文件的执行完毕指令,执行第二预设脚本文件,从所述管理工具中获取所述目标软件许可证的资源配置信息。

4. 根据权利要求1所述的方法,其特征在于,所述根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,包括:

将所述软件许可证的资源配置信息存储至第一数组中;

将所述SLURM作业调度系统中的历史软件许可证的资源配置信息存储至第二数组中;

依次遍历所述第一数组中的各数据,并与所述第二数组中的各数据进行匹配;

若所述第一数据中的目标数据与所述第二数组中的任一数据均不一致,则将所述目标数据添加至所述第二数组中。

5. 根据权利要求1所述的方法,其特征在于,所述根据更新后的软件许可证的资源配置信息对软件许可证进行调度,包括:

响应于目标应用程序的开启指令,根据所述更新后的软件许可证的资源配置信息确定目标软件许可证;

将所述目标软件许可证分配至目标应用程序,以使所述目标应用程序开启;

其中,所述目标软件许可证未被其他应用程序使用。

6. 根据权利要求5所述的方法,其特征在于,所述根据所述更新后的软件许可证的资源配置信息确定目标软件许可证,包括:

检测所述更新后的软件许可证的资源配置信息,确定未被分配至任一应用程序的软件许可证,作为所述目标软件许可证。

7. 根据权利要求1-6中任一项所述的方法,其特征在于,所述软件许可证的资源配置信息包括软件许可证的总数量和/或所述软件许可证的已使用数量。

8. 一种软件许可证的调度方法,应用于SLURM作业调度系统中,所述SLURM作业调度系统包括软件许可证的管理工具,所述管理工具与至少一个软件许可证服务器通信连接,其特征在于,所述装置包括:

资源配置信息获取模块,用于通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;

资源配置信息更新模块,用于根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

9. 一种软件许可证的调度设备,其特征在于,所述软件许可证的调度设备包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-7中任一所述的软件许可证的调度方法。

10. 一种包含计算机可执行指令的存储介质,其特征在于,所述计算机可执行指令在由计算机处理器执行时用于执行如权利要求1-7中任一所述的软件许可证的调度方法。

软件许可证的调度方法、装置、设备及存储介质

技术领域

[0001] 本发明实施例涉及资源调度技术领域,尤其涉及一种软件许可证的调度方法、装置、设备及存储介质。

背景技术

[0002] SLURM是一种开源的,容错较好的,具有高可伸缩性的集群作业调度系统,SLURM作业调度系统可以支持多种资源的管理和分配,包括中央处理器(Central Processing Unit,CPU)、内存、图形处理器(Graphics Processing Unit,GPU)等,同时也支持软件许可证(License)资源的管理和调度。

[0003] 现阶段,主要通过SLURM作业调度系统中提供的资源配置方式来对软件许可证进行调度;该资源配置方式是通过配置软件许可证的总数量,并根据软件许可证的申请情况来控制软件许可证的使用量。

[0004] 但是,现有的方法导致SLURM作业调度系统无法从管理工具中实时获取到软件许可证的使用信息,存在同时将同一软件许可证调度至不同的应用程序,无法对软件许可证进行准确调度,甚至导致软件许可证的调度失败。

发明内容

[0005] 本发明实施例提供一种软件许可证的调度方法、装置、设备及存储介质,以实现SLURM作业调度系统实时获取管理工具中的软件许可证的资源配置信息,可以对软件许可证进行准确调度。

[0006] 第一方面,本发明实施例提供了一种软件许可证的调度方法,应用于SLURM作业调度系统中,所述SLURM作业调度系统包括软件许可证的管理工具,所述管理工具与至少一个软件许可证服务器通信连接,所述方法包括:

[0007] 通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;

[0008] 根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

[0009] 在本实施例的一个可选实现方式中,通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息,包括:

[0010] 解析用于获取所述资源配置信息的配置文件,得到配置信息;其中,所述配置信息包括目标软件许可证服务器的地址以及待获取的软件许可证信息;

[0011] 执行第一预设脚本文件,基于所述地址确定所述目标软件许可证服务器,并从所述目标软件许可证服务器中获取所述待获取的软件许可证信息对应的目标软件许可证的资源配置信息。

[0012] 采用上述技术方案,可以使管理工具准确地获取到软件许可证的资源配置信息,为后续软件许可证的调度工作的准确性提供帮助。

[0013] 在本实施例的一个可选实现方式中,通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息,包括:

[0014] 响应于所述第一预设脚本文件的执行完毕指令,执行第二预设脚本文件,从所述管理工具中获取所述目标软件许可证的资源配置信息。

[0015] 采用上述技术方案,可以使SLURM作业调度系统准确地获取到软件许可证的资源配置信息,进一步为后续软件许可证的调度工作的准确性提供帮助。

[0016] 在本实施例的一个可选实现方式中,所述根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,包括:

[0017] 将所述软件许可证的资源配置信息存储至第一数组中;

[0018] 将所述SLURM作业调度系统中的历史软件许可证的资源配置信息存储至第二数组中;

[0019] 依次遍历所述第一数组中的各数据,并与所述第二数组中的各数据进行匹配;

[0020] 若所述第一数据中的目标数据与所述第二数组中的任一数据均不一致,则将所述目标数据添加至所述第二数组中。

[0021] 采用上述技术方案,可以便于SLURM作业调度系统中的软件许可证的资源配置信息快速更新,且更新准确率较高,不会存在遗漏更新等问题。

[0022] 在本实施例的一个可选实现方式中,所述根据更新后的软件许可证的资源配置信息对软件许可证进行调度,包括:

[0023] 响应于目标应用程序的开启指令,根据所述更新后的软件许可证的资源配置信息确定目标软件许可证;

[0024] 将所述目标软件许可证分配至目标应用程序,以使所述目标应用程序开启;

