



(12)发明专利

(10)授权公告号 CN 104704475 B

(45)授权公告日 2018.04.27

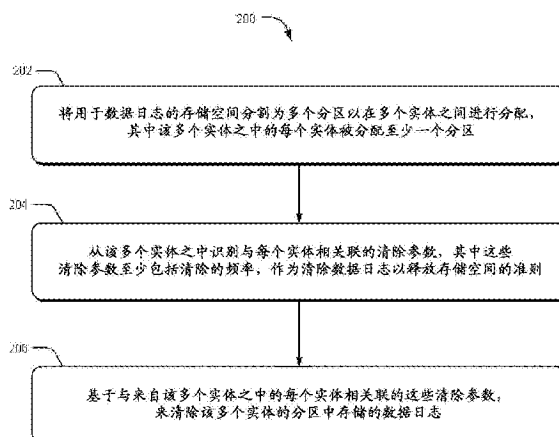
(21)申请号 201380052178.4
 (22)申请日 2013.09.17
 (65)同一申请的已公布的文献号
 申请公布号 CN 104704475 A
 (43)申请公布日 2015.06.10
 (30)优先权数据
 3109/DEL/2012 2012.10.04 IN
 (85)PCT国际申请进入国家阶段日
 2015.04.03
 (86)PCT国际申请的申请数据
 PCT/EP2013/069277 2013.09.17
 (87)PCT国际申请的公布数据
 W02014/053313 EN 2014.04.10
 (73)专利权人 阿尔卡特朗讯
 地址 法国布洛涅-比扬古
 (72)发明人 A·阿查 B·尤尼克里什南
 (74)专利代理机构 北京市金杜律师事务所
 11256
 代理人 王茂华

(51)Int.Cl.
 G06F 11/34(2006.01)
 G06F 17/30(2006.01)
 (56)对比文件
 JP 2010-244137 A,2010.10.28,
 JP 2007-241540 A,2007.09.20,
 WO 2012/025994 A1,2012.03.01,
 JP 2010-244137 A,2010.10.28,
 JP 2007-241540 A,2007.09.20,
 JP 2006-041764 A,2006.02.09,
 JP 2004-038312 A,2004.02.05,
 JP 2010-218313 A,2010.09.30,
 CN 102270225 A,2011.12.07,全文.
 US 2012/0089716 A1,2012.04.12,全文.
 CN 102411533 A,2012.04.11,全文.
 JP 2006-041764 A,2006.02.09,
 审查员 张帅领

权利要求书2页 说明书12页 附图2页

(54)发明名称
 多客户端架构中的数据日志管理

(57)摘要
 描述了用于多客户端架构中的数据日志管理的系统和方法。根据本主题,(多个)系统实施了所描述的用于高效数据日志管理的(多种)方法。该方法包括:识别与多个实体中的每个实体相关联的清除参数,其中这些清除参数表示清除在与该实体相对应的分区中所存储的数据日志的机制,并且其中这些清除参数至少包括用以释放存储空间的与数据日志相关联的关键性点。进一步地,该方法包括:基于这些清除参数来清除在该实体的分区中所存储的数据日志,其中该清除释放了存储空间。



1. 一种用于数据日志管理的方法,包括:

识别与来自多个实体之中的一个实体相关联的清除参数,其中所述清除参数至少包括与数据日志相关联的关键性点,其中重要数据日志与高关键性点相关联,其中普通数据日志与低关键性点相关联,并且其中所述清除参数表示清除由所述实体存储在数据库中的数据日志的准则;以及

基于所述清除参数来清除所述数据日志。

2. 根据权利要求1所述的方法,进一步包括:将所述数据库分割为多个分区以存储数据日志,其中所述多个分区在所述多个实体之间被分配,并且其中来自所述多个实体之中的每个实体被分配来自所述多个分区之中的至少一个分区。

3. 根据权利要求2所述的方法,其中所述分割基于与所述多个实体中的每个实体相关联的分割参数,并且其中所述分割参数定义被分配给所述多个实体中的每个实体的分区的大小。

4. 根据权利要求1所述的方法,其中所述方法进一步包括:

识别与来自所述多个实体之中的每个实体相关联的分片参数,其中所述分片参数定义数据日志成为不同大小的分片准则;以及

基于所识别的分片参数,将由来自所述多个实体之中的每个所述实体生成的所述数据日志分片为不同大小。

5. 根据权利要求4所述的方法,其中针对所述实体而被分片的所述数据日志的所述不同大小基于所述数据库的被分配给所述实体的存储大小以及由所述实体生成的所述数据日志的数目。

6. 根据权利要求1所述的方法,其中所述清除参数进一步包括:清除的频率,以及所存储的数据日志基于其而被清除的时间实例。

7. 根据权利要求2所述的方法,其中对所述数据日志的所述清除进一步基于预定义的时间持续期和被存储在所述分区中的数据日志的阈值数目,其中所述预定义的时间持续期和数据日志的所述阈值指示利用脉冲所存储的数据日志。

8. 根据权利要求2所述的方法,其中所述方法进一步包括:

识别与来自所述多个实体之中的每个实体相关联的覆写因素,其中所述覆写因素定义数据日志从一个分区到另一个分区的覆写准则;以及

基于所述覆写因素,将所述实体的所述数据日志覆写到另一个实体的未使用分区上。

9. 根据权利要求8所述的方法,其中与每个实体相关联的所述覆写因素包括以下各项中的至少一项:可用的存储空间、数据日志的关键性、以及实体的分区的共享能力。

10. 根据权利要求1所述的方法,其中所述方法进一步包括:基于所述多个实体的存储空间的变化要求,动态地分割所述多个实体之中的所述存储空间。

11. 一种用于数据日志管理的系统(102),包括:

处理器(112);

耦合至所述处理器(112)的清除模块(126),被配置为:

识别与来自多个实体之中的一个实体相关联的清除参数,其中所述清除参数至少包括与数据日志相关联的关键性点,其中重要数据日志与高关键性点相关联,其中普通数据日志与低关键性点相关联,并且其中所述清除参数表示清除由所述实体存储在数据库中的数

据日志的准则;以及

基于所述清除参数来清除所述数据日志。

12. 根据权利要求11所述的系统(102),进一步包括分割模块(122),所述分割模块(122)被配置为将所述数据库分割为多个分区以存储数据日志,其中所述多个分区在所述多个实体之间被分配,并且其中来自所述多个实体之中的每个实体被分配来自所述多个分区之中的至少一个分区。

13. 根据权利要求11所述的系统(102),其中所述系统(102)进一步包括分片模块(124),所述分片模块(124)被配置为:

识别与来自所述多个实体之中的每个实体相关联的分片参数,其中所述分片参数定义数据日志成为不同大小的分片准则;以及

基于所识别的分片参数,将由来自所述多个实体之中的每个所述实体生成的所述数据日志分片为不同大小。

14. 根据权利要求12所述的系统(102),其中所述系统(102)进一步包括覆写模块(128),所述覆写模块(128)被配置为:

