



[12] 发明专利说明书

专利号 ZL 02805377. X

[45] 授权公告日 2006 年 3 月 8 日

[11] 授权公告号 CN 1244878C

[22] 申请日 2002.2.25 [21] 申请号 02805377. X

[30] 优先权

[32] 2001. 2. 24 [33] US [31] 60/271, 124

[86] 国际申请 PCT/US2002/005574 2002. 2. 25

[87] 国际公布 WO2002/069097 英 2002. 9. 6

[85] 进入国家阶段日期 2003. 8. 22

[71] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 盖恩 V·班诺特 陈 东

艾伦 G·加拉 马克 E·贾姆帕帕

菲利普·海德伯格

伯克哈德 D·斯坦马彻 - 伯罗

帕夫罗斯 M·弗拉纳斯

审查员 张桂华

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 吴立明 王 勇

权利要求书 5 页 说明书 17 页 附图 7 页

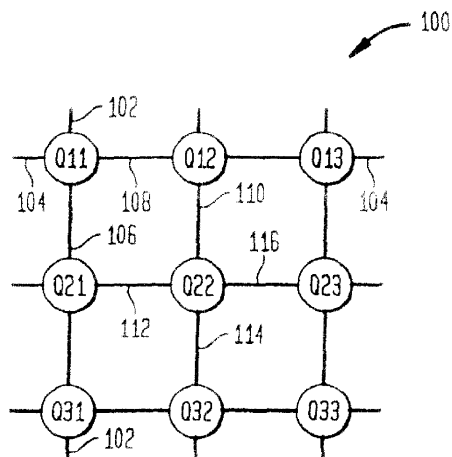
[54] 发明名称

在分布式存储器上实现多维快速傅里叶变换的方法和系统

[57] 摘要

本发明的目的在于一种用于实现一个多维数组的多维快速傅里叶变换(FFT)的方法、系统和程序存储设备,其中该多维数组包含多个元素,所述多个元素最初分布在一个多节点计算机系统中,所述多节点计算机系统包含多个经由一个网络进行通信的节点,其包含:经由该网络跨越该计算机系统的多个节点分布在第一维度上分布该数组的多个元素,以便于进行第一维 FFT;在第一维度上对分布在每个节点处的数组元素上执行第一维 FFT;通过经由网络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布,在第二维度上在每个节点处重新分布经过一维 FFT-转换的元素;以及在第二维度上对在每个节点处重新分布的数组元素执行第二维 FFT,其中随机次序便

于网络的有效利用,由此实现多维 FFT。该“所有-到-所有”重新分布或者数组元素在除了在分布存储器、并行巨型计算机上的多维 FFT 之外的应用中被进一步更高效地实现。



1. 一种用于实现一个多维数组的多维快速傅里叶变换 (FFT) 的方法, 其中该多维数组包含多个元素, 所述多个元素最初分布在一个多节点计算机系统中, 所述多节点计算机系统包含多个经由一个网络进行通信的节点, 该方法包含:

(a) 经由该网络跨越该计算机系统的多个节点在第一维度上分布该数组的多个元素, 以便于进行第一一维 FFT;

(b) 在第一维度上对分布在每个节点处的数组元素执行第一一维 FFT;

(c) 通过经由网络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布, 在第二维度上在每个节点处重新分布经过一维 FFT 转换的元素; 以及

(d) 在第二维度上对在每个节点处重新分布的数组元素执行第二一维 FFT, 其中, 随机次序便于网络的有效利用, 由此实现多维 FFT。

2. 如权利要求 1 所述用于实现一个多维 FFT 的方法, 其特征在于: 该方法进一步包含步骤:

通过经由网络跨越计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布, 在第三维度上在每一节点处重新分布数组元素;

在第三维度上对在每个节点处重新分布的数组元素执行一维 FFT; 以及

重复跨越节点以随机次序重新分布数组元素的步骤和为后续维度对在每一节点处重新分布的元素执行一维 FFT 的步骤。

3. 如权利要求 1 所述用于实现一个多维 FFT 的方法, 其特征在于: 该方法包含一个产生其它节点的一个随机次序以便用于在每一节点处重新分布经过一维 FFT 转换的元素的步骤。

4. 如权利要求 3 所述用于实现一个多维 FFT 的方法, 其特征在于: 多个元素中的每一个经由多个总计分组在计算机系统的节点之间重新分布。

5. 如权利要求 4 所述用于实现一个多维 FFT 的方法, 其特征在于: 该方法进一步包含步骤:

在每一节点处提供多个输出队列;

以产生的随机次序对其它节点迭代多次；以及

在每一次迭代期间，向一个用于每一其它节点的一个输出队列输出这多个总计分组中的至少一个分组。

6. 如权利要求 5 所述用于实现一个多维 FFT 的方法，其特征在于：

5 该方法进一步包含步骤：

提供多个注入先进先出 (FIFO) 缓存器，每个 FIFO 缓存器用于在网络上在至少一个特定方向中传输分组；

对一个节点处的多个输出队列进行迭代以标识在每个队列标题的一个分组；

10 获得与在每个队列标题的分组相关联的可能路由方向；以及

以一个与该分组相关联的可能路由方向，把分组从每个队列的标题移动到一个最不充满的 FIFO 缓存器中。

7. 一种用于实现一个多维数组的多维快速傅里叶变换 (FFT) 的系统，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该系统包含：

(a) 用于经由该网络跨越该计算机系统的多个节点在第一维度上分布该数组的多个元素、以便于进行第一一维 FFT 的装置；

20 (b) 用于在第一维度上对分布在每个节点处的数组元素执行第一一维 FFT 的装置；

(c) 用于通过经由网络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布、在第二维度上在每个节点处重新分布经过一维 FFT -转换的元素的装置；以及

25 (d) 用于在第二维度上对在每个节点处重新分布的数组元素执行第二一维 FFT 的装置，其中，随机次序便于网络的有效利用，由此实现多维 FFT。

8. 如权利要求 7 所述用于实现一个多维 FFT 的系统，其特征在于该系统进一步包含：

30 用于通过经由网络跨越计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布，在第三维度上在每一节点处重新分布数组元素的装置；

用于在第三维度上对在每个节点处重新分布的数组元素执行一维

FFT 的装置；以及

用于重复跨越节点以随机次序重新分布数组元素和为后续维度对在每个节点处重新分布的元素执行一维 FFT 的装置。

5 9. 如权利要求 7 所述用于实现一个多维 FFT 的系统，其特征在于：
该系统包含一个用于产生其它节点的一个随机次序以便用于在每一节点处重新分布经过一维 FFT 转换的元素的装置。

10. 如权利要求 9 所述用于实现一个多维 FFT 的系统，其特征在于：多个元素中的每一个经由多个总计分组在计算机系统的节点之间重新分布。

10 11. 如权利要求 10 所述用于实现一个多维 FFT 的系统，其特征在于该系统进一步包含：

用于在每一节点处提供多个输出队列的装置；

用于以产生的随机次序对其它节点迭代多次的装置；以及

15 用于在每一次迭代期间，向一个用于每一其它节点的一个输出队列输出这多个总计分组中的至少一个分组的装置。

12. 如权利要求 11 所述用于实现一个多维 FFT 的系统，其特征在于该系统进一步包含：

用于提供多个注入先进先出 (FIFO) 缓存器的装置，其中每个 FIFO 缓存器用于在网络上在至少一个特定方向中传输分组；

20 用于在一个节点处的多个输出队列上进行迭代以标识在每个队列标题的一个分组的装置；

用于获得与在每个队列标题的分组相关联的可能路由方向的装置；以及

25 用于以一个与该分组相关联的可能路由方向，把分组从每个队列的标题移动到一个最不充满的 FIFO 缓存器中的装置。

13. 一种用于重新分布一个多维数组的方法，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该方法包含通过经由网络跨越该计算机系统的其它节点以随机次序进行的
30 “所有-到-所有”分布在每个节点处重新分布这些元素，其中该随机次序便于网络的有效利用。

