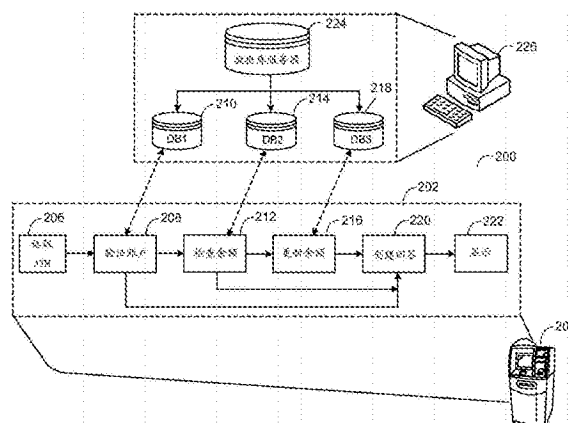




(45)授权公告日 2016.09.14



1. 一种计算机实现的方法,包括:

在初始化进程时存储与所述进程的初始状态有关的信息,其中所述进程的执行包括执行一个或多个执行阶段,所述一个或多个执行阶段以初始执行阶段为开始,并在完成各个所述执行阶段的执行时存储代表所述执行阶段的结束状态的信息;

响应于预定事件中止所述进程的特定执行阶段,同时维持所述进程的用以检查与所述预定事件相关联的状况的能力;以及

基于表示所述预定事件被认为已清除的所述状况,从所保存的初始和结束状态中的一个恢复所述进程的所述特定执行阶段,而不需要关闭所述进程。

2. 如权利要求1所述的计算机实现的方法,其中所述预定事件代表到外部设备的连接的丢失。

3. 如权利要求1所述的计算机实现的方法,其中所述预定事件代表外部设备的错误。

4. 如权利要求2所述的计算机实现的方法,其中当已重建到所述外部设备的连接时,恢复所述特定执行阶段。

5. 如权利要求3所述的计算机实现的方法,其中当已清除所述外部设备的错误时,恢复所述特定执行阶段。

6. 如权利要求1所述的计算机实现的方法,其中从先于所述预定事件发生的特定执行阶段所存储的结束状态恢复所述特定执行阶段。

7. 如权利要求1所述的计算机实现的方法,其中如果所述预定事件发生在所述进程的起动之后的特定执行阶段期间,则从初始状态恢复所述特定执行阶段。

8. 如权利要求1所述的计算机实现的方法,其中执行阶段的执行包括对接收到的数据流执行一个或多个处理动作以便产生与所述执行阶段对应的输出数据。

9. 如权利要求8所述的计算机实现的方法,包括:

存储与一个或多个执行阶段对应的输出数据;以及

当恢复所述特定执行阶段时再现所述输出数据。

10. 如权利要求1所述的计算机实现的方法,其中所述进程是进程级中的部分,并通过数据路径与所述进程级中的第二、不同的进程通信。

11. 如权利要求10所述的计算机实现的方法,包括:

传递检查点命令消息通过所述进程级中的每个进程;以及

在每个进程,在接收到检查点命令消息时,保存与初始或结束状态有关的新的信息,其中所述保存包括暂停所述进程的操作并将所述新的信息保存到存储区域。

12. 如权利要求11所述的计算机实现的方法,包括:

使用新的初始或结束状态覆写旧的保存的初始或结束状态。

13. 如权利要求11所述的计算机实现的方法,其中每个进程与数据队列通信用于为所述进程接收数据并将所述数据排队。

14. 如权利要求11所述的计算机实现的方法,包括:

响应于检测到触发事件生成所述检查点命令消息。

15. 如权利要求14所述的计算机实现的方法,其中所述触发事件包括关于网络事件的信息。

16. 如权利要求11所述的计算机实现的方法,包括:

周期性地生成所述检查点命令消息。

17. 如权利要求11所述的计算机实现的方法,包括:

响应于正在处理的引入数据记录内或从所述引入数据记录导出的所选的数据值的出现生成所述检查点命令消息。

18. 如权利要求11所述的计算机实现的方法,包括:

响应于所述预定事件生成中止命令消息;

传递所述中止命令消息通过所述进程级中的每个进程;以及

在每个进程接收到所述中止命令消息时,中止所述进程的执行并向下一进程传递所述中止命令消息。

19. 如权利要求1所述的计算机实现的方法,包括:

部分基于在恢复处理消息中包含的信息从所保存的初始或结束状态中的所选的一个恢复所述特定执行阶段。

20. 如权利要求18所述的计算机实现的方法,包括:

在初始化所述进程并已保存与所述进程的初始状态有关的信息之后的第一执行阶段期间接收所述中止命令消息;以及

从所保存的初始状态恢复所述特定执行阶段,而不需要关闭和重启所述进程。

21. 一种计算系统,包括:

输入设备或端口,其被配置为在初始化进程时接收并存储与所述进程的初始状态有关的信息,其中所述进程的执行包括执行一个或多个执行阶段,所述一个或多个执行阶段以初始执行阶段为开始,并在完成各个所述执行阶段的执行时存储代表所述执行阶段的结束状态的信息;以及

至少一个处理器,其被配置为:

响应于预定事件中止所述进程的特定执行阶段,同时维持所述进程的用以检查与所述预定事件相关联的状况的能力;以及

基于表示所述预定事件被认为已清除的所述状况,从所保存的初始和结束状态中的一个恢复所述进程的所述特定执行阶段,而不需要关闭所述进程。

22. 如权利要求21所述的计算系统,其中所述预定事件代表到外部设备的连接的丢失。

23. 如权利要求21所述的计算系统,其中所述预定事件代表外部设备的错误。

24. 如权利要求22所述的计算系统,其中当已重建到所述外部设备的连接时,恢复所述特定执行阶段。

25. 如权利要求23所述的计算系统,其中当已清除所述外部设备的错误时,恢复所述特定执行阶段。

26. 如权利要求21所述的计算系统,其中从先于所述预定事件发生的特定执行阶段所存储的结束状态恢复所述特定执行阶段。

27. 如权利要求21所述的计算系统,其中如果所述预定事件发生在所述进程的起动之后的特定执行阶段期间,则从初始状态恢复所述特定执行阶段。

28. 如权利要求21所述的计算系统,其中执行阶段的执行包括对接收到的数据流执行一个或多个处理动作以便产生与所述执行阶段对应的输出数据。

29. 如权利要求28所述的计算系统,其中所述至少一个处理器被配置为:

存储与一个或多个执行阶段对应的输出数据;以及
当恢复所述特定执行阶段时再现所述输出数据。

30. 如权利要求21所述的计算系统,其中所述进程是进程级中的部分,并通过数据路径与所述进程级中的第二、不同的进程通信。

31. 如权利要求30所述的计算系统,其中所述至少一个处理器被配置为:
传递检查点命令消息通过所述进程级中的每个进程;以及

在每个进程,在接收到检查点命令消息时,保存与初始或结束状态有关的新的信息,其中所述保存包括暂停所述进程的操作并将所述新的信息保存到存储区域。

32. 如权利要求31所述的计算系统,其中所述至少一个处理器被配置为:
使用新的初始或结束状态覆盖旧的保存的初始或结束状态。

33. 如权利要求31所述的计算系统,其中每个进程与数据队列通信用于为所述进程接收数据并将所述数据排队。

34. 如权利要求31所述的计算系统,其中所述至少一个处理器被配置为:
响应于检测到触发事件生成所述检查点命令消息。

35. 如权利要求34所述的计算系统,其中所述触发事件包括关于网络事件的信息。

36. 如权利要求31所述的计算系统,其中所述至少一个处理器被配置为:
周期性地生成所述检查点命令消息。

37. 如权利要求31所述的计算系统,其中所述至少一个处理器被配置为:
响应于正在处理的引入数据记录内或从所述引入数据记录导出的所选的数据值的出现生成所述检查点命令消息。

