(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102667758 A
(43)申请公布日 2012.09.12

(21)申请号 201080053045.5

(22)申请日 2010.10.29

(30)优先权数据
12/625,379 2009.11.24 US

(85)PCT申请进入国家阶段日
2012.05.23

(86)PCT申请的申请数据
PCT/US2010/054734 2010.10.29

(87)PCT申请的公布数据
WO2011/066057 EN 2011.06.03

(71)申请人 微软公司
地址 美国华盛顿州

(72)发明人 I·奥斯特罗夫斯基

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 胡利鸣

(51)Int.Cl.
G06F 17/26(2006.01)
G06F 17/40(2006.01)
G06F 9/46(2006.01)
G06F 13/14(2006.01)

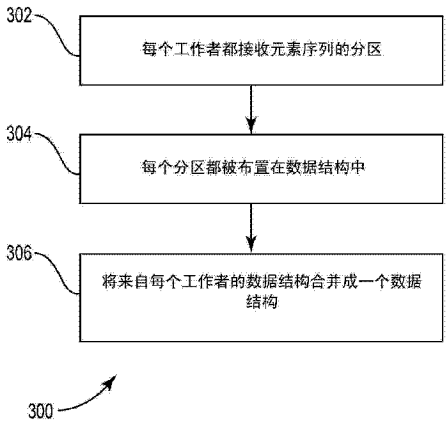
权利要求书 1 页 说明书 6 页 附图 5 页

(54)发明名称

用于多处理器核执行的编组机制

(57)摘要

提供了一种用于在多核处理器上执行的并发编组操作。向编组操作提供元素的序列或组。在一个阶段，每个工作者都接收要被编组的元素序列的分区。将每个分区的元素布置成到包括一个或多个键的数据结构中，每个键都对应于与该键相关联的所接收的元素中的一个或多个的值列表。在另一阶段，合并由每个工作者创建的数据结构，使得整个元素序列的键和相应元素存在于一个数据结构中。递归合并可以以恒定时间完成，该时间与序列的长度不成比列。



CN 102667758 A

1. 一种用于对元素序列进行编组的方法，包括：

接收元素序列的多个分区，其中多核处理器的可用核的每个都接收所述多个分区的单独的分区以用于处理；

并发地将每个单独的分区编组成元素的至少一个值列表的键表，其中每个值列表中的元素都包括共同的键；以及

将所述键表中的每个合并到一起以形成最终的键表，使得所有键以及它们的相应值列表都被包括在所述最终的键表中。

2. 如权利要求 1 所述的方法，其特征在于，元素的所述单独分区的每个元素都具有相关联的键，并且具有相同键的元素被包括在元素的所述值列表中。

3. 如权利要求 2 所述的方法，其特征在于，每个键表都包括多个键，每个键都具有拥有该键的元素的相应值列表。

4. 如权利要求 2 所述的方法，其特征在于，所述键是用键选择器函数来生成的，所述键选择器函数对所述单独分区的每个元素进行操作。

5. 如权利要求 1 所述的方法，其特征在于，所述合并包括生成新的键表。

6. 如权利要求 1 所述的方法，其特征在于，所述合并包括扩充现有键表。

7. 如权利要求 1 所述的方法，其特征在于，所述合并包括：

将键和相应值列表在所述键仅存在于仅仅一个键表中的情况下添加到键表中；以及

串接存在于一个以上键表中的共同的键的值列表。

8. 如权利要求 7 所述的方法，其特征在于，所述合并包括：重复地将部分键表合并到一起以形成完整的键表。

9. 如权利要求 8 所述的方法，其特征在于，所述部分键表被顺序地合并到一起以形成所述完整的键表。

10. 如权利要求 8 所述的方法，其特征在于，所述部分键表被递归地合并到一起以形成所述完整的键表。

11. 如权利要求 1 所述的方法，其特征在于，所述键表是在可操作地耦合到所述多核处理器的存储器中形成的数据结构。

12. 如权利要求 11 所述的方法，其特征在于，所述数据结构包括链表。

13. 如权利要求 12 所述的方法，其特征在于，将所述键表中的每个合并到一起包括：将键和值列表提供到存储器中的新的数据结构中或者将键和值列表添加到所述存储器中的现有数据结构中。

14. 如权利要求 1 所述的方法，其特征在于，在将每个单独的分区编组到键表中以前确定所述元素序列的多个分区的数量。

15. 如权利要求 14 所述的方法，其特征在于，多个分区中的各个分区都包括相等数量的元素。

用于多处理器核执行的编组机制

背景技术

[0001] 具有在多个处理器上执行的并发线程的计算机应用展现出增加性能的巨大前景，但是也向开发者提出巨大挑战。随着处理器制造商在提供处理器时钟频率的显著提高方面遇到障碍，原始顺序处理能力的增长已经平淡化。处理器继续演进，但是针对改善处理器能力的当前焦点是在单个晶片上提供多个处理器核以增加处理器吞吐量。随着处理器核数目增加，以前受益于增加的时钟速度的顺序应用获得显著更少的缩放比例。为了利用多核系统，编写了并发(或并行)应用以包括分布在核上的并发线程。然而，并行化应用是充满挑战的，因为许多常用工具、技术、编程语言、框架、以及甚至开发者本身都是适应于创建顺序的程序的。

[0002] 编组(grouping)操作代表了并行改善可以进行但是大多未被充分利用的一个应用领域。编组操作接收元素序列并且将这些元素放置到预定的组中，其中该序列中的每个元素都在被编组时受到检查。有时，该序列可以包含几百万个或更多元素。将元素顺序地编组为预定的组的性能效果是用于执行编组操作的时间与该序列的元素的数目相关。

