



(12) 发明专利申请

(10) 申请公布号 CN 102650964 A

(43) 申请公布日 2012. 08. 29

(21) 申请号 201110047972. 2

(22) 申请日 2011. 02. 28

(71) 申请人 国际商业机器公司
地址 美国纽约

(72) 发明人 车漾 罗志达 潘志华 牧黎婧

(74) 专利代理机构 中国国际贸易促进委员会专
利商标事务所 11038

代理人 李镇江

(51) Int. Cl.

G06F 11/30(2006. 01)

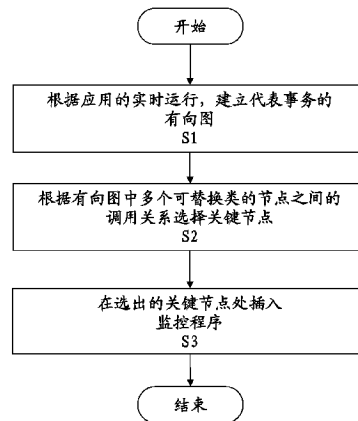
权利要求书 2 页 说明书 7 页 附图 13 页

(54) 发明名称

用于监控面向对象的应用的方法、系统和自监控系统

(57) 摘要

本发明提供了用于监控面向对象的应用的方法、系统和自监控系统。用于监控面向对象的应用的方法包括：根据应用的实时运行，建立代表事务的有向图，所述有向图中包含有多个可替换类的节点；根据有向图中多个可替换类的节点之间的调用关系选择关键节点；在选出的关键节点处插入监控程序。本发明的一个方面使监控面向对象的应用的监控开销大大减小。



1. 一种用于监控面向对象的应用的方法,包括:

根据应用的实时运行,建立代表事务的有向图,所述有向图中包含有多个可替换类的节点;

根据有向图中多个可替换类的节点之间的调用关系选择关键节点;

在选出的关键节点处插入监控程序。

2. 根据权利要求1的方法,还包括:在顶端节点之前增加调用该顶端节点的开始节点,在底端节点之后增加被该底端节点所调用的结束节点,其中顶端节点为不被其它节点调用而仅调用其它节点的节点,底端节点为仅被其它节点调用而不调用其它节点的节点。

3. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:将所述有向图中的所有节点按照入度排序,选取排序中的前n名作为关键节点,其中n为自然数,入度为节点在有向图中被其它节点调用的数量。

4. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:选取有向图中所有入度达到某一阈值的节点作为关键节点,其中入度为节点在有向图中被其它节点调用的数量。

5. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:将所述有向图中所有节点按照入度出度之差排序,选取排序中的前n名作为关键节点,其中n为自然数,入度为节点在有向图中被其它节点调用的数量,出度为节点在有向图中调用其它节点的数量。

6. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:选取有向图中所有入度出度之差达到某一阈值的所有节点作为关键节点,其中入度为节点在有向图中被其它节点调用的数量,出度为节点在有向图中调用其它节点的数量。

7. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:

a. 选择有向图中具有最大入度的节点作为基准节点,入度为节点在有向图中被其它节点调用的数量;

b. 对有向图中与所述基准节点有调用关系的相邻节点进行判断,其中如果所述相邻节点是调用所述基准节点的节点且除了调用所述基准节点之外不再调用其它节点,则从有向图中删除该相邻节点;如果所述相邻节点是被所述基准节点调用的节点且除了被所述基准节点调用之外不再被其它节点调用,则从有向图中删除该相邻节点;

c. 对于每一个未删除的相邻节点,重复过程b,直至所有节点都被遍历。

8. 根据权利要求2的方法,其中根据有向图中多个可替换类的节点之间的调用关系选择关键节点包括:将有向图中所有顶端节点作为关键节点。

9. 根据权利要求1-8中任一个的方法,其中在选出的关键节点处插入监控程序包括在选出的关键节点的前和/或后插入监控程序,若已插入检测程序,则不再重复插入。

10. 一种用于监控面向对象的应用的系统,包括:

事务图建立器,被配置为根据应用的实时运行,建立代表事务的有向图,所述有向图中包含有多个可替换类的节点;

关键节点选择器,被配置为根据有向图中多个可替换类的节点之间的调用关系选择关

键节点；

监控程序插入器，被配置为在选出的关键节点处插入监控程序。

11. 根据权利要求 10 的系统，其中事务图建立器被配置为在顶端节点之前增加调用该顶端节点的开始节点，在底端节点之后增加被该底端节点调用的结束节点，其中顶端节点为不被其它节点调用而仅调用其它节点的节点，底端节点为仅被其它节点调用而不调用其它节点的节点。

12. 根据权利要求 11 的系统，其中关键节点选择器被配置为将所述有向图中的所有节点按照入度排序，选取排序中的前 n 名作为关键节点，其中 n 为自然数，入度为节点在有向图中被其它节点调用的数量。

13. 根据权利要求 11 的系统，其中关键节点选择器被配置为选取有向图中所有入度达到某一阈值的节点作为关键节点，其中入度为节点在有向图中被其它节点调用的数量。

14. 根据权利要求 11 的系统，其中关键节点选择器被配置为将所述有向图中所有节点按照入度出度之差排序，选取排序中的前 n 名作为关键节点，其中 n 为自然数，入度为节点在有向图中被其它节点调用的数量，出度为节点在有向图中调用其它节点的数量。

15. 根据权利要求 11 的系统，其中关键节点选择器被配置为选取有向图中所有入度出度之差达到某一阈值的所有节点作为关键节点，其中入度为节点在有向图中被其它节点调用的数量，出度为节点在有向图中调用其它节点的数量。

16. 根据权利要求 11 的系统，其中关键节点选择器被配置为：

a. 选择有向图中具有最大入度的节点作为基准节点，入度为节点在有向图中被其它节点调用的数量；

b. 对有向图中与所述基准节点有调用关系的相邻节点进行判断，其中如果所述相邻节点是调用所述基准节点的节点且除了调用所述基准节点之外不再调用其它节点，则从有向图中删除该相邻节点；如果所述相邻节点是被所述基准节点调用的节点且除了被所述基准节点调用之外不再被其它节点调用，则从有向图中删除该相邻节点；

c. 对于每一个未删除的相邻节点，重复过程 b，直至所有节点都被遍历。

17. 根据权利要求 11 的系统，其中关键节点选择器被配置为将有向图中所有顶端节点作为关键节点。

18. 根据权利要求 10-17 中任一个的系统，其中检测程序插入器被配置为在选出的关键节点的前和 / 或后插入监控程序，若已插入检测程序，则不再重复插入。

19. 一种自监控系统，包括面向对象的应用和根据权利要求 10-17 中任一个的用于监控面向对象的应用的系统。

用于监控面向对象的应用的方法、系统和自监控系统

技术领域

[0001] 本发明总体上涉及计算机技术,特别地,本发明涉及用于监控面向对象的应用的方法、系统和自监控系统。

背景技术

[0002] 在计算机软件领域,监控计算机软件的运行状态,一直是一项重要的任务。尤其在现代云计算的环境中,实时监控云计算环境中每个节点的软件运行状态,对于了解整个云计算系统的健康状况、维护整个云计算系统的正常运行尤为重要。

[0003] 图 1 是现有技术中的对于面向对象的应用的一个代表事务的有向图。事务是单个逻辑工作单元执行的一系列操作,这些操作组合为一个要么全部成功、要么全部失败的单元。例如,取钱是一个事务。要么取钱成功,要么取钱失败。类是具有相同或相似性质的对象的抽象。例如,图 1 中的 ATM 机取钱、密码认证取钱、指纹认证取钱、瞳孔认证取钱、认证卡取钱就是五个类。这个取钱的事务统一用一个接口来表示。这个接口声明了取钱这个事务,但没有实现,等着每个类分别去实现。在实现时,ATM 机取钱、密码认证取钱、指纹认证取钱、瞳孔认证取钱、认证卡取钱这五个类分别实现了取钱这个接口,但实现的方式不一样。这些具体的实现方式中调用的每个过程就是图 1 中的节点(即方法)。

[0004] 例如,当执行 ATM 机取钱时,节点 10 开始工作,即接受 ATM 机取钱。在该过程中,需要进行密码认证,因此节点 10 调用节点 2。节点 2 在进行密码认证时,需要与密码库或身份数据库进行对比,因此可以调用节点 11 或 8。节点 11 在与密码库进行对比时,需要调用密码库,因此调用节点 12。节点 8 在与身份密码库进行对比时,需要调用身份密码库,因此调用节点 9。图 1 代表了取钱这个完整的事务。

