



(12) 发明专利

(10) 授权公告号 CN 102144373 B

(45) 授权公告日 2013.06.19

(21) 申请号 200980135193.9

(22) 申请日 2009.08.13

(30) 优先权数据

12/203,826 2008.09.03 US

(85) PCT申请进入国家阶段日

2011. 03. 02

(86) PCT 申请的申请数据

PCT/US2009/053770 2009.08.13

(87) PCT申请的公布数据

W02010/027623 EN 2010. 03. 11

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 N·A·艾伦 S·R·巴特雷斯

J · D · 布朗 E · S · 平特 K · 拉曼

J · A · 泰勒

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 潘明嫻

(51) Int. Cl.

H04L 29/08 (2006.01)

(56) 对比文件

US 2007168546 A1, 2007. 07. 19,

CN 101127768 A, 2008. 02. 20,

Carzaniga, A. 等. A routing scheme for content-based networking. 《INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies》. 2004, 第 2 卷 (第 2 期), 918-928.

Carzaniga, A. 等. A routing scheme for content-based networking. 《INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies》. 2004, 第 2 卷 (第 2 期), 918-928.

审查员 李美丽

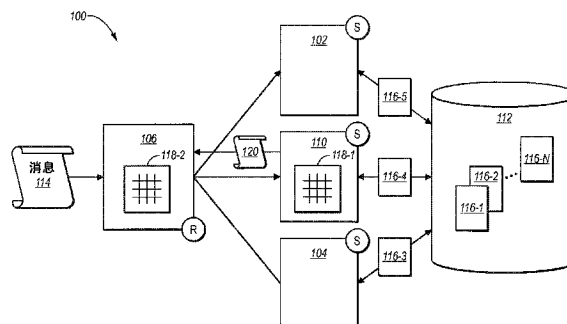
权利要求书3页 说明书12页 附图6页

(54) 发明名称

用于概率动态路由器-服务器网格路由的系统和方法

(57) 摘要

使用不可靠的路由数据来路由消息。一种方法包括从计算机可读通信介质接收消息。计算该消息的各特性性质,以便判断用于该消息的处理的、服务处的服务实例的状态要求。进行获得满足用于处理该消息的状态要求的适当的服务实例的尝试。作出尝试获得满足用于处理该消息的状态要求的适当的服务实例在获得适当的服务实例中是不成功的判断。作为结果,使用路由信息不可靠的本地缓存且在没有各处理节点之间的协调的情况下来重定向该消息。



1. 在计算环境中的处理节点处,一种使用不可靠的路由数据来路由消息的方法,所述方法包括:

从计算机可读通信介质接收消息 (302);

计算所述消息的一个或多个特性性质,以便判断用于所述消息的处理的、服务处的服务实例的状态要求 (304);

尝试获得满足用于处理所述消息的所述状态要求的适当的服务实例 (306),其中当尝试获得满足用于处理所述消息的所述状态要求的适当的服务实例在获得适当的服务实例中是成功的时候,基于所述状态要求处理所述消息 (310);

判断尝试获得满足用于处理所述消息的所述状态要求的适当的服务实例在获得适当的服务实例中是不成功的,且作为结果,使用路由信息不可靠的本地缓存且在没有任何处理节点之间的协调的情况下来重定向所述消息,以便尝试使得所述消息到达具有适当的服务实例或可以成功地获得满足用于所述消息的处理的所述状态要求的适当的服务实例的服务器节点 (312);以及

在计算所述消息的所述一个或多个特性性质之前或之后对所述消息执行独立于所述服务实例的处理。

2. 如权利要求 1 所述的方法,其特征在于,所述独立于所述服务实例的处理是在计算所述消息的所述一个或多个特性性质之前对所述消息执行的。

3. 如权利要求 2 所述的方法,其特征在于,尝试获得满足用于处理所述消息的所述状态要求的适当的服务实例包括尝试从实例协调器获得适当的服务实例,其中所述实例协调器存储网络中未被所述网络中的各处理节点持有的所有服务实例。

4. 如权利要求 3 所述的方法,其特征在于,还包括当所述实例协调器判断在所述实例协调器处或者在所述网络中的所述处理节点中的任何处不存在适当的服务实例时创建新的适当的服务实例的动作。

5. 如权利要求 3 所述的方法,其特征在于,还包括当所述实例协调器判断所述适当的服务实例被所述网络中的所述各处理节点中的一个锁定时向进行请求的处理节点提供关于锁所有者的信息的动作。

6. 如权利要求 3 所述的方法,其特征在于,所述实例协调器将多个特性与各服务实例关联起来,且其中尝试从实例协调器获得适当的服务实例包括发送一个或多个附加特性,对于获得服务实例的这一尝试,不考虑所述附加特性,但是,如果所述处理节点在获得适当的服务实例的尝试中是成功的,则所述附加特性被添加到描述所述服务实例的所述多个特性中。

7. 如权利要求 3 所述的方法,其特征在于,所述实例协调器将多个特性与各服务实例关联起来,且其中尝试从实例协调器获得适当的服务实例包括将所述各特性中的任何一个或多个发送给所述实例协调器,且无需发送与所述服务实例关联的所述各特性中的全部。

8. 在计算环境中的处理节点处,一种使用不可靠的路由信息来路由消息的方法,所述方法包括:

从计算机可读通信介质接收消息 (402);

计算所述消息的一个或多个特性性质,以便判断用于所述消息的处理的、服务处的服务实例的状态要求 (404),并在计算所述消息的所述一个或多个特性性质之前或之后对所

述消息执行独立于所述服务实例的处理；

使用路由信息不可靠的本地缓存且在没有各处理节点之间的协调的情况下，基于所述状态要求确定所述消息的可能路线（406）；以及

根据所述消息的所述可能路线将所述消息定向到服务器节点或另一路由器节点（408）。

9. 如权利要求 8 所述的方法，其特征在于，将所述消息定向到服务器节点包括将所述消息定向到所述处理节点的服务器组件。

10. 如权利要求 9 所述的方法，其特征在于，将所述消息定向到服务器节点包括引用存在于所述处理节点处的服务实例的本地目录。

11. 如权利要求 8 所述的方法，其特征在于，确定所述消息的可能路线包括使用用于路由消息的负载平衡机制。

12. 如权利要求 8 所述的方法，进一步包括：

从所述服务器节点或其他路由器节点接收指示所述消息应被发送到不同的处理节点的重定向消息；以及

更新所述不可靠的本地缓存以便指示带有所述消息的一个或多个特性性质的各消息应被路由到所述不同的处理节点。

13. 如权利要求 12 所述的方法，其特征在于，来自所述服务器节点或其他路由器节点的所述重定向消息含有各特性性质和各处理节点之间的一个或多个附加关联，所述附加特性性质与当前正被处理的所述消息不相关，所述方法还包括根据所述一个或多个附加关联更新所述不可靠的本地缓存的动作。

14. 如权利要求 12 所述的方法，其特征在于，还包括判断如何通过将所述消息定向到所述不同的处理节点来响应于所述重定向消息。

15. 如权利要求 12 所述的方法，其特征在于，还包括判断如何通过将重定向消息提供给从其接收所述消息的系统来响应于所述重定向消息。

16. 如权利要求 12 所述的方法，其特征在于，还包括使用目录服务协议来解析所述重定向消息。

17. 如权利要求 12 所述的方法，其特征在于，还包括使用发现协议来解析所述重定向消息。

18. 如权利要求 8 所述的方法，其特征在于，所述独立于所述服务实例的处理是在计算所述消息的一个或多个特性性质之前对所述消息执行的。

19. 如权利要求 8 所述的方法，其特征在于，所述独立于所述服务实例的处理是在计算所述消息的一个或多个特性性质之后对所述消息执行的。

20. 在一个计算环境中，一个被配置成在网络中路由消息系统，所述系统包括：

被配置成计算消息（114）的一个或多个特性性质以便判断用于所述消息的处理的、服务处的服务实例的状态要求的计算器模块；

实例协调器模块，其中所述实例协调器模块存储网络中未被所述网络中的各处理节点（102, 104, 106, 110）持有的所有服务实例（116）；

服务节点模块，其中所述服务节点模块被配置成尝试从所述实例协调器模块获得满足用于处理所述消息（114）的所述状态要求的适当的服务实例（116），其中当尝试获得满足

用于处理所述消息的所述状态要求的适当的服务实例在获得适当的服务实例中是成功的时,基于所述状态要求处理所述消息,所述服务节点模块还被配置成:当所述实例协调器模块确定在所述实例协调器模块或在所述网络中的任何处理节点不存在适当的服务实例时,创建新的适当的服务实例;以及

