(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 103109292 B
(45)授权公告日 2018.02.27

(21)申请号 201180044786.1

(22)申请日 2011.09.13

(65)同一申请的已公布的文献号
申请公布号 CN 103109292 A

(43)申请公布日 2013.05.15

(30)优先权数据
12/889,574 2010.09.24 US

(85)PCT国际申请进入国家阶段日
2013.03.18

(86)PCT国际申请的申请数据
PCT/US2011/051319 2011.09.13

(87)PCT国际申请的公布数据
WO2012/039992 EN 2012.03.29

(73)专利权人 日立数据系统有限公司
地址 美国加利福尼亚州

(72)发明人 杰弗里·M·克伦普
玛吉·E·蒂拉多

(74)专利代理机构 北京银龙知识产权代理有限公司 11243
代理人 曾贤伟 杨继平

(51)Int.Cl.
G06F 17/30(2006.01)
G06F 15/16(2006.01)

审查员 王静

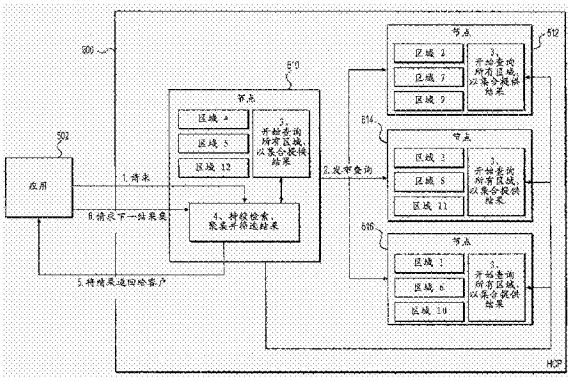
权利要求书2页 说明书13页 附图6页

(54)发明名称

在容错数据库管理系统中聚集查询结果的系统和方法

(57)摘要

独立节点的冗余阵列联网在一起。每个节点运行提供基于对象存储的应用的实例。元数据对象被存储在分布在阵列上节点中的区域的集合中。通过哈希元数据对象属性并提取得出的哈希值的比特的给定集合来识别给定区域。一种管理查询结果的方法包括：由第一节点从客户应用接收通过条件对于对象列表的请求；由第一节点基于所接收到的请求将查询发布给所有节点；由每个节点在节点中的区域上使用存储在区域中的元数据对象来处理查询；由第一节点聚集并筛选来自所有节点的查询结果；以及由第一节点将聚集的和筛选的结果返回给客户应用。



CN 103109292 B

1.一种用于管理查询结果的方法，用在具有多个节点的存储系统中，其中元数据对象被存储在分布在存储系统上节点当中的区域的集合中，其中通过哈希元数据对象属性来识别区域，其中，

该存储系统存储多个内容数据、以及与该多个内容数据相关联的多个原始元数据和多个备份元数据，该存储系统包括：包含第一处理器的第一节点；以及多个第二节点，每一个该第二节点具有第一存储区域和第二存储区域，其中，该第一存储区域存储所述多个原始元数据的一个或多个原始元数据，该第二存储区域存储所述多个备份元数据的一个或多个备份元数据，并且，每一个该第二节点包含第二处理器，该第二处理器被设置成管理所述第一存储区域和所述第二存储区域，

所述方法包括：

当所述第一节点接收到请求满足给定规则的内容数据和/或元数据的列表的第一查询时，所述第一处理器被设置成向所述多个第二节点的每一个发布第二查询，该第二查询请求满足所述给定规则的内容数据和/或元数据的列表；

当所述多个第二节点的每一个接收到所述第二查询时，所述多个第二节点的每一个的所述第二处理器被设置成针对与所述给定规则相关的内容数据和/或元数据，从其自己的第一和第二存储区域中仅搜索出其自己的第一存储区域，并将所述第二查询的结果提供给所述第一节点，以及

所述第一节点的所述第一处理器还被设置成对所述多个第二节点的每一个第二节点发送的结果进行聚集。

2.根据权利要求1所述的方法，

其中，由第二节点进行的处理查询包括向所述第一节点以集合提供所述查询结果；并且

其中，来自所述第二节点的查询结果以集合被聚集、被筛选并被返回给客户应用。

3.根据权利要求2所述的方法，还包括：

在由所述第一节点将聚集和筛选的结果的当前集合返回给所述客户应用之后，在请求和检索来自所述第二节点的下一结果集合之前，等候来自所述客户应用的对于所述下一结果集合的请求。

4.根据权利要求2所述的方法，

其中，将所述查询结果以集合提供给所述第一节点包括提供来自每个第一存储区域的预定数量的对象作为处理查询的结果。

5.根据权利要求1所述的方法，还包括：

由所述第一节点分类所聚集和筛选的结果以产生预定的顺序。

6.根据权利要求1所述的方法，

其中，所述请求还包括通过改变时间查询、通过目录查询、通过处理查询、以及在结果中标记页面中的一个或多个。

7.一种用于管理查询结果的装置，用在具有多个节点的存储系统中，其中元数据对象被存储在分布在存储系统上节点当中的区域的集合中，其中通过哈希元数据对象属性来识别区域，所述装置包括为每个节点提供的处理器、存储器以及查询结果管理模块，其中，

该存储系统存储多个内容数据、以及与该多个内容数据相关联的多个原始元数据和多

个备份元数据，该存储系统包括：包含第一处理器的第一节点；以及多个第二节点，每一个该第二节点具有第一存储区域和第二存储区域，其中，该第一存储区域存储所述多个原始元数据的一个或多个原始元数据，该第二存储区域存储所述多个备份元数据的一个或多个备份元数据，并且，每一个第二节点包含第二处理器，该第二处理器被设置成管理所述第一存储区域和所述第二存储区域，

所述查询结果管理模块被配置为：

当所述第一节点接收到请求满足给定规则的内容数据和/或元数据的列表的第一查询时，所述第一处理器被设置成向所述多个第二节点的每一个发布第二查询，该第二查询请求满足所述给定规则的内容数据和/或元数据的列表；

当所述多个第二节点的每一个接收到所述第二查询时，所述多个第二节点的每一个的所述第二处理器被设置成针对与所述给定规则相关的内容数据和/或元数据，从其自己的第一和第二存储区域中仅搜索出其自己的第一存储区域，并将所述第二查询的结果提供给所述第一节点，

所述第一节点的所述第一处理器还被设置成对所述多个第二节点的每一个第二节点发送的结果进行聚集。

8.根据权利要求7所述的装置，

其中，由第二节点进行的处理查询包括向所述第一节点以集合提供所述查询结果；并且

其中，来自所述第二节点的查询结果以集合被聚集和筛选以被返回给客户应用。

9.根据权利要求8所述的装置，

其中，如果具有所述查询结果管理模块的节点是所述第一节点，则将所述查询结果管理模块配置成：在由所述第一节点将聚集和筛选的结果的当前集合返回给所述客户应用之后，在请求和检索来自所述第二节点的下一结果集合之前，等候来自所述客户应用的对于所述下一结果集合的请求。

10.根据权利要求8所述的装置，

其中，将所述查询结果以集合提供给所述第一节点包括提供来自每个第一存储区域的预定数量的对象作为处理查询的结果。

11.根据权利要求7所述的装置，

其中，如果具有所述查询结果管理模块的节点是所述第一节点，则所述查询结果管理模块被配置成：分类所聚集和筛选的结果以产生预定的顺序。

12.一种节点，其处在根据权利要求7所述的具有多个节点的存储系统中，所述节点包括用于管理所述节点中的元数据对象的元数据管理器，所述元数据管理器包括组织并提供对所述元数据对象的访问，其中所述元数据管理器包括所述节点的查询结果管理模块。

在容错数据库管理系统中聚集查询结果的系统和方法

技术领域

[0001] 本发明通常涉及存储系统，尤其涉及用以在容错数据库管理系统中聚集查询结果的系统和方法。

背景技术

[0002] 基于客户的因公需要和情景，他们对内容平台可能具有不同的聚集需求。一个常见的主题围绕着检索对象列表和这些对象的信息，以传递到在他们设施中的另一个应用实现特定功能(例如，搜索数据或备份数据)。为了这样做，可能需要应用做更多的工作以检索该信息。例如，集成应用可能必须遍历目录、子目录、次-子目录等等来检索对象的列表和给定规则的系统元数据。对于独立节点的冗余阵列被联网在一起并且每个节点群组(cluster)/系统被分区为租户(tenant)和命名空间的情况，这可能必须对于命名空间中的所有目录、相关的所有命名空间、相关的所有租户等进行操作。命名空间是群组的逻辑分区，并且基本上作为特定于至少一个限定应用的对象集。每个命名空间具有相对于其他命名空间私有的文件系统。此外，访问一个命名空间不允许用户访问其他命名空间。租户是一组命名空间以及可能是其他子租户。群组/系统是物理存档实例。见共同授权的美国专利申请No.12/609,804，其在2009年10月30日递交，名称为使用命名空间的分区内容平台中的固定内容存储(Fixed Content Storage Within a Partitioned Content Platform Using Namespaces)，以引用的方式将其合并入本文中。

