



(12)发明专利

(10)授权公告号 CN 103238145 B

(45)授权公告日 2016.11.16

(21)申请号 201180058011.X

(22)申请日 2011.11.21

(65)同一申请的已公布的文献号

申请公布号 CN 103238145 A

(43)申请公布日 2013.08.07

(30)优先权数据

61/419,580 2010.12.03 US

13/248,235 2011.09.29 US

(85)PCT国际申请进入国家阶段日

2013.05.31

(86)PCT国际申请的申请数据

PCT/CN2011/082497 2011.11.21

(87)PCT国际申请的公布数据

W02012/071992 EN 2012.06.07

(73)专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华

为总部办公楼

(72)发明人 赛勒斯·库玛 刘振肖

威廉姆·林奇

(51)Int.Cl.

G06F 12/08(2016.01)

G06F 12/1009(2016.01)

G06F 17/30(2006.01)

(56)对比文件

CN 1377543 A,2002.10.30,

US 5778430 A,1998.07.07,

US 2010023726 A1,2010.01.28,

CN 101350010 A,2009.01.21,

CN 1912870 A,2007.02.14,

审查员 李艳军

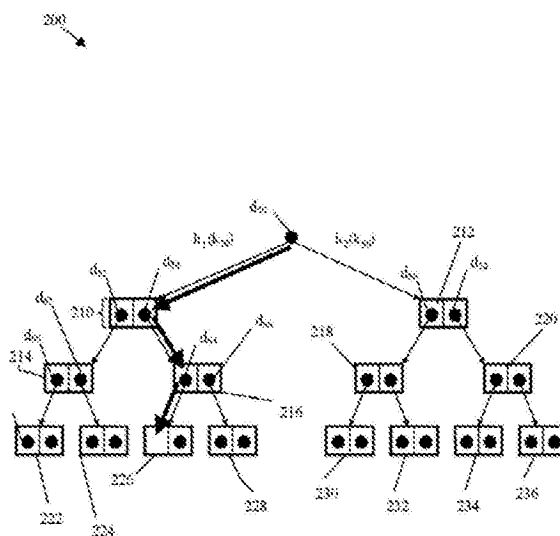
权利要求书3页 说明书10页 附图7页

(54)发明名称

用于网络装备中的高性能、可更新和确定的
哈希表的方法和设备

(57)摘要

一种设备,包括存储装置,所述存储装置包括哈希表,所述哈希表包含多个桶,每个桶能够存储至少一个数据项目,所述设备还包括处理器,所述处理器用于在接收到键后至少应用第一和第二哈希功能来分别产生第一索引和第二索引,所述第一和第二索引在所述哈希表中识别第一和第二潜在桶以用于存储与所述键相关联的新的数据项目,确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目,并且响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,将所述新的数据项目插入所述第一或第二潜在桶中的经确定为具有可用空间的一个桶中。



1. 一种网络设备,其包括:

存储装置,其包括哈希表,所述哈希表包含多个桶,每个桶能够存储至少一个数据项目;以及

处理器,用于:

在接收到键时至少应用第一和第二哈希功能来分别产生第一索引和第二索引,所述第一索引和第二索引在所述哈希表中识别第一和第二潜在桶,用于存储与所述键相关联的新的数据项目,所述第一潜在桶和第二潜在桶分别可容纳多个条目,所述多个条目中的每个条目可包含一个键或一个与所述键相关联的数据项目;

确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目;以及

响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,将所述新的数据项目插入所述经确定为具有可用空间的第一或第二潜在桶中的一个桶中,

其中响应于确定所述第一和第二潜在桶中的至少一者不具有可用空间,所述处理器进一步用于:

确定是否至少一个替代性桶具有可用空间来存储当前存储在所述第一或第二潜在桶中的一个桶中的数据项目,所述至少一个替代性桶是从第一组替代性桶中选择;

响应于确定一个替代性桶具有可用空间,将第一数据项目从所述第一或第二潜在桶移动到所述替代性桶中,以及

将所述新的数据项目插入所述第一数据项目所移走的所述第一或第二潜在桶中。

2. 根据权利要求1所述的网络设备,其中响应于确定所述第一组替代性桶中的每一者都不具有可用空间来存储当前位于所述第一和第二潜在桶中的数据项目,所述处理器进一步经用于确定是否从第二组替代性桶中选择的至少一个替代性桶具有可用空间来存储当前存储在经确定为不具有可用空间的替代性桶中的数据项目。

3. 根据权利要求2所述的网络设备,其中响应于确定来自所述第二组的第一替代性桶具有可用空间,所述处理器进一步用于:

将第一数据项目从经确定为不具有可用空间的替代性桶移动到所述第一替代性桶中;

将当前存储在所述第一或第二潜在桶中的一者中的第二数据项目移动到所述第一数据项目移动离开的所述替代性桶中;以及

将所述新的数据项目插入所述第二数据项目所移走的所述第一或第二潜在桶中。

4. 根据权利要求3所述的网络设备,其中所述处理器用于当确定需要被移走的当前存储的数据项目的数量超过预先确定的阈值后,不将所述新的数据项目插入所述哈希表中。

5. 根据权利要求1所述的网络设备,其中当前存储在所述哈希表的桶中的每个数据项目都具有替代性桶,每个替代性桶分别对应于当将每个数据项目插入所述当前存储的桶中时通过所述第一或第二哈希功能而识别的潜在桶。

6. 一种网络设备实施的方法,用于将数据存储存储在哈希表中并且由至少一个网络组件来实施,所述方法包括:

接收第一键;

通过处理器将第一哈希功能应用到所述第一键以产生第一索引,所述第一索引在所述哈希表中识别第一潜在桶,用于存储与所述第一键相关联的新的数据项目;

通过处理器将第二哈希功能应用到所述第一键以产生第二索引,所述第二索引在所述哈希表中识别第二潜在桶,用于存储所述新的数据项目,所述第一潜在桶和第二潜在桶分别可容纳多个条目,所述多个条目中的每个条目可包含一个键或一个与所述键相关联的数据项目;

通过处理器确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目;以及

响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,将所述新的数据项目插入所述第一或第二潜在桶中的经确定为具有可用空间的一个桶中;响应于确定所述第一和第二潜在桶中的至少一者不具有可用空间:

确定是否至少一个替代性桶具有可用空间来存储当前存储在所述第一或第二潜在桶中的一个桶中的数据项目,所述至少一个替代性桶是从第一组替代性桶中选择;

响应于确定一个替代性桶具有可用空间,将第一数据项目从所述第一或第二潜在桶移动到所述替代性桶中,以及

将所述新的数据项目插入所述第一数据项目移动离开的所述第一或第二潜在桶中。

7.根据权利要求6所述的方法,进一步包括:

响应于确定所述第一组替代性桶中的每一者都不具有可用空间来存储当前位于所述第一和第二潜在桶中的数据项目,确定是否从第二组替代性桶中选择的至少一个替代性桶具有可用空间来存储当前存储在经确定为不具有可用空间的替代性桶中的数据项目。

8.根据权利要求7所述的方法,进一步包括:

响应于确定来自所述第二组的第一替代性桶具有可用空间:

将第一数据项目从经确定为不具有可用空间的替代性桶移动到所述第一替代性桶中;

将当前存储在所述第一或第二潜在桶中的一者中的第二数据项目移动到所述第一数据项目所移走的所述替代性桶中;以及

将所述新的数据项目插入所述第二数据项目所移走的所述第一或第二潜在桶中。

9.根据权利要求8所述的方法,其中如果需要被移走的当前存储的数据项目的数量超过预先确定的阈值,则不将所述新的数据项目插入所述哈希表中。

10.根据权利要求6所述的方法,其中当前存储在所述哈希表的桶中的每个数据项目都具有替代性桶,每个替代性桶分别对应于当将每个数据项目插入所述当前存储的桶中时通过所述第一或第二哈希功能而识别的潜在桶。

11.根据权利要求6所述的方法,其中在所述哈希表中的每个桶都能够存储至少两个数据项目。

12.一种网络设备,其包括:

存储装置,其包括多个桶,每个桶经配置以具有存储至少一个数据项目的容量;以及

处理器,用于根据与新的数据项目相关联的键来识别第一和第二桶来存储所述新的数据项目,所述处理器进一步用于确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目,所述第一潜在桶和第二潜在桶分别可容纳多个条目,所述多个条目中的每个条目可包含一个键或一个与所述键相关联的数据项目,并且所述处理器进一步用于确定当所述第一和第二潜在桶都已充满时,是否多个替代性的桶中的至少一者具有可用空间来存储当前存储在所述第一或第二潜在桶中的一者中的数据项目,其中所述多

