(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106796540 B
(45)授权公告日 2021.01.05

(21)申请号 201580052408.6
(22)申请日 2015.07.20
(65)同一申请的已公布的文献号
申请公布号 CN 106796540 A
(43)申请公布日 2017.05.31
(30)优先权数据
14/445,369 2014.07.29 US
(85)PCT国际申请进入国家阶段日
2017.03.28
(86)PCT国际申请的申请数据
PCT/US2015/041121 2015.07.20
(87)PCT国际申请的公布数据
WO2016/018663 EN 2016.02.04
(73)专利权人 沙特阿拉伯石油公司
地址 沙特阿拉伯达兰市
(72)发明人 哈兰德·S·AL-瓦哈比
(74)专利代理机构 中科专利商标代理有限责任公司 11021
代理人 杨姗
(51)Int.Cl.
G06F 11/14(2006.01)
G06F 11/20(2006.01)
(56)对比文件
US 2002087913 A1,2002.07.04
US 5539883 A,1996.07.23
CN 103197982 A,2013.07.10
US 2010088494 A1,2010.04.08
审查员 黄旭光

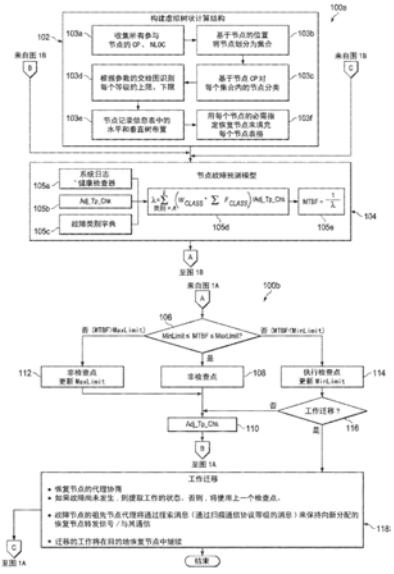
权利要求书3页 说明书15页 附图13页

(54)发明名称

用于分布式计算的主动故障恢复模型

(57)摘要

本公开大体上描述了用于提供用于分布式计算的主动故障恢复模型的方法和系统,包括计算机实现的方法、计算机程序产品和计算机系统。一种计算机实现的方法包括:构建多个计算节点的虚拟树状计算结构;针对虚拟树状计算结构的每个计算节点,由硬件处理器执行节点故障预测模型,以计算与所述计算节点相关联的平均故障间隔时间(MTBF);基于计算出的MTBF与最大和最小阈值之间的比较来确定是否执行所述计算节点的检查点;将过程从所述计算节点迁移至作为恢复节点的不同的计算节点;以及继续在不同计算节点上执行过程。



CN 106796540 B

1.一种计算机实现的方法,包括:

构建多个计算节点的虚拟树状计算结构,所述多个计算节点被映射为以父/子类型关系进行通信,其中针对所述计算节点中的每一个计算节点,一个或多个直接子代被指定为默认恢复节点,并且另一节点被指定为检查点节点;

针对所述虚拟树状计算结构的每个计算节点,由硬件处理器执行节点故障预测模型以计算与所述计算节点相关联的平均故障间隔时间"MTBF";

基于计算出的MTBF与最大和最小阈值之间的比较来确定执行第一计算节点的检查点;

将过程从所述第一计算节点迁移至针对所述第一计算节点指定的默认恢复节点;以及

在针对所述第一计算节点指定的所述默认恢复节点上继续执行所述过程。

2.根据权利要求1所述的方法,还包括:

针对每个计算节点收集至少计算能力参数和节点位置参数;

基于所述计算节点的节点位置参数将所述计算节点划分为集合;以及

基于所述计算能力参数对每个集合内的节点进行排序。

3.根据权利要求2所述的方法,还包括:

识别上限和下限以确定经排序的计算节点的等级;

基于所述计算能力参数以及所述上限和所述下限将每个集合内的计算节点排序为水平等级;

将所述水平等级布置和垂直布置记录到与每个计算节点相关联的相应节点记录信息表中,其中所述垂直布置是至少基于每个计算节点的节点位置参数确定的;以及

用所指定的默认恢复节点填充每个节点记录信息表。

4.根据权利要求3所述的方法,其中,所述上限和所述下限是根据针对每个计算节点所收集的计算能力参数和节点位置参数的交绘图确定的。

5.根据权利要求1所述的方法,其中,所述MTBF是至少基于网络或数据存储故障计算的。

6.根据权利要求1所述的方法,还包括:

当所述第一计算节点的MTBF小于所述最小阈值时创建检查点;以及

将与所述第一计算节点相关联的所述最小阈值更新为等于所述MTBF。

7.根据权利要求6所述的方法,还包括:

确定所述第一计算节点的故障已经发生;以及

使用针对所述第一计算节点采取的最后一个检查点作为过程状态。

8.一种非瞬时性计算机可读介质,存储计算机可读指令,所述指令能够由计算机执行以执行包括以下各项的操作:

构建多个计算节点的虚拟树状计算结构,所述多个计算节点被映射为以父/子类型关系进行通信,其中针对所述计算节点中的每一个计算节点,一个或多个直接子代被指定为默认恢复节点,并且另一节点被指定为检查点节点;

针对所述虚拟树状计算结构的每个计算节点,执行节点故障预测模型以计算与所述计算节点相关联的平均故障间隔时间"MTBF";

基于计算出的MTBF与最大和最小阈值之间的比较来确定执行第一计算节点的检查点;

将过程从所述第一计算节点迁移至针对所述第一计算节点指定的默认恢复节点;以及

在针对所述第一计算节点指定的所述默认恢复节点上继续执行所述过程。

9.根据权利要求8所述的介质,所述操作还包括:

针对每个计算节点收集至少计算能力参数和节点位置参数;

基于所述计算节点的节点位置参数将所述计算节点划分为集合;以及

基于所述计算能力参数对每个集合内的节点进行排序。

10.根据权利要求9所述的介质,所述操作还包括:

识别上限和下限以确定经排序的计算节点的等级;

基于所述计算能力参数以及所述上限和所述下限将每个集合内的计算节点排序为水平等级;

将所述水平等级布置和垂直布置记录到与每个计算节点相关联的节点记录信息表中,其中所述垂直布置是至少基于每个计算节点的节点位置参数确定的;以及

用所指定的默认恢复节点填充每个节点记录信息表。

11.根据权利要求10所述的介质,其中,所述上限和所述下限是根据针对每个计算节点所收集的计算能力参数和节点位置参数的交绘图确定的。

12.根据权利要求8所述的介质,其中,所述MTBF是至少基于网络或数据存储故障计算的。

13.根据权利要求8所述的介质,还包括执行以下操作的指令:

当所述第一计算节点的MTBF小于所述最小阈值时创建检查点;以及

将与所述第一计算节点相关联的所述最小阈值更新为等于所述MTBF。

14.根据权利要求13所述的介质,还包括执行以下操作的指令:

确定所述第一计算节点的故障已经发生;以及

使用针对所述第一计算节点采取的最后一个检查点作为过程状态。

15.一种计算机系统,包括:

至少一个硬件处理器,能够互操作地与存储器耦接并且被配置为:

构建多个计算节点的虚拟树状计算结构,所述多个计算节点被映射为以父/子类型关系进行通信,其中针对所述计算节点中的每一个计算节点,一个或多个直接子代被指定为默认恢复节点,并且另一节点被指定为检查点节点;

针对所述虚拟树状计算结构的每个计算节点,执行节点故障预测模型以计算与所述计算节点相关联的平均故障间隔时间"MTBF";

基于计算出的MTBF与最大和最小阈值之间的比较来确定执行第一计算节点的检查点;

将过程从所述第一计算节点迁移至针对所述第一计算节点指定的默认恢复节点;以及

在针对所述第一计算节点指定的默认恢复节点上继续执行所述过程。

16.根据权利要求15所述的系统,还被配置为:

针对每个计算节点收集至少计算能力参数和节点位置参数;

基于所述计算节点的节点位置参数将所述计算节点划分为集合;以及

基于所述计算能力参数对每个集合内的节点进行排序。

17.根据权利要求16所述的系统,还被配置为:

识别上限和下限以确定经排序的计算节点的等级;

基于所述计算能力参数以及所述上限和所述下限将每个集合内的计算节点排序为水

平等级；

将所述水平等级布置和垂直布置记录到与每个计算节点相关联的节点记录信息表中，其中所述垂直布置是至少基于每个计算节点的节点位置参数确定的;以及

用所指定的默认恢复节点填充每个节点记录信息表。

18.根据权利要求17所述的系统,其中,所述上限和所述下限是根据针对每个计算节点所收集的计算能力参数和节点位置参数的交绘图确定的。

19.根据权利要求15所述的系统,其中,所述MTBF是至少基于网络或数据存储故障计算的。

20.根据权利要求15所述的系统,还被配置为：

当所述第一计算节点的MTBF小于所述最小阈值时创建检查点；

将与所述第一计算节点相关联的所述最小阈值更新为等于所述MTBF；

确定所述第一计算节点的故障已经发生;以及

使用针对所述第一计算节点采取的最后一个检查点作为过程状态。

用于分布式计算的主动故障恢复模型

[0001] 优先权要求

[0002] 本申请要求2014年7月29日递交的美国专利申请No.14/445,369 的优先权,其全部内容通过引用并入本文。

背景技术

[0003] 在分布式计算系统(例如,同构(簇)、异构(网格和云)等) 上执行具有成千上万的科学应用过程的关键/实时科学应用(例如地震数据处理、三维储层不确定性建模和仿真)需要高端计算能力,这可能需要数天或数周来处理数据以生成所需的解决方案。较长工作执行的成功取决于系统的可靠性。由于部署在超级计算机上的大多数科学应用只要其中一个过程故障就可能会故障,因此分布式系统中的容错是复杂计算环境中的重要特征。容许任意类型的计算机处理故障反应性地通常涉及是否允许对一个或多个过程的状态进行定期检查点设置的选择-可广泛应用于高性能计算环境中的有效技术。然而,这种技术具有与选择最优检查点间隔和检查点数据的稳定存储位置相关联的开销问题。此外,当前故障恢复模型通常限于几种类型的计算故障,并且在计算故障的情况下手动地调用当前故障恢复模型,这限制了它们的有用性和效率。

发明内容

[0004] 本公开描述了根据一个实施方式用于提供用于分布式计算的主动故障恢复模型的方法和系统,包括计算机实现的方法、计算机程序产品和计算机系统。一种计算机实现的方法,包括:构建多个计算节点的虚拟树状计算结构,针对所述虚拟树状计算结构的每个计算节点,由硬件处理器执行节点故障预测模型以计算与所述计算节点相关联的平均故障间隔时间(MTBF),基于计算出的MTBF与最大和最小阈值之间的比较来确定是否执行所述计算节点的检查点,将过程从所述计算节点迁移至作为恢复节点的不同的计算节点,以及在所述不同的计算节点上继续执行所述过程。

[0005] 该方案的其他实施方式包括相应的计算机系统、装置和记录在一个或多个计算机可读介质/存储设备上的计算机程序,它们均被配置为执行方法的动作。一个或多个计算机的系统可以被配置为通过在系统上的安装的在操作时使得系统执行动作的软件、固件、硬件或者软件、固件或硬件的组合来执行特定操作或动作。一个或多个计算机程序可以被配置为通过包括指令来执行特定操作或动作,所述指令在被数据处理装置执行时使得该装置执行动作。

[0006] 前述和其他实施方式可以各自可选地以单独或组合的方式包括以下特征中的一个或多个:

[0007] 第一方案,可与一般实现方式组合,还包括:针对每个计算节点收集至少计算能力和节点位置参数值,基于所述计算节点的节点位置参数将所述计算节点划分为集合,以及基于所述计算能力参数对每个集合内的节点进行排序。

[0008] 第二方案,可与前述方案中的任一个组合,还包括:识别上限和下限以确定经排序

的计算节点的等级，基于所述计算能力参数以及所述上限和所述下限将每个集合内的计算节点排序为水平等级，将所述水平等级布置和垂直布置记录到与每个计算节点相关联的节点记录信息表中；以及用指定的恢复节点填充每个节点记录信息表。

[0009] 第三方案，可与前述方案中的任一个组合，其中，所述上限和所述下限是根据针对每个计算节点所收集的计算能力和节点位置参数的交绘图确定的，并且所述垂直布置是至少基于每个计算节点的节点位置参数确定的。

[0010] 第四方案，可与前述方案中的任一个组合，其中，所述MTBF是至少基于网络或数据存储故障计算的。

[0011] 第五方案，可与前述方案中的任一个组合，还包括：当所述计算节点的MTBF小于所述下限时创建检查点；以及将与所述计算节点相关联的所述下限更新为等于所述MTBF。

[0012] 第六方案，可与前述方案中的任一个组合，还包括：确定所述计算节点的故障已经发生；以及使用针对所述计算节点采取的最后一个检查点作为过程状态。

[0013] 在本说明书中描述的主题可以在特定实施方式中实现，以便实现以下优点中的一个或多个。首先，所描述的故障恢复模型系统和方法具有廉价的框架设计，其即使在发生部分/严重的计算节点(例如，计算机服务器等)故障的情况下也允许计算过程的可靠的继续操作——增强业务连续性最优性。故障恢复模型系统允许继续操作并且实现高性能定级以最优地执行故障的工作执行。由于故障恢复模型是主动的 (而不是被动的)，成本针对重新处理工作进一步降低，并允许成本规避，并且从故障恢复实践节省时间和工作量二者。第二，该框架对于大量的计算节点是可扩展的。第三，框架设计考虑了不同的灾难恢复原则因素。第四，所描述的系统和方法将极大地使由不必要的过程检查点设置造成的开销最小化。第五，所描述的系统和方法可以被配置为实践任何类型的负载平衡技术以优化处理。第六，系统和方法不依赖于用于操作的本地式或集中式检查点存储。第七，系统和方法依赖于故障预测模型，以控制检查点过程的最佳布置。第八，所提出的系统和方法设计允许高度的业务连续性最优性。其他优点对于本领域普通技术人员将是显而易见的。第九，该框架设计中的故障预测模型可以捕获和解决任何类型的故障(电源、软件、硬件、网络等)。

