

(12) 发明专利申请

(10) 申请公布号 CN 103259694 A

(43) 申请公布日 2013. 08. 21

(21) 申请号 201310122376. 5

(22) 申请日 2013. 01. 18

(30) 优先权数据

61/587, 928 2012. 01. 18 US

61/644, 345 2012. 05. 08 US

61/652, 794 2012. 05. 29 US

(71) 申请人 马维尔以色列 (M. I. S. L.) 有限公司

地址 以色列约克尼穆

(72) 发明人 C·阿拉德 G·利维

(74) 专利代理机构 北京市金杜律师事务所

11256

代理人 鄭迅 辛鸣

(51) Int. Cl.

H04L 12/26 (2006. 01)

H04L 12/08 (2006. 01)

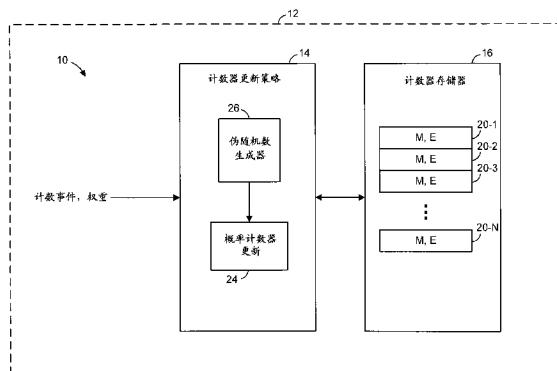
权利要求书3页 说明书14页 附图7页

(54) 发明名称

用于网络设备的空间高效计数器

(57) 摘要

本发明的实施例涉及用于网络设备的空间高效计数器。特别地，涉及一种网络设备，包括存储器和计数器更新逻辑模块。存储器被配置为存储多个比特。计数器更新逻辑模块被配置为基于数据流中多个数据单元的统计抽样来估计多个数据单元中量的计数，并将估计的量的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特。 $m$  个尾数比特代表尾数值  $M$  并且  $e$  个指数比特代表指数值  $E$ 。



1. 一种网络设备,包括:

存储器,被配置为存储多个比特;

计数器更新逻辑模块,被配置为:

基于数据流中多个数据单元的统计抽样来估计所述多个数据单元内量的计数,以及将估计的量的计数在所述存储器中存储为m个尾数比特和e个指数比特,其中所述m个尾数比特代表尾数值M并且所述e个指数比特代表指数值E。

2. 如权利要求1所述的网络设备,其中所述多个数据单元内的所述量的计数是以下之一:

所述多个数据单元内数据单元的计数;或者

所述多个数据单元内长度为q的块的计数,其中q是大于或等于1的整数。

3. 如权利要求1所述的网络设备,其中所述计数器更新逻辑模块被配置为至少部分地通过以下方式估计所述量的计数:

对于所述多个数据单元内的每个数据单元,更新所述m个尾数比特以用增加概率P来增加所述尾数值M;以及

当所述尾数值M滚计时,更新所述e个指数比特以增加所述指数值E。

4. 如权利要求3所述的网络设备,其中所述增加概率P随着所述指数值E的增加而指數地减少。

5. 如权利要求1所述的网络设备,其中所述多个数据单元内的所述量的计数是所述多个数据单元内长度为q的块的计数,并且其中长度q等于对应于所述多个数据单元的指定最小数据单元长度。

6. 如权利要求5所述的网络设备,其中所述多个数据单元是多个包,并且其中所述长度q是字节数。

7. 如权利要求5所述的网络设备,其中所述计数器更新逻辑模块被配置为至少部分地通过以下方式估计所述量的计数:

对于所述多个数据单元内的每个数据单元,

更新所述m个尾数比特以用增加概率P来增加所述尾数值M;以及

当相应的数据单元超过阈值长度时,更新所述m个尾数比特以增加所述尾数值M;以及

当所述尾数值M滚计时,更新所述e个指数比特以增加所述指数值E。

8. 如权利要求7所述的网络设备,其中所述增加概率P基于所述指数值E、所述指定最小数据单元长度和所述长度q。

9. 如权利要求1所述的网络设备,其中所述估计的量的计数等于 $M*2^E$ 。

10. 如权利要求1所述的网络设备,其中m大于或等于2。

11. 如权利要求1所述的网络设备,其中所述计数器更新逻辑模块进一步被配置为接收多个指示符,每个指示符对应于所述多个数据单元中相应的一个,并且其中所述多个数据单元的统计抽样包括所述多个指示符的统计抽样。

12. 如权利要求11所述的网络设备,其中所述多个指示符中的每个指示符是通过所述网络设备发送所述多个数据单元中相应的数据单元的指示符。

13. 如权利要求11所述的网络设备,其中所述多个指示符中的每个指示符是在所述网络设备中处理所述多个数据单元中相应的数据单元的指示符。

14. 一种网络设备中的方法,所述方法包括:

基于数据流中多个数据单元的统计抽样来估计所述多个数据单元内量的计数,以及

将估计的量的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特,其中,所述  $m$  个尾数比特代表尾数值  $M$  并且所述  $e$  个指数比特代表指数值  $E$ 。

15. 如权利要求 14 所述的方法,其中估计所述量的计数包括:

对于所述多个数据单元中的每个数据单元,更新所述  $m$  个尾数比特以用增加概率  $P$  来增加所述尾数值  $M$ ;以及

当所述尾数值  $M$  滚计时,更新所述  $e$  个指数比特以增加所述指数值  $E$ 。

16. 如权利要求 15 所述的方法,其中所述增加概率  $P$  随着所述指数值  $E$  的增加而指数地减少。

17. 如权利要求 14 所述的方法,其中所述多个数据单元内的所述量的计数是所述多个数据单元内长度为  $q$  的块的计数,并且其中所述长度  $q$  等于对应于所述多个数据单元的指定最小数据单元长度。

18. 如权利要求 17 所述的方法,其中估计所述量的计数包括:

对于所述多个数据单元内的每个数据单元,

更新所述  $m$  个尾数比特以用增加概率  $P$  来增加所述尾数值  $M$ ;以及

当相应的数据单元超过阈值长度时,更新所述  $m$  个尾数比特以增加所述尾数值  $M$ ;以及当所述尾数值  $M$  滚计时,更新所述  $e$  个指数比特以增加所述指数值  $E$ 。

19. 如权利要求 18 所述的方法,其中所述增加概率  $P$  基于所述指数值  $E$ 、所述指定最小数据单元长度和所述长度  $q$ 。

20. 如权利要求 14 所述的方法,其中,  $m$  大于或等于 2。

21. 一种网络设备,包括:

存储器,被配置为存储多个比特;

第一计数器,被配置为:

提供第一多个数据单元内第一量的精确计数;

将所述第一量的精确计数在所述存储器中存储为二进制数;

第二计数器,被配置为:

提供第二多个数据单元内第二量的估计的计数;

将所述第二量的估计的计数在所述存储器中存储为  $m$  个尾数比特和  $e$  个指数比特,其中所述  $m$  个尾数比特代表尾数值  $M$  并且所述  $e$  个指数比特代表指数值  $E$ ;以及

控制器,被配置为响应于所需计数器精度的一个或多个指示,(i) 采用所述第一计数器以用于在第一应用中使用,以及 (ii) 采用所述第二计数器以用于在与所述第一应用不同的第二应用中使用。

22. 如权利要求 21 所述的网络设备,其中

所述第一多个数据单元内的所述第一量的精确计数是以下之一:

所述第一多个数据单元内数据单元的精确计数;或者

所述多个数据单元内的长度为  $q_1$  的块的精确计数,其中  $q_1$  是大于或等于 1 的整数;以及

所述第二多个数据单元内的所述第二量的估计的计数是以下之一：

所述第二多个数据单元内数据单元的估计的计数；或

所述多个数据单元内长度为  $q_2$  的块的估计的计数，其中  $q_2$  是大于或等于 1 的整数。

23. 如权利要求 21 所述的网络设备，其中所述第二计数器被配置为至少部分地通过以下方式来提供所述第二量的估计的计数：

对于所述第二多个数据单元内的每个数据单元，更新所述  $m$  个尾数比特以用增加概率  $P$  来增加所述尾数值  $M$ ，其中，所述增加概率  $P$  随着所述指数值  $E$  被增加而指数地减少；

当所述尾数值  $M$  滚计时，更新所述  $e$  个指数比特以增加所述指数值  $E$ 。

24. 如权利要求 21 所述的网络设备，其中，所述所需计数器精度的一个或多个指示包括：

从所述第一应用接收的对计数器资源的第一请求；以及

从所述第二应用接收的对计数器资源的第二请求；以及

其中，与所述对计数器资源的第二请求相比，所述对计数器资源的第一请求对应于更高级别的所需计数器精度。

25. 一种网络设备中的方法，所述方法包括：

使用第一计数器来提供第一多个数据单元内第一量的精确计数；

将所述第一量的精确计数在存储器中存储为二进制数；

使用第二计数器来提供第二多个数据单元内第二量的估计的计数；

将所述第二量的估计的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特，其中所述  $m$  个尾数比特代表尾数值  $M$  并且所述  $e$  个指数比特代表指数值  $E$ ；以及

响应于所需计数器精度的一个或多个指示，(i) 采用所述第一计数器以用于在第一应用中使用，以及 (ii) 采用所述第二计数器以用于在与所述第一应用不同的第二应用中使用。

26. 如权利要求 25 所述的方法，其中提供所述第二量的估计的计数包括：

对于所述第二多个数据单元内的每个数据单元，更新所述  $m$  个尾数比特以用增加概率  $P$  来增加所述尾数值  $M$ ，其中，所述增加概率  $P$  随着所述指数值  $E$  的增加而指数地减少；以及

