(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116545951 A
(43)申请公布日 2023.08.04

(21)申请号 202310093312.0
(22)申请日 2023.02.02
(30)优先权数据
63/306,079 2022.02.02 US
17/731,662 2022.04.28 US
(71)申请人 三星电子株式会社
地址 韩国京畿道
(72)发明人 V·K·阿格拉瓦尔
D·L·赫尔米克
C-C·C-J·A·吴
(74)专利代理机构 北京市柳沈律师事务所 11105
专利代理师 邵亚丽
(51)Int.Cl.
H04L 47/62(2022.01)
H04L 69/04(2022.01)

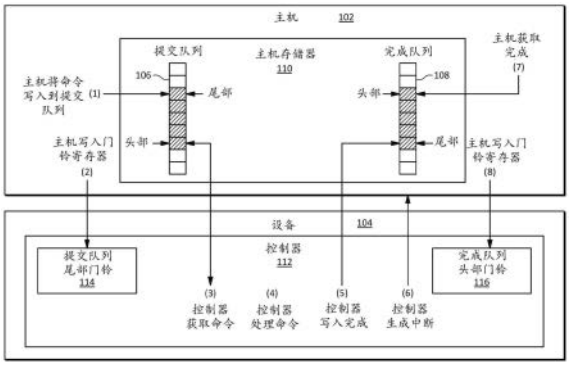
权利要求书2页 说明书24页 附图11页

(54)发明名称

用于队列条目监视的系统、方法和设备

(57)摘要

一种方法可以包括:在设备处接收基于提交给队列的条目的指示,基于该指示获得该条目的测量,对该测量进行编码以生成编码的测量,以及存储该编码的测量。编码可以包括增量编码、熵编码或游程编码中的一种或多种。该方法还可以包括基于队列的状态修改测量。队列的状态可以包括队列中的条目的数量,并且修改测量可以包括将测量设置为一个值。队列的状态可以包括空队列,并且修改测量可以包括重置测量。编码可以包括基于一个或多个参数进行编码。该方法还可以包括确定一个或多个参数中的至少一个。



CN 116545951 A

1.一种用于队列条目监视的方法,包括:

在设备处接收基于提交给队列的条目的指示;

基于所述指示,获得对所述条目的测量;

对所述测量进行编码以生成编码的测量;以及

存储所述编码的测量。

2.根据权利要求1所述的方法,其中,所述编码包括增量编码、熵编码或游程编码中的一个或多个。

3.根据权利要求1所述的方法,还包括:基于所述队列的状态修改所述测量。

4.根据权利要求3所述的方法,其中:

所述队列的状态包括队列中条目的数量;以及

修改所述测量包括将所述测量设置为一个值。

5.根据权利要求3所述的方法,其中:

所述队列的状态包括空队列;以及

修改所述测量包括重置所述测量。

6.根据权利要求1所述的方法,其中,所述编码包括基于一个或多个参数进行编码,所述方法还包括确定所述一个或多个参数中的至少一个。

7.根据权利要求6所述的方法,其中:

所述一个或多个参数包括一个或多个编码技术;以及

所述一个或多个参数包括用于所述一个或多个编码技术中的至少一个的一个或多个设置。

8.根据权利要求6所述的方法,还包括:

在所述设备处接收关于所述队列的一个或多个条目的信息;

其中,确定所述一个或多个参数中的至少一个是基于所述信息的至少一部分。

9.根据权利要求8所述的方法,其中,所述信息包括所述队列的条目的数量、所述队列的条目的频率、所述队列的条目的间距或所述队列的条目的一致性中的一个或多个。

10.根据权利要求6所述的方法,还包括:

确定一个或多个参数的集合的性能;

其中,确定一个或多个参数中的至少一个至少部分地基于所述一个或多个参数的集合的性能。

11.根据权利要求10所述的方法,其中,所述条目是第一条目,并且所述一个或多个参数的集合的性能基于提交给所述队列的第二条目。

12.根据权利要求6所述的方法,其中,所述指示是第一指示,所述测量是第一测量,并且所述条目是第一条目,所述方法还包括:

在所述设备处接收基于提交给所述队列的第二条目的第二指示;以及

基于第二指示,获得对于第二条目的第二测量;

其中,确定所述一个或多个参数中的至少一个参数是基于第一测量和第二测量。

13.根据权利要求1所述的方法,其中,所述测量包括时间戳。

14.根据权利要求1所述的方法,其中,存储所述编码的测量包括在所述设备处存储所述编码的测量的至少一部分。

15.根据权利要求1所述的方法,还包括:

确定所述条目的数据传递大小;

对所述数据传递大小进行编码,以生成编码的数据传递大小;以及

存储所述编码的数据传递大小。

16.根据权利要求1所述的方法,还包括:

从所述队列中获取条目;

基于所述获取来执行操作;以及

基于所述执行,存储所述测量。

17.一种用于队列条目监视的设备,包括:

控制器,被配置为:

接收提交给队列的条目的指示;

基于所述指示,获得对所述条目的测量;

对所述测量进行编码以生成编码的测量;以及

存储所述编码的测量。

18.根据权利要求17所述的设备,还包括指示逻辑,被配置为生成所述指示。

19.一种用于队列条目监视的系统,包括:

主机,被配置为向队列提交条目;以及

使用通信连接耦合到主机的设备,其中,所述设备被配置为:

接收提交给队列的条目的指示;

基于所述指示,获得对所述条目的测量;

对所述测量进行编码以生成编码的测量;以及

存储所述编码的测量。

20.根据权利要求19所述的系统,其中,所述主机被配置为向所述设备发送关于所述队列的一个或多个条目的信息。

用于队列条目监视的系统、方法和设备

[0001] 相关申请的交叉引用

[0002] 本申请要求于2022年2月2日提交的、序列号为63/306,079、名称为“Systems, Methods,and Devices for Command Age Tracking”的美国临时专利申请以及2022年4月28日提交的序列号为17/731,662、名称为“Systems,Methods,And Devices For Queue Entry Monitoring”的美国专利申请的权益,其通过引用并入本文。

技术领域

[0003] 本公开一般涉及队列管理,更具体地,涉及用于队列条目监视的系统、方法和设备。

背景技术

[0004] 通信协议可以使用一个或多个队列来存储诸如请求、命令、完成等的条目。例如,通信协议可以允许主机在提交(submission)队列中存储条目。设备可以从提交队列中获取条目,并且例如以先进先出的顺序处理这些条目。当一个或多个更早存储的条目被设备处理时,条目可能在提交队列中经历时间(age)。

[0005] 背景技术部分中公开的上述信息仅用于增强对发明原理的背景的理解,因此它可能包含不构成现有技术的信息。

发明内容

[0006] 一种方法可以包括:在设备处接收基于提交给队列的条目的指示,基于该指示获得该条目的测量,编码该测量以生成编码的测量,以及存储该编码的测量。编码可以包括增量编码、熵编码或游程编码中的一种或多种。该方法还可以包括基于队列的状态修改测量。队列的状态可以包括队列中条目的数量,并且修改测量可以包括将测量设置为一个值。队列的状态可以包括空队列,并且修改测量可以包括重置测量。编码可以包括基于一个或多个参数进行编码。该方法还可以包括确定一个或多个参数中的至少一个。一个或多个参数可以包括一个或多个编码技术。一个或多个参数可以包括对于一个或多个编码技术中的至少一个的一个或多个设置。该方法还可以包括在设备处接收关于队列的一个或多个条目的信息,其中确定一个或多个参数中的至少一个可以基于该信息的至少一部分。该信息可以由主机提供。该信息可以包括队列的条目的数量、队列的条目的频率、队列的条目的间隔或队列的条目的一致性中的一个或多个。该方法还可以包括确定一个或多个参数的集合的性能,其中确定一个或多个参数中的至少一个可以至少部分地基于该一个或多个参数的集合的性能。该性能可以包括编码的效率。确定性能可以包括在设备处确定性能。该条目可以是第一条目,并且该一个或多个参数的集合的性能可以基于提交给队列的第二条目。该指示可以是第一指示,该测量可以是第一测量,并且该条目可以是第一条目,该方法还可以包括在设备处接收基于提交给队列的第二条目的第二指示,并且基于第二指示获得对于第二条目的第二测量,其中确定一个或多个参数中的至少一个可以基于第一测量和第二测量。该

方法还可以包括在设备处接收基于提交给队列的第三条目的第三指示，以及基于第三指示获得对于第三条目的第三测量，其中确定一个或多个参数中的至少一个可以基于第三测量。一个或多个参数中的至少一个可以包括有损编码技术。有损编码技术可以包括对数编码技术。该队列可以包括用于通信协议的队列。用于通信协议的队列可以包括用于快速非易失性存储器(NVMe)的提交队列。该条目可以包括命令。该测量可以包括时间戳。对编码的测量的存储可以包括在设备处存储编码的测量的至少一部分。

[0007] 一种设备可以包括控制器，该控制器被配置为接收提交给队列的条目的指示，基于该指示获得该条目的测量，对该测量进行编码以生成编码的测量，以及存储该编码的测量。该设备还可以包括被配置为存储该编码的测量的存储器。该设备还可以包括被配置为基于该条目执行操作的设备功能电路。该设备还可以包括被配置为生成该指示的指示逻辑。指示逻辑可以包括门铃寄存器。

[0008] 一种系统可以包括被配置为向队列提交条目的主机，以及使用通信连接耦合到主机的设备，其中该设备可以被配置为接收提交到队列的条目的指示，基于该指示获得条目的测量，编码该测量以生成编码的测量，以及存储该编码的测量。主机可以被配置为使用通信连接向设备发送指示。主机可以被配置为通过访问门铃寄存器来发送指示。主机可以被配置为向设备发送关于队列的一个或多个条目的信息。该信息可以包括队列的条目的数量、队列的条目的频率、队列的条目的间隔或队列的条目的一致性中的一个或多个。

[0009] 一种方法可以包括：在设备处确定提交给队列的条目的数据传递大小，在设备处确定队列的队列深度，以及在设备处基于数据传递大小和队列深度确定队列的工作负荷。该方法还可以包括从设备发送关于队列的工作负荷的信息。该方法还可以包括基于队列的工作负荷来确定超时。超时可以包括设备超时。条目可以是第一条目，队列可以是第一队列，数据传递大小可以是第一数据传递大小，并且队列深度可以是第一队列深度，该方法还可以包括在设备处确定提交给第二队列的第二条目的第二数据传递大小，在设备处确定第二队列的第二队列深度，以及在设备处基于第一数据传递大小、第二数据传递大小、第一队列深度和第二队列深度，确定设备的工作负荷。该方法还可以包括基于设备的工作负荷来确定超时。

[0010] 一种方法可以包括：在设备处接收基于提交给队列的条目的指示，基于该指示获得该条目的测量，确定该条目的数据传递大小，以及存储该测量和数据传递大小。该方法还可以包括从队列中获取条目，并基于获取来确定条目的数据传递大小。该方法还可以包括对测量和数据传递大小进行编码以生成编码的测量和数据传递大小，以及存储编码的测量和数据传递大小。该指示可以是第一指示，该条目可以是第一条目，并且数据传递大小可以是第一数据传递大小，该方法还可以包括在设备处接收基于提交给队列的第二条目的第二指示，基于第二指示获得第二条目的第二测量，确定第二条目的第二数据传递大小，以及累积第一数据传递大小和第二数据传递大小。

[0011] 一种方法可以包括：在设备处接收基于提交给队列的条目的指示，基于该指示获得对该条目的测量，从队列中获取该条目，基于该获取执行操作，以及基于该执行存储该测量。该方法还可以包括对测量进行编码以生成编码的测量，其中存储测量可以包括存储编码的测量。该方法还可以包括确定条目的数据传递大小，并且基于该执行来存储数据传递大小。

附图说明

[0012] 附图不一定是按比例绘制的，在所有附图中，出于说明的目的，类似结构或功能的元件通常可以用类似的附图标记或其部分来表示。附图仅仅是为了便于描述本文描述的各种实施例。附图没有描述本文公开的教导的每个方面，并且不限制权利要求的范围。为了防止附图变得模糊，不是所有的组件、连接等都可以被示出，并且不是所有的组件都具有附图标记。然而，从附图中可以容易地看出组件配置的模式。附图与说明书一起示出了本公开的示例实施例，并且与描述一起用于解释本公开的原理。

[0013] 图1示出了根据本公开的示例实施例的通信协议的队列方案的实施例。

[0014] 图2示出了根据本公开的示例实施例的基于获取命令来监视命令时长(age)的方案的实施例。

[0015] 图3示出了根据本公开的示例实施例的基于向队列提交命令来监视命令时长的方案的实施例。

[0016] 图4示出了根据本公开的示例实施例的具有带有编码的队列监视的系统的实施例。

[0017] 图5示出了根据本公开的示例实施例的用于队列条目的时间戳的增量编码的示例实施例。

[0018] 图6示出了根据本公开的示例实施例的用于队列条目的示例序列的霍夫曼编码的示例实施例。

[0019] 图7示出了根据本公开的示例实施例的用于队列条目的示例序列的熵编码结合游程编码的第一示例实施例。

[0020] 图8示出了根据本公开的示例实施例的用于队列条目的示例序列的熵编码结合游程编码的第二示例实施例。

[0021] 图9示出了根据本公开的示例实施例的具有带有编码的队列监视的系统的示例实施例。

[0022] 图10示出了根据本公开的示例实施例的具有带有调试的队列监视的系统的示例实施例。

[0023] 图11示出了根据本公开的示例实施例的具有带有记录保存的队列监视的系统的示例实施例。

[0024] 图12示出了根据本公开示例实施例的主机装置的示例实施例。

[0025] 图13示出了根据本公开示例实施例的设备的示例实施例。

[0026] 图14示出了根据本公开的示例实施例的用于监视一个或多个队列条目的方法的实施例。

具体实施方式

[0027] 主机可以在提交队列中存储诸如请求、命令等的条目。控制器和/或设备可以从提交队列中获取条目，并例如以先进先出的顺序处理条目。例如，监视队列中的一个或多个条目对于对队列和/或队列可以在其中操作的系统进行管理、调试、剖析、评估等可能是有用的。例如，监视一个或多个条目已经在队列中多长时间(例如，一个或多个条目的时长(age))对于对具有一个或多个队列的系统中的超时进行分析、预测、检测、防止、报告等可

能是有用的。

[0028] 确定提交队列中的一个或多个条目的时长可以涉及存储一个或多个条目的测量，诸如时间戳。然而，取决于实现方式细节，存储队列条目的测量可能会消耗设备中相对大量的存储器或其他位置。

[0029] 本公开包含与监视队列条目相关的许多发明原理。本文公开的原理可以具有独立的效用，并且可以单独实施，并且不是每个实施例都可以利用每个原理。此外，这些原理还可以以各种组合来实施，其中的一些组合可以以协同的方式放大单个原理的一些益处。

