



(12) 发明专利

(10) 授权公告号 CN 1836208 B

(45) 授权公告日 2010.09.08

(21) 申请号 200480001254.X

(51) Int. Cl.

(22) 申请日 2004.07.12

G06F 9/44 (2006.01)

(30) 优先权数据

(56) 对比文件

10/692,216 2003.10.23 US

Xerox 公司. Uniform resource identifiers (URI): generic syntax.

10/692,432 2003.10.23 US

Engineering28 1. 2002, 28(1), 摘要.

10/693,072 2003.10.24 US

Bionoy Ravindran. Engineering Dynamic

(85) PCT 申请进入国家阶段日

Real-Time Distributed Systems. Transactions

2005.05.16

on software engineering28 1. 2002, 摘要、引

(86) PCT 申请的申请数据

言、第 5-7 章.

PCT/US2004/022378 2004.07.12

审查员 胡徐兵

(87) PCT 申请的公布数据

W02005/045559 EN 2005.05.19

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 R·W·麦克卢 R·R·帕朗卡

J·T·普芬宁 A·M·萨顿

M·R·布朗

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 张政权

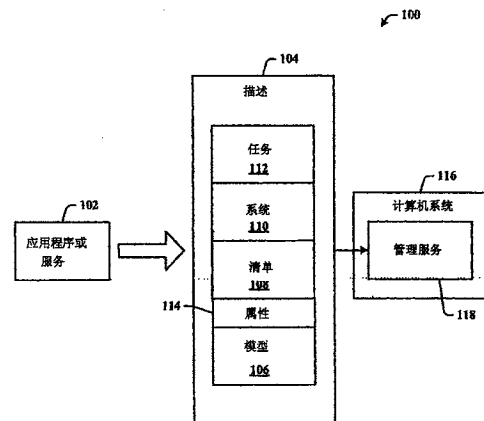
权利要求书 3 页 说明书 14 页 附图 11 页

(54) 发明名称

计算机系统和分布式应用程序基于模型管理的方法和系统

(57) 摘要

基于模型的应用程序管理构架。开发者能根据其构成组件描述应用程序或服务。所需状态可根据功能、配置、安全和性能来描述。该描述在应用程序安装时采用以配置管理服务,而服务帮助确保在诸如配置管理、问题检测、诊断以及恢复的自动化管理动作期间应用程序的可用性。



1. 一种计算机实现的用于管理应用程序或服务的基于模型管理系统,包括下列计算机可执行组件:
  - 一描述组件,其根据其构成组件描述应用程序或服务,且根据功能、配置、系统资源利用、安全、以及性能的至少之一来描述所需状态并描述纠正性动作;以及
  - 一管理服务组件,其在所述应用程序或服务的安装期间至少基于所述描述组件来配置自己。
2. 如权利要求 1 所述的系统,其特征在于,所述管理服务组件确保所述应用程序在包括配置管理、问题检测、诊断和恢复至少之一的管理动作期间的可用性。
3. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一模型组件,其模拟构成组件、健康状态以及恢复、配置设置、和管理性任务的一个或多个。
4. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一清单组件,其包含与模型和源代码属性相关联的由所述管理服务使用的机器可读形式的信息。
5. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一管理系统组件,其包括由从应用程序清单中接收的信息配置的多个服务。
6. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一管理性任务组件,其包括在应用程序清单中定义的管理性任务。
7. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一属性组件,其便于将属性数据插入源代码以指示用于检测所述应用程序诸方面的探测工具和逻辑。
8. 如权利要求 1 所述的系统,其特征在于,所述描述组件包括一管理系统组件,其使用在清单中表述的所需状态。
9. 如权利要求 8 所述的系统,其特征在于,所述管理系统组件使用由管理员更改的所需状态。
10. 如权利要求 9 所述的系统,其特征在于,所述所需状态校验依赖性并仅安装所述必需文件、设置和安全数据。
11. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态根据预定规范订购事件并转送所述事件。
12. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态周期性地收集探测工具数据和计数器数据的至少之一。
13. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态执行自动化管理任务。
14. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态限制对程序功能的访问。
15. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态执行检测问题、诊断根本原因、采取纠正性动作、以及通知管理员何时需要干预的至少之一。
16. 如权利要求 9 所述的系统,其特征在于,一个或多个所述所需状态定制在多个不同计算机上使用的政策。
17. 如权利要求 1 所述的系统,其特征在于,还包括能定义用于监视软件和硬件组件可用性的规则的规则定义语言 (RDL),所述规则定义语言 (RDL) 便于问题测试、诊断、分解、校验和通知的至少之一。

18. 如权利要求 1 所述的系统,其特征在于,还包括一统一资源标识符 (URI),其用以唯一标识抽象资源、物理资源、以及资源集合的至少之一。

19. 如权利要求 1 所述的系统,其特征在于,还包括一 URI 模板,所述 URI 模板无需检索指针就能标识所述指针并能理解所述指针的特征。

20. 如权利要求 1 所述的系统,其特征在于,还包括一探测目录,其无需参照特定实例就利用 URI 模板来描述探测工具。

21. 如权利要求 1 所述的系统,其特征在于,还包括一属性组件,其便于用于监视所述应用程序健康的源代码的属性。

22. 如权利要求 1 所述的系统,其特征在于,还包括一属性组件,其便于确定使用哪一部分源代码来确定和 / 或调整健康,以及何时执行监视规则。

23. 一种计算机实现的基于模型管理的系统,其特征在于,包括下列计算机可执行组件:

一描述组件,其根据构成组件、纠正性动作和所需状态描述应用程序、服务、和 / 或系统,所述构成组件包括以下至少之一:

一模型组件,其还包括组件模型、健康模型、配置模型、管理性任务模型、构架模型、性能模型、以及安全模型的至少之一;

一清单组件,其自所述模型组件的至少之一中生成,包括构成组件信息和所述模型组件之一的源代码属性;

一管理系统组件,其包括与所述应用程序、服务、或系统通过接口相连的一个或多个应用程序接口 (API);以及

一任务组件;以及

一管理服务组件,其在所述应用程序、服务和 / 或系统的安装期间至少基于所述描述组件来配置自己。

24. 如权利要求 23 所述的系统,其特征在于,所述应用程序接口 (API) 便于中央配置、基于角色的访问、系统监视、清单存储和编辑、事件生成和记录、探测、性能计数器处理、本地配置、安装、自动化、以及任务调度的至少之一。

25. 一种用于管理应用程序的基于模型管理的方法,包括:

使用源代码开发对应于所述应用程序组件的一个或多个模型;

执行所述源代码的属性以指示将监视哪些模型或部分;

当安装所述应用程序时产生对应于所述经模拟应用程序组件和源代码属性的清单信息的清单,所述清单信息由管理系统服务使用;

基于所述清单信息配置多个所述管理系统服务;以及

在所述清单中表述所需状态和纠正性动作。

26. 如权利要求 25 所述的方法,其特征在于,还包括校验依赖性并基于所需状态的一个或多个仅安装必需文件、设置和安全的至少之一。

27. 如权利要求 25 所述的方法,其特征在于,还包括基于所述一个或多个所需状态根据预定事件规范来订购事件并转送所述事件。

28. 如权利要求 25 所述的方法,其特征在于,还包括通过基于一个或多个所需状态周期性地收集探测工具信息、计数器信息以及测试来轮询探测工具。

29. 如权利要求 25 所述的方法,其特征在于,还包括基于一个或多个所需状态执行自动化管理任务。

30. 如权利要求 25 所述的方法,其特征在于,还包括基于一个或多个所需状态限制对程序功能的访问。

31. 如权利要求 25 所述的方法,其特征在于,还包括基于所需状态通过检测问题、诊断根本原因、采取纠正性动作、并通知系统管理员何时需要干预来监视系统过程。

32. 如权利要求 25 所述的方法,其特征在于,还包括定制政策并将所述经定制政策应用到不同计算机,以便于问题测试、诊断、分解、校验、以及通知。

33. 如权利要求 25 所述的方法,其特征在于,还包括:

确定服务的一个或多个健康状态;

发布所述应用程序的探测工具;

分析所述经发布探测工具;以及

基于经发布探测工具开发所述服务的健康模型,其中所述健康模型包括组件之间的关系信息。

34. 一种计算机实现的用于管理应用程序或服务的基于模型管理系统,包括:

一装置,用于根据其构成组件描述应用程序或服务,描述纠正性动作,以及根据功能、配置、安全、以及性能的至少之一来描述所需状态;

一装置,用于表述管理信息以及所述应用程序的源代码以便于确定所述应用程序的健康;

一装置,用于使用 URI 来标识抽象或物理资源;以及

一装置,用于在安装所述应用程序以配置管理服务组件期间部分地基于构成组件来配置所述管理服务组件。

35. 如权利要求 34 所述的系统,其特征在于,所述应用程序或服务是分布式的。

36. 一种计算机实现的用于管理应用程序或服务的基于模型管理设备,包括:

使用源代码开发对应于所述应用程序或服务组件的一个或多个模型的装置;

执行所述源代码的属性以指示将监视哪些模型或部分的装置;

产生对应于所述经模拟应用程序或服务组件和源代码属性的清单信息的清单的装置,所述清单信息由管理服务组件使用;

基于所述清单信息配置多个所述管理服务组件的装置;以及

在所述清单中表述所需状态和纠正性动作的装置。

37. 如权利要求 36 所述的设备,其特征在于,还包括校验依赖性并基于一个或多个所需状态仅安装必需文件、设置和安全的至少之一的装置。

## 计算机系统和分布式应用程序基于模型管理的方法和系统

[0001] 相关申请

[0002] 本申请与以下共同未决的美国专利申请相关：序列号 10/692, 216 于 2003 年 10 月 23 日提交的题为“Model-based Management of Computer Systems and Distributed Application”（“计算机系统和分布式应用程序的基于模型管理”）的申请；序列号 10/693, 072 于 2003 年 10 月 24 日提交的题为“Scalable Synchronous and Asynchronous Processing of Rule Monitoring”（“规则监视的可升级同步和异步处理”）的申请；序列号 10/692, 660 于 2003 年 10 月 24 日提交的题为“Rules Definition Language”（“规则定义语言”）的申请；序列号为 10/693, 588 于 2003 年 10 月 24 日提交的题为“Using URI's to Identify Multiple Instances with a Common Schema”（“使用 URI 来标识带有共同模式的多个实例”）的申请；以及序列号 10/692, 432（律师案号 MSFTP522US）2003 年 10 月 23 日提交的题为“Use of Attribution to Describe Management Information”（“使用属性以描述管理信息”）的申请，在此完整引用作为参考。

### 技术领域

[0003] 本发明涉及计算机系统，尤其涉及计算机系统和应用程序的管理。

### 背景技术

[0004] 常规的系统管理是非常特别的。应用程序开发者不具有用于管理其应用程序并获取高可靠性的结构化框架。应用程序的期望行为很大程度上是反工程化的。程序开发者并不具有用于如何思考管理其应用程序并获取高可靠性的结构化引导和框架。此外，操作系统不为开发者进行有效动作而提供整体系统。

[0005] 对于操作者而言，系统的复杂性正在变得太大而难以理解。追踪依赖性和潜在误差花费了大量时间。当得不到探测工具时，操作者需要进行过程转储（process dump），因为这是确定说明请求正在执行和当前状态的唯一方法。

[0006] 当今系统在向用户警告可能和真正问题方面做得较差。用户不能轻易分辨已安装的是什么应用程序，也不知道系统和应用程序是否具有正确的文件和版本、是否为其使用它们而正确配置、是否对环境安全配置、以及它们是否最优地操作并且没有耗尽资源。此外，在多台机器上也不能方便地调试应用程序 - 没有公共的应用程序和交易环境。

