(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 102859490 B
(45)授权公告日 2015.12.02

(21)申请号 201180020780.0
(22)申请日 2011.02.14
(30)优先权数据
12/770,353 2010.04.29 US
(85)PCT国际申请进入国家阶段日
2012.10.25
(86)PCT国际申请的申请数据
PCT/EP2011/052105 2011.02.14
(87)PCT国际申请的公布数据
WO2011/134689 EN 2011.11.03
(73)专利权人 国际商业机器公司
地址 美国纽约
(72)发明人 J·拉特尔曼 C·J·阿切尔 B·E·史密斯 M·A·布洛克萨默
(74)专利代理机构 北京市中咨律师事务所 11247
代理人 张亚非 于静
(51)Int.Cl.
*G06F 9/45*(2006.01)
(56)对比文件
US 2004083462 A1,2004.04.29,
US 2004083462 A1,2004.04.29,
US 6233725 B1,2001.05.15,
US 2006005173 A1,2006.01.05,
审查员 荆苏丹

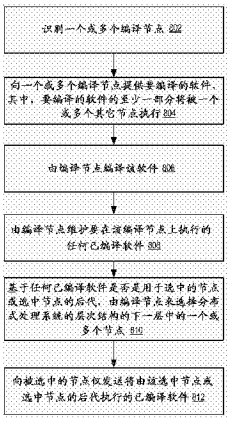
权利要求书3页 说明书20页 附图9页

(54)发明名称

为层次化分布式处理系统编译软件的方法和装置

(57)摘要

为层次化分布式处理系统编译软件包括：向一个或多个编译节点提供要被编译的软件，其中，要被编译的软件的至少一部分将被一个或多个其它节点执行；由编译节点编译软件；由编译节点来维护将在该编译节点上执行的任何已编译软件；基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点选择分布式处理系统的层次结构的下一层中的一个或多个节点；向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件。




CN 102859490 B

1. 一种为层次化分布式处理系统编译软件的方法，该方法包括：

向一个或多个编译节点提供要被编译的软件，其中，要被编译的软件的至少一部分将被一个或多个其它节点执行；

由编译节点编译软件；

由编译节点维护将在该编译节点上执行的任何已编译软件；

基于任何已编译软件是否是用于选中的节点或选中节点的后代、由编译节点选择分布式处理系统的层次结构的下一层中的一个或多个节点；

由编译节点向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件；

由选中的节点接收已编译软件；

确定该已编译软件是否是用于选中的节点或其后代；

如果该已编译软件是用于选中的节点，由该选中节点维护该软件以用于执行；以及

如果该已编译软件被用于后代中的一个，基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点。

2. 如权利要求 1 所述的方法，还包括识别一个或多个编译节点。

3. 如权利要求 2 所述的方法，其中，识别一个或多个编译节点还包括选择针对编译进行了计算优化的一个或多个节点。

4. 如权利要求 2 或 3 所述的方法，其中，识别一个或多个编译节点还包括根据节点在层次化分布式处理系统的拓扑结构中的位置，选择针对编译进行了优化的一个或多个节点。

5. 如权利要求 1 至 3 中的任一个所述的方法，其中，所述层次化分布式处理系统还包括并行计算机，该并行计算机包含：

多个计算节点；

与计算节点耦合的第一数据通信网络，其用于数据通信并针对点对点数据通信进行了优化；

第二数据通信网络，其包含与计算节点耦合的数据通信链路，以将计算节点组织为树，每个计算节点具有专用于并行操作的单独的算术逻辑单元。

6. 如权利要求 4 所述的方法，其中，所述层次化分布式处理系统还包括并行计算机，该并行计算机包含：

多个计算节点；

与计算节点耦合的第一数据通信网络，其用于数据通信并针对点对点数据通信进行了优化；

第二数据通信网络，其包含与计算节点耦合的数据通信链路，以将计算节点组织为树，每个计算节点具有专用于并行操作的单独的算术逻辑单元。

7. 如权利要求 1 至 3 中的任一个所述的方法，其中，所述层次化分布式处理系统还包括混合计算环境，该混合计算环境包含多个计算节点，每个计算节点包含：

具有主计算机架构的主计算机；以及

具有加速器架构的加速器，该加速器架构相对于主计算机架构、针对特定类别的计算功能的执行速度进行了优化，该主计算机和加速器互相适配，以通过系统级消息传递模块来进行数据通信。

8. 如权利要求 4 所述的方法，其中，所述层次化分布式处理系统还包括混合计算环境，该混合计算环境包含多个计算节点，每个计算节点包含 ：

具有主计算机架构的主计算机 ；以及

具有加速器架构的加速器，该加速器架构相对于主计算机架构、针对特定类别的计算功能的执行速度进行了优化，该主计算机和加速器互相适配，以通过系统级消息传递模块来进行数据通信。

9. 一种为层次化分布式处理系统编译软件的装置，包括 ：

用于向一个或多个编译节点提供要被编译的软件的模块，其中，要被编译的软件的至少一部分将被一个或多个其它节点执行 ；

用于由编译节点编译软件的模块 ；

用于由编译节点维护将在该编译节点上执行的任何已编译软件的模块 ；

用于基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点选择分布式处理系统的层次结构的下一层中的一个或多个节点的模块 ；以及

用于由编译节点向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件的模块 ；

用于由选中的节点来接收编译软件的模块 ；

用于确定该已编译软件是否被用于选中的节点或其后代的模块 ；

用于如果该已编译软件被用于选中的节点，由该选中节点来维护软件用于执行的模块 ；以及

用于如果该已编译软件被用于后代中的一个，基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点的模块。

10. 如权利要求 9 所述的装置，还包括用于识别一个或多个编译节点的模块。

11. 如权利要求 10 所述的装置，其中，所述用于识别一个或多个编译节点的模块还包括用于选择针对编译进行了计算优化的一个或多个节点的模块。

12. 如权利要求 10 或 11 所述的装置，其中，所述用于识别一个或多个编译节点的模块还包括 ：用于根据节点在层次化分布式处理系统的拓扑结构中的位置，来选择针对编译进行了优化的一个或多个节点的模块。

13. 如权利要求 9 至 11 中的任一个所述的装置，其中，所述层次化分布式处理系统还包括并行计算机，该并行计算机包含 ：

多个计算节点 ；

与计算节点耦合的第一数据通信网络，其用于数据通信并针对点对点数据通信进行了优化 ；

第二数据通信网络，其包含与计算节点耦合的数据通信链路，从而将计算节点组织为树，每个计算节点具有专用于并行操作的单独的算术逻辑单元。

14. 如权利要求 12 所述的装置，其中，所述层次化分布式处理系统还包括并行计算机，该并行计算机包含 ：

多个计算节点 ；

与计算节点耦合的第一数据通信网络，其用于数据通信并针对点对点数据通信进行了优化 ；

第二数据通信网络，其包含与计算节点耦合的数据通信链路，从而将计算节点组织为树，每个计算节点具有专用于并行操作的单独的算术逻辑单元。

15. 如权利要求 9 至 11 中的任一个所述的装置，其中，所述层次化分布式处理系统还包括混合计算环境，该混合计算环境包含多个计算节点，每个计算节点包含：

具有主计算机架构的主计算机；以及

具有加速器架构的加速器，该加速器架构相对于主计算机架构、针对特定类型的计算功能的执行速度进行了优化，该主计算机和加速器互相适配，以通过系统级消息传递模块来进行数据通信。

16. 如权利要求 12 所述的装置，其中，所述层次化分布式处理系统还包括混合计算环境，该混合计算环境包含多个计算节点，每个计算节点包含：

具有主计算机架构的主计算机；以及

具有加速器架构的加速器，该加速器架构相对于主计算机架构、针对特定类型的计算功能的执行速度进行了优化，该主计算机和加速器互相适配，以通过系统级消息传递模块来进行数据通信。

# 为层次化分布式处理系统编译软件的方法和装置

## 技术领域

[0001] 本发明的领域是数据处理，或更具体而言，用于为层次化分布式处理系统编译软件的方法、装置和产品。

## 背景技术

[0002] 1948 的 EDVAC（电子离散变量自动计算机）计算机系统的开发通常被称为是计算机时代的开始。从那时开始，计算机系统已经演变为极端复杂的设备。今天的计算机与早期的系统诸如 EDVAC 相比要复杂地多。计算机系统典型地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 / 输出设备等的组合。随着半导体处理和计算机架构的进步使计算机的性能越来越高，更为复杂的计算机软件已发展出来，以利用硬件的更高性能，使得今天的计算机系统比仅仅几年前要强大地多。

[0003] 分布式计算是计算机技术取得进展的一个领域。分布式计算通常是指用多个半自主计算机系统来进行计算，所述计算机系统通过数据通信网络来通信。半自主计算机系统互相交互，以实现共同的目标。在分布式计算系统中执行的计算机程序或应用可被称为分布式程序。分布式计算还可指使用分布式计算系统来解决计算问题。在分布式计算中，一个问题可被分为多个任务，每个任务可以被其中一个半自主计算机系统来解决。

[0004] 某些分布式计算系统被优化来执行并行计算。并行计算是在多个处理器上同时执行同一任务（其被划分并特别适应），以更快地获取结果。并行计算基于下列事实 :解决问题的过程通常可被分为更小的任务，这些任务可以通过某种协调来同时执行。

[0005] 并行计算机执行并行算法。并行算法可以被划分从而在很多个不同处理设备上每次执行一片，然后在结束时再次合并到一起以获得数据处理结果。某些算法能容易地分片。将检查从一到十万的数字以找到素数的任务进行划分，例如，可以通过将这些数字的子集分配给每个可用的处理器、然后将肯定结果的列表合并到一起来实现。在本说明书中，执行并行程序的单片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它处理节点包括例如输入 / 输出（“I/O”）节点和服务节点组成。

[0006] 并行计算很有价值，因为由于现代处理器工作的方式，通过并行算法来执行某些类型的大型计算任务比通过串行（非并行）算法更快。构建具有单个快速处理器的计算机比构建吞吐量相同的具有多个慢速处理器的计算机要困难地多。串行处理器的潜在速度也存在一定的理论限制。另一方面，每个并行算法具有串行部分，因而并行算法具有饱和点。在该点之后，增加更多的处理器不会产生更多吞吐量，而只会增加开销和成本。

[0007] 并行计算被设计为还优化并行计算机的节点之间的数据通信需要的一个或多个资源。并行处理器有两种方法来通信，共享存储器或消息传递。共享存储器处理需要对数据的额外锁定并带来了额外的处理器和总线周期的开销，且同时将算法的某些部分串行化。

[0008] 消息传递处理使用高速数据通信网络和消息缓冲区，但该通信增加了数据通信网络上的传输开销和用于消息缓冲区的额外存储器，以及节点之间的数据通信的延迟。并行计算机的设计使用特殊设计的数据通信链路，从而通信开销会较小，但是决定流量大小的

是并行算法。

[0009] 很多数据通信网络架构被用于并行计算机的节点之间的消息传递。例如，计算机节点在网络中可以被组织为环形 (torus) 或网状 (mesh)。而且，计算节点在网络中可以被组织为树。环形网络以回绕 (wrap around) 链路在三维网状中连接节点。每个节点通过该环状网络连接到其六个邻居，且每个节点通过其在网状中的 x、y 和 z 坐标来寻址。在这种方式国，环形网络有助于点到点的操作。在树形网络中，节点典型地连接为二叉树 ：每个节点具有父节点和两个子节点 ( 尽管某些节点可只有零个子节点或一个子节点，这取决于硬件配置 )。尽管树形网络典型地在点对点通信中是低效的，但对于某些共同操作，计算节点同时参与例如 allgather 操作的消息传递操作，树形网络确实提供了高带宽和低延迟。在使用环形和树形网络的计算机中，这两种网络典型地互相独立地实现，具有单独的路由电路、单独的物理链路和单独的消息缓冲区。

## 发明内容

[0010] 为层次化分布式处理系统编译软件的方法、装置和产品，包括 ：向一个或多个编译节点提供要被编译的软件，其中，要被编译的软件的至少一部分将被一个或多个其它节点执行 ；由编译节点编译该软件 ；由编译节点维护将在该编译节点上执行的任何已编译软件 ；基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点来选择分布式处理系统的层次结构的下一层中的一个或多个节点 ；向选中的节点仅发送由该选中的节点或选中节点的后代执行的已编译软件 ；由该选中的节点来接收已编译软件 ；确定该已编译软件是否是用于该选中的节点或其后代之一 ；如果该已编译软件是用于选中的节点，由该选中节点来维护软件以用于执行 ；并且如果该已编译软件是用于后代之一，基于后代为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点。

[0011] 根据如附图所示的本发明的示例性实施例的下列更具体的描述，本发明的上述和其它目标、特征和优势将变得明显，在附图中相同的标号通常表示本发明的示例性实施例的相同部分。

## 附图说明

[0012] 图 1 示出了根据本发明实施例的用于为层次化分布式处理系统编译软件的示例性分布式计算机系统。

[0013] 图 2 给出了根据本发明实施例可在能够为层次化分布式处理系统编译软件的并行计算机中使用的示例性计算节点的框图。

[0014] 图 3A 示出了可在根据本发明实施例的能够为层次化分布式处理系统编译软件的系统中使用的示例性点对点适配器。

[0015] 图 3B 示出了可在根据本发明实施例的能够为层次化分布式处理系统编译软件的系统中使用的示例性全球组合网络 (global combining network) 适配器。

[0016] 图 4 给出了示出示例性数据通信网络的线条图，该数据通信网络针对可在根据本发明实施例的能够为层次化分布式处理系统编译软件的系统中使用的点对点操作进行了优化。

[0017] 图 5 给出了示出示例性数据通信网络的线条图，该数据通信网络针对可在根据本发明实施例的能够为层次化分布式处理系统编译软件的系统中使用的共同操作进行了优化。