[0030] 本文公开的一些原理涉及将一种或多种编码方案用于与队列监视相关的信息。例如，一些实施例可以使用增量编码、熵编码、游程编码、有损编码等中的一种或多种形式，或其一种或多种组合，来编码一个或多个测量，诸如队列中条目的时间戳。根据实现方式细节，这可以压缩测量和/或减少存储测量所涉及的存储器的量。

[0031] 本文公开的一些原理涉及用于确定用于对与队列监视相关的信息进行编码的一个或多个参数的方案。例如，在一些实施例中，静态方案可以(例如，从主机)接收与队列中可以(例如，由主机)提交的一个或多个条目模式相关的信息。基于该信息，该方案可以确定用于编码与队列监视相关的信息的一个或多个参数(例如，一个或多个编码技术、用于编码技术的设置等)。作为另一个示例，在一些实施例中，动态方案可以确定一个或多个编码参数的集合的性能。基于该性能，动态方案可以调整编码参数中的一个或多个。

[0032] 本文公开的一些原理涉及用于对队列和/或队列可以在其中操作的系统中的条目进行管理、调试、剖析、评估等的方案。例如，在一些实施例中，方案可以收集和/或存储与队列操作相关的数据。在一些实施例中，该方案可以使用所收集的数据来基于该数据确定一个或多个工作负荷，和/或对具有一个或多个队列的系统中的超时进行分析、预测、检测、防止、报告等。在一些实施例中，可以对所收集的数据进行编码，取决于实现方式细节，这可以压缩数据和/或减少存储数据所涉及的存储器的量。

[0033] 本文公开的一些原理涉及用于保存与队列和/或队列可以在其中操作的系统中的条目相关的记录的方案。例如，在一些实施例中，方案可以收集和/或存储与一个或多个条目所传递的一个或多个数据量相关的数据、与一个或多个条目相关的一个或多个时间戳或其他测量等。在一些实施例中，收集的和/或存储的数据可以用于对超时进行例如分析、预测、检测、防止、报告等。

[0034] 为了说明的目的，一些实施例可以在一些具体实现方式细节的上下文中描述。然而，这些原理不限于这些或任何其他实施细节。

[0035] 图1示出了根据本公开的示例实施例的通信协议的队列方案的实施例。例如，图1所示的实施例可以与诸如非易失性快速存储器(NVMe)的存储协议一起使用，所述存储协议可以使用诸如外围组件快速互连(PCIe)和/或可以使用诸如以太网的网络的NVMe-over-fabric(NVMe-oF)的互连，但是原理不限于这些协议、通信技术或任何其他实现方式细节。

[0036] 图1所示的实施例可以包括主机102和设备104。主机可以包括位于例如主机存储器110中的提交队列(SQ)106和完成队列(CQ)108。在其他实施例中，提交队列106和/或完成队列108可以位于设备104和/或任何其他位置。设备104可以包括控制器112，控制器112可以包括提交队列尾部门铃寄存器114和/或完成队列头部门铃寄存器116。

[0037] 在一些实施例中，提交队列106和/或完成队列108可以用于例如使设备能够接收

和/或处理来自主机102的一个或多个命令。提交队列106和/或完成队列108可以被实现为例如循环先入先出(FIFO)队列,其中队列的一端可以在逻辑上绕到队列的另一端,以使得条目能够被无限地添加到队列中和从队列中移除(在一些实施例中,受制于最大条目数),即使该队列可以用有限的线性地址空间来实现。参考提交队列106或完成队列108,具有最陈旧的(oldest)未被获取的条目(例如,命令或完成)的插槽(slot)可以被称为头部,下一个可用的未被占用的插槽可以被称为尾部。

[0038] 用于接收和/或处理来自主机102的一个或多个命令的方法的示例实施例可以如下进行。

[0039] 在操作(1),主机102可以从由尾部指针指向的插槽处开始将一个或多个命令放置(例如,写入)提交队列106中的一个或多个插槽中(例如,每插槽一个命令),如图1所示。(在一些实施例中,将条目放置在队列中可以被称为将条目提交到队列中、将条目调度到队列中和/或将条目更新到队列中)。提交队列106的尾部指针然后可以被更新以指向下一个可用的位置。

[0040] 在操作(2),主机102还可以更新(例如,写入)提交队列尾部门铃寄存器(SQ-TDB)114,以发起可以向控制器112通知一个或多个新命令已经被放置在提交队列106中的过程。例如,主机102可以向提交队列尾部门铃寄存器114写入提交队列尾部条目指针的新的值。

[0041] 在一些实施例中,可以用硬件、软件或其组合来监视提交队列尾部门铃寄存器114,以向控制器112提供一个或多个新命令已经被放置在提交队列106中的指示。例如,在一些实施例中,提交队列尾部门铃寄存器114可以被实现为硬件监视的寄存器或存储器位置(例如,诸如控制器112和/或设备104处的PCIe位置的位置),其可以基于寄存器114的更新为设备104生成中断。在一些实施例中,中断可以用作对控制器112和/或设备104的关于一个或多个新命令已经被放置在提交队列106中的指示。

[0042] 在一些实施例中,接收一个或多个新命令被放置在提交队列106中的指示可以使得控制器112能够跟踪提交队列106中可以存在的未被获取的和/或未被处理的命令的数量。在一些实施例中,该信息可以用于例如命令仲裁(arbitration)过程,该命令仲裁过程可以使控制器112能够确定控制器112可以从哪个提交队列(如果有多个提交队列)获取一个或多个命令。

[0043] 在操作(3),控制器112可以从由头部条目指针指向的位置开始从提交队列106中获取(例如,通过读取)一个或多个命令。然后,可以更新头部条目指针以指向提交队列106中的下一个(例如,最陈旧的)未被获取的命令。

[0044] 在一些实施例中,图1所示的方案可以实现一种机制,以使主机102能够跟踪提交队列106的头部的位置。在一些实施例中,这种机制可以基本上实现为提交队列头部门铃寄存器,例如,使用PCIe基地址寄存器。主机102可以使用该信息,例如,用于绕过(wrap)提交队列106,以防止提交队列106的头部和尾部之间的冲突。

[0045] 在操作(4),控制器112可以处理它从提交队列106中获取的一个或多个命令。在一些实施例中,控制器112可以无序地处理一个或多个命令。在一些实施例中,获取和/或处理可以被称为消耗。

[0046] 在操作(5),控制器112可以从由例如可以由完成队列尾部条目指针所指向的下一个可用的插槽处开始,将对应于一个或多个经处理的命令的一个或多个完成放置在完成队

列108中，如由如图1所示的。完成队列尾部条目指针然后可以被更新以指向完成队列108中的下一个可用的插槽。在一些实施例中，完成可以包括可以从先前条目反转(invert)的阶段标签，例如，以向主机102指示完成队列条目(例如，新的完成)是可用于处理的新条目。

[0047] 在操作(6)，控制器112和/或设备104可以为主机102生成中断(例如，基于管脚的中断、消息通知的中断(message signaled interrupt，MSI)、扩展的MSI中断(MSI-X)等)，以向主机102指示一个或多个完成已经被添加到完成队列108。在一些实施例中，针对一个或多个完成的一个或多个中断可以被合并成更少数量的中断。

[0048] 在操作(7)，主机102可以从头部处开始从完成队列108获取(例如，通过读取)一个或多个完成，该头部可以例如由完成队列头部条目指针指向。在一些实施例中，主机102可以处理它从完成队列108中获取的一个或多个完成。在一些实施例中，主机102可以继续获取和/或处理完成，例如，直到它遇到具有从先前获取的完成的阶段标签反转的阶段标签的完成。

[0049] 在操作(8)，主机102可以更新(例如，写入)提交队列头部门铃寄存器(CQ-HDB)116，以发起可以向控制器112通知一个或多个完成已经(例如，通过从完成队列108中读取一个或多个完成)从完成队列108中移除(例如，从完成队列108中释放)的过程。例如，主机102可以向完成队列头部门铃寄存器116写入完成队列头部条目指针的新的值。在一些实施例中，主机102可以在更新相关联的完成队列头部门铃寄存器116之前获取和/或处理一个或多个完成。

[0050] 图2示出了根据本公开的示例实施例的基于获取命令来监视命令时长的方案的实施例。例如，图2所示的方案可以由设备处的控制器(例如，图1所示的设备104处的控制器112)来实现。控制器可以跟踪提交队列(例如，图1所示的提交队列106)中的一个或多个命令的从命令从提交队列被获取的时间处开始的时长。

[0051] 图2所示的实施例可以包括时间戳生成器218和时间戳表220。时间戳生成器218可以基于诸如命令获取222、命令完成224等的事件来提供时间戳(例如，在主机、设备等处的系统时间的指示)。例如，控制器可以从提交队列获取第一命令，如命令获取事件222所示。基于命令获取事件222，时间戳生成器218可以生成第一获取时间戳tsf0，其可以指示获取命令的时间。如图2所示，控制器可以将时间戳tsf0放置在时间戳表220的第一条目中。控制器可以使用时间戳生成器218来生成对应于从提交队列获取的一个或多个附加命令的一个或多个附加获取时间戳tsf1、tsf2、…、tsfm。

[0052] 当命令完成时(例如，在命令完成事件224处)，时间戳生成器218可以生成与命令完成的时间对应的完成时间戳tsc。可以通过从相应的完成时间戳中减去获取时间戳tsf来确定命令完成时的命令的时长，如下所示：命令时长＝tsc-tsf。例如，第一命令在完成时的时长可以由tsc0-tsf0来确定。在一些实施例中，即使命令尚未完成，该公式也可用于量化当前命令时长。该当前命令时长可用于例如信息(例如，用于调试)和/或发起替代命令处理(例如，由于命令超时而使命令失败)。

[0053] 在一些实施例中，可以由设备处的控制器获取的命令的数量可以受到设备处的控制器容量的限制，取决于实现方式细节，该控制器容量可以小于主机可以访问的多个队列中的条目的总数。例如，在一些实施例中，在设备的控制器中最初处理的来自一个或多个队列的条目的数量可以是1024-2048个条目，而一个或多个主机可能已经通过多个队列提交

了数百万个未完成的提交队列条目(SQE)。

[0054] 如上所述,设备处的控制器可以从命令从提交队列中被获取的时间处开始,跟踪设备的提交队列中的一个或多个命令的时长。然而,从主机(例如,图1所示的主机102,和/或主机上的驱动程序、主机上运行的应用、主机上运行的服务等)的角度来看,在主机将命令写入提交队列时,命令可以开始经历时间(age)。例如,在一些实施例中(例如,在NVMe实现方式中),可以从命令被插入提交队列的时间直到当前时间测量命令时长(例如,为了确定命令超时)。

[0055] 例如,主机可以使用命令的时长来检测超时条件。在一些实施例中,如果主机向设备的提交队列写入命令,并且设备在超时时段(例如,预定时段)内没有指示命令已经完成(例如,通过将相应的完成放入完成队列中),则可以发生超时。在命令超时的情况下,主机可以向提交队列重新提交命令。如果主机在特定时间帧内检测到一定数量的超时,则主机可以清除提交队列(例如,通过重置设备)以努力确定是否存在系统配置问题、设备是否可能出现故障等。在这种情况下,调试底层问题可能是有益的。

[0056] 在一些实施例中,将一个或多个超时确定卸载给设备可能是有益的,例如,因为设备和/或设备处的控制器可以访问对调试超时的一个或多个原因(例如,根本原因)有用的信息。然而,如果设备和/或控制器直到获取命令才开始监视提交队列中的命令的时长,则由主机确定的命令的时长和由设备和/或控制器确定的命令的时长之间可能存在差异。例如,如果主机将命令写入提交队列,则在控制器获取命令之前,该命令可能会在队列中等待相当长的时间。因此,在一些实施例中,设备和/或控制器可能无法从主机的角度监视命令的时长(例如,端到端命令时长)。根据实现方式细节,这可能会使调试超时事件的原因变得困难。

[0057] 图3示出了根据本公开的示例实施例的基于向队列提交命令来监视命令时长的方案的实施例。例如,图3所示的方案可以由设备处的控制器(例如,图1所示的设备104处的控制器112)来实现。

[0058] 在图3所示的实施例中,控制器可以跟踪提交队列(例如,图1所示的提交队列106)中的一个或多个命令的时长,该时长从命令被写入提交队列的时间处开始。

[0059] 图3所示的实施例可以包括时间戳生成器318和时间戳表320。时间戳生成器318可以类似于图2所示的时间戳生成器218。图3所示的时间戳表320可以类似于图2所示的时间戳表220,然而,在一些实施例中,图3所示的时间戳表320可以包括对应于一个或多个附加提交队列SQ0,SQ1,…,SQn的一个或多个附加列。在一些实施例中,时间戳表320中的一些或全部可以位于设备处的一个或多个存储器中,例如,位于一个或多个设备存储器、一个或多个控制器存储器(例如,可以与控制器分离、嵌入控制器中的一个或多个控制器存储器,或其组合)等,或其组合中。作为另一个示例,在一些实施例中,时间戳表320中的一些或全部可以位于主机处的一个或多个存储器中。作为另一个示例,时间戳表320中的一些或全部可以位于主机处的一个或多个存储器、设备处的一个或多个存储器等的组合中。

[0060] 时间戳生成器318可以基于诸如命令提交326、命令完成324等的事件来提供时间戳(例如,主机、设备等处的系统时间的指示)。例如,控制器可以接收命令已经被写入第一提交队列SQ0的指示(例如,通过如图1所示的提交队列尾部门铃寄存器114)。基于命令提交(例如,命令提交事件326),控制器可以使用时间戳生成器318来生成第一提交时间戳

tss00,其可以指示提交命令的时间。控制器可以将第一提交时间戳tss00放置在时间戳表320的用于与第一提交队列SQ0相关联的列的第一条目中,如图3所示。控制器可以使用时间戳生成器318来生成对应于提交给第一提交队列SQ0的附加命令的一个或多个附加提交时间戳tss01、tss02、…、tss0m和/或用于其他提交队列SQ1、SQ2、…、SQn的一个或多个附加提交时间戳tsskm。

[0061] 当命令完成时(例如,在命令完成事件324处),控制器可以使用时间戳生成器318来生成对应于命令完成的时间的完成时间戳tsc。可以通过从相应的完成时间戳中减去提交时间戳tss来确定命令完成时命令的时长(例如,从命令被提交到提交队列的时间开始的端到端时长(例如,经过的时间)),如下所示:命令时长=tsc-tss。例如,第一提交队列SQ0中的第一命令在完成时的时长可以由tsc00-tss00来确定。