[0007] 操作者也不能轻易断定应用程序依赖性，是文件、组件、配置设定安全终止，还是类似存储区的装置、网络和路由器。系统既不能警告用户改变可能会损坏其它应用程序，也不能使用该信息来帮助识别根本原因。

[0008] 当前反应式监视是最普遍的，其中警告使用户知道有故障，但不知道问题的起因。先进的脚本和供应商可提供信息更丰富、更可控诉的警告，但缺乏用于执行根本原因分析的基础设施。常常需要用于故障检修的其它诊断。然而，反应式监视的一个问题是警告常常为时已晚 - 应用程序已经不能为用户所用。通过触发 failover 或者用负载平衡装置使服务器离线，有助于监视。然而，系统应当足够智能以在可能问题变成故障之前检测应用程

序中的可能问题。

[0009] 其它问题仅可通过查看多台机器和客户机来检测。示例包括分布式入侵检测和应用程序性能的下降。如果管理员在其控制上具有看到从期望性能的偏离的能力,能够在捕捉到快照时追踪配置变化的根本原因,并在用户抱怨之前解决问题,则可避免许多大规模的网络性能问题和发生故障停机角度的历史数据或趋势来确定。管理员常常不知道他们的复制后备记录是否有问题,且需要首先运行该服务并记录操作方法以建立带有警告和临界阈值的基准。

[0010] 所需要的是用于管理基础设施的改进机制。

## 发明内容

[0011] 为了提供对本发明某些方面的基本理解,以下提供本发明的简要内容。本“发明内容”部分并非是本发明的广泛纵览。它并非旨在标识本发明的关键或重要元素或是勾划本发明的范围。它的唯一目的是用简化形式介绍本发明的某些概念,作为以下提供的更详细说明的序言。

[0012] 在此揭示和声明的本发明在其中一方面中,包括提供使开发者能根据其组件描述应用程序或服务的创新框架的基于模型的管理系统。开发者可根据功能、配置、安全和性能来描述应用程序或服务的所需状态。该描述或模型被提供以应用程序,并由系统在安装时使用以配置管理服务。计算机系统在安装时采用开发者的描述以配置管理服务。管理服务帮助确保在诸如配置管理、问题检测、诊断以及恢复的自动管理动作期间应用程序的可用性。该模型还描述管理员可执行的公共任务。

[0013] 基于模型的管理构架包括以下部分:组成应用程序的组件的诸模型,例如健康状态和恢复、配置设定、以及管理任务;源代码中指示用于监视的探测工具和逻辑的属性;与应用程序一起运送的一个或多个清单,它显示包含有由管理系统服务所使用机器可读形式的来自模型和源代码属性的信息;由应用程序清单中信息所配置的多种服务组成的管理系统;以及在清单中定义的应用程序的管理性任务。

[0014] 基于模型管理构架的系统组件包括确保应用程序可用性所必需的服务。系统使用在清单中示出并由管理员更改的所需状态来执行以下动作:校验依赖性并仅安装必需文件、设定和安全的安装;预订事件并将其按指定进行传递的事件订阅;周期性地收集探测工具和计数器的轮询探测工具;执行自动管理任务的预定任务;限制对程序功能访问的基于角色访问;检测问题、诊断根本问题、采取纠正动作、并在必需干预时通知系统管理员的监视功能;以及用于定制以上政策并应用到许多机器的中央配置。

[0015] 在其中的另一方面,把基于模型管理系统应用到硬件和软件的分布式网络中。相应描述和管理了本地和远程应用程序的组件,以及本地和远程机器和服务。

[0016] 为了完成前述内容和相关目标,本发明的某些说明性方面结合以下说明书和附图进行详细描述。然而这些方面仅仅示出了本发明诸原理可在其中采用的各种方式的其中几种,且本发明旨在包括所有这些方面及其等同方案。结合附图阅读以下详细说明,本发明的其它优点和新特征将变得清楚。

## 附图说明

- [0017] 图 1 示出了根据本发明便于应用程序的基于模型管理的构架。
- [0018] 图 2 示出了与描述基于模型管理构架相关的示图。
- [0019] 图 3A 示出了与本发明基于模型管理构架的模型组件相关联的框图。
- [0020] 图 3B 示出了与本发明基于模型管理构架的清单组件相关联的框图。
- [0021] 图 3C 示出了根据本发明的基于模型管理构架,用于管理应用程序或服务的系统组件的核心系统 API 的框图。
- [0022] 图 3D 示出了本发明基于模型管理构架的系统组件的管理相关 API 的框图。
- [0023] 图 3E 示出了本发明基于模型管理构架的任务组件的子组件。
- [0024] 图 4 示出了基于模型管理过程的一般流程图。
- [0025] 图 5 示出了实现基于模型管理的过程的更详细流程图。
- [0026] 图 6 示出了实现基于模型管理所需状态的过程的流程图。
- [0027] 图 7 示出了可操作以执行所揭示构架的计算机的框图。
- [0028] 图 8 示出了根据本发明示例性计算环境的示意框图。

## 具体实施方式

[0029] 本发明参照附图进行说明,其中贯穿所有附图相同数字代表相同元件。在以下说明中为作解释,陈述有很多具体细节以便提供对本发明的全面理解。然而没有这些具体细节也可实践本发明是显然的。在另外的实例中,为便于描述本发明,众所周知的结构和设备以框图形式示出。

[0030] 如在本申请中所用,术语“组件”和“系统”旨在指计算机相关实体:硬件、硬件和软件的组合、软件、或是执行中的软件。例如,组件可以,但不限于,运行于处理器上的过程、处理器、对象、可执行程序、执行的线程、程序、和 / 或计算机。作为说明,运行于服务器上的应用程序和服务器都可作为组件。一个或多个组件可驻留于过程和 / 或执行的线程中,并且组件可本地化在一台计算机上和 / 或分布在两台或多台计算机上。

[0031] 术语“推断”在此使用时,一般是指从经事件和 / 或数据捕捉的一组观察中推理或推断系统、环境、和 / 或用户的状态的过程。推断用来识别专用环境或动作,或产生例如状态的概率分布。该推断是或然性的 - 即,相关状态的概率分布计算是基于对数据和事件的考虑。推断也可指用来从一组事件和 / 或数据中组成更高层事件的技术。这种推断导致了基于一组观察事件和 / 或存储事件数据对新事件或动作的构建,不管这些事件是否紧密时间接近性相关,及是否这些事件和数据来自一个或若干个事件和数据源。

[0032] 现在参看图 1,根据本发明便于应用程序或服务的基于模型管理的说明性构架 100。该基于模型管理方法使开发者能根据其组成组件描述应用程序或服务 102,并根据功能性、配置、安全以及性能描述所需状态。因而,应用程序或服务描述 104 便于根据一个或多个可管理组件来描述应用程序或服务 102,包括至少一个模型组件 106、清单组件 108、系统组件 110 和任务组件 112。基于模型管理系统 100 利用属性组件 114 来便于源代码从模型组件 106 到清单组件 108 的属性。

[0033] 计算机系统 116 在应用程序 102 安装时使用应用程序或服务描述 104 来配置与计算机操作系统相关联的管理服务 118。然后管理服务 118 帮助确保在诸如配置管理、问题检

测、诊断以及恢复的自动管理动作期间应用程序或服务 102 的可用性。模型 106 还描述了管理员可执行的任务。基于模型管理构架 100 促成所有权的更低的总成本,并在从开发到配置、操作、以及商业分析的应用程序生命周期中使用。通常,开发者由根据应用程序如何工作、其组成组件、开发者定义并选择监视的所需健康状态、至少关于它将如何安装且应用程序或服务将需要哪些设定的配置方面、以及管理性任务及其调度来创建应用程序或服务的一个或多个模型开始。然后模型的源代码归属(加标记)于用来表示的特定区域。

[0034] 模型被积累成探测工具清单。模型往往是文本文档、电子制表文档等等、以及通过代码、脚本、工具变换或者手动地变换成往往是多个 XML 模式的清单,并进一步进行机器处理和机器阅读的结构化文档的形式。也就是说,模型文档更人工可读,而清单则更为机器可读。然后使用清单来促进系统管理。

[0035] 属性子组件 114 与源代码属性相关联。属性用于表达管理信息以及与其相关的代码。如果没有属性,将需要编写两段独立代码 - 一段用于正常的应用程序处理而另一段用于将其展现给管理。源代码中的属性被用来描述应当使用代码的哪些部分(称为探针)来确定和/或纠正健康状况,并指定何时执行监视规则。可从访问现有操作系统 API(应用程序接口)的组件或从载入运行应用程序或服务的组件来展现这些探针。在两种情形中,开发者添加属性以指示应当展现组件中类型的哪些子集,以及应当如何标识它们。使用管理员名空间中的 URI(统一资源标识符)来标识探针。在运行时,通过在计算机上所有探针的目录中标识探针,并跟随有关探针的相关联信息来检索探针。

[0036] 源代码属性还可向监视服务提供指令,例如应当被用作监视规则并在起动、周期性轮询、事件上运行等等时载入属性功能。该属性可以探测工具的相同方式进行自动处理,并置入清单中。因而,属性并非仅仅是探测工具,而且也可用于其它管理目的。也可使用属性以支持管理性任务和/或纠正性动作。

[0037] 现在参看图 2,示出了与描述基于模型管理构架 100 的主要组件相关的示图 200。该构架包括参照图 3A 描述的模型组件 106、参照图 3B 描述的清单组件 108、参照图 3C 和图 3D 描述的系统组件 110、以及参照图 3E 描述的任务组件 112。已经描述了属性,并将贯穿本说明书进行陈述。

[0038] 参照图 3A,有与本发明的基于模型管理构架的模型组件 106 相关联的说明性框图。模型为组成应用程序的组件、健康状态和恢复、配置设定、以及管理性任务进行开发。

[0039] 为了支持,有用于模拟系统的任意及所有组件(以及与其相关联的关系、依赖性、以及服务角色)的组件模型子组件 300。组件模型 300 描述文件、配置、能安装应用程序的不同方法、等等。

[0040] 可开发健康模型子组件 301 以描述各种故障状态,以及应用程序或服务会发生故障的方式。健康模型 301 描述要自动化健康特征需采取的步骤。健康模型 301 表示至少故障状态、状态检测、校验、诊断、以及系统状态的分解。健康状态可根据要符合怎样的标准才能被视为完全健康、完全故障和任意中间状态来进行描述,这些中间状态例如性能降低、部分工作、正在工作的某些定制功能、以及应用程序或服务在传送期望层的服务。健康也可视为功能会是完善的,但性能是指示应用程序或服务不健康的次要标准。

[0041] 配置模型子组件 302 与模拟系统配置相关联。配置模型 302 被用以描述应用程序设定、用户控制、缺省值、各种限制等等。管理性任务模型子组件 303 与模拟管理性任务相



关联,并包括用户可在一系统上采取的动作,诸如可从健康模型 301 调用的开始、停止、添加用户、添加数据库、以及纠正动作。模型 302 列举所有能对应用程序或服务作的动作。构架模型 304 被用以描述分布式环境和相关联配置(例如通常与具有相同或相似的硬件和软件设定和配置的客户机大型网络相关联)以及分布式数据库。因而,本地应用程序可依赖于远程磁盘阵列。在部署时,需要在部署层用清单和 URI 来例举磁盘阵列。因为 URI 是与机器无关的,分布式系统也可获得本发明基于模型管理系统的优点。可开发性能模型 305 以描述开发者希望使用用于监视应用程序或服务性能的尺度的方法。这与系统的健康是紧密相关的。可产生描述与应用程序或服务相关联的安全类型的安全模型 306。

