



(12)发明专利

(10)授权公告号 CN 103460194 B

(45)授权公告日 2016.12.21

(21)申请号 201180069278.9

(72)发明人 吕鹏

(22)申请日 2011.03.15

(74)专利代理机构 中国专利代理(香港)有限公司 72001

(65)同一申请的已公布的文献号
申请公布号 CN 103460194 A

代理人 柯广华 朱海煜

(43)申请公布日 2013.12.18

(51)Int.Cl.

(85)PCT国际申请进入国家阶段日
2013.09.13

G06F 11/30(2006.01)

(86)PCT国际申请的申请数据
PCT/CN2011/000407 2011.03.15

审查员 邹小彬

(87)PCT国际申请的公布数据
W02012/122672 EN 2012.09.20

(73)专利权人 爱立信(中国)通信有限公司
地址 100000 北京市朝阳区利泽东街5号爱立信大厦

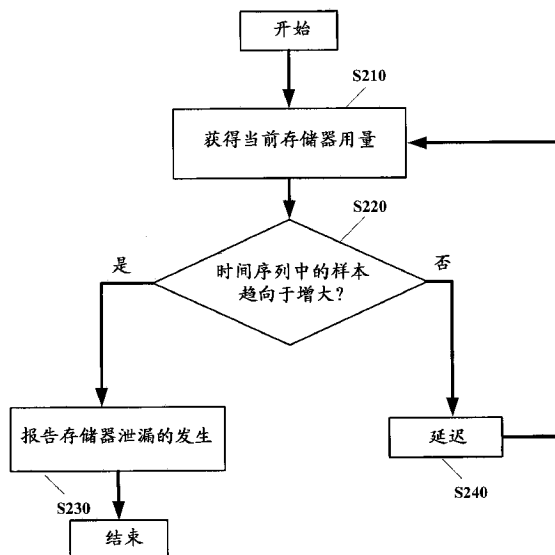
权利要求书1页 说明书6页 附图5页

(54)发明名称

对资源泄漏的检测

(57)摘要

公开一种用于检测计算系统中的资源泄漏的方法。该方法包括下列步骤:获得(S210、S320)表示时间序列中的样本之一的与计算系统中的资源有关的用量;确定(S220、S330)样本是否趋向于增大;以及如果确定样本趋向于增大,则报告(S230、S340)计算系统中的资源泄漏的发生。还公开一种用于检测计算系统中的资源泄漏的设备(600)。



1. 一种用于检测计算系统中的资源泄漏的方法,包括下列步骤:
 - 获得(S210,S320)表示时间序列中的样本之一的与所述计算系统中的资源有关的用量;
 - 确定(S220,S330)所述样本是否趋向于增大;以及
 - 如果确定所述样本趋向于增大,则报告(S230,S340)所述计算系统中的所述资源泄漏的发生,其中确定所述样本是否趋向于增大的所述步骤包括下列步骤:
 - 在获得新用量后,从用于存储用量的数组中去除不小于所述新用量的所有用量;
 - 将所述新用量添加到所述数组中;
 - 确定所述数组中元素的数量是否等于预定参数;以及
 - 如果所述元素的数量等于所述预定参数,则确定所述样本趋向于增大,或者其中确定所述样本是否趋向于增大的所述步骤包括下列步骤:
 - 计算所述样本的移动平均;以及
 - 如果所述移动平均高于预定阈值,则确定所述样本趋向于增大。
2. 如权利要求1所述的方法,其中,通过周期性地测量所述用量来获得所述样本。
3. 如权利要求1所述的方法,其中,所述资源是可用于整个计算系统、应用或进程的存储器或者文件描述符。
4. 一种用于检测计算系统中的资源泄漏的设备(600),包括:
 - 第一单元(610),用于获得表示时间序列中的样本之一的与所述计算系统中的资源有关的用量;
 - 第二单元(620),用于确定由所述第一单元(610)获得的所述样本是否趋向于增大;以及
 - 第三单元(630),用于如果所述第二单元(620)确定所述样本趋向于增大,则报告所述计算系统中的所述资源泄漏的发生,其中所述第二单元配置成:
 - 在获得新用量后,从用于存储用量的数组中去除不小于所述新用量的所有用量;
 - 将所述新用量添加到所述数组中;
 - 确定所述数组中元素的数量是否等于预定参数;以及
 - 如果所述元素的数量等于所述预定参数,则确定所述样本趋向于增大,或者其中所述第二单元配置成:
 - 计算所述样本的移动平均;以及
 - 如果所述移动平均高于预定阈值,则确定所述样本趋向于增大。
5. 如权利要求4所述的设备(600),其中,所述第一单元(610)周期性地测量所述用量,以便获得所述样本。
6. 如权利要求4所述的设备(600),其中,所述资源是所述计算系统中可用的存储器或者文件描述符。

对资源泄漏的检测

技术领域

[0001] 一般来说,本发明涉及计算系统中的资源管理。更具体来说以及在各种实施例中,本发明涉及用于检测计算系统中的资源泄漏的方法、设备、计算机程序产品和计算机可读介质。