[0062] 根据本公开的示例实施例,可以在基于向队列提交命令来监视命令时长的方案中实现一个或多个附加特征。例如,在一些实施例中,命令时长可以包括通过例如主机的一个或多个部分的一个或多个延迟、指示(例如门铃、提交队列条目等)的链路传输的一个或多个延迟、命令解析延迟之前的一个或多个设备接口延迟等。作为另一个示例,如上文关于图1所示的实施例所述,接收一个或多个新命令被放置在提交队列106中的指示可以使控制器112能够跟踪提交队列106中可以存在的未被获取的和/或未被处理的命令的数量。在一些实施例中,这也可以使控制器能够为未被获取的和/或未处理的命令启动定时器。附加地或替代地,在一些实施例中,这样的定时器可以稍后被添加和/或合并到其中控制器了解关于命令的更多信息的命令解析信息中。取决于实现方式细节,来自队列的命令的顺序处理和关于提交队列条目的突发(burst)信息可以允许较早启动的定时器与正确的提交队列条目在被放入(pull in)时鲁棒地对准。因此,一些实施例可以在知道命令和/或命令的任何细节之前开始跟踪每个命令的时间。

[0063] 在一些实施例中(例如,用NVMe实现的实施例),并且取决于实现方式细节,诸如提交队列的队列可以包括多达64K个条目(并且因此每个提交队列多达64K个时间戳),以及多达1024个队列。在一些实施例中,时间戳可以包括64比特数据。因此,在一些实施例中,并且取决于实现方式细节,存储1024个提交队列的时间戳,每个提交队列具有64K个可能的命令,可能涉及每个提交队列使用512KB的存储器和/或控制器使用512MB的存储器。根据实现方式细节,这可能是用于时间戳的禁止性存储器的量。

[0064] 图4示出了根据本公开的示例实施例的具有带有编码的队列监视的系统的实施例。图4所示的实施例可以包括主机402和设备404。主机402可以包括第一通信接口428。设备404可以包括第二通信接口430和控制器434。第一通信接口428和第二通信接口430可以通过通信连接432进行通信。

[0065] 设备404可以使用控制器434来使用通信协议与主机402通信,该通信协议可以使用一个或多个队列440来在主机402和设备404之间交换请求、命令、完成等。在一些实施例中,一个或多个队列440可以位于主机402处,如图4中的虚线轮廓所示。附加地或可替换地,一个或多个队列440可以位于设备404或任何其他位置。在一些实施例中,一个或多个队列440可以部分地位于主机402,部分地位于设备404,和/或部分地位于任何其他位置。

[0066] 在一些实施例中,可以向控制器434提供主机402已经向一个或多个队列440中的至少一个提交了条目的指示436。在一些实施例中,控制器434处的编码逻辑438可以接收指

示436,并基于该指示获得条目的测量(例如,时间戳)。编码逻辑438可以对测量进行编码,以生成编码的测量。在一些实施例中,控制器434可以例如在设备404、主机402和/或任何其他位置存储编码的测量。

[0067] 在一些实施例中,编码逻辑438可以使控制器434能够以类似于图3所示的方式基于向队列提交命令来实现用于监视命令时长的方案。例如,在一些实施例中,指示436可以使控制器434能够监视一个或多个队列条目在条目被提交到队列时开始的时长(例如,端到端命令时长)。

[0068] 在一些实施例中,编码逻辑438可以使用一种或多种编码形式(诸如增量编码、熵编码、游程编码、有损编码等,或者它们的一种或多种组合),对提交给一个或多个队列440的一个或多个条目的一个或多个测量进行编码,其中一些将在下面更详细地描述。取决于实现方式细节,编码可以压缩一个或多个测量和/或减少存储一个或多个编码的测量所涉及的存储器的量。

[0069] 尽管图4所示的系统不限于任何特定的实现方式细节,但是在一些实施例中,主机402和设备404可以实现类似于图1所示的队列方案,例如,使用NVMe协议,其中一个或多个队列440可以实现一个或多个提交队列和/或完成队列。在这样的实施例中,指示436可以例如由门铃机制(例如,提交队列尾部门铃寄存器)来提供。

[0070] 主机402可以用可以被配置为主机的任何类型的装置来实现,例如包括诸如计算服务器、存储服务器、存储节点、网络服务器等的服务器,诸如工作站、个人计算机、平板电脑、智能手机等的计算机,和/或其任意组合。设备404可以用任何类型的可以被配置为设备的装置来实现,包括例如加速器设备、存储设备、网络设备、存储器扩展和/或缓冲器设备、图形处理单元(GPU)、神经处理单元(NPU)、张量处理单元(TPU)等、或其任意组合的设备。

[0071] 第一通信接口428、第二通信接口430和/或通信连接432可以用任何类型的有线和/或无线通信介质、接口、协议等来实现,包括PCIe、NVMe、以太网、NVMe-oF、快速计算链路(CXL)和/或诸如CXL.mem、CXL.cache、CXL.IO等的一致协议、Gen-Z、开放相干加速器处理器接口(Open Coherent Accelerator Processor Interface,OpenCAPI)、加速器高速缓存相干互连(Cache Coherent Interconnect for Accelerator,CCIX)、高级可扩展接口(Advanced eXtensible Interface,AXI)等或其任意组合、传输控制协议/互联网协议(TCP/IP)、光纤信道、InfiniBand、串行AT附件(SATA)、小型计算机系统接口(SCSI)、串行附接SCSI(SAS)、iWARP、包括2G、3G、4G、5G等的任何一代无线网络、任何一代Wi-Fi、蓝牙、近场通信(NFC)等、或其组合。在一些实施例中,通信连接432可以包括通信结构(communication fabric),该通信结构包括一个或多个链路、总线、交换机、集线器、节点、路由器、转换器、中继器等。在一些实施例中,图4所示的系统可以包括一个或多个附加装置,该附加装置具有使用通信连接432和/或其扩展连接的一个或多个附加通信接口。

[0072] 在设备404被实现为存储设备的实施例中,存储设备可以包括例如基于固态介质(例如,固态驱动器(SSD))、磁介质(例如,硬盘驱动器(HDD))、光介质等、或其任何组合的任何类型的非易失性存储介质。例如,在一些实施例中,存储设备可以被实现为基于非与(not-AND,NAND)闪存的SSD、诸如交叉网格非易失性存储器的永久存储器、具有体电阻变化的存储器、相变存储器(PCM)等,或其任意组合。任何这样的存储设备可以使用任何连接器配置,诸如SATA、SCSI、SAS、U.2、M.2等,以诸如3.5英寸、2.5英寸、1.8英寸、M.2、企业和数据

中心SSD形状因子(EDSFF)、NF1等的任何形状因子实现。任何这样的存储设备可以全部地或部分地与服务器机箱、服务器机架、数据室、数据中心、边缘数据中心、移动边缘数据中心和/或其任何组合一起实现和/或结合其使用。

[0073] 如上所述,编码逻辑438可以实现一种或多种编码方案或其组合,以对提交给一个或多个队列440的一个或多个条目的一个或多个测量进行编码,这取决于实现方式细节,可以压缩测量和/或减少存储测量所涉及的存储器的量。这种编码方案的一些示例如下。

[0074] 根据本公开的示例实施例的一些编码方案可以涉及测量(诸如时间戳)的粒度、量化、动态范围、增量编码、重置等。例如,在一些实施例中,可以以纳秒为单位测量的粒度来跟踪主机或设备处的系统时间。此外,可以根据主机或设备的通电时间来跟踪系统时间,该通电时间可以随着以年为单位测量的时间跨度延续。在这些延长的时间段内以这些小粒度跟踪系统时间(例如,以大的动态范围)可能需要具有多于32比特(例如,64比特)的时间戳。

[0075] 然而,在一些实施例中,队列中条目的超时时段可以以秒为单位来测量。例如,在一些实施例中,如果提交给NVMe提交队列的命令在一秒或两秒内没有被处理(例如,如果没有接收到完成),则主机可以认为该命令已经超时。此外,在一些实施例中,主机可能不需要或者不能够以小于毫秒的粒度来测量超时。因此,在一些示例实施例中,确定超时所涉及的动态范围可能仅约为14比特(例如,以1ms的粒度,16秒的范围(当测量几秒的范围时提供安全余量))。在具有每个队列多达64K个条目的队列以及每个控制器多达1024个队列的示例系统中,这可能仅涉及每个队列使用112KB的存储器或者控制器使用112MB的存储器。

[0076] 另外,在一些实施例中,确定队列条目的超时可能只涉及相对时间,而不是绝对系统时间。例如,在一些实施例中,系统时间戳生成器可以跟踪时间,可能以年为单位测量,从主机或设备通电时开始。然而,为了确定队列条目的超时,绝对系统时间可能并不重要。相反,队列中条目的端到端时长可以在该条目被提交给队列并且门铃更新被提供给控制器时开始测量。因此,在一些实施例中,只要队列是空的,空队列的时间戳就可以不断地被重置(例如,重置为零)。这可以被称为在队列的头部重置时间戳。然后,提交给队列的第一条目的提交时间戳可以为零(例如,在图3所示的示例中,对于SQ0,tss0=零),并且可以相对于第一时间戳的零值来确定后续时间戳。取决于实现方式细节,这可以减少在监视队列中条目的时长时所涉及的比特数,因为可以从队列中的第一条目开始相对于零来测量时间,而不是第一条目的可以涉及许多比特的绝对系统时间。在一些实施例中,可以基于队列的状态以例如可以减少监视队列条目时涉及的数据量(例如,比特数)的任何方式修改测量(例如,时间戳)。例如,在某些情况下,队列的状态可以为空或几乎为空。在一些实施例中,基于队列为空或几乎为空的状态,时间戳可以被设置为任何值,例如零或某个相对较低的值,这可以减少测量和/或存储队列中条目的端到端时长时所涉及的比特数。

[0077] 在一些实施例中,用于确定队列条目的时长的起始点可以取决于该条目如何被放置在队列中和/或如何向控制器和/或设备提供指示的细节。例如,在一些实施例中,可以将新条目插入提交队列,然后可以向设备和/或控制器提供门铃指示,以通知设备新条目存在于提交队列中。可以在设备接收门铃指示时启动定时器。取决于实现方式细节,在门铃指示被提供给设备以向其通知新条目之前,可以存在对提交队列中的新条目的一次或多次(可能多次)写入和/或修改。作为另一个示例,在一些实施例中,主机可以向提交队列中插入新条目,然后在门铃指示被提供给设备和/或设备处的控制器(例如,由主机)以向其通知新条

目之前可能经过了一段时间。作为另一个示例,在一些实施例中,主机可以向设备和/或设备处的控制器提供突发更新。例如,在门铃指示被提供给设备和/或设备处的控制器(例如,由主机)以向其通知多个新条目之前,多个条目(例如,十个条目)可以被创建并填充到队列中(例如,由主机)。

[0078] 此外,在一些实施例中,可以使用增量编码,例如,使得后续队列条目的时间戳可以相对于先前条目而不是队列的开始来测量。

[0079] 图5示出了根据本公开的示例实施例的用于队列条目的时间戳的增量编码的示例实施例。在图5所示的实施例中,针对提交队列SQn的表520中的第一条目的提交时间的第一时间戳ts0可以具有绝对初始值(例如,系统时间的绝对值,或者如果时间戳在队列的头部被重置为零,则值为零)。然而,不同于在针对提交队列SQn的表520中存储一个或多个后续条目的提交时间的一个或多个后续戳ts1、ts2、…tsm,可以使用差异时间(例如,时间增量)的表542来存储一个或多个条目(例如,连续条目)之间的时间戳值的变化。

[0080] 例如,在一些实施例中,时间增量的表542可能不具有对于提交队列SQn中第一条目的条目,因为它可能被假设为零。提交队列SQn中第二条目的时间增量可以存储在时间增量的表542中的第一条目Δts1中,作为提交队列SQn中第一条目的时间戳(例如ts0)和提交队列SQn中第二条目的时间戳(例如ts1)之间的差。即Δts1=ts1-ts0。类似地,提交队列SQn中第三条目的时间测量条目可以存储在时间增量的表542中的第二条目Δts2中,作为提交队列SQn中第二条目的时间戳(例如ts1)和提交队列SQn中第三条目的时间戳(例如ts2)之间的差。即Δts2=ts2-ts1。

[0081] 取决于实现方式细节,使用更大的时间戳粒度、时间戳重置和/或增量编码可以减少存储队列条目的时间戳和/或其他测量信息所涉及的存储器的量。在一些实施例中,这些方法提供的潜在存储器减少可以有多种原因。首先,时间增量的值(例如,Δts1,Δts2,…,Δtsm)可以更小和/或具有更小的动态范围,因此涉及使用更少的存储器空间。第二,由主机对队列的访问模式产生的时间增量值模式可以提供使用一个或多个附加编码方案进行进一步压缩的机会。例如,在一些实施例中,主机可以以由没有访问的更长的时间间隙分隔的紧密间隔的写入的突发来向提交队列写入条目。如果突发中的写入间隔得足够近,使得它们在时间戳粒度(例如,1ms)内发生,则连续写入之间的许多时间增量可能为零。因此,时间增量的表可包括相对大量的零(或其他小的值,诸如1),其中的许多可能是相邻的。在一些实施例中,具有相对大的数字或相同和/或相邻值的数据模式可以使用各种编码方案(诸如如下所述的熵编码、游程编码等)来高效地编码。

[0082] 根据本公开的示例实施例的一些附加编码方案可以涉及熵编码,诸如霍夫曼编码。在一些实施例中,熵编码可以确定(例如,测量、估计等)符号在数据序列中出现的概率,并且使用更短的码(例如,更少数量的比特)来编码更频繁的符号。

[0083] 表1示出了根据本公开的示例实施例的霍夫曼码的示例实施例,该霍夫曼码可用于对队列中的条目的增量时间戳测量进行编码。在表1所示的实施例中,x可以指示时间戳的相应部分的标准二进制编码值。

[0084] 被选择在表1中使用的码可以基于提交队列中的条目的增量时间戳的概率分布的以下潜在特征中的一个或多个。(1)队列条目的相对较大部分(例如,大多数队列条目)可以在时间上接近地提交(例如,由在例如虚拟机(VM)上运行的一个或多个应用进行的一个或

多个活动的突发中)，其中突发之间具有相对较大的时间间隙(例如，对于空闲VM)。(2)对于例如由空闲VM引起的空队列，时间戳可以在队列的头部被重置为零。(3)相对大量的队列条目可以具有相同的时间戳和/或增量时间戳值(例如，因为在时间上间隔很近)。

[0085] 在一些实施例中，可以在一个或多个监视窗口(也可以称为跟踪窗口)中监视队列条目。在这样的实施例中，在跟踪窗口中和/或在后续(例如，下一个)跟踪窗口中，相对大量的队列条目可以具有相同的时间戳和/或增量时间戳值。在其他实施例中，增量时间戳的概率分布可以具有不同的特征，因此，可以使用不同的码排列。

[0086] 表1

[0087]

Δtimestamp	比特数	码
0	1	0
1	2	10
<16	7	110<xxxx>
<256	12	1110<xxxxxxxx>
≥256	18	1111<xxxxxxxxxxxxxx>

