(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114138318 B
(45)授权公告日 2024.01.12

(21)申请号 202111231217.X

(22)申请日 2021.10.22

(65)同一申请的已公布的文献号
申请公布号 CN 114138318 A

(43)申请公布日 2022.03.04

(73)专利权人 苏州浪潮智能科技有限公司
地址 215100 江苏省苏州市吴中经济开发区郭巷街道官浦路1号9幢

(72)发明人 甄鹏 许鑫

(74)专利代理机构 济南舜源专利事务所有限公司 37205
专利代理师 侯绪军

(51)Int.Cl.
G06F 8/70(2018.01)

(56)对比文件
CN 111913958 A,2020.11.10
CN 111190858 A,2020.05.22
CN 1629854 A,2005.06.22
CN 113052501 A,2021.06.29
EP 1746501 A1,2007.01.24
CN 102576344 A,2012.07.11
CN 101405696 A,2009.04.08

审查员 罗重凡

权利要求书2页 说明书11页 附图1页

(54)发明名称

软件资产清点方法、系统、终端及存储介质

(57)摘要

本发明提供一种软件资产清点方法、系统、终端及存储介质，包括：本通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息；对获取到的软件信息进行去重处理后暂存在全局存储中；通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数，并将软件信息和相应软件参数保存至知识库的全局存储。发明以知识库的方式规定了对软件资产进行清点的方法，从而实现了在不改变程序逻辑的情况下，增加完善修改清点逻辑、数据的目的，有效避免了遗漏“绿色版软件”等软件信息的问题。

CN 114138318 B

1.一种软件资产清点方法,其特征在于,包括:

通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

对获取到的软件信息进行去重处理后暂存在全局存储中;

通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储;

根据设定的软件特征,从操作系统中查找具有所述软件特征的软件信息,包括:

创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;

获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标;

通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储,包括:

通过指定脚本引用全局存储中的软件信息获取相应的软件参数,所述软件参数包括版本信息和配置文件路径信息;

将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

2.根据权利要求1所述的方法,其特征在于,对获取到的软件信息进行去重处理后暂存在全局存储中,包括:

获取到新的软件信息时,判断全局存储中是否已经存在完全一致的软件信息:

若是,则舍弃新的软件信息;

若否,则保存新的软件信息。

3.根据权利要求1所述的方法,其特征在于,所述方法还包括:

根据知识库的全局存储中的软件信息和相应软件参数生成软件资产列表;

将所述资产列表进行显示输出。

4.一种软件资产清点系统,其特征在于,包括:

特征匹配单元,用于通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

信息存储单元,用于对获取到的软件信息进行去重处理后暂存在全局存储中;

数据嗅探单元,用于通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储;

所述特征匹配单元包括:

进程遍历模块,用于创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;

信息获取模块,用于获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标;

所述数据嗅探单元包括:

信息获取模块,用于通过指定脚本引用全局存储中的软件信息获取相应的软件参数,所述软件参数包括版本信息和配置文件路径信息;

数据保存模块,用于将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

5.一种终端,其特征在于,包括:

处理器;

用于存储处理器的执行指令的存储器;

其中,所述处理器被配置为执行权利要求1-3任一项所述的方法。

6.一种存储有计算机程序的计算机可读存储介质,其特征在于,该程序被处理器执行时实现如权利要求1-3中任一项所述的方法。

软件资产清点方法、系统、终端及存储介质

技术领域

[0001] 本发明涉及软件管理技术领域,具体涉及一种软件资产清点方法、系统、终端及存储介质。

背景技术

[0002] 资产清点功能,指的是安全软件将操作系统中各个维度的软硬件信息进行提取、加工、合并的功能。对应用软件的资产清点功能,就是将形操作系统中的应用软件列表进行提取整理的功能。

[0003] 在传统的技术方案中,对系统中软件进行资产清点,是一件“死板”的事情。在Windows系统下,软件资产清点往往是通过对注册表键HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall进行遍历读取,进而获取到安装于此系统中的软件信息。而在Linux系统下,软件资产清点则可通过调用rpm-devel库的接口,实现类似的操作。

[0004] 但实际上,许多业务系统使用的是“绿色版”的软件——软件从安装至卸载,都不会在注册表、rpm知识库的全局存储中留下任何的信息,因此传统的方案无法对此类软件资产进行清点。