个替代性桶是用阶层式树状结构来布置的,其中响应于确定一个替代性桶具有可用空间,所述处理器进一步用于将第一数据项目从所述第一或第二潜在桶移动到所述替代性桶中,并将所述新的数据项目插入所述第一数据项目移动离开的所述第一或第二潜在桶中。

13.根据权利要求12所述的网络设备,其中响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,所述处理器进一步用于将所述新的数据项目插入所述第一或第二潜在桶中的经确定为具有可用空间的一个桶中。

14.根据权利要求12所述的网络设备,其中所述处理器用于通过如下方式根据与所述新的数据项目相关联的键来识别第一和第二桶来存储新的数据项目:将至少第一和第二哈希功能应用于接收到的键以分别产生第一索引和第二索引,所述第一和第二索引识别哈希表中的第一和第二潜在桶以用于存储与所述键相关联的新的数据项目。

15.根据权利要求14所述的网络设备,其中所述处理器用于当确定需要被移走的当前存储的数据项目的数量超过预先确定的阈值后,不将所述新的数据项目插入所述哈希表中。

16.根据权利要求14所述的网络设备,其中当前存储在所述哈希表的桶中的每个数据项目都具有替代性桶,每个替代性桶分别对应于当将每个数据项目插入所述当前存储的桶中时通过所述第一或第二哈希功能而识别的潜在桶。

用于网络装备中的高性能、可更新和确定的哈希表的方法和 设备

[0001] 相关申请案的交叉参考

[0002] 本发明要求2011年9月29日递交的发明名称为“用于网络装备中的高性能、可更新和确定的哈希表的方法和设备(Method and Apparatus for High Performance, Updatable, and Deterministic Hash Table for Network Equipment)”的第13/248235号美国临时专利申请案的在先申请优先权,该申请案要求2010年12月3日由赛尔西古玛(Sailesh Kumar)等人递交的发明名称为“用于网络装备中的高性能、可更新和确定的哈希表的方法和设备(Method and Apparatus for High Performance, Updatable, and Deterministic Hash Table for Network Equipment)”的第61/419,580号美国临时专利申请案的在先申请优先权,上述两者在先申请的内容都以引入的方式,如重现一样完全并入本文本中。

[0003] 关于由联邦政府赞助的

[0004] 研究或开发的声明

[0005] 不适用。

[0006] 参考缩微胶片附录

[0007] 不适用。

技术领域

[0008] 本发明涉及网络装备技术领域,尤其涉及高性能、可更新和确定哈希表的方法和设备。

背景技术

[0009] 当前的互联网协议(IP)网络可包括多个节点,所述多个节点包含网络核心处的多个路由器以及网络边缘处的多个主机。路由器共同地连接主机之间的通信信道。节点被分配有网络范围内唯一的IP地址,以便将适当并且有效的业务转发到目标节点。路由器可根据包中携带的IP地址在IP网络中对所述包进行路由。可由路由器基于可在每个包中指出的<源地址,目标地址>对而将包转发到适当的目标。

[0010] 许多高速包交换网络使用查找操作来将一个值或一组值与一个表的条目进行匹配。例如,IP路由操作通常应用最长的前缀匹配(LPM)比较技术来执行转发表查找。因此,路由器可利用例如转发表等数据结构来对包执行一个或多个查找操作。在接收到包后,路由器可参考转发表,并且将接收到的包的地址(例如,IP地址前缀)作为键来使用。例如,路由器可使用转发表来查找包的目标地址并且选择与那个地址相关联的输出端口。如果未找到匹配,那么包可能溢出(例如,在各种端口上转发)或被丢弃。

[0011] 查找和更新操作通常与哈希表一起使用,在哈希表中,数据值是根据可适用于所述数据值的对应键的哈希功能的结果来编入索引。哈希功能的应用产生了到哈希表的一个位置或“桶”的索引。哈希表的每个桶可包含固定数目的条目用于存储数据值,所述数据值

具有“哈希”到同一索引的键。键和数据值可以存储在桶中并且从桶中检索,所述桶与通过将哈希功能应用到键而产生的索引是对应的。

[0012] 使用如哈希表等数据结构的一个好处是定位与一个键相关联的值的查找时间大体上是恒定的,无论哈希表中的数据值的数量是多少。因此,即使在存储在桶中的数据值的数量变得相当大时,哈希表实施方案也可保持有效。另一方面,在使用具有查找表的哈希功能时通常产生的一个问题是冲突几乎不可避免。当哈希功能对两个或两个以上不同的键产生相同的索引时,冲突可发生,在这种情况下,尝试将一个数据项目插入与现有项目值相同的位置可能会受阻。因此,不同于完全关联的数据结构,例如内容可寻址存储器(CAM),传统的哈希表不可保证确定的查找。

[0013] 因此,可能需要一种或多种技术来解决冲突,使得从每个键(例如,包地址)中可产生唯一的索引。在这种情况下使用的一种技术涉及将桶的每个位置或槽位链接到一个列表,所述列表含有哈希到同一位置的键-值对。另一种技术包含增加每个桶的大小以容纳多个数据值。然而,已知的是这些技术在存储器带宽需求和处理时间方面是昂贵的。另外,这些技术中的许多技术都利用特殊电路,已知的是这种特殊电路相对昂贵并且复杂。

发明内容

[0014] 在一个实施例中,本发明包含一种设备,所述设备包括存储装置,所述存储装置包括哈希表,所述哈希表包含多个桶,每个桶能够存储至少一个数据项目,所述设备还包括处理器,所述处理器经配置以在接收到键后至少应用第一和第二哈希功能来分别产生第一索引和第二索引,所述第一和第二索引在哈希表中识别第一和第二潜在桶以用于存储与所述键相关联的新的数据项目,确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储新的数据项目,并且响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,将新的数据项目插入所述第一或第二潜在桶中的经确定为具有可用空间的一个桶中。

[0015] 在另一个实施例中,本发明包含一种网络设备实施的方法,用于将数据存储在哈希表中并且由至少一个网络组件来实施,所述方法包括接收第一键,用处理器来将第一哈希功能应用到所述第一键以产生第一索引,所述第一索引在哈希表中识别第一潜在桶用于存储与第一键相关联的新的数据项目,用处理器来将第二哈希功能应用到所述第一键以产生第二索引,所述第二索引在哈希表中识别第二潜在桶以用于存储所述新的数据项目,用处理器来确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目,并且响应于确定所述第一和第二潜在桶中的至少一者具有可用空间,将新的数据项目插入所述第一或第二潜在桶中的经确定为具有可用空间的一个桶中。

[0016] 在第三方面,本发明包含一种设备,所述设备包括存储装置,所述存储装置包括多个桶,每个桶经配置以具有存储至少一个数据项目的容量,所述设备还包括处理器,所述处理器经配置以根据与新的数据项目相关联的键来识别第一和第二桶来存储所述新的数据项目,所述处理器进一步经配置以确定是否所述第一和第二潜在桶中的至少一者具有可用空间来存储所述新的数据项目,并且所述处理器进一步经配置以确定当所述第一和第二潜在桶都已充满时,是否多个替代性的桶中的至少一者具有可用空间来存储当前存储在所述第一或第二潜在桶中的一者中的数据项目,其中所述多个替代性的桶是用阶层式树状结构来布置的。

[0017] 从结合附图和所附权利要求书进行的以下详细描述将更清楚地理解这些和其它特征。

附图说明

[0018] 为了更完整地理解本发明,现在参考以下结合附图和详细描述进行的简要描述,其中相同参考标号表示相同部分。

[0019] 图1A和1B是根据本发明的一个实施例的哈希表的示意图。

[0020] 图2是根据本发明的一个实施例的决策树的示意图。

[0021] 图3是一个波形图,图示了在不同负载下在插入操作期间的溢出概率。

[0022] 图4是一个波形图,图示了在不同负载下每个插入操作所检查的桶的数量。

[0023] 图5是根据本发明的一个实施例的用于将数据项目插入哈希表中的方法的流程图。

[0024] 图6是网络单元的一个实施例的示意图。

[0025] 图7是通用计算机系统的一个实施例的示意图。

具体实施方式

[0026] 最初应理解,尽管下文提供一个或一个以上实施例的说明性实施方案,但可使用任何数目的技术,不管是当前已知还是现有的,来实施所揭示的系统 and/或方法。本发明决不应限于下文所说明的所述说明性实施方案、图式和技术,包含本文所说明并描述的示范性设计和实施方案,而是可在所附权利要求书的范围以及其均等物的完整范围内修改。