[0005] 在现有技术中,当需要监控面向对象的应用时,先监控应用的实时运行,然后根据应用的实时运行,建立如图 1 所示的代表事务的有向图,此后在图中每个节点的前后都插入监控程序,如图 2 所示。在图 2 中共插入了 15 个监控程序。

[0006] 由于需要在每个节点的前后都插入监控程序,监控开销变得不可接受地大。

发明内容

[0007] 本发明的一个方面解决的一个问题是减小监控面向对象的应用的监控开销。

[0008] 根据本发明的一方面,提供了一种用于监控面向对象的应用的方法,包括:根据应用的实时运行,建立代表事务的有向图,所述有向图中包含有多个可替换类的节点;根据有向图中多个可替换类的节点之间的调用关系选择关键节点;在选出的关键节点处插入监控程序。

[0009] 根据本发明的一方面,提供了一种用于监控面向对象的应用的系统,包括:事务图建立器,被配置为根据应用的实时运行,建立代表事务的有向图,所述有向图中包含有多个可替换类的节点;关键节点选择器,被配置为根据有向图中多个可替换类的节点之间的调用关系选择关键节点;监控程序插入器,被配置为在选出的关键节点处插入监控程序。

[0010] 根据本发明的一方面,提供了一种自监控系统,包括面向对象的应用和根据本发明的上述方面的用于监控面向对象的应用的系统。

[0011] 本发明的一个方面带来的一个有益效果是使监控面向对象的应用的监控开销大大减小。

附图说明

[0012] 本申请中所参考的附图只用于示例本发明的典型实施例,不应该认为是对本发明范围的限制。

[0013] 图 1 是现有技术中的对于面向对象的应用的一个代表事务的有向图。

[0014] 图 2 是现有技术中的插入了监控程序的代表事务的有向图。

[0015] 图 3 是根据本发明的一个实施例的流程图。

[0016] 图 4 是根据本发明的一个实施例的增加了开始节点和结束节点的代表事务的有向图。

[0017] 图 5 是根据本发明的一个实施例的插入了监控程序的代表事务的有向图。

[0018] 图 6 是根据本发明的另一个实施例的插入了监控程序的代表事务的有向图。

[0019] 图 7 是根据本发明的另一个实施例的插入了监控程序的代表事务的有向图。

[0020] 图 8 是根据本发明的另一个实施例的插入了监控程序的代表事务的有向图。

[0021] 图 9 示出了根据本发明的另一个实施例最后留下的关键节点。

[0022] 图 10 是根据本发明的另一个实施例的插入了监控程序的代表事务的有向图。

[0023] 图 11 是根据本发明的另一个实施例的插入了监控程序的代表事务的有向图。

[0024] 图 12A 分别示出了利用本发明和不利用本发明时的 CPU 开销。

[0025] 图 12B 分别示出了利用本发明和不利用本发明时的 Java Heap 开销。

[0026] 图 12C 分别示出了若干事务类型在利用本发明和不利用本发明的情况下的插入了监控程序的类的数量和插入的监控程序函数的数量。

[0027] 图 13 示出了根据本发明的一个实施例的自监控系统。

具体实施方式

[0028] 下列讨论中,提供大量具体的细节以帮助彻底了解本发明。然而,很显然对于本领域技术人员来说,即使没有这些具体细节,并不影响对本发明的理解。并且应该认识到,使用如下的任何具体术语仅仅是为了方便描述,因此,本发明不应当局限于只用在这样的术语所表示和 / 或暗示的任何特定应用中。

[0029] 先来介绍一下本发明的原理。由于面向对象的程序设计中广泛使用适配器和代理。这些适配器和代理并不是特别需要监控的节点。因此,可以考虑避免在这些适配器和代理的前后插入监控程序。这些适配器和代理一般不会有高的入度(入度为某节点被其它节点调用的数量)。另外,不同的可替换类的节点之间往往具有可替换性。因此,根据可替换类的节点之间的调用关系从所有节点中选出若干关键节点,并对关键节点插入监控程序,就可以大大减小监控面向对象的应用的开销。

[0030] 图 3 是根据本发明的一个实施例的流程图。根据本发明的一个实施例的用于监控面向对象的应用的方法包括:在步骤 S1 中,根据应用的实时运行,建立代表事务的有向图,

该有向图中包含有多个可替换类的节点；在步骤 S2 中，根据有向图中多个可替换类的节点之间的调用关系选择关键节点；在步骤 S3 中，在选出的关键节点处插入监控程序。

[0031] 优选地，根据本发明的一个实施例，在顶端节点之前增加调用该顶端节点的开始节点，在底端节点之后增加被该底端节点所调用的结束节点。顶端节点为不被其它节点调用而仅调用其它节点的节点，底端节点为仅被其它节点调用而不调用其它节点的节点。图 4 是根据本发明的一个实施例的增加了开始节点和结束节点的代表事务的有向图。图 4 中，节点 10、1、3、5、7 是顶端节点，节点 12、9 是底端节点。

[0032] 下面举例描述选择关键节点并插入监控程序的方式。

[0033] 实施例 1

[0034] 将图中的所有节点按照入度排序，选取排序中的前 n 名作为关键节点，其中 n 为自然数，入度为节点在图中被其它节点调用的数量。在一个例子中，设 $n = 3$ ，但实际上 n 可以为任意自然数。下面是计算出的图 4 中每个节点的入度：

[0035] 节点 1 :1

[0036] 节点 2 :2

[0037] 节点 3 :1

[0038] 节点 4 :1

[0039] 节点 5 :1

[0040] 节点 6 :1

[0041] 节点 7 :1

[0042] 节点 8 :4

[0043] 节点 9 :1

[0044] 节点 10 :1

[0045] 节点 11 :1

[0046] 节点 12 :1

[0047] 所有开始节点 :0

[0048] 所有结束节点 :1

[0049] 当多个节点具有同样的入度时，一般选择最靠近开始节点的节点。当具有同样的入度、最靠近开始节点的节点也有多个时，可从中任意选择。

[0050] 在本例中，入度最高的节点是节点 8，入度第二高的节点是节点 2。由于存在多个入度第三高、且最靠近开始节点的节点 10、1、3、5、7，任意选择其中的节点 1。

[0051] 因此，在选出的三个关键节点 8、2、1 的前和后插入监控程序。由于在给节点 1 的后面插入监控程序时，监控程序已存在，则不再重复插入。插入的结果如图 5 所示。

[0052] 可选地，也可以只在选出的三个关键节点 8、2、1 的前面插入监控程序。

[0053] 可选地，也可以只在选出的三个关键节点 8、2、1 的后面插入监控程序。

[0054] 由于入度较大的节点往往是可替换性较差的节点，只针对它们插入监控程序，可以大大减小监控面向对象的应用的开销。

[0055] 实施例 2

[0056] 选取图中所有入度达到某一阈值的节点作为关键节点。在本例中，设阈值 = 2，但显然阈值也可以为其它值。

[0057] 在本例中,入度达到 2 的节点只有节点 8、2。因此,只在节点 8、2 的前和后插入监控程序,如图 6 所示。当然,也可以只在节点 8、2 的前或后插入监控程序。

[0058] 由于入度较大的节点往往是可替换性较差的节点,只针对它们插入监控程序,可以大大减小监控面向对象的应用的开销。

[0059] 实施例 3

[0060] 将图中所有节点按照入度出度之差排序,选取排序中的前 n 名作为关键节点,其中 n 为自然数,出度为节点在图中调用其它节点的数量。在本例中,设 $n = 3$,但实际上 n 可以为任意自然数。下面是计算出的图 4 中每个节点的入度与出度之差:

[0061] 节点 1 :0

[0062] 节点 2 :0

[0063] 节点 3 :0

[0064] 节点 4 :0

[0065] 节点 5 :0

[0066] 节点 6 :0

[0067] 节点 7 :0

[0068] 节点 8 :3

[0069] 节点 9 :0

[0070] 节点 10 :0

[0071] 节点 11 :0

[0072] 节点 12 :0

[0073] 所有开始节点 :-1

[0074] 所有结束节点 :1

[0075] 因此,在选出的节点 8 和两个结束节点的前和后插入监控程序。由于结束节点的后面无法插入监控程序,因此只在结束节点的前面插入监控程序。插入的结果如图 7 所示。

