



(12) 发明专利申请

(10) 申请公布号 CN 105247492 A

(43) 申请公布日 2016. 01. 13

(21) 申请号 201480020276. 4

代理人 刘瑜 王英

(22) 申请日 2014. 03. 31

(51) Int. Cl.

(30) 优先权数据

G06F 11/34(2006. 01)

61/807, 665 2013. 04. 02 US

14/063, 982 2013. 10. 25 US

(85) PCT国际申请进入国家阶段日

2015. 10. 08

(86) PCT国际申请的申请数据

PCT/US2014/032424 2014. 03. 31

(87) PCT国际申请的公布数据

W02014/165456 EN 2014. 10. 09

(71) 申请人 西部数据技术公司

地址 美国加利福尼亚

(72) 发明人 A·艾哈迈迪-阿尔达卡尼

(74) 专利代理机构 永新专利商标代理有限公司

72002

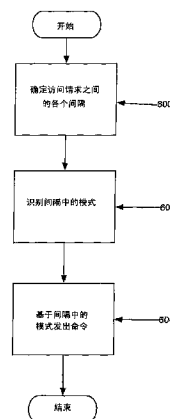
权利要求书3页 说明书8页 附图5页

(54) 发明名称

对使用时间序列建模的用户行为的检测

(57) 摘要

实施例提供了一种预测存储设备何时将被访问的方法。为了增强性能,所述存储设备可以针对所述访问操作主动地作出准备,并因此使得所述存储设备的访问时间响应最小化。随着时间记录用户行为并收集在数据集中。在一个实施例中,在所述数据集中的所述数据点之间的间隔被计算并布置在矩阵中。所述矩阵中的模式被识别并且被用于识别所述用户的下一次可能的访问。预期到下一次预测访问,所述存储设备然后可以采取诸如驱动器旋转启动之类的各种动作,以最小化访问时间响应。



1. 一种预测对存储设备的未来访问的方法,所述方法包括:
确定对所述存储设备的每个访问请求到任何后续访问请求之间的各个间隔;
识别在所述各个间隔中的模式;
基于在所述各个间隔中的所述模式来预测下一访问请求;以及
基于所预测的下一访问请求,主动激活所述存储设备。
2. 根据权利要求1所述的方法,其中,主动激活所述存储设备包括基于硬盘的旋转启动时间,在所预测的下一访问请求之前主动地旋转启动所述硬盘驱动器。
3. 根据权利要求1所述的方法,其中,主动激活所述存储设备包括基于所述各个间隔和容差,在所预测的下一访问请求之前主动激活所述存储设备。
4. 根据权利要求1所述的方法,进一步包括,确定所述各个间隔之间的误差容差。
5. 根据权利要求4所述的方法,进一步包括,将在所述误差容差内发生的多个访问请求聚合成组。
6. 根据权利要求1所述的方法,进一步包括将所述各个间隔布置在矩阵中。
7. 根据权利要求6所述的方法,其中,识别在所述各个间隔中的所述模式:确定在所述矩阵一列中出现的第一组间隔对应于在所述矩阵的一行中出现的第二组间隔。
8. 一种基于预测用户行为来管理电子设备的方法,所述方法包括:
记录在一段时间内的对所述电子设备的用户访问请求的序列;
确定所述用户访问请求的序列之间的各个间隔;
基于对所述用户访问请求的序列进行建模来识别用户访问请求的模式,所述建模基于对所述用户访问请求的间隔的序列进行匹配;
基于所识别的模式,确定下一用户访问请求的预测的时间;以及
基于所述下一用户访问请求的所述预测的时间,在所述电子设备上开始动作。
9. 根据权利要求8所述的方法,其中,识别所述用户访问请求的模式包括:
将所述各个间隔布置在矩阵中,其中,所述矩阵的每列指示从所有其他过去的用户访问请求起的间隔,并且其中,每行指示至后续用户访问请求的间隔;以及
基于识别出现在从所述矩阵中的所选元素起的一行和一系列两者中的间隔的序列,识别所述用户访问请求的模式。
10. 根据权利要求9所述的方法,其中,识别所述用户访问请求的模式包括:识别在容差内的、出现在从所述矩阵中的所选元素起的一行和一系列两者中的间隔的序列。
11. 根据权利要求8所述的方法,其中,在所述电子设备上开始所述动作包括开始硬盘驱动器的旋转启动。
12. 根据权利要求8所述的方法,其中,在所述电子设备上开始所述动作包括开始硬盘驱动器的旋转停止。
13. 根据权利要求8所述的方法,其中,在所述电子设备上开始所述动作包括将存储驱动器从待机状态转变为活动状态。
14. 根据权利要求8所述的方法,其中,在所述电子设备上开始所述动作包括将存储驱动器从活动状态转变为待机状态。
15. 一种被配置为基于对用户访问请求的时间序列进行建模来预期用户行为的存储设备,所述存储设备包括:

至少一个存储介质；

控制器，包括：

事件记录器，其被配置为记录针对来自所述至少一个存储介质的数据的用户访问请求的时间序列；

功率管理器，其被配置为基于活动状态和待机状态来管理所述至少一个存储介质的功耗；

时间序列建模器，其被配置为基于用户访问请求之间的所述时间序列中的间隔来检测用户行为的模式；以及

预测器，其被配置为基于所述用户行为的模式来预测下一用户访问请求，并且预期到所预测的下一用户访问请求而向所述功率管理器发送信号以将所述存储设备从第一功率状态转变为第二功率状态。

