



(12)发明专利

(10)授权公告号 CN 103842986 B

(45)授权公告日 2017.07.04

(21)申请号 201280047496.7

(22)申请日 2012.09.18

(65)同一申请的已公布的文献号
申请公布号 CN 103842986 A

(43)申请公布日 2014.06.04

(30)优先权数据

61/541,051 2011.09.29 US

13/414,593 2012.03.07 US

(85)PCT国际申请进入国家阶段日
2014.03.28

(86)PCT国际申请的申请数据
PCT/US2012/055942 2012.09.18

(87)PCT国际申请的公布数据
W02013/048826 EN 2013.04.04

(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 申旭刚 李向东

(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 15/167(2006.01)

(56)对比文件
US 2003182356 A1,2003.09.25,
CN 1610906 A,2005.04.27,全文.
US 7594234 B1,2009.09.22,全文.
US 6549961 B1,2003.04.15,全文.
US 2003182356 A1,2003.09.25,
CN 101546275 A,2009.09.30,全文.
eagle198699(博主).什么是latch以及如何
导致latch争用.《<http://blog.chinaunix.net/uid-14504139-id-1988457.html>》.2010,

审查员 朱殿尧

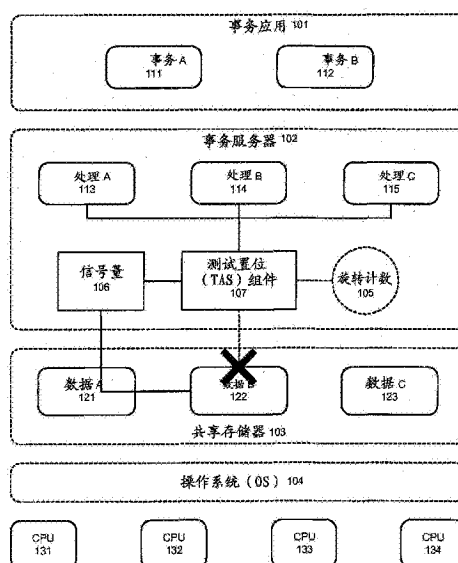
权利要求书4页 说明书8页 附图3页

(54)发明名称

用于在事务中间件机器环境中支持自调谐
锁定机制的系统和方法

(57)摘要

可以在事务中间件系统中支持锁定机制以在存在并行事务时保护共享存储器中的事务数据。事务中间件机器环境包括由运行在多个处理器上的操作系统提供的信号量。多个处理器操作来访问共享存储器中的数据。事务中间件机器环境也包括与一个或多个处理相关联的测试置位(TAS)装配组件。每个所述处理操作来使用TAS装配组件以执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定。另外,在TAS组件执行多次TAS操作并且未能获得锁定之后,处理操作来被在信号量上阻断并且等待对共享存储器中的数据的锁定的释放。



1. 一种用于在事务中间件机器环境中支持锁定机制的系统,包括:

信号量,由运行在多个处理器上的操作系统提供,其中所述多个处理器操作以访问共享存储器中的数据;和

测试置位TAS装配组件,与一个或多个处理相关联,其中每个所述处理操作来使用TAS装配组件执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定;

其中在TAS组件已经执行了指定数目的TAS操作并且未能获得锁定之后,一处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放;

其中所述锁定机制使用旋转计数,旋转计数是允许的TAS操作的指定的最大轮数;

其中旋转计数是利用算法动态地确定的,其中所述算法指定:

如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中CPU空闲率低于CPU空闲率极限,则旋转计数从前一周期的旋转计数增加,其中旋转失败指示所述处理在试图执行了次数为旋转计数的TAS操作之后未能获得锁定,以及

如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

2. 如权利要求1所述的系统,其中在存在多个并行事务时,所述锁定机制保护共享存储器中的事务数据。

3. 如权利要求1所述的系统,其中旋转计数被预配置在元数据中。

4. 如权利要求1所述的系统,其中旋转计数是基于硬件配置和应用情况动态地确定的。

5. 如权利要求1所述的系统,其中旋转计数是利用特殊处理周期性地确定的。

6. 如权利要求1所述的系统,其中当处理在尝试TAS操作指定次数之后未能获得对数据的锁定时,旋转失败发生。

7. 如权利要求1所述的系统,其中当锁定拥有者醒来并且释放锁定时,该处理使用信号量操作来获得锁定。

8. 如权利要求1所述的系统,其中旋转计数被手动地精细调谐以找到优化的值。

9. 一种用于在事务中间件机器环境中支持锁定机制的方法,包括:

提供与运行在多个处理器上的操作系统相关联的信号量,其中所述多个处理器操作以访问共享存储器中的数据;

经由一个或多个处理中的一处理,利用测试置位TAS装配组件来执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定;

在TAS组件已经执行了指定数目的TAS操作并且未能获得锁定之后,经由该处理在信号量上阻断并且等待对共享存储器中的数据的锁定的释放;以及

允许锁定机制使用旋转计数,旋转计数是允许的TAS操作的指定的最大轮数,

其中所述旋转计数是利用算法动态地确定的,其中所述算法指定:

如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中CPU空闲率低于CPU空闲率极限,则旋转计数从前一周期的旋转计数增加,其中旋转失败指示所述处理在试图执行了次数为旋转计数的TAS操作之后未能获得锁定,以及