[0027] 表查找和更新操作可能都需要实质上确定的性能保证。然而,在许多网络系统中,查找操作趋向于需要相对更高的性能。为均衡这些需求,本发明的各方面提议在插入操作期间在哈希表内执行有限次的移动,使得哈希表可以得到重新平衡,同时在查找操作期间保持相对高的吞吐量。由于移动的次数有限,因此更新操作可保持实质上确定的特性,由此确保确定的哈希表性能。尽管限制移动次数可能潜在地降低插入速率,但这个有限的次数可经选择以确保为给定的网络协议正确地操作而保持足够的插入速率。

[0028] 此外,本发明的各方面可利用一个或多个多桶散列法技术来减少冲突。一种此类技术称为d-左散列法(d-left hashing),其中哈希表可分割成d个部分,使得新的数据项目可插入到最少负载的桶中,或在负载相同的情况下插入到最左的桶中。额外地或替代性地,一些方面包含将多个数据项目存储在每个桶中以减少冲突,并且在路由器接口线路板中利用常见存储器技术的突发存取特性。总之,本发明的各实施例在如上所描述的各种限制(例如,存储器带宽、存取大小、存储容量、查找速率,更新速率等)之间提供平衡,以获得满足高端网络装备中所需的性能水平的相对经济的和有效的设计。

[0029] 本文揭示的是实施实质上确定的哈希表的一种系统和方法。在插入操作期间,至少将第一哈希功能和第二哈希功能施加到一个键,以分别产生第一和第二索引值。用第一和第二索引值来识别用于存储与所述键相关联的新数据项目的第一潜在桶和第二潜在桶。如果第一和第二桶都没有可用空间来存储新的数据项目,那么可进行搜索以确定是否有替代性的桶可用于存储当前位于第一或第二潜在桶内的数据项目。如果发现有桶可用于位于第一桶或第二桶中的数据项目,那么可将所述数据项目移动到此替代性的桶,由此腾出空

间以将新的数据项目插入上述数据项目移动离开的桶中。

[0030] 现参考图1A,图示了根据本发明一个实施例的哈希表100并且表示为m行和c列的阵列。如本文所使用,术语“哈希表”可指存储数据并且将键与数据值或相应的表位置进行关联的任何合适的数据结构、阵列、集合、高速缓冲存储器或合适的存储器装置。所属领域的一般技术人员将理解,哈希表100可由例如但不限于路由器、交换机、服务器等一个或多个合适的网络组件和基础设施元件来使用。此外,哈希表100可包括任何合适的存储媒体105和/或通过任何合适的存储媒体105来实施,所述任何合适的存储媒体105例如但不限于随机存取存储器(RAM)、动态随机存取存储器(DRAM)、同步动态随机存取存储器(SDRAM)、减少时延的DRAM(RLDRAM)、嵌入式DRAM(eDRAM)、临时内容可寻址存储器(TCAM)、相变存储器(PCM)、固态驱动器(SSD)、硬盘驱动器(HDD)、相变存储器(PCM)等。

[0031] 哈希表100包括用于存储数据的多个桶110、112、114、116、118、120、122和124。如本文所使用,术语“数据”希望是最宽广的意义并且可以指所有类型的包或任何其它的信息或数据单位,包含但不限于固定长度单元和可变长度包,其中每一者都可以或不可以分为更小的包或单元。此外,术语“包”可指包本身或它的一部分,例如但不限于所有的或一部分的包头部、数据结构值、指针或索引,或者对包或与包相关联的信息的任何其它直接或间接的识别。此外,包可含有一种或多种类型的信息,包含但不限于音频、语音和视频信息。

[0032] 在一些实施例中,桶110、112、114、116、118、120、122和124可容纳多个槽位或条目,其中每个条目可包含一个键或一个与所述键相关联的数据项目。例如,每个条目可以是固定长度(例如32比特)的指针,指向含有存储的项目的主文件中的特定位置。在图1A中,哈希表100的行对应于桶的数量(即,m个桶),且列对应于每个桶中的条目的数量(即,每个桶中有c个条目)。因此,图1的哈希表100将在本文中描述为包括八个桶110、112、114、116、118、120、122和124,每个桶含有两个条目。可是,应理解在其它实施方案中,哈希表100可包括更多或更少的桶,并且每个桶可含有任何合适数量的条目。类似地,尽管桶110、112、114、116、118、120、122和124将描述为含有相等数量的条目,但是在一些实施方案中,一个或多个桶可比哈希表100中的其它桶含有更多或更少的条目。

[0033] 在一个实施例中,使用至少两个哈希功能来产生对应于相应的桶110、112、114、116、118、120、122和124的索引。如本文所使用,术语“哈希功能”可指使用例如键等输入值并且返回或识别与哈希表100中的位置(例如,它所处的桶或条目)相关联的输出值或索引的任何合适的功能或映射算法。应理解尽管本文所描述的实施例可能着重于使用特定数量的哈希功能,但其它实施例可使用任何合适数量的哈希功能。

[0034] 根据一个方面,第一哈希功能 h_1 和第二哈希功能 h_2 接收并且处理新到达的键k以产生相应的输出,从而结合对与所述键相关联的数据的存取和/或存储(例如,在插入和/或更新操作期间)来识别潜在桶110、112、114、116、118、120、122和124。出于方便,在本文中这些数据指的是与所述键相关联的数据项目。然而,熟练的技术人员将很容易了解,所述数据可额外地或替代性地包含所述键本身、与所述键相关联的数据项目的全部或一部分、识别数据项目的全部或一部分的位置信息或指针等。

[0035] 如图1A中所示,第一哈希功能 h_1 和第二哈希功能 h_2 都可以应用到第一键 k_1 ,以产生输出,从而分别识别第一桶114和第二潜在桶120。在一个实施例中,每个所识别的桶114和120都可以作为预期候选者,用于存储与第一键 k_1 相关联的数据项目 d_1 。因此,可以检查桶

114和120中的任一者或两者以确定数据项目 d_1 是否可以插入其中。在这种情况下,尽管桶114和120都分别含有现有的数据项目 d_{14} 和 d_{20} ,但是桶114和120两者都含有可用于存储额外的数据项目的条目。如果存在于相应的桶114和120中的数据项目 d_{14} 和 d_{20} 具有相同的大小并且每个桶114和120具有相同的容量,那么可选择这两个桶114或120中的任一者来插入与第一键 k_1 相关联的数据项目 d_1 。

[0036] 所属领域的一般技术人员将理解,可用各种方法来选择用于存储数据项目 d_1 的桶114或120。简要地说,桶的选择可以是随机地、按顺序地和/或根据指定的策略或优先权(例如,根据最近使用最多或最少的)来完成。在一个方面,例如,可根据可用的容量数量或占用水平来选择桶。额外地或替代性地,可根据将插入到桶中的数据类型来选择桶。然而,出于描述本实例的目的,可假定桶120是任意选择的,用于插入与第一键 k_1 相关联的数据项目 d_1 。

[0037] 现参考图1B,图示了依据第一哈希功能 h_1 和第二哈希功能 h_2 而应用第二键 k_2 。此处,第一哈希功能 h_1 和第二哈希功能 h_2 返回相应的输出,将第一桶112和第二桶120识别为潜在候选者,用于存储与第二键 k_2 相关联的数据项目 d_2 。如图1B中所示,第一识别的桶112含有两个数据项目 d_{11} 和 d_{12} ,并且第二识别的桶120含有两个数据项目 d_1 和 d_{20} 。因此,与关于图1A所描述的情况不同,识别的桶112和120两者都是满的。由于由哈希功能 h_1 和 h_2 识别的桶112和120没有可用空间来存储与第二键 k_2 相关联的数据项目 d_2 ,因此可执行搜索以确定是否可采用替代性的桶来腾出空间以在哈希表100内容纳数据项目 d_2 。