16. 根据权利要求 15 所述的存储设备，进一步包括存储器，其被配置为存储指示用户访问请求之间的间隔的矩阵。

17. 根据权利要求 16 所述的存储设备，其中，所述时间序列建模器被配置为基于对出现在所述矩阵的一行和一系列两者中的间隔的序列进行匹配来检测用户行为的模式。

18. 根据权利要求 15 所述的存储设备，其中，所述预测器被配置为预期到所预测的下一用户访问请求而发送信号以将所述至少一个存储介质从待机状态转变为活动状态。

19. 根据权利要求 15 所述的存储设备，其中，所述预测器被配置为预期到所预测的下一用户访问请求而发送信号以将所述至少一个存储介质从活动状态转变为待机状态。

20. 一种被配置为预测用户行为的设备，所述设备包括：

存储设备，用于存储受到用户访问请求的信息；以及

处理器，其由可执行程序代码所配置以记录在一段时间内的所述设备的用户访问请求的序列，确定所述用户访问请求的序列之间的各个间隔，基于对所述用户访问请求的序列进行建模来识别用户访问请求的模式，基于所识别的模式来确定下一用户访问请求的预测的时间，以及基于所述下一用户访问请求的所述预测的时间来在所述设备上开始动作，其中，所述建模基于对所述用户访问请求的间隔的序列进行匹配。

21. 根据权利要求 20 所述的设备，其中，所述处理器被配置为将所述各个间隔布置于存储在耦合至所述处理器的存储器中的矩阵中，其中，所述矩阵的每列指示从所有其他过去的用户访问请求起的间隔，并且其中，每行指示至后续用户访问请求的间隔；以及

其中，所述处理器被配置为基于识别出现在从所述矩阵中的所选元素起的一行和一系列两者中的间隔的序列来识别所述用户访问请求的模式。

22. 根据权利要求 20 所述的设备，其中，所述存储设备包括硬盘驱动器，并且所述处理器被配置为开始硬盘驱动器的旋转启动。

23. 根据权利要求 20 所述的设备，其中，所述存储设备包括硬盘驱动器，并且所述处理器被配置为开始硬盘驱动器的旋转停止。

24. 根据权利要求 20 所述的设备，其中，所述存储设备包括硬盘驱动器，并且所述处理器被配置为将所述硬盘驱动器从待机状态转变为活动状态。

25. 根据权利要求 20 所述的设备，其中，所述存储设备包括硬盘驱动器，并且所述处理器被配置为将所述硬盘驱动器从活动状态转变为待机状态。

26. 根据权利要求 20 所述的设备,其中,所述处理器被配置为响应于用户访问的模式而开始缓存内容。

27. 根据权利要求 20 所述的设备,其中,所述处理器被配置为响应于用户访问的模式而连接至网络。

对使用时间序列建模的用户行为的检测

背景技术

[0001] 计算设备的响应能力是其性能的重要特性。理想地,计算设备能够尽快地对用户作出响应。

[0002] 例如,存储设备试图尽可能高效和快速地提供对其存储的数据的访问。目前,固态硬盘驱动器(SSD)已逐渐流行,这归因于它们相对于硬盘驱动器(HDD)来说接近于零的数据访问时间延迟(DATD)性能。然而,相对于SSD来说,HDD仍然在较低成本的情况下提供较高的容量存储。因此,HDD由于它们的高容量和耐久性而保持流行。实际上,很多已知的产品,诸如固态混合驱动器(SSHD)目前提供SSD和HDD的组合作为一种存储器解决方案。

[0003] HDD一般具有高DATD,因为它们频繁地被置于待机模式以便节省电力等。例如,移动设备,诸如笔记本电脑、无线网络附加存储(NAS)或任意其他移动设备,频繁地使用侵略性的HDD待机策略。HDD在头部不接触磁盘的各种运动期间还可出于数据安全的原因而被置于待机模式。

[0004] 遗憾的是,当响应于数据访问请求而退出待机模式时,HDD可花费长时间以使得驱动器重新旋转启动。此外,随着硬盘驱动容量的增加,当跳出待机模式时DATD也可能提高。

[0005] 因此,期望提供可最小化HDD的DATD的方法和系统。还期望提供可提高任何计算设备的响应能力的方法和系统,特别是在空闲时间段后。

附图说明

[0006] 现在将参考附图描述实现本发明的各种特征的系统和方法,其中:

[0007] 图1示出了其中可实现本发明的实施例的示例性系统;

[0008] 图1A示出了根据本发明的实施例的被配置为预测用户数据请求的示例性控制器;

[0009] 图2示出了示例性用户行为模式;

[0010] 图3示出了通过本发明的一些实施例的基于时间序列建模的示例性预测访问事件;

[0011] 图4示出了由一些实施例所使用的用于预测各种数据访问事件的示例性时间序列建模;

[0012] 图5示出了由一些实施例所预测的数据访问事件和实际用户数据请求的示例图;以及

[0013] 图6示出了示例性过程流。

具体实施方式

[0014] 概述 – 预测用户行为和数据访问请求

[0015] 本发明的实施例基于预测数据访问请求何时可能发生来改进访问已存储的数据的数据访问时间延迟。在一些实施例中,存储设备观测用户行为并使用时间序列建模来预测或预期用户何时可能访问存储系统。一些实施例在包括仅一个或多个硬盘驱动器或与其

他类型存储设备组合的存储设备上实施。实施例可采用软件、固件等来实施用于预期何时可能发生数据访问请求的预测算法。

[0016] 为了说明的目的,现在将参考包括一个或多个硬盘驱动器的网络附加存储来描述一个示例。那些本领域技术人员将认识到实施例可以在多种多样的存储器类型上实施,诸如直接附加存储,存储区域网络等。

[0017] 现在将描述某些实施例。这些实施例仅以示例的方式呈现,并不是想要限制本发明的范围。实际上,本文描述的新颖方法和系统可以以多种其他形式实施。此外,可以在本文所描述的方法和系统的形式上作出各种省略、替换和改变,而不脱离本发明精神。为了说明实施例中的一些,现在将参考附图。

