(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103942499 B
(45)授权公告日 2017.01.11

(21)申请号 201410076582.1

(22)申请日 2014.03.04

(65)同一申请的已公布的文献号
申请公布号 CN 103942499 A

(43)申请公布日 2014.07.23

(73)专利权人 中天安泰(北京)信息技术有限公司
地址 100071 北京市丰台区小屯路89号航天标准大厦南楼

(72)发明人 汪家祥

(51)Int.Cl.
G06F 21/62(2013.01)
G06F 21/78(2013.01)

(56)对比文件
CN 101082886 A,2007.12.05,全文.
CN 102023817 A,2011.04.20,全文.
WO 2012/145917 A1,2012.11.01,全文.
WO 2012/145916 A1,2012.11.01,说明书第1页第9-11行,第3页第14-23行,第17页第24-31行,第18页第1-23行,第19页第1-5行,第20页第1-17行及第23页第20-21行.
WO 2012/145915 A1,2012.11.01,全文.

审查员 杨怡睿

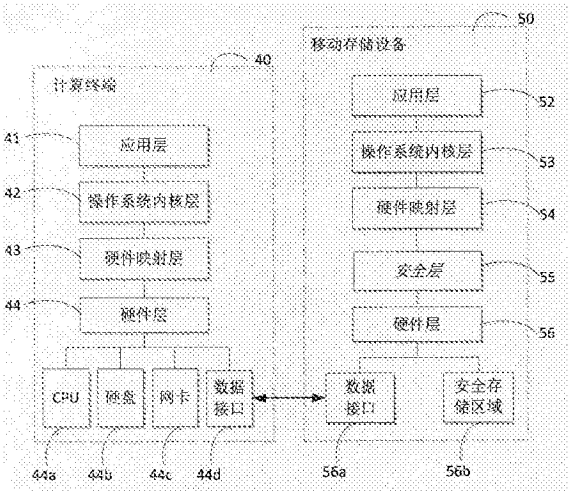
权利要求书4页 说明书34页 附图26页

(54)发明名称

基于移动存储器的数据黑洞处理方法及移动存储器

(57)摘要

本发明提供一种基于移动存储器的数据黑洞处理方法,包括:在计算设备部署数据黑洞系统,使之成为数据黑洞终端;数据黑洞系统是指将计算设备运行过程中的过程数据和运行结果存储至特定存储位置并且能够确保计算设备正常运行的系统;建立数据黑洞空间,包括在所述移动存储器上开辟的数据存储区域;为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系;将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间;阻止对于本地存储设备的数据持久化操作,并且阻止通过本地端口对非数据黑洞终端的数据输出。本发明还提供一种移动存储器。基于移动存储器的数据黑洞处理方法及移动存储器提高数据防泄密的数据安全性。



CN 103942499 B

1.一种基于移动存储器的数据黑洞处理方法,包括:

在计算设备部署数据黑洞系统,使之成为数据黑洞终端;数据黑洞系统是指将计算设备运行过程中的过程数据和运行结果存储至特定存储位置并且能够确保计算设备正常运行的系统;

建立数据黑洞空间,包括在所述移动存储器上开辟的数据存储区域,其中,该数据存储区只能由数据黑洞系统访问,不能被操作系统或应用层软件访问,所述移动存储器与计算设备耦接;

为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系;

将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间;

阻止对于本地存储设备的数据持久化操作,并且阻止通过本地端口对非数据黑洞终端的数据输出,从而保证进入数据黑洞终端或者数据黑洞空间的数据只在数据黑洞空间存在。

2.如权利要求1所述的基于移动存储器的数据黑洞处理方法,其中,部署数据黑洞系统包括部署数据安全存储方法,将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间通过数据安全存储方法实现,数据安全存储方法包括:

接收硬件指令;

如果该硬件指令是存储指令,修改存储指令中的目标地址为当前用户对应的数据黑洞空间的存储地址;和

将修改后的存储指令发送到硬件层执行。

3.如权利要求2所述的基于移动存储器的数据黑洞处理方法,其中,部署数据黑洞系统包括部署数据安全读取方法,数据安全读取方法包括:

接收硬件指令;

如果该硬件指令是读取指令并且其欲读取的数据已经被存储到数据黑洞空间,更改读取指令的源地址为当前用户对应的数据黑洞空间的存储地址;

将修改后的读取指令发送到硬件层执行。

4.如权利要求2所述的基于移动存储器的数据黑洞处理方法,其中,部署数据黑洞系统包括部署数据安全读取方法,数据安全读取方法包括:

接收硬件指令;

如果该硬件指令是读取指令并且其欲读取的数据已经被存储到数据黑洞空间,为用户提供一种选择:读取本地数据或数据黑洞空间数据,并根据用户的选择来读取本地数据或数据黑洞空间数据;

将修改后的读取指令发送到硬件层执行。

5.如权利要求4所述的基于移动存储器的数据黑洞处理方法,其中,读取数据黑洞空间数据包括:

更改读取指令的源地址为当前用户对应的数据黑洞空间的存储地址。

6.如权利要求3或4所述的基于移动存储器的数据黑洞处理方法,其中,接收硬件指令包括:

接收来自硬件抽象层的硬件指令。

7.如权利要求1所述的基于移动存储器的数据黑洞处理方法,其中,部署数据黑洞系统

包括部署数据安全存储方法，将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间通过数据安全存储方法实现，数据安全存储方法包括：

缓存指令运行环境，包括地址寄存器，地址寄存器用于保存下一条将要运行的机器指令的地址，该地址为第一地址；

获取待调度的机器指令片段，其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

分析待调度的机器指令片段中的每一条指令，如果其为存储指令，则修改所述存储指令中的目标地址为对应的数据黑洞空间的存储地址；

在所述第一程序转移指令前，插入第二程序转移指令，生成具有第二地址的重组指令片段，其中，第二程序转移指令指向指令重组平台的入口地址；

将所述地址寄存器中的第一地址修改为第二地址；和

恢复所述指令运行环境。

8.如权利要求1所述的基于移动存储器的数据黑洞处理方法，其中，部署数据黑洞系统包括部署数据安全存储方法，将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间通过数据安全存储方法实现，数据安全存储方法包括：

缓存指令运行环境；

从第一存储位置读取目标地址，根据目标地址获取待调度的机器指令片段；待调度的机器指令片段的最后一条指令为第一程序转移指令；

在第一存储位置保存第一程序转移指令的目标地址；

分析待调度的机器指令片段中的每一条指令，如果其为存储指令，则修改所述存储指令中的目标地址为对应的数据黑洞空间的存储地址；

将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；和

恢复所述指令运行环境，并跳转到第二地址继续执行。

9.如权利要求1所述的基于移动存储器的数据黑洞处理方法，其中，部署数据黑洞系统包括部署数据安全存储方法，将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间通过数据安全存储方法实现，数据安全存储方法包括：

缓存指令运行环境；

获取栈中保存的程序转移指令的地址和参数，计算下一条即将运行的指令地址，该地址为第一地址；

根据第一地址获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

分析待调度机器指令片段中的每一条指令，如果其为存储指令，则修改所述存储指令中的目标地址为对应的数据黑洞空间的存储地址；

替换第一程序转移指令为压栈指令，在压栈指令中记录第一程序转移指令的地址和操作数；

在压栈指令之后加入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；和

恢复所述指令运行环境，并跳转到第二地址继续执行。

10.如权利要求7所述的基于移动存储器的数据黑洞处理方法，其中，部署数据黑洞系统包括部署数据安全读取方法，数据安全读取方法包括：

缓存指令运行环境；所述指令运行环境包括地址寄存器，地址寄存器保存下一条将要运行的机器指令的地址，该地址为第一地址；

获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

分析待调度的机器指令片段中的每一条指令，如果其为读取指令并且其欲读取的数据已经被存储到数据黑洞空间，更改读取指令的源地址为对应的数据黑洞空间的存储地址；

在所述第一程序转移指令前，插入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；

将所述地址寄存器中的第一地址修改为第二地址；和

恢复所述指令运行环境。

11.如权利要求8所述的基于移动存储器的数据黑洞处理方法，其中，部署数据黑洞系统包括部署数据安全读取方法，数据安全读取方法包括：

缓存指令运行环境；

从第一存储位置读取目标地址，根据目标地址获取待调度的机器指令片段；待调度的机器指令片段的最后一条指令为第一程序转移指令；

在第一存储位置保存第一程序转移指令的目标地址；

分析待调度的机器指令片段中的每一条指令，如果其为读取指令并且其欲读取的数据已经被存储到数据黑洞空间，更改读取指令的源地址为对应的数据黑洞空间的存储地址；

将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；和

恢复所述指令运行环境，并跳转到第二地址继续执行。

12.如权利要求9所述的基于移动存储器的数据黑洞处理方法，其中，部署数据黑洞系统包括部署数据安全读取方法，数据安全读取方法包括：

缓存指令运行环境；

获取栈中保存的程序转移指令的地址和参数，计算下一条即将运行的指令地址，该地址为第一地址；

根据第一地址获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

分析待调度的机器指令片段中的每一条指令，如果其为读取指令并且其欲读取的数据已经被存储到数据黑洞空间，更改读取指令的源地址为对应的数据黑洞空间的存储地址；

替换第一程序转移指令为压栈指令，在压栈指令中记录第一程序转移指令的地址和操作数；

在压栈指令之后加入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；和

恢复所述指令运行环境，并跳转到第二地址继续执行。

13.如权利要求7-12中任一项所述的基于移动存储器的数据黑洞处理方法，其中，获取待调度的机器指令片段包括：

从地址寄存器读取待调度的机器指令地址；

以程序转移指令为检索目标，检索所述机器指令地址指向的机器指令及其后续指令，直到发现第一个程序转移指令，称为第一程序转移指令；所述程序转移指令指能够改变机器指令顺序执行流程的机器指令；

将所述第一程序转移指令以及其之前的所有待调度的机器指令作为一个待调度的机器指令片段。

14.如权利要求7-12中任一项所述的基于移动存储器的数据黑洞处理方法，其中，获取待调度的机器指令片段包括：

从地址寄存器读取待调度的机器指令地址；

以程序转移指令为检索目标，检索所述机器指令地址指向的机器指令及其后续指令，直到发现第一个参数地址程序转移指令，称为第一程序转移指令；所述程序转移指令指能够改变机器指令顺序执行流程的机器指令；

将所述第一程序转移指令以及其之前的所有待调度的机器指令作为一个待调度的机器指令片段。

基于移动存储器的数据黑洞处理方法及移动存储器

技术领域

[0001] 本发明涉及计算机安全领域，尤其涉及一种基于移动存储器的数据黑洞处理方法及移动存储器。

背景技术

[0002] 现有的电子信息安全领域包括系统安全、数据安全和设备安全三个子领域。

[0003] 在数据安全领域内，一般采用下面三种技术确保数据安全：

[0004] (1)数据内容安全技术，包括数据加密解密技术和端到端数据加密技术，保障数据在存储和传输过程中内容不被非法读取；

[0005] (2)数据安全转移技术，包括防止非法拷贝、打印或其它输出，保障数据在使用和转移过程中的安全；

[0006] (3)网络阻断技术，包括网络物理阻断和设置网络屏障等技术。

[0007] 根据相关分析，目前针对计算机的所有危害总有效侦测能力最多在50％左右；由于上述技术在应对计算机内核病毒、木马、操作系统漏洞、系统后门以及人为泄密时能力不足，事实上任何计算设备(包括例如计算机、笔记本电脑、手持通信设备等)都可能存在恶意代码。

[0008] 一旦恶意代码进入终端系统，上述的加密技术、防拷贝技术以及网络阻断技术都将失去作用。现有的黑客技术可以利用系统漏洞或系统后门穿透上述安全技术并植入恶意代码，并利用恶意代码取得用户数据。上述技术更无法防范涉密人员的主动或被动泄密，例如，内部人员可以携带存储设备，从内部网络或终端上下载所需的资料并带走存储设备，导致内部泄密；又例如，内部人员可以直接将计算设备带走。

[0009] 综上，防拷贝技术无法保证涉密信息在终端不被非法存储。基于网络过滤无法确保涉密信息不丢失。涉密人员可通过恶意代码或恶意工具造成泄密，还可能因涉密设备或存储介质失控造成泄密。

发明内容

[0010] 本发明的目的是提供一种基于移动存储器的数据黑洞处理方法及移动存储器，提高数据安全性。

[0011] 根据本发明一个方面，提供一种基于移动存储器的数据黑洞处理方法，包括：在计算设备部署数据黑洞系统，使之成为数据黑洞终端；数据黑洞系统是指将计算设备运行过程中的过程数据和运行结果存储至特定存储位置并且能够确保计算设备正常运行的系统；建立数据黑洞空间，包括在所述移动存储器上开辟的数据存储区域，其中，该数据存储区只能由数据黑洞系统访问，不能被操作系统或应用层软件访问，所述移动存储器与计算设备耦接；为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系；将用户在数据黑洞终端操作所产生的数据写重定向到与该用户对应的数据黑洞空间；阻止对于本地存储设备的数据持久化操作，并且阻止通过本地端口对非数据黑洞终端的数据输出，从

而保证进入数据黑洞终端或者数据黑洞空间的数据只在数据黑洞空间存在。

[0012] 根据本发明另一个方面，提供一种移动存储设备，包括：移动版数据安全存取单元以及安全存储空间，其中，移动存储设备本身携带操作系统，安全存储空间对于操作系统及操作系统之上的软件是不可用的，只能由移动版数据安全存取单元访问；其中，当移动存储设备与计算设备耦接时，计算设备的CPU用于执行移动存储设备本身携带的操作系统，用户通过计算设备的I/O与移动存储设备进行交互，移动版数据安全存取单元接收来自移动存储设备本身携带的操作系统的指令并将其发送给计算设备的CPU；其中，移动版数据安全存取单元包括：接收单元，适于接收硬件指令；指令分析单元，适于判断所述硬件指令是否为存储或读取指令，产生判断信号；指令修改单元，根据判断信号，适于当所述硬件指令为存储指令时，将所述存储指令中的目标地址修改为对应的在安全存储空间内的存储地址；还适于当所述硬件指令为读取指令时，查找映射位图，并根据映射位图的数据修改所述读取指令中的读取地址，其中，所述映射位图用于表示计算设备的本地存储空间的地址的数据是否转储到所述安全存储空间；发送单元，适于将修改后的读取或存储指令发送到硬件层执行。

[0013] 可选的，移动存储设备还包括：更新单元，适于在指令修改单元修改所述存储指令之后，更新映射位图中所述目标地址对应的位。

[0014] 可选的，移动存储设备还包括：加解密单元，与所述安全存储空间耦接，适于对进出安全存储空间的数据进行加解密操作。

[0015] 上述方法和设备提高了数据的安全性。具体的，黑洞空间与用户对应，当黑客通过漏洞、后门、木马等恶意代码取得数据权限后将可以对数据进行复制、转储、发送、截留。但所有向外部设备、端口、用户、终端转发出的数据将被重定向到数据黑洞空间(与用户对应的黑洞空间)中，并在数据黑洞空间(与用户对应的黑洞空间)内完成。因此所有的数据窃取、截留、输出等作业都被在数据黑洞空间内实现。当涉密(有数据权限)人员试图将数据私自留存、私自备份、发送、输出时，所有的数据处理作业都在数据黑洞空间(与用户对应的黑洞空间)内完成，使恶意操作无法泄密。

附图说明

[0016] 图1是现有技术中计算设备的系统层次示意图；

[0017] 图2是本发明一个实施例中提供的运行时指令重组方法的流程图；

[0018] 图3是本发明一个实施例中提供的重组指令片段的生成过程示意图；

[0019] 图4是本发明另一个实施例中提供的图2中步骤S102的流程图；

[0020] 图5是本发明另一个实施例中提供的运行时指令重组方法的流程图，利用地址对应表保存已经重组过的指令片段；

[0021] 图6是本发明另一个实施例中提供的运行时指令重组方法的流程图，单独开辟存储位置保存第一程序转移指令的目标地址；

[0022] 图7是本发明另一个实施例中提供的运行时指令重组方法的流程图，针对非固定长度指令集进行反汇编和汇编处理；

[0023] 图8是本发明另一个实施例中提供的运行时指令重组方法的流程图，以压栈指令替代或记录第一程序转移指令；

[0024] 图9a是本发明另一个实施例中提供的运行时指令重组方法的流程图，其中的运行时指令重组方法综合之前多个实施例中的特征；

[0025] 图9b-9d是图9a中的运行时指令重组方法在X86体系处理器上运行时的操作过程示意图；

[0026] 图10是本发明一个实施例中提供的运行时指令重组装置结构示意图；

[0027] 图11是本发明另一个实施例中提供的运行时指令重组装置结构示意图；

[0028] 图12是本发明另一个实施例中提供的指令重组单元结构示意图；

[0029] 图13是本发明另一个实施例中提供的运行时指令重组装置结构示意图；

[0030] 图14是本发明另一个实施例中提供的运行时指令重组装置结构示意图；

[0031] 图15是本发明一个实施例中计算设备的系统层次示意图；

[0032] 图16是本发明一个实施例中提供的数据安全存取过程中的初始化过程的流程图；

[0033] 图17是本发明一个实施例中的Bitmap示意图；

[0034] 图18是本发明一个实施例中提供的数据安全存储方法的流程图；

[0035] 图19是本发明一个实施例中提供的数据安全读取方法的流程图；

[0036] 图20是本发明一个实施例中提供的数据安全存取方法的流程图；

[0037] 图21是本发明一个实施例中提供的数据安全传输方法的流程图；

[0038] 图22是本发明一个实施例中网络环境示意图；

[0039] 图23是本发明一个实施例中提供的数据安全存储装置的结构示意图；

[0040] 图24是本发明一个实施例中提供的数据安全读取装置的结构示意图；

[0041] 图25是本发明一个实施例中提供的数据安全存储和读取装置的结构示意图；

[0042] 图26是本发明另一个实施例中提供的数据安全存储和读取装置的结构示意图；

[0043] 图27是本发明另一个实施例中提供的数据黑洞空间示意图；

[0044] 图28是本发明一个实施例中提供的数据黑洞处理方法的流程图；

[0045] 图29a是本发明一个实施例中提供的计算设备的体系架构示意图，其中运行单机版的数据安全存储和读取方法；

[0046] 图29b是本发明一个实施例中提供的单机版数据安全存储和读取装置的结构示意图；

[0047] 图30是本发明一个实施例中提供的单机版数据黑洞处理方法；

[0048] 图31是本发明一个实施例中提供的使用移动存储器进行安全存储的示意图；

[0049] 图32是本发明一个实施例中提供的移动存储设备的层次结构示意图。

具体实施方式

[0050] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图，对本发明进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0051] 分析

[0052] 如图1所示为现有技术中计算设备的系统层次示意图，从上至下，计算设备包括：

[0053] 用户界面层101，应用层102，操作系统内核层103，硬件映射层104以及硬件层105。

[0054] 其中，用户界面层101是用户与设备之间的接口，用户通过该层与设备(即设备的

其他层次，例如应用层102)进行交互。应用层102指应用软件层。

[0055] 操作系统内核层103是一种基于软件的逻辑层，一般来讲是由软件数据和软件代码组成，相比于界面层101和应用层102，操作系统内核层103的代码拥有更高的权限，可以对计算机系统中的各种软硬件资源进行完整的操作。

[0056] 硬件映射层104是一种基于软件的逻辑层，它一般工作在操作系统内核层，拥有与内核层相同的权限。硬件映射层主要是为了解决将不同类型的硬件的操作模式映射为一种统一的上层接口，向上屏蔽硬件的特殊性。一般来说，硬件映射层主要被操作系统内核层103使用，来完成对各种硬件的操作。

[0057] 硬件层105是指构成计算机系统的所有硬件部件。

[0058] 对于上述计算设备的系统层次的工作过程，下面以保存数据的操作为例进行说明，包括：

[0059] (1)用户通过某应用程序提供的用户界面101，选择执行“保存”功能；

[0060] (2)应用层102调用对应代码，将上述用户操作转化为一个或多个操作系统提供的接口函数(例如，Microsoft 32位平台的应用程序编程接口，win32 API)，即“保存”操作转化成为对一系列操作系统内核层103提供的接口函数的调用；

[0061] (3)操作系统内核层103将每一个操作系统接口函数转化为一个或多个硬件映射层104提供的接口函数；即“保存”操作转化成为对一系列硬件映射层104提供的接口函数的调用；

[0062] (4)硬件映射层104将每一个自己提供的接口函数转化为一个或多个硬件指令调用；最后，

[0063] (5)硬件层105(例如CPU)接收上述硬件指令调用并执行硬件指令。

[0064] 针对该计算设备，当其被恶意代码侵入后，恶意代码可以从计算设备中取得所需数据，窃取数据后其行为模式包括：

[0065] (1)存储行为：将目标数据内容保存到某个存储位置；

[0066] (2)传输行为：将窃取的数据直接通过网络传输到指定的目标地址。

[0067] 另外，使用上述计算设备或信息设备的人员进行内部泄密的行为模式包括：

[0068] (1)主动泄密：涉密人员通过主动拷贝、通过恶意工具穿透安全系统、置入木马等手段直接取得涉密数据，并进行泄密；

[0069] (2)被动泄密：涉密人员使用的电脑或存储介质因保管不善丢失或使用不当(例如将涉密装备直接接入Internet)造成的泄密。

[0070] 上述多种泄密方式使得该计算设备的数据安全无法保障。

[0071] 发明人经研究发现，计算机运行过程中，一CPU地址寄存器用于保存下一条将要运行的机器指令的地址，例如pc(program counter，程序计数器)。获取该寄存器中的数据，并按照该数据指向的地址，读取下一条或者多条将要运行的机器指令，可以实现运行时捕获机器指令的目的。

[0072] 并且，通过修改所述一条或多条机器指令所组成的待调度指令片段(例如在其中插入额外的程序转移指令，本文称为指令重组)，使得在该段指令运行完毕之前重新获得CPU执行权，并再次捕获下一个待调度指令片段，可以实现运行时连续捕获机器指令的目的。

[0073] 并且，在获取到待调度指令片段后，还可以对其中的机器指令进行分析以及处理，从而不仅可以实现运行时指令捕获、重组，还可以实现对预定的目标指令的管理。

[0074] 指令重组或指令追踪

[0075] 基于上述分析和发现，本发明的一个实施例中提供了一种运行时指令重组方法，该方法运行时称为指令重组平台。如图2所示，该方法S100包括：

[0076] S101，缓存指令运行环境；所述指令运行环境包括地址寄存器，地址寄存器保存下一条将要运行的机器指令的地址，该地址为第一地址；

[0077] S102，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令(例如第一跳转指令)；

[0078] S103，在所述第一程序转移指令前，插入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址，即执行该第二程序转移指令后，执行步骤S101；

[0079] S104，将所述地址寄存器中的第一地址修改为第二地址；和

[0080] S105，恢复所述指令运行环境。

[0081] 其中，在步骤S101中，所述缓存指令运行环境可以包括：

[0082] 向缓存栈中压入CPU机器指令运行相关的寄存器数据。

[0083] 在本发明的其他实施例中，缓存或保存指令运行环境也可以在指定的、默认的其他缓存数据结构和地址中进行。

[0084] 在步骤S101中，所述地址寄存器为程序计数器即PC。

[0085] 在步骤S102中，待调度的机器指令片段中只有一条程序转移指令，待调度的机器指令片段包括所述第一程序转移指令以及其之前的所有待调度的机器指令。

[0086] 在步骤S103中，在所述待调度的机器指令片段的最后一条指令(即第一程序转移指令，简称JP1)前，插入第二程序转移指令(简称JP2)，所述JP2指向指令重组平台的入口地址，生成具有第二地址(该地址以A″表示)的重组指令片段。

[0087] 插入第二程序转移指令是为了在CPU运行所述待调度的机器指令片段时，在JP1运行前，重新开始运行所述指令重组平台，这样，指令重组平台就可以继续分析下一段待调度的机器指令片段，从而通过重复本方法来完成对所有运行时指令的重组。

[0088] 在步骤S105中，恢复所述指令运行环境可以包括：

[0089] 从缓存栈中弹出指令运行相关的寄存器数据；其中地址寄存器保存的程序转移指令的目标地址已经修改为以第二地址A″为入口地址的新的机器指令片段。

[0090] 步骤S105执行后，恢复了所述指令运行环境，指令重组平台完成一次运行，CPU执行所述重组指令片段，即CPU将执行以第二地址A″为入口地址的机器指令片段。重组指令片段执行到第二程序转移指令JP2时，所述指令重组平台重新得到CPU控制权(即执行步骤S101)，此时第一程序转移指令的目标地址已经得到，该目标地址为新的第一地址，继而重新执行步骤S101～步骤S105。

[0091] 在本实施例中，上述运行时指令重组方法在X86架构的CPU上执行；在本发明的其他实施例中，上述运行时指令重组方法也可以在MIPS处理器或基于ARM架构的处理器上执行。本领域普通技术人员可以理解，上述方法可以在计算设备中的任何其他类型的指令处理单元上执行。

[0092] 下面结合图3，进一步说明指令重组过程和重组指令片段的生成过程。

[0093] 图3中包括待调度的机器指令集合401(例如已经载入内存中的某程序的机器指令)，其中指令4012为第一程序转移指令，如果指令4012的目标地址为变量，则首先假设指令4012指向机器指令4013；从第一程序转移指令4012以前的包括第一程序转移指令4012的所有待调度的机器指令构成了机器指令片段4011(只包含一个程序转移指令)。

[0094] 当指令重组方法运行后(成为指令重组平台411)，首先缓存指令运行环境；然后获取(例如拷贝)机器指令片段4011；指令重组平台在第一程序转移指令4012前插入了第二程序转移指令4113，第二程序转移指令4113指向指令重组平台411本身，从而生成了重组指令片段4111，重组指令片段的地址为A″；将所述缓存的指令运行环境中的地址寄存器的值A修改为地址A″；最后恢复所述指令运行环境。

