(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 107818125 B
(45)授权公告日 2021.10.15

(21)申请号 201710632383.8

(22)申请日 2017.07.28

(65)同一申请的已公布的文献号
申请公布号 CN 107818125 A

(43)申请公布日 2018.03.20

(30)优先权数据
15/261,883 2016.09.10 US

(73)专利权人 SAP欧洲公司
地址 德国瓦尔多夫

(72)发明人 M.戈尔根斯 D.杜尔纳

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邵亚丽

(51)Int.Cl.
G06F 16/22(2019.01)
G06F 16/242(2019.01)
G06F 8/41(2018.01)
G06F 9/30(2006.01)

(56)对比文件
CN 104040541 A,2014.09.10
CN 104040541 A,2014.09.10
US 2015154255 A1,2015.06.04
CN 101515294 A,2009.08.26
CN 105302055 A,2016.02.03
US 2013151567 A1,2013.06.13
US 2005027701 A1,2005.02.03

审查员 刘明惠

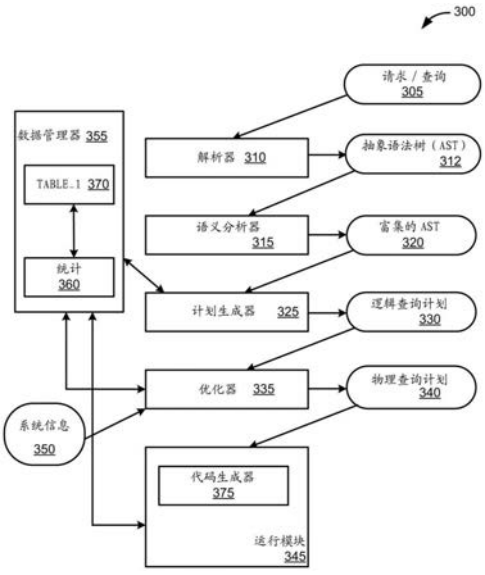
权利要求书4页 说明书12页 附图8页

(54)发明名称

通过SIMD处理器寄存器对数据进行迭代评估

(57)摘要

生成用于处理内存数据库系统中的数据集的可运行代码。该可运行代码基于包括与数据集的第一部分相关联的断言的程序指令。数据集的第一部分被划分为数据部分。数据部分包括与将要分配给处理器处的寄存器中的多个位值相对应的多个数据元素。处理器处的寄存器与在多个数据上执行单个指令相关联。在处理器处，迭代地评估数据部分以确定将迭代地存储到SIMD寄存器中的位向量。基于在处理器处的SIMD寄存器处的迭代存储的位向量，通过调用来自数据集的数据来迭代地确定结果数据集。结果数据集通过处理器提供以进一步使用。



CN 107818125 B

1.一种用于评估内存数据库系统中的数据的计算机实现的方法,所述方法包括:

生成用于处理所述内存数据库系统中的数据集的可运行代码,其中,所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令;

确定处理器处的寄存器的大小以确定将要分配的位值的数量,将要分配的位值的数量等于寄存器的大小,其中,所述寄存器是与并行地在多个数据元素上执行单一指令相关联的单指令多数据SIMD寄存器;

将所述数据集的第一部分划分成一个或多个数据部分,其中,数据部分包括与要分配到处理器处的寄存器中的位值的数量相对应的多个数据元素;

在所述处理器处,迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的一个或多个位向量,每个位向量存储特定一个数据部分中的数据元素的评估结果;以及

在所述处理器处,基于所存储的一个或多个位向量并且基于从所述数据集调用数据,迭代地确定和提供一个或多个结果数据集,

其中,迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的所述一个或多个位向量还包括:

对于每个数据部分:

基于所述断言,基于来自所述数据部分的数据元素的值的评估来确定对应的位向量,其中,所述位向量包括与符合所述断言的评估结果相对应的位值,每个位值对应于特定一个值的评估结果;

在所述处理器处,将位向量加载到所述寄存器中;以及

在所述处理器处,基于所述位向量中的正位值的位置,从来自所述数据集的列确定行的标识符。

2.如权利要求1所述的方法,还包括:

根据可运行代码将所述数据集加载到将要处理的内存数据库系统的主存储器处的数据库表中。

3.如权利要求1所述的方法,其中,所述数据集的所述第一部分包括来自所述数据集的一列或多列数据,并且其中,包括在所述程序指令中的所述断言定义用于确定所述一个或多个结果数据集的规则。

4.如权利要求1所述的方法,还包括:

生成或接收定义将对所述数据集执行的操作的请求;以及

针对所述请求生成逻辑计划,所述逻辑计划定义与所述程序指令和所述数据集相关的逻辑运算的数据流;

其中,所生成的运行计划是与将由内存数据库系统中的运行引擎运行的逻辑计划相对应的物理计划,其中,基于从与所生成的逻辑计划相对应的所确定的物理计划的集合中的选择来确定运行计划,其中,所确定的物理计划的集合基于关于所述运行所使用的内存数据库系统的系统能力的信息、以及基于针对所述数据集的统计数据来确定。

5.如权利要求4所述的方法,其中,生成用于所述程序指令的逻辑计划还包括:

解析所述请求以生成用于执行操作的语法树;

验证所述语法树以确定所述语法树是否对所述数据集有效;以及

当所述语法树对所述数据集有效时，改进所述语法树以包括关于所述数据集的信息。

6.一种对内存数据库系统中的数据进行评估的计算机系统，包括：

处理器；

存储器，与所述处理器相关联，存储指令用于：

生成用于处理所述内存数据库系统中的数据集的可运行代码，其中，所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令；

确定处理器处的寄存器的大小以确定将要分配的位值的数量，将要分配的位值的数量等于寄存器的大小，其中，所述寄存器是与并行地在多个数据元素上执行单一指令相关联的单指令多数据SIMD寄存器；

将所述数据集的第一部分划分成一个或多个数据部分，其中，数据部分包括与将要分配给处理器处的寄存器中的位值的数量相对应的多个数据元素；

在所述处理器处，迭代地评估所述一个或多个数据部分以确定要被迭代地存储到所述寄存器中的一个或多个位向量，每个位向量存储特定一个数据部分中的数据元素的评估结果；以及

在所述处理器处，基于所存储的一个或多个位向量并且基于从所述数据集调用数据，迭代地确定并且提供一个或多个结果数据集，

其中，迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的所述一个或多个位向量还包括：

对于每个数据部分：

基于所述断言，基于来自所述数据部分的数据元素的值的评估来确定对应的位向量，其中，所述位向量包括与符合所述断言的评估结果相对应的位值，每个位值对应于特定一个值的评估结果；

在所述处理器处，将位向量加载到所述寄存器中；以及

在所述处理器处，基于所述位向量中的正位值的位置，从来自所述数据集的列确定行的标识符。

7.如权利要求6所述的系统，其中，所述存储器还存储指令用于：

根据可运行代码将所述数据集加载到将要处理的内存数据库系统的主存储器的数据库表中。

8.如权利要求6所述的系统，其中，所述数据集的第一部分包括来自所述数据集的一列或多列数据，并且其中，包括在所述程序指令中的所述断言定义用于确定所述一个或多个结果数据集的规则。

9.如权利要求6所述的系统，其中，所述存储器还存储指令用于：

生成或接收定义将对所述数据集执行的操作的请求；以及

针对所述请求生成逻辑计划，所述逻辑计划定义与所述程序指令和所述数据集相关的逻辑运算的数据流；

其中，所生成的运行计划是与将由内存数据库系统中的运行引擎执行的逻辑计划相对应的物理计划，其中，基于从与所生成的逻辑计划相对应的所确定的物理计划的集合中的选择来确定运行计划，其中，所确定的物理计划的集合基于关于所述运行所使用的内存数据库系统的系统能力的信息，并且基于针对所述数据集的统计数据来确定。

10.如权利要求9所述的系统，其中，用于生成所述程序指令的所述逻辑计划的指令还包括指令，用于：

解析所述请求以生成用于执行操作的语法树；

验证所述语法树以确定所述语法树是否对所述数据集有效；以及

当所述语法树对所述数据集有效时，改进所述语法树以包含关于所述数据集的信息。

11.一种存储指令的非暂时计算机可读介质，所述指令被运行时使计算机系统：

生成用于处理内存数据库系统中的数据集的可运行代码，其中，所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令；

确定处理器处的寄存器的大小以确定将要分配的位值的数量，将要分配的位值的数量等于寄存器的大小，其中，所述寄存器是与并行地在多个数据元素上执行单一指令相关联的单指令多数据SIMD寄存器；

将所述数据集的第一部分划分成一个或多个数据部分，其中，数据部分包括与将要分配给处理器处的寄存器中的位值的数量相对应的多个数据元素；

在所述处理器处，迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的一个或多个位向量，每个位向量存储特定一个数据部分中的数据元素的评估结果；以及

在所述处理器处，基于所存储的一个或多个位向量并且基于从所述数据集调用数据，迭代地确定并提供一个或多个结果数据集，

其中，迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的所述一个或多个位向量还包括：

对于每个数据部分：

基于所述断言，基于来自所述数据部分的数据元素的值的评估来确定对应的位向量，其中，所述位向量包括与符合所述断言的评估结果相对应的位值，每个位值对应于特定一个值的评估结果；