[0088] 图6示出了根据本公开的示例实施例的用于队列条目的示例序列的霍夫曼编码的示例实施例。图6所示的实施例可以使用例如表1所示的码。

[0089] 图6所示的实施例包括63个队列条目(第0号队列条目到第62号队列条目)的示例序列。第一队列条目(第0号队列条目)可能没有码或增量时间戳，因为由于当队列为空时在队列的头部重置时间戳，它可以被假定为零。接下来的11个队列条目(第1号队列条目到第11号队列条目)可以具有为零的增量时间戳(Δts0＝0，Δts1＝0，…，Δts10＝0)，因为接下来的11个队列条目可以在第一队列条目(第0号队列条目)被提交之后在等于时间戳粒度的时间量内被提交给队列。因此，第1号队列条目到第11号队列条目的码可以都是零。

[0090] 例如，第12号队列条目可以具有等于一个粒度单位的增量时间戳，因为第12号队列条目可以在第一队列条目(第0号队列条目)被提交给队列之后，在等于时间戳粒度的时间量或之后不久的时间被提交给队列。参考表1，一个粒度单位的编码可以是“10”(例如，Δts11＝10)。因此，第12号队列条目的码在图6中显示为10。

[0091] 接下来的13个队列条目(第13号队列条目到第25号队列条目)可以具有为零的增量时间戳(Δts12＝0，Δts13＝0，…，Δts24＝0)，因为接下来的13个队列条目可以在第12号队列条目被提交给队列之后的等于时间戳粒度的时间量内被提交给队列。

[0092] 例如，第26号队列条目可以具有等于一个粒度单位的增量时间戳，因为第26号队列条目可以在第12号队列条目被提交给队列之后的等于时间戳粒度的时间量或之后不久被提交给队列。因此，第26号队列条目的码在图6中显示为10。

[0093] 接下来的24个队列条目(第27号队列条目到第50号队列条目)可以具有为零的增量时间戳(Δts26＝0，Δts27＝0，…，Δts49＝0)，因为第27号队列条目到第50号队列条目可以在第26号队列条目被提交给队列之后的等于时间戳粒度的时间量内被提交给队列。

[0094] 例如，第51号队列条目可以具有等于15个粒度单位的增量时间戳，因为第51号队列条目可以在第26号队列条目被提交给队列之后在等于15个时间戳粒度单位的时间量或之后不久被提交给队列。参考表1，十五个粒度单位的编码可以是“1101111”(110<xxxx>，其中对于十进制值十五，xxxx＝1111)。因此，第51号队列条目的码在图6中显示为1101111。

[0095] 接下来的11个队列条目(第52号队列条目到第62号队列条目)可以具有为零的增量时间戳(Δts51＝0,Δts52＝0,…,Δts61＝0),因为接下来的11个队列条目可以在第51号队列条目被提交给队列之后的等于时间戳粒度的时间量内被提交给队列。

[0096] 为了说明的目的,并且基于图6中所示的示例性增量时间戳,如果时间戳粒度被假设为1ms,则第1号队列条目到第11号队列条目可以在第一队列条目(第0号队列条目)被提交给队列的1ms内被提交给队列。类似地,第13号队列条目到第25号队列条目可以在第12号队列条目被提交到队列的1ms内被提交到队列。

[0097] 在一些实施例中,当在队列尾部提交(例如,插入)条目时,可以更新图6所示的码串。在一些实施例中,命令条目的时间戳可以例如在从图6所示的串中获取时计算。

[0098] 在图6所示的示例实施例中,用于表示63个队列条目的增量时间戳的码串可以是71比特长。为了说明的目的,假设不使用图6所示方法的队列条目的时间戳可以使用32比特数据,则由图6所示方法提供的压缩比可以由63×32/71≈28给出。因此,根据实现方式细节,图6所示的方法可以将存储时间戳数据所涉及的存储器的量减少28倍。然而,图6中所示的示例实施例不限于这些或任何其他特定的实现方式细节。

[0099] 根据本公开的示例实施例的一些附加编码方案可以涉及游程编码。在一些实施例中,游程编码可以用模式的单个副本和指示模式重复次数的计数来代替重复数据模式的序列。

[0100] 图7示出了根据本公开的示例实施例的用于队列条目的示例序列的熵编码结合游程编码的第一示例实施例。为了说明的目的,图7所示的实施例被显示为应用于图6所示的得出的霍夫曼码。然而,在其他实施例中,游程编码可以直接应用于队列条目的序列,或者与一个或多个其他编码方案相结合。

[0101] 图7所示的实施例可以实现固定长度的码,在这个示例中,该码可以是五比特长。5比特码的前4个比特可以是连续比特的数量的计数,以及5比特码的最后一个比特可以指示连续比特是高(1)还是低(0)。在一些实施例中,由前4比特提供的计数可以用计数减1(计数-1)的二进制编码来实现,因为可以不需要对零计数进行编码。

[0102] 参考图7,霍夫曼码中的第一重复模式可以是十一个零的序列(可以表示为11L)。因此,游程编码可以用游程编码“10100”代替霍夫曼码中的前11比特的零,其中前4比特“1010”是11(减1)的二进制编码,最后1比特的“0”表示重复模式是0。

[0103] 霍夫曼码中的下一个模式可以是单个1(可以表示为1H)。游程编码可以用游程码“00001”代替霍夫曼码中的单个1,其中前4比特“0000”是1(减1)的二进制编码,最后1比特“1”表示重复模式是1。

[0104] 霍夫曼码中的下一个模式可以是十四个0的序列(可以表示为14L)。因此,游程编码可以用游程码“11010”替换霍夫曼码中接下来的14比特的零,其中前4比特“1101”是14(减1)的二进制编码,最后1比特“0”表示重复模式是0。

[0105] 霍夫曼码中的下一个模式可以是单个1(可以表示为1H)。游程编码可以用游程码“00001”代替霍夫曼码中的单个1,其中前4比特“0000”是1(减1)的二进制编码,最后1比特“1”表示重复模式是1。

[0106] 霍夫曼码中的下一个模式可以是二十五个0的序列。然而,因为四个计数器比特可以支持最大计数为16,所以该序列可以被分解成16个0和9个0的两个连续序列(可以表示为

16L和9L)。因此,游程编码可以用第一游程码“11110”和第二游程码“10000”来代替25个零的序列,第一游程码“11110”中前4比特“1111”是十六(减一)的二进制编码,最后1比特“0”表示重复模式是0;第二游程码“10000”中前4比特“1000”是九(减一)的二进制编码,最后1比特“0”表示重复模式是0。

[0107] 霍夫曼码中的下一个模式可以是两个1的序列(可以表示为2H)。游程编码可以用游程码“00011”代替霍夫曼码中接下来的两比特1,其中前4比特“0001”是二进制编码2(减1),最后1比特“1”表示重复模式是1。

[0108] 霍夫曼码中的下一个模式可以是单个0(可以表示为1L)。游程编码可以用游程码“00000”代替霍夫曼码中的单个0,其中前4比特“0000”是1(减1)的二进制编码,最后1比特“0”表示重复模式是0。

[0109] 霍夫曼码中的下一个模式可以是四个1的序列(可以表示为4H)。游程编码可以用游程码“00111”代替霍夫曼码中接下来的四比特1,其中前4比特“0011”是4(减1)的二进制编码,最后1比特“1”表示重复模式是1。

[0110] 霍夫曼码中的最终重复模式可以是十一个0的序列(可以表示为11L)。游程编码可以用游程码“10100”代替霍夫曼码中的最后11比特零,其中前4比特“1010”是11(减1)的二进制编码,最后1比特“0”表示重复模式是0。

[0111] 在图7所示的示例实施例中,得到的游程码可以具有50比特。这可以与单独的霍夫曼编码使用的71比特进行比较,如图6所示。因此,游程编码可以提供71/50≈1.4的附加压缩比。取决于实现方式细节,图7所示的霍夫曼编码和游程编码的组合可以将存储器空间减少63×32/50≈40倍。然而,图7中所示的示例实施例不限于这些或任何其他特定的实现方式细节。

[0112] 图8示出了根据本公开的示例实施例的用于队列条目的示例序列的熵编码结合游程编码的第二示例实施例。在图8所示的实施例中,游程编码可以集成到霍夫曼编码中,而不是在霍夫曼编码之后应用。在一些实施例中,这可以被称为混合码。

[0113] 表2示出了可用于对图8所示实施例中的队列条目的增量时间戳测量进行编码的组合的霍夫曼码和游程码的示例实施例。在表2所示的实施例中,x可以指示时间戳的相应部分的标准二进制编码值。

[0114] 表2

[0115]

Δtimestamp	比特数	码
0	可变的	使用前缀为0并且计数器为4比特的固定长度游程编码
1	2	10
<16	7	110<xxxx>
<256	12	1110<xxxxxxxx>
⩾256	18	1111<xxxxxxxxxxxxxx>

[0116] 参考图8,第一队列条目(第0号队列条目)可能没有码或增量时间戳,因为由于当队列为空时在队列头部重置时间戳,它可能被假定为零。接下来的11个队列条目(第1号队列条目到第11号队列条目)可以具有为零的增量时间戳(Δts0=0,Δts1=0,…,Δts10=0)。因此,基于表2,第1号队列条目到第11号队列条目可以用游程编码来编码。因此,第1号队列条目到第11号队列条目的混合码输出可以是游程码“01010”,其中首位的(leading)比特“0”表示重复模式是0,最后四个比特“1010”是11(减1)的二进制编码。

[0117] 下一个队列条目(第12号队列条目)可以具有等于1的增量时间戳。基于表2,对于1的编码可以是“10”(例如,Δts11=10)。因此,在图8中,对于第12号队列条目的混合码显示为10。

[0118] 接下来的13个队列条目(第13号队列条目到第25号队列条目)可以具有为0的增量时间戳(Δts12=0,Δts13=0,…,Δts24=0)。基于表2,第13号队列条目到第25号队列条目可以用游程编码来编码。因此,第13号队列条目到第25号队列条目的混合码输出可以是游程编码“01100”,其中首位的比特“0”表示重复模式是0,最后四个比特“1100”是13(减1)的二进制编码。

[0119] 下一个队列条目(第26号队列条目)可以具有为1的增量时间戳。基于表2,对于1的编码可以是“10”(例如,Δts25=10)。因此,如图8所示,对于第26号队列条目的混合码可以是10。

[0120] 接下来的24个队列条目(第27号队列条目到第50号队列条目)可以具有为0的增量时间戳(Δts26=0,Δts27=0,…,Δts49=0)。因此,基于表2,第27到50号队列条目可以用游程编码来编码。然而,因为4个计数器比特可以支持最大计数为16,所以该序列可以被分解成16个0和8个0的两个连续序列。因此,混合输出编码可以包括第一游程码“01111”以及第二游程码“00111”,第一游程码中首位的比特“0”表示重复模式是0,最后四个比特“1111”是16(减1)的二进制编码,以及第二游程码中首位的比特“0”表示重复模式是0,最后四个比特“0111”是8(减1)的二进制编码。

[0121] 第51号队列条目可以具有为15的增量时间戳。基于表2,对于15的编码可以是“1101111”(例如,Δts51=1101111)。因此,第51号队列条目的混合码可以是1101111,如图8所示。

[0122] 接下来的11个队列条目(第52号队列条目到第62号队列条目)可以具有为0的增量时间戳(Δts51＝0,Δts52＝0,…,Δts61＝0)。基于表2,第52号队列条目到第62号队列条目可以用游程编码来编码。因此,第52号队列条目到第62号队列条目的混合码输出可以是游程码“01010”,其中首位的比特“0”表示重复模式是0,最后四个比特“1010”是11(减1)的二进制编码。

[0123] 在图8所示的示例实施例中,得到的混合码可以具有36比特。这可以与单独的霍夫曼编码使用的71比特进行比较,如图6所示。因此,游程编码可以提供71/36≈2的附加压缩比。取决于实现方式细节,图8所示的混合霍夫曼编码和游程编码可以将存储器空间减少63×32/36≈56倍。然而,图8中所示的示例实施例不限于这些或任何其他特定的实现方式细节。

[0124] 在一些实施例中,可以使用一种或多种可以实现存储空间和准确度之间的折衷的压缩技术来编码与队列条目相关的一个或多个测量。例如,一些实施例可以使用一种或多种非线性算法,诸如对数、指数等,舍入等。这种编码方案可以是有用的,例如,对于诸如时间戳和/或Δtimestamp的相对较大的测量值,其准确度可能不那么有用,并且对于其,用一些准确度换取更紧凑的编码可以是有益的。

[0125] 表3示出了根据本公开的示例实施例的可用于对诸如时间戳的队列条目测量进行编码的码的集合。具有在表3的第一列中所示的十进制范围内的值的Δtimestamp可以使用表3的第二列的相同行中所示的相应二进制码来编码。

[0126] 例如,如果队列条目的Δtimestamp具有十进制值为10,则Δtimestamp可以使用二进制码“11”来编码,如表3的第三列所示,该二进制码“11”可以被解释为十进制值为8。(二进制值11对应于十进制值3,十进制值2^3＝8。)因此,对于十进制值10的编码可能具有十进制2的误差(例如,准确度损失),因为十进制10可能被编码为二进制11,二进制11稍后可能被解释为十进制8。

[0127] 因为8和15之间的十进制值可以使用二进制码“11”来编码,如表3所示,二进制码“11”可以被解释为十进制值8,所以在8到15的十进制范围内的Δtimestamp的准确度的潜在损失可以低至0(对于Δtimestamp为8)到高达7(对于Δtimestamp为15)。

[0128] 在一些实施例中,表3所示的码可以被描述为下舍入实现方式,因为落在两个不同二进制码的十进制解释之间的Δtimestamp可以被下舍入到下一个最低的二进制码。其他实施例可以使用上舍入和/或其他技术。

[0129] 在一些实施例中,表3所示的码可以被描述为实现对数(log)标度编码方案。在一些实施例中,表3所示的码可以被描述为实现有损编码方案,因为例如并且取决于实现方式细节,Δtimestamp值的原始值可能无法在不损失准确度的情况下被恢复。然而,在一些实施例中,准确度的损失可能不是直接由于编码方案的对数性质,而是可能是由用于实现该方案的舍入引起的。

[0130] 表3

[0131]

Δtimestamp (十进制)	要使用的码 (二进制)	码的解释(十进制)	准确度的损失(十进制)
0	0	0	0
1-3	1	$2^1=2$	0 到 1
4-7	10	$2^2=4$	0 到 3
8-15	11	$2^3=8$	0 到 7

[0132]

16-31	100	$2^4=16$	0 到 15
32-63	101	$2^5=32$	0 到 31
64-127	110	$2^6=64$	0 到 63
128-255	111	$2^7=128$	0 到 127
…	…	…	…

[0133] 表4示出了十进制Δtimestamp值(第一列)以及标准二进制表示(第二列)和使用表3所示码的表示。如表4中所示,并且取决于实现方式细节,表3中所示的编码方案可以提供小得多的码,并且因此使用小得多的存储器空间来存储,同时仍然提供可接受的准确度,用于监视队列条目的时长或其他测量。例如,具有十进制值为34的Δtimestamp可以由二进制码“101”来表示,该二进制码“101”可以是标准二进制表示“100010”的一半长度。作为另一个示例,具有十进制值为125的Δtimestamp可以由二进制码“110”来表示,该二进制码“110”可以小于标准二进制表示“1111101”的长度的一半。

