(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 103246508 B
(45)授权公告日 2016.06.22

(21)申请号 201310049241.0
(22)申请日 2013.02.07
(30)优先权数据
13/369,451 2012.02.09 US
(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 C.J.阿彻 J.E.凯里 P.J.桑德斯 B.E.史密斯
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 张丽新
(51)Int.Cl.
G06F 9/44(2006.01)
G06F 9/38(2006.01)
(56)对比文件
US 2007/0294666 A1,2007.12.20,
US 7509244 B1,2009.03.24,
CN 101055532 A,2007.10.17,
CN 101055535 A,2007.10.17,
CN 101055536 A,2007.10.17,
审查员 张静

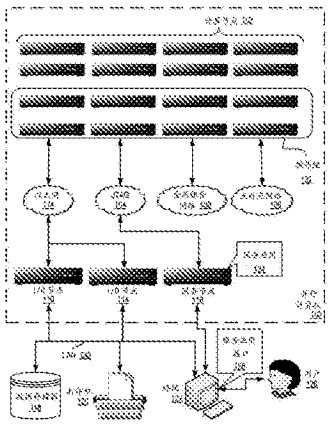
权利要求书2页 说明书13页 附图7页

(54)发明名称

用于开发并行计算机的集合操作的方法和装置

(57)摘要

开发包括多个计算节点的并行计算机的集合操作包括:由集合开发工具向集合开发者呈现图形用户界面("GUI");由该集合开发工具通过GUI从该集合开发者接收对于一个或多个集合基元的选择;由该集合开发工具通过GUI从该集合开发者接收集合基元的连续顺序的详细说明以及对于每个集合基元的输入和输出缓冲器的详细说明;以及由该集合开发工具依赖于所述对于集合基元的选择、集合基元的连续顺序以及对于每个集合基元的输入和输出缓冲器产生执行由所述集合基元指定的集合操作的可执行代码。



CN 103246508 B

1.一种开发包括多个计算节点的并行计算机的集合操作的方法，该方法包括：

由集合开发工具根据对一个或多个集合基元的选择、该一个或多个集合基元的连续顺序以及每个集合基元的输入缓冲器和输出缓冲器的详细说明，产生执行由所述集合基元指定的集合操作的可执行代码；

将集合基元的连续顺序转换成计算机程序指令的一个或多个集合基元模块的执行顺序；

在可执行文件中按该执行顺序插入该计算机程序指令的一个或多个集合基元模块；以及

在可执行文件中的该一个或多个集合基元模块之间插入一个或多个粘结模块，用于在集合操作的执行期间链接集合基元模块，其中，每个粘结模块是基于要在其上执行集合操作的并行计算机的属性而选择的。

2.如权利要求1的方法，还包括：

由该集合开发工具向集合开发者呈现图形用户界面GUI；

由该集合开发工具通过GUI从该集合开发者接收对于一个或多个集合基元的选择；

由该集合开发工具通过GUI从该集合开发者接收集合基元的连续顺序的详细说明以及每个集合基元的输入缓冲器和输出缓冲器的详细说明，

其中，由该集合开发工具通过GUI从集合开发者接收对于一个或多个集合基元的选择还包括：

由所述集合开发工具通过GUI检测指示对于与所述集合基元有关的一个或多个图形图标的选择的用户输入设备活动。

3.如权利要求1的方法，其中至少一个集合基元包括：

多同步基元，该多同步基元在被执行时执行多个计算节点之间的同步。

4.如权利要求1的方法，其中至少一个集合基元包括：

多播基元，该多播基元在被执行时并行地向一组节点发送消息。

5.如权利要求1的方法，其中至少一个集合基元包括：

多组合基元，该多组合基元在被执行时对从多于一个计算节点接收的数据进行操作。

6.如权利要求1的方法，其中至少一个集合基元包括：

多对多基元，该多对多基元在被执行时向一组计算节点发送唯一的数据并从另一组计算节点接收数据。

7.一种用于开发包括多个计算节点的并行计算机的集合操作的装置，该装置包括计算机处理器和操作地耦接到该计算机处理器的计算机存储器，该计算机存储器具有布置在其中的计算机程序指令，该计算机程序指令在被该计算机处理器执行时致使所述装置执行以下步骤：

由集合开发工具根据对一个或多个集合基元的选择、该一个或多个集合基元的连续顺序以及每个集合基元的输入缓冲器和输出缓冲器的详细说明，产生执行由所述集合基元指定的集合操作的可执行代码；

将集合基元的连续顺序转换成计算机程序指令的一个或多个集合基元模块的执行顺序；

在可执行文件中按该执行顺序插入该计算机程序指令的一个或多个集合基元模块；以

及

在可执行文件中的该一个或多个集合基元模块之间插入一个或多个粘结模块，用于在集合操作的执行期间链接集合基元模块，其中，每个粘结模块是基于要在其上执行集合操作的并行计算机的属性而选择的。

8.如权利要求7的装置，该计算机程序指令在被该计算机处理器执行时致使所述装置还执行以下步骤：

由该集合开发工具向集合开发者呈现图形用户界面GUI；

由该集合开发工具通过GUI从该集合开发者接收对于一个或多个集合基元的选择；

由该集合开发工具通过GUI从该集合开发者接收集合基元的连续顺序的详细说明以及每个集合基元的输入缓冲器和输出缓冲器的详细说明，

其中，由该集合开发工具通过GUI从集合开发者接收对于一个或多个集合基元的选择还包括：

由所述集合开发工具通过GUI检测指示对于与所述集合基元有关的一个或多个图形图标的选择的用户输入设备活动。

9.如权利要求7的装置，其中至少一个集合基元包括：

多同步基元，该多同步基元在被执行时执行多个计算节点之间的同步。

10.如权利要求7的装置，其中至少一个集合基元包括：

多播基元，该多播基元在被执行时并行地向一组节点发送消息。

11.如权利要求7的装置，其中至少一个集合基元包括：

多组合基元，该多组合基元在被执行时对从多于一个计算节点接收的数据进行操作。

12.如权利要求7的装置，其中至少一个集合基元包括：

多对多基元，该多对多基元在被执行时向一组计算节点发送唯一的数据并从另一组计算节点接收数据。

用于开发并行计算机的集合操作的方法和装置

技术领域

[0001] 本发明的领域是数据处理，更具体地，是用于开发包括多个计算节点的并行计算机的集合操作的方法、装置和产品。

背景技术

[0002] 1948年EDVAC计算机系统的开发通常被称为计算机时代的开始。自从那时起，计算机系统已经发展为极其复杂的设备。当今的计算机比诸如EDVAC的早期系统复杂得多。计算机系统通常包括硬件和软件组件、应用程序、操作系统、处理器、总线、存储器、输入/输出设备等等的组合。随着半导体处理和计算机架构的进步将计算机的性能推向越来越高，更复杂的计算机软件已经发展为利用更高性能的硬件，得到比仅几年前强大得多的当今的计算机系统。

[0003] 并行计算是已经经历进步的计算机技术领域。并行计算是同一个任务(被分割并具体适配)在多个处理器上的同时执行以便更快地获得结果。并行计算是基于以下事实：解决问题的处理通常可以被划分为更小的任务，这些更小的任务可以通过某些协调而同时执行。

[0004] 并行计算机执行并行算法。并行算法可以被分割以在许多不同的处理设备上每次一段地执行，然后最终再收回到一起以得到数据处理结果。一些算法容易划分成段(piece)。划分检查从1到十万的所有数字以查看哪些是质数的任务可以通过例如向每个可用的处理器分配数字的子集然后将肯定结果的列表放回在一起来进行。在此说明书中，执行并行程序的各个段的多个处理设备被称为“计算节点”。并行计算机由计算节点以及其他处理节点构成，其他处理节点包括例如输入/输出(“I/O”)节点以及服务节点。

[0005] 并行算法是有价值的，因为由于现代处理器工作的方式，经由并行算法比经由串行(非并行)算法更快地进行某些种类的大的计算任务。构造具有单个快速处理器的计算机远比构造具有相同吞吐量的具有许多慢处理器的计算机难得多。还存在对串行处理器的潜在速度的某些理论限制。另一方面，每个并行算法具有串行部分，因此串行算法具有饱和点。在该点之后添加更多的处理器不能产生任何更多的吞吐量，而仅仅是增加开销(overhead)和成本。

[0006] 并行算法还被设计用于优化更多一个资源、即并行计算机的节点之间的数据通信要求。存在两种并行处理器通信方式，共享存储器或者消息传递。共享存储器处理需要对于数据的另外锁定并且强加了另外的处理器和总线循环的开销，并且还串行化该算法的某些部分。

[0007] 消息传递处理使用高速数据通信网络和消息缓存，但是此通信在数据通信网络上添加了传送开销以及消息缓存的另外的存储器需要和节点之间的数据通信中的等待时间。并行计算机的设计使用专门设计的数据通信链接以便通信开销将是小的，但是决定流量的是并行算法。

