(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 105830029 B
(45)授权公告日 2019.12.17

(21)申请号 201480068927.7

(22)申请日 2014.12.04

(65)同一申请的已公布的文献号
申请公布号 CN 105830029 A

(43)申请公布日 2016.08.03

(30)优先权数据
61/917,709 2013.12.18 US
14/449,357 2014.08.01 US

(85)PCT国际申请进入国家阶段日
2016.06.17

(86)PCT国际申请的申请数据
PCT/US2014/068661 2014.12.04

(87)PCT国际申请的公布数据
WO2015/094704 EN 2015.06.25

(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72)发明人 M•A•法尔科

(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 李晓芳

(51)Int.Cl.
G06F 9/48(2006.01)
G06F 9/54(2006.01)

(56)对比文件
CN 103294531 A,2013.09.11,
CN 1601477 A,2005.03.30,
Andrea C.等.Scheduling with Implicit Information in Distributed Systems.《SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems》.1998,第233-243页.
Andrea C.等.Scheduling with Implicit Information in Distributed Systems.《SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems》.1998,第233-243页.

审查员 熊沐阳

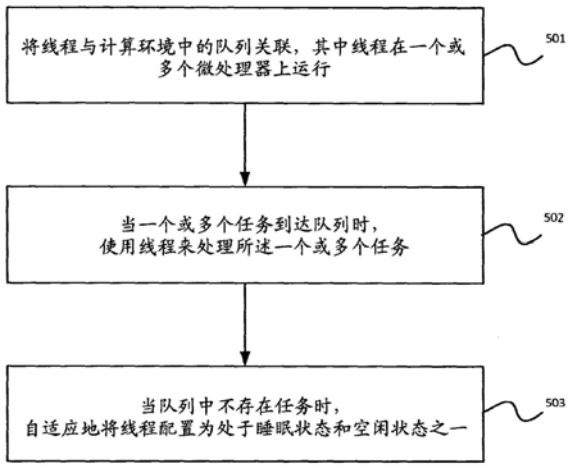
权利要求书2页 说明书5页 附图7页

(54)发明名称

用于在计算环境中支持自适应忙等待的系统和方法

(57)摘要

一种可以在诸如分布式数据网格之类的计算环境中支持队列处理的系统和方法。线程可以与计算环境中的队列关联,其中线程在支持中央处理单元(CPU)的一个或多个微处理器上运行。当一个或多个任务到达队列时,系统可以使用线程来处理所述一个或多个任务。此外,当队列中不存在任务时,系统可以自适应地将线程配置为处于睡眠状态和空闲状态之一。



CN 105830029 B

1.一种用于在计算环境中支持队列处理的方法,包括:

将单个线程与计算环境中的队列关联,其中所述线程在一个或多个微处理器上运行;

如果所述队列不为空,使用所述线程来处理来自所述队列的下一个任务;

如果所述队列为空,等待将被所述队列接收的下一个任务;以及

响应于所述队列的工作负荷,在等待期间自适应地将所述线程配置为处于睡眠状态和空闲状态之一。

2.如权利要求1所述的方法,其中:

所述一个或多个微处理器支持一个或多个中央处理单元CPU。

3.如权利要求2所述的方法,还包括:

当所述线程处于睡眠状态时,允许所述一个或多个CPU切换到另一个线程,以及

当任务到达队列时,将所述线程从睡眠状态唤醒。

4.如权利要求2或3所述的方法,还包括:

基于操作系统OS函数,获得用于将所述线程置于睡眠状态以及将所述线程从睡眠状态唤醒的总成本。

5.如权利要求2或3所述的方法,还包括:

当所述线程处于空闲状态时,允许所述线程使所述一个或多个CPU自旋。

6.如权利要求2或3所述的方法,还包括:

基于队列上的工作负荷,获得用于将所述线程置于空闲状态的成本。

7.如权利要求1-3中任何一项所述的方法,还包括:

比较用于将所述线程置于睡眠状态和将所述线程从睡眠状态唤醒的总成本与用于将所述线程置于空闲状态的成本。

8.如权利要求7所述的方法,还包括:

如果用于将所述线程置于睡眠状态的总成本小于用于将所述线程置于空闲状态的成本,则将所述线程配置为处于睡眠状态;以及

如果相反的情况,则将所述线程配置为处于空闲状态。

9.如权利要求1-3中任何一项所述的方法,还包括:

提供线程池,其中所述线程池包括在一个或多个微处理器上运行的多个线程。

10.如权利要求9所述的方法,还包括:

允许每个所述线程独立地确定它应当处于睡眠状态还是处于空闲状态。

11.一种用于在计算环境中支持队列处理的装置,包括用于执行如权利要求1-10中任何一项所述的方法的各个步骤的单元。

12.一种具有存储在其上的指令的非暂时性机器可读存储介质,当指令被执行时,使系统执行如权利要求1至10中的任何一项所述的方法的步骤。

13.一种用于在分布式数据网格中支持任务处理的系统,该系统包括:

包括一个或多个微处理器和存储器的计算机系统;

在所述存储器中的队列;

线程,所述线程在所述一个或多个微处理器上运行,其中所述线程与计算环境中的所述队列关联,其中所述线程被配置为:

如果所述队列不为空,处理来自所述队列的下一个任务;

如果所述队列为空,等待将被所述队列接收的下一个任务;以及

在等待下一个任务的同时进入睡眠状态和空闲状态之一,其中,睡眠状态和空闲状态之间的选择是响应于所述队列的工作负荷而执行的。

14.如权利要求13所述的系统,其中:

所述一个或多个微处理器支持一个或多个中央处理单元CPU。

15.如权利要求14所述的系统,其中:

当所述线程处于睡眠状态时,所述一个或多个CPU被允许切换到另一个线程,以及

当任务到达队列时,所述线程从睡眠状态被唤醒。

16.如权利要求14或15所述的系统,其中:

所述线程操作以基于操作系统OS函数获得用于将所述线程置于睡眠状态和将所述线程从睡眠状态唤醒的总成本。

17.如权利要求14或15所述的系统,其中:

当所述线程处于空闲状态时,所述线程操作以使得所述一个或多个CPU自旋。

18.如权利要求14或15所述的系统,其中:

所述线程操作以基于队列上的工作负荷获得用于将所述线程置于空闲状态的成本。

19.如权利要求13至15中任何一项所述的系统,其中:

所述线程操作以比较用于将所述线程置于睡眠状态和将所述线程从睡眠状态唤醒的总成本与用于将所述线程置于空闲状态的成本。

20.如权利要求19所述的系统,其中:

当用于将所述线程置于睡眠状态的总成本小于用于将所述线程置于空闲状态的成本时,所述线程被配置为处于睡眠状态;以及

如果相反的情况,则所述线程被配置为处于空闲状态。

21.如权利要求13至15中任何一项所述的系统,还包括:

线程池,其中所述线程池包括在所述一个或多个微处理器上运行的多个线程,并且其中每个所述线程被允许独立地确定它应当处于睡眠状态还是处于空闲状态。

用于在计算环境中支持自适应忙等待的系统和方法

[0001] 版权通知

[0002] 本专利文档的公开内容的一部分包含受版权保护的材料。版权所有者不反对任何人对专利文档或专利公开内容按照在专利商标局的专利文件或记录中出现的那样进行复制再现，但是无论如何在其他方面保留全部的版权权利。

技术领域

[0003] 本发明一般涉及计算机系统，并且特别地涉及计算环境中的线程管理。

背景技术

[0004] 现代计算系统(尤其是较大的组织和企业所采用的那些现代计算系统)继续在规模和复杂度上增加。特别地，在诸如因特网应用之类的领域中，存在着数百万的用户应当能够同时访问该应用的期望，这实际上导致由用户生成和消费的内容量以及涉及这些内容的事务呈指数增长。这样的活动还导致对数据库和元数据存储库的事务调用的数量的相应增加，而数据库和元数据存储库具有有限的容量来适应那种需求。

[0005] 这是本发明的实施例旨在应对的一般领域。

发明内容

[0006] 本文描述的是可以在诸如分布式数据网格之类的计算环境中支持队列处理的系统和方法。线程可以与计算环境中的队列关联，其中线程在支持中央处理单元(CPU)的一个或多个微处理器上运行。当一个或多个任务到达队列时，系统可以使用该线程来处理所述一个或多个任务。此外，当队列中不存在任务时，系统可以自适应地将线程配置为处于睡眠状态和空闲状态之一。

附图说明

[0007] 图1是根据本发明的各种实施例的数据网格集群的图示。

[0008] 图2示出了根据本发明的实施例的在计算环境中支持队列处理的图示。

[0009] 图3示出了根据本发明的实施例的利用自适应忙等待方法来管理计算环境中的线程的图示。

[0010] 图4示出了根据本发明的实施例的利用线程池在计算环境中支持队列处理的图示。

[0011] 图5示出了根据本发明的实施例的用于使用自适应忙等待方法来管理计算环境中的线程的示例性流程图。

[0012] 图6示出了本发明的实施例的功能性配置。

[0013] 图7示出了可以在其上实现本发明的实施例的计算机系统。

具体实施方式

[0014] 本文描述的是可以在计算环境中支持队列处理的系统和方法。

[0015] 分布式数据网格

[0016] 根据实施例，在本文中提到的“数据网格集群”或“数据网格”是包括多个计算机服务器的系统，该多个计算机服务器一起工作来管理分布式或集群环境内的信息和诸如计算之类的相关操作。可以使用数据网格集群来管理跨服务器共享的应用对象和数据。优选地，数据网格集群应当具有低响应时间、高吞吐量、可预测的可缩放性、连续的可用性和信息可靠性。作为这些能力的结果，数据网格集群非常适于在计算密集、有状态的中间层应用中使用。数据网格集群的一些例子(例如，Oracle Coherence数据网格集群)可以将信息存储在存储器中以实现较高的性能，并且可以在使该信息的副本跨多个服务器保持同步时采用冗余，从而在服务器故障的情况下确保系统的弹性以及数据的可用性。例如，Oracle Coherence数据网格集群在可靠的、高度可缩放的对等集群协议之上提供重复式并且分布式的(分区的)数据管理和高速缓存服务。

[0017] 存储器中(in-memory)数据网格可以通过将数据分布在多个一起工作的服务器上来提供数据存储和管理能力。数据网格可以是与应用服务器在同一层中运行或者在应用服务器内运行的中间件。它可以提供数据的管理和处理，并且还可以将处理推送到网格中数据所处的地方。此外，当服务器变得不可操作或从网络断开连接时，存储器中数据网格可以通过自动地和透明地故障转移以及重新分配其集群的数据管理服务来消除各单一故障点。当新的服务器被添加时，或者当故障的服务器被重新启动时，它可以自动地加入集群，并且服务可以被恢复给它，从而透明地重新分布集群负荷。数据网格还可以包括网络级容错特征和透明的软重启能力。

[0018] 根据实施例，数据网格集群的功能是基于使用不同的集群服务。集群服务可以包括根集群服务、分区的高速缓存服务和代理服务。在数据网格集群内，在提供集群服务和消费集群服务两个方面，每个集群节点可以参与多个集群服务。每个集群服务具有服务名称和服务类型，其中服务名称唯一地标识在数据网格集群内的服务，服务类型定义集群服务可以做什么。除了在数据网格集群中的每个集群节点上运行的根集群服务之外，可以存在每种服务类型的多个命名实例。服务可以或者由用户配置，或者由数据网格集群提供作为默认的服务集合。

[0019] 图1是根据本发明的各种实施例的数据网格集群的图示。如在图1中所示，例如Oracle Coherence数据网格集群的数据网格集群100包括多个集群节点101-106，该多个集群节点101-106具有在其上运行的各种集群服务111-116。此外，可以使用高速缓存配置文件110来配置该数据网格集群100。

