

说明书摘要

【方法的主要步骤及优点】

本发明涉及一种基于类别的虚拟机备份方法，包括以下步骤：1) 根据备份需求，设置虚拟机的分类类别；2) 数据源模块获取需备份的虚拟机列表并保存；3) 类别识别模块接收虚拟机列表，根据各虚拟机的磁盘文件格式和文件系统格式鉴别相应虚拟机的类型；4) 基于分类类别和虚拟机类型，重新组织虚拟机列表树；5) 按照分类类别选择虚拟机备份源；6) 对虚拟机进行备份。与现有技术相比，本发明具有简化备份源选择工作、提高工作效率等优点。

权 利 要 求 书

【方法的主要步骤】

1. 一种基于类别的虚拟机备份方法，其特征在于，包括以下步骤：
 - 1) 根据备份需求，设置虚拟机的分类类别；
 - 2) 数据源模块获取需备份的虚拟机列表并保存；
 - 3) 类别识别模块接收虚拟机列表，根据各虚拟机的磁盘文件格式和文件系统格式鉴别相应虚拟机的类型；
 - 4) 基于分类类别和虚拟机类型，重新组织虚拟机列表树；
 - 5) 按照分类类别选择虚拟机备份源；
 - 6) 对虚拟机进行备份。

【对上述主要步骤中的特点作进一步限定】

2. 根据权利要求 1 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的分类类别包括按应用类型分类、按操作系统类型分类、按关键字分类、按存储路径分类或按资源池分类。

3. 根据权利要求 1 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的步骤 3) 中，鉴别相应虚拟机的类型具体为：

- 31) 类别识别模块虚拟机列表中读取虚拟磁盘元数据到本地缓存区；
- 32) 从缓存的虚拟磁盘元数据识别出根分区文件系统元数据；
- 33) 从根分区文件系统元数据读取系统配置文件元数据表项到本地缓存区；
- 34) 从缓存的系统配置文件元数据表项中解析出系统配置文件信息；
- 35) 在系统配置文件的数据区中查找类型信息；
- 36) 用类型信息标示该虚拟机，并返回步骤 31)，继续下一虚拟机的类型识别过程，直到鉴别完列表中的所有虚拟机。

4. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的虚拟磁盘元数据是指磁盘上用来表示磁盘结构及其分区的描述数据，包括磁盘分区表。

5. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的根分区是指磁盘上的系统分区。

6. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所

述的文件系统元数据指磁盘分区上文件系统用来表示文件对象及其关系的描述数据。

7. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的文件系统元数据包括：所承载的文件系统类型、分区起始扇区、分区大小。

8. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的系统配置文件是指记录系统配置、软件配置、用户配置的系统文件，存储于系统分区内。

9. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的系统配置文件信息包括：文件名、文件在分区上的起始位置、文件大小。

10. 根据权利要求 3 所述的一种基于类别的虚拟机备份方法，其特征在于，所述的数据区是指由文件起始位置和文件大小确定的磁盘块区域。

说明书

一种基于类别的虚拟机备份方法

技术领域

本发明涉及一种数据备份技术，尤其是涉及一种基于类别的虚拟机备份方法。

背景技术

【要求介绍与本题目相关现有技术的发展情况，客观地指出其存在的缺点】

随着虚拟化技术的成熟，越来越多的企业开始接受并在数据中心采用虚拟化技术，据统计，当前中国市场的虚拟化普及率已达到 50%（欧美虚拟化市场普及率高达 80%），并且还在快速增长；虚拟化技术的应用为企业节省了大量的成本，但随之也给实施高效便捷的灾备方案带来了更大的挑战。

当前主流的虚拟机（Virtual Machine, VM）备份原理如下：1）通过虚拟化管理软件（Hypervisor）提供的 Public API 获取 VM 列表，选择某（几）个 VMs 作为备份源；2）通过虚拟化管理软件（Hypervisor）提供的 Public API 将属于某（几）个 VMs 的文件拷贝到备份服务器；

上述 VM 备份方法并没有体现 VM 的应用、系统、存储等特性，在步骤 1）中获取的 VM 列表，并没有给出 VM 上的应用类型、承载的操作系统、VM 文件所在的存储等信息，为选择正确的 VM 备份源只得靠用户管理员的记忆或查询簿记信息；在如今，拥有成十上百台 VM 的虚拟化中心，要靠用户的记忆手工选择虚拟机备份源，工作量大，效率低。