发明内容

[0003] 本发明的示例性实施例允许REST(表象化状态转换)客户对符合给定条件的元数据和对象的列表查询内容平台而不需要集成应用来遍历目录、子目录、次子目录等等，以检索给定条件的对象列表和系统元数据。例如，客户应用可以通过改变时间查询、通过目录查询、通过处理(生成、删除、清除)查询、通过命名空间查询、或在结果中标记页面。使用本发明，单个内容平台节点将查询分布到内容平台系统中的所有节点上的所有区域，并且相同的节点在将列表返回给客户应用之前分类结果。通过这种方法，内容平台系统通过在内容平台系统中所有节点上查询、筛选和分类结果并且然后将结果返回给客户应用而承担了该工作的更多负担。

[0004] 本发明的一个方面是针对被联网在一起的独立节点的冗余阵列，其中每个节点运行提供基于对象存储的应用的实例，其中元数据对象被存储在分布在阵列上节点中的区域的集合中，其中通过哈希元数据对象属性并提取得出的哈希值的比特的给定集合来识别给定区域。一种管理查询结果的方法包括：由多个独立节点的第一节点从客户应用接收通过条件对于对象列表的请求；由第一节点基于所接收到的请求将查询发布给所有节点；由每个节点在节点中的所述区域上使用存储在区域中的元数据对象来处理所述查询；由第一节点聚集并筛选来自所有节点的查询结果；以及由第一节点将聚集的和筛选的结果返回给客户应用。

[0005] 在一些实施例中，由每个节点处理查询包括向第一节点以集合提供查询结果，以及来自所有节点的查询结果被以集合聚集和筛选并被返回给客户应用。该方法还包括在由第一节点将聚集的和筛选结果的当前集合返回给客户应用之后，在请求和检索来自所有节点的下一结果集合之前，等候来自客户应用的对于下一结果集合的请求。将查询结果以集合提供给第一节点包括提供来自每个区域的预定数量的对象作为处理查询的结果。该方法还包括由第一节点分类所聚集和筛选的结果以产生预定的顺序。请求包括通过改变时间查询、通过目录查询、通过处理查询、通过命名空间查询、以及在结果中标记页面中一个或多个。

[0006] 本发明的另一方面针对用以在联网在一起的独立节点的冗余阵列中管理查询结果的装置，其中每个节点运行提供基于对象存储的应用的实例，其中元数据对象被存储在分布在阵列上节点中的区域的集合中，其中通过哈希元数据对象属性并提取得出的哈希值的比特的给定集合来识别给定区域。该装置包括为每个节点提供的处理器、存储器、以及查询结果管理模块。该查询结果管理模块被配置为：如果具有查询结果管理模块的节点是从客户应用接收通过条件对于对象列表的请求的第一节点，则基于所接收到的请求将查询发布给所有节点；在节点中的所述区域上使用存储在区域中的元数据对象来处理所述查询；以及如果具有查询结果管理模块的节点是第一节点，则聚集并筛选来自所有节点的查询结果，将聚集的和筛选的结果返回给客户应用。

[0007] 在特定实施例中，节点包括用于管理节点中的元数据对象的元数据管理器，该元数据管理器包括组织和提供对元数据对象的访问，其中，元数据管理器包括节点的查询结果管理模块。

[0008] 本发明的另一个方面针对计算机可读存储介质，其存储多个指令用以在联网在一起的独立节点的冗余阵列中控制数据处理器以管理查询结果，其中每个节点运行提供基于对象存储的应用的实例，其中元数据对象被存储在分布在阵列上节点中的区域的集合中，其中通过哈希元数据对象属性并提取得出的哈希值的给定字节组来识别给定区域，其中计算机可读存储介质被提供在每个节点中。多个指令包括：如果具有计算机可读存储介质的节点是从客户应用接收通过条件对于对象列表的请求的第一节点，则使得所述数据处理器基于接收到的请求将查询发布给所有节点的指令；使得所述数据处理器在节点的区域上使用存储在区域中的元数据对象处理所述查询的指令；如果具有计算机可读存储介质的节点是第一节点，则导致数据处理器通过第一节点聚集并筛选来自所有节点的查询结果，将聚集的和筛选的结果返回给客户应用的指令。

[0009] 在以下特定实施例的详细描述中，本发明的这些和其他特征和优势将对本领域技术人员变得清楚。

附图说明

[0010] 图1是可以应用本发明的方法和装置的固定内容存储器存档系统的简化方框图。

[0011] 图2是独立节点的冗余阵列的简化表示，其中每个节点都是对称的并且支持存档系统群组应用。

[0012] 图3是在给定节点上运行的存档系统群组应用的不同组件的高层表示。

[0013] 图4示出了在群组的给定节点上的元数据管理系统的组件示例。

[0014] 图5是内容平台的简化方框图，示出了通过单个节点将来自客户应用的查询分发到所有其他节点并且聚集将要返回给客户应用的查询结果。

[0015] 图6是流程图的示例，示出了通过单个节点将来自客户应用的查询分发到所有其他节点，聚集查询结果并通过单个节点将其返回给客户应用的过程。

具体实施方式

[0016] 以下将参照形成了公开一部分的附图详细描述本发明，并且在其中通过例证而非限制的方法显示了可以实现本发明的示例性实施例。在附图中，贯穿多个视图，相似的数字描述基本相似的组件。另外，应当注意的是，在如以下所描述的和如在附图中所示出的，具体实施方式提供了各种示例性实施例的同时，本发明不限于本文中所描述和显示的实施例，而是可以扩展到其他实施例，如同本领域技术人员所知或将会知道的那样。在说明书中提及的“一个实施例”、“这个实施例”、或“这些实施例”意为本发明的至少一个实施例中包括结合实施例所描述的特定特征、结构或特点，并且在说明书中的不同位置出现的这些短语不必全部指代相同的实施例。此外，在以下具体实施方式中，为了提供对本发明透彻的理解，阐明了数个特定细节。然而，对于本领域普通技术人员来说，实现本发明并不完全需要这些特定细节。在其他环境下，众所周知的结构、材料、电路、处理和接口没有被详细描述，和/或可以以方框图的形式示出，以便不会不必要的模糊本发明。

[0017] 此外，以下详细描述的一些部分被呈现为计算机内运行的算法和符号表达的形式。这些算法描述和符号表达是数据处理领域技术人员使用的手段，来最有效地将他们的创新实质传达给本领域其他技术人员。算法是导致所期望的结束状态或结果的一系列限定的步骤。在本发明中，为了实现有形结果，所执行的步骤需要有形量的物理操纵。虽然不是必要地，但通常来说，这些量表现为能够被存储、传送、组合、比较、和其他形式操纵的电或磁信号。主要因为通用的原因，已经证明将这些信号指代为位(bits)、值(values)、元素(elements)、符号(symbols)、字符(characters)、术语(terms)、数(numbers)、指令(instructions)等有时是方便的。然而，应当记住，全部这些和相似术语应当与适当的物理量相关联并且只不过是用在这些量上的方便的标签。除非特别做出与以下讨论明显相反的阐述，否则应当理解为在描述、讨论的通篇，使用诸如“处理”、“计算”、“算出”、“确定”、“显示”等术语，可以包括计算机系统或其他信息处理设备的动作和过程，他们将在计算机系统的注册表和存储器中表现为物理(电子)量的数据操纵和变换为在计算机系统存储器或注册表或其他这种信息存储器、传输或显示设备中相似地表现为物理量的其他数据。

[0018] 本发明还涉及用以执行本文中操作的装置。该装置可以是为所需目的而特定构造的，或者可以包含由一个或多个计算机程序选择性触发或重新配置的一个或多个通用计算机。这种计算机程序可以被存储在计算机可读存储器介质中，例如但不限制于光盘、磁盘、只读存储器、随机存取存储器、固态设备和驱动器、或适于存储电子信息的任何其他类型的介质。本文中所呈现的算法和显示不是固有地与任何特定计算机或其他装置相关联的。根据本文中的教导，各种通用系统都可以与程序一起使用，或者可以证实构造更专业的装置来执行所期望的方法步骤是方便的。此外，本发明不参照任何特定编程语言进行描述。将会理解的是，如本文中所描述的，多种编程语言可以被用来实现发明的教导。程序语言的指令可以由一个或多个处理设备运行，例如，中央处理单元(CPU)、处理器或控制器。

[0019] 如同以下将更详细描述的，本发明的示例性实施例提供了用以在容错数据库管理系统中聚集查询结果的装置、方法和计算机程序。

[0020] 1.固定内容分布式数据存储器

[0021] 已经发展了这种需要：对于“固定内容”的高度可用、可靠和持久方式的存档存储代替或补充传统的磁带或光学存储方案。术语“固定内容”典型的是指为了参考或其他目的，希望不进行修改而保存的任何类型的数字信息。这种固定内容的示例包括电子邮件、文档、诊断图像、检测图像、声音记录、影片和视频等等，还有许多其他示例。已经出现了传统的独立节点冗余阵列(RAIN)存储手段，作为为了存储这种固定内容信息资产而产生大型在线存档的选择架构。通过允许节点按需接入或退出群组，RAIN架构将存储器群组隔离于一个或多个节点的故障。通过在多个节点上复制数据，RAIN型存档能够自动补偿节点故障或移除。典型地，RAIN系统主要被生产为从封闭系统中的相同组件设计的硬件器件。