[0095] 指令重组平台411结束运行后，CPU执行以A″为地址的重组指令片段，当执行到第二程序转移指令4113时，指令重组平台411会重新获得CPU控制权。此时，第一程序转移指令4012的目标地址4013已经生成，该目标地址为新的第一地址，指令重组平台根据该目标地址重新开始执行步骤S101～步骤S105，继续分析后续的待调度的机器指令，从而完成了运行时指令重组的方法。

[0096] 根据本发明另一个实施例，如图4所示，在步骤S102中，获取待调度的机器指令片段可以包括：

[0097] S1021，从地址寄存器(例如程序计数器)读取待调度的机器指令地址；

[0098] S1022，以程序转移指令(例如跳转指令)为检索目标，检索所述机器指令地址指向的机器指令及其后续指令，直到发现第一个程序转移指令(称为第一程序转移指令，例如第一跳转指令)；所述程序转移指令指能够改变机器指令顺序执行流程的机器指令，包括Jump程序转移指令、Call调用指令、Return返回指令等；

[0099] S1023，将所述第一程序转移指令以及其之前的所有待调度的机器指令作为一个待调度的机器指令片段，将该机器指令片段保存在指令重组平台中，或其他指令重组平台能够读取的存储位置。

[0100] 在本发明的其他实施例中，获取待调度的机器指令片段也可以以非程序转移指令(例如写入指令、读取指令等)为检索目标，进一步切分机器指令片段。由于在这样的实施例中，也需要保证在调度程序转移指令执行后指令重组平台仍能够获取CPU控制权或执行权，所以程序转移指令需要作为第二检索目标，从而得到粒度更小的机器指令片段。

[0101] 根据本发明另一个实施例，在步骤S102和S103之间，所述运行时指令重组方法还可以包括：

[0102] 利用指令集匹配所述待调度的机器指令片段，得到目标机器指令；所述指令集包括X86，MIPS和ARM指令集；和

[0103] 按照预定的方式，修改所述目标机器指令。

[0104] 不仅可以完成运行时指令监控，还可以进行其他处理过程，相关实施例将在后面详细介绍。

[0105] 进一步的，为了提高指令重组方法的效率，可以将固定地址程序转移指令所指向的待调度指令在步骤S102中一并获取。

[0106] 根据本发明另一个实施例，提供一种运行时指令重组方法，该方法S300包括：

[0107] S301，缓存指令运行环境；所述指令运行环境包括地址寄存器，地址寄存器保存下一条将要运行的机器指令的地址，该地址为第一地址；

[0108] S302，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令，该程序转移指令为参数地址程序转移指令；

[0109] S303，在所述第一程序转移指令前，插入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址，即执行该第二程序转移指令后，执行步骤S301；

[0110] S304，将所述地址寄存器中的第一地址修改为第二地址；

[0111] S305，恢复所述指令运行环境。

[0112] 与之前的实施例中所提供的方法相比，区别在于：在步骤S302中，待调度的机器指令片段中可以包括多条程序转移指令；并且这些程序转移指令中只有一条参数地址程序转移指令，称为第一程序转移指令。

[0113] 需要说明的是，程序转移指令可以包括两类，参数地址程序转移指令和常数地址程序转移指令，其中，常数地址程序转移指令的跳转地址为常数(即立即数)，而参数地址程序转移指令中的参数地址一般在程序转移指令之前的一条机器指令中计算得到。

[0114] 相似地，待调度的机器指令片段的最后一条指令为第一程序转移指令；待调度的机器指令片段包括所述第一程序转移指令以及其之前的所有待调度的机器指令。

[0115] 进一步的，由于程序运行过程中所生成的机器指令具有很高的重复性，为了提高指令重组方法的效率，节省计算设备的计算资源(例如CPU资源)，可以利用少量的存储空间来保存重组指令片段。

[0116] 根据本发明另一个实施例，提供一种运行时指令重组方法。如图5所示，该方法S200包括：

[0117] S201，缓存指令运行环境；所述指令运行环境包括地址寄存器(例如程序计数器)，地址寄存器保存下一条将要运行的机器指令的地址，该地址称为第一地址；一般来说，指令运行环境包括CPU的所有寄存器，包括通用寄存器、状态寄存器、地址寄存器等；

[0118] S202，利用第一地址查找地址对应表；所述地址对应表用于表示第一地址(例如地址A)指向的待调度指令片段是否具有已保存的重组指令片段，地址对应表的数据可以为地址对，也可以以其他形式存储相关数据；

[0119] S203，如果找到相应的记录，将所述第一地址A(即地址寄存器的值A)修改为已保存的重组指令片段的地址(例如地址A')；

[0120] S204，如果没有找到相应的记录，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令(例如第一跳转指令)；

[0121] S205，在所述第一程序转移指令前，插入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址，即执行该第二程序转移指令后，执行步骤S201；

[0122] S206，将所述地址寄存器中的第一地址修改为第二地址；

[0123] S207，恢复所述指令运行环境。

[0124] 进一步的，步骤S206还包括：利用第二地址A"与第一地址A在所述地址对应表中建立地址对(或一条记录)。具有地址A"的重组指令片段被保存在重组指令平台中或重组指令

平台能够访问的存储器中，以供重用。

[0125] 本方法利用地址对应表，节省计算资源，提高运行时指令重组的效率。

[0126] 上述重组方法一般通过在待调度指令片段之中插入所需程序转移指令完成，在本发明其他实施例中，也可以通过其他方式完成重组指令片段的生成。下面将结合实施例详细介绍。

[0127] 根据本发明另一个实施例，提供一种指令重组方法，单独开辟存储位置保存第一程序转移指令的目标地址。如图6所示，该方法S110包括：

[0128] S111，缓存指令运行环境；

[0129] S112，从第一存储位置读取目标地址，根据目标地址获取待调度(即待执行)的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令(例如第一跳转指令)；

[0130] S113，在第一存储位置保存第一程序转移指令的目标地址；

[0131] S114，将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址，即执行该第二程序转移指令后，执行步骤S111；

[0132] S115，恢复所述指令运行环境，并跳转到第二地址继续执行。

[0133] 其中，在步骤S112中，获取待调度的机器指令片段包括：

[0134] S1121、以程序转移指令为检索目标，检索所述机器指令地址指向的机器指令及其后续指令，直到发现第一个程序转移指令(称为第一程序转移指令)；

[0135] S1122、将所述第一程序转移指令以及其之前的所有待调度的机器指令作为一个待调度的机器指令片段，将该机器指令片段保存在指令重组平台中或其他指令重组平台能够读取的存储位置。

[0136] 在步骤S113中，目标地址即程序转移指令的目标地址参数，其可以是立即数或变量参数，对于立即数保存其值，对于变量参数保存其地址/引用。当处理器即将执行某程序转移指令时，其跳转目标地址已经计算完毕。

[0137] 根据本发明另一个实施例，提供一种指令重组方法，针对非固定长度指令集进行反汇编和汇编处理。如图7所示，该方法包括：

[0138] S121，缓存指令运行环境；

[0139] S122，从第一存储位置读取目标地址，根据目标地址获取待调度指令片段，包括：

[0140] 从目标地址开始，获取待调度的一段机器指令，将该段机器指令进行反汇编，并将反汇编结果通过一个词法分析器进行处理并匹配是否其中包含程序转移指令(例如跳转指令)，如果不包含则继续获取下一段待调度的机器指令重复上述操作，直到匹配到程序转移指令为止，该程序转移指令为第一程序转移指令；第一程序转移指令以及之前的所有指令组成待调度指令片段；

[0141] 其中，第一存储位置用于保存下一条将要运行的机器指令的地址；

[0142] S123，在第一存储位置保存第一程序转移指令的目标地址；

[0143] S124，将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；本实施例中，该第一程序转移指令和第二程序转移指令皆为汇编指令；

[0144] S125，将生成的重组后的汇编代码通过汇编器生成对应的机器码；和

[0145] S126，恢复所述指令运行环境，并跳转到第二地址继续执行。

[0146] 根据本发明另一个实施例，提供一种指令重组方法，以压栈指令替代或记录第一程序转移指令。如图8所示，该方法S130包括：

[0147] S131，缓存指令运行环境；

[0148] S132，执行出栈操作获取操作数，计算下一条即将运行的指令地址，该地址为第一地址；其中，栈用于保存程序转移指令(例如跳转指令)的地址和参数；

[0149] S133，根据第一地址获取待调度/执行的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

[0150] S134，替换第一程序转移指令为压栈指令，在压栈指令中记录第一程序转移指令的地址和参数；

[0151] S135，在压栈指令之后加入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；和

[0152] S136，恢复所述指令运行环境，并跳转到第二地址继续执行。

[0153] 本领域普通技术人员可以理解，上述各个实施例中提供的功能或特征可以根据实际的需要叠加在同一个实施例中，这里就不再一一组合给出，下面只举一个例子进行示例性说明。

[0154] 根据本发明另一个实施例，提供一种指令重组方法，如图9a所示，包括：

[0155] (1)缓存指令运行环境，所述指令运行环境包括全部的CPU寄存器；

[0156] 执行出栈操作获取操作数，计算下一条即将运行的指令地址(称为第零地址)，设置第一地址的值为第零地址；其中，栈用于保存程序转移指令的地址和参数；

[0157] (2)利用第一地址来查找地址对应表(也称为地址查找表)，如果找到记录，恢复所缓存的指令运行环境，并跳转到找到的对应地址(例如地址对应表中的地址对)继续执行；

[0158] (3)如果没有找到记录，从第一地址开始获取待执行的机器指令片段，指令片段的结尾为程序转移指令(程序转移指令所在地址为第三地址)；

[0159] (4)从第一地址开始，将机器码进行反汇编，并将反汇编结果通过一个词法分析器进行处理，生成重组后的汇编代码，直到第三地址为止；

[0160] (5)判断第三地址处的代码是否可以进一步处理，即第三地址处的程序转移指令的目标地址为已知量(例如，立即数)，如果可以，将第一地址的值设置为第三地址的目标地址，重新开始执行(3)；

[0161] (6)如果不可以，在生成的重组后的汇编代码最后，加入压栈指令记录当前第三地址的原始地址位置(即第三地址的值)和操作数，并在压栈指令之后加入跳转至重组平台开始的指令，即能够使步骤(1)再次开始执行；

[0162] (7)将生成的重组后的汇编代码通过汇编器生成对应的机器码，并存储于重组地址空间中分配出的地址(第二地址)，并将第二地址和第零地址以对应地址对的形式存储于地址对应表中；

[0163] (8)恢复环境，并跳转到第二地址继续执行。

[0164] 为了方便理解，现以X86体系处理器运行该实施例提供的方法进行说明，参考图9b-9d，指令重组的一个示例过程如下：

[0165] (1)重组平台开始工作后,首先缓存当前指令运行环境;获取栈中保存的程序转移指令的地址和参数,计算下一条即将运行的指令地址,该地址为第零地址,将第一地址的值设置为第零地址。

[0166] (2)利用第一地址来查找地址对应表,如果找到记录,恢复所缓存的指令运行环境,并跳转到找到的对应地址继续执行(图9b);如果没有找到记录,进行如下操作(图9c)。

[0167] (3)-(6)从第一地址开始,将机器码进行反汇编,并将反汇编结果通过一个词法分析器进行处理,生成重组代码;

[0168] 对该段汇编代码进行检索,检查是否包含程序转移指令;

[0169] 对第一个程序转移指令进行分析,判断其跳转目标地址是否为已知量,如果是已知量,则继续寻找,直到找到第一条参数地址程序转移指令,称为第一程序转移指令,该指令的地址为第三地址;

[0170] 在生成的汇编代码(从第一地址到第三地址的机器指令,不包括第一程序转移指令)最后加入压栈指令记录当前第三地址的第一跳转的原始地址位置和操作数;

[0171] 在压栈指令之后加入跳转至重组平台开始的指令(第二程序转移指令)。

[0172] (7)将生成的汇编代码通过汇编器生成对应的机器码,并存储于重组地址空间中分配出的地址(第二地址);

[0173] 将第二地址和第零地址以对应地址对的形式存储于地址对应表中。

[0174] (8)恢复环境,并跳转到第二地址继续执行

[0175] (图9d)处理器开始执行第二地址的指令,之前的待重组指令片段中的程序转移指令已经替换为压栈指令和跳转去重组平台的指令,压栈指令主要的目的是向重组平台提供输入参数。(图9d)当执行到第二程序转移指令时,重组平台重新得到执行,进行上述的步骤(1),通过查看压栈指令中保存的程序转移指令的地址和参数,计算下一条即将运行的指令地址,该地址为第一地址。

[0176] 之后的处理即上述过程的循环。

[0177] 进一步的,为了从系统启动后即执行运行时的指令监控,实现计算设备运行阶段的运行时指令全监控,本发明另一个实施例中,修改计算机启动时的load指令,在原load指令执行前调用本发明提供的指令重组平台,执行上述运行时指令重组方法,由于load指令跳转地址为已知的固定地址,指令重组平台可以事先建立好地址对应表及第一条记录,并建立好第一个重组指令片段。

[0178] 进一步的,根据本发明另一个实施例,提供一种计算机可读介质,其中,所述可读介质中存储有计算机可执行的程序代码,所述程序代码用于执行上述实施例中提供的运行时指令重组方法的步骤。

[0179] 进一步的,根据本发明另一个实施例,提供一种计算机程序,其中,所述计算机程序包含上述实施例中提供的运行时指令重组方法的步骤。

[0180] 针对数据安全的指令重组

[0181] 上述的运行时指令重组方法为进一步的应用提供了基础。下面的实施例中提供了各种针对不同机器指令进行处理的运行时指令重组方法,其中包括:存储/读取指令、I/O指令以及网络传输指令:

[0182] (1)存储/读取指令指计算机系统中所有对外部存储设备(包括但不限于磁盘存储

设备、闪存设备、光存储设备)进行存储/读取的指令或指令组合。

[0183] (2)I/O指令指计算机系统中所有操作外设的地址空间的指令,这些指令最终会影响外设输入输出状态、数据、信号等。外设的地址空间包括但不限于I/O地址空间、内存映射I/O设备地址空间。

[0184] (3)网络传输指令指计算机系统中所有影响网络设备的指令,这些指令最终会影响计算机系统网络设备的传输、状态、数据、信号等所有相关特性。

[0185] 其中,存储/读取指令与I/O指令之间可以存在交集。

[0186] 根据本发明一个实施例,提供一种针对存储/读取指令的运行时指令重组方法S400,包括:

[0187] S401,缓存指令运行环境;所述指令运行环境包括地址寄存器,地址寄存器保存下一条将要运行的机器指令的地址,该地址为第一地址;地址寄存器例如为程序计数器PC;

[0188] S402,利用所述第一地址查找地址对应表;

[0189] S403,如果找到相应的记录,将所述第一地址A修改为已保存的重组指令片段的地址A';

[0190] S404,如果没有找到相应的记录,重组指令片段的生成方法包括:

[0191] S4041,获取待调度的机器指令片段;其中,待调度的机器指令片段的最后一条指令为第一程序转移指令;与步骤S102相同;

[0192] S4042,反汇编所述待调度的机器指令片段,得到汇编指令片段;

[0193] S4043,检索目标汇编指令(即用目标汇编指令作为检索目标,检索汇编指令片段),所述目标汇编指令为存储/读取指令;

[0194] S4044,如果检索得到所述汇编指令片段中的存储/读取指令,修改其中的存储和读取地址为安全存储设备上的地址;修改方式可以为本地地址空间和安全存储设备地址空间之间的直接映射;

[0195] S4045,在所述第一程序转移指令JP1前,插入第二程序转移指令JP2,所述JP2指向指令重组平台(指令重组方法运行时称为指令重组平台,也可以理解为指令重组方法运行时的实例称为指令重组平台)的入口地址;

[0196] S4046,汇编修改过的汇编指令片段,生成具有地址A"的重组机器指令片段;

[0197] S4047,利用重组机器指令片段地址A"与第一地址A在所述地址对应表中建立一条记录(或地址对),具有地址A"的重组指令片段被保存在重组指令平台中;

[0198] S4048,将第一地址A修改为第二地址A";

[0199] S405,恢复所述指令运行环境。

[0200] 本实施例是在反汇编步骤之后进行指令处理的;在其他实施例中,也可以省略反汇编和对应的汇编步骤,直接处理机器指令。

[0201] 在步骤S4044中,针对存储和读取指令进行操作,修改其中的目标和源地址,以实现存储重定位/重定向,确保数据安全。更具体的安全存储/读取的方法将在本发明提供的下面的实施例中介绍。

[0202] 根据本发明一个实施例,提供一种针对I/O指令的运行时指令重组方法S500,包括:

[0203] S501,缓存指令运行环境;所述指令运行环境包括地址寄存器,地址寄存器保存下

一条将要运行的机器指令的地址，该地址为第一地址；

[0204] S502，利用所述第一地址查找地址对应表；

[0205] S503，如果找到相应的记录，将所述第一地址A修改为已保存的重组指令片段的地址A'；

[0206] S504，如果没有找到相应的记录，重组指令片段的生成方法包括：

[0207] S5041，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；与步骤S102相同；

[0208] S5042，反汇编所述机器指令片段，得到汇编指令片段；

[0209] S5043，检索目标汇编指令，所述目标汇编指令为I/O指令；

[0210] S5044，如果检索得到所述汇编指令片段中的I/O指令，将所述I/O指令中的输入指令全部阻止；

[0211] S5045，在所述第一程序转移指令JP1前，插入第二程序转移指令JP2，所述JP2指向指令重组平台的入口地址；

[0212] S5046，汇编修改过的汇编指令片段，生成具有地址A"的重组机器指令片段；

[0213] S5047，利用重组机器指令片段地址A"与第一地址A在所述地址对应表中建立一条记录(或地址对)，具有地址A"的重组指令片段被保存在重组指令平台中；

[0214] S5048，将第一地址A修改为第二地址A"；

[0215] S505，恢复所述指令运行环境。

[0216] 本实施例是在反汇编步骤之后进行指令处理的；在其他实施例中，也可以省略反汇编和对应的汇编步骤，直接处理机器指令。

[0217] 在步骤S5044中，针对I/O指令进行操作，将所述I/O指令中的输入指令全部阻止，以实现彻底阻断对本地硬件设备的写操作；结合上一个实施例中的存储指令处理过程，还可以实现对除存储指令之外的输入指令的阻止，可以提高计算设备中的数据安全性。

[0218] 根据本发明一个实施例，提供一种针对网络传输指令的运行时指令重组方法S600，包括：

[0219] S601，缓存指令运行环境；所述指令运行环境包括地址寄存器，地址寄存器保存下一条将要运行的机器指令的地址，该地址为第一地址；

[0220] S602，利用所述第一地址查找地址对应表；

[0221] S603，如果找到相应的记录，将所述第一地址A修改为已保存的重组指令片段的地址A'；

[0222] S604，如果没有找到相应的记录，重组指令片段的生成方法包括：

[0223] S6041，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；与步骤S102相同；

[0224] S6042，反汇编所述待调度的机器指令片段，得到汇编指令片段；

[0225] S6043，检索目标汇编指令，所述目标汇编指令为网络传输指令；

[0226] S6044，如果检索得到所述汇编指令片段中的网络传输指令，检验所述网络传输指令中的目标地址对应的远端计算设备是否为安全地址(例如白名单)，如果不是，阻止所述网络传输指令；

[0227] S6045，在所述第一程序转移指令JP1前，插入第二程序转移指令JP2，所述JP2指向

指令重组平台的入口地址；

[0228] S6046，汇编修改过的汇编指令片段，生成具有地址A″的重组机器指令片段；

[0229] S6047，利用重组机器指令片段地址A″与第一地址A在所述地址对应表中建立一条记录(或地址对)，具有地址A″的重组指令片段被保存在重组指令平台中；

[0230] S6048，将第一地址A修改为第二地址A″；

[0231] S605，恢复所述指令运行环境。

[0232] 在步骤S6044中，阻止/拒绝网络传输指令可以通过在重组后的代码中插入一到多条指令来将本身的传输指令替换为“取消当前操作的指令”或直接替换为无效指令，要视硬件的不同而定。

[0233] 本实施例是在反汇编步骤之后进行指令处理的；在其他实施例中，也可以省略反汇编和对应的汇编步骤，直接处理机器指令。

[0234] 在步骤S6044中，针对网络传输指令进行操作，检验所述网络传输指令中的目标地址对应的远端计算设备是否为安全地址；如果不是，阻止所述网络传输指令，以实现数据安全传输。

[0235] 上述多个实施例中的地址对应表是由指令重组平台建立并维护的，可以是固定长度的数组结构，也可以是可变长度的链表结构，还可以是其他存储二元数据的适当的数据结构。根据本发明一个实施例，其长度可调节，并且其占用空间可释放。释放地址对应表的操作可以随机进行，也可以周期进行。根据本发明一个实施例，地址对应表还可以包括记录建立时间字段，用于在释放空间删除记录时，按照建立时间的长短删除记录。根据本发明一个实施例，地址对应表还可以包括记录使用次数字段，在查找地址对应表步骤中，如果找到，将改变该字段的值；所述记录使用次数字段也用于在释放空间删除记录时，按照使用次数的多少删除记录。

[0236] 另外，本领域的技术人员可以理解，上述指令重组方法(即运行时指令重组方法)可使用软件或硬件的方法实现：

[0237] (1)如果以软件实现，则上述方法对应的步骤以软件代码的形式存储在计算机可读介质上，成为软件产品；

[0238] (2)如果以硬件实现，则上述方法对应的步骤以硬件代码(例如Verilog)描述，并固化(经过物理设计/布局布线/晶圆厂流片等过程)成为芯片产品(例如处理器产品)。下面将详细介绍。

[0239] 指令重组装置

[0240] 与上述运行时指令重组方法S100相对应，根据本发明一个实施例，提供一种运行时指令重组装置。如图10所示，指令重组装置500包括：

[0241] 指令运行环境缓存和恢复单元501，适于缓存和恢复指令运行环境；所述指令运行环境包括地址寄存器，该地址寄存器(例如程序计数器pc)保存下一条将要运行的机器指令的地址，该地址为第一地址；

[0242] 指令获取单元502，适于在单元501缓存指令运行环境后，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令(例如，第一跳转指令)；

[0243] 指令重组单元503，适于解析、修改所述待调度的机器指令片段，包括：在第一程序

转移指令前，插入第二程序转移指令，生成具有第二地址A″的重组指令片段；所述第二程序转移指令指向装置500，即执行该第二程序转移指令后，装置500的指令运行环境缓存和恢复单元501进行下一次处理；和

[0244] 地址替换单元504，适于将所述缓存的指令运行环境中的地址寄存器的值修改为重组指令片段的地址。

[0245] 所述指令运行环境缓存和恢复单元501分别与指令获取单元502以及地址替换单元504耦接，所述指令获取单元502，指令重组单元503和地址替换单元504依次耦接。

[0246] 装置500执行过程如下：

[0247] 首先，指令运行环境缓存和恢复单元501缓存指令运行环境，例如向缓存栈中压入指令运行相关的寄存器数据；

[0248] 然后，所述指令获取单元502从CPU地址寄存器511读取待调度的机器指令地址，并从所述机器指令地址读取机器指令片段，所述机器指令片段最后一条指令为程序转移指令；

[0249] 例如，指令获取单元502从CPU地址寄存器511读取待调度的机器指令地址；以程序转移指令为检索目标，检索所述机器指令地址对应的机器指令，直到发现第一个程序转移指令(即控制转移指令，包括无条件转移指令和条件转移指令)；所述程序转移指令包括例如Jump/JMP指令、Call指令、RET指令等；将所述第一个程序转移指令及其之前的所有机器指令作为一个待调度的机器指令片段；将该机器指令片段保存在装置500中或其他的装置500能够读取的存储位置；

[0250] 然后，指令重组单元503在所述获取的机器指令片段的最后一条指令前，插入第二程序转移指令，所述第二程序转移指令指向装置500的入口地址，生成具有地址A″的重组指令片段；

[0251] 然后，地址替换单元504将所述缓存的指令运行环境中的地址寄存器的值A修改为地址A″；

[0252] 最后，指令运行环境缓存和恢复单元501恢复所述指令运行环境，例如从缓存栈中弹出指令运行相关的寄存器数据。

[0253] 与上述运行时指令重组方法S300相对应，所述指令获取单元502可以将第一个非常数地址程序转移指令作为第一程序转移指令，以提高重组装置的执行效率。

[0254] 与上述运行时指令重组方法S200相对应，根据本发明另一个实施例，提供一种运行时指令重组装置，能够充分利用运行时指令重复性，提高效率，节省计算资源。

[0255] 如图11所示，指令重组装置600包括：

[0256] 指令运行环境缓存和恢复单元601，适于缓存和恢复指令运行环境；所述指令运行环境包括地址寄存器，地址寄存器保存下一条将要运行的机器指令的地址，该地址为第一地址；

[0257] 指令获取单元602，适于获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

[0258] 指令重组单元603，适于解析、修改所述待调度的机器指令片段，包括：在第一程序转移指令前插入第二程序转移指令，以生成具有第二地址的重组指令片段；所述第二程序转移指令指向装置600，即执行该第二程序转移指令后，装置600的指令运行环境缓存和恢

复单元601进行下一次处理；

[0259] 地址替换单元604，适于将所述缓存的指令运行环境中的地址寄存器的值修改为重组指令片段的地址；和

[0260] 指令检索单元605，适于利用所述第一地址查找地址对应表；所述地址对应表用于表示第一地址A指向的待调度指令片段是否具有已保存的重组指令片段，地址对应表的数据例如为地址对；

[0261] 如果找到相应的记录，指令检索单元605适于调用地址替换单元604，将所述第一地址A(即地址寄存器的值A)修改为已保存的重组指令片段的地址A'；如果没有找到相应的记录，指令检索单元适于利用第二地址A"与地址A在所述地址对应表中建立一条记录。

[0262] 所述指令运行环境缓存和恢复单元601分别与指令检索单元605以及地址替换单元604耦接，所述指令检索单元605分别与指令获取单元602，指令重组单元603和地址替换单元604耦接，所述指令获取单元602、指令重组单元603和地址替换单元604依次耦接。