[0008] 许多数据通信网络架构被用于并行计算机中的节点之间的消息传递。计算节点可

以在网络中被组织为例如“环状结构(torus)”或者“网状结构(mesh)”。而且,计算节点在网络中可以被组织为树。环状网络(mesh network)用环绕的链接来连接三维网状结构中的节点。每个节点通过该环状网络连接到其六个相邻者,每个节点通过其在网状结构中的x、y、z坐标被寻址。以这样的方式,环状网络使其自身适合于点对点操作。在树状网络中,节点通常连接为二叉树:每个节点具有一个双亲和两个孩子(尽管依赖于硬件配置,一些节点可能仅具有0个孩子或者一个孩子)。尽管树状网络通常在点对点通信中效率低,但是树状网络提供高带宽和以及对于某些集合(collective)操作、其中所有计算节点同时参与的消息传递操作诸如例如全聚集(allgather)操作的低等待时间。在使用环状网络和树状网络的计算机中,这两个网络通常彼此独立地实现,具有分离的布线电路、分离的物理链接以及分离的消息缓冲器。

发明内容

[0009] 本说明书中公开了用于开发包括多个计算节点的并行计算机的集合操作的方法、装置和产品。根据本发明的实施例开发这样的集合操作包括:由集合开发工具向集合开发者呈现图形用户界面(“GUI”);由该集合开发工具通过GUI从该集合开发者接收对于一个或多个集合基元的选择;由该集合开发工具通过GUI从该集合开发者接收集合基元的连续顺序的详细说明以及对于每个集合基元的输入和输出缓冲器的详细说明;以及由该集合开发工具依赖于所述对于集合基元的选择、集合基元的连续顺序以及对于每个集合基元的输入和输出缓冲器产生执行由所述集合基元指定的集合操作的可执行代码。

[0010] 本发明的以上以及其他目标、特征和优点将从如在附图中例示的本发明的示例实施例的以下更具体的描述而显而易见,附图中类似的参考标记一般表示本发明的示例实施例的类似的部分。

附图说明

[0011] 图1例示根据本发明的实施例的开发并行计算机的集合操作的示例系统。

[0012] 图2给出根据本发明的实施例在配置为开发集合操作的并行计算机中有用的示例计算节点的框图。

[0013] 图3A给出根据本发明的实施例在用于开发并行计算机的集合操作的系统中有用的示例点对点适配器的框图。

[0014] 图3B给出根据本发明的实施例在用于开发并行计算机的集合操作的系统中有用的示例全局组合网络适配器的框图。

[0015] 图4给出例示根据本发明的实施例在能够开发并行计算机的集合操作的系统中有用的对于点对点操作优化的示例数据通信网络的线条图。

[0016] 图5给出例示根据本发明的实施例在能够开发并行计算机的集合操作的系统中有用的示例全局组合网络的线条图。

[0017] 图6给出根据本发明的实施例的用于开发并行计算机的集合操作的示例方法的流程图。

[0018] 图7给出根据本发明的实施例由支持开发并行计算机的集合操作的集合开发工具给出的示例图形用户界面的线条图。

具体实施方式

[0019] 参考附图描述根据本发明的用于开发并行计算机的集合操作的示例方法、装置和产品，以图1开始。图1例示根据本发明的实施例的可以为其开发集合操作并且可以在其中开发集合操作的示例系统。图1的系统包括并行计算机(100)、数据存储设备(118)形式的用于该计算机的非易失性存储器、打印机(120)形式的用于该计算机的输出设备以及计算机终端(122)形式的用于该计算机的输入/输出设备。

[0020] 在图1的例子中的并行计算机(100)包括多个计算节点(102)。计算节点(102)通过几种独立的数据通信网络耦接用于数据通信，该几种独立的数据通信网络包括高速以太网络(174)、联合测试行动组(“JTAG”)网络(104)、使用二叉树网络拓扑的对于集合操作优化的全局组合网络(106)、以及使用环状网络拓扑的对于点对点操作优化的点对点网络(108)。全局组合网络(106)是包括连接到计算节点(102)的数据通信链接以便将计算节点(102)组织为二叉树的数据通信网络。每个数据通信网络利用在计算节点(102)之间的数据通信链接实现。数据通信链接提供用于并行计算机(100)的计算节点(102)之间的并行操作的数据通信。

[0021] 并行计算机(100)的计算节点(102)被组织为用于并行计算机(100)上的集合并行操作的计算节点的至少一个操作组(132)。计算节点的每个操作组(132)是这样的计算节点的集合：集合并行操作在这些节点上执行。操作组(132)中的每个计算节点被分配了标识该操作组(132)中的特定计算节点的唯一秩(rank)。利用操作组的计算节点之间的数据通信实现集合操作。集合操作是涉及操作组(132)的所有计算节点的那些功能。集合操作是由计算节点的操作组(132)中的所有计算节点同时、即近似在相同时间执行的操作、即消息传递型计算机程序指令。这样的操作组(132)可以包括并行计算机(100)中的所有计算节点(102)或者所有计算节点(102)的子集。集合操作通常围绕点对点操作而建立。集合操作需要在操作组(132)内的所有计算节点上的所有处理利用匹配的变量调用相同的集合操作。“广播”是用于在操作组的计算节点之间移动数据的集合操作的例子。“归约(reduce)”操作是对分布在操作组(132)的计算节点之间的数据执行算术或逻辑功能的集合操作的例子。操作组(132)可以实现为例如MPI“通信器”。

[0022] “MPI”指“消息传递接口”，即现有技术的并行通信库，即用于并行计算机上的数据通信的计算机程序指令的模块。可以改进以用在根据本发明的实施例而配置的系统中的现有技术的并行通信库的例子包括MPI和“并行虚拟机”(“PVM”)库。PVM由田纳西(Tennessee)大学、橡树岭国家实验室(Oak Ridge National Laboratory)和埃默里(Emory)大学开发。MPI由MPI论坛公布，MPI论坛即具有来自定义和维护MPI标准的许多组织的代表的开放组。在撰写本文时MPI是用于在分布式存储器并行计算机上运行并行程序的计算节点之间通信的实际标准。为了易于说明，本说明书有时使用MPI术语，不过像这样使用MPI不是本发明的要求或者限制。

[0023] 一些集合操作具有在操作组(132)中的特定计算节点上运行的单个发起或接收处理。例如，在“广播”集合操作中，在计算节点上的将数据分发到所有其他计算节点的处理是发起处理。例如，在“聚集”操作中，在计算节点上的接收来自其他计算节点的所有数据的处理是接收处理。这样的发起或者接收处理在其上运行的计算节点被称为逻辑根。

[0024] 大多数集合操作是四个基本操作的变型或组合:广播、聚集、散布和归约。用于这些集合操作的接口在由MPI论坛公布的MPI标准中定义。但是,在该MPI标准中未定义用于执行集合操作的算法。在广播操作中,所有处理指定相同的根处理,其缓冲器内容将被发送。除了根之外的处理指定接收缓冲器。在该操作之后,所有缓冲器包含来自该根处理的消息。

[0025] 像广播操作那样,散布操作也是一对多集合操作。在散布操作中,逻辑根将根上的数据划分为片段并且将不同的片段分发到操作组(132)中的每个计算节点。在散布操作中,所有处理通常指定相同的接收计数。发送变量仅对于根处理是重要的,该根处理的缓冲器实际上包含了给定数据类型的发送计数*N元素,其中N是给定组的计算节点中的处理的数量。发送缓冲器被划分并散发到所有处理(包括在逻辑根上的处理)。每个计算节点被分配了称为"秩"的顺序标识符。在该操作之后,根已经按增加的秩的顺序向每个处理发送了发送计数(sendcount)数据元素。秩0接收来自发送缓冲器的第一发送计数数据元素。秩1接收来自发送缓冲器的第二发送计数数据元素,等等。

[0026] 聚集操作是与散步操作的描述完全相反的多对一集合操作。即,聚集是多对一集合操作,其中将来自排序了(ranked)的计算节点的某数据类型的元素聚集到根节点中的接收缓冲器中。

[0027] 归约操作也是多对一集合操作,其包括对两个数据元素进行的算术或逻辑功能。所有处理指定相同的"计数"和相同的算术或逻辑功能。在该归约之后,所有处理已将来自计算节点发送缓冲器的计数数据元素发送到根处理。在归约操作中,来自相应的发送缓冲器位置的数据元素通过算术或逻辑操作被成对地组合以产生根处理的接收缓冲器中的单个相应的元素。可以在运行时定义专用归约操作。并行通信库可以支持预定操作。例如,MPI提供以下预定归约操作:

[0028]

MPI_MAX	最大
MPI_MIN	最小
MPI_SUM	和
MPI_PROD	积
MPI_LAND	逻辑与

[0029]

MPI_BAND	按位与
MPI_LOR	逻辑或
MPI_BOR	按位或
MPI_LXOR	逻辑异或
MPI_BXOR	按位异或

[0030] 除了计算节点之外,并行计算机(100)还包括通过全局组合网络(106)耦接到计算节点(102)的输入/输出("I/O")节点(110,114)。并行计算机(100)中的计算节点(102)可以被划分为处理集以便为了数据通信而将处理集中的每个计算节点连接到相同的I/O节点。因此,每个处理集由一个I/O节点和计算节点(102)的子集组成。整个系统中的计算节点的数量与I/O节点的数量之间的比率通常依赖于并行计算机(102)的硬件配置。例如,在一些

配置中，每个处理集可以由八个计算节点和一个I/O节点组成。在一些其他配置中，每个处理集可以由六十四个计算节点和一个I/O节点组成。但是，这样的例子仅仅是用于说明而不是限制。每个I/O节点提供在其处理集的计算节点(102)和I/O设备集之间的I/O服务。在图1的例子中，I/O节点(110,114)通过使用高速以太网实现的局域网(“LAN”)(130)而连接用于数据通信到I/O设备(118,120,122)。