识别与来自所述多个实体之中的每个实体相关联的覆写因素,其中所述覆写因素定义数据日志从一个分区到另一个分区的覆写准则;以及

基于所述覆写因素,将所述实体的所述数据日志覆写到另一个实体的分区上,其中另一个实体的所述分区未使用。

15. 根据权利要求12所述的系统(102),其中所述分割模块(122)进一步被配置为,基于所述多个实体的存储空间的变化要求,动态地分割所述多个实体之中的所述存储空间。

16. 一种用于数据日志管理的设备,包括:

用于识别与来自多个实体之中的一个实体相关联的清除参数的装置,其中所述清除参数至少包括与数据日志相关联的关键性点,其中重要数据日志与高关键性点相关联,其中普通数据日志与低关键性点相关联,并且其中所述清除参数表示清除由所述实体存储在数据库中的数据日志的准则;以及

用于基于所述清除参数来清除所述数据日志的装置。

17. 根据权利要求16所述的设备,进一步包括:

用于识别与来自所述多个实体之中的每个实体相关联的分片参数的装置,其中所述分片参数定义数据日志成为不同大小的分片准则;以及

用于基于所识别的分片参数将由来自所述多个实体之中的每个所述实体生成的所述数据日志分片为不同大小的装置。

多客户端架构中的数据日志管理

技术领域

[0001] 本主题涉及计算系统,并且特别地但不排他地涉及数据日志在计算系统上的存储。

背景技术

[0002] 在应用的执行或运行期间,众多事件可能发生。例如,该应用可以执行步骤许多次。该应用可以运行多次迭代中的循环。它可以建立和关闭很多连接。线程可以被发起和完成。一个事件可以触发其他事件,这些其他事件可以又触发其他事件。类似地,为了执行的目的,系统的多个实体或硬件组件可以相互通信并且交换数据。这样的数据可能准确地被递送,或者可能被丢弃,或者在交换期间被损坏。因此,在数据的交换或者应用的执行期间,成功的、不成功的、或者未完成的多个事件实时地发生。

[0003] 在事件是丰富的并且实时发生的情形中,为了准确性、排错、以及事件的及时完成的目的而监测这些事件的状态并不总是在物理上可能的。进一步地,为了参考任何先前完成的事件,仅仅依赖于对事件的发生的回溯来收集事件的事实以及结果是麻烦且困难的任務。此外,在某些情形中,难以再现给定环境中所发生的事件。因此,一般而言,与在不同的时间实例发生的不同事件有关的细节被认为并且被存储为数据日志。这样的数据日志包括任何给定进程期间已经发生或者正在发生的事件的历史记录。

[0004] 在应用和进程的执行涉及丰富事件的现今场景中,对所有这些事件保持物理跟踪是不可能的,并且此外,针对这样的事件的数据日志的物理维护也是不可行的。因此,与事件相关联的数据日志一般电子地被记录并且存储在存储器组件中用于未来的目的。数据日志能够被用于各种目的,诸如用于排错的目的,其中用户可以从详查被记入日志的事件而受益。类似地,用户可以进一步从查看与事件执行期间发生的这些事件中的任何事件有关的信息而受益,并且可以在诊断问题或者回答与执行的结果有关的疑问中受到辅助。此外,例行的数据日志分析对于识别安全事故、政策违反、欺诈行为、以及操作问题是有益的。而且,数据日志在审计、法医分析 (forensic analysis)、支持内部调查、建立基线、以及识别操作趋势和长期问题的时候是有用的。

发明内容

[0005] 这个概述被提供以介绍与根据本主题的一种实施方式的在多客户端架构中的数据日志的管理有关的概念。这个概述不意图为识别所要求保护的主题的必要特征,也不意图用于在确定或限制所要求保护的主题的范围中使用。

[0006] 在一个实施方式中,描述了一种用于数据日志管理的方法。该方法包括:识别与多个实体中的每个实体相关联的清除参数,其中这些清除参数表示清除在与该实体相对应的分区中所存储的数据日志的机制,并且其中这些清除参数至少包括关键性点,作为清除这些数据日志以释放存储空间的准则。进一步地,该方法包括基于这些清除参数来清除在该实体的分区中所存储的数据日志。

[0007] 在另一实施方式中,该方法进一步包括:识别与来自该多个实体之中的每个实体相关联的分片参数,其中这些分片参数定义数据日志成为不同大小的分片准则;以及基于所识别的分片参数,将由来自该多个实体之中的每个实体生成的数据日志分片为不同大小,其中该分片允许了对存储空间的优化利用。

[0008] 进一步地,该方法还包括:识别与来自该多个实体之中的每个实体相关联的覆写因素,其中这些覆写因素定义数据日志从一个分区到另一个分区的覆写准则;以及基于这些覆写因素,将该实体的数据日志覆写到另一个实体的分区上,其中另一个实体的该分区未使用。

[0009] 在一个实施方式中,描述了一种用于多客户端架构中的数据日志管理的系统。该系统包括耦合至处理器的清除模块,该清除模块被配置为:识别与多个实体中的每个实体相关联的清除参数,其中这些清除参数表示清除在与该实体相对应的分区中所存储的数据日志的机制,并且这些清除参数至少包括与数据日志相关联的关键性点。该清除模块进一步被配置为基于这些清除参数来清除在该实体的分区中所存储的数据日志,其中该清除释放了存储空间。

[0010] 在另一实施方式中,描述了一种计算机可读程序代码已经被具体化在其上的非瞬态计算机可读介质,该计算机可读程序代码用于执行一种方法。该方法包括:识别与多个实体中的每个实体相关联的清除参数,其中这些清除参数表示清除在与该实体相对应的分区中所存储的数据日志的机制,并且其中这些清除参数至少包括关键性点,作为清除这些数据日志以释放存储空间的准则。进一步地,该方法包括基于这些清除参数来清除在该实体的分区中所存储的数据日志。

附图说明

[0011] 参考随附的示图来描述详细的描述。在各示图中,参考标号的最左边的(多个)数字标识了该参考标号首次出现的示图。贯穿这些示图,相同的标号被用来参考相似的特征和组件。现在通过仅是示例的方式并且参考随附的示图来描述根据本主题的实施例的系统 and/或方法的一些实施例,在随附的示图中:

[0012] 图1图示了根据本主题的一个实施例的实施数据日志管理系统(DLMS)的示例性网络环境;

[0013] 图2图示了根据本主题的一个实施例的用以实施数据日志管理的方法。

[0014] 在本文档中,词语“示例性的”在本文中用来意指“用作示例、实例、或者例证”。本文中被描述为“示例性的”本主题的任何实施例或者实施方式不一定被解释为相对于其他的实施例是优选的或者有利的。

[0015] 本领域的技术人员应当意识到,本文中的任何框图表示了体现本主题的原理的说明性系统的概念视图。类似地,将意识到,任何流程图、流程图、状态转换图、伪代码等表示了可以基本上被表示在计算机可读介质中并且由计算机或处理器如此执行的各种进程,不论这样的计算机或处理器是否明确地被示出。

具体实施方式

[0016] 描述了用以允许数据日志的高效存储的用于多客户端架构中的数据日志管理的

系统和方法。这些方法能够被实施在通过各种网络进行通信的各种计算设备中。尽管本文的描述参考了通信网络的系统,但是如将由本领域的技术人员所理解的,虽然会有一些变化,这些方法和系统可以被实施在能够存储数据日志的其他系统中。