38. 如权利要求31所述的计算系统,其中所述至少一个处理器被配置为:
响应于所述预定事件生成中止命令消息;
传递所述中止命令消息通过所述进程级中的每个进程;以及
在每个进程接收到所述中止命令消息时,中止所述进程的执行并向下一进程传递所述中止命令消息。

39. 如权利要求21所述的计算系统,其中所述至少一个处理器被配置为:
部分基于在恢复处理消息中包含的信息从所保存的初始或结束状态中的所选的一个恢复所述特定执行阶段。

40. 如权利要求38所述的计算系统,其中所述至少一个处理器被配置为:
在初始化所述进程并已保存与所述进程的初始状态有关的信息之后的第一执行阶段期间接收所述中止命令消息;以及
从所保存的初始状态恢复所述特定执行阶段,而不需要关闭和重启所述进程。

41. 一种计算系统,包括:

用于在初始化进程时存储与所述进程的初始状态有关的信息的装置,其中所述进程的
执行包括执行一个或多个执行阶段,所述一个或多个执行阶段以初始执行阶段为开始,且
所述装置在完成各个所述执行阶段的执行时,存储代表所述执行阶段的结束状态的信息;
以及

用于控制所述进程的执行的装置,所述控制包括:

响应于预定事件中止所述进程的特定执行阶段,同时维持所述进程的用以检查与所述

预定事件相关联的状况的能力;以及

基于表示所述预定事件被认为已清除的所述状况,从所保存的初始和结束状态中的一个恢复所述进程的所述特定执行阶段,而不需要关闭所述进程。

42.如权利要求41所述的计算系统,其中所述预定事件代表到外部设备的连接的丢失。

43.如权利要求41所述的计算系统,其中所述预定事件代表外部设备的错误。

44.如权利要求42所述的计算系统,其中当已重建到所述外部设备的连接时,恢复所述特定执行阶段。

45.如权利要求43所述的计算系统,其中当已清除所述外部设备的错误时,恢复所述特定执行阶段。

46.如权利要求41所述的计算系统,其中从先于所述预定事件发生的特定执行阶段所存储的结束状态恢复所述特定执行阶段。

47.如权利要求41所述的计算系统,其中如果所述预定事件发生在所述进程的起动之后的特定执行阶段期间,则从所述初始状态恢复所述特定执行阶段。

48.如权利要求41所述的计算系统,其中执行阶段的执行包括对接收到的数据流执行一个或多个处理动作以便产生与所述执行阶段对应的输出数据。

49.如权利要求48所述的计算系统,所述控制包括:

存储与一个或多个执行阶段对应的输出数据;以及

当恢复所述特定执行阶段时再现所述输出数据。

50.如权利要求41所述的计算系统,其中所述进程是进程级中的部分,并通过数据路径与所述进程级中的第二、不同的进程通信。

51.如权利要求50所述的计算系统,所述控制包括:

传递检查点命令消息通过所述进程级中的每个进程;以及

在每个进程,在接收到检查点命令消息时,保存与初始或结束状态有关的新的信息,其中所述保存包括暂停所述进程的操作并将所述新的信息保存到存储区域。

52.如权利要求51所述的计算系统,所述控制包括:

使用新的初始或结束状态覆写旧的保存的初始或结束状态。

53.如权利要求51所述的计算系统,其中每个进程与数据队列通信用于为所述进程接收数据并将所述数据排队。

54.如权利要求51所述的计算系统,所述控制包括:

响应于检测到触发事件生成所述检查点命令消息。

55.如权利要求54所述的计算系统,其中所述触发事件包括关于网络事件的信息。

56.如权利要求51所述的计算系统,所述控制包括:

周期性地生成所述检查点命令消息。

57.如权利要求51所述的计算系统,所述控制包括:

响应于正在处理的引入数据记录内或从所述引入数据记录导出的所选的数据值的出现生成所述检查点命令消息。

58.如权利要求51所述的计算系统,所述控制包括:

响应于所述预定事件生成中止命令消息;

传递所述中止命令消息通过所述进程级中的每个进程;以及

在每个进程接收到所述中止命令消息时,中止所述进程的执行并向下一进程传递所述中止命令消息。

59.如权利要求41所述的计算系统,所述控制包括:

部分基于在恢复处理消息中包含的信息从所保存的初始或结束状态中的所选的一个恢复所述特定执行阶段。

60.如权利要求58所述的计算系统,所述控制包括:

在初始化所述进程并已保存与所述进程的初始状态有关的信息之后的第一执行阶段期间接收所述中止命令消息;以及

从所保存的初始状态恢复所述特定执行阶段,而不需要关闭和重启所述进程。

重启进程

[0001] 相关申请的交叉引用

[0002] 本申请对2011年2月18日提交的、名称为“Restarting Processes”的美国申请序列No.13/030,998要求优先权,其全部内容通过引用合并于此。

技术领域

[0003] 本说明涉及重启进程。

背景技术

[0004] 由单处理器计算机提供的计算速度在过去数十年已极大地提高。然而,由这样的处理器执行的许多应用可能需要超过甚至最快的单处理器计算机的计算能力。例如,在例如航空预约系统的交易系统中,多个用户可能同时访问计算机资源。这些用户通常期望低响应时间。单处理器计算机可能不能跟上这样的需求。已开发了例如并行处理系统的各种架构来处理这样的应用以便提高性能。通常,并行处理系统使用可能位于单个地点或远程分布的多个处理器。由于它们的处理能力,对于处理大量数据的应用已变得依赖这样的并行处理系统,在某些情况下该处理大量数据的应用可以包括基本上连续的并接近实时的处理。期望这样的处理能力是鲁棒的和耐系统故障的,即容错的。这些能力对于从大规模基于因特网的数据处理到专用网络和通信系统(例如,企业内部的“内联网”等)范围的所有类型和大小的计算机网络是有用的。

发明内容

[0005] 在一个方面,通常,计算机实现的方法包括在正在初始化进程时存储与所述进程的初始状态有关的信息,其中所述进程的执行包括执行至少一个执行阶段,并在完成所述执行阶段的执行时,存储代表所述执行阶段的结束状态的信息;响应于预定事件中止所述进程的执行;以及从所保存的初始和结束状态中的一个恢复所述进程的执行,而不需要关闭所述进程。

[0006] 多个方面可以包括下列内容中的一个或多个。

[0007] 预定事件可以代表到外部设备的连接的丢失。预定事件代表外部设备的错误。当已重建到外部设备的连接时,可以恢复进程的执行。当已清除外部设备的错误时,可以恢复进程的执行。可以从先于预定事件发生的执行阶段存储的结束状态恢复进程的执行。如果预定事件发生在基本上紧跟进程的起动之后的执行阶段期间,则可以从初始状态恢复进程的执行。执行阶段的执行可以包括对接收的数据流执行一个或多个处理动作以便产生与所述执行阶段对应的输出数据。

[0008] 所述方法还可以包括存储与一个或多个执行阶段对应的输出数据,以及当恢复进程的执行时再现输出数据。所述进程可以是进程级的部分,并且通过数据路径与进程级中的第二、不同的进程通信。所述方法还可以包括传递检查点命令消息通过进程级中的每个进程;以及在每个进程处,在接收到所述检查点命令消息时,保存与初始或结束状态有关的