[0042] 注意在此提供的模型数量不是穷尽的,因为开发者可提供许多不同的用于管理应用程序或服务各个方面的模型。

[0043] 本发明可采用各种基于人工智能方案用于完成其各个方面。例如,参照模型,通过自动分类系统和过程可促进用于确定对给定实例或实现可利用哪些模型的过程。此外,可使用这种分类器以建立系统的开始检测系统模式的操作简介,并学习什么是良好状态、较差状态、以及成功和不成功的交易。然后该信息被反馈回相应模型,并用作随后系统的经更新模型。这种分类可采用基于概率和/或统计的分析(例如包括分析实体和成本)以预测或推断用户希望自动执行的动作。例如,可采用支持向量机(SVM)分类器。可采用包括提供不同类型不相关性的贝叶斯网络、判定树、以及概率分类模型等其它分类方法。在此使用的分类还包括被用来开发优先级模型的统计衰退。

[0044] 从本说明书可以理解,本发明可采用显式培训(例如通过一般培训数据)以及隐式培训(例如通过观察用户行为、接收外来信息)的分类器,从而使用分类器根据预定标准来自动确定,例如对于给定实现使用什么初始设定,然后随着系统成熟并体验相对于数据、安装应用程序数量、以及要交互节点数的各种负载条件来调整设定。例如,对于较好理解的 SVM,通过分类器构造器和特征选择模块中的学习或培训阶段来配置 SVM。分类器是把输入属性向量  $x = (x_1, x_2, x_3, x_4, x_n)$  映射到输入属于一类的置信度的函数 - 即  $f(x) = \text{confidence}(\text{class})$ 。在管理系统的情形中,例如,属性是所需状态的系统参数,而类是感兴趣的类别或区域(例如所有磁盘、所有固有过程)。还可采用分类器来捕捉和分析交易记录、寻找模式、并通过寻找成功和不成功模式来诊断系统。

[0045] 配置健康涉及,例如,把队列尺寸从 5 改变到 10,并确定什么在应用程序、服务或系统上影响该改变。同样也适用于安全和性能,其中可采用分类器以监视性能计数器并相应地改变系统以最优化性能。也可模式来监视和分析安全,其影响可用来建议或改变安全政策。因而,可以理解,健康是可应用于系统许多领域的广泛概念。在整个系统范围中,性能可较好而安全则可较差。因而,由本发明提供的跨越本系统许多规则的整体综览是有利的。

[0046] 管理员的所需状态可在代码中得以表达,它在清单中表面化并被传递用于由监视服务进行的监视。系统可基于清单中的指令监视应用程序或服务,并在应用程序或服务不满足性能时警告管理员,以及基于这些指令采取纠正动作。例如,当电子邮件的测试设定未得到维持并在一段时间内落于阈值之下,可添加另一机器直到负载减退,也可使用网络通信量作为增加资源量以处理给定负载的触发器。目标被尽可能地自动化,从而管理员仅在需要手动动作时才会参与。

[0047] 本发明的基于模型管理系统是可调整的。它是基于组件的,其中组件包括大部分。

因而,可将系统减至其最小可管理尺寸并组成备份。在数据库中,例如,具有带有实例的应用程序、数据库、表格、以及经存储过程,并可减小至单个文件。考虑一个 401k 应用程序。401k 应用程序可依赖于数据库、web 服务器、以及客户自己的经营逻辑,减至依赖于操作系统及相关环境的数据库。根据本发明的新颖方面,需要在各个层次上进行管理和报告。通过组件之间的关系来描述应用程序。这些关系可表达单个应用程序是如何装配的(例如 SQL 服务器包含服务、实例、以及数据库)、平台要求(例如操作系统和其它应用程序)以及与其它组件的通信(与 SQL 服务器相连的 web 服务器)。独立的管理员可能会关心数据库和单台机器,财务管理员则会关心 401k 应用程序,而 CIO 会关心所有的应用程序和机器。模型、报告和所需状态应当处理一切,从而可参照单一尺度来确定系统是否按期望运行。

[0048] 所有的模型都系于 URI 名空间中,提供了一种标准方法用于:导航系统、列举所安装的所有组件、并询问该组件它提供什么、视什么为健康、具有什么事件、在最近一天或数小时内发生了什么错误事件、包括什么配置设定、在最近一小时发生什么变化等等。

[0049] 现在参看图 3B,有与本发明基于模型管理构架的清单组件 108 相关联的示图框。与应用程序一起运送的清单包含来自模型的信息和由管理系统服务所使用的机器可读形式的源代码属性。在清单中定义应用程序的管理性任务。可有对应于模型产生的众多清单,包括:与组件依赖性、组件之间关系、以及服务角色相关联的第一清单子组件 307;与事件、探针、规则以及动作相关联的第二清单子组件 308;与设定和断定相关联的第三清单子组件 309;与命令(即 cmdlet)和管理性角色相关联的第四清单子组件 310;与分布式环境相关联的第五清单子组件 311;以及与部署相关联的第六清单子组件 312。

[0050] 清单是开发者和操作团队和管理员之间的“桥梁”,并由在模型中彻底搜寻属性化代码的工具自动创建。由设定引擎使用组件清单 307 来确定如何安装应用程序或服务。它描述了逻辑组件、文件(文件应当安装在哪里)、以及配置设定(或任何设定)。依赖性是在安装之前需要进行定义的,并包括各种角色,从而应用程序可以各种模式安装,带有变化的安全度和不同的操作简介。组件清单 307 使用户和/或系统更容易知道要手动和自动做的是。清单细化可至每个组件一个清单。

[0051] 通常,会安装比真正需要多得多的文件。清单允许仅安装那些需要的文件。这至少改进了性能和安全性。软件依赖性在清单 307 中定义。在应用程序层中,依赖性可对单个机器特定,并定义组件关系和硬件资源。可由清单来描述一计算机,例如,应将应用程序部署在特定制造商的双处理器机器或 4- 处理器机器的接口上。该清单 307 按照本实现所需的硬件细化程度来描述处理器、存储器、驱动器等等。因而,管理可比常规系统中的响应式更为主动。例如当监视到随时间变化的系统温度以及监视电源轨压但发现是足够时,可确定硬盘故障由热故障导致。

[0052] 健康模型 301 被用以产生健康清单 308。使用属性和其它工具来从健康模型 301 填充健康清单 308。事件不是在模型 301 中而是在资源文件中调用的。一工具彻底搜寻了资源文件和属性化源代码,并填充健康清单 308。可通过观察预定事件序列或者监视性能计数器阈值来监测故障状态。可向系统提供如何处理这种故障状态的指令。健康模型被变换成规则。健康清单 308 包括带有诸如事件 1、事件 2、时间 3 等等的参数的规则类型事件序列。

[0053] 配置模型 302 描述所包括的是哪些设置,并变换成向系统提供指令模式以创建安

装组件时设定的设置和声明清单 309。

[0054] 通过小命令和管理角色清单 310 将管理性任务模型 303 变换成动作。例如,如果需要数据备份,小命令是用以推动备份任务的真实代码或 URI。当需要执行许多管理任务时,清单 310 向那些命令或者可能向代码提供 URI 路径。小命令可通过代码中的声明进行处理,或者需要外部代码。管理角色是支持例如管理该应用程序或服务的多类用户以及他们每个可行使的控制层的另一抽象。这与基于角色访问相关联。需要描述各种用户角色及其允许能力的元数据。角色涵盖系统的所有方面 - 允许谁安装、谁能改变监视、谁能查看健康、谁能消除警报、谁能采取以上各种动作等等。

[0055] 任务模型 303 定义开发者认为管理员应当做的事,如在清单 310 中得以表述并由操作团队根据其环境进行定制。这些定制可在类层和实例层上完成。可在类层、实例层上于清单中作改变,并且在运行时可直接作改变。所揭示的基于模型管理构架的一极强大特征是首先在类层上描述能力,而在运行时访问则针对实例空间。

[0056] 构架模型 304 展现了分布式组件清单 311 和部署清单 312。例如,机器之间的网络连接、硬件需求在此描述。部署清单 312 支持至少包括 web 服务器、中间层服务器、以及数据库服务器的应用程序,并包括前端 / 后端应用程序、两个应用程序之间的网络连接,并且还描述单个节点之间的关系。部署时间创建那些在整个构架模型 304 中描述的实例。

[0057] 性能和安全模型 (305 和 306) 的每一个都支持描述那些相关功能和操作的相应清单 (未示出)。

[0058] 返回到基于机器学习的采用,可使用分类器基于例如第一部署期间的要求来选择并动态生成模型编码选定部分的清单。可使用更多或更少的属性来自动生成缺省模型。随着时间的流逝,当系统操作信息变得可用时,可分析该信息使得清单的细化度可基于最近的数据趋势和记录被调整为例如更近地监视特定区域中的系统。然后可按更近监视应用程序或服务所需来重新生成并采用经更新清单。

[0059] 如果清单描述了来自制造商的缺省安装或推荐最佳实施,管理员可能想要改变它们。例如,关于健康规则,管理员可能想要把阈值从 30 改成 40、或者想要安装组件、或者推翻一安全政策。这可通过创建清单的定制版本来替代由制造商绑定的清单。不同版本可在安装期间得到检测,使得用户能有选择缺省清单或定制清单的选项。或者,可有由系统读取列出所有替代的一单独文件,然后可显示它用于由用户选择以应用到缺省清单或在安装期间替代缺省设置。

[0060] 关于分布式应用程序,管理员可更一般地指明他或她想要都在该配置中连接的三个、四个还是六个。管理员可相应地为给定环境定制部署清单 312。

[0061] 现在参看图 3C,示出了根据本发明的基于模型管理构架用于管理应用程序或服务的系统组件 110 的核心系统 API 的框图。系统组件 110 包括根据本发明进行管理的应用程序或服务 314。系统 110 包括用于促进基于模型管理的合作通信中的众多 API。系统 110 包括由应用程序清单中信息配置的多个服务 (参照图 3B 所述)。

[0062] 系统 110 由确保应用程序可用性所需的服务组成,并使用在清单组件 108 中表述并由管理员更改的所需状态来执行以下:安装,用以校验依赖性并仅安装必需文件、设置和安全;事件订阅,用以订阅事件并按指定传送;轮询探测工具,用以周期性地收集探测和计数器;以及,综合性交易或模拟用户交易。确定应用程序是否可用并按期望(所需状态)执

行的最佳方法之一是监视系统象用户一样使用应用程序。这是主动监视。可能的第二种方法是真实用户交易的主动监视,并向系统提交用于分析的聚集数据。这些步骤关闭了循环并显示内部应用程序数据是不够的。基于模型管理也在应用程序外部工作。

[0063] 系统 110 使用在清单组件 108 中表述的所需状态来执行用于自动任务管理的任务调度;基于角色访问,用以限制对程序功能的访问;监视,用以检测问题、诊断根本原因、采取纠正性动作、并通知管理员何时需要干涉;以及中央配置,用以定制以上政策并应用到许多机器中。

[0064] 提供有与应用程序 314 通信的安装 API316,以便于应用程序、应用程序更新以及补丁的安装。安装 API316 通过代码取清单组合,并通过指示系统在该机器上安装该组件、该清单以及该版本来例示组合。安装 API316 具有与之相关联的协议 318 和查看器 320。协议 318 便于向系统 110 的其它组件传送 API 相关数据。查看器 320 显示与安装 API316 相关的数据。安装 API316 不仅便于单机安装而且用于包括本地和远程系统的分布式应用程序或服务,以及硬件供应和抽象。对于分布式数据中心环境,重要的是能够一般地抽象硬件系统,且对于更高的细化度则能进行特定机器抽象。在此如相对 API 所预期的,协议是管理 API 相关数据的传送和接收的规则。如本说明书所述,查看器 320 是显示与 API(这里是安装 API316) 相关数据的程序。API 数据包括但不限于例如声音文件、视频文件以及其它类型的数据文件。

[0065] 系统 110 包括与应用程序 314 通信的配置 API322,以便于配置应用程序 314。配置 API322 具有相关联的模式 323、协议 324 和查看器 326。模式 323 定义在 API322 和应用程序 314 之间传递的数据的结构和内容。协议 324 便于把 API 相关数据传送到系统 110 的其它组件。查看器 326 显示与配置 API322 相关的数据。