[0263] 装置600的执行过程如下：

[0264] 首先，指令运行环境缓存和恢复单元601缓存指令运行环境，例如向缓存栈中压入指令运行相关的寄存器数据；

[0265] 然后，指令检索单元605利用所述缓存的指令运行环境中的地址寄存器的值A查找地址对应表；

[0266] 如果找到相应的记录，指令检索单元605调用地址替换单元604，地址替换单元604将所述地址寄存器的值A修改为记录中的值A'；地址替换单元604调用指令运行环境缓存和恢复单元601，以恢复所述指令运行环境，即从缓存栈中弹出指令运行相关的寄存器数据，本次重组操作结束；

[0267] 如果没有找到相应的记录，所述指令获取单元602从CPU地址寄存器读取待调度的机器指令地址，并从所述机器指令地址读取机器指令片段，所述机器指令片段最后一条指令为程序转移指令。具体的，指令获取单元602从CPU地址寄存器读取待调度的机器指令地址；以程序转移指令为检索目标，检索所述机器指令地址对应的机器指令，直到发现第一个程序转移指令；所述程序转移指令包括Jump指令和Call指令等；将所述第一个程序转移指令及其之前的所有机器指令作为一个待调度的机器指令片段；将该机器指令片段保存在装置600中，或其他的装置600能够读取的存储位置；

[0268] 然后，指令重组单元603在所述获取的机器指令片段的最后一条指令前，插入第二程序转移指令，所述第二程序转移指令指向装置600的入口地址，生成具有地址A"的重组指令片段；

[0269] 然后，指令重组单元603将地址A"发送给指令检索单元605，指令检索单元605利用地址A"与地址A在其中的地址对应表中建立一条记录；以备后续指令重用；

[0270] 然后，地址替换单元604将所述缓存的指令运行环境中的地址寄存器的值A修改为地址A"；

[0271] 最后，指令运行环境缓存和恢复单元601恢复所述指令运行环境，即从缓存栈中弹出指令运行相关的寄存器数据。

[0272] 继续参考图11，其中，指令重组单元603还可以包括：

[0273] 指令解析单元6031，适于利用指令集匹配所述机器指令片段，得到待处理的目标

机器指令(即利用目标指令检索待调度的机器指令片段);所述指令集包括X86,MIPS和ARM指令集;

[0274] 指令修改单元6032,适于按照预定的方式,修改所述目标机器指令。

[0275] 例如,如果所述目标指令为存储/读取指令,所述指令解析单元6031将负责获取待调度的机器指令片段中的存储/读取指令,所述指令修改单元6032修改其中的存储和读取地址为安全存储设备上的地址。其作用和效果与上述对应的方法实施例S400相同,这里不再赘述。

[0276] 又例如,如果所述目标指令为I/O指令,所述指令解析单元6031将负责获取待调度的机器指令片段中的I/O指令,所述指令修改单元6032将所述I/O指令中的输入指令全部阻止。其作用和效果与上述对应的方法实施例S500相同,这里不再赘述。

[0277] 又例如,如果所述目标指令为网络传输指令,所述指令解析单元6031将负责获取待调度的机器指令片段中的网络传输指令,所述指令修改单元6032检验所述网络传输指令中的目标地址对应的远端计算设备是否为安全地址;如果不是,所述指令修改单元适于阻止所述网络传输指令。其作用和效果与上述对应的方法实施例S600相同,这里不再赘述。

[0278] 根据本发明另一个实施例,上述指令重组单元还可以包括反汇编单元和汇编单元。如图12所示,指令重组单元703包括:依次耦接的反汇编单元7031、指令解析单元7032、指令修改单元7033和汇编单元7034。

[0279] 其中,反汇编单元7031适于在解析、修改所述待调度的机器指令片段之前,反汇编所述待调度的机器指令片段,生成待调度的汇编指令片段,发送给指令解析单元7032。

[0280] 汇编单元7034适于在解析、修改所述待调度的机器指令片段之后,汇编重组后的汇编指令片段,得到机器码表示的重组指令片段,发送给指令替换单元。

[0281] 在该实施例中,所述指令解析单元7032和指令修改单元7033将操作待调度的汇编指令片段,操作方法与上述实施例相似,这里不再赘述。

[0282] 与上述运行时指令重组方法S110相对应,根据本发明另一个实施例,提供一种运行时指令重组装置。如图13所示,指令重组装置800包括:

[0283] 指令运行环境缓存和恢复单元801,适于缓存指令运行环境;

[0284] 指令获取单元802和第一存储位置803,其中,指令获取单元802适于从第一存储位置803读取目标地址,并根据目标地址获取待调度/执行的机器指令片段;其中,待调度的机器指令片段的最后一条指令为第一程序转移指令;以及

[0285] 指令重组单元804,适于在第一存储位置803保存第一程序转移指令的目标地址,将第一程序转移指令替换为第二程序转移指令,生成具有第二地址的重组指令片段;所述第二程序转移指令指向装置800的入口地址。

[0286] 其中,指令运行环境缓存和恢复单元801还适于在指令重组单元804替换指令之后,恢复所述指令运行环境,并跳转到第二地址继续执行。

[0287] 装置800的执行过程如下:

[0288] 首先,指令运行环境缓存和恢复单元801缓存指令运行环境;

[0289] 然后,指令获取单元802从第一存储位置803读取目标地址(待调度指令地址),根据目标地址获取待调度的机器指令片段;其中,待调度的机器指令片段的最后一条指令为第一程序转移指令;

[0290] 然后，指令重组单元804在第一存储位置803保存第一程序转移指令的目标地址：(1)对于立即数保存其值，(2)对于变量参数保存其地址/引用，例如保存float类型变量destination_address的地址或引用；

[0291] 然后，指令重组单元804将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；

[0292] 最后，指令运行环境缓存和恢复单元801恢复所述指令运行环境，并跳转到第二地址继续执行。

[0293] 根据本发明另一个实施例，提供一种运行时指令重组装置，与上述方法S130相对应，并且包含上述某些实施例中提供的装置的特征。如图14所示，该装置900包括：

[0294] 指令运行环境缓存和恢复单元901，适于缓存和恢复指令运行环境；

[0295] 指令获取单元902，适于执行出栈操作获取操作数，并利用操作数计算下一条即将运行的指令地址，该地址为第一地址；

[0296] 还适于根据第一地址获取待调度/执行的机器指令片段，其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；

[0297] 指令重组单元903，适于替换第一程序转移指令为压栈指令，在压栈指令中记录第一程序转移指令的地址和操作数；

[0298] 还适于在压栈指令之后加入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向装置900的入口地址；

[0299] 还适于将重组指令片段的第二地址与第一地址在地址对应表中建立一条记录；

[0300] 指令检索单元904，适于利用所述第一地址查找地址对应表；所述地址对应表用于表示第一地址指向的待调度指令片段是否具有已保存的重组指令片段，地址对应表的数据为地址对；

[0301] 如果找到相应的记录，指令检索单元904适于调用指令运行环境缓存和恢复单元901恢复所缓存的指令运行环境，并跳转到找到的对应地址继续执行(本次重组操作完成)；

[0302] 如果没有找到相应的记录，调用指令重组单元903进行重组操作。

[0303] 其中，指令重组单元903还可以包括反汇编单元9031，指令解析单元9032，指令修改单元9033，和汇编单元9034。

[0304] 其中，当指令重组单元903完成重组后，适于调用指令运行环境缓存和恢复单元901恢复所缓存的指令运行环境，并跳转到重组指令片段的地址继续执行(本次重组操作完成)。

[0305] 根据本发明另一个实施例，上述反汇编单元9031可以位于指令获取单元902之中，在获取待调度的指令片段时由其进行反汇编操作。

[0306] 本领域技术人员可以理解，上述装置实施例的附图中的数据流的箭头只是为了便于解释上述实施例中的具体操作流程，并不限定图中各个单元之间的数据流向，装置中各个单元之间为耦接关系。

[0307] 上面通过一些实施例详细的介绍了运行时指令重组方法和装置，其与现有技术相比，具有以下优点：

[0308] 通过指令重组方法，可以在指令运行状态下监控计算设备的指令；

[0309] 利用地址对应表，提高了指令重组效率，节省了计算资源；

[0310] 针对存储和读取指令进行操作，修改其中的目标和源地址，以实现存储重定位/重定向，确保数据安全；

[0311] 针对I/O指令进行操作，将所述I/O指令中的输入指令全部阻止，以实现彻底阻断对本地硬件设备的写操作；还可以实现对除存储指令之外的输入指令的阻止，可以提高计算设备中的数据安全性；

[0312] 针对网络传输指令进行操作，检验所述网络传输指令中的目标地址对应的远端计算设备是否为安全地址；如果不是，阻止所述网络传输指令，以实现数据安全传输。

[0313] 数据安全存取过程

[0314] 图15是本发明一个实施例中计算设备的系统层次示意图。

[0315] 其中，计算设备(例如计算机终端系统)200包括：用户界面层201，应用层202，操作系统内核层203，硬件映射层204，安全层205，和硬件层206。

[0316] 其中，硬件层206进一步包括CPU 2061，硬盘2062(即本地存储设备)以及网卡2063。

[0317] 另外，计算设备200与存储设备10(又称为安全存储设备)耦接。

[0318] 本实施例中，存储设备10为远程磁盘阵列，通过网络连接硬件层206的网卡2063，与计算设备200交换数据。在本发明的其他实施例中，存储设备10也可以是其他已知或未知类型的存储设备。

[0319] 其中，硬盘2062也可以替换为其他类型的本地存储设备，例如u盘和光盘等，这里只是举例说明，并无限制目的。

[0320] 结合上述层次结构，本实施例提供一种数据安全存取过程，包括：

[0321] S1000，初始化；

[0322] S2000，数据写入；和

[0323] S3000，数据读取。

[0324] 参考图16，根据本发明一个实施例，上述的初始化过程S1000包括：

[0325] S1010，建立计算机终端系统200与安全存储设备10的通讯；

[0326] S1020，从安全存储设备10上同步一映射位图(Bitmap)至当前计算机终端系统200，例如保存在计算机终端系统200内存中；所述映射位图用于表示本地存储设备的数据是否已经转移存储到安全存储设备；

[0327] S1030，如果步骤S1020的同步操作失败，在安全存储设备10上建立Bitmap并初始化，然后同步到计算机终端系统200。

[0328] 其中，为了区分计算机终端200上的Bitmap与存储设备10上的Bitmap，下文中，除非另有说明，将计算机终端系统200上的Bitmap称为映射位图或第一映射位图，将安全存储设备10上的Bitmap称为第二映射位图(步骤S1030可以概括为先建立第二映射位图并初始化，然后再同步到计算机终端系统200保存为第一映射位图)。

[0329] 其中，在步骤S1020中，如果从存储设备10上同步第二映射位图至当前计算机终端系统200的操作失败，说明存储设备10与计算机终端系统200之间是第一次连接。

[0330] 其中，步骤S1030可以包括：将计算机终端系统200中的本地存储空间映射到存储设备10上，映射方法/关系为以1扇区(或其他存储的基本单位)为单位的一一映射，并且建立映射位图(Bitmap)。在本发明的其他实施例中，也可以使用其他基本容量为单位建立本

地存储空间到存储设备100上的Bitmap。对于Bitmap，下面将结合附图详细描述。

[0331] 图17为本发明一个实施例中的Bitmap示意图。图中包括本地存储设备(例如图15中的硬盘2062)上的存储介质3000，与本地存储设备通过网络连接的存储设备10上的存储介质4000。

[0332] (1)建立Bitmap的过程描述如下：

[0333] 在存储介质4000上建立与存储介质3000大小相同的存储空间4010，作为一一映射空间。在存储空间4010中保存Bitmap 4020，Bitmap 4020为一位图，其中1位代表1扇区，每一位的数据(0或1)标识/指示存储介质3000上的某扇区是否已经转储到存储介质4000上的存储空间4010，所以映射位图也可以称为转储表。存储设备10上的Bitmap 4020建立完成之后同步到计算机终端系统200中。

[0334] (2)更新Bitmap的过程描述如下：

[0335] 例如，在Bitmap 4020中，已经转储的扇区标记为1，未转储的扇区没有标记；在其他实施例中，转储扇区和非转储扇区所使用的标记可以自由选择。当应用程序或操作系统保存一个数据(例如文件时)，操作系统内部的文件系统将在本地存储设备的存储介质3000上开辟一定量的存储空间，例如扇区3040和扇区3050，并分配给该文件使用，并改写本地的文件分配表。该文件转储时(即写入扇区3040和扇区3050的数据被存储到存储设备10上时)，在存储介质4000上相同的位置分配扇区4040和4050，并在其中保存转储数据，并将Bitmap 4020中扇区3040和扇区3050对应的位数据改为1。

[0336] 结合附图15，根据本发明一个实施例，上述的数据写入过程S2000进一步包括：

[0337] S2010，应用层202通过操作系统内核层203的文件系统发出写文件操作请求，或操作系统内核层203直接发出写文件操作请求；或

[0338] 应用层202直接向硬件映射层204发出写数据操作请求，或操作系统内核层203直接向硬件映射层204发出写数据操作请求；

[0339] S2020，操作系统内核层203将写文件请求解析成硬件端口指令(即硬件指令)，下发至硬件映射层204，端口指令包含写入位置(例如扇区)；

[0340] 需要注意的是，如果步骤S2010是直接向硬件映射层204发出写数据操作请求，则该请求已经为硬件端口指令；

[0341] S2030，安全层205接收来自硬件映射层204的硬件端口指令，并且将端口指令中的写入位置(即扇区)改写为位于存储设备10上的对应存储地址，然后更新第一映射位图，例如将所述扇区对应的位数据修改为1，表示该扇区已经转储；安全层205将修改后的端口指令发送给硬件层206。

[0342] 写入过程执行完成之后，计算机终端系统200并没有存储写入的数据，相应的数据已经重定位存储在安全存储设备10上。

[0343] 需要注意的是，如果写本地硬盘指令本身与写网络硬盘指令不同，那么不仅需要改地址，还需要改存储指令。

[0344] 根据本发明另一个实施例，写入过程S2000还可以包括：

[0345] S2040，将第一映射位图同步到存储设备10上，保存为第二映射位图，从而确保计算机终端系统200上的第一映射位图与存储设备上的第二映射位图实时一致。

[0346] 在本发明的其他实施例中，为了节省系统资源，S2040也可以在本地的计算机终端

系统200关机前统一进行一次。

[0347] 结合附图15，根据本发明一个实施例，上述的数据读取过程S3000进一步包括：

[0348] S3010，将存储设备10上的第二映射位图同步到计算机终端系统200上，保存为第一映射位图；

[0349] S3020，应用层202通过操作系统内核层203的文件系统发出读文件操作请求，或操作系统内核层203直接发出读文件操作请求；或

[0350] 应用层202直接向硬件映射层204发出读数据操作请求，或操作系统内核层203直接向硬件映射层204发出读数据操作请求；

[0351] S3030，操作系统内核层203将读文件请求解析成硬件端口指令，下发至硬件映射层204，端口指令包含读取地址(例如扇区)；

[0352] S3040，安全层205接收来自硬件映射层204的数据读取指令，获取其中的读取地址(源地址)，查找第一映射位图，如果第一映射位图中的位数据表示所述读取地址为转储地址(数据已经转储)，安全层205修改端口指令的读取地址为存储设备10上的地址；安全层205将修改后的端口指令发送给硬件层206。

[0353] 本实施例的优点在于，上述读取过程没有影响用户既有的操作模式，实现了对于安全存储设备(即存储设备10)上已经转储的数据的读取。

[0354] 在步骤S3010中，从存储设备10同步第二映射位图到本地的过程是为了在计算机终端系统200重新启动了以后，保持本地数据与安全存储设备上的数据的一致性。

[0355] 本领域技术人员可以理解，对于上述的数据写入、读取过程以及初始化过程，可以根据实际需要执行所需步骤。

[0356] 数据安全存取方法

[0357] 基于上述数据写入过程和读取过程，下面详细描述本发明提供的数据安全存储和读取方法。

[0358] 本领域技术人员可以理解，上面结合图15来说明数据的读取和存储过程是为了方便理解，并不是限定，在本发明其他实施例中，可以在计算设备的适合层次上执行以上描述的各个步骤。

[0359] 根据本发明一个实施例，提供一种数据安全存储方法；如图18所示，该方法包括如下步骤：

[0360] S4010，接收硬件指令；

[0361] S4020，分析并判断该硬件指令是否为存储指令；

[0362] S4030，如果该硬件指令是存储指令，修改存储指令中的目标地址为对应的安全存储设备上的存储地址；

[0363] S4040，将修改后的存储指令发送到硬件层。

[0364] 根据本发明一个实施例，在步骤S4010中，所述硬件指令是来自硬件映射层的硬件指令。接收来自硬件映射层的硬件指令可以100％的筛查所有发送到CPU等处理器的硬件指令(接口指令)。

[0365] 计算机可以运行Windows操作系统，Windows系统中的硬件抽象层HAL为附图15中的硬件映射层204。在其他实施例中，计算机终端也可以运行其他操作系统，例如Linux，Unix或嵌入式操作系统等，硬件映射层为Linux、Unix或其他嵌入式操作系统中的对应层

次。

[0366] 在步骤S4010中，结合上述运行时指令重组方法，接收硬件指令的过程可以包括：采用运行时指令重组方法(例如S101-S105)获取硬件指令。换句话说，就是可以在运行时指令重组方法获取到机器指令时，处理存储指令(相似的方法例如S404，S504或S604)。通过运行时指令重组方法，可以不仅将计算最终结果重定位存储到安全存储设备，还能够将计算的中间过程(包括操作系统产生的中间过程)全部重定位存储到安全存储设备；通过这样的方式使终端计算设备不完整，并且进一步通过使终端计算设备不完整来达到信息防泄密的目的。

[0367] 另外，在步骤S4010和S4020中，硬件指令可以为X86指令、ARM指令、MIPS指令等类型，可以在终端计算设备内置分析机制，以处理不同类型的CPU指令。

[0368] 根据本发明另一个实施例，在步骤S4030之后，还可以包括：

[0369] S4050、更新第一映射位图，将目标地址(扇区)在第一映射位图中对应的“位”设置为转储标记，例如“1”；并且，将已经更新的映射位图同步到所述安全存储设备，保存为第二映射位图。

[0370] 本实施例中，转储操作对于上层应用以及用户完全透明，不影响现有计算机操作、应用系统的工作流程。

[0371] 本实施例提供的上述方法不仅可以在计算机终端系统中使用，还可以应用在任何包含应用层、操作系统内核层、硬件层的计算设备和智能终端上，实时实现指令级存储重定位/重定向(即基于硬件存储指令的存储重定位/重定向)。

[0372] 根据本发明一个实施例，提供一种数据安全读取方法；参考图19，该方法S5000包括：

[0373] S5010，接收硬件指令；

[0374] S5020，分析并判断该硬件指令是否为读取指令；

[0375] S5030，如果是读取指令，获取读取指令中的源地址(读取地址)，查找第一映射位图，并根据映射位图的数据修改读取指令中的读取地址，实现对转储数据和非转储数据的读取；和

[0376] S5040，将修改后的硬件指令发送到硬件层。

[0377] 根据本发明另一个实施例，在步骤S5010之前，该方法还可以包括：将存储设备上的第二映射位图同步到计算机终端系统200上，保存为第一映射位图。

[0378] 根据本发明另一个实施例，步骤S5010中，所述的硬件指令来自硬件映射层。

[0379] 根据本发明另一个实施例，在步骤S5010中，结合上述运行时指令重组方法，接收硬件指令的过程可以包括：采用运行时指令重组方法(例如S101-S105)获取硬件指令。换一种说法，就是可以在运行时指令重组方法获取到机器指令时，处理读取指令。

[0380] 根据本发明另一个实施例，在步骤S5020中，如果该硬件指令不是读取指令，则可以直接将硬件指令发送给硬件层去执行。

[0381] 根据本发明另一个实施例，步骤S5030还可以进一步分解为：

[0382] S5031，如果是读取指令，获取读取指令中的源地址，判断所述源地址是否为存储设备上的地址；

[0383] S5032，如果所述源地址不是存储设备上的地址，查找第一映射位图，并根据映射

位图的数据修改读取指令中的读取地址。

[0384] 即：在步骤S5031中，如果该读取指令的源地址已经为存储设备上的地址，则计算设备(例如图15中的安全层205)不用再次查找第一映射位图中的数据，可以直接将硬件指令发送给硬件层去执行。

[0385] 根据本发明另一个实施例，为了节约网络资源，在本发明的一些实施例中，安全存储设备10可以作为多个终端系统的共享资源。

[0386] 上面多次提到可以将数据安全存储和读取方法与指令重组方法结合，为了方便理解，下面通过实施例详细介绍。

[0387] 根据本发明一个实施例，提供一种数据安全存取方法。如图20所示，该方法S6000包括：

[0388] S6010，缓存指令运行环境；

[0389] S6011，从第一存储位置读取目标地址，根据目标地址获取待调度/执行的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令(例如第一跳转指令)；

[0390] S6012，在第一存储位置保存第一程序转移指令的目标地址；

[0391] S6013，分析并判断待调度机器指令中的每一条指令是否为存取指令；

[0392] S6014，如果是存取指令(包括存储指令和读取指令)：

[0393] 对于存储指令，修改存储指令中的目标地址为对应的存储设备(即安全存储设备)上的存储地址，并修改第一映射位图；

[0394] 对于读取指令，获取读取指令中的源地址，查找第一映射位图，并根据映射位图的数据修改读取指令中的读取地址；

[0395] 如果写本地硬盘指令本身与写网络硬盘指令不同，或者读取本地硬盘指令本身与读取网络硬盘指令不同，那么不仅需要修改地址，还需要相应的修改存储指令或读取指令；

[0396] S6015，将第一程序转移指令替换为第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向指令重组平台的入口地址；

[0397] S6016，恢复所述指令运行环境，并跳转到第二地址继续执行。

[0398] 本领域技术人员可以理解，该实施例只是为了说明而举例，并不限制安全读取方法、安全存储方法和指令重组方法的组合方式，上述介绍的各种安全读取方法、安全存储方法和指令重组方法可以以各种所需的方式组合使用。

[0399] 数据安全传输方法

[0400] 存储和读取一般是针对本地的存储设备进行的数据交换；传输一般是指通过网络设备进行的数据交换。

[0401] 如图21所示，根据本发明一个实施例，提供一种数据安全传输方法，包括：

[0402] S7010，接收(例如来自硬件映射层的)硬件指令；

[0403] S7020，分析并判断该硬件指令是否为网络传输指令；

[0404] S7030，如果该硬件指令是传输指令，读取目标地址；

[0405] S7040，判断目标地址是否为安全地址；

[0406] S7050，如果是安全地址，将硬件指令发送到硬件层；如果不是安全地址，拒绝该指令；

[0407] S7060，硬件层发送传输指令和数据到目标地址的终端系统；

[0408] S7070，目标地址的终端系统接收并利用数据安全存储方法(在上面实施例中描述)保存数据。

[0409] 根据本发明另一个实施例，在步骤S7040中，判断目标地址是否为安全地址的方法如下。

[0410] 参考图22，安全服务器820通过网络与终端系统800、810连接，终端系统800、810在部署本发明上述实施例中提供的数据安全传输方法时，都已经向安全服务器820进行了注册操作。安全服务器820内部维护一个安全地址表，记录了已经注册的所有终端系统。

[0411] 当安全地址表有更改的时候，安全服务器820自动将更新的安全地址表发送给各个终端，终端系统800的架构包括应用层801，操作系统内核层802，安全层803以及硬件层804，安全层803负责维护该安全地址表。

[0412] 安全层803将根据目标地址是否在安全地址表中，判断目标地址是否为安全地址。即在步骤S7040中，如果目标地址列入了安全地址表，则目标地址为安全地址。

[0413] 上述安全传输方法的实施，使木马或恶意工具即使取得了涉密信息也无法传输所取得的信息。

[0414] 虽然本发明一些实施例中以计算机终端系统作为应用本发明提供的方法的主体，但是，任何手持设备、智能终端等能够提供文件或数据编辑、保存或传输的电子设备，都可以成为应用本发明提供的数据安全存取及传输方法的载体。

[0415] 数据安全存取装置(包括存储、读取装置)

[0416] 与上述的数据安全存储方法相对应，根据本发明一个实施例，提供一种数据安全存储装置。

[0417] 需要注意的是，为了避免混淆，在本发明中：(1)数据安全存储装置指：以硬件形式来实现数据安全存储方法的装置；(2)安全存储设备指：用于转储信息或数据的存储实体，例如磁盘等。

[0418] 参考图23，本实施例提供的数据安全存储装置7100包括：接收单元7110，指令分析单元7120，指令修改单元7130和发送单元7140；所述接收单元7110与指令分析单元7120耦接，指令分析单元7120分别与指令修改单元7130以及发送单元7140耦接，发送单元7140还与指令修改单元7130耦接。

[0419] 其中，接收单元7110适于接收硬件指令，所述硬件指令可以来自硬件映射层；

[0420] 指令分析单元7120适于分析所述硬件指令并判断所述硬件指令是否为存储指令：如果是存储指令，指令分析单元7120还适于将其发送给指令修改单元7130，如果不是存储指令，指令分析单元7120还适于将其发送给发送单元7140；

[0421] 指令修改单元7130适于修改所述存储指令中的目标地址为对应的在安全存储设备上的存储地址，然后将修改后的存储指令发送给发送单元7140；

[0422] 发送单元7140适于将接收到的指令转发给硬件层7200。

[0423] 进一步的，根据本发明另一个实施例，该数据安全存储装置还可以包括：

[0424] 更新单元7150和同步单元7160，更新单元7150与指令修改单元7130耦接，同步单元7160与更新单元7150耦接。

[0425] 其中，更新单元7150适于在指令修改单元7130修改所述存储指令之后，更新映射

位图中所述目标地址对应的位。本实施例中，将存储指令目标地址包含的扇区在第一映射位图中对应的“位”数据置“1”，表示已经转储。

[0426] 其中，同步单元7160适于建立终端计算设备系统(即终端计算设备)与所述安全存储设备之间的通讯，并将映射位图在所述终端计算设备系统和所述安全存储设备之间进行同步。