[0076] 当然,也可以只在选出的节点的前面或后面插入监控程序。

[0077] 由于入度与出度之差较大的节点往往是可替换性较差的节点,只针对它们插入监控程序,可以大大减小监控面向对象的应用的开销。

[0078] 实施例 4

[0079] 选取图中所有入度出度之差达到某一阈值的所有节点作为关键节点。在本例中,设阈值为 2,但实际上阈值可以为任意自然数。

[0080] 在本例中,入度出度之差达到 2 的节点只有节点 8。因此,只在节点 8 的前和后插入监控程序,如图 8 所示。当然,也可以只在节点 8 的前或后插入监控程序。

[0081] 由于入度与出度之差较大的节点往往是可替换性较差的节点,只针对它们插入监控程序,可以大大减小监控面向对象的应用的开销。

[0082] 实施例 5

[0083] 在实施例 5 中,通过以下方式选择关键节点:

[0084] a. 选择有向图中具有最大入度的节点作为基准节点,入度为节点在有向图中被其它节点调用的数量;

[0085] b. 对有向图中与所述基准节点有调用关系的相邻节点进行判断,其中如果所述相

邻节点是调用所述基准节点的节点且除了调用所述基准节点之外不再调用其它节点,则从有向图中删除该相邻节点;如果所述相邻节点是被所述基准节点调用的节点且除了被所述基准节点调用之外不再被其它节点调用,则从有向图中删除该相邻节点;

[0086] c. 对于每一个未删除的相邻节点,重复过程 b,直至所有节点都被遍历。

[0087] 在本例中,具有最大入度的节点是节点 8。与节点 8 有调用关系的相邻节点是节点 2、4、6、7、9,其中只有节点 2 除了调用节点 8 之外还调用节点 11,因此只保留节点 2,删除其它相邻节点。与节点 2 有调用关系的相邻节点是节点 10、1、11。经判断,这三个节点都不符合保留在图中的条件,因此从图中删除。从图 9 中可以看出,只剩下节点 2 和节点 8 作为关键节点。

[0088] 因此,在关键节点 2 和 8 的前和后插入监控程序,如图 10 所示。

[0089] 可选地,也可以只在关键节点 2、8 的前面或后面插入监控程序。

[0090] 由于上面的过程删除的都是与其它节点的调用关系比较单一的节点,保留的是与其它节点的调用关系比较复杂的节点,因此只针对保留的节点插入监控程序,可以大大减小监控面向对象的应用的开销。

[0091] 实施例 6

[0092] 将图中所有顶端节点作为关键节点。图 4 中的顶端节点为节点 10、1、3、5、7,因此在这些节点的前后插入监控程序,如图 11 所示。

[0093] 这种做法的好处是确保每一个类都能被监控到,同时又减小了部分监控开销。

[0094] 当然,也可以采用实施例 1-6 之外的其它选择关键节点并插入监控程序的方式。本领域技术人员阅读到这里,受益于本发明的以上教导,会作出根据图中多个可替换类的节点之间的调用关系选择关键节点、并在选出的关键节点处插入监控程序的任何其它替代方式。

[0095] 图 12A、B 分别示出了利用本发明和不利用本发明时的 CPU 开销、Java Heap 开销。图 12C 分别示出了若干事务类型在利用本发明和不利用本发明的情况下的插入了监控程序的类的数量和插入的监控程序函数的数量。从图中可以看出,利用了本发明时无论是插入了监控程序的类的数量还是插入的监控程序函数的数量都大大减小。本发明大大减小了监控面向对象的应用的开销。

[0096] 图 13 示出了根据本发明的一个实施例的自监控系统 1300。自监控系统 1300 包括面向对象的应用 1301 和用于监控面向对象的应用的系统 1302。用于监控面向对象的应用的系统 1302 包括:事务图建立器 1303,被配置为根据应用的实时运行,建立代表事务的有向图,所述有向图中包含有多个可替换类的节点;关键节点选择器 1304,被配置为根据有向图中多个可替换类的节点之间的调用关系选择关键节点;监控程序插入器 1305,被配置为在选出的关键节点处插入监控程序。

[0097] 所属技术领域的技术人员知道,本发明可以体现为系统、方法或计算机程序产品。因此,本发明可以具体实现为以下形式,即,可以是完全的硬件、完全的软件(包括固件、驻留软件、微代码等)、或者本文一般称为“电路”、“模块”或“系统”的软件部分与硬件部分的组合。此外,本发明还可以采取体现在任何有形的表达介质(medium of expression)中的计算机程序产品的形式,该介质中包含计算机可用的程序码。

[0098] 可以使用一个或多个计算机可用的或计算机可读的介质的任何组合。计算机可

用的或计算机可读的介质例如可以是——但不限于——电的、磁的、光的、电磁的、红外线的、或半导体的系统、装置、器件或传播介质。计算机可读介质的更具体的例子（非穷举的列表）包括以下：有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦式可编程只读存储器（EPROM 或闪存）、光纤、便携式紧凑磁盘只读存储器（CD-ROM）、光存储器件、诸如支持因特网或内部网的传输介质、或者磁存储器件。注意计算机可用的或计算机可读的介质甚至可以是上面印有程序的纸张或者其它合适的介质，这是因为，例如可以通过电扫描这种纸张或其它介质，以电子方式获得程序，然后以适当的方式加以编译、解释或处理，并且必要的话在计算机存储器中存储。在本文件的语境中，计算机可用的或计算机可读的介质可以是任何含有、存储、传达、传播、或传输供指令执行系统、装置或器件使用的或与指令执行系统、装置或器件相联系的程序的介质。计算机可用的介质可包括在基带中或者作为载波一部分传播的、由其体现计算机可用的程序码的数据信号。计算机可用的程序码可以用任何适当的介质传输，包括—但不限于—无线、电线、光缆、RF 等等。

[0099] 用于执行本发明的操作的计算机程序码，可以以一种或多种程序设计语言的任何组合来编写，所述程序设计语言包括面向对象的程序设计语言—诸如 Java、Smalltalk、C++ 之类，还包括常规的过程式程序设计语言—诸如“C”程序设计语言或类似的设计语言。程序码可以完全地在用户的计算上执行、部分地在用户的计算机上执行、作为一个独立的软件包执行、部分在用户的计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在后一种情形中，远程计算机可以通过任何种类的网络——包括局域网（LAN）或广域网（WAN）—连接到用户的计算机，或者，可以（例如利用因特网服务提供商来通过因特网）连接到外部计算机。

[0100] 以下参照按照本发明实施例的方法、装置（系统）和计算机程序产品的流程图和/或框图描述本发明。要明白的是，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理装置，从而生产出一种机器，使得通过计算机或其它可编程数据处理装置执行的这些指令，产生实现流程图和/或框图中的方框中规定的功能/操作的装置。

[0101] 也可以把这些计算机程序指令存储在能指令计算机或其它可编程数据处理装置以特定方式工作的计算机可读介质中，这样，存储在计算机可读介质中的指令产生一个包括实现流程图和/或框图中的方框中规定的功能/操作的指令装置（instruction means）的制造品。

[0102] 也可以把计算机程序指令加载到计算机或其它可编程数据处理装置上，使得在计算机或其它可编程数据处理装置上执行一系列操作步骤，以产生计算机实现的过程，从而在计算机或其它可编程装置上执行的指令就提供实现流程图和/或框图中的方框中规定的功能/操作的过程。

[0103] 附图中的流程图和框图，图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，所述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所

标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的，框图和 / 或流程图中的每个方框、以及框图和 / 或流程图中的方框的组合，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0104] 本文中所述的术语，仅仅是为了描述特定的实施例，而不意图限定本发明。本文中所述的单数形式的“一”和“该”，旨在也包括复数形式，除非上下文中明确地另外指出。还要知道，“包含”一词在本说明书中使用，说明存在所指出的特征、整体、步骤、操作、单元和 / 或组件，但是并不排除存在或增加一个或多个其它特征、整体、步骤、操作、单元和 / 或组件，以及 / 或者它们的组合。

[0105] 权利要求中的对应结构、材料、操作以及所有功能性限定的装置 (means) 或步骤的等同替换，旨在包括任何用于与在权利要求中具体指出的其它单元相组合地执行该功能的结构、材料或操作。所给出的对本发明的描述其目的在于示意和描述，并非是穷尽性的，也并非是要把本发明限定到所表述的形式。对于所属技术领域的普通技术人员来说，在不偏离本发明范围和精神的情况下，显然可以作出许多修改和变型。对实施例的选择和说明，是为了最好地解释本发明的原理和实际应用，使所属技术领域的普通技术人员能够明了，本发明可以有适合所要的特定用途的具有各种改变的各种实施方式。