路由器模块,其中所述路由器模块被配置成使用路由信息不可靠的本地缓存且在没有任何处理节点之间的协调的情况下定向所述消息,以便尝试使得所述消息到达具有适当的服务实例或者可以成功地获得满足用于所述消息的处理的所述状态要求的适当的服务实例的服务器节点。

21. 在计算环境的处理节点处,一种使用不可靠路由信息来路由消息的方法,所述方法包括:

从计算机可读通信介质接收消息;

计算所述消息的一个或多个特性性质,以便判断用于所述消息的处理的、服务处的服务实例的状态要求;

使用路由信息的不可靠的本地缓存且在没有任何处理节点之间的协调的情况下,基于所述状态要求确定所述消息的可能路线;以及

根据所述消息的可能路线,将所述消息定向到所述处理节点的服务器组件。

用于概率动态路由器 – 服务器网格路由的系统和方法

[0001] 背景

[0002] 背景和相关领域

[0003] 计算机和计算系统已经影响了现代生活的几乎每一个方面。计算机一般地涉及工作、休闲、医疗保健、交通、娱乐、家务管理等等。

[0004] 此外,可以由经由网络连接互连到其他计算系统的计算系统能力来增强计算系统功能。网络连接可以包括但不限于经由有线或无线以太网的连接、蜂窝式连接或甚至通过串行、并行、USB 或其他连接的计算机到计算机连接。各连接允许计算系统访问其他计算系统处的服务并快速地和高效地从其他计算系统接收应用程序数据。

[0005] 可以使用消息处理系统来在连网的计算机之间路由消息。消息处理系统的简单安排包括在从客户端机器直接接收消息的单个机器上运行的服务。随着该服务所要求的资源的量因消息的数量或者处理消息的平均成本的增长,构建可以应对该处理负载的单个机器最终可能会变得不切实际地昂贵。一种常见做法是跨越若干机器而执行服务的多个实例,以使得该处理负载在若干较廉价机器中分配。将该处理负载在若干机器中分配还可以用于消除临界点以便改善总体系统可靠性。

[0006] 系统的并行效率衡量添加更多机器的效率如何。在并行效率是 100% 时,机器数量翻番会使得每一机器的处理负载变为一半。并行效率可以是较小的百分数、0、或甚至是负的。负的并行效率意味着添加机器的开销大于总的处理能力的增益。在各处理机器之间的协调是通常引起差的并行效率的常见原因中的一个。例如,在包括若干服务的拓扑中,在消息被路由到一服务之前,有必要保证该服务具有处理该消息所必要的状态信息。说明性地,在拓扑中的特定的服务处可以加载带有特定电子商务购物车的状态信息的状态包。在该拓扑处可以接收更新购物车(例如添加新的项目、移除项目、完成定单等等)的消息。如果该拓扑中的多个服务可以处理购物车功能,则在该拓扑处可以存在对判断各服务中的哪一个包括所讨论的特定购物车的状态包的需求。

[0007] 存在其中可以独立地处理各消息的服务的若干示例。消息独立性允许将消息发送到任何可用的处理节点。还存在具有在各消息和各处理节点之间的明显关联的服务的示例。明显关联允许以低协调水平将传入的消息路由到适当处理节点。明显关联的示例有,网络连接恰好等于必须一起处理的消息集的边界。然而,存在其中需要关联开销来协调消息和服务的若干示例。

[0008] 在此要求保护的本主题不限于解决任何缺点的实施例或仅在诸如以上所描述的那些环境等的环境中操作的实施例。相反,此背景仅被提供为阐释其中可以实践在此描述的一些实施例的一个示例性技术领域。

[0009] 概述

[0010] 在此描述的一个实施例针对使用不可靠的路由数据来路由消息的方法。该方法可以在计算环境中的处理节点处实践。该方法包括从计算机可读通信介质接收消息。计算该消息的一个或多个特性性质,以便判断用于该消息的处理的在一服务处的服务实例的状态要求。作出尝试获得满足用于处理该消息的状态要求的适当的服务实例。当尝试获得满足

用于处理该消息的状态要求的适当的服务实例在获得适当的服务实例中是成功的时,基于该状态要求处理该消息。该方法包括判断尝试获得满足用于处理该消息的状态要求的适当的服务实例在获得适当的服务实例中是不成功的。结果,使用路由信息不可靠的本地缓存且在没有任何处理节点之间的协调的情况下来重定向该消息,以便尝试使该消息到达具有适当的服务实例或者可以成功地获得满足用于该消息的处理的状态要求的适当的服务实例的服务器节点。在计算所述消息的所述一个或多个特性性质之前或之后对所述消息执行独立于所述服务实例的处理。

[0011] 在计算环境中的处理节点处,另一种使用不可靠的路由信息来路由消息的方法,该方法包括:从计算机可读通信介质接收消息;计算该消息的一个或多个特性性质,以便判断用于该消息的处理的、服务处的服务实例的状态要求,并在计算该消息的一个或多个特性性质之前或之后对该消息执行独立于服务实例的处理;使用路由信息不可靠的本地缓存且在没有任何处理节点之间的协调的情况下,基于状态要求确定该消息的可能路线;以及根据该消息的可能路线将该消息定向到服务器节点或另一路由器节点。

[0012] 在一个计算环境中,一个被配置成在网络中路由消息系统,所述系统包括:被配置成计算消息的一个或多个特性性质以便判断用于消息的处理的、服务处的服务实例的状态要求的计算器模块;实例协调器模块,其中实例协调器模块存储网络中未被所述网络中的各处理节点持有的所有服务实例;服务节点模块,其中服务节点模块被配置成尝试从实例协调器模块获得满足用于处理消息的状态要求的适当的服务实例,其中当尝试获得满足用于处理消息的状态要求的适当的服务实例在获得适当的服务实例中是成功的时,基于所述状态要求处理所述消息,所述服务节点模块还被配置成:当所述实例协调器模块确定在所述实例协调器模块或在所述网络中的任何处理节点不存在适当的服务实例时,创建新的适当的服务实例;以及路由器模块,其中所述路由器模块被配置成使用路由信息不可靠的本地缓存且在没有任何处理节点之间的协调的情况下定向所述消息,以便尝试使得所述消息到达具有适当的服务实例或者可以成功地获得满足用于所述消息的处理的所述状态要求的适当的服务实例的服务器节点。

[0013] 在计算环境的处理节点处,一种使用不可靠路由信息来路由消息的方法,所述方法包括:从计算机可读通信介质接收消息;计算所述消息的一个或多个特性性质,以便判断用于所述消息的处理的、服务处的服务实例的状态要求;使用路由信息的不可靠的本地缓存且在没有任何处理节点之间的协调的情况下,基于所述状态要求确定所述消息的可能路线;以及将所述消息定向到所述处理节点的服务器组件。

[0014] 提供本发明内容以便以简化形式介绍下面在详细描述中进一步描述的概念的选集。本发明内容不旨在标识该所要求保护的本主题的关键特征或必要特征,也不旨在用来帮助确定所要求保护的本主题的范围。

[0015] 另外的特征和优点将在下列的描述中阐明,且部分将可从该描述明显看出,或可以通过在此的教导的实践来学习。可以借助于在所附权利要求中特别指出的装置和组合来实现和获得本发明的特征和优点。本发明的特征将从下列的描述和所附权利要求变得更加充分地明显,或者通过下文所阐明的本发明的实践来学习。

[0016] 附图简述

[0017] 为了描述可以获得上述的和其他的优点和特征的方式,将通过参考附图中所阐释

的具体的实施例来呈现以上简要描述的本主题的更加具体的描述。应理解,这些图仅描绘典型的实施例且因此不被认为是对范围的限制,将通过使用附图借助于另外的特异性和细节来描述和解释各实施例,附图中:

[0018] 图 1 阐释包括路由器节点和用于处理消息的服务节点的网络环境;

[0019] 图 2A 阐释其中可以计算消息的各特性的环境;

[0020] 图 2B 阐释查询优化模块;

[0021] 图 2C 阐释在计算中间结果之前和之后变换数据;

[0022] 图 2D 阐释一些实施例的另外的特性计算细节;

[0023] 图 3 阐释处理消息的方法;以及

[0024] 图 4 阐释使用不可靠的路由数据来路由消息的方法。

[0025] 详细描述

[0026] 一些实施例针对其中在各消息和各处理节点之间存在某种不明显关联的服务。例如,不明显关联的一个示例是基于这样的消息中的内容:该消息的位置和格式取决于应用程序定义的协议和/或该消息的值反映见于先前消息中的内容。

[0027] 一些实施例通过使得各处理节点容忍路由错误并采用乐观路由来降低不明显关联的协调水平。在稍后的处理期间将探测错误的路由决定。响应于被错误地路由的消息,丢弃部分完成的工作且再次以乐观方式来重新路由该消息。另外,可以将信息返回给错误地路由消息的节点,以使得将有可能正确地路由该节点路由的将来的消息。尽管由于遭遇了路由错误,一些消息会要求处理更多工作,但处理消息的平均成本下降了,这是因为可以经常正确做出乐观路由,并且可以减少在各节点之间的协调。