[0134] 表4

[0135]

Δtimestamp	标准二进制表示	表 3 中的二进制码
0	0	0
34	100010	101
4	100	10
2	10	1
125	1111101	110

[0136] 尽管本文公开的编码技术不限于任何特定的应用和/或实现方式,但是编码技术

中的一种或多种或其组合在应用于监视提交给诸如NVMe的存储协议中的一个或多个提交队列的命令的时长时可能特别有益。例如，在一些实施例中，队列可以在其间为空的时间可以与调试命令超时无关，因此，重置队列头部的时间戳可以是减少用于存储与命令时长相关的测量的存储器的量的有效技术。此外，增量编码可以进一步减少存储器的使用。作为另一个示例，在一些实施例中，NVMe命令可以以由很少或没有活动(例如，由于空闲的VM)的相对长的时段分隔的紧密间隔的写入的突发的形式被放置在提交队列中。这可以产生时间戳的模式，该模式可以用诸如霍夫曼编码的熵编码来高效地编码。此外，相对长的重复时间戳序列可以用游程编码来有效地编码。作为另一个示例，在一些实施例中，可以使用诸如对数编码(例如，带有舍入)的有损编码方案来进一步减少存储一些NVMe命令之间的相对长的时间增量所涉及的存储器，同时仍然保持足够的准确度。

[0137] 图9示出了根据本公开的示例实施例的具有带有编码的队列监视的系统的示例实施例。例如，图9所示的实施例可以用于实现图4所示的实施例。

[0138] 图9所示的实施例可以包括主机902和设备904。主机902可以包括第一通信接口928。设备904可以包括第二通信接口930和控制器934。第一通信接口928和第二通信接口930可以通过通信连接932通信。出于说明的目的，在主机902处示出了一个或多个队列940，但是一个或多个队列940可以部分地或全部地位于设备904处的控制器934处或任何其他位置。

[0139] 设备904还可以包括时间戳生成器918和/或本地存储器944。在一些实施例中，时间戳生成器918和/或本地存储器944可以部分地或全部地被包括在控制器934中。控制器934可以包括编码逻辑938和选择逻辑946。编码逻辑938可以被配置为使用一个或多个参数950来实现一个或多个编码方案948。例如，一个或多个编码方案948可以包括上述任何编码方案，包括测量粒度和/或量化、基于队列状态重置测量(例如，重置队列头部的时间戳)、增量编码、诸如霍夫曼编码的熵编码、游程编码、对数编码、舍入等。在一些实施例中，编码方案948也可以被称为编码技术，并且参数950也可以被称为(例如，用于编码方案或技术的)设置。在一些实施例中，编码方案948也可以被称为参数。

[0140] 在一些实施例中，编码逻辑938可以被配置为基于接收条目已经被提交给队列940的指示936，获得队列940中的条目的测量(例如，来自时间戳生成器918的时间戳)。编码逻辑可被配置为使用一个或多个参数950利用一个或多个编码方案948来编码测量，并将编码的测量存储在例如本地存储器944中。

[0141] 选择逻辑946可以被配置为选择编码方案948和/或参数950中的一个或多个。例如，在一些实施例中，选择逻辑946可以被配置为监视提交给队列的条目(例如，通过监视作为指示936接收的尾部门铃更新)，并且基于该监视，确定以下中的一个或多个：(1)是否使用游程编码来压缩条目的时间戳；(2)如果使用游程编码，对于编码要使用什么参数(例如，计数器长度、固定的或可变的计数器长度、前缀和/或后缀的使用、混合编码等)；和/或(3)一种或多种其他编码方案，诸如熵编码和/或用于其的一个或多个参数。

[0142] 在一些实施例中，选择逻辑946可以静态方式选择一个或多个编码方案948和/或参数950。例如，在一些实施例中，主机902可以向选择逻辑946提供与可以由主机提交给队列的条目的一个或多个模式相关的信息，诸如队列的条目的数量、队列的条目的频率、队列的条目的间隔、队列的条目的一致性(uniformity)等。例如，如果主机902可以正运行机器

学习应用,则可以预期主机在大多数时间以相对规则的访问模式连续读取数据,因此,命令可以以相对一致的间隔被写入提交队列。作为另一个示例,如果主机902可能正在运行涉及人工输入的终端应用,则访问模式可能更加零星(sporadic),其中,在相对短的时间帧中,密集间隔的命令的偶尔突发被写入提交队列,随后是很少或没有命令提交的较长时间段。基于从主机902接收这种类型的信息,选择逻辑946可以选择可以适合于队列条目的预期模式的一个或多个编码方案948和/或参数950。在一些实施例中,一个或多个选择的编码方案948和/或参数950可以例如在全功率周期中使用。在一些实施例中,选择逻辑946可以研究一个功率周期期间的队列条目行为,并使用该信息来选择一个或多个编码方案948和/或参数950,以在另一个功率周期期间使用。

[0143] 在一些实施例中,选择逻辑946可以包括静态选择逻辑952,静态选择逻辑952可以被配置为以静态方式选择一个或多个编码方案948和/或参数950。例如,在一些实施例中,选择逻辑946可以选择一个或多个初始编码方案948和/或参数950,并且监视和/或评估初始编码方案948和/或参数950的有效性、效率等。在一些实施例中,监视和/或评估可以在一个或多个窗口(例如,可以针对操作的特定持续时间预先配置的窗口)期间执行。基于对有效性、效率等的监视和/或评估,选择逻辑946可以调整编码方案948和/或参数950(例如,选择一个或多个不同的编码方案948和/或参数950)以在后续窗口期间使用。在一些实施例中,选择逻辑946可以在表中存储一个或多个编码方案948和/或参数950的集合,并且添加预定义的标记以指示用于标记之后的编码的表索引。在一些实施例中,初始编码方案948和/或参数950可以基于由主机902提供的信息来确定,如上关于静态选择所述的。在一些实施例中,初始编码方案948和/或参数950可以基于最佳猜测、主机902的先前行为、应用等来确定。

[0144] 在一些实施例中,选择逻辑946可以包括动态选择逻辑954,该动态选择逻辑954可以被配置为基于与队列条目相关的测量(例如,时间戳)相关的一个或多个统计来动态地调整一个或多个编码方案948和/或参数950。例如,在一些实施例中,选择逻辑946可以监视向提交队列进行的命令提交之间的平均时间。如果平均时间达到阈值(这可以指示例如Δtimestamp相对较大),则选择逻辑946可以单独(by itself)或者在可以使用的任何其他编码之上应用有损编码方案,诸如表3中所示的对数方案。

[0145] 图10示出了根据本公开的示例实施例的具有带有调试的队列监视的系统的示例实施例。例如,图10所示的实施例可以用于实现图4所示的实施例。图10所示的实施例可以包括类似于图9所示的一个或多个组件,并且可以由以相同数字结尾的附图标记来表示。

[0146] 图10所示的实施例还可以包括调试逻辑1056,其可以被配置为对一个或多个队列1040中的条目和/或设备1004、主机1002和/或在主机1002上运行的应用、驱动程序、进程、实用程序(utility)等进行管理、调试、剖析、评估和/或等等。调试逻辑1056可以包括数据收集逻辑1058、工作负荷估计逻辑1060、超时确定逻辑1062和/或报告逻辑1064。为了说明的目的,图10所示的实施例的一些操作可以在命令时长相关超时的上下文中描述,例如在可以实现NVMe的系统中。然而,图10所示的实施例不限于任何特定的上下文或实现方式细节。

[0147] 在一些实施例中,命令时长相关超时可以例如由以下因素触发:(1)提交队列,其可以具有用于相对较大工作负荷的一个或多个命令;(2)多个队列,其可以具有用于相对较

大的合计的(aggregated)工作负荷的一个或多个命令;和/或(3)一个或多个底层系统和/或设备问题。如果已经检测到超时,则调试逻辑1056可以用于确定设备1004是否出现故障。如果没有检测到超时,则调试逻辑1056可以用于确定超时是否可能很快发生。调试逻辑的可能特征的一些附加示例可以包括停止读取恢复活动,例如,以遵守时间限制的读取恢复设置,和/或中止命令。

[0148] 在一些实施例中,数据收集逻辑1058可以基于由设备1004从队列1040中的一个或多个所处理的一个或多个命令来收集数据,诸如所传递的数据的量(例如,所传递的逻辑块(NLB)的数量)。数据收集逻辑1058还可以收集和/或分析数据,诸如由队列1040中的一个或多个所处理的命令的数量,和/或队列1040中的一个或多个的平均队列深度。在一些实施例中,数据收集逻辑1058可以确定和/或维护每提交队列的平均命令处理速率和/或平均数据处理速率(例如,NLB的数量)(例如,如果所有队列都不为空)。

[0149] 如果检测到超时(例如,由主机1002和/或超时检测逻辑1062),工作负荷估计逻辑1060可以确定命令和/或基于命令传递的数据量是否反映了一个或多个队列1040的命令仲裁配置,所述命令和/或基于命令传递的数据量可以通过平均队列深度来调整。在一些实施例中,如果命令和/或基于命令传递的数据量(可选地由平均队列深度调整)与命令仲裁配置一致,则可以假设超时与工作负荷无关,并且可能是由一个或多个底层系统和/或设备问题引起的。然而,如果命令和/或基于命令传递的数据量(可选地由平均队列深度调整)与命令仲裁配置不一致,则可以认为超时是由设备的性能(或故障)引起的。在任一情况下,结果可以由报告逻辑1064报告给例如主机1002。

[0150] 在一些实施例中,工作负荷估计逻辑1060可以通过确定对一个或多个提交队列的工作负荷和/或多个提交队列的合计的工作负荷(例如,对于整个系统)的估计(例如,瞬时估计)来执行预测分析(例如,即使在没有超时检测的情况下),并且报告工作负荷是否大于预定阈值。在一些实施例中,超过阈值的工作负荷可以预测即将到来的和/或最终的超时状况。例如,在一些实施例中,工作负荷估计逻辑1060可以如下确定估计的工作负荷:对于i＝1到n,估计的工作负荷＝SUM(针对SQi处理的平均NLB,SQi深度),其中SQi可以指示单个提交队列,n可以指示提交队列的总数。在一些实施例中,如果估计的工作负荷超过阈值,则工作负荷估计逻辑1060可以指示数据收集逻辑1058记录(log)用于调试操作的收集的数据中的一些或所有,和/或指示报告逻辑1064报告(例如,向主机1002和/或更大的系统)可能超时和/或可能存在系统错误配置。

[0151] 图11示出了根据本公开的示例实施例的具有带有记录保存的队列监视的系统的示例实施例。例如,图11所示的实施例可以用于实现图4所示的实施例。图11所示的实施例可以包括类似于图9和/或图10所示的一个或多个组件,并且可以由以相同数字结尾的附图标记表示。

[0152] 图11所示的实施例还可以包括记录保持逻辑1166,在一些实施例中,记录保持逻辑1166可以收集和/或存储与由一个或多个条目传递的一个或多个数据量相关的伴随数据、与一个或多个条目相关的一个或多个时间戳或其他测量等。在一些实施例中,记录保存逻辑1166可以包括数据收集逻辑1168、编码选择逻辑1170和/或滑动窗口逻辑1172。为了说明的目的,图11所示的实施例的一些操作可以在命令时长相关超时的上下文中描述,例如在可以实现NVMe的系统中。然而,图11所示的实施例不限于任何特定的上下文或实现方式

细节。

[0153] 例如,记录保持逻辑1166可以收集和/或保存与关于提交给命令队列的命令所传递的数据量有关的信息。在一些实施例中,该数据量可以被称为数据传递大小。收集和/或保存数据传递大小可以对于例如调试可以具有不同数据传递大小的命令当中的一个或多个超时是有用的。

[0154] 在一些实施例中,数据传递大小可以用逻辑块数量(NLB)来测量。在一些实施例中,数据传递大小可以存储为可以具有任何长度(例如,16比特)的比特字段(例如,在NVMe中)。

[0155] 在一些实施例中,当从队列中获取命令时,可以例如由数据收集逻辑1168收集(例如,记录)命令的数据传递大小。(取决于实现方式细节,数据传递大小可以不可用或不能被确定,直到命令从队列中获取。)一些实施例可以收集和/或保存命令的数据传递大小和时间戳的组合。附加地或可替换地,这种类型的技术可以用于修改时间戳,例如,以标记命令向提交队列的插入(例如,而不是控制器内部的解析开始的时间)。取决于实现方式细节,这可以生成对调试命令超时有用的比特字段记录(例如,相对长的记录)。

[0156] 编码选择逻辑1170可以实现一个或多个级别的编码,以减少用于存储命令的数据传递大小和/或组合的数据传递大小和时间戳的存储器的量。例如,对于第一级编码,编码选择逻辑1170可以以如下顺序或任何其他顺序实现包括以下操作中的一个或多个的编码过程:(1)增量编码,以编码命令之间的时间戳差(例如,Δtimestamp);(2)量化和/或舍入,例如,以减少增量值的更低阶信息;和/或(3)熵编码(例如,霍夫曼编码)以进一步减少量化的增量值。

[0157] 作为另一个示例,对于第二级编码,对于某些命令,可以不收集和/或存储数据传递大小和/或时间戳信息。例如,在一些实施例中,可以只为具有有效增量时间戳的命令收集和/或存储数据传递大小和/或组合的数据传递大小和时间戳。在这样的实施例中,编码选择逻辑1170可以以如下顺序或任何其他顺序实现包括以下操作中的一个或多个的过程:(1)可以累加针对选择和/或命令系列的数据传递大小;(2)累积可以例如在具有有效增量时间戳的命令处开始;(3)可以应用诸如上述第一级编码的编码过程来用后续(例如,下一个)增量时间戳来压缩和/或记录一个或多个累加值;和/或(4)一个或多个累加值可以在随后的(例如,下一个)增量时间戳被清除(例如,清零)。

[0158] 在一些实施例中,编码选择逻辑1170可以被实现为选择逻辑1146的一部分。在一些其他实施例中,编码选择逻辑1170可以至少部分独立地实现,例如,以解决可能特定于记录保持逻辑1166的编码选择的一个或多个方面。在一些实施例中,在选择一个或多个编码方案和/或参数之后,编码选择逻辑1170可以将编码操作移交给编码逻辑1138,并将结果数据存储在例如本地存储器1144中。

[0159] 作为另一示例,记录保持逻辑1166可以收集和/或保存写入提交队列的一个或多个命令的一个或多个时间戳。这可以例如,对于调试超时提供附加信息是有用的。如果在从提交队列中获取命令之后移除和/或丢弃时间戳,则可以减少可用于调试的信息量。例如,如果对于一个提交队列中的命令发生,参考当前SQ的前后(before-and-after)命令序列和/或其他提交队列的前后命令序列可以是有用的。此外,主机可能具有用于检测超时警告的相对较长的等待时间。在一些实施例中,在获取一个或多个命令之后(例如,在命令的完