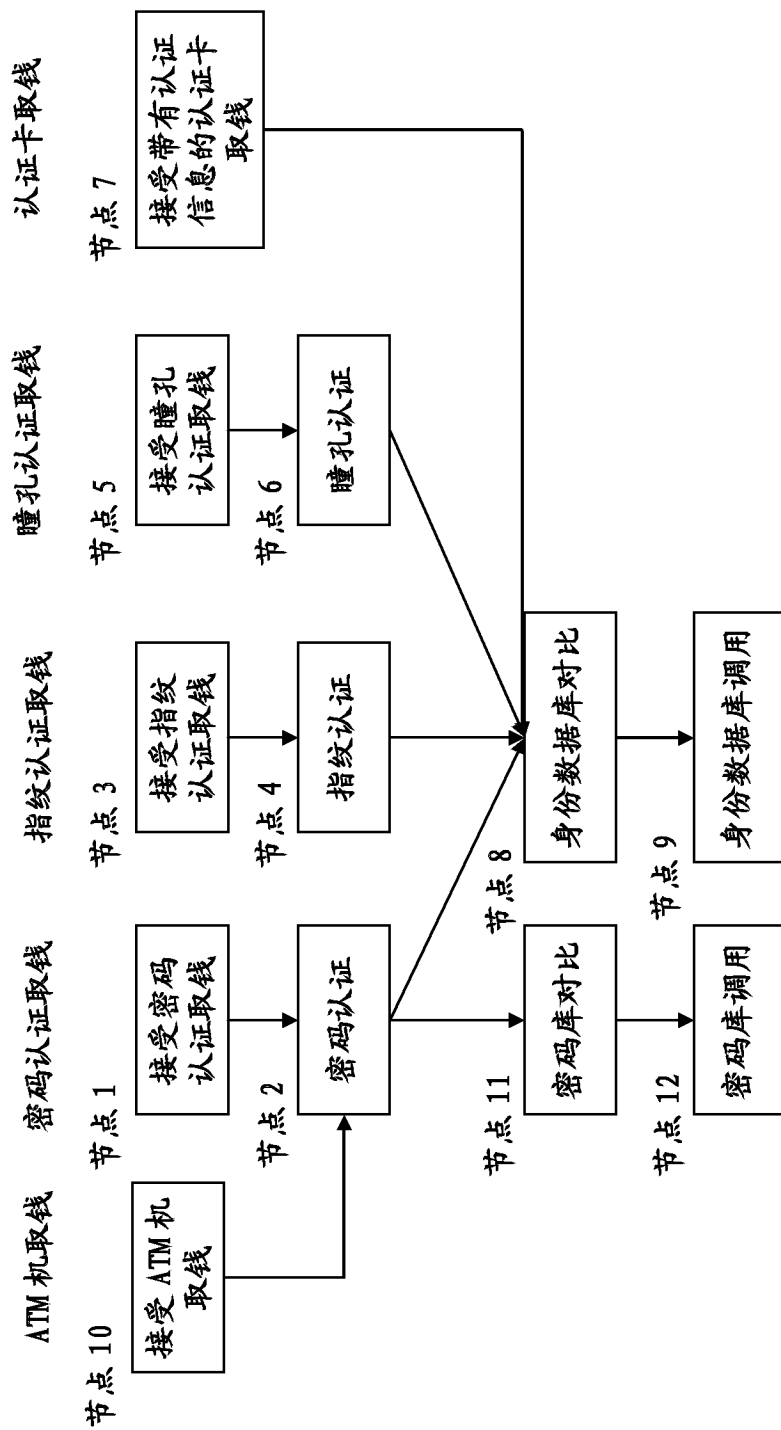


图 1

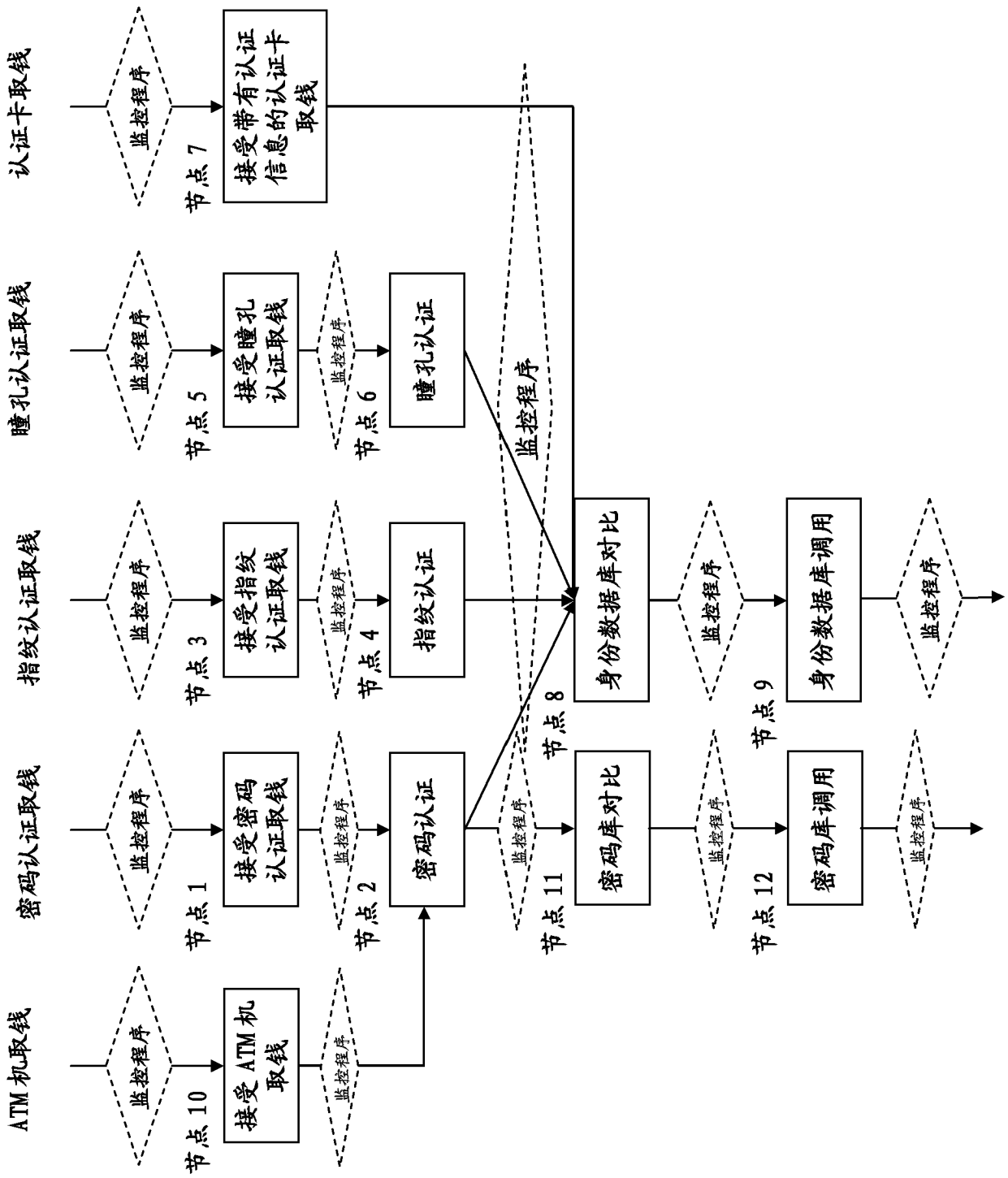


图 2

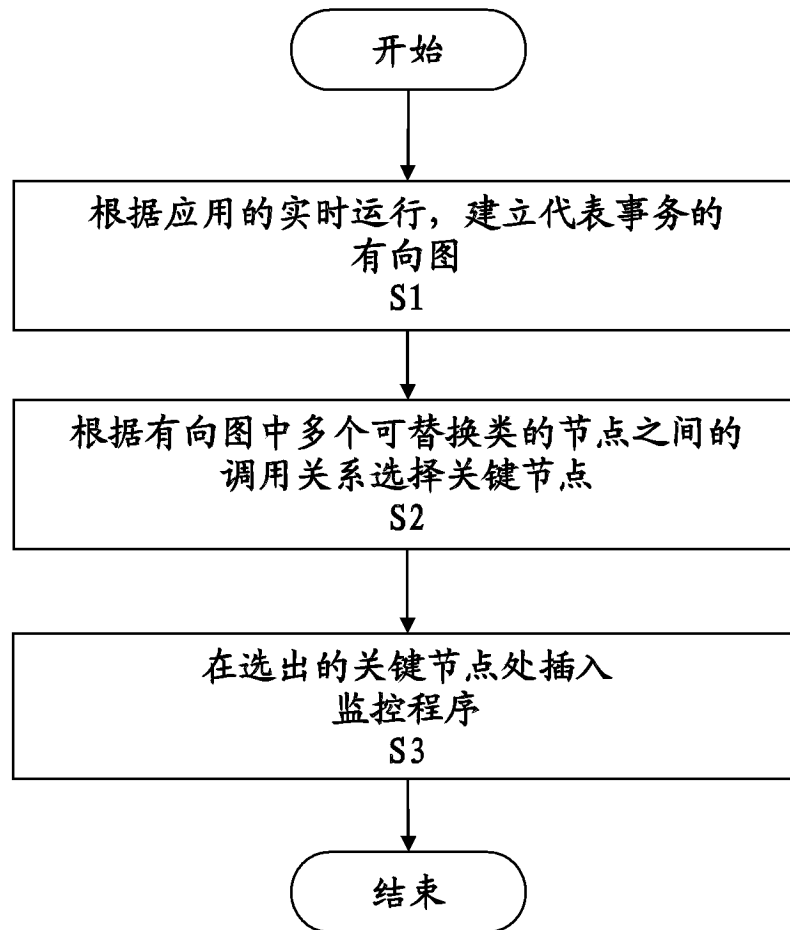


图 3

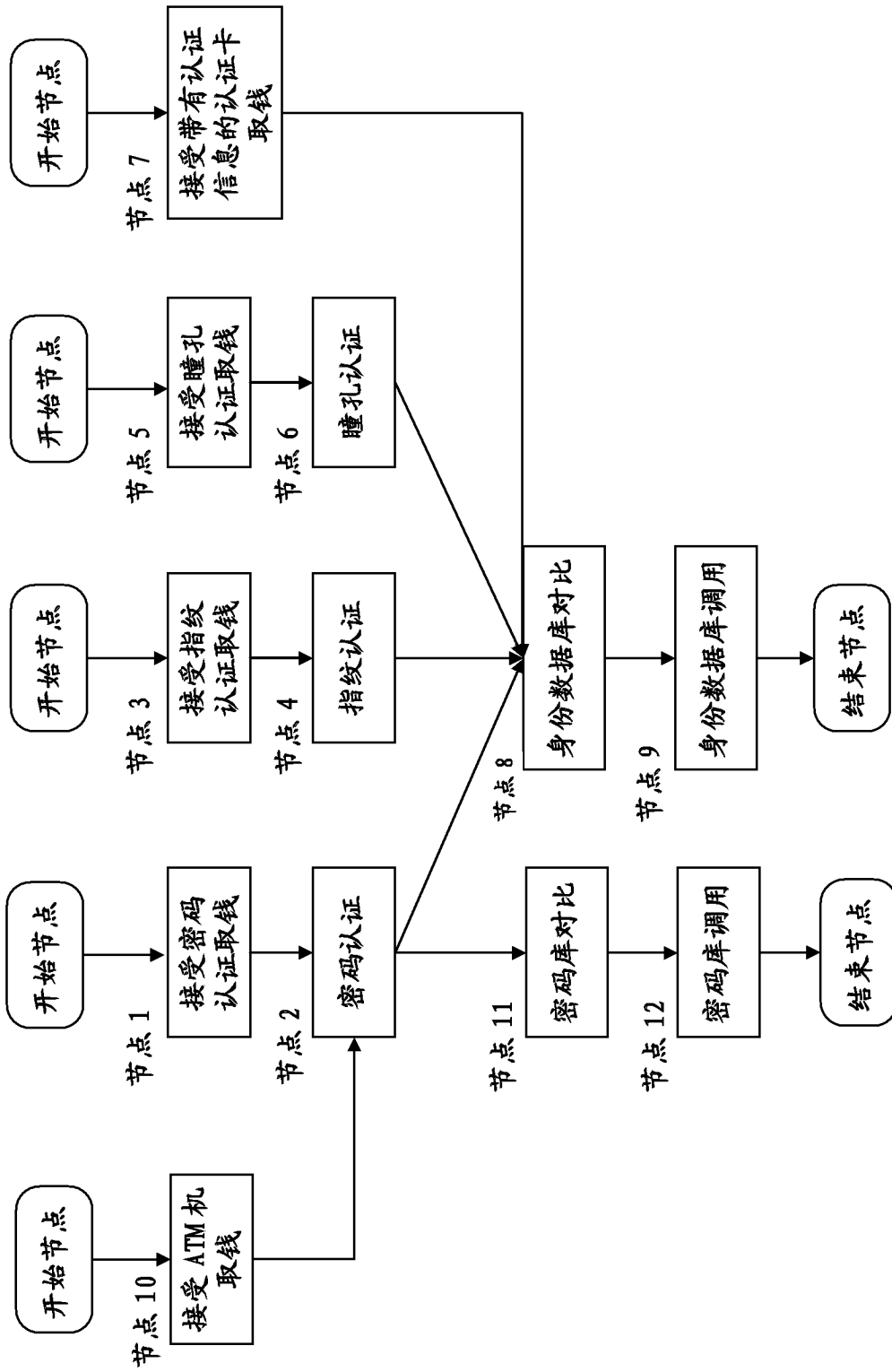