[0018] 图 6 给出了根据本发明实施例的为层次化分布式处理系统编译软件的另一示例性分布式计算系统，其中，该分布式计算系统被实现为混合计算环境。

[0019] 图 7 给出了根据本发明实施例的为层次化分布式处理系统编译软件的示例性方法。

[0020] 图 8 给出了示出根据本发明实施例的为层次化分布式处理系统编译软件的另一示例性方法的流程图。

[0021] 图 9 给出了根据本发明实施例的为层次化分布式处理系统编译软件的系统的示例性用例 (use case) 的框图。

## 具体实施方式

[0022] 参考从图 1 开始的附图来描述根据本发明的实施例的为层次化分布式处理系统编译软件的示例性方法、装置和产品。图 1 示出了根据本发明的实施例的为层次化分布式处理系统编译软件的示例性分布式计算系统。图 1 的系统包括并行计算机 (100)、用于计算机的数据存储设备形式的非易失性存储器 (118)、用于计算机的打印机形式的输出设备 (120)、以及用于计算机的计算机终端形式的输入 / 输出设备 (122)。图 1 的例子中的并行计算机 (100) 包括多个计算节点 (102)。

[0023] 计算节点 (102) 通过若干个独立数据通信网络耦合以进行数据通信，所述数据通信网络包括联合测试行动小组 (“JTAG”) 网络 (104)、针对共同操作进行了优化的全球组合网络 (106)、以及针对点对点操作进行了优化的环形网络 (108)。全球组合网络 (106) 是包含连接到计算节点从而将计算节点组织为树的通信链路的数据通信网络。每个数据通信网络用计算节点 (102) 之间的数据通信链路来实现。数据通信链路为并行计算机的计算节点之间的并行操作提供数据通信。计算节点之间的链路是双向链路，其典型地使用两个单独方向的数据通信路径来实现。

[0024] 此外，并行计算机的计算节点 (102) 被组织为计算机节点的至少一个操作组 (132)，以用于并行计算机 (100) 上的共同并行操作。计算节点的操作组是一组计算节点，共同并行操作在其上执行。共同操作通过操作组的计算节点之间的数据通信来实现。共同操作是涉及操作组中的所有计算节点的那些功能。共同操作是一种操作，是由计算节点的操作组中的所有计算节点同时即大约在同一时刻执行的消息传递计算机程序指令。这样的操作组可以包括并行计算机 (100) 中的所有计算节点或者所有计算节点的子集。共同操作经常围绕点对点操作来建立。共同操作需要操作组中的所有计算节点上的所有进程以匹配的参数来调用同一共同操作。“广播”是用于在操作组中的计算节点之间移动数据的共同操作的例子。“归约”(reduce) 操作是在分布于操作组中的计算节点之间的数据上执行算术或逻辑功能的共同操作的例子。操作组可以被实现为例如 MPI “通信器”。

[0025] “MPI”是指“消息传递接口”，是现有的并行通信库，是用于并行计算机上的数据通信的计算机程序指令模块。可以改进以与根据本发明的实施例的系统一起使用的现有的并行通信库的例子包括 MPI 和“并行虚拟机”(“PVM”) 库，其。PVM 是由田纳西大学、橡树岭

国家实验室和埃默里大学开发的。MPI 由 MPI 论坛公布，该论坛是一个开放组织，其代表来自很多机构，并定义和维护 MPI 标准。在写作本文的时刻，MPI 是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事实上的标准。为了便于解释，本说明书有时使用 MPI 术语，尽管这样使用 MPI 并不是本发明的要求或限制。

[0026] 某些共同操作具有在操作组的特定计算节点上运行的单个发起或接收进程。例如，在“广播”共同操作中，向所有其它计算节点发布数据的计算节点上的进程是发起进程。例如，在“收集”(gather) 操作中，从其它节点接收所有数据的计算节点上的进程是接收进程。在其上运行这样的发起或接收进程的计算节点被称为逻辑根。

[0027] 大多数共同操作是四种基本操作的变化或组合 :广播、收集、散布 (scatter) 和归约。用于这些共同操作的接口在 MPI 论坛所发布的 MPI 标准中定义。但是，用于执共同操作的算法没有在 MPI 标准中定义。在广播操作中，所有进程指定同一根进程，其缓冲区内容将被发送。根以外的进程指定了接收缓冲区。在该操作后，所有缓冲区都包含来自根进程的消息。

[0028] 在散布操作中，逻辑根将根上的数据分段，并将不同的段分发给操作组中的每个计算节点。在散布操作中，所有进程典型地指定同一接收计数。发送参数只对根进程有意义，其缓冲区实际上包含给定数据类型的 sendcount*N 个元素，其中 N 是给定计算节点组中的进程数量。发送缓冲区被分割和散布给所有进程 ( 包括逻辑根上的进程 )。给每个计算节点分配称为“等级”(rank) 的顺序标识符。在该操作后，根以递增的等级顺序向每个线程发送了 sendcount 个数据元素。等级 0 从发送缓冲区接收第一 sendcount 个数据元素。等级 2 从发送缓冲区接收第二 sendcount 个数据元素，等等。

[0029] 收集操作是多对一的共同操作，是与散布操作的描述完全相反。即，收集是多对一的操作，其中某个数据类型的元素从分级的计算节点收集到根节点中的接收缓冲区中。

[0030] 归约操作也是多对一的共同操作，其包括在两个数据元素上执行的算术或逻辑功能。所有进程指定了同一“count”( 计数 ) 和同一算术或逻辑功能。在归约之后，所有进程已从计算节点发送缓冲区向根进程发送了 count 个数据元素。在归约操作中，由算术或逻辑操作将来自相应发送缓冲区位置的数据元素进行成对组合，以生成根进程的接收缓冲区中的单个相应元素。应用特定的归约操作可以在运行时定义。并行通信库可以支持预定义的操作。例如，MPI 提供了下列预定义的归约操作 :

[0031] MPI_MAX 最大值

[0032] MPI_MIN 最小值

[0033] MPI_SUM 总和

[0034] MPI_PROD 乘积

[0035] MPI_LAND 逻辑与

[0036] MPI_BAND 位与

[0037] MPI_LOR 逻辑或

[0038] MPI_BOR 位或

[0039] MPI_LXOR 逻辑异或

[0040] MPI_BXOR 位异或

[0041] 除了计算节点，并行计算机 (100) 包括通过全球组合网络 (106) 耦合到计算节点

(102) 的输入 / 输出 (“I/O”) 节点 (110、114)。并行计算机 (100) 中的计算节点被划分为处理集合，从而一处理集合中的每个计算节点被连接，以与同一 I/O 节点进行数据通信。每个处理集合因此由一个 I/O 节点和计算节点 (102) 的子集构成。整个系统中计算节点数量与 I/O 节点数量的比率典型地取决于并行计算机的硬件配置。例如，在某些配置中，每个处理集合可以由八个计算节点和一个 I/O 节点构成。在某些其它配置中，每个处理集合可以由六十四个计算节点和一个 I/O 节点构成。但是，这样的例子仅用于说明，而不是为了限制。每个 I/O 节点提供其处理集合中的计算节点 (102) 以及一组 I/O 设备之间的 I/O 服务。在图 1 的例子中，I/O 节点 (110、114) 被连接，从而 I/O 设备 (118、120、122) 经局域网 (“LAN”)(130) 进行数据通信，该局域网使用高速以太网来实现。

[0042] 图 1 的并行计算机 (100) 还包括服务节点 (116)，其通过网络 (104) 中的一个耦合到计算节点。服务节点 (116) 提供多个计算节点公用的服务，管理计算节点的配置，将程序载入到计算节点，启动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。服务节点 (116) 运行服务应用 (124)，并通过在计算机终端 (122) 上运行的服务应用接口 (126) 来与用户 (128) 通信。

[0043] 在图 1 的例子中，在其中一个计算节点上安装了层次化分布式编译器 (155)，这是根据本发明的实施例能够为层次化分布式处理系统编译软件的自动计算机器的模块。层次化分布式编译器 (155) 包括计算机程序指令，其用于 ：由编译节点编译软件 ；由编译节点来维护将在该编译节点上执行的任何已编译软件 ；基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点选择分布式处理系统的层次结构的下一层中的一个或多个节点 ；以及向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件。图 1 的其它计算节点 (102) 中的每个也能够接收已编译软件 ；确定该已编译软件是否是用于该节点或其后代 ；如果该已编译软件是用于该节点，维护软件以用于执行 ；以及如果该已编译软件是用于后代中的一个，基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点。

[0044] 构成图 1 示出的示例性系统的节点、网络和 I/O 设备的布置仅用于说明，而不是对本发明的限制。根据本发明的实施例能够为层次化分布式处理系统编译软件的数据处理系统可包括图 1 中未示出的本领域技术人员可以想到的其它节点、网络、设备和架构。尽管图 1 的例子中的并行计算机 (100) 包括十六个计算节点 (102)，读者应注意，根据本发明的实施例能够为层次化分布式处理系统编译软件的并行计算机可以包含任何数量的计算节点。除了以太网和 JTAG，这样的数据处理系统中的网络可以支持很多数据通信协议，包括例如 TCP( 传输控制协议 )、IP( 网际协议 ) 以及本领域技术人员能想到的其它协议。本发明的各种实施例可以在图 1 中示出的那些以外的各种硬件平台上实现。

[0045] 根据本发明的实施例为层次化分布式处理系统编译软件一般可以在包含多个计算节点的并行计算机上实现。事实上，这样的计算机可以包含数千个这样的计算节点。每个计算节点自身也是一类计算机，其由一个或多个计算机处理器 ( 或处理核心 )、其自有的计算机存储器和其自有的输入 / 输出适配器构成。因此，为了进一步说明，图 2 给出了可在根据本发明的实施例的能够为层次化分布式处理系统编译软件的并行计算机中使用的示例性计算节点的框图。图 2 的计算节点 (152) 包括一个或多个处理核心 (164) 以及随机存取存储器 (“RAM”)(156)。处理核心 (164) 通过高速存储器总线 (154) 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 (152) 的其它组件。在 RAM(156) 中存储了应用程序 (158)，这是一个计算机程序指令模块，其使用并行算法来实现并行的用户级数据处理。

[0046] 在 RAM(156) 中还存储了消息模块 (160)，这是一个计算机程序指令库，其实现计算节点之间的并行通信，包括点对点操作以及共同操作。应用程序 (158) 通过调用消息模块 (160) 中的软件例程来执行共同操作。并行通信例程的库可以从头开发以用于根据本发明的实施例的系统，所述开发可使用传统的编程语言例如 C 编程语言，并使用传统的编程方法来编写并行通信例程，所述例程在两个独立的数据通信网络上的节点之间发送和接收数据。或者，现有的库可以被改进以根据本发明的实施例进行操作。现有的并行通信库的例子包括“消息传递接口”(“MPI”) 库和“并行虚拟机”(“PVM”) 库。

[0047] 在 RAM 中还存储了层次化分布式编译器 (155)，这是一种自动计算机器的模块，其能根据本发明的实施例为层次化分布式处理系统编译软件。层次化分布式编译器 (155) 包括计算机程序指令，用于 ：由编译节点编译软件 ；由编译节点维护将在该编译节点上执行的任何已编译软件 ；基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点来选择分布式处理系统的层次结构的下一层中的一个或多个节点 ；向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件。并行计算机中的其它计算节点中的每个也能够接收已编译软件 ；确定该已编译软件是否是用于选中的节点或其后代 ；如果该已编译软件是用于选中的节点，由该选中节点来维护该软件以用于执行 ；以及如果该已编译软件是用于后代中的一个，基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点。

[0048] 在 RAM(156) 中还存储了操作系统 (162)，这是一种计算机程序指令和例程的模块，用于应用程序对计算节点的其它资源的访问。典型地，并行计算机的计算节点中的应用程序和并行通信库会运行单个执行线程，而没有用户登录，也没有安全问题，因为该线程被授权完成对该节点的所有资源的访问。因此，由并行计算机的计算节点上的操作系统执行的任务的数量和复杂度比在有多个线程同时运行的串行计算机上的操作系统执行的要更少和更简单。此外，在图 2 的计算节点 (152) 上没有视频 I/O，这是降低操作系统需求的另一个因素。因此，与通用计算机的操作系统相比，该操作系统可以非常轻量级，可以说是一个精简的版本，或者是专门为特定的并行计算机上的操作开发的操作系统。可以有效地改进、简化以用于计算节点的操作系统包括 $UNIX_{TM}$、$LINUX_{TM}$、微软 $XP_{TM}$、$AIX_{TM}$、IBM 的 $i5/OS_{TM}$以及本领域技术人员能想到的其它操作系统。

[0049] 图 2 的示例性计算节点 (152) 包括若干个通信适配器 (172、176、180、188)，用于实现与并行计算机的其它节点的数据通信。这样的数据通信可以通过 RS-232 连接、通过外部总线例如通用串行总线 (“USB”)、通过数据通信网络例如 IP 网络和本领域技术人员能想到的其它方式来串行地实现。通信适配器实现硬件级别的数据通信，由此一台计算机直接或通过网络向另一台计算机发送数据通信。在根据本发明的实施例的为层次化分布式处理系统编译软件的系统中可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 (IEEE 802.3) 适配器以及用于无线网络通信的 802.11b 适配器。

[0050] 图 2 的例子中的数据通信适配器包括千兆以太网适配器 (172)，其将用于数据通信的示例计算节点 (152) 耦合到千兆以太网 (174)。千兆以太网是在 IEEE 802.3 标准中定

义的网络传输标准，其提供每秒钟 10 亿比特（1 千兆比特）的数据速率。千兆以太网是以太网的变体，其在多模光缆、单模光缆或非屏蔽双绞线上操作。

[0051] 图 2 的例子中的数据通信适配器包括 JTAG 从电路 (176)，其将用于数据通信的示例计算节点 (152) 耦合到 JTAG 主电路 (178)。JTAG 是用于测试访问端口的名为标准测试访问端口和边界扫描架构的 IEEE1149.1 标准的常用名，该标准用于使用边界扫描来测试印刷电路板。JTAG 是如此广泛适应，以至于目前边界扫描或多或少是 JTAG 的同义词。JTAG 不仅被用于印刷电路板，也用于对集成电路进行边界扫描，还可用作对嵌入式系统进行调试的工具，从而提供进入系统的方便的“后门”。图 2 的示例计算节点可以是下列所有三者：它典型地包括安装在印刷电路板上的一个或多个集成电路并可以被实现为嵌入式系统，该嵌入式系统具有其自有的处理器、其自有的存储器和其自有的 I/O 能力。通过 JTAG 从电路 (176) 来进行 JTAG 边界扫描可以有效地配置计算节点 (152) 中的处理器寄存器和存储器，以用于根据本发明的实施例为层次化分布式处理系统编译软件。