成被放置在相应的完成队列中之后),收集和/或保存一个或多个命令的时间戳对于调试超时可以是有用的。

[0160] 在一些实施例中,记录保持逻辑1166可以收集和/或保存滑动时间窗口内的一个或多个命令的一个或多个时间戳。附加地或替代地,记录保持逻辑1166可以收集和/或保存落入一个或多个等待时间仓(bin)内的命令数量的一个或多个计数。在一些实施例中,这可以例如用命令或其他条目的等待时间的一个或多个直方图来实现。在一些实施例中,滑动窗口逻辑1172可以基于例如可用于存储所收集的数据的存储器的量、可以与一个或多个命令超时相关的时间间隔等来确定滑动窗口的大小。例如,在一些实施例中,滑动窗口可以基于可用于每队列64K条目的存储器的量的两倍和/或用于超时时段的时间量的两倍。在一些实施例中,可以使用一个或多个不同的偏移量来存储一个或多个前后命令序列(例如,在指定的限度内)。在一些实施例中,数据传递大小(例如,NLB)可能滞后,因为例如提交队列中的一个或多个未被获取的命令可能不具有对应的数据传递大小信息。在这种情况下,可以在获取命令之后为该命令记录数据传递大小信息。

[0161] 图4、图9、图10和图11所示的实施例中组件的布置是为了说明的目的,在其他实施例中,一些或所有组件可以以其他方式重新布置、组合、分离等。例如,一个或多个队列(440、940、1040或1140)、时间戳生成器(918、1018或1118)、本地存储器(944、1044或1144)、编码逻辑(938、1038或1138)、选择逻辑(946、1046或1146)、调试逻辑(1056)、记录保持逻辑(1166)等中的任何一项可以部分地或全部地位于主机和/或任何其他位置。作为另一示例,尽管编码逻辑、选择逻辑、调试逻辑和/或记录保持逻辑可以被示为控制器的一部分,但是在其他实施例中,这些组件中的任何一个或全部可以部分地或全部地实现为单独的组件、其他组件的一部分等。作为另一个示例,一些实施例可以省略一个或多个组件和/或功能,并且一些实施例可以组合组件中的任何或所有组件。例如,一些实施例可以在单个设备中组合编码逻辑、选择逻辑、调试逻辑和/或记录保持逻辑中的任何一个或全部。

[0162] 在一些实施例中,取决于实现方式细节,如本文所公开的跟踪一个或多个命令时长可以提供现场调试能力、改进的和/或一致的系统性能等。在一些实施例中,本文公开的原理可以通过跟踪总的等待时间来实现固件和/或系统调试。在一些实施例中,时间戳表的命令条目可以被压缩,这取决于实现方式细节,可以导致减少的存储器使用。在一些实施例中,本文公开的原理可以实现系统范围的命令时长跟踪,例如,通过减少跟踪命令时长所涉及的存储器的量。

[0163] 本文描述的任何功能,包括任何主机功能、设备功能等(例如,一个或多个队列(440、940、1040或1140)、时间戳生成器(918、1018或1118)、本地存储器(944、1044或1144)、编码逻辑(938、1038或1138)、选择逻辑(946、1046或1146)、调试逻辑(1056)、记录保持逻辑(1166)等中的任何一项)可以用硬件、软件、固件或其任意组合来实现,包括硬件和/或软件组合逻辑,时序逻辑,定时器,计数器,寄存器,状态机,诸如动态随机存取存储器(DRAM)和/或静态随机存取存储器(SRAM)的易失性存储器、包括闪存的非易失性存储器、诸如交叉网格非易失性存储器的永久存储器、具有体电阻变化的存储器、相变存储器(PCM)等和/或其任意组合,执行存储在任何类型的存储器中的指令的复杂可编程逻辑器件(CPLD)、现场可编程门阵列(FPGA)、专用集成电路(ASIC)CPU(包括诸如x86处理器的复杂指令集计算机(CISC)处理器和/或诸如RISC-V和/或ARM处理器的精简指令集计算机(RISC)处理器)、图形

处理单元(GPU)、神经处理单元(NPU)、张量处理单元(TPU)等。在一些实施例中,一个或多个组件可以被实现为片上系统(SOC)。

[0164] 图12示出了根据本公开示例实施例的主机装置的示例实施例。例如,图12所示的主机装置可以用于实现本文公开的任何主机。图12所示的主机装置1200可以包括处理器1202(其可以包括存储器控制器1204)、系统存储器1206、主机逻辑1208和/或通信接口1210。图12所示的任何或所有组件可以通过一个或多个系统总线1212进行通信。在一些实施例中,图12所示的一个或多个组件可以使用其他组件来实现。例如,在一些实施例中,主机控制逻辑1208可以由执行存储在系统存储器1206或其他存储器中的指令的处理器1202来实现。在一些实施例中,主机逻辑1208可以实现本文公开的任何主机功能,包括例如一个或多个队列(440、940、1040或1140)中的任何一个,向设备和/或选择逻辑提供与主机可以提交给队列的一个或多个条目模式相关的信息。

[0165] 图13示出了根据本公开示例实施例的设备的示例实施例。例如,图13所示的实施例1300可以用于实现本文公开的任何设备。设备1300可以包括设备控制器1302、队列监视逻辑1308、设备功能电路1306和/或通信接口1310。图13所示的组件可以通过一个或多个设备总线1312进行通信。队列监视逻辑1308可以用于例如实现一个或多个队列(440、940、1040或1140)、时间戳生成器(918、1018或1118)、本地存储器(944、1044或1144)、编码逻辑(938、1038或1138)、选择逻辑(946、1046或1146)、调试逻辑(1056)、记录保持逻辑(1166)等中的任何一个。

[0166] 设备功能电路1306可以包括实现设备1300的主要功能的任何硬件。例如,如果设备1300被实现为存储设备,则设备功能电路1306可以包括存储介质,诸如一个或多个闪存设备、闪存转换层(FTL)等。作为另一示例,如果设备1300被实现为网络接口卡(NIC),则设备功能电路1306可以包括一个或多个调制解调器、网络接口、物理层(PHY)、媒体访问控制层(MAC)等。作为另一示例,如果设备1300被实现为加速器,则设备功能电路1306可以包括一个或多个加速器电路、存储器电路等。

[0167] 图14示出了根据本公开的示例实施例的用于监视一个或多个队列条目的方法的实施例。该方法可以在操作1402处开始。在操作1404,该方法可以在设备处接收基于提交给队列的条目的指示。在一些实施例中,该指示可以例如用门铃寄存器来实现。在操作1406,该方法可以基于该指示获得对该条目的测量。例如,在一些实施例中,测量可以被实现为时间戳和/或增量时间戳。在操作1408,该方法可以对测量进行编码以生成编码的测量。例如,可以使用测量粒度和/或量化、基于队列状态重置测量(例如,重置队列头部的时间戳)、增量编码、诸如霍夫曼编码的熵编码、游程编码、对数编码、舍入等或其任意组合中的一个或多个来对测量进行编码。在操作1410,该方法可以存储编码的测量。该方法可以在操作1412处结束。

[0168] 图14所示的实施例以及本文描述的所有其他实施例是示例操作和/或组件。在一些实施例中,可以省略一些操作和/或组件,和/或可以包括其他操作和/或组件。此外,在一些实施例中,操作和/或组件的时间和/或空间顺序可以变化。尽管一些组件和/或操作可以被示为分开的组件,但是在一些实施例中,分开示出的一些组件和/或操作可以被集成到单个组件和/或操作中,和/或被示为单个组件和/或操作的一些组件和/或操作可以用多个组件和/或操作来实现。

[0169] 已经在各种实现方式细节的上下文中描述了上面公开的一些实施例，但是本公开的原理不限于这些或任何其他具体细节。例如，一些功能已经被描述为由某些组件实现，但是在其他实施例中，该功能可以分布在不同位置并具有各种用户界面的不同系统和组件之间。某些实施例被描述为具有特定的过程、操作等，但是这些术语也包含其中特定过程、操作等可以用多个过程、操作等来实现的实施例，或者其中多个过程、操作等可以集成到单个过程、步骤等中的实施例。对组件或元件的引用可以仅指该组件或元件的一部分。例如，对块的引用可以指整个块或一个或多个子块。在本公开和权利要求中使用诸如“第一”和“第二”的术语可能仅仅是为了区分它们所修饰的元素的目的，并且可能不指示任何空间或时间顺序，除非从上下文中显而易见。在一些实施例中，对元件的引用可以指该元件的至少一部分，例如，“基于”可以指“至少部分基于”等。对第一元件的引用并不意味着第二元件的存在。本文公开的原理具有独立的效用，并且可以单独实施，并且不是每个实施例都可以利用每个原理。然而，这些原理也可以以各种组合来实施，其中一些可以以协同的方式放大单个原理的益处。

[0170] 根据本专利公开的发明原理，上述各种细节和实施例可以被组合以产生另外的实施例。由于本专利公开的发明原理可以在布置和细节上进行修改，而不脱离发明构思，因此这种改变和修改被认为落入所附权利要求的范围内。

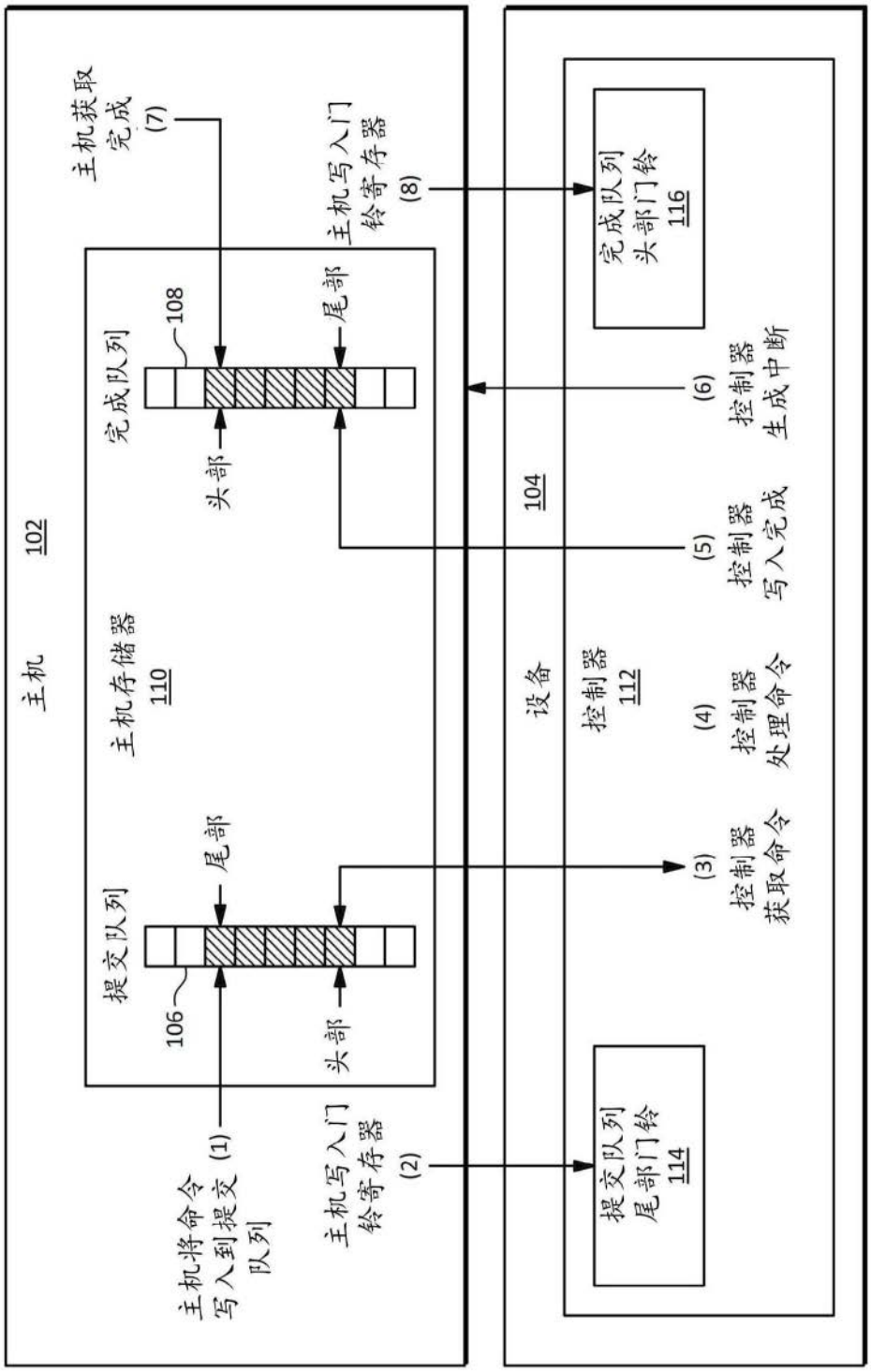

主机
102
主机存储器
110
提交队列
106
尾部
头部
完成队列
108
尾部
头部
主机将命令写入到提交队列
(1)
主机写入门铃寄存器
(2)
主机获取完成
(7)
主机写入门铃寄存器
(8)
设备
104
控制器
112
(3) 控制器获取命令
(4) 控制器处理命令
(5) 控制器写入完成
(6) 控制器生成中断
提交队列尾部门铃
114
完成队列头部门铃
116

图1

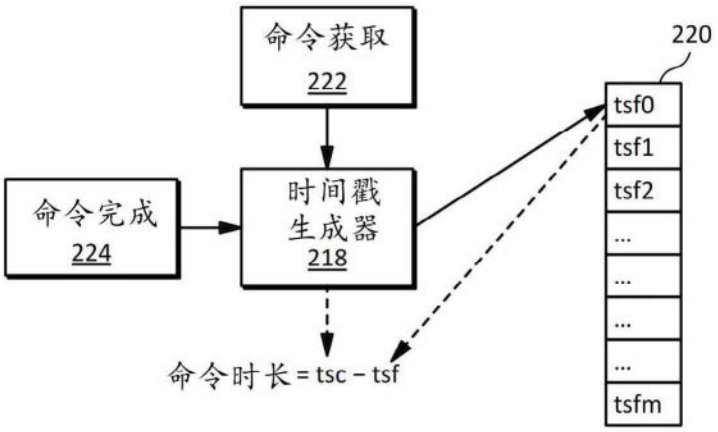

命令获取
222
220
tsf0
tsf1
tsf2
...
...
...
...
tsfm
命令完成
224
时间戳
生成器
218
命令时长 = tsc – tsf