[0022] 图1显示了一个这种可扩展的基于磁盘的存档存储器管理系统。节点可以包括不同硬件，并且因此可以被认为是“异构的”(heterogeneous)。节点典型地具有至存储区域网络(SAN)中的一个或多个存储磁盘的通路，该存储磁盘可以是实际物理存储器磁盘或者虚拟存储器磁盘。在每个节点上支持的存档群组应用(并且，可选地，运行应用的下层操作系统)，可以是相同的或基本上相同的。每个节点上的软件栈(可以包括操作系统)是对称的，而硬件可以是异构的。如图1所示，企业能够使用系统，产生对于许多不同类型的固定内容信息的永久存储，所述固定内容信息诸如文档、电子邮件、卫星图像、诊断图像、检测图像、声音记录、视频等等，以及其他信息。这些类型当然仅仅是示例。通过在独立服务器或者称为存储节点上复制数据，实现了高度可靠性。优选地，每个节点与其对等点(peer)对称。这样，因为优选地任何给定节点能够执行所有功能，因此任何一个节点的故障在存档的可用性上产生很少的影响。

[0023] 如同在共同拥有的美国专利No.7,155,466中所描述的，已知在基于RAIN的存档系统中，并入在每个节点上运行的捕捉、保存、管理和检索数字资产的分布式软件应用。图2显示了一个这种系统。将单独存档的物理范围指代为群组(或系统)。典型地，群组不是单个设备，而是设备集。设备可以是同类的或异构的。典型的设备是运行诸如Linux操作系统的计算机或机器。托管在日常硬件上的基于Linux系统的群组提供了能够变比的存档，从几个存储器节点服务器变比至存储数以千计的数兆字节数据的许多节点。这种构架保证了存储能力能够与组织结构增长的存档需求一直并驾齐驱。

[0024] 在诸如以上描述的存储系统中，典型地将数据随机分布在群组上，以使得一直保护存档不发生设备故障。如果磁盘或节点故障，则群组自动在维持相同数据复本的群组中的其他节点上停止作用。该手段在数据保护的角度上运作良好的同时，为群组计算的数据丢失平均时间(MTDL)可能没有所期望的高。具体来说，MTDL典型地代表计算出的在存档将丢失数据之前的时间量。在数字存档中，不希望任何数据丢失，但是由于硬件和软件组件的特性，一直都存在这种事件发生的可能性(无论多遥远)。由于对象和它们的副本在存档群组中随机分布，因而MTDL可能低于需求而结束，例如，如果由于在给定节点中的给定磁盘(镜像副本被存储在其上)意外地故障了，则所需的对象副本不可用。

[0025] 如图2所示的，实现本发明的示意性群组优选地包括以下通常类型的元件：节点202、一对网络交换机204、配电单元(PDU)206、以及不中断电源(UPS)208。节点202典型地包

括一个或多个日常服务器并包含CPU(例如,Intel×86、适当的随机存取存储器(RAM)、一个或多个硬盘驱动器(例如,标准IDE/SATA、SCSI等)、以及两个或多个网络接口(NIC)卡。典型的节点是具有2.4GHz芯片、512MBRAM以及六(6)个200GB硬盘驱动器的2U机架式单元。然而,这并不是限制。网络交换机204典型地包括能够在节点之间点对点通信的内部交换机205和允许额外的群组访问每个节点的外部交换机207。每个交换机需要足够的端口来操作群组中所有潜在的节点。以太网或GigE交换机可以被用于这个目的。使用PDU206为所有节点和交换机供电,并且使用UPS208保护所有节点和交换机。尽管不意为限制,但是典型的群组可以被连接到网络,诸如公共互联网、企业内部互联网、或其他广域或局域网。在示例性实施例中,在企业环境内实现群组。例如,通过在站点的集体域名命名系统(DNS)命名服务器中导航可以实现。例如,群组域因此可以是现存域的新的子域。在代表性的实现中,在集体DNS服务器中将子域委托给在群组自身中的命名服务器。终端用户使用任何传统的接口或访问工具访问群组。例如,这样可以在任何基于IP的协议(HTTP、FTP、NFS、AFS、SMB、网页服务等)上经由API,或通过任何其他已知或以后将研发的访问方法、服务、程序或工具来实现对群组的访问。

[0026] 客户应用通过诸如标准UNIX文件协议,或HTTPAPI的一个或多个类型的外部网关访问群组。优选地通过虚拟文件系统来公开存档,该虚拟文件系统能够随意符合任何标准的面向UNIX文件协议的设施。这些包括NFS、FTP、SMB/CIFS等。

[0027] 在一个实施例中,联网在一起(例如,经由以太网)成为群组的独立节点的冗余阵列(H-RAIN)上运行存档群组应用。给定节点的硬件可以是异构的。然而,为了得到最大可靠性,每个节点优选地运行分布式应用的实例300(可以是相同实例,或基本上相同的实例),其包括多个运行时组件,如图3中所示。这样,在硬件可以是异构的同时,节点上的软件栈(至少当它涉及本发明时)是相同的。这些软件组件包括网关协议层302、访问层304、文件处理和管理层306以及核心组件层308。为了说明的目的而提出"层"的名称,如同普通技术人员将领会的,可以用其他有意义的方法来体现功能的特征。可以集成或不集成一个或多个层(或其中的组件)。一些组件可以在层之间共享。

[0028] 网关协议层302中的网关协议提供现存应用的透明。具体来说,网关提供诸如NFS310和SMB/CIFS312的本地文件服务,以及网页服务API来建立客户应用。还提供了HTTP支持314。访问层304提供对存档的访问。具体来说,根据本发明,固定内容文件系统(FCFS)316模拟本地文件系统来提供对存档对象的完全访问。FCFS给予了应用对存档内容的直接访问,如同他们是普通文件一样。优选地,存档的内容表现为其原始格式,同时将元数据公开为文件。FCFS316提供了传统观点的目录和访问权限和流程的文件级(file-level)调用,使得管理者能够以他们所熟悉的方式来规定固定内容数据。文件访问调用优选地被用户空间后台程序中断,并被路由给适当的核心元件(在层308)中,其对调用应用动态地产生适当的视图。FCFS调用优选地被存档策略约束,以辅助自主的存档管理。因此,在一个实施例中,管理者或应用不能删除保存期限(给定策略)仍然有效的存档对象。

[0029] 访问层304优选地还包括网页用户界面(UI)318和SNMP网关320。网页用户界面318优选地实现为管理者控制台,提供对文件处理和管理层306中的管理引擎322的交互访问。管理控制台318优选地是密码保护的,提供存档的动态视图的基于网页的GUI,包括存档对象和单独节点。SNMP网关320向存储管理应用提供了对管理引擎322的轻易访问,这使得他

们安全地监视和控制群组动作。管理引擎监视群组动作，包括系统和策略事件。文件处理和管理层306还包括请求管理器处理324。请求管理器324协调来自外部世界(经过访问层304)的全部请求，以及来自核心组件层308中的策略管理器326的内部请求。

[0030] 除策略管理器326以外，核心组件还包括元数据管理器328和一个或多个存储管理器330的实例。元数据管理器328优选地被安装在每个节点上。共同地，元数据管理器在群组中作为分布式数据库，管理所有存档对象。在给定的节点上，元数据管理器328管理存档对象的子集，其中优选地，每个对象在外部文件(“EF”，为了存储而进入存档的数据)和存档数据物理所在的内部文件(每个都是“IF”)组之间映射。相同的元数据管理器328还管理从其他节点上复制的存档对象组。这样，每个外部文件的当前状态对于数个节点上的多个元数据管理器一直都是可用的。如果发生节点故障，则其他节点上的元数据管理器继续提供对先前由故障节点管理的数据的访问。存储管理器330提供了文件系统层，对于分布式应用中的所有其他组件都可用。优选地，它将数据对象存储在节点本地文件系统中。在给定节点中的每个驱动器优选地具有其自己的存储管理器。这使得节点可以移除单个驱动器并且优化吞吐量。存储管理器330还提供系统信息、数据的完整性检查，以及能够直接遍历本地结构。

[0031] 如图3中还显示了，通过通信中间件层332和DNS管理器334，群组管理内部和外部通信。设施332是高效和可靠的基于消息的中间件层，其使能存档组件间的通信。在所示实施例中，层支持多点传送和点对点通信。DNS管理器334运行将所有节点连接至企业服务器的分布式命名服务。优选地，DNS管理器(单独或者与DNS服务相结合)在所有节点上加载平衡请求，以确保最大的群组吞吐量和可用性。