[0427] 具体的，在终端计算设备系统启动时，同步单元7160建立终端计算设备系统与所述安全存储设备的通讯，并将所述安全存储设备上的第二映射位图同步到所述终端计算设备系统，保存为第一映射位图。

[0428] 如果将所述安全存储设备上的第二映射位图同步到所述终端计算设备系统失败，表示终端计算设备系统与安全存储设备是第一次建立连接并通讯，同步单元7160将计算机终端系统中的本地存储空间映射到所述安全存储设备上，并建立第一映射位图和第二映射位图。例如在本实施例中，先在安全存储设备上建立第二映射位图，然后同步到本地，保存为第一映射位图。

[0429] 当更新单元7150更新了第一映射位图(即映射位图)中所述目标地址对应的位，同步单元7160将把更新后的第一映射位图发送给安全存储设备，并在安全存储设备上保存为第二映射位图。

[0430] 所述安全存储设备的位置不限定，可以为远程存储设备或本地存储设备。所述远程存储设备可以只为一个计算设备服务，也可以被多个计算设备共享。

[0431] 根据本发明一个实施例，所述硬件指令可以为硬件端口I/O指令。

[0432] 与上述的数据安全读取方法相对应，根据本发明另一个实施例，提供一种数据安全读取装置，参考图24，数据安全读取装置8100包括：

[0433] 接收单元8110，指令分析单元8120，指令修改单元8130以及发送单元8140；其中，接收单元8110与指令分析单元8120耦接，指令分析单元8120分别与指令修改单元8130以及发送单元8140耦接，指令修改单元8130还与发送单元8140耦接。发送单元8140与硬件层8200耦接。

[0434] 所述接收单元8110适于接收硬件指令，本实施例中，所述硬件指令来自硬件映射层。

[0435] 所述指令分析单元8120适于分析所述硬件指令并判断所述硬件指令是否为读取指令，如果所述硬件指令是读取指令，获取读取指令的源地址并判断所述源地址是否为安全存储设备上的地址。

[0436] 如果所述硬件指令不是读取指令，或者所述源地址是安全存储设备上的地址，指令分析单元8120将所述硬件指令发送到发送单元8140。

[0437] 如果所述源地址不是安全存储设备上的地址，指令修改单元8130查找映射位图，并根据映射位图的数据修改所述读取指令中的读取地址。

[0438] 与上述实施例中的映射位图相同，本实施例中所述映射位图也用于表示本地存储地址的数据是否转储到所述安全存储设备，这里不再赘述。例如，指令修改单元8130查找源地址包含的扇区在第一映射位图中对应的位。如果“位”数据显示为1，表示已经发生转储，如果“位”数据显示为0或NULL(空)，表示没有发生转储。如果已经发生转储，指令修改单元8130将所述源地址(读取地址)改为对应的转储地址，并将修改后的硬件指令发送给发送单

元8140。

[0439] 进一步的，根据本发明另一个实施例，所述数据安全读取装置还可以包括同步单元8150，与指令修改单元8130耦接。

[0440] 同步单元8150适于建立终端计算设备系统与所述安全存储设备的通讯，并将映射位图在所述终端计算设备系统和所述安全存储设备之间进行同步。具体的，同步单元8150在终端计算设备系统启动时，建立终端计算设备系统与所述安全存储设备的通讯，并将所述安全存储设备上的第二映射位图同步到所述终端计算设备系统，保存为第一映射位图，提供指令修改单元8130使用。

[0441] 本实施例中，所述安全存储设备可以为远程存储设备，所述远程存储设备可以被多个终端计算设备系统共享。在本发明的其他实施例中，所述的安全存储设备也可以为本地存储设备。

[0442] 根据本发明另一个实施例，上述数据安全读取装置和数据安全存储装置可以合并为一个装置，其中指令分析单元和指令修改单元既能处理存储指令又能处理读取指令，下面举例进行详细说明。

[0443] 根据本发明另一个实施例，提供一种数据安全存储和读取装置。如图25，数据安全存储和读取装置(简称数据安全存取装置)9100包括：

[0444] 指令运行环境缓存和恢复单元9101，适于缓存和恢复指令运行环境；

[0445] 指令获取单元9102，适于获取下一条即将运行的指令地址，该地址为第一地址；还适于根据第一地址获取待调度/执行的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一程序转移指令；获取待调度的机器指令片段的具体方式在前面的实施例中已经详细描述，这里不再赘述；

[0446] 指令检索单元9104，适于利用所述第一地址查找地址对应表：

[0447] 如果找到相应的记录，指令检索单元9104适于调用指令运行环境缓存和恢复单元9101恢复所缓存的指令运行环境，并跳转到找到的对应地址继续执行(本次重组完成)；

[0448] 如果没有找到相应的记录，调用指令重组单元9103进行重组操作。

[0449] 其中，地址对应表用于表示第一地址指向的待调度指令片段是否具有已保存的重组指令片段，地址对应表的数据可以为地址对。

[0450] 其中，指令重组单元9103进一步包括：

[0451] 指令解析单元9111，是上述指令分析单元7120和指令分析单元8120的有机结合，适于分析所述硬件指令并判断所述待调度/执行的机器指令片段中的每一条硬件指令是否为存储或读取指令；

[0452] 指令修改单元9112，如果指令解析单元9111发现存储或读取指令，指令修改单元9112适于：

[0453] 对于存储指令，修改所述存储指令中的目标地址为对应的在安全存储设备上的存储地址；

[0454] 对于读取指令，查找映射位图，并根据映射位图的指示数据来修改所述读取指令中的读取地址；

[0455] 更新单元9113，适于在指令修改单元9112修改所述存储指令之后，更新映射位图中所述目标地址对应的位，以体现本地数据已经转储；

[0456] 同步单元9114，适于建立终端计算设备系统与所述安全存储设备的通讯，并将映射位图在所述终端计算设备系统和所述安全存储设备之间进行同步。

[0457] 在指令解析单元9111、指令修改单元9112、更新单元9113和同步单元9114操作完成后，指令重组单元9103适于替换第一程序转移指令为压栈指令，在压栈指令中记录第一程序转移指令的地址和操作数；还适于在压栈指令之后加入第二程序转移指令，生成具有第二地址的重组指令片段；所述第二程序转移指令指向装置9100的入口地址；还适于将重组指令片段的第二地址与第一地址在地址对应表中建立一条记录。

[0458] 根据本发明另一个实施例，如图26所示，指令重组单元9103与指令解析单元9111、指令修改单元9112、更新单元9113和同步单元9114作为同一层次的并列单元，其功能不再赘述。继续参考图25，指令重组单元9103获得重组指令片段后，还适于调用指令运行环境缓存和恢复单元9101恢复所缓存的指令运行环境，并跳转到重组指令片段的地址继续执行(重组操作完成)。

[0459] 本领域技术人员可以理解，该实施例只是为了说明而举例，并不限制数据安全读取装置、数据安全存储装置和指令重组装置合并方式，上述介绍的各种数据安全读取装置、数据安全存储装置和指令重组装置可以以各种所需的方式合并。

[0460] 另外，上述安全存储方法和装置还可以与云技术结合，确保云内数据的安全，从而加快云计算(cloud computing)的应用和普及。具体实施例在下面将给予介绍。

[0461] 本领域技术人员可以理解，在安全层实现的上述方法也可以在操作系统内核层至硬件层中的各个层内完成。具体功能的实现位置并不脱离本发明的精神和范围。

[0462] 上述实施例中详细的介绍了本发明提供的安全存储方法和装置，与现有技术相比，具有如下优点：

[0463] 1、数据安全存储方法实现了指令级数据转储即数据全转储，以此为基础，实现了终端计算设备系统全运行周期的数据安全存储方法，一方面，使木马或恶意工具即使取得了涉密信息也无法保存所取得的信息，使数据始终存在于可控的安全范围内；另一方面，本地不再保存在涉密状态下的任何数据，因此防止了涉密人员的主动泄密和被动泄密；

[0464] 2、接收来自硬件映射层的硬件指令可以100％的筛查所有指令，进一步提高数据安全性。

[0465] 上述实施例中还详细的介绍了本发明提供的安全读取方法和装置，与现有技术相比，具有如下优点：

[0466] 1、数据安全读取方法配合数据安全存储方法使数据始终存在于可控的安全范围内，并且保证在安全存储数据(转储)之后，可以将转储数据读出；由于本地将不再保存在涉密状态下的任何数据，因此防止了涉密人员的主动泄密和被动泄密；

[0467] 2、安全存储设备为远程存储设备时，可以为多个终端共享，提高安全存储设备的空间使用效率。

[0468] 数据黑洞处理方法

[0469] 定义：

[0470] 1、数据黑洞系统：是指将计算设备运行过程中的过程数据和运行结果存储至特定存储位置并且能够确保计算设备正常运行的系统；

[0471] 数据黑洞系统破坏了计算设备的完整性，并且通过破坏计算设备的完整性实现了

即使在恶意代码或涉密人员具有最高数据权限时也不会让数据泄密的数据安全系统。

[0472] 2、数据黑洞终端：是指部署了数据黑洞系统的计算设备(例如计算机终端)，数据黑洞终端将其运行过程中所产生的过程数据和结果数据全部转移存储至一特定的存储位置。

[0473] 3、重定向：指计算机在运行过程中所产生的过程数据或结果根据计算机运行要求进行持久化时，在不对计算机任何逻辑和代码进行修改的情况下，将持久化的位置定向至一个特定存储位的处理方法。

[0474] 4、数据写：一种数据持久化操作。

[0475] 5、数据黑洞空间：在下文中定义。

[0476] 6、黑洞存储区：在下文中定义。

[0477] 根据本发明一个实施例，提供一种提高数据安全性的过程A10，包括：

[0478] A11、为用户建立一个数据黑洞空间，包括两种模式(可以任选一种进行)：

[0479] A111 本地部署模式：数据黑洞终端在本地的数据存储设备上创建一个数据存储区，该数据存储区为终端数据重定向的目标区域，该数据存储区称为黑洞存储区；

[0480] 此数据存储区与用户的对应关系可以是一个数据存储区对应多个本机(或本地)用户，也可以是多个存储区对应多个本机(或本地)用户；

[0481] 该数据存储区只能由数据黑洞系统访问，不能被终端计算设备的操作系统或应用层(例如应用软件)访问；

[0482] A112 网络部署模式：在网络上的存储位置创建一个数据存储区，此数据存储区为终端数据重定向的目标区域；

[0483] 此数据存储区与网络终端上的用户的对应关系可以是一一对应关系；该存储区也可以对应本机(或本地)用户。

[0484] 经过上述本地部署模式或者网络部署模式部署，为用户建立了数据黑洞空间(简称黑洞空间)。

[0485] A12、建立用户与重定向存储空间之间的对应关系。

[0486] 当终端用户第一次登录数据黑洞终端时，数据黑洞终端将根据用户信息为其建立对应的数据黑洞的数据存储区。

[0487] A13、重定向终端计算设备所有的数据持久化操作。

[0488] 根据本发明一个实施例，用户登录到数据黑洞终端后，数据黑洞终端确定数据黑洞存储区存在并能建立用户与黑洞存储区之间的对应关系，该用户在本机(数据黑洞终端)上所有的数据写将被重定向至数据存储区。

[0489] 采用上述过程A10后，黑洞空间与用户对应，当黑客通过漏洞、后门、木马等恶意代码取得数据权限后将可以对数据进行复制、转储、发送、截留。但所有向外部设备、端口、用户、终端转发出的数据将被重定向到数据黑洞空间(与用户对应的黑洞空间)中，并在数据黑洞空间(与用户对应的黑洞空间)内完成。因此所有的数据窃取、截留、输出等作业都被在数据黑洞空间内实现。当涉密(有数据权限)人员试图将数据私自留存、私自备份、发送、输出时，所有的数据处理作业都在数据黑洞空间(与用户对应的黑洞空间)内完成，使恶意操作无法泄密。

[0490] 根据本发明一个实施例，如图27所示，能够执行上述过程A10的计算设备称为数据

黑洞服务器，数据黑洞服务器通过网络与计算终端1(图中显示为终端1)、计算终端2(图中显示为终端2)、…、计算终端N(图中显示为终端N)数据连接/耦接。数据黑洞服务器向各个终端部署数据黑洞系统，使各个终端成为数据黑洞终端(图中显示为数据黑洞终端1、数据黑洞终端2、…、数据黑洞终端N)。

[0491] 并且，黑洞存储区(图中显示为映射块1、映射块2、…、映射块N)位于数据黑洞服务器上(或服务器所连接的磁盘阵列服务器)。这样，数据黑洞空间包括数据黑洞服务器的黑洞存储区与各个数据黑洞终端的内存，从而，数据黑洞终端的计算过程数据和结果数据都会被存储到黑洞存储区中。数据黑洞系统破坏了计算设备的完整性，并且通过破坏计算设备的完整性实现了即使在恶意代码或涉密人员具有最高数据权限时也不会让数据泄密的数据安全系统。

[0492] 根据上述过程A10，根据本发明一个实施例，提供一种数据黑洞处理方法S90，如图28所示，包括：

[0493] S91，在计算设备(例如计算机、手持通信设备、智能终端等)部署数据黑洞系统，成为数据黑洞终端；

[0494] S92，建立数据黑洞空间，包括：

[0495] 1)在计算设备本地开辟一个数据存储区(称为黑洞存储区)，以及本地内存；和/或

[0496] 2)在网络一个存储位置开辟一个数据存储区(称为黑洞存储区)，以及本地内存；

[0497] S93，为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系，例如当用户登录数据黑洞终端，使终端用户与数据黑洞空间形成一一对应关系；

[0498] S94，数据黑洞终端将用户操作所产生的“数据写”重定向到与该用户对应的数据黑洞空间，例如重定向到与该用户对应的黑洞存储区；

[0499] S95，阻止对于本地存储设备的数据持久化操作，并且阻止通过本地端口对非数据黑洞终端的数据输出，从而保证进入数据黑洞终端或者数据黑洞空间的数据只在数据黑洞空间中存在。

[0500] 根据本发明的另一个实施例，步骤S91和S92的内容——在计算设备上部署黑洞系统和为用户建立数据黑洞空间可以在一个步骤内完成。

[0501] 根据本发明的另一个实施例，步骤S93可以只在用户第一次登陆黑洞终端时进行，也可以在用户每次登陆黑洞终端时进行。

[0502] 根据本发明的另一个实施例，步骤S93与步骤S94的内容可以在一个步骤中完成，即：

[0503] 当用户发生“数据写”时，按照预设的对应方式，将该用户的“数据写”全部重定向到与该用户对应的数据黑洞空间。

[0504] 其中，预设的对应方式可以包括固定对应，例如，每个用户在黑洞空间对应一定容量的存储空间。预设的对应方式可以包括动态对应，例如，每个用户在黑洞空间先对应预设容量的存储空间，如果用户存储数据超过该预设的容量，为用户分配更大的(例如为预设容量的2、4或8倍等)存储空间。本领域普通技术人员可以理解，用户与存储空间之间的对应方式和分配方式可以按需选择。

[0505] 根据本发明一个实施例，基于上述过程A10，用户登录到数据黑洞终端后，数据黑洞终端确定数据黑洞存储区存在并能建立用户与黑洞存储区之间的对应关系，该用户在本

机(数据黑洞终端)上所有的数据写将被重定向至数据存储区。并且,所有的数据读将根据数据的版本或由用户自行选择读取存储区数据或本机(或本地)数据。

[0506] 根据上述实施例中提供的数据安全读取方法(例如S5000)和装置(数据安全读取装置8100),为了提供用户选择功能,可以做适应性修改。

[0507] 根据本发明一个实施例,提供一种数据安全读取方法S80包括:

[0508] S81,接收硬件指令;

[0509] S82,分析并判断该硬件指令是否为读取指令;

[0510] S83,如果是读取指令,根据映射位图的知识数据的值,如果欲读取的数据已经被转储,则:

[0511] 为用户提供选择操作机会,让用户选择是读取存储区数据还是读取本机(或本地)数据;

[0512] 根据用户的选择来读取存储区数据或本机(或本地)数据,即如果用户选择读取存储区域;

[0513] S84,将修改后的硬件指令发送到硬件层。

[0514] 上述数据安全读取方法S80的其他方面和步骤可以参考数据安全读取方法S5000,这里不再赘述。

[0515] 同理,本实施例中的数据安全读取装置可以适应性修改,例如,将数据安全读取装置8100中的指令修改单元8130修改为还适于执行S83的操作,其他单元可以参考数据安全读取装置8100,这里不再赘述。

[0516] 单机版数据黑洞处理方法

[0517] 在上述步骤S92中,当建立数据黑洞空间为在计算设备本地开辟一个数据存储区(称为黑洞存储区),则该计算设备所执行的数据黑洞处理方法为单机版数据黑洞处理方法。

[0518] 如图29a所示,计算设备70包括:应用层(或者应用层对应的单元)71、操作系统内核层(或者操作系统内核层对应的单元)72、硬件映射层(或者硬件映射层对应的单元)73、安全层(或者安全层对应的单元)74,这些层次或单元与之前的实施例的计算设备200所包括的用户界面层201、应用层202、操作系统内核层203、硬件映射层204、安全层205以及硬件层206对应,不再赘述。

[0519] 移动计算设备70还包括:硬件层75。

[0520] 硬件层75包括设备或单元如下:CPU、网卡和硬盘75a。

[0521] 硬盘75a包括:普通存储区域和安全存储区域75a1。

[0522] 该安全存储区域75a1也可以为加密存储区域,在数据存取之前或之后需要对数据进行加解密处理。

[0523] 另外,当上述数据安全读取方法(例如S5000)和存储方法(例如S4000)应用在独立的计算设备时,上述方法成为单机版的数据安全存储和读取方法;该独立计算设备(例如PC)包括相互独立的本地存储空间和安全存储空间。

[0524] 例如,单机版数据安全存储方法包括:

[0525] 接收硬件指令;

[0526] 如果所述硬件指令是存储指令,将所述存储指令中的目标地址修改为对应的在所

述计算设备上的安全存储空间的存储地址;和

[0527] 将修改后的存储指令发送到硬件层执行。

[0528] 例如,单机版数据安全读取方法包括:

[0529] 接收硬件指令;

[0530] 如果所述硬件指令是读取指令,获取读取指令中的源地址,查找第一映射位图,并根据映射位图的数据修改读取指令中的读取地址;和

[0531] 将修改后的硬件指令发送到硬件层执行。

[0532] 结合前述实施例中提供的安全存储装置和安全读取装置(例如装置7100、装置8100、装置9100等),按需要删减其中不需要的单元,可以成为单机版数据安全存储和读取装置。

[0533] 根据本发明一个实施例,如图29b所示,计算设备包括:相互独立的本地存储空间87和安全存储空间88,以及单机版数据安全存储和读取装置80;其中安全存储空间对于操作系统是不可用的(例如不可见或者不可访问),只能由单机版数据安全存储和读取装置80访问;

[0534] 其中,所述单机版数据安全存储和读取装置80包括:

[0535] 接收单元81,适于接收硬件指令;

[0536] 指令分析单元82,适于判断所述硬件指令是否为存储或读取指令,产生判断信号;

[0537] 指令修改单元83,适于当所述硬件指令为存储指令时,将所述存储指令中的目标地址修改为对应的在安全存储空间内的存储地址;还适于当所述硬件指令为读取指令时,查找映射位图,并根据映射位图的数据修改所述读取指令中的读取地址;所述映射位图用于表示本地存储空间的地址的数据是否转储到所述安全存储空间,映射位图已经在前述实施例中详细描述,这里不再赘述;

[0538] 发送单元84,适于将修改后的读取或存储指令发送到硬件层执行。

[0539] 上述计算设备还可以包括:更新单元85,适于在指令修改单元83修改所述存储指令之后,更新映射位图中所述目标地址对应的位。

[0540] 上述计算设备还可以包括:加解密单元86,适于对进出安全存储空间88的数据进行加密和解密。

[0541] 结合图29a,根据本发明一个实施例,提供一种单机版数据黑洞处理方法,如图30所示,包括:

[0542] Sa1,在计算设备(例如计算机、手持通信设备、智能终端等)部署数据黑洞系统,成为数据黑洞终端;

[0543] Sa2,建立数据黑洞空间,包括:在计算设备本地开辟一个数据存储区(称为黑洞存储区)以及本地内存,其中,数据存储区只能由数据黑洞系统访问,不能被终端计算设备的操作系统或应用层访问;

[0544] Sa3,为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系,例如,当用户登录数据黑洞终端,使终端用户与数据黑洞空间形成一一对应关系;

[0545] Sa4,数据黑洞终端将用户操作所产生的“数据写”重定向到与该用户对应的数据黑洞空间并加密,例如,重定向到与该用户对应的黑洞存储区;

[0546] Sa5,阻止对于本地存储设备(除黑洞存储区外)的数据持久化操作,并且阻止通过

本地端口对非数据黑洞终端的数据输出，从而保证进入数据黑洞终端或者数据黑洞空间的数据只在数据黑洞空间中存在。

[0547] 其中，Sa1表示步骤1。

[0548] 基于移动存储器的数据黑洞处理方法

[0549] 在涉密人员操作移动计算设备(例如笔记本电脑或平板电脑)时，如果不方便与远程安全存储设备(用作黑洞存储区)连接，可以使用移动存储设备作为安全存储设备。将计算设备(包括移动计算设备)的安全性转化为移动存储设备的安全性。

[0550] 根据本发明一个实施例，如图31所示，其中涉密人员通过移动计算设备20操作涉密数据，由于涉密数据不能存放在本地，而且位于网络的安全存储设备不方便连接，此时可以利用指定的移动存储设备作为涉密数据的载体，即利用移动存储设备作为临时的安全存储设备。

[0551] 图中移动计算设备20包括：用户界面层21、应用层22、操作系统内核层23、硬件映射层24、安全层25以及硬件层26与之前的实施例的计算设备200所包括的用户界面层201、应用层202、操作系统内核层203、硬件映射层204、安全层205以及硬件层206对应，不再赘述。

[0552] 为了方便涉密人员的工作，本发明上述实施例中提供的数据安全读取和存储方法可以与安全存储设备整合在一个移动存储设备中，作为便携式设备使用。

[0553] 如图32所示，根据本发明一个实施例，提供一种移动存储设备(即移动存储设备)50，其中包括：应用层(或者应用层对应的单元)52、操作系统内核层(或者操作系统内核层对应的单元)53、硬件映射层(或者硬件映射层对应的单元)54、安全层(或者安全层对应的单元)55。这些层次或单元与之前的实施例的计算设备200所包括的用户界面层201、应用层202、操作系统内核层203、硬件映射层204、安全层205以及硬件层206对应，不再赘述。

[0554] 移动存储设备50还包括：硬件层(或者硬件层对应的单元)56，其中包括数据接口56a以及安全存储区域56b。数据接口56a用于连接其他计算设备(通过相应的数据接口)，安全存储区域56b用于作为数据安全存储和读取方法中的安全存储设备(或者用作黑洞存储区)。

[0555] 计算终端40包括：应用层(或者应用层对应的单元)41、操作系统内核层(或者操作系统内核层对应的单元)42、硬件映射层(或者硬件映射层对应的单元)43以及硬件层(或者硬件层对应的单元)44。其中，硬件层44包括CPU 44a、硬盘44b、网卡44c、数据接口44d(例如USB接口)等硬件单元。

[0556] 其中，数据接口56a与数据接口44d耦接/连接。安全存储区域56b对移动存储设备50上的操作系统是不可用的。

[0557] 移动存储设备50通过数据接口与计算终端40连接，利用计算终端40的计算资源完成移动存储设备本身系统(包括层52～55)的工作，数据保存在安全存储区域56b中。

[0558] 其中，移动存储设备50进行的数据存储的过程包括：

[0559] 步骤A1、移动存储设备50通过数据接口56a、44d与计算终端40耦接；

[0560] 步骤A2、计算终端40重新启动，计算终端40的CPU 44a运行移动存储设备50携带的系统(包括层52～55对应的应用软件和系统软件)；

[0561] 步骤A3、用户通过计算终端40的I/O(输入输出设备，例如键盘44b)操作移动存储

设备50携带的系统；

[0562] 步骤A4、安全层55接收来自硬件映射层54的硬件指令；

[0563] 步骤A5、如果所述硬件指令是存储或者读取指令，安全层55修改所述存储指令中的目标地址或者读取指令中的源地址为对应的在所述移动存储设备上的安全存储区域56b中的存储地址；和

[0564] 步骤A6、将修改后的存储指令发送到计算终端40的CPU 44a。

[0565] 在步骤A4-A5中，安全层55所进行的数据转移存储的过程与之前的实施例中提供数据安全存储和读取方法相同，不再赘述。

[0566] 本实施例中，在安全存储区域56b与计算终端40的本地存储设备44b之间建立映射关系和映射表(即位图)的过程在之前描述的数据安全存储方法中也有详细的记载，不再赘述。

[0567] 另外，本发明上述实施例中提供的数据安全读取和存储方法可以与安全存储设备整合在一个移动计算设备(例如笔记本电脑或者智能手机)中，作为便携式设备使用。

[0568] 基于移动存储器的数据黑洞处理装置

[0569] 上述移动计算设备和移动存储设备可以结合前述实施例中提供的安全存储装置和安全读取装置(例如装置7100、装置8100、装置9100等)，删减其中不需要的单元，完成移动数据安全存储和读取方法。本领域技术人员可以理解，上述移动计算设备和移动存储设备与安全存储装置和安全读取装置的结合方式可以根据需要来设计。

[0570] 根据本发明一个实施例，提供一种移动计算设备。该移动计算设备(例如笔记本电脑或者智能手机)包括：相互独立的本地存储空间和安全存储空间；和数据安全存储和读取装置。其中安全存储空间对于操作系统是不可用的(例如不可见或者不可访问)。