[0038] 例如,可搜索一个或多个其它桶(即,未由哈希功能 h_1 和 h_2 识别的桶)以确定是否现有的数据项目 d_1 、 d_{11} 、 d_{12} 或 d_{20} 可从识别的桶112或120中的一者移动到替代性的桶中。根据一个方面,对替代性桶的搜索可首先对存储在由第一哈希功能 h_1 和第二哈希功能 h_2 识别的桶中的数据项目来执行。这样,所述搜索可开始于对在将现有的数据项目 d_1 、 d_{11} 、 d_{12} 或 d_{20} 插入到识别为用于插入 d_2 的候选者的桶112和120中的过程期间识别的潜在桶进行检查。所属领域的熟练技术人员将了解,对桶进行检查的顺序可根据任何合适的准则。例如,所述顺序可以是预先确定或随机的、根据桶位置、根据时间(例如,最近添加或移动到桶中的数据项目)。在一些方面,可以用深度第一顺序对桶进行搜索,而在其它方面,可以用宽度第一顺序对桶进行搜索。为了清晰,本实例将假定用宽度第一顺序以数据项目 d_1 开始对替代性桶进行搜索。

[0039] 如上所讨论,第一哈希功能 h_1 和第二哈希功能 h_2 产生输出值,将两个桶114和120识别为用于插入数据项目 d_1 的可能候选者。由于选择桶120来存储数据项目 d_1 ,因此未选择的桶114在这种情况下就可定义为待检查的“替代性的桶”。因此,尽管某个桶(例如桶114)起初可能不被选择来插入数据项目(例如 d_1),但哈希表100可维持输出值的日志,所述输出值对应于起初被识别但不用于存储相应的数据项目的替代性的桶。或者,当一个数据项目存储于在插入的时候选择的桶中时,对应于替代性的桶的输出值可随着每个相应的数据项目一起进行存储。

[0040] 如图1B中所示,用于数据项目 d_1 的替代性的桶114具有一个可用于容纳额外的数据项目的条目。因此,数据项目 d_1 可从桶120移动到它的替代性的桶114中,由此有空间用于将数据项目 d_2 插入到桶120中。当然,如果已使用三个或三个以上哈希功能来识别用于插入数据项目 d_1 的桶,那么每个被识别而未选择的桶可以被界定为用于数据项目 d_1 的替代性的桶。如果多个替代性的桶被检查并且确定为在这些情况下是可用的,那么可用的替代性的

桶可经比较以确定(例如,根据占用水平)选择将考虑中的数据项目移动到哪个替代性的桶中。

[0041] 在一些方面,用于一个给定的数据项目和/或至少两个数据项目的多个替代性的桶可经检查以确定是否可以将现有的数据项目移走来腾出空间给新到达的数据项目。或者,当识别出具有可用于给定的数据项目的空间的第一替代性的桶后,检查可终止。在其它方面,可对当前存储在被识别为用于存储新到达的数据项目的潜在候选者的桶中的每个数据项目的至少一个替代性的桶进行检查。如果一个以上替代性的桶被确定为含有可用于容纳识别的桶中的数据项目的条目,那么每个可用的替代性的桶可经比较以确定将使用哪个桶。如上所论述,桶的选择可以是随机的、按顺序的、按照策略的,等。

[0042] 在一个实例中,可假定哈希功能 h_1 和 h_2 在数据项目 d_{11} 的插入期间产生了输出值,将桶112和116识别为潜在的桶。由于数据项目 d_{11} 当前存储在桶112中,因此桶116就被界定为替代性的桶,它可经检查以确定数据项目 d_{11} 是否可移走以腾出空间来将数据项目 d_2 插入桶112中。如图2B所示,桶116是空的,而桶114含有数据项目 d_{14} 。因此(例如,根据布置哈希表100内的数据项目的算法或优先权),在一些情况下,为了腾出空间以插入数据项目 d_2 ,可将数据项目 d_{11} 从桶112移到桶116中,而不是将数据项目 d_1 从桶120移到桶114中。在这种情况下,数据项目 d_2 将被插入到桶112中,而不是插入到桶120中。例如,如果桶的选择根据的是最低占用水平,那么这种结果将是优选的。

[0043] 在另一实例中,可假定四个替代性的桶都没有空间可用于对位于经识别为用于数据项目 d_2 的桶112和120中的对应的数据项目进行存储。在这种情况下,多达八个替代性的桶可类似地经检查以确定是否某个桶具有可用空间来对位于已满的四个替代性桶中的一个桶中的数据项目进行存储。根据四个数据项目 d_1 、 d_{11} 、 d_{12} 和 d_{20} 都不能移到它们相应的替代性桶中的假定,可进行搜索以识别潜在的替代性桶,用于位于已满的四个替代性桶中的一个桶中的数据项目。

[0044] 根据一个方面,当识别出一个没有满(或能存储数据项目)的替代性桶时,可用检查的顺序反复地移走数据项目,所述检查的顺序是以替代性桶没有满的数据项目开始,并且以将新到达的数据插入两个识别的桶中的一个桶中结束。将参考图2进一步描述这方面,图2图示了用于将数据项目布置在哈希表中的决策树200。

[0045] 在插入操作期间,如图1A和1B所示,可使用决策树200在哈希表中搜索替代性的桶。在一个实施例,决策树200可采取二元树的形式,其根基在识别为用于插入数据项目的潜在候选者的每个桶处。在使用两个哈希功能(例如 h_1 和 h_2)来将数据项目插入桶中的实施方案中,已满的每个桶(即,含有 c 个数据项目)将具有它的 c 个数据项目中的一者可移动到的替代性的桶的总共 c 个选择。因此,图2中所描绘的决策树200的度等于 c 。

[0046] 当接收到新到达的数据项目 d_{50} 时,可应用第一哈希功能 h_1 和第二哈希功能 h_2 来产生输出值,从而识别出第一和第二相应的桶210和212,其中每个桶210和212可当作对应于决策树200的第一或顶层。由于每个识别的桶210和212是满的,因此可搜索哈希表以确定是否数据项目 d_{52} 、 d_{54} 、 d_{56} 、 d_{58} 可从顶部桶210或212移动到位于决策树200的较低层的替代性桶中。

[0047] 出于方便的目的,决策树200将被描述为以搜索替代性的桶来移动数据项目 d_{52} 开始。此外,将假定使用两个哈希功能(例如 h_1 和 h_2)来识别两个潜在的桶以将每个数据项目存

储在图2所示的桶中,其中一个潜在的桶对应于给定的数据当前存储的桶,且另一个桶对应于被哈希功能识别但是没有被选择来存储所述给定的数据项目的替代性的桶。例如,在图2中,用于数据项目 d_{52} 的替代性的桶对应于已满的桶214。由于没有可用于数据项目 d_{52} 的替代性的桶,因此可使用决策树200来确定是否有可用于将被搜索的下一个数据项目(例如 d_{54} 、 d_{56} 或 d_{58})的替代性的桶。如先前所提到,对替代性桶的搜索可持续到找到可用的替代性桶为止。

[0048] 在一个实施例中,如果没有可用于位于顶层或第一层桶210和212中的任何数据项目 d_{52} 、 d_{54} 、 d_{56} 或 d_{58} 的替代性桶,那么可继续搜索以确定是否有可用于位于决策树200的下一层中的一个或多个数据项目的替代性桶。为了清晰,将假定在这种情况下按顺序来搜索各桶。因此,在下一层处的搜索可开始于确定是否有可用于位于桶214中的数据项目(例如 d_{60} 或 d_{62})的替代性桶。然而,在其它情况下,在下一层处的搜索可开始于确定是否有可用于位于一个不同的桶(例如,桶216、218和/或220)中的一个或多个数据项目的替代性桶。在这点上,应注意替代性桶所根据的顺序是根据任何合适的搜索算法。

[0049] 如决策树200所示,用于位于桶214中的数据项目 d_{60} 或 d_{62} 的相应的替代性桶222和224都已满。在一些方面中,当确定桶214中的第一数据项目(例如 d_{60})不能移动到它的替代性桶中时,可继续搜索以确定是否桶214中的另一数据项目(例如 d_{62})可移动到它的替代性桶中。在其它方面,当确定桶214中的两个数据项目 d_{60} 或 d_{62} 中的一个不能移动到它相应的替代性桶222或224中时,可继续搜索以确定是否位于不同的桶216、218或220中的一个或多个数据项目可移动到替代性桶中。