如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

10. 如权利要求9所述的方法,还包括在存在多个并行事务时保护共享存储器中的事务数据。

11. 如权利要求9所述的方法,还包括在元数据中预配置所述旋转计数。

12. 如权利要求9所述的方法,还包括基于硬件配置和应用情况动态地确定所述旋转计数。

13. 如权利要求9所述的方法,还包括利用特殊处理周期性地确定所述旋转计数。

14. 如权利要求9所述的方法,其中当处理在尝试TAS操作指定次数之后未能获得对数据的锁定时,旋转失败发生。

15. 如权利要求9所述的方法,还包括当锁定拥有者醒来并且释放锁定时使用信号量获得锁定。

16. 如权利要求9所述的方法,还包括手动地精细调谐所述旋转计数以找到优化的值。

17. 一种用于在事务中间件机器环境中支持锁定机制的系统,包括:

提供单元,用于提供与运行在多个处理器上的操作系统相关联的信号量,其中所述多个处理器操作以访问共享存储器中的数据;

测试置位TAS装配组件,与一个或多个处理相关联,其中每个所述处理操作来使用TAS装配组件执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定;和

允许单元,用于允许锁定机制使用旋转计数,旋转计数是允许的TAS操作的指定的最大轮数;

其中在TAS组件已经执行了指定数目的TAS操作并且未能获得锁定之后,一处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放;

其中所述旋转计数是利用算法动态地确定的,其中所述算法指定:

如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中CPU空闲率低于CPU空闲率极限,则旋转计数从前一周期的旋转计数增加,其中旋转失败指示所述处理在试图执行了次数为旋转计数的TAS操作之后未能获得锁定,以及

如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

18. 如权利要求17所述的系统,还包括保护单元,用于在存在多个并行事务时保护共享存储器中的事务数据。

19. 如权利要求17所述的系统,还包括预配置单元,用于在元数据中预配置所述旋转计数。

20. 如权利要求17所述的系统,还包括动态确定单元,用于基于硬件配置和应用情况动态地确定所述旋转计数。

21. 如权利要求17所述的系统,还包括周期性确定单元,用于利用特殊处理周期性地确定所述旋转计数。

22. 如权利要求17所述的系统,其中当处理在尝试TAS操作指定次数之后未能获得对数据的锁定时,旋转失败发生。

23. 如权利要求17所述的系统,还包括使用单元,用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定。

24. 如权利要求17所述的系统,还包括调谐单元,用于手动地精细调谐所述旋转计数以找到优化的值。

25. 一种用于在事务中间件机器环境中支持锁定机制的系统,包括:

用于提供与运行在多个处理器上的操作系统相关联的信号量的装置,其中所述多个处理器操作以访问共享存储器中的数据;

用于经由一个或多个处理中的一处理,利用测试置位TAS装配组件来执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定的装置;

用于在TAS组件已经执行了指定数目的TAS操作并且未能获得锁定之后,经由该处理在信号量上阻断并且等待对共享存储器中的数据的锁定的释放的装置;和

用于允许锁定机制使用旋转计数的装置,旋转计数是允许的TAS操作的指定的最大轮数,

其中所述旋转计数是利用算法动态地确定的,其中所述算法指定:

如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中CPU空闲率低于CPU空闲率极限,则旋转计数从前一周期的旋转计数增加,其中旋转失败指示所述处理在试图执行了次数为旋转计数的TAS操作之后未能获得锁定,以及

如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

26. 如权利要求25所述的系统,还包括用于在存在多个并行事务时保护共享存储器中的事务数据的装置。

27. 如权利要求25所述的系统,还包括用于在元数据中预配置所述旋转计数的装置。

28. 如权利要求25所述的系统,还包括用于基于硬件配置和应用情况动态地确定所述旋转计数的装置。

29. 如权利要求25所述的系统,还包括用于利用特殊处理周期性地确定所述旋转计数的装置。

30. 如权利要求25所述的系统,其中当处理在尝试TAS操作指定次数之后未能获得对数据的锁定时,旋转失败发生。

31. 如权利要求25所述的系统,还包括用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定的装置。

32. 如权利要求25所述的系统,还包括用于手动地精细调谐所述旋转计数以找到优化的值的装置。

33. 一种用于在事务中间件机器环境中支持锁定机制的系统,包括:

用于提供与运行在多个处理器上的操作系统相关联的信号量的单元,其中所述多个处理器操作以访问共享存储器中的数据;

测试置位TAS装配组件,与一个或多个处理相关联,其中每个所述处理操作来使用TAS装配组件执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定;和

用于允许锁定机制使用旋转计数的单元,旋转计数是允许的TAS操作的指定的最大轮数;

其中在TAS组件已经执行了指定数目的TAS操作并且未能获得锁定之后,一处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放;

其中所述旋转计数是利用算法动态地确定的,其中所述算法指定:

如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中CPU空闲率低于CPU空闲率极限,则旋转计数从前一周期的旋转计数增加,其中旋转失败指示所述处理在试图执行了次数为旋转计数的TAS操作之后未能获得锁定,以及