在所述处理器处，将位向量加载到所述寄存器中；以及

在所述处理器处，基于所述位向量中的正位值的位置，从来自所述数据集的列确定行的标识符。

12.如权利要求11所述的计算机可读介质，还包括指令，所述指令在被运行时导致所述计算机系统：

根据所述可运行代码将所述数据集加载到将要处理的内存数据库系统的主存储器中的数据库表中。

13.如权利要求11所述的计算机可读介质，其中，所述数据集的第一部分包括来自所述数据集的一列或多列数据，并且其中，包括在所述程序指令中的断言定义用于确定所述一个或多个结果数据集的规则。

14.如权利要求11所述的计算机可读介质，还包括指令，所述指令在被运行时导致所述计算机系统：

生成或接收定义将对所述数据集执行的操作的请求；以及

针对所述请求生成逻辑计划，所述逻辑计划定义与所述程序指令和所述数据集相关的逻辑运算的数据流；

其中,所生成的运行计划是与将由内存数据库系统中的运行引擎执行的所述逻辑计划相对应的物理计划,其中,基于从与所生成的逻辑计划相对应的所确定的物理计划的集合中的选择来确定所述运行计划,其中,所确定的物理计划的集合基于关于所述运行所使用的内存数据库系统的系统能力的信息,并且基于针对所述数据集的统计数据来确定。

15.如权利要求14所述的计算机可读介质,其中,用于生成所述程序指令的逻辑计划的指令还包括指令,所述指令在执行时导致所述计算机系统:

解析所述请求以生成用于执行操作的语法树;

验证所述语法树以确定所述语法树是否对所述数据集有效;以及

当所述语法树对所述数据集有效时,改进所述语法树以包含关于所述数据集的信息。

通过SIMD处理器寄存器对数据进行迭代评估

技术领域

[0001] 该领域通常涉及数据处理、数据库系统和计算机处理器。

背景技术

[0002] 计算机程序可以用编程语言形式编写，包括编译或解释语言。计算机程序可以被部署为独立的程序或作为模块或适用于在计算环境中使用的其他单元。该计算机程序可被部署为在计算机上运行或分布在通过通信网络互连的多个计算机上。可以由运行计算机程序的一个或多个可编程处理器执行操作，以通过对相关联的数据进行操作并生成输出来执行功能。

[0003] 数据和程序指令(例如，软件，计算机程序)的集合可以存储在存储单元上并且在由计算机系统运行期间驻留在主存储器内和/或处理器内部。在计算机系统上的处理程序指令包括将处理器寄存器和存储器内的数据操纵和转换成其他数据，该数据类似地表示为存储器或寄存器或其他信息存储器内的物理量。程序指令可以基于对存储在计算机系统上的数据库中的数据的所定义的查询。可以使用查询语句来查询数据库并检索与指定标准相匹配的所选数据。处理器寄存器是可用于计算机系统上的处理器的快速可访问位置。寄存器通常包括一定量的快速存储器，它们可以具有特定的硬件功能，并且/或者可能限制对寄存器的访问，例如只读，只写等。

发明内容

[0004] 根据示例性实施例的一方面，提供了一种用于评估内存数据库系统中的数据的计算机实现的方法，所述方法包括：生成用于处理所述内存数据库系统中的数据集的可运行代码，其中，所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令；将所述数据集的第一部分划分成一个或多个数据部分，其中，数据部分包括与要分配到处理器处的寄存器中的多个位值相对应的多个数据元素；在所述处理器处，迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的一个或多个位向量；以及在所述处理器处，基于所存储的一个或多个位向量并且基于从所述数据集调用数据迭代地确定和提供一个或多个结果数据集。

[0005] 根据示例性实施例的另一方面，提供了一种对内存数据库系统中的数据进行评估的计算机系统，包括：处理器；与存储指令的处理器相关联的存储器，用于：生成用于处理所述内存数据库系统中的数据集的可运行代码，其中，所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令；将所述数据集的第一部分划分成一个或多个数据部分，其中，数据部分包括与将要分配给处理器处的寄存器中的多个位值相对应的多个数据元素；迭代地评估所述一个或多个数据部分以确定要被迭代地存储到所述寄存器中的一个或多个位向量，其中，所述处理器处的所述寄存器与并行地在多个数据元素上执行单个指令相关联；以及在所述处理器处，基于所存储的一个或多个位向量并且基于从所述数据集调用数据迭代地确定并且提供一个或多个结果数据集。

[0006] 根据示例性实施例的另一方面，提供了一种存储指令的非暂时计算机可读介质，所述指令被运行时使计算机系统：生成用于处理所述内存数据库系统中的数据集的可运行代码，其中，所述可运行代码基于包括与所述数据集的第一部分相关联的断言的程序指令；将所述数据集的第一部分划分成一个或多个数据部分，其中，数据部分包括与将要分配给处理器处的寄存器中的多个位值相对应的多个数据元素；迭代地评估所述一个或多个数据部分以确定将迭代地存储到所述寄存器中的一个或多个位向量，其中，所述处理器处的所述寄存器与并行地在多个数据元素上执行单个指令相关联；以及在所述处理器处，基于所存储的一个或多个位向量并且基于从所述数据集调用数据，迭代地确定并提供一个或多个结果数据集。

附图说明

[0007] 权利要求具体阐述了实施例。通过示例的方式而非限制的方式在附图中示出了实施例，其中类似的附图标记表示相似的元件。从下面结合附图的详细描述可以更好地理解实施例及其优点。

[0008] 图1是示出根据一个实施例的用于评估与内存数据库系统中的程序相关联的数据的示例性环境的框图。

[0009] 图2A是示出根据一个实施例的用于逻辑地评估来自在内存数据库系统中运行的程序的程序指令的过程的流程图。

[0010] 图2B是示出根据一个实施例的用于运行来自程序的程序指令的计划生成的过程的流程图。

[0011] 图3是示出根据一个实施例的用于评估将要在内存数据库系统中运行的请求的过程的流程图。

[0012] 图4是示出根据一个实施例的与通过对计算机处理器处的寄存器内的数据集的部分进行迭代(iterating)来评估与数据集有关的请求的过程的相关联的示例性数据结构的框图。

[0013] 图5是示出根据一个实施例的通过处理器寄存器对数据进行迭代评估的过程的流程图。

[0014] 图6是示出根据一个实施例的用于运行与通过处理器寄存器在数据集上的操作有关的程序指令的系统的框图。

[0015] 图7是示出根据一个实施例的用于通过在计算机处理器处生成位向量来确定结果数据来对与数据集上的操作有关的请求的过程进行评估的流程图。

[0016] 图8是示出其中可以实现被描述为用于评估与内存数据库系统中的程序相关联的数据的技术的计算环境的实施例的框图。

具体实施方式

[0017] 本文描述了通过处理器寄存器的迭代评估数据的技术的实施例。在下面的描述中，阐述了许多具体细节以提供对实施例的透彻理解。然而，相关领域的技术人员将认识到，可以在没有一个或多个具体细节，或使用其他方法、组件、材料等的情况下实践实施例。在其他实例中，公知的结构、材料或操作没有详细示出或描述。

[0018] 整个说明书中对“一个实施例”,“本实施例”和类似短语的指代意味着结合该实施例描述的特定特征、结构或特性被包括在一个或多个实施例中的至少一个实施例中。因此,在整个说明书中,在各个地方出现的这些短语不一定都是指相同的实施例。此外,特定特征、结构或特征可以以任何合适的方式组合在一个或多个实施例中。

[0019] 图1是示出根据一个实施例的用于评估与内存数据库系统中的程序相关联的数据的示例性环境100的框图。示例性环境100包括主存储器105、高速缓冲存储器和中央处理单元(CPU)170。主存储器105可以存储指令和数据,其可以与用于在内存数据库系统中执行的一个或多个程序相关联。程序125被定义为在主存储器105中的过程存储器(process memory)110上运行。过程存储器110由在示例性环境100中运行的底层操作系统(OS)(未示出)管理。程序125可以是软件程序,其可以用不同的开发工具和技术而设计和开发。程序125包括程序指令130。程序125可以与程序指令130中定义的与数据集115相关的操作相关联。数据集115可以存储在数据库表中。程序指令130包括数据管理器135和运行引擎175。数据管理器135维护通过由运行引擎175运行程序指令130生成的数据。可以在加载数据时,其中数据从程序指令130被参考,和/或当利用运行指令所需的数据时使用数据管理器135。运行引擎175基于程序指令130在运行时(runtime)期间生成代码。所生成的代码定义如何通过CPU 170的与多个数据上的单个指令相关联的寄存器(SIMD寄存器)155来处理来自数据集115的数据。

[0020] 在一个实施例中,程序指令130可以定义用于查询数据以生成结果数据集的操作。例如,可以基于定义查询的请求来定义指令。查询被定义为查询语言(例如,结构化查询语言(SQL))中的查询字符串,来管理数据库中保存的数据。表1呈现了从使用SQL设计的数据库表-表“T”来查询数据的示例性请求。表2呈现了存储在数据库表“T”中的数据。所呈现的作为SQL字符串的请求可以与诸如程序125的程序相关联。程序125使用SQL字符串来处理接收到的请求。