[0014] 在附图和以下描述中阐述了本说明书的主题的一个或多个实施方式的细节。根据描述、附图和权利要求，本主题的其他特征、方面和优点将变得显而易见。

附图说明

[0015] 图1A-图1C示出了根据一个实施方式用于提供用于分布式计算的主动故障恢复模型的方法。

[0016] 图2示出了根据一个实施方式的可以用于构建节点虚拟树状结构的从节点收集的参数的示例交绘图。

[0017] 图3示出了根据一个实施方式的节点的示例虚拟树状结构。

[0018] 图4A示出了根据一个实施方式的在MTBF的计算中使用的节点性能值。

[0019] 图4B示出了根据一个实施方式的用于计算节点的MTBF的典型理论公式。

[0020] 图5是示出了根据一个实施方式的关于MTBF的检查点间隔布置的图形。

[0021] 图6示出了根据一个实施方式的当部分节点故障发生时节点的示例虚拟树状结构以及如何将恢复模型用于恢复。

[0022] 图7示出了根据一个实施方式的当节点经历半故障时针对虚拟树状结构的节点的检查点设置数据存储节点。

[0023] 图8示出了根据一个实施方式的参与应用计算的节点。

[0024] 图9A和图9B示出了根据一个实施方式的关于独立和从属过程的检查点设置节点请求。

[0025] 图10是示出了根据一个实施方式的用于提供用于分布式计算的主动故障恢复模型的示例计算设备的框图。

[0026] 在各个附图中，相似的附图标记和名称指示相似的元件。

具体实施方式

[0027] 给出以下具体描述以使得本领域任何技术人员能够做出和使用所公开的主题内容，并且在一个或多个特定实施方式的上下文中提供所述描述。对公开的实施方式的各种修改对本领域技术人员而言将显而易见，并且在不背离本公开的范围的情况下，此处定义的一般原理可适用于其他实施方式和应用。因此，本公开并非意在限于所描述的和/或示出的实施方式，而应赋予与此处公开的原理和特征一致的最宽范围。

[0028] 本公开一般地描述了用于提供用于分布式计算的主动故障恢复模型(FRM)以确保在计算节点(例如，计算机服务器等))故障的情况下业务连续性最优性的方法和系统，包括计算机实现的方法、计算机程序产品和计算机系统。尽管以下描述关注特定实施方式，但具体实施方式并不意味着针对其他用途并用符合本公开的方式来限制所描述的主题内容的适用性。

[0029] 在分布式计算系统(例如，同构(簇)、异构(网格和云)等) 上执行具有成千上万的过程的关键/实时科学应用(例如地震数据处理和三维储层不确定性仿真和建模)需要高端计算能力，并且科学应用可能花费数天或有时数周来处理数据以生成所需解决方案。较长执行的成功取决于系统的可靠性。由于部署在超级计算机上的大多数科学应用只要其中一个过程故障就可能会故障，因此分布式系统中的容错是复杂计算环境中的重要特征。容许任意类型的计算机处理故障反应性地通常涉及是否允许对一个或多个过程的状态进行定期检查点设置的选择。

[0030] 检查点设置是可广泛应用于高性能计算环境的有效技术并且是在分布式系统中的过程执行期间发生故障的情况下使用的最有效的容错技术。在检查点设置中，在节点上执行的过程的状态被周期性地保存在例如硬盘、闪存等的可靠且稳定的存储器上。在一些实施方式中，检查点设置创建描述运行过程(例如，上述“过程的状态”)的文件，操作系统可以使用该文件来在稍后重建该过程。例如，检查点文件可以包含关于设置了检查点的过程的堆栈、堆和寄存器的数据。检查点文件还可以包含待决信号状态、信号处理程序、记帐记录、终端状态以及在给定时间点重建过程所需的任意其他必要数据的状态。因此，使得过程能够在取得特定检查点的时刻并从该时刻开始、而不是通过重新开始该过程从起点开始继续执行。

[0031] 在高等级，FRM被配置为维持一致的主题应用/过程吞吐量，保持检查点的最小必要集合以优化/最小化过程返工执行时间，在恢复生存节点之间实现最佳负载平衡策略(以下更详细描述的)，使无磁盘或输入/输出操作最小化，将检查点数据存储在稳定且安全的

存储器中，和/或使存储器开销最小化。在一些实例中，FRM还可以通过使用非阻塞检查点设置来减少检查点设置延迟(过程发起检查点请求和全局检查点过程完成该处理之间的时间)，非阻塞检查点设置除非在参与恢复的节点中托管处理工作的情况下否则不阻塞执行模式的处理工作。结果，减小了处理工作执行延迟。

[0032] 更具体地，所描述的FRM在一些实例中被实现为：1)可扩展的虚拟树状结构，其在无故障计算的情况下支持高性能、稳定的计算系统，并且在发生故障的情况下支持恢复资源的高可用性；以及2)在典型实例中使用的故障预测模型(FPM)，以通过测量针对每个检查点请求的有效性和需要来使基于检查点的算法的协调和上下文切换的开销最小化。

[0033] 虚拟树状结构计算拓扑设计

[0034] 分层树状计算拓扑设计允许多个选项，所述多个选项用于分配恢复节点和可驻留在不同物理位置的远程指定的检查点数据存储节点二者)。在典型的实施方式中，参与分布式计算工作量的所有计算节点(节点)被虚拟地放置到由两个不同参数确定的虚拟树状结构中：1) 计算能力(CP-置于Y轴上)和2)节点位置(NLOC-置于X轴上)以构建虚拟树状结构。在其他实施方式中，可以在Y轴或X轴上收集和/或使用其他参数，以允许有形/有意义的分类并构建所描述的虚拟树状结构。

[0035] 节点故障预测模型

[0036] 故障预测长期以来被认为是一个具有挑战性的研究问题，主要是由于缺乏来自实际生产系统的实际故障数据。然而，计算的平均故障间隔时间(MTBF)(用于表示节点的可靠性的统计参数)可以是针对节点的不久的将来的预定义时间段内的故障率的良好指示符。

[0037] 在一些实施方式中，在分布式计算环境中的故障可以被分类为五个不同的类别，其必须被考虑以确保鲁棒且全面的故障恢复模型。例如，类别可以包括：1)崩溃故障-服务器停止，但是正常工作直到它停止为止；2)遗漏故障(接收或发送遗漏)-服务器无法响应传入请求、服务器无法接收传入消息、服务器无法发送消息；3)定时故障 -服务器的响应在指定的时间间隔之外；4)响应故障(值或状态转换故障)-服务器的响应不正确；响应的值错误，服务器偏离正确的控制流；5)任意故障-s服务器可以在任意时间产生任意响应。

[0038] 一般来说，分布式计算节点结构(例如，节点N0；N1；N2；……； Nn)通过本地或全局的网络连接(例如，互联网或其他网络云)来连接。每个节点通常具有其自己的物理存储器和本地盘(例如，独立的计算设备)，并且部署稳定的共享存储器用于节点之间的大数据集共享。在科学、实时等的应用中，节点的过程之间的通信可以通过消息传递接口(MPI)、被指定用于过程之间的全局发送/接收请求的共享存储器和/或其他通信方法来实现。通常每个过程驻留在不同的节点上，但两个或更多个单独的过程可以在单个节点上执行。

[0039] 假设过程之间的通信信道是稳定和可靠的，并且分布式计算系统中的每个节点是易失性的(意味着节点可以由于故障而离开分布式计算系统，或者在修复之后加入分布式计算系统——还假设故障停止模型，其中故障节点将与计算环境隔离——节点的故障将导致故障节点上的所有过程停止工作(故障节点上的受影响过程的所有数据都丢失)。这里，可以使用FRM恢复/备用节点(而不是挂起应用，直到故障物理上修复)以从每个受影响的过程的最后一个检查点继续处理。

[0040] 通常特定节点的节点故障情形将由驻留在与虚拟树状结构中的特定节点相同等级的任意节点确定(例如，由节点的软件代理预测——每个节点通常具有其自己的服务守

护进程“代理”)/协作地(向每个节点中保存的描述计算环境的结构的记录表中的指定的参与节点) 通知。特定节点故障的预测是允许在不久的将来在一段时间内评估故障风险并且采取主动步骤以保存与特定节点相关联的过程状态的更精细进展(更高粒度)的重要指示符。因此,如果在已经预测可能发生故障时,特定节点发生严重故障,可以规避大量的重新处理时间,这是由于可以用于恢复与特定节点相关联的过程的可用节点状态的更精细粒度。典型地,节点故障的预测与检查点的获取/存储的成本相平衡。

[0041] 图1A-图1C示出了根据一个实施方式用于提供用于分布式计算的主动故障恢复模型的集合方法100(划分为子方法100a、100b和100 c)。在其他实施方式中,提供用于分布式计算的主动故障恢复模型可以包括更多或更少的步骤/操作,其包括更多的每个所描述的步骤/操作。方法100(或其任意单独子方法)可以视情况由任何合适的系统、环境、软件和/或硬件或系统、环境、软件和/或硬件的组合(例如,在下文的图10中描述的计算机系统)来执行。在一些实施方式中,方法100的各个步骤可以并行、组合、循环或以任意顺序执行。

[0042] 构建计算节点的虚拟树状结构化模型

[0043] 转向参照图1A,在102,使用分布式计算系统中的可用节点来构建节点的虚拟树状结构化模型。树状结构被认为是“虚拟的”,原因在于节点的树实际上没有布置在树状结构中,而是用这种方式映射以按父/子类型关系进行通信。如本领域普通技术人员将理解的,计算能力(CP)和节点位置(NLOC)参数的使用仅是构建根据本公开的节点的虚拟树状接口模型的一种可能实施方式,并且还可以在其他实施方式中使用其他参数(例如,计算硬件类型和/或版本、软件版本等)。CP和NLOC参数的使用并不意味着以任何方式限制所描述的主题,并且在本公开的范围内设想了其他参数。在典型实施方式中,认为CP在节点的虚拟树状结构化模型的Y轴上,并且可以认为NLOC(或其他参数) 在节点的虚拟树状结构化模型的X轴上。

[0044] 在103a,针对参与例如计算过程的处理的分布式计算系统的所有节点收集至少计算能力(CP)、节点位置(NLOC)和/或其他参数。在一些实施方式中,将该收集的数据放入数据结构、文件等中(例如,节点记录信息表)以供虚拟树创建过程(未示出)使用。在一些实施方式中,每个节点知道所有其他节点和相关联的参数。例如,每个节点可以有权访问包含针对分布式计算系统中的节点所收集的参数信息的数据结构/文件。该信息可用于允许每个节点知道其兄弟、后代等。方法100a从103a前进至103b。

[0045] 在103b,节点基于它们的位置(NLOC)被划分成集合。方法100 a从103b前进至103c。

[0046] 在103c,基于节点CP参数在每个集合内对节点进行排序。方法 100a从103c前进至103d。

[0047] 在103d,根据从节点收集的参数的交绘图确定每个等级的下限和上限(即,阈值)。现在转到图2,图2示出了根据一个实施方式的可以用于构建节点虚拟树状结构的从节点收集的参数的示例交绘图。如图所示,每个节点(注意,#号/图案可以表示颜色以指示不同的位置(NLOC)——例如,所有“蓝色”绘制的节点在特定位置,而所有“绿色”绘制的节点在不同的特定位置)在一些实施方式中可以根据X 轴202上的存储器参数值(例如,从低到高-例如8GB-64GB的计算机服务器存储范围)和Y轴204上的CP参数值(从低到高-例如1.6-3.5GHz处理器时钟的范围)来绘制。如普通技术人员将理解的,这仅是产生交绘图的许多可能方法

中的一个。设想根据本公开的任何合适参数的使用在本公开的范围内。

[0048] 在一些实施方式中，节点的水平布置(即节点是其一部分的水平“线”)是根据图2的交绘图内的节点的位置而基于CP的。例如，基于节点的CP参数和位置，节点可以在虚拟树状结构中的底部、中间或顶部位置。在所示的示例中，在一些实例中，水平布置可以通常是最底部附接的节点将是具有最高CP参数值(更高的计算能力)的节点，而节点在树状结构中放置的位置越高，CP参数值越低(计算能力越低)。

[0049] 在一些实施方式中，节点的垂直布置(例如，沿着上述水平“线”的左或右-例如，节点304b在下面的图3中放置)取决于不同的可分类准则，例如物理位置、子网、带宽速度、电源线等，并在默认情况下平衡虚拟树状结构。例如，如果在交绘图中使用的x轴准则引导树结构中的特定节点的布置，则虚拟树状结构是平衡的并可以用作关于计算环境中是否应用正确的物理灾难恢复设置的指示符。用于垂直地对节点进行分离的附加准则可以尤其包括物理位置、子网、带宽速度、电源线和/或其他附加准则。方法100a从103d前进至103e。