14. 如权利要求 13 所述用于重新分布一个多维数组的方法，其特

征在于：该方法包含一个产生其它节点的一个随机次序以便用于在每一节点处重新分布这些元素的步骤。

5 15. 如权利要求 14 所述用于重新分布一个多维数组的方法，其特征在于：多个元素中的每一个经由多个总计分组在计算机系统的节点之间重新分布。

16. 如权利要求 15 所述用于重新分布一个多维数组的方法，其特征在于该方法进一步包含步骤：

在每一节点处提供多个输出队列；

以产生的随机次序对其它节点进行迭代多次；以及

10 在每一次迭代期间，向一个用于每一个其它节点的一个输出队列输出这多个总计分组中的至少一个分组。

17. 如权利要求 16 所述用于重新分布一个多维数组的方法，其特征在于该方法进一步包含步骤：

15 提供多个注入先进先出 (FIFO) 缓存器，每个 FIFO 缓存器用于在网络上在至少一个特定方向中传输分组；

通过在一个节点处的多个输出队列上进行迭代以标识在每个队列标题的一个分组；

获得与在每个队列标题的分组相关联的可能路由方向；以及

20 以一个与该分组相关联的可能路由方向，把分组从每个队列的标题移动到一个最不充满的 FIFO 缓存器中。

18. 一种用于重新分布一个多维数组的系统，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该系统包含：一个用于通过经由网络跨越该计算机系统的其它节点的、以随机
25 次序进行的“所有-到-所有”分布、在每个节点处重新分布这些元素的装置，其中该随机次序便于网络的有效利用。

19. 如权利要求 18 所述用于重新分布一个多维数组的系统，其特征在于：该系统包含一个用于产生其它节点的一个随机次序以便用于在每一节点处重新分布这些元素的装置。

30 20. 如权利要求 19 所述用于重新分布一个多维数组的系统，其特征在于：多个元素中的每一个经由多个总计分组在计算机系统的节点之间重新分布。

21. 如权利要求 20 所述用于重新分布一个多维数组的系统, 其特征在于该系统进一步包含:

用于在每一节点处提供多个输出队列的装置;

用于以产生的随机次序对其它节点迭代多次的装置; 以及

5 用于在每一次迭代期间, 向一个用于每一个其它节点的一个输出队列输出这多个总计分组中的至少一个分组的装置。

22. 如权利要求 21 所述用于重新分布一个多维数组的系统, 其特征在于该系统进一步包含:

10 用于提供多个注入先进先出(FIFO)缓存器的装置, 其中每个 FIFO 缓存器用于在网络上在至少一个特定方向中传输分组;

用于对在一个节点处的多个输出队列进行迭代以标识在每个队列标题的一个分组的装置;

用于获得与在每个队列标题的分组相关联的可能路由方向的装置; 以及

15 以一个与该分组相关联的可能路由方向, 把分组从每个队列的标题移动到一个最不充满的 FIFO 缓存器中。

在分布式存储器上实现
多维快速傅里叶变换的方法和系统

5 交叉引用

本发明要求享受于 2001 年 2 月 24 号提出的、标题为 MASSIVELY PARALLEL SUPERCOMPUTER 的共同拥有的、待决美国临时专利申请 60 / 271,124 的利益，其全部内容和公开就好像在此被充分阐述的那样通过引用被明确地包含在此。这个专利申请另外涉及以下在同一日期提出
10 出的、共同拥有的待决美国专利申请，其中这些申请中每一个的全部内容和公开就好像在此充分阐述的那样通过引用被明确地包含在此。美国专利申请 (YOR920020027US1、YOR920020044US1 (15270))，
“Class Networking Routing”；美国专利申请 (YOR920020028US1 (15271))，“A Global Tree Network for Computing Structures”；
15 美国专利申请 (YOR920020029US1 (15272))，“Global Interrupt and Barrier Networks”；美国专利申请 (YOR920020030US1 (15273))，
“Optimized Scalable Network Switch”；美国专利申请 (YOR920020031US1、YOR920020032US1 (15258))，“Arithmetic Functions in Torus and Tree Networks”；美国专利申请
20 (YOR920020033US1、YOR920020034US1 (15259))，“Data Capture Technique for High Speed Signaling”；美国专利申请 (YOR920020035US1 (15260))，“Managing Coherence Via Put / Get Windows”；美国专利申请 (YOR920020036US1、YOR920020037US1 (15261))，“Low Latency Memory Access And Synchronization”；
25 美国专利申请 (YOR920020038US1 (15276))，“Twin-Tailed Fail - Over for Fileservers Maintaining Full Performance in the Presence of Failure”；美国专利申请 (YOR920020039US1 (15277))，
“Fault Isolation Through No - Overhead Link Level Checksums”；美国专利申请 (YOR920020040US1 (15278))，“Ethernet Addressing Via Physical Location for Massively Parallel
30 Systems”；美国专利申请 (YOR920020041US1 (15274))，“Fault Tolerance in a Supercomputer Through Dynamic

Repartitioning”；美国专利申请（YOR920020042US1（15279））、
“Checkpointing Filesystem”；美国专利申请（YOR920020043US1
（15262））、“Efficient Implementation of Multidimensional
Fast Fourier Transform on a Distributed - Memory Parallel
5 Multi Node Computer”；美国专利申请（YOR920010211US2（15275））、
“A Novel Massively Parallel Supercomputer”；以及美国专利
申请（YOR920020045US1（15263））、“Smart Fan Modules and
System”。

技术领域

10 本发明因为例如应用于在科学、数学、工程等等领域中的计算，
所以通常涉及一个分布式存储器、信息传递并行多节点计算机和相关的
系统软件的领域。尤其是，本发明的目的在于一个用于在分布式存
储器、并行巨型计算机上高效地实现一个多维快速傅里叶变换（即，
“FFT”）的系统和方法。

15 背景技术

线性转换，诸如傅里叶变换（即，“FT”）已经被广泛地用于解
决在科学、数学、工程等等领域问题的范围。FT 把一个给定问题改变
成为一个可以被更容易地解决的问题，而且 FT 在许多不同的应用中使
用。例如对于一个 N 个变量的系统，FT 实质上表示该 N 个变量从坐标
20 空间到动量空间的变化，其中每个变量的新值取决于所有老的变量的
值。这样一个 N 个变量的系统通常作为一个 N 个元素的数组存储在一
台计算机上。FT 通常使用快速傅里叶变换（即，“FFT”）进行计算。
该 FFT 在许多标准课本中进行了描述，诸如出版社出版的数值方法等
（剑桥大学出版社在 1986, 1992 年出版，ISBN：0 - 521 - 43064 -
25 X，由 W. H. Press、S. A. Teukolsky、W. A. Vetterling 和 Brian
P Flannery 所著的“Fortran 数值方法”中的第 490 - 529 页）。大
部分计算机生产商提供了库函数调用来为它们的特定的处理器优化
FFT。例如，在工程和科学子程序库中，在 IBM 的 RS / 6000 处理器上
对 FFT 进行了充分的优化。这些库例行程序需要执行该 FFT 所必须的
30 数据（即，上述元素）被驻留在一个节点的本地存储器中。

在一个多维 FFT 中，一个多维数组的 N 个元素被分布在跨越一个
分布式存储器、并行多节点计算机的节点的多个维度中。在分布存储

器、并行多节点计算机上执行的许多应用把它们执行时间的一大部分花费在计算该多维 FFT 上。因为该分布存储器、并行多节点计算机的一个动机就是更快的执行，所以对该分布数组多维 FFT 的快速计算具有关键的重要性。该数组的 N 个元素最初以某种对于一个应用所特有的任意方式跨越节点分布。为了计算该多维 FFT，则重新分布该元素数组以便在每个节点上的数组的一部分都包含在该 x 维度中元素的一个完整行。然后执行在每个节点上、在该 X 维度中每个行上的一个一维 FFT。因为该行对于一个节点来说是本地的，以及因为在每个行上的每个一维 FFT 与其它无关，在每个节点上执行的一维 FFT 不须要和任何其它节点进行通信并且可以使用上述的库例行程序执行。在一维 FFT 之后，重新分布数组元素以便在每个节点上的、该数组的一部分包含在 y 维度中的一个完整行。此后，执行在每个节点上、在该 y 维度中每个行上的一个一维 FFT。如果该数组有超过两个的维度，则为每个超过该 x 维度和 y 维度的连续数组维度重复该重新分布和一维 FFT。该产生的数组可以重新分布成为为该应用所特有的某个任意方式。

依次处理 x 维度和 y 维度对于多维 FFT 来说不是很重要的。代之以，该数组的维度可以以任何次序进行处理。对于某些应用或者某些计算机来说，某种次序可以利用某些性能的优点而且因此具有一个比其它次序更快的执行速度。例如，数组以某种为该应用所特有的任意方式跨越该节点的初始分布，可以符合在 y 维度中为一维 FFT 所必须分布。在这种情况下，让多维 FFT 在处理 x 维度和任何其它的剩余维度之前首先处理 y 维度可能是更快的。