[0066] 还包括便于分布式环境多对一管理的管理 API328。该 API328 与受管理应用程序 314 以及远程系统(未示出)通信。API328 具有相关联的协议 330 和查看器 332。

[0067] 系统 110 包括与应用程序 314 通信的性能计数器 API334,以便于追踪在管理应用程序 314 中使用的计数器变量。该计数器 API334 具有相关联的协议 336 和查看器 338。协议 336 便于把 API 相关数据传送到系统 110 的其它组件。查看器 338 显示与计数器 API334 相关的数据。性能计数器由应用程序 314 展现并通过查看器 338 发表计数器。

[0068] 提供有与应用程序 314 通信的探测工具 API340,以便于配置探测工具并将探测数据传递给应用程序 314。探测工具 API340 与协议 342 和展现该探测的查看器 344 相关联。协议 342 便于把 API 相关数据传送到系统 110 的其它组件。查看器 344 显示与探测工具 API340 相关的数据。探测工具 API340 通过 IPC(进程间通信)346 与受管理应用程序 314 通信。IPC 是同一计算机上或者网络上一程序和另一程序之间数据的自动交换。当用户使用剪贴板手动地从一文件到另一文件剪切并黏贴数据时,执行了 IPC 功能的一个示例。计数器一直通过共享存储器发表,而探测则按需递送。探测工具 API340 还包括以类似于事件模式的方式描述探测类型表面的模式 348。还可包括探测记录(未示出);然而,许多管理员喜欢利用事件记录。

[0069] 系统 110 包括作为跟踪并缓存组件和模式信息的存储器的目录 347。该模式信息来自安装时的清单,以及部分是动态的并在运行时更新。目录 347 包括目录 API 并提供对事件、计数器、探测工具、以及配置数据的访问,这里仅命名存储的一些类型的数据。协议 351

和查看器 353 便于对目录 347 的访问。中央配置数据库包含了跨多个受管理节点上目录的累积或聚集综览。

[0070] 系统 110 包括与应用程序或服务 314 通信的事件 API350, 以便于实现和追踪在管理应用程序 314 中使用的事件。事件 API350 与作为所发生所有事件存储器的事件记录 352 通过接口相连。事件 API350 与协议 354 和查看器 356 相关联。协议 354 便于把 API 相关数据传送到系统 110 的其它组件。查看器 356 显示与事件 API350 相关的数据。与应用程序 314 的通信根据定义在其间传递的数据结构和内容的事件模式 358。事件在描述或发生时发表。该模式描述事件的表面。

[0071] 系统 110 包括与应用程序 314 通信的自动化 API360, 以便于自动化可正常地通过与应用程序 314 交互而完成的过程。自动化 API360 具有相关联的协议 362 和外壳 364。协议 362 便于把 API 相关数据传送到系统 110 的其它组件。外壳 364 向自动化 API360 提供用户界面, 以便于用于输入并显示与自动化过程相关数据的用户交互, 并便于自动化过程的用户控制。

[0072] 系统 110 还包括与应用程序 314 和自动化 API360 都通信的经调度任务 API366。经调度任务 API366 便于对至少自动化 API360 和经管理应用程序 314 调度作业或程序。它维持要运行的作业列表并相应地分配资源。经调度任务 API366 具有相关联的协议 368 和查看器 370。协议 368 便于把 API 相关数据传送到系统 110 的其它组件。查看器 370 显示与经调度任务 API366 相关的数据。任务模式 372 定义在任务 API 和其它组件之间传递的数据的结构和内容。

[0073] 从任务和小命令模型接收自动化和任务数据。这些特征可通过管理外壳本地或者远程地自动化。调度系统可运行这些例如在凌晨 3 点的备份。

[0074] 可以理解, 在图 3C 中所述的组件可代表那些本地实现, 而图 3D 所述的组件可代表那些与分布式实现相关联的, 从而分析跨许多机器和软件系统而发生。因而, 在一分布式实现中, 图 3D 的组件至少与图 3C 的本地系统之一通信, 但通常是与有线和 / 或无线体制中的多个这种本地系统通信。在本地实现中, 系统 110 还可包括图 3D 的任意或所有组件, 其中包括本地监视服务 API365。本地监视服务 API365 还包括协议 367、查看器 369 以及模式 371, 每一个都促进与其它 API 的组件相似的功能。在分布式系统中, 本地监视服务 365 可把监视信息传递给分布式监视服务, 如下所述。

[0075] 现在参看图 3D, 示出了本发明基于模型管理结构的系统组件 110 的管理相关 API 的框图。提供了配置数据库子组件 374, 其中通过中央配置 API376 向它提供访问和控制。中央配置 API376 与系统 110 的所有子组件通过接口相连, 且具有相关联的用于通信和交互的协议 378 和查看器 380、以及描述诸如断定和缺省值的配置设置和属性的形态的模式组件 382。协议 378 便于把 API 相关数据传送到系统 110 的其它组件。

[0076] 还提供有用作管理系统的操作相关数据 (例如报告、当前状态、以及历史数据) 存储库的操作数据库子组件 383。监视 API384 与操作数据库 383 以及基于模型管理系统的子组件通过接口相连, 并还具有相关联的协议 385、查看器 386 以及模式 387。协议 385 便于把 API 相关数据传送到系统 110 的其它组件。查看器 386 显示与监视 API384 相关的数据。模式 387 至少参照结构中每个数据元素可包含内容的结构和类型提供整个操作数据库 383 的定义。

[0077] 中央配置可触及所有 API,并由管理员使用以设置可包括分布式应用程序情景细节的配置细节,诸如应用程序应当安装在哪台机器上。配置还包括监视配置。例如,所有机器必需显示 5 分钟的不低于 80% 的 CPU 利用率。因而,监视系统使用该配置系统。监视是管理员如何通过管理系统确保应用程序在运行、被配置、以及在每个模型上安装。它还可包括确保期望功能、安全的所需量、正常执行、以及按用户期望递送数据。因而,监视涵盖所有这些领域。一般过程是安装、配置、按需运行任务、消耗事件、提供探测、配置、以及存储数据和结果。健康清单以对监视系统指令的规则形式向监视系统提供工作指令。一般而言,清单包含运行时指令,并且运行时实现所需状态。

[0078] 监视服务是本地服务,也是中央或分布式机制。对于分布式实现,健康包括本地机器的,以及本地和远程机器之间的关系。例如,给定 10 台机器的集合,只要 6 台正常发挥功能,就认为该系统是健康的。然而,如果不超过 5 台机器在运行,系统健康状态降级至警戒状态。如果不超过 4 台机器在运行,系统健康可视为故障状态。如果一个或多个集合中机器故障或离线,硬件抽象便于引入一个或多个备份系统,或者使应用程序 / 服务在线。因而,可基于指令控制空闲资源或共享资源池。该特征在数据中心环境中特别有用。可实现自动化动作以确保系统保持了优化或至少最少功能。

[0079] 基于模型管理构架的一方面使开发者能创作表述系统被视为健康所必需符合标准的大量规则。监视 API384 包括便于规则隐式并行处理的规则运行时引擎。规则引擎接收按使用规则定义语言 (RDL) 表述规则的中间形式来表述规则的输入指令。规则引擎还接收来自配置数据库 374 用以例示规则代码的配置数据。翻译器读取输入指令并将它们转换成并行的执行形式。运行时引擎读取经翻译的指令并便于并行执行。通过将指定要运行那些规则的配置数据、以及运行规则所需的参数载入运行时引擎来例示规则代码。可在运行时改变规则参数,诸如仅当检测到问题时才激活具有大型系统影响的规则。因而,规则以及可相应改变的阈值都是动态的。监视 API384 还与系统 110 的所有子组件相连。

[0080] 还提供由管理员使用的清单存储和编辑服务 388。清单服务 388 具有相关联的协议 389 和查看器 390 以向管理员展现这些清单功能。清单服务 388 通过协议 389 和查看器 390 向管理员供应清单,使得管理员在安装之前可查看并改变清单。清单服务 388 还便于根据更新和定制更改清单的版本。

[0081] 还提供有与基于模型管理系统的所有子组件通过接口相连的基于角色访问 API391,且它具有相关联的协议 392 和查看器 393。协议 392 便于把 API 相关数据传送到系统 110 的其它组件。查看器 393 显示与基于角色 API391 相关的数据。该 API391 在监视和配置组件之上的层次进行说明,以基于模型管理系统向各种组件和诸方面提供对访问的全面管理。基于角色访问 API391 不必包括协议 392 和查看器 393,因为这些功能可由系统 110 的其它组件促成。

[0082] 系统还包括用于基于机器学习和控制的分类器 394。如上所述,可用多种方法采用分类器 394 以提高系统性能和健康,仅列举若干。为便于基于机器学习,分类器 394 与中央配置服务 376 通过接口相连,从而可访问该系统的所有组件并使用其数据。

[0083] 现在参看图 3E,示出了本发明基于模型管理构架任务组件 112 的主要子组件。任务通过管理任务模型来描述。任务落于三个子组件中:监视子组件 395、检修子组件 396、以及管理子组件 397。

[0084] 监视子组件 395 的任务包括检查健康、安全、补丁、配置、性能、以及应用程序数据。检修子组件 396 的任务包括诊断健康状态、处理警告、以及更新事件、探测工具、和性能记录。管理子组件 397 的任务包括中央配置 / 政策、调度、以及更新部署。管理不仅包括单个系统的管理,还包括管理例如许多机器、应用程序、和系统、政策、备份时间、改变、以及更新。

[0085] 在基于模型管理构架中采用 URI 以唯一地标识抽象或物理资源或资源集合。资源的模式可由带有资源占位符的 URI 来标识。带有占位符的 URI 称为 URI 模板。系统的目录依赖 URI 模板来不参照特定实例而描述探测工具。URI 模板使得无需真正检索特定实例的探针就能标识探针并理解其特征。保护离开实例而预定义探测工具的能力使得规则的部署和创作更简便,且使相关联操作系统可进行管理。

[0086] 基于模型管理构架采用 RDL 以基于监视软件和硬件可用性的目的来激活规则的定义。以 RDL 编写的规则由运行时引擎执行,作为监视服务的一部分。RDL 的目的是测试断定、使用运行时信息实施约束、作推断、执行关联、以及向其它组件传送动态测试的结果。RDL 定义规则类型(即类),而通过指定例示所需的参数值独立的 XML(可扩展标记语言)文档被用以创建规则类型的实例。有用于描述系统应当对问题检测、诊断、分解、校验以及警告所采取的步骤序列的模式。这就是模型中所描述的、清单中所表述的、并由监视系统所执行 / 管理的内容。

[0087] 基于模型管理构架采用事件和性能计数器的更新值以指示服务、测试或综合交易的健康模式(或状态),如前所述。健康模型 301 是怎样服务或组件可能会故障的图形和 / 或文本表示,它帮助管理员理解各种事件和服务的性能计数器的重要性,并基于观察到的探测数据来有效判定是否要作任何动作。开发者用随后从模型和源代码属性中生成的相应文件来构建健康模型 301。

[0088] 健康模型 301 包括对组件关系以及依赖性的描述。取决于被检测问题的环境,系统可遍历关系树并尝试基于其它组件的健康确定根本原因。该方法由根据本发明利用的模型和清单所支持。

[0089] 本揭示构架获得了服务定义模型系统的应用,其各方面是本受让人诸专利申请的主题,第一是 2003 年 10 月 \_\_\_\_\_ 日提交美国专利申请系列号为 \_\_\_\_\_ 的题为“Architecture for Distributed Computing System and Automated Design, Deployment, and Management of Distributed Application”(“分布式计算系统的构架和分布式应用程序的自动化设计、部署和管理”)的申请,以及第二为 2003 年 10 月 \_\_\_\_\_ 日提交美国专利申请系列号为 \_\_\_\_\_ 的题为“Integrating Design, Deployment, and Management Phases for an Application”(“集成应用程序的设计、部署与管理阶段”)的申请。