[0050] 在103e，在节点记录信息表中做出每个节点的水平和/或垂直树布置条目。方法100a从103e前进至103f。

[0051] 在103f，每个节点的节点记录信息表用与该节点的相关联的指定检查点和/或恢复节点来填充。可以基于虚拟树状结构中的水平/垂直位置针对虚拟树状结构中的每个节点来确定祖先节点和后代节点，并且利用该节点记录信息来更新节点记录信息表。此外，对于每个特定节点，可以将另一个节点(例如，一个或多个直接子代)指定为特定节点的默认恢复节点，并且可以将另一节点指定为特定节点的检查点节点。通常，检查点节点不是虚拟树状结构中特定节点的兄弟、子代或祖先节点。在一些实施方式中，用于特定节点的指定恢复节点和检查点节点可以是相同的。方法100a从103f前进至104(节点故障预测模型)。

[0052] 现在转到图3，图3示出了根据一个实施方式(例如，根据上面的103a-103f构建的)节点的示例虚拟树状结构300。如上面103b中所述，302a、302b、……、302n示出了节点的位置(NLOC)的集合，例如不同网络子网中的位置。如上面103c和103d中所述，304a、304b、……、304n是通过CP参数排序的节点，并且通过从节点收集的参数的交绘图划分为水平等级。例如，节点304a可以具有比节点304n更高的CP参数值。此外，节点(例如，306a和306b)基于上述(或其他)不同的可分类准则(例如物理位置、子网、带宽速度等)在同一水平等级内垂直地分离。在一些节点内示出了示例唯一节点标识(节点ID)值(例如，304n对于等级1、节点1显示“N1(1)”，节点306b对于等级2、节点13显示“N2(13)”)等(尽管未示出N2以下的节点ID)。设想任何合适的唯一节点标识符在本公开的范围内。

[0053] 注意，虚拟树状结构的拓扑是自适应的。例如，如果将更多或更少的节点添加至特定位置，则节点CP值改变，节点被替换为更高的 CP/存储器模型等，可以更新该NLOC划分的集合内的关系树，并且还可以更新与其他NLOC划分的集合中的其他节点的关系。例如，如果将新节点添加到分布式计算系统，则可以再次执行虚拟树状结构创建过程。在一些实例中，树可以被部分或全部重构。

[0054] 转向图4A，图4A示出了根据一个实施方式的在MTBF的计算中使用的节点性能值400a。这里，402是故障开始点(故障开始的时间(或“停机时间”))，而404是恢复开始点404(处理重新开始的时间(或“正常运行时间”))。406是故障之间的时间(“停机时间”和“正常运行时间”之差是在两个事件之间运行的时间量)。408表示故障。

[0055] 转向图4B,图4B示出了根据一个实施方式的用于计算节点的MTB F的典型理论公式400b。这里,MTBF是操作时段的总和(例如,对于节点)除以观察到的故障408的数量(例如,同样对于该节点)。如本领域普通技术人员将理解的,节点故障的预测中也可以使用利用根据本公开的使用更多或更少的数据值的MTBF或类似值的其他变形,并且结果动作如下所述。方法100a从104前进至方法100b(图1B)。

[0056] 在图1B,执行是否检查点过程状态和/或迁移过程(工作)的决定100b。检查点时间计算是使(例如,基于节点的MTBF)将节点的检查点带到其中认为更有必要的时间的系统开销最小。

[0057] 在106,将在方法100a中计算的节点的MTBF与最小阈值(MinLim it)和最大阈值(MaxLimit)进行比较。初始地,MinLimit和MaxLim it被设置为某个预定的时间值。可以根据需要改变MinLimit值(例如,以确定何时对节点执行下一次健康检查)。还可以根据需要改变MaxL imit(例如,以反映增加的MTBF值)。

[0058] 如果在106,MTBF在MinLimit和MaxLimit之间(例如,大于或等于MinLimit并且小于或等于MaxLimit),则方法100b前进至108。在1 08,不采取节点的检查点。方法100b从108前进至110。

[0059] 在110,基于由该节点的软件代理针对该节点执行的新的MTBF计算来调整采取下一个检查点的时间(根据故障评估节点的当前状态——在上一时段中发生了多少故障)。用这种方式,可以根据节点的状态动态地调整检查点间隔。例如,在第一检查点之后,如果针对下一检查点设置了五分钟,则等待五分钟。在下一检查点之后,基于最近五分钟内的故障(如果有)执行MTBF评估。基于所计算的MTBF,可以向上或向下调整检查点间隔(例如,如图5所示)。方法100b从110前进至关于图1A描述的104。

[0060] 如果在106处,MTBF大于MaxLimit,则方法100b前进至112。在 112,不采取节点的检查点,并且MaxLimit被更新为等于MTBF。在一些实施方式中,MaxLimit高于特定阈值可以发起关于MaxLimit太高的警报的生成。方法100b从112前进至110。

[0061] 如果在106处,MTBF小于MinLimit,则方法100b前进至114。在 114,采取节点的检查点,并且MinLimit被更新为等于当前计算的MTB F值。在一些实施方式中,检查点可以尤其包括过程状态(寄存器内容)、存储器内容、通信状态(例如,打开的文件和消息信道)、相关的内核上下文和/或排队的工作。方法100b从114前进至116。

[0062] 在116,执行关于工作(过程)是否应当被迁移的决定是基于主动故障恢复的阈值(例如,在106确定的MTBF＜MinLimit值)。可以根据特定节点中的故障频率主观地强制阈值。如果确定不应当迁移工作,则方法100b前进至110。如果确定应当迁移工作,则方法100b前进至1 18以执行工作迁移。

[0063] 现在转向图5,图5是示出了根据一个实施方式的关于MTBF的检查点间隔布置的图形。如图所示,随着MTBF减小(例如,分别与示例检查点间隔相对应的504a、504b和504c),检查点之间的时间(检查点间隔)(例如,502a、502b和502c)变短。随着MTBF增加(例如,在504d),检查点间隔减小(例如,在502d)。这是为了确保随着故障风险增加(由于MTBF减少),如果节点故障则以较短的间隔为节点创建检查点以最小化开销并且最大化业务连续性的最优性是有利的 (可以在更接近实际故障时间的点恢复,以使损失的处理最小化)。

[0064] 现在转到图1C,图1C示出了用于在节点之间迁移工作的方法流 100c。

[0065] 在119a，恢复节点软件代理协商以确定哪个节点应当托管要迁移的工作。在一些实施方式中，协商用于负载平衡目的。在其他实施方式中，可以使用其他参数/准则用于协商目的。方法100c从119a前进至109b。

[0066] 在119b，执行关于节点的故障是否已经发生的确定。如果确定没有发生节点的故障，则方法100c前进至119c。如果确定发生了故障，则方法100c前进至119d。

[0067] 在119c，提取节点的过程状态。在一些实施方式中，过程状态可以尤其包括过程状态(寄存器内容)、存储器内容、通信状态(例如，打开的文件和消息信道)、相关的内核上下文和/或排队的工作。方法100c从119c前进至119e。

[0068] 在119d处，使用最后一个检查点来代替当前节点状态(因为节点已经故障并且“停机”/不可用于从中检索过程状态)。在一些实施方式中，检查点可以尤其包括过程状态(寄存器内容)、存储器内容、通信状态(例如，打开的文件和消息信道)、相关的内核上下文和/或排队的工作。方法100c从119d前进至119e。

[0069] 在119e，故障的祖先恢复节点软件代理将保持向新的恢复节点转发信号/通信(例如，通过以通信协议级搜索消息)。注意，在一些实施方式中，如果故障节点被修复，则转发信号/通信的责任可以由修复的节点执行(这时将向修复的节点移交转发信号/通信的过程)。方法100c从119e前进至119f。

[0070] 在119f，执行从“停机”节点到恢复节点的过程转移。转移后的状态通常包括过程的地址空间、执行点(寄存器内容)、通信状态 (例如，打开的文件和消息信道)和/或其他操作系统相关状态。方法 100c从119f前进至119g。

[0071] 在119g，该过程继续在恢复节点上执行。方法100c从119g停止。返回图1B，针对特定停机节点的方法118从118停止。在典型实施方式中，如果已经对故障节点执行了工作迁移，则故障节点与当前计算运行隔离，并且故障节点即使被修复也不能返回到相同的工作族并参与 (例如，从虚拟树状结构中的一个或多个节点的节点记录信息表中将节点移除，并且必须等待直到新的计算运行开始)。对于使用节点故障预测模型的不同节点，处理返回到图1A。在其他实施方式中，可以修复故障节点(例如，修复的节点可以被重新集成到虚拟树状结构节点记录信息表中，由节点故障预测模型过程，并且开始处理、转发信号/通信等)。

[0072] 返回图1A所示，在104处，在一些实施方式中，执行节点故障预测模型以尤其评估每个节点的当前机器状态，以确定是否执行节点的检查点，将工作(过程)从一个节点迁移至另一个节点等。每个节点的MTBF的计算由软件代理计算，该软件代理在驻留在树结构中与要针对故障条件评估的特定节点相同等级中的至少一个节点上和/或在特定节点自身上驻留/执行。例如，如图3所示，节点304b所在的等级中的任何节点可以确定节点304b的MTBF，并向适当节点通知该确定。

[0073] 如104所示，在一些实施方式中，可以在故障预测模型中使用的值包括例如通过“健康检查”类型/“心跳”程序产生的一个或多个系统日志105a(例如，网络连接故障/拒绝、分组丢失、计算速度降级、低内存条件、存储/网络问题等)，对检查的时间段的调整105b(例如，表示等待执行下一次健康检查的时间段的动态计算的值，该下一次健康检查是由参与计算的每个节点上执行的功能执行的)。在动态计算检查的时间段值105b之后调用该功能以确定下一时间段和周期性地 (例如，tp时间段)收集的故障类别字典105c。在一些实施方

式中，可以对问题、故障等的类型进行加权(例如，网络/存储更重要等)。

[0074] 在105d处，通过用主观指派给每个故障类别的权重值对每个故障进行分类，来计算故障类型频率，以在计算每个tp时间段的故障频率时度量影响。例如，可以监测电力供应和网络连接性，并且可以针对特定时间段tp计算它们的故障类型频率。在其他实施方式中，可以使用任何适当的算法来计算故障类型频率。在105e，在一些实施方式中，可以通过将1除以计算的故障类型频率来计算MTBF。

[0075] 图6示出了根据一个实施方式的当部分节点故障发生时节点的示例虚拟树状结构600以及如何将恢复模型用于恢复。实现虚拟树状结构中的恢复首先通过来自要恢复的故障节点的后代的同一等级的任何节点的通知而实现，否则将通知发送给故障节点的祖先。

[0076] 在一个示例中，如果节点602故障，则节点602的直接子代(后代)604或具有比该父节点更高的计算能力的进一步后代606将检测到已经出现问题(例如，与节点602的连接丢失、来自节点602的数据接收停止、节点502的心跳检测指示节点故障等)。问题是哪些其他节点应该替代节点602以处理节点602执行的工作，以优化业务连续性。选项是父节点604或后代节点606或608。在这种情况下，可以向节点606 或608指派节点602的工作(由于其高得多的计算能力CP)，以在更短的时间段内完成最初在节点602上运行的工作，以优化业务连续性。使用节点602的后代的决定还可以取决于由后代节点的负载平衡分析所确定的子节点的负载。

[0077] 通过分配来自节点的兄弟后代的任何活动的、可用的和轻微加载(由系统中的负载平衡确定的)节点(如果故障节点的直接后代中没有一个活动)来恢复任何故障的节点被称为代-停止。例如，如果节点610故障，则节点610的子代应该检测节点610的故障。然而，在该示例中，节点610的所有子代也都停机。然后问题变成哪个(哪些)节点应当代替节点510来完成节点510的工作。这里，向祖先节点612通知(例如，通过驻留在两个不同节点上的两个工作之间的通信协议——例如消息传递接口(MPI)和/或其他协议)节点610的故障，并且将寻找其相同等级的亲戚来接管至少故障节点610的工作(以及在一些实例中也是节点610的后代的工作)的处理。这里，节点612与节点614通信(注意节点614可以在不同的子网中-参照图3)以查看其及其子代是否可以承担故障节点610的工作的处理。在该示例中，假设节点614接受该故障恢复任务，则视情况，其可以将工作委托给其直接子代，所述直接子代可以将工作委托给它们的子代等(例如，基于计算能力、主题、负载平衡等)以优化业务连续性。还要注意，在该示例中，节点614 的一个或两个直接子节点还可以与不同父节点的子节点(例如，节点 616)通信，以便也参与辅助对最初链接到故障节点610或者节点614 及其对其右边的兄弟618(或该等级的其他节点612)的工作的操作。通常在这种情况下，应该从最自下而上(具有更高的CP值)的节点开始执行恢复，其中其他兄弟的后代节点的最近的后代节点将参与叶节点的恢复，直到那时叶节点才会参与恢复其祖先树。

[0078] 现在转向图7，图7示出了根据一个实施方式的当节点经历半故障时针对虚拟树状结构的节点的检查点设置数据存储节点。例如，节点610具有用于存储检查点数据的指定的一个或多个检查点设置数据存储节点702。同样，节点704还具有被指定为节点610的检查点设置数据存储节点的一个或多个检查点设置数据存储节点702。在一些实施方式中，多个节点可以共享相同的检查点设置数据存储节点。在其它实施方式中，检查点设置数据存储节点702仅由一个节点或几个节点(例如，在相同子网中、兄弟等)使用以扩展检查点设置数据