当所述尾数值  $M$  滚计时，更新所述  $e$  个指数比特以增加所述指数值  $E$ 。

27. 如权利要求 25 所述的方法，进一步包括接收所述所需计数器精度的一个或多个指示，其中接收所述一个或多个指示包括：

从所述第一应用接收对计数器资源的第一请求；以及

从所述第二应用接收对计数器资源的第二请求；以及

其中，与所述对计数器资源的第二请求相比，所述对计数器资源的第一请求对应于更高级别的所需计数器精度。

## 用于网络设备的空间高效计数器

[0001] 交叉引用的相关申请

[0002] 本公开要求了以下申请的权益:U.S. 临时专利申请 No. 61/587,928, 名称为“Increasing Device Counters”,2012 年 1 月 18 日 提交;U.S. 临时专利申请 No. 61/644,345, 名称为“Increase Device Counters”,2012 年 5 月 8 日提交;U.S. 临时专利申请 No. 61/652,794, 名称为“Increase Device Counters”,2012 年 5 月 29 日提交,这些申请的公开内容通过引用整体上结合于此。

### 技术领域

[0003] 本公开内容总体上涉及计数器,并且更具体地,涉及通信网络中使用的计数器。

### 背景技术

[0004] 这里提供的背景技术描述是出于总体上呈现本公开内容的上下文的目的。到在本背景技术部分中所描述程度的当前具名的发明人的工作,以及在提交申请时没有被作为背景技术的说明书的方面,既没有被明示也没有被暗示地承认为是针对本公开内容的现有技术。

[0005] 通信网络典型地采用诸如网桥和路由器这样的网络设备来在网络内转发信息。对通信网络的有效管理需要监控网络设备的各种操作属性,以及网络本身的操作属性。监控这样的属性经常需要计数器,近些年对计数器的使用已经获得引人注目的发展。用来容纳网络应用中使用的所有不同计数器所需要的物理空间也已相应地增长。

### 发明内容

[0006] 在一个实施例中,网络设备包括存储器和计数器更新逻辑模块。该存储器被配置用于存储多个比特。该计数器更新逻辑模块被配置用于基于数据流中的多个数据单元的统计抽样来估计该多个数据单元内量的计数,并将该估计的量的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特。该  $m$  个尾数比特代表尾数值  $M$ ,并且该  $e$  个指数比特代表指数值  $E$ 。

[0007] 在另一个实施例中,网络设备中的方法包括基于数据流中的多个数据单元的统计抽样来估计该多个数据单元内量的计数,并将该估计的量的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特。该  $m$  个尾数比特代表尾数值  $M$ ,并且该  $e$  个指数比特代表指数值  $E$ 。

[0008] 在另一个实施例中,网络设备包括被配置为存储多个比特的存储器、第一计数器、第二计数器和控制器。该第一计数器被配置为提供第一多个数据单元内的第一量的精确计数,并将该第一量的精确计数在存储器中存储为二进制数字。该第二计数器被配置为提供第二多个数据单元内的第二量的估计的计数,并将该第二量的估计的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特。该  $m$  个尾数比特代表尾数值  $M$  并且该  $e$  个指数比特代表指数值  $E$ 。该控制器被配置为响应于所需计数器精度的一个或多个指示,(i) 采用该第一计

数器以用于在第一应用中使用,以及 (ii) 采用该第二计数器以用于在与第一应用不同的第二应用中使用。

[0009] 在另一个实施例中,网络设备中的方法包括使用第一计数器来提供第一多个数据单元内的第一量的精确计数,并将该第一量的精确计数在存储器中存储为二进制数;使用第二计数器来提供第二多个数据单元内的第二量的估计的计数,并将该第二量的估计的计数在存储器中存储为  $m$  个尾数比特和  $e$  个指数比特。该  $m$  尾数比特代表尾数值  $M$  并且该  $e$  个指数比特代表指数值  $E$ 。该方法还包括响应于所需计数器精度的一个或多个指示, (i) 采用该第一计数器以用于在第一应用中使用,以及 (ii) 采用该第二计数器以用于在与第一应用不同的第二应用中使用。

## 附图说明

[0010] 图 1 是根据一个实施例的利用本公开内容的计数技术的示例计数器的框图。

[0011] 图 2 是根据一个实施例的使用统计计数器估计网络设备中计数的示例方法的流程图。

[0012] 图 3 是根据一个实施例的图示一个示例计数器更新过程的操作的框图。

[0013] 图 4 是根据一个实施例的利用本公开内容的计数器配置技术的示例计数器系统的框图。

[0014] 图 5 是根据一个实施例及场景的图 4 的计数器系统的计数器块内存储器的示例配置的框图。

[0015] 图 6 是根据一个实施例的配置网络设备中的计数器资源的示例方法的流程图。

[0016] 图 7 是根据一个实施例的配置网络设备中的计数器资源的另一个示例方法的流程图。

## 具体实施例

[0017] 在一些应用中,并不需要穿过网络设备的数据单元(例如,包或者元)的精确计数或者这些数据单元内的总字节数。例如,指示数据单元流是否是相对高带宽或者生成示出跨各种范围或类别的数据单元的分布的直方图的一些应

[0018] 用,可在不知任何精确数据单元计数或字节计数的情况下传递有用的信息。在下面描述的一些实施例中,网络设备中的计数器使用统计方法估计数据单元计数或字节计数,以便无需必须降低最大计数器值就可减少用于这样的应用的计数器数目。为了表示统计计数器的值,一个或多个比特被分配给尾数值,并且一个或多个比特被分配给指数值。在一个实施例中,如果确实发生的话,则通过将尾数值增加 1 或者当已经达到最大尾数值时将指数值增加 1(并滚计(roll over)尾数值)来增加计数器值。此外,尽管传统的二进制(精确)计数器对于每个检测到的数据单元(或字节)都增加计数器值,而在一个实施例中,统计计数器仅以随指数值增加而降低的概率增加计数器值。因此,在该实施例中,对数据单元计数的概率随着指数值(以及因此的计数器值)的增加而降低,但是每个相继“被计数”的数据单元将大量增加计数器值。在统计计数器计数字节而不是数据单元的一些实施例中,对于每个检测到的数据单元仍以特定的概率增加计数器值,但该概率是以如下因数增长的,该因数与被检测到的数据单元的长度或由被检测到的数据单元的多个最低有效位

表示的量成正比例。

[0019] 在下面描述的其他实施例中，网络设备是可配置的，从而使得可以基于特定客户端 / 应用的需要来分配有限的计数器资源。在这些实施例中，应用请求计数器资源（例如，比特存储和计数器更新逻辑），这些计数器资源能够提供所需数量的具有各种计数器参数的计数器。在一个实施例中，计数器参数包括每个请求的计数器（例如，在实施例中，不管计数器是精确或统计计数器，和 / 或大量的尾数比特）所需要的准确度或精度，以及计数器比特的数目。在一个实施例中，为每个精确计数器指定单一数目的比特，而为每个统计计数器指定多个尾数比特和多个指数比特。继而，在一个实施例中，通过根据所需要的计数器参数设置计数器更新逻辑，以及通过根据确定的比特分配来分配存储器的比特来对计数器资源进行配置。因此，可以按照满足特定应用的需求而无浪费过量比特的方式来分配有限的计数器资源。例如，在一个实施例中，最大计数器值可以被设置为最低可接受的水平，只要不需要精确的数据单元或字节计数就可以使用统计计数器来代替比特密集的二进制计数器，并且统计计数器的准确度可被设置为最低可接受的水平。图 1 是根据一个实施例的利用本发明公开内容的数据单元（例如，包或元（cell））计数技术和 / 或字节计数技术的示例计数器块 10 的框图。例如，计数器块 10 被包括在网络设备 12 中，网络设备 12 为诸如路由器、网桥或其他网络交换机。在一个实施例中，网络设备 12 被配置为在以太网通信系统内操作。为了解释方便，下面具体参考包来描述图 1 的实施例。然而，可以理解，其他类似的实施例作为替代关于其他适合类型的数据单元操作。例如，在一些实施例中，实施例对元（或字节，或元内的其他单元等）而不是包进行计数。

[0020] 计数器块 10 包括计数器更新策略模块 12，该计数器更新策略模块 12 包括管理计数器块 10 内的 N 个计数器的操作的逻辑。为了改变计数器块 10 中的计数器的当前值，计数器更新策略模块 14 操纵计数值比特 20-1 至 20-N 内的特定比特，其存储在存储器 16 中，代表计数器值。计数器比特 20-1 至 20-N 各自包括一个或多个尾数比特以及一个或多个指数比特。在一些实施例和 / 或场景中，尾数比特和 / 或指数比特的数目因计数器不同而不同。例如，在各种实施例和 / 或场景中，计数值比特 20-1 比计数值比特 20-2 包括更多或更少的尾数比特，和 / 或更多或更少的指数比特。在一个备选实施例中，存储器 16 中仅包括对应于单个计数器的计数值比特 20-1。在一个实施例中，计数器块 10 内的每个计数器都具有等于  $M \times 2^E$  的计数器值 C，其中 M 是由对应的尾数比特所表示的尾数值，并且 E 是由对应的指数比特所表示的指数值。为了解释方便，下面具体参考对应于计数值比特 20-1 的计数器的操作来描述图 1 的实施例。在一些实施例中，其他 (N-1) 计数器中的每个计数器以类似方式来操作。