[0031] 图1的并行计算机(100)还包括经过网络(104)之一耦接到计算节点的服务节点(116)。服务节点(116)提供对多个计算节点共同的服务、管理计算节点的配置、将程序加载到计算节点中、开始计算节点上的程序执行、取回计算节点上的程序操作的结果，等等。服务节点(116)运行服务应用(124)并通过在计算终端(122)上运行的服务应用接口(126)与用户(128)通信。

[0032] 图1的并行计算机(100)支持根据本发明的实施例开发的集合操作。这样的集合操作可以在并行计算机(100)本身的计算节点(102)上或者在其他自动的计算机器上开发。根据本发明的实施例开发集合操作包括：由集合开发工具向集合开发者呈现图形用户界面(“GUI”)。在此说明书中使用术语“集合开发者”指代开发可在执行集合操作的并行计算机上执行的软件的用户(128)。集合开发工具是在执行时致使包括计算机硬件和软件的聚合的自动计算机器执行根据本发明的实施例的集合开发的计算机程序指令的模块。

[0033] 集合开发工具还通过GUI从集合开发者接收对于一个或多个集合基元的选择。集合基元是在执行时执行预定的集合任务的计算机程序指令的模块。每个基元是可以按各种方式与其他基元组合以形成合成的集合操作的构建块。这样的集合基元的例子包括：

[0034] • 多同步基元，当执行时，执行多个计算节点之间的同步；

[0035] • 多播基元，当执行时，并行地向一组节点发送消息；

[0036] • 多组合基元，当执行时，对从多于一个计算节点接收的数据进行操作；以及

[0037] • 多对多基元，当执行时，向一组计算节点发送唯一的数据并从另一组计算节点接收数据。

[0038] 集合开发工具在接收到一个或多个集合基元的详细说明之后还通过GUI从集合开发者接收集合基元的连续顺序的详细说明以及对于每个集合基元的输入和输出缓冲器的详细说明。集合基元的顺序指定一个接一个的集合基元的执行的顺序。输入和输出缓冲器的详细说明可以按各种方式实现，包括例如作为表示具体大小的阵列的全局变量或指针的定义。

[0039] 然后集合开发工具依赖于集合基元的选择、集合基元的连续顺序以及每个集合基元的输入和输出缓冲器产生执行由集合基元指定的集合操作的可执行代码。

[0040] 集合开发工具使得集合开发者能够在图形界面中而不是通过乏味的源代码的合成来开发集合操作。以此方式，复杂的集合操作可以由开发者迅速并有效地“建立”而不需要开发者编写将被执行以实施集合操作的每个计算机程序指令。

[0041] 通常为并行计算机实现根据本发明的实施例开发并行计算机的集合操作，并行计算机包括通过至少一个数据通信网络组织用于集合操作的多个计算节点。此外，对于并行计算机开发集合操作也可以在这样的并行计算机中执行。事实上，这样的计算机可以包括数千个计算节点。每个计算节点本身又是一种由一个或多个计算处理核、其自己的计算机存储器以及其自己的输入/输出适配器组成的计算机。为了进一步说明，图2给出在根据本

发明的实施例能够开发集合操作的并行计算机中有用的示例计算节点(102)的框图。图2的计算节点(102)包括多个处理核(165)以及RAM(156)。图2的处理核(165)可以配置在一个或多个集成电路晶片上。处理核(165)通过高速存储器总线(155)连接到RAM(156)并通过总线适配器(194)和扩展总线(168)连接到计算节点的其他组件。存储在RAM(156)中的是应用程序(226),即使用并行算法执行并行的用户级数据处理的计算机程序指令的模块。

[0042] 还存储在RAM(156)中的是并行通信库(161),即在计算节点之中执行并行通信、包括点对点操作以及集合操作的计算机程序指令的库。可以使用诸如C编程语言的传统编程语言以及使用传统的编程方法编写在两个独立数据通信网络上的节点之间发送和接收数据的并行通信例程来从头开发并行通信例程的库以用在根据本发明的实施例的系统中。或者,可以改进已有的现有技术库以根据本发明的实施例而操作。现有技术的并行通信库的例子包括“消息传递接口”(“MPI”)库以及“并行虚拟机”(“PVM”)库。

[0043] 还存储在计算节点(102)的RAM(156)中的是集合开发工具(228),即当执行时致使图2的计算节点(102)支持根据本发明的实施例的集合操作开发的计算机程序指令的模块。具有本领域知识的读者将认识到,仅仅为了说明而不是限制的目的,图2的集合开发工具(228)在计算节点上执行。根据本发明的实施例配置的集合开发工具可以在任何自动计算机器上执行。

[0044] 图2的集合开发工具(228)通过以下支持根据本发明的实施例的集合操作开发:向集合开发者呈现GUI(230);通过GUI(230)从集合开发者接收对于一个或多个集合基元的选择(232);通过GUI(230)从集合开发者接收集合基元的连续顺序的详细说明(234)以及对于每个集合基元的输入和输出缓冲器的详细说明(236);以及依赖于集合基元的选择(232)、集合基元的连续顺序(234)以及对于每个集合基元的输入和输出缓冲器(236)产生执行由集合基元指定的集合操作的可执行代码(238)。在一些示例实施例中,可执行代码(238)可以被包括为并行通信库(161)中的库函数、被包括为并行应用(226)的模块、或者按具有本领域知识的读者将想到的其他方式被包括。

[0045] 还存储在RAM(156)中的是操作系统(162),即用于应用程序对计算节点的其他资源的访问的计算机程序指令和例程的模块。并行计算机的计算节点中的应用程序和并行通信库典型地运行执行的单线程而不用用户登录并没有安全问题,因为该线程被赋予对于该节点的所有资源的完全访问的权利。因此并行计算机中的计算节点上的操作系统要执行的任务的数量以及复杂性比具有同时运行的许多线程的串行计算机上的操作系统的任务更小并且更不复杂。另外,在图2的计算节点(102)上没有视频I/O,即降低对操作系统的要求的另一个因素。因此,与通用计算机相比,操作系统(162)可以是非常轻量级的(lightweight)削减版本的操作系统或者专门为了具体的并行计算机上的操作而开发的操作系统。可以有用地改进、简化以用在计算节点中的操作系统包括UNIX™、Linux™、Windows XP™、AIX™、IBM的i/OS™以及如本领域技术人员将想到的其他操作系统。

[0046] 图2的示例计算节点(102)包括用于实现与并行计算机的其他节点的数据通信的几个通信适配器(172,176,180,188)。这样的数据通信可以通过RS-232连接、通过诸如USB的外部总线、通过诸如IP网络的数据通信网络以及按本领域技术人员将想到的其他方式串行地执行。通信适配器实现硬件级的数据通信,通过该硬件级的数据通信,一个计算机直接地或者通过网络向另一计算机发送数据通信。在根据本发明的实施例为其开发集合操作的

并行计算机中的装置中有用的通信适配器的例子包括用于有线通信的调制解调器、用于有线网络通信的以太网(IEEE802.3)适配器以及用于无线网络通信的802.11b适配器。

[0047] 图2的例子中的数据通信适配器包括为了数据通信而将示例的计算节点(102)耦接到吉比特以太网(174)的吉比特以太网适配器(172)。吉比特以太网是提供每秒十亿比特(一个吉比特)的数据速率的在IEEE802.3中定义的网络传输标准。吉比特以太网是在多模光纤电缆、单模光纤电缆或者非屏蔽双绞线上工作的以太网的变型。

[0048] 图2的例子中的数据通信适配器包括为了数据通信将示例的计算节点(102)耦接到JTAG主电路(178)的JTAG从电路(176)。JTAG是名为标准测试访问端口和边界扫描架构的IEEE1149.1标准的普通名称,用于使用边界扫描来测试印刷电路板的测试访问端口。JTAG如此广泛适用使得这时边界扫描或多或少与JTAG意思相同。JTAG不仅用于印刷电路板,而且还用于进行集成电路的边界扫描,并且作为用于调试嵌入系统、提供进入系统中的便利的替换接入点的机制也是有用的。图2的示例的计算节点可以是这些中的所有三个:其通常包括安装在印刷电路板上的一个或多个集成电路并且可以实现为具有其自己的处理核、其自己的存储器以及其自己的I/O能力的嵌入式系统。通过JTAG从属(176)的JTAG边界扫描可以有效地配置计算节点(102)中的处理核寄存器和存储器以用于将连接的节点动态地再分配到在根据本发明的实施例的为其开发集合操作的系统中有用的计算节点的块。

[0049] 图2的例子中的数据通信适配器包括点对点网络适配器(180),该点对点网络适配器(180)为了数据通信将示例的计算节点(102)耦接到诸如例如配置为三维环状结构或者网状结构的网络的对于点对点消息传递操作最佳的网络(108)。点对点适配器(180)提供通过以下六个双向链接在三个通信轴x、y和z上的在六个方向上的数据通信:+x(181)、-x(182)、+y(183)、-y(184)、+z(185)和-z(186)。

[0050] 图2的例子中的数据通信适配器包括全局组合网络适配器(188),该全局组合网络适配器(188)为了数据通信将示例的计算节点(102)耦接到诸如例如配置为二叉树的网络的对于集合消息传递操作最佳的全局组合网络(106)。全局组合网络适配器(188)通过三个双向链接为该全局组合网络适配器(188)支持的每个全局组合网络(106)提供数据通信。在图2的例子中,全局组合网络适配器(188)通过以下三个双向链接为全局组合网络(106)提供数据通信:两个到孩子节点(190)以及一个到双亲节点(192)。