存储节点的数量，使得一个或多个检查点数据存储节点的故障不会导致大量检查点设置数据的丢失。如果节点故障，那么具有恢复该节点任务的节点可以访问节点记录信息表，以确定故障节点的默认指定恢复节点和检查点设置数据存储节点。

[0079] 在典型实施方式中，每个过程仅维护一个永久性检查点。这减少了总体存储开销，并消除了对垃圾收集活动以清理未使用/放弃的检查点的需要。在一些实施方式中，每个节点的检查点数据被保存在处于节点的相同等级(例如，兄弟)的节点中，因为那些节点同时故障的概率较低。在典型实施方式中，检查点数据存储节点用这种方式实现，以使捕获本地工作的安全状态的风险最小化。检查点数据存储节点具有用于当前相关工作的信息，该执行模式包括驻留在用于那些工作的存储器中或在队列中排队的工作集数据。

[0080] 现在转到图8，图8示出了根据一个实施方式的参与应用计算的节点。如前所述，将由节点的软件代理发起检查点请求，在所述节点的软件代理中预测模型的程度或期望的可靠性需要检查点。如果节点具有独立的过程(例如，线程等)X，则其仅使用其自身在树中的相应检查点存储节点执行检查设置动作，而不将请求传播到其它节点(由于没有其他节点参与独立过程X)。

[0081] 然而，如果过程是从属过程(例如，依赖于其他过程)，则将应用最小检查点方法，其中检查点发起者节点识别自上一次检查点/ 正常通信以来与其通信的所有过程，并将向它们全部传播请求。在接收到请求时，每个过程进而识别已经与其通信的所有过程，并向它们传播请求等直到不能识别更多过程为止。

[0082] 参照图8，识别从属过程节点802和独立过程节点804二者。例如，对于示例从属过程节点，节点806是用于特定过程的顶级节点。由相应的箭头指示执行从属于节点806的过程的过程的节点。对于从属过程节点(例如，806)，故障被(例如，父节点810)传送给所有参与的过程节点(例如，812等)，原因在于所有参与的过程节点一起工作，并且其他节点有必要保存它们过程的状态直到恢复其他从属过程节点 (例如，806)为止，并且然后从属过程可以在它们停止的地方继续。然而，节点808正在执行独立的过程并且没有从属，所以恢复仅需要关注独立节点本身。

[0083] 参照图9A和图9B，图9A和图9B分别示出了根据一个实施方式的关于独立过程900a和从属过程900b的检查点设置节点请求。图9A示出了独立过程。例如，如果独立过程N3 902接收到检查点请求，则执行 N3的检查点而不考虑其他过程。在从属过程N2 902b如图9B所示的情况下，一旦接收到检查点请求，过程N2 902b就将检查点请求传递到直接从属于它的过程(例如从属过程N3 904b)。从属过程904B然后将检查点请求传递到直接从属于它的过程(例如，从属过程N41 906b和N46 9 08b)等。注意，由于通知检查点/向从属过程请求检查点的时间，对于低级从属过程(例如，从属过程N41 906b和N46 908b)的检查点可能稍微晚于上述“父”从属过程(例如，从属过程N3 904b)的检查点。在一些实施方式中，每个从属过程可以向请求从属过程通知其检查点设置操作何时完成。

[0084] 转向图10，图10是示出了根据一个实施方式的用于提供用于分布式计算的主动故障恢复模型的示例计算设备1000的框图。在一些实施方式中，EDCS 1000包括计算机1002和网络1030。在其他实施方式中，多个计算机和/或网络可以一起工作以执行上述方法。

[0085] 示出的计算机1002旨在包括计算设备(例如计算机服务器)，但是还可以包括台式计算机、膝上型/笔记本计算机、无线数据端口、智能电话、个人数字助理(PDA)、平板计算设

备、这些设备内的一个或多个处理器、或任意其他合适的处理设备(包括计算设备的物理和/ 或虚拟实例)。计算机1002可以包括计算机,该计算机包括可以接受用户信息的输入设备(例如键区、键盘、触摸屏或其他设备(未示出)) 以及输出设备(未示出),该输出设备传达与计算机1002的操作相关联的信息,包括数字数据、视觉和/或音频信息或用户界面。

[0086] 在一些实施方式中,计算机1002可以用作客户端和/或服务器。在典型的实施方式中,计算机1002充当并行处理节点,以及还充当根据本公开的软件代理或其他应用、过程、方法等(即使未示出)(例如,应用1007)。示出的计算机1002可通信地与网络1030耦接。在一些实施方式中,计算机1002的一个或多个组件可以被配置为在并行处理和/或基于云计算的环境中操作。计算机1002的实施方式也可以使用消息传递接口(MPI)或其他接口通过网络1030进行通信。

[0087] 在更高的层面,计算机1002是可操作为接收、发送、处理、存储或管理与根据一个实施方式提供用于分布式计算的主动故障恢复模型相关联的数据和信息的电子计算设备。根据一些实施方式,计算机 1002还可以包括或通信地耦接到仿真服务器、应用服务器、电子邮件服务器、网络服务器、缓存服务器、流传输数据服务器、分析服务器和/或任何其他服务器。

[0088] 计算机1002可以通过网络1030从应用1007(例如,在另一计算机1002上执行的应用)接收请求,并通过在适当的软件应用1007中处理所述请求来响应于所接收的请求。另外,还可以从内部用户(例如,从命令控制台或通过其他适当的访问方法)、外部或第三方、其他自动化应用以及任意其他适当的实体、个人、系统或计算机向计算机10 02发送请求。

[0089] 计算机1002的每个组件可以使用系统总线1003进行通信。在一些实施方式中,计算机1002的任意和/或所有组件(不论硬件和/或软件)可以使用应用编程接口(API)1012和/或服务层1013通过系统总线1003与彼此和/或接口1004进行接口连接。API 1012可以包括针对例程、数据结构和对象类的规范。API 1012可以是独立于或依赖于计算机语言,并且指的是完整的接口、单个功能或甚至是一组API。服务层 1013向计算机1002和/或计算机1002是其一部分的系统提供软件服务。计算机1002的功能可以对于使用该服务层的所有服务消费者是可访问的。软件服务(例如由服务层1013提供的软件服务)通过定义的接口提供可重用的、定义的业务功能。例如,接口可以是以JAVA、C++或以可扩展标记语言(XML)格式或其它合适格式提供数据的其它合适语言所编写的软件。虽然被示为计算机1002的集成组件,但是备选实施方式可以将API 1012和/或服务层1013示为作为相对于计算机1002的其他组件独立的组件。此外,在不脱离本公开的范围的情况下,API 10 12和/或服务层1013的任意或所有部分可以被实现为另一软件模块、企业应用或硬件模块的子模块或副模块。

[0090] 计算机1002包括接口1004。虽然在图10中被示为单个接口1004,但是可以根据计算机1002的特定需要、期望或特定实现而使用两个或更多个接口1004。接口1004由计算机1002用于与连接到网络1030的分布式环境(包括并行处理环境)(无论是否示出)中的其它系统通信。通常,接口1004包括以合适的组合以软件和/或硬件编码的逻辑,并且可操作为与网络1030通信。更具体地,接口1004可以包括支持与通过网络1030的通信相关联的一个或多个通信协议的软件。

[0091] 计算机1002包括处理器1005。虽然在图10中被示为单个处理器 1005,但是可以根据计算机1002的特定需要、期望或特定实现而使用两个或更多个处理器。通常,处理器1005

执行指令并操纵数据以执行计算机1002的操作。具体地，处理器1005执行提供用于分布式计算的主动故障恢复模型所需的功能。

[0092] 计算机1002还包括保存计算机1002和/或计算机作为其一部分的系统的其他组件的数据的存储器1006。虽然在图10中被示为单个存储器1006，但是可以根据计算机1002的特定需要、期望或特定实现而使用两个或更多个存储器。虽然存储器1006被示为计算机1002的集成组件，但是在备选实施方式中，存储器1006可以在计算机1002外部。在一些实施方式中，存储器1006可以保存和/或引用相对于方法100描述的任何数据(例如，检查点数据覆盖得分、同一性得分、深度比等) 和/或根据本公开的任意其他适当的数据中的一个或多个。

[0093] 应用1007是根据计算机1002和/或计算机1002是其一部分的系统的特定需要、期望或特定实现来提供功能(尤其是针对一些实施方式提供用于分布式计算的主动故障恢复模型所需的功能)的算法软件引擎。例如，应用1007可以用作软件主机、科学处理应用、检查点设置应用、恢复应用和/或根据本公开的任意其他类型的应用(无论是否示出)(或其一部分)。尽管被示为单个应用1007，但是应用1007可以被实现为计算机1002上的多个应用1007。另外，虽然被示出为与计算机1002集成，但是在备选实施方式中，应用1007可以在计算机1002外部并且与计算机802分开执行。

[0094] 可以存在与执行根据本公开的功能的分布式计算机系统相关联的任意数量的计算机1002。此外，在不脱离本公开的范围的情况下，术语“客户端”、“用户”和其他适当的术语可以适当地互换使用。此外，本公开包含许多用户/过程可以使用一个计算机1002，或者一个用户/过程可以使用多个计算机1002。

[0095] 在本说明书中描述的主题和功能操作的实施可以在数字电子电路中、在有形实施的计算机软件或固件中、在计算机硬件中实现，包括在本说明书中公开的结构及其结构等同物、或它们中的一个或多个的组合。在本说明书中描述的主题的实施可以被实现为在有形非瞬时计算机存储介质上编码的一个或多个计算机程序，即计算机程序指令的一个或多个模块，用于由数据-处理装置执行或者控制数据数据处理装置的操作。备选地或另外地，程序指令可以在人工产生的传播信号 (例如，机器产-生的电、光或电磁信号)上编码，所述信号被产生以对信息进行编码以传输到合适的接收机装置，以由数据处理装置执行。计算机存储介质可以是机器可读存储设备、机器可读存储基板、随机或串行存取存储器设备、或它们中的一个或多个的组合。

[0096] 术语“数据处理装置”是指数据处理硬件，并且涵盖用于处理数据的所有类型的装置、设备和机器，例如包括可编程处理器、计算机、多处理器或计算机。该装置还可以是或还包括专用逻辑电路，例如中央处理单元(CPU)、协处理器(例如，图形/视觉处理单元(GP U/VPU))、FPGA(现场可编程门阵列)-或ASIC(专用集成电路)。在一些实施方式中，数据处理装置和/或专用逻辑电路可以是基于硬件和/或基于软件的。可选地，装置可以包括为计算机程序创建运行环境的代码，例如，构成处理器固件、协议栈、数据库管理系统、操作系统或者上述各项中的一项或多项的组合的代码。本公开考虑具有或不具有常规操作系统(例如LINUX、UNIX、WINDOWS、MAC OS、ANDROID、 IOS或任意其它合适的常规操作系统)的数据处理装置的使用。

[0097] 可以以任何形式的编程语言(包括编译或解释语言、或声明或程序语言)来写计算

机程序(也可以称作或描述为程序、软件、软件应用、模块、软件模块、脚本或代码),所述编程语言包括:编译或解释语言、或者声明或程序语言,并且可以以任何形式来部署计算机程序,包括部-署为单独的程序或者部署为适合于用于计算环境的模块、组件、子例程、或者其它单元。计算机程序可以、但无需与文件系统中的文件相对应。程序可以存储在保存其他程序或数据(例如,存储在标记语言文档中的一个或多个脚本)的文件的一部分中、专用于所讨论的程序的单个文件中、或者存储在多个协同文件中(例如,存储一个或多个模块、子程序或代-码部分的文件)。计算机程序可以被部署为在一个计算机上或者在位于一个站点或分布在多个站点并且通过通信网络互连的多个计算机上执行。尽管各图中所示的程序的部分被示为通过各种对象、方法或其他过程实现各种特征和功能的单独模块,但是视情况程序可以替代地包括多个子模块、第三方服务、组件、库等。相反,各种组件的特征和功能可以视情况组合成单个组件。

[0098] 本说明书中描述的过程和逻辑流可以由一个或多个可编程计算机来执行,所述一个或多个可编程计算机执行一个或多个计算机程序以通过操作输入数据并且产生输出来执行功能。过程和逻辑流也可以由专用逻辑电路(例如CPU、FPGA或ASIC)来执行,并且装置也可以实现为专用逻辑电路(例如CPU、FPGA或ASIC)。

[0099] 适合于执行计算机程序的计算机可以基于通用或专用微处理器、这两者或任何其它类型的CPU。通常,CPU将从只读存储器(ROM)或随机存取存储器(RAM)或者这二者接收指令和-数据。计算机的必不可少的元件是用于执行或运行指令的CPU和用于存储指令和数据的一个或多个存储器设备。通常,计算机还将包括用于存储数据的一个或多个大容量存储设备(例-如,磁盘、磁光盘或光盘),或可操作耦接以便从所述一个或多个大容量存储设备接收或向其发送数据。然而,计算机不需要具有这些设备。此外,计算机可以嵌入在另一设备中,例如,移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏机、全球定位系统(GPS)接收机或者便携式存储设备(例如,通用串行总线(USB)闪存驱动器),仅举几个例子。