[0571] 其中，所述数据安全存储和读取装置包括：

[0572] 接收单元，适于接收硬件指令；

[0573] 指令分析单元，适于判断所述硬件指令是否为存储或读取指令，产生判断信号；

[0574] 指令修改单元，适于当所述硬件指令为存储指令时，将所述存储指令中的目标地址修改为对应的在安全存储空间内的存储地址；还适于当所述硬件指令为读取指令时，查找映射位图，并根据映射位图的数据修改所述读取指令中的读取地址；所述映射位图用于表示本地存储空间的地址的数据是否转储到所述安全存储空间；

[0575] 发送单元，适于将修改后的读取或存储指令发送到硬件层执行。

[0576] 本实施例中，硬件指令来自硬件映射层。根据本发明另一个实施例，上述的移动计算设备还包括：更新单元，适于在指令修改单元修改所述存储指令之后，更新映射位图中所述目标地址对应的位。

[0577] 上述移动计算设备(例如笔记本)，用于保护个人或企业用户数据外部应用授权后的数据安全保护。系统假定个人或企业用户在PC、笔记本上存有涉密数据，但因为系统有后门、漏洞、木马或其它未知的恶意代码而无法保障PC/笔记本上数据不会被泄密，同时也无法保证设备丢失后的数据安全保护。企业可用在数据从内网导出数据时，实现对数据的使用过程的保护和监控。

[0578] 本领域技术人员可以理解，上述移动计算设备(例如笔记本)也可以是独立计算机(例如PC)。

[0579] 根据本发明一个实施例，提供一种移动存储设备。该移动存储设备(例如U盘)包括：数据接口，安全存储空间，以及数据安全存储和读取装置；所述数据接口适于与计算设备耦接；所述计算设备包括本地存储空间，用于运行移动存储设备上的操作系统，并用于为所述数据安全存储和读取装置提供计算资源。

[0580] 数据安全存储和读取装置包括：

[0581] 接收单元，适于接收硬件指令；

[0582] 指令分析单元，适于判断所述硬件指令是否为存储或读取指令，产生判断信号；

[0583] 指令修改单元，适于当所述硬件指令为存储指令时，将所述存储指令中的目标地址修改为对应的在安全存储空间内的存储地址；还适于当所述硬件指令为读取指令时，查找映射位图，并根据映射位图的数据修改所述读取指令中的读取地址；所述映射位图用于表示本地存储空间的地址的数据是否转储到所述安全存储空间；和

[0584] 发送单元，适于将修改后的读取或存储指令发送到计算设备的硬件层执行。

[0585] 根据本发明另一个实施例，上述移动存储设备还包括：更新单元，适于在指令修改单元修改所述存储指令之后，更新映射位图中所述目标地址对应的位。

[0586] 根据本发明另一个实施例，硬件指令可以来自硬件映射层。

[0587] 上述移动存储设备(例如U盘)，以部署了数据安全存储和读取装置(或数据安全存储和读取方法)的U盘/移动硬件盘作为导出数据载体，用于保护导出数据的安全。核心是确保导出到外部的数据在非可控环境中使用时不会在数据使用过程中留下数据痕迹，同时确保在有系统后门、漏洞、木马或其它未知的恶意代码的环境中，数据不被复制或截留。

[0588] 上述实施例中，映射位图用于表示本地存储空间的地址的数据是否转储到所述安全存储空间。在本发明其他实施例中，也可以使用文件对应表的形式，即本地数据以文件的形式被转移存储到所述安全存储空间。

[0589] 本发明提供的上述方法和装置，相对于现有技术，具有如下优点：

[0590] A.可实现数据操作的过程追踪，具有对恶意代码、后门和木马数据操作的追踪能力；

[0591] B.具有在安全域内部实现文件操作授权，并确保文件授权后仍具有完全的监控能力；

[0592] C.能实现安全域间的文件授权，在授权后仍具有完全监控能力，并可对授权文件的实现定期、定次使用、定期销毁的能力；

[0593] D.可实现终端使用与服务器数据的全加密。

[0594] 本领域的技术人员(本领域的普通技术人员)可以理解，上述的数据安全存储方法、读取方法及传输方法可使用软件或硬件的形式来实现：

[0595] (1)如果以软件实现，则上述方法对应的步骤以软件代码的形式存储在计算机可读介质上，成为软件产品；

[0596] (2)如果以硬件实现，则上述方法对应的步骤以硬件代码(例如Verilog)的形式描述，并固化(经过物理设计/布局布线/晶圆厂流片等过程)成为芯片产品(例如处理器产品)。

[0597] 具体的，如本领域的普通技术人员将意识到的那样，本发明可以具体实现成一种系统、方法或计算机程序产品。因此，本发明可以采用完全硬件实施例、完全软件实施例(包

括固件、驻留软件、微码等)的形式、或者组合了软件和硬件方面的实施例的形式，它们在此可以总称为“电路”、“模块”或“系统”。

[0598] 此外，本发明可以采用在表达有计算机可用的程序代码的任何有形的介质中具体实现的计算机程序产品的形式。

[0599] 一个或多个计算机可用或计算机可读介质的任何组合都可以被使用。计算机可用或计算机可读介质可以是(但不限于)例如电子的、磁的、光的、电磁的、红外的或半导体的系统、装置、设备或传播介质。计算机可读介质的更为具体的例子(非穷举列表)将包括以下：具有一个或多个导线的电气连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦写可编程只读存储器(EPROM或闪存)、光纤、便携式致密盘只读存储器(CD-ROM)、光存储设备、诸如支持因特网或内部网的那些传输介质，或者磁存储设备。

[0600] 注意，计算机可用或计算机可读介质甚至可以是纸或可以打印程序的另外的合适的介质，因为程序可以经由例如对纸或其他介质的光学扫描而被电气捕获、接着被编辑、被翻译或者以合适的方式来进行其他处理，如果必要，并且接着被存储在计算机存储器中。在本文档的上下文中，计算机可用的或计算机可读的介质可以是可以包含、存储、通信、传播或传送程序以供由指令执行系统、装置或设备或结合其来使用的任意介质。计算机可用介质可以包括其中包含计算机可用程序代码的传播的数据信号，其可以是在基带中或者可以作为载波的一部分。计算机可用程序代码可以通过使用任何合适的介质来传输，这些介质包括但不限于无线、有线、光缆、RF等等。

[0601] 用于执行本发明的操作的计算机程序代码可以用一种或多种编程语言的任何组合来编写，这些语言包括诸如Java、Smalltalk、C++等等之类的面向对象的编程语言和诸如“C”编程语言或类似的编程语言之类的传统过程语言。程序代码可全部在用户的计算机上、部分地在用户的计算机上作为单机软件包执行、部分地在用户计算机上且部分地在远程计算机上执行、或者全部在远程计算机或服务器上执行。在后面这种情况下，远程计算机可以经由任何类型的网络连接到用户计算机，这些网络包括局域网(LAN)或广域网(WAN)或者可以连接到外面计算机的连接(例如，通过使用因特网服务提供商的因特网)。

[0602] 应该注意到并理解，在不脱离后附的权利要求所要求的本发明的精神和范围的情况下，能够对上述详细描述的本发明做出各种修改和改进。因此，要求保护的技术方案的范围不受所给出的任何特定示范教导的限制。

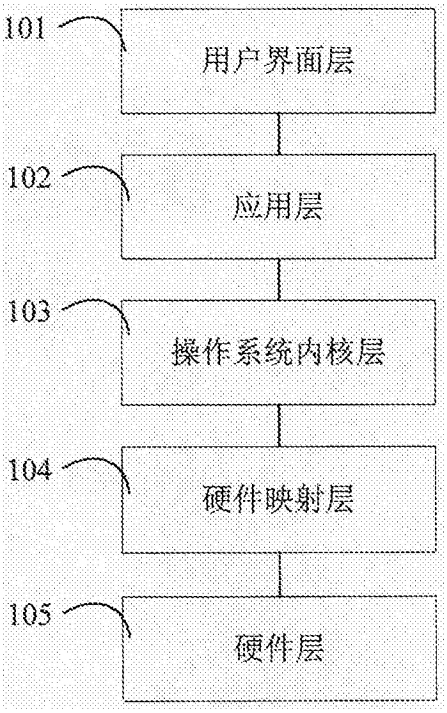

101
用户界面层
102
应用层
103
操作系统内核层
104
硬件映射层
105
硬件层

图1

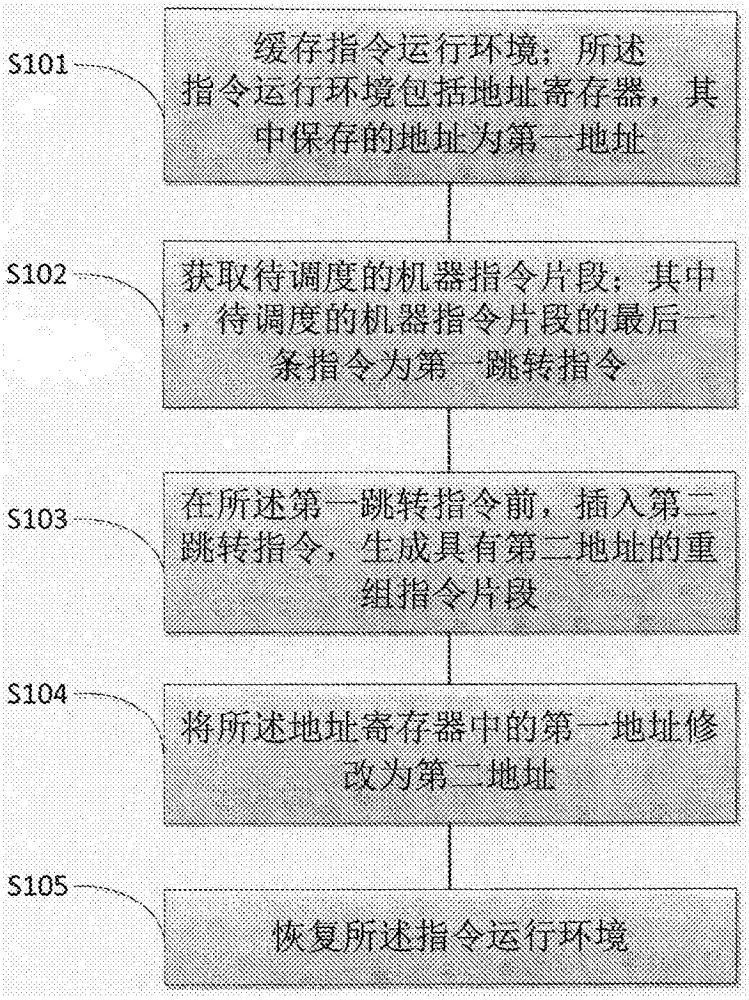

S101
缓存指令运行环境；所述指令运行环境包括地址寄存器，其中保存的地址为第一地址
S102
获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一跳转指令
S103
在所述第一跳转指令前，插入第二跳转指令，生成具有第二地址的重组指令片段
S104
将所述地址寄存器中的第一地址修改为第二地址
S105
恢复所述指令运行环境

图2

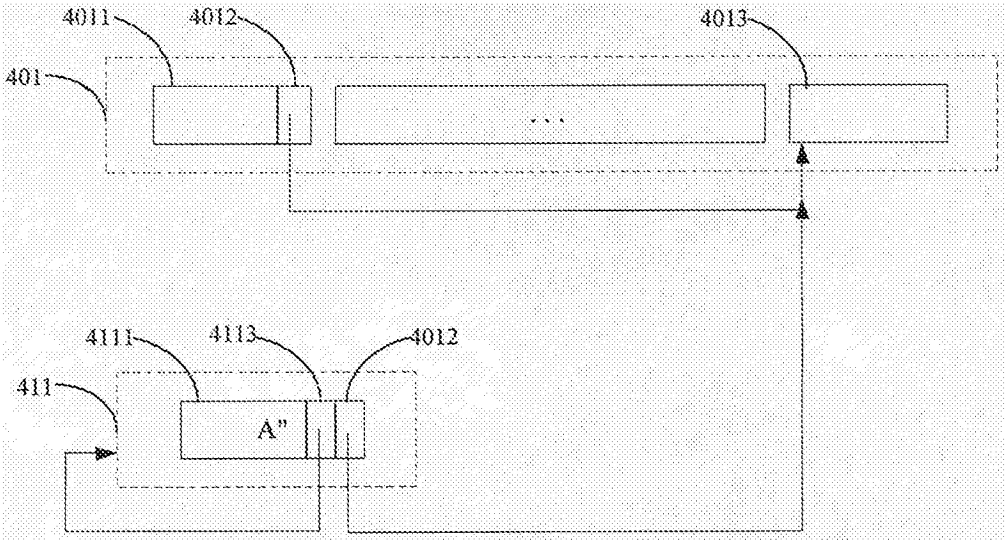

4011
4012
4013
401
...
4111
4113
4012
411
A"

图3

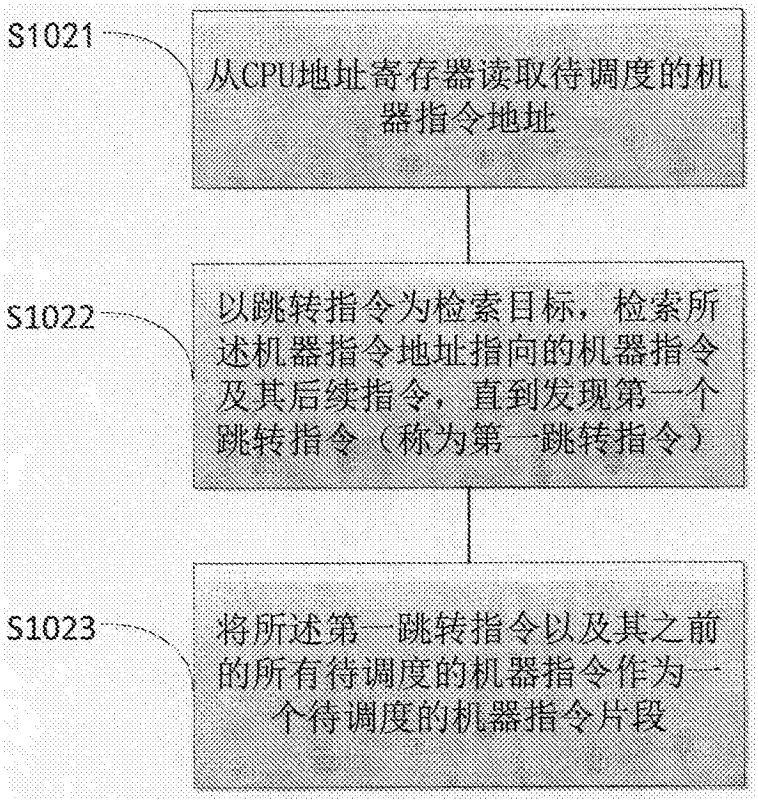

S1021
从CPU地址寄存器读取待调度的机器指令地址
S1022
以跳转指令为检索目标，检索所述机器指令地址指向的机器指令及其后续指令，直到发现第一个跳转指令（称为第一跳转指令）
S1023
将所述第一跳转指令以及其之前的所有待调度的机器指令作为一个待调度的机器指令片段

图4

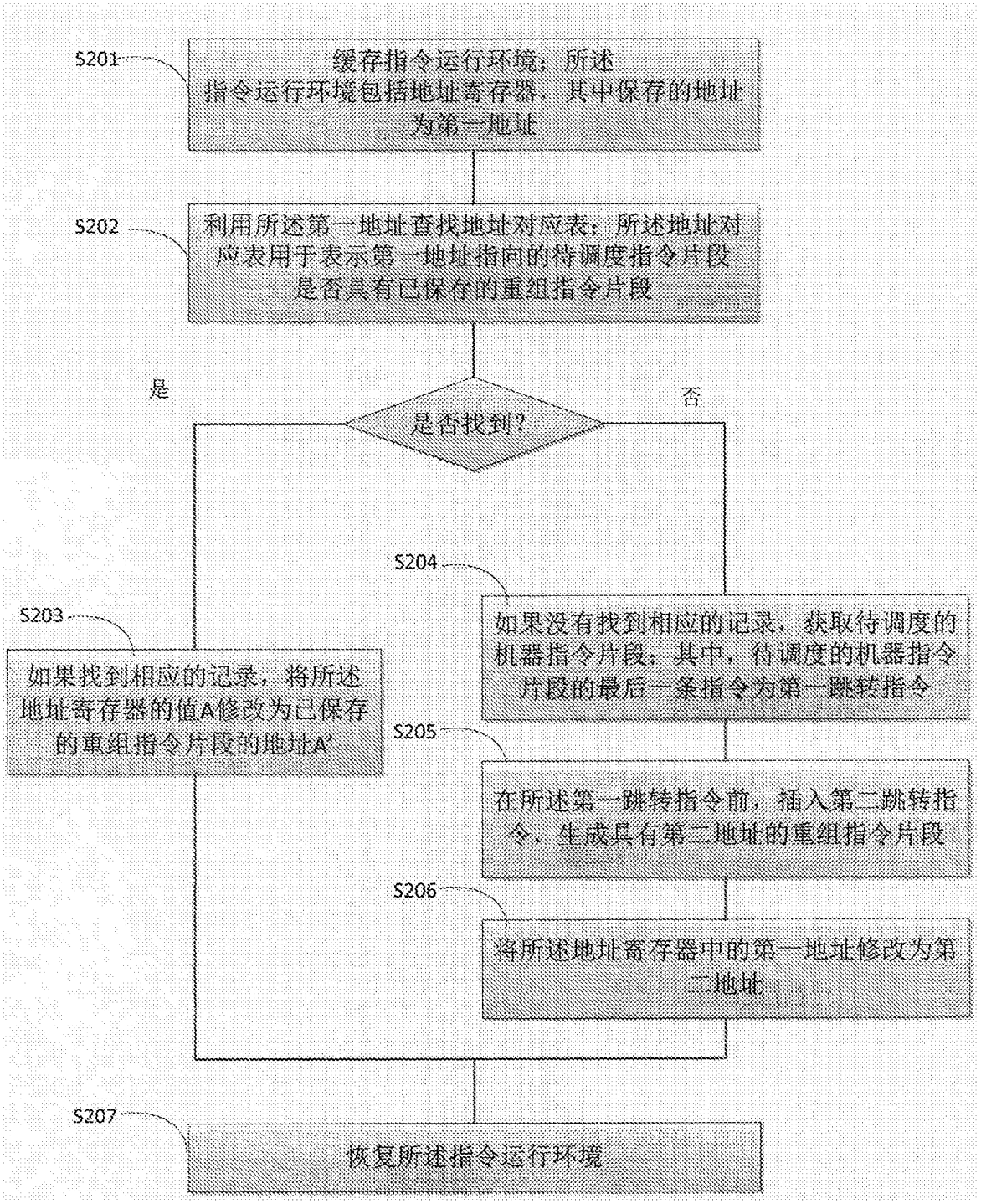

S201
缓存指令运行环境；所述指令运行环境包括地址寄存器，其中保存的地址为第一地址
S202
利用所述第一地址查找地址对应表；所述地址对应表用于表示第一地址指向的待调度指令片段是否具有已保存的重组指令片段
是否找到？
是
否
S203
如果找到相应的记录，将所述地址寄存器的值A修改为已保存的重组指令片段的地址A'
S204
如果没有找到相应的记录，获取待调度的机器指令片段；其中，待调度的机器指令片段的最后一条指令为第一跳转指令
S205
在所述第一跳转指令前，插入第二跳转指令，生成具有第二地址的重组指令片段
S206
将所述地址寄存器中的第一地址修改为第二地址
S207
恢复所述指令运行环境

图5

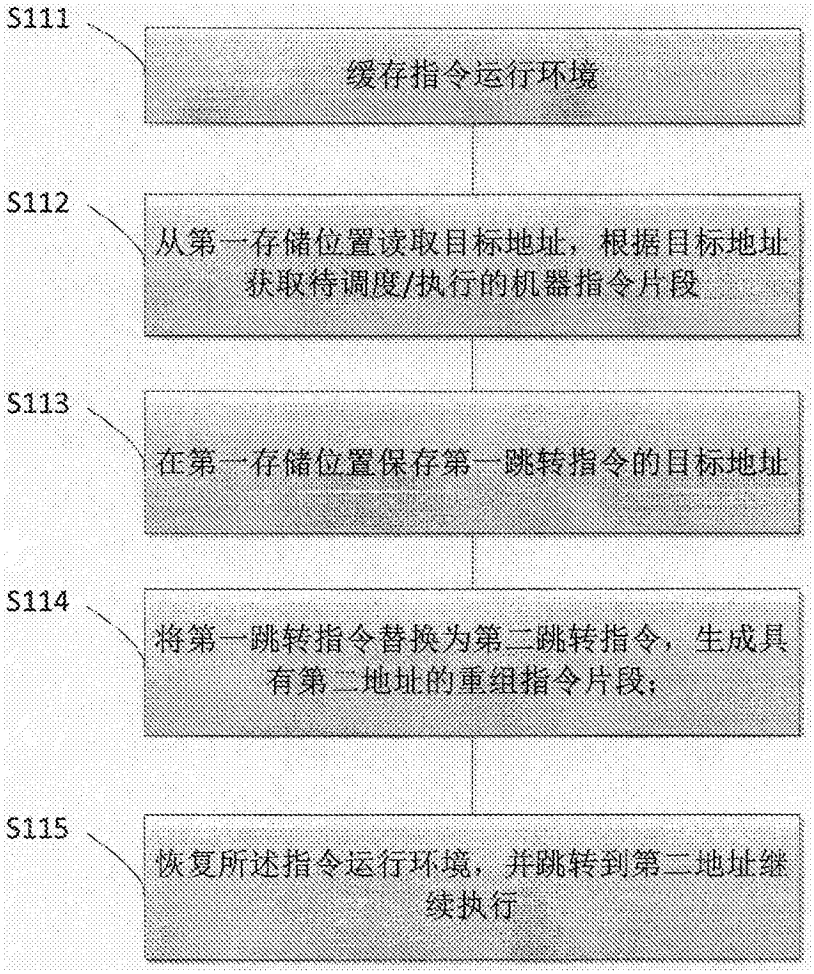

S111
缓存指令运行环境
S112
从第一存储位置读取目标地址，根据目标地址获取待调度/执行的机器指令片段
S113
在第一存储位置保存第一跳转指令的目标地址
S114
将第一跳转指令替换为第二跳转指令，生成具有第二地址的重组指令片段；
S115
恢复所述指令运行环境，并跳转到第二地址继续执行

图6

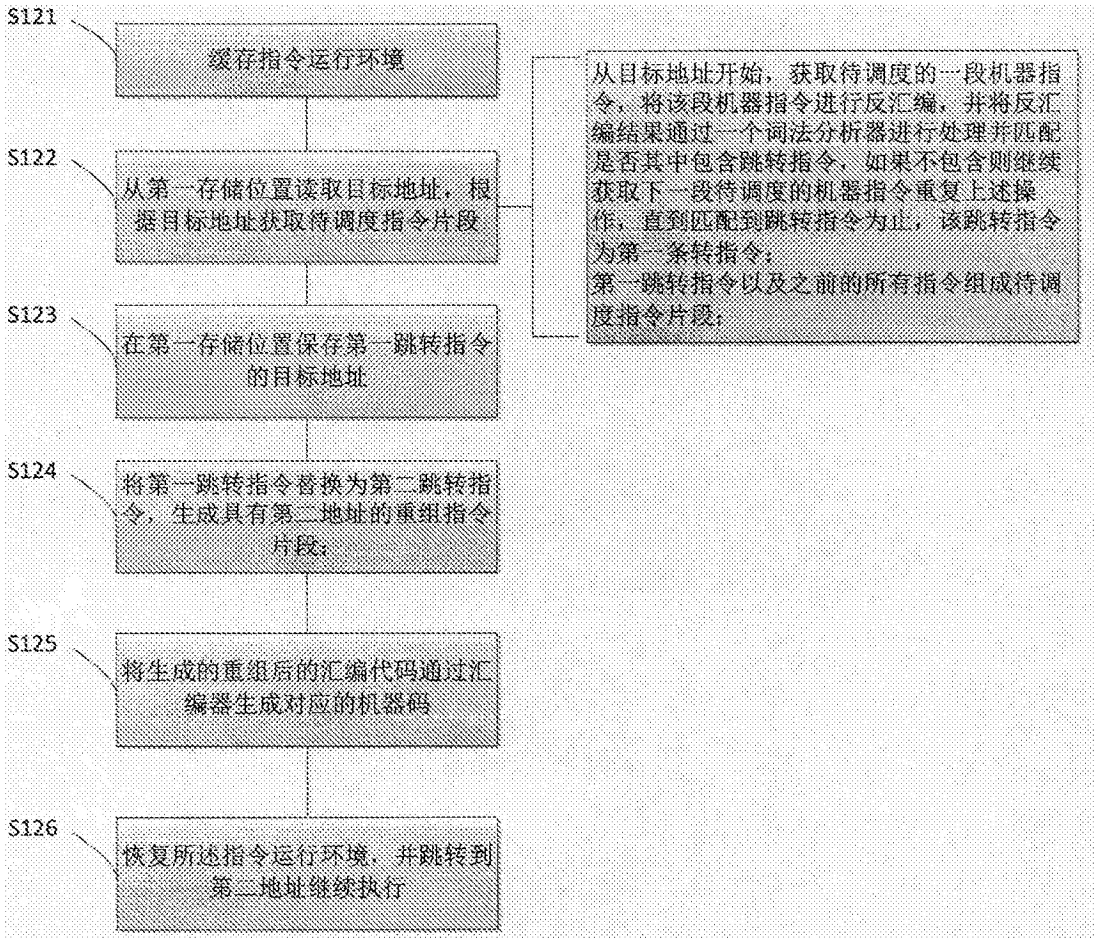

S121
缓存指令运行环境
S122
从第一存储位置读取目标地址，根据目标地址获取待调度指令片段
从目标地址开始，获取待调度的一段机器指令，将该段机器指令进行反汇编，并将反汇编结果通过一个词法分析器进行处理并匹配是否其中包含跳转指令，如果不包含则继续获取下一段待调度的机器指令重复上述操作，直到匹配到跳转指令为止，该跳转指令为第一条转指令；
第一跳转指令以及之前的所有指令组成待调度指令片段；
S123
在第一存储位置保存第一跳转指令的目标地址
S124
将第一跳转指令替换为第二跳转指令，生成具有第二地址的重组指令片段；
S125
将生成的重组后的汇编代码通过汇编器生成对应的机器码
S126
恢复所述指令运行环境，并跳转到第二地址继续执行