如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

34. 如权利要求33所述的系统,还包括用于在存在多个并行事务时保护共享存储器中

的事务数据的单元。

35. 如权利要求33所述的系统,还包括用于在元数据中预配置所述旋转计数的单元。

36. 如权利要求33所述的系统,还包括用于基于硬件配置和应用情况动态地确定所述旋转计数的单元。

37. 如权利要求33所述的系统,还包括用于利用特殊处理周期性地确定所述旋转计数的单元。

38. 如权利要求33所述的系统,其中当处理在尝试TAS操作指定次数之后未能获得对数据的锁定时,旋转失败发生。

39. 如权利要求33所述的系统,还包括用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定的单元。

40. 如权利要求33所述的系统,还包括用于手动地精细调谐所述旋转计数以找到优化的值的单元。

用于在事务中间件机器环境中支持自调谐锁定机制的系统和方法

[0001] 版权通知

[0002] 本专利文件的公开的一部分包含受版权保护支配的材料。版权所有人不反对任何人传真复制专利文件或专利公开,因为它出现在专利商标局专利文档或记录中,但在别的方面保留所有版权。

技术领域

[0003] 本发明一般涉及计算机系统和诸如中间件之类的软件,并且具体地涉及支持事务中间件机器环境。

背景技术

[0004] 事务中间件系统、或面向事务的中间件包括可以在组织之内处理多种事务的企业应用服务器。随着诸如高性能网络和多处理器计算机之类的新技术的发展,需要进一步改善事务中间件的性能。这些一般是本发明的实施例意欲解决的领域。

发明内容

[0005] 这里描述的是可以在事务中间件系统中支持的用于在存在并行事务时保护共享存储器中的事务数据的锁定机制。事务中间件机器环境包括运行在多个处理器上的操作系统,每个操作以访问共享存储器中的数据。事务中间件机器环境也包括由操作系统提供的信号量(semaphore)、和与一个或多个处理相关联的测试置位(TAS)装配组件。每个处理可以使用TAS装配组件执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定。此外,在TAS组件在执行指定数目的TAS操作后未能获得锁定之后,处理可以在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放。

附图说明

[0006] 图1示出根据本发明的实施例的支持自调谐锁定机制的事务中间件机器环境的例示。

[0007] 图2示出了根据本发明的实施例的用于在事务中间件机器环境中支持自调谐锁定机制的示范性流程图。

[0008] 图3是根据本发明的实施例的用于在事务中间件机器环境中支持锁定机制的系统的功能框图。

具体实施方式

[0009] 这里描述的是用于支持诸如Tuxedo之类的事务中间件系统的系统和方法,其可以利用具有多处理器的快速机器、和高性能网络连接。可以在事务中间件系统中支持锁定机制以在存在并行事务时保护共享存储器中的事务数据。事务中间件机器环境包括由运行在

多个处理器上的操作系统提供的信号量。多个处理器可以访问共享存储器中的数据。事务中间件机器环境也包括与一个或多个处理相关联的测试置位 (TAS) 装配组件。每个所述处理操作来使用TAS装配组件以执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定。另外,在TAS组件执行多个TAS操作并且未能获得锁定之后,处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放。

[0010] 根据本发明的实施例,系统包括例如64位处理器技术的高性能硬件、高性能大容量存储器、和冗余无限带和以太网的组合,与诸如WebLogic套之类的应用服务器或中间件环境一起以提供完整的JavaEE应用服务器综合体,其包括大规模并行存储器内网格、可以被快速提供、并可以根据需要分级。根据实施例,系统可以被布置为完全的、一半、或四分之一机架、或其它配置,其提供应用服务器网格、存储区域网络、和无限带 (IB) 网络。中间件机器软件可以提供应用服务器、中间件和其它功能,诸如例如,WebLogic服务器、JRockit或Hotspot JVM、Oracle Linux或Solaris、和Oracle VM。根据实施例,系统可以包括经由IB网络与彼此通信的多个计算节点、IB交换机网关、和存储节点或单元。当被实施为机架配置时,机架的不使用的部分可以被保持为空或由填充物占据。

[0011] 根据本发明的实施例,这里称为“Sun Oracle Exalogic”或“Exalogic”的系统是解决诸如Oracle中间件SW套、或Weblogic之类的主机托管的中间件或应用服务器软件的容易部署的技术方案。如同这里描述的,根据实施例,系统是“盒中网格”,其包括一个或多个服务器、存储单元、用于存储联网的IB结构、和托管中间件应用所需的所有其它组件。通过利用例如实应用群和Exalogic开放存储来平衡大规模平行网格结构,可以为各种中间件应用实现重要的性能。系统传送有线性I/O可分级性的改善的性能、使用并管理简单、并实现任务关键的可用性和可靠性。

[0012] 根据本发明的实施例,Tuxedo是启动构造、执行、和管理高性能、分布式业务应用、并且已经由许多多层应用开发工具用作事务中间件的一组软件模块。Tuxedo是可以在分布式计算环境中用于管理分布式事务处理的中间件平台。它被证明是用于解锁企业传统应用和将它们扩展到面向服务的架构、同时实现无限的可分级性和基于标准的互操作性的平台。

[0013] 根据本发明的实施例,诸如Tuxedo系统之类的事务中间件系统可以利用诸如Exalogic中间件机器之类的具有多处理器的快速机器、和诸如无限带 (IB) 网络之类的高性能网络连接的优势。

[0014] 自调谐锁定机制

[0015] 根据本发明的实施例,自调谐锁定机制可以在事务中间件系统中被支持以在存在并行事务时保护共享存储器中的事务数据。利用自调谐锁定机制,事务中间件机器环境可以在诸如具有大规模并行事务的应用之类的事务应用情况中实现重要的吞吐量改进。