发明内容

[0005] 针对现有技术的上述不足,本发明提供一种软件资产清点方法、系统、终端及存储介质,以解决上述技术问题。

[0006] 第一方面,本发明提供一种软件资产清点方法,包括:

[0007] 通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

[0008] 对获取到的软件信息进行去重处理后暂存在全局存储中;

[0009] 通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。

[0010] 进一步的,根据设定的软件特征,从操作系统中查找具有所述软件特征的软件信息,包括:

[0011] 创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;

[0012] 获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标。

[0013] 进一步的,对获取到的软件信息进行去重处理后暂存在全局存储中,包括:

[0014] 获取到新的软件信息时,判断全局存储中是否已经存在完全一致的软件信息:

[0015] 若是,则舍弃新的软件信息;

[0016] 若否,则保存新的软件信息。

[0017] 进一步的,通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件

参数,并将软件信息和相应软件参数保存至知识库的全局存储,包括:

[0018] 通过指定脚本引用全局存储中的软件信息获取相应的软件参数,所述软件参数包括版本信息和配置文件路径信息;

[0019] 将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

[0020] 进一步的,所述方法还包括:

[0021] 根据知识库的全局存储中的软件信息和相应软件参数生成软件资产列表;

[0022] 将所述资产列表进行显示输出。

[0023] 第二方面,本发明提供一种软件资产清点系统,包括:

[0024] 特征匹配单元,用于通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

[0025] 信息存储单元,用于对获取到的软件信息进行去重处理后暂存在全局存储中;

[0026] 数据嗅探单元,用于通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。

[0027] 进一步的,所述特征匹配单元包括:

[0028] 进程遍历模块,用于创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;

[0029] 信息获取模块,用于获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标。

[0030] 进一步的,所述信息存储单元用于:

[0031] 获取到新的软件信息时,判断全局存储中是否已经存在完全一致的软件信息:

[0032] 若是,则舍弃新的软件信息;

[0033] 若否,则保存新的软件信息。

[0034] 进一步的,所述数据嗅探单元包括:

[0035] 信息获取模块,用于通过指定脚本引用全局存储中的软件信息获取相应的软件参数,所述软件参数包括版本信息和配置文件路径信息;

[0036] 数据保存模块,用于将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

[0037] 进一步的,所述系统还包括:

[0038] 列表生成单元,用于根据知识库的全局存储中的软件信息和相应软件参数生成软件资产列表;

[0039] 列表显示单元,用于将所述资产列表进行显示输出。

[0040] 第三方面,提供一种终端,包括:

[0041] 处理器、存储器,其中,

[0042] 该存储器用于存储计算机程序,

[0043] 该处理器用于从存储器中调用并运行该计算机程序,使得终端执行上述的终端的方法。

[0044] 第四方面,提供了一种计算机存储介质,所述计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述各方面所述的方法。

[0045] 本发明的有益效果在于,本发明提供的软件资产清点方法、系统、终端及存储介

质,通过通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息,并对获取到的软件信息进行去重处理后暂存在全局存储中,然后通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。本发明以知识库的方式规定了对软件资产进行清点的方法,从而实现了在不改变程序逻辑的情况下,增加完善修改清点逻辑、数据的目的,有效避免了遗漏“绿色版软件”等软件信息的问题。

[0046] 此外,本发明设计原理可靠,结构简单,具有非常广泛的应用前景。

附图说明

[0047] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,对于本领域普通技术人员而言,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。

[0048] 图1是本发明一个实施例的方法的示意性流程图。

[0049] 图2是本发明一个实施例的系统的示意性框图。

[0050] 图3为本发明实施例提供的一种终端的结构示意图。

具体实施方式

[0051] 为了使本技术领域的人员更好地理解本发明中的技术方案,下面将结合本发明实施例中的附图,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。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0052] 下面对本发明中出现的关键术语进行解释。