[0021] 计数器更新策略模块 14 基于由计数器更新策略模块 14 所接收到的“计数事件”和“权重”指示符来增加计数器的计数器值。在一个实施例中，每个“计数事件”指示符指示特定的包符合一个或多个标准，诸如具有一个或多个包特征（例如，属于如由一个或多个包字段值所标识的特定流，超过特定字节长度，具有在特定地址范围内的源或目的地地址，等等）和 / 或为特定的包状态（例如，接受的，转发的，列队的，丢弃的等等）。在各种不同的实施例中，“计数事件”指示符由包检测器（图 1 中未示出）生成，该包检测器对接收到的包（例如，由网络设备 12 接收到的包），或者对头部、描述符，或其他与接收到的包相关联的信息进行操作。为了方便说明，“检测包”在这里指的是确定该包具有一个或多个特定特征和

/ 或处于一个或多个特定状态，并且“被检测到的包”在这里指的是已经被检测为具有必需特征和 / 或处于必需状态的包。

[0022] 当计数器更新策略模块 14 接收到“计数事件”指示符时，计数器更新策略模块 14 仅以概率 P 增加由计数值比特 20-1 所表示的计数器值。在一个实施例中，当计数器值被增加时，其是通过将尾数值 M 加 1 来增加的，除非已经达到最大尾数值（例如，可以通过计数值比特 20-1 的尾数比特表示的最大值），在这种情况下，指数值 E 被增加 1 并且尾数值 M 滚计。

[0023] 如上所述，计数器更新策略模块 14 仅以概率 P 来增加计数器值。在对包进行计数的一个实施例中，针对给定“计数事件”指示符而增加计数器值的概率是  $P = 1/2^E$ 。因此，随着指数和计数器值以上述方式来增加，每个附加的检测到的包将被“计数”（即，使得计数器值增加）的机会越来越小。然而，这是通过以下事实来补偿的，即随着指数值增加，被计数的每个被检测到的包将以越来越大的数量增加计数器值。尽管该方法仅提供对被检测到的包的实际计数的统计近似，但与精确的二进制计数器相比其允许固定数目的比特来表示高得多的最大计数器值。例如，具有 8 比特的二进制计数器可以指示计数器值最大到  $2^8 - 1$ ，或者 255，而具有 8 比特的统计计数器可以指示大得多的值（例如，对于 3 个尾数比特和 5 个指数比特来说，可达到  $7 * 2^{31}$ ，或者 15032385536）。最大计数器值随着分配给指数（例如，计数值比特 20-1 中的指数比特）的比特数量被增加而急剧增加，并且随着分配给尾数（例如，计数值比特 20-1 中的尾数比特）的比特数量被增加而达到较小程度。计数器值中的分辨率，以及由此的相对于被检测到的包的实际计数的计数器准确度和精度，也随着尾数比特数量被增加而增加。在一个实施例中，至少两个尾数比特被包括在计数值比特 20-1 中。

[0024] 在一个实施例中，为了测试是否应当响应于接收到“计数事件”指示符而增加计数器值，计数器更新策略模块 14 内的概率计数器更新模块 24 进行以下两个操作：(1) 基于由计数值比特 20-1 内的指数比特所指示的当前指数值 E，计算为每个“计数事件”指示符增加计数器值的概率  $P = 1/2^E$ ；以及 (2) 比较计算的概率 P 与由伪随机数生成器 26 生成并从其接收的伪随机数。在一个实施例中，伪随机数生成器 26 以均匀分布方式在 0 和 1 之间生成数，并且概率 P 总是大于 0 而小于 1。在一个实施例中，伪随机数生成器 26 提供伪随机数以用于由计数器块 10 内的 N 个计数器中的每一个来使用。

[0025] 在一些实施例和 / 或场景中，计数器块 10 内的一个或多个计数器对字节而不是对包进行计数。在一个实施例中，为了对基于“计数事件”指示符的字节计数进行近似，对字节的“块”而不是对单个字节进行计数，因此，计数器值应当乘以字节块大小（例如，通过数字处理器对由计数值比特 20-1 所表示的计数器值进行抽样）以便确定估计的字节计数。例如，在一个实施例中，计数在 q 字节的块中执行，其中 q 是计数量单元（即，以 q 的倍数来执行计数）。在各种实施例中，q 反映了最小包长度（例如，用于以太网包的 64 字节），和 / 或提供对于特定联网应用来说足够的计数准确度 / 精度的任何其他适合量。在一些实施例和 / 或联网应用中，q 等于 1（例如，一个字节）。

[0026] 在一字节计数的实施例中，概率计数器更新模块 24 针对每个“计数事件”指示符计算增加概率 P，其中 P 基于以下来确定：(1) 对应于“计数事件”指示符的包的包长度 L；(2) 计数量单元 q；以及 (3) 当前指数值 E。具体地，在一个实施例中，包长度 L 被标准化为计数量 q 以确定标准化的包长度  $L_q = L/q$ ，继而通过将  $L_q$  的 E 个最低有效位 (LSB) 除以  $2^E$

来确定 P。同样,在该实施例中,当  $L_q$  具有除 E 个 LSB 之外的一个或多个非零比特时,将尾数值 M 增加附加量。下面将参考图 3 对该实施例进行更详细的描述。

[0027] 每个包的长度是基于“权重”指示符来确定的,在各种实施例中,其是包长度 L、标准化的包长度  $L_q$ ,或者指示包长度的另一个合适度量。在一个实施例中,计数器更新策略模块 14 从另一个处理单元(图 1 中未示出)接收检测到的包的“权重”指示符,并且因此概率计数器更新模块 24 知道每个包的字节长度。如果所有包具有相同长度,则在一个实施例中对包而不是对字节(例如,可以独立于计数器 10 来确定字节,简单地通过将计数器值乘以固定包长度)计数是足够的。因此,在该实施例和 / 或场景中,增加概率 P 等于  $1/2^E$ 。在所有包具有相同长度或对包而不是对字节进行计数的一些实施例中,没有“权重”指示符被提供给计数器更新策略模块 14。

[0028] 在一个实施例中,其中计数器是字节计数器,其中 P 等于  $L_q$  的 E 个 LSB 除以  $2^E$  的结果,并且其中计数器值  $C = M * 2^E$ ,对于示例的包计数器,尾数值和指数值以如上所述相同方式被增加(即,将尾数值增加 1 直到达到最大值,在该最大值点尾数值滚计并且指数值加 1)。然而,在一些实例中,尾数值 M 和 / 或指数值 E 可增加多于 1,如下参考图 3 所描述的。为了确定估计的字节计数,在一个实施例中,计数器值  $C = M * 2^E$  乘以 q(例如,通过对计数器值 C 进行抽样的数字处理器),从而使得估计的字节数目等于  $C_{est} = q * M * 2^E$ 。

[0029] 在上述的字节计数实施例中,计数器对等于最小包长度(以字节)的单元进行计数。然而,在各种其他实施例中,计数器对任何其他适合量的单元进行计数。因此,通过改变 q 能够获得针对比特、半字节、字节、字等的计数器估计,不管是单独的还是以任何大小的块。

[0030] 表 1 图示了对于一个包计数器和一个字节计数器的示例情况,如何通过使用类似于计数器块 10 的计数器的统计计数器而不是精确计数器来保存计数器比特(以及由此的片区域):

[0031]

计数器类型	精确计数器		统计计数器	
	大小(比特)	最大值	大小(比特)	最大值
包	29	$2^{29}$	16 (11m, 5e)	$2^{43}-1$
字节	35	$2^{35}$	16 (11m, 5e)	$2^{49}-1(q=64)$ 或 $2^{43}-1(q=1)$
总和	64 比特		32 比特	

[0032] 如表 1 中所示,一半的比特可以用来提供大得多的包及字节计数器值。代价是由于由尾数比特、计数量 q 所提供的计数分辨率所导致的计数不准确度,以及由于统计计数方法所导致的固有的不准确度。如上所述,尾数比特的数目确定诸如计数器块 10 的计数器之类的统计计数器的准确度和精度。

[0033] 图 2 是根据一个实施例的使用统计计数器估计网络设备中的计数的示例方法 50 的流程图。在各种实施例中,方法 50 被全部地或部分地由图 1 的网络设备 12 中的计数器更新策略模块 14 实施。如图 1 一样,具体参考包来描述图 2 的实施例。然而,再次应当理解其他实施例作为替代关于其他适合类型的数据单元(如,元)进行操作。

[0034] 方法 50 估计量的计数,其中每个量是任何适当的计数单元(例如,包,最小包长度中的字节的数目等等)。此外,在一个实施例中,关于特定包流中的包对量进行计数,其中包流可以按照任何适当的方式来定义(例如,具有特定头部字段值的包,已经接收到的包,已被丢弃的包,等等)。在一个实施例中,对于多个包中的每个包重复方法 50。

[0035] 在块 52,接收包指示符。该包指示符对应于相关“流”内的特定包,并且指示该包已被检测到。在各种实施例中,包指示符是从实施方法 50 的相同网络设备中的包处理单元接收到的信号,或者从不同网络设备接收到的信号。例如,在一个实施例中,接收到的信号正常在“低”电平,并且每当信号转变为“高”电平时指示包(反之亦然)。在一个实施例中,包指示符类似于参考图 1 所描述的“计数事件”指示符中的一个指示符。

[0036] 在块 54,确定当前计数器值 C。在一个实施例中,通过从存储器中读取比特(诸如在图 1 的存储器 16 中存储的计数值比特 20-1 中所包括的尾数和指数比特)确定计数器值。在一个实施例中,计数器值 C 等于  $M \times 2^E$ ,其中 M 是由 m 个尾数比特所表示的尾数值,并且 E 是由 e 个指数比特所表示的指数值。