[0051] 示例的计算节点(102)包括多个算术逻辑单元("ALU")。每个处理核(165)包括ALU(166),并且分离的ALU(170)专用于全局组合网络适配器(188)的独占使用,以供用在进行归约操作、包括全局归约(allreduce)操作的算术和逻辑功能中。在该并行通信库(161)中的归约例程的计算机程序指令可以将用于算术或逻辑功能的指令锁存到指令寄存器(169)中。当归约操作的算术或者逻辑功能是例如"求和"或者"逻辑或"时,集合操作适配器(188)可以使用由全局组合网络(106)上的节点(190,192)提供的数据和由计算节点(102)上的处理核(165)提供的数据、通过使用在处理核(165)中的ALU(166)或者通常更快地通过使用专用ALU(170)来执行算术或逻辑操作。

[0052] 但是,通常仅当在全局组合网络适配器(188)中进行算术操作时,全局组合网络适配器(188)仅用于组合从孩子节点(190)接收的数据并将该结果沿该网络(106)向上传递到双亲节点(192)。类似地,全局组合网络适配器(188)可以仅用于传输从双亲节点(192)接收的数据并将该数据沿该网络(106)向下传递到孩子节点(190)。也就是说,计算节点(102)上

的处理核(165)都不贡献(contribute)更改ALU(170)的输出的数据，该ALU(170)的输出然后沿该全局组合网络(106)向上或向下传递。因为ALU(170)通常在ALU(170)接收到来自处理核(165)之一的输入之前不输出任何数据到网络(106)上，所以处理核(165)可以将身份元素注入到专用(ALU)170中以在ALU(170)中进行特定算术操作，以便防止更改ALU(170)的输出。但是将身份元素注入ALU中通常消耗多个处理循环。为了在这些情况下进一步增强性能，示例的计算节点(102)包括用于将身份元素注入ALU(170)中以降低防止更改ALU输出所需的处理核资源量的专用硬件(171)。该专用硬件(171)注入与该ALU进行的特定算术操作对应的身份元素。例如，当全局组合网络适配器(188)对从孩子节点(190)接收的数据进行按位或时，专用硬件(171)可以将0注入ALU(170)中以改进整个全局组合网络(106)的性能。

[0053] 为了进一步说明，图3A给出在根据本发明的实施例为其开发集合操作的系统中有用的示例点对点适配器(180)的框图。该点对点适配器(180)设计为用在对于点对点操作优化的数据通信网络、即按三维环状结构或网状结构组织计算节点的网络中。图3A的例子中的点对点适配器(180)提供沿着x轴通过四个单向数据通信链接的到-x方向上的下一节点(182)和来自该下一节点(182)、以及到+x方向上的下一节点(181)和来自该下一节点(181)的数据通信。图3A的点对点适配器(180)还提供沿着y轴通过四个单向数据通信链接的到-y方向上的下一节点(184)和来自该下一节点(184)、以及到+y方向上的下一节点(183)和来自该下一节点(183)的数据通信。图3A的点对点适配器(180)还提供沿着z轴通过四个单向数据通信链接到的-z方向上的下一节点(186)和来自该下一节点(186)、以及到+z方向上的下一节点(185)以及来自该下一节点(185)的数据通信。

[0054] 为了进一步说明，图3B给出在根据本发明的实施例为其开发集合操作的系统中有用的示例的全局组合网络适配器(188)的框图。全局组合网络适配器(188)设计为用在对于集合操作优化的网络、即按二叉树组织并行计算机的计算节点的网络中。在图3B的例子中的全局组合网络适配器(188)提供通过四个单向数据通信链接的到全局组合网络的孩子节点(190)以及来自这些孩子节点(190)的数据通信，以及还提供通过两个单向数据通信链接的到全局组合网络的双亲节点(192)以及来自该双亲节点(192)的数据通信。

[0055] 为了进一步说明，图4给出例示在根据本发明的实施例的为其开发集合操作的系统中有用的对于点对点操作优化的示例的数据通信网络(108)的线条图。在图4的例子中，点表示并行计算机的计算节点(102)，并且点之间的虚线表示计算节点之间的数据通信链接(103)。利用与例如图3A中例示的点对点数据通信适配器类似的点对点数据通信适配器来实现这些数据通信链接，并且这些数据通信链接在三个轴x、y和z上并且在+x(181)、-x(182)、+y(183)、-y(184)、+z(185)和-z(186)六个方向上来回。由对于点对点操作优化的此数据通信网络将链接和计算节点组织为三维网(105)。该网(105)具有在每个轴上的连接该网(105)的相对侧的网(105)中的最外侧计算节点的环绕链接。这些环绕链接形成环状结构(107)。环状结构中的每个计算节点具有在该环状结构中的由一组x、y和z坐标唯一地指定的位置。读者将注意到，为了清楚已经省略了在y和z方向上的环绕链接，但是这些链接以与x方向上的例示的环绕链接类似的方式而配置。为了清楚说明，图4的数据通信网络例示为仅具有27个计算节点，但是读者将认识到，用于在根据本发明的实施例的为其开发集合操作的系统中使用的对于点对点操作优化的数据通信网络可以仅包含几个计算节点或者可以包含几千个计算节点。为了易于说明，图4的数据通信网络被例示为仅具有三个维度，但

是读者将认识到，用于在根据本发明的实施例的为其开发集合操作的系统中使用的对于点对点操作优化的数据通信网络实际上可以实现为二维、四维、五维等等。现在一些超级计算机使用五维网状或者环状网络，包括例如IBM的Blue Gene Q™。

[0056] 为了进一步说明，图5给出例示在根据本发明的实施例的为其开发集合操作的系统中有用的示例的全局组合网络(106)的线条图。图5的示例的数据通信网络包括连接到计算节点以便将计算节点组织为树的数据通信链接(103)。在图5的例子中，点表示并行计算机的计算节点(102)，并且点之间的虚线(103)表示计算节点之间的数据通信链接。利用类似于例如图3B中例示的全局组合网络适配器的全局组合网络适配器来实现数据通信链接，除了一些例外，每个节点通常提供到两个孩子节点和来自两个孩子节点的数据通信以及到双亲节点以及来自该双亲节点的数据通信。全局组合网络(106)中的节点可以被刻画为物理根节点(202)、分支节点(204)以及叶子节点(206)。物理根(202)具有两个孩子但是没有双亲，并且之所以这样称呼是因为物理根节点(202)是物理地配置在二叉树的顶部的节点。叶子节点(206)每个具有双亲，但是叶子节点不具有孩子。分支节点(204)每个具有一个双亲和两个孩子。链接和计算节点由此由对于集合操作优化的此数据通信网络组织为二叉树(106)。为了清楚的说明，图5的数据通信网络被例示为仅具有31个计算节点，但是读者将认识到用在根据本发明的实施例的为其开发集合操作的系统中的对于集合操作优化的全局组合网络(106)可以仅包含几个计算节点或者可以包含数千个计算节点。

[0057] 在图5的例子中，树中的每个节点被分配了称为“秩”的单元标识符(250)。秩实际上标识正执行根据本发明的实施例的并行操作的任务或处理。使用秩来标识节点假定仅一个这样的任务正在每个节点上执行。至于多于一个参与任务在单个节点上执行的情况，秩标识这样的任务而不是标识节点。秩唯一地标识任务在树网络中的位置以供在树网络的点对点和集合操作两者中使用。此例子中的秩被分配为整数，以0分配给根任务或根节点(202)、1分配给树的第二层中的第一节点、2分配给树的第二层中的第二节点、3分配给树的第三层中的第一节点、4分配给树的第三层中的第二节点等等而开始。为了易于例示，在此仅示出了树的前三个层的秩，但是树网络中的所有计算节点都被分配了唯一的秩。

[0058] 为了进一步说明，图6给出例示根据本发明的实施例开发用于并行计算机的集合操作的示例方法的流程图。类似于在图1和2的例子中给出的，根据本发明的实施例为其开发集合操作的并行计算机可以包括多个计算节点。

[0059] 图6的方法包括由集合开发工具向集合开发者呈现(602)图形用户界面(“GUI”)。呈现给集合开发者的GUI可以包括许多不同的GUI对象，比如：表示集合基元的图形形状；用于指定集合基元之间的连续顺序的图形连接符；用于指定输入和输出缓冲器的大小、类型和变量名的下拉列表、文本输入字段等，以及具有本领域知识的读者将想到的其他工具。可以通过在诸如监视器的显示屏幕上显示GUI来执行呈现这样的GUI。

[0060] 图6的方法还包括由集合开发工具通过GUI从集合开发者接收(604)对于一个或多个集合基元的选择。接收(604)对于一个或多个集合基元的选择可以按各种方式执行，包括例如通过检测指示对于与集合基元有关的一个或多个图形图标的选择的用户输入设备活动、通过接收拖动并放下表示集合基元的GUI对象、调用下拉选择列表中的基元的选择的来自用户输入设备(如键盘、鼠标和麦克风等)的输入、以及具有本领域知识的读者将想到的其他方式。

[0061] 从集合开发者的角度来看，集合基元包括收入、操作和输出。从集合开发工具的角度来看，集合基元是计算机程序指令的模块，在某些实施例中是源代码的模块。这样的集合基元的例子包括：