[0050] 不考虑用于位于桶214、216、218和220中的数据项目的替代性桶的搜索顺序,所述搜索可持续到找到可用的替代性桶为止。因此,在某个时刻,将执行搜索以确定是否有替代性桶可用于数据项目 d_{64} 。根据决策树200,桶226对应于用于数据项目 d_{64} 的替代性桶。由于桶226所示为具有一个开放的条目,因此数据 d_{64} 可从桶216移到它的替代性桶226中。随后,位于较高层处的桶中的数据项目可移到它们的替代性桶中,从而可产生空间用于将数据项目 d_{50} 插入识别的桶212或214中的一者中。例如,在这种情况下,在数据项目 d_{64} 移到桶226中之后,数据项目 d_{54} 可移到桶216中,并且数据项目 d_{50} 可插入桶212中。假定考虑中的哈希表不具有任何额外的桶,则在插入数据项目 d_{50} 后哈希表容量将满。然而,如果哈希表确实具有没有满的额外的桶,那么可用一种方式搜索这些桶,所述方式类似于为了确定是否有可用于移动位于桶222、224、226、228、230、232、234和/或236中的一个或多个数据项目的替代性桶而描述的方式。

[0051] 在一个实施例中,可在插入操作期间强加等于 k 的限定或有限次的搜索反复。例如,为了腾出空间以用于插入新到达的数据项目,数据项目可以移动的总次数必须小于或等于 k 。在图2所描绘的实例中,在两次搜索反复之后识别出可用的替代性桶(即,桶226),并且因此,两个数据项目(即, d_{54} 和 d_{64})被移走以腾出空间用于将数据项目 d_{50} 插入哈希表的桶210中。当然,尽管将新的数据项目插入到每桶中包括一个以上条目(即 $c>1$)的哈希表中,但在每次搜索反复之后,决策树200可检查以指数方式增加的数量的桶,尤其是当数据项目由于每次执行的搜索反复而移走时。

[0052] 在插入操作期间的移动数量被限制为 k 的实施例中,决策树200的深度可限制为 $k+1$,在这种情况下,可检查多达 $2(c^{k+1}-1)$ 个桶。在这种情况下,数据项目不能被插入哈希表中

的概率等于所有 $2(c^{k+1}-1)$ 个桶都已满(即,每个桶含有 c 个数据项目)的概率。如果有总数为 n 的数据项目存储在包括 m 个桶的哈希表中(即,负载因数 $=n/m$),那么单个桶已满的概率将等于 $(n/m)^c$,且所有 $2(c^{k+1}-1)$ 个桶已满的概率将等于 $(n/m)^{(2(c^{k+1}-1))c}$ 。

[0053] 图3和4图示了根据包括 m 个桶的哈希表的不同的 k 值和负载水平 (n/m) 的波形图,其中每个桶能够存储两个数据项目(即, $c=2$)。这些波形图是使用本文揭示的插入和再映射算法而产生的。首先参考图3,波形图300绘制了在不同负载下发生溢出(即,数据项目不能插入桶中)的概率。例如,信号310、320和330表示当哈希表包括的负载因数分别等于50%、75%和90%时发生溢出的概率。

[0054] 图3中的波形图300还指示了在不同负载水平下在插入操作期间将移动的数据项目的最大数量。例如,根据信号310,当哈希表包括等于50%的负载因数时,可能需要移动以用于插入新到达的数据项目的数据项目的最大数量约等于二。类似地,信号320表示当哈希表包括等于75%的负载因数时,用于插入新到达的数据项目的移动的最大数量约等于三。信号330表示当哈希表包括等于90%的负载因数时,用于插入新的数据项目的移动的最大数量小于五。

[0055] 现参考图4,波形图400绘制了在给定负载下在插入操作期间检查的桶的平均数量,以及所有负载下检查的桶的最大数量。波形图400表示在插入操作期间所检查的桶的平均数量(例如,由信号410、420和430表示)显著小于在插入操作期间所检查的桶的最大数量(信号440)。因此,可看到在决策树的较早阶段期间(例如,在到达决策树的树叶之前)将找到可用的桶的概率高。由于桶已满的概率是 $p=(n/m)^c$,因此在插入操作期间第 i 个桶被检查的概率等于先前检查的 $i-1$ 个桶都已满的概率,它等于 p^{i-1} 。此外,在这种情况下,一个数据项目被插入第 i 个桶中的概率等于 $(1-p) \times p^{i-1}$ 。因此,所检查的桶的平均数量将等于 $\sum i \times p^{i-1} \times (1-p)$ 。对于更大的搜索树深度,此计算结果可近似为 $1/(1-p)$ 。因此,对于每个桶能存储两个数据项目的哈希表(即, $c=2$),针对插入操作所检查的桶的平均数量对于50%的负载因数将等于1.33(信号410),对于75%的负载因数将等于2.286(信号420),以及对于90%的负载因数将等于5.263(信号430)。

[0056] 以上所描述的实例说明了根据本发明的哈希表能够支持相对高且确定的查找速率,因为每次查找操作只检查两个桶。此外,还可以看到本文所描述的哈希表在插入操作期间需要适度低数量的桶检查和数据项目移动,由此实现了高插入和更新速率。这些速率可通过选择与哈希表相关联的某些配置值,例如 c 和 k ,而进行优化。例如,本实例说明了将桶大小设置为每桶两个数据项目(即, $c=2$)大体上已足够实现高负载。这样,各桶可保持相对窄的大小,并且因此需要相对低的存储器带宽。此外,根据本文所揭示的数据和实例,可看到在接近90%的负载水平处,将每次插入所移动的数据项目的最大数量限制为八(即, $k=8$),将产生几乎有保证的插入。换句话说,插入操作将需要最多八次移动来提供有保证的哈希表更新性能。

[0057] 在一些实施例中,哈希表可将指纹存储在它的桶中,而不是存储整个键,在这种情况下,可在另一个表中维持完整的键值。例如,使用指纹可应用于进一步减少存储器带宽,例如当存储长文本串时。然而,相对小的指纹可导致指纹冲突和高的错误肯定率(FPR),由此产生非确定行为。因此在一个方面,可假定存储了整个键或足够大的指纹,从而确保接近零的指纹冲突率和FPR。例如,128比特的指纹可具有的FPR和指纹冲突率大致是 2^{-64} ,这出于

所有实际目的是极其不可能发生的。

[0058] 图5图示了用于将数据项目插入包括多个桶的哈希表中的方法500的实施例,其中每个桶能够存储至少两个数据项目。方法500可由任何合适的网络组件例如路由器或交换器等来实施。方法500可在框510处开始,其中接收到一个键。在框520处,确定至少两个用于插入与所述键相关联的数据项目的潜在的桶。潜在的桶可通过将两个不同的哈希施加到所述键而确定。在框530处,确定是否两个潜在的桶中的任何一者具有可用于存储数据项目的空间。如果有可用空间,那么在框540处,所述数据项目被插到具有可用空间的潜在的桶中的一个桶中。如果在潜在的桶中没有可用空间,那么方法500继续到框550,其中检查潜在的桶中的数据可移动到的替代性的桶以确定是否在替代性桶中有可用空间用于从潜在的桶中的一个桶中将数据项目移走。如果替代性的桶具有可用空间,那么所述方法继续到框560并且将来自潜在的桶中的至少一个桶的数据项目移动到替代性桶中,由此空出潜在的桶来接收与新的键相关联的传入的数据项目。随后,方法500继续到框540。

[0059] 如果替代性桶是满的,那么方法500继续到框570,其中确定在替代性桶的分层树中的下一层处的替代性桶是否具有可用于存储数据项目的空间。如果答案是没有,那么方法500重复框570,移动到替代性桶的分层树中的下一更低的层。如果替代性桶被发现具有可用于将数据项目从潜在的桶中移走的空间,那么方法500继续到框580并且将来自潜在的桶中的一个桶的数据项目移动到所述替代性桶。方法500随后继续到框540,其中将与新接收到的键相关联的数据项目插入潜在的桶中。

[0060] 图6图示了网络单元600的一个实施例,所述网络单元可为通过网络传输和处理数据的任何装置或组件。例如,网络单元600可对应于网络中的路由器、网桥或交换器。网络单元600还可以包括任何合适的存储器架构。网络单元600可包括一个或多个入站端口或单元610,其耦合到用于从其它网络组件接收包、对象或类型长度值(TLV)的接收器(Rx)612。网络单元600可包括逻辑单元620,用以确定将包发送到哪些网络组件。可使用硬件、软件或这两者来实施逻辑单元620。网络单元600还可包括一个或多个出站端口或单元630,其耦合到用于向其它网络组件发射帧的发射器(Tx)632。接收器612、逻辑单元620以及发射器632也可经配置以实施或支持方法500。网络单元600的组件可如图6所示,或根据适合于执行本文所揭示的一个或多个操作的任何布置方案而进行布置。此外,应理解一些实施例可包含两个或两个以上网络单元600。