[0037] 在块 56,确定增加概率 P。在一个实施例中,增加概率 P 是随着指数值 E 被增加(并且由此随着计数器值 C 增加)而指数地减少的值。在方法 50 表示包计数器的操作(即,其中计数器值 C 是对包的数目的估计)的一个实施例中,增加概率 P 等于  $1/2^E$ 。在方法 50 表示字节计数器的操作的一个实施例中,并且其中包具有等于计数量 q 的最小字节长度,增加概率 P 等于  $L_e / 2^E$ ,其中  $L_e$  等于  $L/q$ (例如,下面参考图 3 进行更详细的描述)的 E 个 LSB。

[0038] 在块 60,以概率 P 来增加计数器值 C,其中 P 是在块 56 确定的概率 P。在一个实施例中,对是否增加计数器值 C 的确定是通过比较伪随机数与概率 P 来做出的。在一个实施例中,伪随机数是由与图 1 的伪随机数生成器 26 类似的伪随机数生成器来生成的。在方法 50 表示包计数器的操作的一个实施例中,增加计数器值 C 包括将尾数值 M 增加 1,或者如果 M 达到其最大值并滚计则增加指数值 E(并重置 M)。在方法 50 表示字节计数器的操作的一个实施例中,增加计数器值 C 包括根据图 3 的过程或适合的等同过程来增加尾数值 M 和 / 或指数值 E。

[0039] 在块 62,m 个尾数比特和 e 个指数比特的更新值(即,代表对应于增加的计数器值 C 的尾数值 M 和指数值 E 的二进制值)被写入计数器存储器。在一些实施例中,只有在作为块 60 的增加结果而使得比特值被改变时才在块 62 写入或重写这些比特。在一个实施例中,例如,当在块 60 没有发生增加的情况下,在块 62 没有比特被重写。在其他实施例中,比特在块 62 被重写而不管块 60 处的结果(例如,对每个尾数比特和 / 或每个指数比特发生写入操作,即使仅仅是以相同的比特值来重写这些比特)。

[0040] 在一些实施例中,方法 50 的块的发生顺序与图 2 中可见的顺序的不同。例如,在一个实施例中,当在块 52 接收包指示符之前在块 56 确定增加概率 P。在另一个示例实施例中,当在块 62 写入更新的计数器值比特之前或者同时,在块 52 接收针对第二个检测到的包的包指示符(在方法 50 的第二次迭代中),对于方法 50 的第一次迭代中的第一个被检测到

的包,在块 60 以概率 P 增加计数器值 C,和 / 或在块 56 确定增加概率 P。

[0041] 图 3 是根据一个实施例的图示示例的计数器更新过程的操作的框图。在一个实施例中,图 3 中所示的操作对应于在其中计数器更新策略模块 14 接收“计数事件”指示符以及指定包长度的对应“权重”指示符的场景。

[0042] 在图 3 中,一组初始计数器值比特 100 包括  $m$  个尾数比特和  $E$  个尾零(例如,等于由  $e$  个指数比特所表示指数的当前整数值的多个尾零)。添加  $E$  个尾零用来将尾数值  $M$  转换为当前整数计数值,  $M*2^E$ 。然而,应当理解,图 3 以及这里所附的描述仅仅是旨在有助于对根据一个实施例的计数器更新过程的理解。例如,在一些实施例中,实际的实施方式执行等同于图 3 的操作的操作而无需在计数器值比特和整数格式之间转换(例如,无需插入  $E$  个尾零等)。

[0043] 包长度比特 110 表示检测到的包的量长度  $L_q$ (即,包长度  $L$  被标准化到计数量  $q$ ,从而使得  $L_q = L/q$ )。例如,在包是如 511 量一样大的一个实施例中,包长度比特 110 包括至少  $n = 9$  个比特。如图 3 中所示,与初始计数器值比特 100 的被移位的尾数比特重叠的包长度比特 110 的最高有效位(MSB)被添加到尾数值  $M$ ,从而形成第一和比特 120。在图 3 的示例实施例中,该添加仅发生在  $n$  比  $E$  大的情况下(即,如果包长度比阈值长度大,其中阈值长度随着  $E$  的增加而增加),在这种情况下,包长度比特的  $(n-E)$  个 MSB 被添加到尾数值  $M$ 。在一个实施例中,与初始计数器值比特 100 的  $E$  个尾零重叠的包长度比特 110 的 LSB 没有被添加到初始计数器值比特 100 或者第一和比特 120。然而,在一个实施例中,比特 130 以等于  $L_e/2^E$  的增加概率  $P$  被添加到第一和比特 120,其中  $L_e$  等于包长度比特 110 的  $E$  个 LSB。因此,虽然在该实施例中包长度比特的  $E$  个 LSB 没有被添加到计数器值,但它们影响包的增加概率  $P$ 。由于  $E$  随着计数器值增加,当计数器值远大于  $L_q$  时,增加概率  $P$  降低到  $L_q/2^E$ 。

[0044] 比特 130 当被添加到比特 120 时,尾数值  $M$  增加了 1。形成的新计数器值比特 140 提供新计数器值的整数表示,其根据增加概率  $P$  与比特 130 的添加结合或不与其结合。在一个实施例中,一旦该新计数器值 140 被确定,则新计数器值 140 被转换到指数表示  $M*2^E$ 。在一个实施例中,为了转换新计数器值,新计数器值比特 140 的  $m$  个 MSB(起始于且包括在比特 140 中所包括的最高有效(即最左侧)“1”值)在存储器中被存储为  $m$  个尾数比特,并且比特 140 中该  $m$  个尾数比特的右边的比特数等于指数值  $E$ 。在一个实施例中,将指数值  $E$  转换到二进制数提供了存储在存储器中的  $e$  个指数比特。

[0045] 以上述方式将新计数器值比特 140 转换到指数表示的结果使得固有地发生了两个操作。首先,当尾数值  $M$  滚计(即,以“回绕(wrap-around)”形式)时,新尾数值  $M$  等于 0 加上加法的任意余数。此外,如果表示尾数值  $M$  的比特  $m$  的数目相当小,则尾数值  $M$  可以在单个的计数器更新操作中环绕多次,从而使得  $E$  增加多于 1。然而,再次地,在一些实施例中,实际实施方式作为替代执行等同于图 3 的操作的其他操作而无需在计数器值比特和整数格式之间转换。

[0046] 在一些实施例中,数字处理器(例如,CPU)通过实施数字计数器来提高计数器 10 的有效最大计数器值,其中该数字计数器偶尔(如周期性地)抽样计数器 10 的尾数和指数比特,处理器计算对应于尾数和指数比特的每个抽样组的计数器值,并将尾数和指数比特重置为 0。在一些实施例和 / 或应用中,数字处理器将每个计数器值添加到该数字计数器

值。在其他实施例和 / 或应用中，数字处理器利用计数器 10 的抽样值，但是并不保留或计算数字计数器值。

[0047] 图 4 是根据一个实施例的利用本公开内容的计数器配置技术的示例计数器系统 300 的框图。在一个实施例中，计数器系统 300 例如被包括在诸如路由器、网桥或其他网络交换机之类的网络设备中。在一个实施例中，计数器系统 300 被包含在被配置为运行于以太网系统内的网络设备中。

[0048] 计数器系统 300 包括表示网络设备内的各种应用的客户端 302-1 至 302-N，其中每个客户端执行利用一个或多个计数器的操作。这些应用可能需要对符合特定标准的数据单元（例如，包或元）的数目的计数，这些标准例如诸如有多少数据单元处于特定状态，和 / 或有多少数据单元具有特定特征。例如，流量测量应用可以基于由网络设备接收（或转发）的数据单元的数目或者被识别为属于特定时间帧中的特定流的数据单元的数目来生成流量带宽的指示符。作为另一个示例，调试应用可以基于被网络设备丢弃的数据单元的数目来生成网络性能的指示符。其他应用作为替代可能需要对在符合标准的数据单元中所包括的字节的总数目进行计数。例如，在包具有可变长度的系统中，更精确的流量测量应用可以基于由网络设备所接收（或转发）的包中的字节的总数目来生成流量带宽的指示符。

[0049] 为了支持所需的计数器，计数器系统 300 包括计数器资源 304（例如，计数器逻辑和用于计数值比特的储存器）。在图 4 中所示的实施例中，计数器资源 304 包括多个计数器资源块 306，每个计数器资源块 306 包括计数器逻辑 310，计数器逻辑 310 部分地通过操纵存储在相应存储器 312 中的计数值比特管理一个或多个计数器的操作。为了清楚目的，在图 4 中计数器逻辑 310 和存储器 312 仅针对单个计数器资源块 306 示出。在一个备选实施例中，计数器资源 304 仅包括单个计数器资源块 306。

[0050] 在一个实施例中，包括在计数器资源 304 中的每个计数器资源块 306 可被配置为单个计数器或者多个计数器。此外，在一个实施例中，计数器资源块 306 的每个计数器可被配置为作为计数器估计器（例如，具有与如上关于图 1、图 2 和 / 或图 3 所述的操作类似的操作的统计计数器）、或精确的二进制计数器来运行。因此，在各种不同实施例中，每个计数器资源块 306 的计数器逻辑 310 可被配置为管理多个二进制计数器、多个统计计数器或者一个或多个二进制计数器与一个或多个统计计数器的混合体的操作。根据这些不同的场景，相应的存储器 312 可以存储二进制计数器比特的一个或多个组，和 / 或尾数和指数比特的一个或多个组。在一个备选实施例中，一些计数器资源块 306 仅可以被配置为精确计数器，而其他计数器资源库仅可以被配置为统计计数器。