[0062] • 多同步基元，当执行时，执行多个计算节点之间的同步；

[0063] • 多播基元，当执行时，并行地向一组节点发送消息；

[0064] • 多组合基元，当执行时，对从多于一个计算节点接收的数据进行操作；以及

[0065] • 多对多基元，当执行时，向一组计算节点发送唯一的数据并从另一组计算节点接收数据。

[0066] 图6的方法还包括由集合开发工具通过GUI从集合开发者接收(606)集合基元的连续顺序(serial order)的详细说明以及对于每个集合基元的输入和输出缓冲器的详细说明。接收(606)集合基元的连续顺序的详细说明可以按各种方式执行，包括例如通过接收来自用户输入设备的输入以及按具有本领域知识的读者将想到的其他方式，该输入拖动并放下表示集合基元的两个GUI对象之间的GUI连接符(connector)、在文本字段输入指定该顺序的文本、从下拉选择列表选择顺序。

[0067] 图6的方法还包括接收(606)对于每个集合基元的输入和输出缓冲器的详细说明，接收(606)对于每个集合基元的输入和输出缓冲器的详细说明可以通过接收定义了大小、元素的数量(计算节点的数量)、表示每个缓冲器的变量名或者指针等等的来自用户输入设备的输入来执行。

[0068] 图6的方法还包括由集合开发工具依赖于对集合基元的选择、集合基元的连续顺序以及对于每个集合基元的输入和输出缓冲器产生(608)执行由集合基元指定的集合操作的可执行代码。产生(608)可执行代码可以通过在可执行文件中插入用于每个集合基元的计算机程序指令的模块(“集合基元模块”)并且根据预定规则在集合基元模块之间插入“粘结(glue)模块”来执行。在本说明书中作为术语使用的“粘结模块”指配置为为了在集合操作的执行期间链接集合基元模块的目的而插入在集合基元模块之间的计算机程序指令的模块。指定要插入在集合基元模块之间的粘结模块的预定规则可以基于要在其上执行集合操作的并行计算机的许多因素。例如不同的粘结模块可以与不同的集合基元、不同的网络拓扑、计算节点架构、操作组中的计算节点的数量、用于计算节点的通信适配器的存储器资源等等相关联。在一些实施例中，在集合操作开发之前，用这样的并行计算机特性配置集合开发工具。在其他实施例中，集合开发者可以在开发集合操作时向集合开发工具提供这样的并行计算机特性。

[0069] 该集合开发工具可以自动地、无需集合开发者的交互而在集合基元模块之间插入这样的粘结模块。以此方式，可以产生集合操作而不用集合开发者编写任何源代码。因此该集合开发工具为集合开发者提供了用于容易地构造集合操作的工具，同时还为集合开发者提供了通过其详细地指定(specify)集合操作的部分的手段。一旦集合开发工具产生了可执行代码，该代码就可以部署在并行计算机上并执行。

[0070] 图7给出由支持根据本发明的实施例的开发并行计算机的集合操作的集合开发工具呈现的示例的图形用户界面的线条图。图7的示例的GUI(702)包括两栏(pane)：集合基元栏和集合操作栏。集合基元栏包括可用于由集合开发者(用户)选择的每个集合基元的图形图标。在图7的例子中，集合基元栏包括：

[0071] • 表示多同步基元(704)的图形图标，该多同步基元在执行时执行多个计算节点之间的同步；

[0072] • 表示多播基元(705)的图形图标，该多播基元在执行时并行地向一组节点发送消息；

[0073] • 表示多组合基元(706)的图形图标，该多组合基元在执行时对从多于一个计算节点接收的数据进行操作；以及

[0074] • 表示多对多基元(718)的图形图标，该多对多基元在执行时向一组计算节点发送唯一的数据并从另一组计算节点接收数据。

[0075] 这些图形图标的每个可以按任意组合以及按任意数量被选择、拖动或者放下到集合操作栏中。在此例子中，集合开发者已经通过将对应于多同步基元(704)、多组合基元(706)和多对多基元(718)的每个的图标拖动到集合操作栏而选择了这些基元用于集合操作。集合开发者还指定了要在执行多组合基元(708)时实施的“或”操作。

[0076] 一旦集合开发者选择了将形成集合操作的集合基元，则集合开发者就可以指定集合基元的连续顺序以及对于每个所选集合基元的输入和输出缓冲器。在图7的例子中，集合开发者已经指定了输入缓冲器(706、710、712、716)和输出缓冲器(706、712、714)。一些缓冲器可以操作为一个集合操作的输出缓冲器以及另一个集合操作的输入缓冲器。集合开发者还可以通过配置为接收诸如大小、类型、表示缓冲器的变量或指针等等的数据项的文本字段或者其他GUI对象来指定缓冲器的其他特性。

[0077] 集合开发者还拉动、拖动或者另外提供用户输入以形成集合基元和缓冲器之间的箭头(720)以便指定集合基元之间的连续顺序。在此例子中，集合基元的连续顺序以多同步基元(704)开始，以多组合基元(708)继续，并且以多对多基元(718)结束。

[0078] 在集合开发者选择了集合基元、指定了集合基元的顺序并且指定了集合基元的输入和输出缓冲器之后，集合开发者可以开始将执行集合操作的可执行代码的产生。用户可以通过图7的示例GUI(702)按各种方式开始这样的产生，包括调用为这样的产生指派的GUI按钮、从下拉列表或者菜单列表选择动作、输入预定义的一组键盘键击以及具有本领域知识的读者将想到的其他方式等等。

[0079] 以此方式，集合开发者可以有效并迅速地建立和产生用于集合操作的可执行代码，而实际上编写非常少的可执行代码——如果有的话。除了通过本发明的集合开发工具提供给集合开发者的集合开发的容易性和高效之外，还为开发者提供了通过其详细指定集合操作的特性的有力的手段。

[0080] 如本领域技术人员将认识到的，本发明的方面可以实现为系统、方法或计算机程序产品。因此，本发明的方面可以采取完全硬件实施例、完全软件实施例(包括固件、驻留软件、微代码等)或者组合了软件和硬件方面的实施例的形式，它们可以在此统称为“电路”、“模块”或“系统”。此外，本发明的方面可以采取体现在一个或多个计算机可读介质中的计算机程序产品的形式，该计算机可读介质具有体现在其上的计算机可读的程序代码。

[0081] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读传输介质或者计算机可读存储介质。计算机可读存储介质例如可以是但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件、或者以上的任意适当的组合。计算机可读存储介质的更具体的例子(非穷举的列表)将包括以下：具有一个或多个导线的电连接、

便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文的上下文中,计算机可读存储介质可以是任何可以包含或存储程序的有形介质,该程序由指令执行系统、装置或者器件使用或者与其结合使用。

[0082] 计算机可读传输介质可以包括在基带中或者作为载波的部分的传播的数据信号,其中承载了计算机可读的程序代码。这种传播的信号可以采用多种形式的任意形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读传输介质可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0083] 计算机可读介质上包含的程序代码可以使用任何适当的介质来传输,包括但不限于无线、有线、光纤电缆、RF等等,或者上述的任意合适的组合。

[0084] 可以以一种或多种编程语言的任意组合来编写用于执行本发明的方面的操作的计算机程序代码,所述编程语言包括诸如Java、Smalltalk、C++等的面向对象的程序设计语言以及诸如“C”编程语言或类似编程语言的传统过程编程语言。程序代码可以完全地在用户的计算机上执行、部分地在用户计算机上执行、作为独立的软件包执行、部分在用户的计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在后一情形中,远程计算机可以通过任意类型的网络、包括局域网(LAN)或广域网(WAN)连接到用户的计算机,或者,可以使得(例如使用因特网服务提供商通过因特网)连接到外部计算机。

[0085] 参照根据本发明的实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本发明的方面。将理解,流程图和/或框图中的每个方框以及流程图和/或框图中各方框的组合可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出机器,以便经由计算机或其它可编程数据处理装置的处理器执行的这些指令创建用于实现流程图和/或框图的方框中指定的功能/动作的部件。

[0086] 也可以把这些计算机程序指令存储在可以使得计算机、其它可编程数据处理装置或其它设备以特定方式工作的计算机可读介质中,以便存储在计算机可读介质中的指令产生包括实现流程图和/或框图的方框中指定的功能/动作的制造品(article of manufacture)。

[0087] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,以致使在计算机、其它可编程装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,以便在计算机或其它可编程装置上执行的指令提供用于实现在流程图和/或框图的方框中指定的功能/动作的过程。

[0088] 附图中的流程图和框图例示根据本发明的各个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表模块、程序段或代码的部分,所述模块、程序段或代码的部分包含一个或多个用于实现指定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能可以以不同于附图中所标注的顺序发生。例如,示出为连续的两个方框实际上可以基本并行地执行,或者这些块有时可以按相反的顺序执行,这依赖于所涉及的功能而定。也要注意的是,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合可以由进行指

定的功能或动作的专用的基于硬件的系统来实现，或者可以由专用硬件与计算机指令的组合来实现。

[0089] 根据以上描述将理解，在不偏离本发明的真正精神的情况下，可以在本发明的各个实施例中做出修改和改变。本说明书中的描述仅仅是用于例示的目的并且不应被解释为限制的意思。本发明的范围仅由以下权利要求的语言限制。