[0016] 图1示出根据本发明的实施例的支持自调谐锁定机制的事务中间件机器环境的例示。如图1所示,事务中间件机器包括支持操作系统 (OS) 104的多个CPU131-134、和包括多种事务数据121-123的共享存储器103。具有多个并行事务的事务应用101可以在事务服务器102中的多个处理113-115上运行,每个处理可以使用原子TAS(测试置位) 组件107以实施有效锁定机制。锁定机制在存在并行事务时可以保护共享存储器中的事务数据。此外,事务应用中的处理可以使用由OS提供的信号量机制106以必要时获得对数据122的锁定。

[0017] 根据一个实施例,当处理想要得到对数据122的锁定时,处理可以执行许多轮的TAS操作。系统可以指定目标旋转计数,其是允许的TAS操作的轮数。可以或者预配置或者动态地确定目标旋转计数。

[0018] 如果在到达目标旋转计数之前锁定变得可用,则处理能够以比由OS提供的信号量机制更少的代价获得锁定。另一方面,如果锁定在此期间不可用,则处理可以被配置为阻断信号量,并且等到锁定拥有者醒来并且释放锁定。

[0019] 根据本发明的实施例,可以在硬件配置和应用情况的上下文中决定目标旋转计数值。用户可以手动地精细调谐旋转计数值以便找到优化值。在一些情况中决定可能不是明显的,因为在CPU使用率和时间之间存在折衷以得到锁定。例如,用户可以必须花费更多的CPU功率来执行更多的TAS操作以便在更短时间帧内得到锁定。因此,可以不存在对每一情况都明显的优化的目标旋转计数值。

[0020] 用户级信号量实施方式的一个缺点是事务应用不能根据具体的机器类型实时地动态调整目标旋转计数。一般,用户级信号量仅仅使用静态配置的目标旋转计数值,并且用户可以仅仅通过经验实践手动地调整目标旋转计数值。因为最优的旋转计数值是机器相关的,并且没有用于所有平台的一个适合所有的值,因此更适当的方法是采用动态并且实时地计算目标旋转计数值的机制。

[0021] 根据本发明的实施例,目标旋转计数值可以被存储在共享存储器中。诸如Tuxedo守护进程处理之类的特定处理可以根据集中在前一周期中收集的操作信息周期性地改变旋转计数值。例如,Tuxedo守护进程可以默认每5秒一次更新目标旋转计数值。

[0022] 根据一个实施例,算法可以用于配置目标旋转计数值。如果CPU空闲率低,或太多TAS操作未能获得锁定并且系统切换到信号量,则算法可以增加目标旋转计数值。此外,如果CPU空闲率太高,则算法可以降低目标旋转计数值。

[0023] 图2示出了根据本发明的实施例的用于在事务中间件机器环境中支持自调谐锁定机制的示范性流程图。如图2所示,在步骤201处,系统可以提供与运行在多个处理器上的操作系统相关联的信号量,其中多个处理器操作来访问共享存储器中的数据。然后,步骤202处,一个或多个处理中的一个处理可以使用测试置位(TAS)装配组件以执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定。最后,在步骤203处,在TAS组件已经执行指定数目的TAS操作并且未能获得锁定之后,处理操作可以在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放。

[0024] 配置Tuxedo中的SPINCOUNT值

[0025] 根据本发明的实施例,诸如Tuxedo配置文件中的SPINCOUNT参数之类的元数据可以用于指定目标旋转计数。SPINCOUNT可以或者是静态配置的值、或者是动态的值。

[0026] 例如,Tuxedo可以使用SPINCOUNT的静态设置值以确定公告牌锁定等待在系统级信号量上被阻断之前旋转多少次。此算法的缺点是由用户设置的值不是特定平台上的SPINCOUNT的最优值,因为SPINCOUNT的最优值取决于诸如CPU量、工作负荷、等待公告牌(BB)锁定的数目等等之类的许多动态系数。

[0027] 另外,Tuxedo可以动态地调谐SPINCOUNT的值同时考虑运行时环境。算法可以用于确定用于SPINCOUNT参数的适当的值。如果太多TAS操作在前一周已经失败并且系统切换到信号量,并且存在足够的CPU空闲率,则系统可以增加SPINCOUNT。另一方面,如果CPU空闲

率太高,则系统可以降低SPINCOUNT。

[0028] 以上所述算法基于例如CPU空闲率的CPU使用率,和例如为获得锁定在每10000次操作中SPIN失败的比率的SPIN失败率。SPIN失败率可以指示多少次经由信号量而不是经由TAS操作获得锁定。

[0029] 当处理处于活动状态时,处理可以处于SPIN模式。如果处理在试图执行TAS操作许多次之后未能获得锁定,其被称为SPIN失败,则处理可以在信号量上被阻断。处理可以试图反复地执行TAS操作以便在SPIN失败发生之前得到锁定。例如SPINCOUNT的可配置参数可以用于指定要被调用并执行的TAS操作的轮数。

[0030] 根据本发明的实施例,最小空闲CPU率和SPIN失败率可以利用配置文件中的元数据定义。例如,在Tuxedo配置文件中,可以存在用于定义最小空闲CPU率的MINIDLECPU参数,其值具有1-100的范围,其中20作为默认值。另外,可以存在用于定义SPIN失败率的FACTOR参数,其值具有1-10000的范围,其中1000作为默认值。