[0053] CPU中央处理器(central processing unit,简称CPU)作为计算机系统的运算和控制核心,是信息处理、程序运行的最终执行单元。

[0054] 知识库有两种含义:一种是指专家系统设计所应用的规则集合,包含规则所联系的事实及数据,它们的全体构成知识库。这种知识库是与具体的专家系统有关,不存在知识库的共享问题;另一种是指具有咨询性质的知识库,这种知识库是共享的,不是一家所独有的。从今后的发展来看,巨型知识库将会出现,还依赖于硬件及软件条件的发展。下一代计算机所应考虑的重要问题之一是知识库的设计,以知识库为背景的知识库公共管理系统机构设计。知识库的概念来自两个不同的领域,一个是人工智能及其分支-知识工程领域,另一个是传统的知识库的全局存储领域。由人工智能(AI)和知识库的全局存储(DB)两项计算机技术的有机结合,促成了知识库系统的产生和发展。知识库是基于知识且具有智能性的系统(或专家系统)。并不是所有具有智能的程序都拥有知识库,只有基于知识的系统才拥有知识库。许多应用程序都利用知识,其中有的还达到了很高的水平,但是,这些应用程序可能并不是基于知识的系统,它们也不拥有知识库。一般的应用程序与基于知识的系统之间的区别在于:一般的应用程序是把问题求解的知识隐含地编码在程序中,而基于知识的系统则将应用领域的问题求解知识显式地表达,并单独地组成一个相对独立的程序实体。

[0055] 知识库存在以下特点:

[0056] 1、知识库中的知识根据它们的应用领域特征、背景特征(获取时的背景信息)、使用特征、属性特征等而被构成便于利用的、有结构的组织形式。知识片一般是模块化的。

[0057] 2、知识库的知识是有层次的。最低层是“事实知识”,中间层是用来控制“事实”的知识(通常用规则、过程等表示);最高层次是“策略”,它以中间层知识为控制对象。策略也常常被认为是规则的规则。因此知识库的基本结构是层次结构,是由其知识本身的特性所确定的。在知识库中,知识片间通常都存在相互依赖关系。规则是最典型、最常用的一种知识片。

[0058] 3、知识库中可有一种不只属于某一层次(或者说在任一层次都存在)的特殊形式的知识——可信度(或称信任度,置信测度等)。对某一问题,有关事实、规则和策略都可标以可信度。这样,就形成了增广知识库。在知识库的全局存储中不存在不确定性度量。因为在知识库的全局存储的处理中一切都属于“确定型”的。

[0059] 4、知识库中还可存在一个通常被称作典型方法库的特殊部分。如果对于某些问题的解决途径是肯定和必然的,就可以把其作为一部分相当肯定的问题解决途径直接存储在典型方法库中。这种宏观的存储将构成知识库的另一部分。在使用这部分时,机器推理将只限于选用典型方法库中的某一层体部分。

[0060] 作为一款实用的安全软件,有必要针对系统中“绿色软件”在部署、运行过程中留下的蛛丝马迹,实现对这些软件资产的清点。因此,本发明提出了一种无需读取注册表、rpm知识库的全局存储,就可以依靠应用软件痕迹进行软件资产清点的方法。

[0061] 图1是本发明一个实施例的方法的示意性流程图。其中,图1执行主体可以为一种软件资产清点系统。

[0062] 如图1所示,该方法包括:

[0063] 步骤110,通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

[0064] 步骤120,对获取到的软件信息进行去重处理后暂存在全局存储中;

[0065] 步骤130,通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。

[0066] 上述方法为知识库的运行方法,知识库定位为资产清点程序的数据来源,以及软件识别逻辑的来源,由一个个Software(软件)节点组成。每个Software节点分为三部分,分别是静态信息(Info,用于直接形成为数据汇总中的信息)、特征匹配(Characteristic,用于在系统中寻找软件存在的蛛丝马迹,比如进程名、文件路径特征)、数据嗅探(Sniff,在已知软件资产存在的情况下,对其特征进行提取,比如版本、配置文件路径)。

[0067] 特征匹配阶段,负责“寻找”软件资产。此步骤的核心,是如代码1所示的逻辑匹配节点,即知识库编写者可以根据需要,对多个信息匹配节点获取到的信息进行逻辑操作,这些操作包括不限于与、或、非、去重等等。在逻辑节点之下,实际执行特征匹配工作的,是信息匹配节点。在程序代码中,会有与信息匹配节点相对应的处理函数,利用节点中的参数信息在操作系统中执行信息匹配。特征匹配阶段所获取到的信息,最终将会以内存数据的方式暂存在全局存储中。

[0068] 数据嗅探阶段,负责获取上一步“找到”的软件资产的信息。与特征匹配阶段不同,此阶段不适用逻辑匹配节点,相对而言“较为平静”,仅仅是执行一个个的信息获取节点,以