[0090] 现在参看图 4,示出了基于模型管理的过程的流程图。尽管为了简单解释以例如流程图形式在此示出的一个或多个方法被示为并描述为一系列动作,但可以理解和认识到本发明不受限于这些动作的顺序,因为根据本发明某些动作可以与在此所示和所述的不同顺序发生和 / 或其它动作并发。例如,本领域技术人员将理解并认为方法可有选择地表示为一系列相互关联的状态或事件,诸如状态图。此外,要根据本发明实现方法并不需要所有的示出动作。

[0091] 在 400, 要安装的应用程序或服务根据其组件进行描述。在 402, 应用程序或服务根据功能、配置、安全以及性能在所需状态中进行描述。在 404, 在安装期间提供描述以及应用程序或服务, 从而由系统使用描述来配置系统的管理服务。然后过程抵达停止框。

[0092] 现在参看图 5, 所示是实现基于模型管理的过程的更详细流程图。在 500, 开发了应用程序组件、健康状态以及恢复、配置设置以及管理任务的模型。在 502, 用户根据环境定制了系统 / 规则 / 模型。在 504, 将属性插入源代码以指示用于监视的探测工具和逻辑。在 506, 提供了模型信息和源代码属性的清单以由管理系统服务使用。以机器可读形式提供了由管理系统服务使用的清单。在 508, 管理系统服务的一个或多个基于清单信息进行配置。在 510, 在清单中定义应用程序的管理性任务, 诸如向系统登记小命令、设置时间表等等。然后过程抵达停止框。

[0093] 现在参照图 6, 示出了实现基于模型管理的所需状态的过程的流程图。在 600, 从清单访问所需状态。在 602, 校验依赖性并仅安装必需的文件、设置和安全特征。在 604, 如清单中指定地订阅并转送事件。在 606, 周期性地收集探测数据和计数器数据, 以及所执行的测试和综合交易。在 608, 执行自动管理任务。在 610, 限制对程序功能的访问。然而, 根据本发明促进基于模型的管理并不需要包括它。在 612, 检测问题、诊断根本问题、采取纠正性动作、并通知系统管理员何时干涉。在 614, 所有以上政策被定制用于对许多其它类型机器和系统的应用程序。然后过程抵达停止框。

[0094] 现在参照图 7, 示出了可操作来执行本发明构架的计算机的框图。为了提供本发明诸方面的其它上下文环境, 图 7 和以下论述旨在对适合本发明诸方面在其中实现的适当计算环境 700 提供简要、一般的说明。尽管本发明是在运行于一台或多台计算机上的计算机可执行指令的一般上下文环境中说明的, 本领域技术人员将认识到本发明也可结合其它程序模块和 / 或作为硬件和软件的组合来实现。通常, 程序模块包括执行具体任务或实现具体抽象数据结构的例程、程序、组件、数据结构、等等。另外, 本领域技术人员将理解本发明的方法也可通过其它计算机系统配置来实践, 包括单处理器或多处理器计算机系统、微型计算机、大型计算机、以及个人计算机、手持式计算装置、基于微处理器的或可编程的消费电器等等, 其中每个装置都可有效地与一个或多个相关联装置耦合。本发明所说明的诸方面也可在任务由经通信网络连接的远程处理设备执行的分布式计算环境中实践。在分布式计算环境中, 程序模块可置于本地和远程存储设备。

[0095] 再参照图 7, 实现本发明各方面的示例性环境 700 具有计算机 702, 该计算机 702 具有处理单元 704、系统存储器 706、及系统总线 708。系统总线 708 耦合包括, 但不限于将系统存储器 706 耦合到处理单元 704 的系统组件。处理单元 704 可以是各种可用处理器的任一种。双微处理器和其它多处理器构架也可被用作处理单元 704。

[0096] 系统总线 708 可以是若干类总线结构的任一种, 包括存储器总线或存储器控制器、外围总线、和 / 或使用各种可用总线构架任一种的本地总线。系统存储器 706 包括只读存储器 (ROM) 710 和随机存储器 (RAM) 712。包含在计算机 702 元件间传送如起动时信息的基本例程的基本输入 / 输出系统 (BIOS), 存储在诸如 ROM、EPROM、EEPROM 的非易失性存储器 710 上。RAM 712 也可包括诸如用于缓存数据的静态 RAM 的高速 RAM。

[0097] 计算机 702 还包括硬盘驱动器 714、磁盘驱动器 716 (例如读取或写入可移动磁盘 718) 和光盘驱动器 720 (例如读取 CD-ROM 722, 或读取或写入其它诸如数字化视频盘 (DVD)



的大容量光学介质)。硬盘驱动器 714、磁盘驱动器 716、和光盘驱动器 720 分别通过硬盘驱动器接口 724、磁盘驱动器接口 726、和光盘驱动器接口 728 与系统总线 708 相连。这些驱动器和与之相关联的计算机可读介质提供数据、数据结构、计算机可执行指令等等的非易失性存储。对于计算机 702，驱动器和介质容纳了具适当数字化格式的广播编程的存储。尽管以上计算机可读介质的描述是指硬盘、可移动磁盘和 CD，本领域技术人员将理解，其它类型的计算机可读介质，诸如 zip 盘、磁带、闪存卡、数字化视频盘、盒式磁带等等，也能用于示例性操作环境，而且，任意这种介质可包含执行本发明方法的计算机可执行指令。

[0098] 众多程序模块，包括操作系统 730、一个或多个应用程序 732、其它程序模块 734、和程序数据 736，可存储在驱动器和 RAM712 中。操作系统、应用、模块、和 / 或数据的全部或部分也可被高速缓存在 RAM712 中。

[0099] 可以理解本发明可用各种可购买的操作系统或操作系统的组合来实现。

[0100] 用户可通过键盘 738 和诸如鼠标 740 的定位装置向计算机 702 输入指令和信息。其它输入装置（未示出）可包括话筒、IR 远程控制、游戏杆、游戏垫、卫星天线、扫描仪等等。这些和其它输入装置通常通过与系统总线 708 耦合的输入端口接口 742 连接到处理单元 704，但也可通过其它接口相连，如并行端口、游戏端口、通信串行总线（USB）端口、IR 接口等等。监视器 744 或其它类型显示装置也通过接口，如视频适配器 746 和系统总线 708 相连。除了显示器 744，计算机通常包括其它外围输出装置（未示出），如扬声器和打印机等。

[0101] 计算机 702 可以在使用与一台或多台远程计算机，诸如远程计算机 748 经有线和 / 或无线通信的逻辑连接的网络化环境中运行。远程计算机 748 可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐装置、对等装置或其它共同网络节点，而且通常包括上述与计算机 702 相关的许多或全部部件，尽管为简化起见仅示出了存储器存储装置 750。所述逻辑连接包括局域网（LAN）752 和广域网（WAN）754。这样的网络化环境在办公室、公司范围计算机网络、内联网以及因特网是常见的。

[0102] 当用于 LAN 网络环境中时，计算机 702 通过有线或无线的通信网络接口或适配器 756 与局域网 752 连接。适配器 756 可有助于与 LAN752 的有线或无线通信，其中包括用于与无线适配器 756 通信的无线访问节点。当用于 WAN 网络环境中时，计算机 702 包括调制解调器 758、或连接于 LAN 上的通信服务器、或其它用于在广域网 754 如因特网中建立通讯的装置。可以是内置式或外置式、有线或无线装置的调制解调器 758 与系统总线 708 通过串行端口接口 742 连接。在网络化环境中，与计算机 702 相关的程序模块或其一部分可存储在远程存储器 / 存储装置 750 中。可以理解的是，所示网络连接是示例性的，且其它用于在计算机间建立通讯连接的技术也可以使用。

[0103] 计算机 702 与任意有效地置于无线通信中的无线装置或实体的通信是可操作的，例如打印机、扫描仪、台式和 / 或便携式计算机、便携式数据助理、任何关联于无线可检测标记的设备或地点（例如亭子、新闻架、厕所）、以及电话。这包括至少 Wi-Fi 和蓝牙™ 无线技术。因而，通信可以是带有常规网络或至少两个装置之间特别通信的预定结构。

[0104] Wi-Fi 或无线保真，使得家中沙发、旅馆房间内的床、或工作中的会议室无需接线就可与因特网连接。Wi-Fi 是能使例如计算机的这种装置在室内外收发数据的像蜂窝电话这类的无线技术；可在基站范围内的任何地方。Wi-Fi 网络使用称为 IEEE802.11(a, b, g 等) 的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络能用于计算机之间的相

互连接、与因特网、有线网络（使用 IEEE802.3 或以太网）的连接。Wi-Fi 网络具有 11 兆比特 / 秒 (Mbps) (802.11b) 或 54Mbps (802.11a) 的数据速率, 或包含两个频带（双频带）在无许可证的 2.4 和 5GHz 的无线频带上操作, 因此网络可提供类似于在许多办公室中使用的基本 10BaseT 有线以太网的实际性能。

[0105] 现在参看图 8, 示出了根据本发明示例性计算环境 800 的示意框图。系统 800 包括一台或多台客户机 802。客户机 802 可以是硬件和 / 或软件（例如线程、过程、计算装置）。例如客户机 802 可通过采用本发明容纳 cookie 和 / 或相关联的上下文信息。系统 800 还可包括一台或多台服务器 804。服务器 804 也可以是硬件和 / 或软件（例如线程、过程、计算装置）。例如, 服务器 804 可采用本发明容纳线程来执行变换。在客户机 802 和服务器 804 间的一可能通信可能是以适于在两个或多个计算机过程间传送的数据包形式进行。数据包可包括例如 cookie 和 / 或相关联的上下文信息。系统 800 包括可用来便于客户机 802 和服务器 804 间通信的通信框架 806（例如诸如因特网的全球通信网络）。

[0106] 通信可通过有线（包括光纤）和 / 或无线技术来推动。客户机 802 可与一个或多个用来存储客户机 802 本地信息（例如 cookie 和 / 或相关联上下文信息）的客户机数据库 808 有效连接。类似地, 服务器 804 可与一个或多个用来存储服务器 804 本地信息的服务器数据库 810 有效连接。

[0107] 如上所述, 所揭示的基于模型管理构架可应用于企业类型的系统管理。例如, 客户机 802 之一不仅可管理本地应用程序或服务, 也可管理那些远程节点例如服务器 804。所有方面应用以支持从本地客户机的单个实例到跨多个网络节点的远程系统和应用程序上的多个实例的健康监视。可从本地层次到公司层次及更上层采用基于机器的学习, 以自动化并改进系统性能和能力。

[0108] 以上所述包括本发明的诸多示例。当然, 为描述本发明而对每一能想到的组件或方法组合进行描述是不可能的, 但本领域普通技术人员明白本发明的更多排列和组合是可能的。因此, 本发明旨在包含所有这样的在所附权利要求书精神和范围内的变更、修改、和变化。此外, 就用于具体实施方式或权利要求书的术语“具有”而言, 这种术语意在以类似于术语“包括”在权利要求书中作连接词的方式作包含意义解。