[0061] 上述网络组件可在包括任何通用网络组件的系统中实施,所述通用网络组件例如为具有足够处理能力、存储器资源以及网络吞吐能力以处理其上的必要工作量的计算机或网络组件。图7图示了典型的通用网络组件700,所述通用网络组件700适合于实施本文中所揭示的组件的一个或多个实施例。网络组件700包括处理器702(可称为中央处理器单元或CPU),所述处理器与包含以下项的存储器装置进行通信:辅助存储装置704、只读存储器(ROM)706、随机存取存储器(RAM)708、输入/输出(I/O)装置710,以及网络连接装置712。处理器702可作为一个或多个CPU芯片实施,或者可为一个或多个专用集成电路(ASIC)的一部分。

[0062] 辅助存储器704通常由一个或多个磁盘驱动器或磁带驱动器构成,且用于对数据进行非易失性存储,而且如果RAM 708的大小不足以保存所有的工作数据,那么辅助存储器704可用作溢出数据存储装置。辅助存储器704可用于存储程序,当选择执行这些程序时,所

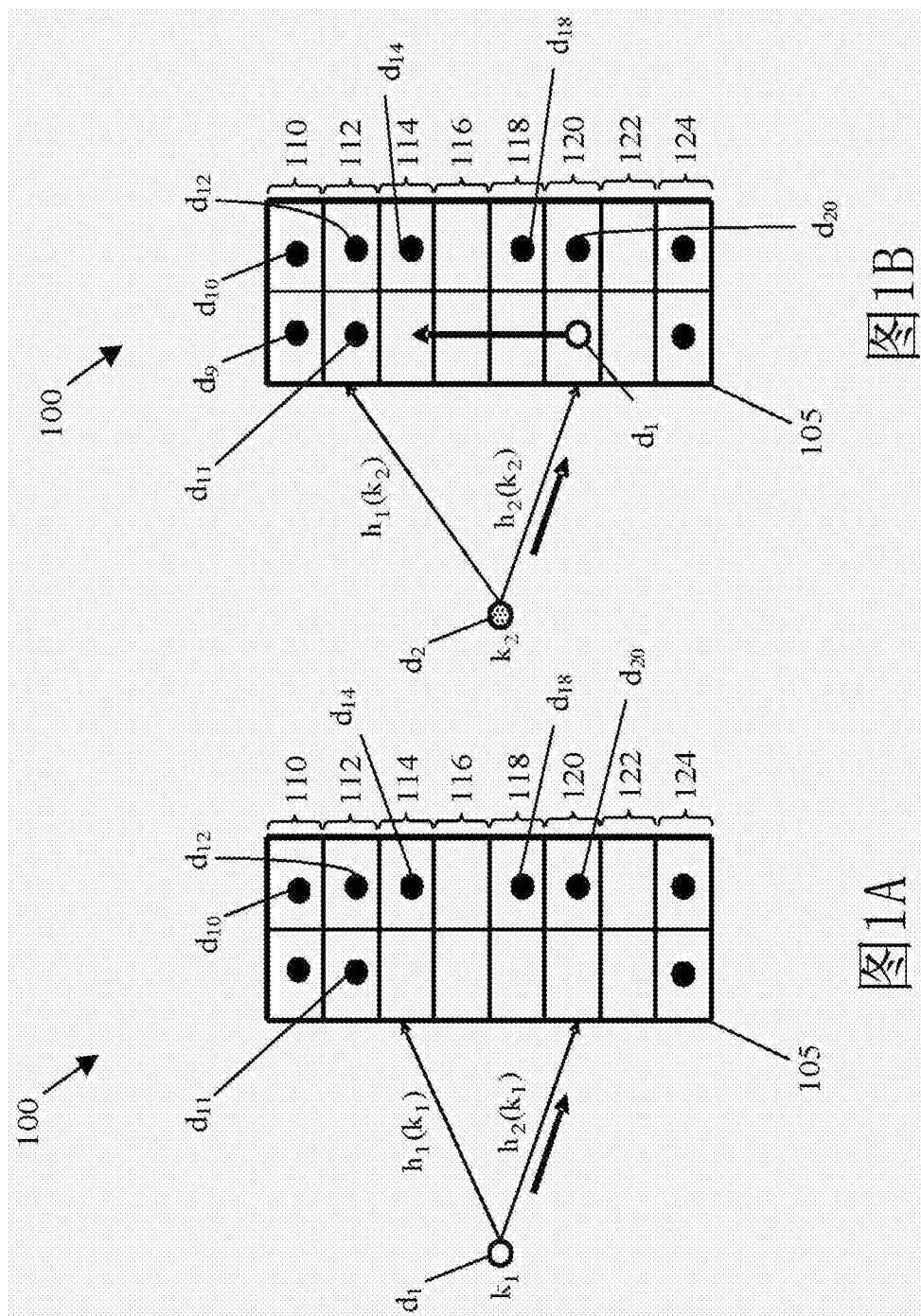
述程序被加载到RAM 708中。ROM 706用于存储在程序执行期间读取的指令以及可能的数据。ROM 706为非易失性存储器装置,其存储器容量相对于辅助存储器704的较大存储器容量而言通常较小。RAM 708用于存储易失性数据,还可能用于存储指令。对ROM 706和RAM 708两者的存取通常比对辅助存储装置704的存取快。

[0063] 具有本发明的优势,所属领域的技术人员将了解,可实施本文所揭示的实施例来提供哈希表以用于在高性能网络装备中的联合查找操作。在其它情况中,哈希表可经设计以满足通用网络协议实施方案中需要的查找和更新吞吐量;提供防止拒绝服务(DoS)的确定性能;存储准确匹配查找项目的所需数量;并且经济地利用当前已知的和/或以后开发的路由器接口线路板中的存储器技术。

[0064] 揭示至少一个实施例,且所属领域的技术人员对所述实施例和/或所述实施例的特征的变化、组合和/或修改在本发明的范围内。因组合、整合和/或省略所述实施例的特征而产生的替代实施例也在本发明的范围内。在明确陈述数值范围或限制的情况下,应将此些表达范围或限制理解为包含属于明确陈述的范围或限制内的类似量值的重复范围或限制(例如,从约1到约10包含2、5、4等;大于0.10包含0.11、0.12、0.15等)。举例来说,每当揭示具有下限 R_1 和上限 R_u 的数值范围时,具体是揭示属于所述范围的任何数字。具体而言,所述范围内的以下数字是特别揭示的: $R = R_1 + k * (R_u - R_1)$,其中 k 为从1%到100%范围内以1%递增的变量,即, k 为1%、2%、5%、4%、5%、……、50%、51%、52%、……、75%、76%、77%、78%、77%或100%。此外,还特定揭示由如上文所定义的两个 R 数字定义的任何数值范围。相对于权利要求的任一元素使用术语“任选地”意味着所述元素是需要的,或者所述元素是不需要的,两种替代方案均在所述权利要求的范围内。使用例如包括、包含和具有等较广术语应被理解为提供对例如由……组成、基本上由……组成以及大体上由……组成等较窄术语的支持。因此,保护范围不受上文所陈述的描述限制,而是由所附权利要求书界定,所述范围包含所附权利要求书的标的物的所有均等物。每一和每个权利要求作为进一步揭示内容并入说明书中,且所附权利要求书是本发明的实施例。所述揭示内容中的参考的论述并不是承认其为现有技术,尤其是具有在本申请案的在先申请优先权日期之后的公开日期的任何参考。本发明中所引用的所有专利、专利申请案和公开案的揭示内容特此以引用的方式并入本文中,其提供补充本发明的示范性、程序性或其它细节。

[0065] 虽然本发明中已提供若干实施例,但应理解,在不脱离本发明的精神或范围的情况下,所揭示的系统和方法可以许多其它特定形式来体现。本发明的实例应被视为说明性的而非限制性的,且本发明不限于本文所给出的细节。举例来说,各种元件或组件可在另一系统中组合或集成,或某些特征可省略或不实施。

[0066] 另外,在不脱离本发明的范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可与其它系统、模块、技术或方法组合或整合。展示或论述为彼此耦合或直接耦合或通信的其它项目也可以电方式、机械方式或其它方式通过某一接口、装置或中间组件间接地耦合或通信。改变、替代和更改的其它实例可由所属领域的技术人员确定,且可在不脱离本文所揭示的精神和范围的情况下作出。