[0028] 正如所述的,一些实施例允许通过使用最佳猜想且无需在各处理节点之间协调来基于该消息的特性性质路由消息。这可以通过维持路由数据不可靠的本地缓存来完成。路由数据不可靠的本地缓存可以包括,例如,在各消息特性、消息元数据特性或服务节点的其他特性之间的关联。此外,该路由数据可以包括基于所接收的重定向消息的更新。例如,错误地接收消息的节点可以提供为该数据标识正确的节点的信息。各实施例还可以包括用于决定如何基于该原始的消息传送机制或特性而处理该重定向消息的功能。进一步,路由可以包括将负载均衡机制包括为用于最佳猜想路由的定义的部分。

[0029] 一些实施例可以包括由实例协调器基于消息的特性性质而实现的原子创建负载原语。例如,作为原子创建负载操作的部分,可以在各特性和服务实例之间提供附加关联。被执行为提供关联的查找可以基于多于一个特性。此外,可以通过使用服务实例的本地目录来实现查找的优化。下面在此进一步包括这些实施例的更多详细描述。

[0030] 各实施例可以包括用于作为失败的创建负载操作的结果而生成重定向消息的功能。在一些实施例中,可以使用目录服务或发现协议来决定重定向地址。此外,用于其他服务实例的路由数据可以作为该重定向消息的部分被包括。

[0031] 现在参见图 1,阐释互连各处理节点的网络 100。可以使用计算硬件和软件来实现各处理节点,如在此稍后定义的,计算硬件和软件包括适当的处理器、存储器设备、计算机存储、在计算机可读介质上实现的计算机可执行指令等等。各处理节点中的一些能够执行应用程序服务的实例。各处理节点中的一些在此被称为服务器节点,且由图 1 中所阐释的示例服务器节点 102 和 104 来被阐释。各处理节点中的一些能够在各处理节点中间路由消

息。这些节点在此被称为路由器节点,且由路由器节点 106 阐释。随着时间的推移且在没有协调的情况下,可以存在对节点数量、在各节点之间的连接或节点的角色(例如获得或失去路由器或服务器能力)的改变。因此,网络的任何描述都表示特定时刻的快照。各处理节点并不必定永久地被划分成路由器和服务器。具体地,处理节点可以起到两者(由路由器-服务器节点 110 阐释,路由器-服务器节点 110 在此可以被称为路由器节点或服务器节点)的作用或不起到两者中的任何一者的作用。

[0032] 包括一个或多个服务实例 116 在内的应用程序在网络 100 上执行,一个或多个服务实例 116 在此一般地由 116 指代,且具体地由 116-X 指代,其中 X 是用于指示特定的服务实例 116 的变量。应用程序可以持续地创建和破坏服务实例 116。因而,当前的服务实例 116 的集合可以不断地改变。服务实例 116 处理消息且随着时间的推移可以积累实例状态。在本示例中,服务实例 116 每次只存在于各服务器节点中的一个。当服务实例 116 不位于任何服务器节点时,其由所有服务器节点都可以访问的实例协调器 112 持有。实例协调器 112 的一个典型的实现是持久地存储服务实例 116 的实例状态的数据库。

[0033] 应用程序可以不断地在各种处理节点处从外部源接收消息 114。处理消息可以要求现有的服务实例 116 的实例状态、要求之前从来没有处理过消息的新的服务实例 116 的白板、或没有实例状态要求。各处理要求可以是应用程序的定義的部分,且因而视应用程序不同而不同,且可能不容易从消息明显看出。一些实施例的一种功能是将消息定向到满足消息的处理要求的服务器节点而不要求在各处理节点之间的过多的协调。在一些实施例中,这可以通过计算特性来完成,特性可以被用来判断消息 114 的处理要求。

[0034] 可以使用例如在题为“Query-Oriented Message Characterization(面向查询的消息表征)”的美国申请第 12/203,790 号中所描述的技术来执行特性计算,该申请与此同时提交,且通过引用以其整体合并于此。图 2A-2D 阐释对于一些实施例如何执行特性计算。现在参见图 2A,阐释一示例。图 2A 阐释查询引擎 202。查询引擎 202 包括处理查询 204 的功能,其中查询 204 是针对诸如包括可从其他源获得的消息数据或非消息数据 210 在内的消息 114 等的各种数据源的查询。具体地,一些实施例可以被实践为使得可以使用对消息内容、元数据或其他信息的查询来指定消息特性。查询引擎可以包括对各种语言 206 的支持。在一个具体的示例中,可以通过将 XPath 表达式用作查询语言来表达查询。

[0035] 通常,诸如 XPath 等的查询语言 206 具有用于访问有限的多种格式的且来自有限的多种源的信息的天生功能,同时非天生地包括用于访问其他信息的功能。例如,XPath 包括用于访问诸如使用 XML 来格式化的消息等的 XML 结构化数据结构中的信息的天生功能,但是可以不包括用于确定来自其他服务的其他信息的功能。尽管如此,可以通过包括扩展 212 来将查询语言扩展成包括用于访问其他服务的功能。在 XPath 查询语言中,扩展被称为选择器。另外,一些实施例可以包括使用对查询语言的扩展来规范化对不同的存储位置的访问的功能。在一些实施例中,规范化对不同的存储位置的访问可以使用双方商定的数据结构。可以通过合并各查询并同时或并行执行它们来执行对计算相同消息的多个特性的优化,如下面更详细地描述。

[0036] 正如图 2A 中所阐释的,信息源可以包括消息 114,消息 114 包括消息数据在内。消息可以包括诸如信封数据、消息正文中的数据、消息的首部中的数据等的信息。如上所述,查询引擎 202 可以包括用于提取消息数据的功能。例如,在一个实施例中,查询引擎可

以包括支持用于从 XML 格式化消息中提取数据的 XPath 查询语言的功能。还可以或者替代地可以使用其他查询语言 206。值得注意的是,查询引擎 202 还可以包括用于调用各种应用程序编程接口 (API) 214 的功能。API 214 包括用于与各信息源交互以便从各源获得数据的经编程的功能。值得注意的是,在一些方面,可以将各语言 206 认作 API。

[0037] 图 2A 进一步阐释非消息数据 210。非消息数据可以是来自多个不同的源中的任何一个的数据,且可以包括关于消息数据的元数据或不直接地呈现该消息数据的其他数据。与消息 114 中的数据相关联的元数据可以包括诸如指示用于发送消息 114 的协议的协议数据、环境数据、本地性质、日时等等的信息。

[0038] 如先前所述,图 2A 阐释查询引擎 202 针对各数据源执行查询 204。基于查询 204,查询引擎 202 生成可以是数据实例值的中间结果 216。中间结果 216 可以包括数据的表或数据的其他形式。例如,中间结果 216 可以包括诸如具体的日时(该日时可以与消息 114 相关联或不相关联)、用于传输消息 114 的具体协议或其他信息等的信息。中间结果通常不是无单位的结果,而是代表某种具体的单位。例如,中间结果 216 可以代表日时单位、协议单位、传输单位或某种其他的具体单位。另外,各中间结果可以是一个或多个不同的数据类型。例如,中间结果可以是整数、浮点、串或其他数据类型。另外,中间结果的集合可以具有不同的数据类型的混合。例如,时间可以被表达为一个或多个整数,而协议可以被表达为一个或多个串。时间整数和协议串两者都可以被包括在中间结果 216 的相同的集合中。

[0039] 中间结果 216 可以被特性计算模块 220 用来创建特性 218。特性 218 可以是例如使用散列算法或基于中间结果 216 计算数值的其他数值方法来计算的数值。例如,在一个实施例中,特性 218 可以是表示全局唯一标识符的无单位的 128 位散列数。可以使用被配置成计算散列或诸如例如数值表示等的其他表示的计算机硬件和软件来实现特性计算模块 220。

[0040] 如下面将更详细地讨论的,可以实践其中在消息特性 218 的计算和消息收发基础设施之间进行协调的一些实施例。具体地,消息收发基础设施可以对它可以潜在地提供给查询 204 的信息编目。例如,消息收发基础设施可以提供关于传输的信息、关于协议的信息等等。消息收发基础设施可以承诺信息在特定时间的可用性。在一些实施例中,该承诺与在该消息收发基础设施处的某种动作的某种功能或性能相关。可以在特性计算模块 220 处的特性计算之前执行对查询 204 的分析,以判断将需要什么信息。可以执行特性计算的优化,以便在经受基于信息可用性的约束的更方便的时间执行用于特性计算的计算。