[0003] 概述

[0004] 提供本概述以便以简化的形式介绍将在以下的详细描述中进一步描述的一些概念。本发明内容并不旨在标识出所要求保护的主题的特定特征或必要特征，也不旨在用于限定所要求保护的主题的范围。

[0005] 本公开针对一种用于在多核处理器上执行的并发编组操作。向编组操作提供元素的序列或组。在一个阶段，诸如处理器中的逻辑核或物理核之类的每个工作者都接收要被编组的元素序列的分区。将每个分区的元素布置成到包括一个或多个键的数据结构中，每个键都对应于与该键相关联的所接收元素中的一个或多个的值列表。在另一阶段，合并由每个工作者创建的数据结构，使得整个元素序列的键和相应元素存在于一个数据结构中。

[0006] 附图简述

[0007] 包括附图来提供了对各实施例的进一步理解，且这些附图被合并在本发明书内并构成其一部分。附图示出各实施例，并且与说明书一起用于解释本发明的原理。其他实施例和各实施例的许多预期优点将随着参考下面的详细描述进行更好的理解而得到认识。附图的元素不一定相对于彼此而缩放。相同的附图标记指代对应的类似部分。

[0008] 图 1 是示出示例计算设备的框图。

[0009] 图 2 是示出图 1 的计算设备的多核处理系统的示例的示意图。

[0010] 图 3 是一框图，其示出了用于诸如图 2 的示例之类的多核处理系统的编组操作的示例性方法。

[0011] 图 4 是示出实现来自图 3 的方法的键表的示例性数据结构的示意图。

[0012] 图 5 是示出了组合图 4 的两个键表的示例的示意图。

[0013] 图 6 是一框图，其示出了用于在诸如图 2 的示例之类的多核处理系统中递归地组合图 4 的键表的示例的示意图。

[0014] 详细描述

[0015] 在以下具体实施例中，对附图进行了参考，附图构成了实施例的一部分且在其中作为示例示出了可在其中实践本发明的各特定实施例。可以理解，可以使用其它实施例并且可以做出结构上或逻辑上的改变而不背离本发明的范围。因此，以下详细描述并不旨在限制，并且本发明的范围由所附权利要求来限定。应理解，此处描述的各示例性实施例的特征可相互组合，除非另外具体注明。

[0016] 图 1 示出了可用作操作环境并且包括诸如计算设备 100 之类的计算设备的示例性计算机系统。在一基本配置中，计算设备 100 通常包括具有至少两个处理单元(即，处理器 102)的处理器体系结构以及存储器 104。取决于计算设备的确切配置和类型，存储器 104 可以是易失性的(如随机存取存储器(RAM))、非易失性的(诸如只读存储器(ROM)、闪存等)或两者的某种组合。该基本配置在图 1 中由虚线 106 来例示。该计算设备可采取若干形式中的一种或多种。这些形式包括个人计算机、服务器、手持式设备、消费电子产品(诸如视频游戏控制台)或其他设备。

[0017] 计算设备 100 还可具有附加特征或功能。例如，计算设备 100 还可包括附加存储(可移动和 / 或不可移动)，包括但不限于 :磁盘或光盘或固态存储器，或者闪速存储设备，诸如可移动存储 108 和不可移动存储 110。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等的任何合适的方法或技术实现的易失性和非易失性、可移动和不可移动介质。存储器 104、可移动存储 108 和不可移动存储 110 都是计算机存储介质的示例。计算机存储介质包括，但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘(DVD)或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁性存储设备、通用串行总线(USB)闪存驱动器、闪存卡、或能用于存储所需信息且可以由计算设备 100 访问的任何其它介质。任何这样的计算机存储介质都可以是计算设备 100 的一部分。

[0018] 计算设备 100 包括允许计算设备 100 与其它计算机 / 应用 115 进行通信的一个或多个通信连接 114。计算设备 100 还可包括诸如键盘、定点设备(例如，鼠标)、笔、语音输入设备、触摸输入设备等的输入设备 112。计算设备 100 还可包括诸如显示器、扬声器、打印机等的输出设备 111。

[0019] 计算设备 100 可被配置成运行操作系统软件程序以及一个或多个软件应用，这些构成系统平台。在一个示例中，计算设备 100 包括被称为托管的或运行时环境的软件组件。托管环境可被包括为操作系统的一部分或者可在稍后被包括为软件下载。托管环境通常包括针对常见编程问题的预先编码的解决方案以帮助软件开发者创建诸如应用等在托管环境中运行的软件程序。

[0020] 被配置成在计算设备 100 上执行的计算机应用包括至少一个进程(或任务)，所述至少一个进程(或任务)是执行程序。每个进程提供用于执行该程序的资源。一个或多个线程在该进程的上下文中运行。线程是操作系统在处理器 102 中向其分配时间的基本单元。线程是为执行而调度的进程内的实体。进程的各线程可共享其虚拟地址空间和系统资源。每个线程可包括异常处理程序、调度优先级、线程位置存储、线程标识符、以及直到该线程被调度的线程上下文(或线程状态)。在并行应用中，可在处理器 102 上并发地执行各线程。

[0021] 图 2 是可在计算设备 100 中被实现为用于并发执行各线程的处理器 102 的示例多核处理器 200。这一示例包括在单个晶片 202 上实现的多核。该示例性多核处理器 200 包括四个物理处理器核 204、206、208、210、或简称四个物理核，其中这些物理核中的每个都可