背景技术

[0002] 存储器泄漏(或者在这个上下文中,泄漏)在计算机程序消耗存储器、但是无法将其重新释放给操作系统时发生。它能够通过降低可用存储器的量来削弱计算机的性能。最终,在最坏情况下,过多的可用存储器可变得已被分配并且系统或装置的全部或部分停止工作,应用失败,或者系统因超负荷而不可接受地慢下来。

[0003] 但是,特别是在大型系统、例如电信软件中,难以具有对存储器泄漏的准确且及时的检测或预测,部分是因为当泄漏开始时通常不会有直接征兆。

[0004] 开发了多种方法和技术来解决存储器泄漏问题。例如,开发了一种方法,其中设置存储器用量的固定阈值,以及如果总存储器用量的峰值超过该固定阈值,则确定存储器处于存储器泄漏状态。图1示出用于检测计算机中的存储器泄漏的示意过程。如图1所示,在步骤S110,测量计算机的当前存储器用量。然后,在步骤S120,如果确定所测量用量大于预定阈值TH1,则进行到步骤S130,其中向操作系统报告泄漏的发生;否则,该过程继续进行步骤S140,其中经过预定间隔,然后该过程返回到步骤S120。

[0005] 但是,该方法未能考虑如下事实:存储器用量通常与系统的服务负荷成比例,并且因而在一些情况下,高于阈值的存储器用量的主要峰值并非暗示存储器泄漏,而只是高负荷。一旦负荷下降,则存储器用量将返回其正常等级。此外,突然泄漏可引起存储器在短时期耗尽,而对于这种方法,如果阈值TH1不适当地设置为高值,则在存储器容量完全不可用以前,可能没有充分时间预先发出警告。

[0006] 另一种方式是在开发阶段,采用诸如从IBM(Armonk, New York, U.S.A.)可获得的Rational Purify[®]之类的调试程序来执行用于查找潜在泄漏的一系列测试。但是,对于大型且复杂的系统,不可能在测试中调查全部代码分支,并且因而只能降低泄漏的可能性。另外,调试是耗费时间的过程,并且将显著延长开发过程。

发明内容

[0007] 目的是消除上述缺点中的至少一些,并且提供用于检测资源泄漏的改进解决方案。

[0008] 本发明的方面包括一种用于检测计算系统中的资源泄漏的方法。该方法可包括下列步骤:获得表示时间序列中的样本之一的与计算系统中的资源有关的用量;确定样本是否趋向于增大;以及如果确定样本趋向于增大,则报告计算系统中的资源泄漏的发生。

[0009] 在按照本发明的一个优选实施例中,在确定步骤中,如果存在时间序列中的升序样本集合,则确定样本趋向于增大。更优选地,在确定步骤中,升序样本集合被表征如下:

$$(S_1, \dots, S_i, S_{i+1}, \dots, S_k), S_{i+1} > S_i, 1 \leq i \leq k-1$$

[0010]

$$S_i = m_x, S_{i+1} = m_y, x < y$$

[0011] 其中, k 是升序样本集合中的元素的数量, S_i 是升序样本集合中的第 i 元素, m_x 和 m_y 是时间序列中的第 x 和第 y 样本。

[0012] 在按照本发明的另一个优选实施例中, 通过周期性地测量用量来获得样本。

[0013] 在按照本发明的另一个优选实施例中, 资源是可用于整个计算系统、应用或进程的存储器或者文件描述符。

[0014] 本发明的另一方面包括用于检测计算系统中的资源泄漏的设备。该设备包括: 第一单元, 用于获得表示时间序列中的样本之一的与计算系统中的资源有关的用量; 第二单元, 用于确定由第一单元获得的样本是否趋向于增大; 以及第三单元, 用于如果第二单元确定样本趋向于增大, 则报告计算系统中的资源泄漏的发生。

[0015] 还提供一种在运行于计算机上时适合执行上述方法的计算机程序产品以及包含适合执行上述方法中的步骤的计算机可执行程序代码的计算机可读介质。

附图说明

[0016] 通过以下如附图中所示的优选实施例的更具体描述, 本发明的上述及其它目的、特征和优点将变得清楚, 附图中参考标号表示各个视图中的相同部分。

[0017] 图1示出用于检测计算机中的存储器泄漏的示意过程。

[0018] 图2是示出按照本发明的一个示范实施例、用于存储器泄漏检测的方法步骤的流程图。

[0019] 图3是示出按照本发明的一个示范实施例、用于存储器泄漏检测的方法步骤的流程图。

[0020] 图4是示出用于从时间序列来搜索升序样本集合的示范算法的流程图。

[0021] 图5示出通过图4的算法所获得的仿真图。

[0022] 图6是示出按照本发明的用于检测存储器泄漏的设备的一个实施例的框图。

具体实施方式

[0023] 虽然本发明涵盖各种修改和备选构造, 但是在附图中示出并且下面将详细描述本发明的实施例。但是, 应当理解, 具体描述和附图并不是要将本发明限制到所公开的具体形式。相反, 意图是, 要求保护的发明的范围包括落入所附权利要求所表达的本发明范围内的其全部修改和备选构造。