[0100] 适合于存-储计算机程序指令和数据的计算机可读介质(视情况,暂时或非暂时的)包括所有形式的非易失性存储器、介质和存储器设备,其包括例如半导体存储器设备、例如可擦除可编程只读存储器(E PROM)、电可擦除可编程只读存储器(EEPROM)和闪存设备;磁盘(例如内部硬盘或可移动盘);磁光盘;以及CD-ROM、DVD+/-R、DVD-R AM和DVD-ROM盘。存储器可以存储各种对象或数据,包括:高速缓存区、类(class)、框架、应用、备份数据、工作、网页、网页模板、数据库表格、存储商业信息和/或动态信息的存储库、以及包括任意参数、变量、算法、指令、规则、约束、对其的引用在内的任何其它适当的信息。另外,存储器可以包括任意其他适当的数据,诸如日志、策略、安全或访问数据、报告文件等。处理器和存储器可以由专用逻辑电路来补充或者并入到专用逻辑电路中。

[0101] 为了提供与用户的交互,本说明书中描述的主题的实施可以实现在计算机上,该计算机具有用于向用户显示信息的显示设备(例如, CRT(阴极射线管)、LCD(液晶显示器)、LED(发光二极管)或等离子监视器)和用户可以向计算机提供输入的键盘和指点设备(例如,鼠标、轨迹球或轨迹板)。还可以使用触摸屏(诸如具有压敏性的平板计算机表面、使用电容或电感测的多点触摸屏或其它类型的触摸屏) 向计算机提供输入。其它类型的设备也可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何类型的传感反馈,例如,视觉反馈、听觉反馈或触觉反馈;以及可以以任意形式(包括声音、语音或触觉输入)来接收

来自用户的输入。此外，计算机可以通过向用户使用的设备发送文档或者从该设备接收文档，来与用户交互；例如，通过响应于从用户客户端设备上的网络浏览器接收到的请求而向所述网络浏览器发送网页，来与用户交互。

[0102] 术语“图形用户界面”或GUI可以以单数或复数形式使用，以描述一个或多个图形用户界面以及特定图形用户界面的每一次显示。因此，GUI可以表示任意图形用户界面，包括但不限于网络浏览器、触摸屏或处理信息并且有效地向用户呈现信息结果的命令行界面(CLI)。通常，GUI可以包括多个UI元素，其中一些或全部与网络浏览器相关联，诸如可由商业套件用户操作的交互式字段、下拉列表和按钮。这些和其他UI元素可以与网络浏览器的功能相关或表示网络浏览器的功能。

[0103] 本说明书中描述的主题的实施可以实现在计算系统中，该计算系统包括后端-组件(例如，数据服务器)、或包括中间件组件(例如，应用服务器)、或者包括前端组件(例如，具有用户通过其可以与本说明书中描述的主题的实现进行交互的图形用户界面或者网络浏览器的客户端计算机)、或者一个或多个此类后端组件、中间件组件或前端--组件的任意组合。可以通过任意形式或方式的有线和/或无线数字数据通信(例如，通信网络)来互连系统的组件。通信网络的示例包括局域网(LAN)、无线电接入网络(RAN)、城域网(MAN)、广域网(WAN)、全球微波接入互操作性(WIMAX)、使用例如802.11a/b/g/ n和/或802.20的无线局域网(WLAN)、互联网的全部或一部分、和/ 或一个或多个位置处的任意其它通信系统。网络可以在网络地址之间传递例如互联网协议(IP)分组、帧中继帧、异步传输模式(ATM)单元、语音、视频、数据和/或其它适合信息。

[0104] 计算系统可以包括客户端和服务器。客户端和服务器一般相互远离并且通常通过通信网络进行交互。客户端和服务器的关系通过在相应计算机上运行并且相互具有客户端-服务器关系的计算机程序来产生。

[0105] 在一些实施方式中，计算系统的任意或所有组件(硬件和/或软件)可以使用应用编程接口(API)和/或服务层彼此和/或与接口进行接口连接。API可以包括针对例程、数据结构和对象类的规范。API可以是独立于或依赖于计算机语言，并且指的是完整的接口、单个功能或甚至是一组API。服务层向计算系统提供软件服务。计算系统的各种组件的功能可以经由该服务层对于所有服务消费者是可访问的。软件服务通过定义的接口提供可重用的、定义的业务功能。例如，接口可以是以JAVA、C++或以可扩展标记语言(XML)格式或其它合适格式提供数据的其它合适语言所编写的软件。API和/或服务层可以是与计算系统的其他组件相关的集成组件和/或独立组件。此外，在不脱离本公开的范围的情况下，服务层的任意或所有部分可以被实现为另一软件模块、企业应用或硬件模块的子模块或副模块。

[0106] 尽管本说明书包含许多具体实现细节，然而这些细节不应被解释为对要求保护的范围或任何发明的范围构成限制，而是用于说明特定于具体发明的具体实施例的特征。在单个实现中，还可以组合实现本说明书中在独立实现的上下文中描述的特定特征。相反地，在单个实现的上下文中描述的各种特征也可在多个实现中单独地实现，或以适当的子组合来实现。此外，虽然特征可以在上面描述为在某些组合中起作用并且甚至最初如此要求保护，但是来自所要求保护的组合的一个或多个特征在一些情况下可以从组合中删除，并且所要求保护的组合可以针对子组合或子组合的变体。

[0107] 类似地，虽然在附图中以特定顺序描绘了操作，但是这不应被理解为要求这些操

作以示出的特定顺序或以顺序次序执行，或者需要执行所有示出的操作来实现期望的结果。在特定环境中，多任务处理和并行处理可能是有利的。此外，在上述的实施方式的各种系统模块和组件的分离和/或集成不应被理解为在所有实施方式中要求这样的分离和/或集成，并且应该理解的是，所描述的程序组件和系统一般可以一起集成在单个软件产品或封装为多个软件产品。

[0108] 描述了本主题的具体实施方式。对于本领域技术人员显而易见的是，所描述的实施方式的其他实施方式、改变和置换在所附权利要求的范围内。例如，在权利要求书中记载的动作可以以不同顺序来执行，并且仍然实现期望结果。

[0109] 因此，示例实施方式的上述描述不限定或限制本公开。在不脱离本公开的精神和范围的情况下，还可以有其他改变、替换和变化。

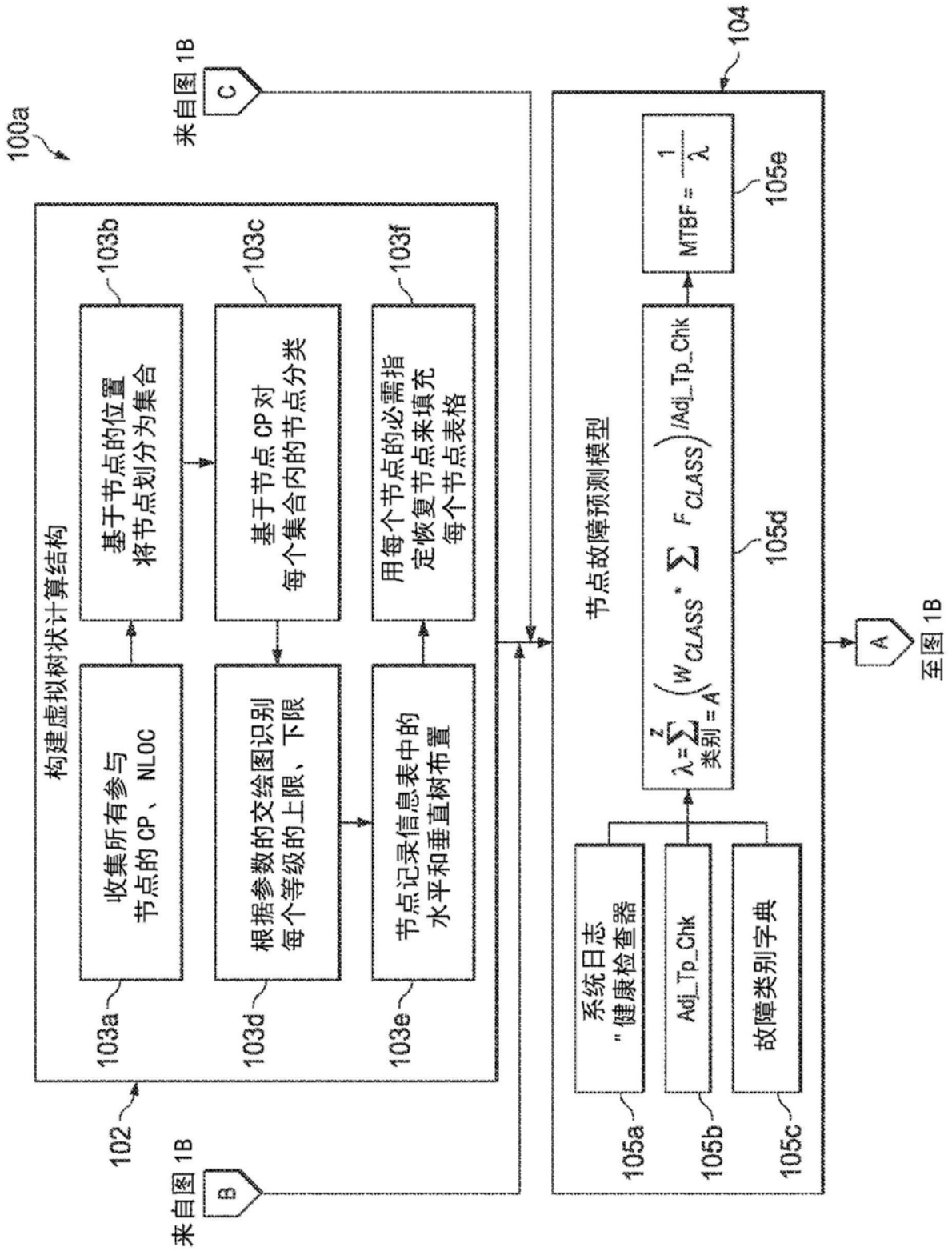

100a
来自图 1B
C
104
构建虚拟树状计算结构
基于节点的位置
将节点划分为集合
103b
基于节点 CP 对
每个集合内的节点分类
103c
用每个节点的必需指
定恢复节点来填充
每个节点表格
103f
收集所有参与
节点的 CP、NLOC
103a
根据参数的交绘图识别
每个等级的上限、下限
103d
节点记录信息表中的
水平和垂直树布置
103e
102
来自图 1B
B
节点故障预测模型
MTBF = 1/λ
105e
$\lambda=\sum_{\text{类别}=A}^{Z}\left(W_{CLASS}*\sum F_{CLASS}\right)/Adj\_Tp\_Chk$
105d
系统日志
"健康检查器
105a
Adj_Tp_Chk
105b
故障类别字典
105c
A
至图 1B

图1A

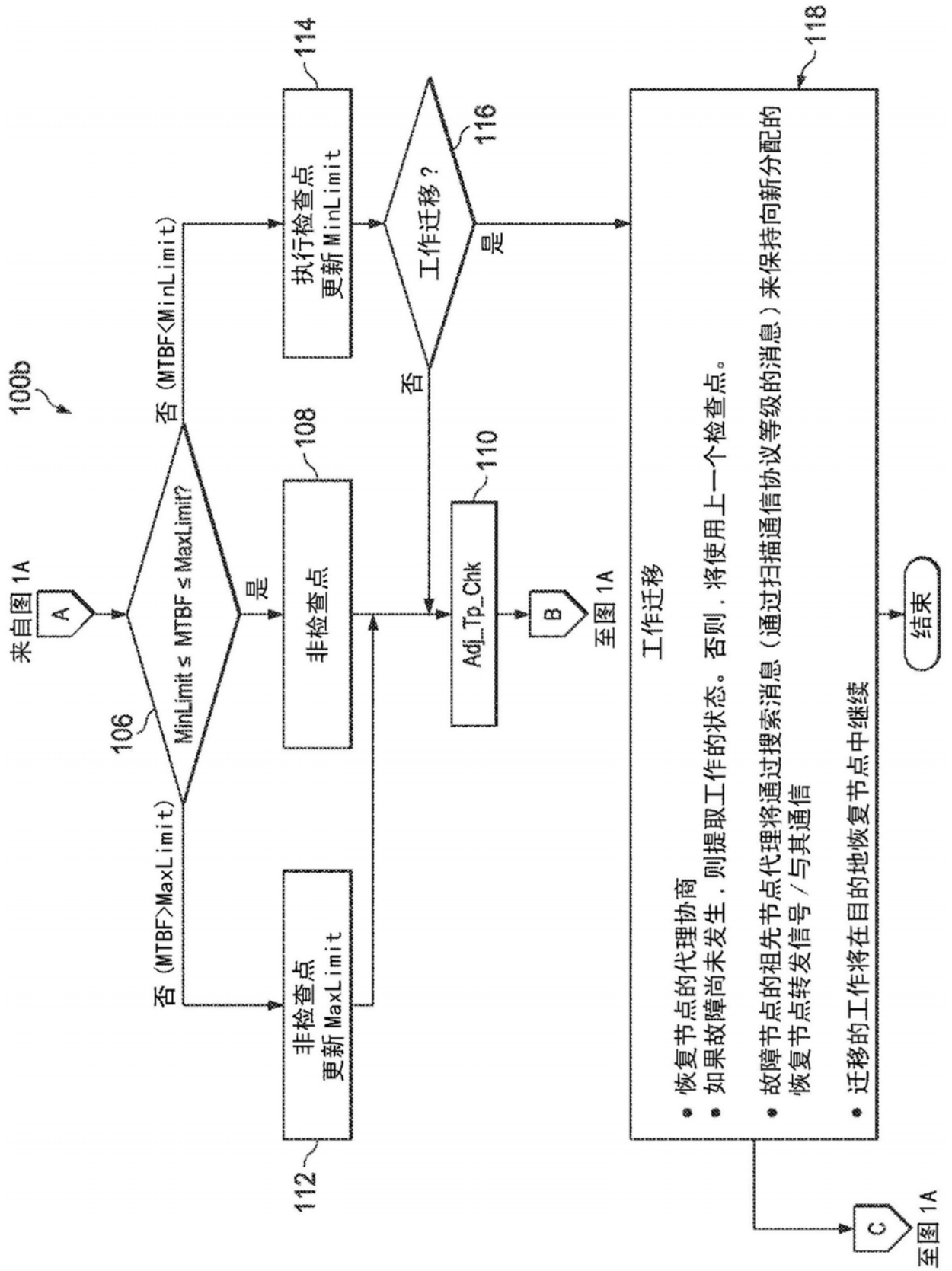

100b
来自图 1A
A
106
MinLimit ≤ MTBF ≤ MaxLimit?
否 (MTBF>MaxLimit)
是
否 (MTBF<MinLimit)
非检查点
更新 MaxLimit
112
非检查点
108
执行检查点
更新 MinLimit
114
工作迁移？
116
否
是
Adj_Tp_Chk
110
B
至图 1A
工作迁移
118
恢复节点的代理协商
如果故障尚未发生，则提取工作的状态。否则，将使用上一个检查点。
故障节点的祖先节点代理将通过搜索消息（通过扫描通信协议等级的消息）来保持向新分配的恢复节点转发信号／与其通信
迁移的工作将在目的地恢复节点中继续
结束
C
至图 1A