[0018] 图 1 示出了示例性系统。图 1A 示出了根据本发明的实施例的被配置为预测用户数据请求的示例性控制器。图 2 示出了示例性用户行为模式,其可通过实施例进行分析和预测。图 3 示出了通过本发明一些实施例的基于时间序列建模的示例性预测访问事件。图 4 示出了由一些实施例所使用的用于预测各种数据访问事件的示例性时间序列建模。图 5 示出了由一些实施例所预测的数据访问事件和实际用户数据请求的示例图。图 6 示出了示例性过程流。现在将在下文进一步描述图 1-5。

[0019] 示例性系统 - 图 1

[0020] 图 1 示出了本发明的实施例的示例性系统。本发明的实施例可在任意类型的存储设备或提供文件通知服务的设备上实施。为了说明的目的,示出了示例性系统 100,其具有经由网络 106 为客户端(或主机设备)104 服务的网络附加存储(NAS)102。该系统 100 还可用在其中期望预测能力的其他设备中,诸如移动设备、媒体流/回放设备、笔记本电脑、平板电脑、桌上型计算机、智能电视或其他类型的电子设备。现在将在下文简略描述具有这些部件的示例性实施例。

[0021] 示例性存储设备 - NAS 102

[0022] NAS 102 是文件级存储设备,其能够连接至网络(诸如网络 106),并作为文件服务器提供对在其存储介质中存储的文件的访问。在一个实施例中,利用已知的硬件、软件和固件实现 NAS 102。例如,在一个实施例中,NAS 102 被配置为具有嵌入式操作系统的装置。NAS 102 可以支持多种操作系统,诸如 UNIX、Linux、Windows 等。如将进一步描述的,NAS 102 还可以包括多种存储介质,诸如布置在独立磁盘冗余阵列(RAID)中的一个或多个硬盘。此外,NAS 102 可以支持多种协议,诸如 NFS、SMB/CIFS、AFP 等。

[0023] 如图所示,NAS 102 可以包括网络接口 110、控制器 112、存储接口 114 以及磁盘存储阵列 116。将在下文简略描述这些部件。

[0024] 网络接口 110 充当 NAS 102 至/从网络 106 的网络通信接口。例如,在一个实施例中,网络接口 110 可以包括一个或多个千兆以太网、以太网、通用串行总线(USB)或其他用于与网络 106 通信的接口。这样的部件对于那些本领域技术人员来说是公知的。

[0025] 控制器 112

[0026] 控制器 112 表示管理 NAS 102 的磁盘 116 并将它们向客户端 104 呈现为一个逻辑单元或卷的硬件和软件。在一些实施例中,控制器 112 还可以包括一个或多个其他部件以补充其操作,诸如片上 RAID 控制器、存储器或磁盘高速缓存等。此外,在本实施例中,控制器 112 被配置为观察与来自客户端 104 的数据请求相关的用户行为,检测在该行为中的模

式,并预测各种未来数据请求。控制器 112 然后可以经由存储接口 114 管理或操纵各个磁盘 116 的操作和功率状态。在一些实施例中,控制器 112 采用时间序列建模以预测未来数据请求并相应地管理磁盘 116 以改进它们的数据访问时间和 / 或最小化它们的 DATD。下文进一步描述时间序列建模算法和实施方式的示例。

[0027] 存储接口 114

[0028] 存储接口 114 充当控制器 112 和磁盘 206 之间的接口。存储接口 114 可支持多种通信,诸如 SAS、SATA、SCSI 等。

[0029] 磁盘 116

[0030] 磁盘 116 表示 NAS 102 的用于设备存储数据的存储介质和相关的电子器件。在一个实施例中,磁盘 116 可以被实施为硬盘驱动器,诸如由西部数据技术公司提供的那些硬盘。当然,在其他实施例中,NAS 102 可以包括其他类型的存储介质和设备,诸如固态驱动器、混合驱动器等。能够被配置为 RAID 的一部分的任意类型的存储驱动器可以被实施为本发明的实施例的一部分。

[0031] 可替代地,如上文所述,在其他实施例中,系统 100 可以包括直接附加存储设备或外部驱动器。例如,系统 100 可以包括一个或多个外部驱动器,诸如西部数据技术公司提供的 MyBook 或 Passport 设备。

[0032] 客户端 104

[0033] 客户端 104 表示可在 NAS 102 上存储和 / 或访问文件的各种客户端设备。例如,客户端 104 可以是桌上型电脑、笔记本电脑、平板电脑、智能电话等。实施例支持能够访问存储在 NAS 102 上的文件的任意设备。

[0034] 网络 106

[0035] 网络 106 为系统 100 的部件间的数据通信提供通信基础设施。网络 106 可以包括已知的网络元件,诸如集线器、交换机、路由器、防火墙等,以使这些通信便利和安全。在实施例中,网络 106 可以包括局域网、广域网等。此外,网络 106 可以包括有线和无线链路或部件以承载其通信。

[0036] 实施时间序列建模的示例性控制器 - 图 1A

[0037] 图 1 进一步,图 1A 示出了被配置为基于时间序列建模预测未来数据请求的控制器 112 的更详细的框图。在一些实施例中,控制器 112 可以实现时间序列建模器,其包括四个主要部件:(1) 事件记录器 118;(2) 检测算法 120;(3) 预测器算法 122;(4) 功率管理器 124。现在将更进一步地描述这些部件。

[0038] 事件记录器 118

[0039] 事件记录器 118 记录与从客户端 104 接收的数据请求相关的信息。事件记录器 118 可以记录数据请求的多个方面,诸如请求的日期 / 时间、文件类型、文件大小等。事件记录器 118 可以以多种多样的方式在控制器 112 中实施。例如,事件记录器 118 可以是控制器 112 内的固件,其被配置为执行记录数据请求的功能。作为另一示例,事件记录器 118 可以是在控制器 112 中运行的软件,其随着时间持续记录设备的用户行为。在一些实施例中,事件记录器 118 记录来自客户端 104 的数据请求的部分或全部,并且每个数据点(或者挑选出的数据点)本质上充当训练点并且收集的数据点持续地增加整体预测的精度。

