



(12) 发明专利

(10) 授权公告号 CN 112534401 B

(45) 授权公告日 2024. 07. 12

(21) 申请号 201980049799.4
 (22) 申请日 2019.05.22
 (65) 同一申请的已公布的文献号
 申请公布号 CN 112534401 A
 (43) 申请公布日 2021.03.19
 (30) 优先权数据
 15/993,284 2018.05.30 US
 (85) PCT国际申请进入国家阶段日
 2021.01.26
 (86) PCT国际申请的申请数据
 PCT/US2019/033573 2019.05.22
 (87) PCT国际申请的公布数据
 W02019/231793 EN 2019.12.05

(72) 发明人 G·A·迪基
 (74) 专利代理机构 北京林达刘知识产权代理事务所(普通合伙) 11277
 专利代理师 刘新宇
 (51) Int.Cl.
 G06F 8/41(2006.01)
 G06F 16/901(2006.01)
 (56) 对比文件
 US 2012284255 A1,2012.11.08
 US 2013290298 A1,2013.10.31
 US 2015088856 A1,2015.03.26
 US 2017083573 A1,2017.03.23
 US 2017147644 A1,2017.05.25
 审查员 王琼丽

(73) 专利权人 起元技术有限责任公司
 地址 美国马萨诸塞州

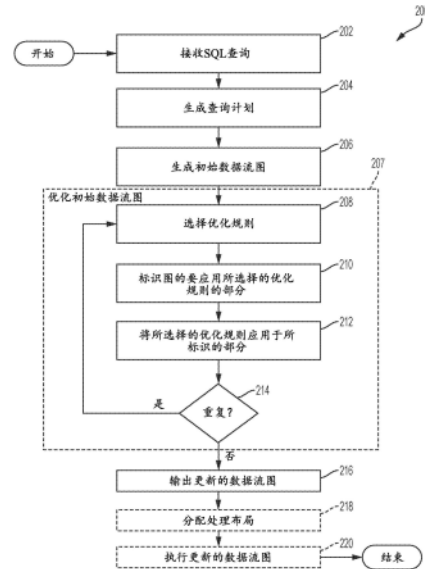
权利要求书6页 说明书24页 附图24页

(54) 发明名称

用于数据流图优化的系统和方法

(57) 摘要

一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路。



1. 一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:

获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示该第一多个节点中的节点之间的数据流的第一多个链路;

通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点;以及

至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图,

其中,生成该更新的数据流图包括:

使用数据流子图模式匹配语言表达来确定将被应用于该初始数据流图的一部分的优化规则,

其中:

该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分,以及

该数据流子图模式匹配语言表达标识:一系列节点,这些节点中的每个节点表示相应的数据处理操作,以及该一系列节点中的特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

2. 如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,这些处理器可执行指令进一步使该至少一个计算机硬件处理器进行以下操作:

将处理布局分配给该更新的数据流图中的一个或多个节点中的每个节点,其中,根据所分配的一个或多个处理布局来执行该更新的数据流图。

3. 如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,该第二多个节点具有比该第一多个节点更少的节点并且该第二多个链路具有比该第一多个链路更少的链路。

4. 如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,生成该更新的数据流图包括:

选择第一优化规则;

标识该初始数据流图的将被应用该第一优化规则的第一部分;以及

将该第一优化规则应用于该初始数据流图的该第一部分。

5. 如权利要求4所述的至少一个非暂态计算机可读存储介质,其中,生成该更新的数据流图进一步包括:

选择不同于该第一优化规则的第二优化规则;

标识该初始数据流图的将被应用该第二优化规则的第二部分;以及

将该第二优化规则应用于该初始数据流图的该第二部分。

6. 如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,数据流子图模式匹配语言表达包括表示将被在数据流图中标识的各个模式的表达。

7. 如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,该数据流子图模式

匹配语言表达用符号标识该一系列节点中的该特定节点,该符号指示该特定节点的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

8.如权利要求4所述的至少一个非暂态计算机可读存储介质,其中,应用该第一优化规则包括应用选自以下各项组成的组的优化:去除冗余数据处理操作、强度降低优化、组合操作优化、宽度减小优化和去重优化。

9.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,生成更新的数据流图包括:标识表示冗余操作的第一节点;以及从该初始数据流图去除该第一节点。

10.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,生成更新的数据流图包括:标识该初始数据流图中的表示相应的排序数据处理操作的两个节点;以及用表示排序数据处理操作的单个节点来替换这两个节点。

11.如权利要求10所述的至少一个非暂态计算机可读存储介质,其中,在该初始数据流图中,这两个节点彼此不相邻。

12.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,生成更新的数据流图包括:用表示第二数据处理操作的第二节点来替换表示第一数据处理操作的第一节点,该第二数据处理操作具有比该第一数据处理操作弱的类型。

13.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,生成更新的数据流图包括:标识该初始数据流图中的表示相应的联接数据处理操作的两个节点;以及用表示联接数据处理操作的单个节点来替换这两个节点。

14.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,获得该自动生成的初始数据流图包括:自动生成该初始数据流图。

15.如权利要求1所述的至少一个非暂态计算机可读存储介质,其中,自动生成该初始数据流图包括:

获得结构化查询语言查询,即SQL查询;

针对该SQL查询生成查询计划;以及

使用该查询计划生成该初始数据流图。

16.一种用于数据流图的方法,包括:

使用至少一个计算机硬件处理器进行以下操作:

获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;

通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点;以及

至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图,

其中,生成该更新的数据流图包括:

使用数据流子图模式匹配语言表达来确定将被应用于该初始数据流图的一部分的优

化规则,

其中:

该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分,以及

该数据流子图模式匹配语言表达标识:一系列节点,这些节点中的每个节点表示相应的数据处理操作,以及该一系列节点中的特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

17. 如权利要求16所述的方法,进一步包括:

将处理布局分配给该更新的数据流图中的一个或多个节点中的每个节点,其中,执行该更新的数据流图是根据所分配的一个或多个处理布局进行的。

18. 如权利要求16所述的方法,其中,生成该更新的数据流图包括:

选择第一优化规则;

标识该初始数据流图的将被应用该第一优化规则的第一部分;以及

将该第一优化规则应用于该初始数据流图的该第一部分。

19. 如权利要求18所述的方法,其中,标识该初始数据流图的该第一部分包括标识表示第一数据处理操作的第一节点,该第一数据处理操作与由连接到该第一节点的第二节点表示的第二数据处理操作交换。

20. 如权利要求18所述的方法,其中,应用该第一优化规则包括应用选自以下各项组成的组的优化:去除冗余数据处理操作、强度降低优化、组合操作优化、宽度减小优化和去重优化。

21. 如权利要求16所述的方法,进一步包括:

获得结构化查询语言查询,即SQL查询;

针对该SQL查询生成查询计划;以及

使用该查询计划生成该初始数据流图。

22. 一种数据处理系统,包括:

至少一个计算机硬件处理器;以及

存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由该至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:

获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;

通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点;以及

至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图,

其中,生成该更新的数据流图包括:

使用数据流子图模式匹配语言表达来确定将被应用于该初始数据流图的一部分的优

化规则,

其中:

该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分,以及

该数据流子图模式匹配语言表达标识:一系列节点,这些节点中的每个节点表示相应的数据处理操作,以及该一系列节点中的特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

23. 如权利要求22所述的数据处理系统,其中,这些处理器可执行指令进一步使该至少一个计算机硬件处理器进行以下操作:

将处理布局分配给该更新的数据流图中的一个或多个节点中的每个节点,

其中,执行该更新的数据流图是根据所分配的一个或多个处理布局进行的。

24. 如权利要求22所述的数据处理系统,其中,生成该更新的数据流图包括:

选择第一优化规则;

标识该初始数据流图的将被应用该第一优化规则的第一部分;以及

将该第一优化规则应用于该初始数据流图的该第一部分。

25. 如权利要求24所述的数据处理系统,其中,标识该初始数据流图的该第一部分包括标识表示第一数据处理操作的第一节点,该第一数据处理操作与由连接到该第一节点的第二节节点表示的第二数据处理操作交换。

26. 如权利要求25所述的数据处理系统,其中,应用该第一优化规则包括应用选自自由以下各项组成的组的优化:去除冗余数据处理操作、强度降低优化、组合操作优化、宽度减小优化和去重优化。

27. 如权利要求22所述的数据处理系统,其中,这些处理器可执行指令进一步使该至少一个计算机硬件处理器进行以下操作:

获得结构化查询语言查询,即SQL查询;

针对该SQL查询生成查询计划;以及

使用该查询计划生成该初始数据流图。

28. 一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些处理器可执行指令包括:

用于获得自动生成的初始数据流图的手段,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;

用于通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图的手段,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点;以及

用于至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图的手段,

其中,生成该更新的数据流图包括:

使用数据流子图模式匹配语言表达来确定将被应用于该初始数据流图的一部分的优

化规则,

其中:

该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分,以及

该数据流子图模式匹配语言表达标识:一系列节点,这些节点中的每个节点表示相应的数据处理操作,以及该一系列节点中的特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

29. 一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:

获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及

通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点,该生成包括:

至少部分地通过标识该初始数据流图的将被应用第一优化规则的第一部分中的交换节点来标识该初始数据流图的该第一部分,其中,标识该交换节点包括标识一系列节点中的特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果;以及
将该第一优化规则应用于该初始数据流图的该第一部分。

30. 如权利要求29所述的至少一个非暂态计算机可读存储介质,其中,这些处理器可执行指令进一步使该至少一个计算机硬件处理器进行以下操作:

将处理布局分配给该更新的数据流图中的一个或多个节点中的每个节点,

其中,执行该更新的数据流图是根据所分配的一个或多个处理布局进行的。

31. 如权利要求29所述的至少一个非暂态计算机可读存储介质,其中,标识该初始数据流图的该第一部分包括使用数据流子图模式匹配语言表达来标识该交换节点,该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分,其中该数据流子图模式匹配语言表达标识:该一系列节点,这些节点中的每个节点表示相应的数据处理操作,以及该一系列节点中的该特定节点,该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的处理的结果。

32. 如权利要求29所述的至少一个非暂态计算机可读存储介质,其中,应用该第一优化规则包括应用选自自由以下各项组成的组的优化:去除冗余数据处理操作、强度降低优化、组合操作优化、宽度减小优化和去重优化。

33. 如权利要求29所述的至少一个非暂态计算机可读存储介质,其中,这些处理器可执行指令进一步使至少一个计算机硬件处理器进行以下操作:

获得结构化查询语言查询,即SQL查询;

针对该SQL查询生成查询计划;以及

使用该查询计划生成该初始数据流图。

34. 一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质, 这些指令在由至少一个计算机硬件处理器执行时, 使该至少一个计算机硬件处理器进行以下操作:

获得自动生成的初始数据流图, 该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路; 以及

通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图, 该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路, 其中, 该第二多个节点包括表示第一数据处理操作的节点和表示第二数据处理操作的另一节点, 该生成包括:

从以下各项中选择要应用于该初始数据流图的第一部分的第一优化规则: 去除冗余数据处理操作、强度降低优化、宽度减小优化和去重优化; 以及

将该第一优化规则应用于该初始数据流图的该第一部分,

其中, 生成该更新的数据流图包括:

使用数据流子图模式匹配语言表达来确定将被应用于该初始数据流图的一部分的优化规则,

其中:

该数据流子图模式匹配语言表达用于标识数据流图的具有交换节点的部分, 以及

该数据流子图模式匹配语言表达标识: 一系列节点, 这些节点中的每个节点表示相应的数据处理操作, 以及该一系列节点中的特定节点, 该特定节点在该一系列节点中的顺序能够相对于该一系列节点中的一个或多个其他节点改变而不改变由该一系列节点所表示的数据处理操作进行的结果。

35. 如权利要求34所述的至少一个非暂态计算机可读存储介质, 其中, 这些处理器可执行指令进一步使该至少一个计算机硬件处理器进行以下操作:

将处理布局分配给该更新的数据流图中的一个或多个节点中的每个节点,

其中, 执行该更新的数据流图是根据所分配的一个或多个处理布局进行的。

36. 如权利要求34所述的至少一个非暂态计算机可读存储介质, 其中, 这些处理器可执行指令进一步使至少一个计算机硬件处理器进行以下操作:

获得结构化查询语言查询, 即SQL查询;

针对该SQL查询生成查询计划; 以及

使用该查询计划生成该初始数据流图。

用于数据流图优化的系统和方法

技术领域

[0001] 本发明涉及数据处理系统。

背景技术

[0002] 数据处理系统可以使用一个或多个计算机程序来处理数据。数据处理系统利用的一个或多个计算机程序可以被开发为数据流图。数据流图可以包括表示要对输入数据进行的数据处理操作的组件(称为“节点”或“顶点”)、以及组件之间的表示数据流的链路。数据流图的节点可以包括:表示相应输入数据集的一个或多个输入节点;表示相应输出数据集的一个或多个输出节点;以及表示要对数据进行的数据处理操作的一个或多个节点。用于执行由数据流图进行编码的的计算的技术在名称为“Executing Computations Expressed as Graphs[执行表达为图的计算]”的美国专利号5,966,072以及名称为“Managing Parameters for Graph-Based Computations[管理基于图的计算的参数]”的美国专利7,716,630中进行了描述,这些美国专利中的每一个通过引用以其全文并入本文。

发明内容

[0003] 一些实施例涉及一种数据处理系统,该数据处理系统包括:至少一个计算机硬件处理器;以及存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由该至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示该第二数据处理操作的另一节点;以及至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图。

[0004] 一些实施例涉及一种方法,该方法包括使用至少一个计算机硬件处理器进行以下操作:获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示该第二数据处理操作的另一节点;以及至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图。

[0005] 一些实施例涉及一种存储处理器可执行指令的至少一个非暂态计算机可读存储

介质,这些指令在由至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:获得自动生成的初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示该第二数据处理操作的另一节点;以及至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图。

[0006] 一些实施例涉及一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些处理器可执行指令包括:用于获得自动生成的初始数据流图的手段,该初始数据流图包括表示第一多个数据处理操作的第一多个节点和表示在该第一多个节点中的节点之间的数据流的第一多个链路;以及用于通过迭代地应用数据流图优化规则来更新该初始数据流图以生成更新的数据流图的手段,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点和表示该第二多个节点中的节点之间的数据流的第二多个链路,其中,该第二多个节点包括表示第一数据处理操作的节点和表示该第二数据处理操作的另一节点;以及用于至少部分地通过使用第一计算机系统进程执行该第一数据处理操作并且使用不同于该第一计算机系统进程的第二计算机系统进程执行该第二数据处理操作来执行该更新的数据流图的手段。

[0007] 一些实施例涉及一种数据处理系统,该数据处理系统包括:至少一个计算机硬件处理器;以及存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由该至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:获得结构化查询语言 (SQL) 查询;针对该SQL查询生成查询计划;使用该查询计划生成初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点;以及通过使用至少一个数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点。

[0008] 一些实施例涉及一种方法,该方法包括使用至少一个计算机硬件处理器进行以下操作:获得结构化查询语言 (SQL) 查询;针对该SQL查询生成查询计划;使用该查询计划生成初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点;以及通过使用至少一个数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点。

[0009] 一些实施例涉及一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些指令在由至少一个计算机硬件处理器执行时,使该至少一个计算机硬件处理器进行以下操作:获得结构化查询语言 (SQL) 查询;针对该SQL查询生成查询计划;使用该查询计划生成初始数据流图,该初始数据流图包括表示第一多个数据处理操作的第一多个节点;以及通过使用至少一个数据流图优化规则来更新该初始数据流图以生成更新的数据流图,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点。

[0010] 一些实施例涉及一种存储处理器可执行指令的至少一个非暂态计算机可读存储介质,这些处理器可执行指令包括:用于获得结构化查询语言 (SQL) 查询的手段;用于针对

该SQL查询生成查询计划的手段;用于使用该查询计划生成初始数据流图的手段,该初始数据流图包括表示第一多个数据处理操作的第一多个节点;以及用于通过使用至少一个数据流图优化规则来更新该初始数据流图以生成更新的数据流图的手段,该更新的数据流图包括表示第二多个数据处理操作的第二多个节点。

[0011] 前述内容为本发明的非限制性发明内容,其由所附权利要求限定。

附图说明

[0012] 将参照以下附图描述各个方面和实施例。应当理解,附图不一定按比例绘制。出现在多个附图中的项在其出现的所有附图中由相同或相似的附图标记指示。

[0013] 图1A是本文所描述的技术的一些实施例可以在其中操作的说明性计算环境的框图。

[0014] 图1B是根据本文所描述的技术的一些实施例的用于根据输入结构化查询语言(SQL)查询自动生成数据流图的说明性过程的流程图。

[0015] 图2是根据本文所描述的技术的一些实施例的用于根据输入SQL查询自动生成数据流图的说明性过程200的流程图。

[0016] 图3A展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以去除一个或多个冗余数据处理操作。