[0052] 图 2 的例子中的数据通信适配器包括点对点适配器 (180)，其将用于数据通信的示例计算节点 (152) 耦合到针对点对点消息传递操作进行了优化的网络 (108)，例如，被配置为三维环形或网状的网络。点对点适配器 (180) 通过六个双向链路：+x(181)，-x(182)，+y(183)，-y(184)，+z(185) 和 -z(186) 来提供三个通信坐标 x、y 和 z 上的六个方向上的数据通信。

[0053] 图 2 的例子中的数据通信适配器包括全球组合网络适配器 (188)，其将用于数据通信的示例计算节点 (152) 耦合到网络 (106)，该网络针对例如被配置为二叉树的全球组合网络上的共同消息传递操作进行了优化。全球组合网络适配器 (188) 通过以下三个双向链路来提供数据通信：两个到子节点 (190) 和一个到父节点 (192)。

[0054] 示例计算节点 (152) 包括两个算术逻辑单元（“ALU”）。ALU(166) 是每个处理核心 (164) 的组件，且单独的 ALU(170) 专用于全球组合网络适配器 (188) 的排他使用，以用于执行归约操作的算术和逻辑功能。并行通信库 (160) 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。例如，当归约操作的算术或逻辑功能为“总和”或“逻辑或”时，全球组合网络适配器 (188) 可通过使用处理器 (164) 中的 ALU(166) 来执行算术或逻辑操作，或典型地通过使用专用 ALU(170) 来更快得多地执行。

[0055] 图 2 的示例计算节点 (152) 包括直接存储器访问（“DMA”）控制器 (195)，这是用于直接存储器访问的计算机硬件，以及 DMA 引擎 (197)，这是用于直接存储器访问的计算机软件。图 2 的 DMA 引擎 (197) 典型地被存储在 DMA 控制器 (195) 的计算机存储器中。直接存储器访问包括对计算节点的存储器的读写，其降低了中央处理单元 (164) 上的操作负担。DMA 传输实质上将存储器块从一个位置拷贝到另一位置，典型地从一个计算机节点到另一节点。尽管 CPU 可以启动 DMA 传输，CPU 并不执行它。

[0056] 为了进一步说明，图 3A 示出了可在根据本发明的实施例的能够为层次化分布式处理系统编译软件的系统中使用的示例性点对点适配器 (180)。点对点适配器 (180) 被设计用于针对点对点操作进行了优化的数据通信网络，该网络将计算节点组织为三维环形或网状。图 3A 的例子中的点对点适配器 (180) 通过到或从 -x 方向 (182) 上的下一节点以及到或从 +x 方向 (181) 上的下一节点这四个单向数据通信链路来提供沿着 x 轴的数据通信。点对点适配器 (180) 还使用到或从 -y 方向 (184) 上的下一节点以及到或从 +y 方向 (183)

上的下一节点这四个单向数据 (180) 通信链路来提供沿着 y 轴的数据通信。图 3A 中的点对点适配器 (180) 还使用到或从 -z 方向 (186) 上的下一节点以及到或从 +z 方向 (185) 上的下一节点这四个单向数据通信链路来提供沿着 z 轴的数据通信。

[0057] 为了进一步说明，图 3B 示出了可在根据本发明的实施例的能为层次化分布式处理系统编译软件的系统中使用的示例性全球组合网络适配器 (188)。全球组合网络适配器 (188) 被设计用于针对共同操作进行了优化的网络，该网络将并行计算机的计算节点组织为二叉树。图 3B 的例子中的全球组合网络适配器 (188) 通过四个单向数据通信链路 (190) 来提供到或从两个子节点的数据通信。全球组合网络适配器 (188) 还通过两个单向数据通信链路 (192) 来提供到或从父节点的数据通信。

[0058] 为了进一步说明，图 4 给出了示出示例性数据通信网络的线条图，该数据通信网络针对可在根据本发明的实施例的能够为层次化分布式处理系统编译软件的系统中使用的点对点操作进行了优化。在图 4 的例子中，点表示并行计算机的计算机节点 (102)，且点之间的虚线表示计算节点之间的数据通信链路 (103)。所述数据通信链路以类似于针对图 3A 的例子示出的点对点数据通信适配器来实现，所述数据通信网络位于 x、y 和 z 三个轴上，并且到或从六个方向 +x(181)，-x(182)，+y(183)，-y(184)，+z(185) 和 -z。针对点对点操作进行了优化的该数据通信网络将链路和计算节点组织为三维网状网络 (105)。网状网络 (105) 在每个轴上具有回绕的链路，其将网状网络 (105) 中最远端的计算节点连接到网状网络 (105) 的相对一侧。这些回绕的链路构成环形 (107) 的一部分。环形中的每个计算节点在该环形网络中具有由一组 x、y 和 z 坐标唯一指定的位置。读者应注意，为了清楚起见，忽略了 y 和 z 方向上的回绕链路，但它们是和 x 方向上示出的回绕链路类似地配置的。为了说明清楚，图 4 的数据通信网络被示出仅有 27 个计算节点，但读者将认识到，针对可在根据本发明的实施例为层次化分布式处理系统编译软件中使用的点对点操作进行了优化的数据通信网络可以仅包含几个计算节点，或者可以包含数千个计算节点。

[0059] 为了进一步说明，图 5 给出了示出示例性数据通信网络的线条图，该数据通信网络针对可在根据本发明的实施例的能够为层次化分布式处理系统编译软件的系统中使用的共同操作进行了优化。图 5 的示例数据通信网络包括连接到计算节点以将计算节点组织为树的数据通信链路。在图 5 的例子中，点表示并行计算机的计算节点 (102)，且点之间的虚线 (103) 表示计算节点之间的数据通信链路。数据通信链路用类似于例如图 3B 示出的全球组合网络适配器来实现，每个节点典型地提供到或从两个子节点的数据通信以及到或从父节点的数据通信，但有某些例外。二叉树 (106) 中的节点可以被表征为物理根节点 (202)、分支节点 (204)、和叶子节点 (206)。根节点 (202) 具有两个子节点但没有父节点。叶子节点 (206) 每个具有一父节点，但叶子节点没有子节点。分支节点 (204) 每个同时具有父节点和两个子节点。针对共同操作进行了优化的该数据通信网络由此将链路和计算节点组织为二叉树 (106)。为了说明简单，图 5 的数据通信网络示出了仅有 31 个计算节点，但读者将认识到，针对可在根据本发明的实施例为层次化分布式处理系统编译软件中使用的共同操作进行了优化的数据通信网络可以仅包含几个计算节点，或者可以包含数千个计算节点。

[0060] 在图 5 的例子中，树中的每个节点被分配了称为“等级”(250) 的单元标识符。节点的等级唯一地标识了该节点在树形网络中的位置，以用于树形网络中的点对点操作和共

同操作这两者。该例子中的等级被分配为整数，开始 0 被分配给根节点 (202)，1 被分配给树的第二层中的第一个节点，2 被分配给树的第二层中的第二个节点，3 被分配给第三层中的第一个节点，4 被分配给第三层中的第二个节点，以此类推。为了描述简单，这里仅示出了前三层的等级，但树形网络中的所有计算节点都被分配了唯一的等级。

[0061] 为了进一步说明，图 6 给出了根据本发明的实施例的为层次化分布式处理系统编译软件的另一示例性分布式计算系统，其中，该分布式计算系统被实现为混合计算环境。“混合计算环境”这一术语在本说明书中使用时，是指一种计算环境，其中包含操作地耦合到计算机存储器的计算机处理器，从而以在存储器中存储的且在处理器上执行的计算机程序指令的执行的形式来实现数据处理。图 6 的混合计算环境 (600) 包括一个计算节点 (603)，该节点代表较小的单独的混合计算环境，它在和其它类似的计算节点 (602) 结合时，一起构成较大的混合计算环境。

[0062] 图 6 的示例计算节点 (603) 可实现主要的 (principal) 用户级计算机程序执行、从在服务节点上执行的服务应用接受管理服务例如初始程序载入等，该服务节点通过数据通信网络连接到计算节点 (603)。示例计算节点还可以耦合到一个或多个输入 / 输出 (I/O) 节点来进行数据通信，所述 I/O 节点使得计算节点能够获取对数据存储和其它 I/O 功能的访问。I/O 节点和服务节点可通过局域网 (“LAN”) 连接到示例计算节点 (603)、到更大的混合计算环境中的其它计算节点 (602)、以及到 I/O 设备，该局域网用高速以太网或本领域技术人员可以想到的其它结构 (fabric) 类型的数据通信结构来实现。在包含计算节点 (603) 的更大的混合计算环境中可用的 I/O 设备可包括用于计算环境的数据存储设备形式的非易失性存储器、用于混合计算环境的打印机形式的输出设备、以及计算机终端形式的用户 I/O 设备，该用户 I/O 设备执行服务应用接口，其向用户提供接口，用于配置混合计算环境中的计算节点，并启动由计算节点来执行主要的用户级计算机程序指令。

[0063] 图 6 的例子中的计算节点 (603) 以展开图的形式来示出，以有助于更具体地说明混合计算环境 (600)，其可与其它混合计算环境例如其它计算节点 (602) 组合，来构成更大的混合计算环境。图 6 的例子中的计算节点 (603) 包括主计算机 (610)。主计算机 (610) 被称为“主机”，其意义在于它是实现计算节点和任何特定计算节点以外的混合计算环境的其它组件之间的接口功能的主计算机。即，是由主计算机来执行初始启动过程、开机自检、基本 I/O 功能、从服务节点接受用户级程序载入等。

[0064] 图 6 的例子中的主计算机 (610) 包括通过高速存储器总线 (653) 操作地耦合到计算机存储器即随机存取存储器 (“RAM”) (642) 的计算机处理器 (652)。每个主计算机 (610) 中的处理器 (652) 具有定义主计算机架构的一组架构寄存器 (654)。

[0065] 图 6 的示例计算节点 (603) 还包括一个或多个加速器 (604、605)。加速器 (604) 是“加速器”，因为每个加速器具有加速器架构，其相对于主计算机架构、针对特定类别的计算功能的执行速度进行了优化。这样的加速计算功能包括例如矢量处理、浮点操作以及本领域技术人员能想到的其它功能。图 6 的例子中的每个加速器 (604、605) 包括通过高速存储器总线 (651) 操作地耦合到 RAM(640) 的计算机处理器 (648)。在主计算机和加速器 (604、605) 的 RAM(640、642) 中存储的是操作系统 (645)。可在根据本发明的实施例的混合计算环境的主计算机和加速器中使用的操作系统包括 $UNIX_{TM}$、$LINUX_{TM}$、微软 $XP_{TM}$、微软 $Vista_{TM}$、微软 $NT_{TM}$、$AIX_{TM}$、IBM 的 $i5/OS_{TM}$ 以及本领域技术人员能想到的其它操作系统。主计算机中的

操作系统不需要和加速器上使用的操作系统相同。

[0066] 每个加速器 (604、605) 的处理器 (648) 具有一组架构寄存器 (650) 来定义加速器架构。每个加速器的处理器 (648) 的架构寄存器 (650) 和主计算机 (610) 中的处理器 (652) 的架构寄存器 (654) 不同。架构寄存器是可被在每个架构上执行的计算机程序指令访问的寄存器，例如指令寄存器、程序计数器、存储器索引寄存器、栈指针等。由于具有不同的架构，主计算机和加速器支持相同的指令集是不寻常的，尽管是可能的。这样，为了在加速器 (604) 的处理器 (648) 上执行而编译的计算机程序指令一般不期望本机地 (natively) 在主计算机 (610) 的处理器 (652) 上执行，反之亦然。此外，由于主计算机和加速器之间的硬件架构的典型差异，为了在主计算机 (610) 的处理器 (652) 上执行而编译的计算机程序指令一般不期望本机地在加速器 (604) 的处理器 (652) 上执行，即使加速器支持主机的指令集。图 6 的例子中的加速器架构相对于主计算机架构、针对特定类别的计算功能的执行速度进行了优化。即，对于加速器针对其进行了优化的一个或多个功能，这些功能在加速器上将会比在主计算机的处理器上执行要更快进行。

[0067] 混合计算环境的例子包括数据处理系统，其转而包含一台或多台主计算机，每台具有 x86 处理器，以及加速器，其架构寄存器实现 PowerPC 指令集。为了在主计算机的 x86 处理器上执行而编译的计算机程序指令不能本机地在加速器的 PowerPC 处理器上执行。此外，读者将认识到，本说明书中描述的某些示例混合计算环境基于 LANL 走鹃 (roadrunner) 计划（以新墨西哥州的州鸟命名）所开发的洛斯阿拉莫斯国家实验室（“LANL”）超级计算机架构，该超级计算机架构著名地首先实现了“千万亿次”(petaflop) 计算，即每秒钟一千万亿次浮点操作。LANL 超级计算机架构包括很多具有 AMD 皓龙双核处理器的主计算机，其耦合到很多具有 IBM 单元 (Cell) 处理器的加速器，所述皓龙处理器和单元处理球具有不同的架构。