[0040] 检测算法 120

[0041] 控制器 112 执行检测算法 120 以识别由事件记录器 118 收集的先前数据内的各种模式。在一些实施例中,检测算法 120 确定各种数据请求的到达之间的间隔以产生一组间隔的时间序列。检测算法 120 然后可以基于这些间隔的时间序列中存在的模式来识别或检测用户行为。

[0042] 预测器算法 122

[0043] 控制器 112 还可执行预测器算法 122。一旦显现出一种模式,预测器算法 122 提供用于预测用户的未来行为的逻辑,以及何时期望来自客户端 104 的一个或多个未来数据请求。预测器算法 122 可以以多种格式输出其预测结果,诸如时间偏移、置信因子等。

[0044] 功率管理器 124

[0045] 最后,控制器 112 可以包括功率管理器 124。功率管理器 124 管理诸如存储接口 114 和磁盘 116 之类的 NAS 102 的功率状态和功耗。在一些实施例中,控制器 112 被配置为使用功率管理器 124 基于预测器算法 122 的预测结果来预期用户数据请求使用。例如,响应于预测器算法 122 预测的未来数据请求,控制器 112 可以用信号通知功率管理器 124 转变磁盘 116 的功率状态以在下一数据请求的预测时间开始旋转硬盘 116。此外,控制器 112 可以当距离预测的下数据请求存在长的闲置时段或时间时用信号通知功率管理器 124 以使磁盘 116 和 / 或存储接口 114 掉电或减少其功耗。针对控制器 112 和功率管理器 124 何时改变功率状态和操作的多种阈值可以例如基于旋转启动时间 (spin up time)、通信延迟、读取 / 写入延迟等而改变。此外,控制器 112 和功率管理器 124 可以基于多种标准修改其功率管理方案,诸如电池荷电状态、数据请求频率等。

[0046] 时间序列建模的示例

[0047] 为了说明本发明的理念,现在将参考图 2-5 在下文进一步描述检测和预测算法。为了说明的目的,参考图 2 示出了用户行为和数据请求序列的简单示例。

[0048] 图 2- 示例性用户行为 (简化的)

[0049] 如图 2 所示,该图表可以表示位于客户端 104 处的用户自从他已经初始地打开 NAS 102 开始以 100 小时的间隔已经访问设备上的数据的行为。在该示例中,控制器 112 因此可以根据这样的用户行为来进行简单的预测。

[0050] 图 3- 对用户行为的预测

[0051] 例如,图 3 示出了由控制器 112 确定的预测的数据请求事件。如图所示,各个“用户数据请求”点 (用于指示来自客户端 104 的用户请求) 被标记为 R_n 而由控制器 112 预测的“未来事件预测”点被标记为 P_n , 其中 n 为 1、2、3 等。如能够看出的那样, R_2 和 R_1 之间的时间间隔可以表示为 Δ_1 而 R_3 和 R_2 之间的时间间隔可以表示为 Δ_2 。

[0052] 在一个实施例中,如果发现 Δ_1 和 Δ_2 是相等的或处于特定容差内,预测器算法 122 可以预测一模式并因此预测未来数据请求。也就是说,下一用户请求可以被预测为远离 $R_3 \Delta_1$ 或 Δ_2 (即,大约 100 小时)。

[0053] 解释波动和 / 或变化

[0054] 在一些实施例中,为了对用户请求行为中的一些波动作出解释,预测器算法 122 可以包括误差缓存。例如,该算法可以被公式化并简化为如下:

[0055]
$$\text{if } |\Delta_{n+1} - \Delta_n| \leq \epsilon_r \Rightarrow \Delta'_{n+2} = \min(\Delta_n, \Delta_{n+1}) - \epsilon_r - t_s, \text{ 其中, } \min$$
 是最小化函数,

[0056] ϵ_r 是用于解释用户请求中的波动的缓冲, 以及

[0057] t_s 是 DATD, 诸如用于驱动器旋转启动以及设备准备好响应用户请求所花费的时间, 以及最后

[0058] Δ'_{n+2} 是未来预测间隔 (在图 3 中示出为 R_3 和 P_1 之间的间隔)。

[0059] 在一个实施例中, 为了简化计算, 检测算法 120 被配置为以矩阵形式表达变化量 Δ , 如下面所建议的:

[0060]

$$\begin{bmatrix} \Delta_{1,1} & \cdots & \Delta_{1,j} \\ \vdots & \ddots & \vdots \\ \Delta_{i,1} & \cdots & \Delta_{i,j} \end{bmatrix}$$

[0061] 其中, $\Delta_{i,j}$ 指的是 R_i 和 R_j 之间的时间间隔。

[0062] 如能够看出的那样, 该矩阵是其主对角线上的所有元素都等于 0 的方阵。另外, 该矩阵的下三角部分包括与该矩阵的上三角部分相同的数据, 以使得下三角部分可以被忽略 (诸如节省存储器) 并基于对上述矩阵的这些简化得到下面的矩阵:

[0063]

$$\begin{bmatrix} 0 & \Delta_{1,2} & \cdots & \Delta_{1,j} \\ 0 & 0 & \cdots & \Delta_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

[0064] 因此, 任意行 i 包括涉及数据请求点 R_i 至所有其他未来数据请求点的所有变化量 Δ 。同样任意列 j 包括涉及从所有其他过去的的数据请求点至 R_j 的间隔的所有变化量 Δ 。为了使得该示例可视化, 图 4 示出了说明在该矩阵中表达的间隔的图表。

[0065] 时间序列建模 - 另一示例

[0066] 在一些实施例中, 控制器 112 被配置为基于预测器算法 122 执行时间序列建模以识别该矩阵内的模式。现在将参考下述假设的用户模式来提供检测算法 120 和时间序列建模的一个示例。

[0067]

用户数据请求点	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
时间单位 (小时)	5	7	10	19	21	24	33	35	38	48	58	68

[0068] 根据这一数据集, 控制器 112 可构建下述简化的矩阵如下:

[0069]

0	2	5	14	16	19	28	30	33	43	53	63
0	0	3	12	14	17	26	28	31	41	51	61
0	0	0	9	11	14	23	25	28	38	48	58

0	0	0	0	2	5	14	16	19	29	39	49
0	0	0	0	0	3	12	14	17	27	37	47
0	0	0	0	0	0	9	11	14	24	34	44
0	0	0	0	0	0	0	2	5	15	25	35
0	0	0	0	0	0	0	0	3	13	23	33
0	0	0	0	0	0	0	0	0	10	20	30
0	0	0	0	0	0	0	0	0	0	10	20
0	0	0	0	0	0	0	0	0	0	0	10

[0070] 在一个实施例中,可以基于下述伪代码实现检测算法 120 以识别用户访问请求中的模式。

[0071] clc

[0072] G = [] ;

[0073] l = [5 7 10 19 21 24 33 35 38 48 58 68] ; %间隔向量始于 i = 1

[0074] l_size = size(l') ;

[0075] for i = 1:l_size ;

[0076] for j = 1:l_size

[0077] Delta(i, j) = abs(l(i)-l(j)) ;

[0078] end

[0079] end

[0080] Delta = triu(Delta)

[0081] for i = 1:l_size

[0082] col_v = Delta(:, i) ;

[0083] row_v = Delta(i, :) ;

[0084] t_col = col_v(col_v ~= 0) ;

[0085] t_row = row_v(row_v ~= 0) ;

[0086] RESULT = ismember(col_v, t_row) ;

[0087] [row_i, col_i] = find(RESULT == 1) ;

[0088] if isempty(Delta(row_i, i))

[0089] %显示“这是空的”

[0090] % Delta(row_i, i)

[0091] else

[0092] detected_delta = Delta(row_i, i)

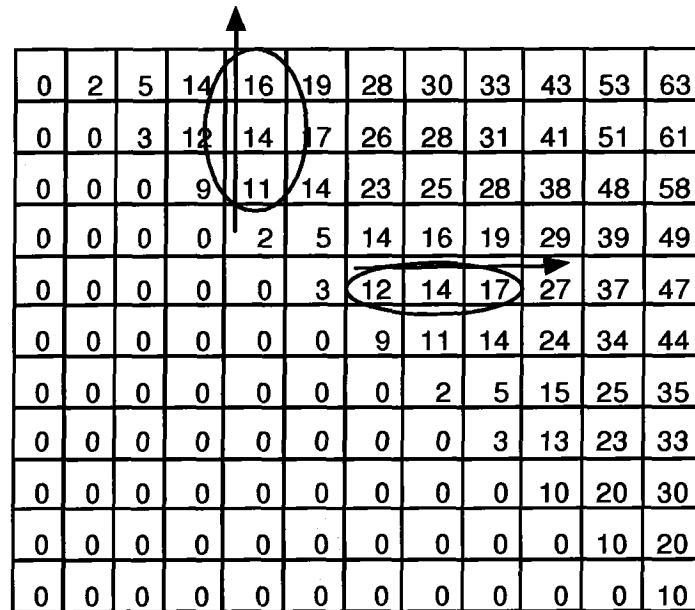
[0093] G(end+1) = l(i)+detected_delta ;

[0094] end

```
[0095] end
[0096] figure
[0097] plot(1,1,'bx',G,1,'ro');
[0098] legend(' User Data Request', 'Future Event Prediction');
```

[0099] 上述代码还执行对用户行为的预测并产生如图 5 所示的下述附图。如能够看到的那样,用户已经利用以不同的模式到达数据请求与 NAS 102 交互。下文的示例性矩阵示出了已经通过检测算法 120 对从矩阵中的所选元素起的一行和一列两者共同的间隔的序列进行匹配来识别的模式。

[0100]



0	2	5	14	16	19	28	30	33	43	53	63
0	0	3	12	14	17	26	28	31	41	51	61
0	0	0	9	11	14	23	25	28	38	48	58
0	0	0	0	2	5	14	16	19	29	39	49
0	0	0	0	0	3	12	14	17	27	37	47
0	0	0	0	0	0	9	11	14	24	34	44
0	0	0	0	0	0	0	2	5	15	25	35
0	0	0	0	0	0	0	0	3	13	23	33
0	0	0	0	0	0	0	0	0	10	20	30
0	0	0	0	0	0	0	0	0	0	10	20
0	0	0	0	0	0	0	0	0	0	0	10

[0101] 在一些实施例中,上述系统及其部件的一些或全部由一个或多个物理或虚拟计算系统执行或实施。例如,在一些实施例中,具有包括中央处理器 (CPU)、输入 / 输出 (I/O) 部件、存储设备和 / 或存储器的部件的计算系统 (诸如服务器、移动设备或其他计算设备) 可以用于执行上述系统的部件的一些或全部。I/O 部件可以包括显示器、到网络的网络连接、计算机可读介质驱动器和 / 或其他 I/O 设备 (例如,键盘、鼠标、触摸屏、扬声器等)。

[0102] 图 6 示出了用于预测存储设备 (诸如存储设备 102) 的未来访问的示例处理流程。在阶段 600, 存储设备 102 中的事件记录器 118 记录接收到对存储在磁盘 116 上的文件的数据请求。控制器 112 然后确定存储设备 102 的每个访问请求到任意后续访问请求之间的各个间隔。例如,控制器 112 可产生数据结构,诸如数组或矩阵,其中,每个元素指示间隔 (或 Δ)。在一些实施例中,该矩阵可进一步被简化或减小,因为该矩阵是方阵,并因此下三角半区可以被忽略。正如所提到的,该矩阵可以由控制器 112 存储在存储器 (未示出) 中。在上文示出了这样一个矩阵的示例。

[0103] 在阶段 602, 控制器 112 识别在各个间隔中的模式。例如,如上文所提到的,控制器 112 可以执行程序代码或与实现预测算法 (诸如上述预测器算法 122) 的硬件或固件部件进行合作。