[0025] 其中,所述目标软件许可证未被其他应用程序使用。

[0026] 采用上述技术方案,可以避免将已使用的软件许可证分配至应用程序,可以从源头避免软件许可证的调度作业的失败。

[0027] 在本实施例的一个可选实现方式中,所述根据所述更新后的软件许可证的资源配置信息确定目标软件许可证,包括:

[0028] 检测所述更新后的软件许可证的资源配置信息,确定未被分配至任一应用程序的软件许可证,作为所述目标软件许可证。

[0029] 采用上述技术方案,可以实时地响应目标应用程序的开启指令,根据更新后的软件许可证的资源配置信息确定目标软件许可证;将目标软件许可证分配至目标应用程序,以使目标应用程序开启,可以保证目标应用程序的正常开启,不会出现由于软件许可证的调度失败而导致应用程序无法正常开启的情况。

[0030] 在本实施例的一个可选实现方式中,所述软件许可证的资源配置信息包括软件许可证的总数量和/或所述软件许可证的已使用数量。

[0031] 第二方面,本发明实施例还提供了一种软件许可证的调度方法,应用于SLURM作业调度系统中,所述SLURM作业调度系统包括软件许可证的管理工具,所述管理工具与至少一个软件许可证服务器通信连接,所述装置包括:

[0032] 资源配置信息获取模块,用于通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;

[0033] 资源配置信息更新模块,用于根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

[0034] 第三方面,本发明实施例还提供了一种软件许可证的调度设备,所述软件许可证的调度设备包括:

[0035] 一个或多个处理器;

[0036] 存储装置,用于存储一个或多个程序,

[0037] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如本发明实施例中任一实施例所述的软件许可证的调度方法。

[0038] 第四方面,本发明实施例还提供了一种包含计算机可执行指令的存储介质,所述计算机可执行指令在由计算机处理器执行时用于执行如本发明实施例中任一实施例所述的软件许可证的调度方法。

[0039] 本发明实施例通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息;根据资源配置信息,更新SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度,解决了现有技术中同时将同一软件许可证调度至不同的应用程序,无法对软件许可证进行准确调度的问题,SLURM作业调度系统可以实时获取到管理工具中的软件许可证的资源配置信息,实现了对软件许可证的准确调度。

附图说明

[0040] 图1是本发明实施例中的一种软件许可证的调度方法的流程图;

[0041] 图2是本发明实施例中的一种软件许可证的调度方法的流程图;

[0042] 图3是本发明实施例中的一种软件许可证的调度方法的流程图;

[0043] 图4是本发明实施例中的一种实通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息的示意图;

[0044] 图5是本发明实施例中的一种SLURM作业调度系统实时更新软件许可证资源配置信息的示意图;

[0045] 图6是本发明实施例中的一种软件许可证的调度装置的结构示意图;

[0046] 图7是本发明实施例中的一种软件许可证的调度设备的结构示意图。

具体实施方式

[0047] 下面结合附图和实施例对本发明实施例作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明实施例,而非对本发明实施例的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明实施例相关的部分而非全部结构。

[0048] 图1是本发明实施例中的一种软件许可证的调度方法的流程图,本实施例可适用于SLURM作业调度系统实时获取软件许可证的资源配置信息,并对软件许可证进行准确调度的情况,该方法可以由软件许可证的调度装置来执行,该装置可以通过软件和/或硬件的方式实现,该装置可以由独立的硬件设备完成,例如在SLURM作业调度系统中增加独立的服务器或者计算机等,也可以集成在多媒体指挥系统中已有的服务器中,本实施例中对其不

加以限定；该方法可以应用于SLURM作业调度系统中，SLURM作业调度系统与软件许可证的管理工具通信连接，管理工具与至少一个软件许可证服务器通信连接。具体的，参考图1，该方法具体包括如下步骤：

[0049] 步骤110、通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息。

[0050] 其中，管理工具可以为通用的软件许可证管理工具“FLEXlm”，其可以使软件许可证能够在网上浮动；可以理解的是，浮动的软件许可证有利于应用程序（软件）的使用和对软件许可证的管理，这使得应用程序能够高效地使用有效的许可，并使管理者能够控制允许使用它的节点；需要说明的是，本实施例中的管理工具也可以为其他自定义的软件许可证管理工具，本实施例中对其不加以限定。

[0051] 软件许可证的资源配置信息可以包括软件许可证的总数量和/或软件许可证的已使用数量；示例性的，软件许可证的资源配置信息可以为：软件许可证的总数量“100”，已使用数量“10”；软件许可证的资源配置信息还可以为：第一类别特征的软件许可证的总数量“10”，已使用数量“1”，第二类别特征的软件许可证的总数量“90”，已使用数量“9”；其中，第一类别特征和第二类别特征可以根据各软件许可证的属性信息进行区分，例如，可以根据各软件许可证的大小、创建时间或者修改时间等属性信息进行区分。

[0052] 在本实施例的一个可选实现方式中，可以通过与SLURM作业调度系统通信连接的软件许可证的管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息；例如，可以获取第一软件许可证服务器中存储的软件许可证的资源配置信息，也可以获取第二软件许可证服务器中存储的软件许可证的资源配置信息，其中，第一软件许可证服务器与第二软件许可证服务器的网际互联网协议（Internet Protocol, IP）地址不同。

[0053] 在本实施例的一个具体例子中，可以每间隔设定时间（例如，1分钟、5分钟或者1小时等，本实施例中对其不加以限定），通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息。

[0054] 在本实施例的另一个具体例子中，可以在接收到目标应用程序的开启指令时，即需要SLURM作业调度系统为目标应用程序调度一个软件许可证，以使目标应用程序顺利开启时，通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息。

[0055] 步骤120、根据资源配置信息，更新SLURM作业调度系统的软件许可证的资源配置信息，并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

[0056] 在本实施例的一个可选实现方式中，在通过管理工具获取到软件许可证服务器中存储的软件许可证的资源配置信息之后，可以根据获取到的资源配置信息对SLURM作业调度系统的软件许可证的资源配置信息进行更新；进一步的，可以根据更新后的资源配置信息对软件许可证进行调度。