[0032] 在所示实施例中，ArC应用实例在基本操作系统336上运行，诸如RedHat Linux9.0、Fedora Core6等。通信中间件是任何传统的分布式通信机制。其他组件可以包括FUSE(USErspace中的文件系统)，其可以被用于固定内容文件系统(FCFS)316。NFS网关310可以由标准nfsd Linux Kernel NFS驱动来实现。每个节点中的数据库可以被实现，例如PostgreSQL(本文中也被称为Postgres)，其是对象关系型数据库管理系统(ORDBMS)。节点可以包括网页服务器，诸如Jetty，其是Java HTTP服务器和小服务程序容器。当然，以上机制仅仅是说明性的。

[0033] 给定节点上的存储管理器330负责管理物理存储设备。优选地，每个存储管理器实例负责单个根目录，在单个根目录中所有文件根据其放置算法被放置。可以在节点上同时运行多个存储管理器实例，并且每个通常代表系统中不同的物理磁盘。存储管理器提取在其余系统中使用的驱动器和接口技术。当存储管理器实例被请求以写入文件时，它生成其负责的用以表达的完全路径和文件名。在代表性的实施例中，存储在存储管理器上的每个对象被接收为原始数据而被存储，存储管理器随后在存储数据时将其自己的元数据添加到文件，以寻迹不同类型信息。外部文件(EF)存储随后由查询引擎(Query Engine)进行查询中将需要的信息。例如，元数据包括而不限于：EF长度(外部文件长度字节)、IF段尺寸(该片内部文件的尺寸)、EF保护表达(EF保护模式)、IF保护作用(该内部文件的表达)、EF创建时间戳(外部文件时间戳)、签名(写入(PUT)时内部文件的签名，其包括签名类型)、以及EF文件名(外部文件文件名)。与内部文件数据一起存储该附加元数据提供了额外程度的保护。具体来说，清除(scavenging)能够通过内部文件存储的元数据在数据库中创建外部文件记录。其他策略可以验证内部文件哈希相对于内部文件以确认内部文件保持完好。

[0034] 内部文件可以是数据“组块(chunk)”,代表在存档对象中的原始“文件”的一部分,并且他们可以被放置在不同节点上以存档条带(stripe)并保护块(block)。然而,并不必须将外部文件分离为更小的组块单元;在替选实施例中,内部文件可以是外部文件的完整副本。典型地,在元数据管理器中,对于每个存档对象提供一个外部文件条目,同时对于每个外部文件条目可以存在许多内部文件条目。典型地,内部文件布局取决于系统。在给定实现中,磁盘上的该数据的真实物理格式被存储为一系列可变长度记录。

[0035] 请求管理器324负责运行通过与系统中的其他组件交互而执行存档动作所需的操作组。请求管理器支持不同类型的许多同时动作,能够重新执行任何失败的处理,并支持可能花费长时间来运行的处理。请求管理器还确保在存档中的读/写操作被适当操作并且确保所有请求在所有时间上处于已知状态。其还提供处理控制,用以配合节点上的多个读/写操作以满足给定的客户请求。此外,请求管理器为最近使用过的文件高速缓存元数据管理器条目,并为会话和数据块提供缓冲。

[0036] 群组的主要职责是在磁盘上可靠地存储无限数量的文件。可以将给定节点想作是“不可靠的”,感觉像它是因为任何原因而不可到达或者其他方式不可用的。收集这种潜在的不可用节点有助于创建可靠的和高度可用的存储。通常,存在两种类型的信息需要被存储:文件本身和关于文件的元数据。固定内容分布式数据存储额外的细节可以在美国专利公开2007/0189153和2006/0026219中找到,以引用的方式将他们合并入本文中。

[0037] II.元数据管理

[0038] 元数据管理系统负责组织给定元数据(诸如系统元数据)和提供对它的访问。该系统元数据包括位于存档中的文件的信息以及配置信息、在管理UI上显示的信息、公制(metrics)、不可修复的策略违规信息等。尽管不详细说明,但是其他类型的元数据(例如,与存档文件相关的用户元数据)也可以使用现在描述的元数据管理系统管理。

[0039] 在群组的代表性实施例中,元数据管理系统为元数据对象组提供持续性,这可以包括一个或多个以下对象类型(仅仅是说明性的):

[0040] ExternalFile(外部文件):存档用户所意识到的文件;

[0041] InternalFile(内部文件):存储管理器存储的文件;典型地,在外部文件和内部文件之间可以是一对多关系。

[0042] ConfigObject(配置对象):用于配制群组的名称/值对;

[0043] AdminLogEntry(管理日志条目):将在管理者UI上显示的消息;

[0044] MetricsObject(公制对象):时间戳密钥/值对,代表在时间点上一些存档的量度(例如,文件数量);以及

[0045] PolicyState(策略状态):一些策略的违规。

[0046] 每个元数据对象可以具有唯一名称,优选地从不改变。元数据对象被组织为区域(region)。区域包括授权区域副本和“错误容点”(TPOF)数量(零或多的组)的备份区域副本。具有零个副本,元数据管理系统是可变比的,但是可能不是高度可用的。通过哈希一个或多个对象属性(例如,对象名称,诸如完全合格的路径名,或其部分)和提取给定数量比特的哈希值来选择区域。这些比特包括区域数量。选择的比特可以是低位比特、高位比特、中间位比特,或单个比特的任何组合。在代表性实施例中,给定的比特是哈希值的低位比特。对象的一个或多个属性可以使用任何传统的哈希函数而被哈希。这些包括而不限于,基于

Java的哈希函数，诸如java.lang.string.hashCode等。优选地，包括区域数量比特数由配置参数控制，本文中指代为regionMapLevel(区域映射层)。例如，如果该配置参数被设定为6，则这样的结果是2^6=64个区域。当然，区域数量大是允许的，并且可以使用命名空间分区方案自动地调整区域数量。

[0047] 每个区域可以被冗余存储。如以上所说明的，存在区域的一个授权副本，和零个或多个备份副本。备份副本的数量由元数据TPOF配置参数控制，如同已经描述过的。优选地，区域副本被分布在群组的所有节点上，以便平衡每个节点授权区域副本的数量，并平衡每个节点全部区域副本的数量。

[0048] 元数据管理系统将元数据对象存储在每个节点上运行的数据库中。该数据库被用来支持区域映射。示例性数据库使用PostgreSQL实现，其可以作为开源使用。优选地，对于每个区域副本存在概要(schema)，并且在每个概要中，存在对于每种类型的元数据对象的表格。概要仅仅是命名空间，可以拥有表格、索引、过程和其他数据库对象。每个区域优选地具有其自己的概要。每个概要具有完整的表格组，每个元数据对象一个表格。这些表格中的一个表格的行对应于单个元数据对象。同时，Postgres是优选的数据库，任何传统的关系型数据库(例如，Oracle、IBM DB/2等)都可以被使用。

[0049] 如图4中所示的，每个节点400具有一组过程或组件：一个或多个区域管理器(RGM)402a-n、元数据管理器(MM)404、至少一个元数据管理客户(MMC)406、以及具有一个或多个概要410a-n的数据库408。RGM、MM和MMC组件与虚拟机412(诸如Java虚拟机)一起运行。对于每个区域副本存在一个RGM。这样，对授权的区域副本具有RGM、对每个备份区域副本具有RGM、并且对于每个不完全的区域副本具有RGM。对于每个RGM 402还具有由其管理该概要的数据库概要410。数据库还存储区域映射405。每个节点优选地具有区域映射的相同全局查看，以及由同步方案实施的要求。区域管理器RGM402负责在区域副本上操作(作为可能的情况，可能是授权的、备份的或不完全的)，并且用以运行由元数据管理器客户406和其他区域管理器402提交的请求。通过任何传统的手段将请求提供给RGM，这种方法是诸如图3中所示的通信中间件或其他消息层。区域管理器提供这些请求运行的环境，例如通过提供对数据库的连接，该环境被配置为在由该RGM管理的概要上操作。每个区域管理器将它的数据存储在数据库408中。元数据管理器404是顶级组件，负责节点上的元数据管理。其负责生成和销毁区域管理器(RGM)并组织RGM所需的资源，例如群组配置信息和数据库连接池。优选地，(给定节点中的)给定元数据管理器用做领导者并且负责决定哪些元数据管理器(在节点集或子集上)负责哪些区域副本。领导者选择算法，诸如欺负(bully)算法或其变形，可以被用来选择元数据管理器领导者。优选地，尽管可能每个节点运行多个MM，但是每个节点具有单个元数据管理器。一旦区域拥有关系被命名空间分区方案建立(如以下将描述的)，每个元数据管理器负责相应地调整其一个或多个区域管理器集。系统组件(例如，管理引擎、策略管理器等)与元数据管理器MM通过元数据管理器客户交互。MMC负责(使用区域映射)定位RGM来执行给定请求，以此将请求发布给所选RGM，并且以此如果所选的RGM不可用(例如，因为节点已经故障)则重新尝试请求。在后种情况中，当在节点上接收了新的区域映射时，重新尝试请求将会成功。

[0050] 如以上所提到的，区域映射用于识别负责每个区域的每个副本的节点。虚拟机412(和其中的每个RGM、MM和MMC组件)具有对区域映射405的访问通路；区域映射的副本420在