新的信息,其中所述保存包括暂停进程的操作以及将新的信息保存到存储区域。所述计算机实现的方法还可以包括以新的初始或结束状态覆写旧的保存的初始或结束状态。

[0009] 每个进程可以与数据队列通信用于为进程接收数据和将所述数据排队。所述方法还可以包括响应于检测触发事件生成检查点命令消息。触发事件可以包括关于网络事件的信息。所述方法还可以包括周期性地生成检查点命令消息。

[0010] 所述方法还可以包括响应于在正在处理的引入数据记录内或从所述引入数据记录导出的所选的数据值的出现,而生成检查点命令消息。所述方法还可以包括响应于预定事件生成中止命令消息;传递所述中止命令消息通过进程级中的每个进程;以及在每个进程处接收所述中止命令消息时,中止进程的执行并向下一进程传递所述中止命令消息。所述方法还可以包括部分基于在恢复处理消息中包含的信息从保存的初始或结束状态中的所选的一个恢复进程的执行。所述方法还可以包括在基本上紧跟初始化进程并已保存与进程的初始状态有关的信息之后的第一执行阶段期间接收中止命令消息;以及从保存的初始状态恢复进程的执行,而不需要关闭和重启进程。

[0011] 在另一方面,通常,存储计算机程序的计算机可读存储介质包括使得计算系统执行下列操作的指令:在正在初始化进程时存储与所述进程的初始状态有关的信息,其中进程的执行包括执行至少一个执行阶段,并在完成所述执行阶段的执行时存储代表所述执行阶段的结束状态的信息;响应于预定事件中止进程的执行;以及从保存的初始和结束状态中的一个恢复进程的执行,而不需要关闭进程。

[0012] 在另一方面,通常,计算系统包括输入设备或端口,其被配置为在正在初始化进程时接收和存储与所述进程的初始状态有关的信息,其中进程的执行包括执行至少一个执行阶段,并在完成所述执行阶段的执行时存储代表执行阶段的结束状态的信息;以及至少一个处理器,其被配置为:响应于预定事件中止进程的执行,以及从保存的初始和结束状态中的一个恢复进程的执行,而不需要关闭进程。

[0013] 在另一方面,计算系统包括用于在正在初始化进程时存储与所述进程的初始状态有关的信息的装置,其中进程的执行包括执行至少一个执行阶段,并在完成所述执行阶段的执行时存储代表所述执行阶段的结束状态的信息;用于控制所述进程的执行的装置,所述控制包括响应于预定事件使得中止进程的执行,以及从保存的初始和结束状态中的一个恢复进程的执行,而不需要关闭进程。

[0014] 多个方面可以包括下列优点中的一个或多个。

[0015] 可以在不同的执行阶段中执行多进程处理系统中的进程。在系统故障的事件中,从最近完成检测点终止和重启处理系统可能消耗过度量的处理时间和资源。在处理系统已响应于异常情况终止其活动之后,可能需要由具有这样的系统的丰富经验的信息技术专家手动地重新初始化所述处理系统。这可能导致相当长的系统停机时间。在某些示例中,还可能需要单独的进程来检测系统故障并通知专家。像这样,为了提高效率并减少处理资源消耗,在重建到进程的故障连接时,可以从处理系统内的进程的最近记录的检查点执行所述处理系统内的进程,而非重启整个系统。在一个实现方式中,可以通知系统中的单个进程以暂停处理直到重建故障连接为止,而非终止和重启整个系统。

[0016] 根据下列描述和根据权利要求,本发明的其他特征和优点将变得清楚。

附图说明

- [0017] 图1是多进程数据处理系统的框图。
[0018] 图2和图3图示示例多进程数据处理系统。
[0019] 图4是图示示例检查点操作进程的流程图。
[0020] 图5和图6是示例复原机制的流程图。

具体实施方式

[0021] 参考图1, 数据处理系统100提供以流水线方式排列的多个进程用于处理数据。在示例系统100内, 从数据源102(例如, 在用作Web服务器的服务器104上执行的应用)接收数据, 并向在计算机系统108上执行的或以分布的方式(例如, 具有两个或多个联网的计算机终端)执行的多进程数据处理模块106传送所述数据。数据处理模块106监视、控制并执行系统100的数据处理方面。为了提供这样的处理, 数据处理模块106包括能够存储要由一个或多个进程116、118处理的数据的一个或多个队列110、112、114。在该实例中, 如所示, 在初始数据队列110中存储从数据源102接收的数据, 并向初始进程116周期性地提供所述数据。进程116处理数据(例如, 变换、过滤、确认内容等)并向一个或多个下游数据队列112、112'提供处理后的数据。可以向随后的进程118、118'提供来自队列112、112'的数据, 并在转而向其他下流数据队列114、114'递送结果之前执行其他(或类似)处理。图示的数据处理模块106的队列和进程布局是可以使用的许多可能的处理方案中的一个。例如, 数据处理模块106可以包括可以位于示出的进程的上游、下游、或独立于示出的进程的另外的进程(例如, 用于并行或串行执行)。在某些示例中, 可以向例如关系数据库管理系统(RDBMS)的目标应用120(或多个应用)输出来自最后一组队列(例如, 队列114和114')的数据。

[0022] 在数据处理模块106中包括的进程可以与外部设备和/或其他处理系统(例如, 计算机系统122)通信。例如, 进程可以与向来自其他系统的进程提供消息的Java消息服务(JMS)队列通信。在某些情况下, 数据处理模块106中的进程可以与一个或多个数据库通信(例如, 位于外部系统122中)。例如, 模块106可以基于从在一个或多个自动取款机(ATM)处的对应的客户会话接收的信息执行对银行数据库中的客户金融账户的更新。

[0023] 通过示例的方式, 图2图示具有用于提供数据以用于在中央位置的处理的远程执行处理模块202(由ATM204执行)的处理系统200。在图示的示例中, 例如读取ATM进程206的初始进程能够从ATM接收客户账户数据(例如与交易相关联), 并向验证账户进程208传递所述数据用于认证账户详情。在该实例中, 验证账户进程208可以对照个人识别号(PIN)数据库210检验由客户输入的PIN。一旦已认证客户的身份, 就可以向下游的检查余额进程212传送进一步的数据记录, 所述检查余额进程212可以与第二、不同的数据库214通信例如用于检查所识别的客户账户的余额。在完成进一步的交易之后, 可以向下游的更新余额进程216发送另外的数据, 所述更新余额进程216可以与第三数据库218通信例如用于更新与客户账户相关联的余额信息。创建回答进程220可以准备可以向输出显示进程222提供的交易的输出概要(例如, 用于在ATM204上向用户显示)。对于系统级监视(例如, 系统质量保证)或其它应用, 数据库210、214和218可以与主数据服务器224通信。在某些实现方式中, 可以例如由独立的计算机系统226执行所述数据库。

[0024] 可以在不同的执行阶段执行这样的多进程处理系统中的进程。进程的执行可以包括在不同的执行阶段中进程内的一个或多个任务的执行。通过将执行分段为这样的阶段，可以例如由数据处理中的多个逻辑端点或断点终止不同的执行阶段。每个执行阶段可以具有一个或多个处理活动来实现该执行阶段的目标。作为示例，可以以一个或多个方式在不同的执行阶段中执行验证账户进程208。例如，作为第一执行阶段，验证账户进程208最初可以从客户接收个人识别号(PIN)信息。在从客户接收PIN信息时的各种处理活动可以包括例如在ATM显示器上显示提示，以及运行例程来检验PIN信息的数据条目。在下一执行阶段，进程208可以建立到数据库210的连接，并使用PIN信息作为识别客户的记录的密钥。一旦进程208已完成其与客户的记录的交易，就可以终止到数据库210的连接。在最后的执行阶段，进程208可以基于上述交易生成结果。像这样，这些执行阶段中的每一个包括不同的逻辑端点(或处理断点)，在所述逻辑端点处可以临时暂停和/或恢复验证账户进程208。