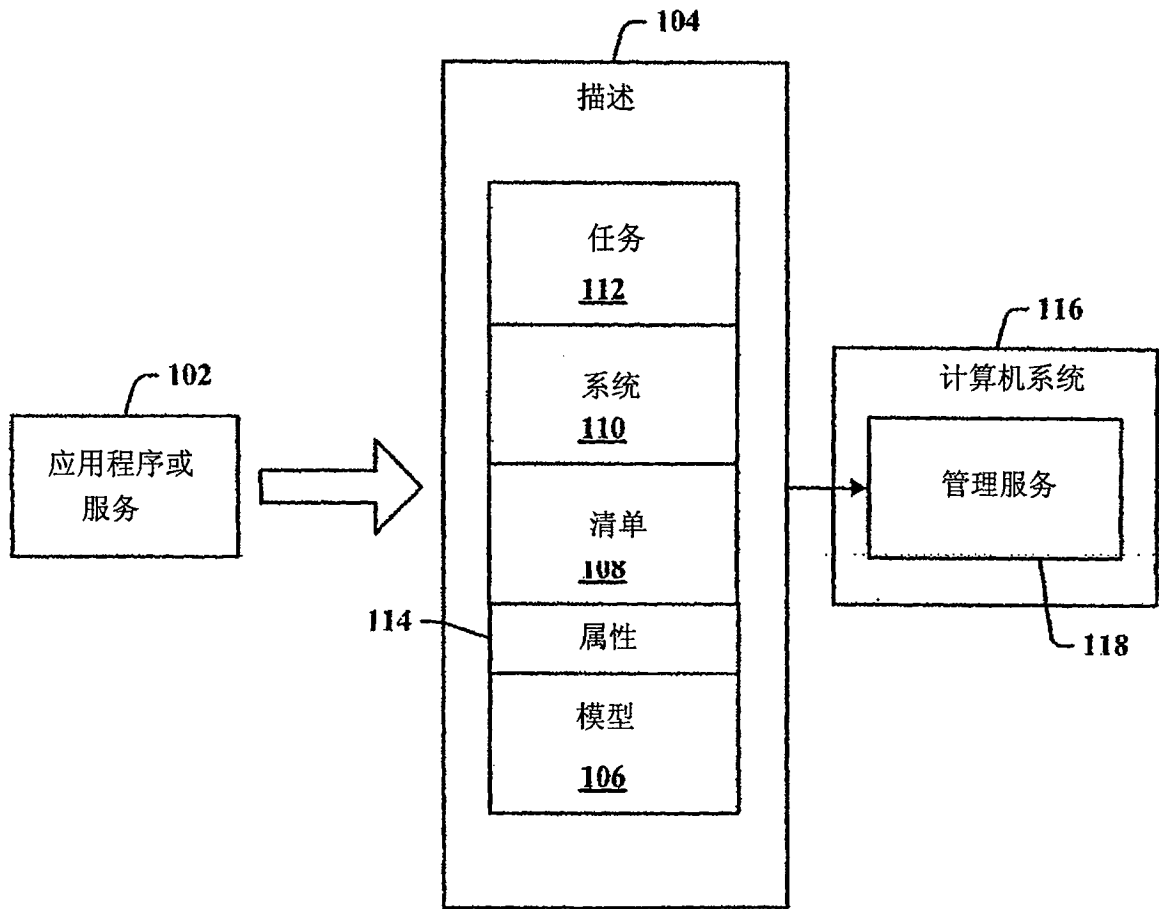


图 1

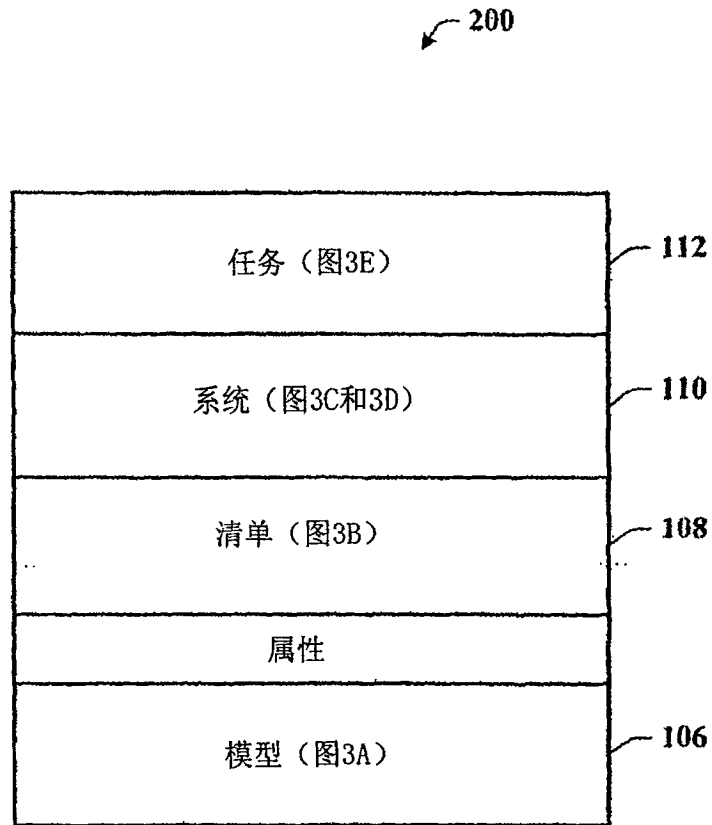


图 2

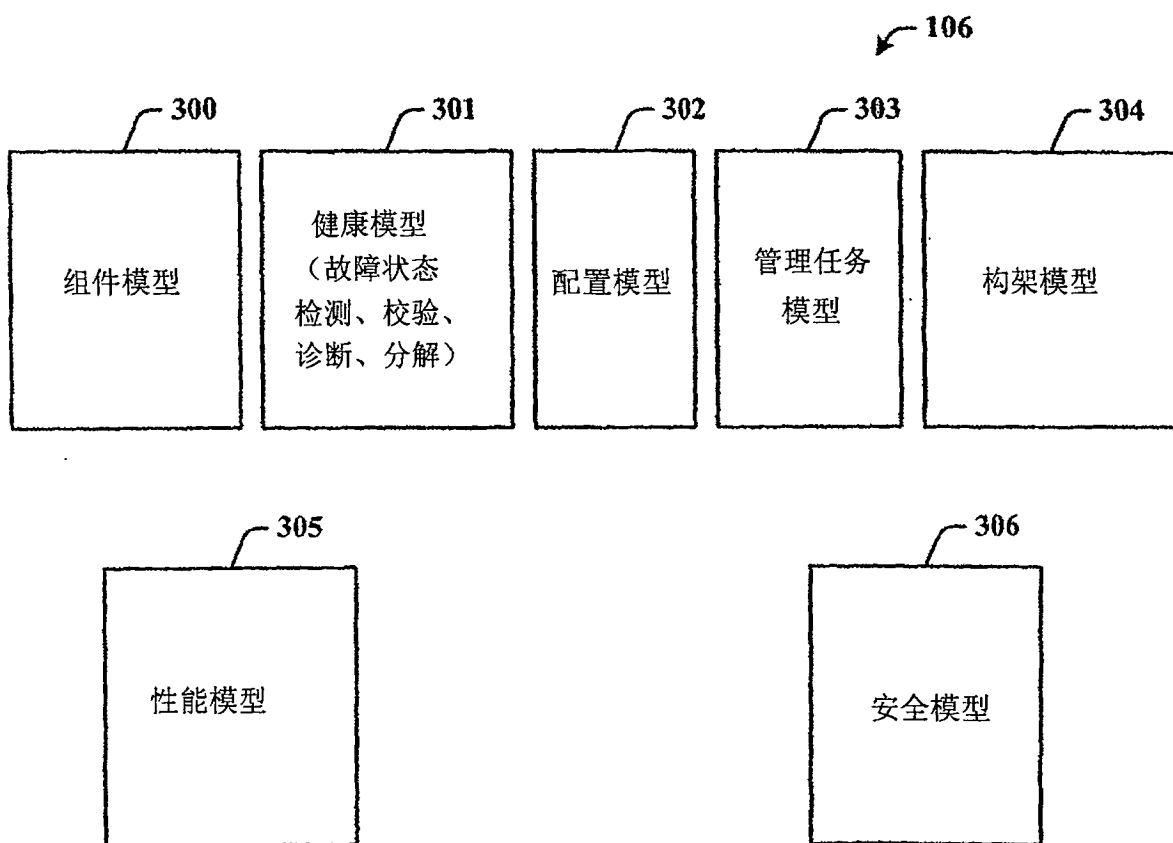