[0017] 数据日志一般被存储在系统的存储器组件中,来保留与发生在不同时间实例的不同事件有关的细节。这些数据日志包括已经发生的事件的历史记录。在一种组织中,数据日志能够由许多来源和多个事件所生成,这些来源和事件包括:安全软件,诸如反病毒软件、防火墙、和入侵检测及预防系统;服务器、工作站和联网装备上的操作系统;以及应用。因为针对每个来源的数据日志为了未来使用而被存储和维护,所以由该组织所管理的数据日志的数目、容积和品种是大量的。

[0018] 由不同来源针对不同事件生成的数据日志被存储用于传输、分析和处理,或者倾向于销毁。归因于存储数据日志的服务器处的有限存储空间,数据日志还定期地被销毁以便更新存储器用于存储新的数据日志。例如,一个组织可以利用其服务器上的100GB的空间以用于存储不同应用的操作和各种进程的处理一天天地生成的日志。如果在各实例中该组织的全部各种事件每天生成接近2GB的数据日志,则有可能该服务器可能只剩下少于2GB的数据存储空间,诸如已被利用的空间是99GB。在这样的情形中,为了有效地存储下一天生成的全部数据日志,该服务器可能不得不销毁一些先前存储的或者不必要的数据日志,从而可用的自由空间现在大于2GB。

[0019] 为了管理极度大量的所生成的数据日志并且通过销毁先前存储的或者不必要的数据日志来释放存储空间,处置数据日志的服务器现今利用数据日志管理的技术,也称为日志管理技术。

[0020] 常规地被利用的数据日志管理技术基于不同业务进程的不时变化的要求,来管理对数据日志的存储、处理、传输、和清除。特别地,为了清除数据日志并且释放存储空间,这些日志管理技术基于存储日期来删除数据日志,即更早被存储的数据日志首先被删除,来为将被存储的新数据日志准备存储空间。这样的清除技术类似于先入先出(FIFO)的技术,其中在时间上更早被存储的数据先于在稍后的时间实例所存储的数据而被释放。尽管这样的清除技术允许存储空间的释放,然而在清除之前没有考虑到与数据日志相关联的关键性的任何重要性。这可能经常导致关键数据日志的破坏,这些关键数据日志相对于一些非关键数据日志而言对于未来使用可能是有用的,而这些非关键数据日志可能仍然被存储在存储空间中。

[0021] 进一步地,如早前所描述的,不同的应用和来源可能生成将被存储在一个存储空间上的数据日志。尽管存储来自多个来源的数据日志,但是一个特定应用的数据日志可能比另一个特定应用的数据日志更加关键。然而,尽管管理来自不同来源和各种应用的日志,但是FIFO方法以相同的方式对所有的应用进行加权,由此不允许向具体应用的数据日志提供任何特定的优先级,或者不允许向该应用提供任何具体的存储容量。

[0022] 根据本主题的一种实施方式,本文描述了使重要且关键的数据日志的存储优先的数据日志管理的方法。所描述的方法一方面允许了来自不同应用的数据日志的高效存储,另一方面提供了创建用于新数据日志的自由存储空间的高效清除机制。根据本主题的所描述的实施方式,可以从不同的应用接收数据日志,这些应用诸如安全软件,包括反病毒软件、防火墙、和入侵检测及预防系统;服务器、工作站和联网装备上的操作系统。进一步地,

数据日志可以涉及调用日志、安全日志、联网日志、和通信日志。尽管关于通信日志描述了本描述,但是如将由本领域的技术人员所理解的,虽然有一些变化,所描述的技术也能够被实施在其他类型的数据日志中。

[0023] 进一步地,可以通过不同的服务器和用户处置/利用的不同功能来接收数据日志,诸如邮件服务器、中央目录服务器、数据库服务器、文件服务器、打印服务器、web服务器、应用服务器、笔记本、平板设备、网络接入适配器。

[0024] 在本主题的一个实施方式中,用于数据日志的存储空间被划分为多个分区,其中每个分区专用于存储一个实体的数据日志,该实体包括应用或者个体的用户。每个实体还可以由为了存储数据日志的目的而协作的应用或用户的组合所形成。在所述实施方式中,每个分区在被用于一个特定实体(诸如一个用户或者一个应用)时,被提供具有能够被用于存储该特定实体的数据日志的预定义的存储空间。例如,如果三个应用将它们的数据日志存储在一个组织中的一个共同服务器处,则根据本主题的所述实施方式,这个服务器的存储空间可以被分割为三个分区。将存储空间划分为不同的分区可以允许对该存储空间的基于各种实体的管理,从而不同的数据日志管理技术能够个别地被实施在每个分区上。

[0025] 在本主题的一个实施方式中,存储空间的分割可以基于不同应用的要求,诸如整个存储空间可以不等同地被分割,并且不同的应用可以利用不同量的存储空间用于数据日志。例如,在这三个应用之中,基于它们的具体要求,整个存储空间的30%可以被提供给第一应用,该空间的60%可以被提供给第二应用,并且最后,剩余的10%可以被分配给第三应用。特定于每个应用的具体要求可以由数据日志管理系统来确定,数据日志管理系统可以被配置为基于所确定的要求来分割存储空间。

[0026] 进一步地,基于可用存储空间的分区,用于每个实体(诸如所描述的场景中的这三个应用)的数据日志管理技术能够是不同的,并且基于每个应用的要求。换句话说,基于每个分区中的数据日志的优先级和重要性,用于不同分区的数据日志管理(诸如清除技术、存储技术)能够变化以适合业务要求。例如,在三个不同应用的数据日志被存储在三个不同数据分区处的上面所描述的情形中,可以使得一个应用相比于其他时间实例而言在一天中的一个特定时间段期间存储关键数据。在这样的情形中,可以利用使得足够的存储空间在所要求的特定时间段是可用的技术,来实施存储这个特定应用的数据日志的数据分区。将会理解,在这样的情形中,用于其他两个应用的数据日志管理的技术可能与第一应用不同。

[0027] 类似地,如上面所描述的,基于不同应用的具体要求,数据日志清除针对不同应用能够是不同的。在本主题的一个实施方式中,每个分区中存储的数据日志它们自身能够被分片为更小的大小,以增加将最大的数据日志存储在每个应用的分区中的概率。例如,在应用生成大小70MB的数据日志并且被分配给该应用的分区的可用总空间具有550MB的场景中,能够被数据日志的存储所填充的最大空间将是490MB并且剩余的60MB空间将总是未被利用。因此,在这样的场景中,将数据日志分片为更小大小的数据日志,这些分区就能够高效地被利用。在本示例中,70MB的数据日志可以被分片为60MB和10MB,从而大小60MB的数据日志的片段能够被容纳在该分区的存储的未被利用的部分中。该分片使得该存储的完整利用成为可能。进一步地,对数据日志的分片还允许了高效的处理,由此避免了归因于处理更大数据日志的任何性能影响。