在如上所述的多维 FFT 的实现中，在该一维 FFT 之间的数组的每次重新分布都是一个“所有-到-所有”通信或者重新分布的一个示例。在所有-到-所有重新分布中，该分布存储器、并行多节点计算机中的每个节点使用多个分组把唯一的数据（即，数组中的元素）发送到所有其它的节点。如上所述，在分布存储器、并行多节点计算机上多维 FFT 的快速计算具有关键性的意义。在如上所述的实现中通常执行时间的一大部分被花费在跨越该分布存储器、并行多节点计算机的节点的数组重新分布上。尤其是，执行时间的一大部分被花费在跨越该分布存储器、并行多节点计算机中的节点的“所有-到-所有”数组元素的重新分布上。

因此在本技术领域有一个提供一个用于在该分布式存储器、并行巨型计算机上高效地实现多维 FFT 的系统和方法的需要。特别地，在本技术领域有一个用于提供一个用于在该分布存储器、并行巨型计算机上高效地实现“所有-到-所有”重新分配来高效地实现多维 FFT 的系统和方法的需要。

发明内容

因此本发明的一个目的是提供一个用于在一个分布存储器、并行巨型计算机上分布的一个数组上高效地实现多维 FFT 的系统和方法。

本发明的另一个目的是提供一个用于通过在该分布存储器、并行巨型计算机上高效地实现“所有-到-所有”重新分布，来在该数组上高效地实现多维 FFT 的系统和方法。

本发明的还有另一个目的是提供一个用于在不同于在分布存储器、并行巨型计算机上的多维 FFT 的应用中高效地实现“所有-到-所有”重新分布的系统和方法。

依据本发明的一个实施例，提供了一种用于实现一个多维数组的多维快速傅里叶变换 (FFT) 的方法，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该方法包含：经由该网络跨越该计算机系统的多个节点在第一维度上分布该数组的多个元素，以便于进行第一一维 FFT；在第一维度上对分布在每个节点处的数组元素执行第一一维 FFT；通过经由网络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布，在第二维度上在每个节点处重新分布经过一维 FFT-转换的元素；以及在第二维度上对在每个节点处重新分布的数组元素执行第二一维 FFT，其中，随机次序便于网络的有效利用，由此实现多维 FFT。

依据本发明的另一个实施例，提供了一种用于实现一个多维数组的多维快速傅里叶变换 (FFT) 的系统，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该系统包含：用于经由该网络跨越该计算机系统的多个节点在第一维度上分布该数组的多个元素、以便于进行第一一维 FFT 的装置；用于在第一维度上对分布在每个节点处的数组元素执行第一一维 FFT 的装置；用于通过经由网

络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分布、在第二维度上在每个节点处重新分布经过一维FFT-转换的元素的装置；以及用于在第二维度上对在每个节点处重新分布的数组元素执行第二一维FFT的装置，其中，随机次序便于网络的有效利用，
5 由此实现多维FFT。

依据本发明的还有一个实施例，提供了一种可触知地包含指令程序的程序存储设备，其中的指令程序可由一个机器执行来执行一种用于实现一个多维数组的多维快速傅里叶变换(FFT)的方法，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机
10 系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该方法包含：经由该网络跨越该计算机系统的多个节点在第一维度上分布该数组的多个元素，以便于进行第一一维FFT；在第一维度上对分布在每个节点处的数组元素执行第一一维FFT；通过经由网络跨越该计算机系统的其它节点的、以随机次序进行的“所有-到-所有”分
15 布，在第二维度上在每个节点处重新分布经过一维FFT-转换的元素；以及在第二维度上对在每个节点处重新分布的数组元素执行第二一维FFT，其中，随机次序便于网络的有效利用，由此实现多维FFT。

依据本发明的一个进一步实施例，提供了一种用于重新分布一个多维数组的方法，其中该多维数组包含多个元素，所述多个元素最初
20 分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该方法包含通过经由网络跨越该计算机系统的其它节点以随机次序进行的“所有-到-所有”分布在每个节点重新分布这些元素，其中该随机次序便于网络的有效利用。

依据本发明的还有一个进一步实施例，提供了一种用于高效地重新分布一个多维数组的系统，其中该多维数组包含多个元素，所述多
25 个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该系统包含：一个用于通过经由网络跨越该计算机系统的其它节点以随机次序进行的“所有-到-所有”分布在每个节点处重新分布这些元素的装置，其中该随机次序
30 便于网络的有效利用。

依据本发明的还有一个进一步实施例，提供了一种可触知地包含一个指令程序的程序存储设备，该指令程序可由一个机器执行来执行

一种用于高效地重新分布一个多维数组的方法，其中该多维数组包含多个元素，所述多个元素最初分布在一个多节点计算机系统中，所述多节点计算机系统包含多个经由一个网络进行通信的节点，该方法包含通过经由网络跨越该计算机系统的其它节点以随机次序进行的“所有-到-所有”分布在每个节点处重新分布这些元素，其中该随机次序便于网络的有效利用。

附图说明

鉴于以下结合附图进行的详细说明，本发明的目的，特征以及优点对于本领域技术人员将变得明显，其中附图包含：

10 图 1 说明了依据本发明的一个示例分布存储器、并行巨型计算机，该并行巨型计算机包含 9 个经由一个使用一个 2 维 3x3Torus 网络的多维网格互连的节点；

图 2 说明了依据本发明，一个来自图 1 中的分布存储器、并行巨型计算机的示例节点的更详细表示；

15 图 3 说明了依据本发明的一个示例二维 9 行 9 列数组，其可以被高效地实现用于多维 FFT；

图 4 说明了依据本发明，一个跨越在图 1 中的巨型计算机的节点的、图 3 中的二维数组的示例分布；

20 图 5 说明了依据本发明，一个跨越在图 1 中的巨型计算机的节点分布的二维数组的一个示例第一一维 FFT；

图 6 说明了依据本发明，一个在图 5 中的第一一维 FFT 之后产生的二维数组的一个示例重新分布；

图 7 说明了一个依据本发明，图 6 中的重新分布数组的示例第二一维 FFT；

25 图 8 说明了一个依据本发明，描述在图 4 - 7 中说明的二维 FFT 的实现的示例方法流程图；

图 9 说明了一个依据本发明，描述用为在该分布存储器、并行巨型计算机上的其它节点指定的分组填充在该示例节点上的输出队列的示例方法流程图；以及

30 图 10 说明了一个依据本发明，描述在该示例节点的输出队列中的分组为了在该 Torus 网络 100 上的后续插入，如何被流入 FIFO 中的示例方法流程图。

具体实施方式

本发明的目的是一个用于在分布存储器、并行巨型计算机上高效地实现多维快速傅里叶变换（即，“FFT”）的系统和方法。尤其是，本发明实现了分布在该分布存储器、并行巨型计算机的节点处的元素的一个高效“所有-到-所有”重新分布以得到多维 FFT 的一个高效实现。

依据本发明，FFT 作为一系列一维转换在该分布存储器、并行巨型计算机上实现，其需要一个或多个跨越该分布存储器、并行巨型计算机中的节点的一个多维数组的“所有-到-所有”重新分布。该分布式存储器、并行巨型计算机利用一个基于 Torus 的网络用于互连该巨型计算机以及在该巨型计算机的节点之间进行通信。如将在下面描述的那样，每个节点实现了一个硬件路由器，用于跨越经由基于 Torus 的网络互连的该巨型计算机节点，高效地路由包含该数组元素的分组。因此依据本发明，本发明把作为该多维数组的一系列一维转换的多维 FFT 的实现和上述硬件路由连接起来，以获得高效的 FFT 实现。

进一步依据本发明，该分布存储器、并行巨型计算机包含多个节点，每个节点都包含至少一个在一个本地存储器上进行操作的处理器。该节点作为一个多维网格进行互连而且它们经由网格链接进行通信。不用损失普遍性以及为了让这个发明描述可为本领域的一个技术人员容易地理解，巨型计算机的多维节点网格将被描述为一个示例 2 维网格。尽管在下面描述中仅仅描述了 2 维节点网格的事实，但是可以考虑到在本发明的范围之内，基于本发明的示教可以容易地提供其它维度的节点网格。要注意到：该分布存储器、并行巨型计算机可能使用一个 3 维或者更大的基于 Torus 的体系结构。另外，不用损失普遍性以及为了让这个发明描述可为本领域的一个技术人员容易地理解，由该多维 FFT 使用的多维数组将被描述为一个示例 2 维数组。尽管有在下面描述中仅仅描述了 2 维数组的事实，但是可以考虑到在本发明的范围之内，基于本发明的示教可以容易地提供附加维度的数组。要进一步注意到：在基于 Torus 的体系结构的维度数目和在数组中的维度数目之间没有对应关系。该数组必须具有足够的大小以便它能够跨越该巨型计算机的节点或者一个节点子集分布，用于依据本发明实现多维 FFT。