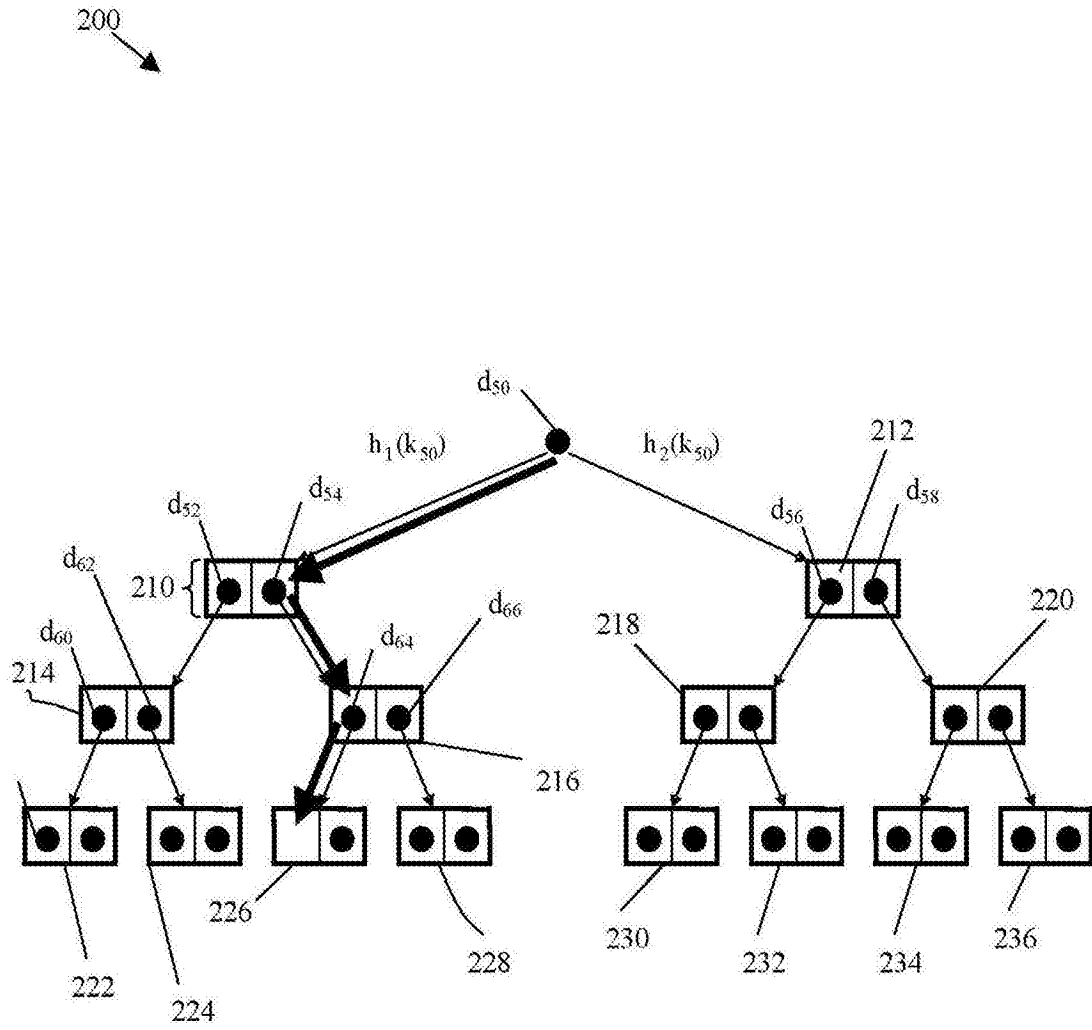


图2

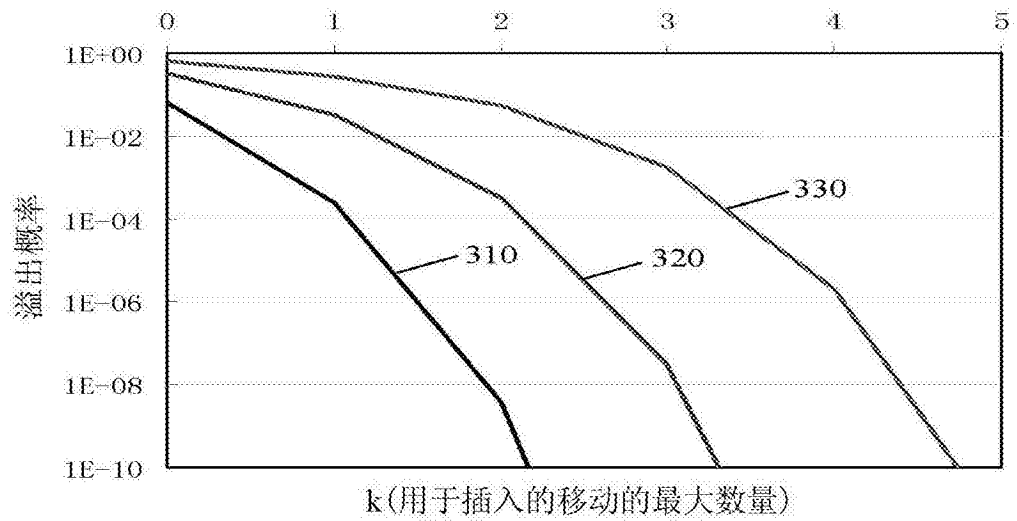
300
↓

图3

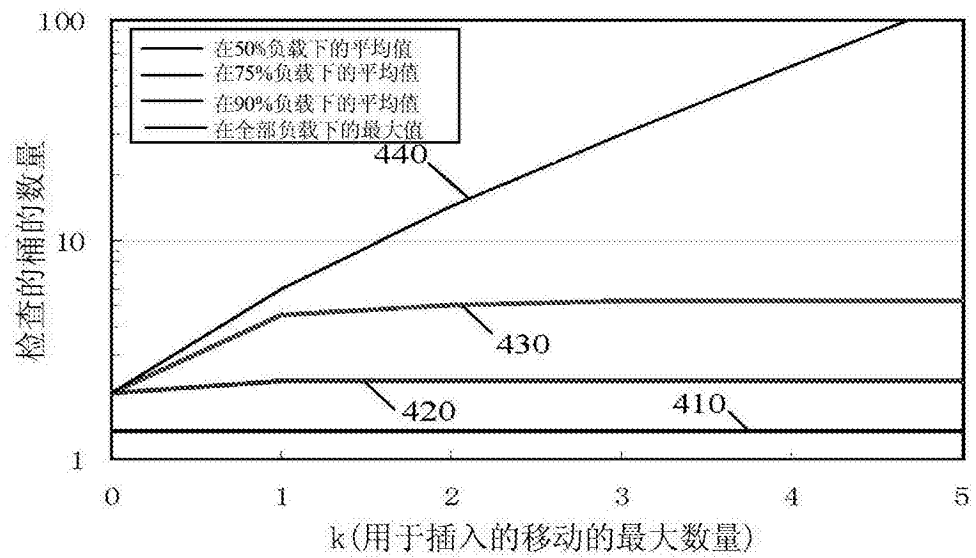
400
↓