[0028] 此外,在所述实施方式中,数据日志能够基于具体应用的要求而被分片,并且因此

能够针对不同的应用以不同的形式而被实施。也就是说,可以确定的是,对于一个应用而言,为了对空间的高效利用,分片应当被完成到每数据日志10MB的大小。然而,对于另一个应用而言,为了高效的存储空间利用,分片可以被限制为8MB。进一步地,尽管已经描述了能够在不同的分区中实施分片,但是本领域的技术人员将会理解,在存储空间没有在不同实体之间进行分区并且各种应用和用户的数据日志共同被存储的情形中,数据日志的分片也能够被执行。

[0029] 在另一实施方式中,能够通过利用数据日志清除的不同技术的方式,来变化针对不同应用的数据日志管理。例如,在上面所提到的场景中,为了确保在特定时间段存储空间可用于存储针对第一应用的数据日志,清除数据日志的方法可以针对该特定的数据分区而被修改,使得数据日志清除相比于在任何其他时间段的平常清除而言发生在该特定时间段之前。换句话说,针对该存储空间的清除实例或者清除发生的时间可以被修改以适合该数据日志存储的要求。

[0030] 因此,基于存储空间的分割和经分割的空间向不同实体的分配,基于不同实体的个别的具体要求,不同的数据日志管理技术能够被实施用于不同实体。将会理解,每个实体的具体要求能够被用户定义以实施数据日志管理技术。

[0031] 类似地,在本主题的另一实施方式中,在所分配的存储空间耗尽的情形中,被分配给其他应用的数据分区的未使用存储空间也可以由任何具体应用利用以用于存储关键数据日志的目的。例如,在早前描述三个应用之中,如果第二应用正在保存关键数据日志并且被分配给第二应用的存储空间被耗尽,则可以利用被分配给其他两个应用的未使用的存储空间用于由第一应用对这些关键数据日志的存储。

[0032] 根据本主题的一种实施方式,将数据日志存储到被分配给其他应用和实体的分区上基于与每个分区相关联的覆写因素(overwriting factor)。该覆写因素可以包括如下的参数,诸如可用的存储空间、数据日志的关键性、以及每个实体的分区可以基于其而在不同实体之间进行共享的分区的共享能力。已经关于附图的具体实施例详细地解释了在不同场景中对这些参数的实施和使用。

[0033] 根据本主题的另一实施方式,还可以基于可用的存储空间来动态地修改对存储空间的分割,从而基于各实体不时修改的要求来改变经分割的存储空间的大小。例如,在总可用空间是30GB并且存储空间在各应用之间的分割基于30%、20%和50%准则的情形中。然而,在如果第三应用的要求上升超出50%并且存储空间对于前两个应用是可用的情形中,则分割可以动态地被修改为20%、10%和70%以将更多的存储空间提供给第三应用。在一个实施方式中,数据日志管理系统可以被配置为重新评估可用空间,并且可以基于要求来动态地执行重新配份(re-portioning)。

[0034] 基于对存储空间的动态分配,为了有效的数据日志管理,能够考虑到不同实体的变化的要求。在本主题的另一实施方式中,用于各分区的清除机制基于与分区相关联的实体的选择。也就是说,对于将数据日志存储到它们被分配的分区实体而言,能够基于不同的因素按照各实体的要求来定制清除机制。在一个实施方式中,这些实体可以基于数据日志已经被存储的持续期来清除数据。

[0035] 本领域的技术人员将会理解,数据日志在不同的实例并且针对不同的目的而被存储。某些数据日志在每个预定时间段之后被存储,以将工作系统的状态记入日志。其他的数

据日志是当正在被监测的应用或系统的运转中存在任何特定错误时所生成的错误日志。一般而言,错误数据日志在短的突发时间段内生成,短的突发时间段归因于错误的突然发生,错误的突然发生归因于某些进程的不寻常事件或失效。这样的错误数据日志在稍后的时间实例对于该错误的排错并且对于更详细地理解该错误是重要的。并且因此,还将会意识到,错误数据日志比平常数据日志具有更为关键的重要性。

[0036] 根据本主题的一种实施方式,在短的突发时间段中、或者利用脉冲、或者当最大数据日志被存储的时间持续期中所存储的数据日志,在更长的持续期内被维持并且不被清除,而其他的数据日志更早地被清除,由此释放了存储空间。不是在短的突发期间被存储并且很可能是平常数据日志的数据日志能够以对关键数据(诸如错误数据日志)的较少破坏而被清除。将会进一步意识到,为了高效且恰当的数据日志管理,能够按照各实体的选择来变化清除的时间和清除的频率。

[0037] 在本主题的又另一实施方式中,基于与每个数据日志相关联的关键性点来清除数据日志。例如,极度重要和关键的数据日志(诸如错误数据日志)可以与最高的关键性点相关联,并且并不是非常重要的数据日志(诸如平常的数据日志)可以与最低的关键性点相关联。基于每个数据日志的关键性点,根据所述实施方式,可以完成对数据日志的清除。例如,每个实体可以基于与数据日志相关联的关键性点,来定义将首先被清除的数据日志。一个实体可以定义,相比于清除其他的数据日志,首先清除具有最低关键性点的数据日志。类似地,其他实体可以选取首先清除具有最高关键性点的数据日志。

[0038] 因此,基于与不同数据日志相关联的关键性点,清除机制可以被选取并且被应用于释放存储空间。这可以允许不同的实体定义在不同时间实例的并且具有不同优先级的对不同类型的数据日志的清除。

[0039] 应当注意,本描述仅说明了本主题的原理。将因此意识到,本领域的技术人员将能够设计出尽管本文没有明确描述但是体现了本主题的原理并且被包括在其精神和范围内的各种布置。此外,本文所记载的所有示例主要意图为明确地仅用于教导目的,以辅助读者理解本发明的原理以及由(多位)发明人为了促进本领域所贡献的概念,并且将被解释为没有对这样具体记载的示例和条件作出限制。此外,记载本发明的原理、方面、和实施例的本文中的所有陈述,以及它们的具体示例,意图为涵盖它们的等价物。

[0040] 已经关于图1和2详细地解释了提供虚拟分发列表的系统和方法应该被实施的方式。尽管所描述的用于提供虚拟分发列表的系统和方法的各方面能够被实施在任何数目的不同计算系统、传输环境、和/或配置中,但是在以下的(多个)示例性系统的上下文中描述了各实施例。

[0041] 本领域的技术人员还将意识到,如本文所使用的词语“在……期间”、“在……同时”以及“当……时”不是意味着一经发起的动作就立即发生一个动作的确切术语,而是在该初始动作与由该初始动作所发起的反应之间可以存在某个小的但合理的延迟,诸如传播延迟。

[0042] 图1图示了根据本主题的一个实施例的用于实施数据日志的实施了数据日志管理系统(DLMS) 102的网络环境100。为了解释的目的,该DLMS在后文中被称为系统102。本文所描述的系统102能够被实施在包括各种网络设备的任何网络环境中,这些网络设备包括路由器、网桥、服务器、计算设备、存储设备,等等。在一个实施方式中,系统102通过网络106而