[0024] 除非本描述的上下文中另外限定, 否则, 本文所使用的所有科技术语具有与本发明所属领域的技术人员普遍理解的同样的含义。

[0025] 此外, 作为非限制性示例, 本发明的方法和设备通过在存储器泄漏情形中使用来示出。但是, 应当理解, 本发明能够应用于其数量或使用受到限制并且因差错或故障而部分或完全不能再使用的其它种类的资源。例如, 一种资源能够是指配给类似UNIX的操作系统中的各进程或任务的文件描述符, 并且如果没有适当释放, 则该进程将因而遭受主存储器

信息转储。其它种类的这类资源包括但不限于计算能力和通信信道。

[0026] 按照本发明的一个实施例,周期性地或随机地监测资源用量,监测它是否具有稳定地增大或上升的趋势,如果这种趋势存在,则确定资源泄漏发生。这种方式可排除资源用量的峰值只归因于操作期间高负荷的情况。

[0027] 注意,在本发明中,多种方法和设备能够用于有效且有力地“探测”增大的趋势或稳定增大,这将详细描述。

[0028] 如以下附图所示,方法和设备能够用于多种计算系统中,例如包括但不限于服务器、个人计算机、膝上型计算机、嵌入式计算机等。此外,方法能够采取软件、硬件、固件或者其组合的形式来实现。

[0029] 为了便于说明,假定以下实施例应用于包括处理器、存储器、I/O装置、操作系统和应用的计算系统。计算系统还包括用于检测存储器泄漏的设备。虽然结合具有上述结构的计算系统来描述按照本发明的以下实施例,但是并不是要将本发明限定于任何特定系统。

[0030] 图2是示出按照本发明的一个示范实施例、用于存储器泄漏检测的方法步骤的流程图。

[0031] 参照图2,在步骤210,该设备测量或获得计算系统的当前存储器用量。当前值连同先前测量的值一起构成时间序列。换言之,每个值能够被看作时间序列中的样本之一。注意,本文中的词语“测量”表示由该设备本身或者该设备外部的其它组件、例如由操作系统所提供并且由该设备所调用的基础功能性或例程来执行所述测量。

[0032] 然后,该过程进行到步骤S220,其中设备确定时序序列的样本之中是否存在稳定上升,即,样本是否趋向于增大。如果发现这种上升,则该过程进行到步骤S230;否则,该过程进行到步骤S240。

[0033] 在步骤S230,该设备生成指示存储器泄漏发生的消息,并且将其传送给操作系统,以便将采取适当动作以防止计算系统的性能降级。

[0034] 另一方面,在步骤240,该设备等待预定间隔,然后返回到步骤S210。通过这个间隔,周期性地测量或获得时间序列的样本。但是,固定间隔不是必需的,并且实际上,测量可随机地进行。例如,该设备可配置成响应于例如从操作系统接收到触发信号而发起泄漏检测过程。

[0035] 图3是示出按照本发明的一个示范实施例、用于存储器泄漏检测的方法步骤的流程图。

[0036] 参照图3,在步骤310,该设备“监听”来自操作系统的触发信号或命令。如果接收到该信号,则该过程进行到步骤S320;否则,该设备继续监听该信号或命令。

[0037] 在步骤320,该设备测量或获得计算系统的当前存储器用量,以便形成由当前值和历史值组成的时间序列。

[0038] 然后,该过程进行到步骤S330,其中该设备确定时间序列的样本是否趋向于增大。如果发现这种趋势,则该过程进行到步骤S340;否则,该过程进行到步骤S310。

[0039] 在步骤S340,该设备生成存储器泄漏发生的消息,并且将该消息传送给操作系统,操作系统进而采取适当动作。

[0040] 注意,在上述实施例中,存储器用量表示整个计算系统的存储器用量。但是,如图2和图3所示的实施例也能够应用于需要监测正运行的程序或任务的存储器用量的状况。这

些变化和修改处于本发明的精神和原理之内。

[0041] 现在说明如何确定时间序列的样本是否趋向于增大或上升,即,是否存在时间序列的样本之中的稳定增大或者增大的趋势。

[0042] 假定以间隔T来周期性地测量存储器用量,由此生成本文中表示为 $(m_1, m_2, m_3, \dots, m_n, \dots)$ 的时间序列,其中 m_n 表示在第n时间点测量的样本。按照本发明的一个实施例,所述趋势或稳定增大由以下的升序样本集合来表征:

$$(S_1, \dots, S_i, S_{i+1}, \dots, S_k), S_{i+1} > S_i, 1 \leq i \leq k-1 \quad (1)$$

[0043]

$$S_i = m_x, S_{i+1} = m_y, x < y \quad (2)$$

[0044] 其中,k是升序样本集合中的元素的数量, S_i 是升序样本集合中的第i元素, m_x 和 m_y 是时间序列中的第x和第y样本。

[0045] 换言之,如果从时间序列 $(m_1, m_2, m_3, \dots, m_n, \dots)$ 能够找出多个样本(例如,这个实施例中的k个元素)在时间上是上升的,则确定存在增大的趋势或稳定增大。直观地,在条件(1)和(2)下的升序样本集合表现为随时间上升的多个“波谷”,但是样本可波动。