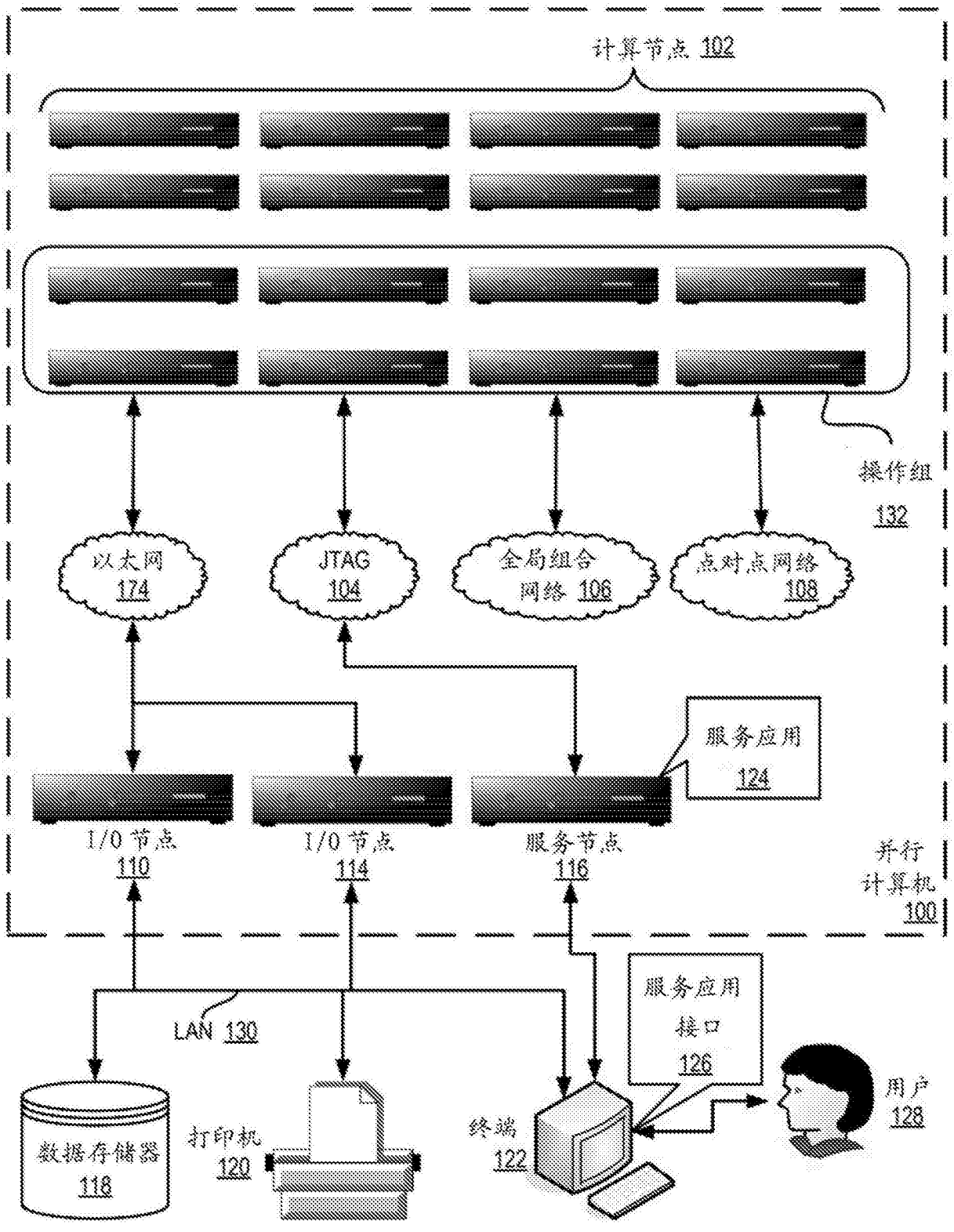

计算节点 102
操作组
132
以太网
174
JTAG
104
全局组合
网络 106
点对点网络
108
服务应用
124
I/O 节点
110
I/O 节点
114
服务节点
116
并行
计算机
100
LAN 130
服务应用
接口
126
用户
128
数据存储器
118
打印机
120
终端
122

图1

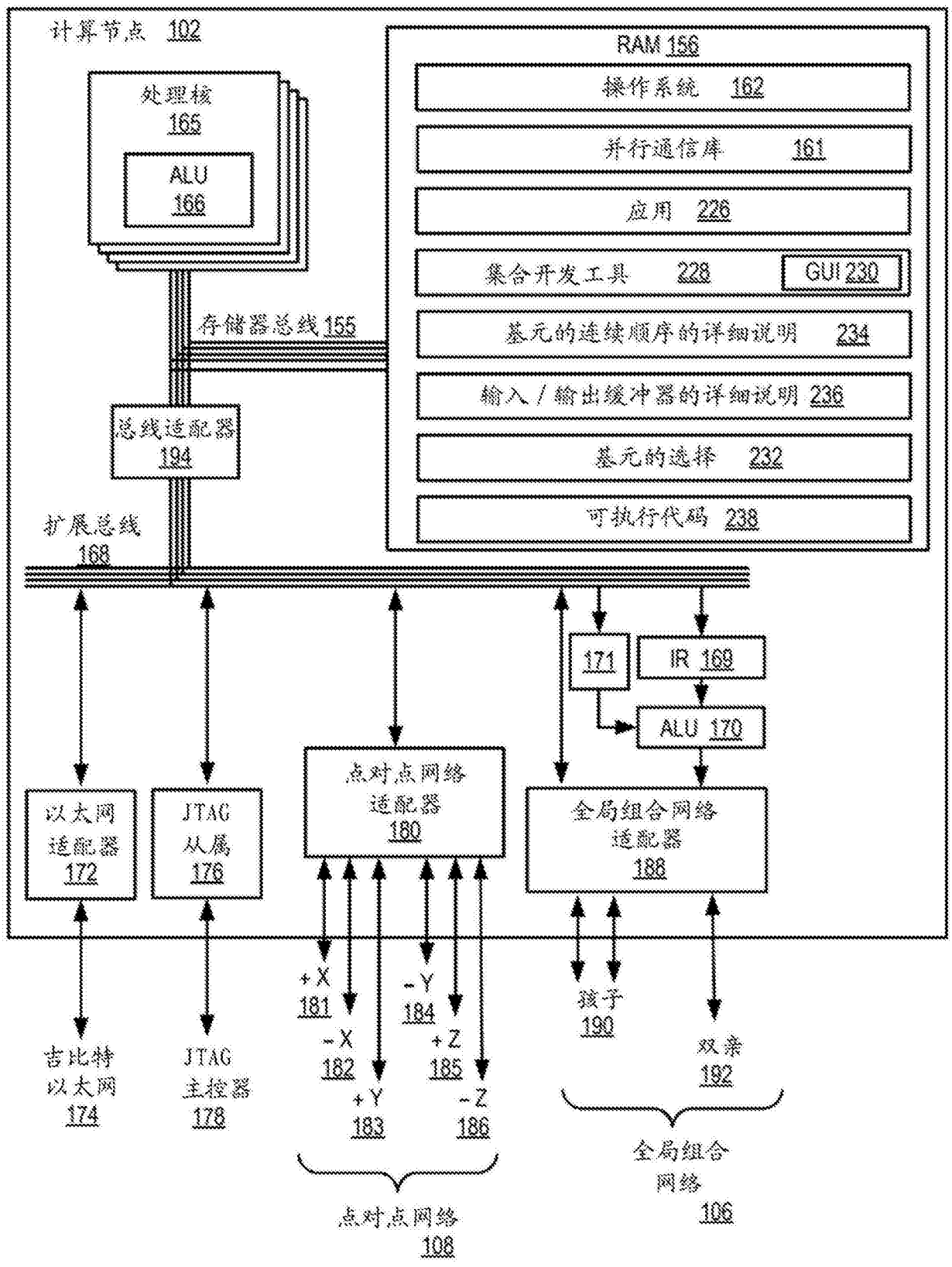

计算节点 102
处理核 165
ALU 166
RAM 156
操作系统 162
并行通信库 161
应用 226
集合开发工具 228
GUI 230
基元的连续顺序的详细说明 234
输入/输出缓冲器的详细说明 236
基元的选择 232
可执行代码 238
存储器总线 155
总线适配器 194
扩展总线 168
171
IR 169
ALU 170
以太网适配器 172
JTAG 从属 176
点对点网络适配器 180
全局组合网络适配器 188
+X 181
-X 182
+Y 183
-Y 184
+Z 185
-Z 186
孩子 190
双亲 192
吉比特以太网 174
JTAG 主控器 178
点对点网络 108
全局组合网络 106