[0031] 当给定最小空闲CPU率和SPIN失败率时,系统可以在每个扫描元件中调谐SPINCOUNT。然后,如果SPIN失败率太高(例如SPIN失败率被设置为大于 $(1/FACTOR*1.1)$),则系统可以增加SPINCOUNT,或空闲的CPU时间是足够的(例如,空闲率 $>MINIDLECPU\%+0.05$),其中新的 $SPINCOUNT=旧的SPINCOUNT+旧的SPINCOUNT*(cpu_idletime/cpu_usertime)$,并且最大SPINCOUNT可以被设置为10,000,000。否则,当空闲CPU率太低(例如,空闲率 $<MINIDLECPU\%-0.05$),其中新的 $SPINCOUNT=旧的SPINCOUNT/4$ 并且最小SPINCOUNT被设置为50000时,系统可以减小SPINCOUNT。

[0032] 以下列表1是用于SPINTUNING配置的Tuxedo示例。

```

*资源
IPCKEY          123456
DOMAINID        simpapp
MASTER          ALLEN
[0033] MAXACCESSERS  10
MAXSERVERS      5
MAXSERVICES     10
MODEL           SHM
LDBAL           N
OPTIONS          EXALOGIC,SPINTUNING

[0034] *机器
[0035] ALLENHOST LMID="ALLEN"
[0036] APPDIR="/home/allen/Workspace/Tuxedo11gR1PS2/simpdir"
[0037] TUXCONFIG="/home/allen/Workspace/Tuxedo11gR1PS2/simpdir/tuxconfig"
[0038] TUXDIR="/home/allen/Software/OraHome/tuxedo11gR1PS2"
[0039] SPINTUNING_TARGET=1000
[0040] SPINTUNING_MINIDLECPU=20
[0041] *组
[0042] GROUP1
[0043] LMID=ALLEN GRPNO=10PENINFO=NONE

```

[0044] *服务器

[0045] DEFAULT:

[0046] CLOPT="-A"

[0047] simpserv SRVGRP=GROUP1SRVID=1

[0048] *服务

[0049] TOUPPER

[0050] 列表1

[0051] 如以上示例所示,配置文件的*机器部分包括用于配置调谐目标的属性“SPINTUNING_TARGET”。“SPINTUNING_TARGET”的值是数字,其可以大于或等于“0”并且小于或等于例如“10000”。0值指示使用内置到二进制中的值。内置的值是1000。“SPINTUNING_TARGET”的默认值是0。

[0052] 上述示例中,SPINTUNING_TARGET的值意指每一个千锁定中最多一次经由系统信号量对公告牌锁定。系统可以随满足SPINTUNING_TARGET的较大的值花费更多的CPU,而增加SPINCOUNT的值。只有当选项SPINTUNING被指定时属性才可以用非零值设置。

[0053] 如以上示例所示,配置文件的*机器部分包括用于指定空闲CPU率的属性“SPINTUNING_MINIDLECPU”。空闲CPU率可以由系统使用以动态地找到合适的SPINCOUNT。SPINCOUNT越大,系统使用越多的CPU。用户可以经由“SPINTUNING_MINIDLECPU”设置最小空闲CPU率以避免消耗太多CPU。

[0054] “SPINTUNING_MINIDLECPU”的值是数字的并且按百分比。它可以大于或等于“0”并且小于或等于“100”。0值指示应当使用内置到传送的二进制中的值。例如,在Tuxedo中,内置的值可以被设置为20,并且“SPINTUNING_MINIDLECPU”的默认值是0。只有当指定选项SPINTUNING时属性才可以用非零值设置。

[0055] 根据一些实施例,图3是示出用于在根据如上所述的本发明的原理配置的事务中间件机器环境中支持锁定机制的系统1000的功能框图。系统1000的功能框可以由硬件、软件、或硬件和软件组合实施以实现本发明的原理。本领域的技术人员应当理解,在图3中描述的功能框可以被结合或分成子块以实施如上所述的本发明的原理。因此,这里的描述可以支持这里描述的功能框的任何一种可能的组合或分离或进一步定义。

[0056] 如图3所示,用于在事务中间件机器环境中支持锁定机制的系统1000包括提供单元1100和TAS装配组件1200。提供单元1100提供与运行在多个处理器上的操作系统相关联的信号量,其中多个处理器操作来访问共享存储器中的数据。TAS装配组件1200与一个或多个处理相关联,其中每个所述处理操作来使用TAS装配组件1200以执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定。在TAS组件1200已经执行指定数目的TAS操作并且未能获得锁定之后,处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放。

[0057] 在一些实施例中,系统1000还包括用于在存在多个并行事务时保护共享存储器中的事务数据的保护单元1300。

[0058] 在一些实施例中,系统1000还包括用于允许锁定机制使用旋转计数的允许单元1400,旋转计数是允许的TAS操作的指定的最大轮数。

[0059] 在一些实施例中,系统1000还包括用于预配置元数据中的旋转计数的预配置单元

1500。

[0060] 在一些实施例中,系统1000还包括用于基于硬件配置和应用情况动态地确定旋转计数的动态确定单元1600。

[0061] 在一些实施例中,系统1000还包括用于利用特殊处理周期性地确定旋转计数的周期性确定单元1700。

[0062] 在一些实施例中,利用算法动态地确定旋转计数,其中如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中的CPU空闲率低于CPU空闲率极限,则该算法指定旋转计数从前一周期的旋转计数增加,并且如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