[0025] 在某些情况下，可能发生趋向于影响正常系统操作过程的一个或多个事件。这样的事件可以是由在处理系统中包括的硬件或软件模块引起的异常或错误。例如，硬件异常或错误可以包括来自一个或多个硬件单元的重置、中断或其他信号。还可以由算术逻辑单元生成异常用于例如除以零、溢出、指令解码错误、未定义的指令等的数值错误。基于一个或多个这样的事件的发生，可能需要临时停止数据库中的一个或多个数据库的操作用于进行校正动作(例如，维护、切换到次级系统等)。

[0026] 可能需要停止操作和矫正动作的其他事件可以包括检测数据库210-224中的一个或多个数据库的故障。这样的故障可能由于各种原因发生。例如，存储器分配中可能存在错误或向存储器空间写入时可能存在冲突。在例如当进程试图从用尽的账户提取资金时在底层的数据操作中也可能存在错误。除了存在临时故障的事件以外，还可能通过操作者干预触发的事件。在某些实现方式中，操作者可以校正引起故障的情况，或者系统可以及时校正所述情况。所述事件的示例在没有限制的情况下可以包括连接到网络的一个或多个设备的故障、用于维护的一个或多个设备或软件服务的关闭、设备或软件服务的故障或切换、例如存储空间资源的耗尽、处理单元的负载、一个或多个软件服务的超时。

[0027] 为了检测并解决这样的事件，数据处理系统可以使用通常被称为检查点操作技术的一个或多个技术来在故障事件中或当使得系统离线用于维护或切换时保证最小系统停机时间。检查点操作技术总地涉及将进程的当前状态的详情存储为检查点记录，使得所述进程可以使用存储的信息来稍后从该状态重启。例如，验证账户进程204可以在完成每个执行阶段时(先于启动下一执行阶段或其他处理的执行)在检查点记录中保存其当前状态。

[0028] 检查点记录可以包括各种类型的信息，例如进程值、关于成功处理的记录的信息、以及与进程的当前执行阶段有关的其他详情。例如，检查点记录可以包括关于数据队列(例如，图1的数据队列)中的当前位置的信息，从所述位置正在处理数据。像这样，在停止操作之后，可以从该队列位置恢复处理。沿这些线，在从系统故障复原之后，进程能够从存储的中间检查点状态而非从初始状态重启。

[0029] 作为示例，如果PIN数据库210故障，则验证账户进程208可以产生异常来使得终止整个处理系统。在重启时，处理系统中的进程(或进程中的部分)可以从它们的最近检查点状态继续处理。在该示例中，由于在客户提供他的PIN信息之后的时间点发生故障和重启，因此PIN信息被回复到进程并且不需要从客户重新收集。像这样，可能不存在再次提示客户

提供他的PIN信息的需要。

[0030] 在系统故障的事件中,从最近完成检测点终止和重启处理系统可能消耗过度量的处理时间和资源。在处理系统已响应于异常情况终止其活动之后,处理系统可能需要由具有这样的系统的丰富经验的信息技术专家手动重新初始化。这可能导致相当长的系统停机时间。在某些示例中,单独的进程可能需要被设计为检测系统故障并通知专家。在名称为“Continuous Flow Checkpointing Data Processing”的美国专利No.6,584,581中、在名称为“Overpartitioning System and method for increasing checkpoints in component-based parallel applications”的美国专利No.5,819,021中、以及在名称为“Methods and Systems for Reconstructing the State of a Computation”的美国专利No.5,712,971中描述了检查点操作系统的示例,上述专利中的每一个的内容整体合并于此。

[0031] 为了提高效率并降低处理资源消耗,在重建到进程的故障连接时从处理系统内的进程的最近记录的检查点执行所述进程而非重启整个系统。在一个实现方式中,可以通知系统中的单个进程以暂停处理直到重建故障连接为止,而非终止并重启整个系统。

[0032] 图3示出多进程系统300的框图,所述多进程系统300包括数据源进程302、进程304a-n、数据接收进程306,以及与其他进程(进程302、304a-n)中的每一个通信的容错管理器308。在某些实现方式中,容错管理器308可以作为多进程系统300内的另一进程执行。在某些情况下,容错管理器302可以是在单独的计算机系统(未示出)上运行的或在专用处理器中实现的应用,所述专用处理器是例如在例如名称为“Continuous Flow Checkpointing Data Processing”的美国专利No.6,584,581中描述的检测点处理器,所述专利的内容整体合并于此。

[0033] 可以实现一个或多个技术来建立进程302-306和容错管理器308之间的通信。例如,单独的异常信道310a-n可以被用于传送关于可能在进程302-306中发生的异常情况的信息。信道310a-n可以是有线网络系统、无线网络系统、或有线或无线网络系统的组合的部分。可以由进程302-306使用信道310a-n来向容错管理器308传送关于进程302-306的错误信息。例如,如果与进程304a通信的外部设备故障,则进程304a可以立即产生错误标志并经由异常信道310b向容错管理器308传送所述错误。

[0034] 除了异常信道310a-n以外,容错管理器308可以通过对应的通信信道312a-e向进程302-306发送命令消息(例如,检查点命令消息)。通信信道312a-e被布置为依序从容错管理器308向进程302-306中的每一个传送命令消息。例如,可以通过信道312b-d首先向数据源进程302传送来自容错管理器308的消息,然后顺次传递所述消息通过进程304a-n中的每一个和数据接收进程306。数据接收进程306可以使用信道312e向容错管理器308传送命令消息。

[0035] 为了存储与进程302、304a-n、306中的每一个相关联的相应的检查点数据,存储区域(例如,存储器)可以被分配给每个进程。每个进程在不同的执行阶段的结束周期性地暂停其当前操作,并且在相关联的存储区域中存储其检查点数据。例如,数据源进程302可以在处理中的不同执行阶段的结束(例如引入数据流中的逻辑断点)周期性地暂停其当前操作,并在存储区域312中存储检查点信息。以这种方式,当执行进程302、304a-n、306中的每一个时,对应的存储区域312、314a-n、316周期性地保存检查点数据。可以使用各种类型的

存储技术来实现存储区域312-316,例如在诸如硬盘驱动器的磁性介质的非易失性存储器上。在某些示例中,可以由两个或更多进程共享单个存储区域。检查点数据可以包括关于当前状态的信息和/或与进程302-306相关联的数据以便允许在稍后的时间重新构建那些状态。

[0036] 在某些示例中,数据存储区域312-316还可以存储与进程相关联的数据队列(例如,图1的队列110、112和114)。例如,存储区域316可以包括用于从进程304n接收处理后的数据的队列,并且从所述队列可以输出或公布(例如,打印、或显示)数据。

[0037] 容错管理器308通过生成并通过通信信道312a-e向每个进程302-306依序传递检查点命令消息来管理检查点操作。检查点命令消息传递通过每个进程302-306,使得所述进程可以在接收到消息时对其当前状态执行检查点操作。像这样,检查点命令消息行进到数据源进程302,然后在被返回到容错管理器308之前依序通过每个进程304a-n和数据接收进程306。可以以固定的时间间隔自动启动该检查点操作。例如,容错管理器308可以以例如每五分钟的预定周期率启动检查点命令消息。所述周期率可以被设定为默认值,或由用户调整。