用于处理至少一个应用线程，而至少一个其他物理核并发地处理另一线程。如图所示，物理核 204、206、208、210 与晶片 202 上的存储器控制器 212 和高速缓存 214 相邻。这些核中的每个都与高速缓存分层结构相关联。在一个示例中，处理器 102 的架构包括物理核 204、206、208、210 中的高速缓存（比如 L1 和 L2 高速缓存）以及高速缓存 214 中的 L3 高速缓存、由存储器控制器 212 来服务的存储器 104 等等。该示例中的高速缓存 L1、L2 和 L3 可以表示晶片上存储器，因为这些高速缓存位于晶片 202 上，而存储器分层结构可以进一步扩展到晶片外存储器，比如存储器 104。在示例性晶片 202 中，队列 216 部署在晶片上的存储器控制器 212 与高速缓存 214 之间。晶片 202 可包括其他特征 218 或诸如存储器接口、杂项输入 / 输出块、专有互连、扩展卡接口之类的特征的组合。

[0022] 每个物理核都能够有效地和并发地执行并发进程的多个线程。这些物理核常常被称为“同时多线程”核或常常简称为“SMT”核，并且每个物理核上的并发执行的线程共享包括在该单个物理核内的硬件资源。在多核处理系统 200 的示例中，每个物理核都能够进行多线程化。能够进行多线程化的每个物理核都可以向操作系统呈现如其支持的并发执行线程一样多的逻辑核。在示例性的多核处理系统 200 中，每个物理核 204、206、208、210 都能够并发执行两个线程，并由此向操作系统提供八个并发逻辑核。

[0023] 在一些示例中，单个处理器（未示出）或多核处理器 102 可以根据性能考虑因素作为多处理器架构的一部分被包括在内。非一致存储器访问（NUMA）和对称多处理（SMP）系统提供了可用多处理器架构的两个共同示例。多处理器架构内的每个处理器或逻辑核都能够执行线程。多处理器架构可以进一步与分布式系统中的其他多处理器架构相组合。逻辑核、物理核、处理器和多处理器系统的许多可用或以后开发的组合可以被用来实现编组机制，但是该编组机制不限于任何特定的处理器系统或架构。能够并发地执行并发编组机制的线程或组件的每个单元都在此统称为“核”或“工作者（worker）”。

[0024] 图 3 示出了用于编组操作 300 的示例性方法。向编组操作 300 提供元素的序列或组，比如值序列。在一个阶段，在 302，诸如处理器 200 中的逻辑核或物理核之类的每个工作者都接收要被编组的元素序列的分区。在 304，将每个分区的元素布置成到包括一个或多个键（key）的数据结构中，每个键都对应于与该键相关联的所接收元素中的一个或多个的值列表。在另一阶段，在 306，合并由每个工作者创建的数据结构，使得整个元素序列的键和相应元素存在于一个数据结构中。

[0025] 供用于编组操作 300 的多个核处理器或多核处理器包括一定数目的核（比如在处理系统 200 中为八个，每个核都执行线程）或工作者。有时（比如当多核处理器 200 正在处理其他应用时、当多核处理器 200 包括空闲核时、二者的组合或其他情况），仅仅处理器核的子集可用于处理编组操作。操作系统可以提供关于编组操作的可用核的数据。编组操作 300 还可以被实现为存储在存储器中的软件产品、或者被实现为包括计算设备 100 的处理器 102 和存储器 104 的系统。

[0026] 元素序列还可以被广范围地定义，但是下面的示例是为了图解说明而包括在内的。输入序列要么用编组操作本身、用在先数据并行操作、要么用其他方式被分割成多个分区。给每个分区都分配相应的核以处理编组操作中的分区。在一个示例中，分区的数目对应于用于处理编组操作的可用核的数目。在一个示例中，该序列可以具有预定的长度，并且在分区以前向编组操作提供该长度。在这种情况下，编组操作可以将序列分成逻辑分区，比如

具有大致相同元素数量的分区、对每个的处理都为相同时间长度的分区、或者其他逻辑分区。在未向编组操作提供预定的元素长度的情况下，编组操作可以应用要么已知、要么待发现的算法以用于在给定数目的工作者中对元素进行分区。序列中的元素可以为任何类型，并且在所述示例中使用了一定数据类型的元素。例如，元素序列可以包括整数、浮点数、字母数字串等等。

[0027] 附加于接收元素序列，编组操作还接收键选择器函数作为输入，以提供关于如何对元素序列进行编组的参数。编组操作参照元素序列中的元素来对键选择器函数求值，并且获得与每个元素相关联的键。因此，键选择器函数可以表示为：

[0028] f(element)=key

[0029] 在一个示例中，键选择器函数可以是整数序列中的每个元素的最低有效位。在另一示例中，键选择器函数是字母数字串中的第一个字符。具有相同或共同的键的所有元素都被一起编组到数据结构中。在使用 C# (C-sharp) 的一个示例中，该数据结构可以表示为：

[0030] Dictionary<TKey,LinkedList<TValue>>

[0031] 当然，可以替代于链表使用其他数据结构。每个工作者都检查相应分区中的每个元素以确定合适的键，查找该数据结构中的合适链表，并且将该元素插入到该链表中。每个数据结构在存储器分层结构中的位置可以由其大小来规定。在一些示例中，数据结构可以被创建在晶片上存储器中以获得增加的效率。在另一示例中，可以给每个元素预分配所接收的分区中的键。