已经被复制到JVM之后，也显示在图4中。区域映射因此对于给定节点中的JVM和数据库都可用。在这种示例的实施例中，每个元数据对象具有属性（例如名称），其被哈希为0x0和0x3fffffff之间所包括的整数字段，也就是30比特值。这些值能够充分地表示带符号的32比特整数，而不会出现溢出问题（例如，当向范围的高端添加1时）。30比特允许高达近10亿个区域，甚至对于大型群组都是充足的。区域表示哈希值集，并且所有区域的集覆盖所有可能的哈希值。对于每个区域，具有不同比特的位置，并且不同比特的位置优选地是固定顺序。这样，每个区域以数字来识别，其优选地通过提取哈希值的RegionLevelMap（区域层映射）比特而得到。其中，配置参数被设置为6，这允许64个区域，所得出的哈希值是数字0x0至0x3f。

[0051] 如以上说明的，区域副本是三（3）种状态之一：“授权的”“”“备份的”和“不完全的”。如果区域副本是授权的，那么对区域的所有请求都去向该副本，并且对每个区域具有一个授权的副本。如果区域副本是备份的，那么副本（从授权的区域管理器过程）接收备份请求。如果加载了元数据，但是副本还没有同步（典型地，关于其他备份副本），那么区域副本是不完全的。直到同步完成，在副本变为备份副本的时刻，不完全的区域副本还不适合升级到其他状态。每个区域具有一个授权的副本和给定数量（如同由元数据TPOF配置参数设定的）的备份的或不完全的副本。

[0052] 通过在授权区域副本和其TPOF备份副本之间实施给定的协议（或“合同”），备份区域副本与授权区域副本被保持同步。现在介绍该协议。

[0053] 通过简要背景，当MMC上接收到更新请求时，MMC在本地区域映射上浏览以查找授权区域副本的位置。MMC将更新请求发送到与授权区域副本相关联的RGM，该RGM随后对该更新请求承认（commit）。更新还被发送（通过与授权的区域副本相关联的RGM）给每个TPOF备份副本的RGM。然而，授权的RGM为了指示成功，不需要等待与备份区域副本相关联的每个RGM来承认更新；相反的，当与备份区域副本相关联的RGM接收到更新时，其立即返回或试图返回（给授权的RGM）确认。当接收到备份请求时并且在它被运行之前发布该确认。在没有故障发生的情况下，一旦授权的RGM接收到所有确认，其告知随后将成功结果返回给呼叫者的MMC。然而，如果发生了给定故障事件，则协议保证受冲击的RGM（无论备份的还是授权的）从服务中移除其自身（和潜在受影响的节点），并且MM领导者发布新的区域映射。尽管任何方便的技术都可以被使用，但是优选地，RGM通过卸掉JVM将其自身从服务中移除。新映射指定对丢失的区域副本的替代。以这种方式，每个备份区域副本都是对授权区域副本的“热备份”并且如果并且需要时（因为授权的RGM故障或者负载平衡的目的等），适合将每个备份区域副本升级为授权的。

[0054] 还存在更新过程可能失败的许多方式。例如，因此授权的区域管理器（在等待确认的同时）可以遇到这样的例外：这种例外指示备份管理器过程已经不运行，或即使备份管理器已经发布了确认，但备份管理器过程仍可能不能本地处理更新请求；或备份区域管理器过程在发布确认的同时可能遇到指示授权的区域管理器过程已经不运行的例外，等等。如以上说明的，如果给定备份RGM不能处理更新，则将其自身从服务中移除。此外，当备份RGM或授权的RGM不运行时，发布新的区域映射。

[0055] 元数据管理系统将区域的副本保持同步。对授权的区域副本中的对象完成的更新被复制到备份区域副本上。一旦更新被授权的RGM承认，那么对于所有备份区域副本应用相

同的更新。元数据管理系统确保在故障节点上的任何这种故障(无论节点层级、区域管理器层级等)都导致区域副本的重新分配;从而,保证了剩余区域副本的完整性。如果包含授权的RGM的节点故障,那么备份RGM或者处于同步(当前运行更新或者当前没有运行更新),或者他们仅仅通过被中断的更新而不同步。在后种情况下,重新同步是容易的。因为备份区域被保持为与授权区域同步,因此升级(从备份至授权)是瞬间的。

[0056] 节点故障还很有可能丢失备份区域。通过在某个其他节点上创建新的、不完全的区域来恢复备份区域。一旦创建不完全的区域,它就开始记录更新并开始从授权区域复制数据。当复制完成时,应用积累的更新,产生最新的备份。新备份区域随后通知MM领导者,它已经是最新的,这将导致MM领导者发出包括该区域升级(从不完全的到备份的)的映射。

[0057] 应当注意的是,不需要区域的数量对应于节点的数量。更通常来说,区域的数量与独立节点阵列中节点的数量是无关的。元数据管理的额外细节可以在美国专利公开2006/0026219中找到。

[0058] III.由节点聚集查询结果

[0059] 本发明的示例性实施例允许REST(表象化状态转换)客户对符合给定标准的对象和元数据列表查询平台内容,而不需要整合应用来遍历目录、子目录、次子目录等等为给定标准检索对象和系统元数据的列表。本发明的特征包括客户应用能够通过改变时间进行查询、通过目录查询、通过处理(生成、删除、清除)查询、通过命名空间查询、以及在结果中标记页面等。单个内容平台节点将查询分布到内容平台系统中的所有节点上的所有区域并且在将列表返回给客户应用之前使得相同节点分类结果。根据特定实施例,在元数据管理器中实现数据库查询。

[0060] “改变时间”是用户最后修改对象(具体来说,其元数据,因为内容平台系统中的内容是只读的)的时间。例如,自从1970年1月1日开始以毫秒数测量时间。“通过目录查询”的作用是检索在逻辑上驻存于相同文件系统目录的内容平台系统中的所有对象。内容平台系统通过对其数据库运行SQL查询来实现这点。“通过处理查询”的作用是检索由特定类型操作最近访问的内容平台系统中的所有对象。例如,其可以返回最近的动作是创建的所有对象,或者最近被删除的所有对象。“通过命名空间查询”的作用是检索在特定内容平台系统命名空间中的所有对象,并且仅仅检索这些对象。“在结果中标记页面”的作用是重复查询集的结果集,而不是单个对象。例如,可能1000个对象满足查询。在传统的重复中,客户应当每次一个地检索并检验这些对象,这需要1000次重复。在标记页面方案中,他们可以以50、100或一些其他数量成批返回给客户,这减少了遍历结果所必需的重复数量。

[0061] III.A.查询定义

[0062] 图5是内容平台500的简化方框图,示出了通过单个节点510将来自客户应用502的查询分发到所有其他节点512、514、516。查询结果随后在他们被返回给HTTP客户应用502之前被筛选和分类。内容平台500包括联网在一起的独立节点的冗余阵列。在每个节点中处理查询,以通过查找符合查询条件的对象来提供查询结果。筛选查询结果意味着在结果集中,仅仅包括符合REST客户给定的一个或多个条件的结果。由操作类型(即,“生成”、“删除”、“元数据改变”等)筛选对象。例如,如果客户只希望看到transaction=create records(处理=生成记录),那么访问所有记录,并且仅仅包括符合transaction=create(处理=生成)的记录。

[0063] 在步骤1,中,应用502向查询分布节点510的第一节点或领导节点发送查询。在步骤2中,在从应用502接收到请求的基础上,第一节点510将查询发布给内容平台500中的每个其他节点512、514、516。在步骤3中,内容平台500中的每个节点可查询节点内的所有授权区域并提供结果集。在步骤4中,第一节点510持续检索、聚集、以及筛选并分类来自所有节点的结果。在步骤5中,第一节点510将结果集返回给应用502。在步骤6中,应用502可以向第一节点510发布下一组结果的请求,并且重复以上步骤2-5。每个区域优选地映射至优化的数据库,以优化通过使用数据库索引来处理关于对象改变时间的查询。

[0064] 图6流程图的示例示出了来自客户应用502的查询通过第一节点510分发到所有其他节点,并且聚集查询结果并将其返回给客户应用502的过程。在该示例中,查询在提供的UUID(通用唯一标示符)、目录路径、和change_time(改变时间)中选择100个对象。由查询产生的顺序是(uuid,change_time,fn_hash)。

[0065] 在步骤602中,客户应用502以特定条件对对象列表做出请求。在步骤604中,接收请求的第一节点510向所有其他节点发布查询。在步骤606中,每个节点在其中的区域上开始查询。在步骤608中,每个节点为节点上的每个区域接收首个100个结果。在步骤610中,第一节点510从所有节点接收原始结果集。在步骤612中,第一节点聚集并筛选并分类结果(这是持续进行的,直到客户应用502停止请求下一结果集)。在步骤614中,第一节点510将结果返回给客户应用502。在步骤616中,客户应用接收结果,直到完成(这是直到客户应用502停止请求下一结果集才完成的)。在步骤618中,客户应用502将下一结果集的请求发送给第一节点502。在步骤620中,第一节点502请求来自所有节点的聚集并筛选并分类的额外结果,直到完成(这是直到客户应用502停止请求下一结果集才完成的)。