[0038] 可以实现一个或多个技术用于启动用于为在系统中包括的进程存储检查点的操作。例如,一个或多个外部触发器可以启动用于存储检查点信息的操作。在一个实例中,网络消息可以通知容错管理器308即将到来的网络关闭,从而触发检查点操作。在某些实施方式中,可以响应于正在处理的数据记录内或从所述数据记录导出的值触发检查点操作。例如,处理后的数据记录可以包括可以被视为检查点操作可发生的逻辑点的时间戳或断点值。

[0039] 随着在数据正在被系统处理的时段期间存储检查点信息,可以先于处理数据存储信息。在一个实施方式中,可以在初次初始化多进程系统300时,例如在起动期间,触发初始检查点操作。容错管理器308可以传递初始检查点命令消息通过进程302-306中的每一个。在图3中示出的示例中,首先向数据源进程302传送初始检查点消息。数据源进程302立即进行检查点操作,例如将代表所述数据源进程302的初始状态的数据存储到相关联的数据存储空间312,并向下游的下一进程304a传递初始检查点消息。该初始检查点状态被称为检查点状态零。

[0040] 类似地,以串行方式,进程304-306中的每一个可以对应地将其初始状态和相关联的数据值存储到适当的存储区域作为检查点状态零。在示例中,初始状态和相关联的数据值可以包括全局变量的初始值、参考数据信息、和包括计数器的初始值的审核变量。

[0041] 在进程302-306中的每一个已存储它们的初始状态之后,初始检查点命令消息通过信道312e返回到容错管理器308。基于在其通过进程302、304a-n、306的往返之后返回到容错管理器308的消息,警告容错管理器进程302-306已完成检查点状态零。在某些实施方式中,虽然下游进程正在保存它们的当前状态,但是源和其他上游进程可以继续接收数据并执行其他功能,而不需要等待所有进程保存它们的状态。

[0042] 类似地,对于进程302-306的每个不同的执行阶段可以执行另外的检查点操作。像这样,随着存储代表初始检查点信息的数据,容错管理器308可以启动例如代表与随后的检查点周期(例如,检查点状态1、2、3、……、n)相关联的信息的另外的信息的存储。为了启动随后的检查点信息的存储,可以使用例如传播进一步的检查点命令消息通过进程302、

304a-n、306的技术。在接收检查点命令消息时,进程304a可以完成任何正在进行的任务或暂停任何未完成的任务。在某些示例中,进程304a可以删除在数据存储314中存储的先前创建的检查点记录,并重新请求存储空间。然后进程304a可以对于其当前状态和相关联的数据创建的新的检查点记录。在某些情况下,早先的检查点记录永久地存储在存储器中,并没有被新的检查点记录覆写。在美国专利No.6,584,581中提供了在检查点操作记录中存储的信息的另外的示例,所述专利的内容整体合并于此。

[0043] 在某些实施方式中,进程304a可以与外部数据库318(在计算机系统320上执行)进一步通信。有时,到数据库318的连接可能故障,或可能使得数据库318离线用于维护。所述故障可以是执行数据库318的计算机系统320的硬件故障。在这样的情况下,进程304a可以在异常信道310a上产生错误标志来通知容错管理器308连接丢失。

[0044] 在接收到错误的通知时,容错管理器308可以生成并传播中止命令消息通过进程302-306。通常首先通过信道312a向数据源进程302传送中止命令消息,然后通过信道312b-d通过进程302-306中的每一个,最后通过信道312e回到容错管理器308。在接收到中止命令消息时,进程302-306中的每一个以相对较小的延迟(如果存在)中止其当前活动,并刷新/丢弃自最近检查点状态以来可能已处理的任何未完成的任务或记录。在进程已中止活动之后,其可以向下一个下游进程传递中止命令消息。以这样的方式,中止命令消息在返回到容错管理器308之前一直传播到接收进程306。容错管理器308等待直到其从接收进程306接收到中止命令消息为止,这确保所有进程302-306已中止当前处理任务(例如,处于静止状态中)。

[0045] 在数据库318由于计算机系统320中的硬件故障而发生故障时,进程302-306被指示为中止它们的处理。在某些实施方式中,在系统已完全中止其处理之后,进程302可以等待反映校正故障所需的平均时间量的时间的可指定的时间量,并再次从最近的保存的检查点状态开始处理。在某些实施方式中,进程304a可以周期性地轮询数据库318来检查其状况(即,检查数据库318是否是操作的)。在某些示例中,计算机系统320可以被配置为当数据库318被回复为操作状态时,自动通知进程304a。当重建与数据库318的连接时,处理系统300可以再次从最近保存的检查点状态开始处理。

[0046] 在这方面,进程304a通知容错管理器308已重建连接。容错管理器308对于进程302-306中的每一个确定最近成功完成的检查点状态,并向进程302-306中的每一个发送恢复处理消息。正如其他命令消息,恢复处理消息经由通信信道312a-e依序传播通过进程302-306中的每一个。恢复处理消息指定检查点状态,从所述检查点状态进程302-306将恢复处理。

[0047] 在某些实例中,容错管理器308可以在当前正在执行前面的检查点操作的同时启动另外的检查点操作。例如,在进程304n正在处理任意检查点状态(例如,与检查点命令消息N对应的检查点状态N)时,容错管理器308可以通过生成并向源进程302传输随后的检查点命令消息N+1来开始随后的检查点状态(例如,检查点状态N+1)。沿这些线,当检查点命令消息N仍然正在行进通过进程302-306时,生成新的随后的检查点命令消息N+1并传递所述检查点命令消息N+1通过进程302-306是可能的。以这样的方式,容错管理器308可以引起更频繁的进程状态的检查点操作,而不必等待直到完成先前的检查点操作状态为止。

[0048] 在某些情况下,在一个或多个检查点命令消息传输通过进程302、304a-n、306时,

系统故障可能发生。例如,考虑容错管理器308通过生成检查点命令消息N已启动检查点状态N的情况。在检查点命令消息N正在被进程302-306处理时,进程中的一个(例如,进程304a)和外部系统(例如,数据库312)之间的连接可能故障。在被警告所述情况时,容错管理器308可以通过传递中止命令消息通过进程302-306来响应。中止命令消息可以到达仍然正在处理检查点状态N(例如,存储与检查点N相关联的检查点信息)的进程(例如,进程304n)。基于中止命令的接收,进程304n可以采取一个或多个动作。例如,进程304n可以完成检查点状态N并中止所有进一步的处理。在另一情况下,进程304n可以立即丢弃与自先前的检查点状态N-1以来的当前和随后的状态相关联的结果,并中止进一步的处理。作为结果,当系统300实现静止时,进程302-306中的每一个可以处于不同的检查点状态。例如,进程304n上游的所有进程(例如,数据接收进程306)可能已完成检查点状态N,而进程304n下游的所有进程(例如,进程304a和数据源进程302)可能仅已完成检查点状态N-1。

[0049] 当系统300准备好恢复处理时,容错管理器308经由通信信道312a-e向每个进程传输一个或多个恢复处理消息。恢复处理消息向进程指示最早的、完全进行(commit)(或完成)的检查点状态(例如,检查点状态N-1),所述进程将从所述检查点状态执行。在示例中,可能已完成检查点状态N的进程可以仅将从检查点状态N-1到检查点状态N再现结果。以该方式,进程302-306可以避免重复它们的较早的努力。在示例中,从检查点状态N-1到检查点状态N重演结果涉及再现在两个检查点状态之间可能已发生的较早的处理动作的结果。

[0050] 在示例中,系统故障可以在起动之后基本上立即发生。在这样的情况下,进程302-306中的许多可能仅完成了检查点状态零。这些进程302-306可以基于在对应的检查点记录中存储的初始化数据和起动值从检查点状态零恢复处理。