[0021]

```
Select a
From T
Where b>100;
```

[0022] 表1

[0023]

a	b	c
14	5	结果1
21	3	结果2
13	101	结果3
16	90	结果1
14	102	结果3

[0024] 表2

[0025] 可以评估和运行程序125中定义的程序指令130。例如,如果程序指令130与表1中呈现的SQL字符串相关联,则当运行程序125时,基于列“b”的值对数据集115进行过滤。运行引擎175基于执行对来自列“b”的值的过滤以确定来自列“a”的值的数据的结果集的程序

125来生成可运行程序。如果指令与表2中呈现的数据集相关，则可以基于如where子句中定义的断言“b>100”过滤列“b”。确定与符合该断言的来自列“b”的值对应的来自列“a”的值。结果数据包括与高于100的来自列“b”的值“101”和“102”对应的来自列“a”的值“13”和“14”。

[0026] 在一个实施例中，当运行来自程序125的程序指令130时，CPU 170的高速缓冲存储器被用于存储在操作期间由程序125频繁引用的程序指令或数据。快速访问这些指令和数据提高了程序执行的总体速度。CPU 170可以在与三个高速缓存级别的高速缓存存储器相关联-高速缓存“1”150、高速缓存“2”145和高速缓存“3”140。CPU 170包括寄存器，包括单指令多数据(SIMD)寄存器155和其他寄存器160。寄存器是CPU 170可用的快速可访问位置。寄存器可以保存(hold)通过高速缓存存储器级别-高速缓存“3”140、高速缓存“2”145和高速缓存“1”150从主存储器105传送的到达寄存器的值。可以在运行诸如程序指令130之类的程序指令时存储寄存器中的数据。

[0027] SIMD寄存器155与并行地在多个数据元素上执行单个指令相关联。因此，在这样的SIMD寄存器155中，在多个数据元素上同时执行如程序指令(例如，130)中定义的公共操作，而不是以标量方式遍历表中的行。其他寄存器160可以存储基于程序指令生成的值，以标量方式定义数据迭代。

[0028] 图2A是示出根据一个实施例的用于逻辑地评估来自在内存数据库系统中运行的程序的程序指令的过程200的流程图。在一个实施例中，程序可以是诸如图1的程序125。程序可以包括定义与操纵和/或评估来自数据集的数据相关联的操作的程序指令。程序指令可以与存储在数据库表中的数据相关联。程序指令可以与从数据库表查询数据的请求相关联。基于指令，关于table_1 220执行表扫描操作215。例如，table_1 220可以是诸如关于图1讨论并在上表2中表示的表“T”。程序指令可以与针对表“T”定义的查询、并具有所定义的“where”子句的SQL字符串相关联。程序指令可以与表1中呈现的SQL字符串相关联。基于程序指令，table_1 220根据程序指令在210进行过滤。

[0029] 程序指令可以包括将被施加用于评估table_1 220中的数据的表达式，例如用于过滤数据。使用来自表1的SQL字符串和与表2对应的table_1 220的示例，“where”子句过滤来自table_1 220的行，其中“b”值大于100。过滤表210步骤与断言表达式树225相关联。可以基于在程序中定义的指令的评估来生成断言表达式树225。表达式树可以以树状数据结构表示程序指令，其中，节点表示方法调用、二进制操作、数据值等。表达式树可以是二叉树，其中内部节点对应于运算符，且叶节点对应于操作数。基于过滤(在210处)，在205处获取(fetch)与所定义的程序指令对应的数据。因此，过程200中定义的数据流从过程的底部到顶部定义，从表扫描215开始并以获取数据205结束。

[0030] 图2B是示出根据一个实施例的用于运行来自程序的程序指令的计划生成的过程230的流程图。基于定义用于评估来自程序的程序指令的逻辑计划的过程200，生成用于运行程序指令的计划。当运行来自程序的程序指令时，可以利用来自CPU的寄存器来提高程序的性能。寄存器可以是SIMD寄存器，例如SIMD寄存器155(图1)。为了利用SIMD寄存器，过程230可以被定义用于将经处理的数据分配到SIMD寄存器中的代码生成。可以针对表_1202定义SIMD扫描240(如过程200所述)。SIMD扫描240与针对来自程序指令的操作而定义的断言表达式树225对应。基于SIMD扫描240，执行获取数据235。以SIMD方式处理数据的获取，使得

获取的数据被处理进入相应定义的SIMD寄存器。SIMD寄存器允许一个微指令在多个数据项上(例如在来自数据库表中多行的多个数据值上)同时操作。通常需要重复连续的指令的操作可以利用一条指令来执行,以增强性能和计算时间。

[0031] 图3是示出根据一个实施例的用于评估将在内存数据库系统中运行的请求305的过程300的流程图。请求305可以是与内存数据库系统中的数据集相关联的查询。基于这样的请求,可以定义例如图1的程序125的程序。请求305可以包括基于相关联的例如存储在数据库表中的数据集定义生成结果的表达式。请求305可以与table_1 370相关联。table_1 370可以包括诸如表2中呈现的数据的数据。请求305可以包括对table_1 370中的数据的SQL查询,该查询可以包括定义用于通过查询table_1 370生成结果数据的规则的断言和表达式。SQL查询可以是诸如表1中呈现的SQL查询。请求305由解析器310作为语句(statement)接收。请求305被解析器310读取并分析以生成抽象语法树(AST)312。在AST 312中,来自请求305的语句在逻辑上被划分并且被定位在沿着树枝定位操作数和操作符的树结构中。

[0032] 解析器310将生成的AST 312发送到语义分析器315,语义分析器315生成富集的抽象语法树(AST)320。语义分析器315在所定义的表table_1 370的上下文中验证从解析器310接收到的AST 312。AST 312的验证包括确定所定义的操作数和操作符是否被逻辑地构造以便对来自table_1 370的数据有效地执行。基于验证,语义分析器315更新了AST以生成富集的AST 320,其还包括用于table_1 370中的数据的元数据。例如,富集的AST 320可以包括关于table_1 370中的行数、存储在来自table_1 370的特定列中的数据的类型等的信息。计划生成器325从语义分析器315接收富集的AST 320,以生成用于请求305的逻辑查询计划330。计划生成器325获取富集的AST 320并将其转换为定义基于富集的AST 320的步骤的逻辑查询计划330。逻辑查询计划330可以与图2的处理过程200对应,其中table_1 220对应于table_1 370。

[0033] 基于所定义的逻辑查询计划330,优化器335确定物理查询计划340。在一个示例中,所确定的物理查询计划340可以被解释为生成结果数据集。在另一示例中,所确定的物理查询计划340可用于生成运行请求305的代码。优化器335耦合到数据管理器355以接收关于存储在表_1 370中的数据的信息。优化器335通过数据管理器355接收对来自table_1 370的数据的统计360。例如,该统计360包括列中的数据的最小值和/或最大值、表中的行数、表中的列数以及对table_1 370中的数据的其他统计指标。此外,优化器335接收关于其中将运行程序指令305的系统环境的系统信息350。以这种方式,优化器335接收其中将运行程序指令305的引擎的能力的信息、以及关于将根据程序指令305处理的数据的信息。优化器335可以基于所接收的系统信息350和统计360生成一组物理计划。来自该组物理查询计划的计划可以是代码生成计划。优化器335可以基于来自数据管理器355的输入和系统信息350来确定用于代码生成的最佳物理计划。

[0034] 用于代码生成的物理查询计划340可以从优化器335提供给运行模块345,运行模块345可以运行由代码生成器375定义的运行时生成的程序指令。如关于图1所描述的,运行模块345执行由请求305定义的操作,并通过高速缓存存储器级别将数据从内存数据库系统的主存储器的table_1 370传送到处理器中的SIMD寄存器。

[0035] 运行模块345包括代码生成器375,代码生成器375可以在处理来自table_1 370的

数据的同时，为利用例如图1的SIMD寄存器155的SIMD寄存器的所接收的请求305生成编译形式的可运行代码。代码生成器375可以定义如何基于在请求305中定义的操作和断言来评估来自table_1 370的数据。代码生成器375利用由优化器335提供的物理查询计划340以在运行时生成代码，其定义了在如处理请求305中定义的处理数据期间使用来自处理器的哪种寄存器。可以设计代码生成计划以优化数据到寄存器的分配，并且还可以定义利用SIMD寄存器以增加执行程序的性能。

[0036] 当代码生成器375与利用SIMD寄存器以处理数据相关联时，则所生成的代码还可以定义如何分离和分割经处理的数据以优化处理的运行。程序指令305可以定义与存储在table_1 370中的大量数据相关联的操作，其可能不会立刻通过在处理器处的SIMD寄存器处理。因此，来自table_1 370的数据可以以子集来处理，这些子集可以被定义为行的部分。这些子集或部分可以被定义为在大小上与SIMD寄存器的大小相对应。因此，SIMD寄存器的大小可以被确定并作为系统信息350的一部分被接收。在一些实施例中，优化器335可以确定大小。在一些其他实施例中，优化器335可以检查以确定SIMD寄存器是否存在并且是否可用。代码生成器375可以用于确定SIMD寄存器的大小并选择将使用哪个SIMD寄存器。