[0066] 管理查询结果的过程(即,发布和处理查询,并且聚集和筛选查询结果)可以实现为查询结果管理模块。在特定实施例中,在内容平台的每个节点的元数据管理器中提供查询结果管理模块。

[0067] III.B.CPExternalFileQueryRequest(CP外部文件查询请求)

[0068] CPExternalFileQueryRequest(CP外部文件查询请求)是来自元数据管理器客户对特定授权的区域号码的请求。该请求从符合QueryParameters(查询参数)的external_file(外部文件)表格中返回1批。CPExternalFileQueryRequest将调用如以上篇幅中描述的相同的查询。如以上所述,查询被排序为(uuid,change_time,fn_hash)。在返回批次之前,列表进一步被分类(在存储器中)以产生该确切的顺序:

[0069] (uuid,change_time,fn_hash,directory,file_name,version_id)。该特征可以实现为软件模块以提供第一节点510和其他节点之间的通信,用以请求下一组/批结果。“uuid”是对象的通用唯一标示符。在这种情况下,其识别对象所在的命名空间。“change_time”反映了记录最后被修改的日期和时间。“fn_hash”代表对命名空间中的对象名称应用哈希函数的结果,并且被用于识别对象的缩写。参见美国专利公开2006/0026219。“version_id”是命名空间中对象的特定版本的唯一标示符。

[0070] III.C.RemoteQueryBatchIterator(远程查询批次迭代器)

[0071] RemoteQueryBatchIterator(远程查询批次迭代器)是发送CPExternalFileQueryRequest消息以检索批次的批次迭代器的简单扩展。这与通常查询本地区域的批次迭代器的典型实现稍有不同。查询引擎被约束为特定区域号码和关于生成的

映射尺寸。该特征可以实现为软件模块，以便当第一节点510从客户应用502接收到以特定条件对对象列表的查询请求时，从第一节点510向其他节点发布查询。

[0072] III.D.ExternalFileQuery(外部文件查询)

[0073] ExternalFileQuery(外部文件查询)类是MetadataManagerClient.Operation实现，其合并系统中所有查询引擎。因为从RemoteQueryBatchIterator返回的每个查询引擎都被严格排序，PriorityQueueIterator(优先队列迭代器)能够将这些查询引擎高效的合并。返回的查询将是MetadataIterator<ExternalFile>类型。该特征可以实现为软件模块，以聚集和筛选由第一节点502从所有节点收集的结果。

[0074] 用以合并所有区域的算法相当直接：(1)迭代给定节点中的所有区域，(2)生成RemoteQueryBatchIterator(Region,QueryParameters)，(3)生成PriorityQueryIterator(Collection<RemoteQueryBatchIterator>iterators,QueryKeyExtractor)，以及(4)返回PriorityQueryIterator。

[0075] 根据本发明的特定实施例，以上所述的用以聚集查询结果的技术是元数据查询引擎的一部分，有助于为内容平台与搜索引擎、备份服务器、策略服务器、策略引擎、使用RBS的应用、使用XAM的应用等的集成提供支持。

[0076] 当然，如图1和图5中所示的系统配置是可以实现本发明的存储存档的纯粹示例，并且本发明不限于特定硬件配置。实现本发明的计算机和存储系统还可以具有已知的I/O设备(例如，CD和DVD驱动器、软盘驱动器、硬驱动器等)，其可以存储并读取用以实现以上描述的本发明的模块、程序和数据结构。这些模块、程序和数据结构可以被编码在这种计算机可读介质上。例如，本发明的数据结构能够被存储在独立于用在本发明中的程序所在的一个或多个计算机可读介质的计算机可读介质上。系统的组件可以通过数字数据通信的任何形式或介质互相连接，例如，通信网络。通信网络的示例包括局域网、广域网，例如互联网、无线网、存储区域网等。

[0077] 在说明中，为了解释的目的呈现了数个细节，以便于提供对本发明的透彻理解。然而，对本领域技术人员将清楚的是，为了实现本发明，并不需要所有这些特定细节。还会注意到，本发明可以描述为通常以流程图、流向图、结构图或框图描述的过程。尽管流程图可以将操作描述为有序的过程，但是许多操作可以并行或同时执行。此外，可以重新安排操作的顺序。

[0078] 如本领域所公知的，以上描述的操作可以由硬件、软件或软件和硬件的某些组合执行。本发明实施例的不同方面可以使用电路和逻辑设备(硬件)来完成，同时其他方面可以使用存储在机器可读介质(软件)上的指令来完成，如果使用处理器运行会导致处理器执行实现本发明实施例的方法。此外，本发明的一些实施例可以在硬件中单独执行，而其他实施例可以以软件单独执行。此外，所描述的不同功能可以在单个单元中执行，或者可以以任何方法散布在多个组件上。当通过软件执行时，基于存储在计算机可读介质上的指令，方法可以由诸如通用计算机的处理器运行。如果需要，指令可以存储在压缩和/或加密格式的介质上。

[0079] 从前述，将清楚的是，本发明提供了用以在容错数据库管理系统中聚集查询结果的方法、装置和存储在计算机可读介质上的程序。此外，在本文中已经描述和说明特定实施例的同时，本领域普通技术人员将会领会，通过计算以实现相同目的的任何布置都可以取

代公开的特定实施例。该公开意欲覆盖本发明的任何和所有改变或变形，并且其还被理解为在所附权利要求中使用的术语不应当被解释为将本发明限制为说明书中公开的特定实施例。相反的，本发明的范围将完全由所附权利要求确定，其将根据要求阐释所建立的规则而被解释，以及这种要求所授权的等同物的全部范围。

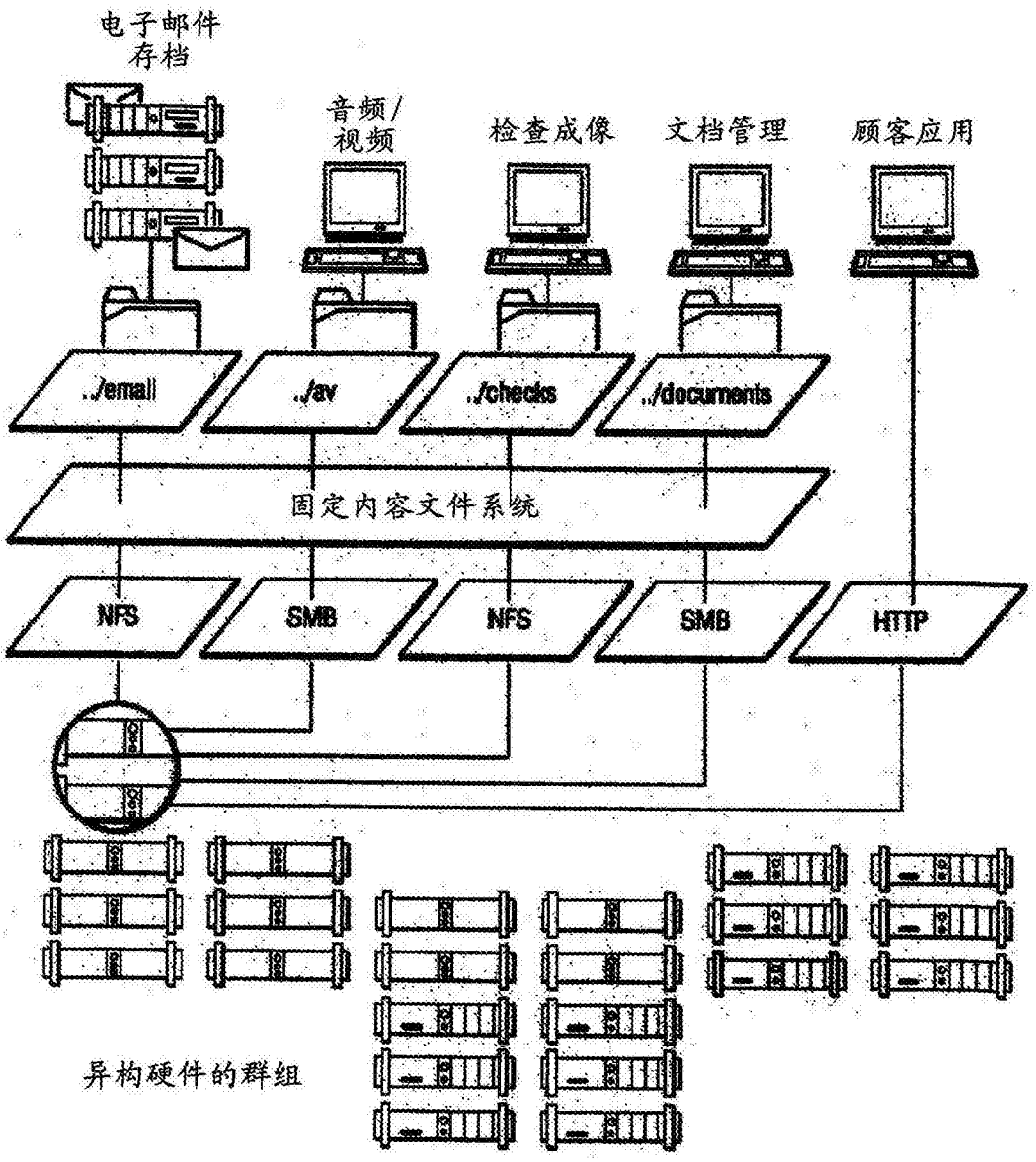

电子邮件
存档
音频/
视频
检查成像
文档管理
顾客应用
./email
./av
./checks
../documents
固定内容文件系统
NFS
SMB
NFS
SMB
HTTP
异构硬件的群组