[0046] 图4是示出用于从时间序列来搜索升序样本集合的示范算法的流程图。

[0047] 参照图4,在步骤S410,获得本文中表示为m的存储器用量的新值作为当前值。如上所述,能够周期性地或随机地测量该值。

[0048] 然后,在步骤S420,如果确定本文中表示为S的升序样本的数组为空,则该过程进行到步骤S430,其中将当前值m作为第一元素 S_1 记录在数组S中,并且进一步进行到步骤S410;否则,进行到步骤S440。

[0049] 在步骤S440,从数组S中去除不小于当前值m的任何元素。随后,在步骤S450,将当前值记录为数组S的尾部或最后一个元素,本文中表示为 S_j 。

[0050] 然后,该过程进行到步骤460以确定数组S中的元素的数量是否等于预定参数k,即,时间序列中是否存在升序样本集合。如果情况是这样,则该过程进行到步骤S470,其中报告稳定上升的事件;否则,返回步骤S410。

[0051] 下面是用于实现图4的算法的举例说明性的C伪代码段。

[0052]

```

1 static int S[k] ; /* define an array with k elements */
2
3 while(1) /* Loop to measure memory usage periodically */
4 {
5     get the current memory usage "m";
6
7     for(int j = 0; j < k; j++) {
8         if (S[j]==0) {
9             S[j]=m;
10            break;
11        } else if( m <= S[j] ) {
12            S[j]=m;
13            set S[j+1]..S[k-1] as zero;
14            break;
15        }
16    }
17    if ( j== k-1 ) {
18        /*
19         * The pattern is found. All elements in array S[] are non-zero.
20         * And S[i+1]>S[i]. It means continuous increase in memory usage,
21         * indicating there could be memory leak problem!
22         * Stop looping and exit the algorithm.
23         */
24        return 1;
25    }
26
27    delay( Delta_t ); /* delay delta_t interval to measure memory usage */
28 }

```

[0053] 图5示出通过图4的算法所获得的仿真图。如从图5看到的,在定时 t_1 、 t_2 、 t_3 、……、 t_{21} 、……(其间具有固定间隔 T)周期性地测量存储器用量。在定时 t_{18} ,该过程检测到升序样本集合:

[0054] $(S_1, \dots, S_j, S_{j+1}, \dots, S_k), S_{i+1} > S_i, 1 \leq i \leq k-1$

[0055] 在定时 t_{18} 之后,存储器用量突然增大,从而引起仿真测试下的系统的崩溃。如果该过程用作存储器泄漏的检测器,则能够避免崩溃。

[0056] 仍然参照图5,虽然存储器用量在定时 t_8 与 t_9 之间呈现迅速增大,但是它在后续定时 t_{10} 和 t_{11} 因降低的负荷而下降。通过上述算法,将会良好地识别并且因而忽略这个虚假信号。

[0057] 在该示例中,样本的数量 k 被设置为8。应当注意,这个参数连同间隔 ΔT 能够被调整以适合多种场合。例如,值得考虑的是,对于缓慢存储器泄漏采用具有较大值的 ΔT ,而对于快速存储器泄漏采用具有较小值的 ΔT 。对于样本的数量 k ,发现较大的 k 将提高检测精度和可靠性,但是可能需要更多时间。

[0058] 如上所述,时间序列中的增大的趋势能够通过搜索其中的升序样本集合来检测。在将详细描述的一个实施例中,使用基于移动平均值(MA)的算法。

[0059] 仍然假定以间隔 ΔT 来周期性地测量存储器用量,由此生成本文中表示为 $(m_1, m_2, m_3, \dots, m_n, \dots)$ 的时间序列。按照实施例,趋势或稳定增大是基于时间序列的移动平均值的改变来确定的。具体来说,如果移动平均值大于预定阈值,则确定时间序列具有稳定增大;否则,没有稳定增大发生。备选地,移动平均值可由其变化率来替代。

[0060] 在获得存储器用量的新测量结果、例如时间序列中的第 n 样本时,时间序列的移动

平均值MA能够更新如下：

$$[0061] \quad MA = \frac{\sum_{i=i-h+1}^i m_i}{h} \quad (3)$$

[0062] 其中， m_i 是时间序列中的第*i*样本，以及*h*是求平均的样本的数量。

[0063] 注意，样本的数量和阈值是可调整的，以便适合多种场合。例如，较低等级的数量和阈值将产生对存储器泄漏的快速响应，但是冒着错误检测的风险。另一方面，对于这些参数的较高等级，预期精度和可靠性，但是灵敏度降低。

[0064] 图6是示出按照本发明的用于检测存储器泄漏的设备的一个实施例的框图。

[0065] 图6所示的设备可采取如下形式来实现：运行于操作系统上或与之集成在一起的用于监测实际存储器使用状态的软件工具，例如专用集成电路(ASIC)、现场可编程门阵列(FPGA)之类的用于特定目的的电路，或者它们的组合。它能够实现为用于在开发阶段调试检测存储器泄漏的独立工具。