图1B

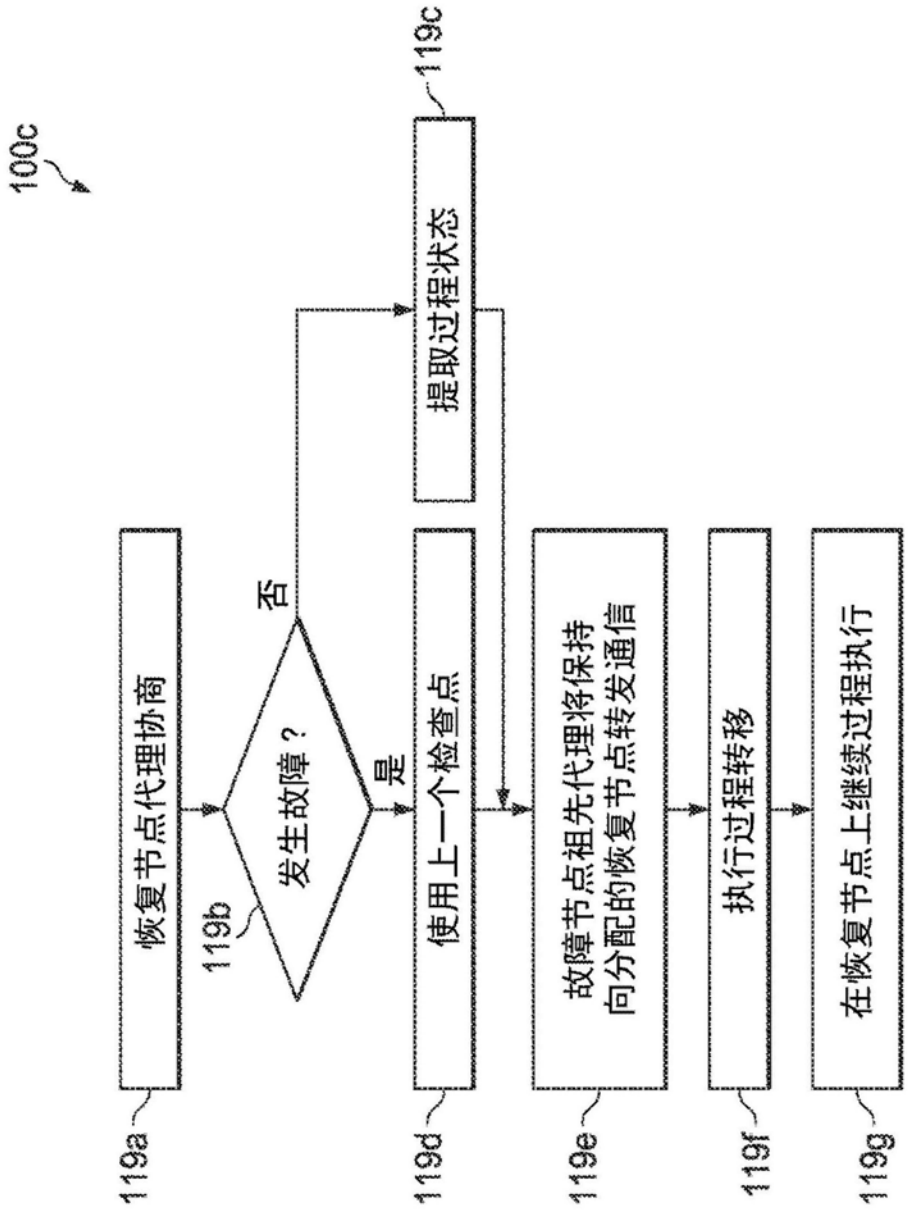

100c
恢复节点代理协商
119a
发生故障？
119b
否
是
提取过程状态
119c
使用上一个检查点
119d
故障节点祖先代理将保持
向分配的恢复节点转发通信
119e
执行过程转移
119f
在恢复节点上继续过程执行
119g

图1C

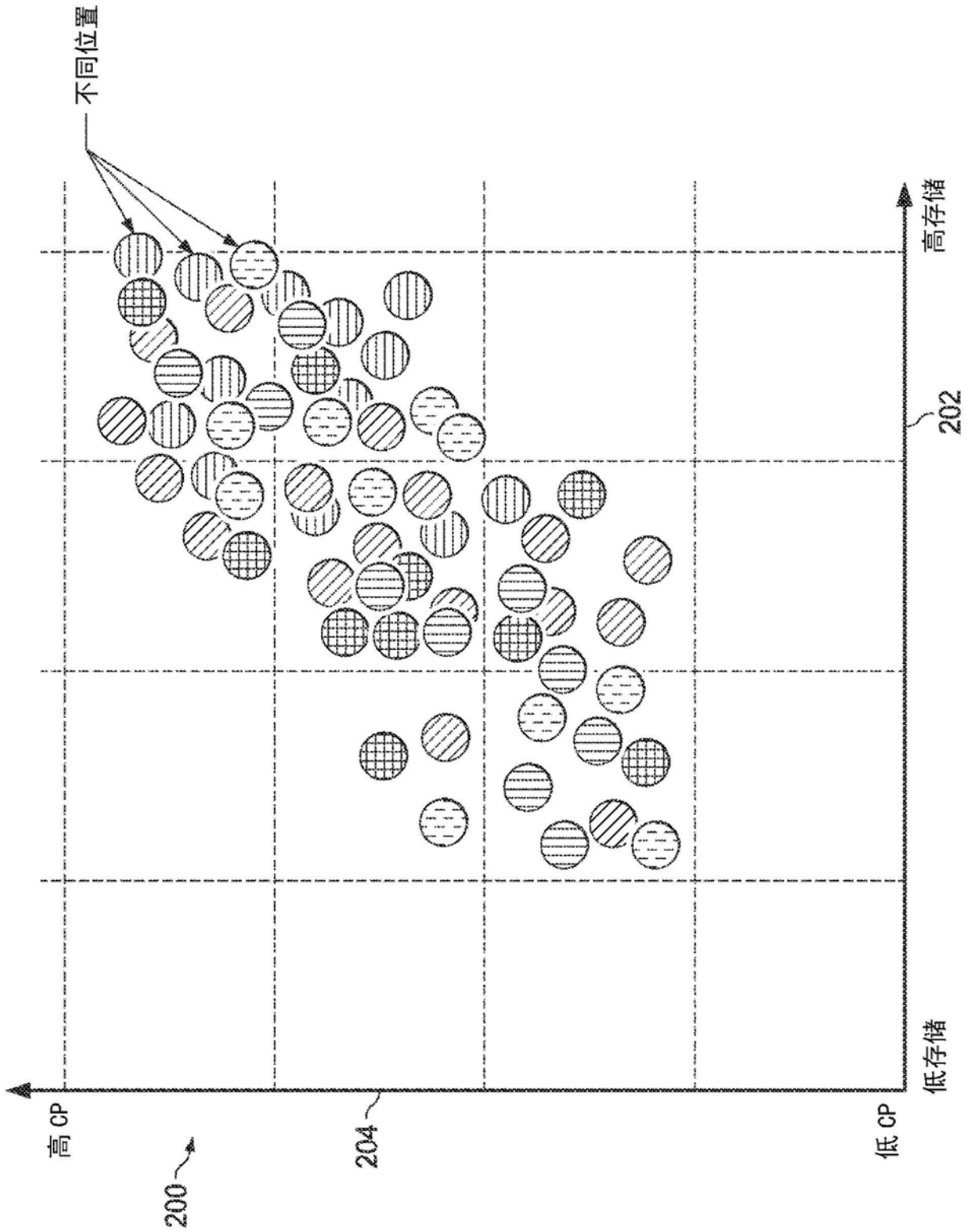


图2

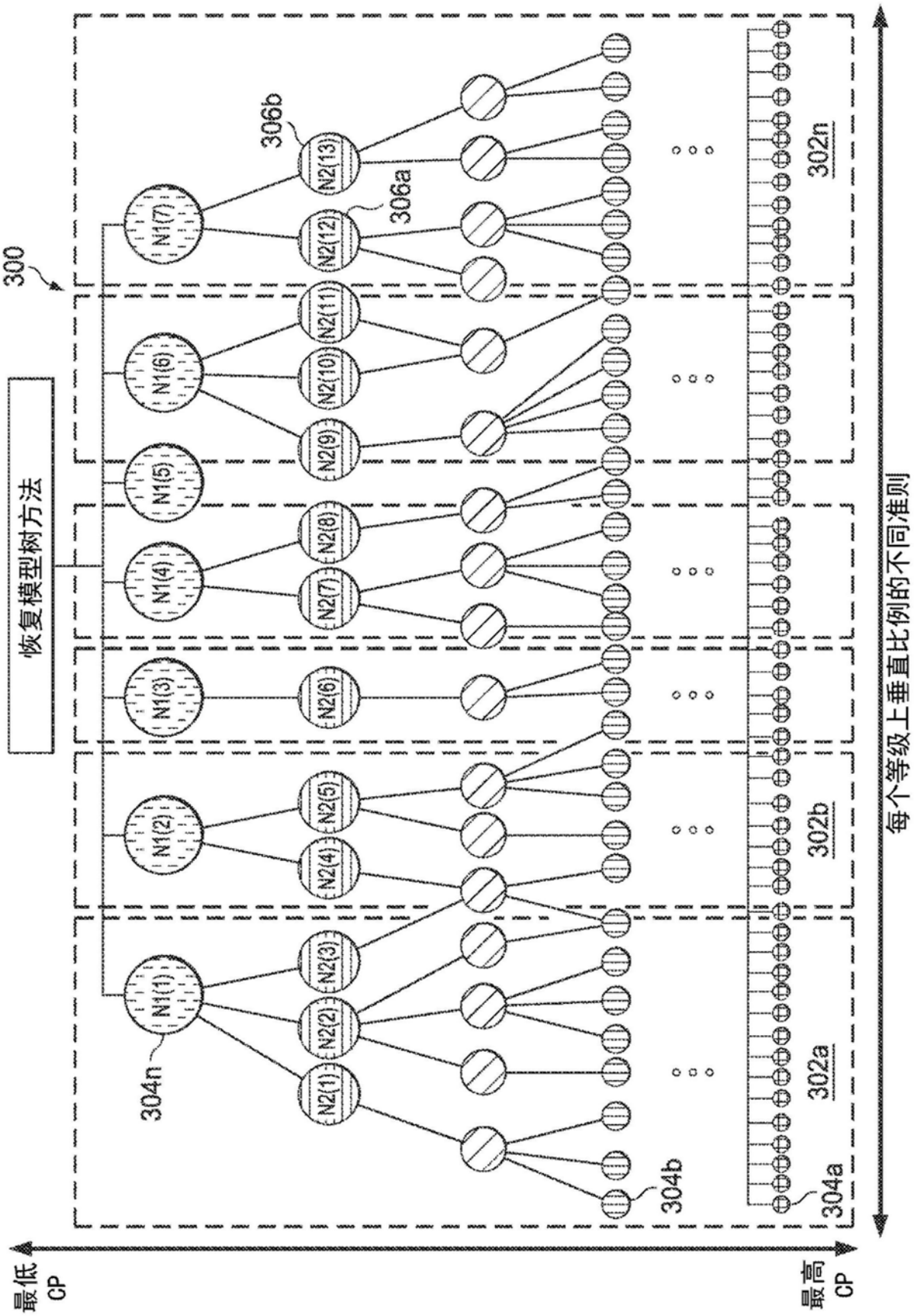

300
恢复模型树方法
N1(1)
N1(2)
N1(3)
N1(4)
N1(5)
N1(6)
N1(7)
N2(1)
N2(2)
N2(3)
N2(4)
N2(5)
N2(6)
N2(7)
N2(8)
N2(9)
N2(10)
N2(11)
N2(12)
N2(13)
304n
304b
304a
306b
306a
302a
302b
302n
最低 CP
最高 CP
每个等级上垂直比例的不同准则