[0041] 如下面将更详细地讨论的,可以实践其中在查询之前和/或之后执行信息的变换的一些实施例。

[0042] 再次参见图 2A,阐释具有特殊性的更详细的示例。考虑希望计算其特性的消息 114。可以在不考虑现在或过去如何生成消息 114 的前提下预先假定这一消息 114 的存在。因而,这可以是正被发送的、正被接收的或者甚至是与消息操作没有任何联系的被无中生有地创建的消息。可以以各种格式来表示该消息。作为示例,考虑使用简单对象访问协议 (SOAP) 1.2 格式来表示的消息。这样的消息将具有用于消息信封、消息正文和任何数量的消息首部的存储位置。该消息还可以使得在消息信封中不包含的元数据与之相关联,这些元数据例如本地消息性质、传送性质或周围环境中的信息。此元数据可以由在 210 处阐释的非消息数据表示。因而,数据源可以是指该消息内的信息的源或者该消息外的信息的源。

[0043] 为了计算消息 114 的特性,可以利用所有可用的信息源。特性的计算将通常仅要求可用的信息的子集。此子集由包括一个或多个查询 204 的查询规范 205 描述。每一查询包括一标识符和一查询过程。查询过程定义如何从可用的信息中提取值。

[0044] 作为查询规范 205 的示例,在一个实施例中,使用 XPath 表达式来指定查询过程。例如,该消息可以是 SOAP 格式的采购定单,其片段如下所示:

[0045]

```
<s:Envelope>
  <s:Header>
    ...消息中所包括的首部数据...
  </s:Header>
  <s:Body>
    <po:PurchaseOrder purchaseOrderNumber="123">
      ...由应用程序定义的采购定单数据...
    </po:PurchaseOrder>
  </s:Body>
</s:Envelope>
```

[0046] XPath 表达式“/s:Envelope/s:Body/po:PurchaseOrder/@purchaseOrderNumber”指定消息的一部分。在本示例中, XPath 表达式指定名为 Envelope(信封)的元素内的名为 Body(正文)的元素内的名为 PurchaseOrder(采购定单)的元素上的名为 purchaseOrderNumber(采购定单号)的属性的值。在本示例中, XPath 表达式被命名为“PONumber(采购定单号)”,以便创建在标识符 PONumber 和由评估该 XPath 表达式而得到的事实之间的关联,(即表示采购定单单元 123 的号 123)。

[0047] 一旦向查询引擎 202 提供包括查询 204 和诸如消息 114 的必要的信息源和/或对生成非消息源 210 数据的源的访问权在内的查询规范 205,查询引擎 202 就计算在中间结果 216 中阐释的已命名查询结果的表。

[0048] 在所阐释的示例中,根据已命名查询结果 216 来定义特性 218 的计算以从如何访问或组织信息中抽象出计算过程。通过使得新的信息源与现有信息源统一或者通过用新的访问方法扩展查询引擎,可以将新的信息源添加到系统。例如,标准 XPath 语言仅提供对消息数据的访问。可以用新的函数扩展 XPath 语言以便访问非消息数据,正如由扩展 212 所阐释的。

[0049] 在一个实施例中,HTTP Referer(引用物)首部不是消息数据的部分,但是可以使用 XPath 表达式“z:GetProtocolData()/Referer”来指定非消息数据 210 的部分以相似的方式访问。在此情况中,消息内不含有协议数据中的 Referer 性质的值。尽管 SMTP From(SMTP 来自)首部来自不同的信息源,但也可是使用 GetProtocolData(取协议数据)函数来访问它。因而,可以根据开发者的方便来实现将信息分组到相同或不同的访问方法。

[0050] 现在参见图 2B,现在将讨论查询引擎 202 的另外的细节,且尤其是关于优化查询处理。如果针对诸如消息数据 208 和非消息数据 210 等的相同信息源执行多个查询 204,

则当将这些查询合起来而非每次一个时,往往可能更高效地执行查询的集合。在一个实施例中,为做到这一点,查询引擎 202 包括查询优化模块 222,查询优化模块 222 在使用语言 206 来执行经优化的查询规范 224 之前首先将原始的查询规范 205 变换成经优化的查询规范 224(如图 2A 中所阐释的 API 214)。经优化的查询规范 224 在被处理时产生相同的查询结果 216 的表。

[0051] 在一个实施例中,查询引擎 202 的查询优化器 222 将具有公共子表达式的各查询结合在一起,以使得单个公共子表达式仅被评估一次。因而,作用于含有两个查询“s:Envelope/s:Body/PurchaseOrder1”和“/s:Envelope/s:Body/PurchaseOrder2”的查询规范 205 的查询引擎 202 可以仅需要扫描消息 114 的 Envelope 元素和 Body 元素一次就满足这两个查询。

[0052] 现在参见图 2C,阐释查询引擎 202 关于在处理之前和之后变换数据的另外的特征。在所阐释的实施例中,查询引擎 202 与其他组件 228 和 230 组成处理流水线 226 的部分。这些组件 228 和 230 分别作用于对该引擎的输入和输出。在组件 228 处,在由该引擎读取信息源之前,可以将一个或多个变换应用到各信息源,且在组件 230 处,在计算特性 218(见图 2A)之前,可以将一个或多个变换应用到各查询结果。信息源中的每一事实和每一指定的查询结果可以使对其应用单独地精心制作的变换;或者,可以将各变换应用到一组事实或查询结果。

[0053] 应用程序常常具有用于计算特性 218 的优选时间。对应用程序来说,取决于所做出的决定的类型,通常希望尽可能迟地或者尽可能早地计算特性 218。然而,直到所有必要信息可用之前,应用程序不能计算该特性 218,。当发送消息时会发生这种冲突的一个示例。可能希望尽可能早地计算该特性,以使得在观察到对发送消息的任何响应之前该特性是已知的。然而,直到部分地或完全地发送消息之前,计算该特性所必需的信息可能是不可用的。直到很迟之前都不可用的信息的示例是在消息被写到线上时由传送系统指派的消息标识符。

[0054] 现在参见图 2D,阐释解决这些问题的一个实施例的示例。为了推理出冲突,应该知道特性计算将使用什么信息以及何时该信息将可用。在由应用程序 232 发送消息 114 之前,反省消息收发基础设施 234 以便标识此特定配置将生成的各种信息。消息收发基础设施 234 还可以作出关于何时每一事实将可用的一个或多个声明。各声明可以是各事实将在特定的时间或处理阶段可用的承诺。也是在发送消息 114 之前,可以反省查询规范 205(见图 2A)以确定此特定的查询规范 205 将请求的各种信息。

[0055] 图 2D 相对于时间线轴 T 阐释消息收发基础设施 234。时间线轴 T 阐释时间在向下方向上增加。在 $T_{\text{开始}}$,消息 114 从该应用程序 232 发送到消息收发基础设施 234。在一些实施例中,大约在消息 114 被发送的时候,将查询规范 205 中的查询 204 将要求的信息标识符的列表与消息 114 关联起来。值得注意的是,各实施例可以被实现为在消息 114 被发送之前、在消息 114 被发送的时候、或者在一些实施例中是在消息 114 被发送之后关联信息标识符的列表。另外,将消息 114 与调用查询引擎 202 和特性计算模块 220(见图 2A)的回调关联起来。组件 236-1-236-N 可以作用于消息 114。当各组件(在这里概括地指 236 且具体地由 236-X 代表,其中 X 是标识特定组件的数字)作用于消息 114 时,各组件概念性地向在每一所标识的事实变为可用将被查询 204 要求的信息标识符的列表添加校验标记。在一

个实施例中,当有可能执行得到事实的的具体过程时,该事实变得可用。此过程可以简单地返回该事实的预先计算的,或者替代地可以要求执行另外的计算。因而,尽管实审可以在特定的时间为查询引擎 202 可用,但直到查询引擎 202 在稍后的时间请求该值之前,在明确的意义上不可知道该事实的值,如果查询引擎 202 曾经这样选择的话。一旦所有所标识的信息可用,则可以调用回调以便完成特性计算。在所阐释的示例中,图 2D 示出使关于消息 114 的信息对查询引擎 202 可用。在时间 T_1 ,使由组件 236-1 提供的信息对查询引擎 202 可用。在时间 T_2 ,使由组件 236-2 提供的信息对查询引擎 202 可用。在时间 T_N ,使由组件 236-N 提供的信息(这表示可以在该消息收发基础设施 234 中实现任何数量的组件 236)对查询引擎 202 可用。

[0056] 时间线轴 T 包括代表消息 114 被从消息收发基础设施 234 传出(例如通过该将消息传输到通信线)的时间 $T_{\text{传输}}$ 。通信线可以是包括网络线缆或无线传输介质在内的不同数量的介质中的任何一种。计算的完成可以早于或晚于消息被发送发生,这取决于由组件 236 所作的承诺。