[0032] 图 4 示出了数据结构 400 的示例，该数据结构包括具有一组键 404、406 的键表 402，其中每个键都分别具有相关联的链表 408、410。键表 402 是通过评估分配给第一工作者的经分区的序列的元素而产生的示例性结果。每个链表 408、410 都包括来自具有该键的经分区的序列的元素，比如元素 412、414。示例性键表 402 中的键是从包括 arrow、anvil、axe、dust 和 door 的序列中的字母数字串的第一个字符中确定的。根据键选择器函数，键 404 是“A”，并且键 406 是“D”。

[0033] 在该示例中，用在编组操作中的每个工作者都基于相同的键选择器函数来创建键表，该键表然后被应用于分配给该工作者的序列分区中的每个元素。因此，在使用八个逻辑核来处理编组操作的情况下，该序列可以被分成八个分区，给每个分区都分配相应的逻辑核以用于编组，并且每个相应的逻辑核都将基于键选择器函数和所分配分区中的元素来构造键表。

[0034] 图 5 示出了两个键表、即由第一工作者创建的键表 402 和由第二工作者创建的键表 502 的示例。键表 502 还包括一组键 504、506，其中每个键分别具有相关联的链表 508、510。每个链表 508、510 都包括诸如元素 512 和 514 之类的来自分配给第二工作者的经分区的序列的元素，这些元素都具有相同的键。示例性键表 502 中的键也是从包括 bow、brass、bucket、day、dish 和 doctor 的序列中的字母数字串的第一个字符中确定的。根据键选择器函数，键 504 是“B”，并且键 506 是“D”。在该示例中，分配给第一工作者的分区中元素的数目不同于分配给第二工作者的元素的数目。

[0035] 在各个键表已经被构造以后，合并由所述工作者创建的多个键表。在一个示例中，来自第二键表的数据被插入到第一键表中。如果某个键仅存在于第二表中，则该键和值列

表对、即该键和相关联的链表对，被插入到第一表中。在新键表被创建的示例中，该键和值列表对在该键仅存在于一个表中的情况下被插入到该新键表中。如果相同的键存在于一个以上的键表中，则将两个链表串接并且与经合并的表中的合适键相关联。例如，来自键表 502 的数据被插入到键表 402 中，从而创建图 5 中的经合并的键表 522。键 404 和相关联的链表 408 作为键 522 和相应值列表 526 被包括在经合并的键表 522 中。键 504 和相关联的链表 508 作为键 528 和相应值列表 530 被插入到经合并的键表 522 中。键 406 和 506 存在于表 402 和 502 二者中，并且作为键 532 被放置到经合并的表中。与键 406 和 506 相关联的链表 410、510 被串接以形成经合并的表 522 中的值列表 534。

[0036] 在其他示例中，两个以上的键表可以合并到一起。在不同键的数目与整个序列中元素的数目相比相对小的情况下，可以将键表顺序地组合。换言之，两个工作者的键表被合并成较大的键表，第三工作者的键表与该较大的键表合并以形成更大的键表，并且以此类推直到由每个工作者从其相应分区中创建的键表被合并成适于从编组操作中输出的单个键表。

[0037] 附加地，键表组可被递归地合并以形成适于输出的单个键表。例如，小键表组可被并发地合并以形成附加的键表，并且附加的键表可被递归地合并以便又形成附加键表，并且以此类推直到剩下单个键表。

[0038] 图 6 示出了利用五个工作者 602、604、606、608、610 的递归编组操作 600。每个工作者都接收要被编组的元素的序列 611 的分区，并且与其他工作组并行地分别创建相应的键表 612、614、616、618、620。键表的小组或子组被并发地合并在一起以创建组合的键表。在一些架构中，在小组是由工作者彼此之间的“距离”来选择的情况下可以获得优点。(距离是常常由跳(hop)、带宽、等待时间等等来度量的度量。)在一些多核处理器或处理系统中，核快速地获得对它们所接近的存储器的访问，而更远离的存储器被更慢地访问。键表 612 和 614 被合并以形成第一级 621 中的组合键表 622。与合并形成组合键表 622 并发地，键表 616 和 618 被合并以形成仍在第一级 621 中的组合键表 624。未准备好作为包括序列的每个元素的完整键表输出的键表被描述为部分键表。在所使用的示例性算法中，部分键表对被递归地合并到一起。多个部分键表对被并发地合并。然而，在该特定示例中，来自工作者 610 的部分键表 620 未与另一表合并，因为存在奇数个工作者。

[0039] 部分键表的小组被递归地合并到一起，直到它们形成完整的键表。在该递归合并期间，部分键表的多个小组在多核处理器中被并发地合并到一起。在第一递归级 631，剩下从第一级转入的三个部分键表、即键表 622、624 和 620。键表 622 和 624 被合并到一起以形成键表 632。再次，在该等级下存在奇数个键表，并且部分键表 620 转入下一递归级、即第二递归级 641。在第二递归级 641，剩下两个部分键表、即键表 632、键表 620，这些键表然后被合并到一起以形成键表 642。键表 642 是完整的键表，因为它包含了序列 611 中的所有元素，并且因此适于从编组操作 600 中输出。

[0040] 对其他编组操作的编组操作使用递归合并的优点是，该合并可以相对于合并等级的数目而言以恒定的时间完成，而不是以取决于序列中的元素数量的时间完成。在部分键表对在给定工作者数目 W 的每个等级下被合并到一起的情况下，下面的表达式用于确定用于生成完整键表的等级的数目、比如 $2^{N-1}<W<2^{N}$、或者可替代地：