发明内容

【详细说明本发明的技术方案：

- （1）对方法的主要步骤和工艺参数范围进行描述；**
- （2）对上述主要步骤和工艺参数范围的特点作进一步限定；**
- （3）改进之处及其相应带来的有益效果】**

本发明的目的就是为了解决上述现有技术存在的缺陷而提供一种简化备份源

选择工作、提高备份效率的基于类别的虚拟机备份方法。

本发明的目的可以通过以下技术方案来实现：

(1) 对方法的主要步骤和工艺参数范围进行描述：

一种基于类别的虚拟机备份方法，包括以下步骤：

- 1) 根据备份需求，设置虚拟机的分类类别；
- 2) 数据源模块获取需备份的虚拟机列表并保存；
- 3) 类别识别模块接收虚拟机列表，根据各虚拟机的磁盘文件格式和文件系统格式鉴别相应虚拟机的类型；
- 4) 基于分类类别和虚拟机类型，重新组织虚拟机列表树；
- 5) 按照分类类别选择虚拟机备份源；
- 6) 对虚拟机进行备份。

(2) 对上述主要步骤和工艺参数范围的特点作进一步限定：

所述的分类类别包括按应用类型分类、按操作系统类型分类、按关键字分类、按存储路径分类或按资源池分类。

所述的步骤3)中，鉴别相应虚拟机的类型具体为：

- 31) 类别识别模块虚拟机列表中读取虚拟磁盘元数据到本地缓存区；
- 32) 从缓存的虚拟磁盘元数据识别出根分区文件系统元数据，包括：所承载的文件系统类型、分区起始扇区、分区大小；
- 33) 从根分区文件系统元数据读取系统配置文件元数据表项到本地缓存区；
- 34) 从缓存的系统配置文件元数据表项中解析出系统配置文件信息，包括：文件名、文件在分区上的起始位置、文件大小；
- 35) 在系统配置文件的数据区中查找类型信息；
- 36) 用类型信息标示该虚拟机，并返回步骤31)，继续下一虚拟机的类型识别过程，直到鉴别完列表中的所有虚拟机。