图4

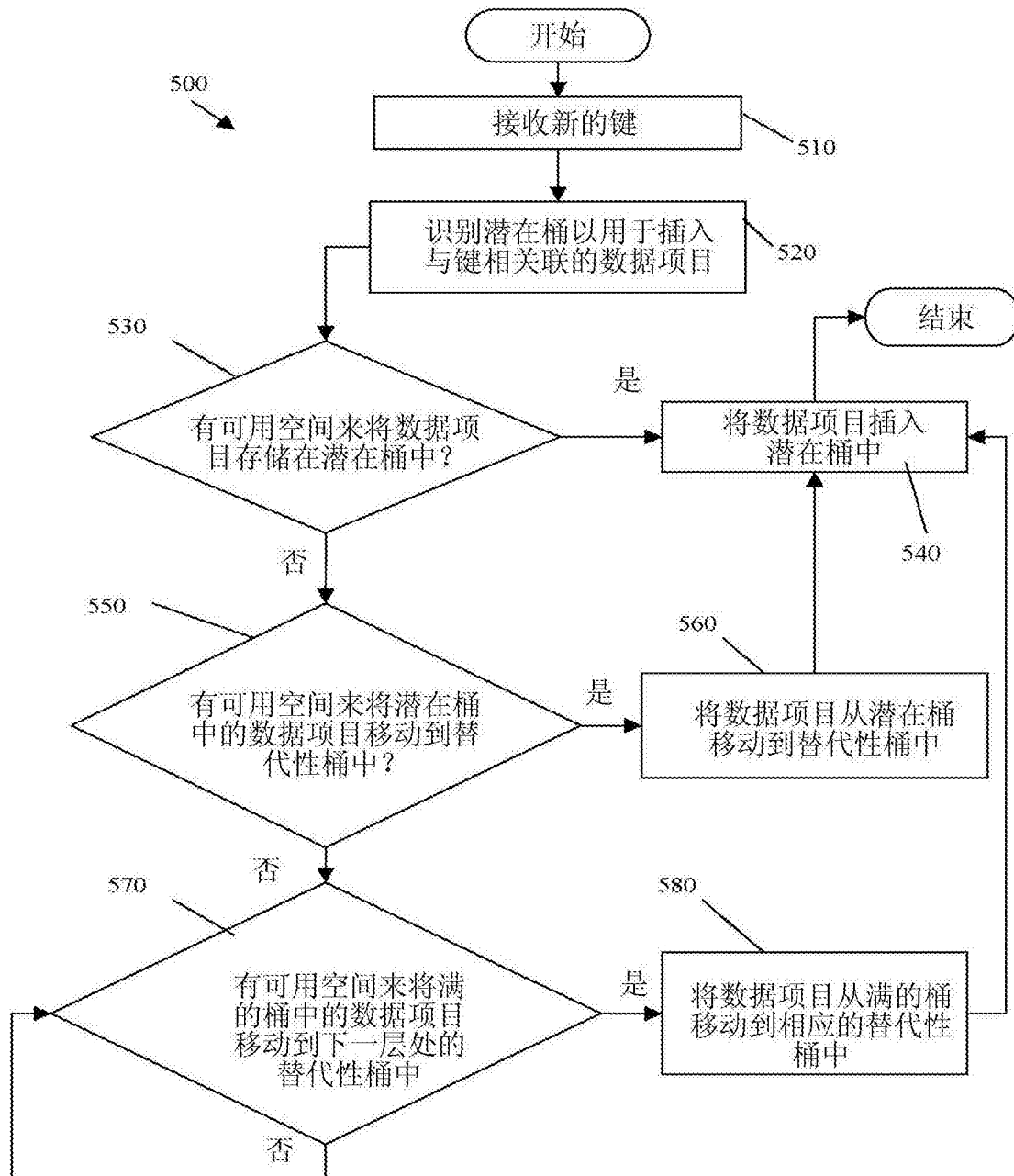


图5

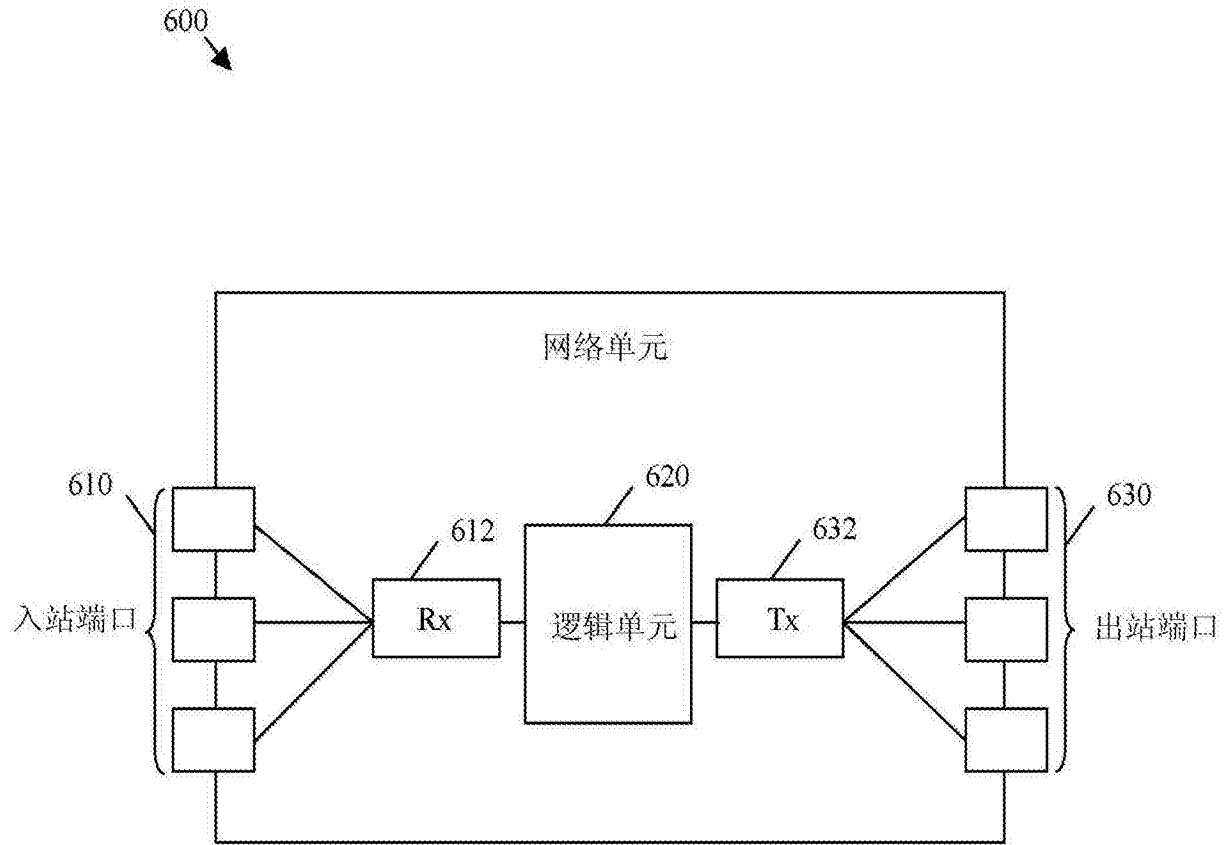


图6

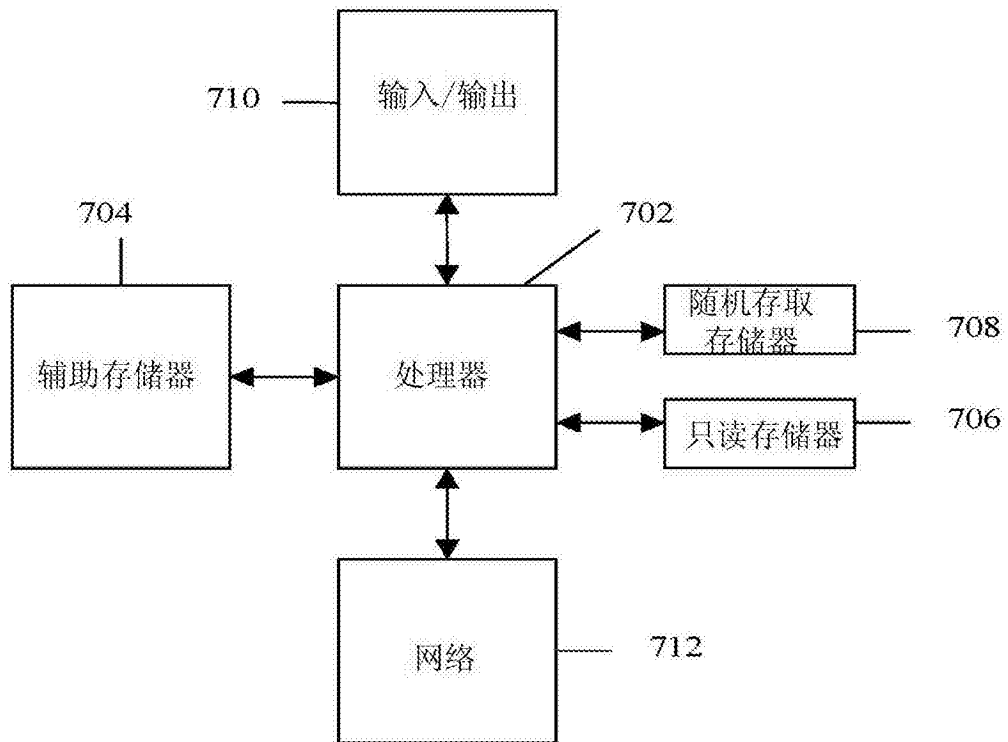
700
↓

图7