图 1 是依据本发明的一个分布存储器、并行巨型计算机的示例说明，该并行巨型计算机包含 9 个经由一个使用一个 2 维 3x3Torus 网络 100 的多维网格互连的节点。要注意到为了简单和清楚起见，在示例方式中的节点数目局限于 9 个节点，以及取决于该分布存储器、并行巨型计算机的特定体系结构需求，节点的数目可以显著地变化。图 1 描述了标记为 Q11 - Q33 的 9 个节点，其中的一对通过一个网格链接互连。总计，9 个节点 Torus 网络 100 由 18 个网格链接互连，其中每个节点经由一个相应的网格链接直接互连到在该 Torus 网络 100 中的四个其它节点。要注意到不同于一个网格，示例 2 维 Torus 网络 100 不包含边沿节点。例如，节点 Q11 经由网格链接 102 互连到节点 Q31；经由网格链接 104 互连到节点 Q13；经由网格链接 106 互连到节点 Q21；以及经由网格链接 108 最后互连到节点 Q12。作为另一个示例，节点 Q22 经由网格链接 110 互连到节点 Q12；经由网格链接 112 互连到节点 Q21；经由网格链接 114 互连到节点 Q32 以及经由网格链接 116 最后互连到节点 Q23。其它的节点以类似的方式进行互连。

进一步参考图 1，在节点之间通信的数据（即，数组中的元素）在一个或多个分组中在该网络上进行传输。对于在一对节点之间的任何给定通信，如果要被通信的数据数量超过了由该 Torus 网络 100 支持的分组大小，则需要多个分组。一个分组包含一个分组标题和由该分组携带的数据。该分组标题包含由 Torus 网络 100 所需要的从一个源节点传输分组到一个目的节点的信息。在当前专利申请中的分布存储器、并行巨型计算机中，在该网络上的每个节点由一个逻辑地址标识而且该分组标题包含一个目的地址，以便该分组被自动地路由到在该网络上由一个目标标识的节点。

图 2 是依据本发明，来自图 1 中的分布存储器、并行巨型计算机的一个示例节点，例如节点 Q11 的一个更详细表示 200。节点 Q11 包含至少一个在本地存储器 204 上进行操作的处理器 202。该节点进一步包含一个路由器 206，其路由，即发送和接收，在网格链接 102、104、106 和 108 上的分组，这些网格链接分别把节点 Q11 连接到它的邻居节点 Q31、Q13、Q21 和 Q12，如同在图 1 中特别地说明的那样。还有进一步，该节点包含一个接收缓存器 208 用于缓存由路由器 206 接收的分组，这些分组的目的地为本地处理器 202。本地处理器 202 可以容

易地定期轮询接收缓存器 208 以便确定在该接收缓存器中是否有分组，并且然后检索在该接收缓存器 208 中缓存的分组。取决于一个特定的应用和分组，本地处理器 202 可以写入分组的内容到存储器 204 中。

- 5 进一步参考图 2，节点 Q11 包含四个特别被标记为 X+、X-、Y+和 Y- 的注入先进先出（即，“FIFO”）缓存器 810。

处理器把输出分组放置到本地存储器 2104 的一个或多个输出队列 212 中，这些输出队列 212 存储目的地为其它节点的分组直到它们能够被放置到注入 FIFO 210 中为止。当注入 FIFO 不是充满时，处理器把
10 输出分组放置到该注入 FIFO 210 中。依据到达一个注入 FIFO 210 标题的一个特定分组，由路由器 206 把该分组从注入 FIFO 210 中删除，并且该路由器 206 把该分组插入到一个朝向一个用于该特定分组的
目的节点的网格链接 102、104、106 和 108 上。该四个注入 FIFO 210 由路由器 206 以及本地处理器 202 的硬件相等地处理。

- 15 还进一步参考图 2，路由器 206 包含几个同时路由的特点。该路由首先表示虚拟的开通式路径选择。例如，如果一个在一个网格链接上的
进来分组的目的地不是节点 Q11 的本地处理机 202，则路由器 206 把该分组转发到外出网格链接 102、104、106 和 108 中的一个上。路
由器 206 执行该转发而不涉及本地处理器 202。该路由进一步表示最短
20 路径路由选择。例如，一个由节点 Q11 发送到节点 Q13 的分组（参见
图 1 和 8）经由表示一个最短路径路由的网格链接 104 进行传播。任何
其它通路将是相对比较长的。作为另一个示例，由节点 Q11 发送到节
点 Q22 的分组可以经由网格链接 106 和 112 进行传播，或者作为替换
经由网格链接 108 和 110 进行传播。这类路由表示一个自适应类型的
25 路由选择。因此，在这里可以有一个网格链接的选择，通过这个网格
链接选择，经由基于 Torus 的网络 100，一个分组可以在传输中离开
一个节点前往另一个节点。在先前的示例中，分组可以经由网格链接
106 或者 108 离开节点 Q11。自适应路由选择允许路由器 206 对一个分
组选择较不忙的外出网格链接或者基于某些其它的条件选择外出网格
30 链接。要注意到：自适应路由选择不仅仅在一个分组的源节点，例如
节点 Q11 处执行，而是在一个分组经由图 1 中基于 Torus 的网络 100
去往该分组的目的地节点的路上所通过的每个中间节点处执行。在下面

参考图 9 和 10 的描述特别地描述了本发明如何执行分组经由 Torus 网络 100，跨越巨型计算机的节点的上述路由。

图 3 是一个依据本发明的包含 81 个元素的示例二维 9 行乘 9 列数组 300，其可以被高效地实现用于该多维 FFT。要注意到：示例二维数组 300 被容易地扩展到其它包含一个不同的行和列数的二维数组（例如，10 行乘 11 列的二维数组），其可以依据本发明被使用用于在分布存储器、并行巨型计算机上实现 FFT。在数组 300 中，数组的第一行包含元素 A11、A12... A19，而数组的第一列包含元素 A11、A21... A91。

图 4 是一个依据本发明，图 3 中的二维数组 300 如何跨越在图 1 中的节点 Q11 - Q33 分布的示例分布说明 400。要注意到：该数组最初以某种对于一个应用所特有的任意方式跨越节点分布。依据本发明，该数组被重新分布以便在每个节点 Q11... Q33 上的一部分数组都包含在图 4 中说明的分布。这个重新分布类似于如下参考图 5 和 6 描述的那样。如在该分布说明 400 中特别描述的那样，图 1 中的每个节点都包含图 3 中的二维数组 300 的一部分。例如，节点 Q11 包含数组 300 的第一行，即，元素 A11、A12... A19。作为另一个示例，节点 Q12 包含数组 300 的第二行，即元素 A21、A22... A23。要注意到，如在图 4 的分布说明 400 中特别描述的那样，图 1 中的其它节点 Q13 - Q33 包含数组 300 的相应行 3 到 9。在图 4 的示例分布中，一个特定节点到一个数组元素特定行的分配不是重要的。相反，要注意到任何分配都是可行的。对于各种应用和 / 或计算机，某些分配可以利用由应用和 / 或计算机提供的性能的优点并且因此产生比其它分配更快的执行。例如，执行多维 FFT 最快速的方法可能是把节点 Q1 和 Q12 的分配由在图 4 中说明的那些颠倒过来。

图 5 是一个描述了在图 4 的二维数组上的一个第一一维 FFT 的示例说明 500，这些二维数组经由图 1 中的二维 Torus 网络 100 跨越节点 Q11 - Q33 进行分布。如上面特别提及的那样，依据本发明的多维 FFT 通过执行一系列的一维 FFT 来完成。因此依据本发明，二维数组 300 的多维 FFT 可以被实现为一系列的一维 FFT。因此，在分布在每个节点处的每个元素行上执行一个一维的 FFT。例如，为分布在节点 Q11 处的元素，即在被分给节点 Q11 的第一行数组 300 中的元素执行一个一维 FFT。为在每个节点 Q12 - Q33 处的元素（即，元素行）执行一