[0051] 图4是描绘在多线程系统内的进程(例如,图3的进程302)的示例执行的流程图。在起动时,进程立即将其初始状态存储到数据存储中作为检查点状态零(步骤402)。然后进程可以在不同的执行阶段(例如,执行阶段1、2、……、N-1)中执行。在每个执行阶段结束时,进程可以将其结束状态保存到数据存储中,例如,结束状态可以被保存为检查点状态。例如,在第一执行阶段之后,进程可以将第一执行阶段的结束状态保存为检查点状态1(状态404)。类似地,在随后的执行阶段之后,进程可以将执行阶段的结束状态保存为检查点状态2、……、N-1和N(步骤406-410)。

[0052] 图5是描绘在外部系统故障或使得外部系统离线用于维护的事件中执行的示例步骤的流程图。例如,外部系统可以是与处理系统中的进程通信的数据库(例如,图3的数据库318)。当使得外部系统离线用于维护或外部系统故障时,可以由例如与故障的外部系统通信的进程使错误标志出现(步骤502)。响应于所述错误标志可以生成中止命令消息并传播所述中止命令消息通过进程(步骤504)。当进程接收到中止命令消息时,中止进程中的每一个的当前活动(步骤506)。此外,进程可能丢弃自最近的检查点状态以来执行的任何交易。然后中止进一步的动作直到重建到外部系统的故障的连接为止(步骤508)。当重建所述连接时,向进程中的每一个传输恢复处理消息(步骤510)。恢复处理消息指示检查点状态,从所述检测点状态进程将恢复处理。像这样,进程中的每一个从与它们相关联的存储区域检索关于检查点状态的相关信息(步骤512)。

[0053] 图6是描绘在执行进程的同时在存储和从检查点恢复时执行的示例步骤的流程图。例如,在初始化进程时,与进程的初始状态有关的信息被存储在相关联的存储区域中

(步骤602)。然后在不同的执行阶段中执行进程。像这样,在每个执行阶段结束时,进程存储代表执行阶段的结束状态的信息(步骤604)。当预定事件发生时,例如,丢失到外部设备的连接(步骤608),中止进程的执行(步骤606)。同时,进程检查以便查看触发中止的事件是否已清除(例如,重建到外部设备的连接)。在这期间,不关闭进程,但中止进程直到事件被认为清除为止。进程的执行从最近保存的初始或结束状态恢复(步骤610)。

[0054] 可以使用用于在计算机上执行的软件来实现上面描述的技术。例如,所述软件在一个或多个编程或可编程计算机系统(其可以是例如分布式、客户端/服务器、或网格的各种架构)上执行的一个或多个计算机程序中形成过程,每个所述计算机系统包括至少一个处理器,至少一个数据存储系统(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备或端口以及至少一个输出设备或端口。所述软件可以形成例如提供与数据流图的设计和配置有关的其他服务的更大的程序中的一个或多个模块。图的节点和元素可以被实现为在计算机可读介质中存储的数据结构,或符合在数据存储库中存储的数据模型的其他有组织的数据。

[0055] 可以在例如CD-ROM的可以由通用或专用可编程计算机读取的存储介质上提供所述软件,或可以经由到执行所述软件的计算机的网络的通信介质递送(被编码在传播信号中)所述软件。可以在专用计算机上、或使用例如协处理器的专用硬件执行所有功能。可以由不同的计算机执行由所述软件指定的计算的不同部分的分布式的方式实现所述软件。每个这样的计算机程序被优选地存储在或下载到可以由通用或专用可编程计算机读取的存储介质或设备(例如,固态存储器或介质、或磁性或光学介质),用于当由计算机系统读取存储介质或设备以便执行在这里描述的过程时,配置和操作计算机。本发明系统还可以被认为被实现为使用计算机程序配置的计算机可读存储介质,其中这样配置存储介质使得得以具体和预定方式操作计算机系统以便执行在这里描述的功能。

[0056] 已描述了本发明的许多实施例。然而,将理解可以进行各种修改而不背离本发明的精神和范围。例如,上面描述的步骤中的某些可以是顺序无关的,从而可以以不同于描述的顺序的顺序执行。

[0057] 应理解上述描述意图图示而并不限制本发明的范围,本发明的范围由所附权利要求的范围定义。例如,可以以不同的顺序执行上面描述的许多功能步骤,而不实质上影响整体处理。其他实施例在下列权利要求的范围内。