[0041] N=ceil(log2(W))

[0042] 其中 N 是用于生成完整键表的等级的数目。图 6 的示例示出了使用三个合并等级(第一级 621、第一递归级 631 和第二递归级 641)的五个工作者。如果使用八个工作者创建八个键表，则在并发地合并键表对以后剩下四个部分键表。在第一递归级中，在并发地合并了部分键表对以后剩下两个部分键表。在第二递归级中，在合并了最终的两个部分键表以后剩下完整的键表。因此，在工作者的数目为 5 至 8 之间(包括 8)的情况下，使用 3 级合并来创建最终的键表。并发地用于基于更小的分区来创建键表的工作者越多，则创建每个键表的时间在总体上被降低。但是，用于创建完整键表的合并等级的数目仅仅成对数地增长。相应地，所公开的编组操作可以与现有技术的编组操作相比显著改善性能。

[0043] 尽管此处说明并描述了具体实施例，但本领域技术人员可以理解，可用各种替换和 / 或等价实现来替换所示出并描述的具体实施例而不背离本发明的范围。本申请旨在覆盖此处讨论的具体实施例的任何改编或变型。因此，本发明旨在仅由权利要求书及其等效方案来限制。

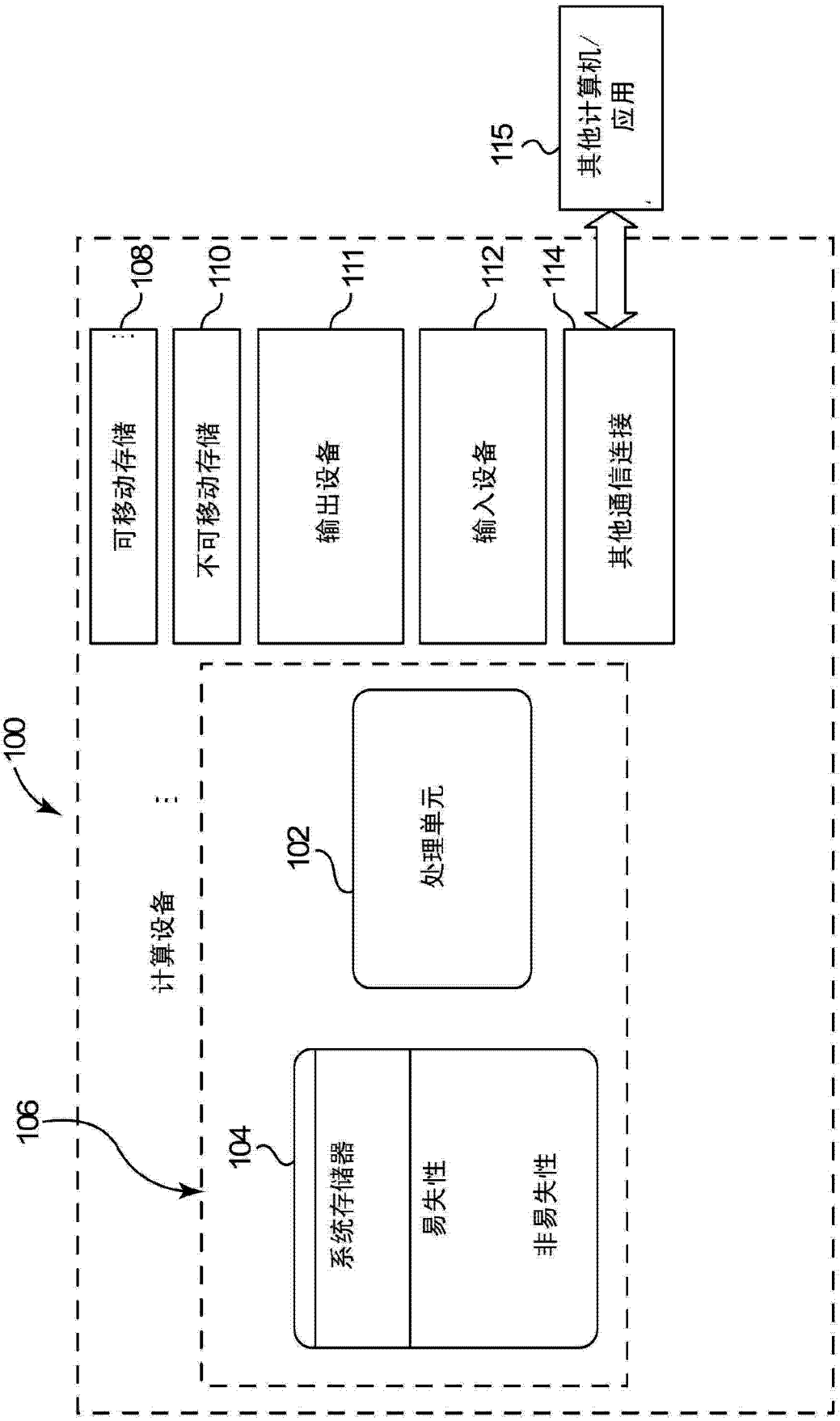

100
计算设备
102
处理单元
104
系统存储器
易失性
非易失性
106
108
可移动存储
110
不可移动存储
111
输出设备
112
输入设备
114
其他通信连接
115
其他计算机/应用