[0057] 在本实施例的一个具体例子中，若获取到的资源配置信息为：软件许可证的总数量“100”，已使用数量“10”；SLURM作业调度系统的软件许可证的资源配置信息为：软件许可证的总数量“90”，已使用数量“5”，则可以将SLURM作业调度系统的软件许可证的资源配置信息更新为：软件许可证的总数量“100”，已使用数量“10”；进一步的，可以根据更新后的资源配置信息对软件许可证进行调度，例如，可以根据更新后的资源配置信息将未使用的软件许可证A调度至目标应用程序，以使目标应用程序顺利开启。

[0058] 本实施例的方案,通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息;根据资源配置信息,更新SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度,解决了现有技术中同时将同一软件许可证调度至不同的应用程序,无法对软件许可证进行准确调度的问题,SLURM作业调度系统可以实时获取到管理工具中的软件许可证的资源配置信息,实现了对软件许可证的准确调度。

[0059] 图2是本发明实施例中的一种软件许可证的调度方法的流程图,本实施例是对上述各技术方案为进一步细化,本实施例中的技术方案可以与上述一个或者多个实施例中的各个可选方案结合。如图2所示,软件许可证的调度方法可以包括如下步骤:

[0060] 步骤210、解析用于获取资源配置信息的配置文件,得到配置信息。

[0061] 其中,配置信息可以包括目标软件许可证服务器的地址以及待获取的软件许可证信息;其中,目标软件许可证服务器的地址可以为目标软件许可证服务器的IP地址,也可以为目标软件许可证服务器的物理地址(MAC地址),可以理解的是,通过解析配置文件从而得到目标软件许可证服务器的地址可以唯一地确定目标软件许可证服务器;待获取的软件许可证信息可以包括待获取的软件许可证的类别特征、每个类别特征下的软件许可证的数量,或者每个软件许可证的属性信息等(例如,软件许可证的大小、创建时间或者内容特征等),本实施例中对其不加以限定。示例性的,解析得到的配置信息可以为:目标软件许可证服务器的地址“192.168.0.123”,以及待获取的软件许可证信息“类别特征为A、创建时间为2021.6.1的软件许可证”。

[0062] 还需要说明的是,本实施例中涉及到的配置文件的后缀名可以为“.conf”,其可以用于获取资源配置信息,通过对配置文件进行解析,可以得到上述配置信息。

[0063] 在本实施例的一个可选实现方式中,SLURM作业调度系统通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息,可以首先对用户自定义的配置文件进行解析,从而得到配置信息,这样可以有助于后续快速地获取到目标软件许可证服务器中存储的资源配置信息。

[0064] 步骤220、执行第一预设脚本文件,基于地址确定目标软件许可证服务器,并从目标软件许可证服务器中获取待获取的软件许可证信息对应的目标软件许可证资源。

[0065] 其中,第一预设脚本文件可以和配置文件共同作用,从而从目标软件许可证服务器中获取到与待获取的软件许可证信息对应的目标软件许可证资源,第一预设脚本文件为用户自定义文件,其后缀名为“.sh”,本实施例中对第一预设脚本文件的具体代码内容不作限定。

[0066] 在本实施例的一个可选实现方式中,在解析得到目标软件许可证服务器的地址以及待获取的软件许可证信息之后,可以执行第一预设脚本文件,确定与解析得到的地址对应的目标软件许可证服务器,并从软件许可证服务器中获取与待获取的软件许可证信息对应的目标软件许可证资源。

[0067] 示例性的,若目标软件许可证服务器的地址“192.168.0.123”,以及待获取的软件许可证信息“类别特征为A、创建时间为2021.6.1的软件许可证”,则通过执行第一预设脚本文件,可以根据地址“192.168.0.123”唯一地确定目标软件许可证服务器,并进一步的可以从目标软件许可证服务器中获取到与“类别特征为A、创建时间为2021.6.1的软件许可证”

的软件可证信息对应的目标软件许可证的资源配置信息(例如,类别特征为A、创建时间为2021.6.1的软件许可证的总数量为“100”,已使用量为“10”)。

[0068] 步骤230、响应于第一预设脚本文件的执行完毕指令,执行第二预设脚本文件,从管理工具中获取目标软件许可证的资源配置信息。

[0069] 其中,第二预设脚本文件可以在第一预设脚本文件执行完毕后,即管理工具从目标软件许可证服务器中获取到目标软件许可证的资源配置信息之后,使SLURM作业调度系统获取到管理工具获取到的目标软件许可证的资源配置信息;第二预设脚本文件也为用户自定义文件,其后缀名为“.sh”,本实施例中第二预设脚本文件的具体代码内容也不作限定。可以理解的是,本实施例中涉及到的第一预设脚本文件与第二预设脚本文件所实现的功能并不相同,因此,第一预设脚本文件与第二预设脚本文件的代码内容也是不同的。

[0070] 在本实施例的一个可选实现方式中,在第一预设脚本文件执行完毕之后,即在管理工具从目标软件许可证服务器中获取到目标软件许可证的资源配置信息之后,可以继续执行第二预设脚本文件,进而从管理工具中获取目标软件许可证的资源配置信息。

[0071] 示例性的,在管理工具获取到与“类别特征为A、创建时间为2021.6.1的软件许可证”的软件可证信息对应的目标软件许可证的资源配置信息“类别特征为A、创建时间为2021.6.1的软件许可证的总数量为“100”,已使用量为“10””之后,可以进一步的执行第二预设脚本文件,以使SLURM作业调度系统获取到目标软件许可证的配置信息。

[0072] 可选的,本实施例中获取到的目标软件许可证的资源配置信息中除了包括软件许可证的总数量以及软件许可证的已使用数量之外,还可以包括每个软件许可证的属性信息(例如,每个软件许可证的内容信息、大小、生成时间或者修改时间等),通过属性信息可以准确地确定已使用的软件许可证和未使用的软件许可证,这样可以便于后续更为准确地对软件许可证进行调度,可以避免对同一软件许可证进行重复调度。