[0017] 图3B展示了根据本文所描述的技术的一些实施例改变交换数据处理操作的顺序以促进将优化规则应用于另一说明性数据流图。

[0018] 图3C展示了根据本文所描述的技术的一些实施例将优化规则应用于另一说明性数据流图以去除一个或多个冗余数据处理操作。

[0019] 图3D展示了根据本文所描述的技术的一些实施例将优化规则应用于又一说明性数据流图以去除一个或多个冗余数据处理操作。

[0020] 图4A展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行强度降低优化。

[0021] 图4B展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行另一强度降低优化。

[0022] 图5A展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行组合操作优化。

[0023] 图5B展示了根据本文所描述的技术的一些实施例将优化规则应用于另一说明性数据流图以进行组合操作优化。

[0024] 图5C展示了根据本文所描述的技术的一些实施例将优化规则应用于又一说明性数据流图以进行组合操作优化。

[0025] 图5D展示了根据本文所描述的技术的一些实施例将优化规则应用于又一说明性数据流图以进行组合操作优化。

[0026] 图6展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以去除未引用的数据处理操作。

[0027] 图7展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行宽度减小优化。

[0028] 图8A展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行去重优化。

[0029] 图8B展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以通过拉链式组合进行去重优化。

[0030] 图9展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图以进行串行到并行的优化。

[0031] 图10A展示了根据本文所描述的技术的一些实施例的初始数据流图。

[0032] 图10B展示了根据本文所描述的技术的一些实施例通过迭代地将优化规则应用于图10A所示的初始数据流图而获得的更新的数据流图。

[0033] 图10C展示了根据本文所描述的技术的一些实施例的图10A的初始数据流图的另一视图。

[0034] 图10D展示了根据本文所描述的技术的一些实施例的图10B的更新的数据流图的另一视图。

[0035] 图11是可以用于实施所描述的技术的一些实施例的说明性计算系统环境的框图。

具体实施方式

[0036] 本文所描述的技术的方面涉及通过改进用于使用数据流图来进行数据处理操作的常规技术来提高数据处理系统的速度、吞吐量和准确性。

[0037] 一些数据处理系统使用数据流图来处理数据。在许多情况下,数据流图是自动生成的而不是手动指定的。例如,一些数据处理系统可以根据结构化查询语言(SQL)查询自动生成数据流图。在这种情况下,用户或计算机程序可以将输入SQL查询提供给数据处理系统,并且数据处理系统可以通过根据SQL查询生成数据流图并执行所生成的数据流图来执行SQL查询。作为另一个示例,数据处理系统可以从用户或计算机程序接收输入查询的表示(该表示不是SQL查询),并且可以根据查询的表示自动生成数据流图。作为又一示例,数据处理系统可以从另一数据处理系统接收输入数据流图。该输入数据流图可能不适合在该数据处理系统中执行(即使在该输入数据流图可能适合在该另一数据处理系统中执行的情况下)。因此,该数据处理系统根据该输入数据流图生成适合在该数据处理系统上执行的新的数据流图。

[0038] 发明人已经认识到,可以在用于(例如根据如以上所讨论的SQL查询、其他查询表示或输入数据流图)自动生成数据流图的常规技术上改进。虽然可以在数据处理系统上执行自动生成的数据流图,但是用于自动生成数据流图的常规技术所产生的数据流图的执行需要大量的计算资源(例如,处理器资源、存储器资源、网络资源等)和大量的时间。例如,自动生成的数据流图:(1)可能会包括表示冗余数据处理操作的节点;(2)可能会需要进行其结果后续未被使用的数据处理操作;(3)在可以进行并行处理的情况下可能会需要不必要地进行串行处理;(4)可能会将数据处理操作应用于比获得期望的结果所需的更多的数据;(5)可能会将计算分解为在多个节点上进行,这会显著增加在由计算机程序中的专用线程、专用计算机程序(例如,操作系统中的进程)和/或专用计算设备对每个数据流图节点进行数据处理的情况下进行计算的计算成本;(6)在需要较少计算的较弱类型的数据处理操作(例如,组内排序操作或组内上卷(roll-up)操作等)就足够时,可能会需要进行需要更

多计算的较强类型的数据操作(例如,排序操作、上卷操作等);和/或(7)可能会需要重复处理工作。

[0039] 发明人已经进一步认识到,即使一些常规的优化技术被用作数据流图生成过程的一部分,用于生成数据流图的常规自动化技术所产生的数据流图的执行也可能会需要大量的计算资源和大量的时间。例如,可以通过根据SQL查询生成查询计划并根据所生成的查询计划生成数据流图来根据SQL查询生成数据流图。但是,即使查询计划的生成涉及了进行某些优化,但所产生的数据流图(根据查询计划产生的)也可能需要花费大量的计算资源。实际上,根据查询计划生成数据流图的常规技术通常会导致效率低下,并且可能并未完善到足以产生可以在计算方面高效的方式执行的数据流图。

[0040] 发明人已经认识到,如果进一步处理和优化自动生成的数据流图以减少计算资源用于执行所生成的数据流图的量,则数据处理系统的性能将被改善。为此,发明人开发了本申请中所描述的一些数据流图优化技术。本文所描述的数据流图优化技术通过减少用于执行至少部分地通过使用数据流图优化技术生成的数据流图的计算资源(例如,处理器资源、存储器资源、网络资源等)的量来改善数据处理系统的性能(例如,吞吐量、速度、准确性等)。

[0041] 本文所描述的数据流图优化技术的另一益处是,作为数据处理系统的一部分而存在的数据流图优化器允许其他数据处理系统组件的开发人员和/或数据处理系统的用户依赖于数据流图优化器,而不是尝试将自组织优化作为其自己工作的一部分来进行。该数据流图优化器不仅能够减少这些开发人员和/或用户所做的工作,而且还能防止他们不经意间将错误引入数据处理系统,这当然还由于减少了错误数量而改善了数据处理系统。

[0042] 应当理解,本文所描述的数据流图优化技术可以但不一定会产生在某种意义上“最优”的数据流图。替代地,这些优化技术通常尝试通过在执行数据流图之前改变数据流图来提高执行数据流图的计算效率,从而改善数据处理系统在执行数据流图时的性能。

[0043] 本文所描述的一些实施例解决了发明人已经认识到的用于自动生成数据流图的常规技术所具有的所有以上所描述的问题。然而,并非本文所描述的每个实施例都解决了这些问题中的每一个,并且一些实施例可能不能解决这些问题中的任何一个。这样,应当理解,本文所描述的技术的实施例不限于解决用于自动生成数据流图的常规技术的以上所讨论的问题中的所有或任何一个。例如,本文所描述的技术的一些实施例可以应用于优化手动指定的数据流图,因为这种数据流图也可能包括效率低下且需要比必要的更多的计算资源的问题。

[0044] 因此,一些实施例提供了用于根据SQL查询和/或其他输入自动生成数据流图的新颖技术。本文提供了这样的其他输入的示例。在一些实施例中,数据处理系统可以:(1)获得结构化查询语言(SQL)查询;(2)针对该SQL查询生成查询计划;(3)使用该查询计划生成初始数据流图;并且(4)通过使用至少一个数据流图优化规则来更新初始数据流图,从而生成更新的数据流图。

[0045] 在一些实施例中,可以保存更新的数据流图(例如,保存在非易失性存储器中)以供后续使用。附加地或可替代地,在一些实施例中,更新的数据流图可以由数据处理系统执行。在执行之前,数据处理系统可以将处理布局分配给更新的数据流图的一个或多个节点中的每个节点。

[0046] 在一些实施例中,初始数据流图可以包括第一多个节点,该第一多个节点表示将在数据处理系统执行初始数据流图时被执行的相应的多个数据处理操作。更新的数据流图可以包括第二多个节点,该第二多个节点表示将在数据处理系统执行更新的数据流图时被执行的相应的多个数据处理操作。在一些实施例中,该第二多个节点具有比该第一多个节点更少的节点。在这样的实施例中,更新的数据流图中的节点数量少于初始数据流图中的节点数量。相对于在执行初始数据流图时数据处理系统将使用的计算资源,图中的节点数量的减少可以减少替代地执行更新的数据流图所需的计算资源量。

[0047] 在一些实施例中,数据处理系统可以通过将一个或多个数据流图优化规则(本文提供了其示例)应用于初始数据流图的一个或多个部分来从初始数据流图生成更新的数据流图。可以迭代地应用优化规则。例如,在一些实施例中,数据处理系统可以通过以下方式更新初始数据流图:(1)选择第一优化规则;(2)标识该初始数据流图的将被应用该第一优化规则的第一部分;以及(3)将该第一优化规则应用于该初始数据流图的该第一部分。随后,数据处理系统可以通过以下方式继续更新初始数据流图:(1)选择不同于该第一优化规则的第二优化规则;(2)标识该初始数据流图的将被应用该第二优化规则的第二部分;以及(3)将该第二优化规则应用于该初始数据流图的该第二部分。

[0048] 对于正在更新的初始数据流图,可以考虑优化规则的应用的多种方式。例如,在一些实施例中,针对每个特定的优化规则,数据处理系统可以标识数据流图的适用特定优化规则的部分,并将该优化规则应用于所标识的部分。作为另一个示例,在一些实施例中,针对数据流图的每个特定部分,数据处理系统可以标识可以应用于该特定部分的优化规则,并且将所标识的优化规则应用于该特定部分。在这样的实施例中,可以以拓扑方式对初始数据流图进行排序,并且可以遍历(例如,从左到右)以拓扑方式排序的图,以标识可以被应用优化规则的特定部分。

[0049] 在一些实施例中,数据处理系统可以采用数据流子图模式匹配语言来标识初始数据流图的将被应用一个或多个优化规则的一个或多个部分。数据流子图模式匹配语言可以包括表示要在数据流图中标识的相应模式的一个或多个表达。本文提供了此类表达的示例。

[0050] 在一些实施例中,为了标识初始数据流图的将被应用特定优化规则的部分,数据处理系统可以确定一个或多个节点是否可与一个或多个其他节点交换。换句话说,数据处理系统可以确定是否可以在不改变处理结果的情况下更改数据流图中一个或多个节点的出现顺序。这是一个有价值特征,因为如果存在交换节点,则通过改变至少一些交换节点的顺序,优化规则可能变得适用于图的一部分,而否则该规则可能不适用。

[0051] 例如,优化规则可能涉及标识初始数据流图中的表示相应的排序操作的两个相邻节点,其中,第二排序操作会使第一操作的效果无效,因此应删除第一操作(参见例如图3B和图3C所示的示例)。根据定义,这种优化规则将不会被应用于不具有表示排序操作的相邻节点的数据流图。然而,如果表示第一排序操作的第一节点要与一个或多个其他节点交换,则可以改变第一节点与一个或多个其他节点中的至少一个节点的顺序,使得表示第一排序操作的第一节点与表示第二排序操作的第二节点相邻地放置。作为以这种方式对节点进行交换的结果,去除冗余的第一排序操作的优化规则可以被应用于数据流图。因此,在一些实施例中,标识初始数据流图的第一部分可以包括标识表示排序数据处理操作的第一节点,

该排序数据处理操作与由连接到该第一节点的第二个节点表示的数据处理操作交换。

[0052] 在一些实施例中,当从初始数据流图生成更新的数据流图时,可以应用多种类型的优化规则中的任何一种或多种。通过举例且非限制性,将优化规则应用于初始数据流图可以涉及去除一个或多个冗余数据处理操作、去除一个或多个未引用的数据处理操作、进行一次或多次强度降低优化、进行一次或多次组合操作优化、进行一次或多次宽度减小优化和/或进行一次或多次去重优化。

[0053] 在一些实施例中,优化规则可以体现在程序代码中,该程序代码在被执行时,使得对数据流图进行对应的优化。例如,用于去除冗余的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(从将被应用该规则的数据流图中)去除表示被确定为冗余的数据处理操作的至少一个节点。该程序代码可以以任何编程语言来编写,因为本文所描述的技术的各方面在这方面不受限制。

[0054] 作为又一示例,用于去除一个或多个未引用的数据处理操作的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(从将被应用该规则的数据流图中)去除表示结果未被引用和/或未被使用的数据处理操作(例如,由于在后续处理中不需要或不依赖排序所产生的顺序,因此未被引用的排序数据处理操作)的至少一个节点。

[0055] 作为又一示例,用于进行强度降低的优化规则可以体现在这样的程序代码中该程序代码在被执行时,使得(在将被应用该规则:的数据流图中)表示第一数据处理操作的第一节点(例如,表示排序数据处理操作的节点)被表示比该第一数据处理操作更弱类型的第二数据处理操作的第二个节点(例如,表示组内排序数据处理操作的节点)替换。

[0056] 作为又一示例,用于进行组合操作优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(在将被应用该规则的数据流图中)表示多个操作的多个节点被表示该多个操作的组合的单个节点替换。

[0057] 作为又一示例,用于进行宽度减小优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得一些数据(例如,一列或多列数据)在进行后续操作之前在图中的特定部分处被删除,因为这些数据(即所删除的数据)在后续操作中不会被使用并且不需要作为处理的一部分进行传播。作为又一示例,数据流图中的节点可以被配置为进行若干个计算,并且这些计算中的一些计算的结果可能未被使用。因此,在一些实施例中,用于进行宽度减小优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得特定节点由被配置为仅进行结果被使用的那些计算的另一个节点替换;不必要的计算不再会被进行。

[0058] 作为又一示例,用于进行去重优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得数据流图的将被应用该规则的不同分支合并。

[0059] 应当理解,可以以多种方式中的任何方式来实施上文介绍的和下文更详细讨论的技术,因为这些技术不限于任何特定实施方式。本文仅出于说明性目的提供了实施方式细节的示例。此外,由于本文所描述的技术的方面不限于使用任何特定技术或技术组合,因此本文所披露的技术可以单独使用或以任何合适的组合使用。

[0060] 图1A是根据本文所描述的技术的一些实施例的说明性数据处理系统100的图示。如图1A中所展示的,数据处理系统100包括查询输入模块104、查询计划生成器106、数据流图生成器108、图优化器110、布局分配模块112和图执行引擎115。

[0061] 数据处理系统100被配置为访问(例如,从中读取数据和/或向其中写入数据)数据存储装置102-1、102-2、…、和102-n。数据存储装置102-1、102-2、…、和102-n中的每一个可以存储一个或多个数据集。数据存储装置可以以任何合适的方式存储任何合适类型的数据。数据存储装置可以使用数据库系统(例如,关系数据库系统)或以任何其他合适的方式将数据存储为平面文本文件、电子表格。在一些实例中,数据存储装置可以存储交易数据。例如,数据存储装置可以存储信用卡交易、电话记录数据或银行交易数据。应当理解,数据处理系统100可以被配置为访问任何合适类型的任何合适数量的数据存储装置,因为本文所描述的技术的方面在这方面不受限制。可以将数据处理系统100可以被配置为从其读取数据的数据存储装置称为数据源。可以将数据处理系统100可以被配置为向其写入数据的数据存储装置称为数据接收器。

[0062] 在一些实施例中,数据存储装置102-1、102-2、…、102-n可以是相同类型(例如,全部可以是关系数据库)或不同类型(例如,一种可以是关系数据库,而另一种可以是将数据存储存储在平面文件中的数据存储装置)。数据存储装置可以是SQL server数据存储装置、ORACLE数据存储装置、TERADATA数据存储装置、平面文件数据存储装置、多文件数据存储装置、HADOOP数据存储装置、DB2数据存储装置、Microsoft SQL SEVER数据存储装置、INFORMIX数据存储装置、SAP数据存储装置、MongoDB数据存储装置、元数据数据存储装置、和/或任何其他合适类型的数据存储装置,因为本文所描述的技术的方面在这方面不受限制。

[0063] 在一些实施例中,查询输入模块104可以被配置为接收输入SQL查询。在一些实施例中,查询输入模块104可以被配置为从用户接收输入SQL查询。例如,查询输入模块104可以被配置为生成图形用户界面,用户可以通过该图形用户界面输入SQL查询。作为另一个示例,查询输入模块104可以被配置为接收由用户通过图形用户界面(该图形用户界面不一定由查询输入模块104本身生成)提供的信息。在一些实施例中,查询输入模块104可以被配置为从另一个计算机程序接收输入SQL查询。例如,查询输入模块104可以暴露可以通过其提供输入SQL查询的应用编程接口(API)(例如,开放数据库互连(ODBC)API和Java数据库互连(JDBC)API),可以响应于表明要访问SQL查询的通知来访问SQL查询,或者以任何其他合适的方式从另一个计算机程序接收输入SQL查询。

[0064] 由查询输入模块104接收的SQL查询可以涉及从单个数据存储装置读取数据和/或向其写入数据。可替代地,由查询输入模块104接收的SQL查询可以涉及从多个数据存储装置读取数据和/或向其写入数据。当数据存储装置是不同类型时,SQL查询可以称为联合SQL查询。在一些实施例中,SQL查询可以涉及从联合数据库读取数据和/或向其写入数据。

[0065] 在一些实施例中,查询计划生成器106被配置为根据由查询输入模块104接收的SQL查询来生成查询计划。如果执行了SQL查询,则所生成的查询计划可以标识要进行的一个或多个数据处理操作。所生成的查询计划可以进一步指定要执行所标识的数据处理操作的顺序。这样,所生成的查询计划可以表示为了执行由查询输入模块104接收的SQL查询而要进行的一系列数据处理操作。查询计划生成器106可以被配置为以任何合适的方式生成查询计划。例如,在一些实施例中,查询计划生成器106可以实施在名称为“Managing Data Queries[管理数据查询]”的美国专利号9,116,955中描述的用于生成查询计划的任何技术,该美国专利通过引用以其全文并入本文。

[0066] 在一些实施例中,数据流图生成器108被配置为根据由查询计划生成器106生成的查询计划来生成初始数据流图。数据流图生成器108可以被配置为以任何合适的方式根据查询计划生成初始数据流图。例如,在一些实施例中,数据流图生成器108可以实施在名称为“Managing Data Queries[管理数据查询]”的美国专利号9,116,955中描述的用于生成查询计划的任何技术,该美国专利通过引用以其全文并入本文。

[0067] 在一些实施例中,数据流图可以包括表示要对输入数据进行的数据处理操作的组件(称为“节点”或“顶点”)、以及组件之间的表示数据流的链路。数据流图的节点可以包括:表示相应输入数据集的一个或多个输入节点;表示相应输出数据集的一个或多个输出节点;以及表示要对数据进行的数据处理操作的一个或多个节点。在一些实施例中,输入节点可以表示联合数据库或任何其他类型的数据库。类似地,在一些实施例中,输出节点可以表示联合数据库或任何其他类型的数据库。

[0068] 在一些实施例中,可以使用不同的相应计算机系统进程来执行由数据流图中的不同节点表示的不同数据处理操作。例如,数据流图可以包括表示第一数据处理操作(例如,“排序”操作)的第一节点和表示不同于第一数据处理操作的第二数据处理操作(例如,“联接”操作)的第二节点,并且在一些实施例中,可以使用第一计算机系统进程来执行第一数据处理操作,并且可以使用不同于第一计算机系统进程的第二计算机系统进程来执行第二数据处理操作。在一些实施例中,第一计算机系统进程和第二计算机系统进程可以在同一计算设备上执行,并且例如可以由同一操作系统来管理。在其他实施例中,第一计算机进程和第二计算机系统进程可以在不同的计算设备上执行。