[0051] 图 5 是根据一个实施例和场景的图 4 的计数器系统的计数器块 306 内的存储器 312 的示例配置 313A、313B 的框图。在一个实施例中，计数值比特 314 的每个组对应于单个计数器。在一个实施例中，第一配置 313A 是由请求两个二进制计数器和两个计数器估计器的客户端（例如，图 4 的客户端 302-1）所产生的。因此，存储器 312 被配置以使得计数值比特 314-1 和 314-2 专用于相应的二进制计数器，并且以使得计数值比特 314-3 和 314-4 专用于相应的计数器估计器（统计计数器）。在一个实施例中，4 组计数值比特 314-1 至 314-4 中的每一组基于不同计数事件而被更新。例如，在一个实施例和场景中，每当网络设备接收到包时，计数值比特 314-1 被更新以便增加计数器，每当包被丢弃时，计数值比特 314-2 被更新以便增加计数器等等。

[0052] 第二配置 313B 是由相同的应用（例如，客户端 302-1）或者请求 6 个计数器估计器而不要二进制计数器的不同应用（例如客户端 302-2）所产生的。因此，存储器 312 被配置以使得计数值比特 314-1 至 314-6 专用于相应的计数器估计器。例如，第一配置 313A 可以代表在其中客户端 302 要求两个计数器的 100% 准确度的场景，而第二配置 313B 可以代表在其中客户端 302 要求更多的计数器而较少的准确度或精度的场景。在一个实施例中，六组计数值比特 314-1 至 314-6 中的每一组基于不同计数事件而被更新。例如，在一个实施例和场景中，每当网络设备接收到包时，计数值比特 314-1 被增加（以增加概率  $P_1$ ），每当包被确定为超过特定包长度时计数值比特 314-2 被增加（以增加概率  $P_2$ ），等等。

[0053] 再次参考图 4，示例计数器系统 300 还包括控制器 324（例如，在一个实施例中的软件控制器），其中控制器 324 包括计数器资源管理器 326 和流管理器 328 两者。为了便于说明，参考单个客户端（客户端 302-1）来描述控制器 324 的操作。在一个实施例中，客户端 302-1 向计数器资源管理器 324 请求计数器资源。在一个实施例中，该请求包括计数器的所需数目以及那些计数器中的每个计数器所需要的属性（例如，计数器准确度，以及计数器是对数据单元进行计数还是对字节进行计数）。继而，在一个实施例中，计数器资源管理器 324 将一个或多个计数器资源块 306 与客户端 302-1 相关联。特别地，在一个实施例中，计数器资源管理器 324 配置硬件映射表来将该特定计数器块 306 与客户端 302-1 相关联。

[0054] 在一个实施例中，计数器资源管理器 324 还基于在来自客户端 302-1 的请求中所指定的计数器参数来配置相关联的计数器资源块 306。在一个实施例中，在每个相关联的计数器资源块 306 内，该配置包括逻辑 310 的配置（例如，对于精确或统计操作），并且包括比特的分配（例如，对尾数和指数比特或一个或多个计数器，和 / 或对二进制计数器的比特）。在各种实施例中，计数器资源管理器 324 对请求的统计计数器进行配置以使其以与上述参考图 1、图 2 和 / 或图 3 所描述的类似方式来进行操作。

[0055] 在一个实施例中，客户端 302-1 进一步请求流管理器 328 为各种计数器安装(install) 流。在一个实施例中，每个安装的流与计数器相关联。在一个实施例中，一旦被安装，则流中每个被检测到的数据单元用作对应计数器的计数事件，类似于参考图 1 所描述的“计数事件”指示符。

[0056] 示例计数器系统 300 还包括生成伪随机数的伪随机数生成器 330，所述伪随机数被提供给被配置为支持至少一个统计计数器的每个计数器资源块 306 的计数器逻辑 310（即，与由计数器逻辑 310 所确定的增加概率进行比较）。在一个备选实施例中，每个计数器资源块 306 包括不同的专用伪随机数生成器，其类似于伪随机数生成器 330（例如，类似于图 1 中所示的实施例，其中计数器块 10 包括专用的伪随机数生成器 26）。

[0057] 在一些实施例中，在计数器操作期间，客户端 302-1 还发送附加信息给每个相关联的计数器块 306 的计数器逻辑 310，该信息为诸如与图 1 的“权重”指示符类似的数据单元长度（例如，在其中数据单元长度被用来计算统计数据单元计数器的增加概率的实施例和场景中），和 / 或与图 1 的“计数事件”指示符类似的数据单元指示符。在一些实施例中，客户端 302-1 还与该附加信息一起发送计数器索引，或者其他计数器标识符，从而使得计数器逻辑 310 可以标识该附加信息所对应的计数器。例如，在一个实施例中，客户端 302-1 将计数器编号与每个计数事件指示符一起发送，从而使得接收该信息的计数器逻辑 310 可以响应于该指示符标识哪个计数器应当被增加。

[0058] 下面的表 2-5 以不同的表图示了计数器系统 300 的一个实施例的各种不同示例场景, 其中这些不同的表对应于来自客户端 302-1 至 302-N 中的不同客户端的请求, 或者对应于来自客户端 302-1 至 302-N 中的单个客户端在不同时间的请求。在对应于表 2-5 的场景的实施例中, 基于每个请求所确定的计数器参数包括指示每个所请求的计数器是数据单元(例如, 包或元)计数器还是字节计数器的第一计数器类型, 指示每个所请求的计数器是精确计数器还是统计计数器的第二计数器类型, 以及指定二进制计数器比特大小(对于每个所请求的精确计数器), 或者尾数比特大小和指数比特大小两者(对于每个所请求的统计计数器)的比特分配。在表 2-5 的每一个表中还示出了当使用对应的计数器参数时所形成的最大计数值。

[0059] 此外, 对应于表 2-5 的实施例是如下实施例, 其中提交请求的客户端被分配计数器资源块 306 中的单个块, 并且其中每个存储器 312 可以存储多达 72 个比特以用来表示由客户端所请求的所有计数器的计数器值。然而, 在其他实施例中, 单个客户端 302 可被分配多于一个的计数器资源块 306, 和 / 或每个计数器资源块 306 包括具有多于或少于 72 个比特的存储器 312。在各种实施例中, 每个计数器资源块 306 可以支持最大 2 至 8 个计数器。在其他实施例中, 每个计数器资源块 306 可以支持多于 8 个计数器。

[0060] 在第一示例场景中, 客户端 302-1 至 302-N 中的一个客户端请求一个精确数据单元计数器和一个精确字节计数器:

[0061]

计数器编号(Contr#)	计数器参数					最大值
	测量量	精确/统计	二进制计数器大小(比特)	M 大小(比特)	E 大小(比特)	
1	数据单元	精确	33	N/A	N/A	$2^{33}-1$
2	字节	精确	39	N/A	N/A	$2^{39}-1$

[0062] 表 2

[0063] 在第二示例场景中, 客户端 302-1 至 302-N 中的一个客户端请求 3 个精确包计数器:

[0064]

计数器编号(Contr#)	计数器参数					最大值
	测量量	精确/统计	二进制计数器大小(比特)	M 大小(比特)	E 大小(比特)	
1	数据单元	精确	24	N/A	N/A	$2^{24}-1$

[0065]

2	数据单元	精确	24	N/A	N/A	$2^{24}-1$
3	数据单元	精确	24	N/A	N/A	$2^{24}-1$

[0066] 表 3

[0067] 由于存储器 312 的比特预算固定在 72 个比特,因此这 3 个数据单元计数器无法平均地专用于与表 2 的第一示例场景相关的计数器值一样多的比特,其转而限制表 3 中所示的最大计数器值。

[0068] 在第三示例场景中,客户端 302-1 至 302-N 中的一个客户端请求两个统计数据单元计数器和两个统计字节计数器:

[0069]

计数器编号(Contr#)	计数器参数					最大值
	测量量	精确/统计	二进制计数器大小(比特)	M 大小(比特)	E 大小(比特)	
1	数据单元	统计	N/A	14	4	$2^{30}$
2	数据单元	统计	N/A	14	4	$2^{30}$
3	字节	统计	N/A	14	4	$2^{30}$
4	字节	统计	N/A	14	4	$2^{30}$

[0070] 表 4

[0071] 如表 4 中所示,与表 2 和表 3 的精确计数器相比,每个所请求的计数器具有高得多的最大计数器值,尽管事实上是在表 4 中请求了更高数目的计数器。然而,与用精确计数器应达到的水平相比,这种配置导致每个计数器的较低水平的计数器准确度和精度。

[0072] 在第四示例场景中,客户端 302-1 至 302-N 中的一个客户端请求三个统计数据单元计数器和三个统计字节计数器:

[0073]

计数器编号(Contr#)	计数器参数					最大值
	测量量	精确/统计	二进制计数器大小(比特)	M 大小(比特)	E 大小(比特)	
1	数据单元	统计	N/A	9	3	$2^{17}$
2	数据单元	统计	N/A	9	3	$2^{17}$
3	数据单元	统计	N/A	9	3	$2^{17}$

[0074]

4	字节	统计	N/A	9	3	$2^{17}$
5	字节	统计	N/A	9	3	$2^{17}$
6	字节	统计	N/A	9	3	$2^{17}$

[0075] 表 5

[0076] 如表 5 中所示, 每个所请求的计数器仍然具有非常高的最大计数器值, 尽管事实上是请求了相对大数目的计数器。然而, 与表 4 中示出的配置相比, 该配置导致了每个计数器更高级别的不准确度和不精确度, 这是因为尾数比特的数目较小。