[0037] 代码生成器375可以在使用SIMD寄存器的存储器的运行时期间生成代码。表3包括基于所接收的例如表1中的SQL字符串(与例如表2的表相关联)的SQL字符串的示例性伪运行时生成的代码。对于表3中的示例，假设SIMD寄存器的大小是256位。因此，迭代是在每个包括256个数字的“i”部分上执行的。伪代码示例被定义为有助于理解代码生成的人造和非形式化语言。伪代码是运行时所生成的代码的“基于文本”的详细算法设计。

[0038]

```
int *a;
int *b;
```

[0039]

```
for (i = in 256 number sections)
{
        for ( # of simd_passes)
        {
          load (b -> simd_register)
         compare_gt ( simd_register, 100);
         // 长度为 8 比特的位向量
        //在正确位置存储在 result_vec 中
        }

        for (j= ones in bitvector)
          {
              calculate original row_id = i* 256 + j;
              return_to_user (a[row_id]);
           }
}
```

[0040] 表3

[0041] 图4是示出根据一个实施例的与用于通过在计算机处理器处的寄存器内的数据集的部分上迭代来评估与数据集相关的请求的过程相关联的示例性数据结构的框图。Table_1 405是包括列“a”410和列“b”415的数据的表。Table_1 405可以类似于上面关于图1呈现的表2,但是包括列“a”和列“b”而没有列“c”。

[0042] 该请求可以包括对table_1 405的一个或多个查询。例如,查询可以是诸如表1中定义的查询。基于对table_1 405中的数据的查询,可以确定其中来自值“b”大于100的列“a”的值。如关于图3所讨论的,可以评估所定义的请求以确定代码生成计划。代码生成计划可以定义来自列“b”的值按照部分通过过程寄存器进行评估以优化性能。过程寄存器可以与在多个数据上执行单个指令(SIMD)相关联。可以基于在用于评估数据的程序指令中定义的断言,迭代地对如阵列410中呈现的列“b”的值进行迭代地评估。在来自表1的示例性查询中,断言在where子句“b>100”中定义。来自列“b”的值的阵列410可以被划分为多个部分。该部分的大小可以与其中运行程序的计算机系统的计算机处理器处的SIMD寄存器407的大小对应。可以确定SIMD寄存器407的大小。例如,可以假设来自列“b”的值是32位数,并且SIMD寄存器可以具有一组8位寄存器部分,以定义256位大小的寄存器。在256位的SIMD寄存器407中,可以包括256位值,其可以与基于在程序指令中定义的断言(即“b>100”)对来自阵列

410的256个数字的评估结果对应。

[0043] 可以针对阵列417定义每个部分包括256个数字的一组部分。例如，来自阵列417的值是32位宽的值(整数或浮点数)。基于对第一部分(来自阵列417的部分“1”425)的评估，位阵列b’420可以被定义为包括与来自部分“1”425的值的评估结果对应的位值。位阵列b’420与来自列“b”410的值的部分“1”425对应。位阵列b’420可被划分为8个数(32位大小)的部分，以与SIMD寄存器407中定义的8位的部分对应。位阵列b’420可以作为位向量“1”430存储在SIMD寄存器407中并且基于确定的值，可以定义结果输出数据。SIMD寄存器407中的位向量“1”430可以被划分成寄存器部分，诸如寄存器部分440。例如，寄存器部分440可以包括8位值，并且大小为8位。例如，基于位阵列b’420中的正位值的位置，可以定义来自table_1405的列“a”的相应值。迭代地，可以针对列“b”定义另外的位阵列以与阵列410中定义的另外的部分对应。

[0044] 基于针对位阵列b’420的所确定的值，可以利用诸如0或1的值填充SIMD寄存器。“0”与来自b的其中不满足在来自程序指令的断言中定义的条件的值对应。“1”与来自b的其中满足来自程序指令的断言中定义的条件的值对应。基于可以解释为正位值的“1”，可以确定来自列“a”的对应行以便针对列a生成结果数据的第一集合--结果数据集“1”450。结果数据集“1”450与被评估的第一部分“1”425对应。结果数据集“1”450可由处理器传递。例如，可以经由与计算机系统相关联的显示屏上的用户界面来传递结果数据集“1”450。在第二次迭代期间，可以生成第二位向量来代替SIMD寄存器中的第一位向量，其中可以定义评估来自阵列417的第二部分的位值。基于第二位向量中的正位值，可以定义来自列“a”的相应值以便从列“a”生成256个结果值的第二结果数据集。

[0045] 迭代地，可以评估在阵列410中定义的所有部分，以确定例如将一个接一个地被存储到处理器的SIMD寄存器中的位向量“1”430的位向量。以这种迭代的方式，基于SIMD寄存器中生成的结果，所提供的结果是以片段提供的。因此，对来自列“b”的给定部分的评估结果可能不会被推送回到计算环境的主存储器中。

[0046] 评估请求的过程可能涉及包括用于评估来自数据库表的数据的一个或多个简单断言(例如上述断言“b>100”)的复杂断言。例如，复杂断言可能在以下内容中:WHERE(s>10AND t<20)或(u＝30和v>40)。参数s、t、u和v可以指代来自数据库表的列。示例的复杂断言包括四个简单断言。对于这样的复杂断言，大小为256的位向量b1用于第一断言“s>10”，大小为256的位向量b2用于第二断言“t<20”，b3和b4等等。然后，可以采取四个位向量，并以“SIMD”方式进行连接。可以在原始定义的复杂断言中被定义的连接处执行SIMD操作。例如，可以对位向量执行诸如(b1SIMD_AND b2)SIMD_OR(b3SIMD_AND b4)的所执行的SIMD操作。以这种方式，仅以3个SIMD指令执行256行的“与”和“或”操作，而不是如果以标量方式执行所需要运行的256*3个指令。由于断言评估与运行的其余部分密切耦合，因此表数据和位向量不会通过缓存级别进行上下传送。

[0047] 图5是示出根据一个实施例的用于通过处理器寄存器迭代评估数据的过程500的流程图。在510，生成用于处理内存数据库系统中的数据集的可运行计划。可运行计划与用于通过在内存数据库系统的处理器中的SIMD寄存器处理数据的代码生成相关联。可运行计划与运行时生成的程序指令相关联。基于包括与数据集的第一部分相关联的断言的源代码定义运行时生成的程序指令。运行时生成的程序指令可以基于所接收到的包括对数据集的

查询的请求。查询可以定义与数据集的第一部分相关联的断言。例如，数据集可以是诸如表2中呈现的表的数据库表。运行时生成的程序指令可以基于包括在软件应用中的程序语句。数据集的第一部分可以是来自数据库表中的一个或多个列。利用表1中的示例性查询，与断言相关联的数据集的第一部分是来自列“b”的数据，断言是验证来自列“b”的值是否大于或等于100的语句。

[0048] 在520，数据集的第一部分被划分成一个或多个数据部分。数据部分包括与在处理器中存储的可分配到寄存器中的多个位值相对应的多个数据元素。数据集的第一部分可以被划分成与在520定义的数据部分相对应的多个大小相同的块。可以用于过程500的过程中的寄存器可以是与在多个数据上执行单个指令相关联的寄存器(SIMD寄存器)。可以在生成运行计划之前确定SIMD寄存器的大小。SIMD寄存器的大小可用于确定要包括在数据部分中的值的数量。来自数据部分的值基于程序指令中定义的断言被迭代地评估。数据部分可以是例如图4的部分“1”425。在应用于来自表2的数据的表1的示例性查询中，断言的评估可以存储为位向量中的位值，例如0或1。位向量可以是位向量“1”430。例如，如果SIMD寄存器具有256位的大小，则可以评估来自数据集的第一部分的256个数字，以确定以位值呈现的256个评估结果。

[0049] 在530，对数据的第一部分定义的一个或多个数据部分被迭代地评估以确定一个或多个位向量。一个或多个位向量被迭代地存储到在处理器处的SIMD寄存器中。在540，在处理器处，迭代地确定并提供一个或多个结果数据集。一个或多个结果数据集的确定基于SIMD寄存器处所确定的一个或多个位向量，并且还基于该数据集。基于针对当前迭代定义的相应位向量中的正位值确定结果数据集。当确定正位值时，可以确定来自数据集的相应行标识符。这样的行标识符可用于确定用于生成由程序指令定义的结果数据的数据集的值。例如，利用基于来自表1的查询的指令，可以基于根据断言(b>100)对来自列“b”的部分进行迭代评估来确定表2的行标识符。可以使用这样的行标识符来确定来自列“a”的哪些值将被包括在如查询(Select a)中定义的结果数据集中。在另一示例中，结果数据可以是存储在数据集中的值的导出值。

[0050] 图6是示出根据一个实施例的用于通过处理器寄存器运行与数据集上的操作相关的程序指令的系统600的框图。系统600包括主存储器610、三个高速缓存存储器级别L1 625、L2 620和L3 615以及CPU 630。主存储器610包括基于接收到的请求675生成的编译程序670。所接收的请求675可以是诸如图3的请求305。基于接收到的包括对存储在主存储器610上的表660中的数据的查询的请求675，可以生成程序指令。程序指令可以是例如图1的程序指令125，编译程序670可以基于这样的程序指令生成。编译程序670可以由诸如图1的运行引擎175的运行引擎生成。编译程序670可以与表660相关联。表660包括列的集合，例如列“a”和列“b”。表660可以是如上所呈现的表2。编译程序670可以定义如何使用来自表660中的数据集的数据来生成结果数据并将其提供给终端用户。所生成的结果数据可以通过CPU 630提供。