[0020] 队列处理等待

[0021] 图2示出了根据本发明的实施例的在计算环境中支持队列处理的图示。如图2中所示，诸如分布式数据网格之类的计算环境200可以使用一个或多个线程(例如，线程211)来支持队列处理。线程211可以负责处理到达队列201的各种任务，例如，以便支持基于分布式数据网格的消息传送总线。

[0022] 此外，当队列201中不存在任务时，系统允许线程211等待直到下一个任务(例如，任务203)到达队列201。在等待任务203到达的时候，线程211可被配置为处于不同的状态

(即,使用不同的策略)。

[0023] 根据本发明的实施例,线程211可以被配置为处于空闲状态,其中系统可以使CPU 202自旋(即,CPU不会被切换成运行另一个线程)。通过在等待的时候使CPU 202自旋,新任务203一到达队列201,系统就可以使用线程211来处理任务203。因此,系统可以实现低延迟时间和高性能。

[0024] 作为折衷,当线程211处于空闲状态时,CPU时间会被浪费,因为当CPU 202自旋时,CPU 202不会被切换成运行其它线程。因此,当工作负荷重并且两个连续任务之间的间隔时间趋于短时,这种方法是有利的。

[0025] 作为替代,线程211可以被置于睡眠状态,在这种情况下,CPU202被允许切换成运行另一个线程,直到例如当任务203到达队列101时,线程211被唤醒。

[0026] 另一方面,利用这第二种方法,系统需要执行两个操作(即,将线程211置为睡眠的第一操作以及将线程211唤醒的第二操作)以便使用线程211进行队列处理。

[0027] 因此,当两个连续任务之间的间隔时间长于系统将线程211置为睡眠以及随后当下一个任务203到达时唤醒它所花费的总CPU时间时,该方法是有利的。

[0028] 此外,系统可以利用高性能网络,诸如由Coherence数据网格集群使用的InfiniBand(IB)网络。其结果是,总的延迟时间可以显著地减小。在这种情况下,执行这两个操作所花费的CPU时间可以贡献总延迟时间中的很大的一部分。

[0029] 自适应忙等待

[0030] 图3示出了根据本发明的实施例的利用自适应忙等待方法来管理计算环境中的线程的图示。如图3中所示,线程311可以用于在计算环境300中处理队列301。

[0031] 当队列301为空时,系统可以使用自适应忙等待方法来确定应当使用哪种策略来配置在等待新任务到达队列301的时候的线程311(即,决定是将线程311置为睡眠还是使CPU 302自旋)。

[0032] 利用自适应忙等待方法,系统可以获得将线程311置为睡眠以及随后唤醒它两者可能需要的总CPU时间。然后,系统可以首先将用于把线程311置于睡眠状态和从睡眠状态唤醒线程311的总成本与通过使CPU 302自旋而将线程置于空闲状态的成本进行比较。

[0033] 对应地,如果用于将线程311置于睡眠状态的总成本小于用于将线程311置于空闲状态的成本,则系统可以将线程311配置为处于睡眠状态。如果相反的情况,系统可以将线程311配置为处于空闲状态。

[0034] 如图3中所示,系统可以利用由操作系统(OS)303提供的函数(function)313来确定用于执行操作的成本。例如,在CPU 302之上运行的OS函数313可以提供关于线程311已经消耗了多少CPU时间的信息。系统可以通过将执行操作前后由线程311消耗的总CPU时间进行比较来估计用于执行操作的成本。

[0035] 此外,系统可以估计系统使CPU 302自旋而不是将线程311置为睡眠可能需要的CPU时间。例如,系统可以基于两个连续任务到达队列301之间的平均间隔时间是多长来获得这样的信息。

[0036] 根据本发明的实施例,系统可以基于队列301上的工作负荷来自适应地配置线程311。例如,当队列301上的工作负荷轻时,系统可以将线程311配置为在等待新任务的时候处于睡眠状态。然后,当工作负荷变重时,用于保持CPU自旋所需要的CPU时间可以显著地减

少。

[0037] 因此，系统可以确定保持CPU 302空闲(或自旋)实际上更便宜，而不是经历将线程311置为睡眠并且随后在新任务到达队列301之后唤醒它的过程更便宜。随后，当工作负荷之后再次变轻时，系统可以将线程311重新配置为在等待新任务的时候处于睡眠状态。

[0038] 此外，代替使用自适应忙等待方法，系统可以设置阈值，该阈值定义有多少CPU时间被允许能够在自旋中(即，在空闲状态中)浪费。否则，当用于自旋的CPU时间超过该阈值时，系统可以将线程311设置为处于睡眠状态。

[0039] 因此，这种方法的缺点在于，始终如一的阈值对于动态队列处理系统来说可能并不总是优化的。即使在队列301上的工作负荷轻并且线程311可以被切换成运行其它任务时，始终如一的阈值也无法阻止用于自旋的CPU时间的一部分被浪费。

[0040] 图4示出了根据本发明的实施例的利用线程池在计算环境中支持队列处理的图示。如图4中所示，计算环境400中的线程池410可以包含在CPU 420上运行的多个线程，例如线程A-C 411-413。此外，线程池410可以与例如队列A-C 401-403的多个队列关联。

[0041] 此外，线程A-C 411-413可以被用来处理到达队列A-C 401-403的任务。对于线程池410中的每个线程A-C 411-413，系统可以独立地确定在等待新任务到达队列A-C 401-403的时候是使CPU 420自旋还是将线程置为睡眠。

[0042] 因此，系统可以允许线程A-C 411-413基于队列A-C 401-403上的不同工作负荷而被不同地进行配置。

[0043] 图5示出了根据本发明的实施例的用于使用自适应忙等待方法来管理计算环境中的线程的示例性流程图。如图5中所示，在步骤501，系统可以将线程与计算环境中的队列关联，其中线程在一个或多个微处理器上运行。然后，在步骤502，当一个或多个任务到达队列时，系统可以使用该线程来处理所述一个或多个任务。此外，在步骤503，当队列中不存在任务时，系统可以自适应地将线程配置为处于睡眠状态和空闲状态之一。