[0057] 在一个实施例中,回调的完成被用来解决在发送消息和接收消息之间的竞争。应用程序 232 制止处理可能依赖于先前所发送的消息 114 的特性 218 的任何所接收的消息,直到所有那些特性都已经被计算。

[0058] 下列讨论现在引用可以执行的若干方法和方法动作。应注意,尽管可以以某种次序来讨论这些方法动作,或者在流图中将其阐释为以特定次序发生,但除非特别说明,否则不要求任何特定排序,或者由于一动作依赖于在执行该动作之前完成的另一动作而要求特定排序。

[0059] 现在参见图 3,阐释诸如图 1 中的节点 102、110 和 104 等的服务器节点的操作。图 3 阐释方法 300。方法 300 包括接收消息的动作(动作 302)。可以从诸如网络介质、计算机总线或其他通信介质等的计算机可读通信介质接收该消息(例如消息 114)。

[0060] 方法 300 还包括计算该消息的各特性性质以便确定用于该消息的处理的一服务(例如服务 102、104 或 110 中的一个)处的服务实例(例如服务实例 116 中的一个)的状态要求(动作 304)。在一些实施例中,如以上在上面的图 2A-2D 以及所附描述中所阐释,可以执行计算各特性性质。

[0061] 方法 300 还包括尝试获得满足用于处理消息 114 的状态要求的适当的服务实例(例如服务实例 116)。在 308,判定框阐释取决于该尝试是否成功而执行的不同动作。当尝试获得满足用于处理该消息的状态要求的适当的服务实例 116 在获得适当的服务实例 116 中是成功的时,基于状态要求处理该消息(动作 310)。或者,方法 300 可以包括判断尝试获得满足用于处理该消息的状态要求的适当的服务实例 116 在获得适当的服务实例 116 中是不成功的,且作为结果,使用路由信息不可靠的本地缓存且在没有各处理节点之间的协调的情况下来重定向消息 114(动作 312),以便尝试使得消息 114 到达具有适当的服务实例 116 或者可以成功地获得满足用于该消息的处理的状态要求的适当的服务实例 116 的服务器节点。

[0062] 尽管未在图 3 中阐释,但方法 300 还可以包括对消息 114 执行独立于服务实例 116 的处理。独立于服务实例 116 的处理的示例是执行静态配置的协议和消息变换。

[0063] 现在更详细地阐释方法 300 的细节,在消息到达诸如节点 102、104 或 110 等的服

务器节点时,可以作出该服务器节点满足消息 114 的处理要求的初始假定。在对消息 114 执行某种量的处理之后,该服务器节点达到进一步的处理要求服务实例 116 的点。

[0064] 为了标识适当的服务实例 116,该服务器节点可以使用该消息和本地可用的状态但在没有各处理节点之间的协调的情况下计算特性性质。例如,如果消息 114 到达服务器节点 110,则服务器节点 110 可以使用消息 114 和本地可用的状态但在没有其他处理节点 106、102 和 104 之间的协调的情况下来计算特性性质。特性计算由应用程序定义。特性可以是来自消息传送过程的信息、该消息的部分或者甚至整个消息本身。作为示例,该服务器节点可以标识该消息是具有特定格式的采购定单请求、标识该采购定单格式包括在固定位置处的采购定单标识符、以及提取该采购定单标识符以便形成该特性。

[0065] 然后,该服务器节点咨询实例协调器 112,以便获得具有该特性的适当状态的适当的服务实例 116。此咨询可以包括由实例协调器 112 原子执行的若干动作:

[0066] 具体地,该咨询可以包括判断是否已经存在与这一特性相关联的服务实例 116。例如,实例协调器 112 可以判断是否有适当的服务实例 116 存在于实例协调器 112 处或网络 100 中的任何处理节点处。如果不存在具有该特性的服务实例 116,则创建新的服务实例 116,且将服务实例 116 与该特性关联起来。如果存在具有该特性的服务实例 116,则该咨询可以包括判断服务实例 116 存在于何处。服务实例 116 可以已经存在于进行请求的服务器节点(在本示例中是服务器节点 110)处。如果服务实例 116 存在于实例协调器 112 处,则服务实例 116 被传递到请求服务器节点 110。如果服务实例 116 存在于诸如服务器节点 102 或 104 等的某个其他服务器节点处,则实例协调器 112 可以拒绝该负载请求并将出错消息传输给服务器节点 110。出错消息包括该服务实例 116 当前存在于其处的服务器节点(例如 102 或 104)的标识符。

[0067] 还可以在其中可以对服务实例 116 放置锁的环境中实现各实施例。在这些实施例中的一些中,可以包括用于在实例协调器 112 确定适当的服务实例 116 被网络中的各处理节点中的一个锁定时向进行请求的处理节点提供关于锁所有者的信息的功能。在一个实施例中,实例协调器 112 可以将关于服务实例 116 上的锁的信息提供给进行请求的系统。

[0068] 作为获得服务实例 116 的部分,服务器节点 110 可以提供它希望使之与服务实例 116 关联的一个或多个附加特性。尽管这些附加特性不被用作初始查找的部分,但在执行关联之后,对附加特性中的一个的将来的查找将匹配服务实例 116。如果服务器节点 110 包括附加特性,则作为原子创建或负载过程的部分而执行这些关联,并且在一些实施例中,只有当该创建或负载过程成功获得服务实例 116 时才执行这些关联。具体地,尝试从实例协调器 112 获得适当的服务实例 116 可以包括发送一个或多个附加特性。对于获得服务实例 116 的这种尝试,不考虑这些附加特性,但是,如果该处理节点在获得适当的服务实例 116 的尝试中是成功的,则这些附加特性被添加到描述服务实例 116 的多个特性。作为示例,服务器节点 110 可以使用从采购定单标识符导出的特性来查找服务实例 116,并且提供从发货人跟踪号导出的附加特性。将来,即使不知道采购定单标识符,也可以使用发货人跟踪号来寻找服务实例 116。因而,在一些实施例中,实例协调器将多个特性与各服务实例关联起来。尝试从实例协调器获得适当的服务实例可以包括将各特性中的任何一个或多个发送给实例协调器且无需发送与该服务实例相关联的所有特性。在所阐释的示例中,可以提供采购定单标识符或发货人跟踪号中的任一个且无需提供其他标识符或号。

[0069] 在本发明的一些实施例中,实例协调器 112 接受多于一个的用于查找的特性,并返回匹配各特性中的任何一个的服务实例 116。在各特性不唯一地定义服务实例 116 时的结果可以是基于特性规范的性质支持特定的服务实例 116、基于服务实例 116 的性质支持特定的服务实例 116、或者因被含糊地指定而拒绝该查找操作。

[0070] 在一些实施例中,服务器节点 110 维护其当前拥有的服务实例 116 的本地目录。此本地目录允许该服务器节点制止在不存在要执行的附加关联时咨询实例协调器 112。

[0071] 在一些实施例中,如果服务器节点 110 成功获得服务实例 116,则其将所接收的消息 114 分派给应用程序的剩余部分。否则,服务器节点 110 构造含有由该实例协调器返回的地址的重定向消息并指示该消息的发送方将该消息转发到此新地址。例如,假设实例协调器 112 指示特定的服务实例 116-3 位于服务器节点 104。服务器节点 110 可以构建指示消息 114 应被转发到服务节点 104 的重定向消息 120,该重定向消息 120 被发送到路由器节点 106(路由器节点 106 将消息 114 发送给服务器节点 110)。

[0072] 在一些实施例中,新地址不是物理机器地址。例如,它可以是要在消息的发送方已知的目录中查找的逻辑地址或服务名称。作为另一示例,可以使用动态发现协议而非目录服务来决定该新地址。

[0073] 在一些实施例中,不同于提供路由该消息所严格要求的信息,实例协调器 112 提供用于服务实例 116 的路由信息以便改善总体路由质量。

[0074] 来自服务器节点 110 或其他路由器节点的重定向消息可以含有在各特性性质和各处理节点之间的一个或多个附加关联。附加特性性质并不必与当前正被处理的消息相关。可以根据该一个或多个附加关联更新不可靠的本地缓存。

[0075] 下列描述一般地包括关于路由器节点(例如路由器节点 106 和 110)的功能的进一步细节。另外,可以包括关于处理重定向消息 120 的各细节。

[0076] 图 4 阐释方法 400。方法 400 包括接收一消息(动作 402)。可以从计算机可读通信介质接收该消息。方法 400 还包括计算该消息的各特性性质以便判断对用于该消息的处理的一服务处的服务实例 116 的状态要求(动作 404)。方法 400 包括使用路由信息不可靠的本地缓存且在没有各处理节点之间的协调情况下,判断该消息的可能路线(动作 406)。根据该消息的可能路线,将该消息定向到服务器节点或另一路由器节点。