图 4

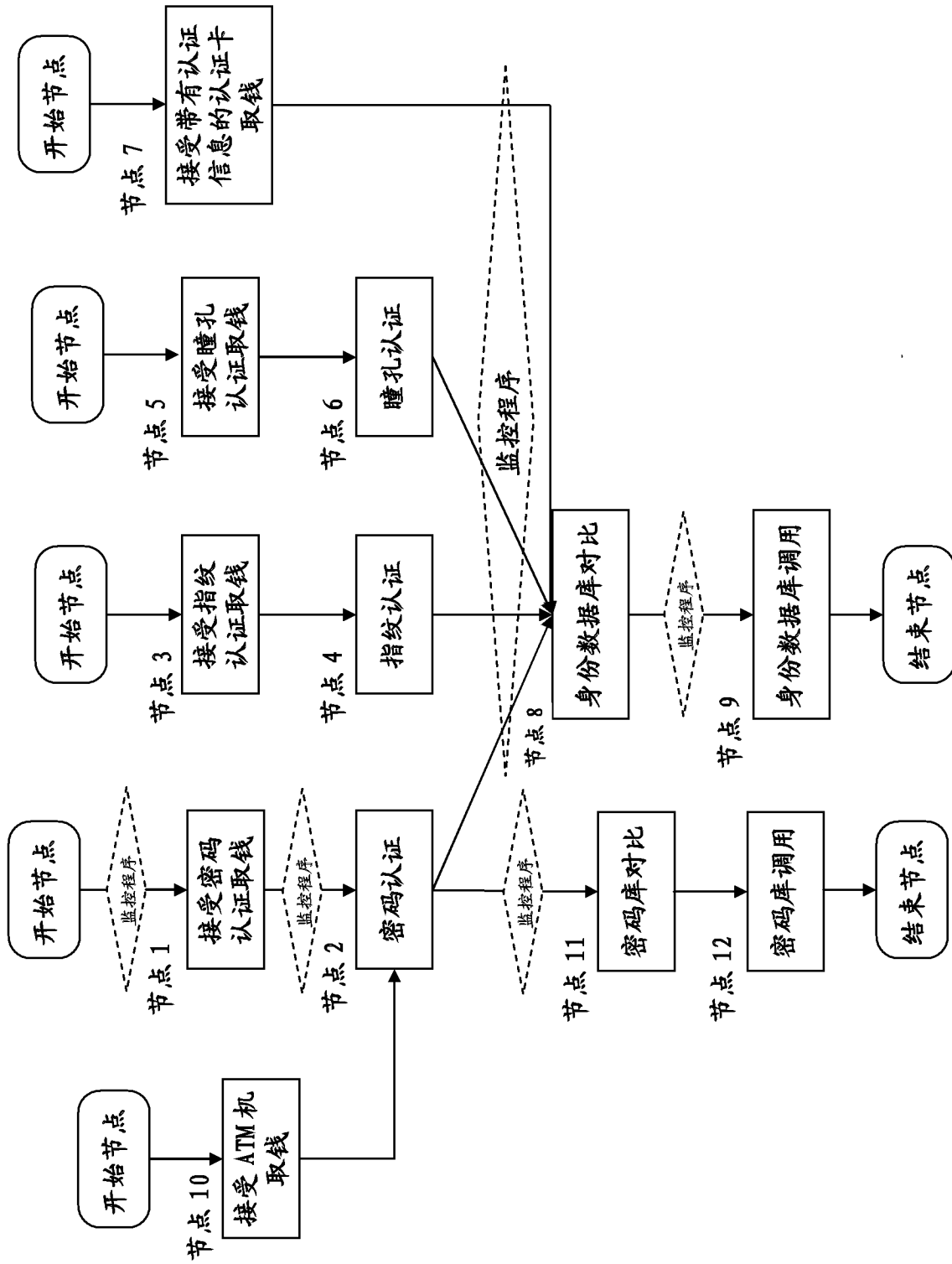


图 5

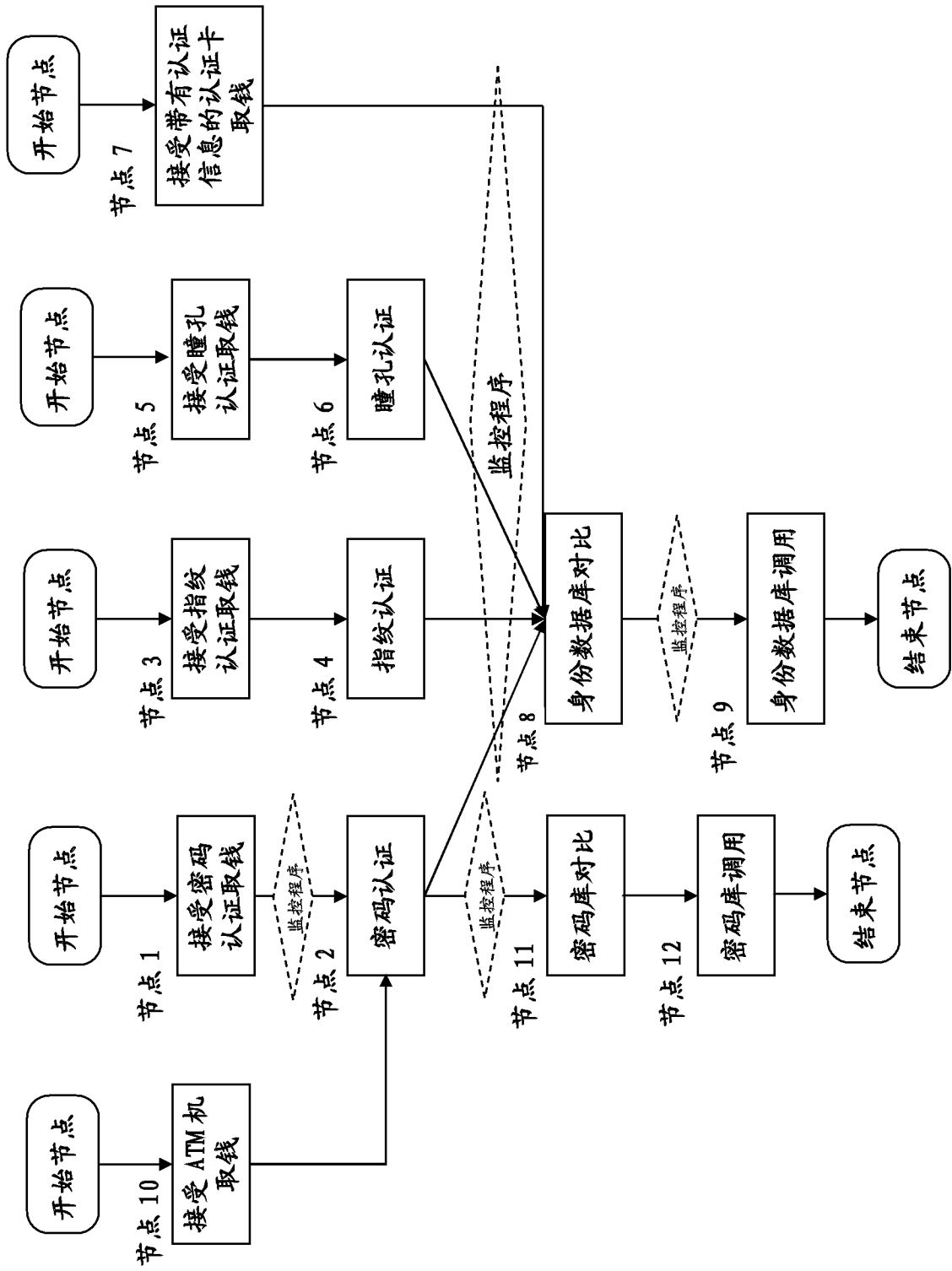


图 6

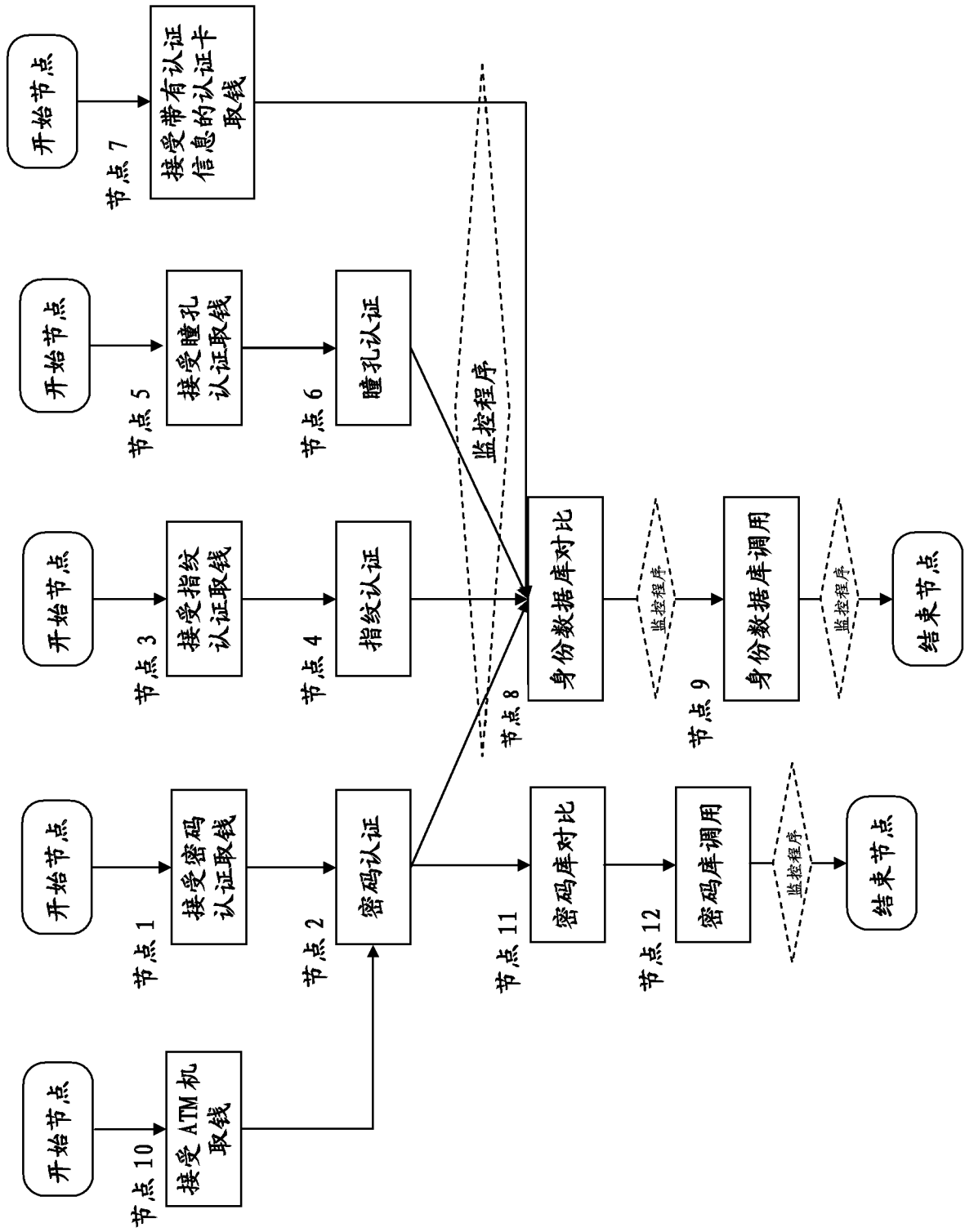


图 7