[0044] 参考图6，示出了本发明的实施例的系统600。图6示出了由系统600实现的功能性配置的图示。系统600包括计算模块610、管理器620、处理器630、任务处理器640和配置模块650。

[0045] 计算模块610向管理器620提供队列。管理器620将线程与计算模块610中的队列关联。该线程在诸如处理器630之类的一个或多个微处理器上运行。当一个或多个任务到达队列时，任务处理器640使用该线程来处理所述一个或多个任务。当队列中不存在任务时，配置模块650自适应地将该线程配置为处于睡眠状态和空闲状态之一。

[0046] 图7示出了包括众所周知的硬件元件的计算机系统700的图示。即，计算机系统700包括中央处理单元(CPU)710、鼠标720、键盘730、随机存取存储器(RAM)740、硬盘750、盘驱动器760、通信接口(I/F)770、以及监视器780。计算机系统700可以充当构成系统600的服务器节点。

[0047] 根据本发明的实施例，计算模块610、管理器620、处理器630、任务处理器640和配置模块650由一个或多个计算机系统700提供。计算模块610、管理器620、处理器630、任务处理器640和配置模块650由CPU 710实现。在另一方面，多于一个的处理器可以被使用，使得计算模块610、管理器620、处理器630、任务处理器640和配置模块650能够被实现。即，计算模块610、管理器620、处理器630、任务处理器640和配置模块650当中的任何一个可以彼此

物理地远离。

[0048] 在还有另一方面中，系统600可以通过使用充当计算模块610、管理器620、处理器630、任务处理器640和配置模块650的多个硬连线电路来实现。

[0049] 可以通过使用一个或多个常规的通用或专用数字计算机、计算设备、机器或微处理器而方便地实现本发明，其中所使用的数字计算机、计算设备、机器或微处理器包括根据本公开内容的教导编程的一个或多个处理器、存储器和/或计算机可读存储介质。熟练的程序员基于本公开内容的教导可以容易地准备出适当的软件编码，这对于软件领域的技术人员将是显而易见的。

[0050] 在一些实施例中，本发明包括计算机程序产品，它是具有存储于其上/其中的指令的存储介质或计算机可读介质，所述指令可以用于对计算机进行编程以执行本发明中的任何过程。存储介质可以包括但不限于任何类型的盘，包括软盘、光盘、DVD、CD-ROM、微驱动器、以及磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪速存储器设备、磁卡或光卡、纳米系统(包括分子存储器IC)、或者适于存储指令和/或数据的任何类型的介质或设备。

[0051] 本发明的以上描述是为了说明和描述的目的而给出的。它不意图穷举或者将本发明限定到所公开的精确形式。对本领域执业人员来说，许多修改和变化将是显而易见的。所述修改和变化包括所描述的特征的任何相关组合。实施例的选择和描述是为了最好地解释本发明的原理及其实践应用，从而针对各种实施例以及适于预想到的特定用途的各种修改的情况，使得本领域的其他技术人员能够理解本发明。本发明的范围旨在由以下的权利要求及其等价物来限定。

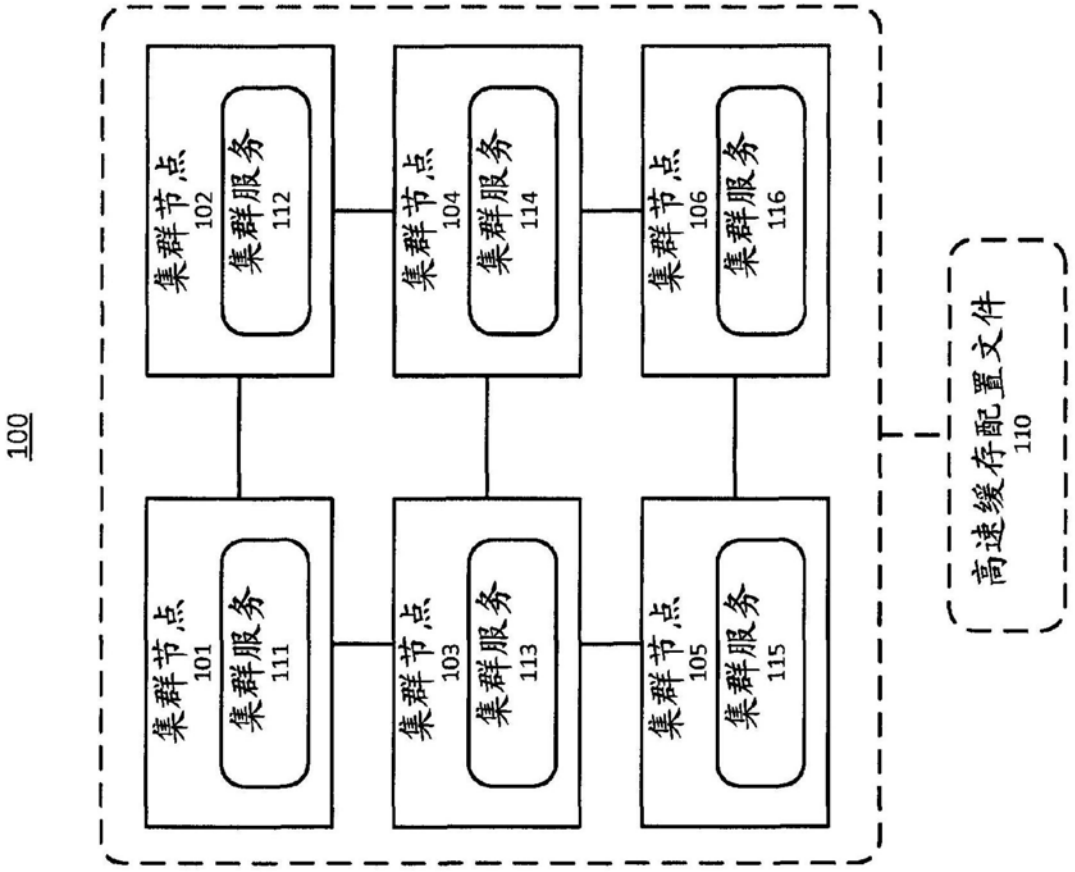

100
集群节点
101
集群服务
111
集群节点
102
集群服务
112
集群节点
103
集群服务
113
集群节点
104
集群服务
114
集群节点
105
集群服务
115
集群节点
106
集群服务
116
高速缓存配置文件
110