所述的虚拟磁盘元数据是指磁盘上用来表示磁盘结构及其分区的描述数据，包括磁盘分区表。

所述的根分区是指磁盘上的系统分区。

所述的文件系统元数据指磁盘分区上文件系统用来表示文件对象及其关系的描述数据。

所述的文件系统元数据包括：所承载的文件系统类型、分区起始扇区、分区大

小。

所述的系统配置文件是指记录系统配置、软件配置、用户配置的系统文件，存储于系统分区内。

所述的系统配置文件信息包括：文件名、文件在分区上的起始位置、文件大小。

所述的数据区是指由文件起始位置和文件大小确定的磁盘块区域。

(3) 改进之处及其相应带来的有益效果：

与现有技术相比，本发明具有以下优点：

1) 本发明通过类别识别模块识别 VM 类型，并根据类别组织 VM，简化了 VM 的管理、备份源的选择工作及提高了备份效率；

2) 本发明以虚拟化平台为单元，可提供对整个虚拟化数据中心的中心化保护；

3) 本发明是主机无关的存储级保护，不会对上层主机产生影响，可靠性高；

4) 本发明只需一个用户界面（可以安装在任何地方，只要保证与数据中心联通即可），就可以对数据中心的所有虚拟化服务器进行备份；

5) 本发明能实现虚拟化服务器关机情况下的备份。

附图说明

【请提供方法的流程图，要求是黑白线条图，最好用 CAD 或 VISO 画，不能使用照片】

图 1 为本发明的流程示意图；

图 2 为 MBR 类型磁盘逻辑结构示意图；

图 3 为 NTFS 文件系统逻辑结构示意图；

图 4 为 ExtX 文件系统逻辑结构示意图；

图 5 为按应用类型分组前后 VM 列表效果图；

（5a）为按应用类型分组前 VM 列表，（5b）为按应用类型分组后 VM 列表；

图 6 为按操作系统类型分组前后 VM 列表效果图；

（6a）为按操作系统类型分组前 VM 列表，（6b）为按操作系统类型分组后 VM 列表；

图 7 为客户端、控制台及 Hypervisor 的结构图。

具体实施方式

【即具体实施例，是对上述技术方案的举例说明，应当详细描述方法步骤及具体参数】

下面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其他所有实施例，都属于本发明的保护范围。

如图 1 所示，一种基于类别的虚拟机备份方法，包括以下步骤：

1) 根据备份需求，设置虚拟机的分类类别，所述的分类类别包括按应用类型分类、按操作系统类型分类、按关键字分类、按存储路径分类或按资源池分类等。

所述的按应用类型分类，是指根据 VM 中的关键应用程序分类，所述关键应用程序主要是指数据库（DB）应用，包括：Oracle、DB2、Sybase、SQL Server、Mysql、Domino、Exchange 等当前所有主流数据库；

所述的按系统类型分类，是指根据 VM 所承载的操作系统分类，主要分为：Linux、Windows 及 Unix 三大类别，Linux 操作系统包括：Redhat 发行版、CentOS 发行版、Federal 发行版等所有当前主流 linux 发行版，Windows 操作系统包括：Windows 2000 及以后的所有 Windows 发行版本，Unix 操作系统主要是指：HP-Unix、IBM AIX、Sun Solaris；

所述的按关键字分类，是指根据 VM 机器名中包含的关键字对 VM 进行分类的一种方法，一般用来过滤 VM。

2) 客户端的数据源模块通过 Hypervisor 提供的 Public API 获取需备份的虚拟机列表并保存，若获取失败，则结束备份，若获取成功，则继续进行。

3) 类别识别模块接收虚拟机列表，根据各虚拟机的磁盘文件格式和文件系统格式鉴别相应虚拟机的类型。

鉴别相应虚拟机的类型具体为：

31) 类别识别模块虚拟机列表中读取虚拟磁盘元数据到本地缓存区；

所述虚拟磁盘元数据，是指磁盘上用来表示磁盘结构及其分区的描述数据，通常存储在磁盘的第一个扇区，又被称为主引导记录 MBR（Master Boot Record）共 512Byte，最后 64Byte 为磁盘分区表 DPT（Disk Partition Table），包含 4 个分区表项，每个 16Byte，偏移 0x04 字节表示该分区的文件系统类型、偏移 0x08 字节表示该分区的起始位置、最后 4Byte 表示该分区的大小；

32) 从缓存的虚拟磁盘元数据识别出根分区文件系统元数据, 包括: 所承载的文件系统类型、分区起始扇区、分区大小等;

所述根分区, 通常是指磁盘上的系统分区, 也即操作系统文件所在的磁盘分区, 在虚拟磁盘元数据中用 0x80 标示一个系统分区, 普通分区用 0x00 标示; 所述的识别根分区, 是指从包含在虚拟磁盘元数据中的磁盘分区表 DPT(Disk Partition Table) 找到标示为 0x80 的 16Byte 表项;

33) 从根分区文件系统元数据读取系统配置文件元数据表项到本地缓存区;

所述文件系统元数据, 是指磁盘分区上文件系统用来表示文件对象(目录/文件)及其关系的描述数据, 如: Windows NTFS 文件系统用主文件表 MFT(Master File Table) 来确定文件在磁盘上的存储位置, 每个文件对应一个文件记录, MFT 正是这些文件记录的集合;

34) 从缓存的系统配置文件元数据表项中解析出系统配置文件信息, 包括: 文件名、文件在分区上的起始位置、文件大小等; 所述的解析元数据表项, 是指基于特定的文件系统元数据格式分离出文件名、文件大小等信息的过程;

35) 在系统配置文件的数据区中查找类型信息; 所述数据区, 是指由[文件起始位置, 文件大小]确定的磁盘块区域;

36) 用类型信息标示该虚拟机, 并返回步骤 31), 继续下一虚拟机的类型识别过程, 直到鉴别完列表中的所有虚拟机。

4) 基于分类类别和虚拟机类型, 重新组织虚拟机列表树, 客户端将该虚拟机列表树通过 TCP/IP 网络发送给控制台, 由控制台负责分组显示 VM 列表树。

5) 按照分类类别选择虚拟机备份源, 在控制台可按类别, 选择某(几)类或者某(几)台 VM 作为备份源。

6) 对虚拟机进行备份, 客户端通过 Hypervisor 提供的 Public API 读取 VM 数据块, 并通过网络将数据发送给控制台, 由控制台将数据存储于介质中。