[0063] 在一些实施例中,当处理在尝试TAS操作特定次数之后未能获得对数据的锁定时,旋转失败发生。

[0064] 在一些实施例中,系统1000还包括用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定的使用单元1800。

[0065] 在一些实施例中,系统1000还包括用于手动地精细调谐旋转计数以找到优化的值的精细调谐单元1900。

[0066] 其它实施例包括用于在事务中间件机器环境中支持锁定机制的设备,包括用于提供与运行在多个处理器上的操作系统相关联的信号量的装置,其中多个处理器操作来访问共享存储器中的数据;用于经由一个或多个处理的一个处理使用测试置位(TAS)装配组件来执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定的装置;用于在TAS组件已经执行指定数目的TAS操作并且未能获得锁定之后,经由该处理在信号量上阻断并且等待对共享存储器中的数据的锁定的释放的装置。

[0067] 在另一个实施例中,该设备还包括用于在存在多个并行事务时保护共享存储器中的事务数据的装置。

[0068] 在另一个实施例中,该设备还包括用于允许锁定机制使用旋转计数的装置,旋转计数是允许的TAS操作的指定的最大轮数。

[0069] 在另一个实施例中,该设备还包括用于预配置元数据中的旋转计数的装置。

[0070] 在另一个实施例中,该设备还包括用于基于硬件配置和应用情况动态地确定旋转计数的装置。

[0071] 在另一个实施例中,该设备还包括用于使用特殊处理周期性地确定旋转计数的装置。

[0072] 在实施例中,还包括其中利用算法动态地确定旋转计数的设备,其中如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中的CPU空闲率低于CPU空闲率极限,则该算法指定旋转计数从前一周期的旋转计数增加,并且如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

[0073] 在另一个实施例中,其中当处理在尝试用于特定次数的TAS操作之后未能获得对数据锁定时,旋转失败发生。

[0074] 在另一个实施例中,该设备还包括用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定的装置。

[0075] 在另一个实施例中,该设备还包括用于手动地精细调谐旋转计数以找到优化的值

的装置。

[0076] 还有一个实施例包括用于在事务中间件机器环境中支持锁定机制的系统,包括用于提供与运行在多个处理器上的操作系统相关联的信号量的单元,其中多个处理器操作来访问共享存储器中的数据;与一个或多个处理相关联的测试置位(TAS)装配组件,其中每个所述处理操作来使用TAS装配组件执行一个或多个TAS操作以便获得对共享存储器中的数据的锁定,其中在TAS组件已经执行指定数目的TAS操作并且未能获得锁定之后,处理操作来在信号量上被阻断并且等待对共享存储器中的数据的锁定的释放;

[0077] 在另一个实施例中,系统还包括用于在存在多个并行事务时保护共享存储器中的事务数据的单元。

[0078] 在另一个实施例中,系统还包括用于允许锁定机制使用旋转计数的单元,旋转计数是允许的TAS操作的指定的最大轮数。

[0079] 在另一个实施例中,系统还包括用于预配置元数据中的旋转计数的单元。

[0080] 在另一个实施例中,系统还包括用于基于硬件配置和应用情况动态地确定旋转计数的单元。

[0081] 在另一个实施例中,系统还包括用于使用特殊处理周期性地确定旋转计数的单元。

[0082] 另一个实施例包括其中利用算法动态地确定旋转计数的系统,其中如果在前一周期中旋转失败的数目超过旋转失败极限并且在前一周期中的CPU空闲率低于CPU空闲率极限,则该算法指定旋转计数从前一周期的旋转计数增加,并且如果CPU空闲率超过CPU空闲率极限,则旋转计数从前一周期的旋转计数减小。

[0083] 另一个实施例包括其中当处理在尝试指定次数的TAS操作之后未能获得对数据锁定时,旋转失败发生的系统。

[0084] 另一个实施例包括还包括用于当锁定拥有者醒来并且释放锁定时使用信号量获得锁定的单元的系统。

[0085] 并且还有另一个实施例包括还包括用于手动地精细调谐旋转计数以找到优化的值的单元的系统。

[0086] 本发明可以利用包括一个或多个处理器、存储器和/或根据本公开的教导编程的计算机可读存储介质的一个或多个传统的通用或专门数字计算机、计算设备、机器、或微处理器方便地执行。基于本公开的教导,本领域的技术人员可以容易地准备适当的软件编码,这对软件领域的技术人员将是显而易见的。

[0087] 在一些实施例中,本发明包括计算机程序产品,其是在其上/其中存储有指令的存储介质或计算机可读介质(媒介),所述指令可以用于编程计算机以实施本发明的处理中的任何一个。存储介质可以包括但是不局限于包括软盘、光盘、DVD、CD-ROM、微驱动器的任何类型的盘、和磁光盘、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、闪速存储设备、磁或光卡、纳米系统(包括分子存储器IC),或适合于存储指令和/或数据的任何类型的介质或设备。

[0088] 为了例示和描述目的已经提供本发明的以上描述。它不意欲是详尽的或将本发明限制到公开的精确形式。本领域的技术人员将清楚许多修改和变化。选择并描述实施例以便最佳地说明本发明的原理和它的实际应用,从而使得本领域的其他技术人员理解本发明的各种实施例和适合于特定预期使用的各种修改。意图由以下权利要求和它们的相等物限