图1

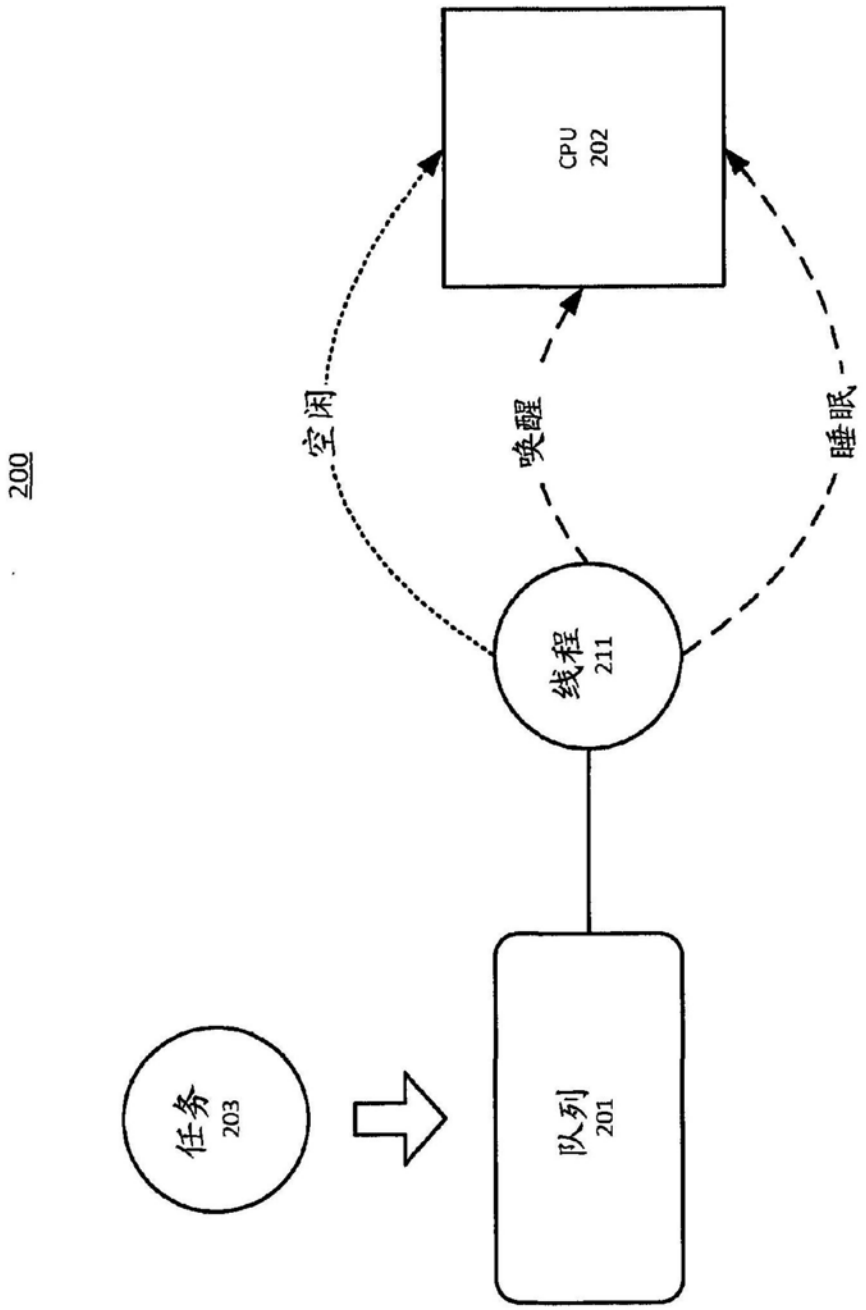


图2

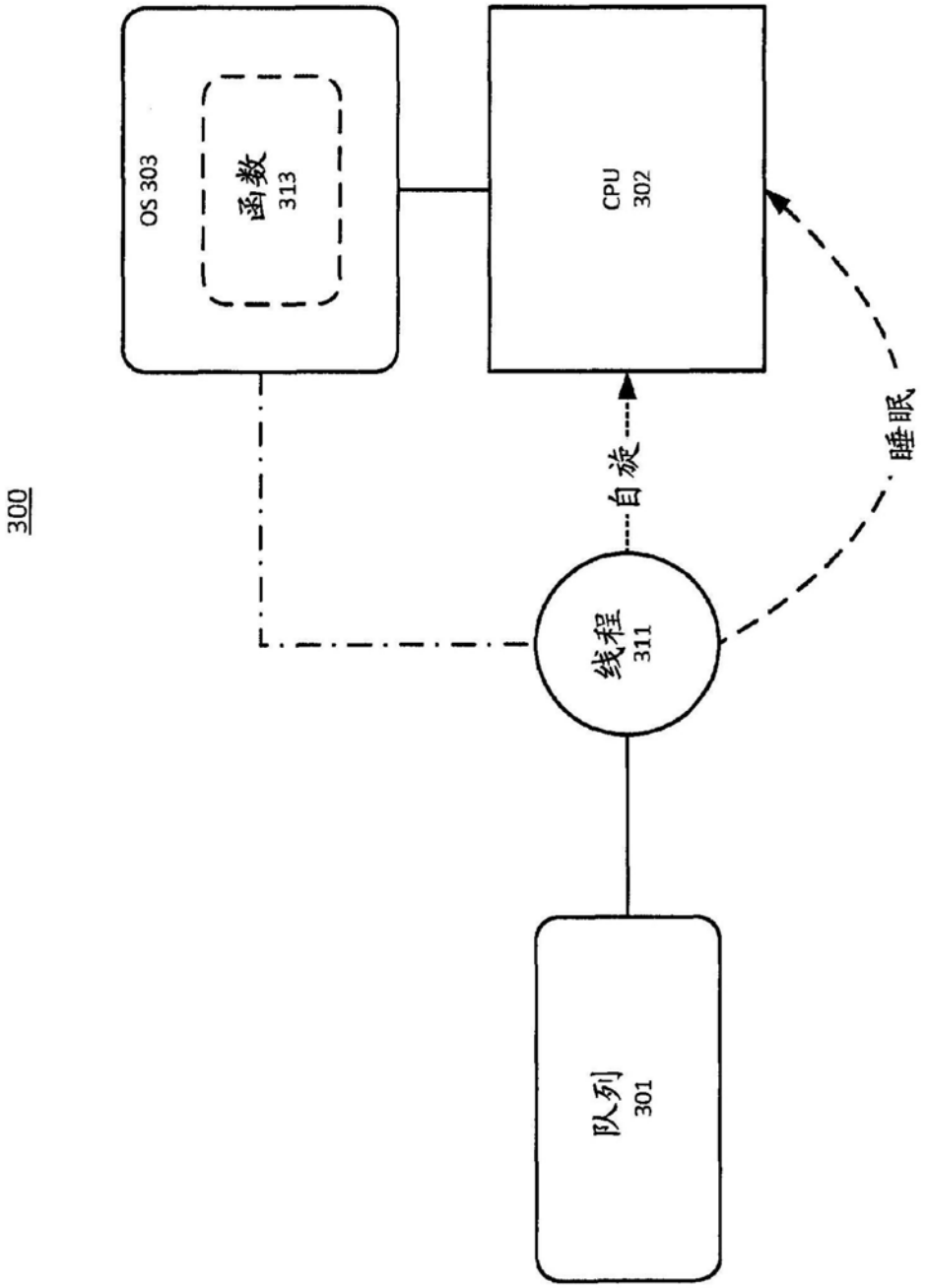


图3

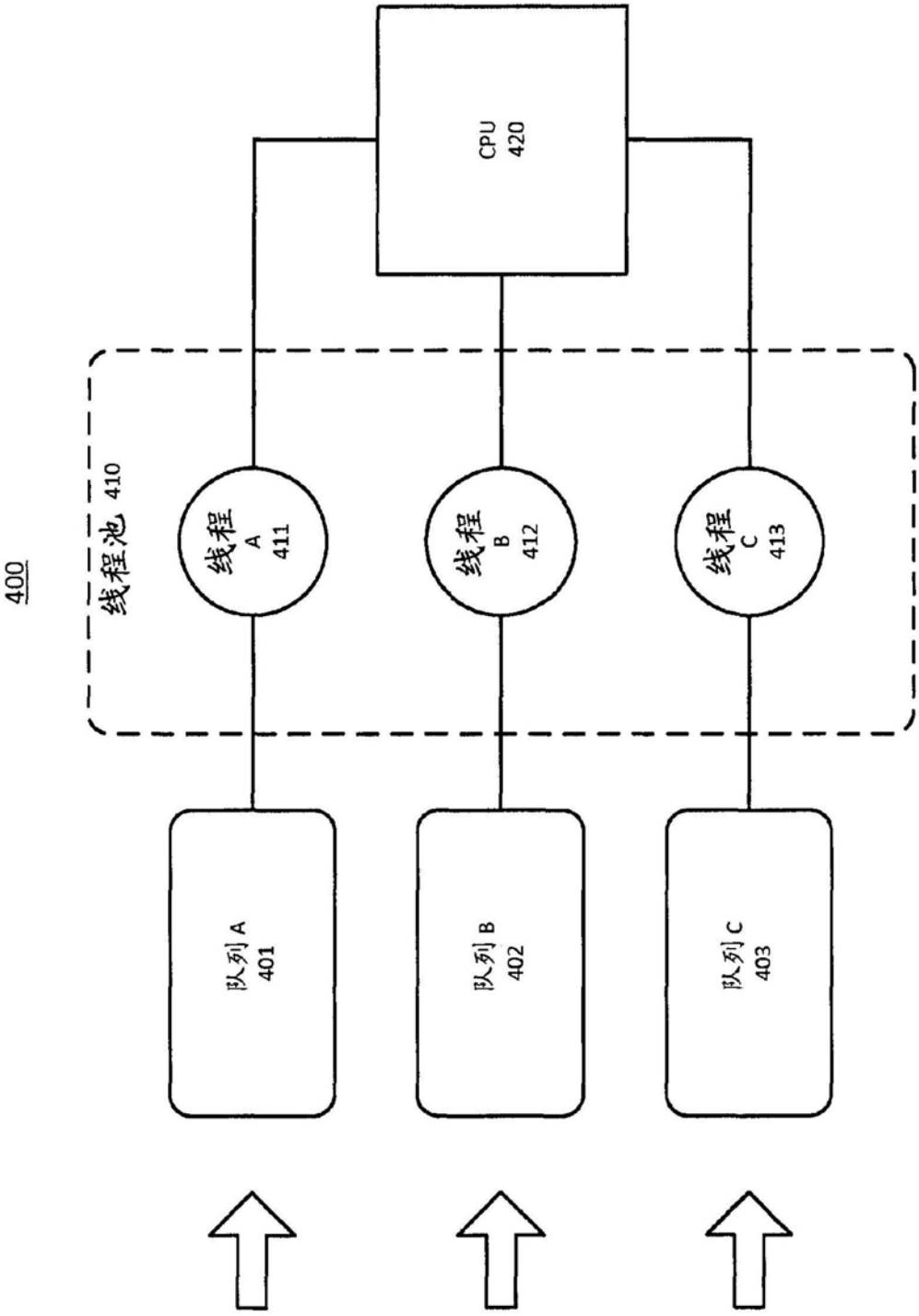