图7

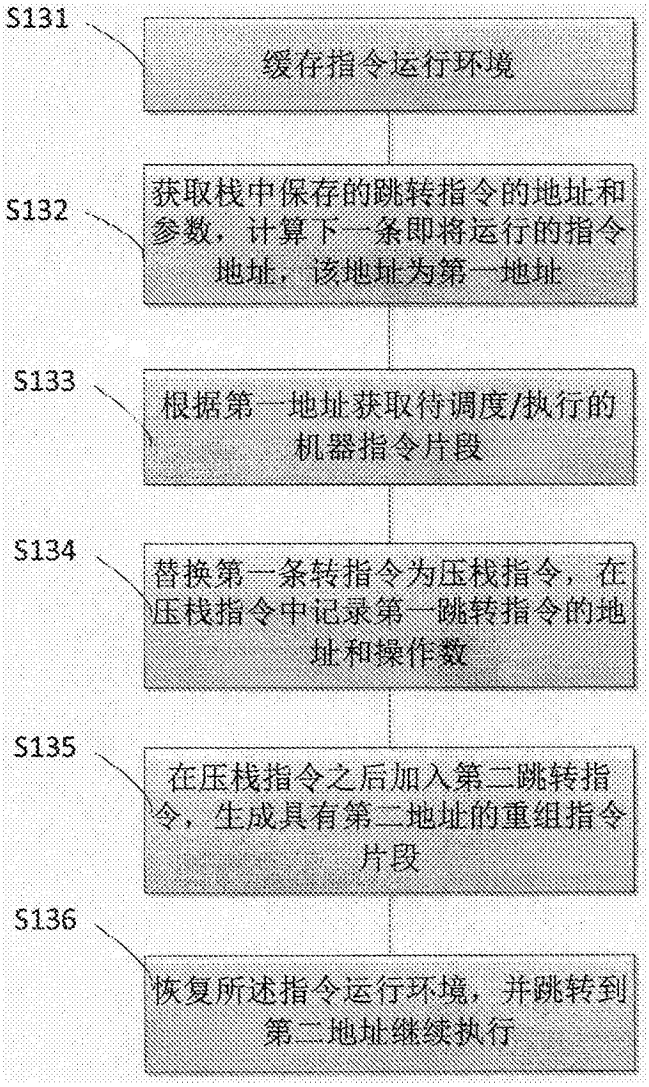

S131
缓存指令运行环境
S132
获取栈中保存的跳转指令的地址和参数，计算下一条即将运行的指令地址，该地址为第一地址
S133
根据第一地址获取待调度/执行的机器指令片段
S134
替换第一条转指令为压栈指令，在压栈指令中记录第一跳转指令的地址和操作数
S135
在压栈指令之后加入第二跳转指令，生成具有第二地址的重组指令片段
S136
恢复所述指令运行环境，并跳转到第二地址继续执行

图8

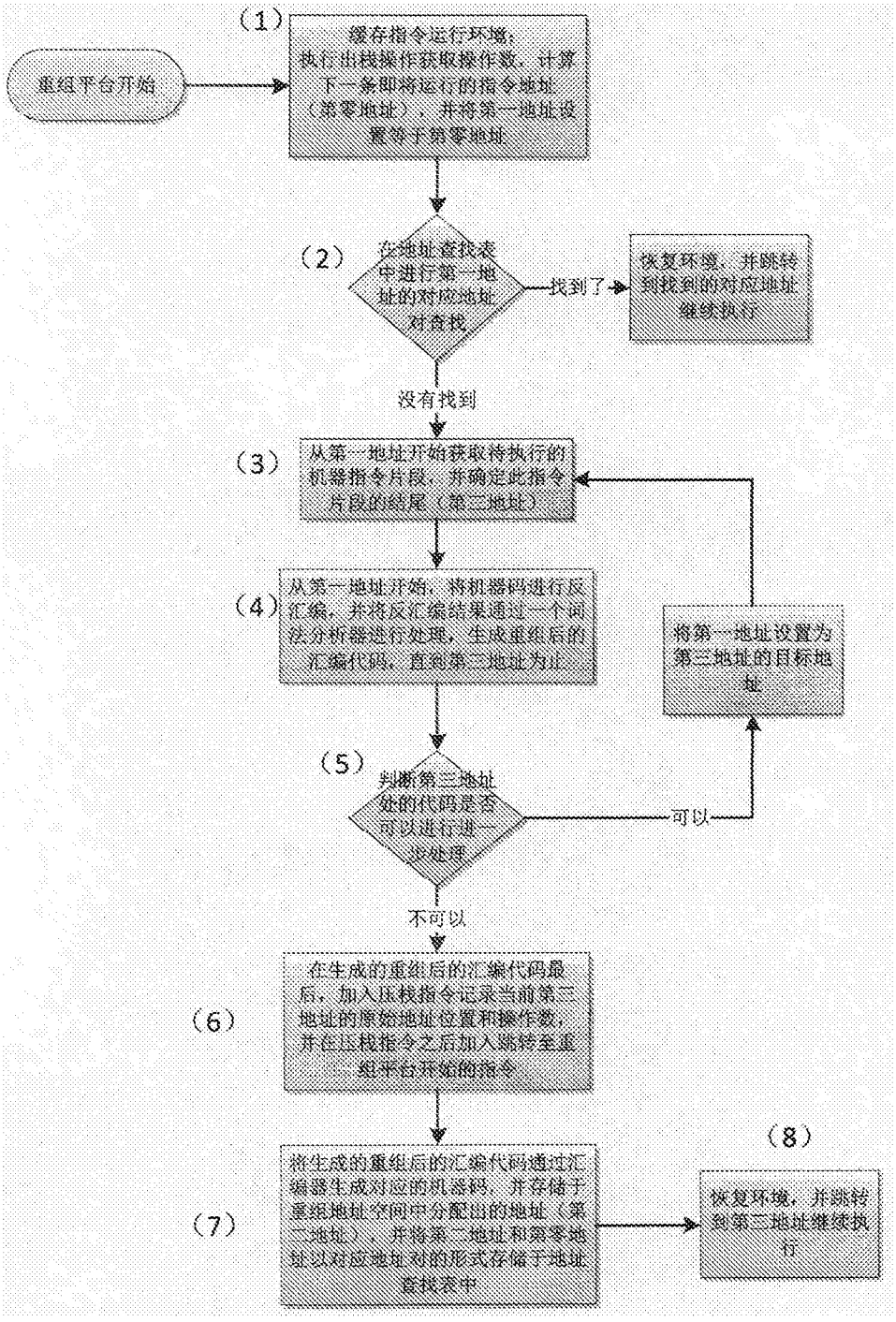

重组平台开始
(1)
缓存指令运行环境，执行出栈操作获取操作数，计算下一条即将运行的指令地址（第零地址），并将第一地址设置等于第零地址
(2)
在地址查找表中进行第一地址的对应地址对查找
找到了
恢复环境，并跳转到找到的对应地址继续执行
没有找到
(3)
从第一地址开始获取待执行的机器指令片段，并确定此指令片段的结尾（第二地址）
(4)
从第一地址开始，将机器码进行反汇编，并将反汇编结果通过一个词法分析器进行处理，生成重组后的汇编代码，直到第二地址为止
(5)
判断第三地址处的代码是否可以进行进一步处理
可以
将第一地址设置为第三地址的目标地址
不可以
(6)
在生成的重组后的汇编代码最后，加入压栈指令记录当前第三地址的原始地址位置和操作数，并在压栈指令之后加入跳转至重组平台开始的指令
(7)
将生成的重组后的汇编代码通过汇编器生成对应的机器码，并存储于重组地址空间中分配出的地址（第二地址），并将第二地址和第零地址以对应地址对的形式存储于地址查找表中
(8)
恢复环境，并跳转到第三地址继续执行

图9a

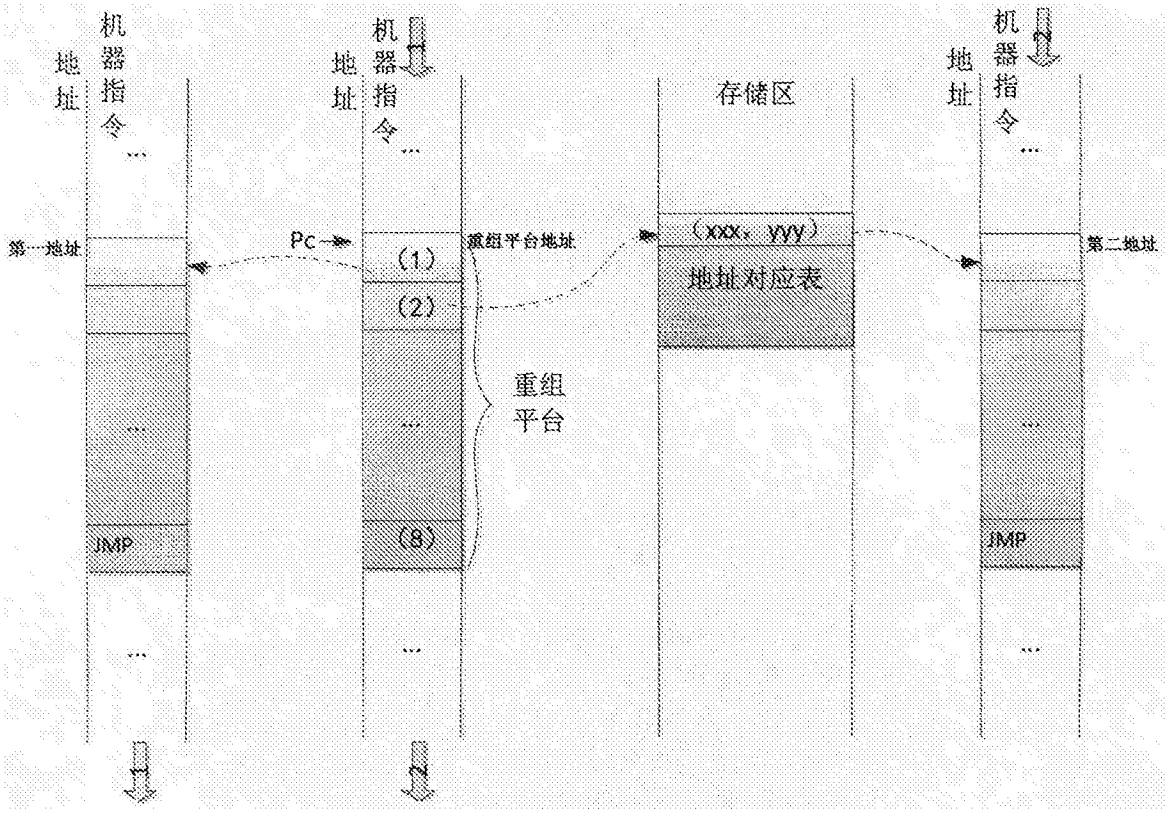


图9b

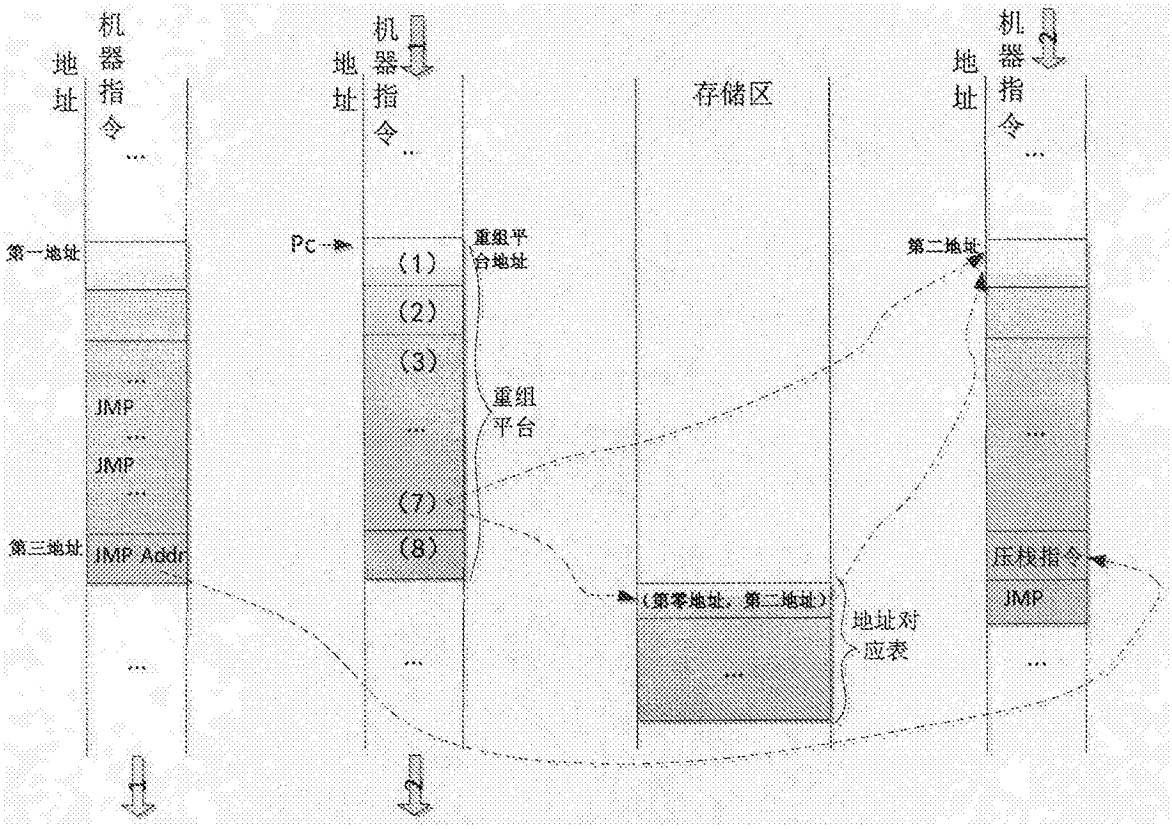

地址
机器指令
...
第一地址
JMP
JMP
第三地址
JMP Addr
Pc
(1)
(2)
(3)
(7)
(8)
重组平台地址
重组平台
存储区
(第零地址，第二地址)
地址对应表
第二地址
压栈指令
JMP

图9c

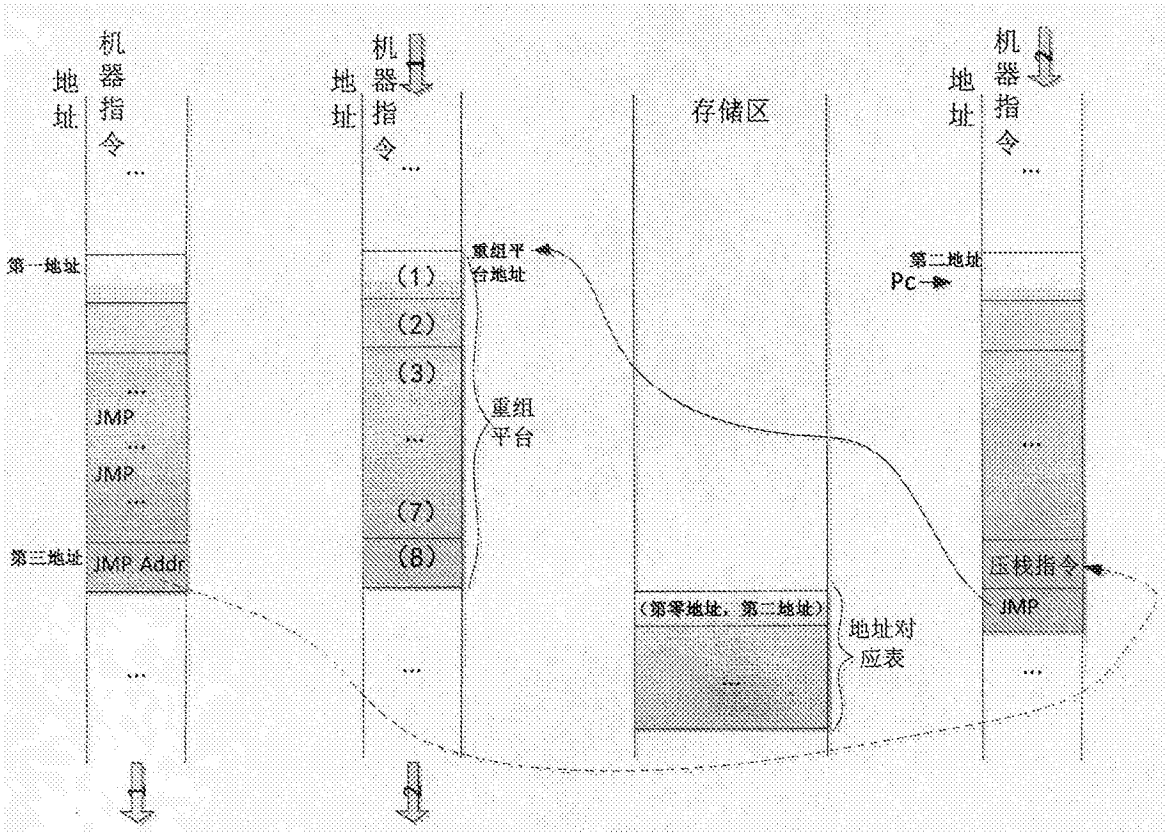

地址
机器指令
存储区
第一地址
JMP
JMP
第三地址
JMP Addr
(1)
(2)
(3)
(7)
(8)
重组平台地址
重组平台
第二地址
Pc
应栈指令
JMP
(第零地址，第二地址)
地址对应表

图9d

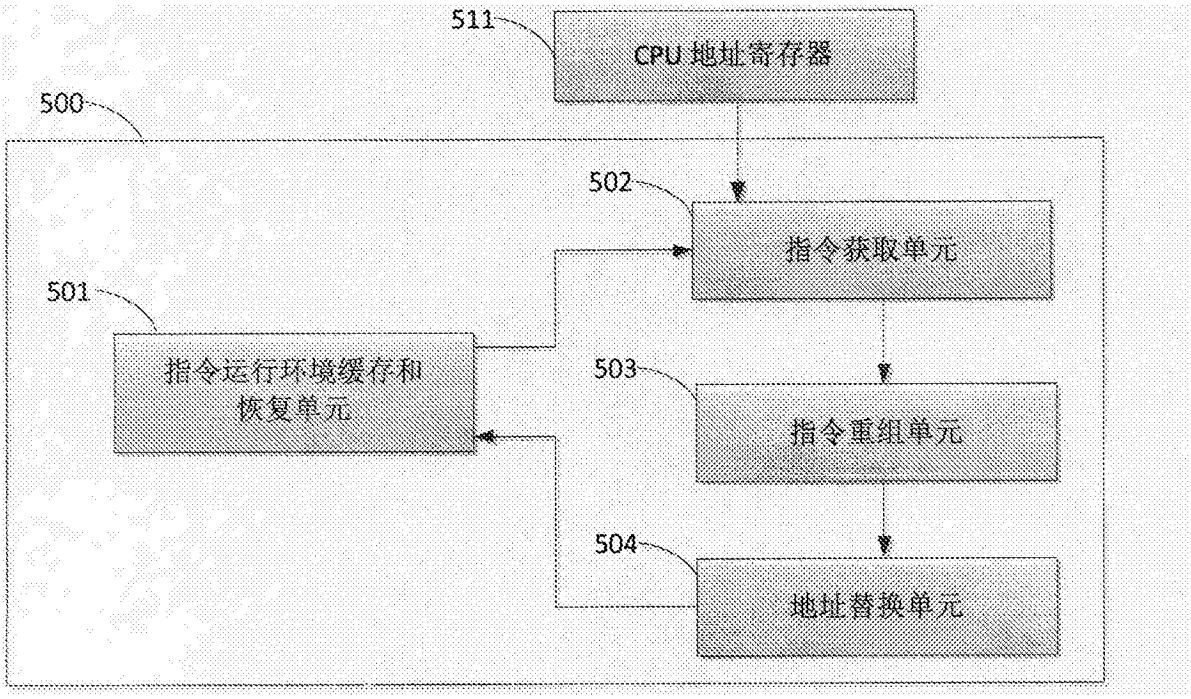

511
CPU 地址寄存器
500
502
指令获取单元
501
指令运行环境缓存和恢复单元
503
指令重组单元
504
地址替换单元

图10

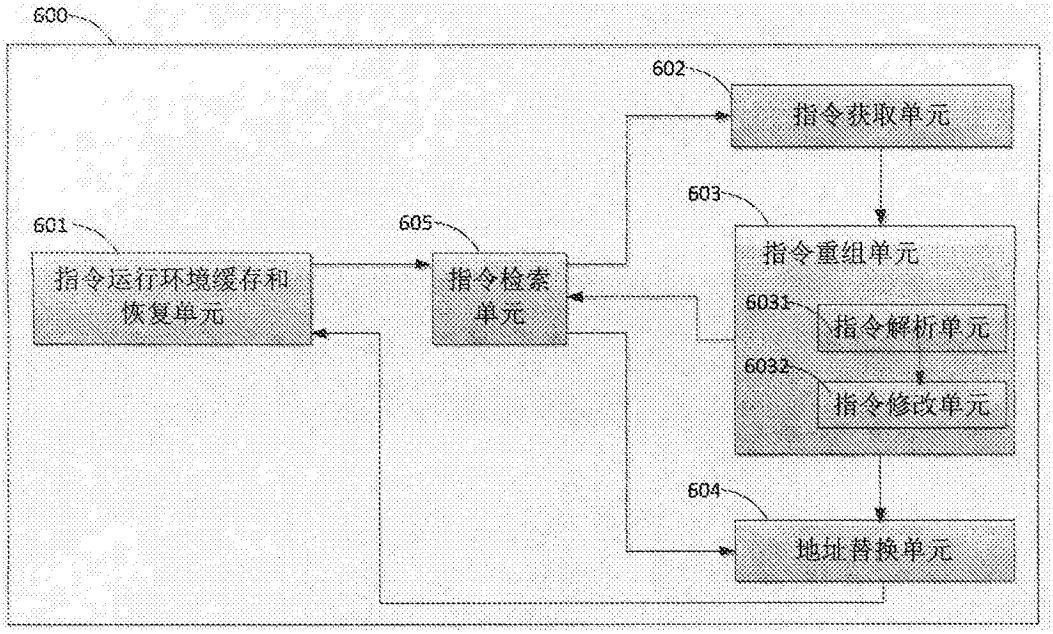

600
602
指令获取单元
603
601
605
指令运行环境缓存和恢复单元
指令检索单元
指令重组单元
6031
指令解析单元
6032
指令修改单元
604
地址替换单元

图11

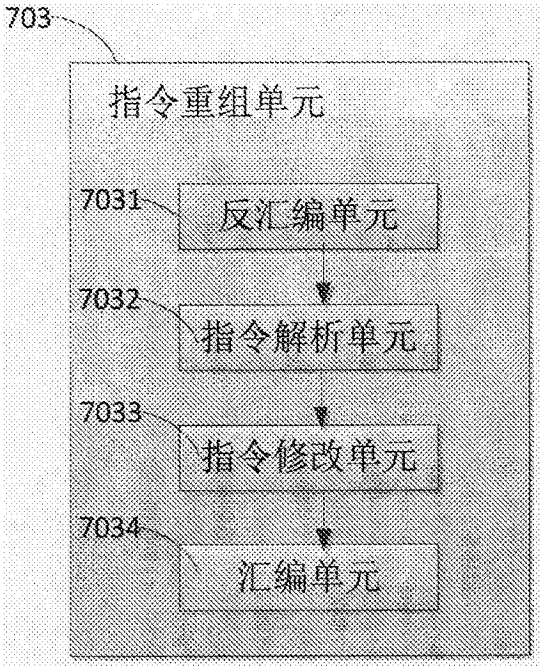

703
指令重组单元
7031
反汇编单元
7032
指令解析单元
7033
指令修改单元
7034
汇编单元

图12

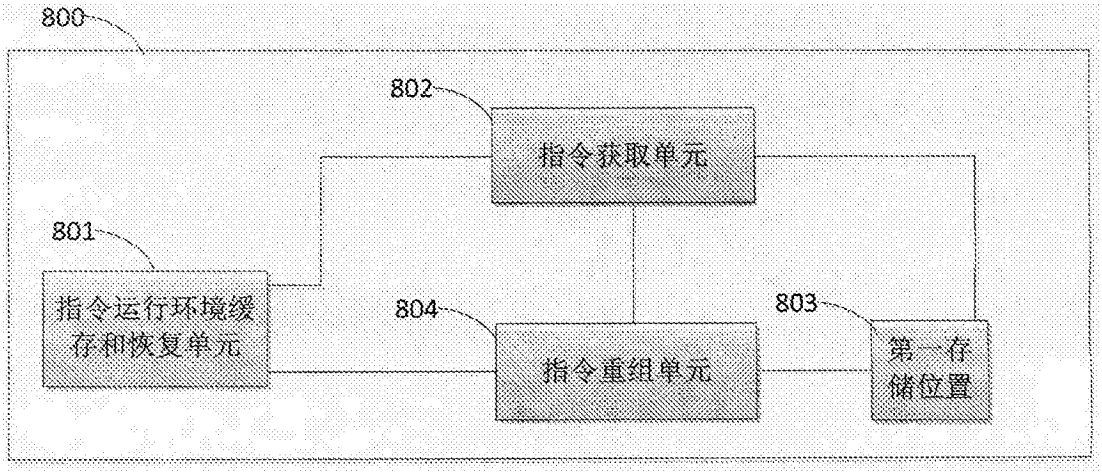

800
802
指令获取单元
801
指令运行环境缓存和恢复单元
804
指令重组单元
803
第一存储位置

图13

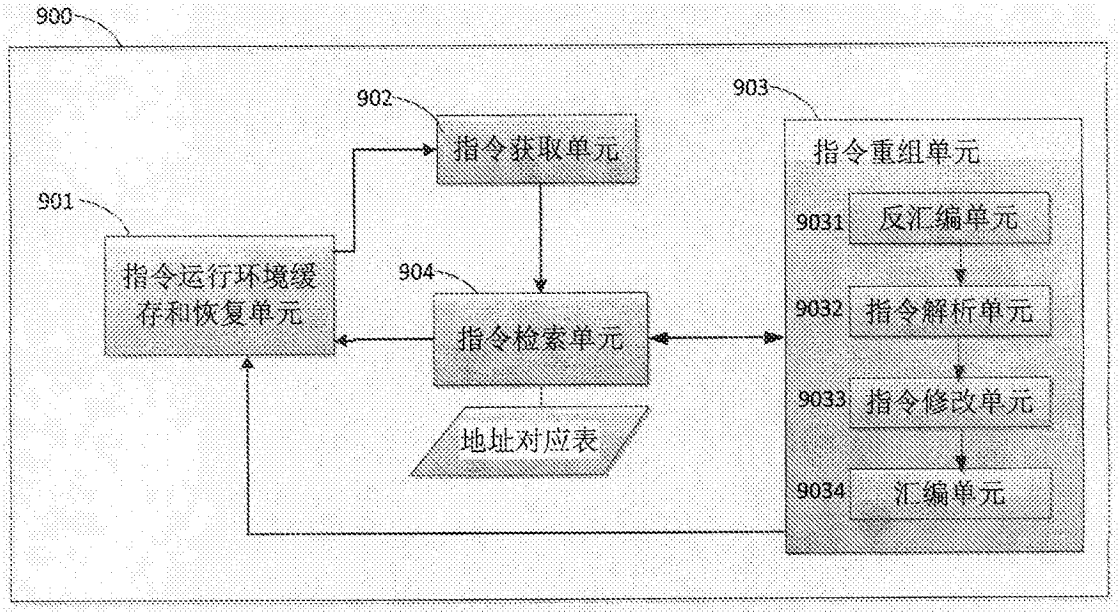

900
902
指令获取单元
903
指令重组单元
901
指令运行环境缓存和恢复单元
904
指令检索单元
地址对应表
9031
反汇编单元
9032
指令解析单元
9033
指令修改单元
9034
汇编单元