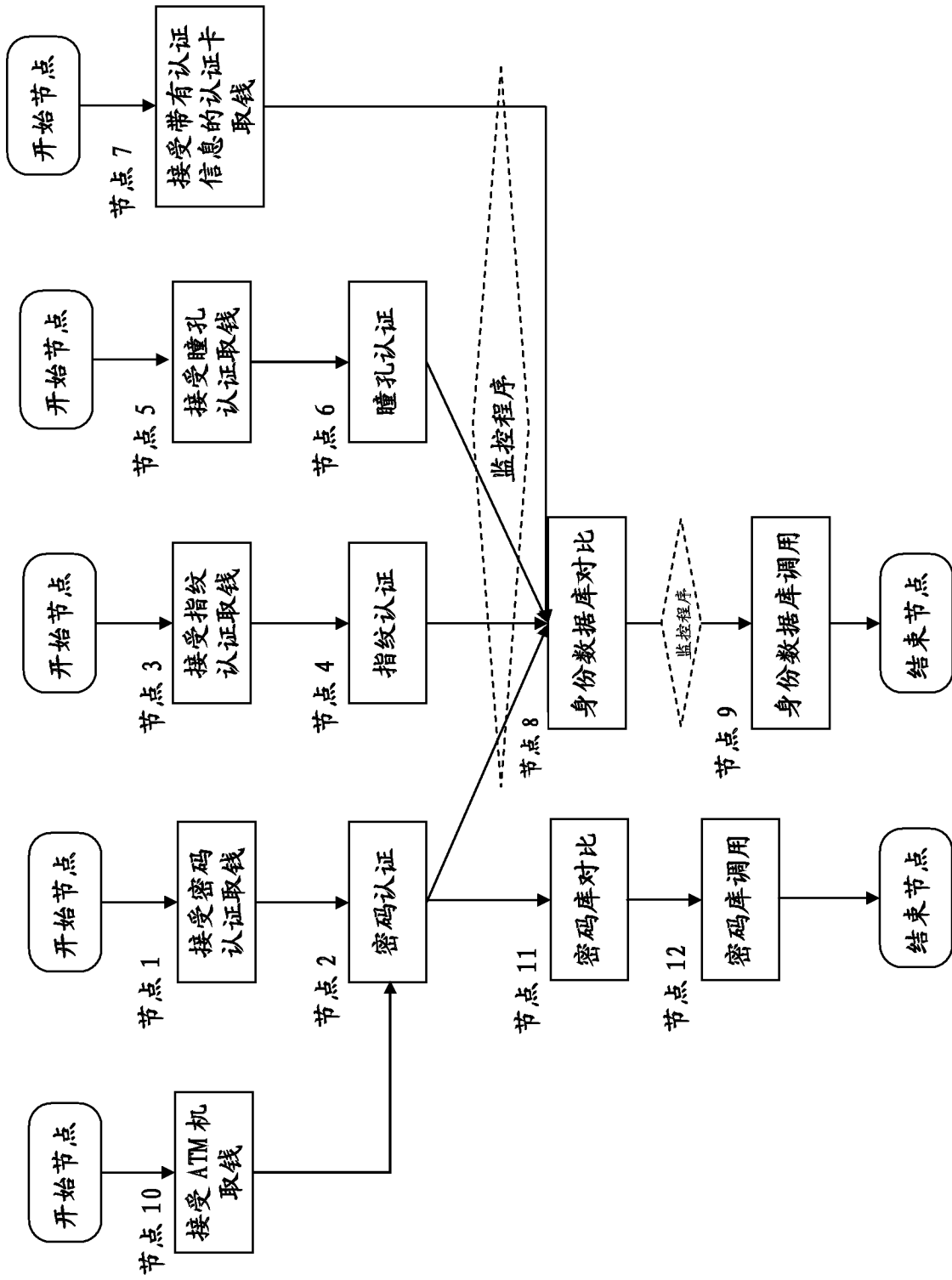


图 8

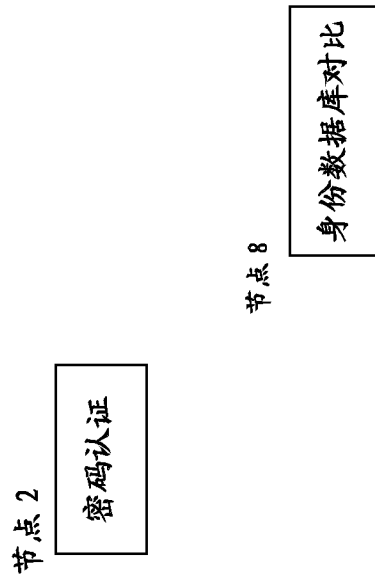


图 9

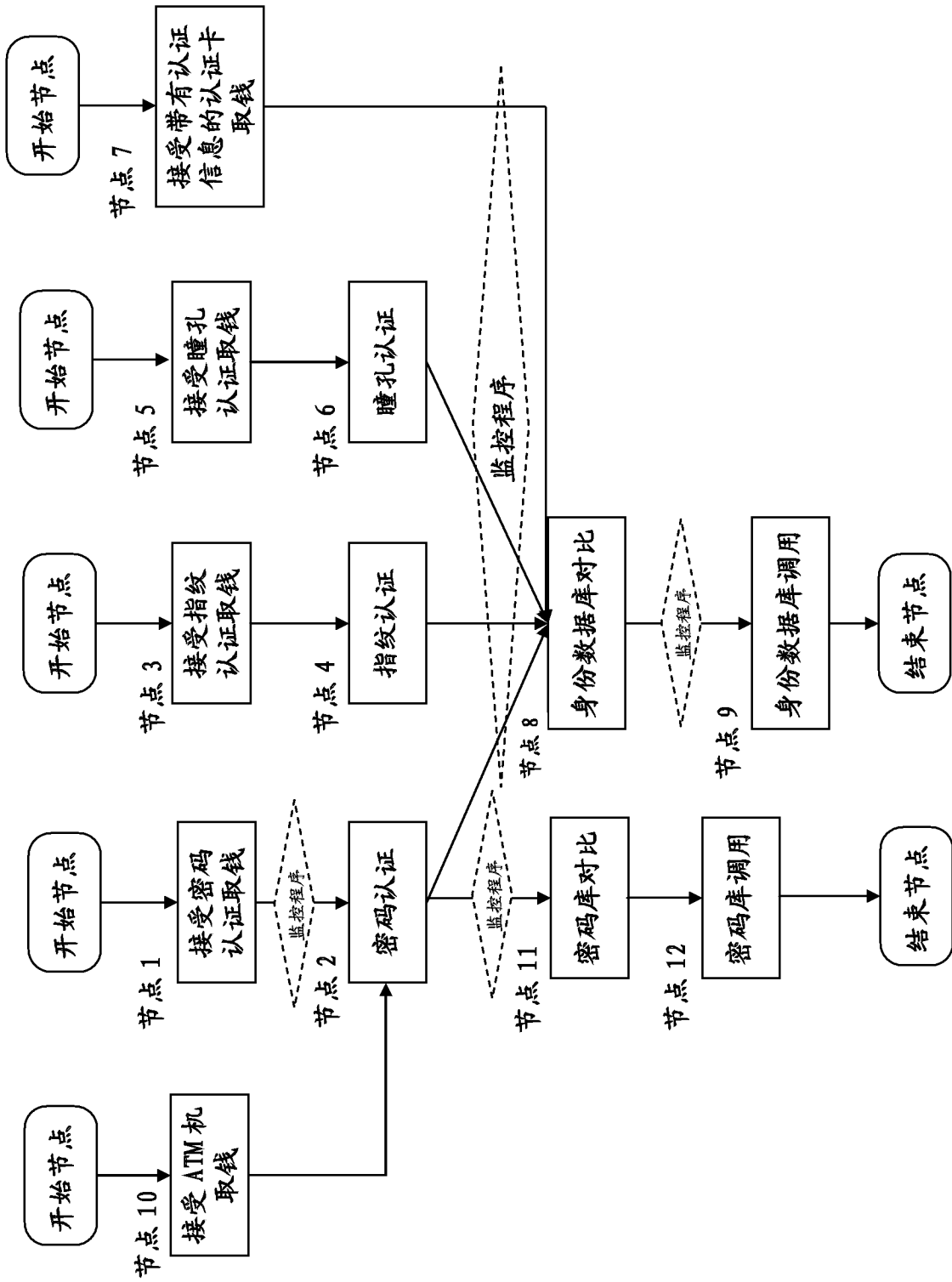


图 10

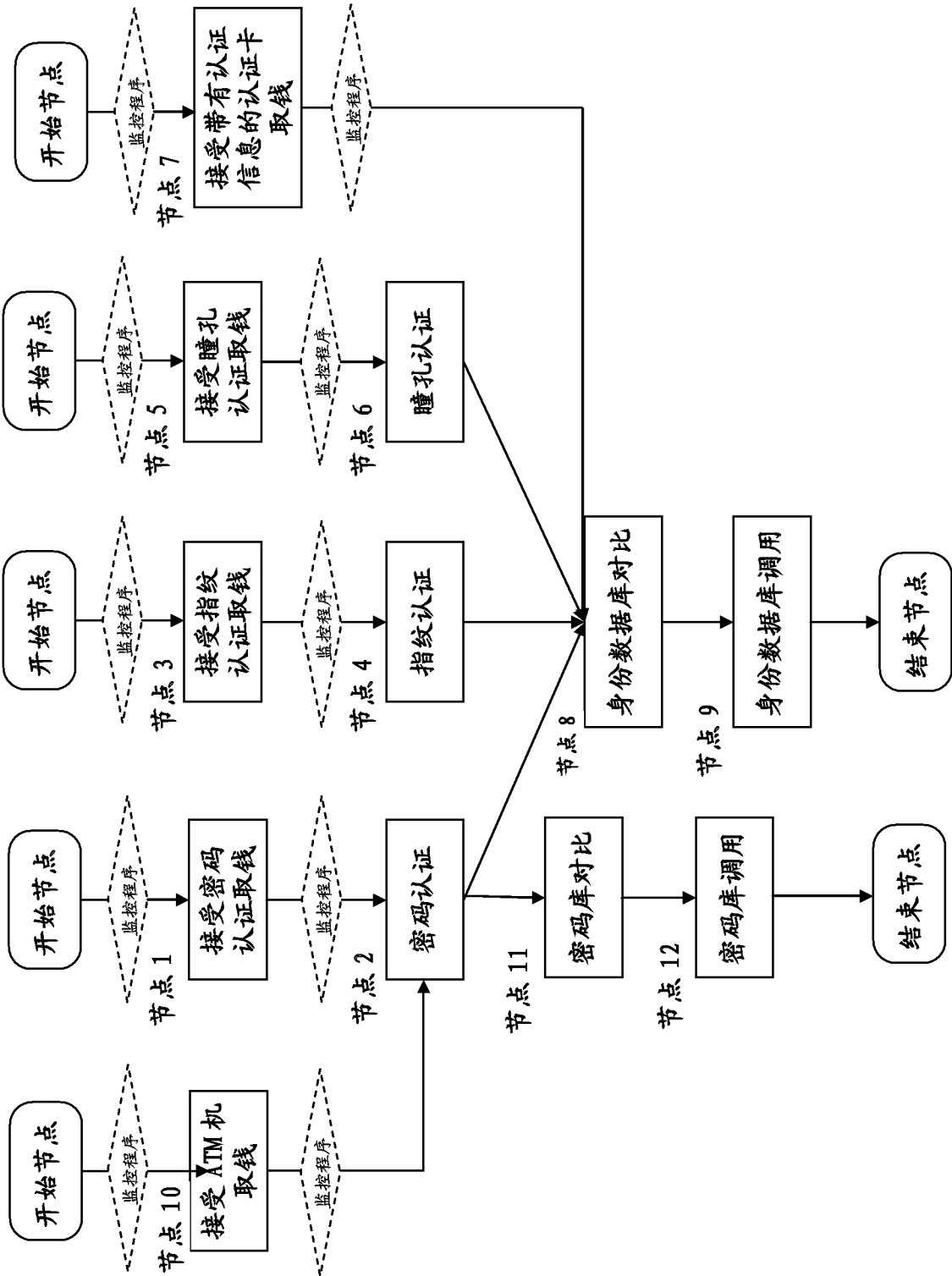