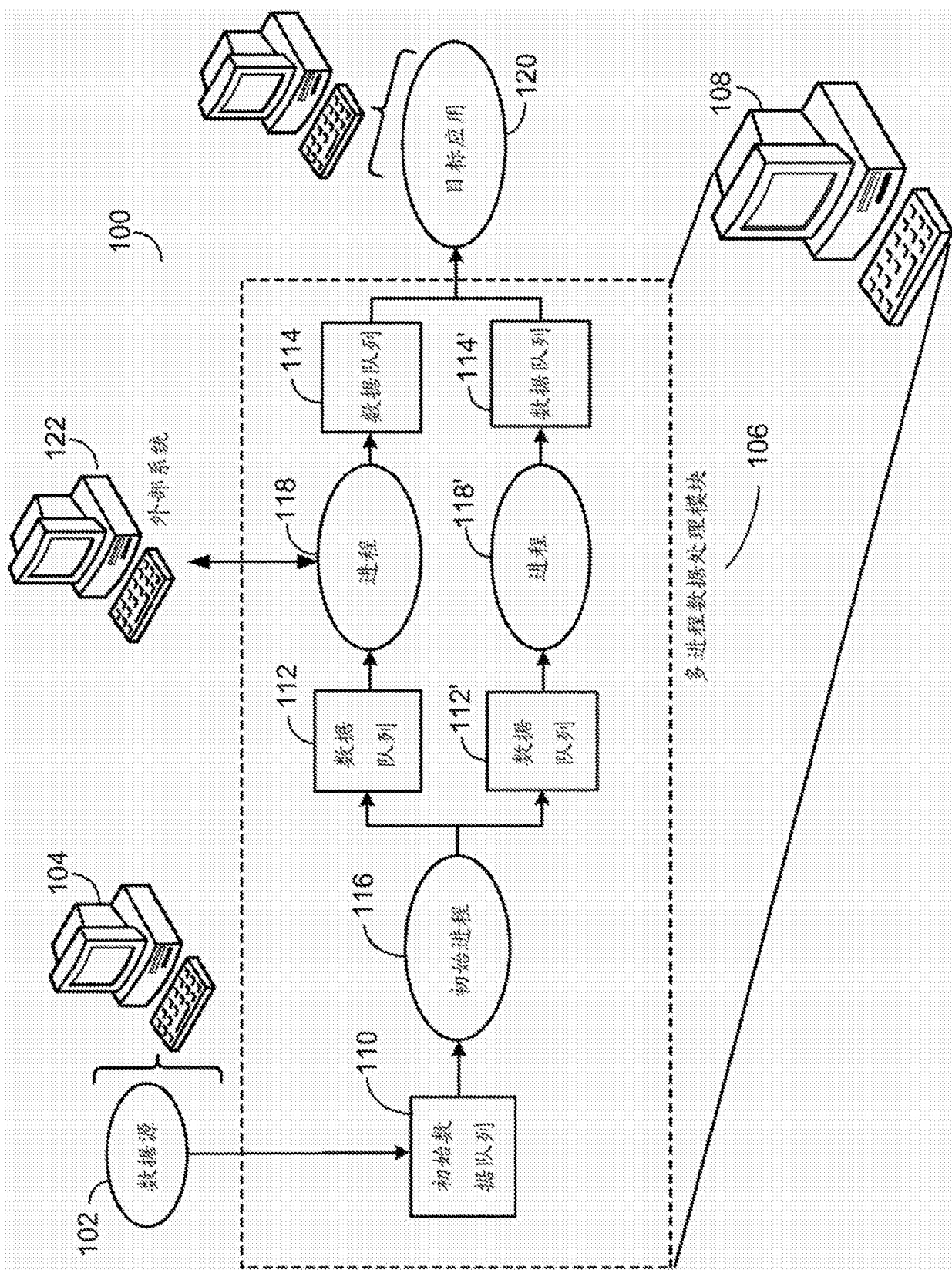


图1

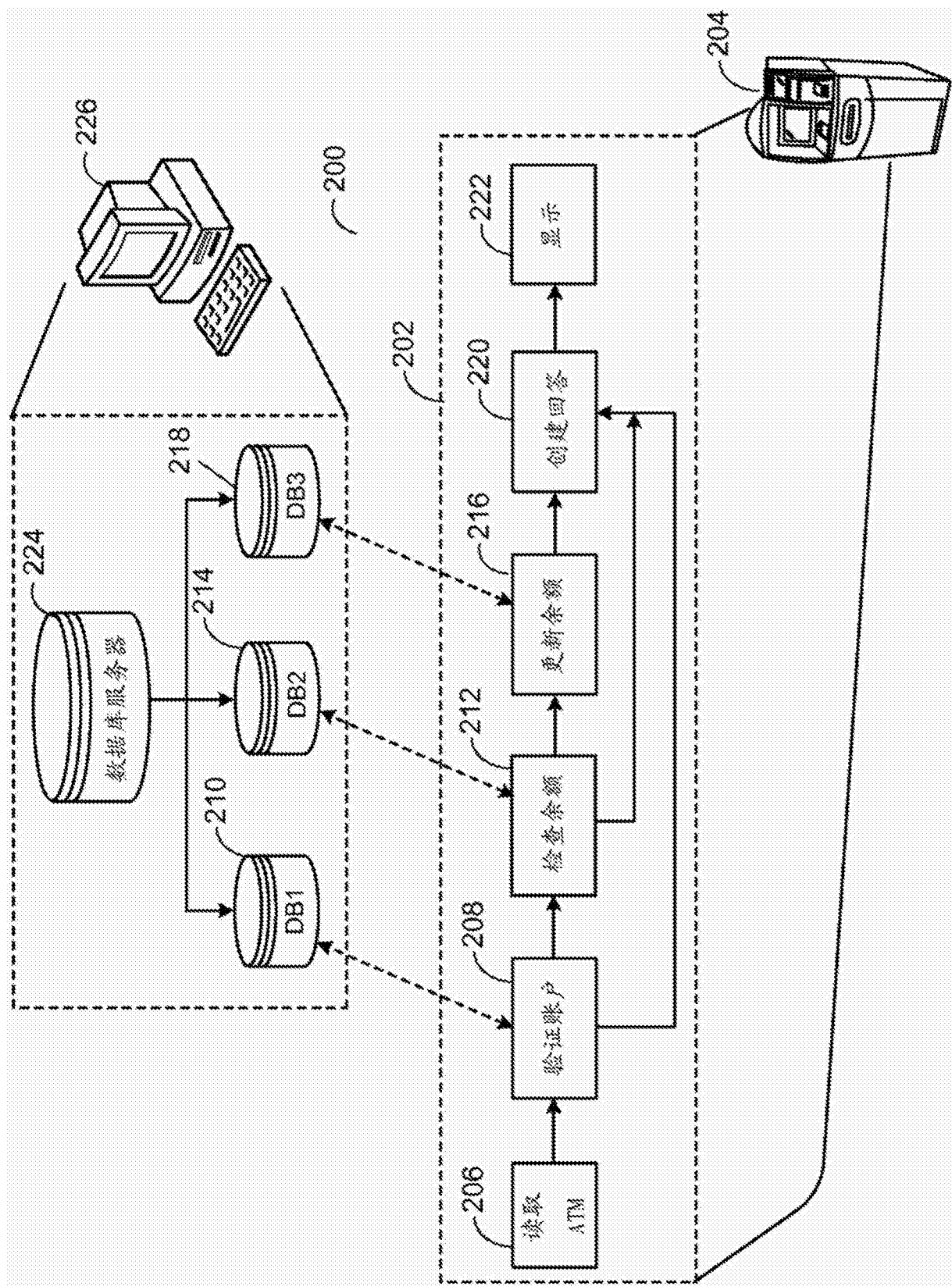


图2

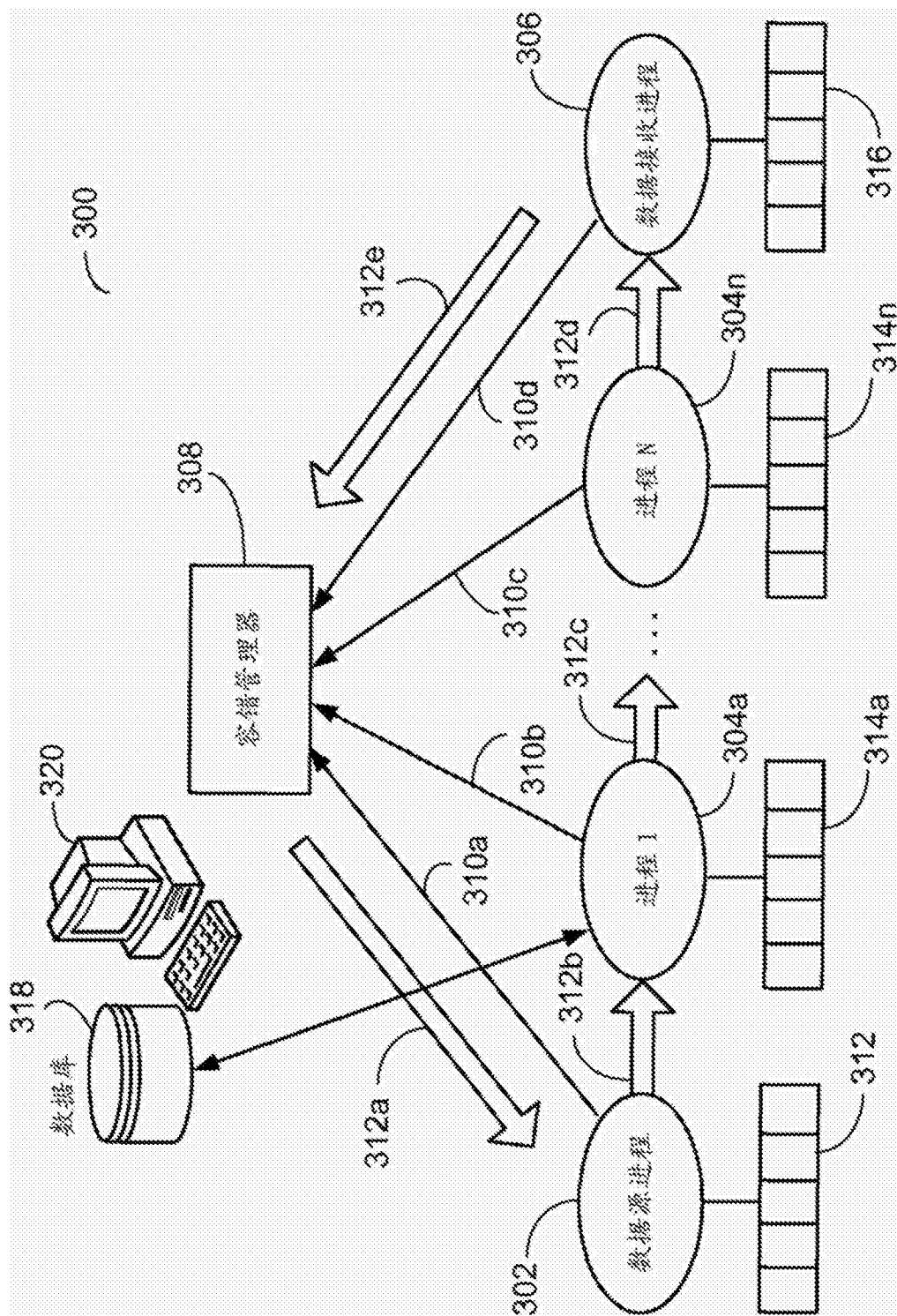