图2

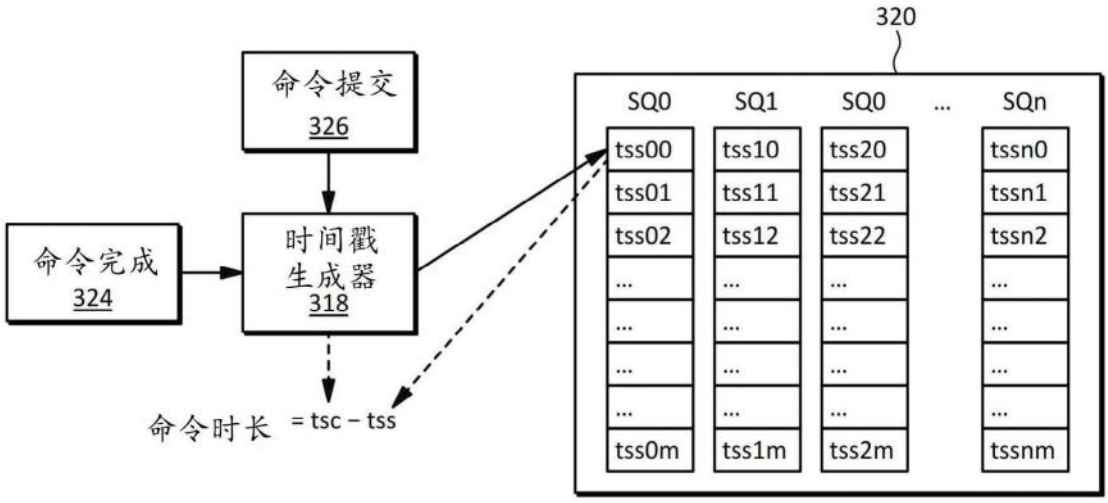

320
命令提交
326
SQ0
SQ1
SQ0
...
SQn
tss00
tss10
tss20
tssn0
tss01
tss11
tss21
tssn1
tss02
tss12
tss22
tssn2
tss0m
tss1m
tss2m
tssnm
命令完成
324
时间戳
生成器
318
命令时长 = tsc – tss

图3

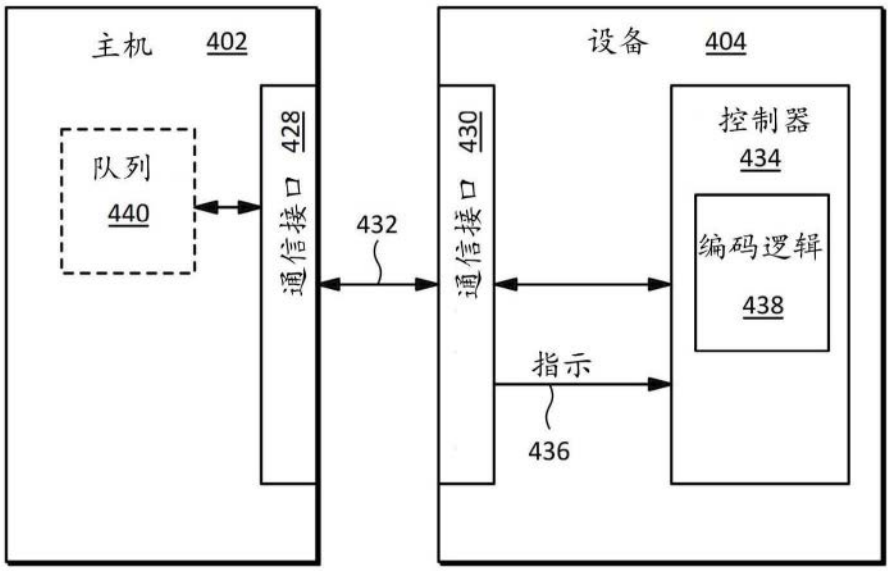

主机 402
队列 440
通信接口 428
432
设备 404
通信接口 430
控制器 434
编码逻辑 438
指示
436

图4

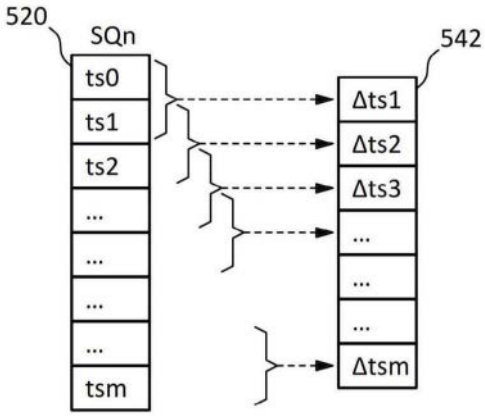

520
SQn
ts0
ts1
ts2
...
...
...
...
tsm
542
Δts1
Δts2
Δts3
...
...
...
Δtsm

图5

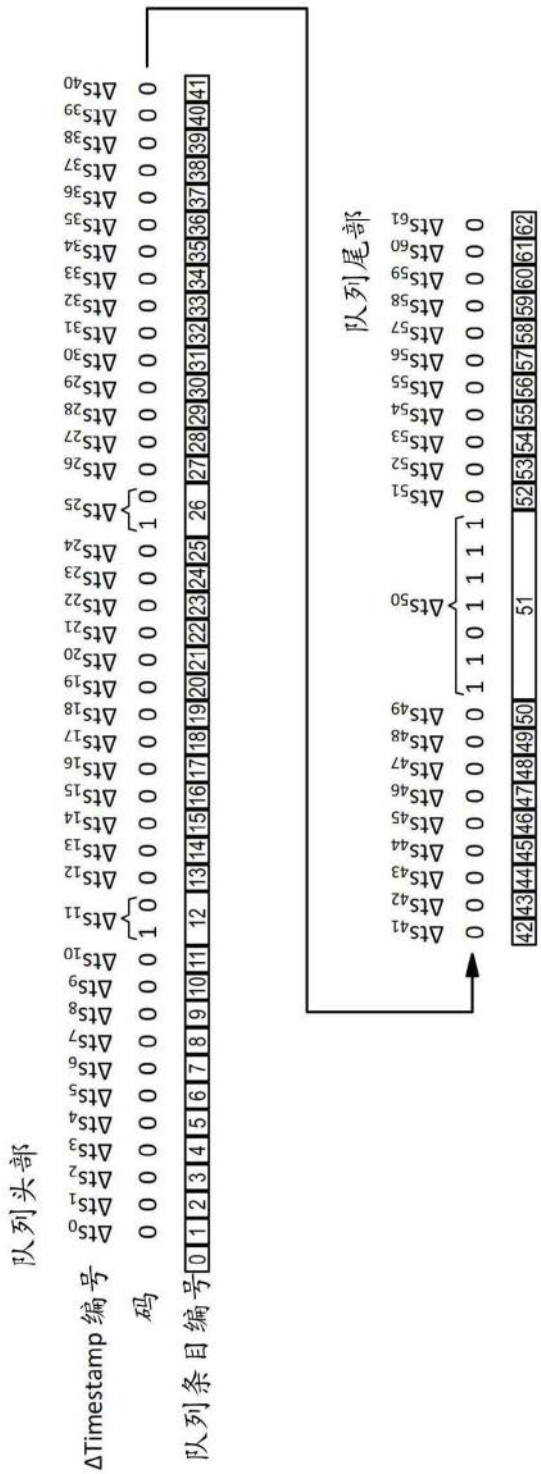


图6

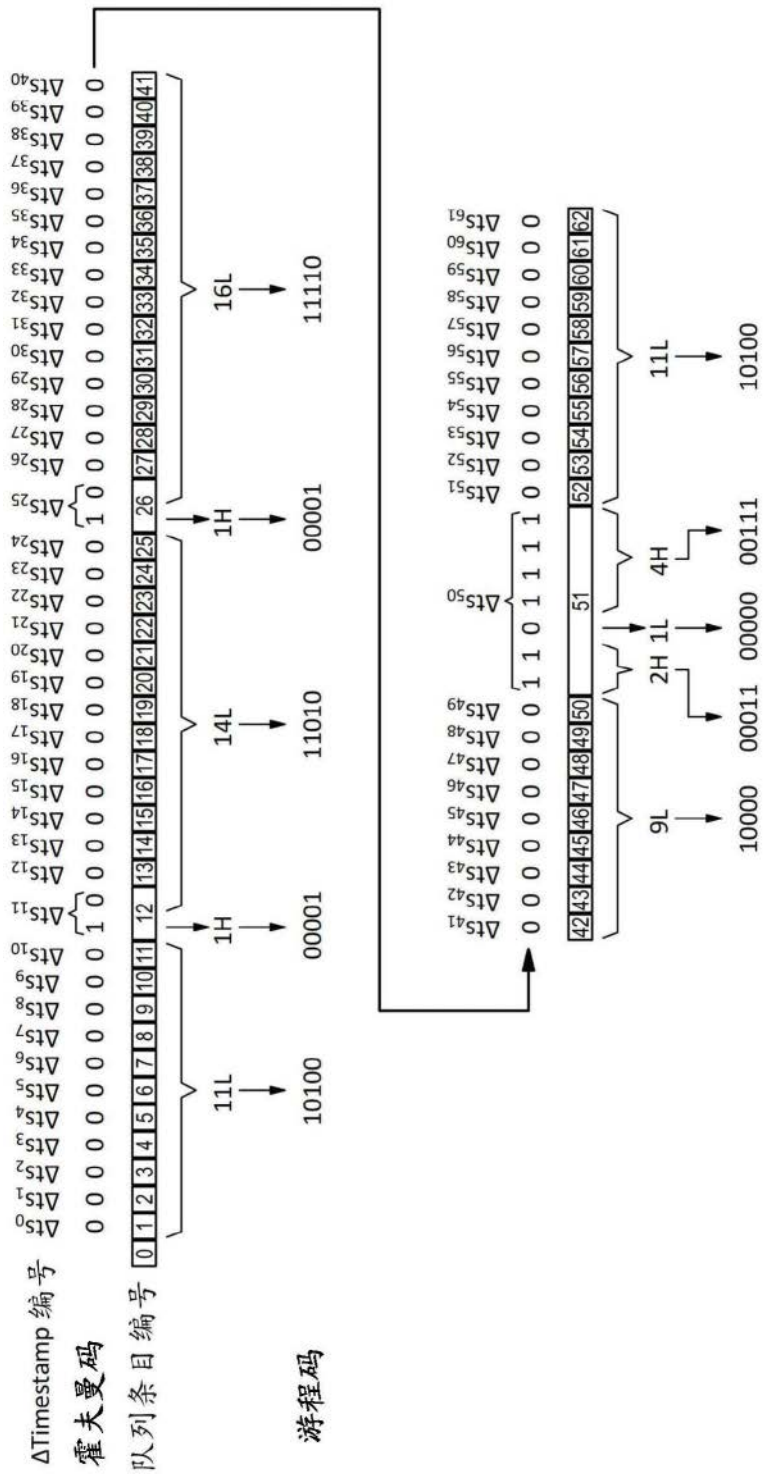

ΔTimestamp 编号
霍夫曼码
队列条目编号
游程码
11L
1H
14L
1H
16L
10100
00001
11010
00001
11110
9L
2H 1L
4H
11L
10000
00011
00000
00111
10100

图7

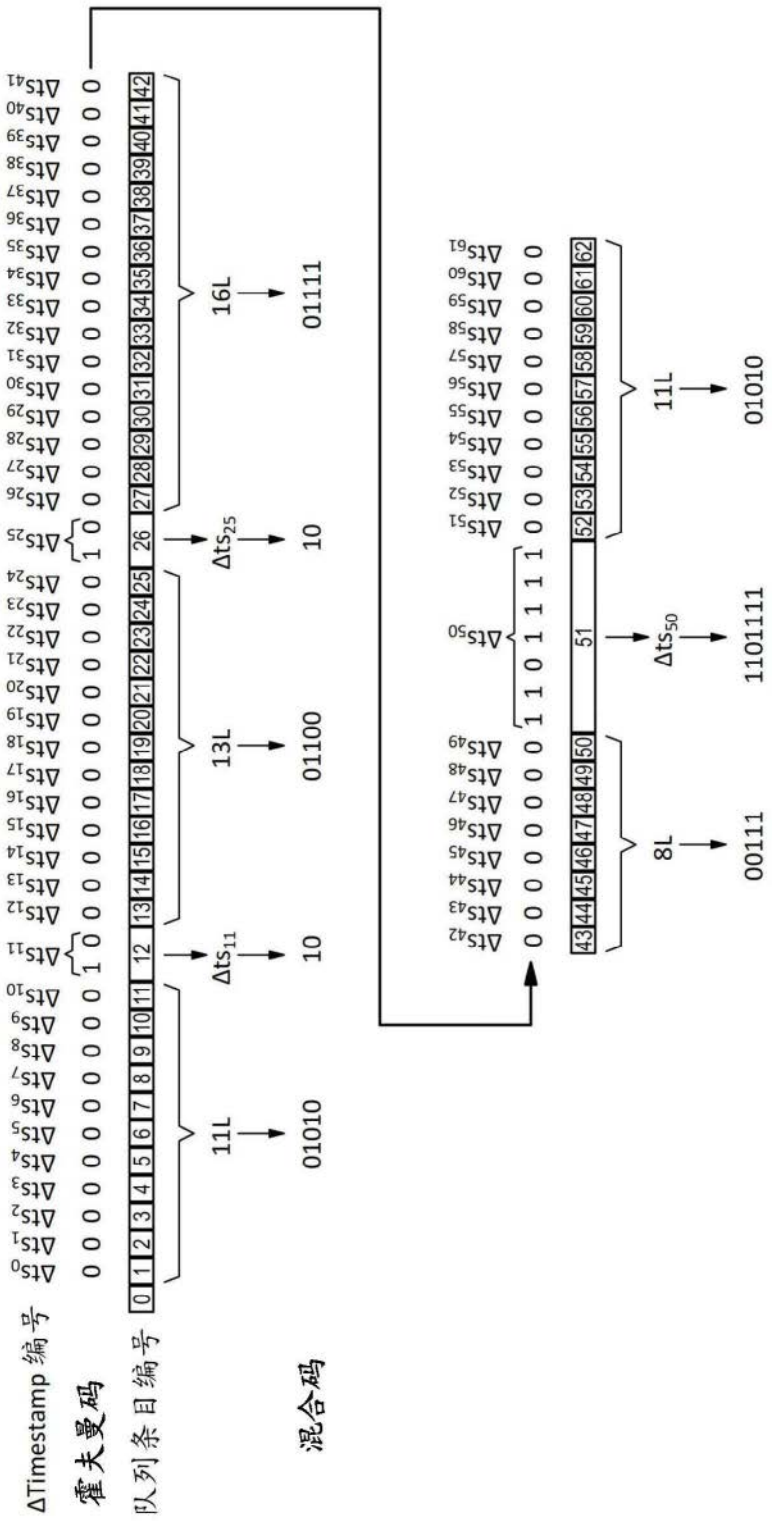

ΔTimestamp 编号
霍夫曼码
队列条目编号
混合码
11L
13L
16L
8L
11L
01010
01100
01111
00111
01010
11101111
10
Δts11
Δts25
Δts50