[0051] 在一个实施例中，来自表660的列可以与在编译程序670中定义的断言相关联。列可以是列“b”。断言可以在请求(查询)675中定义。例如，查询请求可以被定义为SQL字符串：“select a,where b>100,from table_1”。列“b”可以是在运行针对程序670在运行时生成的代码期间在部分中被迭代地评估。根据所定义的断言执行列“b”中的数据的迭代评估，以

确定值是否符合断言。列“b”中的值可以被划分为多个部分，使得来自部分的评估的结果位值可以适合于CPU 630处的SIMD寄存器635。部分的大小，即来自列“b”的值的数量，可以被定义为与SIMD寄存器635的大小对应。以这种方式，可以定义诸如b'和b″之类的部分。在第一迭代645中，可以基于程序指令来评估来自列“b”650的第一部分。可以评估来自第一部分的值以确定它们是否符合被定义的断言。在第一次迭代期间对值的评估通过中间操作利用高速缓冲存储器级别L1 625、L2 620和L3 615来存储数据。在第一迭代645期间，第一部分的评估结果被存储在SIMD寄存器的位向量“1”640处。位向量“1”640可以是诸如图4中的位向量“1”430。基于存储在位向量“1”640处的值，可以定义与来自表660的行对应的行标识符。可以确定的行标识符(row_ids)与来自位向量“1”640的正位值相关联，其对应于表660中满足断言的行。当针对第一迭代确定这样的行标识符时，可以查询表660以提供与列“b”650的第一部分对应的第一结果数据集。第一结果数据集可以从主存储器被调用并通过CPU 630提供。第一结果数据集可以包括来自列“a”的值，该值基于所确定的行标识符被调用。

[0052] 在随后的迭代中，SIMD寄存器635可以擦除先前存储的位向量“1”640，并允许随后的位向量存储在那里。在随后的迭代中，待确定的行标识符可能不与来自位向量的值的连续数相对应。可以使用公式来确定与初始数据库表660相关联的行标识符。在位向量的值的位置、随后的迭代次数与在数据库660的值的位置之间存在相关性。例如，公式(1)可以用于迭代地确定与位向量中的正位值相对应的行的行标识符。在示例性公式(1)中，当SIMD寄存器635的大小为256位时，位向量中所存储的值的数目为256。

[0053] (1)row_id＝i*256+j，其中“i”是后续的迭代次数，其中“i”从0开始到n，“j”是位向量中正位值的位置。

[0054] 以迭代方式，可以通过后续查询表660迭代地提供结果数据集。因此，存储在结果位向量(例如，图4的结果向量“1”430)中对列“b”650的数据评估可以不用从CPU 630向主存储器610来回推送数据，以确定最终迭代中的整个结果集。通过利用SIMD寄存器635，基于程序指令对来自数据库表的值进行评估的性能得到提高。经由代码生成，SIMD扫描数据与执行期间的数据获取的紧密耦合提高了评估的性能。避免从CPU 630向主存储器610来回推送数据，这优化了对来自数据库表的值的评估过程。

[0055] 图7是示出根据一个实施例的用于通过在计算机处理器处生成位向量以确定结果数据来对与数据集上的操作有关的请求进行评估的过程700的流程图。

[0056] 在705，在内存数据库系统中接收该请求。该请求包括与包括数据集的数据库表的列相关联的断言。在710，确定内存数据库系统中处理器处的SIMD寄存器的大小。确定SIMD寄存器的大小以确定将要包括在SIMD寄存器中的位值的数量。在715，生成用于处理内存数据库系统中的数据库表的可运行代码。可运行计划是将由与内存数据库系统相关联的运行引擎运行的物理计划。在720，与断言相关联的数据库表的列被加载在内存数据库系统的主存储器中。主存储器可以是诸如图6的主存储器610。程序指令被存储在内存数据库系统的主存储器中。在725，来自数据库表的列的数据被分成一个或多个数据部分。数据部分包括多个数据元素，其数量与将要分配给处理器处的SIMD寄存器的位值的数量对应。在730，确定第一位向量，该第一位向量基于来自该列的数据的第一数据部分的数据的值的评估被确定。基于程序指令中定义的断言执行评估。第一位向量包括位值，例如0和1，其与来自第一数据部分的值与断言的符合性的评估结果对应。在处理器处，第一位向量被加载到SIMD寄

存器中。在735，确定来自数据库表的列的行的标识符。行的标识符(ids)与来自数据库表的满足断言的行对应。例如，在图4的例子中，列“a”的值来自于与所确定的行ids对应的所确定的行，其中列“b”的值高于100。

[0057] 一些实施例可以包括被写为一个或多个软件组件的上述方法。这些组件以及与每个组件相关联的功能可被客户端、服务器、分布式或对等计算机系统使用。这些组件可以用与例如功能性、声明性、程序性、面向对象、较低级语言等的一个或多个编程语言相对应的计算机语言来编写。它们可以经由各种应用程序编程接口链接到其他组件，然后编译成服务器或客户端的一个完整应用程序。或者，组件可以在服务器和客户端应用程序中实现。此外，这些组件可以经由各种分布式编程协议链接在一起。一些示例性实施例可以包括用于在分布式编程环境中实现这些组件中的一个或多个组件的远程过程调用。例如，逻辑级可以驻留在远离包含接口级(例如，图形用户界面)的第二计算机系统的第一计算机系统上。这些第一计算机系统和第二计算机系统可以在服务器-客户端、对等或某个其他配置中进行配置。从移动和手持设备到瘦客户端以及胖客户端甚至其他服务器，客户端可以在复杂性方面不同。

[0058] 上述软件组件作为指令被有形地存储在计算机可读存储介质上。术语“计算机可读存储介质”应被视为包括存储一组或多组指令的单个介质或多个介质。术语“计算机可读存储介质”应被认为包括能够经历一组物理变化的物理物品，以物理存储、编码或以其他方式携带用于由计算机系统执行的一组指令，该指令使计算机系统执行本文所描述、表示或示出的任何方法或处理步骤。计算机可读存储介质可以是非暂时计算机可读存储介质。非暂时计算机可读存储介质的示例包括但不限于：诸如硬盘，软盘和磁带的磁介质；诸如CD-ROM、DVD和全息设备的光学介质；磁光介质；以及专门配置用于存储和执行的硬件设备，例如专用集成电路(“ASIC”)、可编程逻辑器件(“PLD”)以及ROM和RAM器件。计算机可读指令的示例包括诸如由编译器生成的机器代码和包含由使用解释器的计算机运行的较高级代码的文件。例如，可以使用Java、C++或其他面向对象的编程语言和开发工具来实现实施例。另一实施例可以以硬连线电路代替，或与机器可读软件指令组合来实现。

[0059] 图8是示例性计算机系统800的框图。计算机系统800包括运行存储在计算机可读存储介质855上的软件指令或代码以执行上述方法的处理器805。处理器805可以包括多个核。计算机系统800包括用于从计算机可读存储介质855读取指令并将指令存储在存储器810或随机存取存储器(RAM)815中的介质读取器840。存储器810提供用于保存静态数据的大空间，其中至少可以存储一些指令以待以后执行。根据一些实施例，诸如一些内存计算系统实施例，RAM 815可以具有足够的存储容量来存储RAM 815中处理所需的大部分数据，而不是在存储器810中。在一些实施例中，处理所需的所有数据可以存储在RAM 815中。存储的指令可以被进一步编译以生成指令的其他表示并动态地存储在RAM 815中。处理器805从RAM 815读取指令并按照指示执行动作。根据一个实施例，计算机系统800还包括输出设备825(例如，显示器)，以将执行结果中的至少一些提供为输出，包括但不限于提供给用户的视觉信息，以及为用户或另一设备提供输入数据和/或以其他方式与计算机系统800交互的装置的输入设备830。这些输出设备825和输入设备830中的每一个可以由一个或多个附加外围设备连接，以进一步扩展计算机系统800的能力。可以提供网络通信器835以将计算机系统800连接到网络850，并且依次连接到与网络850连接的其他设备，包括其他客户端、服

务器、数据存储和接口。计算机系统800的模块经由总线845互连。计算机系统800包括访问数据源860的数据源接口820。数据源860可以经由一个或多个以硬件或软件实现的抽象层来访问。例如，数据源860可以被网络850访问。在一些实施例中，可以经由诸如语义层之类的抽象层访问数据源860。

[0060] 数据源是信息资源。数据源包括可以能够进行数据存储和检索的数据的源。数据源可以包括诸如关系、事务、分层、多维(例如，OLAP)、面向对象的数据库等的数据库。其他的数据源包括表格数据(例如，电子表格、分隔文本文件)、使用标记语言做标签的数据(例如，XML数据)、事务数据、非结构化数据(例如文本文件、屏幕刮除(scrapping))、分层数据(例如在文件系统中的数据、XML数据)、文件、多个报告以及通过诸如基础软件系统(例如，ERP系统)生成的开放数据库连接(ODBC)等已建立的协议可访问的任何其他数据源等等。数据源还可以包括其中数据不被有形地存储或以其他方式临时存在的数据源，例如数据流、广播数据等。这些数据源可以包括相关联的数据基础、语义层、管理系统、安全系统等。