[0068] 在图 6 的例子中，主计算机 (610) 和加速器 (604、605) 互相适配，以便由系统级别的消息传递模块（“SLMPM”）(646) 和至少两种不同结构类型的两个数据通信结构 (628、630) 来进行数据通信。数据通信结构 (628、630) 是实现在主计算机和加速器之间耦合的数据通信的数据通信硬件和软件的配置。数据通信结构类型的例子包括外围设备互连（“PCI”）、PCI 快速（“PCIe”）、以太网、无限带宽 (Infiniband)、光纤通道、小型计算机系统接口（“SCSI”）、外部串行高级技术附件（“eSATA”）、通用串行总线（“USB”）等本领域技术人员可以想到的结构。在图 6 的例子中，主计算机 (610) 和加速器 (604、605) 互相适配，以经 PCIe 通信适配器 (660) 由 PCIe 结构 (630) 和经以太网通信适配器 (661) 由以太网结构 (628) 来进行数据通信。PCIe 和以太网的使用仅用于说明，而不是对本发明的限制。本领域的读者将立刻认识到，根据本发明的实施例的混合计算环境可以包括其它结构类型的结构，例如，PCI、无限带宽、光纤通道、SCSI、eSATA、USB 等。

[0069] SLMPM(646) 是计算机程序指令的模块或库，其向用户级应用暴露应用编程接口（“API”），以用于实现主计算机 (610) 和加速器 (604、605) 之间的基于消息的数据通信。根据本发明的实施例的可被改进以用于 SLMPM 的基于消息的数据通信库的例子包括：

[0070] • 消息传递接口或“MPI”，其在两个版本中是行业标准接口，首先在超级计算 (Supercomputing)1994 公布，没有被任何主要标准机构认可，

[0071] • LANL 超级计算机的数据通信和同步接口（“DACS”），

[0072] • POSIX 线程库（“Pthreads”），用于分布式多线程处理的 IEEE 标准，

[0073] • 开放多处理接口（“OpenMP”），行业所认可的用于并行编程的规范，以及

[0074] • 本领域技术人员能想到的其它库。

[0075] 在该例子中，为了支持主计算机 (610) 和加速器 (604) 之间的基于消息的数据通信，主计算机 (610) 和加速器 (604) 都具有 SLMPM(646)，从而基于消息的通信可以在任何数据通信耦合的两端发起和接收。

[0076] 该例子中的 SLMPM(646) 一般通过下列方式来操作以进行混合计算环境 (600) 中的数据处理 :针对主计算机 (610) 和加速器 (604、605) 之间的多种数据通信模式来监视数据通信性能，接受根据数据通信模式将数据从主计算机发送到加速器的请求 (668)，根据请求的数据通信模式来确定是否发送数据，以及如果根据请求的数据通信模式不发送数据，则 :选择另一数据通信模式并根据选中的数据通信模式来发送数据。在图 6 的例子中，被监视的性能被示为由 SLMPM(646) 在计算节点 (603) 操作期间存储在主计算机 (610) 的 RAM(642) 中的被监视的性能数据 (674)。

[0077] 数据通信模式指定了数据通信结构类型、数据通信链路以及数据通信协议 (678)。数据通信链路 (656) 是主计算机和加速器之间的数据通信连接。在图 6 的例子中，主计算机 (610) 和加速器 (604) 之间的链路 (656) 可以包括 PCIe 连接 (638) 或经以太网 (606) 的以太网连接 (631、632)。在图 6 的例子中的主计算机 (610) 和加速器 (605) 之间的链路 (656) 可以包括 PCIe 连接 (636) 或经过以太网 (606) 的以太网连接 (631、634)。尽管在图 6 的例子中的主计算机和加速器之间仅为每个结构类型示出了一条链路，本领域技术人员将立刻认识到，每种结构类型可以有任何数量的链路。

[0078] 数据通信协议是将信息从主计算机 (610) 发送到加速器 (604) 所需的用于数据表示、信令、验证和错误检测的一组标准规则。在图 6 的例子中，SLMPM(646) 可以选择若干种协议 (678) 中的一种，以用于主计算机 (610) 和加速器之间的数据通信。这样的协议 (678) 的例子包括通过发送和接收操作 (681) 来执行的共享存储器传输（“SMT”）(680)、以及通过 PUT 和 GET 操作 (683) 来执行的直接存储器访问（“DMA”）(682)。

[0079] 共享存储器传输是一种将主计算机和加速器之间的数据传递至共享存储器空间 (658) 的数据通信协议，该空间是为这样的目的而分配的，从而在任一时刻只有数据的一个实例位于存储器中。考虑下列作为图 6 的主计算机 (610) 和加速器 (604) 之间的示例共享存储器传输。应用 (669) 请求 (668) 根据 SMT(680) 协议从主计算机 (610) 向加速器 (604) 传输数据 (676)。这样的请求 (668) 可包括为该共享存储器分配的存储器地址。在该例子中，共享存储器段 (658) 被示为位于加速器 (604) 上的存储器位置，但读者将认识到，共享存储器段可以位于加速器 (604) 上、主计算机 (610) 上、在主计算机和加速器两者之上、或完全离开本地计算节点 (603)—只要该段能够被主机和加速器在需要时访问。为了实现共享存储器传输，主计算机 (610) 上的 SLMPM(646) 通过类似于 TCP 协议中的握手过程来建立与加速器 (604) 上执行的 SLMPM(646) 之间的数据通信连接。SLMPM(646) 然后创建包含头部和净荷数据的消息 (670) 并将该消息插入到用于特定结构的特定链路的消息发送队列。在创建消息时，SLMPM 在消息的头部插入加速器的标识以及在加速器上执行的进程的标识。SLMPM 还将来自请求 (668) 的存储器地址插入到消息中，或者在头部或者作为净荷数据的一部分。SLMPM 还将要传送的数据 (676) 插入到消息 (670) 中作为消息净荷数据的一部

分。消息然后被通信适配器 (660、61) 经过结构 (628、630) 发送到在加速器 (604) 上执行的 SLMPM，在此该 SLMPM 根据消息中的存储器地址在 RAM(640) 中的共享存储器空间 (658) 中存储净荷数据，即被发送的数据 (676)。

[0080] 直接存储器访问 (“DMA”) 是一种有于在主计算机和加速器之间传递数据的通信协议，其降低了计算机处理器 (652) 上的操作负担。DMA 传输实质上实现了将存储器块从一个位置拷贝到另一个位置，典型地从主计算机到加速器或反之亦然。主计算机和加速器中的一个或两者都可包含 DMA 控制器和 DMA 引擎，即用于直接存储器访问的计算机硬件和软件的集合。直接存储器访问包括以对其处理器的降低的负担来读写加速器和主计算机的存储器。例如，加速器的 DMA 引擎可以读写为了 DMA 目的而分配的存储器，而加速器的处理器执行计算机程序指令，或以其他方式继续操作。即，计算机处理器可以发出指令来执行 DMA 传输，但由 DMA 引擎而不是处理器来实现该传输。

[0081] 在图 6 的例子中，只有加速器 (604) 包含 DMA 控制器 (685) 和 DMA 引擎 (684)，而主计算机没有。在该实施例中，主计算机上的处理器 (652) 通过向加速器发送根据 SMT 协议的消息、指示加速器执行远程“GET”操作，来启动从主机到加速器的 DMA 数据传输。在图 6 的例子示出的配置中，加速器 (604) 是唯一包含 DMA 引擎的设备，这是为了说明而不是限制。本领域的读者将立刻认识到，在很多实施例中，主计算机和加速器两者都可以包含 DMA 控制器和 DMA 引擎，而在另外的实施例中，只有主计算机包含 DMA 控制器和 DMA 引擎。

[0082] 为了在图 6 的混合计算环境中实现 DMA 协议，某些存储器区域被分配用于 DMA 引擎的访问。分配这样的存储器可以与其它加速器或主计算机独立地实现，或者可以由另一加速器或主计算机发起并与之合作来完成。根据 SMA 协议分配的共享存储器区域，例如，可以是能被 DMA 引擎使用的存储器区域。即，在图 6 的混合计算环境 (600) 中的 DMA 数据通信的初始设置和实现可以至少部分通过共享存储器传输或另一带外数据通信协议 ( 即相对于 DMA 引擎是带外的 ) 来实现。分配存储器来实现 DMA 传输的延迟相对较大，但一旦被分配，DMA 协议提供了高带宽数据通信，其与很多其它数据通信协议相比需要较少的处理器使用率。

[0083] 直接“PUT”操作是将数据从源设备的 DMA 引擎发送到目标设备的 DMA 引擎的模式。直接“PUT”操作允许数据被发送并存储在目标设备上，而目标设备的处理器很少干涉。为了实现目标设备的处理器对直接“PUT”操作的最小干涉，源 DMA 引擎将要被存储在目标设备上的数据和目标设备的存储位置的特定标识一起传输。源 DMA 知道目标设备上的特定存储位置，因为在目标设备上存储数据的特定存储位置之前已经由目标 DMA 引擎提供给源 DMA 引擎。

[0084] 远程“GET”操作，有时被称为“rGET”，是从源设备上的 DMA 引擎向目标设备上的 DMA 引擎发送数据的另一模式。远程“GET”操作允许数据被发送和存储在目标设备上，而源设备的处理器很少干涉。为了实现源设备的处理器对远程“GET”操作的最小干涉，源 DMA 引擎在可被目标 DMA 引擎访问的存储位置中存储数据，直接或通过共享存储器传输以带外方式来通知目标 DMA 引擎关于准备被发送的数据的存储位置和大小，并且目标 DMA 引擎从该存储位置取回数据。

[0085] 针对多种数据通信模式来监视数据通信性能包括监视用于数据通信链路 (656) 的消息发送请求队列 (662-165) 中的若干请求 (668)。在图 6 的例子中，每个消息发送请

求队列 (662-165) 与特定的数据通信链路 (656) 关联。每个队列 (662-165) 包括消息的条目，所述条目包括将由通信适配器 (660、661) 沿着与队列相关的数据通信链路 (656) 来发送的数据 (676)。

[0086] 针对多种数据通信模式来监视数据通信性能还可包括监视共享存储器空间 (658) 的使用率。在图 6 的例子中，共享存储器空间 (658) 在加速器的 RAM(640) 中分配。使用率是已被存储了用于发送到目标设备、而还没有被目标设备读取或接收的数据的所分配的共享存储器空间的比例，该比例是通过跟踪对所分配的共享存储器的写入和读取来进行监视的。在图 6 的混合计算环境 (600) 中，共享存储器空间是有限的，事实上任何存储器都是有限的。于是，由于共享存储器空间的空间限制，共享存储器空间 (658) 在执行应用程序 (660) 时可能被填满，从而从主计算机 (610) 到加速器的数据传输会被减慢甚至停止。

[0087] 在本发明的某些实施例中，图 6 的混合计算环境 (600) 可以被配置为作为并行计算环境来操作，其中应用程序 (669) 的两个或更多个实例在并行计算环境中的两台或多台主计算机 (610) 上执行。在这样的实施例中，监视各数据通信模式中的数据通信性能还可以包括 :将在并行计算环境的两台或更多台主计算机上执行的应用程序 (669) 的多个实例的数据通信性能信息 (674) 进行汇集。汇集的性能信息 (674) 可用来计算数据通信模式的平均通信延迟、特定结构类型的数据通信链路中的平均请求数、并行计算环境中的多台主计算机和加速器之间的平均共享存储器使用率、以及本领域技术人员所能想到的其它信息。这样的度量的任何组合可以被 SLMPM 用于确定是否根据请求的数据通信模式来发送数据，且如果根据请求的数据通信模式该数据将不被发送，选择另一数据通信模式来发送数据。

[0088] 图 6 的 SLMPM(646) 从主计算机 (610) 上的应用程序 (669) 接收请求，以根据数据通信协议从主计算机 (610) 向加速器 (604) 发送数据 (676)。这样的数据 (676) 可以包括为了被加速器 (604) 执行而编译的计算机程序指令，例如加速器应用程序的可执行文件、用于加速器应用程序的工作数据 (work piece data)、加速器应用程序执行所必须的文件例如库、数据库、驱动器等。接收根据数据通信模式来发送数据 (676) 的请求 (668) 可包括接收通过特定的结构类型来发送数据的请求、接收通过特定的数据通信链路从主计算机向加速器发送数据的请求、或接收根据协议来从主计算机向加速器发送数据的请求。

[0089] 根据数据通信模式来发送数据 (676) 的请求 (668) 可以实现为通过 API 对 SLMPM(646) 的的用户级应用函数调用，该调用根据协议、结构类型和链路来显式地指定数据通信模式。被实现为函数调用的请求可根据函数自身的操作来指定协议。例如，dacs_put() 函数调用可表示通过实现为 DACS 库的 SLMPM 所暴露的 API 进行的调用，以在 DMA“PUT”操作的默认模式中发送数据。从调用应用和编写调用应用的程序员的角度来看，这样的调用表示请求 SLMPM 库根据默认模式来发送数据，程序员已知该默认模式为与显式的 API 调用关联的默认模式。被调用的函数，在该例子中是 dacs_put()，在实施例中可以用多种结构类型、协议和链路来编码，以自己决定是否根据请求的数据通信模式，即根据被调用函数的默认模式，来发送数据。在另一例子中，dacs_send() 指令可表示通过实现为 DACS 库的 SLMPM 所暴露的 API 来进行的调用，以在 SMT “发送”操作的默认方式中发送数据，其中，在实施例中被调用的函数 dacs_send() 再次以多种结构类型、协议和链路来编码，以自己决定是否根据请求的模式来发送数据。

[0090] 函数调用中的特定加速器的标识可以有效地指定结构类型。这样的函数调用可以包括特定加速器的标识作为调用参数。特定加速器的标识例如通过使用 PCIe ID 来有效地指定 PCI 结构类型。在另一个类似的例子中，特定加速器的标识通过使用以太网适配器的介质访问控制（“MAC”）地址来有效地指定以太网结构类型。除了以这种方式来实现在主机上执行的应用进行的函数调用的加速器 ID 从而指定结构类型外，函数调用可以仅包含特定加速器的全局唯一标识作为调用的参数，由此仅指定从主计算机到加速器的链路而不是结构类型。在该情形下，被调用的函数可以实现默认的结构类型以和特定协议一起使用。例如，如果 SLMPM 中被调用的函数被配置为以 PCIe 作为默认结构类型来和 DMA 协议一起使用，且 SLMPM 接收根据 DMA 协议来向加速器 (604) 发送数据的请求，即 DMA PUT 或 DMA 远程 GET 操作，则被调用的函数显式地指定用于 DMA 的默认结构类型，即 PCIe 结构类型。