图2

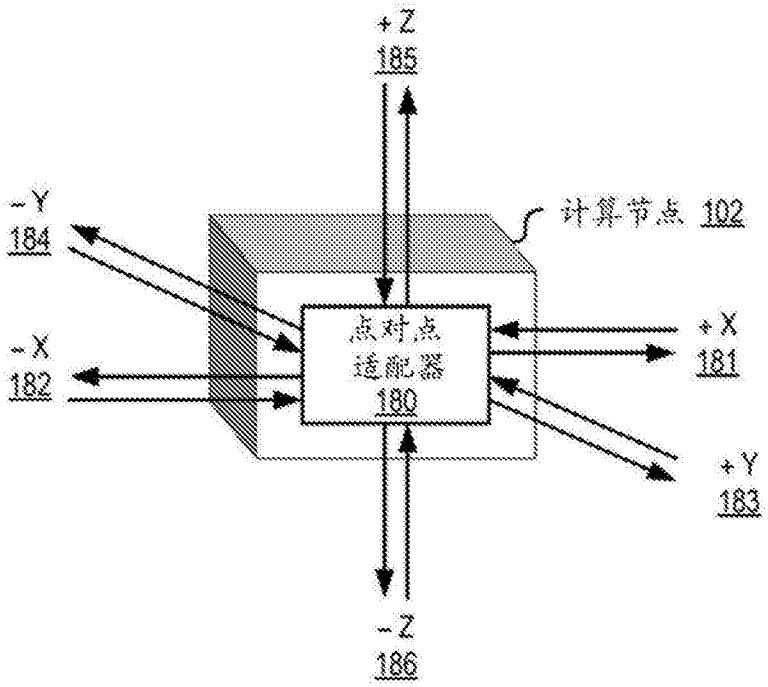

+Z
185
-Y
184
计算节点 102
点对点
适配器
180
+X
181
-X
182
+Y
183
-Z
186

图3A

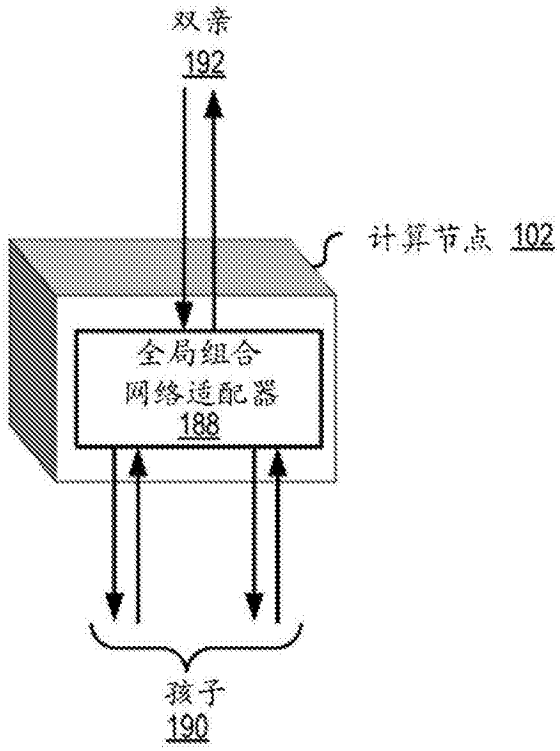

双亲
192
计算节点 102
全局组合
网络适配器
188
孩子
190

图3B

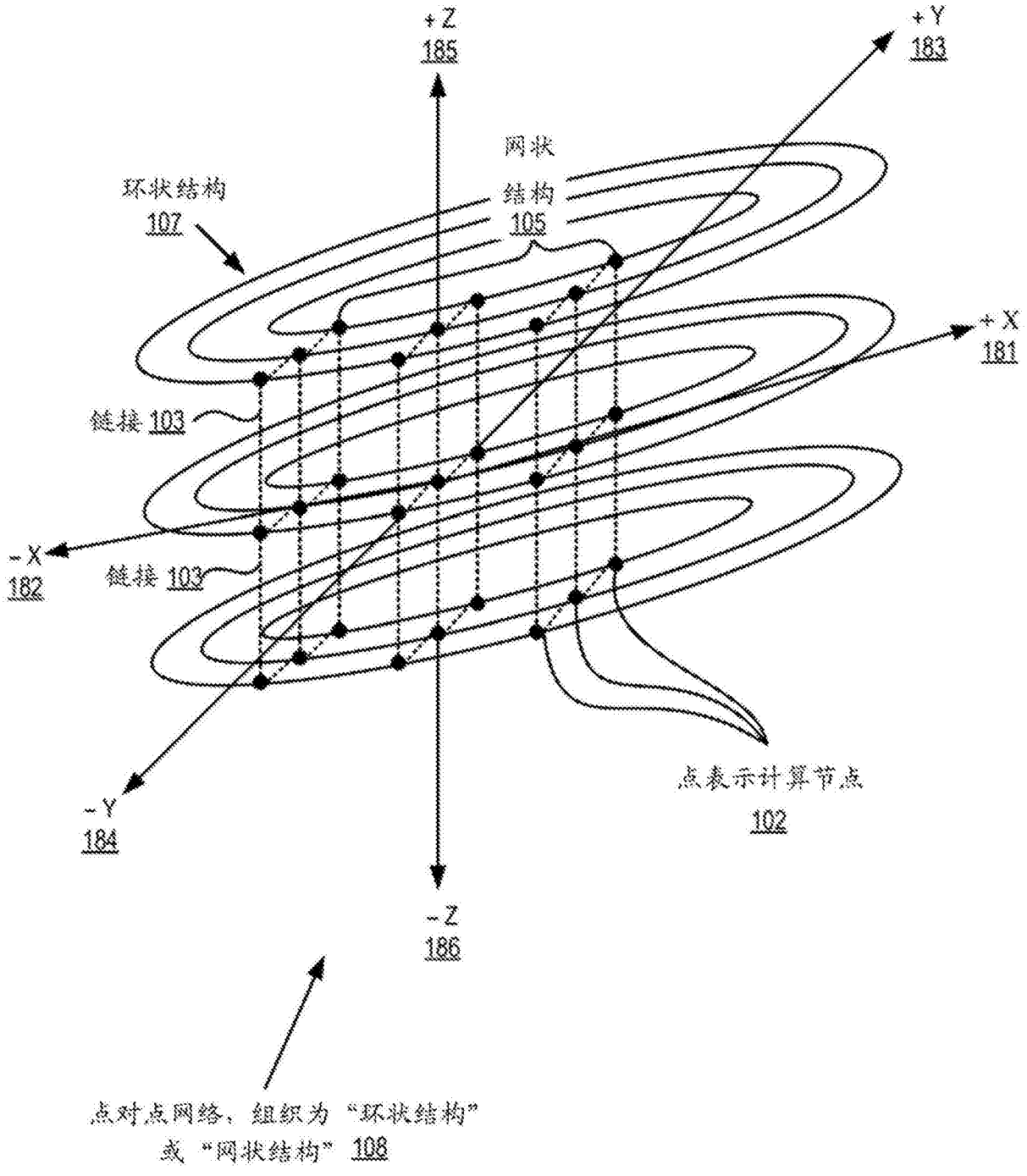

+Z
185
+Y
183
网状
结构
105
环状结构
107
+X
181
链接 103
-X
182
链接 103
-Y
184
点表示计算节点
102
-Z
186
点对点网络、组织为“环状结构”
或“网状结构” 108

图4

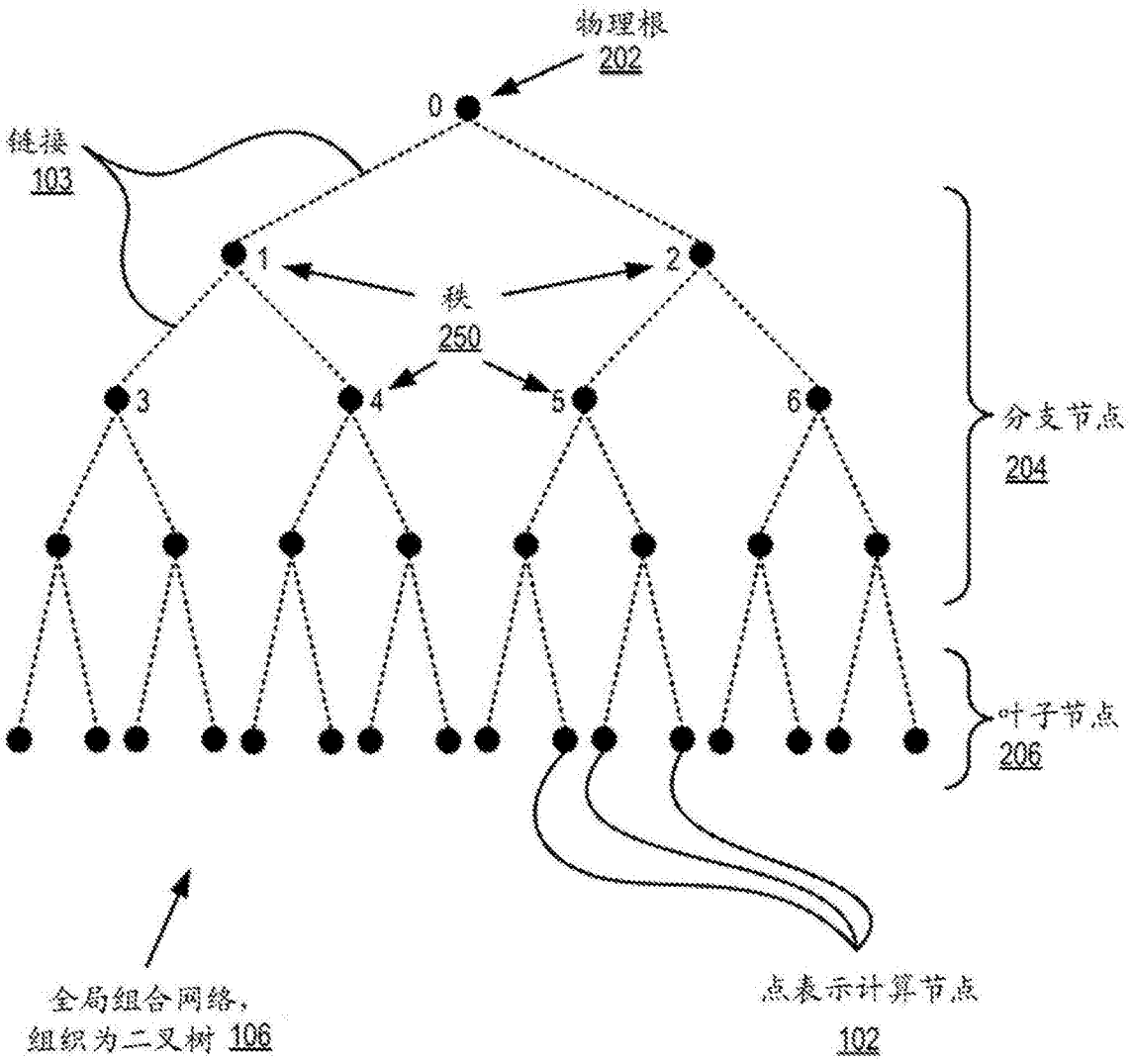

物理根
202
0
链接
103
1
2
秩
250
3
4
5
6
分支节点
204
叶子节点
206
全局组合网络，
组织为二叉树 106
点表示计算节点
102