图14

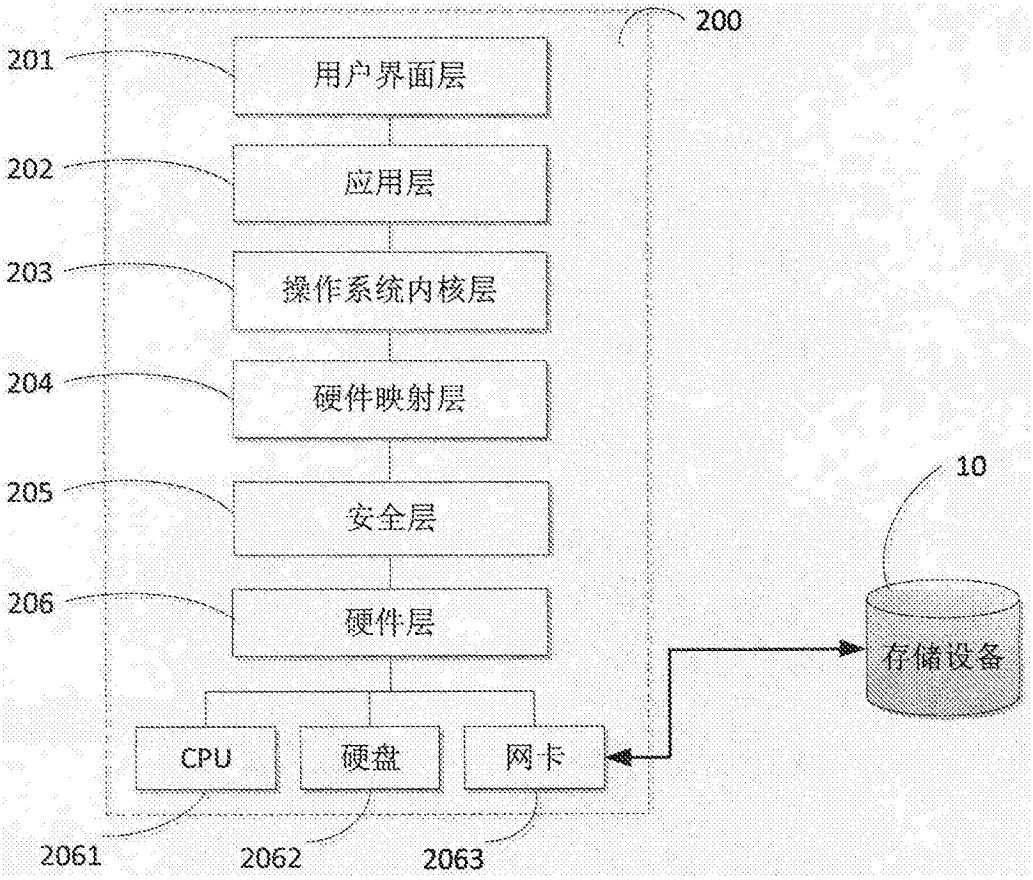

200
201
用户界面层
202
应用层
203
操作系统内核层
204
硬件映射层
205
安全层
206
硬件层
CPU
硬盘
网卡
2061
2062
2063
10
存储设备

图15

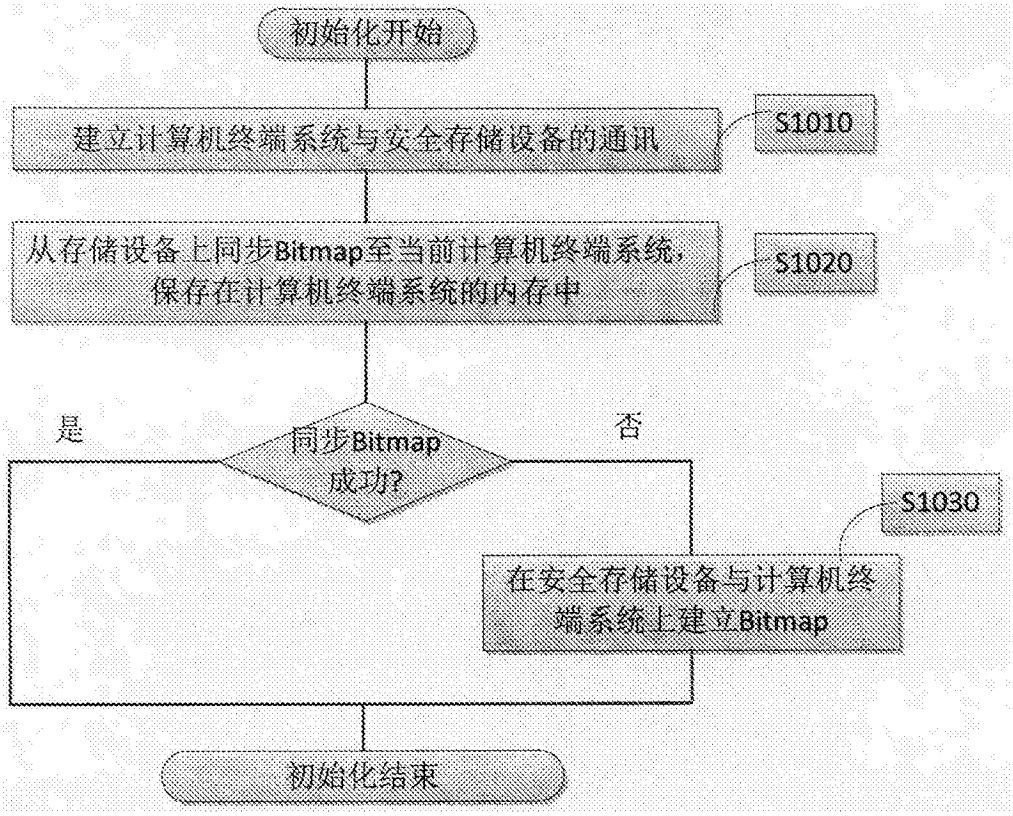

初始化开始
建立计算机终端系统与安全存储设备的通讯
S1010
从存储设备上同步Bitmap至当前计算机终端系统，保存在计算机终端系统的内存中
S1020
是
同步Bitmap成功?
否
S1030
在安全存储设备与计算机终端系统上建立Bitmap
初始化结束

图16

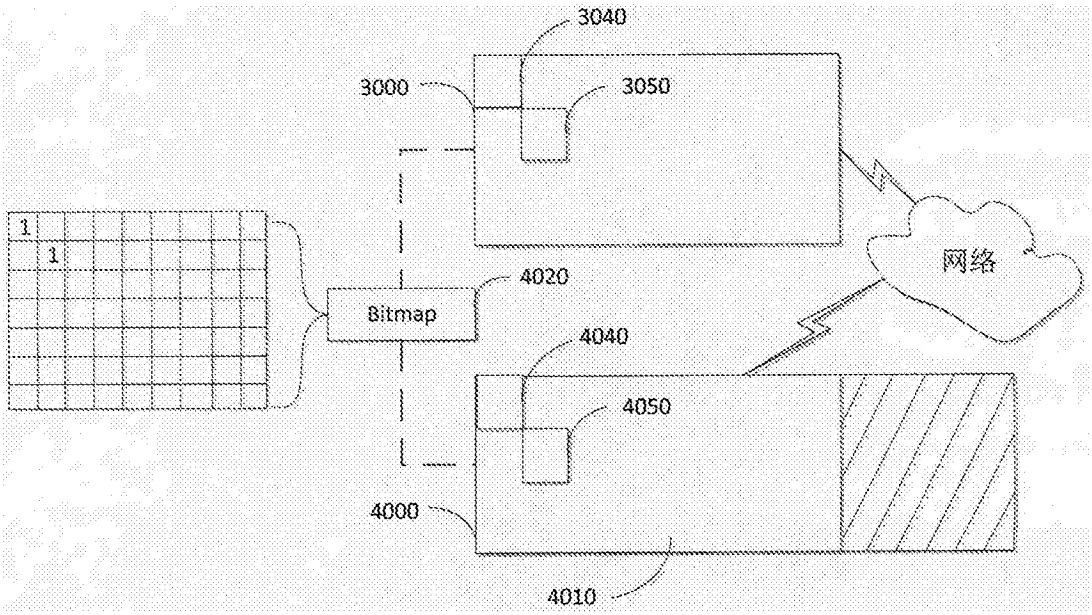

3040
3000
3050
1
1
网络
Bitmap
4020
4040
4050
4000
4010

图17

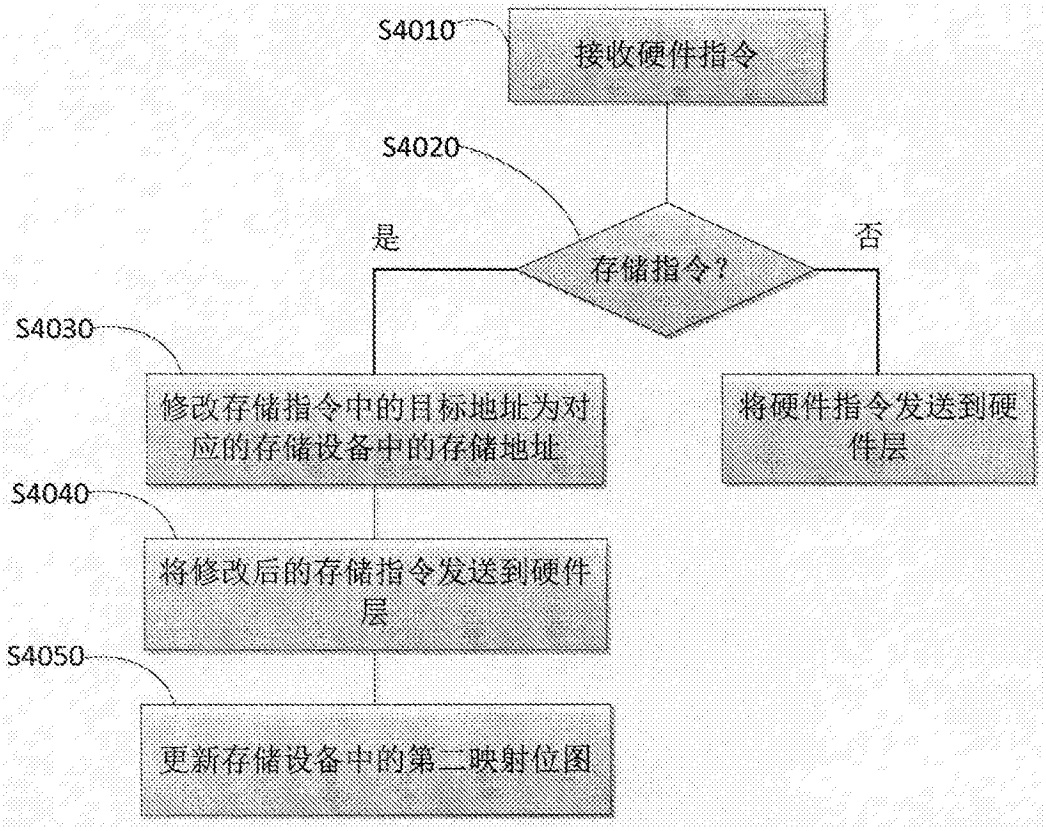

S4010
接收硬件指令
S4020
存储指令?
是
否
S4030
修改存储指令中的目标地址为对应的存储设备中的存储地址
将硬件指令发送到硬件层
S4040
将修改后的存储指令发送到硬件层
S4050
更新存储设备中的第二映射位图

图18

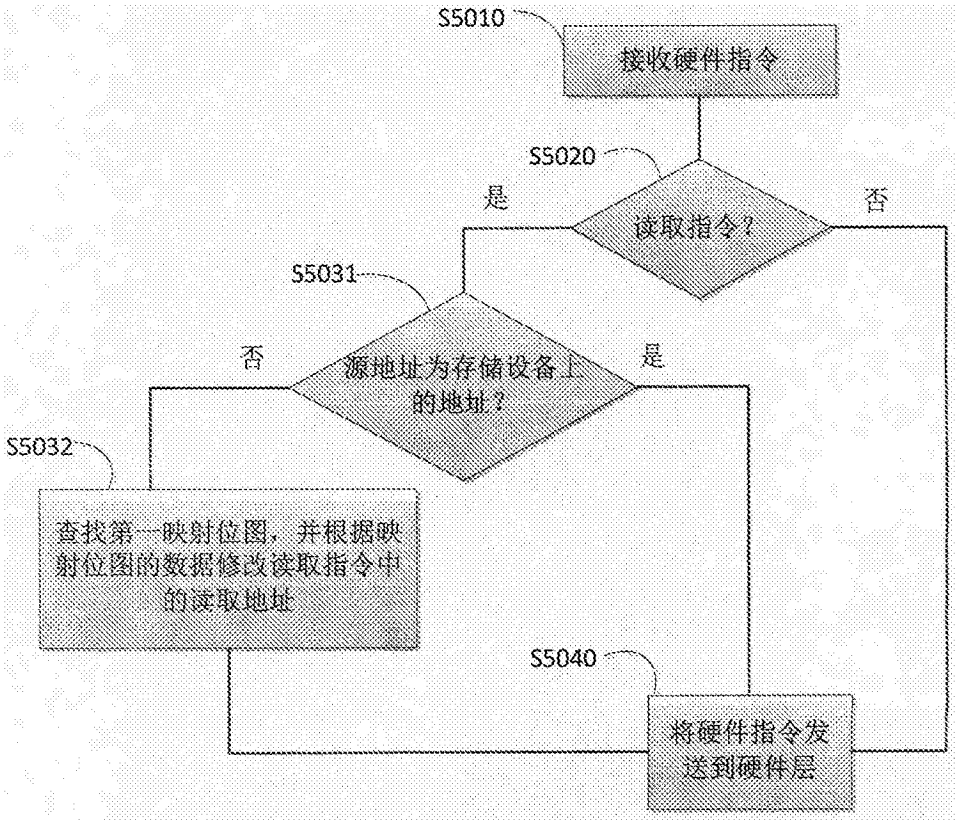

S5010
接收硬件指令
S5020
读取指令？
是
否
S5031
源地址为存储设备上的地址？
否
是
S5032
查找第一映射位图，并根据映射位图的数据修改读取指令中的读取地址
S5040
将硬件指令发送到硬件层

图19

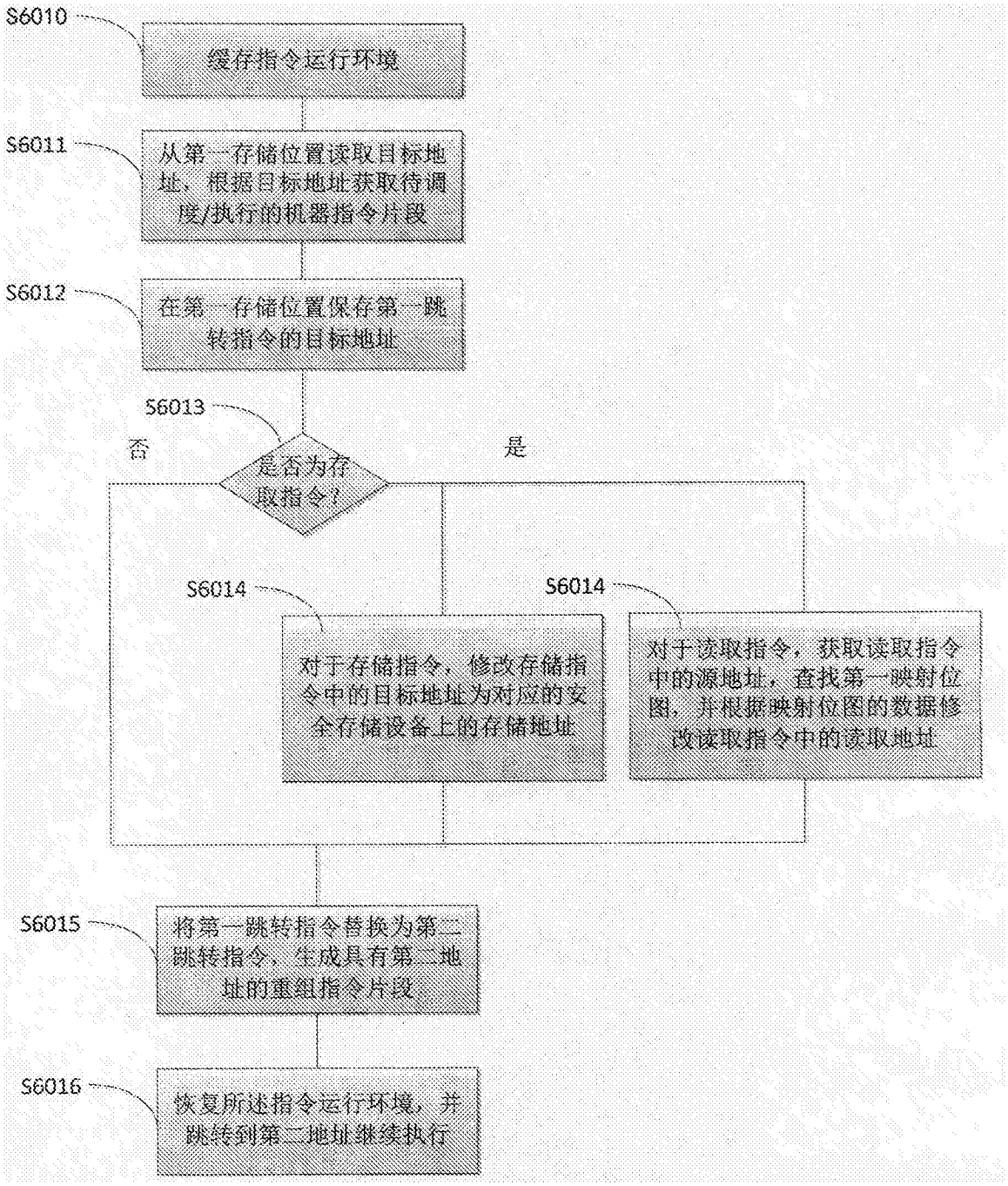

S6010
缓存指令运行环境
S6011
从第一存储位置读取目标地址，根据目标地址获取待调度/执行的机器指令片段
S6012
在第一存储位置保存第一跳转指令的目标地址
S6013
否
是否为存取指令？
是
S6014
对于存储指令，修改存储指令中的目标地址为对应的安全存储设备上的存储地址
S6014
对于读取指令，获取读取指令中的源地址，查找第一映射位图，并根据映射位图的数据修改读取指令中的读取地址
S6015
将第一跳转指令替换为第二跳转指令，生成具有第二地址的重组指令片段
S6016
恢复所述指令运行环境，并跳转到第二地址继续执行

图20

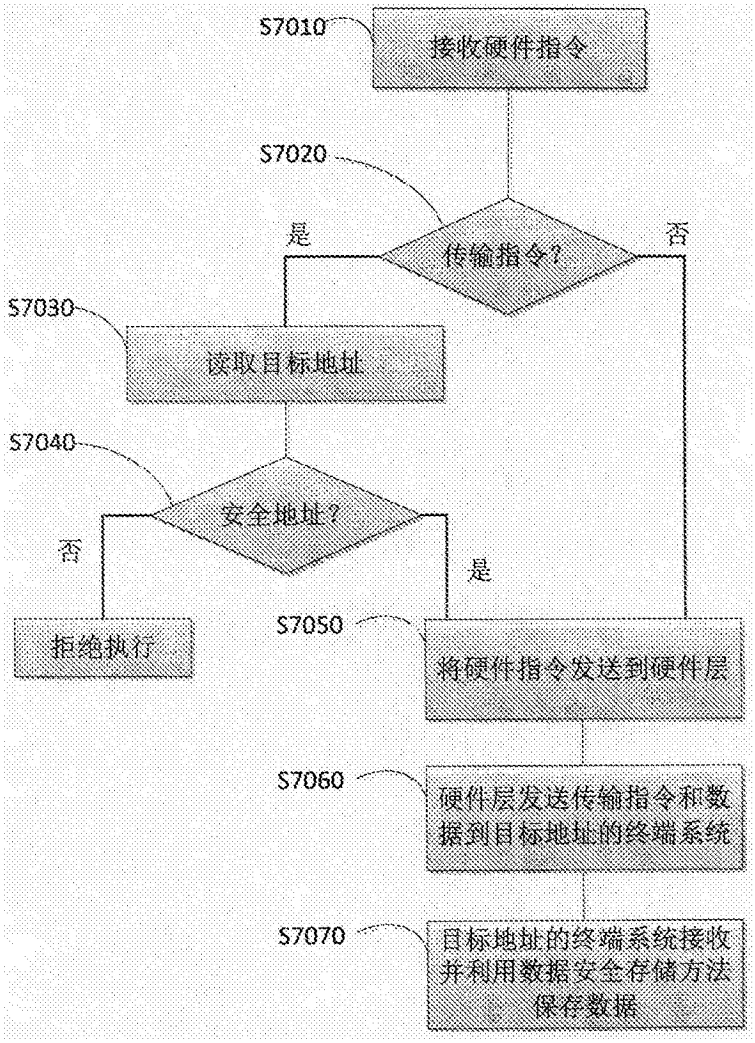

S7010
接收硬件指令
S7020
传输指令?
是
否
S7030
读取目标地址
S7040
安全地址?
否
是
拒绝执行
S7050
将硬件指令发送到硬件层
S7060
硬件层发送传输指令和数据到目标地址的终端系统
S7070
目标地址的终端系统接收并利用数据安全存储方法保存数据