[0066] 参照图6，设备600包括第一单元610、第二单元620和第三单元630。在实施为软件工具的设备的条件下，第一至第三单元610-630能够采取分开的组件或模块的形式来实现。第一单元610负责通过调用操作系统提供的例程来获得或测量计算系统的存储器用量的多个样本。如上所述，样本构成时间序列。而且，能够周期性地或随机地测量存储器用量。

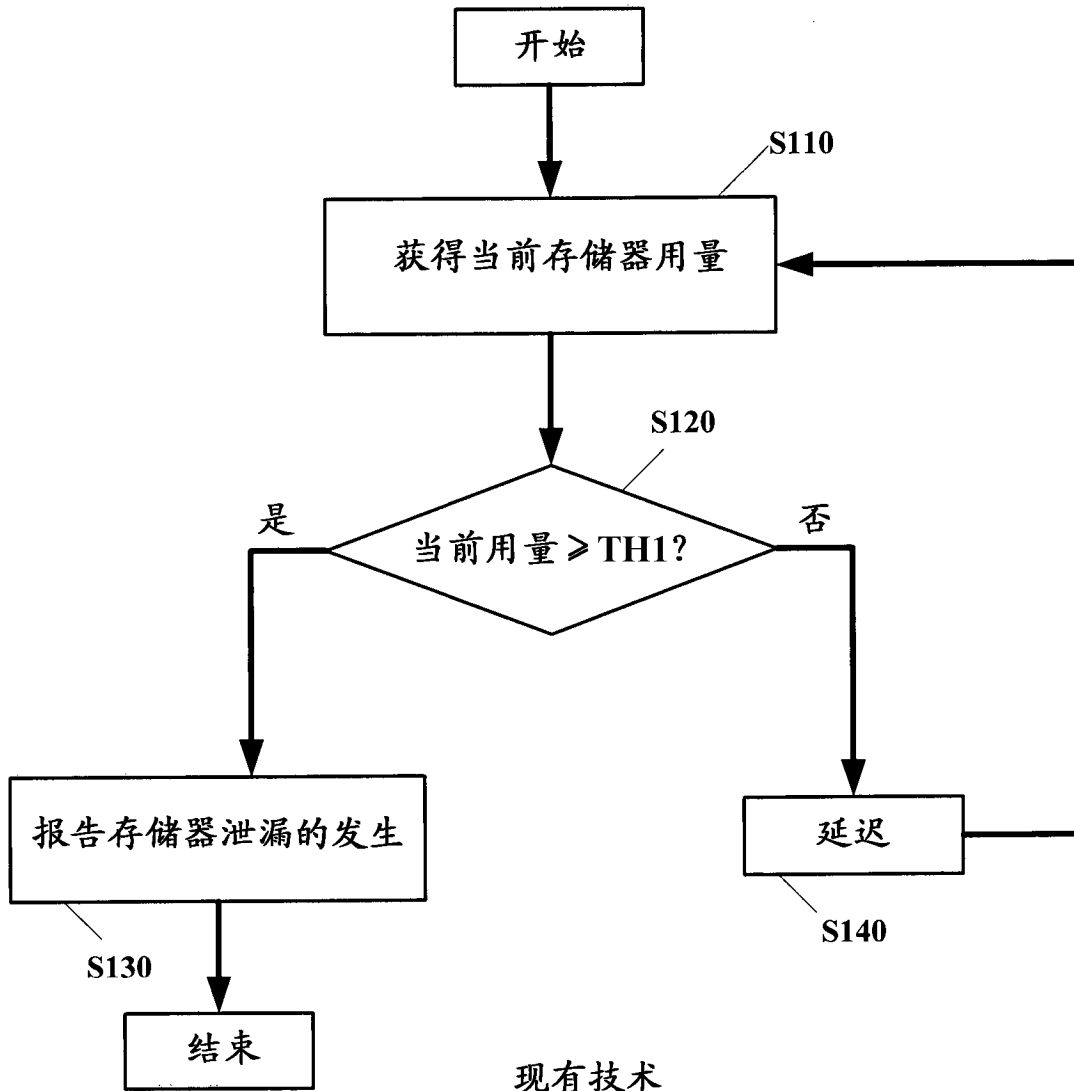
[0067] 第二单元620与第一单元610和第三单元630进行通信。它确定由第一单元610获得的样本是否具有增大的趋势。如果发现这种趋势，则第三单元630将向操作系统报告存储器泄漏的发生。在这个实施例中，第二单元620能够配置成通过使用如上所述的方法来执行确定。

[0068] 与用于检测存储器泄漏的常规技术相比，本文中结合图2、图3、图4、图5和图6所公开的方法和设备实现对存储器泄漏的即时、准确和可靠的检测，以便避免系统崩溃。具体来说，能够有效地过滤出动态服务负荷所引起的存储器用量峰值。