[0104] 在阶段 604, 控制器 112 基于在各个间隔中的模式预测下一访问请求。特别地,控制器 112 基于预测器算法的输出确定下一数据请求的时间。预测的下一数据请求可以被提

供为绝对时间、时间偏移、范围等。预测的下一数据请求还可以具有相关联的元数据,诸如置信因子、数据请求的类型等。

[0105] 在阶段 604,控制器 112 可以基于下一预期的数据请求发出一个或多个命令。例如,控制器 112 可以基于预测的下一访问请求激活或停止主动激活的存储设备。例如,控制器 112 可以将磁盘 116 从一个功率状态转变为另一功率状态,诸如从活动转变为待机状态。控制器 112 还可以激活或停止其他部件,诸如网络接口 110、存储接口 114 等,例如,以用于节省电池消耗或最大化性能。此外,控制器 112 可以基于下一预测的数据请求来命令高速缓存或预取存储在磁盘 116 上的各种文件。

[0106] 在一个实施例中,控制器 112 可以被配置为采用时间序列建模来优化存储设备 102 的电池寿命。例如,控制器 112 可以包括对存储设备 102 的功率管理特别是磁盘 116 的功耗进行控制的功率管理器 124。在一些实施例中,功率管理器 124 可以转变磁盘 116 的功率状态以当请求了长闲置时间时或当预测到下一数据请求时开始对磁盘 116 的旋转停止或旋转启动 (spin down or up)。

[0107] 此外,控制器 112 还可以对各种容差作出解释,诸如磁盘 116 的旋转启动延迟等。容差可以基于磁盘 116 的已知特性或基于由检测算法 120 提供的元数据。

[0108] 在前面部分中描述的过程、方法和算法中的每个可以被实施在由一个或多个计算机、计算机处理器或被配置为执行计算机指令的机器所执行的代码模块中并全部或部分地由代码模块自动执行。所述代码模块可以被存储在任意类型的非暂时性计算机可读存储介质或有形计算机存储设备上,诸如硬盘驱动器、固态存储器、光盘等。该过程和算法可以被部分或全部实施为专用电路。公开的过程和过程步骤的结果可以永久地或以其他方式存储在任意类型的非暂时性计算机存储设备中,例如,易失性或非易失性存储设备。

[0109] 上文所公开的特定实施例的特征和属性可以以不同方式进行组合以形成其他的实施例,其全部落入本公开的范围。尽管本公开提供了特定实施例和应用,对本领域技术人员显而易见的其他实施例,包括并不提供本文所阐述的所有特征和优点的实施例,同样在本公开的范围。因此,旨在仅参考所附权利要求来限定本公开的范围。