图21

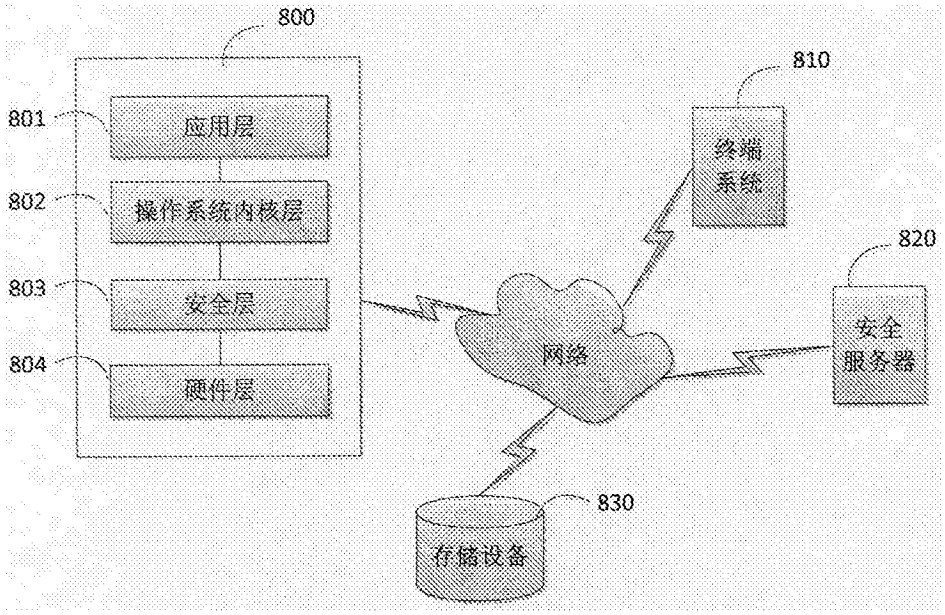

800
801
应用层
802
操作系统内核层
803
安全层
804
硬件层
810
终端系统
820
安全服务器
网络
830
存储设备

图22

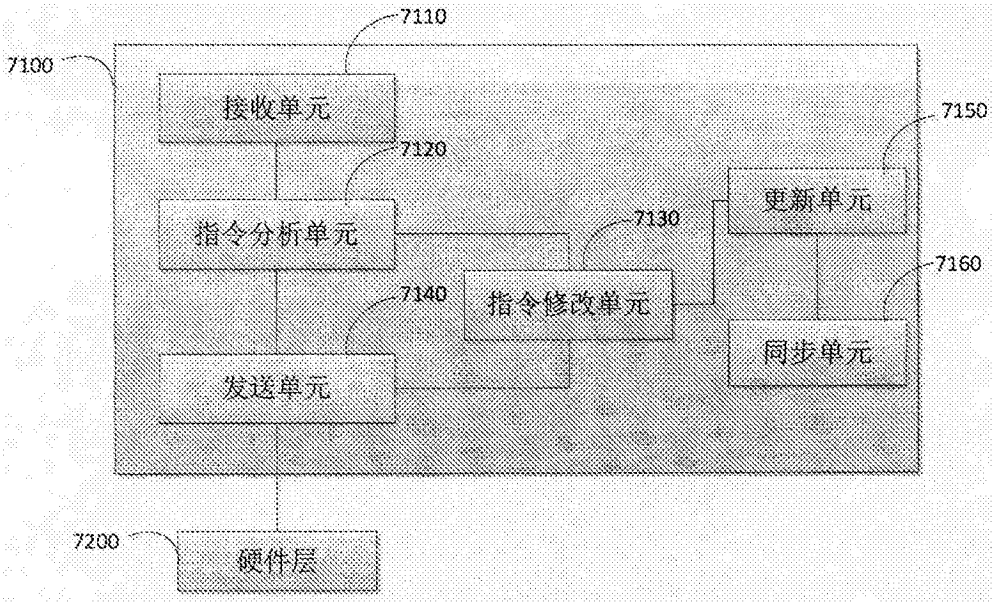

7110
7100
接收单元
7120
指令分析单元
7130
指令修改单元
7140
发送单元
7150
更新单元
7160
同步单元
7200
硬件层

图23

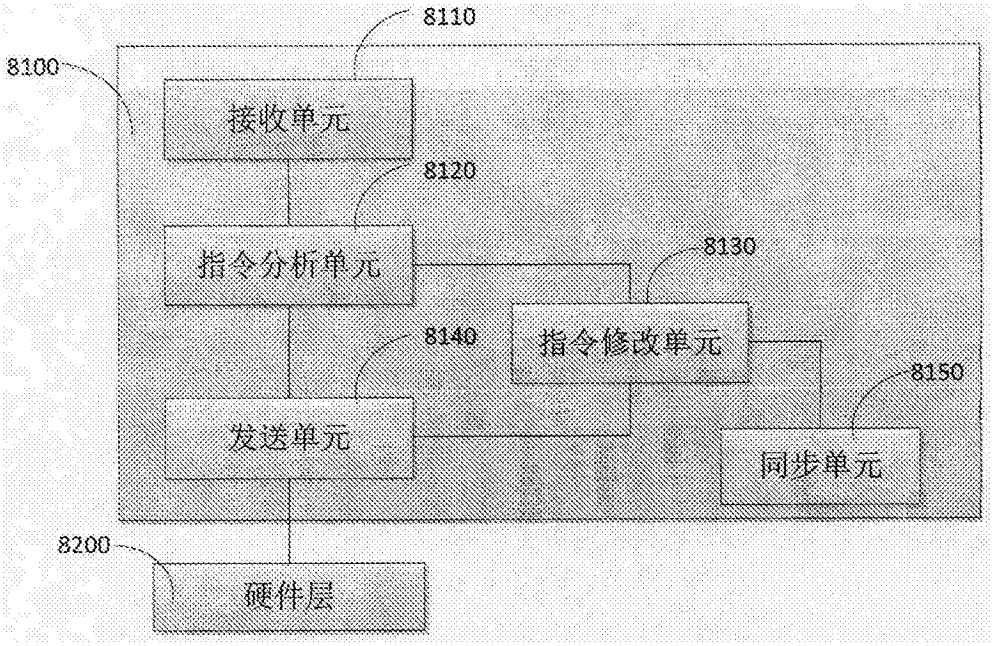

8110
8100
接收单元
8120
指令分析单元
8130
指令修改单元
8140
8150
发送单元
同步单元
8200
硬件层

图24

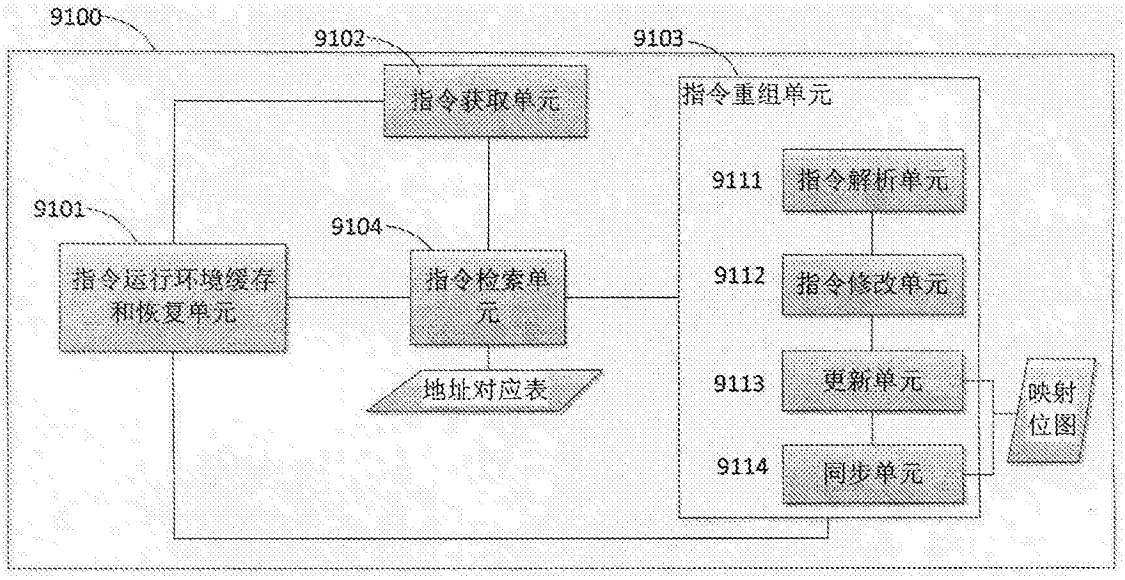

9100
9102
9103
指令获取单元
指令重组单元
9111
指令解析单元
9101
9104
指令运行环境缓存和恢复单元
指令检索单元
9112
指令修改单元
9113
更新单元
映射位图
地址对应表
9114
同步单元

图25

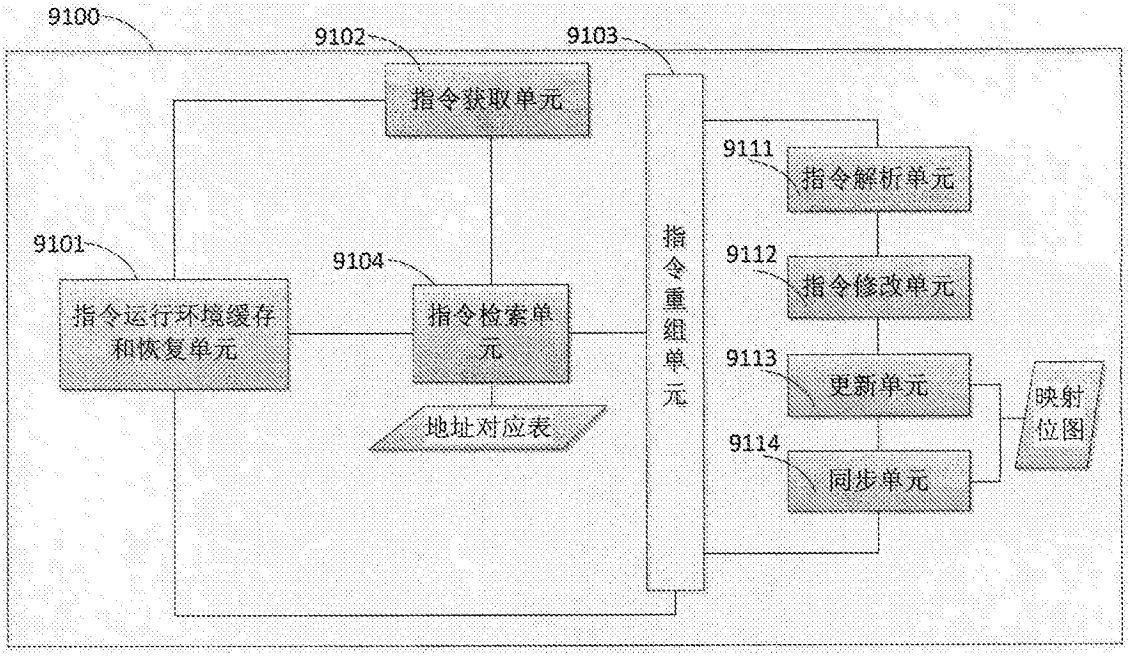

9100
9102
9103
指令获取单元
9111
指令解析单元
9101
9104
指令运行环境缓存和恢复单元
指令检索单元
指令重组单元
9112
指令修改单元
9113
更新单元
映射位图
地址对应表
9114
同步单元

图26

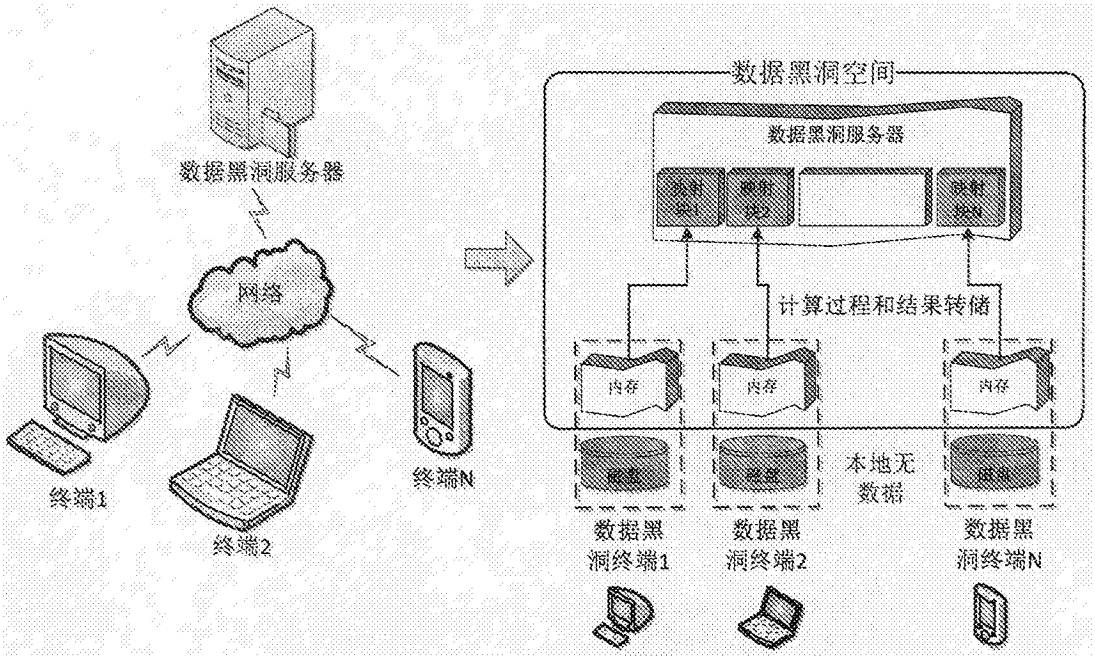

数据黑洞服务器
网络
终端1
终端2
终端N
数据黑洞空间
数据黑洞服务器
计算过程和结果转储
内存
内存
内存
本地无数据
数据黑洞终端1
数据黑洞终端2
数据黑洞终端N

图27

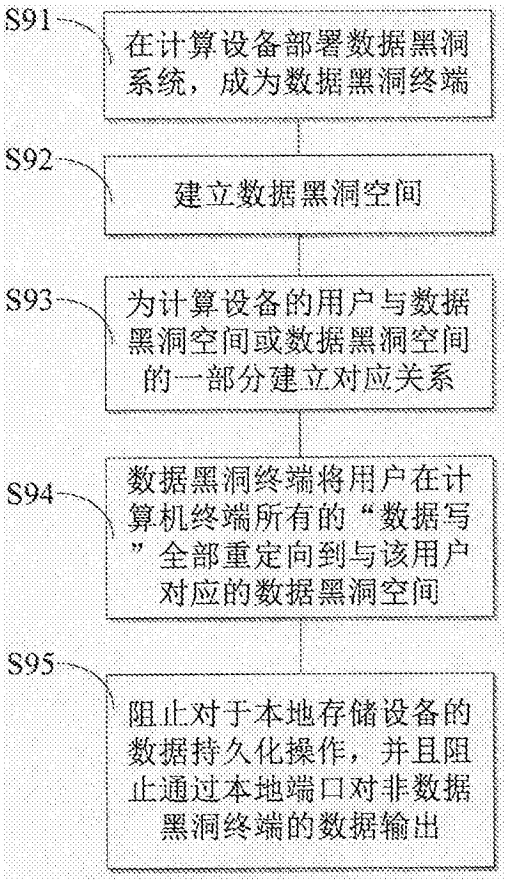

S91
在计算设备部署数据黑洞系统，成为数据黑洞终端
S92
建立数据黑洞空间
S93
为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系
S94
数据黑洞终端将用户在计算机终端所有的“数据写”全部重定向到与该用户对应的数据黑洞空间
S95
阻止对于本地存储设备的数据持久化操作，并且阻止通过本地端口对非数据黑洞终端的数据输出

图28

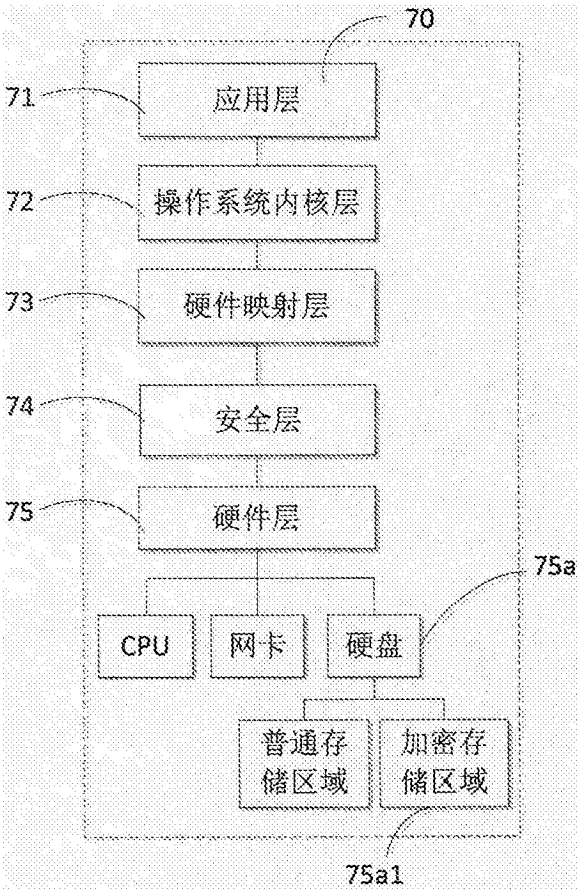

70
71
应用层
72
操作系统内核层
73
硬件映射层
74
安全层
75
硬件层
75a
CPU
网卡
硬盘
普通存储区域
加密存储区域
75a1

图29a

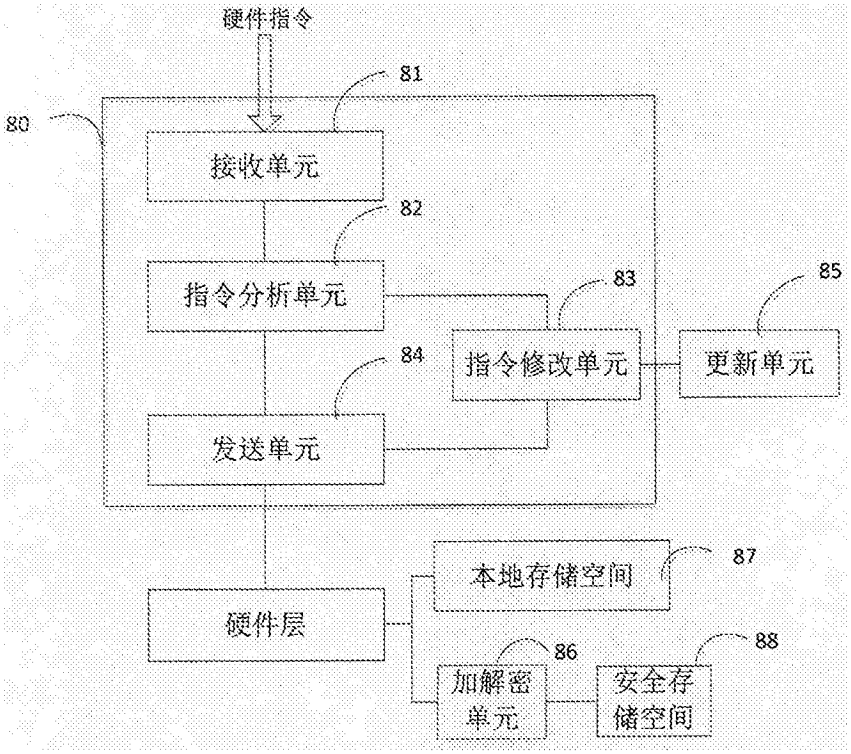

硬件指令
80
81
接收单元
82
指令分析单元
83
指令修改单元
85
更新单元
84
发送单元
本地存储空间
87
硬件层
86
加解密单元
88
安全存储空间

图29b

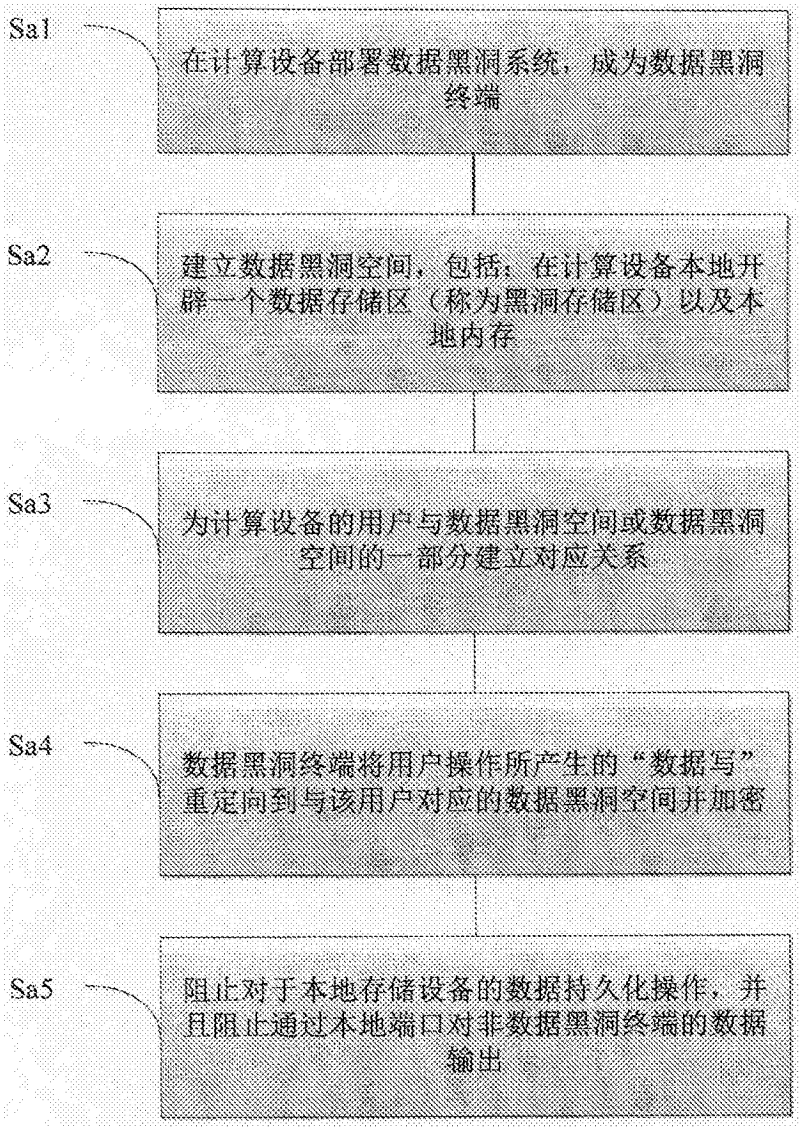

Sa1
在计算设备部署数据黑洞系统，成为数据黑洞终端
Sa2
建立数据黑洞空间，包括：在计算设备本地开辟一个数据存储区（称为黑洞存储区）以及本地内存
Sa3
为计算设备的用户与数据黑洞空间或数据黑洞空间的一部分建立对应关系
Sa4
数据黑洞终端将用户操作所产生的“数据写”重定向到与该用户对应的数据黑洞空间并加密
Sa5
阻止对于本地存储设备的数据持久化操作，并且阻止通过本地端口对非数据黑洞终端的数据输出

图30

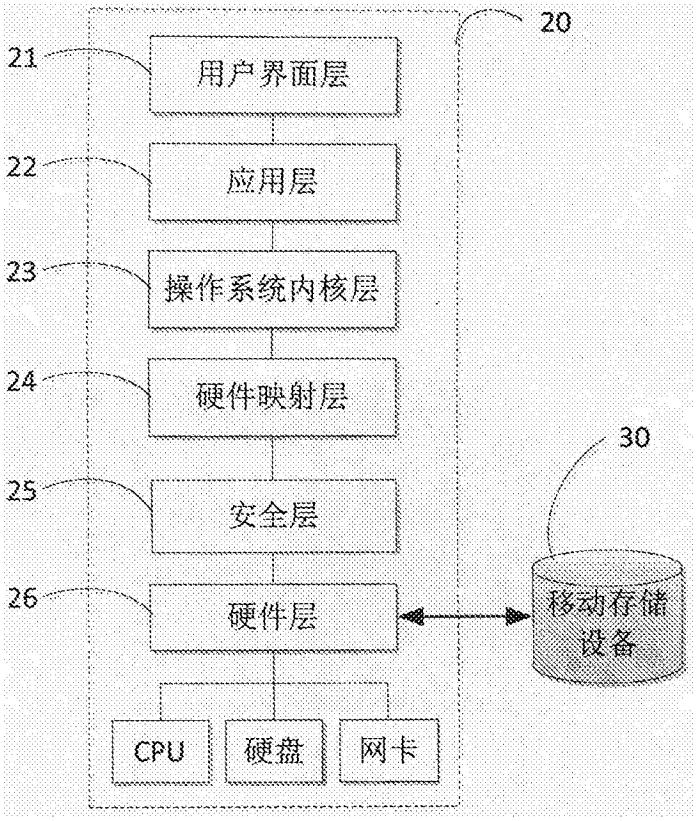

20
21
用户界面层
22
应用层
23
操作系统内核层
24
硬件映射层
30
25
安全层
26
硬件层
移动存储设备
CPU
硬盘
网卡

图31

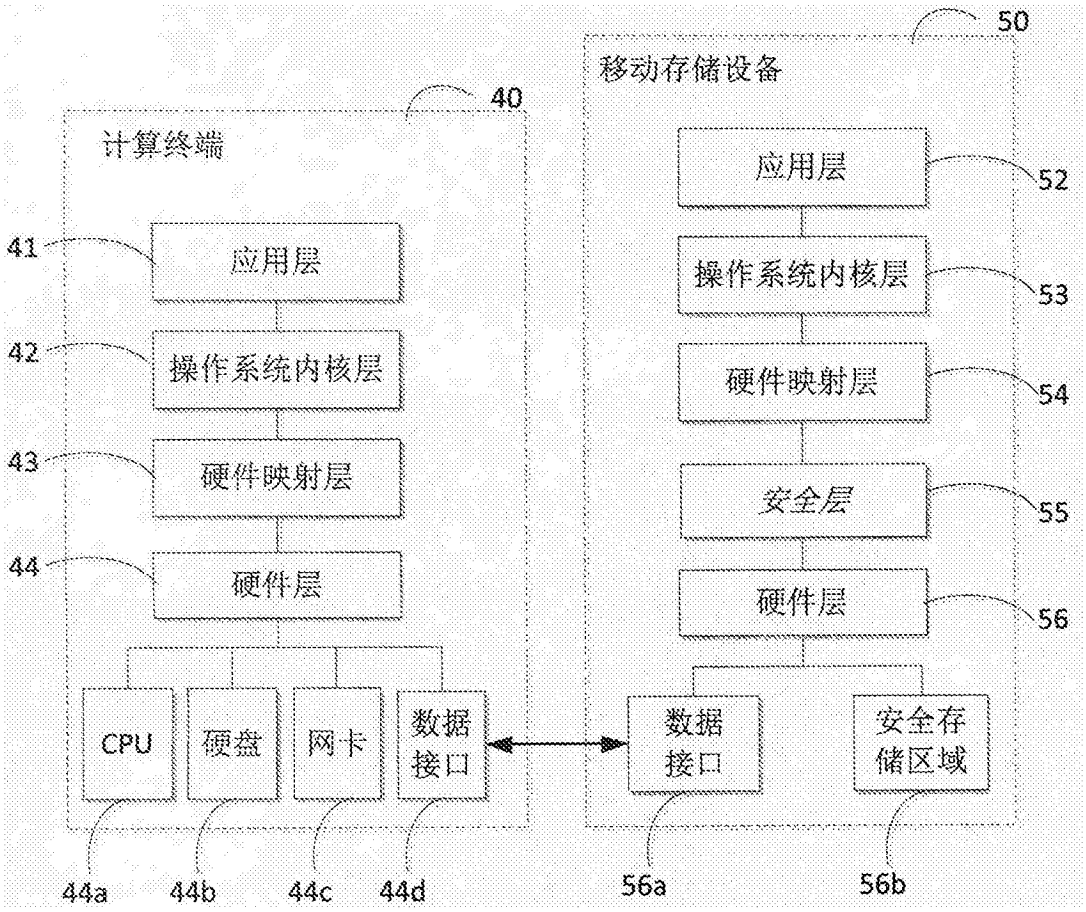

50
移动存储设备
应用层
52
操作系统内核层
53
硬件映射层
54
安全层
55
硬件层
56
数据接口
安全存储区域
56a
56b
40
计算终端
41
应用层
42
操作系统内核层
43
硬件映射层
44
硬件层
CPU
硬盘
网卡
数据接口
44a
44b
44c
44d

图32