[0069] 在一些实施例中,用于执行由数据流图中的节点表示的数据处理操作的计算机系统进程可以是配置为执行用于对数据处理操作进行编码的处理器可执行指令的计算机程序的实例。计算机系统进程可以是单线程进程或多线程进程。计算机系统进程可以与一个或多个计算机系统资源相关联,通过举例且非限制性,该一个或多个计算机系统资源包括:表示对数据处理操作进行编码的处理器可执行指令、存储器(例如,保存可执行代码、进程专用输入数据和/输出数据、调用栈、计算堆和/或其他数据的物理存储器和/或虚拟存储器的区域)、进程标识符(例如,由操作系统用于标识计算机系统进程的标识符)、安全属性(例如,指示进程的一个或多个拥有者和/或允许计算机系统进程进行的操作的许可)和/或指定计算机系统进程的状态的信息。

[0070] 在一些实施例中,可以至少部分地通过以下方式根据查询计划生成初始数据流图:生成初始数据流图以包括针对在查询计划中标识的数据处理操作的至少一个子集(例如,一些或全部)中的每一个的节点。随后,查询计划中指定的数据处理操作的顺序可以用于生成连接初始数据流图中的节点的链路。例如,当所生成的查询计划指示在第二数据处理操作之前进行第一数据处理操作时,所生成的初始数据流图可以具有第一节点(表示第一数据处理操作)和第二节点(表示第二数据处理操作)以及指定从第一节点到第二节点的路径的一个或多个链路。

[0071] 在一些实施例中,根据查询计划生成初始数据流图包括将一个或多个节点添加到表示输入和/或输出数据源的图。例如,生成初始数据流图可以包括为在执行SQL查询期间要从其中读取数据记录的每个数据源添加输入节点。每个输入节点可以配置有与相应数据源相关联的参数值。这些值可以指示如何访问数据源中的数据记录。作为另一个示例,生成

初始数据流图可以包括为在执行SQL查询期间要向其写入数据记录的每个数据接收器添加输出节点。每个输出节点可以配置有与相应数据接收器相关联的参数值。这些值可以指示如何将数据记录写入数据源。在一些实施例中,初始数据流图可由图执行引擎运行。在其他实施例中,初始数据流图不可由图执行引擎运行。

[0072] 在一些实施例中,图优化器110被配置为通过使用一个或多个数据流图优化规则来更新由数据流图生成器108生成的初始数据流图,从而生成更新的数据流图。图优化器110可以被配置为将本文所描述的多种类型的优化规则中的任何一种或多种应用于初始数据流图。例如,图优化器110可以被配置为通过以下方式来更新初始数据流图:去除一个或多个冗余数据处理操作、去除一个或多个未引用的数据处理操作、进行一次或多次强度降低优化、进行一次或多次组合操作优化、进行一次或多次宽度减小优化和/或进行一次或多次去重优化。图优化器110可以被配置为以任何合适的方式进行操作,并且例如可以被配置为根据参照图2所描述的说明性过程200或其一个或多个变体进行操作。

[0073] 在一些实施例中,布局分配模块112可以确定由图优化器110生成的更新的数据流图中的相应节点表示的一个或多个数据处理操作中的每一个的处理布局。数据处理操作的处理布局可以指定要使用多少个计算设备来进行数据处理操作,并且可以标识要用于进行数据处理操作的特定计算设备。因此,在一些实施例中,布局分配模块112可以针对更新的数据流图中的一个或多个节点中的每一个,确定是要使用单个设备(例如,单个处理器、单个虚拟机等)还是多个设备(例如,多个处理器、多个虚拟机等)以及应当使用哪些设备来进行数据处理操作。在一些实施例中,布局分配模块可以将不同的并行性程度分配给更新的数据流图中的不同节点。这样,应当理解,可以将不同的处理布局分配给在执行由图优化器110生成的更新的数据流图期间要进行的不同的数据处理操作。

[0074] 在一些实施例中,更新的数据流图可以包括表示不同的数据处理操作的多个(例如,两个或更多个)节点,并且可以使用不同的进程来执行数据处理操作。例如,可以使用一个或多个计算机系统进程来执行由第一节点表示的数据处理操作(例如,在数据处理操作是并行化的时可以使用多个计算机系统进程),并且可以使用一个或多个其他计算机系统进程来执行由更新的数据流图中的不同于第一节点的第二节点表示的数据处理操作。

[0075] 在一些实施例中,图执行引擎115被配置为执行一个或多个数据流图。例如,在初始数据流图可运行的一些实施例中,图执行引擎115可以被配置为执行由数据流图生成器108生成的任何初始数据流图。作为另一个示例,图执行引擎115可以被配置为执行由图优化器110生成的任何更新的数据流图。图执行引擎可以包括用于执行数据流图的协作系统或任何其他合适的执行环境。用于开发和执行数据流图的环境的方面在名称为“Executing Computations Expressed as Graphs[执行表示为图的计算]”的美国专利号5,966,072以及名称为“Managing Parameters for Graph-Based Computations[管理基于图的计算的参数]”的美国专利7,716,630中进行了描述,这些美国专利中的每一个通过引用以其全文并入本文。

[0076] 图1B是根据本文所描述的技术的一些实施例的用于根据输入结构化查询语言(SQL)查询自动生成数据流图的说明性过程120的流程图。过程120可以由任何合适的数据处理系统进行,例如可以由参照图1A所描述的数据处理系统100进行。

[0077] 过程120在动作122处开始,在该动作中,接收SQL查询。可以使用查询输入模块104

来接收SQL查询。这可以以任何合适的方式来完成,包括以参照过程200的动作202所描述的任何方式。

[0078] 接下来,过程120进行到动作124,在该动作中,根据在动作122处接收的SQL查询生成查询计划。可以使用查询计划生成器106来生成查询计划。这可以以任何合适的方式来完成,包括以参照过程200的动作204所描述的任何方式。

[0079] 接下来,过程120进行到动作126,在该动作中,根据在动作124处获得的查询计划生成初始数据流图。可以由数据流图生成器108生成初始数据流图。这可以以任何合适的方式来完成,包括以参照过程200的动作206所描述的任何方式。

[0080] 接下来,过程120进行到动作128,在该动作中,通过将一个或多个优化规则应用于初始数据流图来从初始数据流图生成更新的数据流图。可以由图优化器110生成更新的数据流图。这可以以任何合适的方式来完成,包括以参照过程200的动作207所描述的任何方式。

[0081] 更新的数据流图可以被存储以供后续使用或可以由数据处理系统执行。在执行之前,可以将处理布局分配给由更新的数据流图中的节点表示的一个或多个数据处理操作。可以由布局分配模块112将处理布局分配给数据处理操作。

[0082] 图2是根据本文所描述的技术的一些实施例的用于根据输入SQL查询自动生成数据流图的说明性过程200的流程图。可以使用任何合适的数据处理系统来执行过程200,该数据处理系统包括例如参照图1A所描述的数据处理系统100。

[0083] 过程200在动作202处开始,在该动作中,接收SQL查询。在一些实施例中,作为用户将SQL查询作为输入提供给数据处理系统的结果,可以由执行过程200的数据处理系统接收SQL查询。用户可以通过图形用户界面或任何其他合适类型的界面输入SQL查询。在其他实施例中,可以通过另一个计算机程序将SQL查询提供给数据处理系统。例如,SQL查询可以由计算机程序提供,该计算机程序被配置为使数据处理系统执行一个或多个SQL查询,该一个或多个SQL查询中的每一个可以已由用户指定或自动生成。SQL查询可以是任何合适的类型,并且可以以任何合适的格式提供,因为本文所描述的技术的方面在这方面不受限制。

[0084] 接下来,过程200进行到动作204,在该动作中,根据在动作202处接收的SQL查询生成查询计划。如果执行了SQL查询,则所生成的查询计划可以标识要进行的一个或多个数据处理操作。所生成的查询计划可以进一步指定要执行所标识的数据处理操作的顺序。这样,所生成的查询计划可以表示为了执行在动作202处接收的SQL查询而要进行的一系列数据处理操作。可以使用任何合适的类型的查询计划生成器(例如,查询计划生成器106)来生成所生成的查询计划。在名称为“Managing Data Queries[管理数据查询]”的美国专利号9,116,955中描述了用于生成查询计划的一些说明性技术,该美国专利通过引用以其全文并入本文。

[0085] 接下来,过程200进行到动作206,在该动作中,使用在动作202处接收的SQL查询根据在动作204处生成的查询计划生成初始数据流图。在一些实施例中,可以至少部分地通过以下方式根据查询计划生成初始数据流图:生成初始数据流图以包括针对在查询计划中标识的数据处理操作的至少一个子集(例如,一些或全部)中的每一个的节点。在一些实施例中,查询计划中的单个节点可以导致初始数据流图中包含多个节点。随后,查询计划中指定的数据处理操作的顺序可以用于生成连接初始数据流图中的节点的链路。例如,当所生成

的查询计划指示在第二数据处理操作之前进行第一数据处理操作时,所生成的初始数据流图可以具有第一节点(表示第一数据处理操作)和第二节点(表示第二数据处理操作)以及指定从第一节点到第二节点的路径的一个或多个链路。

[0086] 在一些实施例中,根据查询计划生成初始数据流图包括将一个或多个节点添加到表示输入和/或输出数据源的图。例如,生成初始数据流图可以包括为在执行SQL查询期间要从其中读取数据记录的每个数据源添加输入节点。每个输入节点可以配置有与相应数据源相关联的参数值。这些值可以指示如何访问数据源中的数据记录。作为另一个示例,生成初始数据流图可以包括为在执行SQL查询期间要向其写入数据记录的每个数据接收器添加输出节点。每个输出节点可以配置有与相应数据接收器相关联的参数值。这些值可以指示如何将数据记录写入数据源。

[0087] 应当理解,在动作206处生成的初始数据流图不同于在动作204处生成的查询计划。数据流图可以由图执行引擎(例如,图执行引擎115)执行,而查询计划不能由图执行引擎执行——查询计划是用于生成数据流图的中间表示,该数据流图由图执行引擎执行以便执行SQL查询。查询计划是不可执行的,并且即使在关系数据库管理系统的背景下,也需要对其进行进一步处理以生成执行策略。相比而言,数据流图可由图执行引擎执行,以便进行SQL查询。另外,即使在由关系数据库系统进行进一步处理之后,所得的执行策略也不允许从其他类型的数据源和/或数据接收器读取数据和/或向其写入数据,而数据流图在这方面不受限制。

[0088] 在一些实施例中,在动作206处生成的初始数据流图可以包含表示数据处理操作的节点,该数据处理操作不在动作204处生成的查询计划中。相反地,在一些实施例中,在动作206处生成的初始数据流图可以不包含表示数据处理操作的节点,该数据处理操作在动作204处生成的查询计划中。此类情况可能是由于在根据查询计划生成数据流图的过程期间可能进行的各种优化而引起的。在一些实施例中,在动作206处生成的初始数据流图可以包含表示除了正在数据库计算机系统(例如,关系数据库管理系统)上进行的数据处理操作之外的数据处理操作的节点。

[0089] 在一些实施例中,查询计划和数据流图可以在不同类型的数据结构中实施。例如,在一些实施例中,查询计划可以在有向图中实施,在该有向图中,每个节点具有单个父节点(例如,诸如二叉树等树),而数据流图可以在有向非循环图中实施,该有向非循环图可能具有带有多个父节点的至少一个节点。

[0090] 接下来,过程200进行到动作207,在该动作中,更新初始数据流图以获得更新的数据流图。这可以以多种方式中的任何一种来完成。例如,在所展示的实施例中,在动作208处选择数据流图优化规则。接下来,执行过程200的数据处理系统标识将在动作208处标识初始数据流图的将被应用优化规则的部分。在动作212处,将所选择的优化规则应用于图的所标识的部分。接下来,过程200进行到决策框214,在该决策框处,确定是否存在要应用于数据流图的至少又一个部分的优化规则。如果确定存在要应用于图的至少又一个部分的优化规则(例如,在动作208处选择的优化规则可以被应用于图的不同于在动作210处标识的部分的另一部分,不同的优化规则可以被一起选择等等),过程200返回到动作208。否则,过程200进行到动作216。

[0091] 在一些实施例中,针对在动作208处选择的每个特定的优化规则,数据处理系统可

以标识数据流图的适用所选择的优化规则的部分,并将优化规则应用于所标识的部分。一旦所有这种部分都被标识出,就可以选择不同的优化规则。然而,也可以选择先前应用过的优化规则,使得可以考虑将同一优化规则多次应用于数据流图(与贪婪地选择优化规则并且在应用过一次后不再使用该规则的方法相比,这可能产生更优的数据流图)。可以以任何合适的顺序来选择优化规则,因为本文所描述的技术的方面在这方面不受限制。作为一个示例,在进行去重优化之后,可以去除表示冗余操作的任何节点,并且可以去除任何空节点。在去除空节点后,可以进行宽度减小优化操作等。

[0092] 在一些实施例中,动作208和210的顺序可以改变。在这样的实施例中,数据处理系统可以首先标识数据流图的一部分,并且然后选择可以应用于所标识的数据流图部分的优化规则。在这样的实施例中,可以以拓扑方式对初始数据流图进行排序,并且可以遍历(例如,从左到右)以拓扑方式排序的图,以标识可以被应用优化规则的特定部分。

[0093] 在一些实施例中,数据处理系统可以采用数据流子图模式匹配语言来标识初始数据流图的将被应用一个或多个优化规则的一个或多个部分。数据流子图模式匹配语言可以包括用于标识数据流图中的子图的特定类型的一个或多个表达。在一些实施例中,执行过程200的数据处理系统可以被配置为使用子图模式匹配语言中的表达来标识数据流图的将被应用一个或多个优化规则的部分。特定表达可以促进标识要被应用特定优化规则或多个优化规则的一个或多个部分。在一些实施例中,当数据流图优化器(例如,图优化器110)配置有一个或多个新的优化规则时,图优化器可以配置有以子图模式匹配语言编写的一个或多个新的表达,以促进标识数据流图的可以被应用(多个)新的优化规则的部分。

[0094] 例如,模式匹配语言可以包括用于标识至少阈值长度(例如,至少两个、三个、四个、五个等)的一系列节点的表达,这些节点表示可以使用组合操作优化规则由图中的单个节点组合并表示的一系列相应的计算。标识这种模式可以促进组合操作优化规则的应用,该应用将在以下包括参照图5A至图5D进一步描述。一个这种表达的非限制性示例是“A→B→C→D”,这可以帮助标识一系列可以组合的四个连续的数据处理操作。

[0095] 作为另一个示例,模式匹配语言可以包括用于标识数据流图的其中某些类型的节点可以与其他节点交换的部分的表达。这可以促进将多种不同类型的优化规则应用于数据流图。当数据处理系统确定可以在不改变处理结果的情况下更改数据流图中一个或多个节点的顺序时,这允许数据处理系统考虑对数据流图的结构进行改变(如通过交换操作可获得的自由度所允许的),以便标识可以被应用优化规则的部分。作为考虑基于交换的更改的结果,一个或多个优化规则可以变得适用于图的否则将不适用该一个或多个规则的部分。

[0096] 例如,优化规则可能涉及标识初始数据流图中的表示相应的排序操作的两个相邻节点,其中,第二排序操作会使第一操作的效果无效,因此应删除第一操作(参见例如如图3B和图3C所示的示例)。根据定义,这种优化规则将不会被应用于不具有表示排序操作的相邻节点的数据流图。然而,如果表示第一排序操作的第一节点要与一个或多个其他节点交换,则可以改变第一节点与一个或多个其他节点中的至少一个节点的顺序,使得表示第一排序操作的第一节点与表示第二排序操作的第二节点相邻地放置。作为以这种方式对节点进行交换的结果,去除冗余的第一排序操作的优化规则可以被应用于数据流图。

[0097] 因此,在一些实施例中,子图匹配语言可以包括用于标识数据流图的符合数据流图中的节点的顺序可以改变的情况的子图的一个或多个表达。作为一个示例,表达“A*→

(\dots) \rightarrow B” (其中,A和B中的每一个可以是任何合适的数据处理操作,诸如排序、合并等)可以用于找到数据流图中的具有以下各项的部分:节点“A”(即,表示操作“A”的节点)和节点B(表示操作B)、以及在节点A与节点B之间的与节点A交换的一个或多个节点(例如,在改变节点的顺序情况下,由这些节点进行的处理的结果不会改变)。如果这样的部分被标识出,则可以通过将节点A移动成与节点B相邻来改变数据流图,以获得部分“AB”。作为特定示例,如果数据流图具有节点ACDB,并且操作A与操作C和D交换,则可以将数据流图更改为“CDAB”。进而,数据处理系统可以考虑优化规则是否适用于部分“AB”。例如,如果操作A是排序并且操作B是排序,则数据处理系统可以尝试确定是否可以如图5B的示例中那样用的单个排序来替换这两个排序。

[0098] 作为另一个示例,“ $A\rightarrow(\dots)\rightarrow B^*$ ”可以用于找到数据流图的具有以下各项的部分:节点A、第二节点B、以及在这两个节点之间的与节点B交换的一个或多个节点。作为特定示例,如果数据流图具有节点ACDB,并且操作B与操作C和D交换,则可以将数据流图更改为“ABCD”。进而,数据处理系统可以考虑优化规则是否适用于部分“AB”。

[0099] 作为另一个示例,表达“ $A\rightarrow(\dots)\rightarrow B^{**}$ ”可以用于找到数据流图的具有以下各项的部分:节点A、节点B、以及在节点A与节点B之间的不与节点B交换的一个或多个节点(例如,C和D)。在这种情况下,系统可能会尝试“强行”进行交换,其中,如果可能的话,将节点C和D推到节点A的左侧。作为特定示例,如果数据流图具有节点ACEDB,并且操作B与操作E交换但不与操作C和D交换,则可以将数据流图更改为“CDABE”——将B与E交换并将C和D推到A的左侧。

[0100] 作为又一个示例,表达“ $A^{**}\rightarrow(\dots)\rightarrow B$ ”可以用于找到数据流图的具有以下各项的部分:节点A、节点B、以及在节点A与节点B之间的不与节点A交换的一个或多个节点(例如,C和D)。在这种情况下,系统可能会尝试“强行”进行交换,其中,如果可能的话,将节点C和D推到节点B的右侧。作为特定示例,如果数据流图具有节点ACEDB,并且操作A与操作E交换但不与操作C和D交换,则可以将数据流图更改为“EABCD”——将A与E交换并将C和D推到A的右侧。