5 维的 FFT。该结果是一个由第一一维 FFT 转换的元素数组。更详细地，在每个节点处在每一行上的一维 FFT 的结果是一行和在图 5 中特别说明的具有同样长度的一行。例如，在图 4 中的节点 Q11 处第一行，其包含元素 A11、A12... A19，上执行的一个一维 FFT 产生在图 5 的节点 Q11 处的一个第一行，其包含元素 B11、B12 ... B19。此外，在每一节点处在每一行上执行的一维 FFT 与在另一个节点处在任何其它行上执行的一维 FFT 无关。在图 4 中说明的数据特定分布允许每一节点执行分布在那个节点处的元素行上的一维 FFT，而不和在图 1 的 Torus 网络 100 上的任何其它节点进行通信。因此，因为不需要在节点之间的通信，这些一维 FFT 被快速地执行。要注意到：在每个节点除了产生在图 5 中的行之外，在图 4 中的原有行可以继续存在以及为一个特定的应用所感兴趣，但是如在图 6 和 7 中详细说明的那样，原有的行不再为在一系列为依据本发明的多维 FFT 所需要的 FFT 中的第二一维 FFT 所需要。

15 图 6 是一个依据本发明的示例“所有-到-所有”重新分布说明 600，其描述了每一个经由图 5 中的第一维度 FFT 转换产生的元素行为执行第二维度 FFT 如何跨越节点 Q11 - Q33 重新分布。更详细地，如图 6 中详细地描述的那样，每一个分布在图 5 的每一节点 Q11... Q33 处产生的元素行经由该 Torus 网络 100 重新分布以便每一个连续的节点接收一个连续的元素列。这个高效的重新分布是“所有-到-所有”重新分布，其允许依据本发明的、在分布存储器、并行巨型计算机上的多维 FFT 的一个高效实现。例如，第一节点 Q11 接收第一元素列，即来自每一节点 Q11... Q33 的第一元素。作为另一个示例，节点 Q12 接收第二元素列，即来自每一节点 Q11... Q33 的第二元素。为在图 5 中的每一列执行这个重新分布。在图 6 的示例重新分布中，一个特定节点到一个数组元素特定行的分配不是重要的。相反，要注意到任何分配都是可行的。对于各种应用和 / 或计算机，某些分配可以利用由应用和 / 或计算机提供的性能的优点并且因此产生比其它分配更快的执行。例如，执行多维 FFT 最快速的方法可能是把节点 Q11 和 Q12 的分配由在图 6 中说明的那些颠倒过来。在下面参考图 9 和 10 的描述特别地描述了本发明如何执行数组元素经由 Torus 网络 100，跨越巨型计算机的节点的“所有-到-所有”重新分布。在每一节点 Q11... Q33

处的元素的“所有-到-所有”重新分布是快速的，这是因为它利用了 Torus 网络 100 的通信特征的优点。在图 6 中说明的重新分布中，每一个来自 Q11... Q33 的节点发送单个数组元素到所有其他节点。以下描述假定该数组的每一个元素是一些比由单个分组携带的数据数量要大的数据量。因此，需要多个分组来经由 Torus 网络 100 把数组的每一元素传送到一个目的节点。这非常类似于典型的现实的重新分布，其中由于非常大的数组尺寸，每一节点通常需要许多分组来发送许多数组元素到所有其他节点。

图 7 是一个依据本发明描述了在图 6 的二维数组上的一个第二一维 FFT 的示例说明 700，这些二维数组经由图 1 中的二维 Torus 网络 100 跨越节点 Q11 - Q33 进行重新分布。如上面详细地指出的那样，依据本发明的多维 FFT 通过执行一系列一维的 FFT 来完成，其中图 7 描述了依据本发明在那个系列中的第二一维 FFT。因此，在被分配给如图 5 中说明的每一节点的元素列上执行一个一维的 FFT。例如，为分布在节点 Q11 处的元素，即作为一行分配给节点 Q11 以形成图 5 的第一列的、在图 6 中的元素 B11、B21... B91 执行一个一维 FFT。另外，在每一节点 Q12 - Q33 处的元素行上执行一维的 FFT（即，从图 5 的连续元素行中分配）。在每一行上的一维 FFT 的结果是和图 7 中特别说明的具有同样长度的行。例如，在图 6 中节点 Q11 处的第一行上执行的一个一维 FFT 产生在图 7 的在节点 Q11 处的一个第一行，其中图 6 中节点 Q11 处的第一行包含元素 B11、B21... A91，图 7 中在节点 Q11 的第一行包含元素 C11、C21... C91。如上参考第一 FFT 所述的那样，在每一节点处在每一行上执行的一维 FFT 与在另一个节点处在任何其它行上执行的一维 FFT 无关。在图 6 中说明的数据特定分布允许每一节点执行分布在那个节点处的元素行上的一维 FFT，而不用和在图 1 的 Torus 网络 100 上的任何其它节点进行通信。因此，因为不需要在节点之间的通信，这些一维 FFT 被快速地执行。

图 8 是一个示例方法流程图，其说明了在图 1 中的、利用一个 2 维 Torus 网络 100 用于在该巨型计算机的节点 Q11... Q33 之间的通信的、分布的分布存储器、并行巨型计算机上的一个数组的二维 FFT 实现。在下面的描述中，将基于图 1 - 7 为了高效地执行二维 FFT 对图 8 进行描述。在步骤 802 处，开始在图 3 中说明的一个二维数组在图 1

的分布的分布存储器、并行巨型计算机中的多维 FFT。要注意到在步骤 702 处,在图 3 中说明的数组以某种可能是一个应用所特有的任意方式跨越这些节点分布。在步骤 804 处,如在图 4 中详细说明的那样,数组 300 的元素(即,数据)被高效地跨越节点 Q11... Q33 重新分布。

5 在步骤 806 处,如在图 4 说明的那样,每一个节点在存储在那个节点处的一行数组元素上执行一个第一一维 FFT(从一系列的一维 FFT 当中),并且结果在图 5 中进行了详细的说明。如参考图 5 和 6 进行描述的那样,在步骤 808 处,一维 FFT 转换的元素列跨越使用图 1 中基于 Torus 的体系结构的巨型计算机中的节点 Q11... Q33 进行重新分布。

10 在步骤 810 处,每一个节点在一个在图 6 中说明的一个第一一维 FFT 转换了的元素的连续列上执行一个第二一维 FFT,这些元素作为在图 6 中分布为一个元素行。第二一维 FFT 的结果在图 7 中进行了说明。在步骤 812 处,结束在图 3 中说明的二维数组在图 1 的巨型计算机中的多维 FFT。如上详细所述的那样,在两个一维的 FFT 之间有一个跨越

15 节点 Q11... Q33 的快速元素重新分布。

以上描述的、在一个跨越一个分布存储器、并行巨型计算机的节点分布的元素数组上的多维 FFT 和跨越这些节点重新分配元素一起来说明本发明。更详细地,本发明利用基于 Torus 的体系结构的高效硬件路由和一系列一维 FFT 一起来实现在分布式存储器、并行巨型计算机上的多维 FFT 的一个高效实现。如上所述,依据本发明的示教可以被使用用于执行在其它数组维数数量中、在其它数组尺寸中、以及在 Torus 网络维度的其它数量,例如 3 维 Torus 中的高效多维 FFT。另外,依据本发明的示教可以被使用用于在一个任意维度的 Torus 网络上、在分布存储器、并行巨型计算机的节点之间执行“所有-到-所有”通信。

20

25

图 9 是一个示例方法流程图 900,其描述了依据本发明在该分布存储器、并行巨型计算机上用目的地为其它节点,例如节点 Q22 和 Q33 的分组填充在图 2 的一个示例节点 Q11 的一个或多个输出队列 212。在图 6 中说明的“所有-到-所有”重新分配依据本发明如下所述实现。

30 假定 Q_{xy} 表示一个具有一个 X 坐标值 x 和一个 Y 坐标值 y (例如, $x = 1$; $y = 1$) 的通用节点(例如,节点 Q11)。因此,依据“所有-到-所有”重新分布,对于所有可能的值 a 和 b ,节点 Q_{xy} (例如,节点 Q11)