[0061] 在上述描述中，阐述了许多具体细节以提供对实施例的透彻理解。然而，相关领域的技术人员将认识到，可以在没有一个或多个具体细节或使用其他方法，组件，技术等的情况下实施实施例。在其他实例中，未示出或详细描述公知的操作或结构。

[0062] 尽管本文所示的和描述的过程包括一系列步骤，但是将理解，不同的实施例不受所示出的步骤顺序的限制，因为一些步骤可以以不同的顺序发生，一些步骤可以与除在此示出和描述的之外的其他步骤同时进行。此外，根据一个或多个实施例，可以不需要所有示出的步骤来实现方法。此外，应当理解，可以与本文所示和所描述的装置和系统相关联地以及与未示出的其他系统相关联地实现该过程。

[0063] 实施例的上述描述和说明，包括摘要中所描述的内容并不旨在是穷尽性地或将一个或多个实施例限制于所公开的精确形式。虽然为了说明的目的在这里描述了一个或多个实施例的具体实施例和示例，但是如相关领域的技术人员将认识到的，在一个或多个实施例的范围内，可以进行各种等同的修改。这些修改可以鉴于上述详细描述做出。确切地说，范围将由所附权利要求确定，这些权利要求将根据所建立的权利要求构造的理论进行解释。

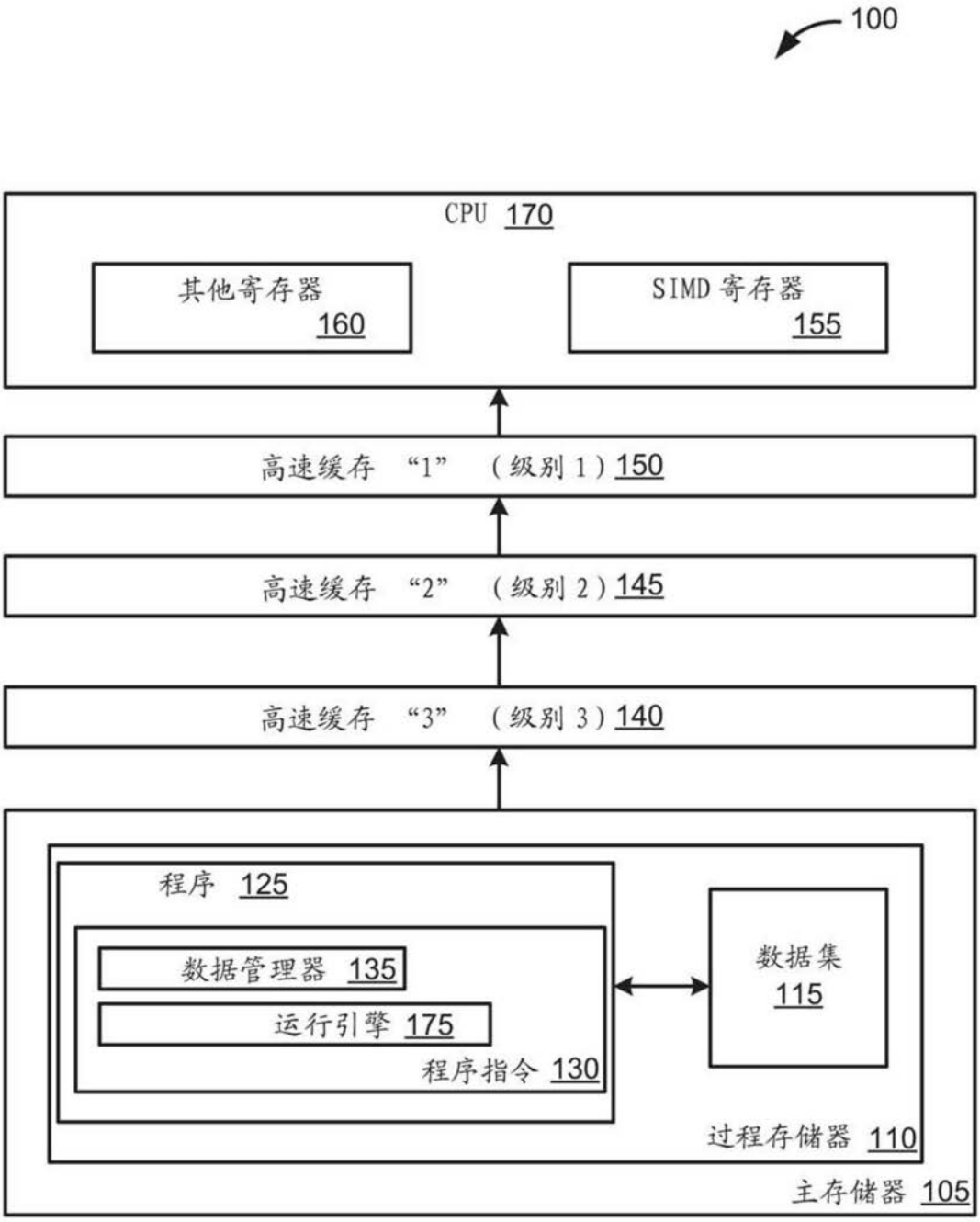

100
CPU 170
其他寄存器 160
SIMD 寄存器 155
高速缓存 “1” (级别 1) 150
高速缓存 “2” (级别 2) 145
高速缓存 “3” (级别 3) 140
程序 125
数据管理器 135
运行引擎 175
程序指令 130
数据集 115
过程存储器 110
主存储器 105