本发明如下实施例基于如下备份系统: 原备份系统由客户端 1 和控制台 3 组成, 两者之间通过以太网连接, 采用的传输协议是 TCP/IP, 同时客户端 1 与 Hypervisor2 也通过 TCP/IP 传输。为了在该系统中实现基于类别的虚拟机备份, 只需在原客户端的数据源模块 11 中加入类型识别模块 12 即可。整个系统的结构图如图 7 所示。类型识别模块负责 VM 的类型识别及 VM 列表重组织, 并将新列表发送给控制台显示, 类型识别模块对控制台来说是透明的。

为使普通技术人员对本发明中的分类过程有更直观的了解,本发明如下实施例基于如下的约定对整个备份过程进行说明;

1.假定根分区是 MBR 磁盘分区, MBR 类型磁盘逻辑结构如图 2 所示;

2.假定 Windows 根分区是 MBR 磁盘上的 NTFS 分区, NTFS 文件系统的逻辑结构如图 3 所示;

3.假定 Linux 根分区是 MBR 磁盘上的 ExtX 分区, ExtX 文件系统的逻辑格式如图 4 所示。

实施例 1

本发明实施例通过基于类别的虚拟机备份提供一种基于应用类型的虚拟机批量备份方法。本方法通过按关键应用类型分组组织 VM, 完成所有同类型虚拟机的一键保护, 分类效果如图 5 所述, 备份流程如图 1 所示, 该方法包括如下步骤:

1.通过控制台设置客户端的虚拟化分类策略为: 应用类型;

2.客户端通过 Hypervisor 提供的 Public API 获取 VM 列表, 成功则获得如图 5 (a) 所示的 VM 视图, 失败则转步骤 18, 结束任务;

3.通过 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 的 MBR 虚拟磁盘元数据, 成功则继续, 否则转步骤 18, 结束任务;

4.从上述的 MBR 磁盘元数据的 0x1BE 偏移处得到磁盘分区表 DPT, 找到系统标识为 0x80 的根分区项, 分别从该分区项的 0x04、0x08 及 0x012 偏移处提取出根分区的文件系统类型、起始扇区和分区大小;

5.以上述根分区的起始扇区为参数, 调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 的 DBR 分区元数据, 成功则继续, 否则转步骤 18, 结束任务;

6.从上述的 DBR 分区元数据的 0x30 偏移处得到 NTFS 文件系统主文件表的起始扇区;

7.以上述主文件表的起始扇区为参数, 调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取主文件表内容, 成功则继续, 否则转步骤 18, 结束任务;

8.从主文件表内容中找到路径名为 C:/Windows/System32/config/SYSTEM 的系统注册表文件的文件记录, 并从文件记录中提取出配置文件的起始扇区及大小;

9.以上述配置文件起始扇区为参数, 调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取配置文件的内容, 成功则继续, 否则转步骤 18, 结束任务;

10.在上述的配置文件内容中查找应用类型字符串, 找到则将虚拟机加入对应

的列表；

11.若 VM 类型匹配还未完成，则继续进行下一 VM 的类型匹配，转步骤 3，否则继续执行；

12.基于类型重新组织 VM 目录树，并将结果通过网络发送给控制台；控制台将显示如图 5（b）所示的 VM 目录结构；

13.通过控制台选择 VM 备份源，并将选择的数据源通过网络传给客户端；

14.若 VM 拷贝未完成，则继续，否则转步骤 18，结束任务；

15.通过 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 数据块，成功则继续，否则转步骤，结束任务；

16.通过网络将 VM 数据块发送给控制台并存储到介质，成功则继续，否则转步骤 18，结束任务；

17.若当前 VM 的数据块拷贝还未完成，则继续 VM 的数据块拷贝，转步骤 15，否则转步骤 14；

18.任务结束。

实施例 2

本发明实施例通过基于类别的虚拟机备份提供一种基于操作系统类型的虚拟机批量备份方法。本方法通过按 VM 所承载的操作系统类型分组组织 VM，完成所有同类型虚拟机的一键保护，分类效果如图 6 所述，备份流程如图 1 所示，该方法包括如下步骤：