[0077] 如上表 2-5 的示例实施例和场景中所示, 在分配固定数目的计数器资源 (例如, 存储器的比特) 给多个计数器时可以提供大程度的灵活性, 伴随着在计数器的数目、每个计数器的准确度或精度以及每个计数器的最大计数器值之间的许多可能的折衷。因此, 仅在要求 100% 或非常接近 100% 准确度或精度时, 可通过请求精确计数器来使得计数器资源被更高效地利用。例如, 为检测“肥流”或“大象流”(即, 传递大量信息的流) 的应用请求计数器的客户端可以仅需要对给定时间帧中包的数目的大致近似值。如果是这样, 该客户端可请求具有高最大值 (由于指数比特的数目大) 但是准确度小于 100% 的统计计数器。可接受统计计数器的其他应用包括生成特定类型的直方图的数据的应用, 该数据位诸如队列填充级别 (例如, 为了延迟估计), 或设备内先入先出 (FIFO) 缓冲器级别 (例如, 为了基于专用集成电路 (ASIC) 中的统计现象来评估结构设计)。其他应用可能需要精确计数器, 或者统计计数器中的尾数比特的数目高的惊人以具有非常高准确度的计数器。例如, 可能需要精确计数器的应用包括用于调试应用的丢包计数器, 其中预期仅有少量包可能被丢弃。

[0078] 图 6 是根据一个实施例的配置网络设备中的计数器资源的示例方法 350 的流程图。在一个实施例中, 方法 350 由图 4 的控制器 324 和计数器资源 304 来实施。

[0079] 在块 352, 使用第一计数器提供对第一多个数据单元内的第一量的精确计数。在一个实施例中, 第一量类似于上述的计数量 q, 并且是任何合适的单元 (例如, 包, 元, 字节的块, 等等)。在一个实施例中, 第一多个数据单元对应于特定流内的数据单元。在块 354, 第一量的精确计数在存储器中被存储为二进制数据。

[0080] 在块 356, 使用第二计数器提供对第二多个数据单元内的第二量的估计的计数。在一个实施例中, 第二量类似于以上参考图 1 描述的计数量 q, 并且是任何合适的单元 (例如, 包, 元, 字节的块, 等等)。在一些场景中, 第二量不同于第一量。在各种实施例中, 第二计数器被布置为根据参考图 1 描述的方法和 / 或图 3 的过程 (或等效过程) 来操作。在一个实施例中, 第二多个数据单元对应于特定流内的数据单元, 其中该流与使用第一计数器监视的流相同或者不同。在块 360, 第二量的估计的计数在存储器中被存储为表示尾数值 M 的 m 个尾数比特和表示指数值 E 的 e 个指数比特。

[0081] 在块 362, 响应于对所需计数器精度的一个或多个指示, 第一计数器被采用以用于在第一应用中使用, 并且第二计数器被采用以用于在第二应用中使用。例如, 在一个实施例中, 第一计数器被采用以用于由图 4 的客户端 302-1 使用, 并且第二计数器被采用以用于由图 4 的客户端 302-2 使用。在一个实施例中, 所需计数器精度的指示是从请求计数器资源的客户端 / 应用接收的。在一个实施例和 / 或场景中, 与来自第二应用的对计数器资源的

第二请求相比,来自第一应用的对计数器资源的第一请求对应于较高级别的所需计数器精度。在一个实施例中,所需计数器精度的每个指示包括一个或多个计数器参数,诸如计数器类型(统计或精确),以及用于统计计数器的尾数比特的数目。

[0082] 在一些实施例中,图6中所示的块并不以所描述的顺序发生。例如,在一个实施例和/或场景中,第一计数器和第二计数器并不存在,直到适当的计数器逻辑被配置以支持这些计数器。在这样的实施例中,块362发生在块352、354、356和360之前。

[0083] 图7是根据一个实施例的配置网络设备中的计数器资源的另一个示例方法400的流程图。在一个实施例中,方法400全部或部分地由图4的控制器324来实施。

[0084] 在块410,接收到对计数器资源的请求。例如,在一个实施例中,该请求是从类似于图4的客户端302-1的客户端接收的。在一个实施例中,计数器参数的请求指定了每个所请求的计数器的计数器类型(统计或精确),以及对每个所请求的计数器的比特分配。在一个实施例中,对特定的所请求的计数器的比特分配指定了表示整数值的比特的数目(用于精确计数器),或者表示尾数值的尾数比特的数目和表示指数值的指数比特的数目(用于统计计数器)。在一些实施例中,计数器参数组还包括其他计数器参数,诸如每个所请求的计数器是否是“权重使能的”(即,计数器是否采用了类似于参考图1所描述的“权重”指示符)。在一个实施例中,如果计数器是字节计数器而不是数据单元计数器,则该计数器是权重使能的。

[0085] 在块420,响应于在块410接收到的请求,将计数资源与客户端相关联。在一个实施例中,类似于图4的计数器资源块306的一个或多个资源块,以基于客户端的需求(例如,所需计数器的数目和所需准确度或精度)所确定的块的数目,与该客户端相关联。在一个实施例中,在块420,至少部分地通过配置映射表来将资源与客户端相关联。

[0086] 在块430,在块420与客户端相关联的计数资源是根据在块410接收的请求来配置的。在一个实施例中,为此,方法400配置计数资源块内的逻辑以根据由请求所指定的计数器参数来操作。对于精确计数器,计数器逻辑被配置为每当发生计数事件时增加每个计数器值。对于统计计数器,计数器逻辑被配置为对每个计数事件计算增加概率P,并且以概率P增加尾数(以及指数值,如果尾数值滚计的话)。例如,在一个实施例中,对于每个统计计数器,计数器逻辑被配置为以类似于图1的计数器块10中的计数器的方式和/或根据图3的过程(或适合的等效的过程)来操作统计计数器。进一步地,在一个实施例中,在块430,用于每个计数器的适当数目的存储比特(为了满足在块410处接收的客户端请求中的计数器参数)被分配给每个计数器。在一些实施例中,在块430还基于存储器中的可用比特的数目来配置计数器。例如,在一个实施例中,方法400基于针对计数器资源的每个请求来确定“最佳”匹配,但是仍将配置比特分配,即使按照计数器类型、计数器准确度/精确度级别以及比特的可用数目不能完全支持请求的计数器参数。

[0087] 在块440,根据来自客户端的请求来安装流。根据不同实施例,该客户端请求是在块410接收的请求,或者单独的请求。在一个实施例中,一旦流被安装,则流中每个被检测到的包用来作为对应计数器的计数事件,类似于参考图1所描述的“计数事件”指示符。

[0088] 在一些实施例中,至少部分上述的计数技术是以硬件或固件来实施的。例如,在一个实施例中,至少图1的计数器10的计数器更新策略模块14和存储器16是以硬件实施的,从而允许计数器10实时且线速地对数据单元或字节进行计数。作为另一个示例,在一个实

施例中,图 4 的计数器系统 300(以及在一些实施例中,计数器矩阵连接 320)的计数器资源 304 是以硬件实施的。在一些实施例中,例如,块、操作、技术等是在客户集成电路 (IC)、专用集成电路 (ASIC)、可编程逻辑设备 (PLD)、现场可编程逻辑阵列 (FPGA)、可编程逻辑阵列 (PLA) 等中实施的。在一个实施例中,图 1 的控制器 324 是全部或部分以软件实施的。

[0089] 本公开内容的实施例可以在任何类型的在有线或无线通信系统中使用的网络设备中实施,例如,这些系统包括使用在包括或连接到有线或无线 LAN、或者有线或无线 WAN、因特网、基于电缆和卫星的通信系统(如因特网、数据、视频和语音通信系统)、无线电话系统(包括蜂窝电话系统,基于语音的因特网协议 (VoIP) 系统,基于家庭的无线电话系统)等中的网络设备。

[0090] 虽然本发明是参考特定示例来进行描述的,这仅旨在示例而并不是限制本发明,在不脱离本发明实质和范围的情况下可以对公开的实施例做出改变、增加和 / 或删除,这对于本领域普通技术人员来说是显然的。