图3

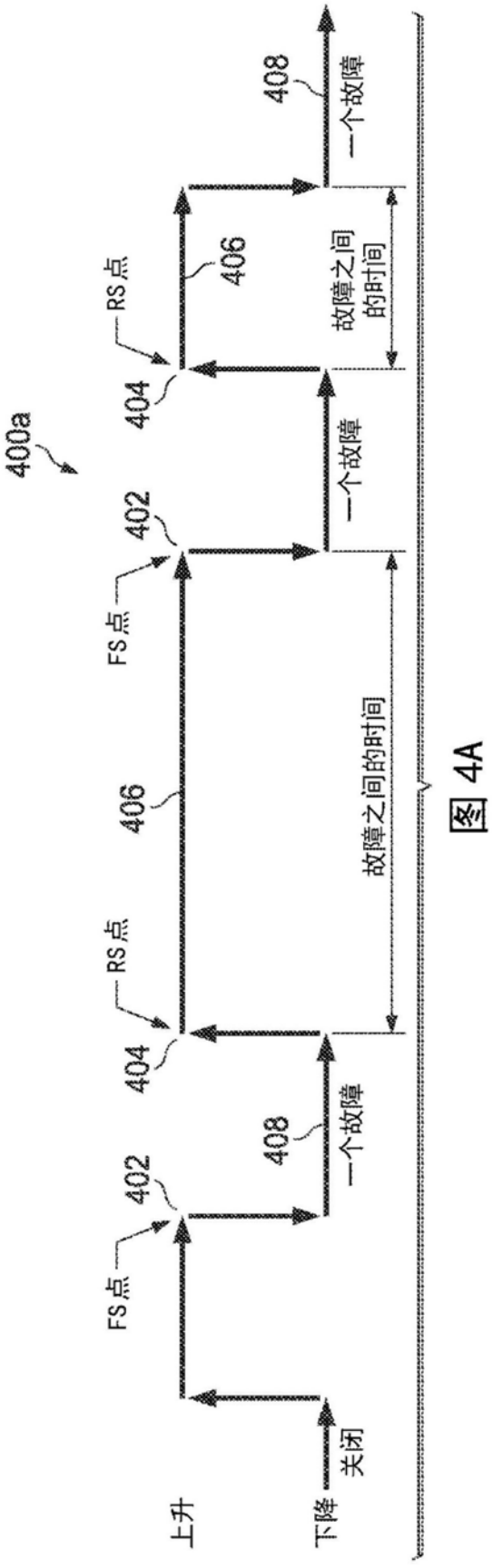


图 4A

图4A

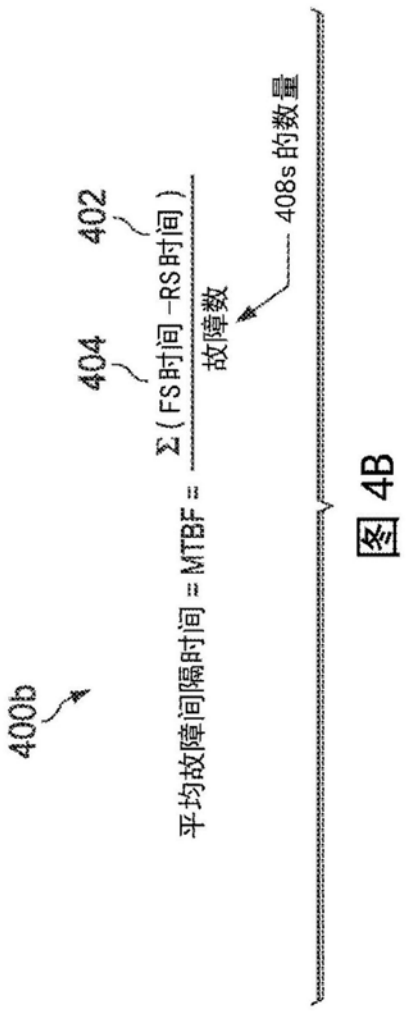


图 4B

图4B

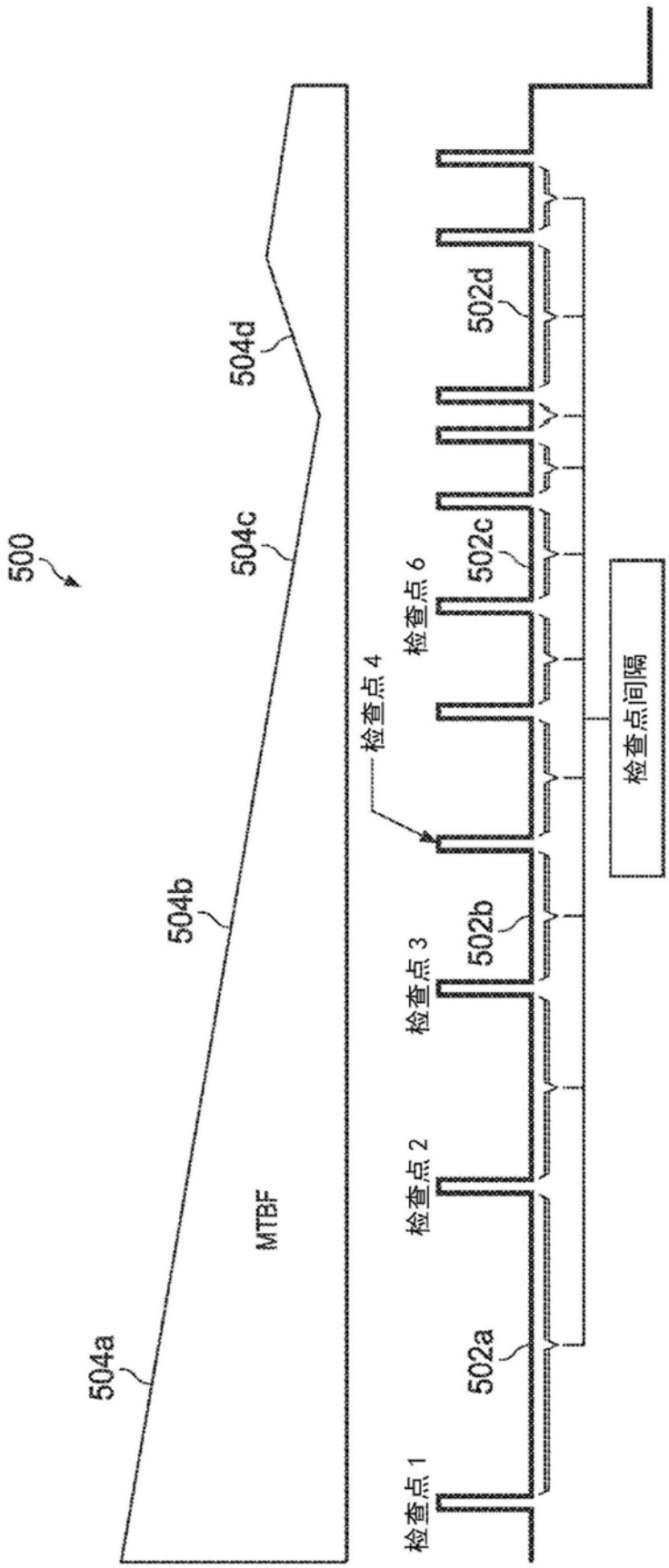

500
504a
504b
504c
504d
MTBF
检查点 1
检查点 2
检查点 3
检查点 4
检查点 6
502a
502b
502c
502d
检查点间隔

图5

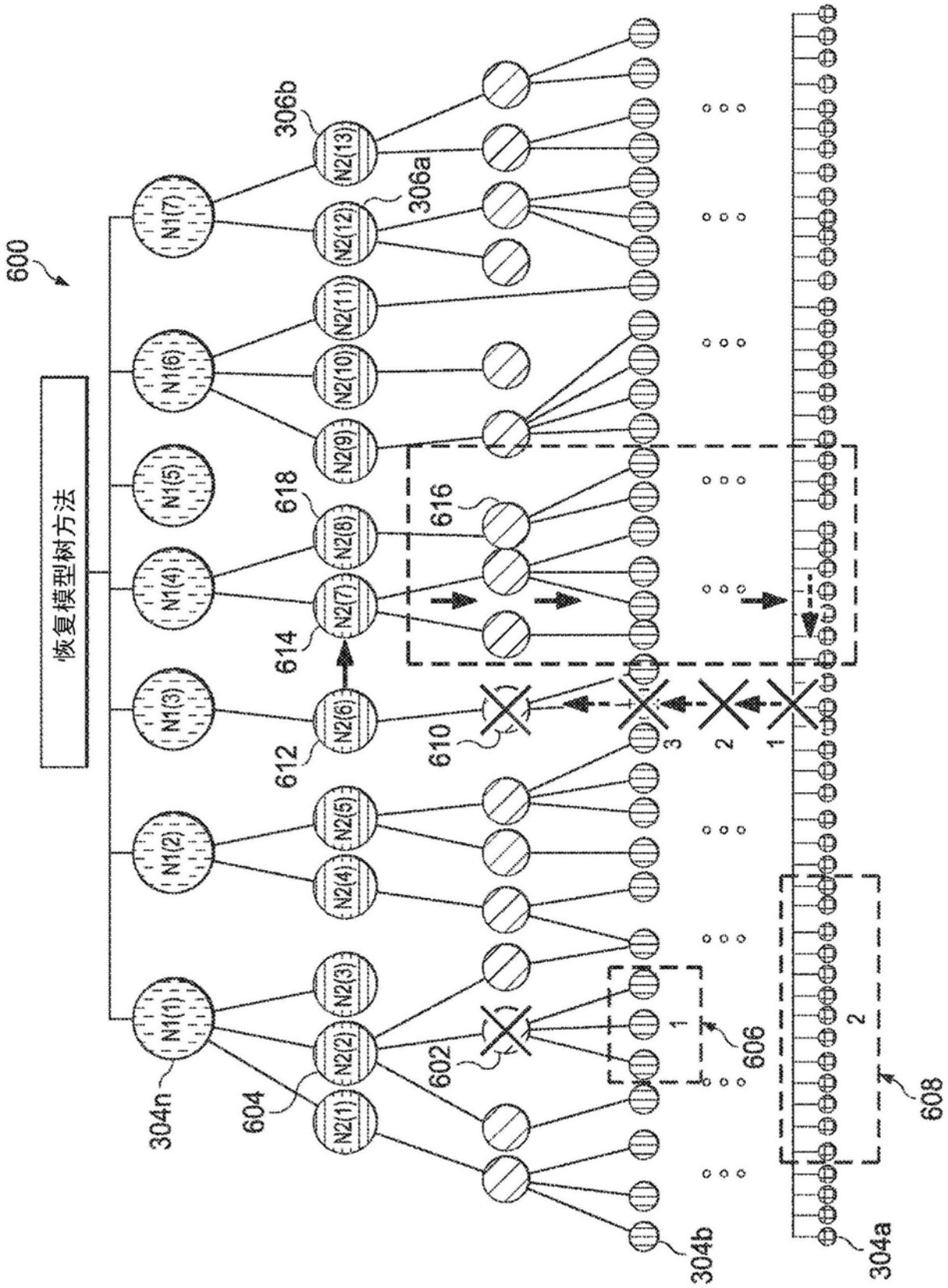

600
恢复模型树方法
N1(1)
N1(2)
N1(3)
N1(4)
N1(5)
N1(6)
N1(7)
N2(1)
N2(2)
N2(3)
N2(4)
N2(5)
N2(6)
N2(7)
N2(8)
N2(9)
N2(10)
N2(11)
N2(12)
N2(13)
304n
604
602
606
608
610
612
614
616
618
306b
306a
304b
304a
1
2
3