400
CPU
420
线程池 410
线程
A
411
线程
B
412
线程
C
413
队列 A
401
队列 B
402
队列 C
403

图4

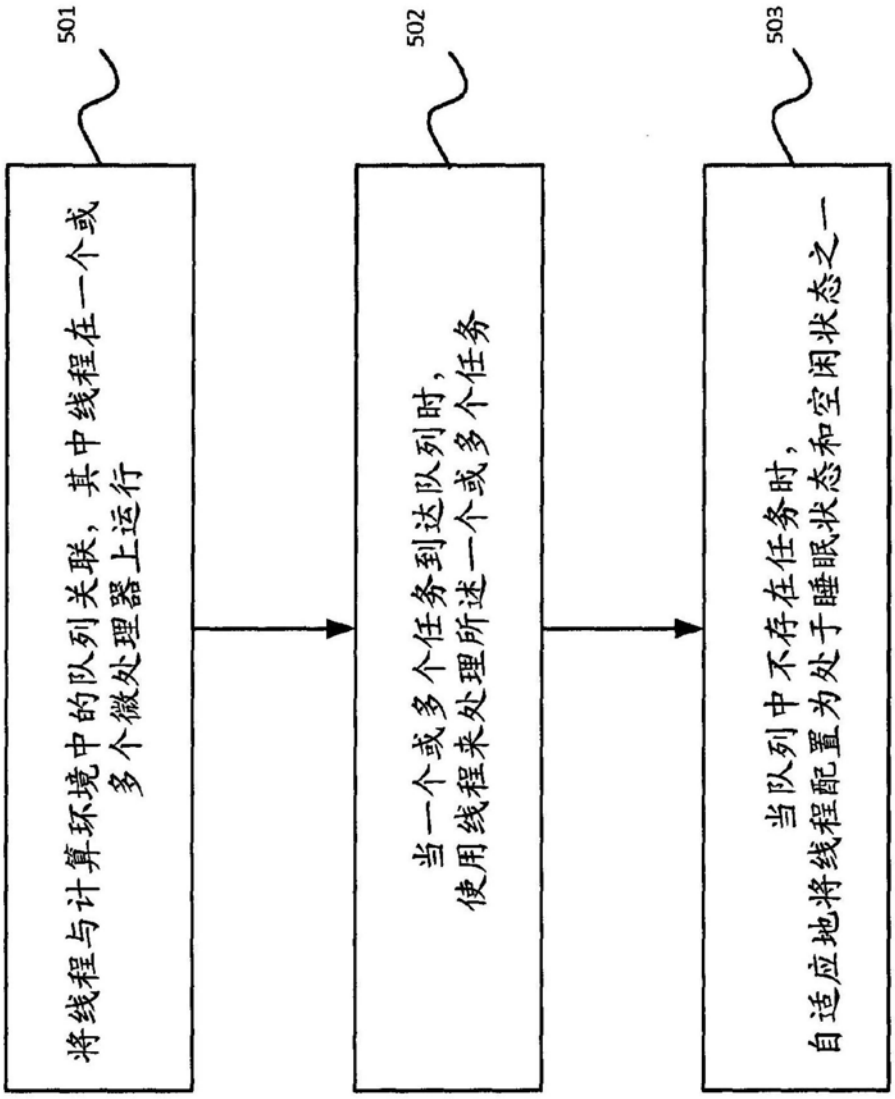

501
将线程与计算环境中的队列关联，其中线程在一个或多个微处理器上运行
502
当一个或多个任务到达队列时，使用线程来处理所述一个或多个任务
503
当队列中不存在任务时，自适应地将线程配置为处于睡眠状态和空闲状态之一