[0077] 可以执行方法 400 以便执行独立于服务实例 116(见图 1)的处理。独立于服务实例 116 的处理的示例是执行静态配置的协议和消息变换。

[0078] 在诸如节点 110 等的路由器-服务器节点的情况中,该服务器节点可以是同一处理节点的服务器组件。

[0079] 在一些方面中,路由器节点的操作类似于服务器节点的操作。然而,代替向诸如实例协调器 112 等的权威来源咨询服务实例 116,诸如路由器节点 106 等的路由器节点咨询潜在地不可靠的本地路由数据。例如,路由器节点 106 可以包括含有潜在地不可靠的本地路由数据的路由表 118-2。路由器节点 106 可以不时地患有健忘症,这使得它忘记路由数据中的一些或所有;或者,由于错误的猜想而包括了错误的路由数据。不存在以及时方式或甚至确实告知路由器节点 106 各处理节点已经被添加到网络 100 或从网络 100 移除、各处理节点已经改变改变它们的类型或者与其他处理节点的连接性、或者服务实例 116 从一个位置移动到另一位置的期望。路由器节点 106 使用路由表 118-2 中的路由数据来作出正确路线

的最好的猜想判断。最好猜想可以基于先前所路由的消息 114 和来自先前在重定向消息 120 中接收的实例协调器 112 的路由数据等。

[0080] 在一些实施例中,最好猜想包括诸如循环负载平衡等的负载平衡机制,或将各消息定向到被认为未充分利用的各处理节点。

[0081] 当路由器节点 106 接收其先前所路由的消息 114 的重定向消息 120 时,路由器节点 106 决定如何处理路由故障。取决于传送机制、特性和路由数据,路由器节点 106 本身可以转发消息 114。在其他情况中,路由器节点 106 可以将重定向消息 120 返回给原始的发送方。例如,尽管未在图 1 中阐释,但另外的路由器节点可能已经将消息 114 传输到路由器节点 106。路由器节点 106 可以使用表 118-2 中的不可靠的本地路由数据来判断该消息应被发送到路由器节点 110。路由器节点 110 可以判断消息 114 被错误地发送到路由器节点 110。响应于此判断,路由器节点 110 将重定向消息 120 发送给路由器节点 106。在一些实施例中,路由器节点 110 还可以发送指示哪个节点被认为是消息 114 的正确处理节点的信息。可以从本地路由表 118-1 的不可靠信息或者从来自实例协调器 112 的可靠协调信息导出这种指示哪个节点被认为是消息 114 的正确处理节点的信息。响应于该重定向消息,路由器节点 106 可以将消息 114 转发给重定向消息 120 中所指示的处理节点。

[0082] 在一些实施例中,如果可能,路由器节点总是返回重定向消息。从单工消息源接收到的消息被转发,而从双工消息源接收到消息返回重定向消息。

[0083] 本发明的各实施例可以包括或利用包括计算机硬件在内的专用或通用计算机,如下面更详细地讨论。本发明的范围内的实施例还包括用于运载或存储计算机可执行指令和/或数据结构的物理介质和其他计算机可读介质。这样的计算机可读介质可以是可由通用或专用计算机系统访问的任何可获得的介质。存储计算机可执行指令的计算机可读介质是物理存储介质。运载计算机可执行指令的计算机可读介质是传输介质。因而,作为示例而非限制,本发明的各实施例可以包括至少两种截然不同的计算机可读介质:物理存储介质和传输介质。

[0084] 物理存储介质包括 RAM、ROM、EEPROM、CD-ROM 或其他光学盘存储、磁盘存储或其他磁存储设备、或者可以用来存储以计算机可执行指令或数据结构的形式的所期望的程序代码方法且可由通用或专用计算机访问的任何其他介质。

[0085] “网络”被定义为允许在计算机系统和/或模块和/或其他电子设备之间传输电子数据的一个或多个数据链路。当在网络或另一通信连接(硬布线、无线或硬布线或无线的组合)上将数据传递或提供给计算机时,该计算机适当地将该连接看作是传输介质。传输介质可以包括可以用来运载以计算机可执行指令或数据结构的形式的所期望的程序代码方法且可由通用或专用计算机访问的网络和/或数据链路。以上的组合也应被包括在计算机可读介质的范围内。

[0086] 进一步,一旦到达各种计算机系统组件,以计算机可执行指令或数据结构的形式的程序代码方法就可以被自动地从传输介质传递到物理存储介质(或反之亦然)。例如,在网络或数据链路上所接收的计算机可执行指令或数据结构可以被缓冲在网络接口模块(例如,“NIC”)内的 RAM 中,且然后,最终被传递到计算机系统 RAM 和/或被传递到在计算机系统处的较不易失性物理存储介质。因而,应理解,物理存储介质可以被包括在还利用(或甚至主要利用)传输介质的计算机系统组件中。

[0087] 计算机可执行指令包括例如引起通用计算机、专用计算机或专用处理设备执行某些功能或功能组的指令和数据。计算机可执行指令可以是例如二进制数、诸如汇编语言等的中间格式指令或甚至源代码。尽管已经用对结构特征和 / 或方法论动作来说专用的语言描述了本主题,但应理解,在所附权利要求中界定的本主题不必限于以上所描述的所描述的特征或动作。相反,所描述的特征和动作是作为实现权利要求的示例形式而公开的。

[0088] 本领域的技术人员应明白,可以在具有多种类型的计算机系统的网络计算环境中实践本发明,这些多种类型的计算机系统包括个人计算机、台式计算机、膝上型计算机、消息处理器、手持式设备、多处理器系统、基于微处理器的或可编程的消费性电子设备、网络 PC、小型计算机、大型计算机、移动电话、PDA、寻呼机、路由器、交换机等等。还可以在分布式系统环境中实践本发明,分布式系统环境中,通过网络而(被硬布线数据链路、无线数据链路、或硬布线和无线数据链路的组合)连接的本地计算机系统和远程计算机系统两者均执行任务。在分布式系统环境中,程序模块可以位于本地存储器存储设备和远程存储器存储设备两者。

[0089] 可以在不偏离其精神或必要特性的前提下以其他具体形式实现本发明。在所有方面,所描述的实施例都应被认为仅仅是说明性的且不是限制性的。因此,本发明的范围由所附权利要求而非由前述的描述来指示。在权利要求的等效性的意义或范围内的所有改变都应被包含在它们的范围内。