[0069] 按照本发明的一个实施例，提供一种在运行于计算机上时适合执行上述方法的计算机程序产品。

[0070] 按照本发明的另一个实施例，提供一种包含适合执行上述方法中任一个的步骤的计算机可执行程序代码的计算机可读介质。

[0071] 应当注意，上述实施例是说明本发明而不是限制本发明，可由本领域的技术人员设计替代实施例，而没有背离所附权利要求的范围。诸如“包括”、“包含”之类的用语并不排除存在、但在说明书和权利要求书中没有列出的元件或步骤。还要注意，如本文和所附权利要求书中使用的单数形式“一”、“一个”和“该”包括复数对象，除非上下文另加明确说明。本发明能够通过包括若干不同元件的硬件或者通过适当编程的计算机来实现。在列出若干部件的单元权利要求中，这些部件之中的若干部件能够具体实施在同一硬件项目中。诸如第一、第二、第三之类的词语的使用并不表示任何顺序，而能够只是解释为名称。



现有技术

图 1

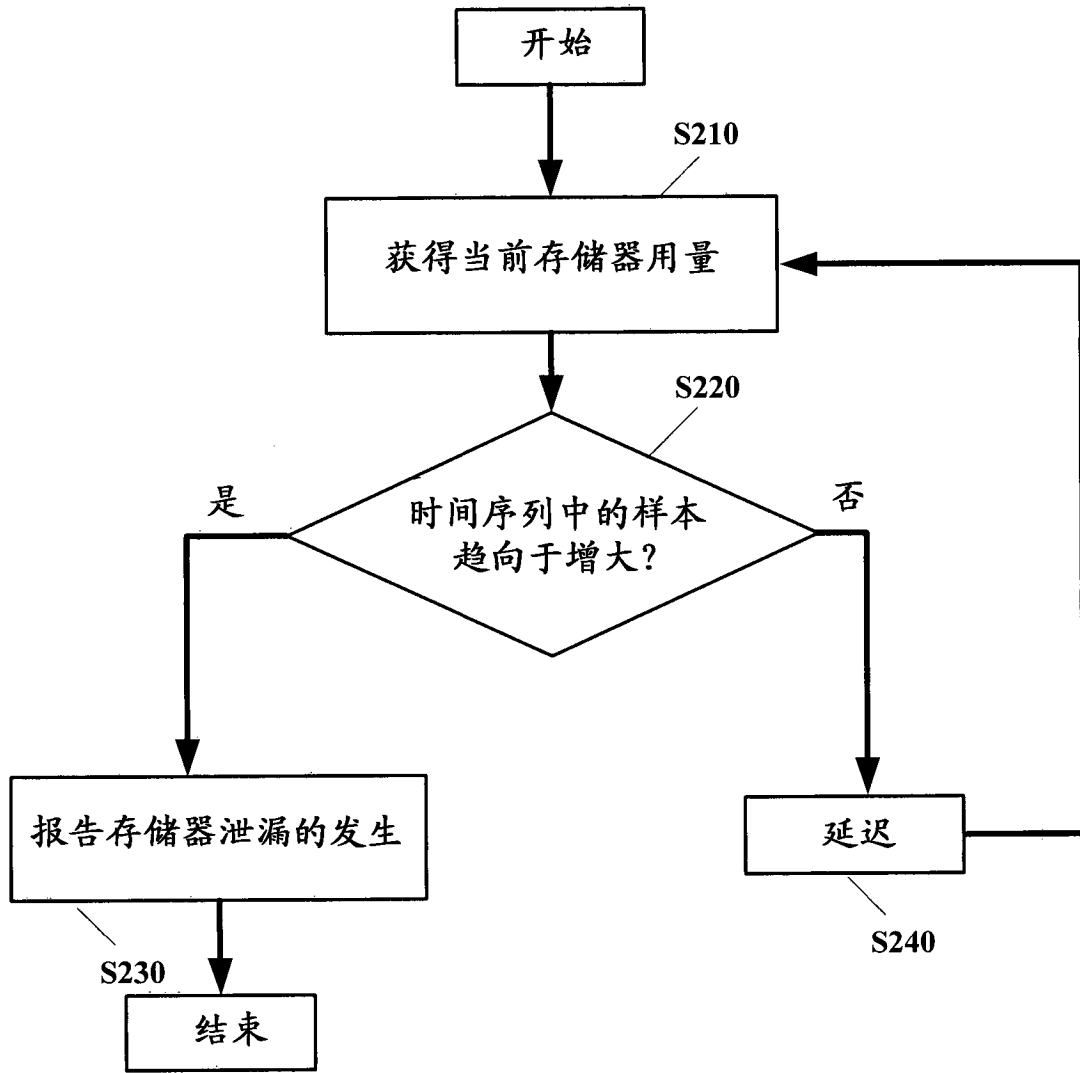