[0091] 在其中每种结构类型只有一条链路将单台主机适配于单个加速器的混合计算环境中，函数调用的参数中的特定加速器标识还可以有效地指定链路。在其中每种结构类型有多于一条链路来适配主计算机和加速器（例如两条 PCIe 链路将主计算机 (610) 连接到加速器 (604)）的混合计算环境中，被调用的 SLMPM 函数可以针对函数调用的参数中指定的加速器、为加速器标识所指定的结构类型来实现默认链路。

[0092] 在图 6 的例子中的 SLMPM(646) 还基于监视的性能 (674) 来确定是否根据请求的数据通信模式来发送数据 (676)。确定是否根据请求的数据通信模式来发送数据 (676) 可包括确定是否由请求的结构类型来发送数据、是否通过请求的数据通信链路来发送数据、或者是否根据请求的协议来发送数据。

[0093] 在根据本发明的实施例的混合计算环境中，其中监视各数据通信模式中的数据通信性能包括监视用于数据通信链路的消息发送请求队列 (662-165) 中的请求数，确定是否根据请求的数据通信模式来发送数据 (676) 可以通过确定消息发送请求队列中的请求数是否超过预定的阈值来实现。在根据本发明的实施例的混合计算环境中，其中针对多种数据通信模式来监视数据通信性能包括监视共享存储器空间的使用率，确定是否根据请求的数据通信模式来发送数据 (676) 可以通过确定共享存储器空间的使用率是否超过预定的阈值来实现。

[0094] 如果根据请求的数据通信模式将不发送数据，SLMPM(646) 基于监视的性能来选择另一数据通信模式来发送数据，并根据选中的数据通信模式来发送数据 (676)。选择另一数据通信模式来发送数据可以包括：基于监视的性能来选择由其来发送数据的另一数据通信结构类型，选择通过其来发送数据的数据通信链路，以及选择另一数据通信协议。作为例子，考虑请求的数据模式是 DMA 传输，其使用经 PCIe 结构 (630) 的链路 (638) 到加速器 (604) 的 PUT 操作。如果监视的数据性能 (674) 指示与链路 (638) 关联的发送消息请求队列 (662) 中的请求数超过预定的阈值，SLMPM 可以选择另一结构类型，即以太网架构 (628)，以及链路 (631、632)，由此来发送数据 (676)。还考虑监视的性能 (676) 指示共享存储器空间 (658) 的当前使用率低于预定阈值而队列 (662) 中未解决的 DMA 传输的数量超过预定的阈值。在该情形下，SLMPM(646) 也可以选择另一协议，例如共享存储器传输，由此来发送数据 (674)。

[0095] 由 SLMPM 来选择另一数据通信模式来发送数据 (672) 还可以包括基于数据通信消息大小 (672) 来选择数据通信协议 (678)。基于数据通信消息大小 (672) 来选择数据通信

协议 (678) 可以通过确定消息大小是否超过预定的阈值来实现。对于较大的消息 (670)，DMA 协议可以是优选的协议，因为进行较大消息 (670) 的 DMA 传输时的处理器使用率典型地比进行同等大小消息的共享存储器传输时的处理器使用率更小。

[0096] 如上所述，SLMPM 还可以根据选中的数据通信模式来发送数据。根据选中的数据通信模式来发送数据可以包括由选中的数据通信结构类型来发送数据，通过选中的数据通信链路来发送数据，或根据选中的协议来发送数据。通过经由设备驱动器指示用于选中的数据通信模式的数据通信结构类型的通信适配器根据选中的数据通信模式的协议来发送消息 (670)，SLMPM(646) 可以实现根据选中的数据通信模式来发送数据，其中，所述消息在消息头中包括加速器的标识，且在消息净荷中包括要被发送的数据 (676)。

[0097] 在图 6 的例子中，计算节点之一的主计算机 (610) 和加速器 (604) 两者上面都安装了层次化分布式编译器 (155)。为了完整，在主计算机和加速器中都示出了层次化分布式编译器 (155)。事实上，根据本发明的实施例的层次化分布式编译器 (155) 可安装在主计算机上，或安装在一个或多个加速器上，或安装在主计算机和一个或多个加速器这两者上，如本领域的技术人员可想到的。图 6 的层次化分布式编译器 (155) 是能够根据本发明的实施例为层次化分布式处理系统编译软件的自动计算机器的模块。层次化分布式编译器 (155) 包括计算机程序指令，其用于 ：由编译节点编译软件 ；由编译节点来维护将在该编译节点上执行的任何已编译软件 ；基于任何已编译软件是否是用于选中的节点或选中节点的后代、由编译节点来选择分布式处理系统的层次结构的下一层中的一个或多个节点 ；向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件。图 16 的其它计算节点 (602) 中的每个还能够接收已编译软件 ；确定该已编译软件是否是用于选中的节点或其后代 ；如果该已编译软件是用于该节点，维护该软件以用于执行 ；以及如果该已编译软件是用于后代中的一个，基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送到该选中的另一节点。

[0098] 为了进一步说明，图 7 示出了根据本发明的实施例的为层次化分布式处理系统编译软件的示例性方法。编译是将用计算机语言（通常是高级编程语言）编写的源代码转换为另一典型地可执行的计算机语言（典型地具有二进制形式且有时被称为目标代码）的过程。根据本发明的实施例的编译典型地用根据本发明的在编译节点上安装的层次化分布式编译器来实现。这样的层次化分布式编译器典型地能够编译软件，以在若干不同类型的目标计算机上使用。这样，分布式层次化编译器可以将部分未编译源软件编译为目标对象代码，以在不同类型的计算机上使用。

[0099] 层次化分布式处理系统可以用多种方式来实现，例如，以树形结构来实现。这样的树形结构可以是 k 叉的 (k-ary)，即任何阶，二叉，或本领域技术人员能想到的任何其它形式的树形结构。或者，根据本发明的层次化分布式处理系统可以以本领域技术人员不认为是树形结构的其它形式来实现。图 7 的方法可以在与上述示例分布式计算系统类似的分布式计算机系统中实现 ：图 1-5 的示例并行计算机、图 6 的示例混合计算环境、以及本领域技术人员所能想到的其它形式。

[0100] 图 7 的方法包括识别 (802) 一个或多个编译节点。如上所述，一个或多个编译节点将用计算机语言（典型地是高级编程语言）编写的源代码转换为另一通常可执行的计算机语言（典型地具有二进制形式并且有时被称为目标代码）。编译节点编译要在该编译节

点自己上面执行的软件部分，以及要在层次化分布式网络的另一节点上执行的其它软件部分。

[0101] 根据图 7 的方法来识别 (802) 一个或多个编译节点可通过选择针对编译进行了计算优化的一个或多个节点来实现。选择针对编译进行了计算优化的一个或多个节点可以包括基于其 I/O 能力、处理能力和存储器能力来识别节点。通常这些能力的特定平衡对于编译是最优的。这样的最优平衡对于编译不同类型的软件可以是不同的，从而选择针对编译进行了计算优化的一个或多个节点可包括基于要编译的特定软件来选择一个或多个节点。

[0102] 根据图 7 的方法来识别 (802) 一个或多个编译节点还可以通过下列方式来实现：根据节点在层次化分布式处理系统的拓扑结构中的位置，来选择针对编译进行了优化的一个或多个节点。例如，在树形结构的层次化数据处理系统中，作为根节点的节点或者具有很多后代的节点可位于拓扑结构中，从而该节点被优化，以为其后代编译软件。在本说明书中使用的术语后代是指位于层次化数据处理系统中编译节点下面的层次中、并且在层次化处理系统包含该编译节点的分支上的节点。

[0103] 图 7 的方法还包括向一个或多个编译节点提供 (804) 要编译的软件，其中，该要编译的软件的至少一部分将在一个或多个其它节点上执行。向一个或多个编译节点提供 (804) 要编译的软件可通过下列方式来实现：将要编译的软件在消息中发送给编译节点，将该软件下载到编译节点，由系统管理员在编译节点上安装该软件，或者本领域技术人员能想到的向一个或多个编译节点提供要编译的软件的任何其它方式。

[0104] 图 7 的方法还包括由编译节点编译 (806) 软件。由编译节点编译 (806) 软件可以通过下列方式来实现：识别要在编译节点上执行的软件部分，并将未编译软件的用计算机语言编写的代码转换为用于在编译节点上执行的可执行计算机语言。由编译节点编译 (806) 软件可以通过下列方式来实现：识别要在另一节点上执行的软件部分，识别另一节点的目标执行环境，以及将未编译软件的用计算机语言编写的软件转换为用于在该另一节点上执行的可执行计算机语言。

[0105] 图 7 的方法还包括由编译节点来维护 (808) 要在该编译节点上执行的任何已编译软件。由编译节点来维护 (808) 要在该编译节点上执行的任何已编译软件可通过下列方式来实现：存储要在该编译节点上执行的任何已编译软件以用于执行。

[0106] 图 7 的方法还包括基于任何已编译软件是否是用于选中的节点或选中节点的后代、由编译节点来选择 (810) 分布式处理系统的层次结构的下一层中的一个或多个节点，以及向被选中的节点仅发送 (812) 将由该选中节点或选中节点的后代执行的已编译软件。基于任何已编译软件是否是用于选中的节点或选中节点的后代、由编译节点来选择 (810) 分布式处理系统的层次结构的下一层中的一个或多个节点可以通过下列方式来实现：遍历层次化分布式处理系统的拓扑结构的表示、以识别要执行已编译软件的节点的位置，确定执行已编译软件的那些节点所在的层次化数据处理系统的分支，识别编译节点的哪个子节点也在层次化数据处理系统的该分支上。

[0107] 向被选中的节点仅发送 (812) 将由该选中节点或选中节点的后代执行的已编译软件可通过下列方式来实现：创建消息，该消息包含将由选中节点或选中节点的后代执行的已编译软件部分，并将该消息发送到选中的节点。向被选中的节点仅发送 (812) 将由该选中节点或选中节点的后代执行的已编译软件还可以通过下列方式来实现：通知选中节点

关于已编译软件的位置以下载到该节点来执行已编译软件，或者本领域技术人员所能想到的向被选中的节点仅发送 (812) 将由该选中节点或选中节点的后代执行的已编译软件的任何其它方法。

[0108] 被选中的节点可以执行或者可以不执行已编译软件。即，已编译软件可由选中节点下面的层次中的节点来执行。因此，为了进一步说明，图 8 给出了示出根据本发明的实施例的为层次化分布式处理系统编译软件的另一示例性方法的流程图。图 8 的方法与图 7 的方法类似，这在于图 8 的方法包括 ：识别 (802) 一个或多个编译节点 ；向一个或多个编译节点提供 (804) 要编译的软件，其中，要编译的软件的至少一部分将由一个或多个其它节点执行 ；由编译节点编译 (806) 该软件 ；由编译节点来维护 (808) 要在该编译节点上执行的任何已编译软件 ；基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点来选择 (812) 分布式处理系统的层次结构的下一层中的一个或多个节点 ；以及向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件。

[0109] 图 8 的方法还包括由层次化分布式处理系统中在编译节点下面的节点来实现的其它步骤。图 8 的方法包括由选中的节点来接收 (814) 已编译软件。已编译软件可以在消息中接收，由编译节点来识别以下载，或者以本领域技术人员能想到的其它方式来接收。

[0110] 图 8 的方法还包括确定 (816) 该已编译软件是否是用于选中的节点或其后代。确定 (816) 该已编译软件是否是用于选中的节点或其后代可以通过下列方式来实现 ：从编译节点接收要执行已编译软件的特定部分的节点标识，并确定该标识的节点是否是选中节点，或者该标识的节点是否是其后代中的一个。

[0111] 如果该已编译软件是用于选中的节点，图 8 的方法还包括由该选中节点来维护 (818) 该软件以用于执行。由该选中节点来维护 (818) 软件以用于执行可以通过下列方式来实现 ：存储要在被选中节点上执行的任何已编译软件以用于执行。

[0112] 如果该已编译软件是用于后代中的一个，图 8 的方法还包括基于后代、为已编译软件选择 (820) 层次化分布式处理系统的下一层中的另一节点，并将已编译软件发送 (822) 到该选中的另一节点。基于后代、为已编译软件选择 (820) 层次化分布式处理系统的下一层中的另一节点可以通过下列方式来实现 ：遍历层次化分布式处理系统的拓扑结构表示以识别要执行已编译软件的节点的位置，确定要执行已编译软件的那些节点所在的层次化数据处理系统的分支，识别编译节点的哪个子节点也在层次化数据处理系统的该分支上。

[0113] 向被选中的另一节点发送 (822) 已编译软件可通过下列方式来实现 ：创建消息，该消息包含将由选中的另一节点或该选中的另一节点的后代执行的已编译软件部分，并将该消息发送到选中的另一节点。向被选中的另一节点发送 (822) 已编译软件还可以通过下列方式来实现 ：通知选中的另一节点关于已编译软件的位置以下载到该节点来执行已编译软件，或者本领域技术人员所能想到的向被选中的另一节点发送 (822) 已编译软件的任何其它方法。

[0114] 为了进一步说明，图 9 示出了根据本发明的实施例的为层次化分布式处理系统编译软件的系统的示例性用例的框图。在图 9 的例子中，编译膝上型电脑 (702) 具有未编译的软件 (722)。未编译的软件 (722) 具有由计算机 (704) 执行的软件部分 (724)、由一个或多个图形处理单元 ( “GPU”) (704) 执行的软件部分 (726 和 728)、由并行计算机 (712) 中

的计算节点执行的软件部分 (728、730、732、734)、由混合计算环境前端节点 (714) 执行的软件部分 (736)、由混合计算系统中的主计算机 (718) 执行的软件部分 (738、740、742)、和由混合计算系统中的加速器 (720) 执行的软件部分 (744、746、748 和 750)。