图 11

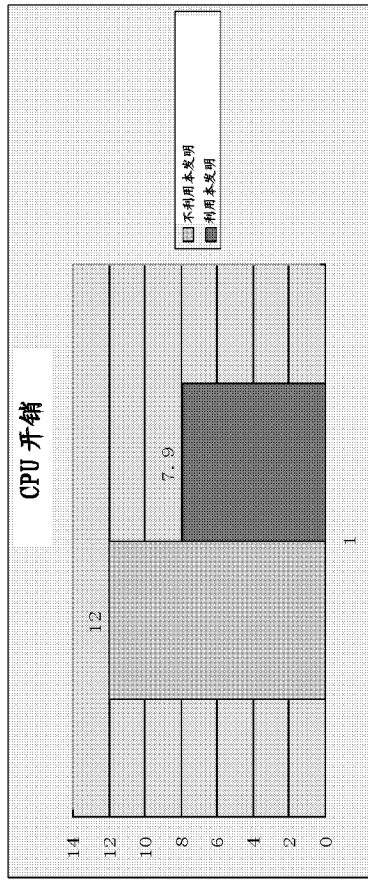


图 12A

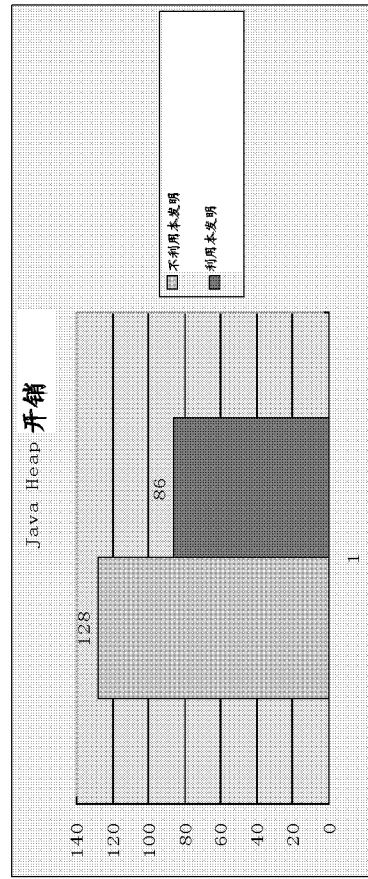


图 12B

事务类型	不利用本发明		利用本发明	
	插入了监控程序的类的数量	插入了监控程序的函数的数量	