图6

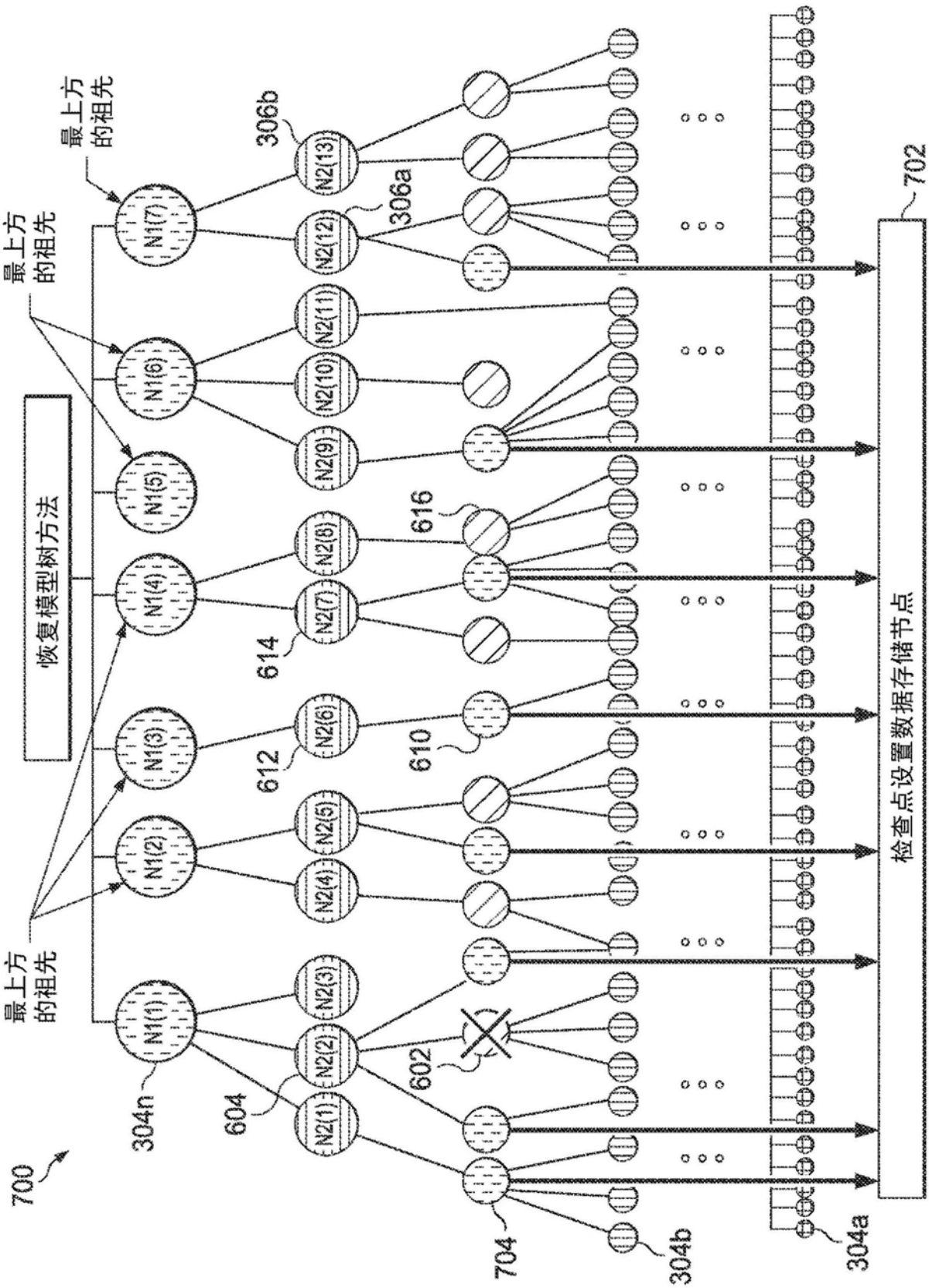

700
最上方的祖先
最上方的祖先
最上方的祖先
恢复模型树方法
N1(1)
N1(2)
N1(3)
N1(4)
N1(5)
N1(6)
N1(7)
N2(1)
N2(2)
N2(3)
N2(4)
N2(5)
N2(6)
N2(7)
N2(8)
N2(9)
N2(10)
N2(11)
N2(12)
N2(13)
304n
604
602
612
610
614
616
306a
306b
704
304b
304a
702
检查点设置数据存储节点

图7

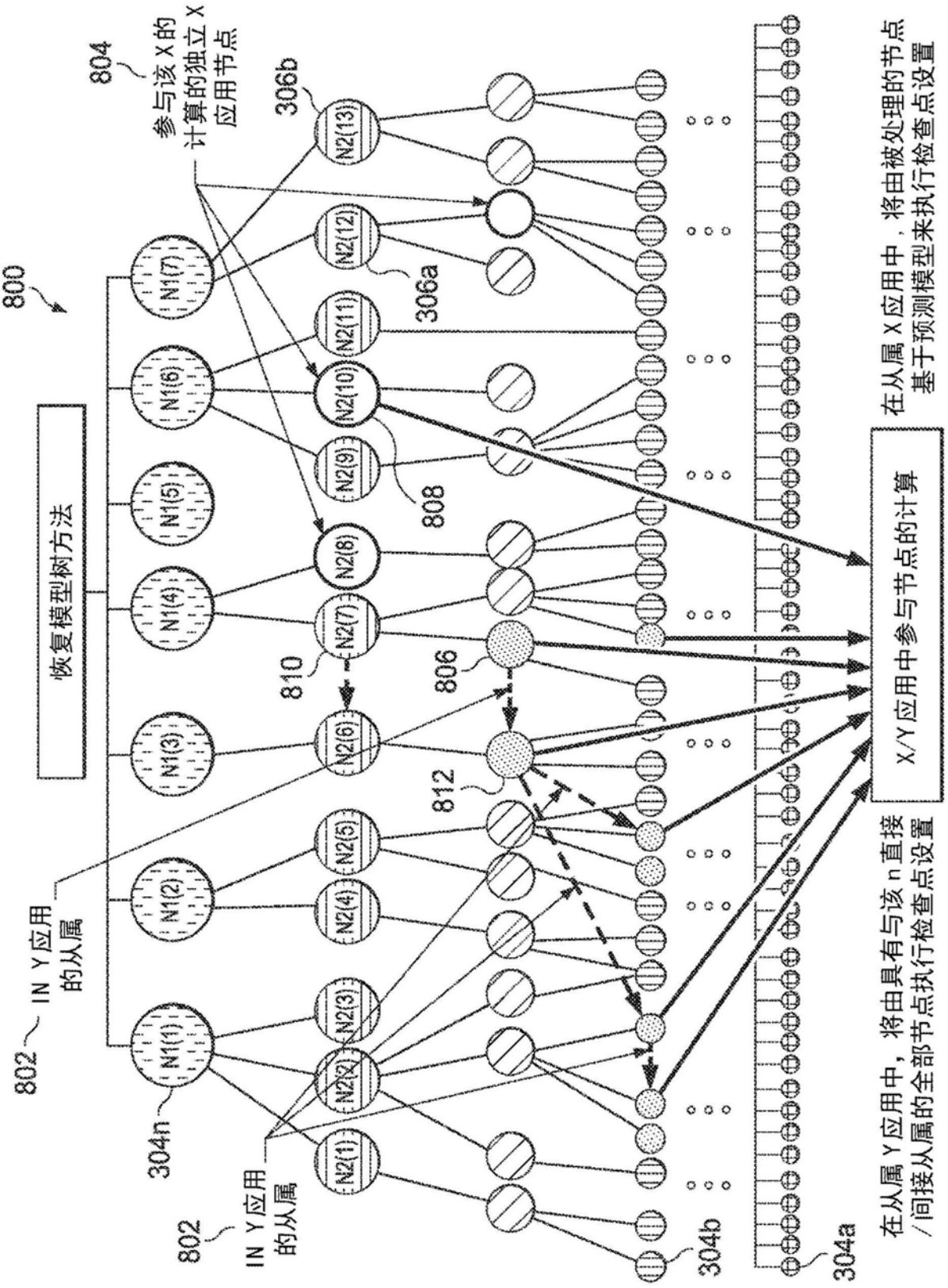

800
恢复模型树方法
802
IN Y应用的从属
804
参与该X的计算的独立X应用节点
306b
306a
808
810
806
812
304n
304b
304a
X/Y应用中参与节点的计算
在从属X应用中，将由被处理的节点基于预测模型来执行检查点设置
在从属Y应用中，将由具有与该n直接/间接从属的全部节点执行检查点设置

图8

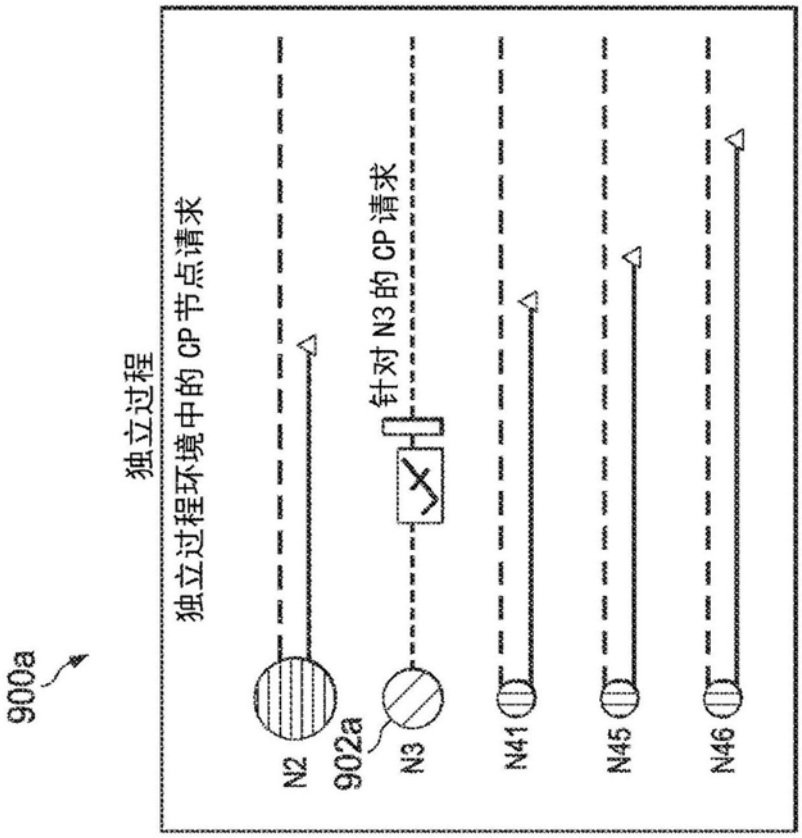


图9A

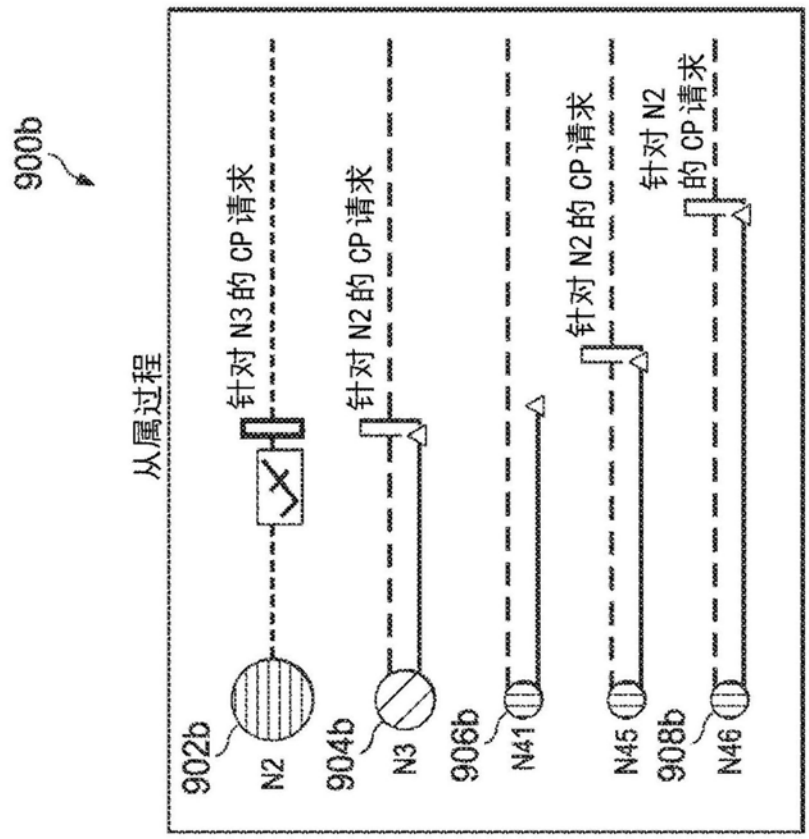

900b
从属过程
针对 N3 的 CP 请求
针对 N2 的 CP 请求
针对 N2 的 CP 请求
针对 N2 的 CP 请求
902b
N2
904b
N3
906b
N41
N45
908b
N46

图9B

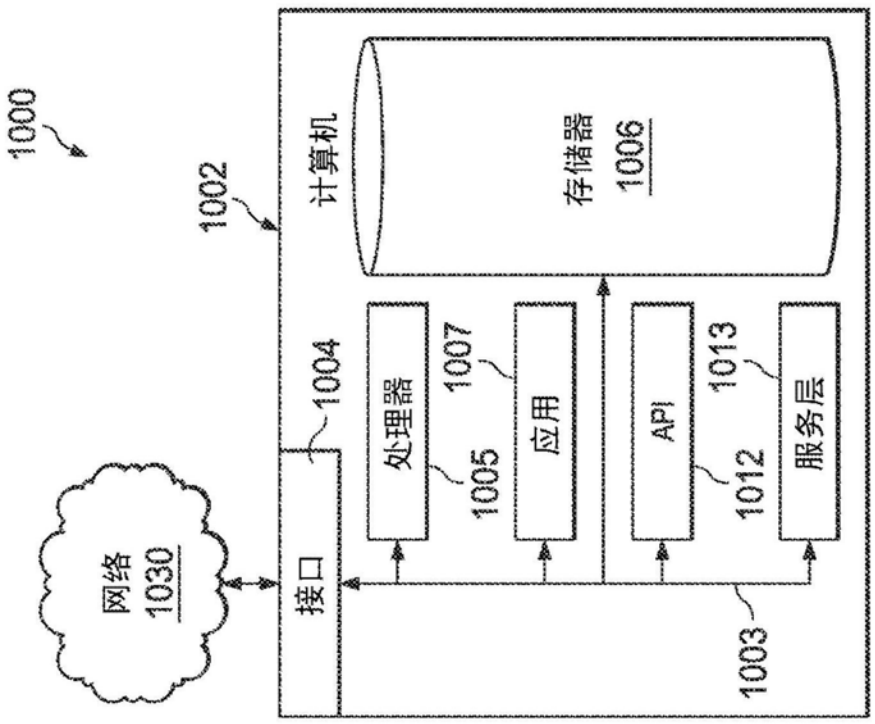

1000
1002
计算机
存储器 1006
网络 1030
接口
1004
处理器
1005
1007
应用
API
1012
1013
服务层
1003

图10