图 3A

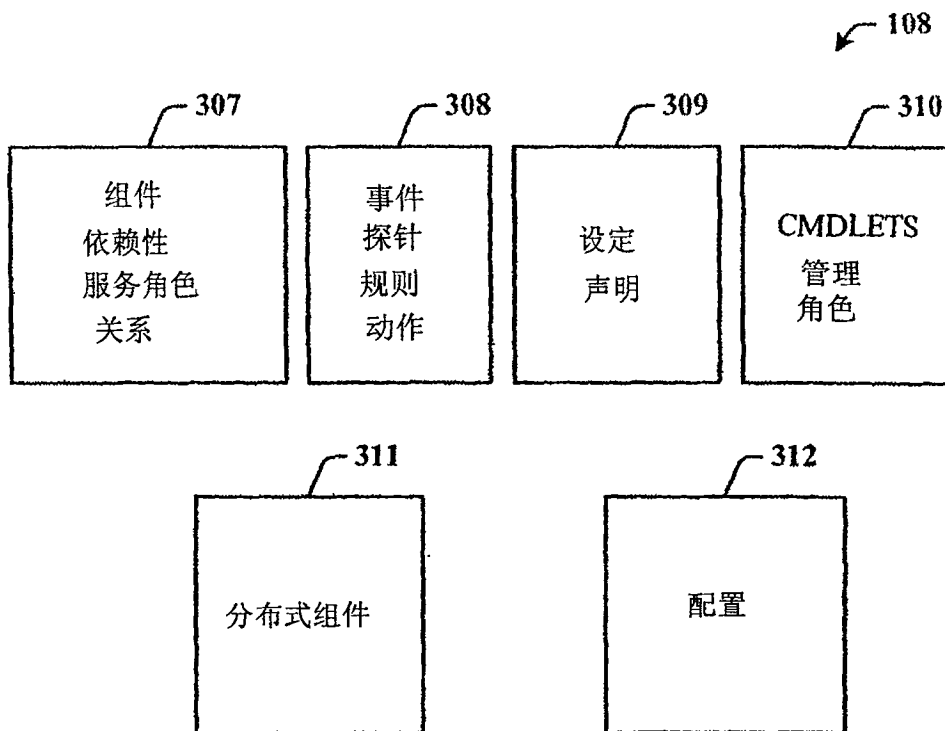


图 3B

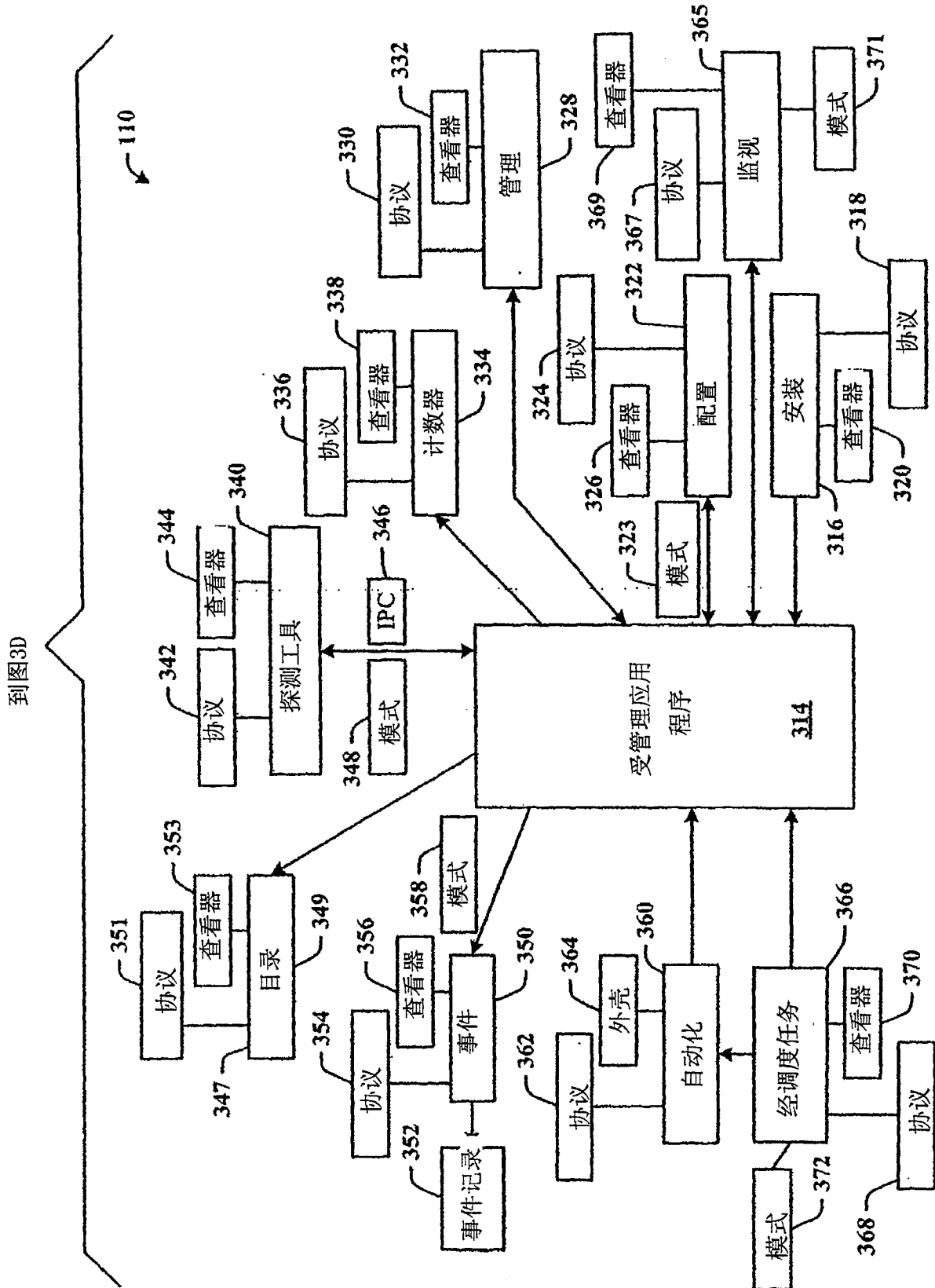
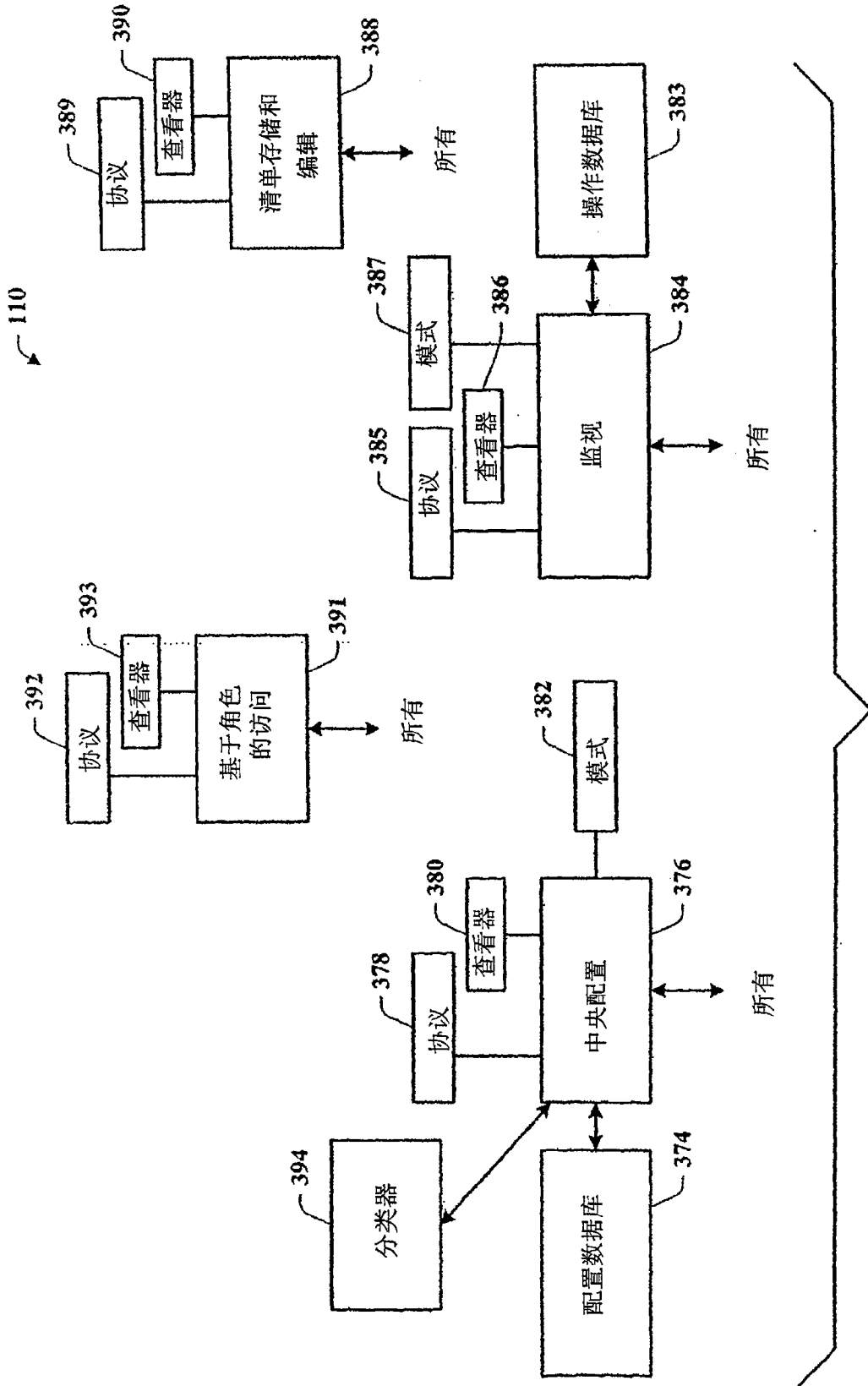