需要发送多个总分组（例如， k 个分组）到每个节点 Q_{ab} （例如，在图 1 中说明的 Q_{12} 、 Q_{13} ； Q_{21} 、 Q_{22} 、 Q_{23} ；以及 Q_{31} 、 Q_{32} 、 Q_{33} ；要注意到 Q_{11} 不把分组发送到它本身）。为了尽快地执行该重新分布，必须高效地使用 Torus 网络 100 的网格链接。如果分组没有以一个高效的次序进行调度，则网格链接的使用可能是非常低效率的。例如，如果每个节点首先仅仅在正 $X+$ 方向发送分组，则所有在负 X 方向的网格链接将是空闲的，因此该重新分布不会被尽快地执行而且该多域 FFT 不会被最高效地实现。依据本发明，该快速重新分布利用基于 Torus 网络 100 的自适应路径选择能力，以便分组调度被高效地实现，如下面详细说明的那样。

因此参考图 9，有通过 Torus 网络 100 互连的 $N_x * N_y$ 节点（即，在图 1 中的 $3 * 3 = 9$ 节点）需要交换分组，这些分组包含二维数组中的元素。在步骤 902 处，开始该示例方法。在步骤 904 处，在每一节点 $Q_{11} \dots Q_{33}$ 处创建一个为在 Torus 网络 100 上的每一节点分配一个在 $0, \dots, N_x * N_y - 2$ 之间的唯一编号的数组（即，`random_map []` 数组）。因为一个节点不把分组发送给它本身，所以交换分组的节点总数是 0 到 $N_x * N_y - 2$ 。注意到在步骤 904 处的分配是随机地产生的。在这一点上假定一个节点发送该数组的一个元素到另一个节点所需要的分组总数是 k 个分组（例如，6 个分组）。此后，假定总共 k 个分组 = d 个循环 * b 个分组，其中 d 是必要的循环数目以及每次循环为 k 个分组总量传送 b 个分组。注意到 b 可以为性能所必需进行选择而且可以同样地等于 1。例如，为了传送总共 6 个分组，能够为总数 6 个分组选择在 3 个循环中的每一个上让每个循环传送 2 个分组。因此，在步骤 906 处，为 i_d 从 1 到 d 的循环启动一次循环。在步骤 908 处，一个队列计数器被初始化为零。假定有 L 个输出队列 212（ L 大于或等于 1）用于存储分组（或者分组的简短描述符以便不需要复制实际的分组），以及所有用于一个给定目的地的分组（或者分组的描述符）将被放置到相同的输出队列中。在图 9 的嵌套循环内在步骤 912 处以循环的次序选择一个特定输出队列 i_L 。在步骤 910 处，为从节点 0 到节点 $N_x * N_y - 2$ 的 i_N 值初始化一个循环，该 i_N 值作为一个到在步骤 904 处创建的数组（即，`random_array []`）的索引。因为在步骤 904 中创建的数组为一个特定 i_N 值进行索引，所以可从该随机数组中

获得一个随机节点值。在步骤 912 处，以循环次序选择一个第一队列。在步骤 914 处，为每个 d 循环、从 1 到 b 分组的 ib 初始化一个循环。随后，如步骤 914 和 916，多个目的为一个给定随机节点 i_N 的 b 个分组（例如，从上面的示例中 $b = 2$ 个分组）被添加到同分组 [node, 5 id, ib] 相同的一个输出队列 i_L 中。在步骤 918 处，一旦已经完成了所有 d 循环，则结束该方法。总而言之参考流程图 900，在一个 d 循环期间一个特定节点 “i”（例如，在图 2 的节点 Q11 上的处理器 202）将首先把包含用于目的地为一个节点模数 $(i+1, N_x * N_y - 1)$ 的数组中的一个元素的数据的 b 个数量的分组放置在一个第一输出队列中，10 然后特定节点 “i” 将把包含用于目的地为一个节点模数 $(i+2, N_x * N_y - 1)$ 的数组中的一个元素的数据的 b 个分组放置到下一个输出队列中，诸如此类直到到达节点模数 $(i + (N_x * N_y - 1), N_x * N_y - 1)$ 为止。当为一个给定循环已经把 b 个分组插入到该输出队列中时，重复这个处理过程直到 d 循环已经全部都完成了为止。上述重新分布实现了在图 1 的 Torus 网络 100 上非常高的网格链接使用，由此高效地15 实现了依据本发明的多维 FFT。

图 10 是一个示例方法流程图 1000，其描述了依据本发明，在一个或多个在图 2 中的示例节点 Q11 上的输出队列 212 中的分组如何被流入到一个注入 FIFO 210 中用于在该 Torus 网络 100 上的后续插入。在20 详细地描述图 10 之前，要注意到图 9 中的填充和图 10 中的流出可以彼此同时执行。在步骤 1002 处，开始该示例方法。在步骤 1004 处确定是否全部 L 输出队列 212 都为空。在步骤 1006 处，为 i_L 从 1 到 L 启动一个循环，以在所有 L 输出队列上从 1 至 L 进行迭代。在步骤 1008 处确定一个特定输出队列 i_L 是否是空的。如果输出队列 i_L 是空的，25 步骤 1006 处该方法继续到下一个 i_L 输出队列。否则，在步骤 1010 处，为在该输出队列 i_L 标题的一个分组，获得用于经由 Torus 网络 100 路由该分组的可能方向。例如参考图 1，假定节点 Q11 把一个目的地为节点 Q22 的分组放置到一个输出队列 i_L 中。该分组可以以 X+方向（经由网格链接 108）从节点 Q11 传播继之以 Y 方向（经由网格链接 110）以30 到达节点 Q22，或者它可以在 Y 方向（经由网格链接 106）传播继之以 X+方向（经由网格链接 112）以到达节点 Q22。现在回到图 10，在步骤 1012 处，进一步确定是否在用于分组的可能方向中的、在图 2 中的

全部 FIFO 210 是满的。如上所述，每一注入 FIFO 210 具有一个与它相关联的逻辑方向（例如，X+），其表示任何被放置到该注入 FIFO 210 中的分组能够在相关的逻辑方向（例如，X+方向）中进行移动。如果用于分组方向的注入 FIFO 210 是满的，则该方法跳过当前输出队列并且通过

5 通过在步骤 1006 处迭代到下一个输出队列继续。否则，在步骤 1014 处，该分组在一个用于那个分组的可能方向中，从该输出队列移动到一个最不充满的 FIFO 212。要注意到分组以一个循环次序从输出队列中除去，用于插入到在图 2 中说明的注入 FIFO 210 中。在该分组被移动了之后，在步骤 1008 处为一个在那个输出队列中的下一个可用分组

10 继续该方法。一旦全部的输出队列都是空的，则该方法在步骤 1016 处结束。

为了更充分地说明描述“全部-到-全部”路由的图 9 和 10，假定在图 5 中的节点 Q11 处的元素行，即元素 B11、B12... B19，将经由 Torus 网络 100 跨越在图 6 中说明的节点 Q12... Q33 进行重新分布。

15 假定节点的随机映射具有以下在 random_map 数组 = {Q32; Q22; Q13; Q21; Q23; Q33; Q12; 和 Q31} 中的值。因此，来自节点 Q11 的数组元素次序和它们的目的节点如下：{B12 到 Q12; B13 到 Q13; B14 到 Q21; B15 到 Q22; B16 到 Q23; B17 到 Q31; B18 到 Q32 以及 B 19 到 Q33}。这些数组元素被放置到节点 Q11 的 FIFO 210 中，其如下所示：{B18

20 经由 X+或者 Y-到 Q32; B15 经由 X+或者 Y+到 Q22; B13 经由 X-到 Q13; B14 经由 Y+到 Q21; B16 经由 Y+或者 X-到 Q23; B19 经由 X-或者 Y-到 Q33; B12 经由 X+到 Q12; 以及 B17 经由 Y-到 Q31}。因此例如，在节点 Q11 上 FIFO 210 可以如在下面在表格 1 中说明的那样进行填充。

表格 1

X+	X-	Y+	Y-
B18 到 Q32	B13 到 Q13	B15 到 Q22	B14 到 Q21
B12 到 Q12	B19 到 Q33	B16 到 Q23	B17 到 Q31

25

尽管注入 FIFO 的数目在上面被描述为等于网格链接到一个节点的数目（例如，4 个 FIFO 和 4 个网格链接）的事实，一个被局限于至少一个特定网格链接的注入 FIFO 的使用在当注入 FIFO 的数量不等于网格链接数目时也是非常适合的。例如，如果有比网格链接更少的注入

FIFO，则一个缓存器的使用可以被限制到几个特定网格链接中的至少一个。对于另一个示例，如果有比网格链接更多的注入 FIFO，则那里可以有几个使用被限于至少同一个特定网格链接的注入 FIFO。