期嗅探得到相关软件的参数数据,并暂存在全局存储中。

[0069] 为了便于对本发明的理解,下面以本发明软件资产清点方法的原理,结合实施例中对软件资产进行清点的过程,对本发明提供的软件资产清点方法做进一步的描述。

[0070] 具体的,所述软件资产清点方法包括:

[0071] S1、通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息。

[0072] 创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标。

[0073] 具体的,本实施例中软件特征值为“tomcat”。

[0074] 具体执行方法如下:

[0075] </Info>

```
<!--特征匹配部分：每一个符合条件的特征结果都将产生一个汇报实例-->
<Characteristic>
<!--逻辑匹配节点：去掉子元素返回集合中的重复元素-->
<Logic key="Deduplicate">
<!--信息匹配节点：根据参数提供的进程名，返回进程对应的可执行程序路径，return 为 All 时返回所有路径-->
<Method key="PathByProcess" return="All" parameter="tomcat"/>
</Logic>
```

[0076]

```
<!--逻辑匹配节点：第一个返回有效值的子元素值即为节点返回值-->
<Logic key="Or">
<!--信息匹配节点：直接返回引用值-->
<Method key="ReturnReference" reference="PathByProcess"/>
<!--信息匹配节点：根据路径匹配返回值（在此例中实际上不需要此节点，再次仅作为单项逻辑节点的示例）-->
<Method key="PathMatch" parameter="*/Apache Software Foundation/Tomcat*"/>
</Logic>
```

[0077]

```
</Characteristic>
```

[0078] 具体的,软件信息如下：

```
<Repository>
  <Software>
    <!--固有信息部分-->
    <Info>
      <!--名称-->
      <Name>Tomcat</Name>
      <!--产品家族-->
      <Family>Tomcat</Family>
      <!--简介，用于展示-->
```

[0079]

```
      <Description>Tomcat是由Apache软件基金会下属的Jakarta项目开发的一个Servlet容器，实现了对Servlet和JavaServer Page（JSP）的支持，并提供了作为Web服务器的一些特有功能，如Tomcat管理和控制平台、安全域管理和Tomcat阀等。由于Tomcat本身也内含了一个HTTP服务器，它也可以被视作一个单独的Web服务器。</Description>
      <!--分类-->
      <SoftwareType>Web Server</SoftwareType>
      <!--图标，Base64编码的jpg图片-->

      <Icon></Icon>
```

[0080] S2、对获取到的软件信息进行去重处理后暂存在全局存储中。

[0081] 获取到新的软件信息时，判断全局存储中是否已经存在完全一致的软件信息：若是，则舍弃新的软件信息；若否，则保存新的软件信息。

[0082] 在本发明的其他实施方式中，也可以采用其他数据去重算法进行软件信息去重。

[0083] S3、通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数，并将软件信息和相应软件参数保存至知识库的全局存储。

[0084] 通过指定脚本引用全局存储中的软件信息获取相应的软件参数，所述软件参数包

括版本信息和配置文件路径信息;将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

[0085] 具体执行方法如下：

```
<!—数据嗅探部分-->
  <Sniff>
    <!--版本信息嗅探-->
    <Item key="Version">
      <!--信息获取节点：获取指定 bat 脚本的执行结果，引用 PathByProcess 结果作为基础路径，并使用正则表达式进行匹配，仅支持 Windows 系统-->
      <Method key="BatExecute" parameter="../bin/Version.bat" reference="PathByProcess" platform="Windows" regex="(?&lt;=Server number: )\d+(.\d+)*"/>
    </Item>
    <!--配置文件路径信息嗅探-->
    <Item key="ConfigPath">
      <Method key="RelativePath" parameter="../config.inf" reference="PathByProcess" platform="Windows"/>
    </Item>
  </Sniff>
</Software>
<Software>
  <!--其它中间件项目-->
</Software>
```

[0086]

[0087]

[0088] S4、以知识库的全局存储中的键值对数据为基础生成资产列表,并对资产列表显示输出。

[0089] 如图2所示,该系统200包括：

[0090] 特征匹配单元210,用于通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息;

[0091] 信息存储单元220,用于对获取到的软件信息进行去重处理后暂存在全局存储中;