图1

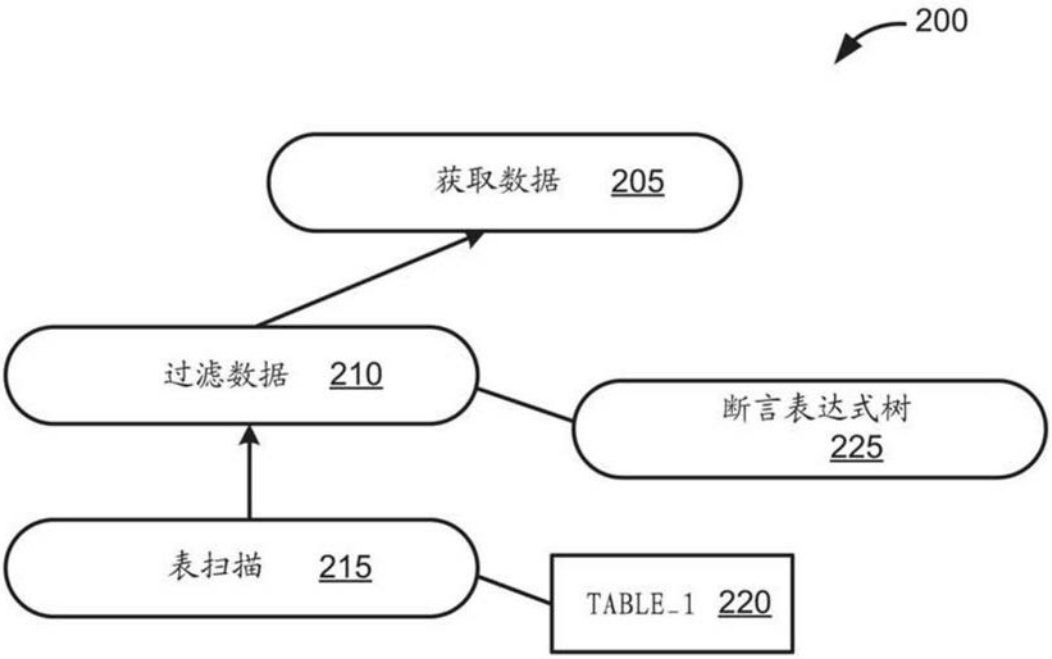

200
获取数据 205
过滤数据 210
断言表达式树 225
表扫描 215
TABLE_1 220

图2A

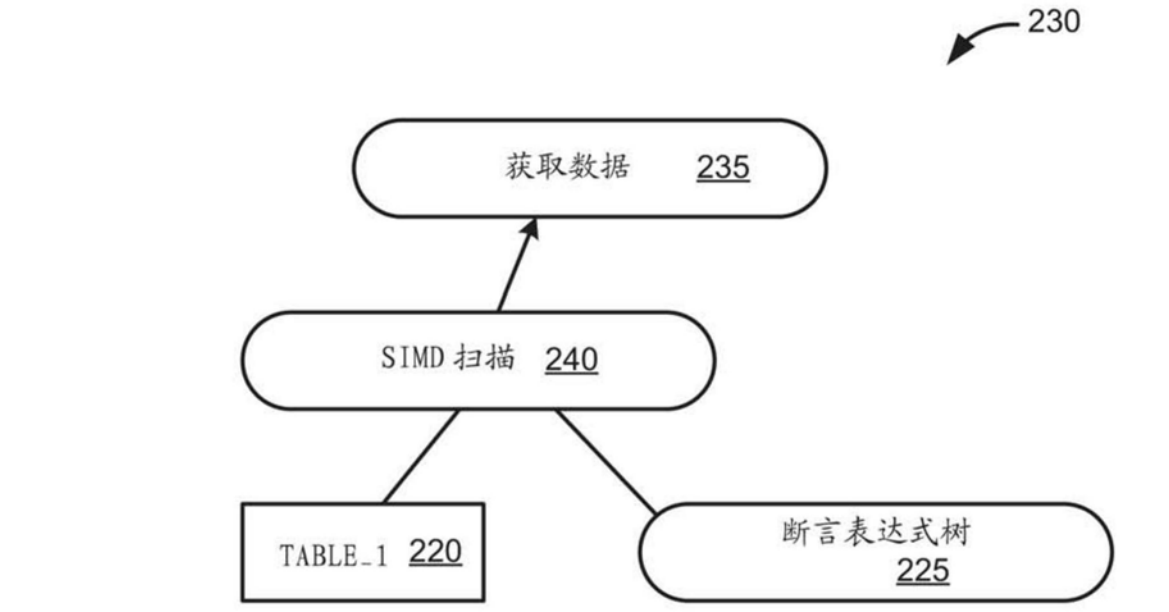

230
获取数据 235
SIMD 扫描 240
TABLE_1 220
断言表达式树 225

图2B

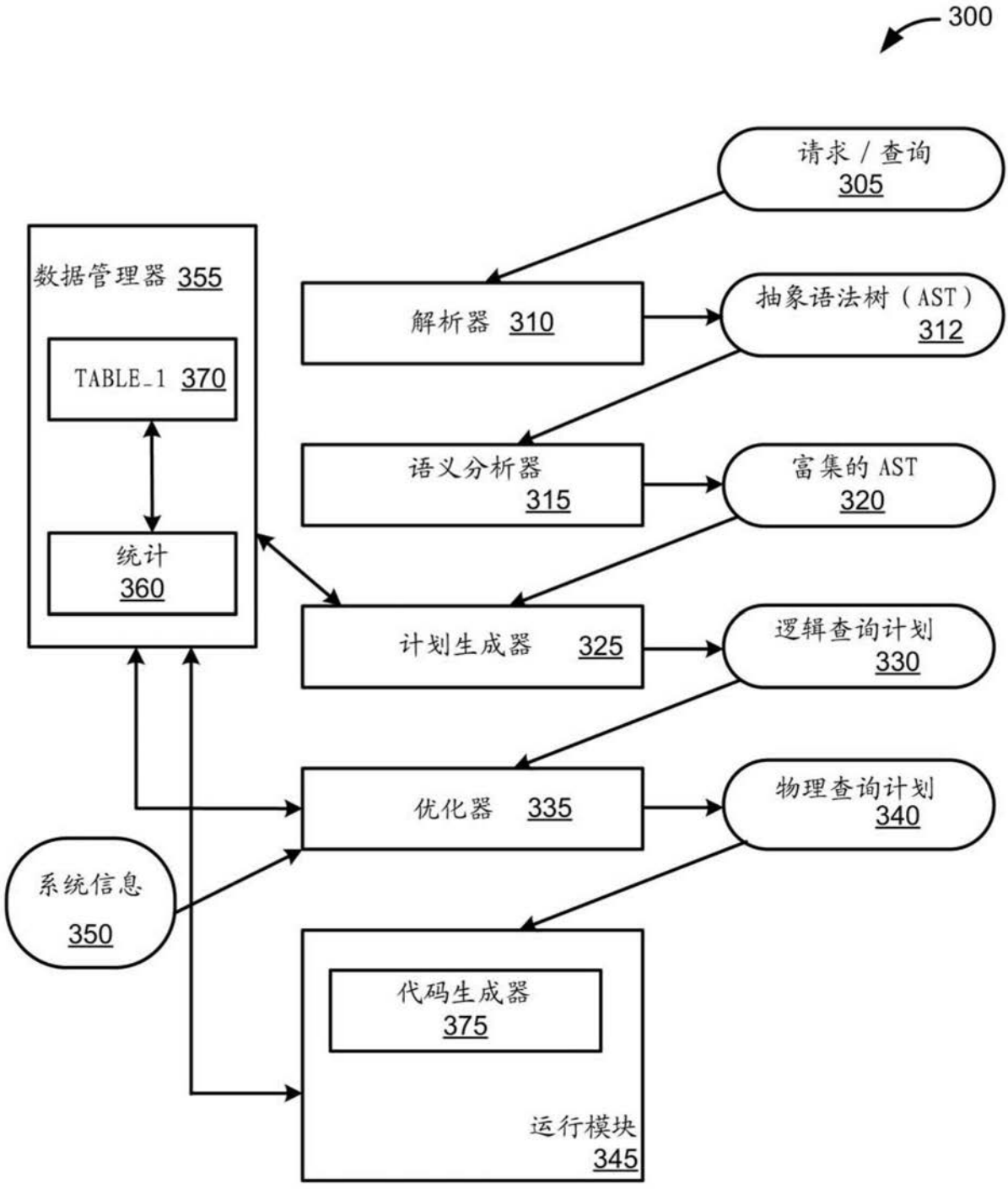

300
请求 / 查询
305
数据管理器 355
TABLE_1 370
统计
360
解析器 310
抽象语法树（AST）
312
语义分析器
315
富集的 AST
320
计划生成器 325
逻辑查询计划
330
优化器 335
物理查询计划
340
系统信息
350
代码生成器
375
运行模块
345

图3

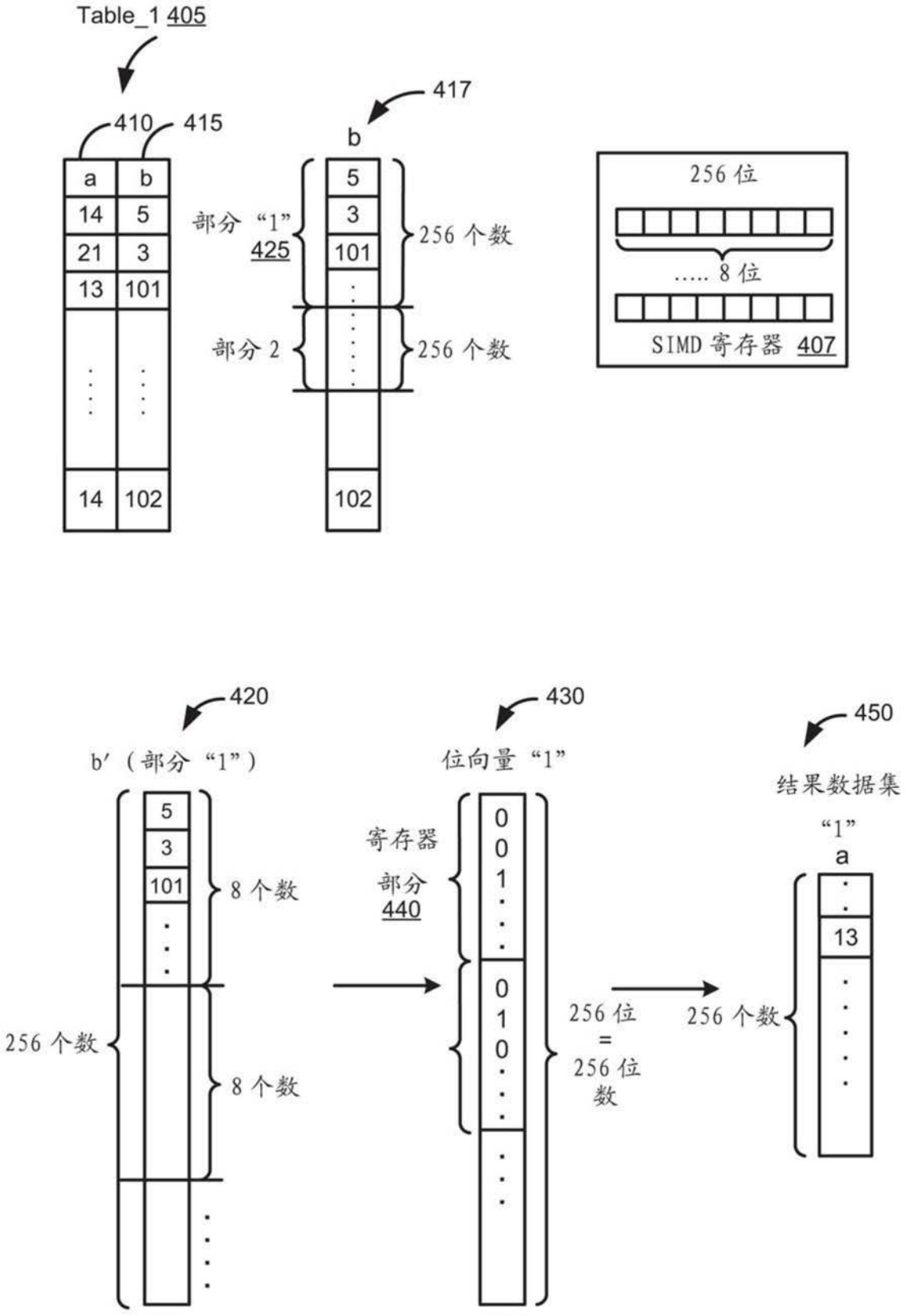


Table_1 405
410
415
a
b
14
5
21
3
13
101
14
102
417
b
5
3
101
102
部分 "1" 425
256 个数
部分 2
256 个数
256 位
..... 8 位
SIMD 寄存器 407
420
b′（部分 "1"）
5
3
101
8 个数
256 个数
8 个数
430
位向量 "1"
寄存器
部分
440
0
0
1
0
1
0
256 位
=
256 位
数
450
结果数据集
"1"
a
13
256 个数