[0101] 应当理解,以上所描述的子图匹配语言的表达的示例是说明性的。在一些实施例中,作为以上所描述的示例的补充或替代方案,一个或多个其他表达也可以是子图匹配语言的一部分。

[0102] 在一些实施例中,在动作207处,当从初始数据流图生成更新的数据流图时,可以应用多种类型的优化规则中的任何一种或多种。例如,将优化规则应用于初始数据流图可以涉及去除一个或多个冗余数据处理操作、去除一个或多个未引用的数据处理操作、进行一次或多次强度降低优化、进行一次或多次组合操作优化、进行一次或多次宽度减小优化和/或进行一次或多次去重优化。

[0103] 在一些实施例中,优化规则可以体现在程序代码中,该程序代码在被执行时,使得对数据流图进行对应的优化。例如,用于去除冗余的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(从将被应用该规则的数据流图中)去除表示被确定为冗余的数据处理操作的至少一个节点。在图3A至图3C中展示了将优化规则应用于数据流图以去除一个或多个冗余数据处理操作的示例,如以下更详细地描述的。

[0104] 作为另一个示例,用于进行强度降低的优化规则可以体现在这样的程序代码中该

程序代码在被执行时,使得(在将被应用该规则:的数据流图中)表示第一数据处理操作的第一节点(例如,表示排序数据处理操作的节点)被表示比该第一数据处理操作更弱类型的第二数据处理操作的第二节点(例如,表示组内排序数据处理操作的节点)替换。在图4A和图4B中展示了将优化规则应用于数据流图以进行强度降低优化的示例,如以下更详细地描述的。

[0105] 作为另一个示例,用于进行组合操作优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(在将被应用该规则的数据流图中)表示多个操作的多个节点被表示该多个操作的组合的单个节点替换。在图5A至图5D中展示了将优化规则应用于数据流图以进行组合操作优化的示例,如以下更详细地描述的。

[0106] 作为又一示例,用于去除一个或多个未引用的数据处理操作的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得(从将被应用该规则的数据流图中)去除表示结果未被引用和/或未被使用的数据处理操作(例如,由于在后续处理中不需要或不依赖排序所产生的顺序,因此未被引用的排序数据处理操作)的至少一个节点。在图6中展示了将这种优化规则应用于数据流图的示例,如以下更详细描述的。

[0107] 作为另一个示例,用于进行宽度减小优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得一些数据(例如,一列或多列数据、多行数据等)在进行后续操作之前在图中的特定部分处被删除,因为这些数据(即所删除的数据)在后续操作中不会被使用并且不需要作为处理的一部分进行传播。在图7中展示了将这种优化规则应用于数据流图的示例,如以下更详细描述的。

[0108] 作为另一个示例,用于进行去重优化的优化规则可以体现在这样的程序代码中:该程序代码在被执行时,使得数据流图的将被应用该规则的不同分支合并。在图8A和图8B中展示了将这种优化规则应用于数据流图的示例,如以下更详细描述的。

[0109] 作为另一个示例,用于进行串行到并行优化的优化规则可以体现在这样程序代码中:该程序代码在被执行时,使得被串行地进行的处理被并行地进行。在图9中展示了将这种优化规则应用于数据流图的示例,如以下更详细描述的。

[0110] 应当理解,以上所描述的优化规则和优化是说明性的非限制性示例。作为过程200的一部分,作为以上描述的优化规则和/或优化的替代或补充方案,可以将一个或多个其他优化规则和/或优化应用于初始数据流图。

[0111] 接下来,过程200进行到动作216,在该动作中,输出更新的数据流图。在一些实施例中,在动作216处,可以存入更新的数据流图(例如,将其存储在非易失性存储器中)以供后续使用。

[0112] 作为进行存储的补充或替代方案,可以执行更新的数据流图。在执行更新的数据流图的一些实施例中,在过程200的动作218处,将处理布局分配给更新的数据流图的一个或多个节点。用于表示数据处理操作的节点的处理布局可以指定要使用多少个计算设备来进行数据处理操作,并且可以标识要用于进行数据处理操作的特定计算设备。这可以以任何合适的方式来完成,包括通过使用2018年3月29日提交的名称为“Systems and Methods for Performing Data Processing Operations Using Variable Level Parallelism[用于使用可变水平并行性来进行数据处理操作的系统和方法]”的美国专利申请号15/939,829中描述的任何布局分派技术,该美国专利申请通过引用以其全文并入本文。在一些实施

例中,可以较早地(例如,在动作207期间)确定是要使用单个设备还是多个计算设备来处理节点(例如,是否要应用并行处理以及要采用何种水平的并行),其中,将在动作218处分配要用于计算的特定计算设备。

[0113] 在动作218处分配了处理布局之后,可以执行更新的数据流图。例如,当处理200由数据处理系统进行时,数据处理系统100可以使用图执行引擎115来执行更新的数据流图。在一些实施例中,在动作207处生成的更新的数据流图一经生成就可以立即执行,而无需任何用户输入。在其他实施例中,更新的数据流图可以在动作207处生成,但是其执行可以仅响应于要这样做的命令而开始,该命令可以由用户通过界面或由另一计算机程序(例如,通过API调用)来提供。

[0114] 应当理解,过程200是说明性的,并且存在变型。例如,在一些实施例中,可选动作218和220可以被省略,并且过程200可以在生成并存储了更新的数据流图之后就完成。作为另一个示例,过程200可以用于优化从另一源(例如,另一数据处理系统)提供的数据流图,而不是如所展示的实施例中的情况那样用于优化根据输入SQL查询生成的数据流图。在这样的实施例中,可以省略动作202至204,并且可以在动作206处从另一源提供的数据流图生成初始数据流图。这样的生成可能涉及将接收到的数据流图变换成被配置为与执行过程200的数据处理系统一起使用的数据流图。

[0115] 以下参照图3A至图9提供了将优化规则应用于数据流图的说明性示例。这些附图中所示的每个数据流图都可以是正被优化(例如,作为过程200的动作207的一部分)的较大数据流图的子图。例如,这些附图所示的一个或多个数据流图中的每一个都可以在过程200的动作206处生成的初始数据流图的子图、和/或通过作为动作207的一部分对初始数据流图进行变换而获得的一个或多个数据流图的子图。

[0116] 图3A展示了将优化规则应用于说明性数据流图300以去除一个或多个冗余数据处理操作。如图3A中所示,数据流图300包括表示重新分区数据处理操作的节点302(该节点对数据进行分区以在不同计算设备上并行处理),随后是表示串行化操作的节点304(该节点操作以组合所有数据,以便由单个计算设备来进行串行处理)。由于后续的串行化命令将使重新分区的效果无效,因此无需进行重新分区数据处理操作。因此,数据流图优化器(例如,数据处理系统100中的图优化器110)可以去除表示重新分区操作的节点302。结果,数据流图的部分300被变换为部分305。

[0117] 图3B展示了根据本文所描述的技术的一些实施例改变交换数据处理操作的顺序以促进将优化规则应用于另一说明性数据流图。如图3B中所示,数据流图310包括表示排序数据处理操作的节点311(该排序根据键A进行),随后是一个或多个节点(未示出),随后是表示另一排序数据处理操作的节点312(该排序根据键B进行)。在该示例中,确定由节点311表示的排序操作是否与由节点311与312之间的节点表示的(多个)数据处理操作交换可以促进将一个或多个优化规则应用于数据流图310。例如,如果由节点311表示的排序操作与由节点311与312之间的节点表示的(多个)数据处理操作交换,则改变节点的顺序允许将节点311和312彼此相邻地放置(如数据流图313中所示的)。进而,这允许考虑是否可以将一个或多个优化规则应用于所产生的图313(在改变由节点311表示的排序操作在图中出现的位置的顺序之前,这些规则可能不适用于数据流图310)。例如,在这种情况下,根据键B进行的后续重新排序(节点312)会使根据键A进行的排序(节点311)的效果无效。因此,由节点311

表示的排序操作是不必要的,并且可以去除该节点(如图3C中所示),从而产生数据流图314。

[0118] 图3D展示了将优化规则应用于另一说明性数据流图320以去除冗余数据处理操作。如图3D中所示,数据流图320包括表示根据键A进行的排序操作的节点322。然而,当已经根据该相同的键对数据进行过排序时,数据流图优化器可以去除表示根据键A进行的排序操作的节点322,从而产生数据流图323。在图3A至图3D中所示的所有示例中,去除冗余节点会产生与不去除此类节点的情况相比在执行时消耗更少的计算资源的数据流图。

[0119] 图4A展示了将优化规则应用于说明性数据流图400以进行强度降低优化并获得数据流图402。如图4A中所示,数据流图400包含节点401,该节点表示用于根据主要键A对传入数据进行排序(例如,按姓氏进行排序)并且然后根据次要键进行排序(例如,按名字对具有相同姓氏的人进行排序)的排序操作。然而,当图优化器检测到已经根据键A对传入节点401中的数据进行了排序(例如,已经按姓氏进行了排序)时,图优化器可以通过用组内排序操作(由图402中的节点403表示)来替换排序操作以进行强度降低优化,从而降低排序操作的强度但获得相同的结果并避免了进行不必要的计算。

[0120] 图4B展示了将优化规则应用于说明性数据流图410以进行另一强度降低优化并获得数据流图412。如图4B所示,数据流图410包含表示要根据主要键A和次要键B进行的上卷操作的节点411。然而,当图优化器检测到已经根据键A对传入节点411中的数据进行了排序时,图优化器可以通过用分组上卷操作(由图412中的节点413表示)来替换上卷操作以进行强度降低优化,从而降低上卷操作的强度但获得相同的结果并避免了进行不必要的计算。

[0121] 图5A展示了将优化规则应用于说明性数据流图500以进行组合操作优化。如图5A中所示,数据流图500包括节点502、504和506的序列,每个节点表示相应的计算。在一些实施例中,在数据流图的执行期间,由分开的节点表示的数据处理操作可以由在一个或多个计算设备上运行的不同进程来执行。图优化器可以通过用单个节点(例如,数据流图505中的节点508)来替换三个节点的序列来进行组合操作优化,使得所有操作都由在单个计算设备上执行的单个进程来进行,这减少了进程间(以及潜在的设备间)通信的开销。

[0122] 图5B展示了将优化规则应用于另一说明性数据流图510以进行组合操作优化。如图5B中所示,数据流图510包括表示使用键A1和B1对数据集A与数据集B进行的联接操作的节点512,随后是表示使用键A1和C1对由节点512表示的联接操作的输出与数据集C进行的联接操作的节点514。在该示例中,图优化器可以通过以下操作来进行组合操作优化:用表示使用键A1、B1和C1对数据集A、B和C进行的联接操作的单个节点(如由数据流图515中的节点516所示)来替换表示相应的联接操作的两个分开的节点。以这种方式,联接处理由在单个计算设备上执行的单个进程进行,这减少了进程间(以及潜在的设备间)通信的开销。

[0123] 图5C展示了将优化规则应用于另一说明性数据流图520以进行组合操作优化。如图5C中所示,数据流图520包括表示根据键A进行的过滤操作的节点522;随后是表示根据键B进行的另一过滤操作的节点524。在该示例中,图优化器可以通过以下操作来进行组合操作优化:用表示根据键A和B两者进行过滤的过滤操作的单个节点(如由数据流图525中的节点526所示)来替换表示相应的过滤操作的两个分开的节点522和524。以这种方式,过滤由在单个计算设备上执行的单个进程进行,这减少了进程间(以及潜在的设备间)通信的开销。

[0124] 图5D展示了将优化规则应用于另一说明性数据流图530以进行组合操作优化。如图5D中所示,数据流图530包括表示根据键A进行的过滤操作的节点532,随后是表示上卷操作的节点534。在该示例中,图优化器可以通过以下操作来进行组合操作优化:用表示对根据键A选择的输入进行处理的上卷操作的单个节点(如由数据流图535中的节点536所示)来替换两个分开的节点522和524。以这种方式,在单个设备上执行的单个进程可以进行上卷操作和过滤操作的等效操作,这减少了进程间(以及潜在的设备间)通信的开销。

[0125] 图6展示了将优化规则应用于说明性数据流图600以去除不必要的数据处理操作。如图6中所示,数据流图600包括表示根据键A进行的排序操作的节点602,随后是表示重新格式化命令的节点604,随后是表示无序写入命令的606。在写入命令是无序的并且因此由节点602表示的排序操作对数据施加的顺序通过写入操作后不会被保留的情况下,图优化器可以去除表示排序操作的节点,这会产生数据流图605。

[0126] 图7展示了根据本文所描述的技术的一些实施例将优化规则应用于说明性数据流图700以进行宽度减小优化。如图7中所示,数据流图700包括将数据列A的值设置为存储在列B和列C中的数据的逻辑“或”的节点702,随后是表示对由节点702提供的数据列A进行的过滤操作的节点704和表示使用键D的排序操作的节点706。在过滤操作之后,数据列A在下游计算中不会被使用。因此,在一些实施例中,图优化器可以通过去除数据列A中的数据使得不进一步传播这些数据来进行宽度减小优化(例如,通过引入用于将这些数据从沿着节点之间的链路移动的数据中删除的节点,如由在数据流图705中节点708所示)。这减少了将数据携带通过数据流图705中的节点704下游的后续计算所需的计算资源的量(例如,通过减少所利用的网络、存储器和处理资源)。

[0127] 图8A展示了将优化规则应用于说明性数据流图800以进行去重优化。如图8A中所示,数据流图800包括节点802和804,这两个节点都表示从同一基础文件(A.dat)的读取操作。图优化器可以通过用单个节点(例如,如由数据流图805中的节点806所示)来替换这两个节点以进行去重优化,使得数据仅被读取一次。即使该数据随后被数据流图805中的不同部分使用,不读取同一数据两次也减少了用于访问数据的计算资源的量。

[0128] 图8B展示了将优化规则应用于说明性数据流图811以通过拉链式组合进行去重优化。如图8B中所示,数据流图811包括节点810,该节点表示从称为“A.dat”的文件读取数据的数据处理操作。然后,该数据在两个不同的分支中被处理。第一分支包含节点812(提取数据列“A.f”并将该列中的数据加上“1”)和节点814(根据列“A.k”对数据进行排序)。第二分支包含节点816(提取数据列“A.g”,并将该列中的数据乘以3)和节点818(根据“A.k”列对数据进行排序)。在图8A的示例中,去重涉及删除进行相同处理(从相同文件读取相同数据)的节点。然而,在图8B的示例中,由图811的不同分支所进行的处理是不同的。但是,在不同分支中进行的处理相似到足以使这些分支可以被交叉合拢(fold)到一起成为单个路径。该交叉合拢可以通过将路径从左到右组合在一起(类似于拉链连接两侧的方式)来进行。

[0129] 在该示例中,图优化器可以更改图811,使得由节点812和816表示的计算被串行地(而不是并行地)进行。在数据流图821中示出了结果。接下来,图优化器可以更改图821,使得表示相应排序操作的节点826和828被组合成单个节点836。在数据流图831中示出了结果。如可以看到的,图831中的节点数量相对于图811有所减少,并且对相同数据的处理可以在相同位置进行,从而减少了所需的计算资源。另外,例如,通过将由节点832和834表示的

操作组合到单个节点中,可以将进一步的优化应用于所产生的图831。

[0130] 图9展示了将优化规则应用于说明性数据流图900以进行串行到并行优化。如图9中所示,数据流图900包括:表示应用于先前在k个(多个)计算设备上处理的数据的串行化操作的节点902、以及表示(根据键A)进行排序操作的节点904。在一些实施例中,数据流图优化器可以更改数据流图,使得应用于串行化的数据的一个或多个操作替代地以并行化的方式被应用。在该示例中,数据流图优化器可以更改图900以去除串行化操作,并允许以k个并行的方式应用排序(如使用数据流图905中的节点906所示)。然后,可以使用合并操作来组合这k个并行排序的结果,如使用数据流图905中的节点908所示。

[0131] 图10A展示了根据本文所描述的技术的一些实施例的初始数据流图1000。图10B展示了根据本文所描述的技术的一些实施例通过迭代地将优化规则应用于图10A所示的初始数据流图获得的更新的数据流图1050。通过将图10A的初始数据流图1000与图10B的更新的数据流图1050进行比较可以看出,更新的数据流图具有比初始数据流图更少的节点和链路,并且可以比初始数据流图更高效地被执行。以下详细描述了根据本文所描述的技术的一些实施例的应用于初始数据流图1000的多种优化。

[0132] 在图10A和图10B的说明性示例中,向图10A的初始数据流图1000应用多次组合操作优化,以便生成图10B的更新的数据流图1050。例如,在更新的数据流图1050的生成期间,使用组合操作优化来组合由节点1002表示的数据处理操作(“expr”,其可以是用于对流过节点1002的数据进行任何合适的计算的表达,并且在这种情况下该计算是对数据进行格式化以供输出)与由节点1004表示的写入数据处理操作(“写入文件”),该更新的数据流图包含被配置为进行这两个数据处理操作的节点1054。作为另一个示例,在更新的数据流图1050的生成期间,使用组合操作优化来组合由节点1006表示的数据处理操作(“expr”)和由节点1008表示的过滤数据处理操作(“过滤”),该更新的数据流图包含被配置为进行这两个数据处理操作的节点1056。作为又一个示例,在更新的数据流图1050的生成期间,使用组合操作优化来组合由节点1010表示的数据处理操作(“expr”)和由节点1012表示的写入数据处理操作(“写入文件”),该更新的数据流图包含被配置为进行这两个数据处理操作的节点1058。如本文所讨论的,在一些实施例中,因为与单个数据流图节点相关联的数据处理操作是由在单个计算设备上执行的单个进程进行的,所以组合节点减少了进程间(以及潜在的设备间)通信的开销,从而提高了数据处理系统的性能。