[0073] 步骤240、根据资源配置信息,更新SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

[0074] 本实施例的方案,通过解析用于获取资源配置信息的配置文件,得到配置信息;执行第一预设脚本文件,基于地址确定目标软件许可证服务器,并从目标软件许可证服务器中获取与待获取的软件许可证信息对应的目标软件许可证的资源配置信息,可以使管理工具准确地获取到软件许可证的资源配置信息,为后续软件许可证的调度工作的准确性提供帮助;响应于第一预设脚本文件的执行完毕指令,执行第二预设脚本文件,软件许可证的资源配置信息从管理工具中获取目标软件许可证的资源配置信息,可以使SLURM作业调度系统准确地获取到软件许可证的资源配置信息,进一步为后续软件许可证的调度工作的准确性提供帮助。

[0075] 图3是本发明实施例中的一种软件许可证的调度方法的流程图,本实施例是对上述各技术方案的进一步细化,本实施例中的技术方案可以与上述一个或者多个实施例中的各个可选方案结合。如图3所示,软件许可证的调度方法可以包括如下步骤:

[0076] 步骤310、通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息。

[0077] 步骤320、将软件许可证的资源配置信息存储至第一数组中;将SLURM作业调度系统中的历史软件许可证的资源配置信息存储至第二数组中;依次遍历第一数组中的各数

据,并与第二数组中的各数据进行匹配;若第一数据中的目标数据与第二数组中的任一数据均不一致,则将目标数据添加至第二数组中。

[0078] 其中,第一数组和第二数组初始皆为空数组,本实施例中对第一数组和第二数组的维度不加以限定。

[0079] 在本实施例的一个可选实现方式中,在通过管理工具获取到软件许可证服务器中存储的软件许可证的资源配置信息之后,可以将获取到的资源配置信息存储在第一数组中,并将SLURM作业调度系统中的历史软件许可证的资源配置信息存储至第二数组中;进一步的,可以依次遍历存储在第一数组中的每一个数据,并与第二数组中存储的各数据进行匹配,即确定第一数组中的各数据是否与第二数组中存储的数据完全相同;如果第一数组中存储的目标数据与第二数组中存储的任一数据均不一致,即均不匹配,则可以将目标数据添加至第二数组中。其中,目标数据可以为获取到的资源配置信息中的任一数据,本实施例中对其不加以限定,例如,软件许可证的总数量、软件许可证的已使用数量之外或者每个软件许可证的属性信息等。

[0080] 这样设置的好处在于,可以便于SLURM作业调度系统中的软件许可证的资源配置信息快速更新,且更新准确率较高,不会存在遗漏更新等问题。

[0081] 步骤330、响应于目标应用程序的开启指令,根据更新后的软件许可证的资源配置信息确定目标软件许可证;将目标软件许可证分配至目标应用程序,以使目标应用程序开启。

[0082] 其中,目标软件许可证未被其他应用程序使用。

[0083] 在本实施例的一个可选实现方式中,在根据获取到的目标软件许可证的资源配置信息,更新SLURM作业调度系统的软件许可证的资源配置信息之后,如果接收到目标应用程序的开启指令,可以根据更新后的资源配置信息确定目标软件许可证,并将所确定的目标软件许可证分配至待开启的目标应用程序,从而使目标应用程序正常开启。

[0084] 在本实施例的一个可选实现方式中,根据更新后的软件许可证的资源配置信息确定目标软件许可证,可以包括:检测更新后的软件许可证的资源配置信息,确定未被分配至任一应用程序的软件许可证,作为目标软件许可证。

[0085] 可选的,可以根据更新后的资源配置信息中软件许可证的总数量、已使用的软件许可证的数量以及以使用的软件许可证的属性信息,在未被使用的软件许可证中选择一个未分配至任一应用程序的软件许可证,并将其作为目标软件许可证。

[0086] 示例性的,若目标应用程序为数据处理程序,在接收到目标应用程序的开启指令时,可以根据更新后的资源配置信息中软件许可证的总数量以及已使用的软件许可证的数量,确定一个未被使用的软件许可证作为目标软件许可证,并分配至数据处理程序,这样保证了分配至数据处理程序的目标软件许可证未被分配至其他应用程序,可以使该数据处理程序正常开启。

[0087] 这样设置的好处在于,可以避免将已使用的软件许可证分配至应用程序,可以从源头避免软件许可证的调度作业的失败。

[0088] 本实施例的方案,可以实时地响应目标应用程序的开启指令,根据更新后的软件许可证的资源配置信息确定目标软件许可证;将目标软件许可证分配至目标应用程序,以使目标应用程序开启,可以保证目标应用程序的正常开启,不会出现由于软件许可证的调

度失败而导致应用程序无法正常开启的情况。

[0089] 为了使本领域技术人员更好地理解本实施例涉及到的软件许可证的调度方法,下面采用一个具体示例进行说明,具体过程包括有:

[0090] 步骤一、实时获取软件许可证的管理工具中的软件许可证的资源配置信息。

[0091] 可选的,可以通过配置文件及第一预设脚本文件的方式,配置要获取的软件许可证服务器地址及要获取的软件许可证信息,通过脚本文件的调用来获取到管理工具里的浮动软件许可证数据,得到软件许可证的总数量和已使用数量。

[0092] 在本实施例中,实通过管理工具获取软件许可证服务器中存储的软件许可证的资源配置信息的示意图可以如图4所示,具体的,参考图4,其主要包括如下步骤:

[0093] 步骤410、获取名称为license_feature.conf的配置文件。

[0094] 步骤420、确定配置文件是否为空。

[0095] 若否,则执行步骤430。

[0096] 步骤430、对配置文件进行解析,得到配置信息。

[0097] 其中,配置信息可以包括:待获取的软件许可证的名称、软件许可证服务器地址、查询命令或者软件许可证的类别特征(feature)等。

[0098] 步骤440、循环解析各行软件可证配置的数据,并后台执行步骤450-480。

[0099] 步骤450、是否全部解析完成。

[0100] 若是,结束。

[0101] 否则,执行步骤460。

[0102] 步骤460、根据查询命令执行管理工具命令,并查询配置的类别特征数据。

[0103] 步骤470、获取与类别特征数据对应的软件许可证总数量和已使用数量。

[0104] 步骤480、将软件许可证总数量和已使用数量存入文件。

[0105] 需要说明的是,在本实施例中名称为license_feature.conf的配置文件的格式可以为:服务器名称license服务器端口地址执行命令类型feature名称,例如,lsdyna7 licenseserver2 lstc_qrun lsdyna7。

[0106] 步骤二、实时根据管理工具中的数据更新SLURM作业调度系统的软件许可证资源配置信息,实现SLURM作业调度系统的软件许可证的实时调度。

[0107] 在具体实现中,可以通过每间隔2分钟执行一次sync_license.sh脚本文件(第二预设脚本文件),更新SLURM作业调度系统中的软件许可证资源配置信息。

[0108] 在本实施例中,SLURM作业调度系统实时更新软件许可证资源配置信息的示意图可以如图5所示,具体的,参考图5,其主要包括如下步骤:

[0109] 步骤510、执行名称为license_analysis.sh的脚本文件,获取管理工具中的资源配置信息。

[0110] 步骤520、将资源配置信息存入数组。

[0111] 步骤530、获取SLURM作业调度系统中的资源配置信息。

[0112] 步骤540、遍历数组中的数据,并与SLURM作业调度系统中的资源配置信息进行匹配。

[0113] 步骤550、是否全部遍历结束。

[0114] 若是,执行步骤560;

- [0115] 否则,执行步骤570。
- [0116] 步骤560、遍历SLURM作业调度系统中的资源配置信息,并与数组中的资源配置信息进行匹配。
- [0117] 步骤561、管理工具解析到的资源配置信息在SLURM作业调度系统中是否已配置。
- [0118] 若是,执行步骤562;
- [0119] 否则,返回执行步骤560。
- [0120] 步骤562、更新管理工具中的资源配置信息至SLURM作业调度系统中。
- [0121] 步骤570、若数组中有新的数据,且SLURM作业调度系统中没有,则向SLURM作业调度系统中插入新的数据,并更新SLURM作业调度系统中的资源配置信息。
- [0122] 步骤三、SLURM作业调度系统根据更新后的资源配置信息,对软件许可证进行实时调度。
- [0123] 本实施例的方案,可以实现SLURM作业调度系统的软件许可证资源与商业应用的软件许可证服务器的数据同步,保证通过SLURM作业调度系统启动的作业,和调度系统外用户单独启动的作业不存在软件许可证资源抢占的情况,实现SLURM作业调度系统作业运行的稳定;同时,也可以实现SLURM作业调度系统对软件许可证资源的灵活调整,自动根据软件许可证服务器的实际数据进行调整,减少管理员操作;另外,也实现了SLURM作业调度系统对软件许可证资源的监控,在作业提交及软件许可证的管理方面提供支撑。
- [0124] 图6是本发明实施例中的一种软件许可证的调度装置的结构示意图,该装置可以执行上述各实施例中涉及到的软件许可证的调度方法。参照图5,该装置包括:资源配置信息获取模块610以及资源配置信息更新模块620。
- [0125] 资源配置信息获取模块610,用于通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;
- [0126] 资源配置信息更新模块620,用于根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。
- [0127] 本实施例的方案,通过资源配置信息获取模块通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;通过资源配置信息更新模块根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度,解决了现有技术中同时将同一软件许可证调度至不同的应用程序,无法对软件许可证进行准确调度的问题,SLURM作业调度系统可以实时获取到管理工具中的软件许可证的资源配置信息,实现了对软件许可证的准确调度。
- [0128] 在本实施例的一个可选实现方式中,资源配置信息获取模块610,具体用于
- [0129] 解析用于获取所述资源配置信息的配置文件,得到配置信息;其中,所述配置信息包括目标软件许可证服务器的地址以及待获取的软件许可证信息;
- [0130] 执行第一预设脚本文件,基于所述地址确定所述目标软件许可证服务器,并从所述目标软件许可证服务器中获取与所述待获取的软件许可证信息对应的目标软件许可证的资源配置信息。
- [0131] 在本实施例的一个可选实现方式中,资源配置信息获取模块610,还具体用于

[0132] 响应于所述第一预设脚本文件的执行完毕指令,执行第二预设脚本文件,软件许可证的资源配置信息从所述管理工具中获取所述目标软件许可证的资源配置信息。

[0133] 在本实施例的一个可选实现方式中,资源配置信息更新模块620,具体用于

[0134] 将所述软件许可证的资源配置信息存储至第一数组中;

[0135] 将所述SLURM作业调度系统中的历史软件许可证的资源配置信息存储至第二数组中;

[0136] 依次遍历所述第一数组中的各数据,并与所述第二数组中的各数据进行匹配;

[0137] 若所述第一数据中的目标数据与所述第二数组中的任一数据均不一致,则将所述目标数据添加至所述第二数组中。

[0138] 在本实施例的一个可选实现方式中,资源配置信息更新模块620包括:目标软件许可证确定子模块,用于

[0139] 响应于目标应用程序的开启指令,根据所述更新后的软件许可证的资源配置信息确定目标软件许可证;

[0140] 将所述目标软件许可证分配至目标应用程序,以使所述目标应用程序开启;