图1

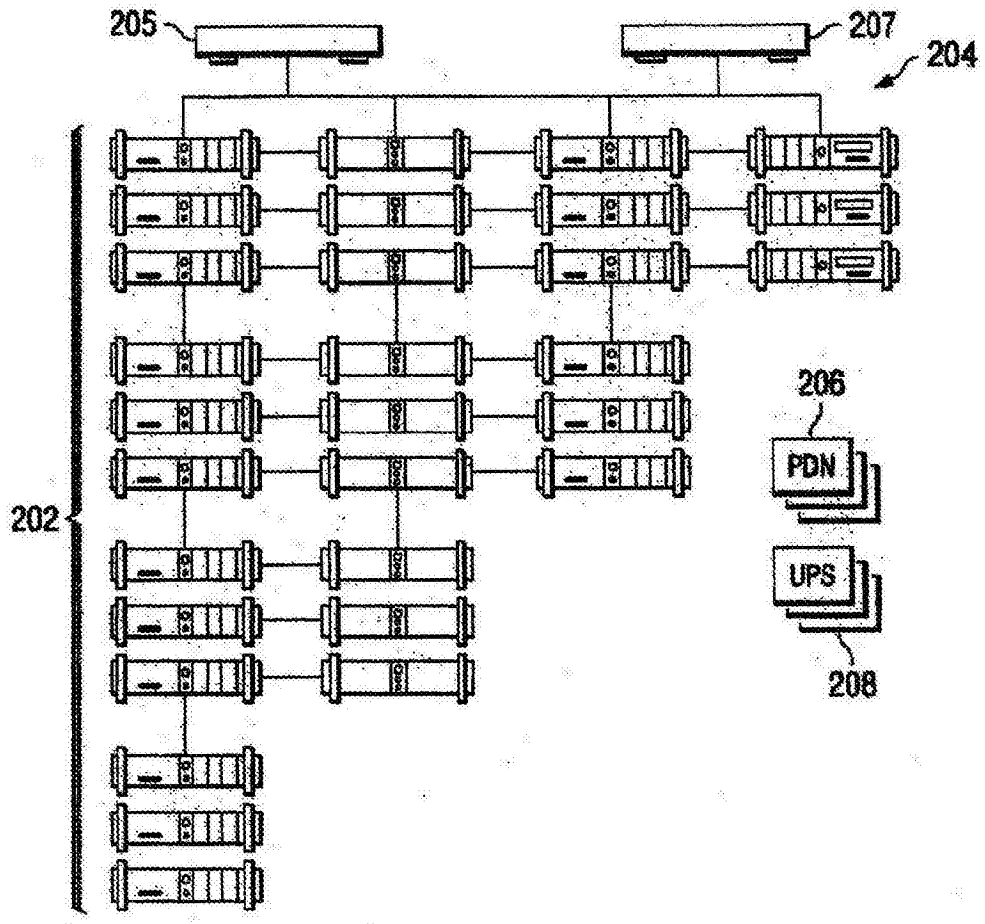


图2

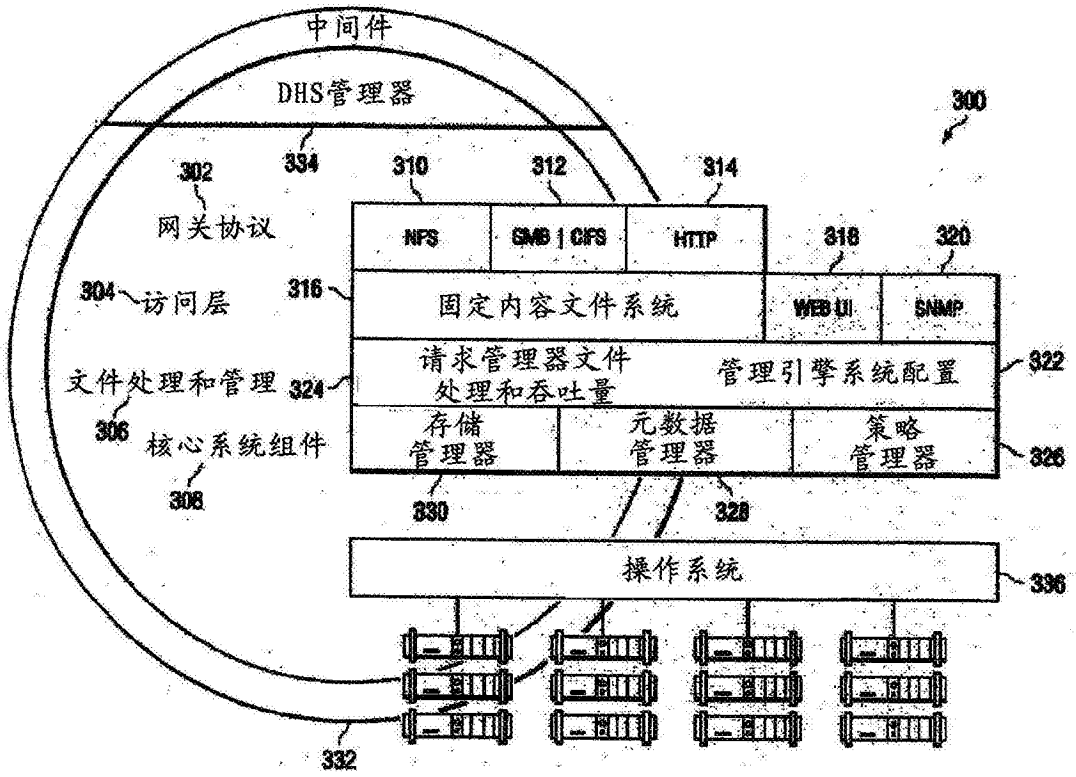

中间件
DHS管理器
300
334
310
312
314
302
网关协议
NFS
SMB | CIFS
HTTP
318
320
304
访问层
316
固定内容文件系统
WEB UI
SNMP
文件处理和管理
324
请求管理器文件
处理和吞吐量
管理引擎系统配置
322
306
核心系统组件
存储
管理器
元数据
管理器
策略
管理器
326
308
330
328
操作系统
336
332