插入了监控程序的类的数量	插入了监控程序的函数的数量
JDBC(DB2)	43	135	6	15
JSP	2799	2799	1	1
Servlet	1481	2962	1	2
EJB	1545	8439	1	1
...				

图 12C

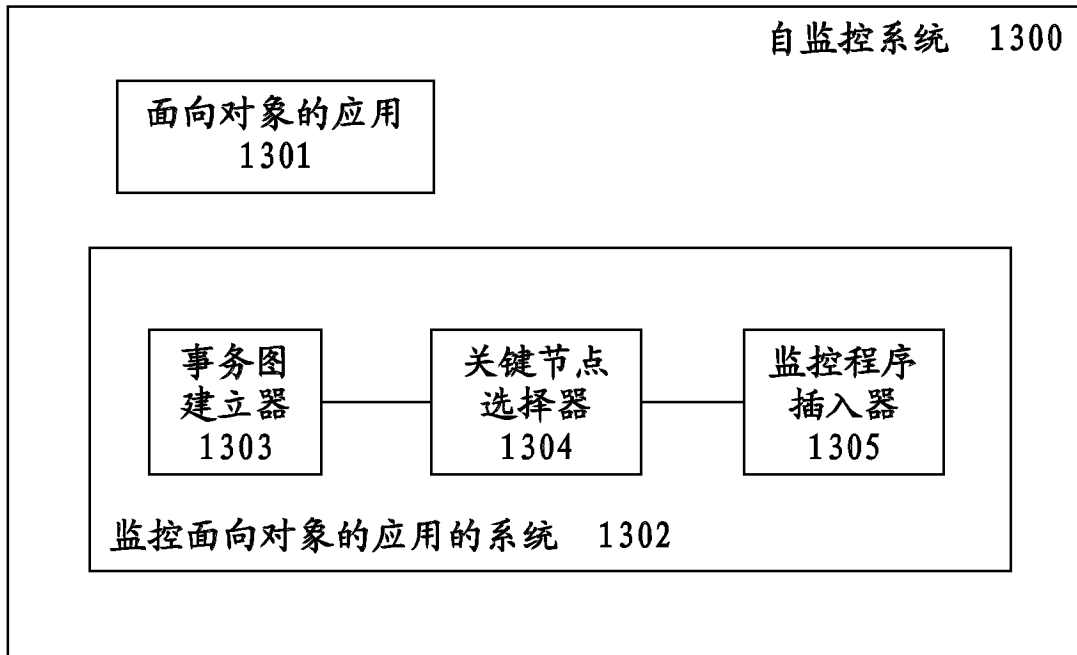


图 13