定本发明的范围。

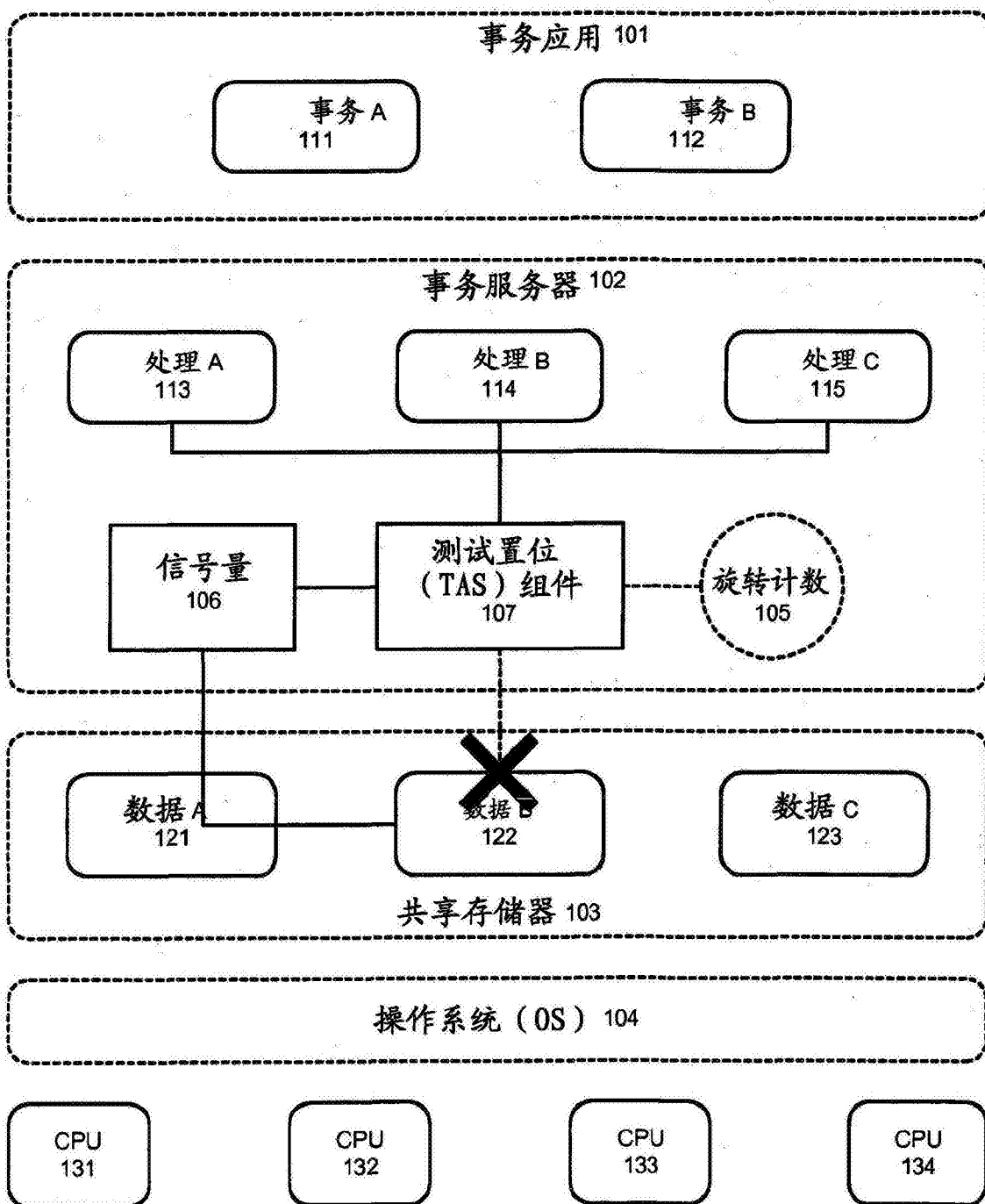


图1