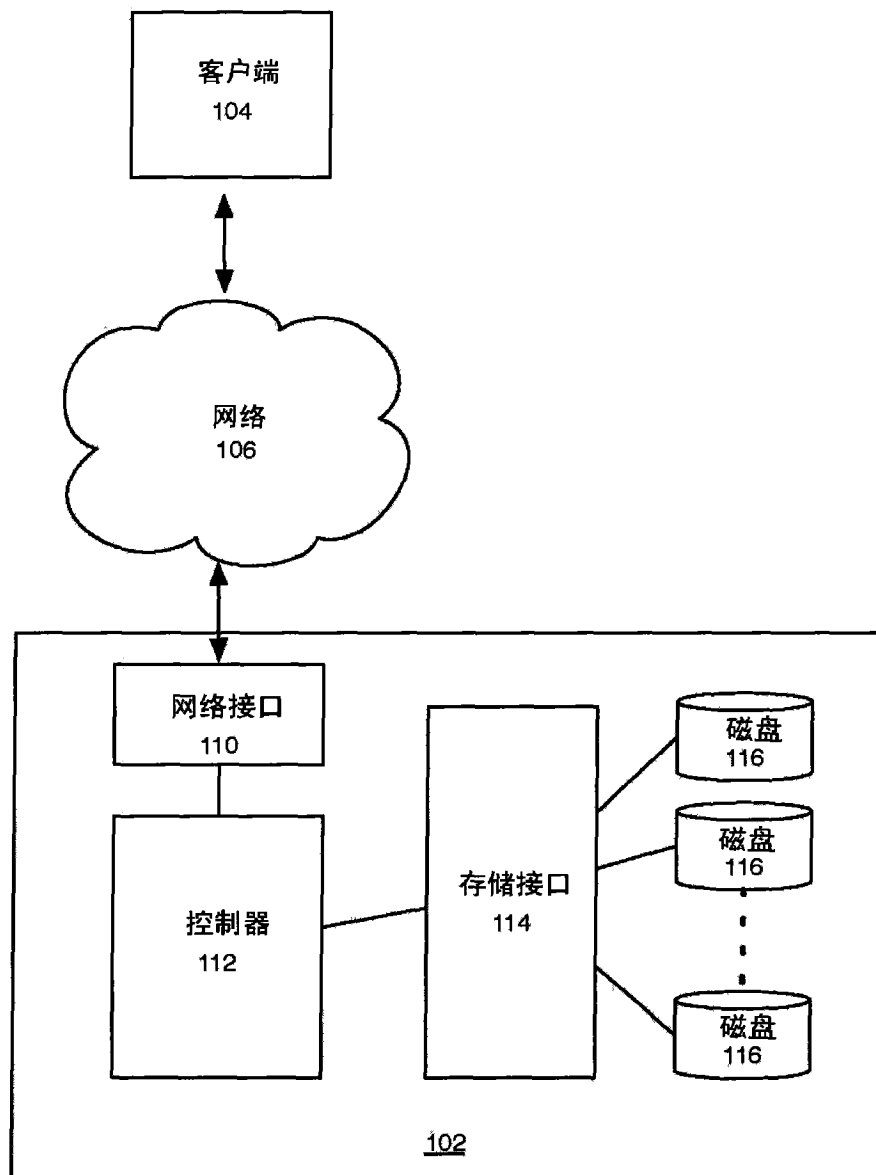


图 1

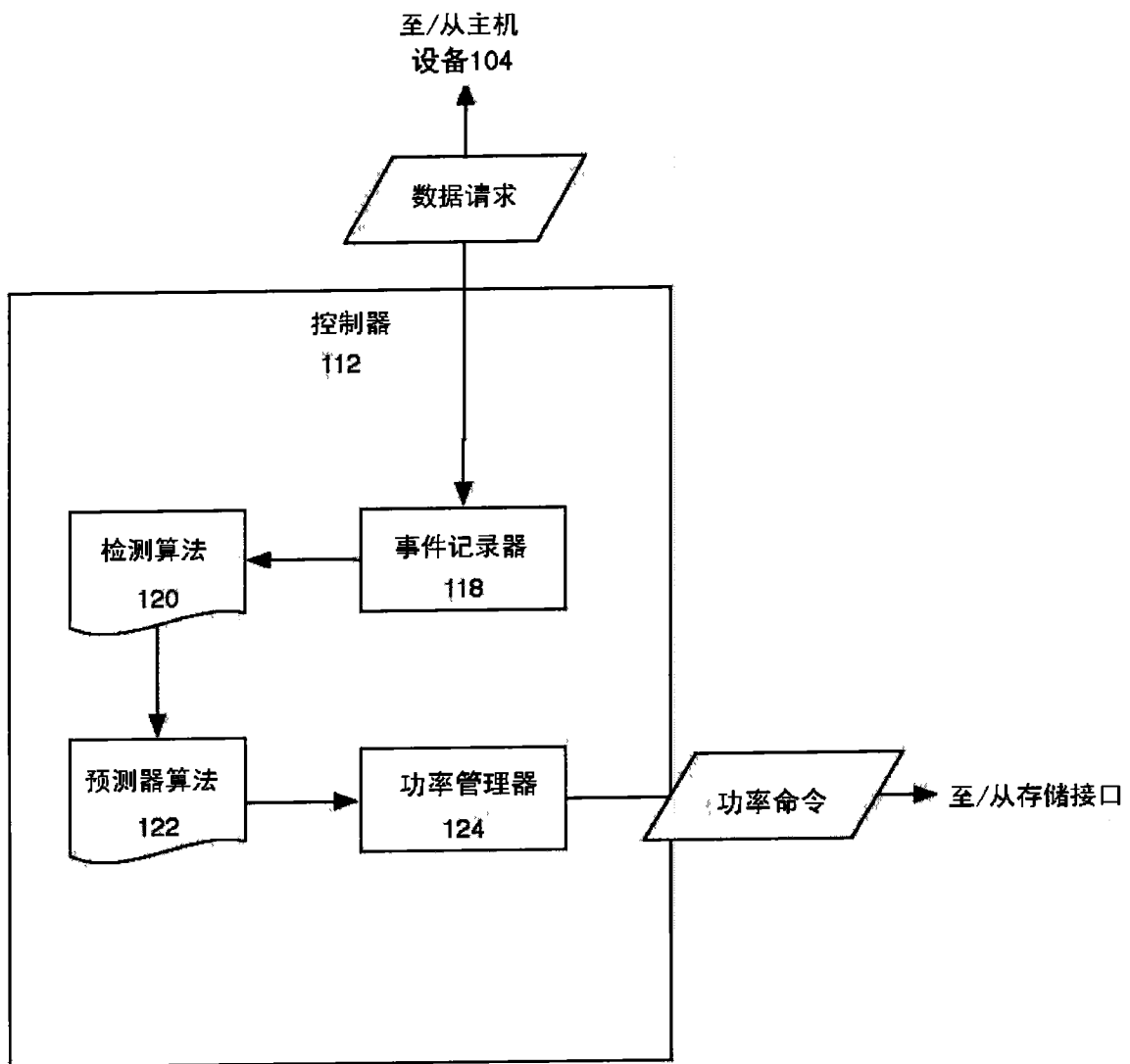


图 1A

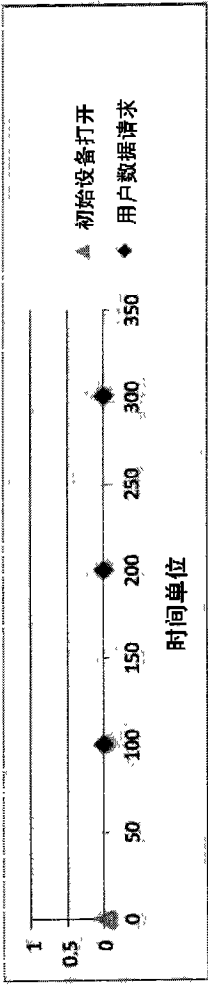


图 2

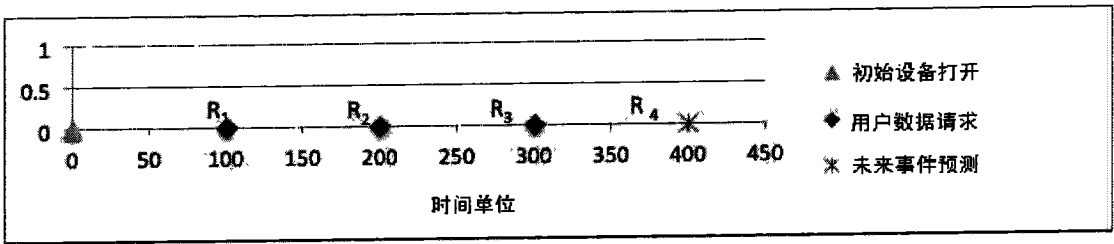