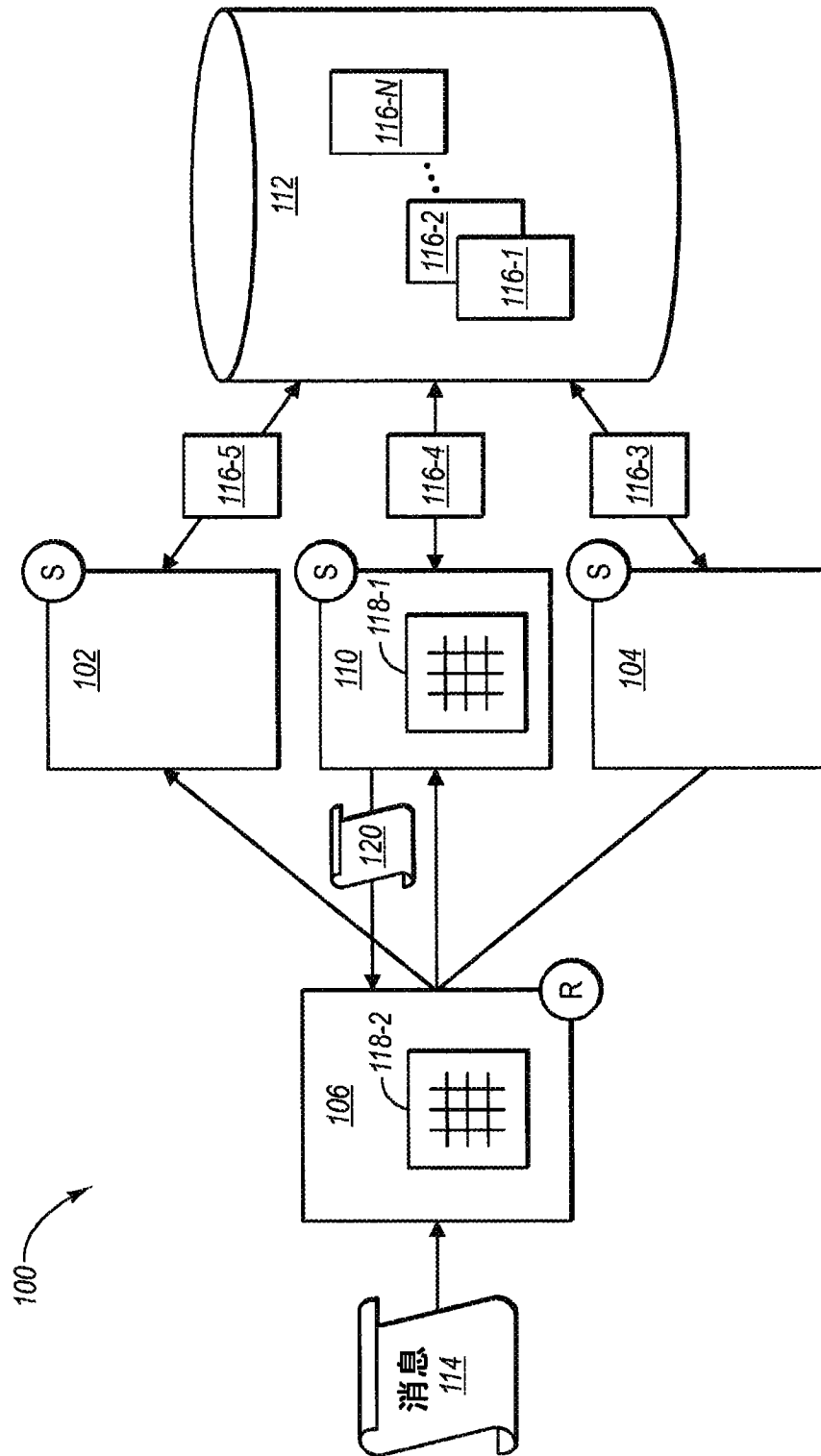


图 1

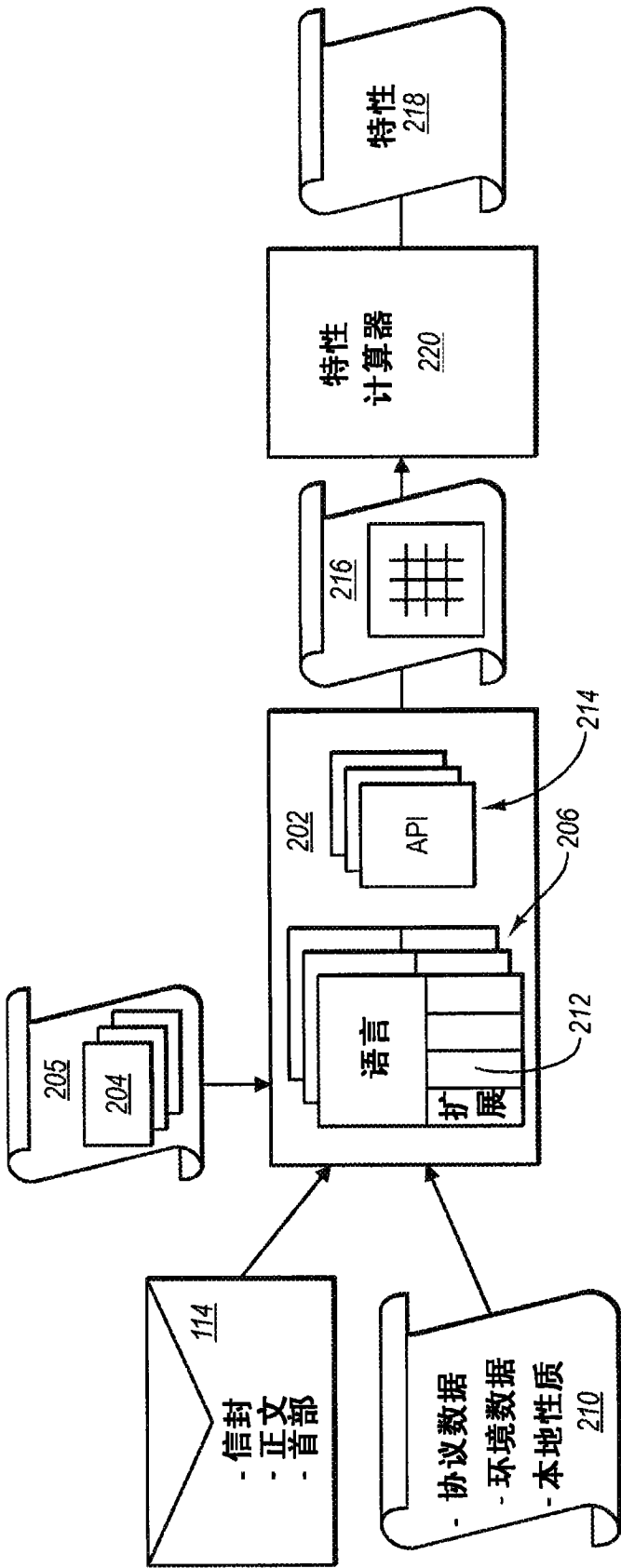


图 2A

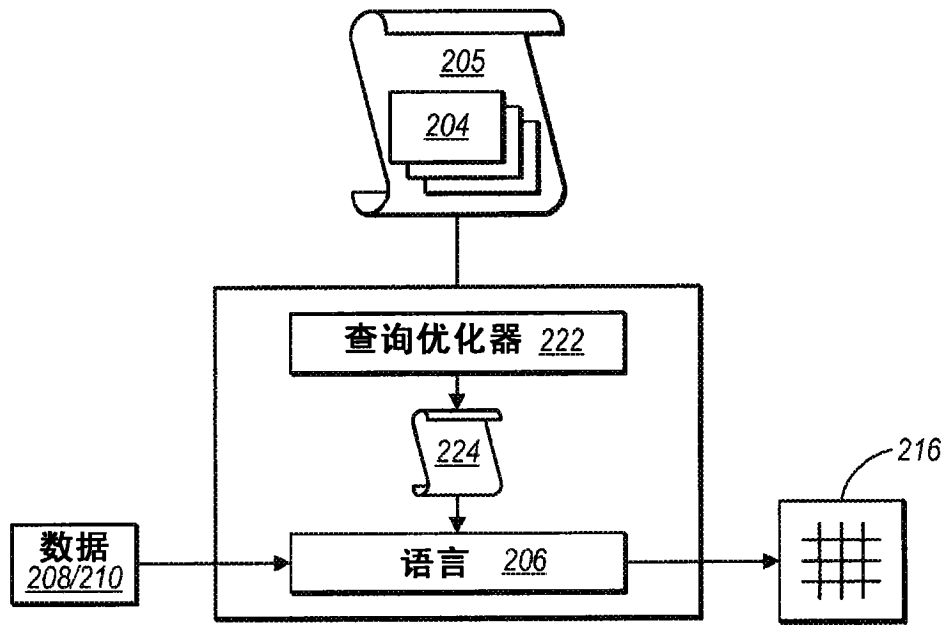


图 2B

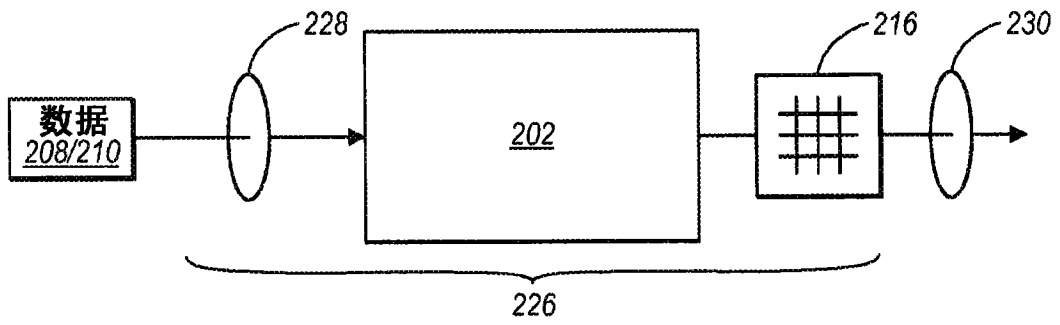