1.通过控制台设置客户端的虚拟化分类策略为：操作系统类型；

2.客户端通过 Hypervisor 提供的 Public API 获取 VM 列表，成功则获得如图 6（a）所示的 VM 视图，失败则转步骤 25，结束任务；

3.通过 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 的 MBR 虚拟磁盘元数据，成功则继续，否则转步骤 25，结束任务；

4.从上述的 MBR 磁盘元数据的 0x1BE 偏移处得到磁盘分区表 DPT，找到系统标识为 0x80 的根分区项，分别从该分区项的 0x04、0x08 及 0x012 偏移处提取出根分区的文件系统类型、起始扇区和分区大小；

5.以根分区的起始扇区为参数，调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 分区元数据，成功则继续，否则转步骤 25，结束任务；

6.从上述分区元数据 0x04 及 0x48 偏移处的值确定操作系统类型，若 0x04 处

值为 FAT/NTFS，则为 Windows 系统，分区元数据又被称为 DBR，转步骤 7；若 0x48 处值为 0，则为 Linux 系统，分区元数据又被称为 Super Block，转步骤 10；

7.从上述的 DBR 分区元数据的 0x30 偏移处得到 NTFS 文件系统主文件表的起始扇区；

8.以上述主文件表的起始扇区为参数，调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取主文件表内容，成功则继续，否则转步骤 25，结束任务；

9.从主文件表内容中查找系统配置文件的文件记录，并从文件记录中提取出配置文件的起始扇区及大小，转步骤 16；

10.从上述的 Super Block 分区元数据的 0x18、0x20 及 0x28 偏移处分别获取数据块大小 (S)、每块组的总数据块数 (N) 和每块组的总索引节点数 (I)；

计算给定块号 (n)，所在块组 (Gn)、块组逻辑偏移 (Fn) 的公式如下：

$$G_n = (n-1) / N \quad (1)$$

$$F_n = (S * N) * G_n \quad (2)$$

计算给定索引节点号 (i)，所在块组 (Gi)、块组的逻辑偏移 (Fi) 的公式如下：

$$G_i = (i-1) / I \quad (3)$$

$$F_i = (S * N) * G_i \quad (4)$$

11.从当前索引节点 (默认为 2 号)，通过公式 (3) 及公式 (4) 计算得到块组的逻辑偏移；

12.以上述的块组逻辑偏移为参数，调用 Hypervisor 提供的 Public API 从虚拟磁盘文件获取上述块组内容，成功则继续，否则转步骤 25，结束任务；

13.从上述块组内容中找到当前索引节点及索引节点类型 (目录/文件) 及其数据区所在的块号，通过公式 (1) 及公式 (2) 计算数据区所在块组的逻辑偏移；

14.以上述的块组逻辑偏移为参数，调用 Hypervisor 提供的 Public API 从虚拟磁盘文件获取索引节点内容，成功则继续，否则转步骤 25，结束任务；

15.若步骤 13 中所述索引节点为目录，则从上述索引节点内容中找到下一级目录的索引节点号并转步骤 11，否则转步骤 17；

16.以前述配置文件起始扇区为参数，调用 Hypervisor 提供的 Public API 从虚拟磁盘文件读取配置文件的内容，成功则继续，否则转步骤 25，结束任务；