图 1

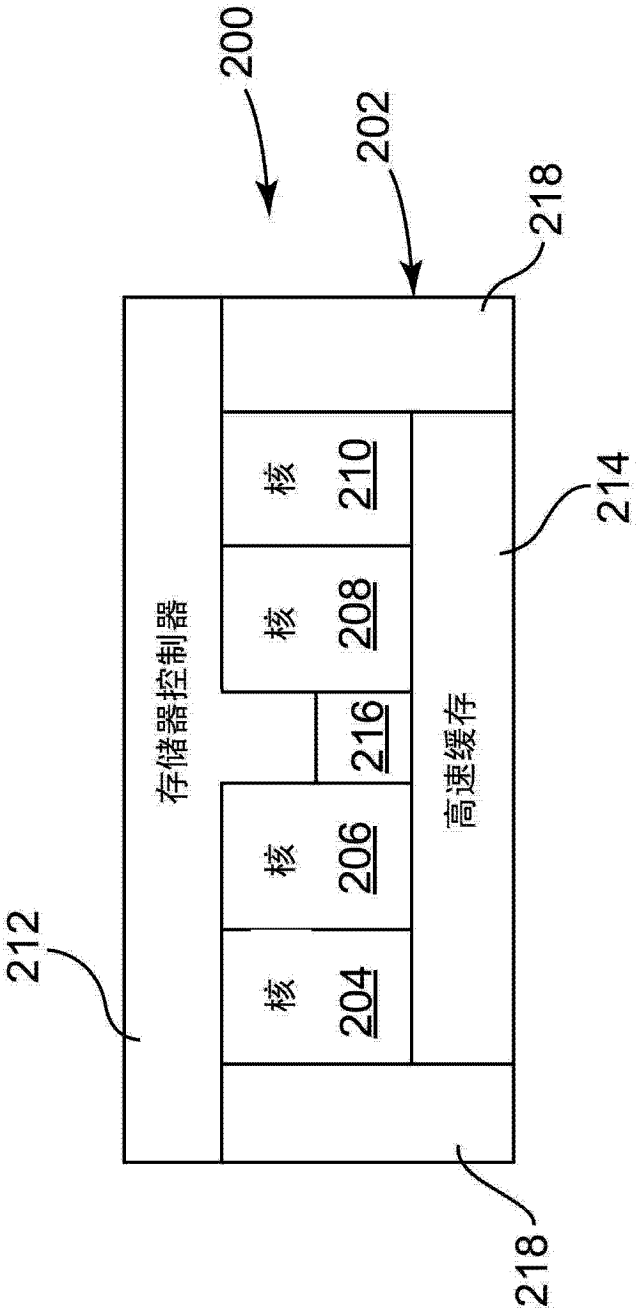


图 2

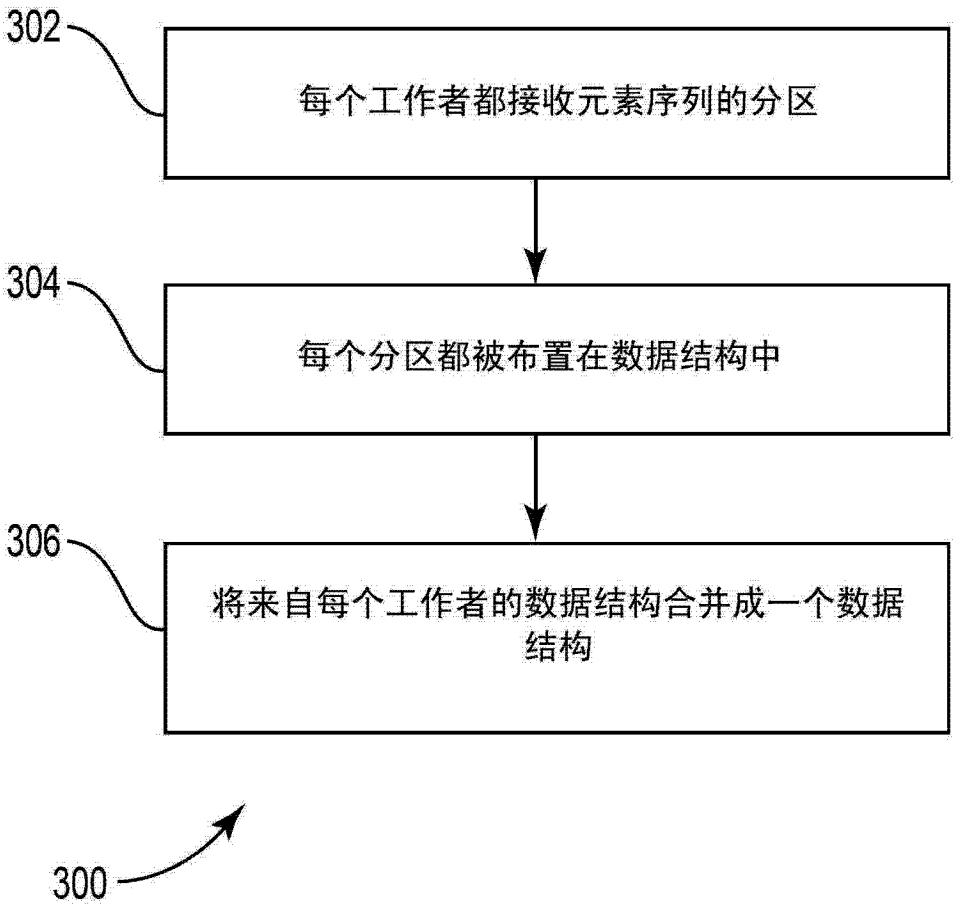

302
每个工作者都接收元素序列的分区
304
每个分区都被布置在数据结构中
306
将来自每个工作者的数据结构合并成一个数据结构
300

图 3

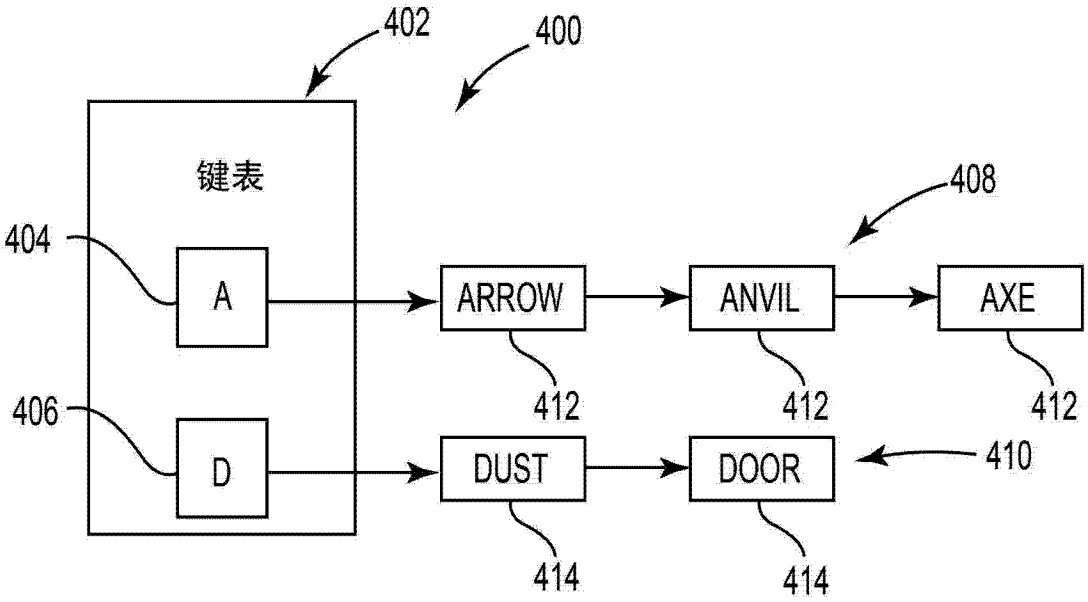

402
400
键表
404
A
406
D
ARROW
ANVIL
AXE
408
412
412
412