图 3C



到图3C

图 3D

112

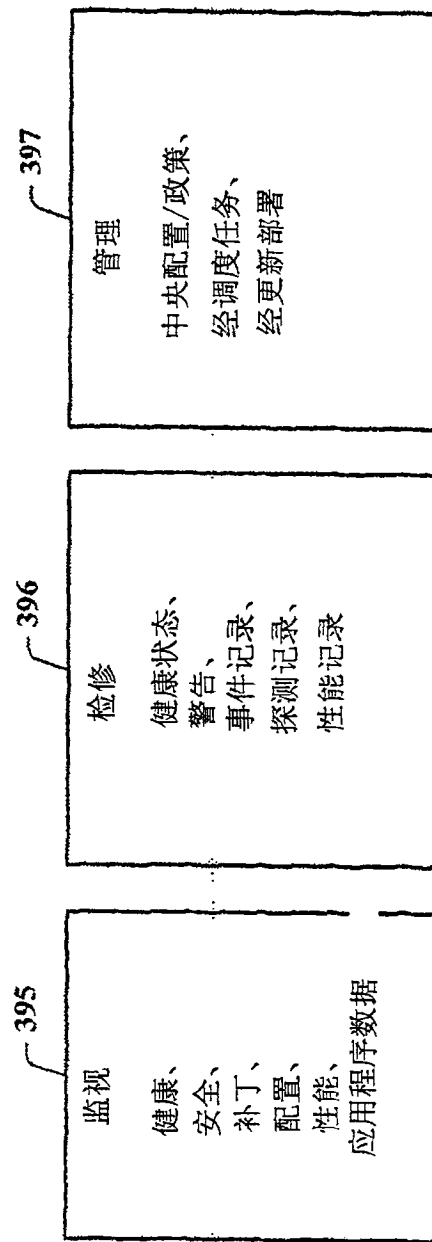


图 3E



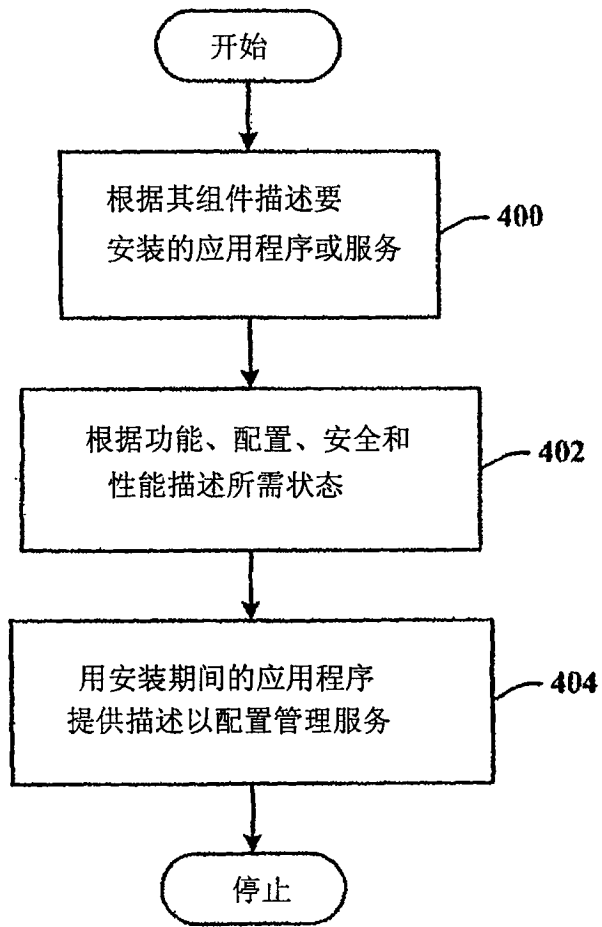


图 4

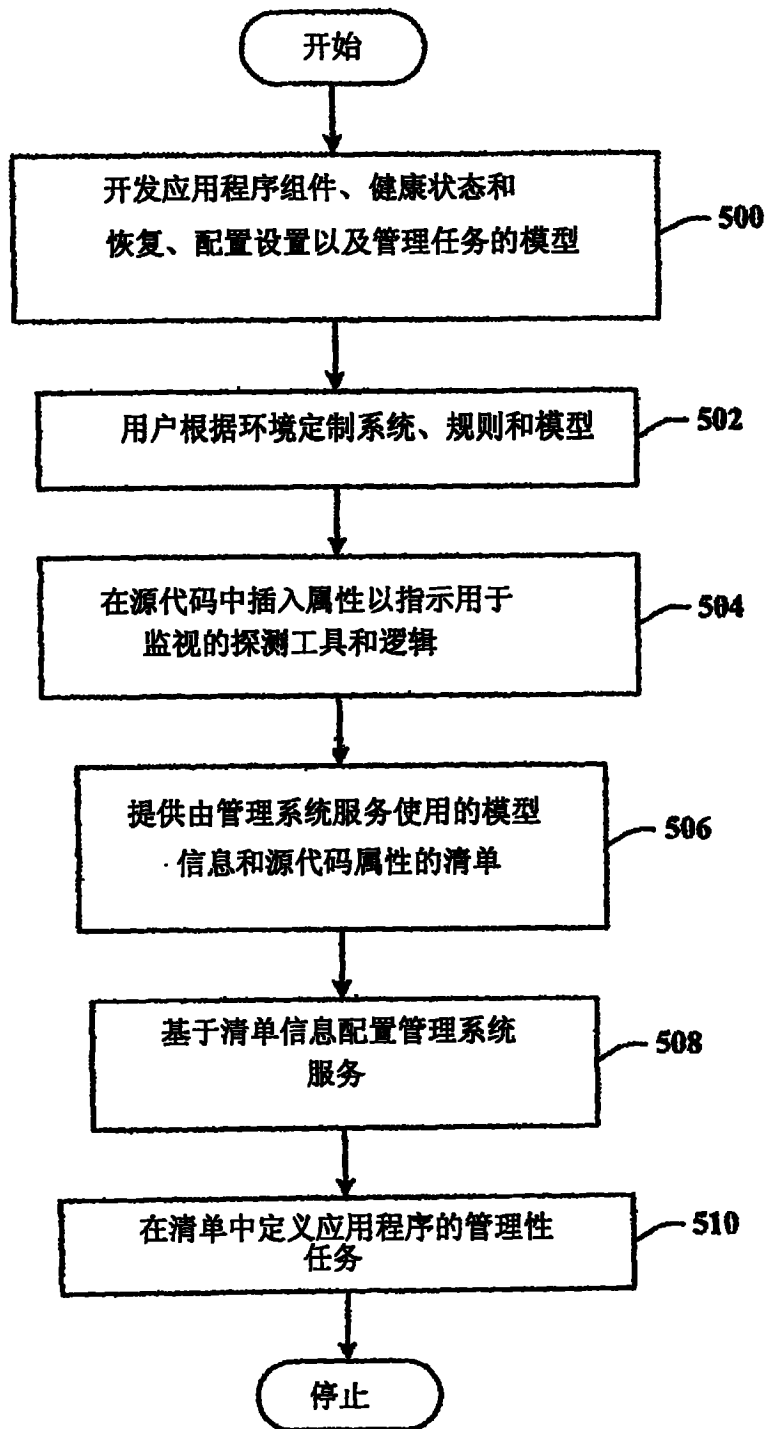


图 5

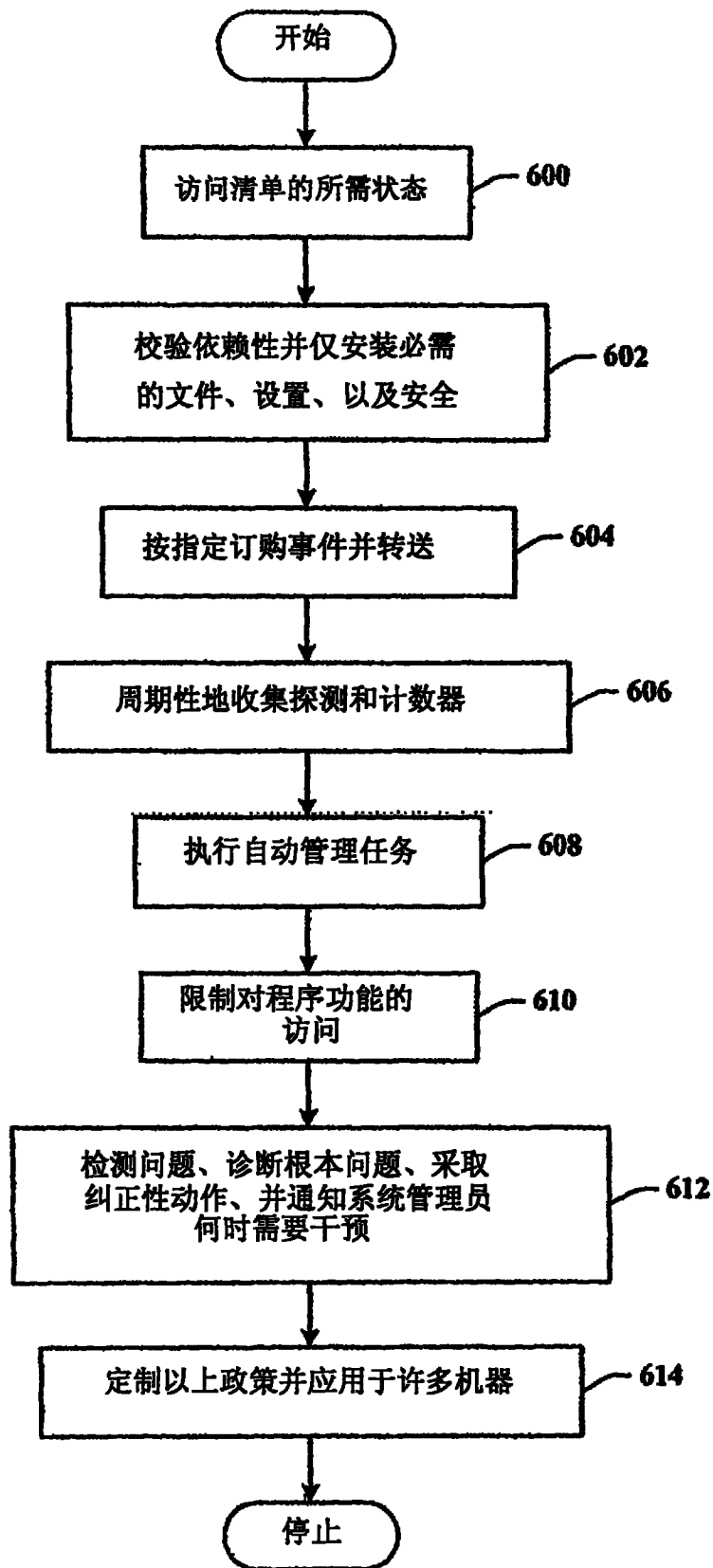


图6

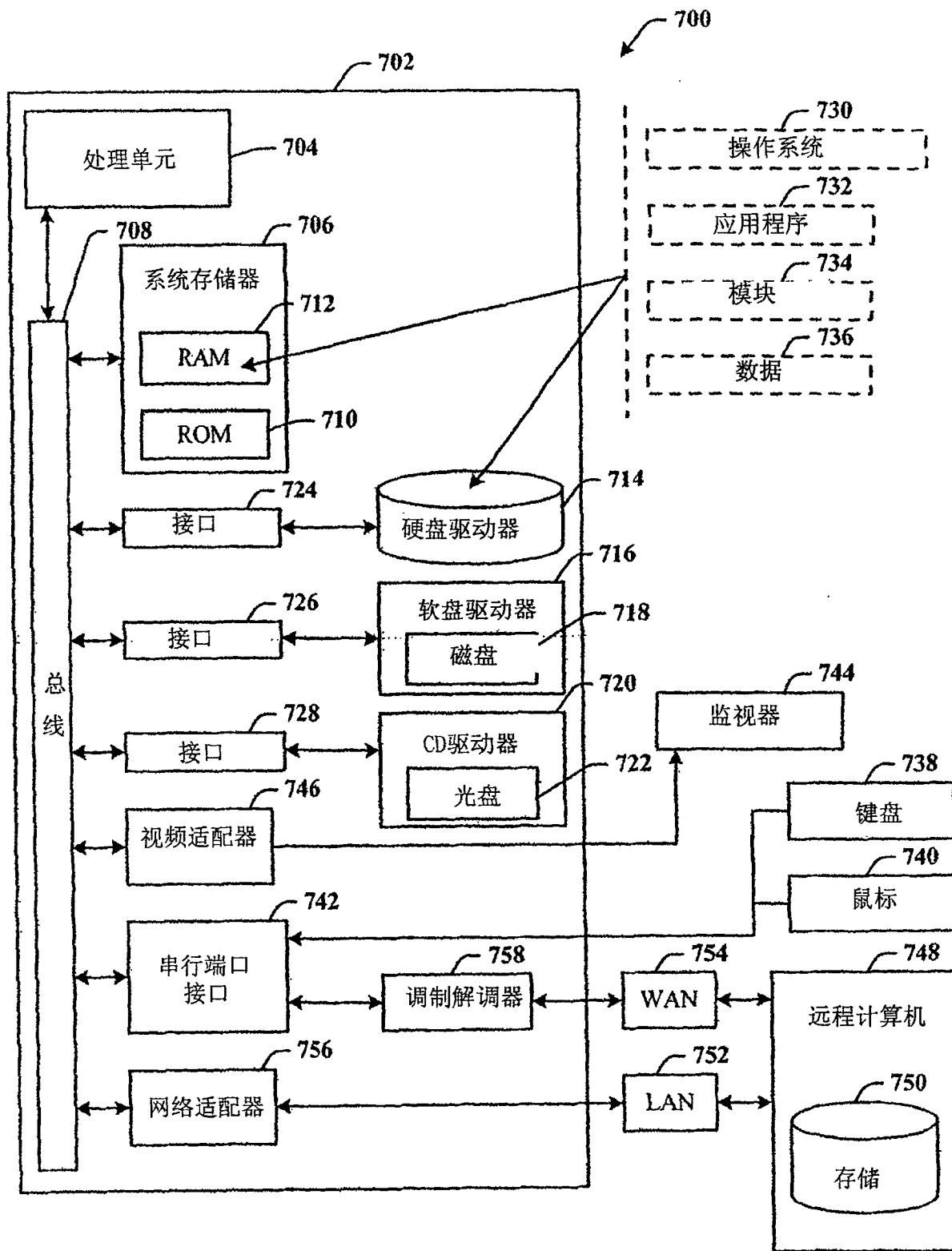


图 7

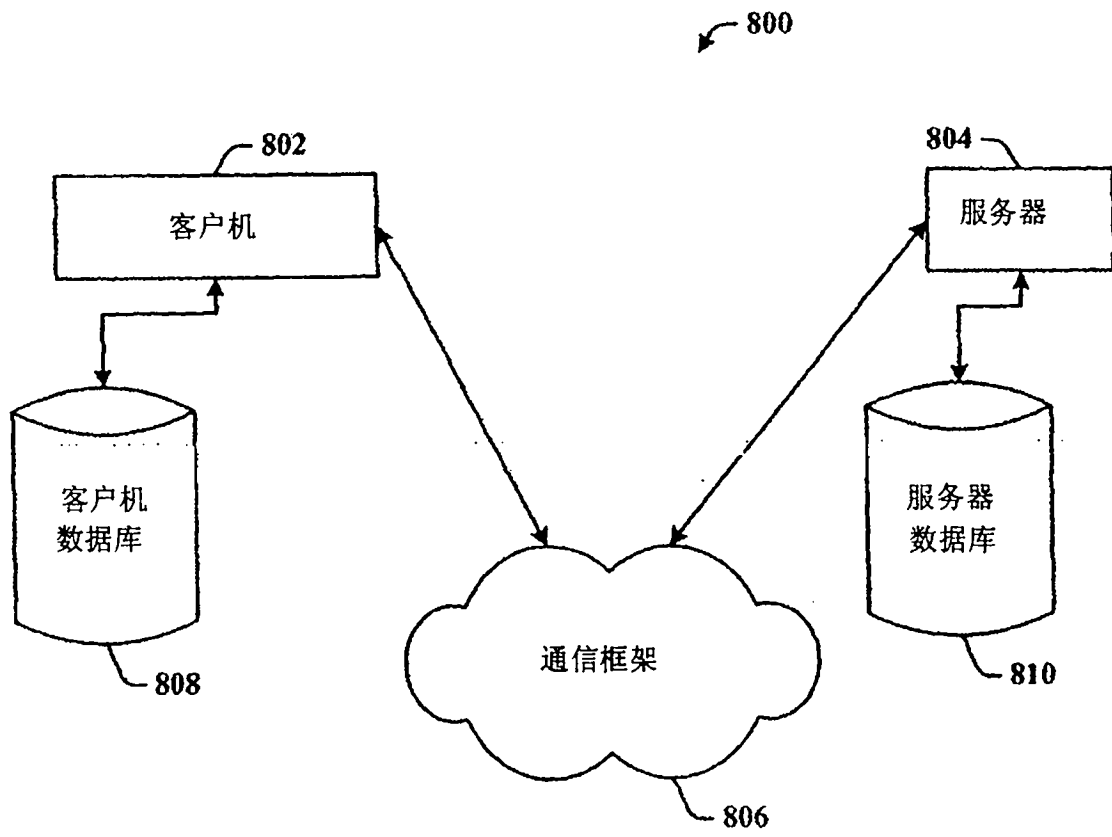


图 8