17.在上述的配置文件内容中查找操作系统字符串，找到则将虚拟机加入对应

的列表；

18.若 VM 类型匹配还未完成，则继续进行下一 VM 的类型匹配，转步骤 3，否则继续执行；

19.基于类型重新组织 VM 目录树，并将结果通过网络发送给控制台；控制台将显示如图 6（b）所示的 VM 目录结构；

20.通过控制台选择 VM 备份源，并将选择的数据源通过网络传给客户端；

21.若 VM 拷贝未完成，则继续，否则转步骤 25，结束任务；

22.通过 Hypervisor 提供的 Public API 从虚拟磁盘文件读取 VM 数据块，成功则继续，否则转步骤 25，结束任务；

23.通过网络将 VM 数据块发送给控制台并存储到介质，成功则继续，否则转步骤 25，结束任务；

24.若当前 VM 的数据块拷贝还未完成，则继续 VM 的数据块拷贝，转步骤 22，否则转步骤 21；

25.任务结束。

说明书附图

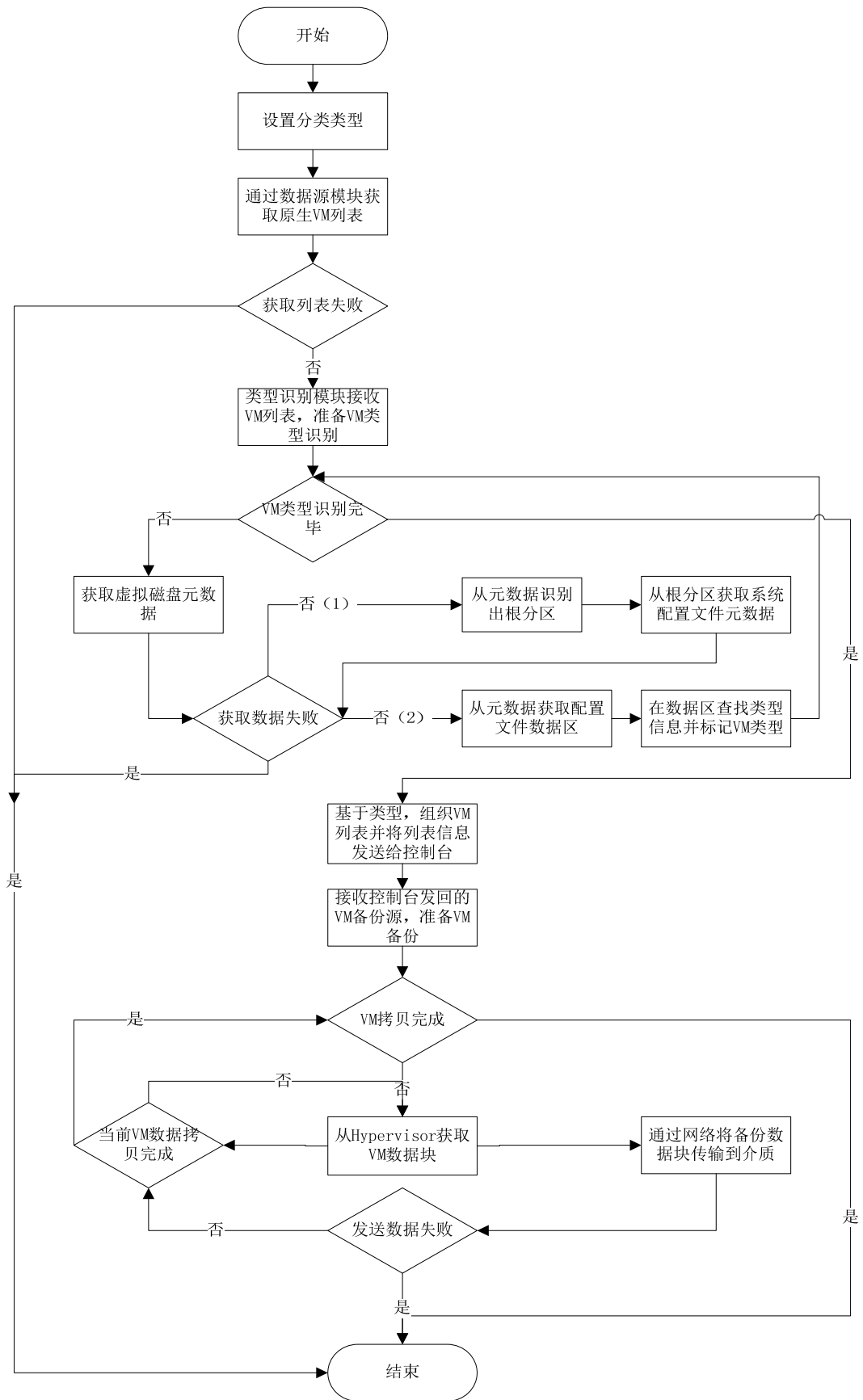


图 1

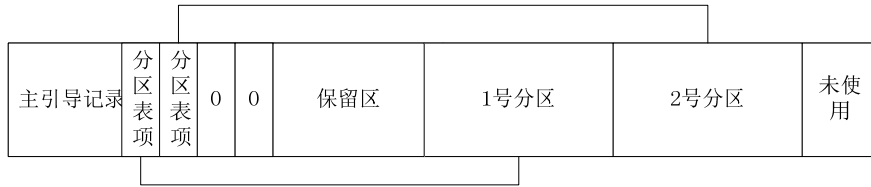