个别地连接至一个或多个客户端设备104-1、104-2、104-3、……、104-N,并且在后文中共同地被称为(多个)客户端设备104。这些客户端设备104可以包括多个应用,这些应用可能正在运行以执行如由不同用户和个别组织所要求的若干功能。进一步地,这些客户端设备104还可以是任何网络元件的内部组件,诸如3G网络,其中这些客户端设备104是BTS的内部组件,诸如控制器、调制解调器板、以及无线电板。

[0043] 系统102能够被实施为各种服务器和通信设备。能够实施所描述的(多种)方法的服务器和计算系统包括但不限于:邮件服务器、中央目录服务器、数据库服务器、文件服务器、打印服务器、web服务器、应用服务器,等等。系统102还可以被实施为计算设备,诸如膝上型计算机、台式计算机、笔记本、工作站、大型计算机、服务器,等等。本文所描述的系统102还能够被实施在包括各种网络设备的任何网络环境中,这些网络设备包括路由器、网桥、服务器、计算设备、存储设备,等等。

[0044] 这些客户端104可以被实施为但不限于:台式计算机、手持式设备、膝上型或者其他便携式计算机、平板计算机、移动电话、PDA、智能电话、安全仪器、监控装备,等等。进一步地,这些客户端104可以包括如下的设备,这些设备能够交换数据以提供通向不同的通信设备和计算系统的连接性。这样的设备可以包括但不限于:数据卡、移动适配器、无线(WiFi™)适配器、路由器、无线调制解调器、无线通信设备、无绳电话、无线本地环路(WLL)站,等等。因为客户端设备104可以是静止的或者移动的,所以它们也可以被理解为移动台、终端、接入终端、订户单元、站,等等。

[0045] 网络106可以是无线网络或者有线网络,或者它们的组合。网络106能够是彼此互连并且用作单个大网络(例如,互联网或内联网)的个体网络的集合。这样的个体网络的示例包括但不限于:全球移动通信系统(GSM)网络、通用移动通信系统(UMTS)网络、个人通信服务(PCS)网络、时分多址(TDMA)网络、码分多址(CDMA)网络、下一代网络(NGN)、公共交换电话网络(PSTN)、以及综合服务数字网络(ISDN)。取决于技术,网络106包括各种网络实体,诸如网关、路由器;然而,这些细节为了容易理解已经被省略。

[0046] 在一个实施例中,系统102与数据库108相关联,数据库108存储从来自不同地理区域的客户端设备104接收的数据日志。尽管数据库108被图示在系统102外部,但是将理解,数据库108也可以在系统102内部。进一步地,数据库108能够被实施为,例如,单个储存库、分布式储存库、或者分布式储存库的集合。

[0047] 在一个实施方式中,系统102包括(多个)处理器112。处理器112可以被实施为一个或多个微处理器、微计算机、微控制器、数字信号处理器、中央处理单元、状态机、逻辑电路、和/或基于操作指令来操控信号的任何设备。除了其他的能力之外,该(些)处理器还被配置为取回并执行存储器中存储的计算机可读指令。

[0048] 可以通过使用专属硬件以及能够执行软件的硬件联合适当软件,来提供附图所示出的各种元件的功能,包括被标记为“(多个)处理器”的任何功能框。当由处理器提供时,这些功能可以由单个专属的处理器、由单个共享的处理器、或者由多个个体处理器(其中的一些可以是共享的)来提供。此外,对术语“处理器”的明确使用不应当被解释为排他地指代能够执行软件的硬件,并且可以隐含地不带限制地包括:数字信号处理器(DSP)硬件、网络处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、用于存储软件的只读存储器(ROM)、随机访问存储器(RAM)、非易失性存储器。其他常规的和/或定制的硬件也可以被包

括。

[0049] 此外,系统102包括(多个)接口114。这些接口114可以包括允许系统102与网络106的实体进行交互或者彼此进行交互的各种软件和硬件接口。这些接口114可以促进各种各样的网络和协议类型之内的多个通信,这些网络和协议类型包括有线网络(例如,LAN、电缆等)和无线网络(例如,WLAN、蜂窝网络、基于卫星的网络等)。

[0050] 在本主题的另一实施例中,系统102还可以包括存储器116。存储器116可以耦合至处理器112。存储器116能够包括本领域中已知的任何计算机可读介质,包括例如:易失性存储器,诸如静态随机访问存储器(SRAM)和动态随机访问存储器(DRAM);和/或非易失性存储器,诸如只读存储器(ROM)、可擦除可编程ROM、闪存、硬盘、光盘、以及磁带。

[0051] 进一步地,系统102可以包括(多个)模块118和数据120。模块118和数据120可以耦合至处理器112。除了其他的事物之外,这些模块118还包括执行特定的任务或者实施特定的抽象数据类型的例程、程序、对象、组件、数据结构,等等。这些模块118也可以被实施为,(多个)信号处理器、(多个)状态机、逻辑电路、和/或基于操作指令来操控信号的任何其他设备或者组件。

[0052] 进一步地,这些模块118能够被实施在硬件中、由处理单元执行的指令中,或者由它们的组合来实施。该处理单元能够包括计算机、处理器、状态机、逻辑阵列、或者能够处理指令的任何其他适合的设备。该处理单元能够是通用处理器,该通用处理器执行指令以促使该通用处理器执行所要求的任务,或者该处理器单元能够专用于执行所要求的功能。

[0053] 在本主题的另一方面中,这些模块118可以是机器可读指令(软件),其当由处理器/处理单元执行时,执行所描述的功能中的任何功能。这些机器可读指令可以被存储在电子存储器设备、硬盘、光盘、或者其他的机器可读存储介质或非瞬态介质上。在一个实施方式中,这些机器可读指令还能够经由网络连接而被下载到该存储介质。

[0054] 在一种实施方式中,(多个)模块118包括分割模块122、分片模块124、清除模块126、覆写模块和(多个)其他模块130。(多个)其他模块130可以包括对系统102所执行的应用或功能进行补充的程序或经编码的指令。在所述实施方式中,数据120包括实体数据132、数据日志数据134、关键性数据136、以及其他数据138。除了其他的事物之外,其他数据138还可以用作储存库,该储存库用于存储作为(多个)模块118中的一个或多个模块的执行结果而被处理、被接收、或者被生成的数据。尽管数据120被示出在系统102内部,但是可以理解,数据120能够位于外部储存库(附图中未示出)中,该外部储存库可以耦合至系统102。系统102可以通过(多个)接口114与该外部储存库进行通信以从数据120中获得信息。

[0055] 如之前所提到的,系统102被配置为针对不同的实体来管理数据日志,以允许数据日志的高效存储和它们的清除机制。数据日志可以由客户端设备104不时地由于各种原因而提供。某些数据日志可能在每个预定时间段之后被存储,以存储不同的客户端设备和应用的状态。然而,其他的数据日志可以在关键情形中被存储,这些关键情形诸如任何客户端设备中的故障,或者任何应用中的意外错误。这样的数据日志还可以包括错误数据日志,这些错误数据日志存储了错误在其期间发生的不同准则,诸如错误的可能原因、当错误发生时设备的状态、以及示出错误何时发生的时间戳。在本主题的一个实施方式中,由客户端设备104提供的所有数据日志都由系统102存储在数据库108中。