[0092] 数据嗅探单元230,用于通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。

[0093] 可选地,作为本发明一个实施例,所述特征匹配单元包括:

[0094] 进程遍历模块,用于创建多个软件节点,并利用多个软件节点并发遍历操作系统的所有进程,查找具有软件特征值的目标进程;

[0095] 信息获取模块,用于获取目标进程的软件信息,所述软件信息包括名称、产品家族、简介、分类和图标。

[0096] 可选地,作为本发明一个实施例,所述信息存储单元用于:

[0097] 获取到新的软件信息时,判断全局存储中是否已经存在完全一致的软件信息:

[0098] 若是,则舍弃新的软件信息;

[0099] 若否,则保存新的软件信息。

[0100] 可选地,作为本发明一个实施例,所述数据嗅探单元包括:

[0101] 信息获取模块,用于通过指定脚本引用全局存储中的软件信息获取相应的软件参数,所述软件参数包括版本信息和配置文件路径信息;

[0102] 数据保存模块,用于将软件信息和相应软件参数以键值对的形式保存至知识库的全局存储。

[0103] 可选地,作为本发明一个实施例,所述系统还包括:

[0104] 列表生成单元,用于根据知识库的全局存储中的软件信息和相应软件参数生成软件资产列表;

[0105] 列表显示单元,用于将所述资产列表进行显示输出。

[0106] 图3为本发明实施例提供的一种终端300的结构示意图,该终端300可以用于执行本发明实施例提供的软件资产清点方法。

[0107] 其中,该终端300可以包括:处理器310、存储器320及通信单元330。这些组件通过一条或多条总线进行通信,本领域技术人员可以理解,图中示出的服务器的结构并不构成对本发明的限定,它既可以是总线形结构,也可以是星型结构,还可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0108] 其中,该存储器320可以用于存储处理器310的执行指令,存储器320可以由任何类型的易失性或非易失性存储终端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。当存储器320中的执行指令由处理器310执行时,使得终端300能够执行以下上述方法实施例中的部分或全部步骤。

[0109] 处理器310为存储终端的控制中心,利用各种接口和线路连接整个电子终端的各个部分,通过运行或执行存储在存储器320内的软件程序和/或模块,以及调用存储在存储器内的数据,以执行电子终端的各种功能和/或处理数据。所述处理器可以由集成电路(Integrated Circuit,简称IC)组成,例如可以由单颗封装的IC所组成,也可以由连接多颗相同功能或不同功能的封装IC而组成。举例来说,处理器310可以仅包括中央处理器

(Central Processing Unit,简称CPU)。在本发明实施方式中,CPU可以是单运算核心,也可以包括多运算核心。

[0110] 通信单元330,用于建立通信信道,从而使所述存储终端可以与其它终端进行通信。接收其他终端发送的用户数据或者向其他终端发送用户数据。

[0111] 本发明还提供一种计算机存储介质,其中,该计算机存储介质可存储有程序,该程序执行时可包括本发明提供的各实施例中的部分或全部步骤。所述的存储介质可为磁碟、光盘、只读存储记忆体(英文:read-only memory,简称:ROM)或随机存储记忆体(英文:random access memory,简称:RAM)等。

[0112] 因此,本发明通过通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息,并对获取到的软件信息进行去重处理后暂存在全局存储中,然后通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数,并将软件信息和相应软件参数保存至知识库的全局存储。本发明以知识库的方式规定了对软件资产进行清点的方法,从而实现了在不改变程序逻辑的情况下,增加完善修改清点逻辑、数据的目的,有效避免了遗漏“绿色版软件”等软件信息的问题,本实施例所能达到的技术效果可以参见上文中的描述,此处不再赘述。

[0113] 本领域的技术人员可以清楚地了解到本发明实施例中的技术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,本发明实施例中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,该计算机软件产品存储在一个存储介质中如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,包括若干指令用以使得一台计算机终端(可以是个人计算机,服务器,或者第二终端、网络终端等)执行本发明各个实施例所述方法的全部或部分步骤。

[0114] 本说明书中各个实施例之间相同相似的部分互相参见即可。尤其,对于终端实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例中的说明即可。

[0115] 在本发明所提供的几个实施例中,应该理解到,所揭露的系统和方法,可以通过其它的方式实现。例如,以上所描述的系统实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,系统或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0116] 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单元来实现本实施例方案的目的。