[0141] 其中,所述目标软件许可证未被其他应用程序使用。

[0142] 在本实施例的一个可选实现方式中,目标软件许可证确定子模块,具体用于

[0143] 检测所述更新后的软件许可证的资源配置信息,确定未被分配至任一应用程序的软件许可证,作为所述目标软件许可证。

[0144] 在本实施例的一个可选实现方式中,所述软件许可证的资源配置信息包括软件许可证的总数量和/或所述软件许可证的已使用数量。

[0145] 本发明实施例所提供的软件许可证的调度装置可执行本发明任意实施例所提供的软件许可证的调度方法,具备执行方法相应的功能模块和有益效果。

[0146] 图7为本发明实施例中的一种软件许可证的调度设备的结构示意图,如图7所示,该软件许可证的调度设备包括处理器70、存储器71、输入装置72和输出装置73;软件许可证的调度设备中处理器70的数量可以是一个或多个,图7中以一个处理器70为例;软件许可证的调度设备中的处理器70、存储器71、输入装置72和输出装置73可以通过总线或其他方式连接,图7中以通过总线连接为例。

[0147] 存储器71作为一种计算机可读存储介质,可用于存储软件程序、计算机可执行程序以及模块,如本发明实施例中的软件许可证的调度方法对应的程序指令/模块(例如,软件许可证的调度装置中的资源配置信息获取模块610以及资源配置信息更新模块620)。处理器70通过运行存储在存储器71中的软件程序、指令以及模块,从而执行软件许可证的调度设备的各种功能应用以及数据处理,即实现上述的软件许可证的调度方法。

[0148] 存储器71可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序;存储数据区可存储根据终端的使用所创建的数据等。此外,存储器71可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实例中,存储器71可进一步包括相对于处理器70远程设置的存储器,这些远程存储器可以通过网络连接至软件许可证的调度设备。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0149] 输入装置72可用于接收输入的数字或字符信息,以及产生与软件许可证的调度设备的用户设置以及功能控制有关的键信号输入。输出装置73可包括显示屏等显示设备。

[0150] 本发明实施例还提供一种包含计算机可执行指令的存储介质,所述计算机可执行指令在由计算机处理器执行时用于执行一种软件许可证的调度方法,该方法包括:

[0151] 通过所述管理工具获取所述软件许可证服务器中存储的软件许可证的资源配置信息;

[0152] 根据所述资源配置信息,更新所述SLURM作业调度系统的软件许可证的资源配置信息,并根据更新后的软件许可证的资源配置信息对软件许可证进行调度。

[0153] 当然,本发明实施例所提供的一种包含计算机可执行指令的存储介质,其计算机可执行指令不限于如上所述的方法操作,还可以执行本发明任意实施例所提供的软件许可证的调度方法中的相关操作。

[0154] 通过以上关于实施方式的描述,所属领域的技术人员可以清楚地了解到,本发明可借助软件及必需的通用硬件来实现,当然也可以通过硬件实现,但很多情况下前者是更佳的实施方式。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计算机可读存储介质中,如计算机的软盘、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory, RAM)、闪存(FLASH)、硬盘或光盘等,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)执行本发明各个实施例所述的方法。

[0155] 值得注意的是,上述软件许可证的调度装置的实施例中,所包括的各个单元和模块只是按照功能逻辑进行划分的,但并不局限于上述的划分,只要能够实现相应的功能即可;另外,各功能单元的具体名称也只是为了便于相互区分,并不用于限制本发明的保护范围。