图 2

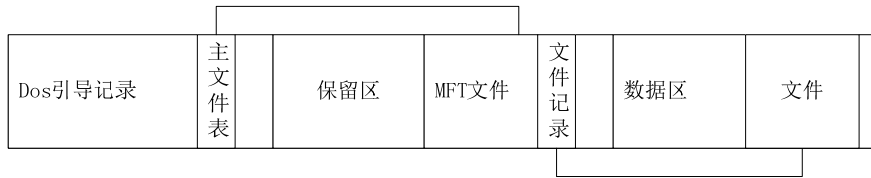


图 3

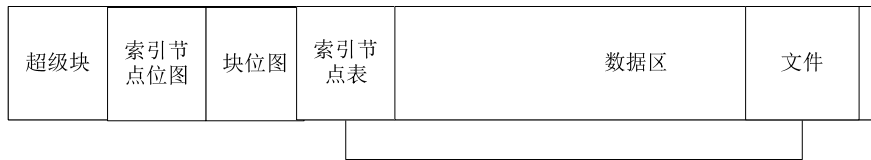


图 4

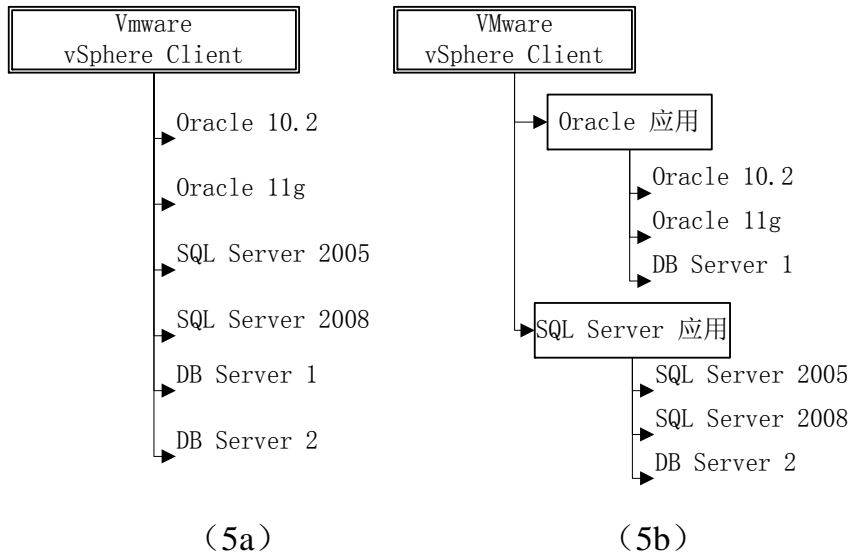


图 5

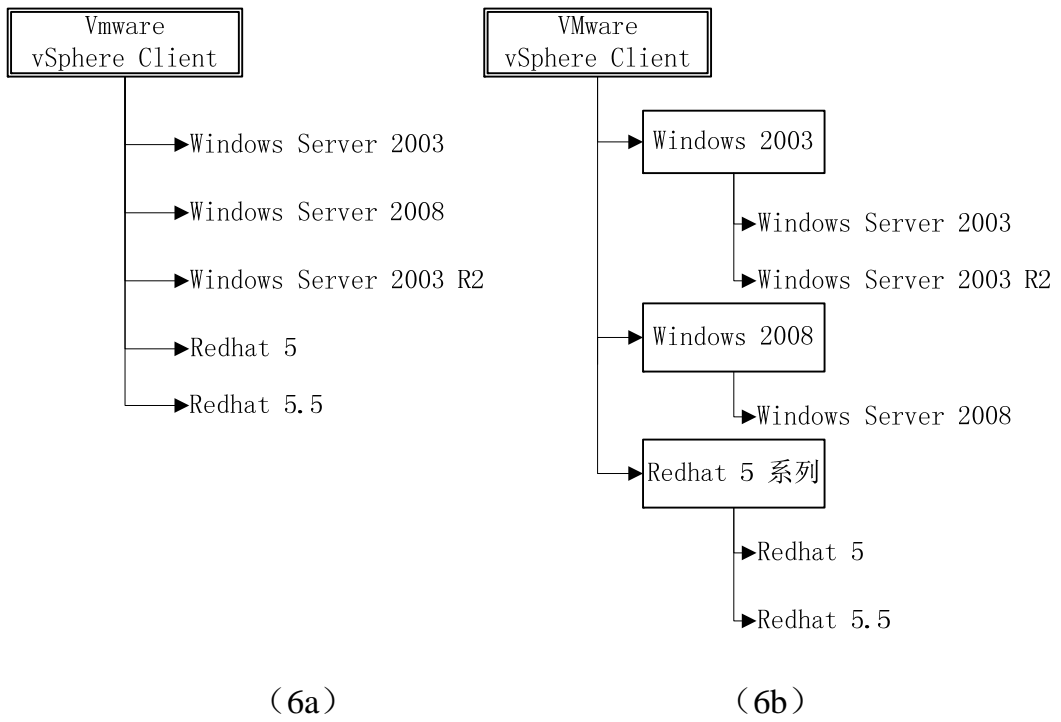


图 6

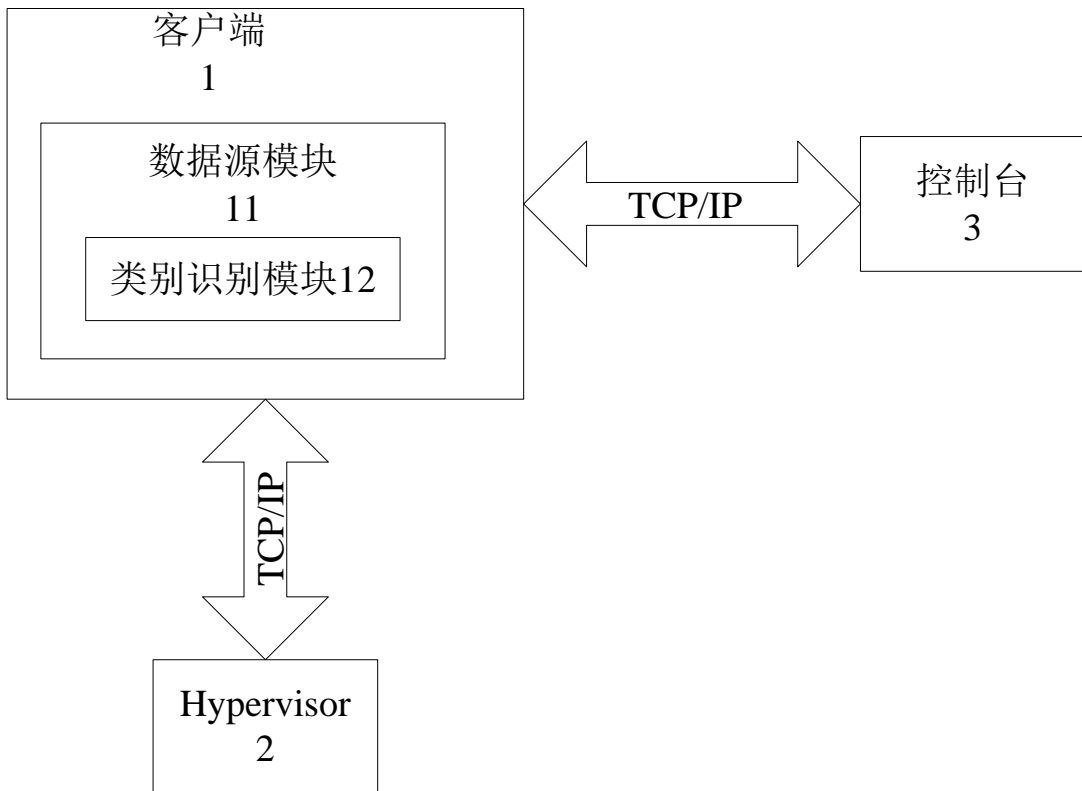


图 7