DUST
DOOR
410
414
414

图 4

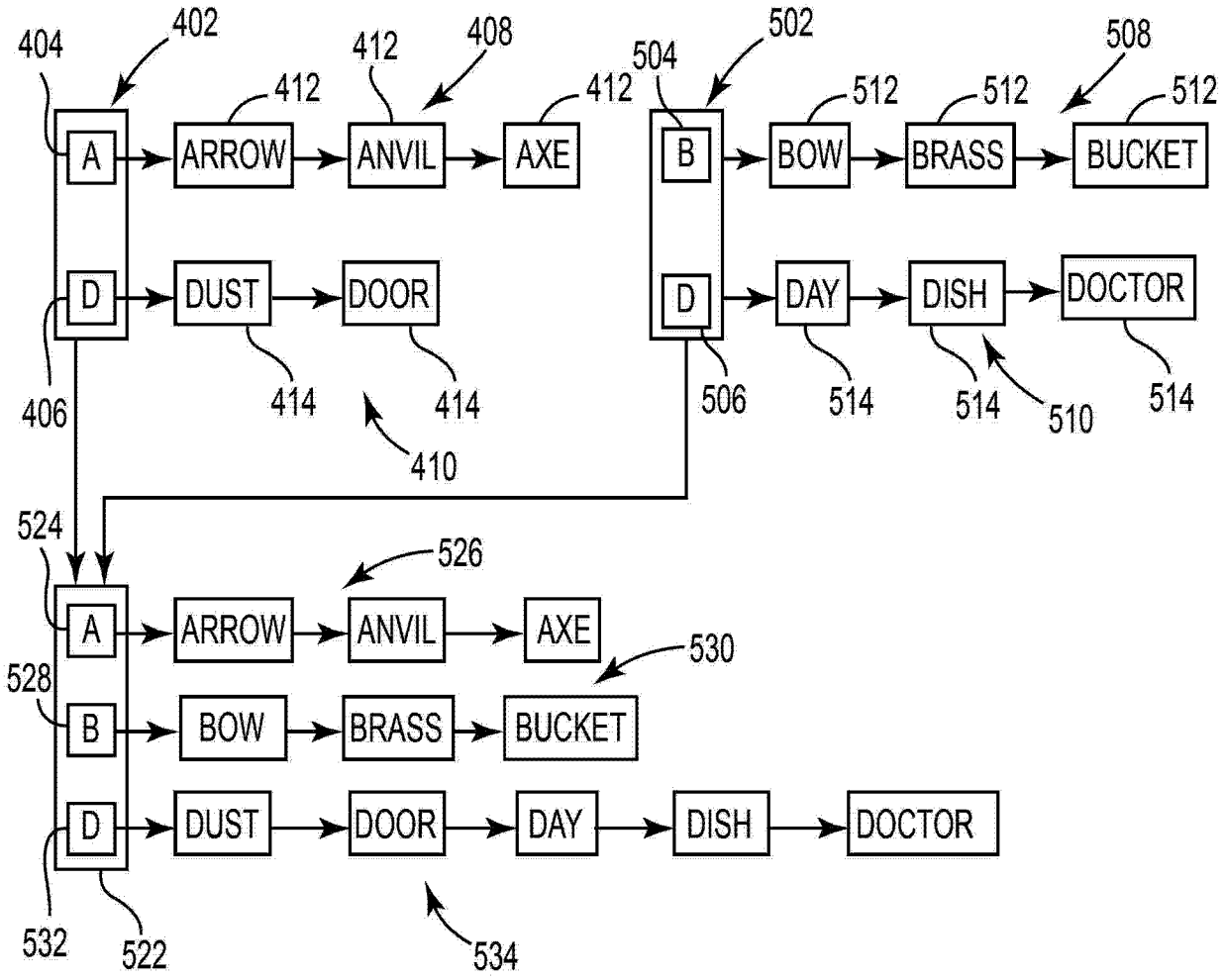

404
402
412
408
412
412
504
502
508
512
512
512
A
ARROW
ANVIL
AXE
B
BOW
BRASS
BUCKET
D
DUST
DOOR
D
DAY
DISH
DOCTOR
406
414
414
410
506
514
514
510
514
524
526
A
ARROW
ANVIL
AXE
530
528
B
BOW
BRASS
BUCKET