[0115] 在图 9 的例子中，编译膝上型电脑 (702) 编译用于计算机 (704) 的软件部分 (724)，并将该已编译软件部分发送到计算机 (704) 来执行。在图 9 的例子中，编译膝上型电脑 (702) 还编译用于特定 GPU(708) 的软件部分 (726 和 728)，并将这些已编译软件部分发送到计算机 (704)，其转而将该已编译软件部分发送到特定的 GPU，该 GPU 将会执行该已编译软件部分。

[0116] 在图 9 的例子中，编译膝上型电脑 (702) 编译用于并行计算机的特定计算节点 (712) 的软件部分 (728、730、732 和 734)，并将那些已编译软件部分发送到并行计算机的 I/O 节点 (710)，其转而将该已编译软件部分发送到计算节点 (712) 的根节点。计算节点的根节点然后识别哪个子节点具有将执行所述已编译软件部分的后代，并向每个子节点仅发送将由该子节点或其后代执行的部分。每个子节点然后接收那些部分，确定它是否将执行所述部分或者是否其后代中的一个将执行所述部分，并仅将用于其后代的那些部分发送到同一分支上作为后代的子节点。通过这种方式，一层到一层，已编译软件的部分被发送到特定的计算机节点，以执行该已编译软件。

[0117] 在图 9 的例子中，编译膝上型电脑 (702) 编译用于混合计算环境前端节点 (714) 的软件部分 (736)，并将该已编译软件部分发送到混合计算机环境前端节点 (714) 来执行。在图 9 的例子中，编译膝上型电脑 (702) 还编译用于特定主计算机 (718) 的软件部分 (726、738、740 和 742)，并将那些已编译软件部分发送到混合计算环境前端节点 (714)，其转而将该已编译软件部分发送到特定的主计算机，该主计算机将执行该已编译软件部分。在图 9 的例子中，编译膝上型电脑 (702) 还编译用于主计算机 (718) 的特定加速器 (720) 的软件部分 (744、746、748 和 750)，并将那些已编译软件部分发送到混合计算环境前端节点 (714)，其转而将该已编译软件部分发送到用于那些加速器的特定主计算机，其转而将该已编译软件部分发送到特定的加速器，该加速器将执行该已编译软件部分。

[0118] 在上面的例子中，用单个编译节点对为层次化分布式处理系统编译软件进行了一般地讨论。这是为了说明而不是限制。事实上，在本发明的很多实施例中，多于一个节点可以根据本发明为层次化分布式处理系统编译软件。

[0119] 所属技术领域的技术人员知道，本发明可以实现为系统、方法或计算机程序产品。因此，本公开可以具体实现为以下形式，即：可以是完全的硬件、也可以是完全的软件（包括固件、驻留软件、微代码等），还可以是硬件和软件结合的形式，本文一般称为“电路”、“模块”或“系统”。此外，在一些实施例中，本发明还可以实现为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，该计算机可读介质中包含计算机可读的程序代码。

[0120] 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信号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，或者任意以上的组合。计算机可读存储介质的更具体的例子（非穷举的列表）包括：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)、可擦式可编程只读存储器 (EPROM 或闪存)、光纤、便携式紧凑磁盘只读存储器 (CD-ROM)、光存储器件、磁存储器件、

或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0121] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括——但不限于——电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还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。

[0122] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于——无线、电线、光缆、RF 等等，或者上述的任意合适的组合。

[0123] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如 Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如"C"语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户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。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网 (LAN) 或广域网 (WAN)—连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接）。

[0124] 以上参照本发明实施例的方法、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。应当理解，流程图和 / 或框图的每个方框以及流程图和 / 或框图中各方框的组合，都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，从而生产出一种机器，这些计算机程序指令通过计算机或其它可编程数据处理装置执行，产生了实现流程图和 / 或框图中的方框中规定的功能 / 操作的装置。

[0125] 也可以把这些计算机程序指令存储在能使得计算机或其它可编程数据处理装置以特定方式工作的计算机可读介质中，这样，存储在计算机可读介质中的指令就产生出一个包括实现流程图和 / 或框图中的方框中规定的功能 / 操作的指令装置 (instruction means) 的制造品 (manufacture)。

[0126] 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，使得在计算机、其它可编程数据处理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，从而使得在计算机或其它可编程装置上执行的指令能够提供实现流程图和 / 或框图中的方框中规定的功能 / 操作的过程。

[0127] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、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一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规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也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。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，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的

组合来实现。

[0128] 本说明书的描述仅是为了说明的目的，而不应理解为具有限制意义。本发明的范围仅由所附权利要求书限定。

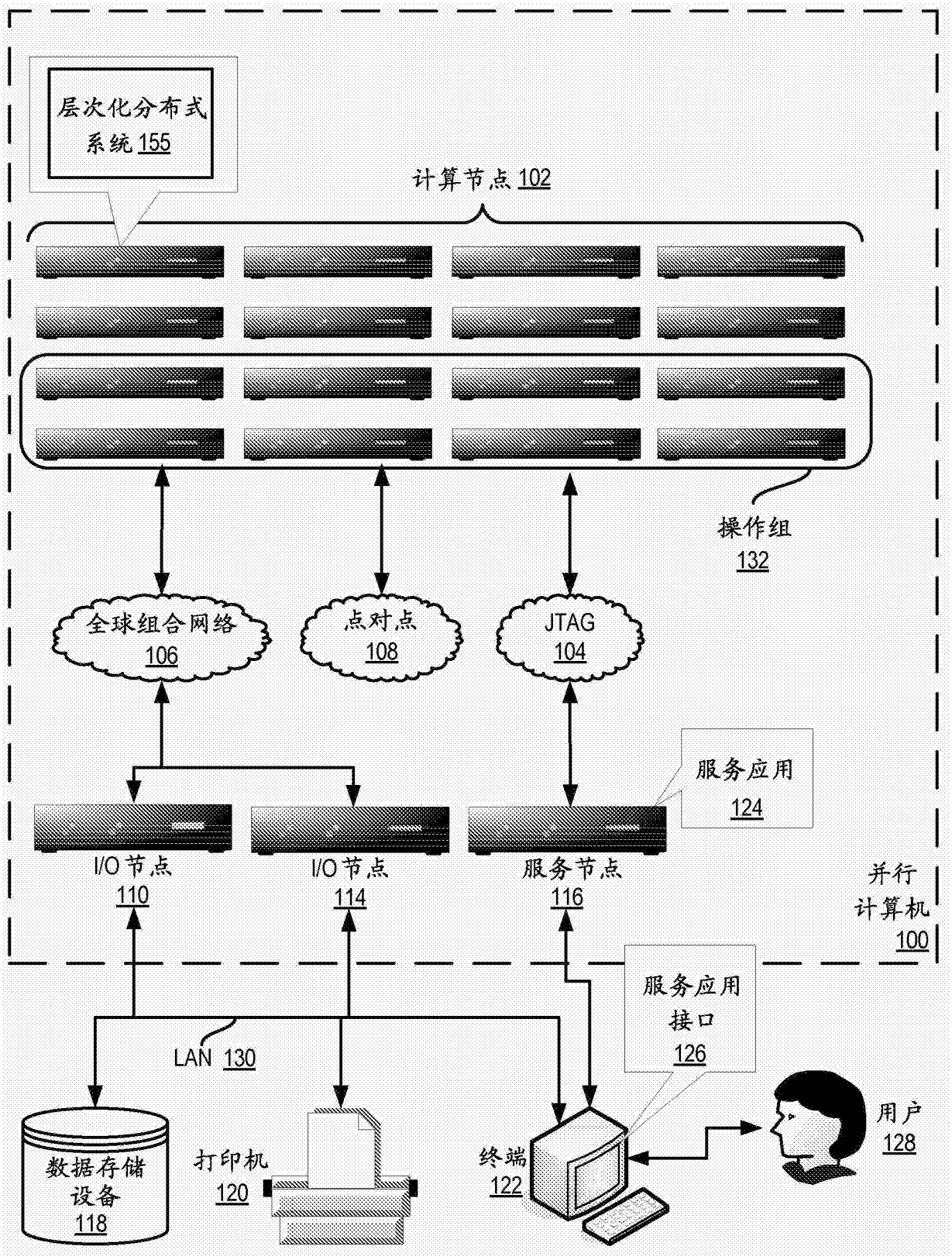

层次化分布式
系统 155
计算节点 102
操作组
132
全球组合网络
106
点对点
108
JTAG
104
服务应用
124
I/O 节点
110
I/O 节点
114
服务节点
116
并行
计算机
100
服务应用
接口
126
LAN 130
数据存储
设备
118
打印机
120
终端
122
用户
128

图 1

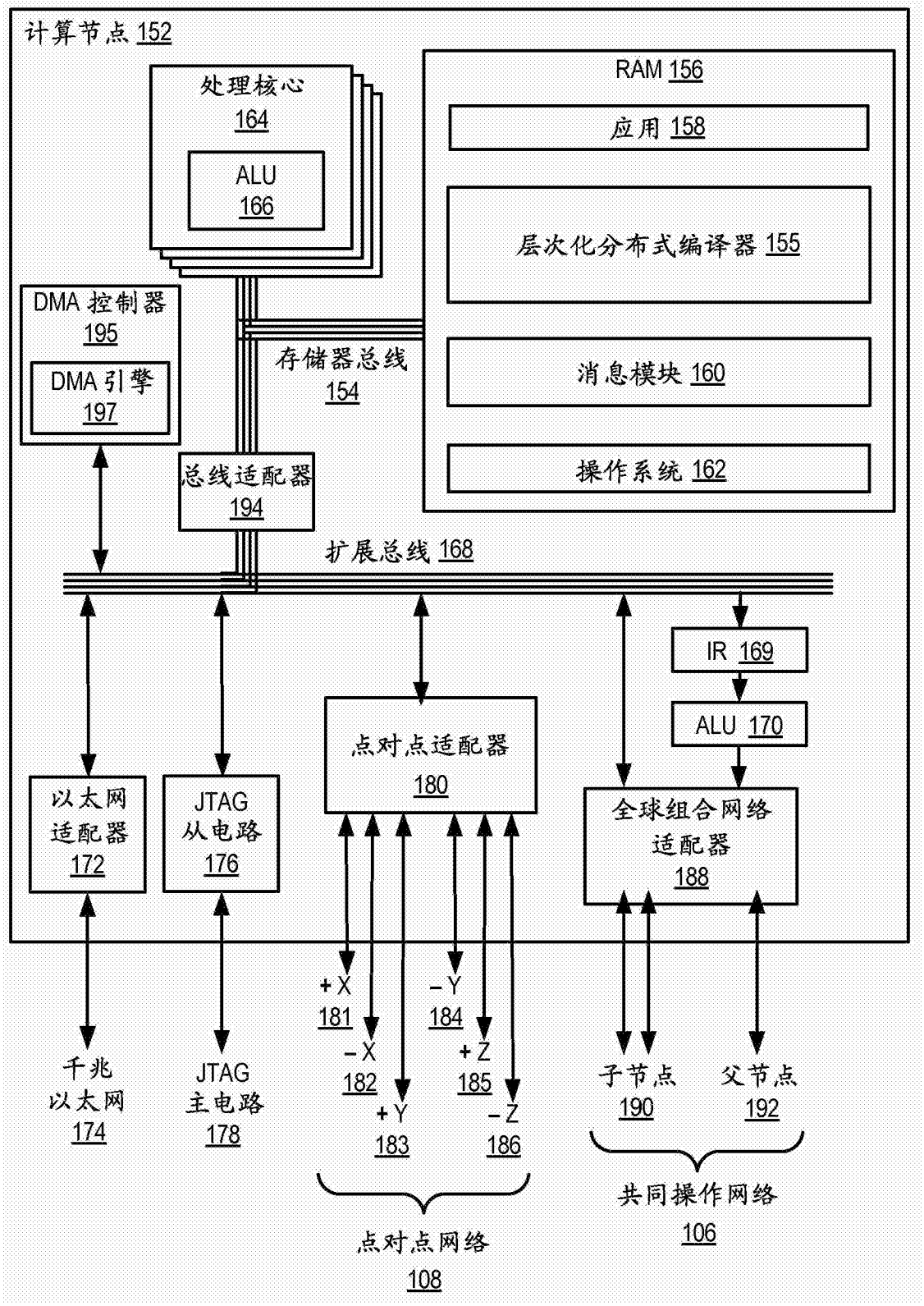

计算节点 152
处理核心 164
ALU 166
RAM 156
应用 158
层次化分布式编译器 155
消息模块 160
操作系统 162
DMA 控制器 195
DMA 引擎 197
存储器总线 154
总线适配器 194
扩展总线 168
IR 169
ALU 170
点对点适配器 180
以太网适配器 172
JTAG 从电路 176
全球组合网络适配器 188
+X 181
-X 182
+Y 183
-Y 184
+Z 185
-Z 186
千兆以太网 174
JTAG 主电路 178
子节点 190
父节点 192
点对点网络 108
共同操作网络 106