图5

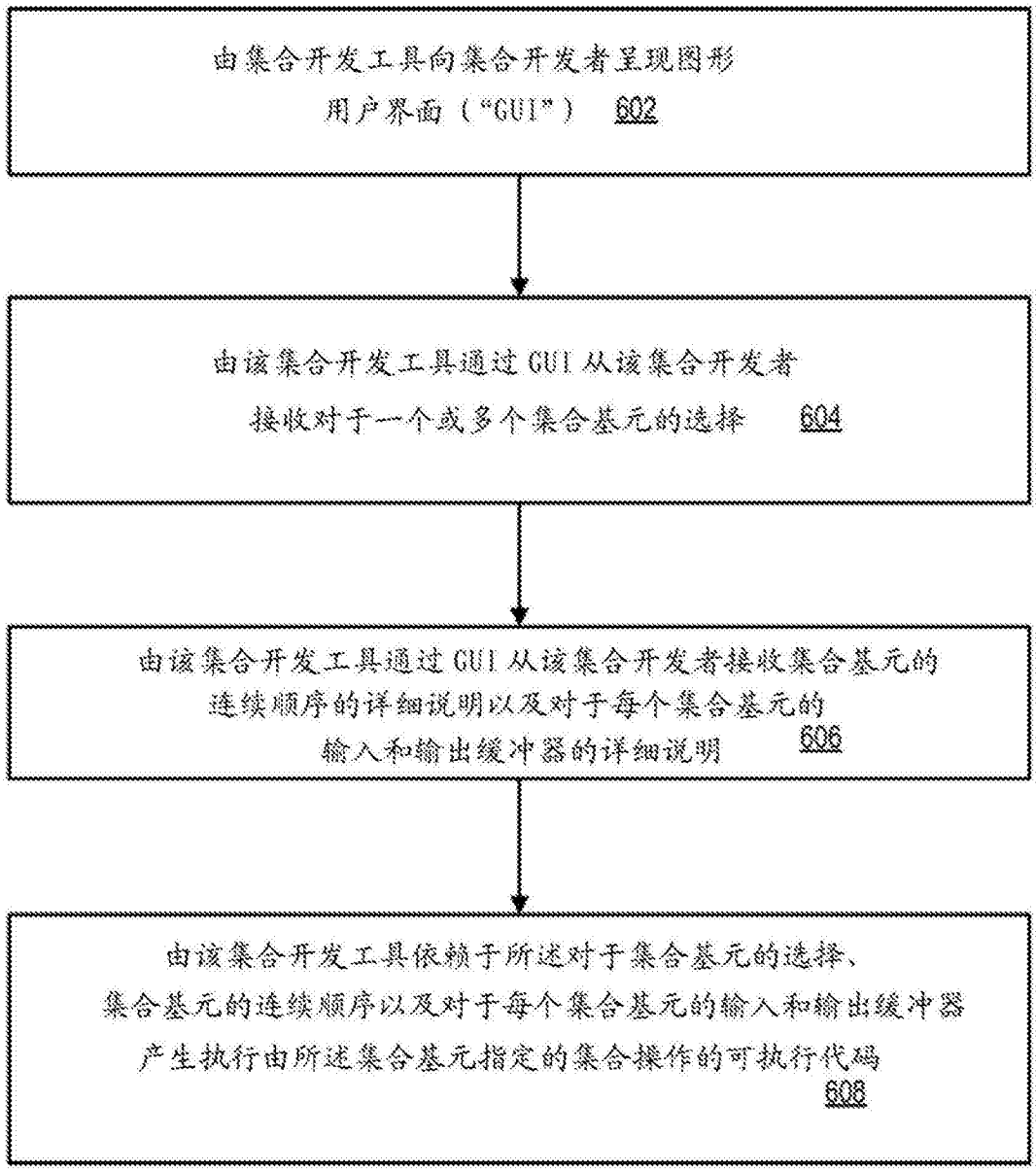

由集合开发工具向集合开发者呈现图形用户界面（“GUI”） 602
由该集合开发工具通过 GUI 从该集合开发者接收对于一个或多个集合基元的选择 604
由该集合开发工具通过 GUI 从该集合开发者接收集合基元的连续顺序的详细说明以及对于每个集合基元的输入和输出缓冲器的详细说明 606
由该集合开发工具依赖于所述对于集合基元的选择、集合基元的连续顺序以及对于每个集合基元的输入和输出缓冲器产生执行由所述集合基元指定的集合操作的可执行代码 608

图6

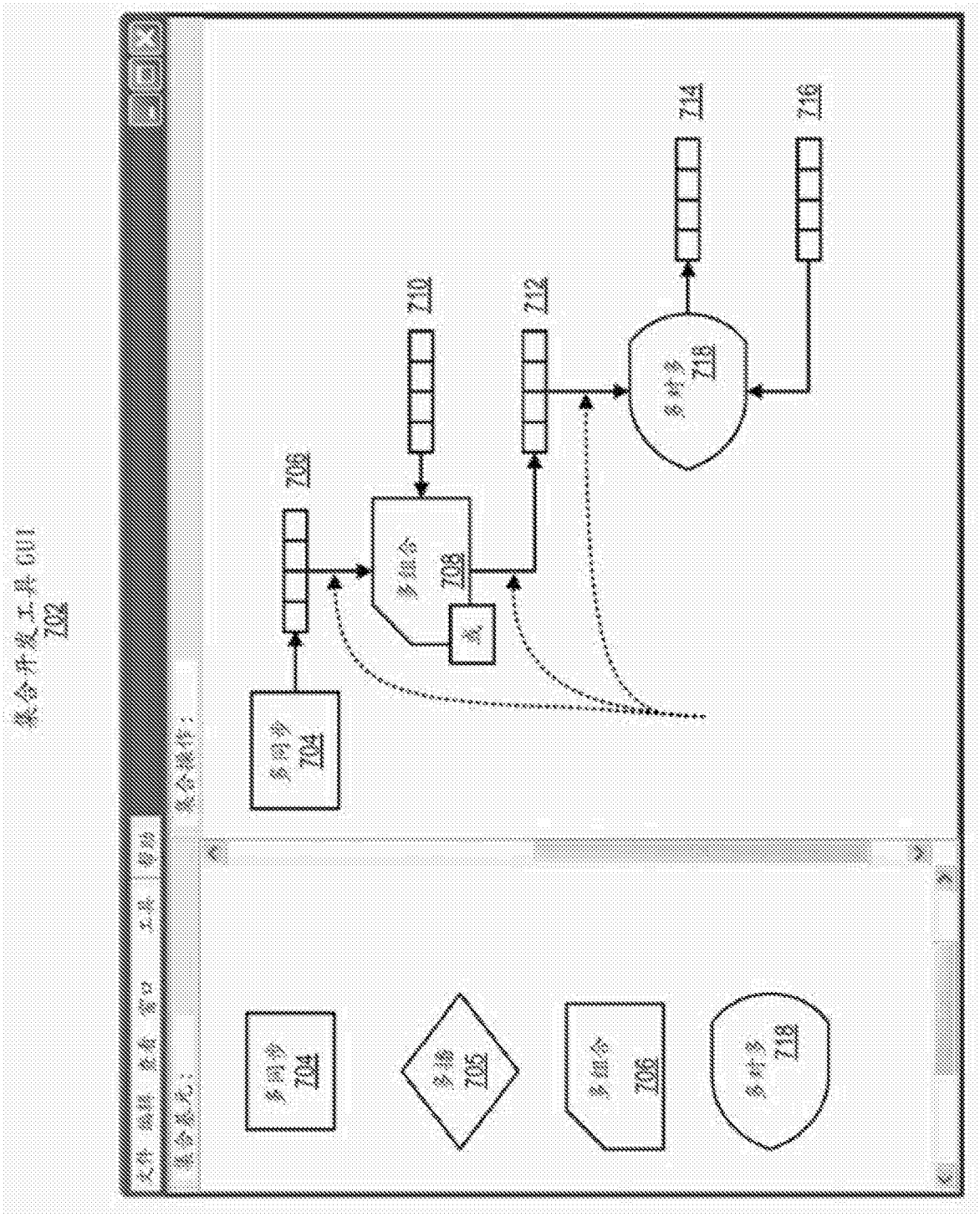

集合开发工具 GUI
202
文件 编辑 查看 窗口 工具 帮助
集合表达式:
集合操作:
多对多
204
多端
205
多组合
206
多对多
218
多对多
204
多组合
208
集
208
210
212
多对多
218
214
216

图7