[0117] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0118] 尽管通过参考附图并结合优选实施例的方式对本发明进行了详细描述,但本发明并不限于此。在不脱离本发明的精神和实质的前提下,本领域普通技术人员可以对本发明

的实施例进行各种等效的修改或替换,而这些修改或替换都应在本发明的涵盖范围内/任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应所述以权利要求的保护范围为准。

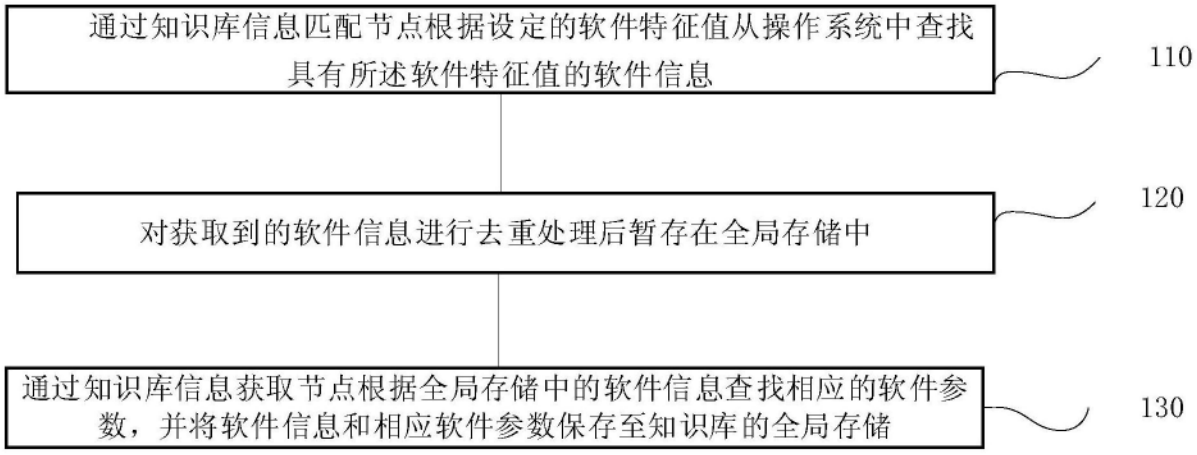
通过知识库信息匹配节点根据设定的软件特征值从操作系统中查找具有所述软件特征值的软件信息
110
对获取到的软件信息进行去重处理后暂存在全局存储中
120
通过知识库信息获取节点根据全局存储中的软件信息查找相应的软件参数，并将软件信息和相应软件参数保存至知识库的全局存储
130

图1


系统 200
特征匹配单元 210
信息存储单元 220
数据嗅探单元 230

图2

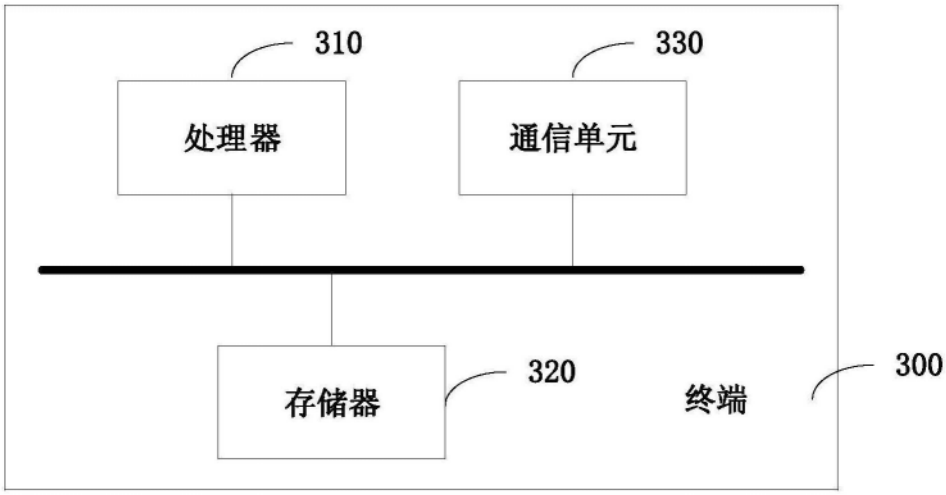
310
330
处理器
通信单元
存储器
320
终端
300

图3