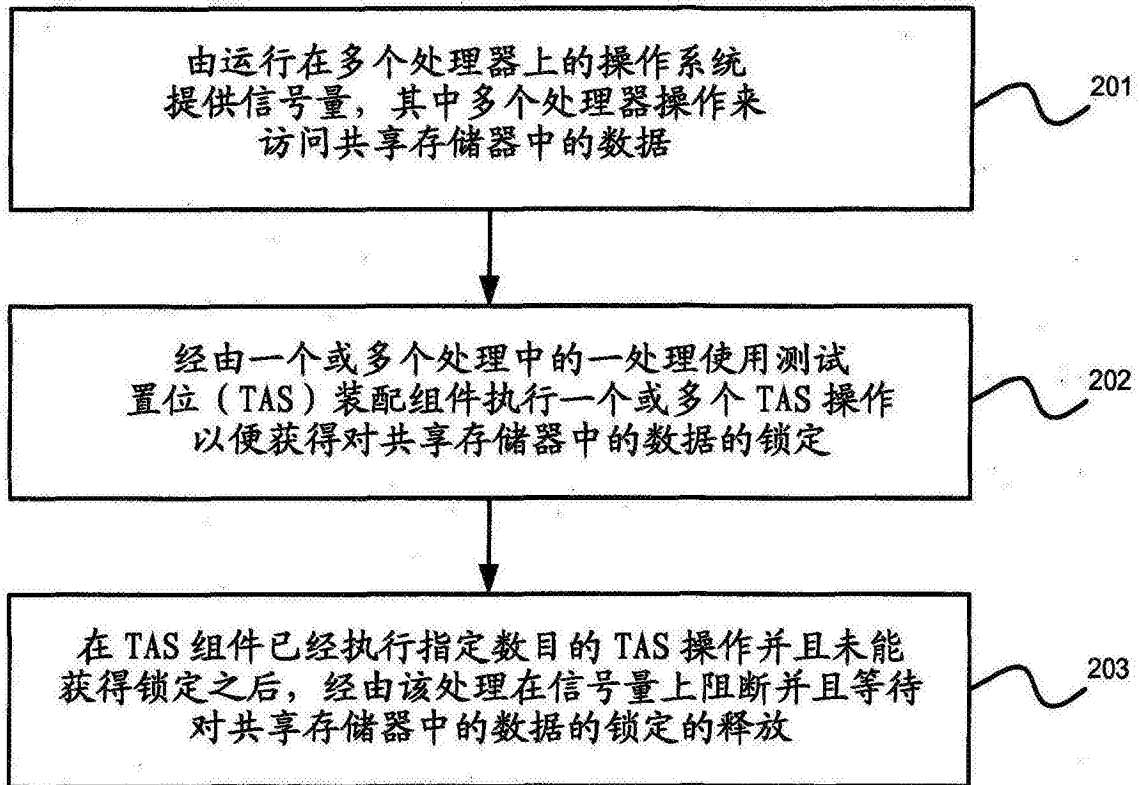


图2

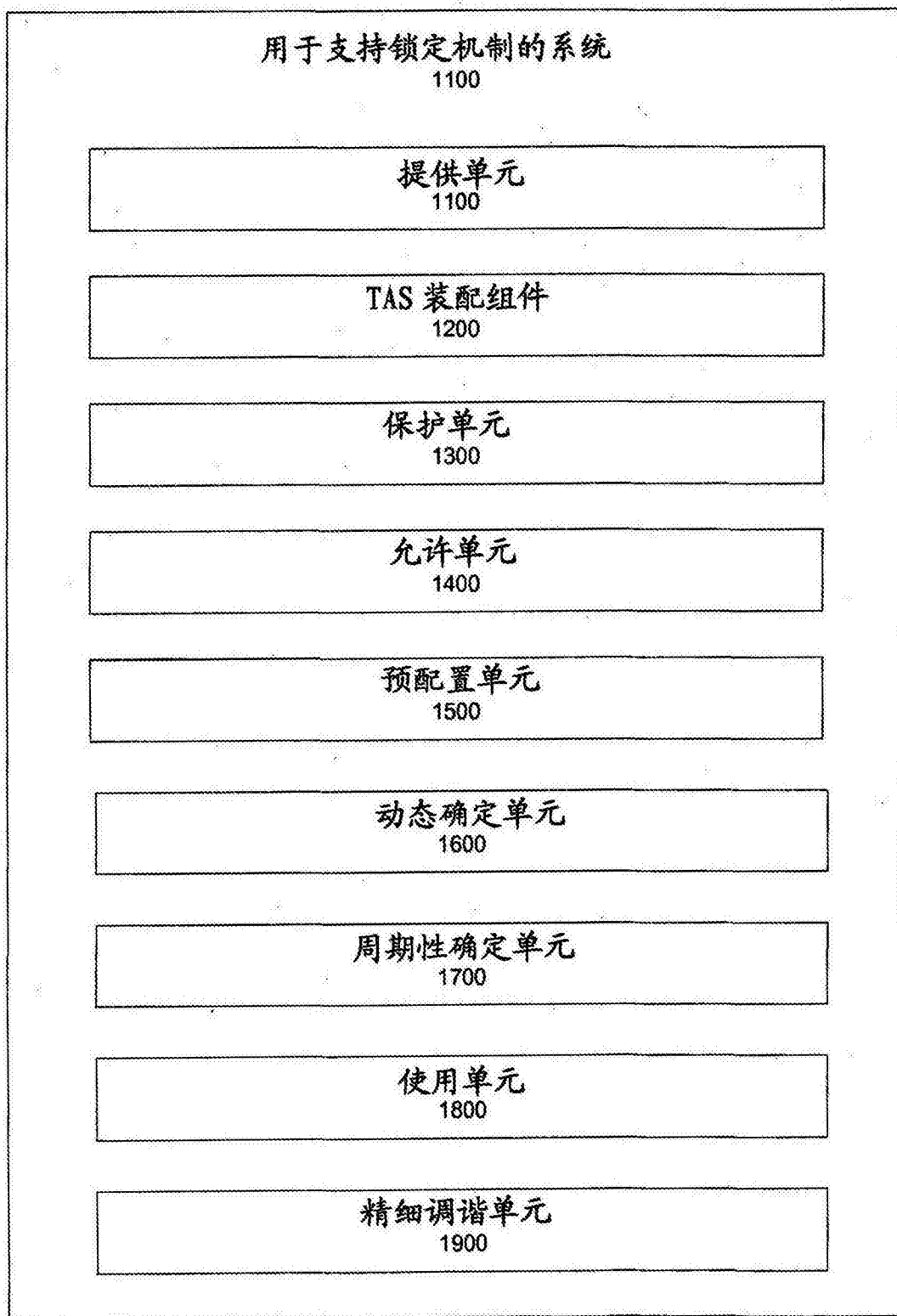


图3