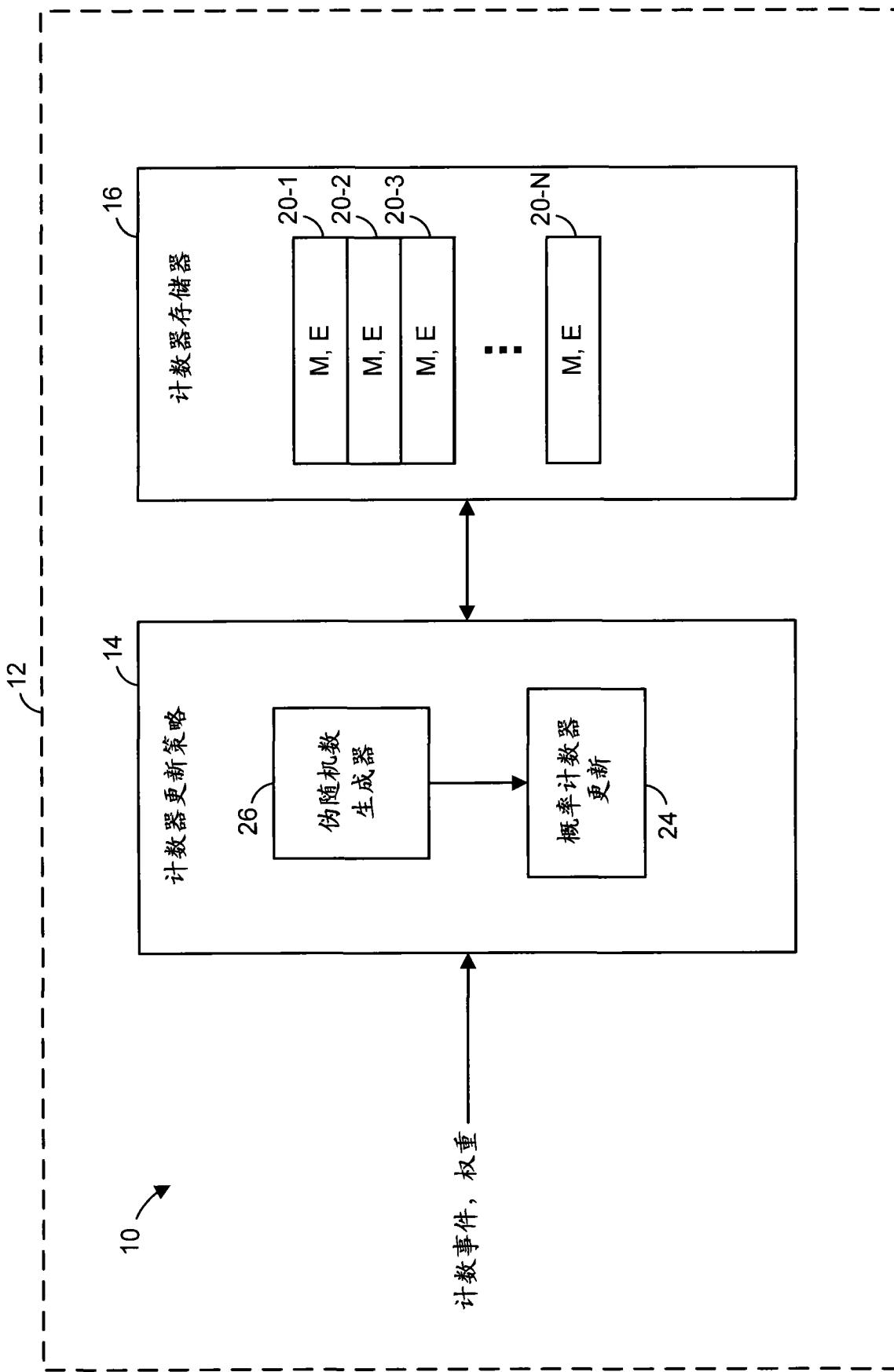


图 1

50

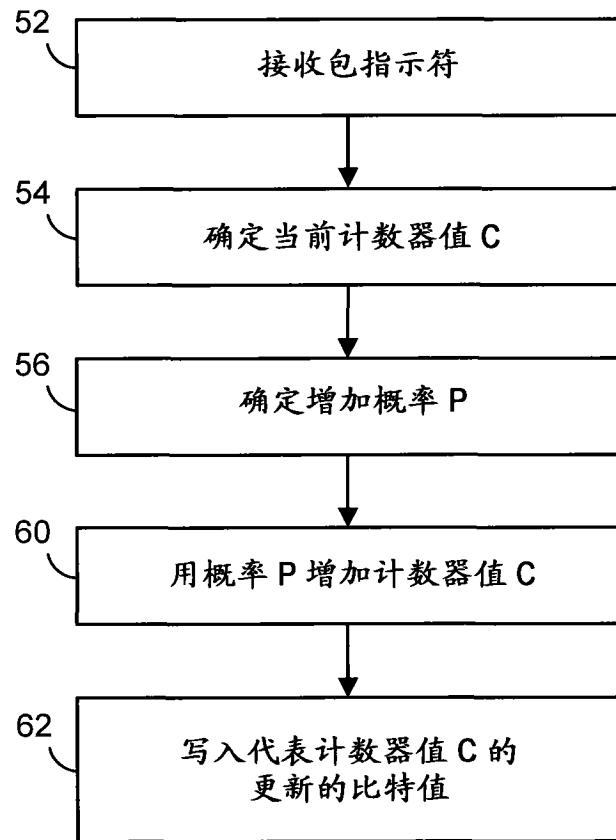
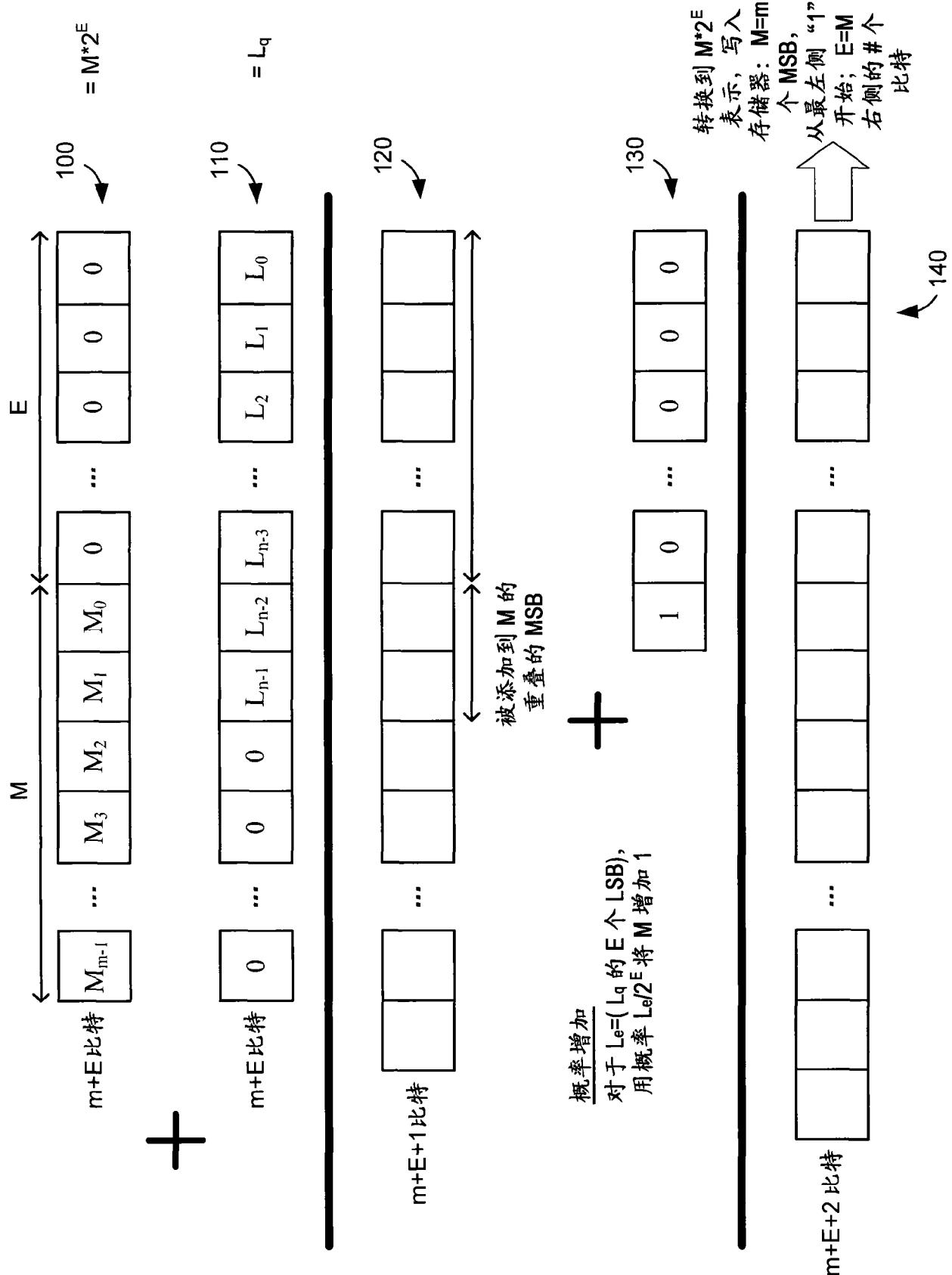
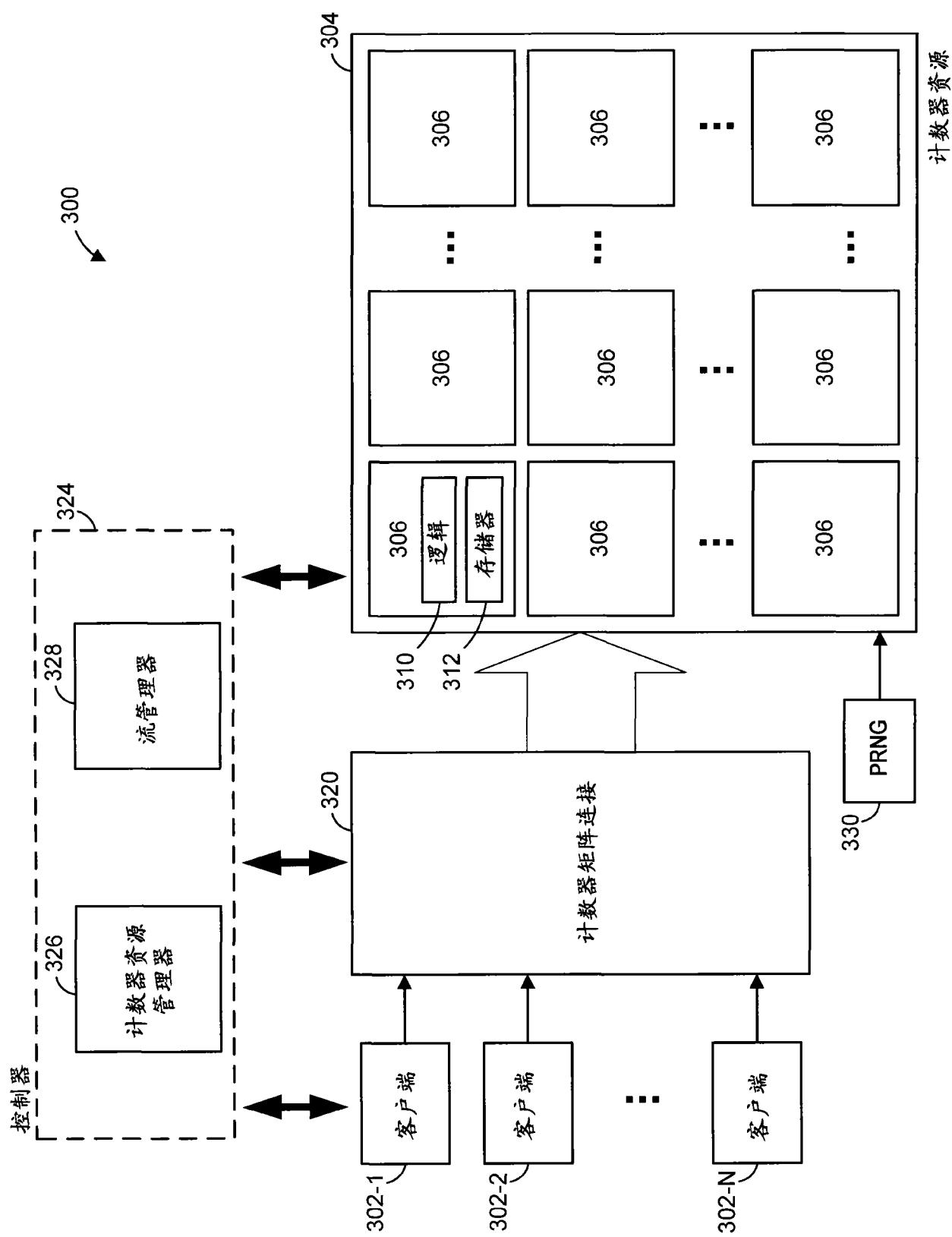