- 5 虽然上面参考多维 FFT 的高效实现描述了数组重新分布的实现，该“所有-到-所有”重新分布也非常适合于经由图 1 中的 Torus 网络 100 的、任何类型的数组重新分布。

虽然已经通过参考它的最佳实施例详细地显示和描述了本发明，但是本领域的那些技术人员必须理解：在其中可以进行上述及其它形式和细节中的变化而不背离本发明的精神和范围。

10

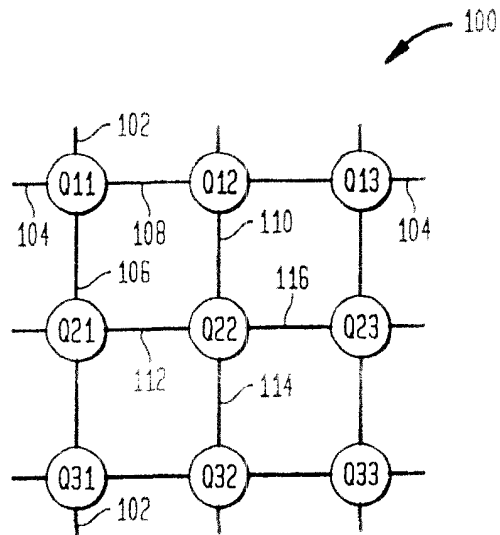


图 1

A11	A12	A13	A14	A15	A16	A17	A18	A19
A21	A22	A23	A24	A25	A26	A27	A28	A29
A31	A32	A33	A34	A35	A36	A37	A38	A39
A41	A42	A43	A44	A45	A46	A47	A48	A49
A51	A52	A53	A54	A55	A56	A57	A58	A59
A61	A62	A63	A64	A65	A66	A67	A68	A69
A71	A72	A73	A74	A75	A76	A77	A78	A79
A81	A82	A83	A84	A85	A86	A87	A88	A89
A91	A92	A93	A94	A95	A96	A97	A98	A99

图 3

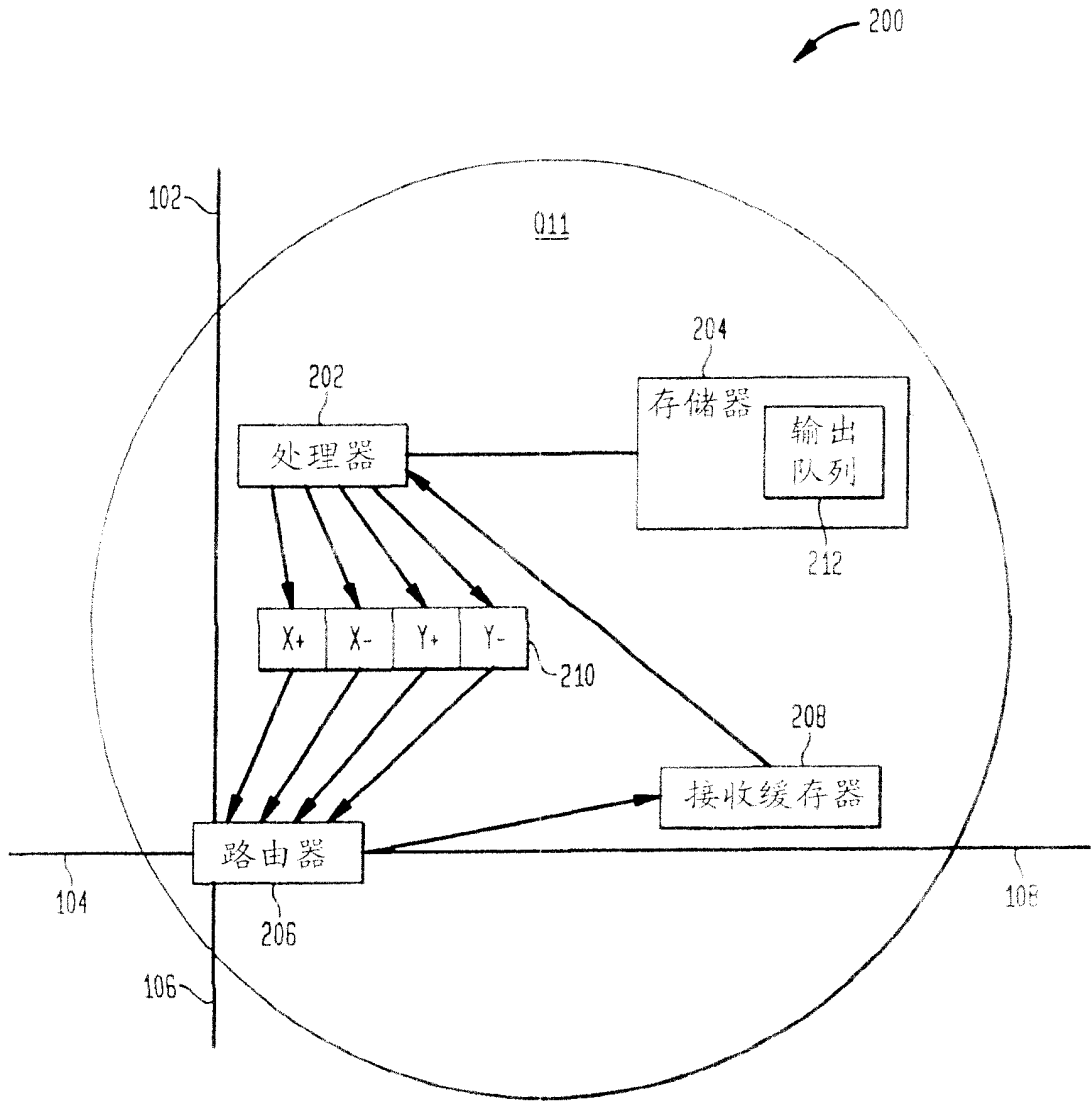


图 2

400

Q11	A11	A12	A13	A14	A15	A16	A17	A18	A19
Q12	A21	A22	A23	A24	A25	A26	A27	A28	A29
Q13	A31	A32	A33	A34	A35	A36	A37	A38	A39
Q21	A41	A42	A43	A44	A45	A46	A47	A48	A49
Q22	A51	A52	A53	A54	A55	A56	A57	A58	A59
Q23	A61	A62	A63	A64	A65	A66	A67	A68	A69
Q31	A71	A72	A73	A74	A75	A76	A77	A78	A79
Q32	A81	A82	A83	A84	A85	A86	A87	A88	A89
Q33	A91	A92	A93	A94	A95	A96	A97	A98	A99

图 4

500

Q11	B11	B12	B13	B14	B15	B16	B17	B18	B19
Q12	B21	B22	B23	B24	B25	B26	B27	B28	B29
Q13	B31	B32	B33	B34	B35	B36	B37	B38	B39
Q21	B41	B42	B43	B44	B45	B46	B47	B48	B49
Q22	B51	B52	B53	B54	B55	B56	B57	B58	B59
Q23	B61	B62	B63	B64	B65	B66	B67	B68	B69
Q31	B71	B72	B73	B74	B75	B76	B77	B78	B79
Q32	B81	B82	B83	B84	B85	B86	B87	B88	B89
Q33	B91	B92	B93	B94	B95	B96	B97	B98	B99

图 5

600

Q11	B11	B21	B31	B41	B51	B61	B71	B81	B91
Q12	B12	B22	B32	B42	B52	B62	B72	B82	B92
Q13	B13	B23	B33	B43	B53	B63	B73	B83	B93
Q21	B14	B24	B34	B44	B54	B64	B74	B84	B94
Q22	B15	B25	B35	B45	B55	B65	B75	B85	B95
Q23	B16	B26	B36	B46	B56	B66	B76	B86	B96
Q31	B17	B27	B37	B47	B57	B67	B77	B87	B97
Q32	B18	B28	B38	B48	B58	B68	B78	B88	B98
Q33	B19	B29	B39	B49	B59	B69	B79	B89	B99

图 6

700

Q11	C11	C21	C31	C41	C51	C61	C71	C81	C91
Q12	C12	C22	C32	C42	C52	C62	C72	C82	C92
Q13	C13	C23	C33	C43	C53	C63	C73	C83	C93
Q21	C14	C24	C34	C44	C54	C64	C74	C84	C94
Q22	C15	C25	C35	C45	C55	C65	C75	C85	C95
Q23	C16	C26	C36	C46	C56	C66	C76	C86	C96
Q31	C17	C27	C37	C47	C57	C67	C77	C87	C97
Q32	C18	C28	C38	C48	C58	C68	C78	C88	C98
Q33	C19	C29	C39	C49	C59	C69	C79	C89	C99