图 2

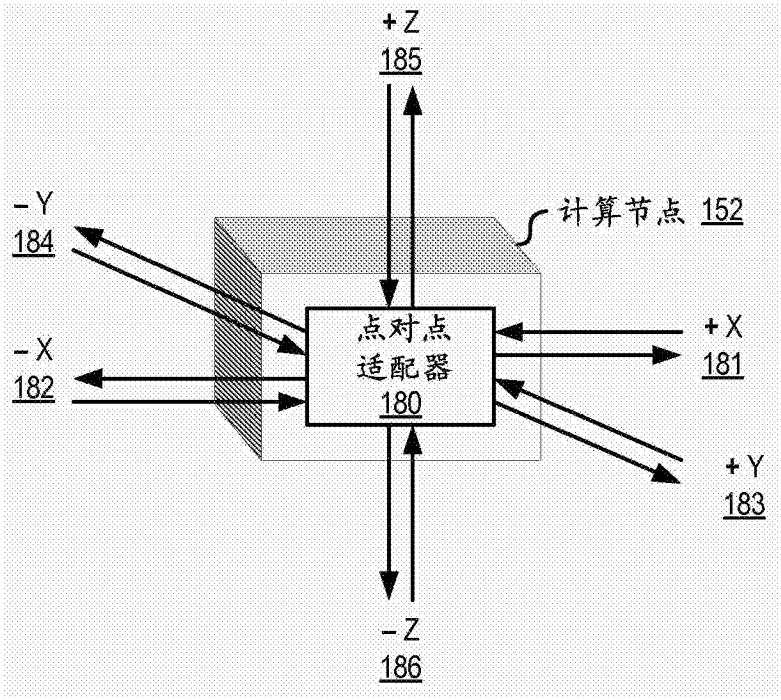

+Z
185
-Y
184
计算节点 152
点对点
适配器
180
+X
181
-X
182
+Y
183
-Z
186

图 3A

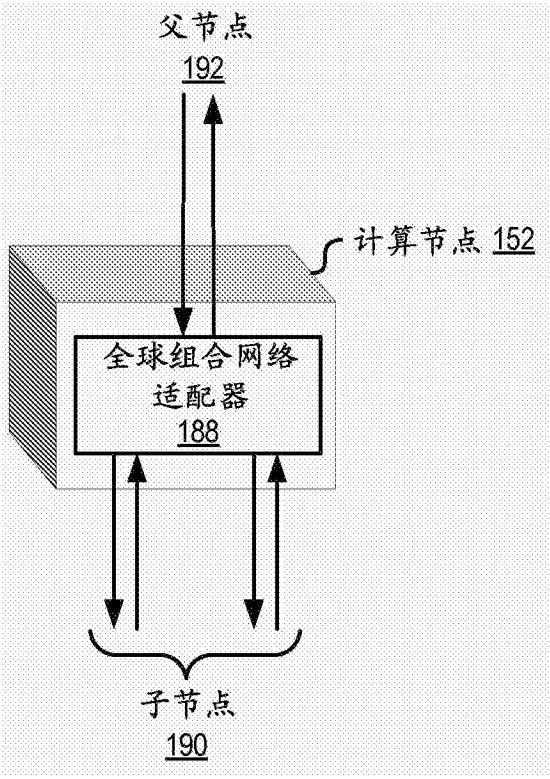

父节点
192
计算节点 152
全球组合网络
适配器
188
子节点
190

图 3B

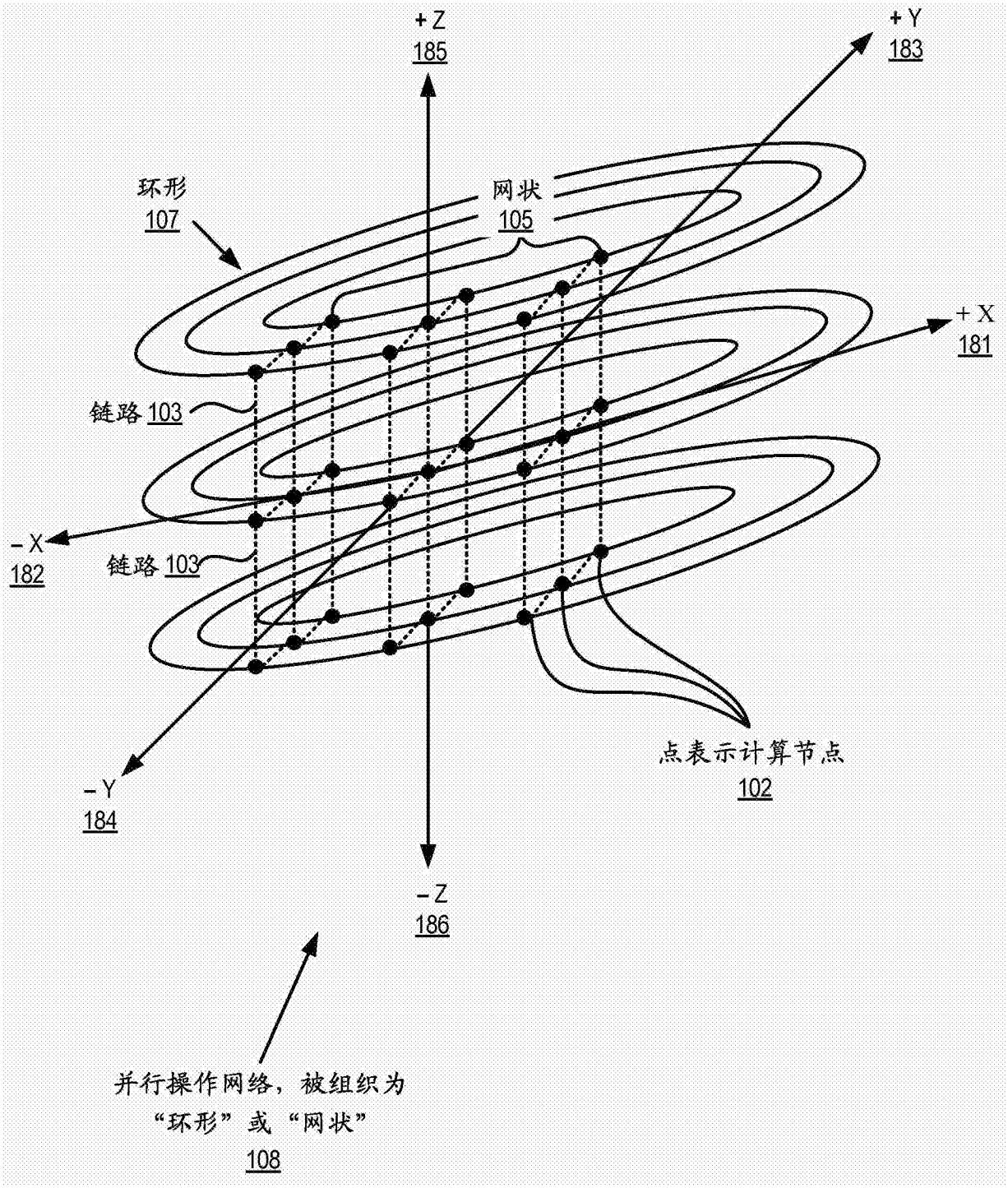

+Z
185
+Y
183
环形
107
网状
105
+X
181
链路 103
-X
182
链路 103
-Y
184
点表示计算节点
102
-Z
186
并行操作网络，被组织为
“环形”或“网状”
108

图 4

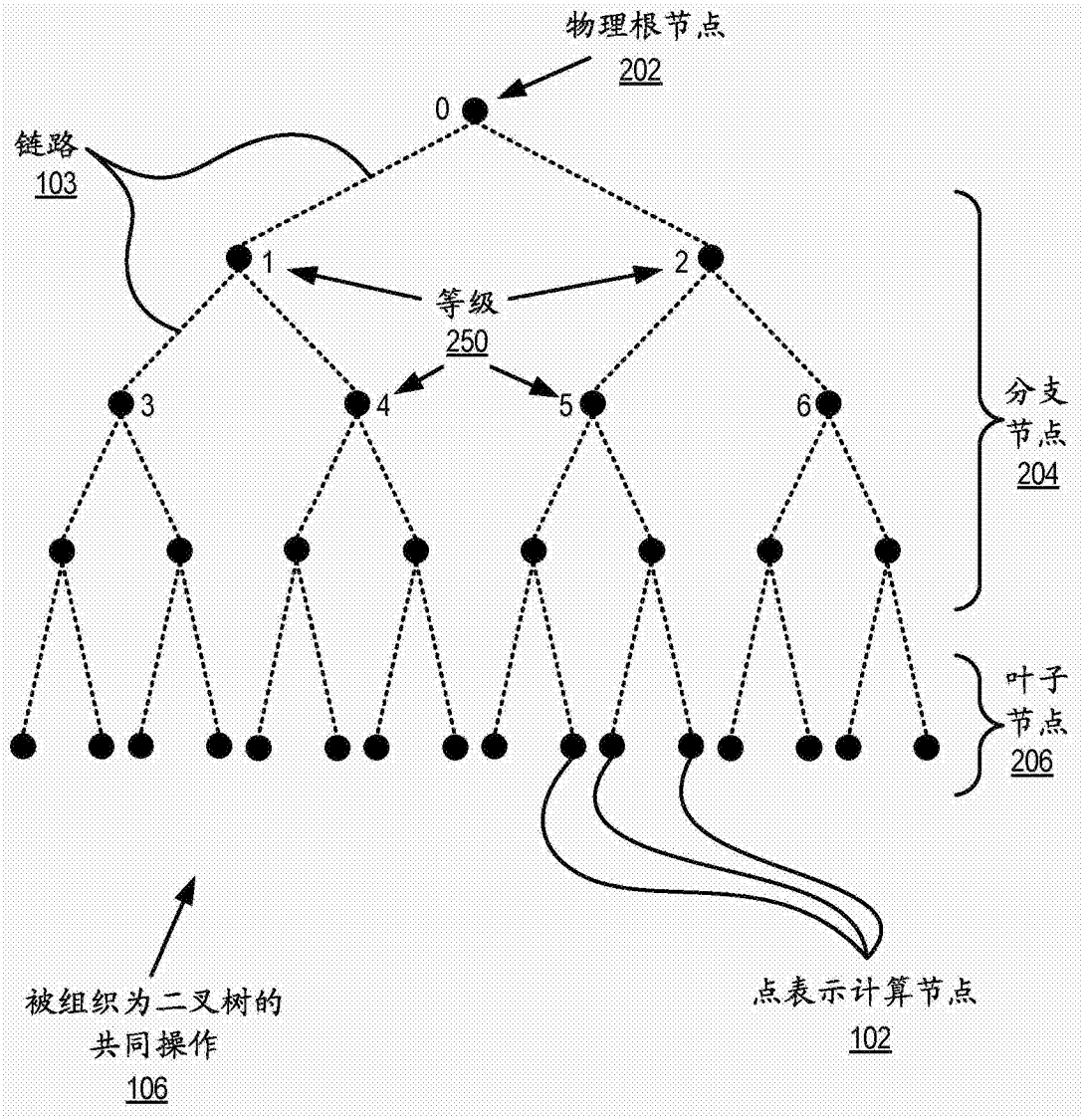

物理根节点
202
0
链路
103
1
2
等级
250
3
4
5
6
分支
节点
204
叶子
节点
206
被组织为二叉树的
共同操作
106
点表示计算节点
102

图 5

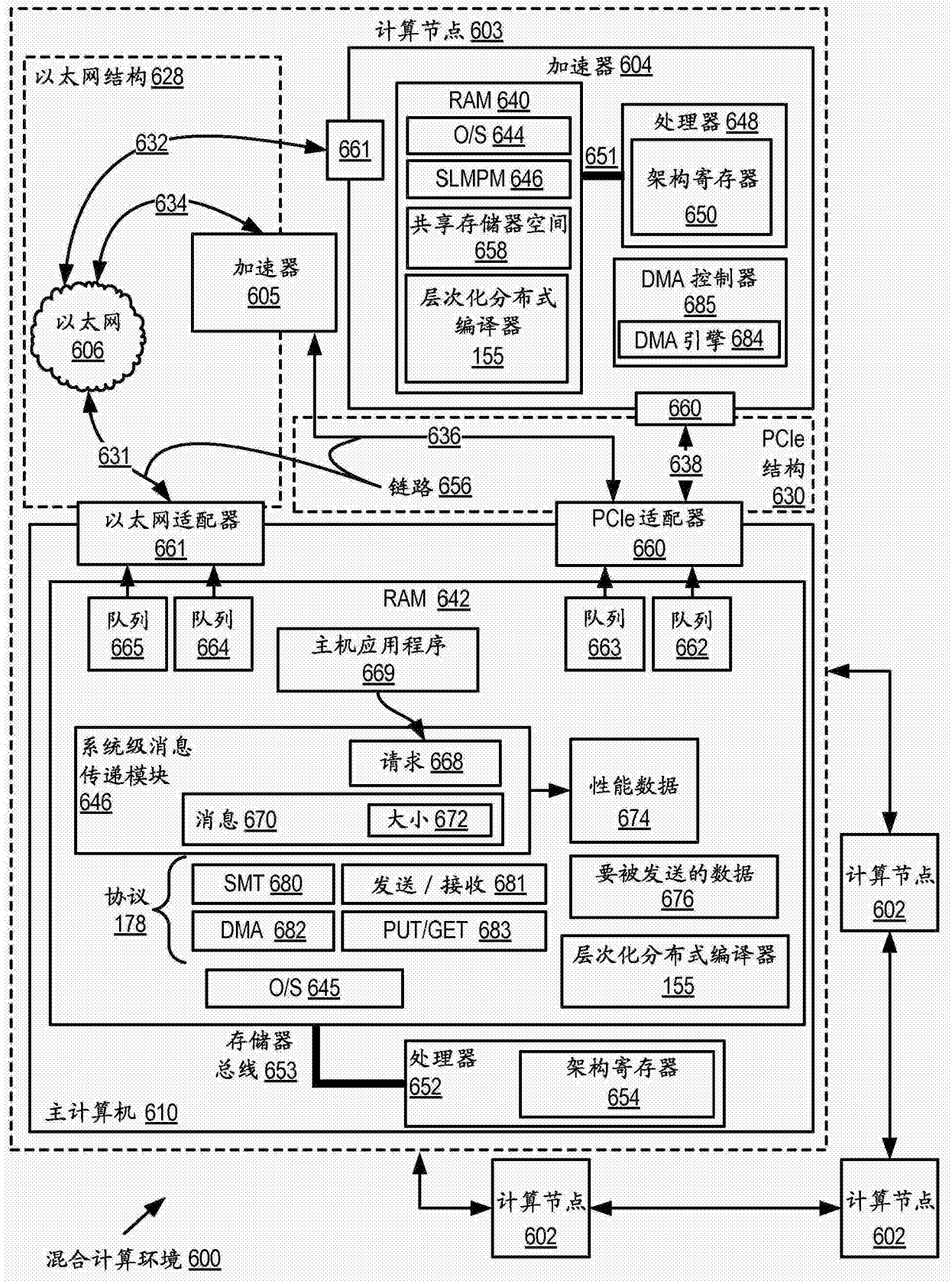

计算节点 603
以太网结构 628
加速器 604
RAM 640
O/S 644
SLMPM 646
共享存储器空间 658
层次化分布式编译器 155
处理器 648
架构寄存器 650
651
DMA 控制器 685
DMA 引擎 684
661
632
634
加速器 605
以太网 606
660
636
638
PCIe 结构 630
631
链路 656
以太网适配器 661
PCIe 适配器 660
RAM 642
队列 665
队列 664
主机应用程序 669
队列 663
队列 662
系统级消息传递模块 646
请求 668
消息 670
大小 672
性能数据 674
协议 178
SMT 680
发送 / 接收 681
DMA 682
PUT/GET 683
O/S 645
要被发送的数据 676
层次化分布式编译器 155
存储器总线 653
处理器 652
架构寄存器 654
主计算机 610
计算节点 602
计算节点 602
计算节点 602
混合计算环境 600