[0156] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。



图1

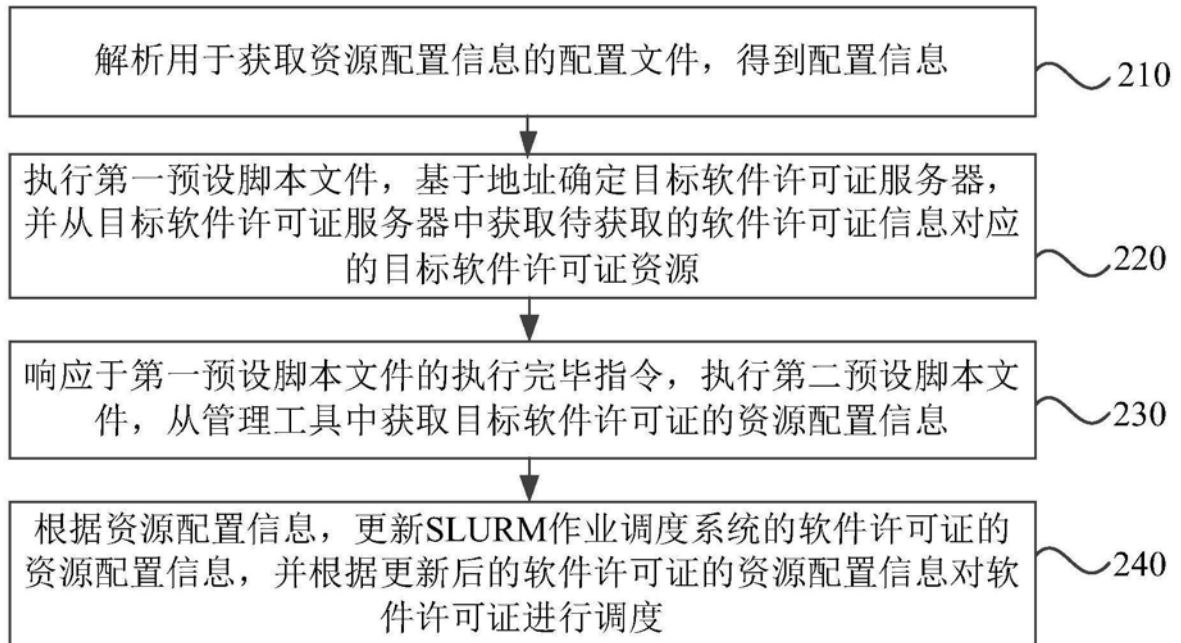


图2

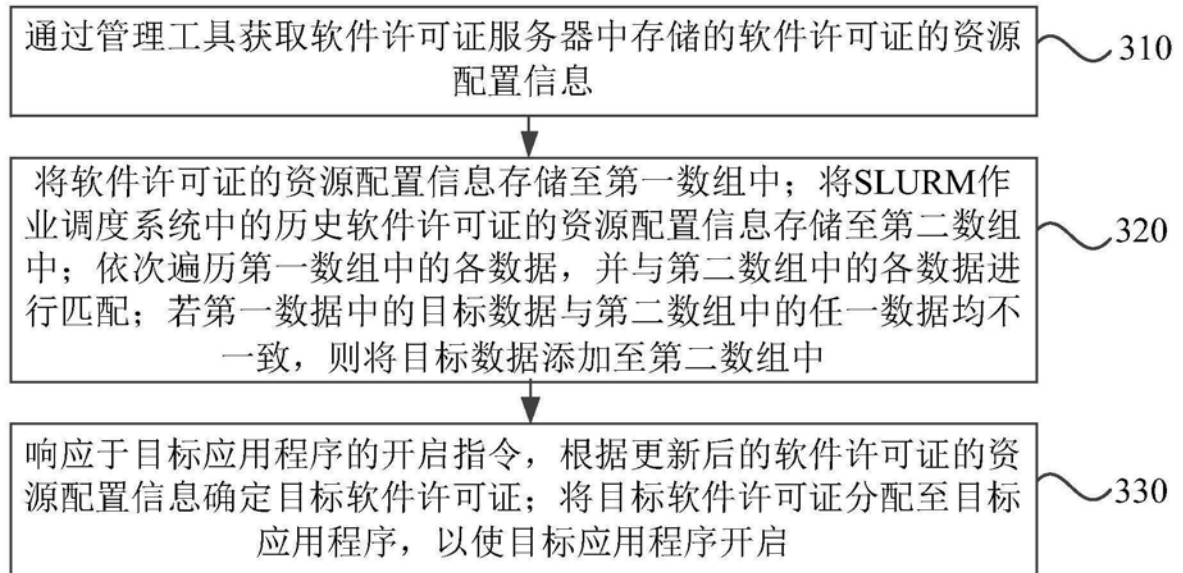


图3