[0056] 为了数据日志的高效存储,在本主题的一个实施方式中,系统102的分割模块122

被配置为,将数据库108的存储空间在不同实体之间进行分割。每个实体可以是单个客户端设备104,或者可以是多个客户端设备104的组合。进一步地,每个实体可以包括一个应用或者可以包括一组应用。在所述实施方式中,分割模块122可以基于某些分割参数来分割数据库108,这些分割参数可以定义不同实体的存储空间要求。也就是说,每个实体所要求的存储空间量可以由与这些实体中的每个实体相关联的分割参数来定义。在一个实施方式中,每个实体的要求可以基于以下各项中的一项或多项:与每个实体相关联的关键性、由该实体生成的数据日志的大小、以及该实体生成这些数据日志的频率。因此,基于这些准则,针对每个实体的分割参数可以被确定并且与该实体相关联。

[0057] 在一个实施方式中,分割模块122可以基于与这些实体中的每个实体相关联的分割参数,将数据库108分割为多个分区。与这些实体相关联的这些分割参数可以由分割模块122接收,以确定用于每个实体的存储空间。也就是说,分割模块122可以基于与这些实体中的每个实体相关联的分割参数,在数据库108中将分离的存储空间分配给每个实体。例如,分割模块122可以将数据库108的存储空间分割到5个不同的实体中。每个实体可以具有与之相关联的关联分割参数。基于与这些实体中的每个实体的关联分割参数,分割模块122可以将数据库108分割,使得第一实体被分配可用存储空间的20%份额,第二实体被分配可用存储空间的50%份额,并且剩余的实体每个都被分配10%的份额。

[0058] 将会理解,可能不是数据库108的整个存储空间都由系统102利用以用于存储数据日志的目的,并且相应地,仅是被利用以用于数据日志存储的目的的存储空间可以由分割模块122在不同的实体之间进行分割。

[0059] 将数据库108的存储空间分割为用于不同实体的不同分区,可以允许系统102基于这些实体的具体要求来将数据日志管理的不同技术应用至不同的分区。

[0060] 在本主题的一个实施方式中,系统102的分片模块124被配置为,为了在所分配的分区上的高效存储,而将各实体所生成的数据日志分片为更小大小的数据日志。数据日志的分片可以针对被存储到数据库108上的所有数据日志而进行,或者可以基于个别实体的具体要求。在所述实施方式中,数据日志的分片可以基于被分配给每个实体的存储空间的大小以及由每个实体生成的数据日志的大小。例如,分片模块124可以确定对五个实体中的如下三个实体的数据日志进行分片,对于这三个实体而言,所生成的数据日志具有大于预定大小的尺寸。

[0061] 进一步地,也可能发生的是,这些实体可以定义并提供输入给分片模块124,以将它们生成的数据日志分片。在这样的场景中,分片模块124可以针对已经提供了与分片有关的任何输入的实体,来分割数据日志。在一个实施方式中,这些实体可以用分片参数的形式将输入提供给分片模块124。这些分片参数可以包括:数据日志将被分片到的大小、超出其分片就应当发生的数据日志的任何阈值大小,并且类似地可以由分片模块124基于其来执行对数据日志的分片。

[0062] 在一个实施方式中,分片模块124被配置为,基于被分配给一个实体的分区大小以及由该实体生成的数据日志的数目,来将数据日志分片。在所述实施方式中,分片模块124可以基于分区大小与数据日志的数目的比率,来确定经分片的数据日志的大小。换句话说,经分片的数据日志的大小能够表示为:

$$[0063] \quad \text{Size of Fragmented Data Log} = \frac{\text{Size of Partitan}}{\text{Number of Data logs Generated}} \quad (1)$$

[0064] 基于上面的定义关系(1),将会理解,分片模块124可以用不同的大小将用于不同实体的数据日志分片,以高效地存储数据日志并且更好地利用存储分区空间。

[0065] 将会理解,被存储到数据库108上的数据日志将以规律的间隔而被清除来释放存储空间,以用于存储新近的数据日志。然而,不应当以其他不重要的数据日志为代价来进行对重要数据日志的清除。因此,系统102利用了用于在清除重要的数据日志之前清除不太重要的数据日志的高效清除机制。

[0066] 在本主题的另一实施方式中,清除模块126被配置为高效地清除数据库108的存储空间中存储的数据日志。清除模块126被配置为,基于每个分区的实体,以不同的方式来清除不同分区的数据日志。在一个实施方式中,对一个分区中的数据日志的清除可以基于这些实体的要求。诸如,一个实体可以在每天的任何特定的给定时间实例存储敏感的数据日志,比如在上午11:00到中午12:00之间存储每个支行的数据日志的银行应用。在每天的这个时间实例期间存储的数据日志对组织是重要的,并且可以要求被分配给这个实体的分区中的存储空间以使得没有数据日志被错过。在这样的场景中,清除模块126可以确定,每天在上午11:00之前清除被分配给这个实体的分区中存储的较旧的数据日志,以确保可用的存储空间。将会理解,在除了上午11:00之外的不同时间实例,可以针对其他分区来清除数据日志。

[0067] 几个其他的实体可以在每个预定义的时间段之后存储数据日志,以存储各应用和客户端设备104的状态。然而,在应用的错误和失效的情形中,可以由该实体利用急始脉冲(sudden impulse)来存储数据日志。换句话说,在这样的意外错误或失效期间的短时间跨度内,可以生成并且存储许多数据日志。因此,对于不同的实体,可以定义预定义的时间持续期和日志的阈值,可以基于其来确定这些数据日志是否利用急始脉冲而被存储。例如,如果对于一个实体,预定义的时间持续期是5分钟并且数据日志的阈值是50,则在多于50个数据日志在5分钟的跨度内被存储的情形中,这些数据日志可以被标记为利用急始脉冲而被存储的数据日志。进一步地,因为这样的数据日志对于进一步的分析和故障发现也可能是重要的,所以在一个实施方式中,清除模块126可以确定,在利用急始脉冲在短时间跨度中被存储的数据日志之前清除以平常间隔被存储的数据日志。

[0068] 在另一实施方式中,清除模块126可以基于与不同实体的数据日志相关联的关键点来清除数据日志。如之前所描述的,不同的数据日志可以基于与它们相关联的重要性而具有不同的关键性。也就是说,不同的数据日志可以存储如下的信息,该信息可以具有与存储这些数据日志的实体有关的变化的重要性。因此,与不同的数据日志相关联的关键性点可以是不同的。清除模块126可以基于相关性将关键性点与数据日志相关联,例如,具有较低重要性的数据日志,诸如在每个预定时间段之后被生成并被存储的平常的数据日志,可以与低关键性点相关联。类似地,与高重要性相关联的数据日志,诸如错误数据日志,可以与高关键性点相关联。在所述实施方式中,清除模块126可以确定,在高关键性点的数据日志之前清除具有低关键性点的数据日志,以释放该实体的分区中的存储空间。

[0069] 在一个实施方式中,清除模块126还可以基于实体的可以根据清除参数而被定义的要求,来清除来自被分配给实体的分区的数据日志。每个实体可以定义清除参数,以定义