[0133] 在图10A和图10B的说明性示例中,向图10A的初始数据流图1000应用多次用于减少冗余的优化,以便生成图10B的更新的数据流图1050。例如,在更新的数据流图1050中,由初始数据流图1000中的节点1030、1032和1034表示的排序操作被去除,并且没有任何对应项。这些排序操作被去除是因为已经通过分别经由节点1022、1026和1028应用的排序操作对传入这些节点中的数据进行了排序。经由节点1022、1026和1028处的处理所施加的排序顺序会被后续数据处理操作(例如,排序上卷操作)保留,并且数据的排序顺序的保持使得无需在节点1030、1032和1034处进行进一步的排序。因此,在更新的数据流图1050的生成期间去除节点1030、1032和1034。另一方面,保留由节点1022、1026和1028表示的排序操作。例如,初始数据流图1000中的节点1022和1026对应于更新的数据流图1050中的节点1070和1072。相比而言,由于在先前的全外部合并联接操作不保留排序顺序,因此由节点1036表示的排序操作不是冗余的,所以保留该排序操作(节点1074是更新的数据流图1050中的对应

节点)。

[0134] 图10A和10B还展示了去除不必要的数据处理操作的另一示例。如图10A中所示,初始数据流图1000包括表示排序操作的节点1024,最后接着的是表示无序写入命令的节点1004。在写入命令是无序的并且因此由节点1024表示的排序操作对数据施加的顺序通过写入操作后不会被保留的情况下,可以去除表示排序操作的节点1024。如从图10B可以看出的,没有对应于节点1024的排序操作——该操作已被去除。

[0135] 作为去除不必要的数据处理操作的又一个示例,初始数据流图1000包括由节点1040、1045、1046和1047表示的布局 and 分区操作,这些节点是冗余的,因为在它们中的每一个之前存在相应的布局 and 分区节点,并且布局没有被另外地改变。因此,从初始数据流图1000去除这些节点——数据流图1050中不存在对应的节点。在应用于初始数据流图1000的相关优化中,用由节点1088表示的聚集操作来替换由节点1048表示的布局 and 分区操作。相比而言,保留由于节点1044表示的布局 and 分区数据处理操作,因为该节点不是冗余的;该节点负责建立最终的分区和布局,并产生最终图1050中的节点1080“按照键分区”。

[0136] 图10A和图10B还展示了宽度减小优化的示例。如可以看出的,通过引入图10B的更新的数据流图1050中所示的“重新格式化_隐式”节点1064,减少了使用由节点1014表示的读取数据处理操作所处理的数据的列数。在该示例中,图优化器通过确定由节点1012表示的写入操作要写出哪些列来检测读入数据中的后续将不会被使用的多个数据列。

[0137] 图10C展示了根据本文所描述的技术的一些实施例的图10A的初始数据流图1000的另一视图。在该视图中,数据流图节点的标签已被其首字母缩写替换。类似地,图10D展示了图10B的更新的数据流图的另一视图,在该视图中,数据流图节点的标签已被其首字母缩写替换。

[0138] 图11展示了可以在其上实施本文所描述的技术的合适计算系统环境1100的示例。计算系统环境1100仅是合适的计算环境的一个示例,并且不旨在对本文所描述的技术的使用或功能的范围提出任何限制。也不应当将计算环境1100解释为具有与示例性操作环境1100中展示的部件中的任何一个或组合相关的任何依赖性 or 要求。

[0139] 本文所描述的技术与许多其他通用或专用计算系统环境或配置一起操作。可以适合与本文所描述的技术一起使用的众所周知的计算系统、环境和/或配置的示例包括但不限于:个人计算机、服务器计算机、手持或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络PC、小型计算机、大型计算机、包括上述系统 or 设备中任一个的分布式计算环境等。

[0140] 计算环境可以执行计算机可执行指令,诸如程序模块。通常,程序模块包括进行特定任务 or 实施特定抽象数据类型的例程、程序、对象、组件、数据结构等。本文所描述的技术还可以在分布式计算环境中实践,其中,任务由通过通信网络链接的远程处理设备进行。在分布式计算环境中,程序模块可以位于包括存储器存储设备的本地和远程计算机存储介质中。

[0141] 参照图11,用于实施本文所描述的技术的示例性系统包括呈计算机1110形式的通用计算设备。计算机1110的部件可以包括但不限于处理单元1120、系统存储器1130、以及将包括系统存储器的各种系统部件耦合到处理单元1120的系统总线1121。系统总线1121可以是包括存储器总线或存储器控制器、外围总线和使用各种总线架构中的任何架构的本地总

线的若干类型的总线结构中的任何总线结构。通过举例而非限制,此类架构包括工业标准架构 (ISA) 总线、微通道架构 (MCA) 总线、增强ISA (EISA) 总线、视频电子标准协会 (VESA) 本地总线、以及外围部件互连 (PCI) 总线 (也称为夹层 (Mezzanine) 总线)。

[0142] 计算机1110通常包括各种计算机可读介质。计算机可读介质可以是可由计算机1110访问的任何可用介质,并且包括易失性和非易失性介质、可移除和不可移除介质。通过举例而非限制,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以任何方法或技术实施的用于存储诸如计算机可读指令、数据结构、程序模块、或其他数据等信息的易失性和非易失性介质、可移除和不可移除介质。计算机存储介质包括但不限于RAM、ROM、EEPROM、闪存存储器或其他存储器技术、CD-ROM、数字通用盘 (DVD) 或其他光盘存储装置、磁带盒、磁带、磁盘存储装置或其他磁存储设备、或者可用于存储期望信息并可由计算机1010访问的任何其他介质。通信介质通常以诸如载波或其他传输机制等已调制数据信号来实施计算机可读指令、数据结构、程序模块或其他数据,并且包括任何信息递送介质。术语“已调制数据信号”是指以编码信号中的信息的这种方式使其特征中的一个或多个被设置或改变的信号。通过举例而非限制,通信介质包括有线介质 (诸如有线网络或直接有线连接) 以及无线介质 (诸如声学、RF、红外线及其他无线介质)。上述中的任一个的组合也应包括在计算机可读介质的范围内。

[0143] 系统存储器1130包括呈易失性和/或非易失性存储器形式的计算机存储介质,诸如只读存储器 (ROM) 1131和随机存取存储器 (RAM) 1132。包含诸如在启动期间帮助在计算机1110内的元件之间传输信息的基本例程的基本输入/输出系统1133 (BIOS) 通常存储在ROM 1131中。RAM 1132通常包含可由处理单元1120立即访问和/或当前正在操作的数据和/或程序模块。通过举例而非限制,图11展示了操作系统1134、应用程序1135、其他程序模块1036和程序数据1137。

[0144] 计算机1110还可以包括其他可移除/不可移除、易失性/非易失性计算机存储介质。仅通过举例,图11展示了读取或写入不可移除的非易失性磁介质的硬盘驱动器1141、读取或写入可移除的非易失性存储器1152 (诸如闪存存储器) 的闪存驱动器1151、以及读取或写入可移除的非易失性光盘1156 (诸如CD ROM或其他光学介质) 的光盘驱动器1155。可以用于示例性操作环境中的其他可移除/不可移除、易失性/非易失性计算机存储介质包括但不限于磁带盒、闪存卡、数字通用盘、数字录像带、固态RAM、固态ROM等。硬盘驱动器1141通常通过诸如接口1140等不可移除存储器接口连接至系统总线1121,并且磁盘驱动器1151和光盘驱动器1155通常由诸如接口1150等可移除存储器接口连接至系统总线1121。

[0145] 以上所讨论并在图11中展示的驱动器及其相关联的计算机存储介质为计算机1110提供了对计算机可读指令、数据结构、程序模块和其他数据的存储。例如,在图11中,硬盘驱动器1141被展示为存储操作系统1144、应用程序1145、其他程序模块1146和程序数据1147。注意,这些部件与操作系统1134、应用程序1135、其他程序模块1136和程序数据1137可以是相同或不同的。这里给予了操作系统1144、应用程序1145、其他程序模块1146和程序数据1147不同的编号,以说明至少它们是不同的副本。用户可以通过诸如键盘1162和定点设备1161 (通常称为鼠标、轨迹球或触模板) 等输入设备将命令和信息输入到计算机1110中。其他输入设备 (未示出) 可以包括麦克风、操纵杆、游戏手柄、圆盘式卫星天线、扫描仪等。这些和其他输入设备通常通过耦合到系统总线的用户输入接口1160连接到处理单元

1120,但可以由诸如并行端口、游戏端口或通用串行总线(USB)等其他接口和总线结构来连接。监视器1191或其他类型的显示设备也经由诸如视频接口1190等接口连接至系统总线1121。除了监视器之外,计算机还可以包括可通过输出外围接口1195连接的其他外围输出设备,诸如扬声器1197和打印机1196。

[0146] 计算机1110可以使用与一个或多个远程计算机(诸如远程计算机1180)的逻辑连接在互联网环境中操作。远程计算机1180可以是个人计算机、服务器、路由器、网络PC、对等设备、或其他常见网络节点,并且通常包括以上关于计算机1110描述的元件中的许多或全部,但是在图11中仅展示了存储器存储设备1181。图11中描绘的逻辑连接包括局域网(LAN)1171和广域网(WAN)1173,但是还可以包括其他网络。这种联网环境在办公室、企业范围计算机网络、内联网和互联网中是普遍的。

[0147] 当在LAN联网环境中使用时,计算机1110通过网络接口或适配器1170连接到LAN1171。当在WAN联网环境中使用时,计算机1110通常包括调制解调器1172或用于通过诸如互联网等WAN 1173建立通信的其他装置。调制解调器1172可以是内置的或外置的,其可以经由用户输入接口1160或其他适当的机制连接到系统总线1121。在联网环境中,关于计算机1110描绘的程序模块或是其部分可以存储在远程存储器存储设备中。通过举例而非限制,图11展示了远程应用程序1185驻留在存储器设备1181上。应当理解,所示出的网络连接是示例性的,并且可以使用在计算机之间建立通信链路的其他手段。

[0148] 如此描述了本发明的至少一个实施例的若干方面之后,应当理解,本领域技术人员会容易想到各种更改、修改以及改进。

[0149] 这种更改、修改以及改进旨在是本披露内容的一部分、并且旨在落入本发明的精神和范围之内。此外,尽管指示了本发明的优点,但应当理解,并非本文所描述的技术的每个实施例都将包括每个描述的优点。一些实施例可以不实施在本文中描述为有利的任何特征,并且在一些情况下,可以实施描述的特征中的一个或多个以实现进一步的实施例。因此,先前的描述和附图仅是通过举例的方式。

[0150] 本文所描述的技术的上述实施例可以以多种方式中的任一种来实施。例如,可以使用硬件、软件或其组合来实施这些实施例。当在软件中实施时,可以在任何合适的处理器或处理器集合上执行软件代码,无论该处理器或处理器集合是设置在单个计算机中还是分布在多个计算机中。此类处理器可以被实施为集成电路(其中,一个集成电路部件中具有一个或多个处理器),包括名称为诸如CPU芯片、GPU芯片、微处理器、微控制器或协处理器等本领域已知的商用集成电路部件。可替代地,处理器可以以定制电路系统(诸如ASIC)或者通过配置可编程逻辑设备产生的半定制电路系统来实施。作为又一替代方案,处理器可以是较大电路的一部分或半导体器件,无论是商用的、半定制的还是定制的。作为具体示例,一些商用微处理器具有多个核,使得这些核中的一个或子集可以构成处理器。然而,处理器可以使用呈任何合适格式的电路系统来实施。

[0151] 进一步地,应当理解,计算机可以实施为多种形式中的任何形式,诸如机架式计算机、台式计算机、膝上型计算机或平板计算机。另外,计算机可以嵌入在通常不被认为是计算机但具有合适的处理能力的设备中,该设备包括个人数字助理(PDA)、智能手机或任何其他合适的便携式或固定式电子设备。

[0152] 而且,计算机可以具有一个或多个输入设备和输出设备。这些设备可以尤其用于

呈现用户接口。可以用于提供用户接口的输出设备的示例包括用于对输出进行视觉呈现的打印机或显示屏,以及用于对输出进行听觉呈现的扬声器或其他声音生成设备。可以用于用户接口的输入设备的示例包括键盘和定点设备(诸如鼠标、触摸板以及数字化平板)。作为另一个示例,计算机可以通过语音识别或以其他可听格式接收输入信息。

[0153] 这种计算机可以由呈任何合适的一个或多个网络来互连,该一个或多个网络包括局域网或广域网,诸如企业网或互联网。这种网络可以基于任何合适的技术并且可以根据任何合适的协议来运行,并且可以包括无线网络、有线网络或光纤网络。

[0154] 而且,本文中概括的各种方法或过程可以被编码为在采用各种操作系统或平台中的任一种的一个或多个处理器上可执行的软件。另外,这种软件可以使用多种合适的编程语言和/或编程或脚本工具中的任一种来编写,并且还可以被编译为可执行的机器语言代码或在框架或虚拟机上执行的中间代码。

[0155] 在这方面,本发明可以被实施为用一个或多个程序进行编码的计算机可读存储介质(或多个计算机可读介质)(例如,计算机存储器、一个或多个软盘、压缩光盘(CD)、光盘、数字视频盘(DVD)、磁带、闪存存储器、现场可编程门阵列或其他半导体器件中的电路配置、或其他有形计算机存储介质),该一个或多个程序在一个或多个计算机或其他处理器上执行时进行实施以上讨论的本发明的各种实施例的方法。如从前文示例将清楚明白,计算机可读存储介质可以将信息保留足够的时间,以便提供呈非暂态形式的计算机可执行指令。这种计算机可读存储介质可以是可运送的,使得其上存储的一个或多个程序可以载入到一个或多个不同的计算机或其他处理器上,以实施如上讨论的本发明的各个方面。如本文使用的,术语“计算机可读存储介质”仅涵盖可被看作是制品(即,制造物品)或机器的非暂态计算机可读介质。可替代地或附加地,本发明可以实施为除了计算机可读存储介质之外的计算机可读介质,诸如传播信号。

[0156] 术语“程序”或“软件”在本文中在一般意义上使用,以指代任何类型的计算机代码或计算机可执行指令集,这些计算机代码或计算机可执行指令集可以被用来对计算机或其他处理器进行编程以实施如上讨论的本发明的各个方面。另外,应该理解,根据此实施例的一个方面,当被执行时进行本发明的方法的一个或多个计算机程序不必驻留在单个计算机或处理器上,而是可以用模块化的方式分布在多个不同的计算机或处理器中以实施本发明的各个方面。

[0157] 计算机可执行指令可以呈由一个或多个计算机或其他设备执行的许多形式,诸如程序模块。通常,程序模块包括进行特定任务或实施特定抽象数据类型的例程、程序、对象、组件、数据结构等。通常,程序模块的功能可以如各个实施例所期望的而组合或分布。

[0158] 而且,数据结构可以以任何合适的形式存储在计算机可读介质中。为了简化说明,可以将数据结构示出为具有通过数据结构中的位置而相关的字段。这种关系同样可以通过为针对字段的存储指派传达字段之间关系的计算机可读介质中的位置来实现。然而,可以使用任何合适的机制来建立数据结构的字段中的信息之间的关系,包括通过使用指针、标签、或建立数据元素之间的关系的其他机制。

[0159] 本发明的各个方面可以单独使用、组合使用、或以前文所述的实施例中未确切讨论的各种布置使用,并且因此在其应用中不局限于其在前文描述中所阐述或附图中所展示的部件的细节和布置。例如,一个实施例中描述的方面可以以任何方式与其他实施例中描

述的方面组合。

[0160] 而且,本发明可以实施为其示例已被提供的一种方法。作为该方法的一部分而进行的动作可以以任何合适的方式排序。因此,可以构造实施例,在这些实施例中,动作以与所展示的顺序不同的顺序进行,这可以包括同时进行一些动作,即使这些动作在说明性实施例中所示出为顺序动作。

[0161] 进一步地,一些动作被描述为由“用户”做出。应当理解,“用户”不必是单个人,并且在一些实施例中,可归因于“用户”的动作可以由个人团队和/或个人与计算机辅助工具或其他机构的组合来进行。

[0162] 在权利要求中使用诸如“第一”、“第二”、“第三”等序数术语来修饰权利要求要素本身并不意味着一个权利要求要素优先于另一个权利要求要素的任何优先级、优先权或顺序,或者进行方法的动作的时间顺序,而是仅作用于将具有某个名称的一个权利要求要素与具有同一名称(但使用序数术语)的另一个要素进行区分的标签,从而区分这些权利要求要素。

[0163] 而且,在本文中使用的措辞和术语是用于描述的目的,而不应当被视为是限制性的。本文使用的“包括(including)”、“包括(comprising)”或“具有(having)”、“包含(containing)”、“涉及(involving)”及其变化形式旨在涵盖其后所列的项和其等效形式、以及附加项。