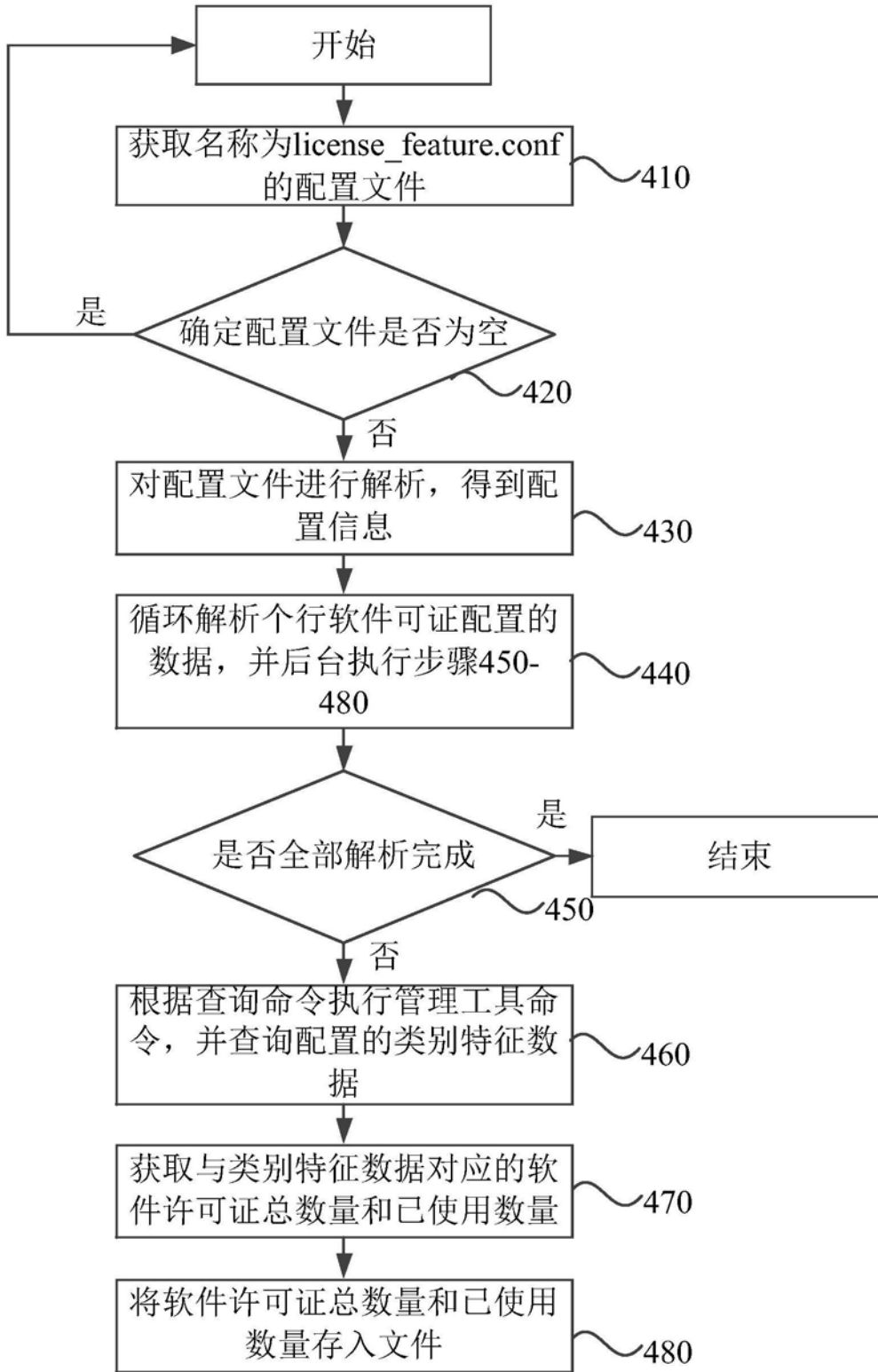


图4

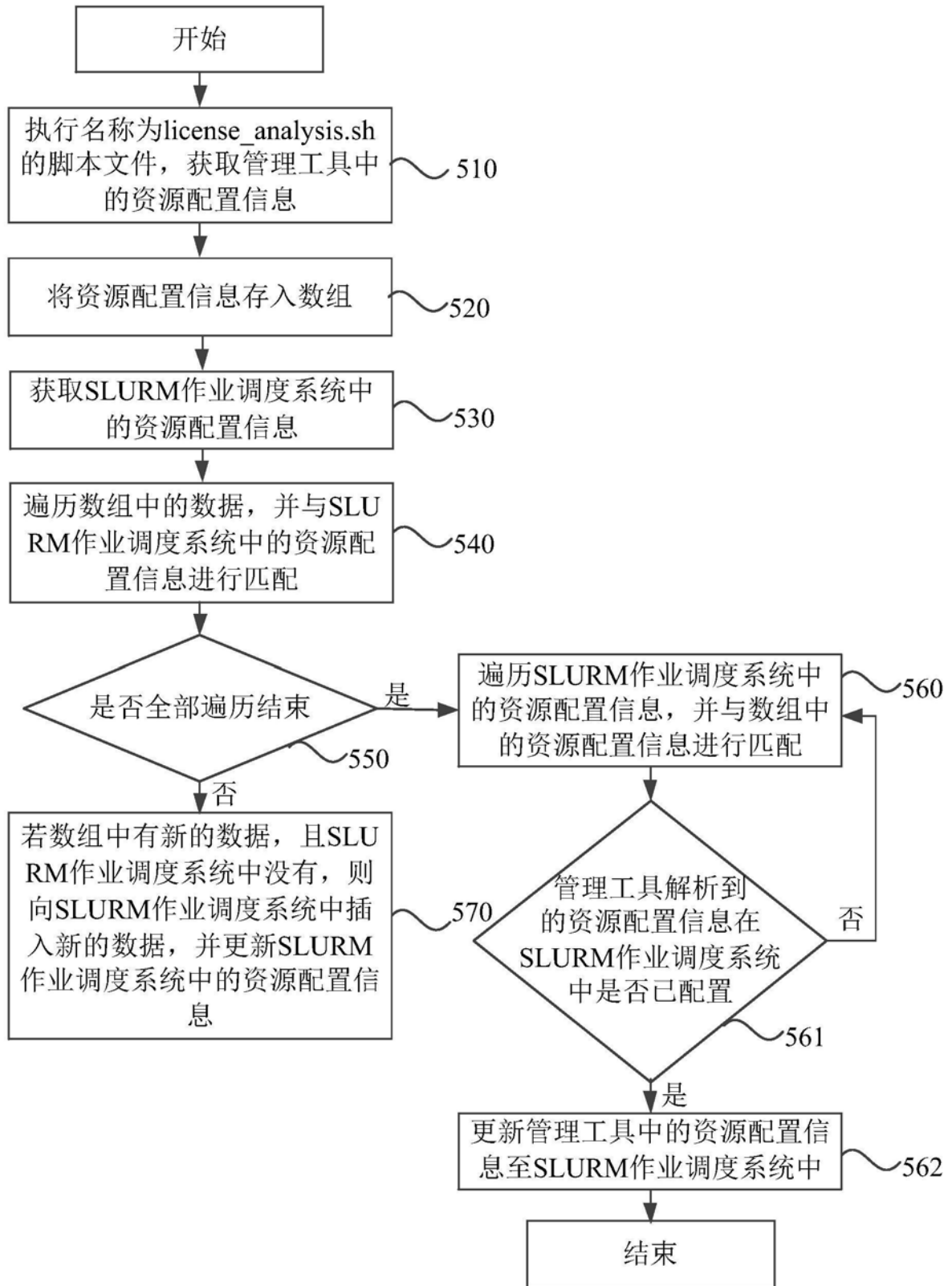


图5

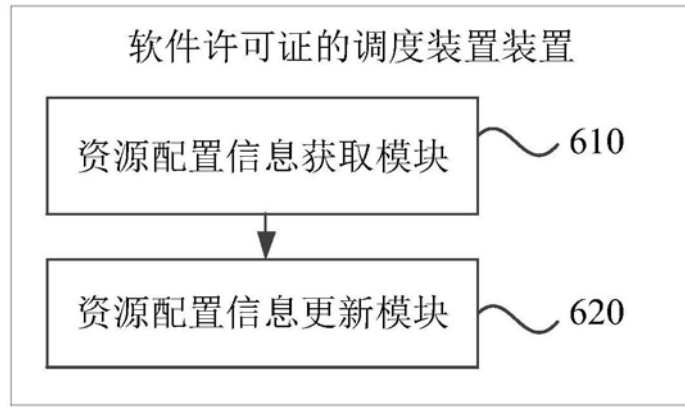


图6

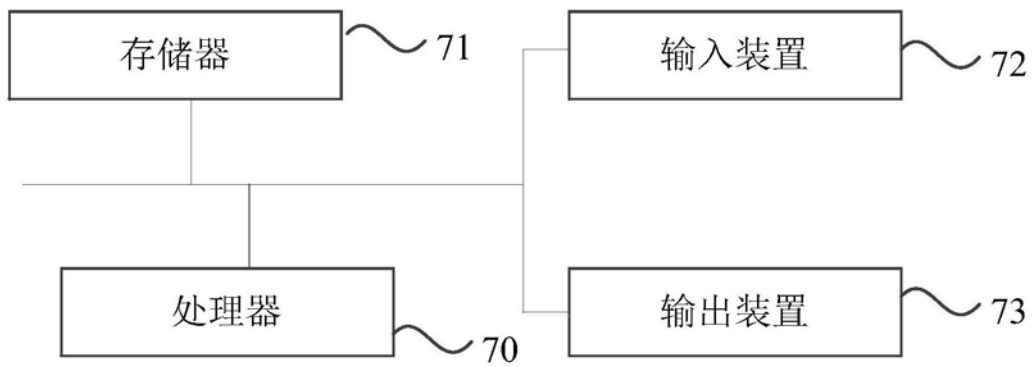


图7