清除数据日志的时间、清除数据日志的频率、将首先被清除的数据日志的关键性点、以及在其期间所存储的数据日志应当被清除或者不应当被清除的时间实例。因此,基于不同实体的要求,数据日志清除机制能够由系统102独立地实施在各种分区上。

[0070] 根据本主题的一种实施方式,系统102的覆写模块128被配置为管理数据库108的存储空间,使得所分配的分区中的存储空间的耗尽期间,由每个实体存储在它们各自分区中的数据日志也能够被存储在其他分区中。在所述实施方式中,覆写模块128可以确定其他实体的分区中的未使用存储空间,以存储其存储空间接近耗尽的任何实体的数据日志。例如,数据库108可以由分割模块122在10个实体之间进行分割,并且仅10%的可用存储空间被分配给第8实体。进一步地,第8实体在各实例在非常短的时间跨度中存储大数目的数据日志。在这样的场景中,如果在存储大数目的数据日志的事件期间第8实体的10%存储空间被耗尽,则覆写模块128可以利用第3实体的未使用存储空间来存储这些数据日志。

[0071] 在另一实施方式中,覆写模块128可以基于与每个分区相关联的覆写因素,将一个实体的数据日志存储到另一实体的存储分区上。基于这些覆写因素,各实体可以定义针对什么类型的数据日志,向其他实体的存储空间的分区上的覆写可以由覆写模块128执行。换句话说,一个实体可以定义为存储仅用于由它自己生成的某个类型的数据日志的数据,将被存储到另一实体的分区上。例如,在上面所提到的10个实体的场景中,第8实体可以定义:仅错误数据日志可以被存储到另一实体的分区上,并且在存储空间耗尽的情形中拒绝其他的数据日志。类似地,另一实体可以定义:仅具有高关键性点的数据日志可以被存储到其他实体的分区上,可以由覆写模块128携带。在所述实施方式中,每个实体还可以独立地定义它们的分区的将由覆写模块128利用以用于存储其他实体的数据日志的目的的最大大小。

[0072] 在一个实施方式中,基于这些覆写因素,一些实体还可以定义为,不将它们的分区共享用于存储来自其他实体的数据日志,而不论可用的存储空间。因此,将会理解,这些实体还可以通过这些覆写因素的方式,来定义关于数据日志的覆写的对数据日志的管理。在一个实施方式中,覆写因素可以包括如下的参数,诸如,可用的存储空间、数据日志的关键性、以及定义不同实体的要求的分区的共享能力。将会理解,不同的实体可以用不同的方式来定义不同的参数,以适合它们的要求和需要。

[0073] 又在本主题的另一实施方式中,系统102的分割模块122被配置为,动态地修改对已经被分割的数据库108的分割。这样的动态分割可以基于各实体的变化的要求,这些变化的要求可以取决于数据日志的数目和所生成的数据日志的大小。例如,在数据库108的总可用空间是30GB并且分割模块122已经将数据库108在3个应用之间分别以30%、20%和50%的份额进行了分割的情形中,任何一个实体(比如实体3)的要求可能突然增加,这是由于应用的存储失效以及错误数据日志的存储。在这样的情形中,分割模块122可以动态地将这些实体的份额分别修改为20%、10%和70%,以将更多的存储空间提供给第三实体。

[0074] 图2图示了根据本主题的一个实施例的用于多客户端架构中的数据日志管理的方法200。方法200被描述的顺序不意图被解释为一种限制,并且任何数目的所描述的方法框能够以任何顺序被组合以实施方法200,或者任何替换方法。另外,不偏离本文所描述的主题的精神和范围,可以从该方法中删除个别框。此外,该方法能够被实施在任何适合的硬件、软件、固件、或者它们的组合中。

[0075] 可以在计算机可执行指令的一般上下文中描述该方法。一般而言,计算机可执行

指令能够包括执行特定的功能或者实施特定的抽象数据类型的例程、程序、对象、组件、数据结构、进程、模块、函数,等等。该方法也可以被实践在分布式计算环境中,其中由通过通信网络而被链接的远程处理设备来执行各功能。在分布式计算环境中,计算机可执行指令可以被定位在本地和远程计算机存储介质(包括存储器存储设备)的两者中。

[0076] 本领域的技术人员将容易认识到,该方法的步骤能够由经编程的计算机来执行。在本文中,一些实施例也意图为覆盖程序存储设备,例如,数字数据存储介质,它们是机器或计算机可读的并且对指令的机器可执行或计算机可执行的程序进行编码,其中所述指令执行所描述的方法的步骤中的一些步骤或所有步骤。这些程序存储设备可以是,例如,数字存储器、诸如磁盘和磁带的磁存储介质、硬驱动器、或者光可读数字数据存储介质。各实施例还意图为覆盖被配置为执行示例性方法的所述步骤的通信网络和通信设备两者。

[0077] 参考图2,在框202处,用于数据日志的存储空间被分割为多个分区以在多个实体之间进行分配,其中该多个实体之中的每个实体被分配至少一个分区。在一个实施方式中,可以利用数据日志管理系统(诸如系统102)来分割存储空间。在所述实施方式中,系统102的分割模块122可以将可用于存储数据日志的存储空间分割为多个分区。在另一实施方式中,存储空间的分割可以基于多个实体的要求。

[0078] 在框204处,从该多个实体之中识别与每个实体相关联的清除参数,其中这些清除参数至少包括清除的频率,作为清除数据日志以释放存储空间的准则。这些清除参数可以定义准则,对分区中的数据日志的清除可以基于这些准则而发生。例如,一个实体可以将清除的频率定义为一天三次,而另一实体可以将该频率定义为一天一次。类似地,这些清除参数还可以包括:可以由各实体定义的清除数据日志的时间、与这些实体相关联的将首先被清除的数据日志的关键性点、以及在其期间所存储的数据日志应当被清除或者不应当被清除的时间实例。

[0079] 在框206处,基于与来自该多个实体之中的每个实体相关联的清除参数,来清除该多个实体的分区中存储的数据日志。在一个实施方式中,被清除的数据日志可以包括存储空间的分区中存储的经分片的数据日志以及未分片的数据日志。对数据日志的清除可以基于各实体的个别要求。例如,来自该多个实体之中的一个实体可以要求在清除错误数据日志之前清除平常的数据日志。类似地,相比于每12小时一次的其他实体的数据日志的平常清除,另一实体可以要求在12小时中清除数据日志两次。

[0080] 尽管已经关于具体实施例描述了本主题,但是这一描述并非有意于在限制性的意义上进行解释。对本领域的技术人员而言,一经参考本主题的描述,所公开的实施例的各种修改以及本主题的替换实施例将变得明显。因此被考虑到的是,不偏离如所定义的本主题的精神或范围,这样的修改能够被做出。