图 3

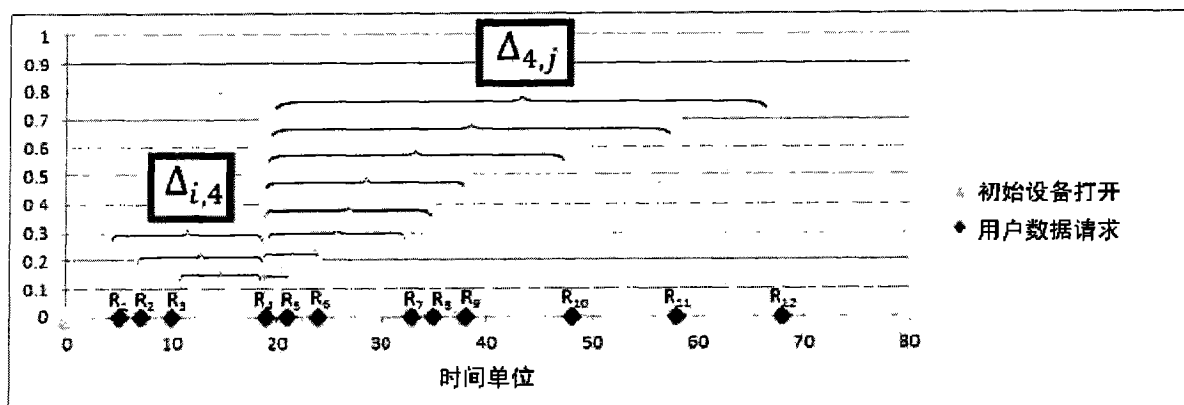


图 4

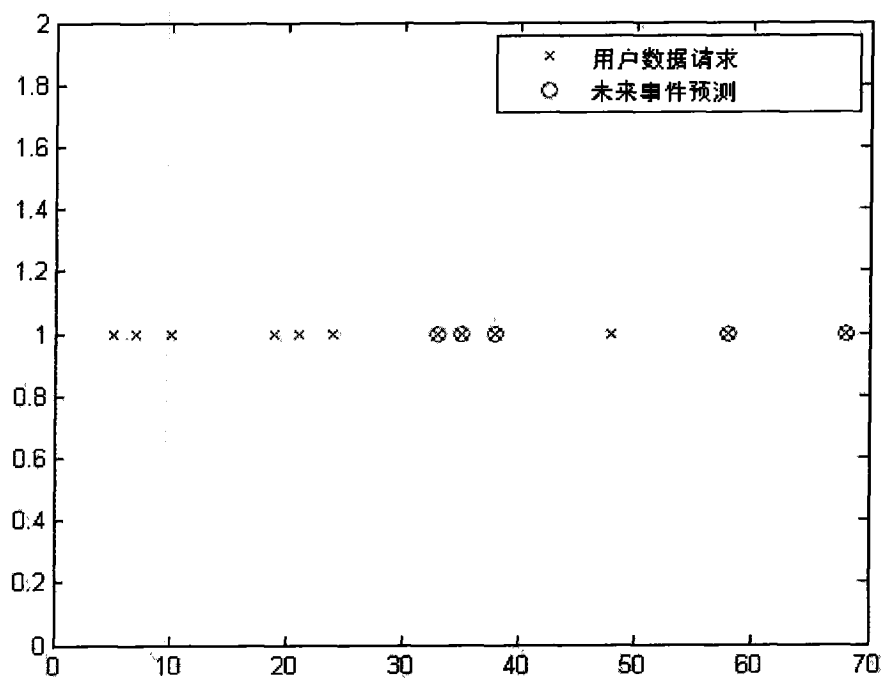


图 5

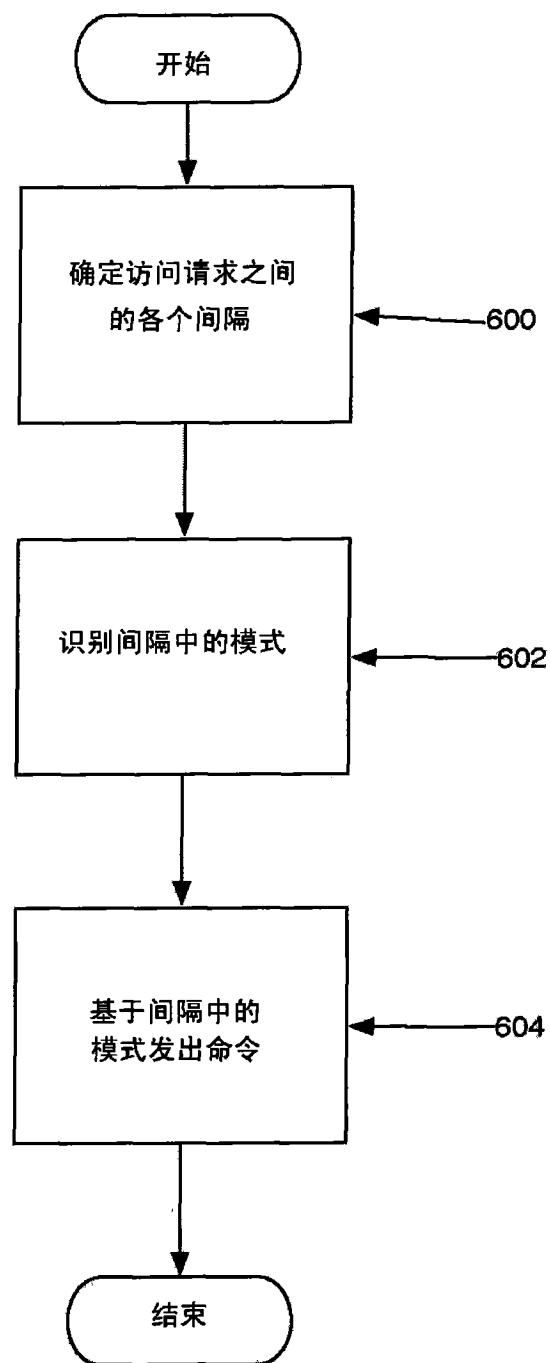


图 6