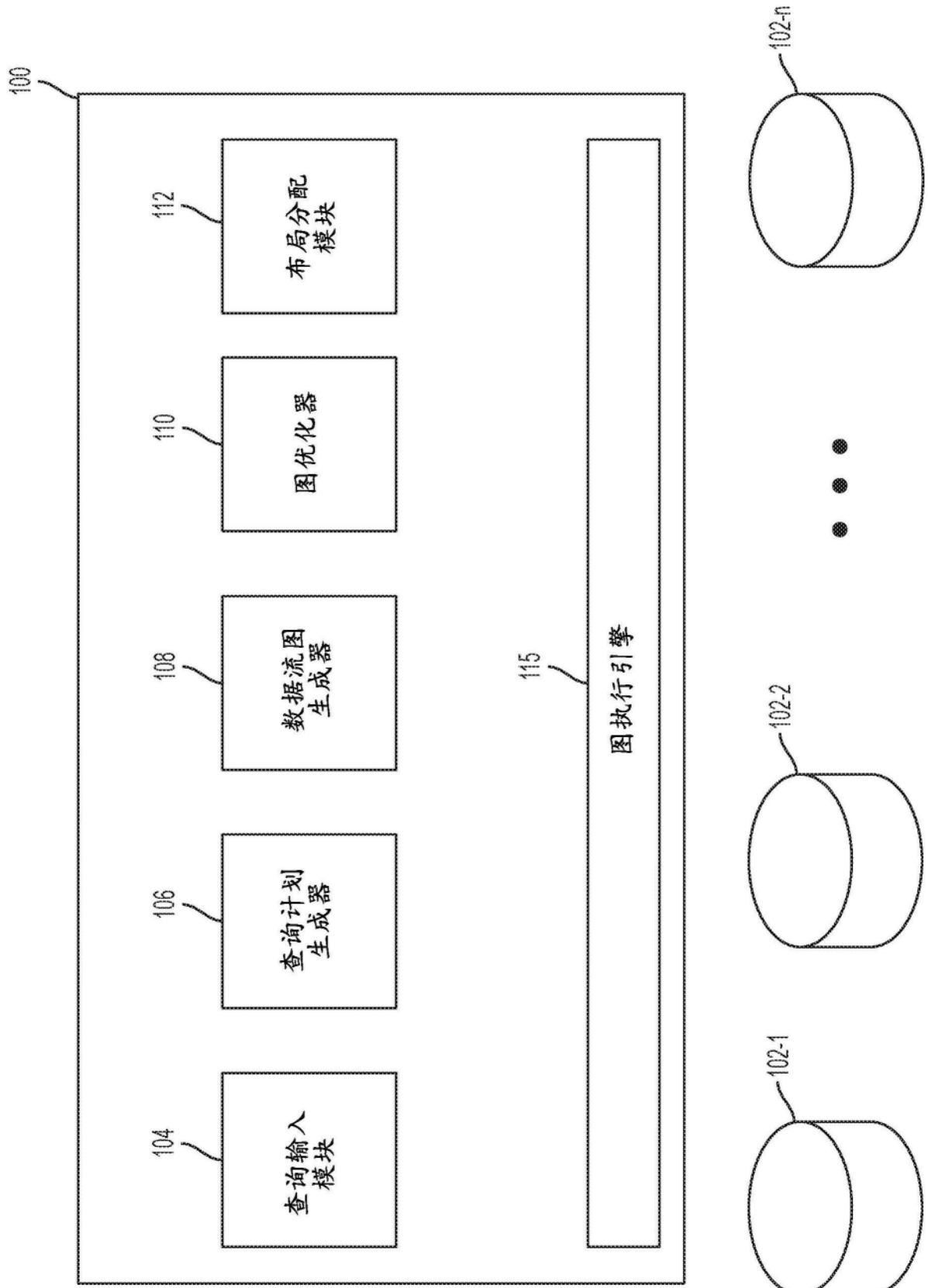


图1A

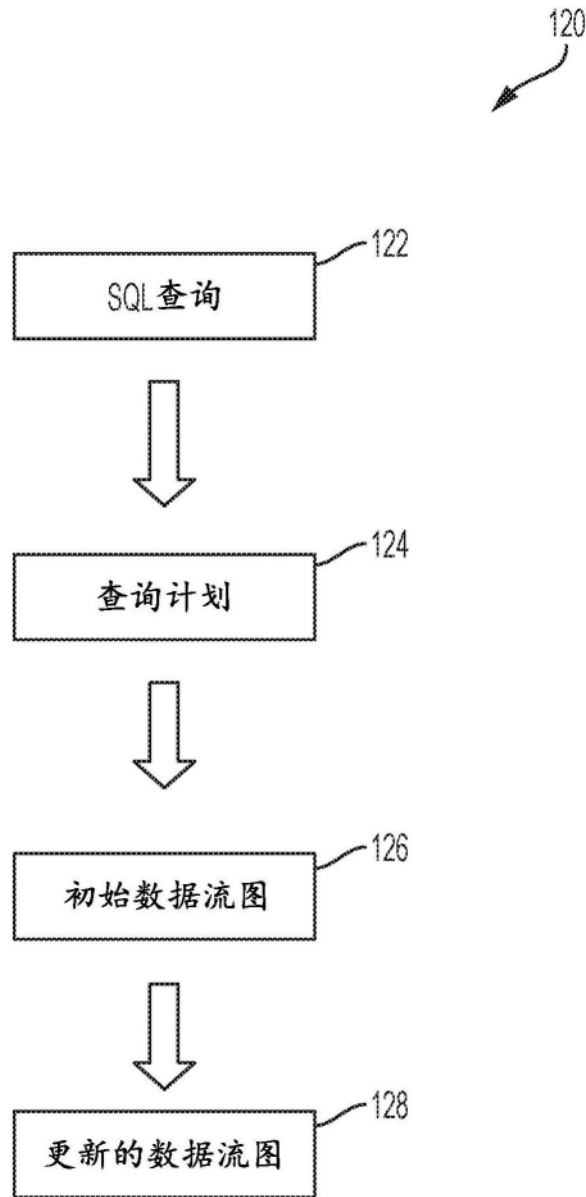


图1B

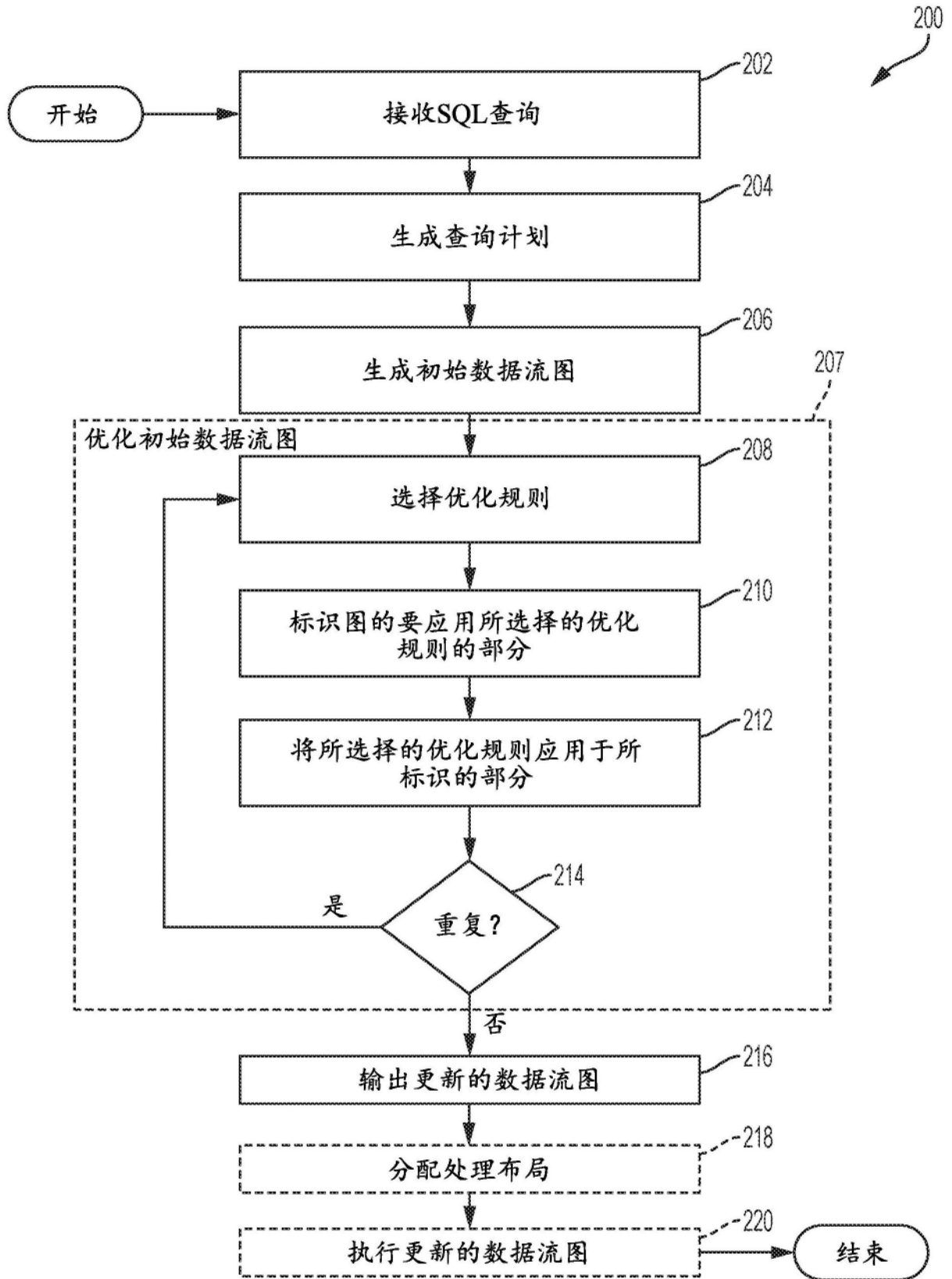


图2

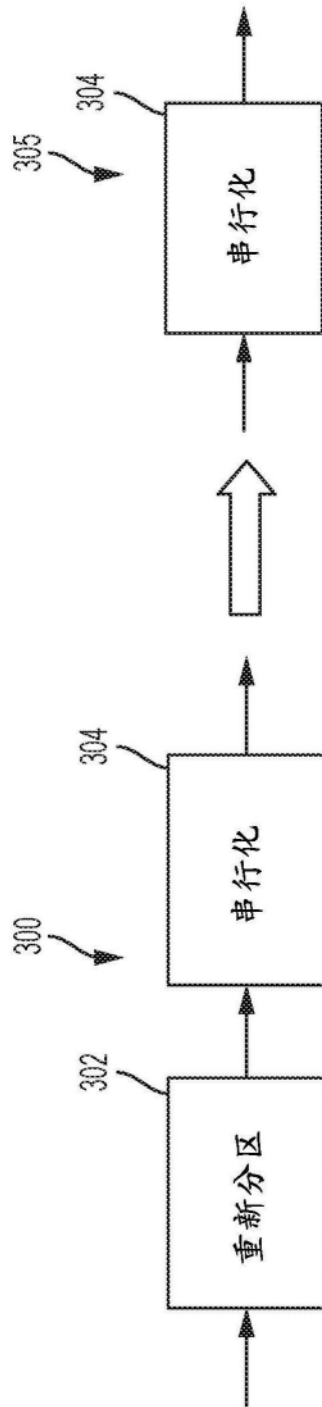


图3A

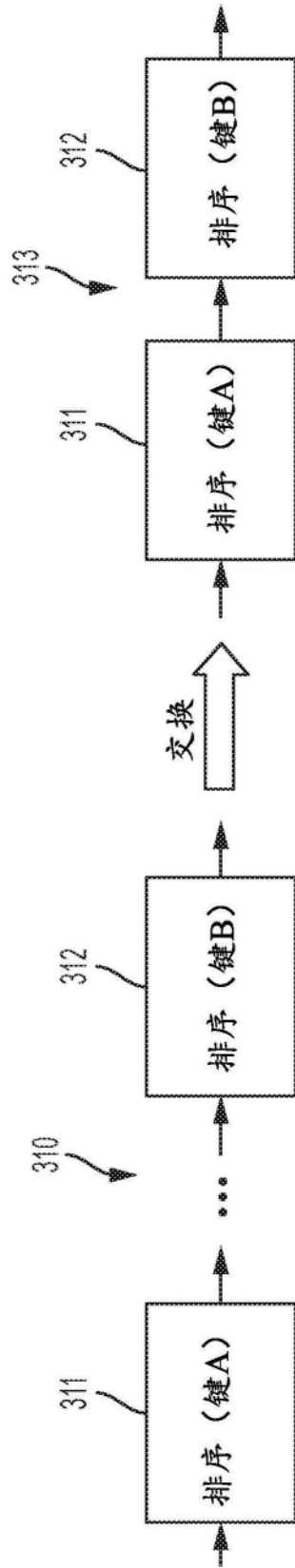


图3B

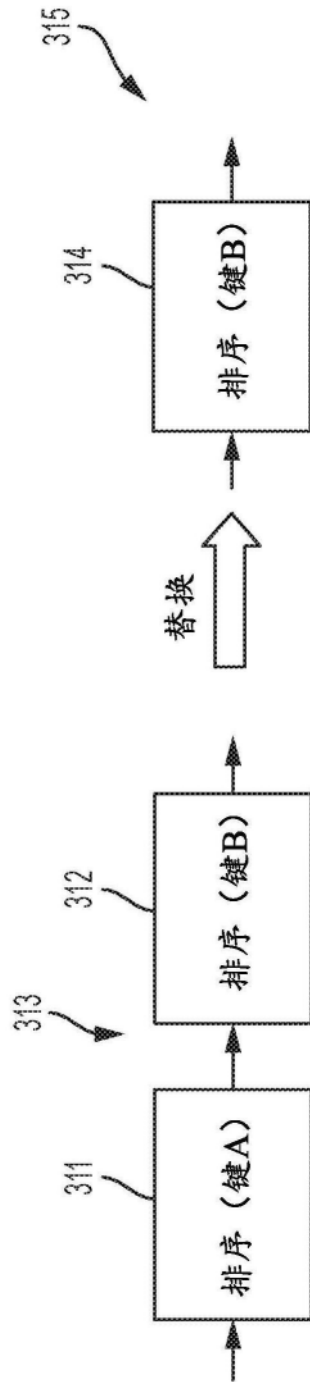


图3C

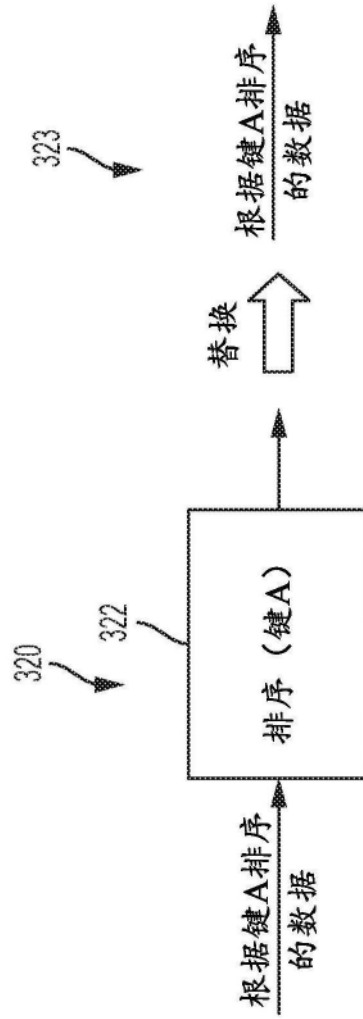


图3D