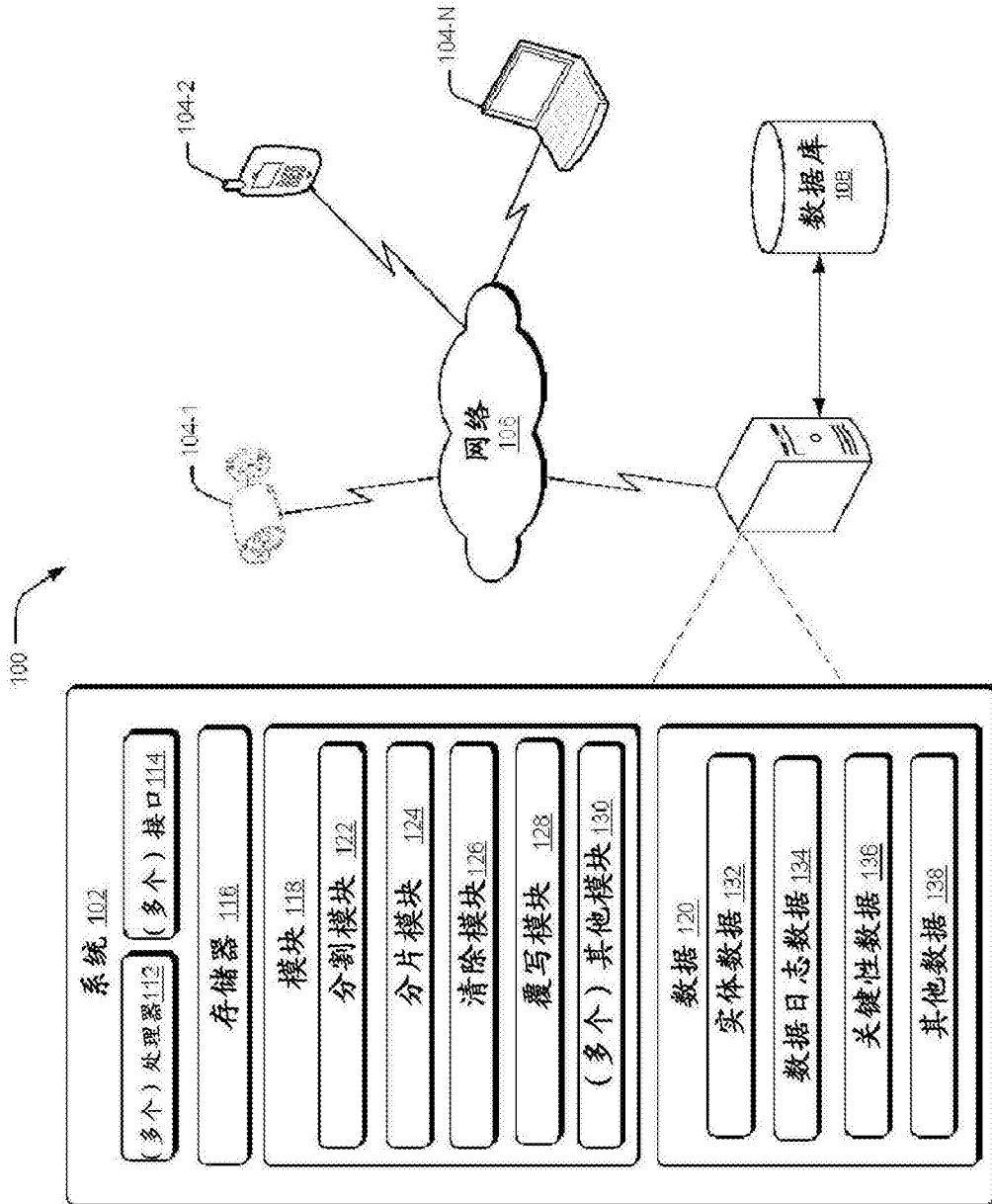


图1

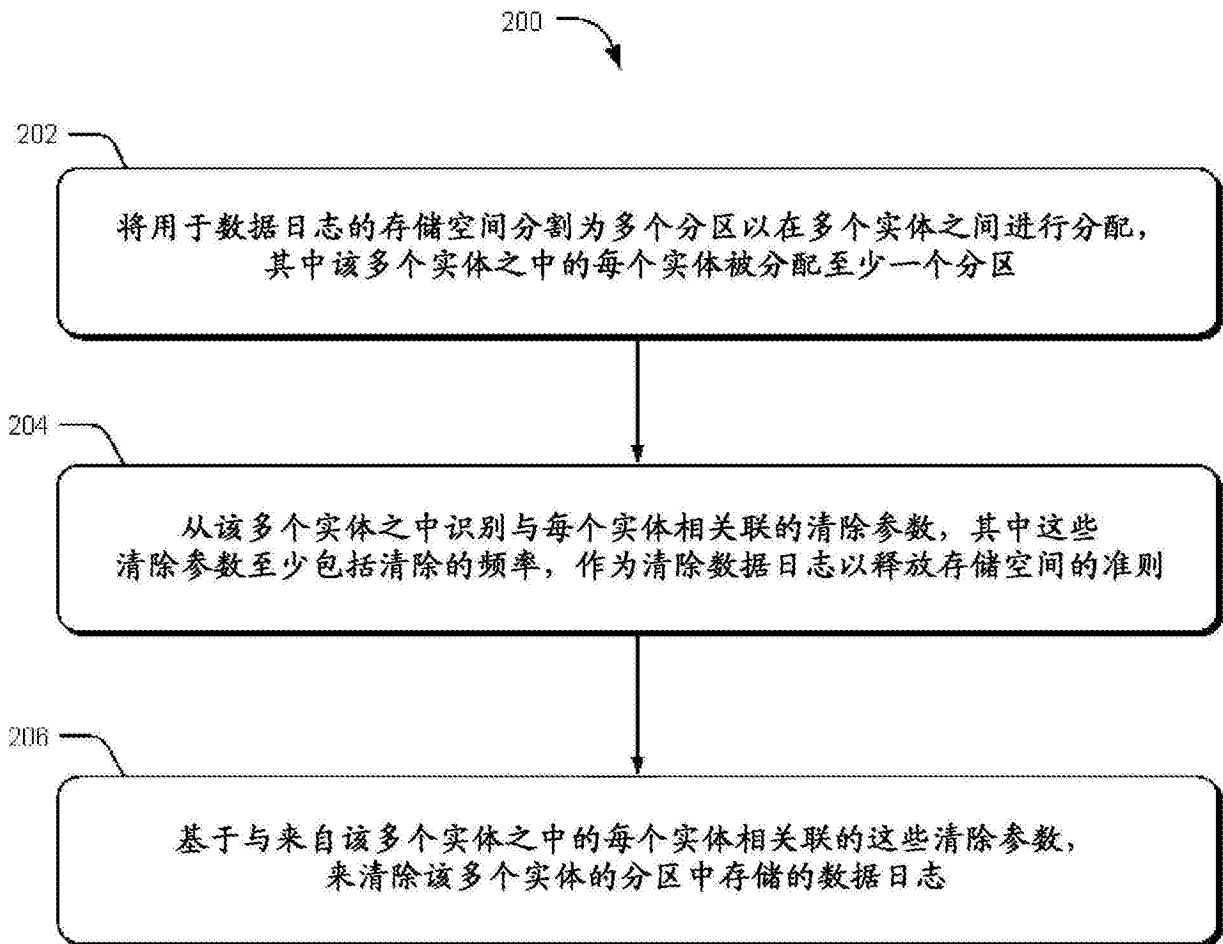


图2