图 2

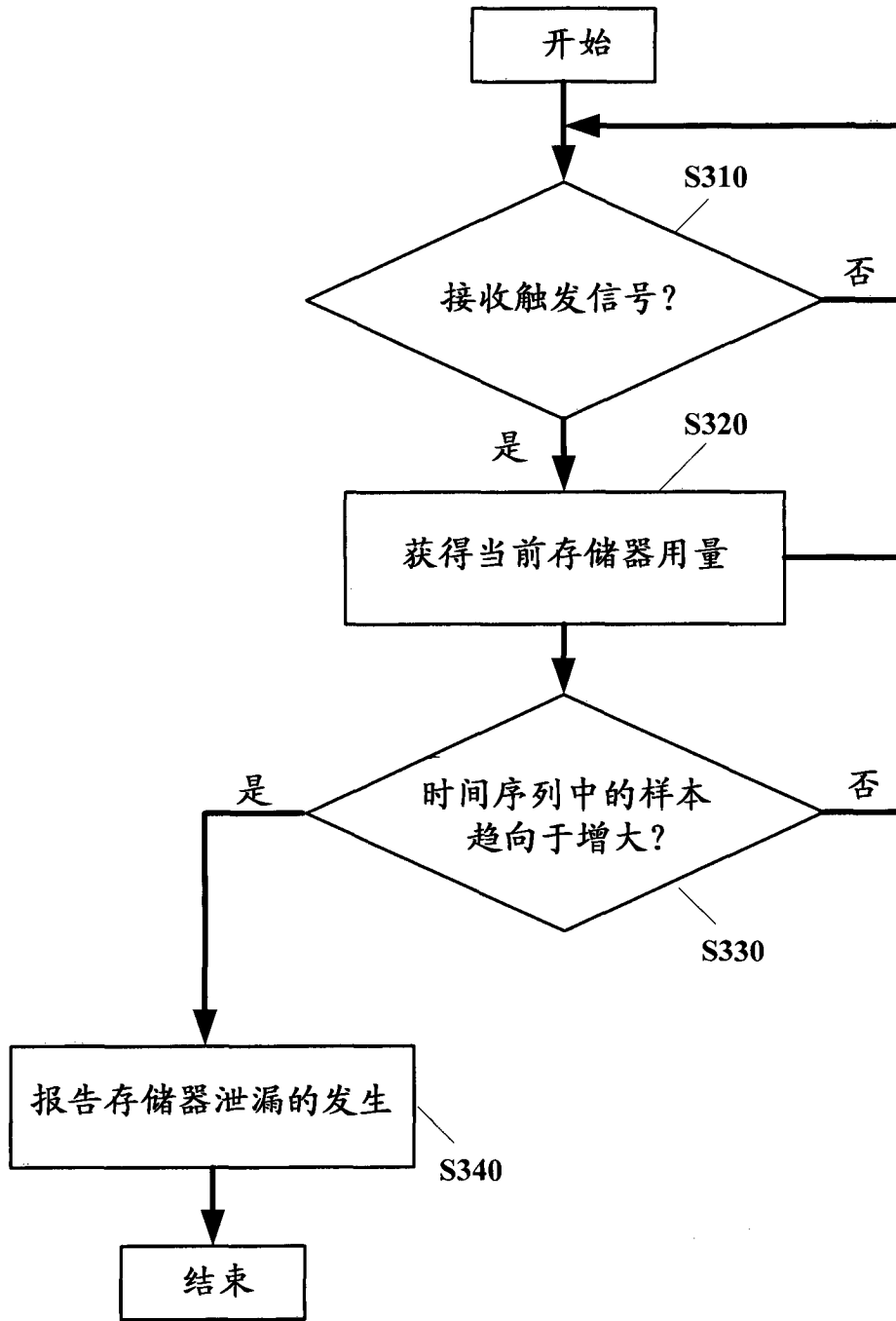


图 3

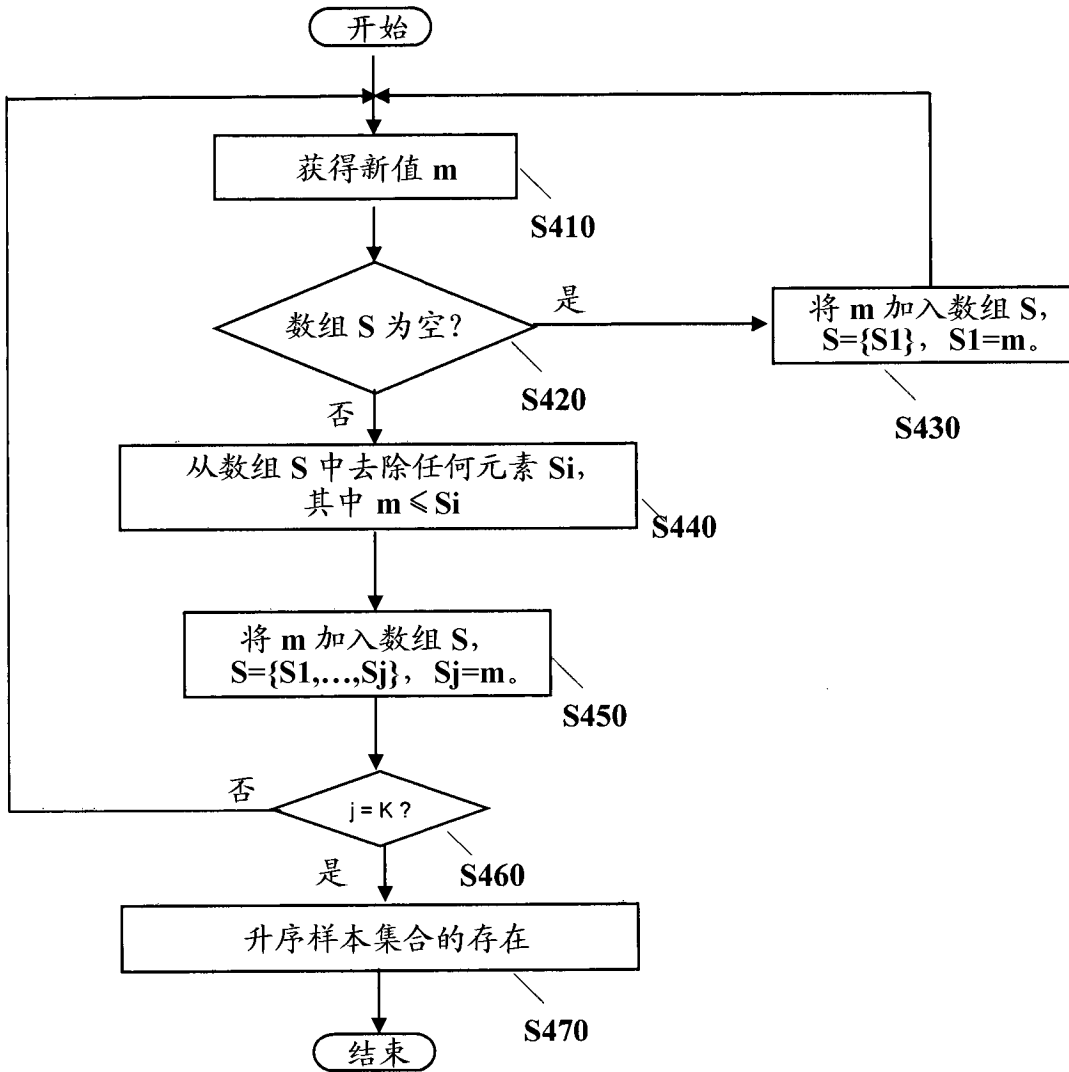


图 4

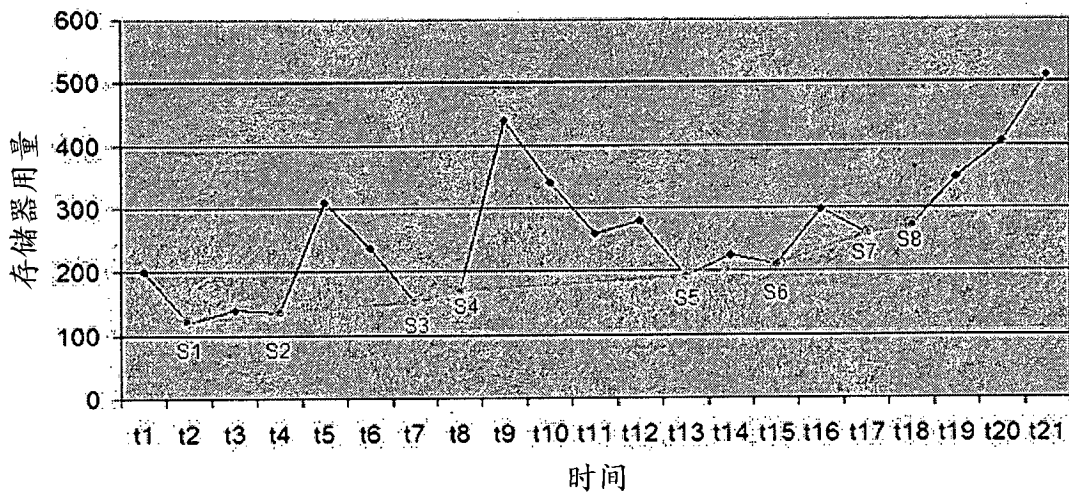


图 5

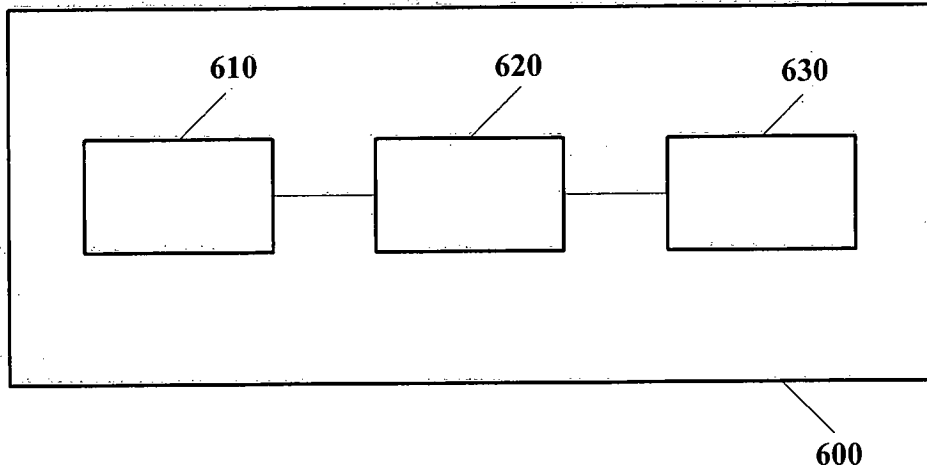


图 6