图5

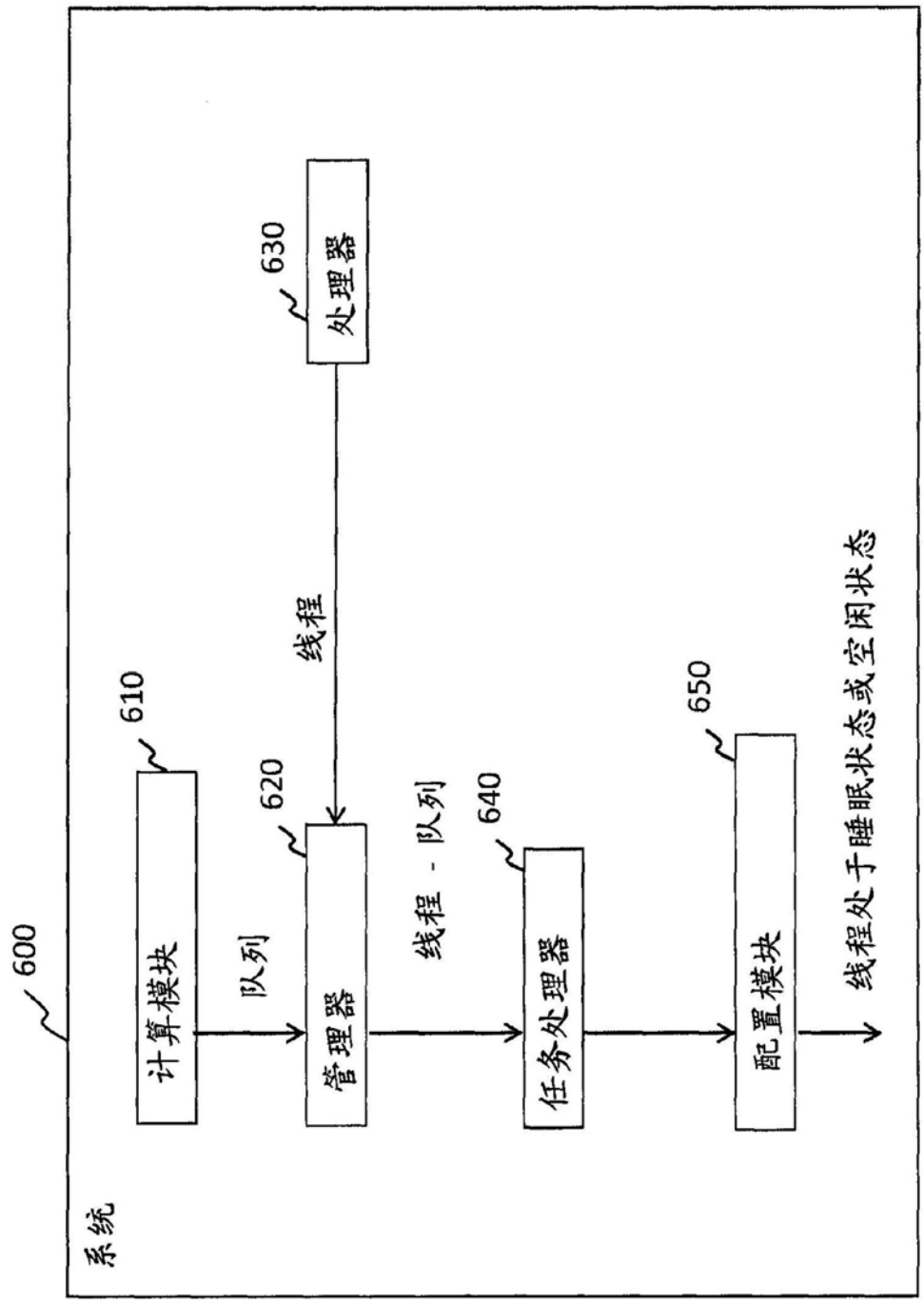

600
系统
计算模块
610
队列
管理器
620
线程
处理器
630
线程 - 队列
任务处理器
640
配置模块
650
线程处于睡眠状态或空闲状态

图6

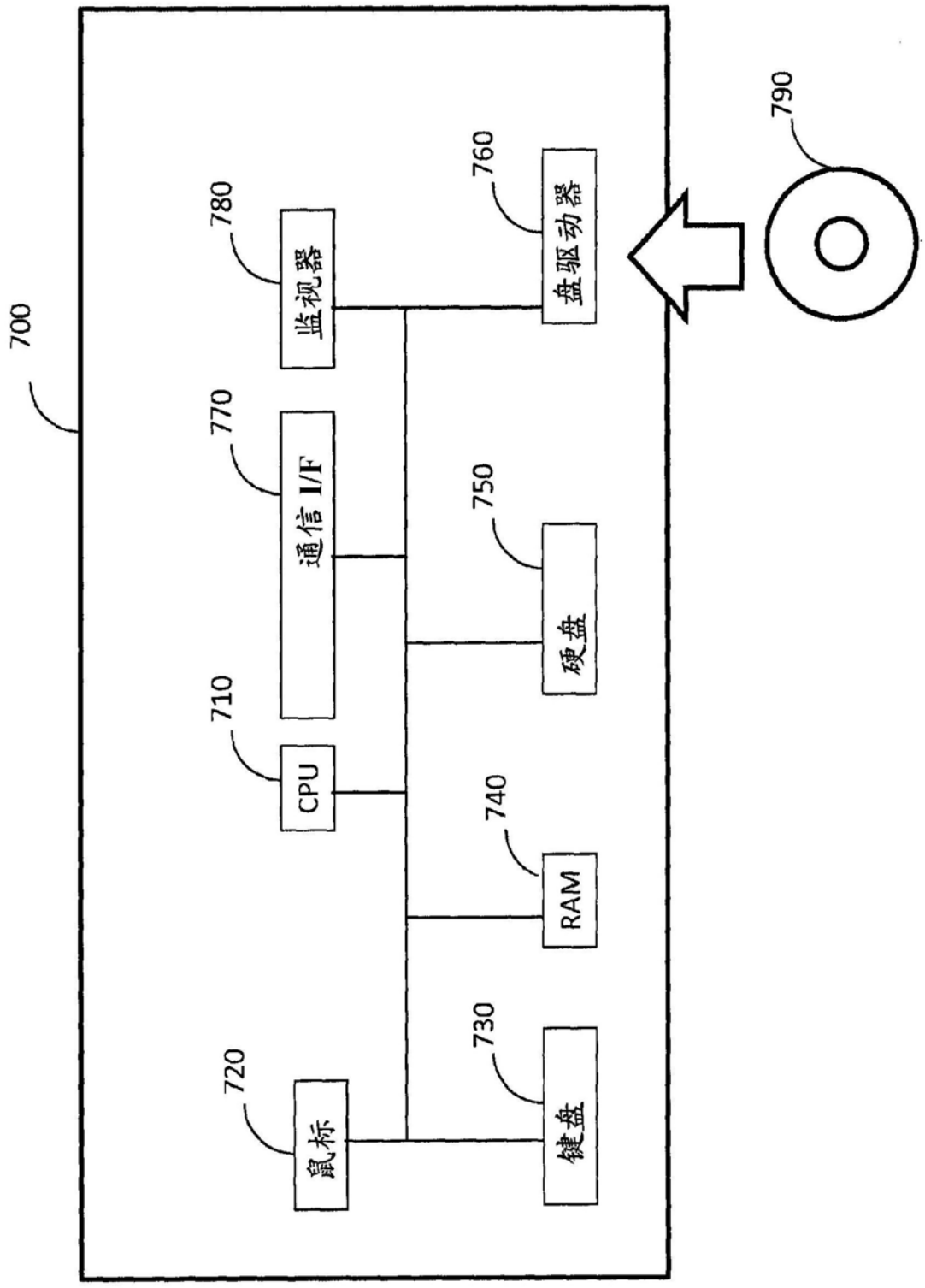

700
790
760
盘驱动器
780
监视器
770
通信 I/F
750
硬盘
710
CPU
740
RAM
730
键盘
720
鼠标

图7