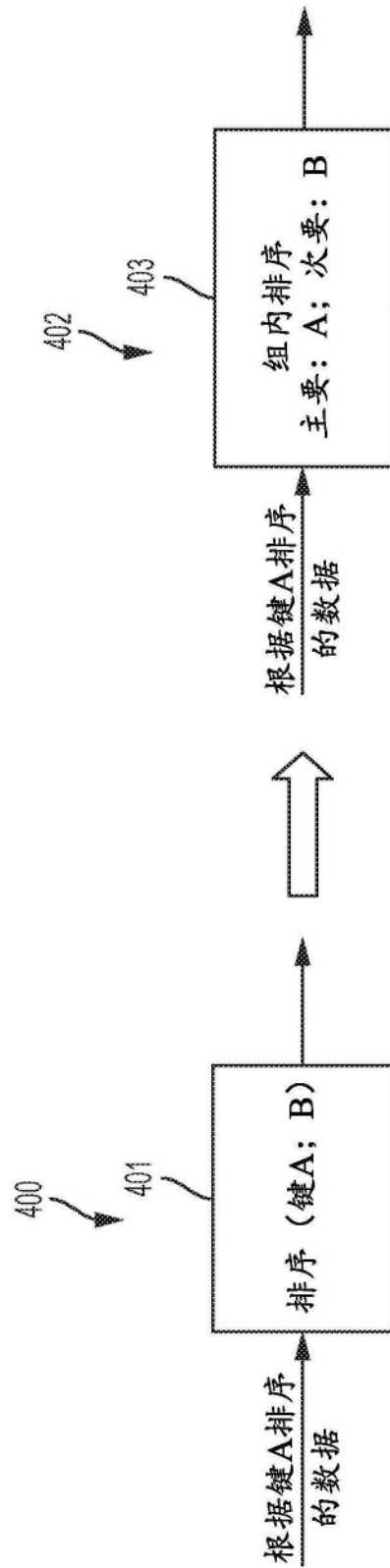


图4A

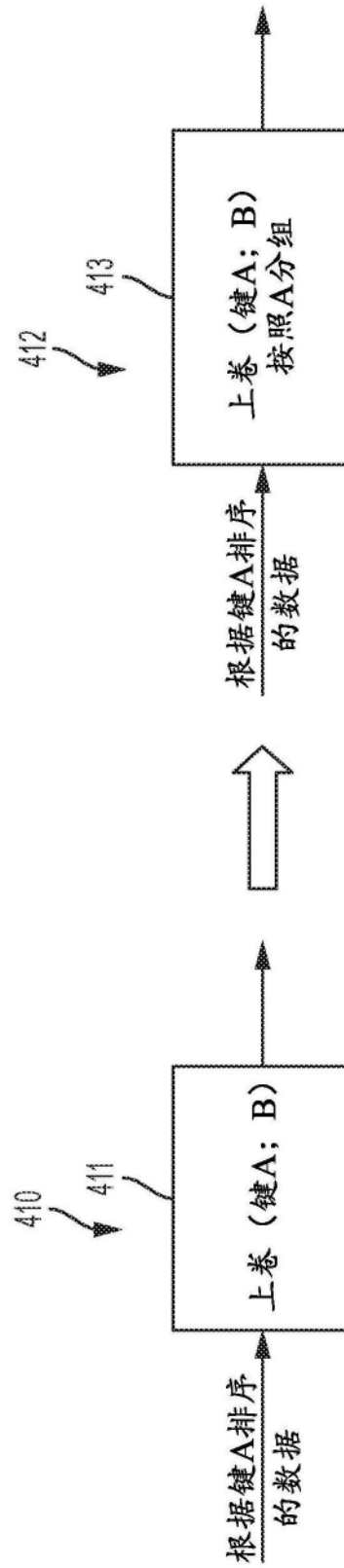


图4B

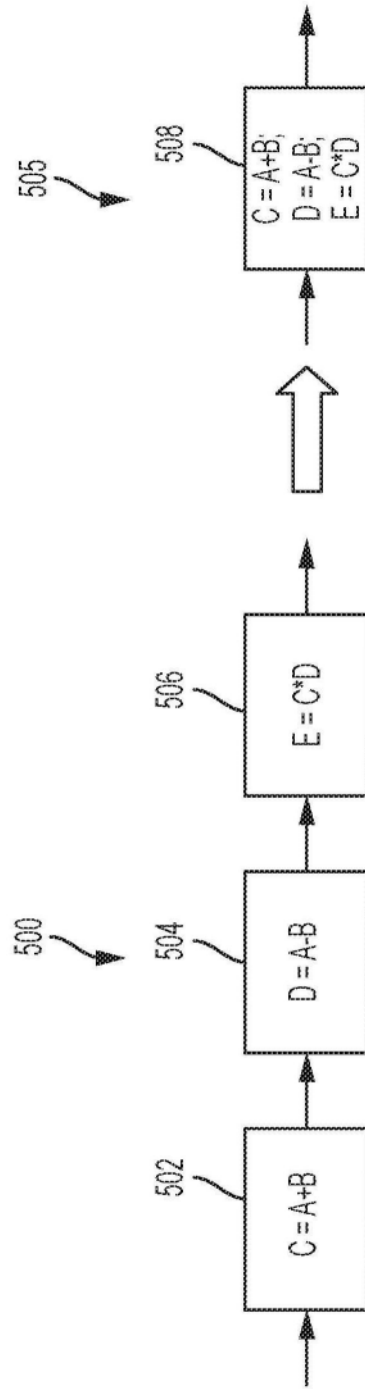


图5A

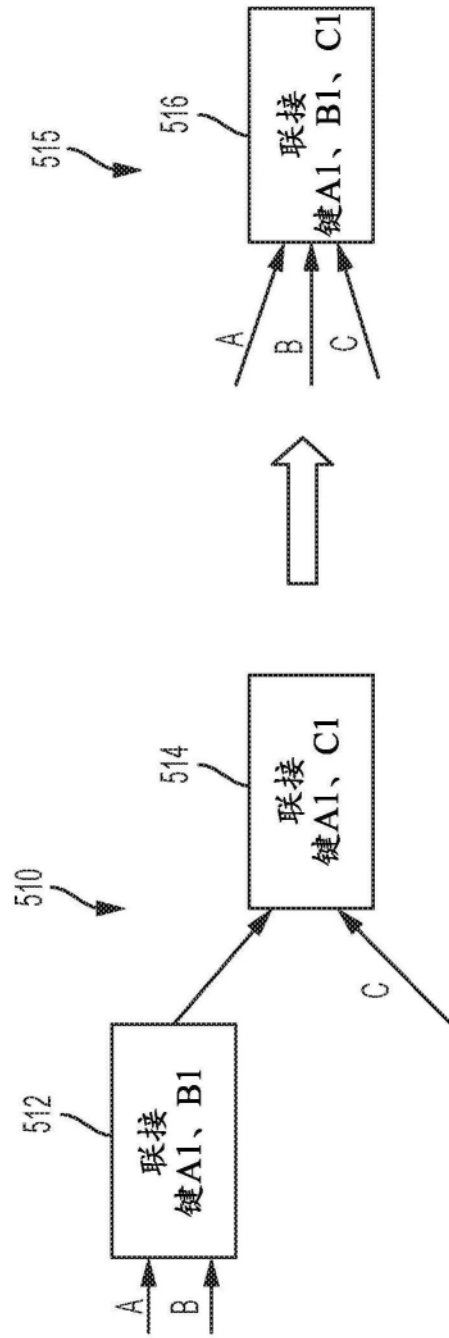


图5B

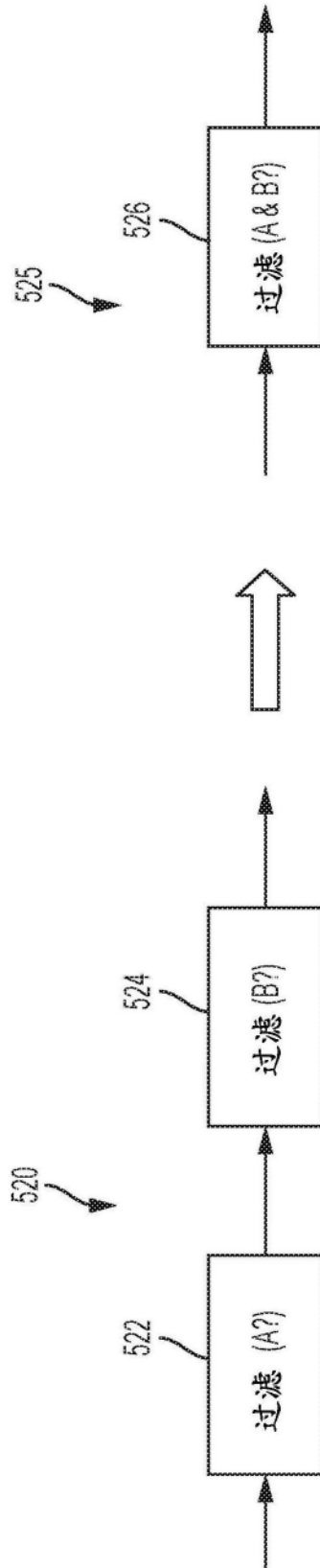


图5C

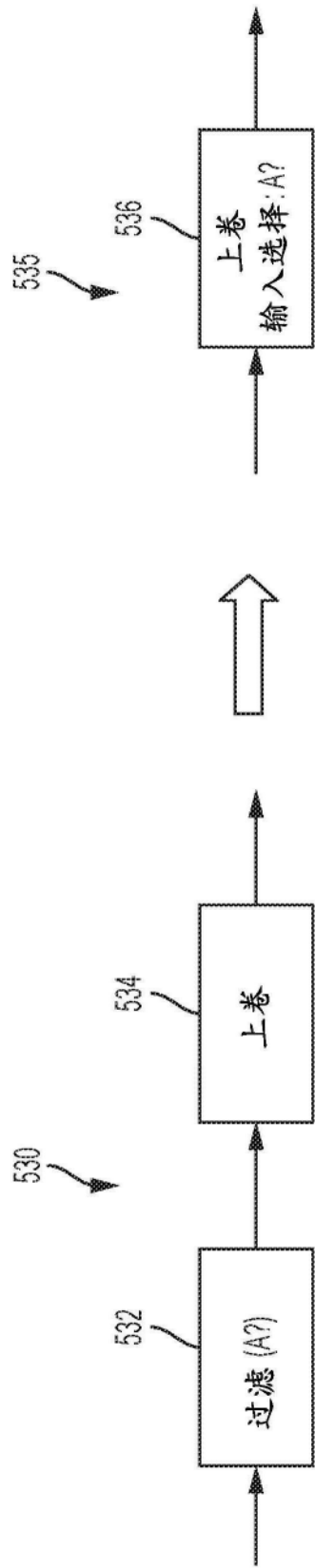


图5D

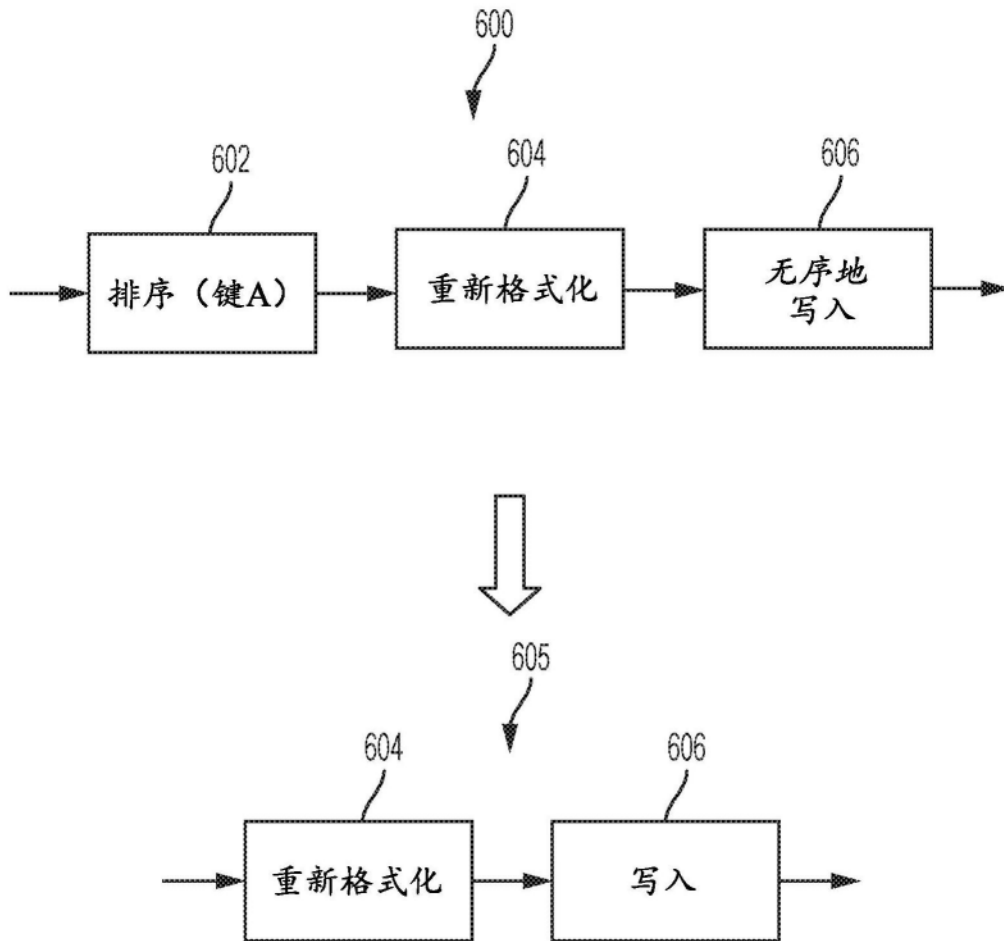


图6

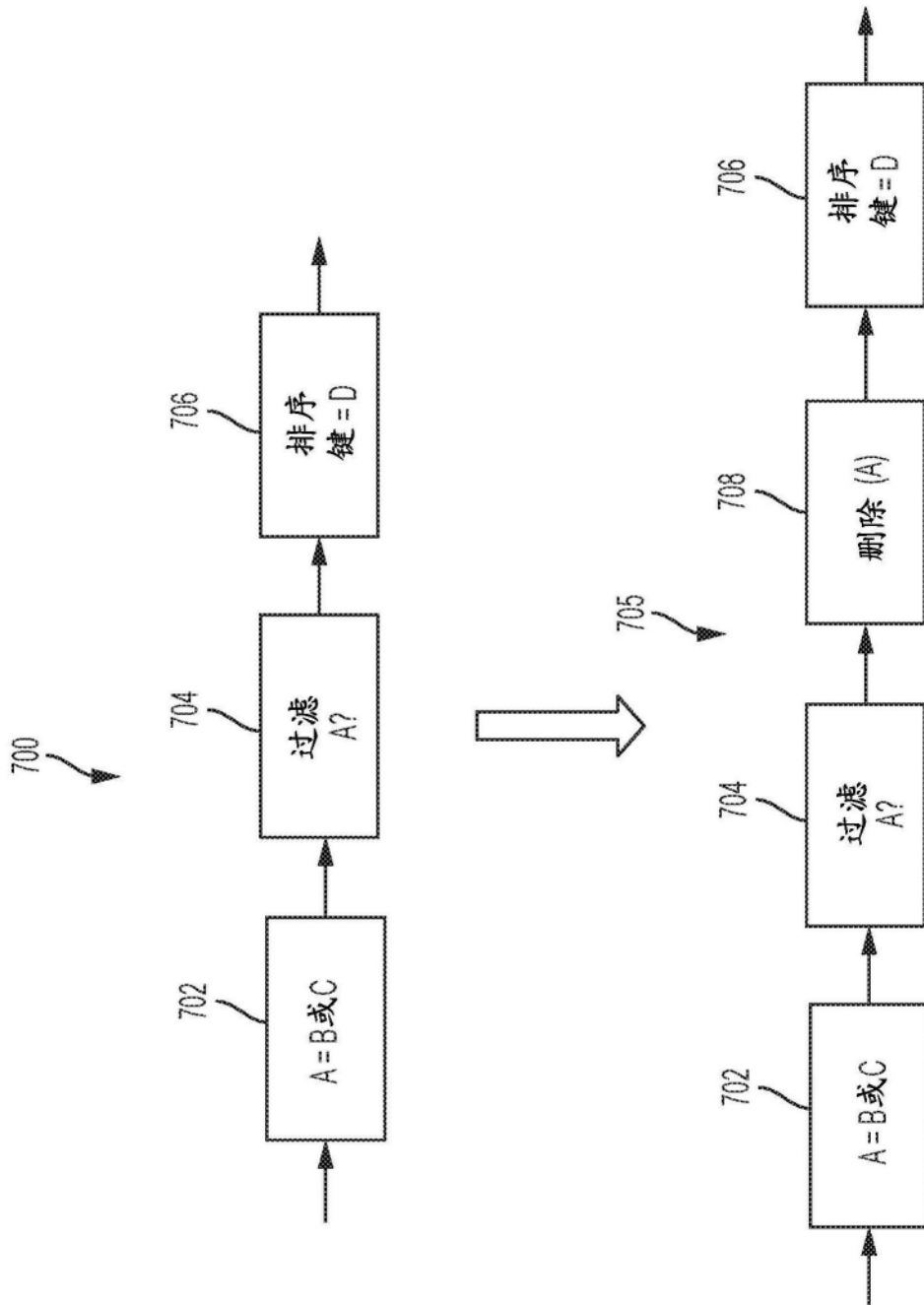