图 2





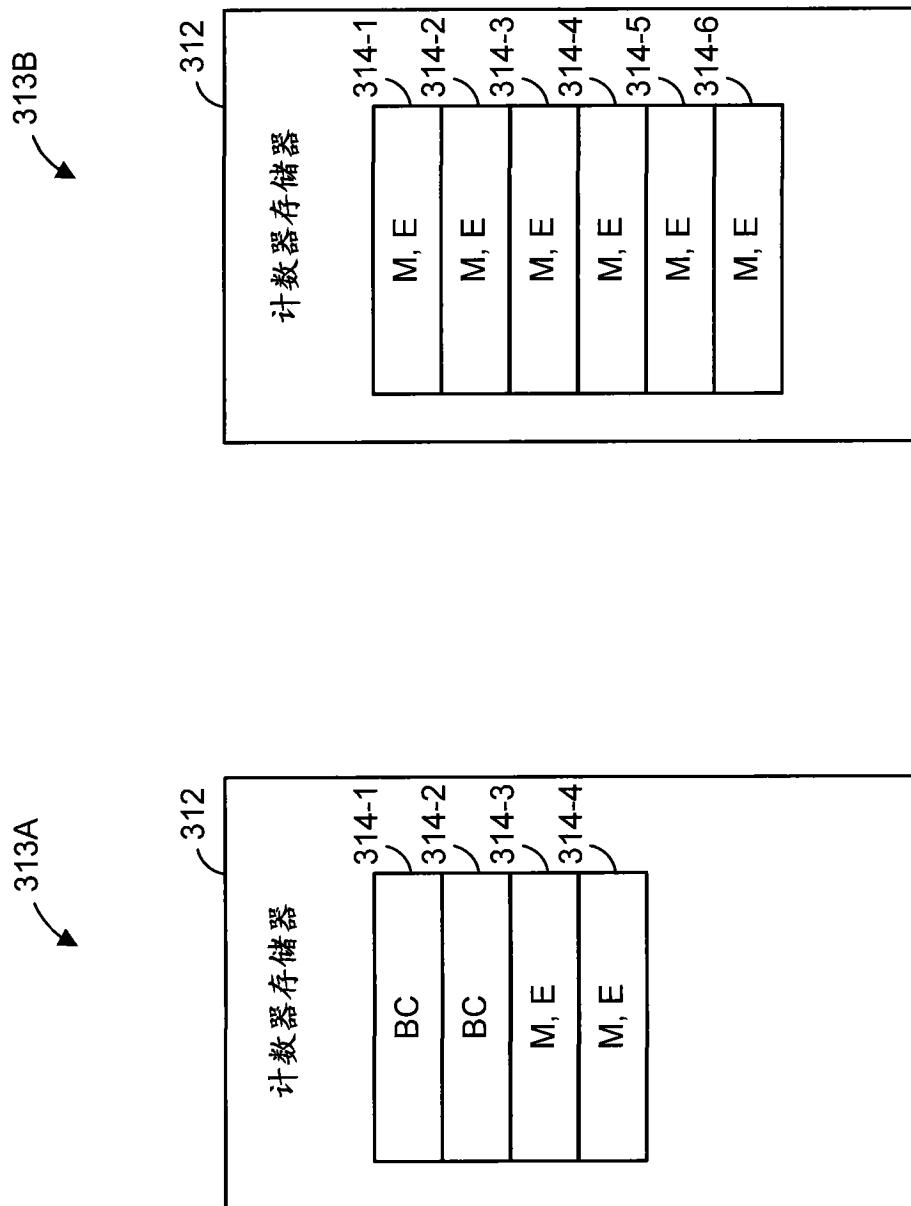


图 5

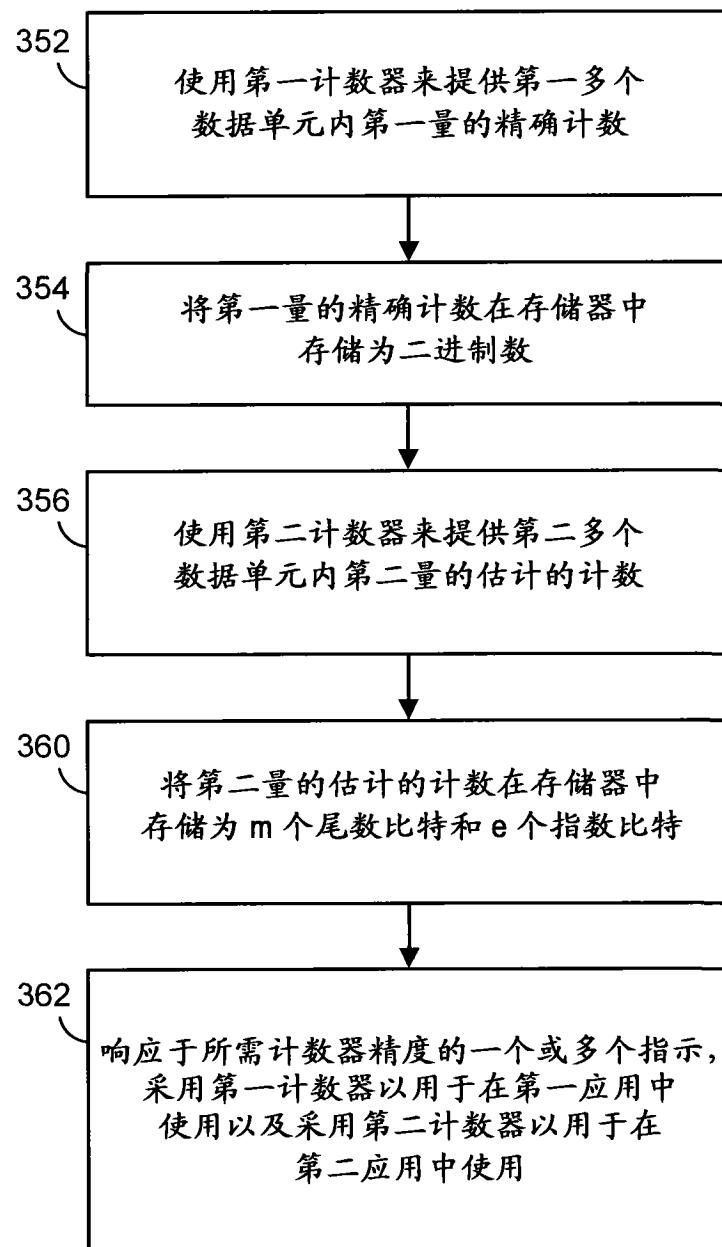
350

图 6

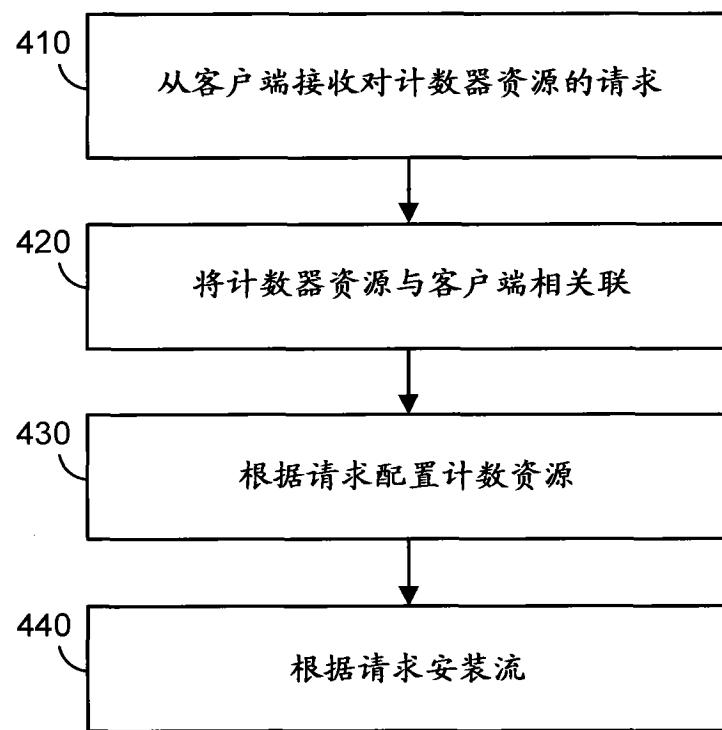
400

图 7