图8

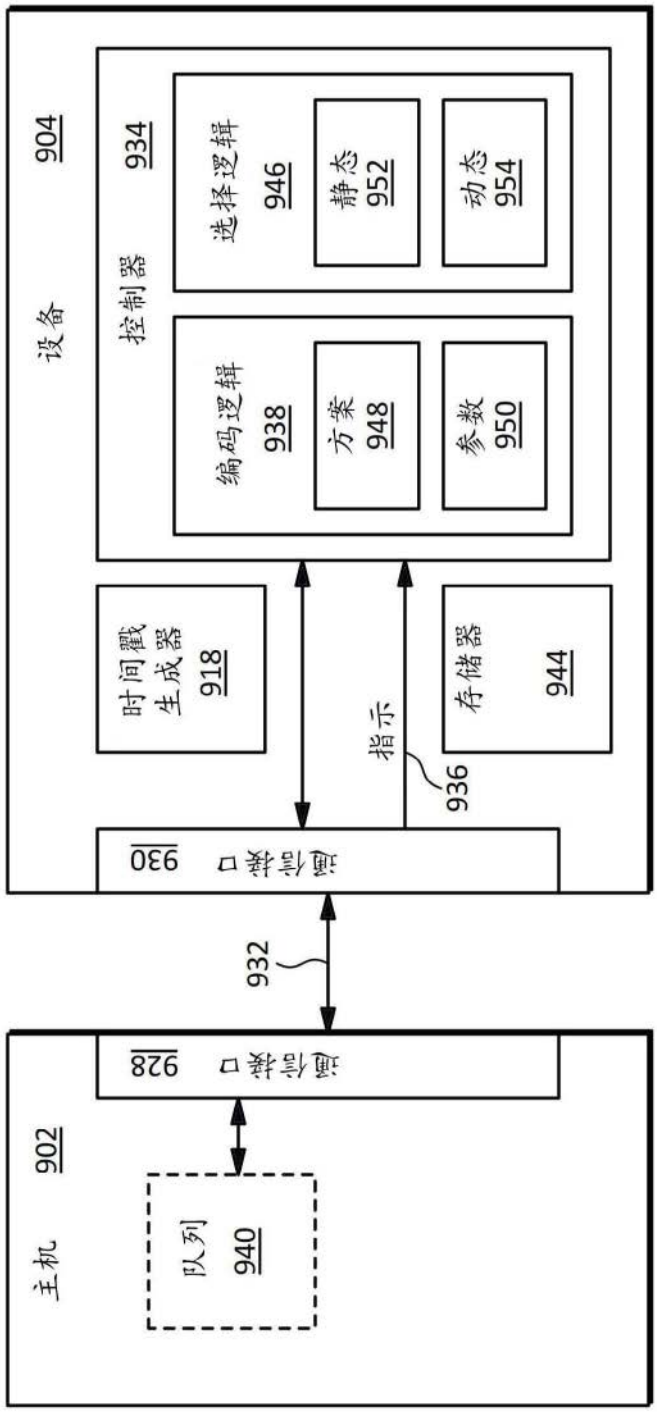

设备 904
控制器 934
选择逻辑 946
静态 952
动态 954
编码逻辑 938
方案 948
参数 950
时间戳生成器 918
存储器 944
指示 936
通信端口 930
932
通信端口 928
主机 902
队列 940

图9

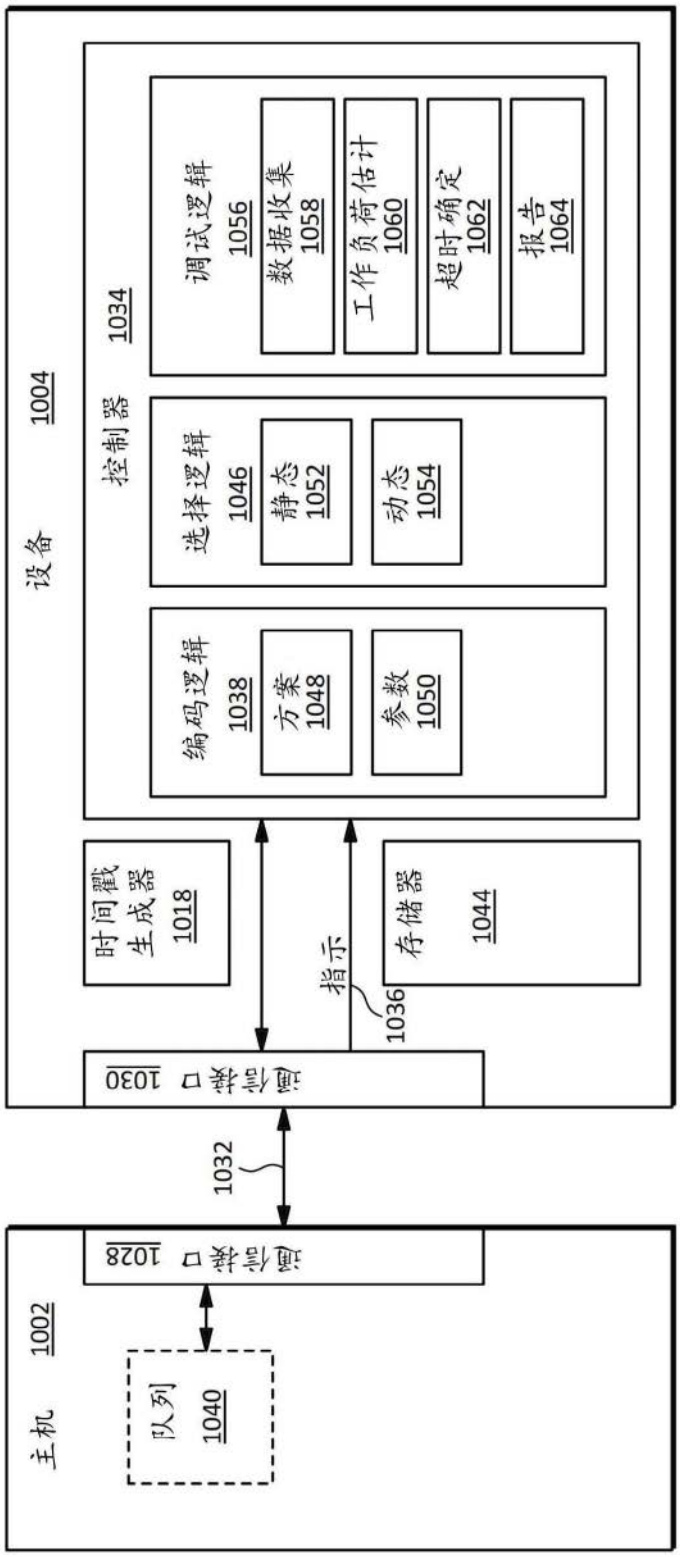

主机 1002
队列 1040
网络接口 1028
1032
网络接口 1030
设备 1004
控制器 1034
编码逻辑 1038
方案 1048
参数 1050
选择逻辑 1046
静态 1052
动态 1054
调试逻辑 1056
数据收集 1058
工作负荷估计 1060
超时确定 1062
报告 1064
时间戳生成器 1018
指示 1036
存储器 1044

图10

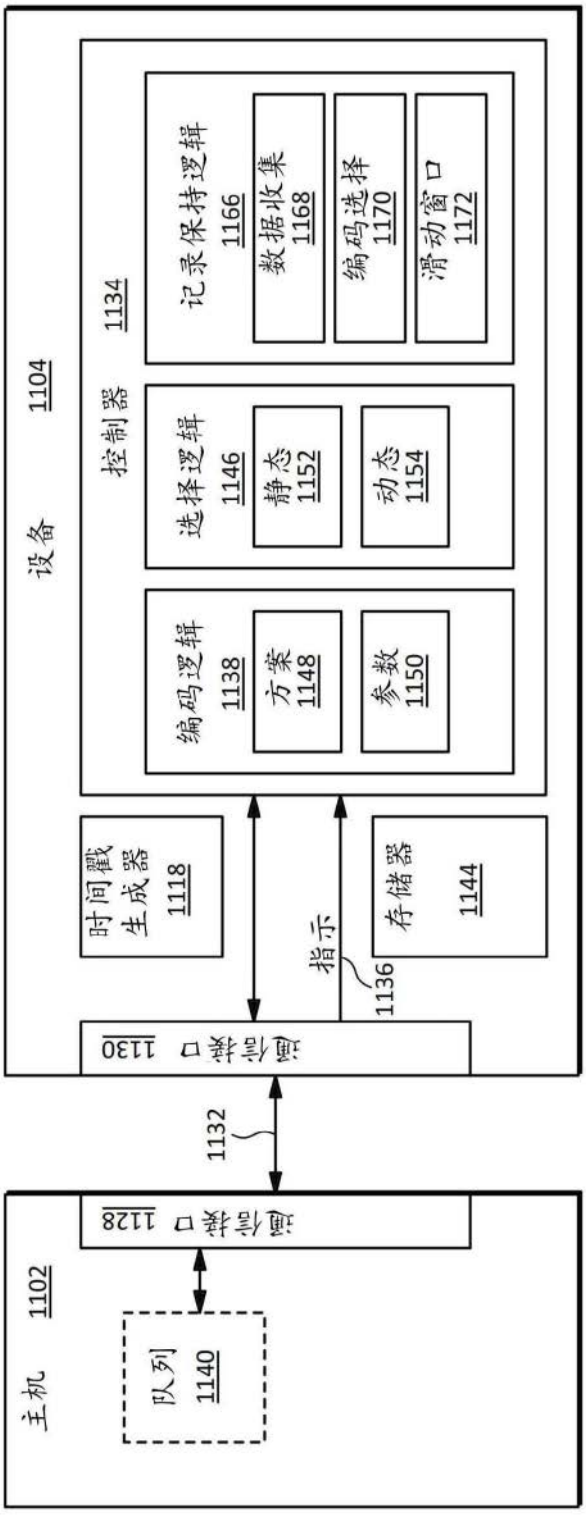

主机 1102
队列 1140
通信端口 1128
1132
通信端口 1130
设备 1104
控制器 1134
记录保持逻辑 1166
数据收集 1168
编码选择 1170
滑动窗口 1172
选择逻辑 1146
静态 1152
动态 1154
编码逻辑 1138
方案 1148
参数 1150
时间戳生成器 1118
存储器 1144
指示 1136

图11

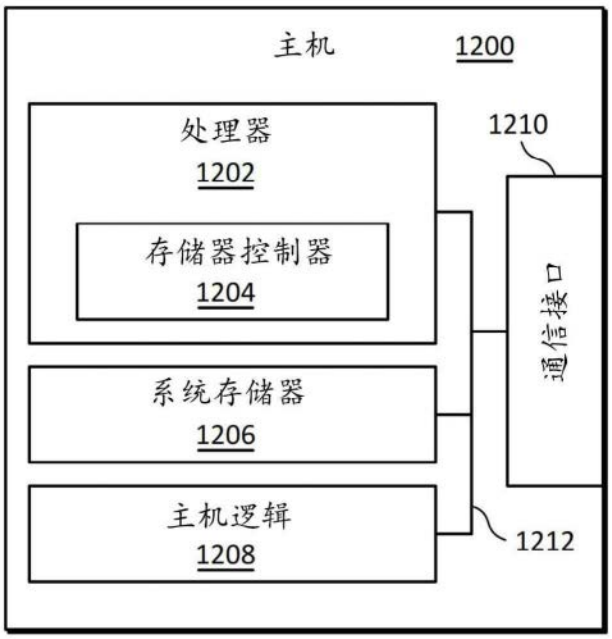

主机
1200
处理器
1202
存储器控制器
1204
系统存储器
1206
主机逻辑
1208
1210
通信接口
1212

图12

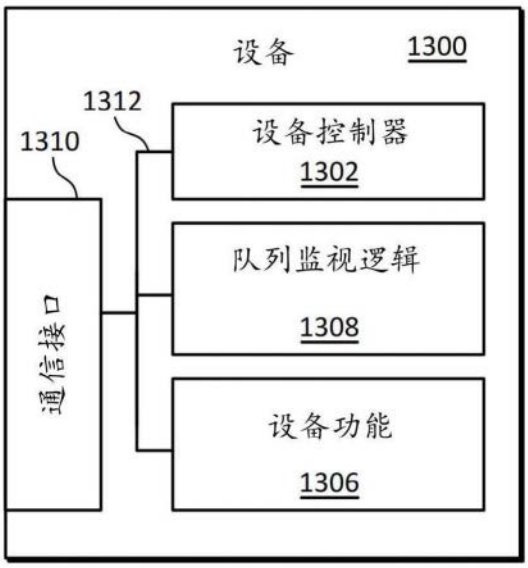

设备
1300
1312
1310
设备控制器
1302
队列监视逻辑
1308
设备功能
1306
通信接口

图13

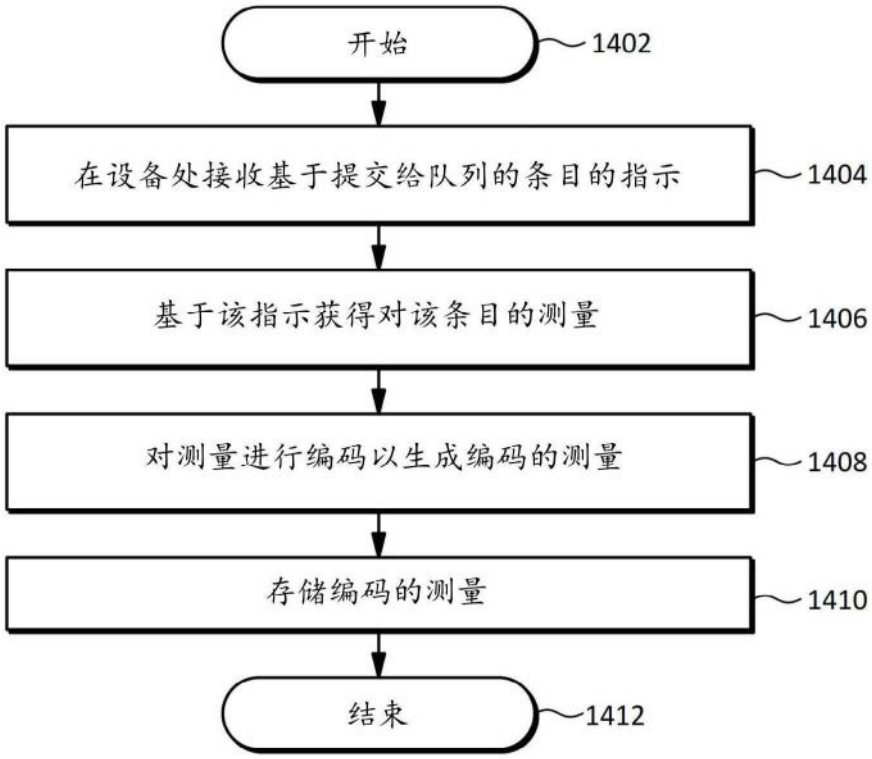

开始
1402
在设备处接收基于提交给队列的条目的指示
1404
基于该指示获得对该条目的测量
1406
对测量进行编码以生成编码的测量
1408
存储编码的测量
1410
结束
1412

图14