图7

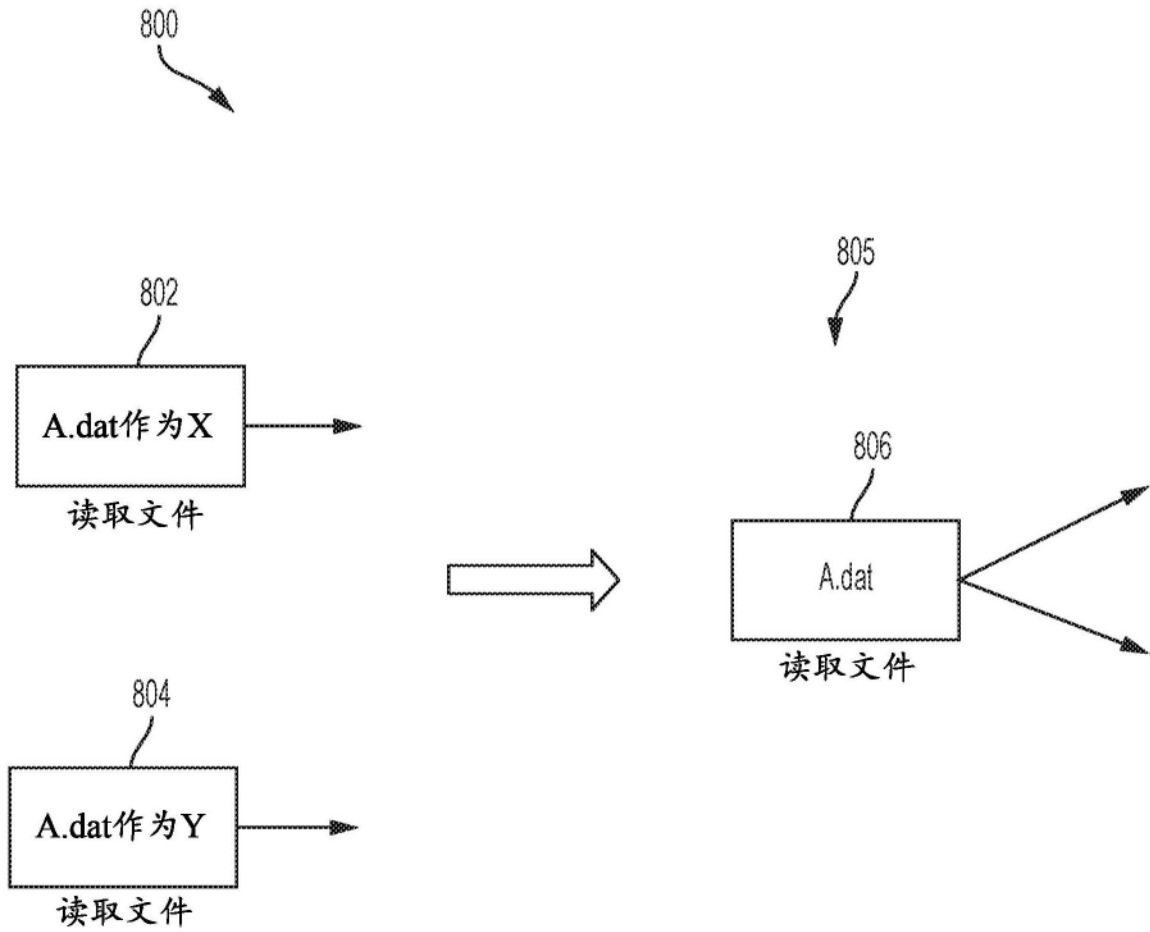


图8A

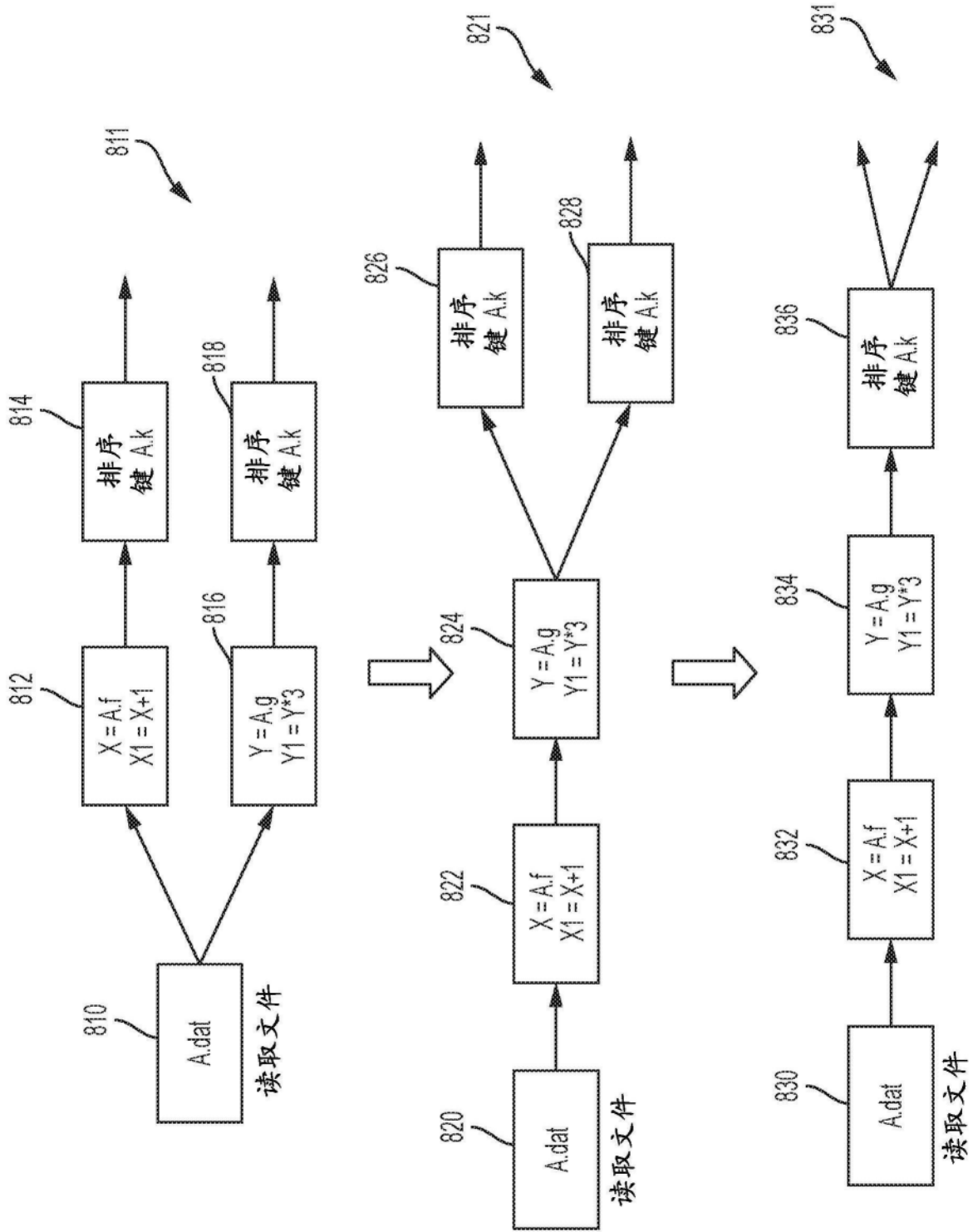


图8B

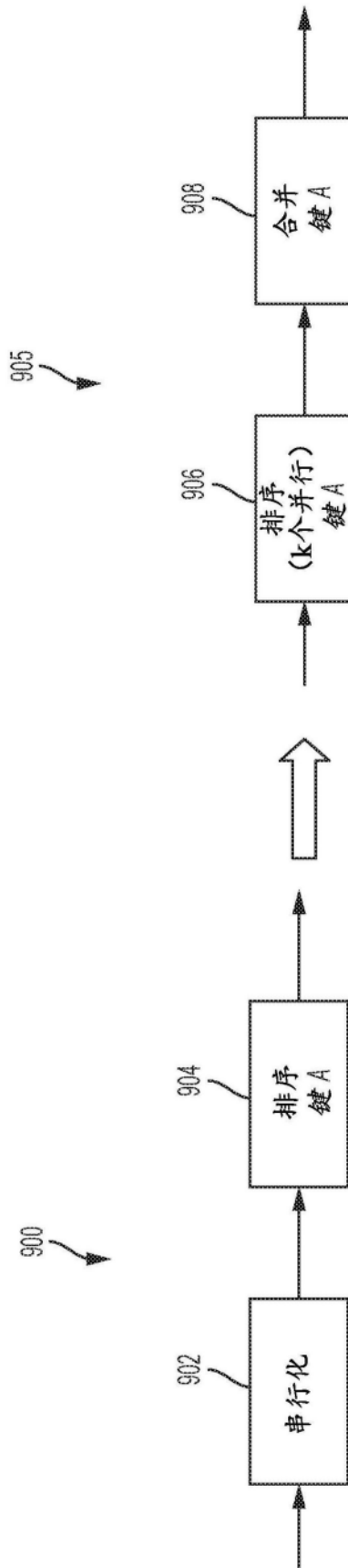


图9

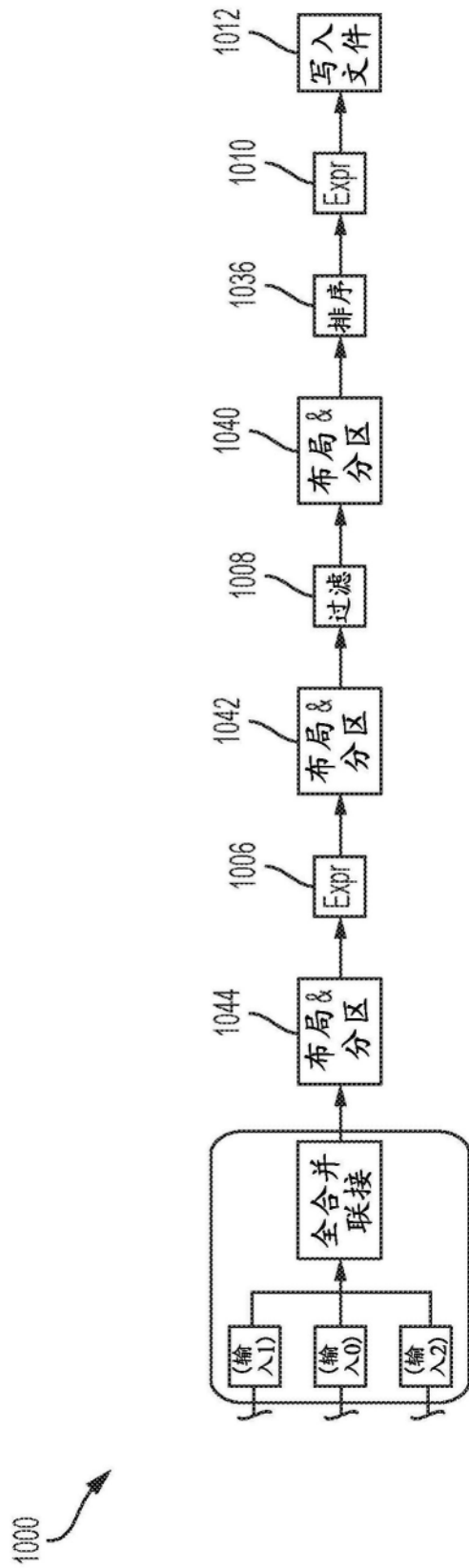


图10A续

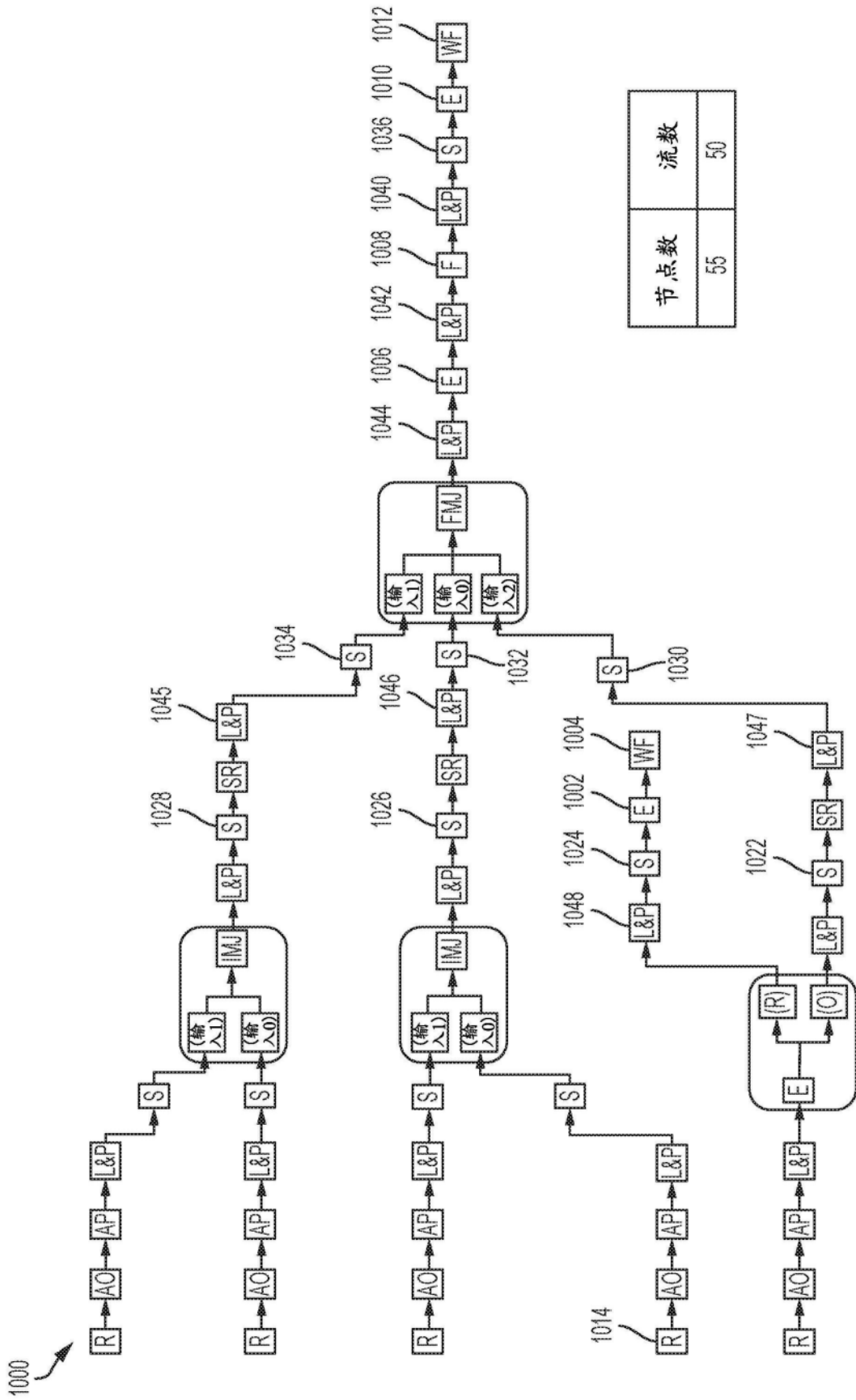


图10C

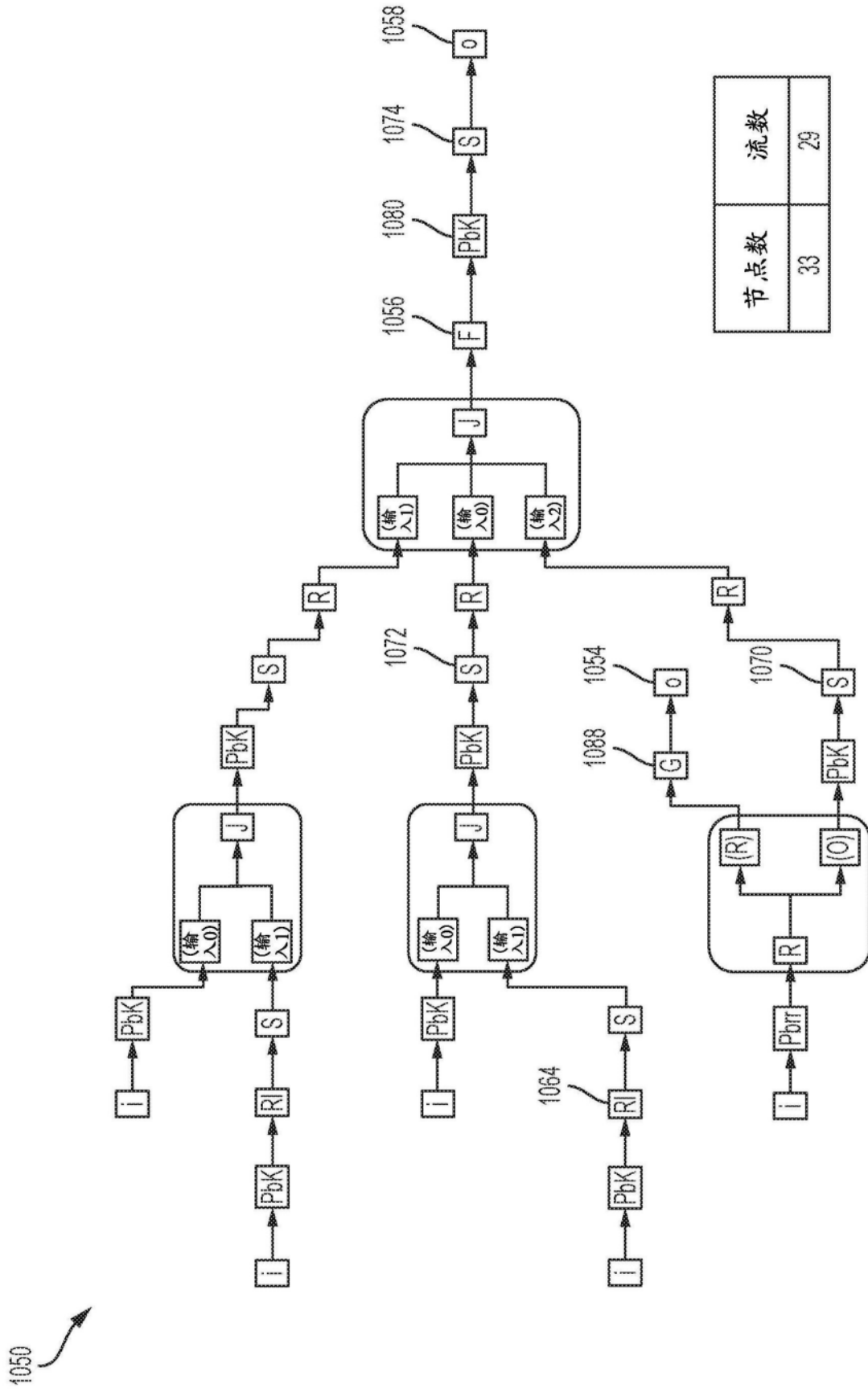


图10D

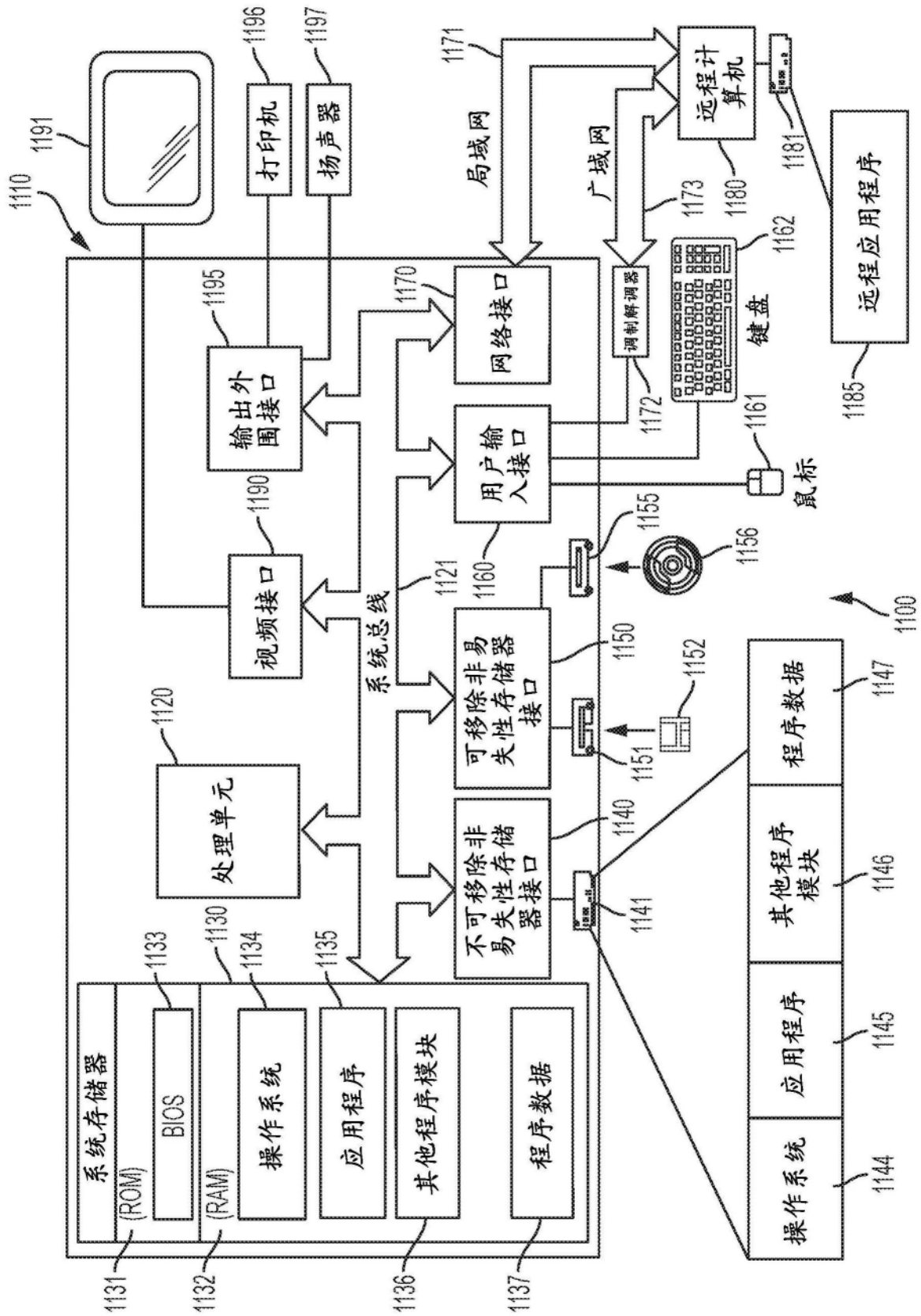


图11