图3

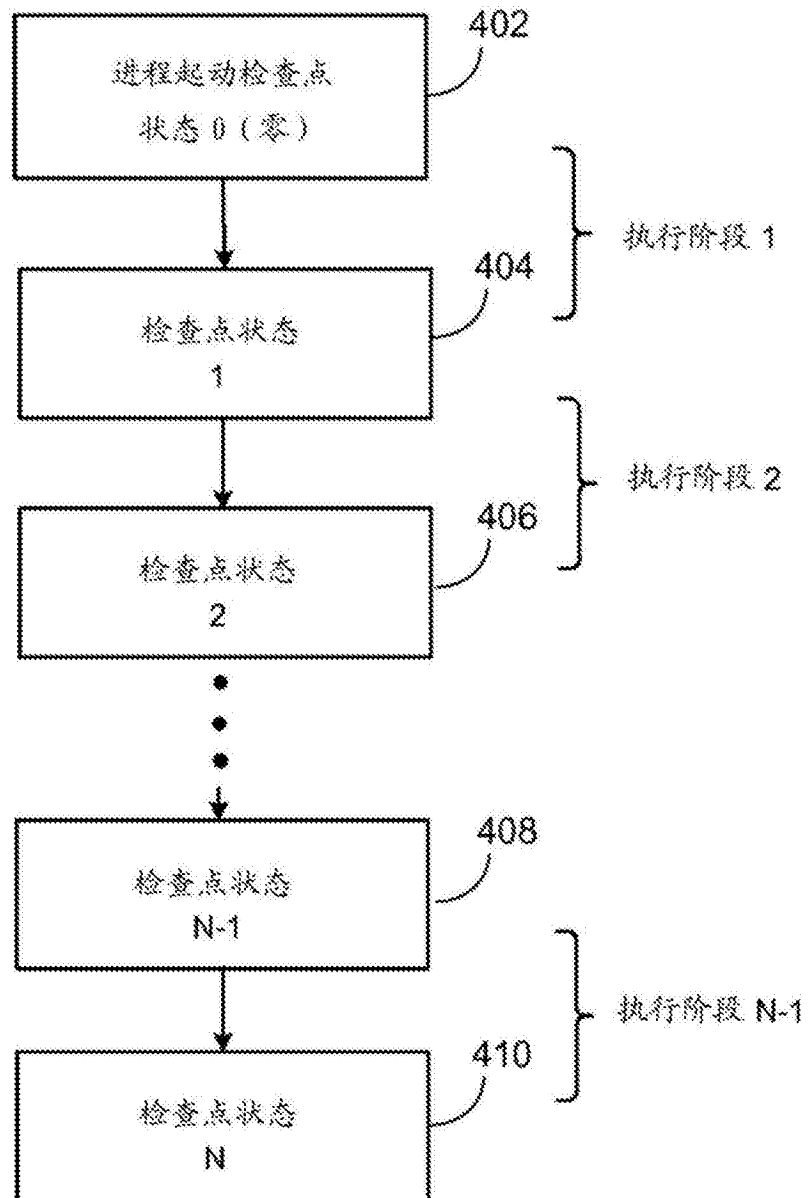


图4

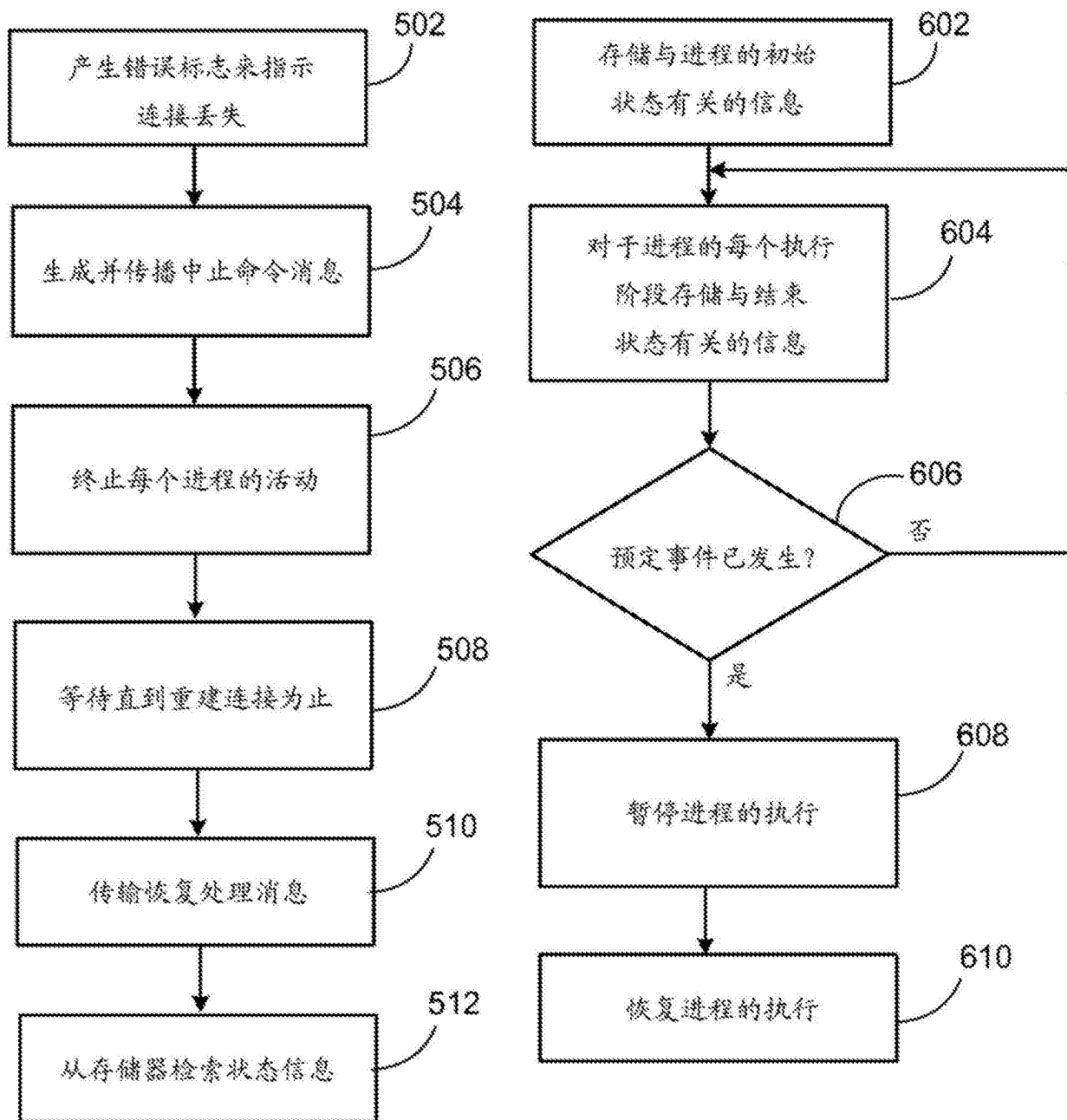


图6

图5