图 6

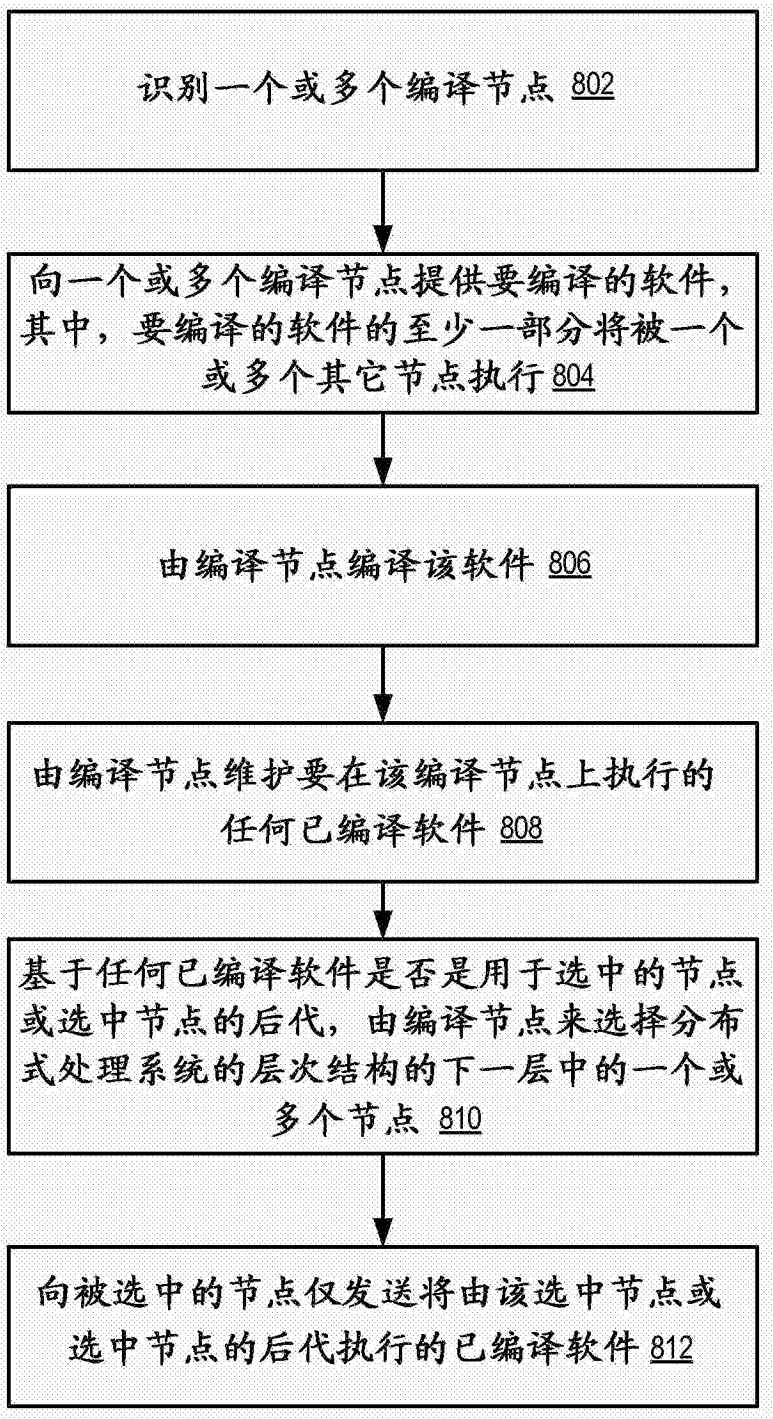

识别一个或多个编译节点 802
向一个或多个编译节点提供要编译的软件，其中，要编译的软件的至少一部分将被一个或多个其它节点执行 804
由编译节点编译该软件 806
由编译节点维护要在该编译节点上执行的任何已编译软件 808
基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点来选择分布式处理系统的层次结构的下一层中的一个或多个节点 810
向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件 812

图 7

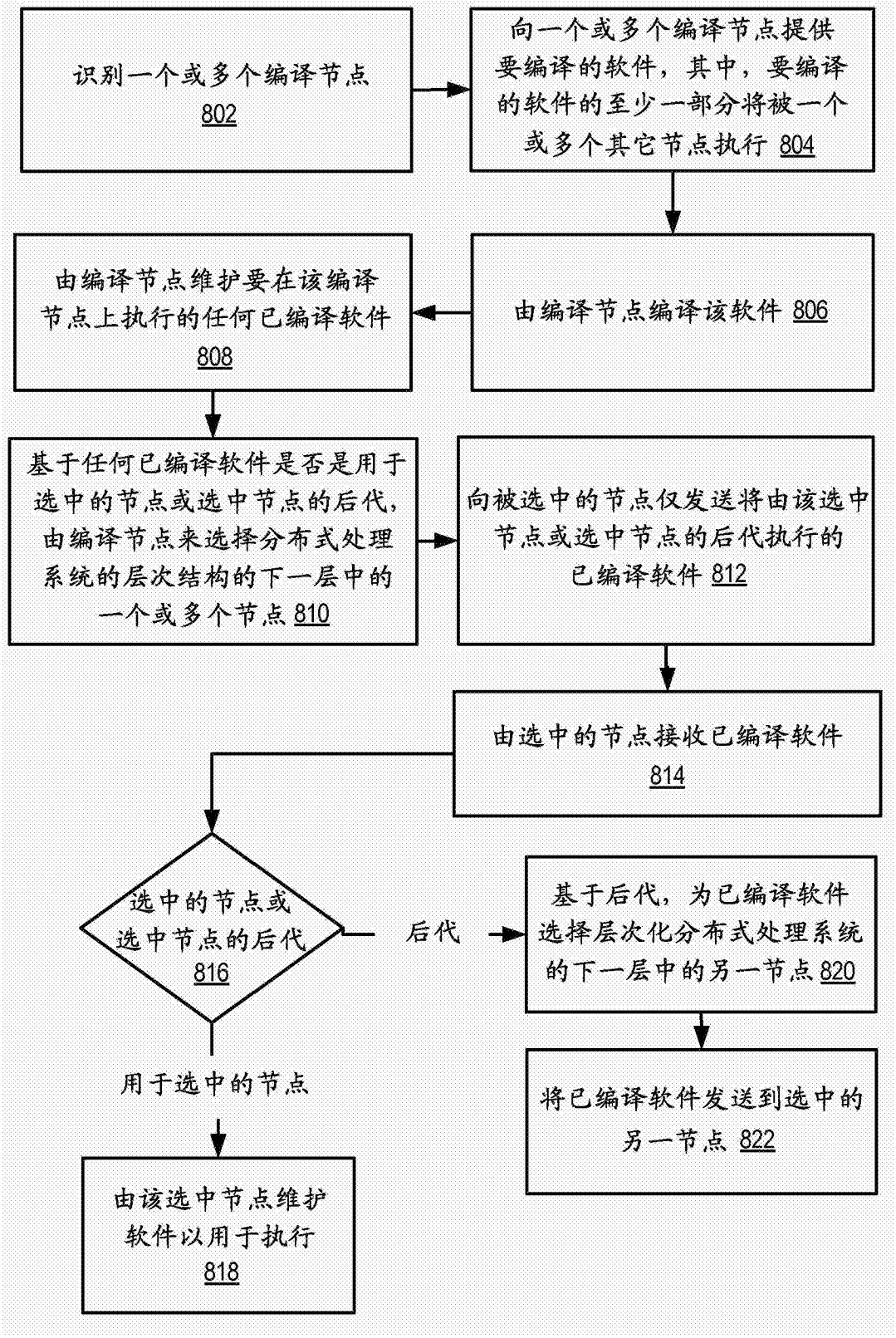

识别一个或多个编译节点 802
向一个或多个编译节点提供要编译的软件，其中，要编译的软件的至少一部分将被一个或多个其它节点执行 804
由编译节点编译该软件 806
由编译节点维护要在该编译节点上执行的任何已编译软件 808
基于任何已编译软件是否是用于选中的节点或选中节点的后代，由编译节点来选择分布式处理系统的层次结构的下一层中的一个或多个节点 810
向被选中的节点仅发送将由该选中节点或选中节点的后代执行的已编译软件 812
由选中的节点接收已编译软件 814
选中的节点或选中节点的后代 816
后代
基于后代，为已编译软件选择层次化分布式处理系统的下一层中的另一节点 820
将已编译软件发送到选中的另一节点 822
用于选中的节点
由该选中节点维护软件以用于执行 818

图 8

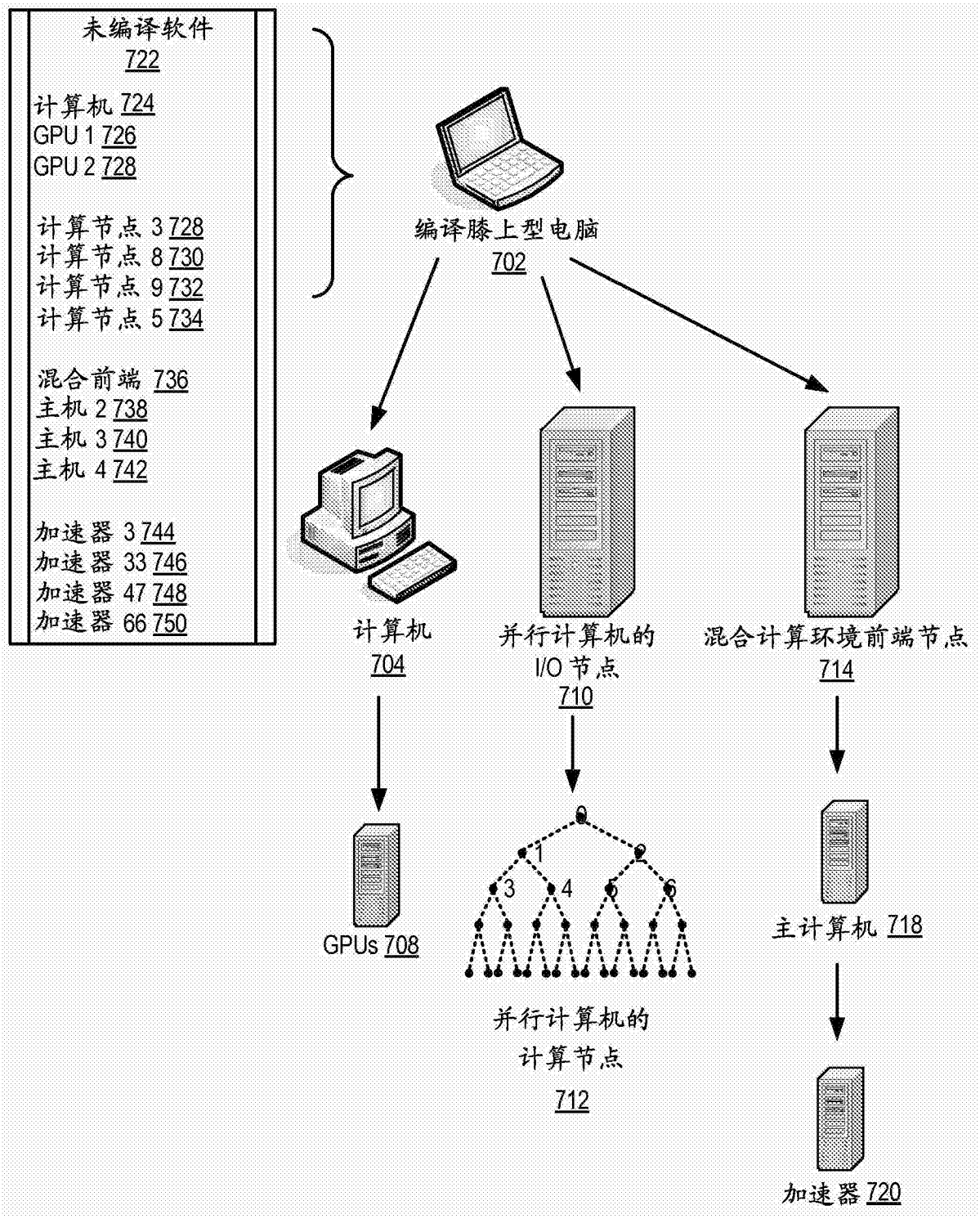

未编译软件
722
计算机 724
GPU 1 726
GPU 2 728
计算节点 3 728
计算节点 8 730
计算节点 9 732
计算节点 5 734
混合前端 736
主机 2 738
主机 3 740
主机 4 742
加速器 3 744
加速器 33 746
加速器 47 748
加速器 66 750
编译膝上型电脑
702
计算机
704
并行计算机的
I/O 节点
710
混合计算环境前端节点
714
GPUs 708
0
1
2
3
4
5
6
并行计算机的
计算节点
712
主计算机 718
加速器 720

图 9