D
DUST
DOOR
DAY
DISH
DOCTOR
532
522
534

图 5

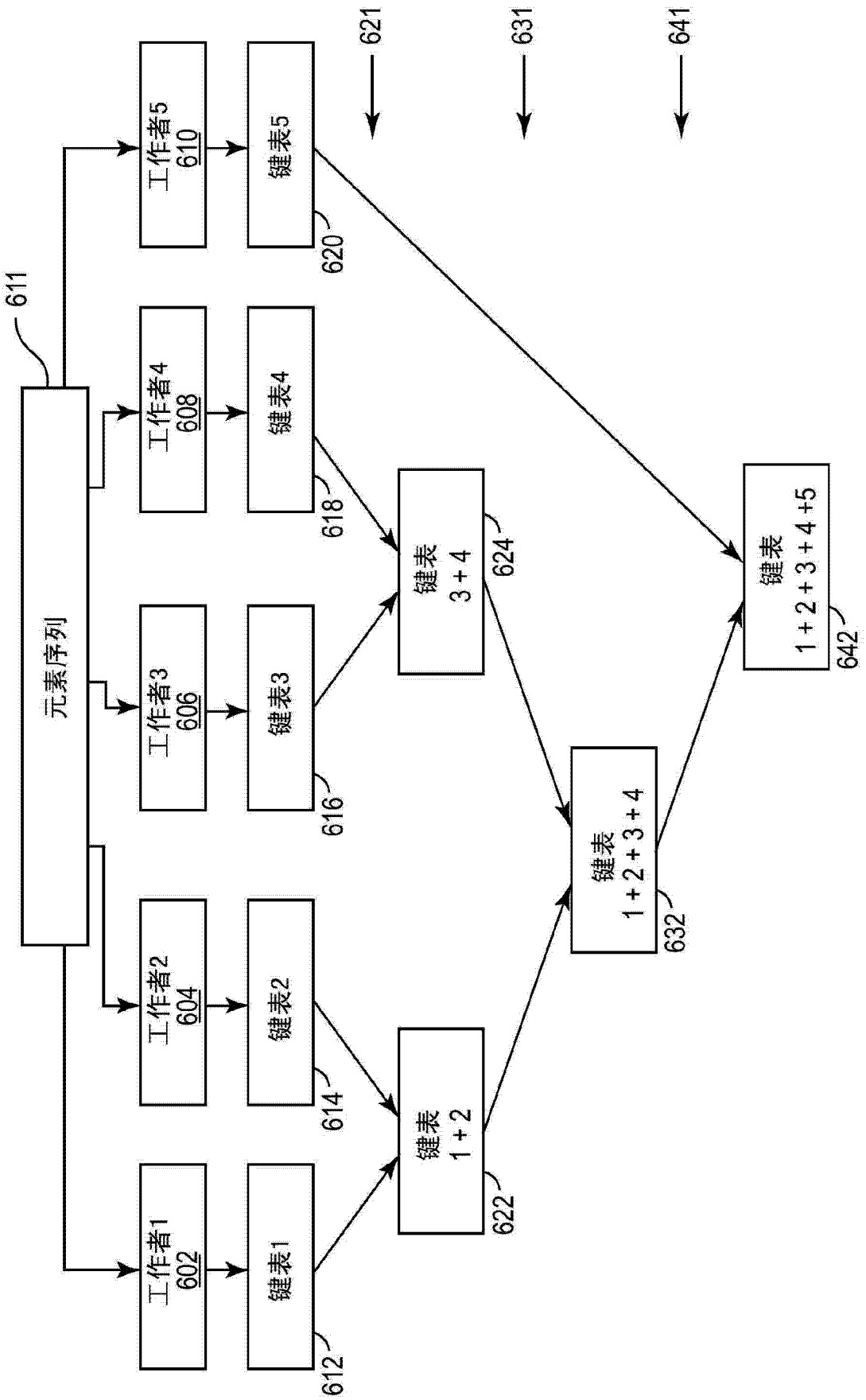

621
631
641
611
元素序列
工作者1 602
工作者2 604
工作者3 606
工作者4 608
工作者5 610
键表1
键表2
键表3
键表4
键表5
612
614
616
618
620
键表 1 + 2
键表 3 + 4
键表 1 + 2 + 3 + 4
键表 1 + 2 + 3 + 4 +5
622
624
632
642

图 6