图 2C

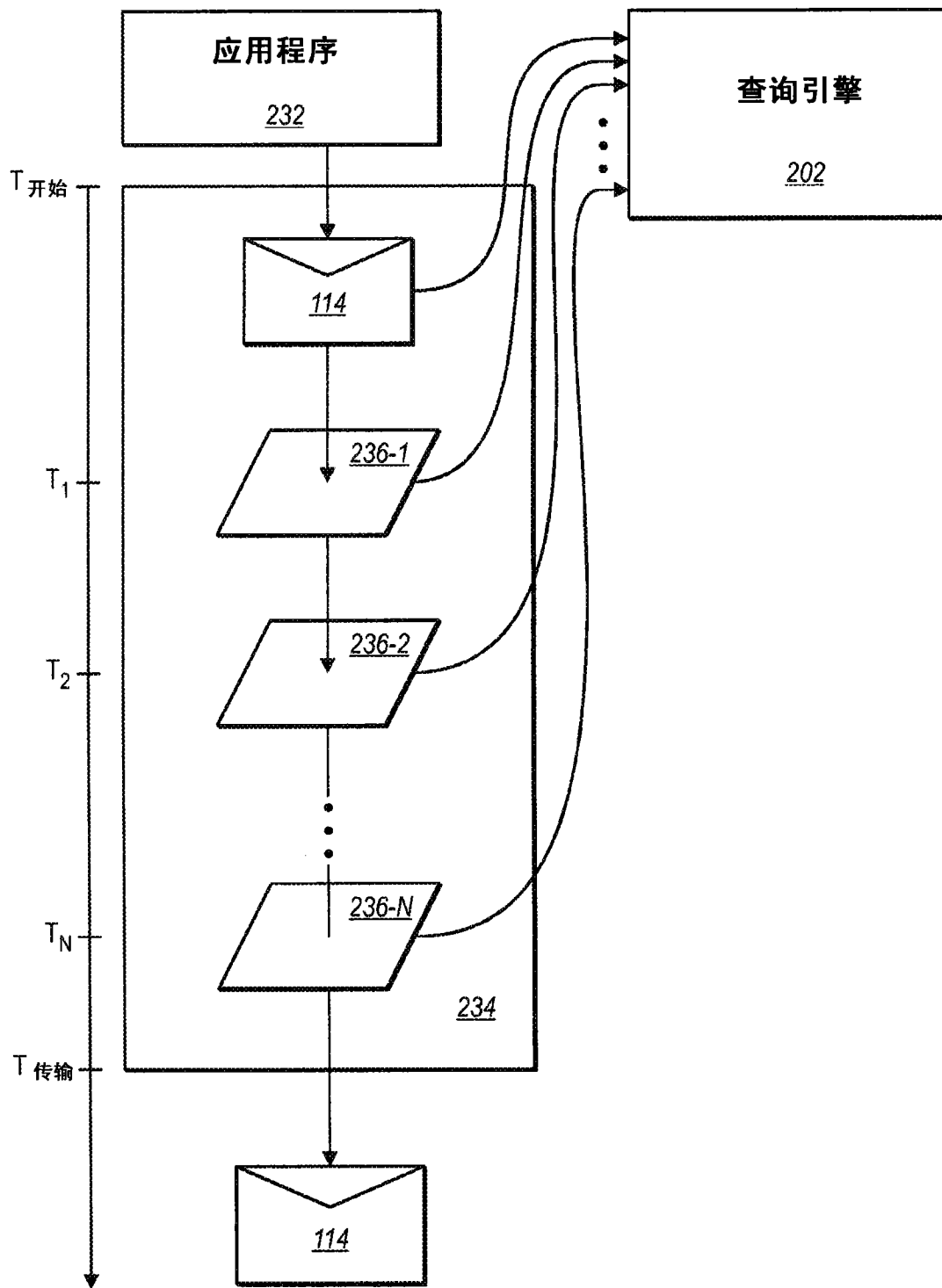


图 2D

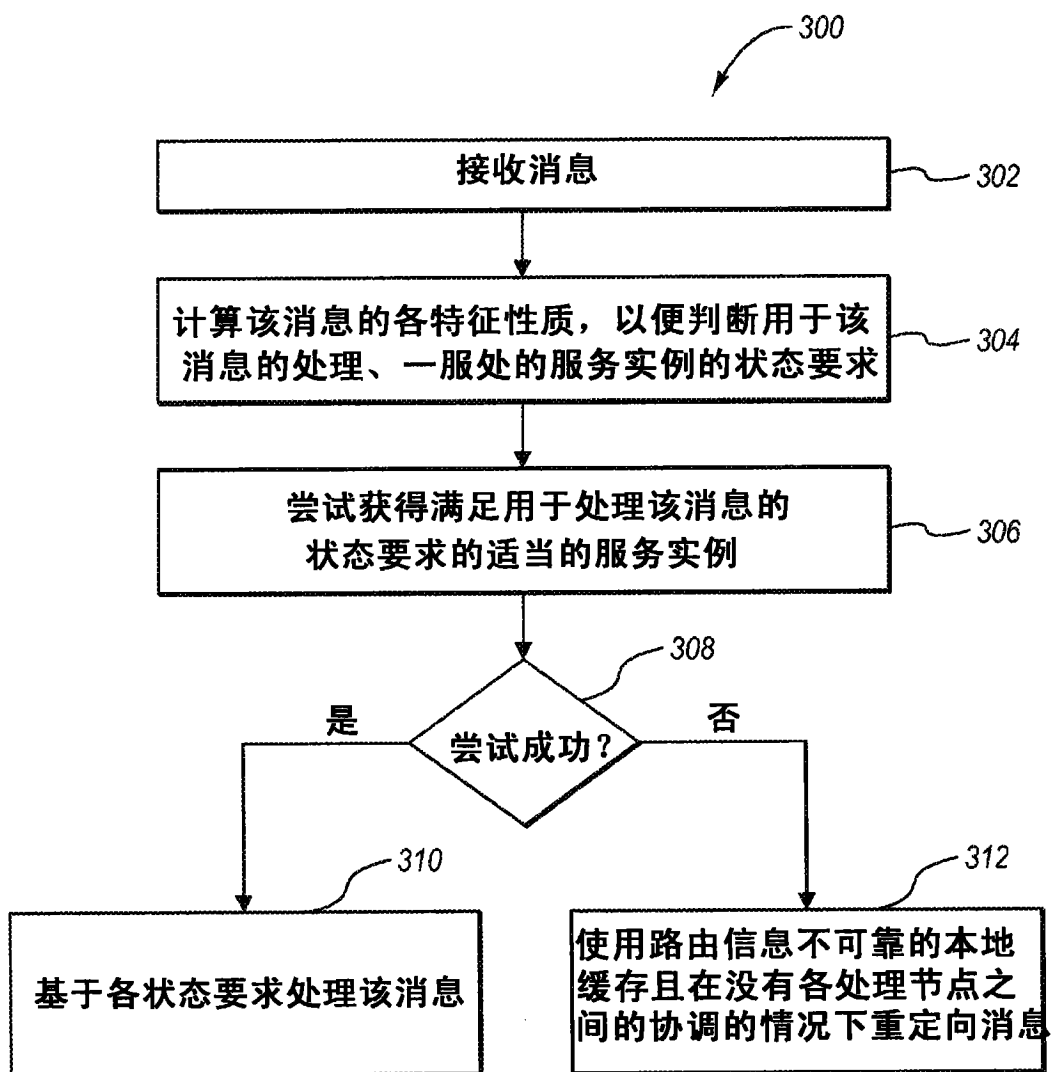


图 3

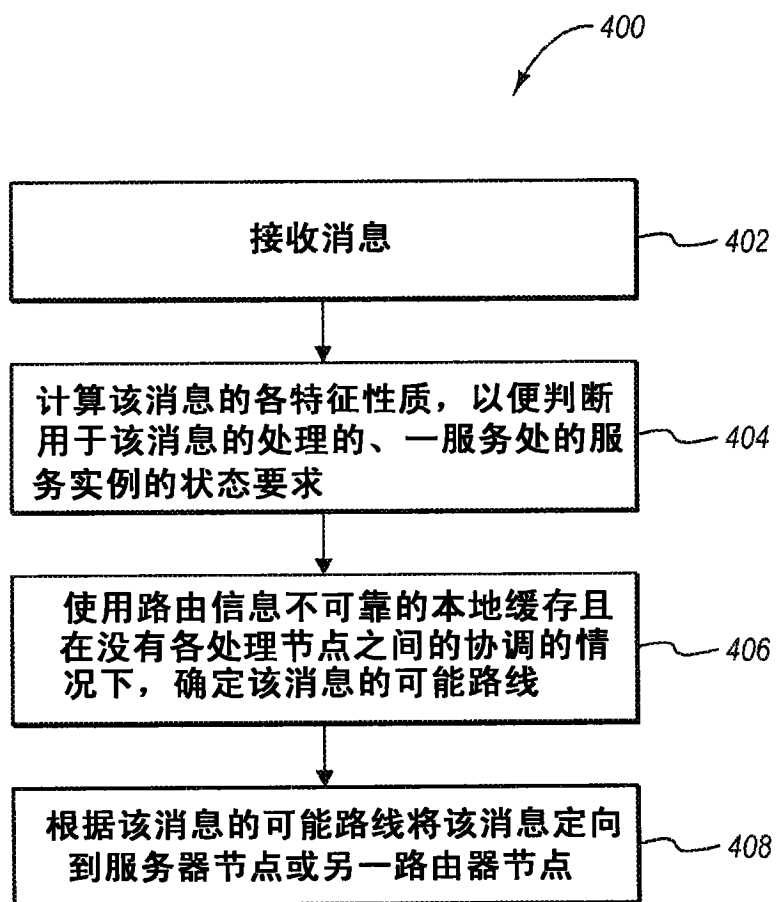


图 4