图 7

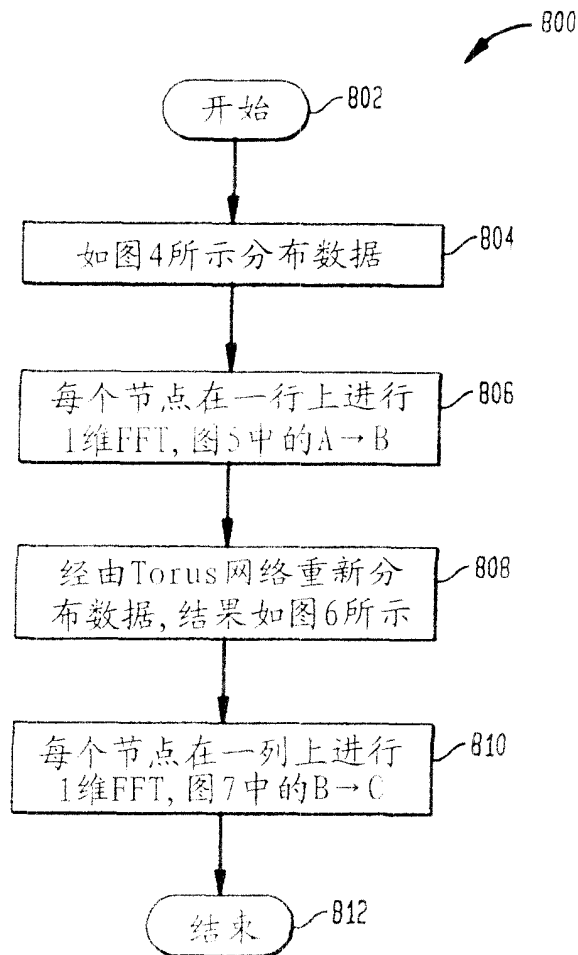


图 8

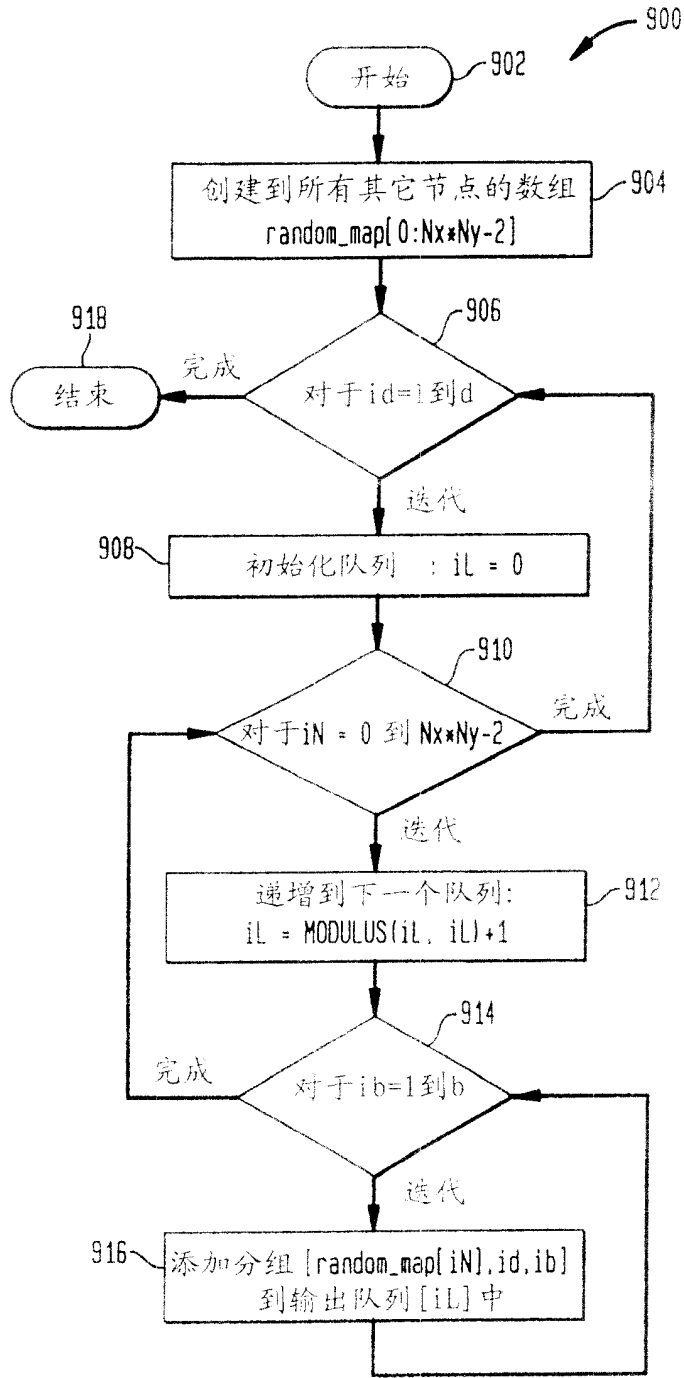


图 9

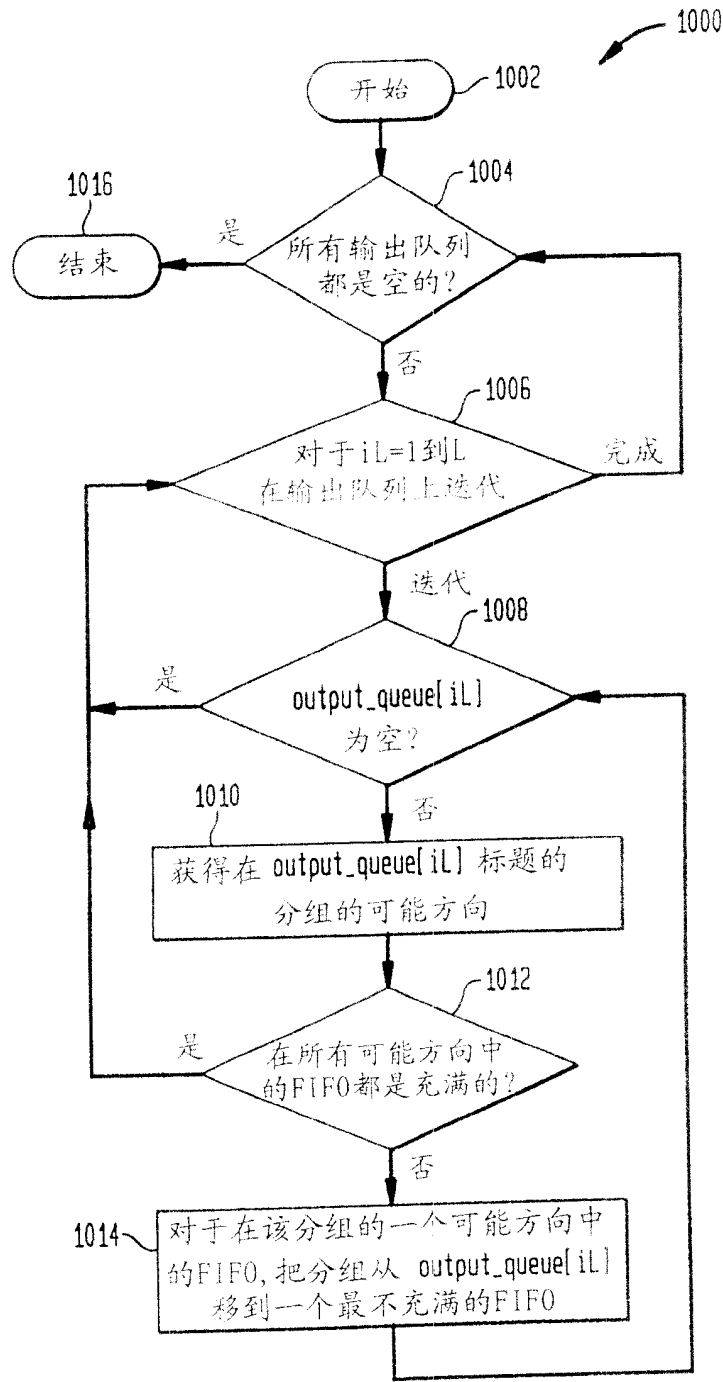


图 10