图3

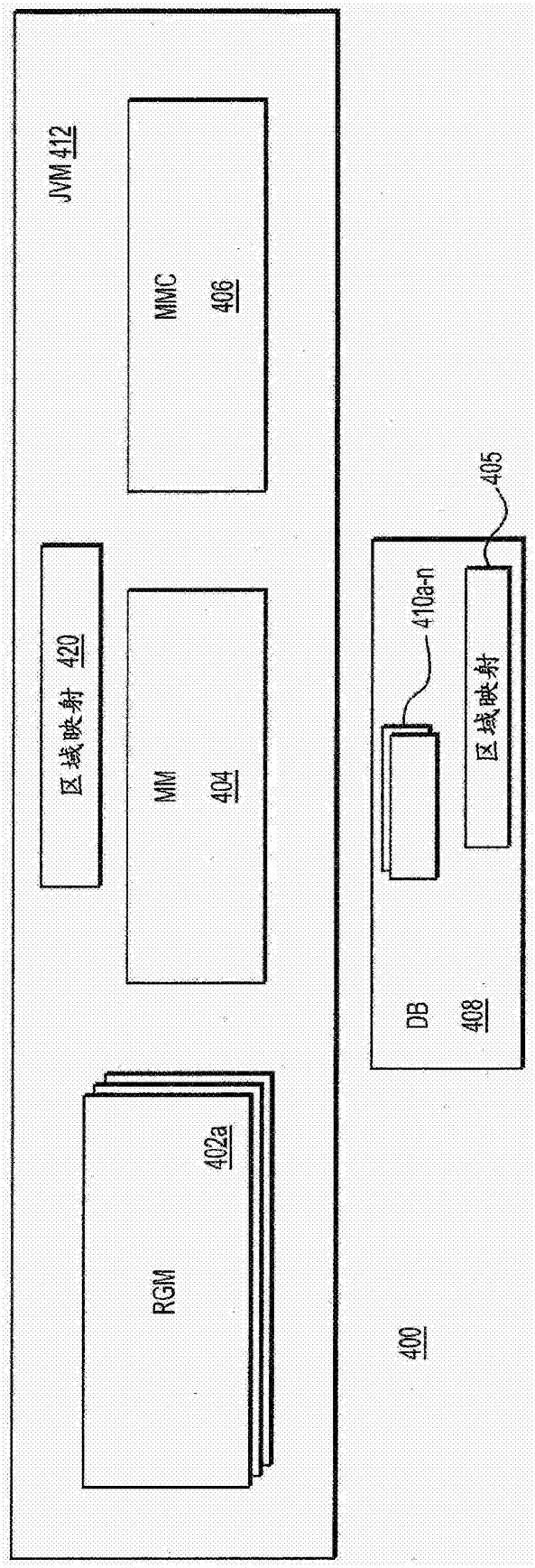

JVM 412
MMC 406
区域映射 420
MM 404
RGM 402a
400
405
410a-n
区域映射
DB 408

图4

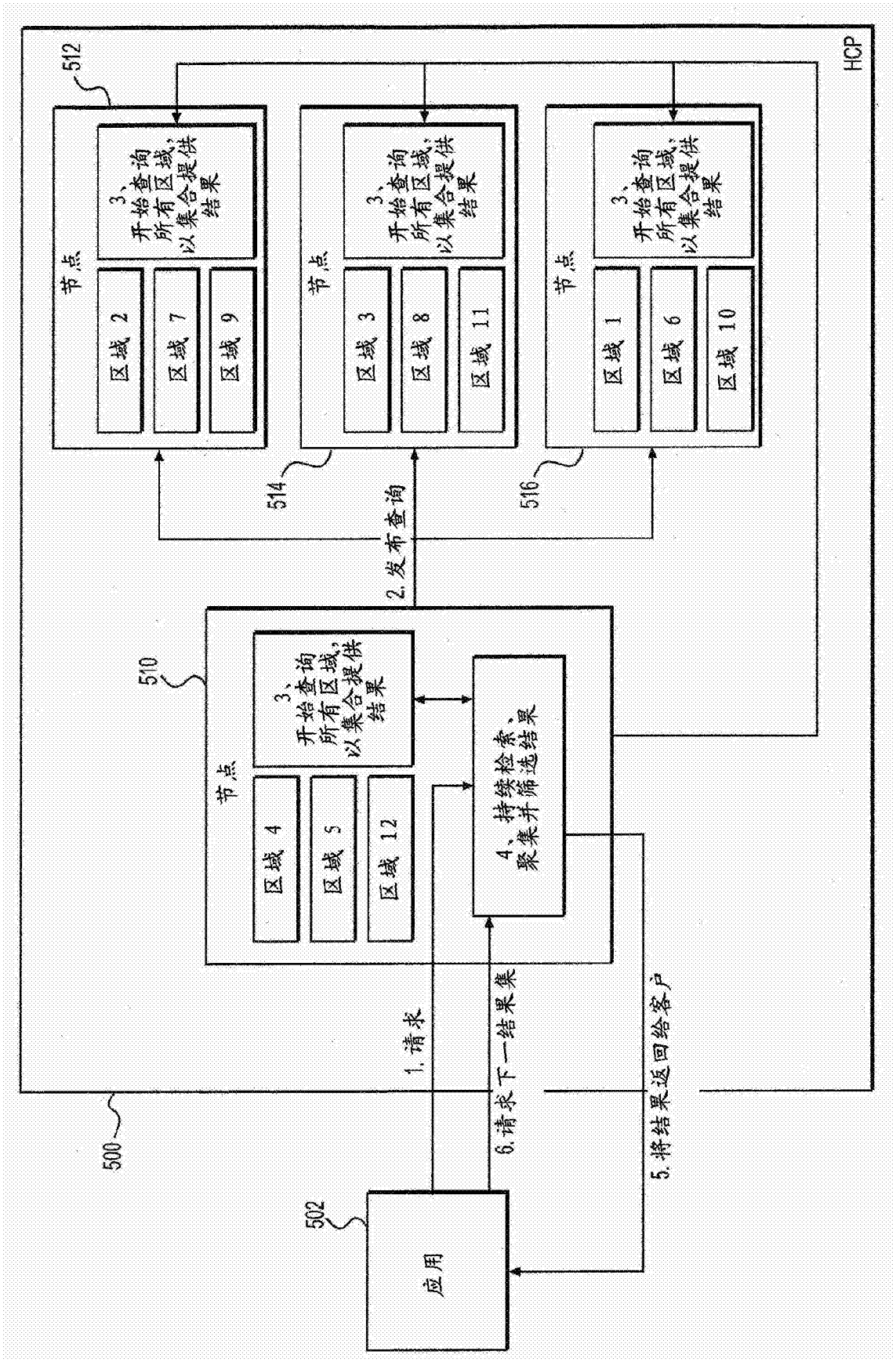

HCP
512
节点
3、开始查询所有区域，以集合提供结果
区域 2
区域 7
区域 9
514
节点
3、开始查询所有区域，以集合提供结果
区域 3
区域 8
区域 11
516
节点
3、开始查询所有区域，以集合提供结果
区域 1
区域 6
区域 10
2. 发布查询
510
节点
3、开始查询所有区域，以集合提供结果
区域 4
区域 5
区域 12
4、持续检索、聚集并筛选结果
1. 请求
6. 请求下一结果集
5. 将结果返回给客户
500
502
应用

图5

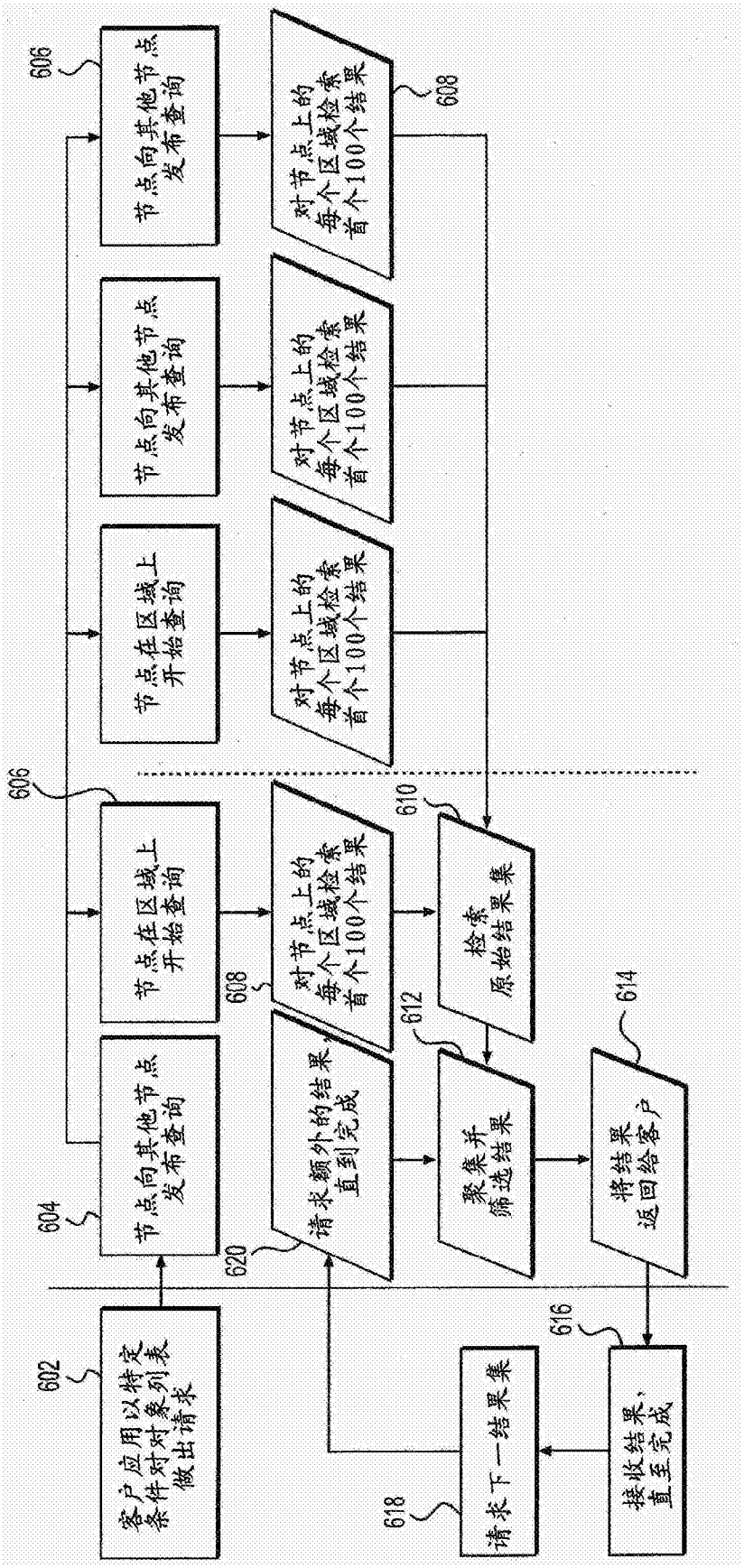

602
客户应用以特定条件对象列表做出请求
604
节点向其他节点发布查询
606
节点在区域上开始查询
608
对节点上的每个区域检索首个100个结果
610
检索原始结果集
612
聚集并筛选结果
614
将结果返回给客户
616
接收结果，直至完成
618
请求下一结果集
620
请求额外的结果直到完成

图6