图4

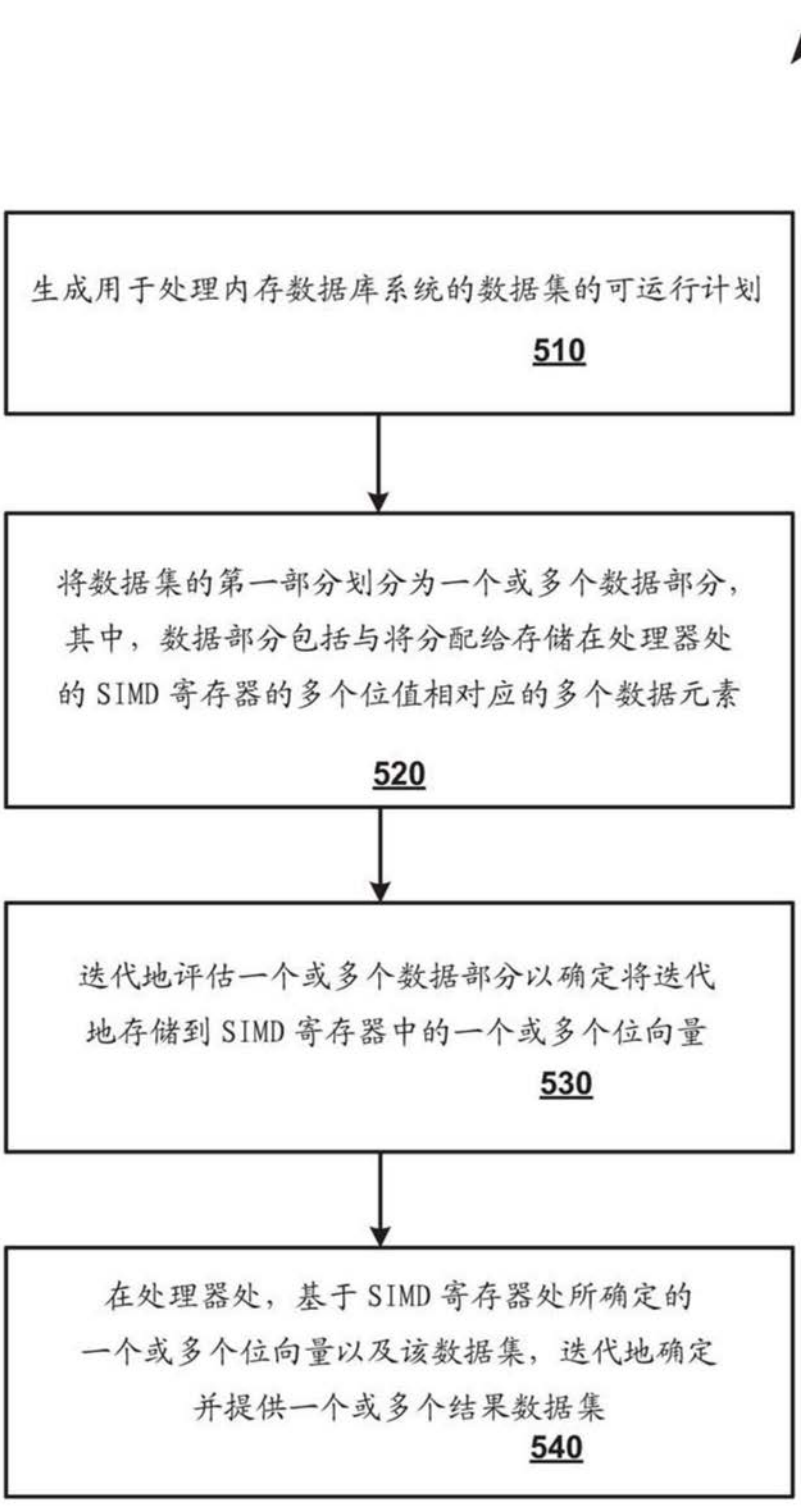

500
生成用于处理内存数据库系统的数据集的可运行计划
510
将数据集的第一部分划分为一个或多个数据部分，其中，数据部分包括与将分配给存储在处理器处的 SIMD 寄存器的多个位值相对应的多个数据元素
520
迭代地评估一个或多个数据部分以确定将迭代地存储到 SIMD 寄存器中的一个或多个位向量
530
在处理器处，基于 SIMD 寄存器处所确定的一个或多个位向量以及该数据集，迭代地确定并提供一个或多个结果数据集
540

图5

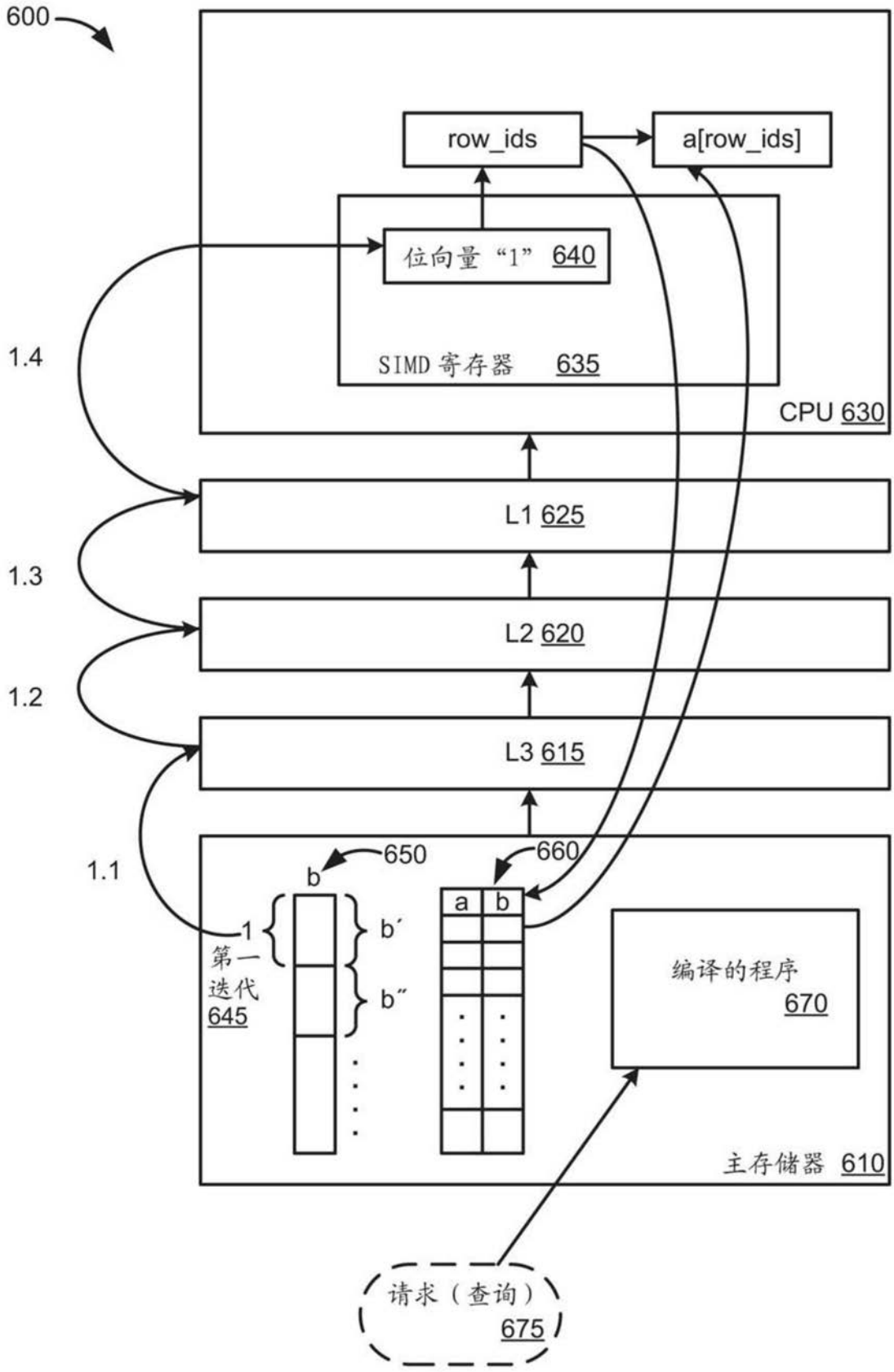

600
row_ids
a[row_ids]
位向量 “1” 640
SIMD 寄存器 635
CPU 630
1.4
L1 625
1.3
L2 620
1.2
L3 615
1.1
650
660
b
a
b
1
第一迭代 645
b′
b″
编译的程序 670
主存储器 610
请求（查询） 675

图6

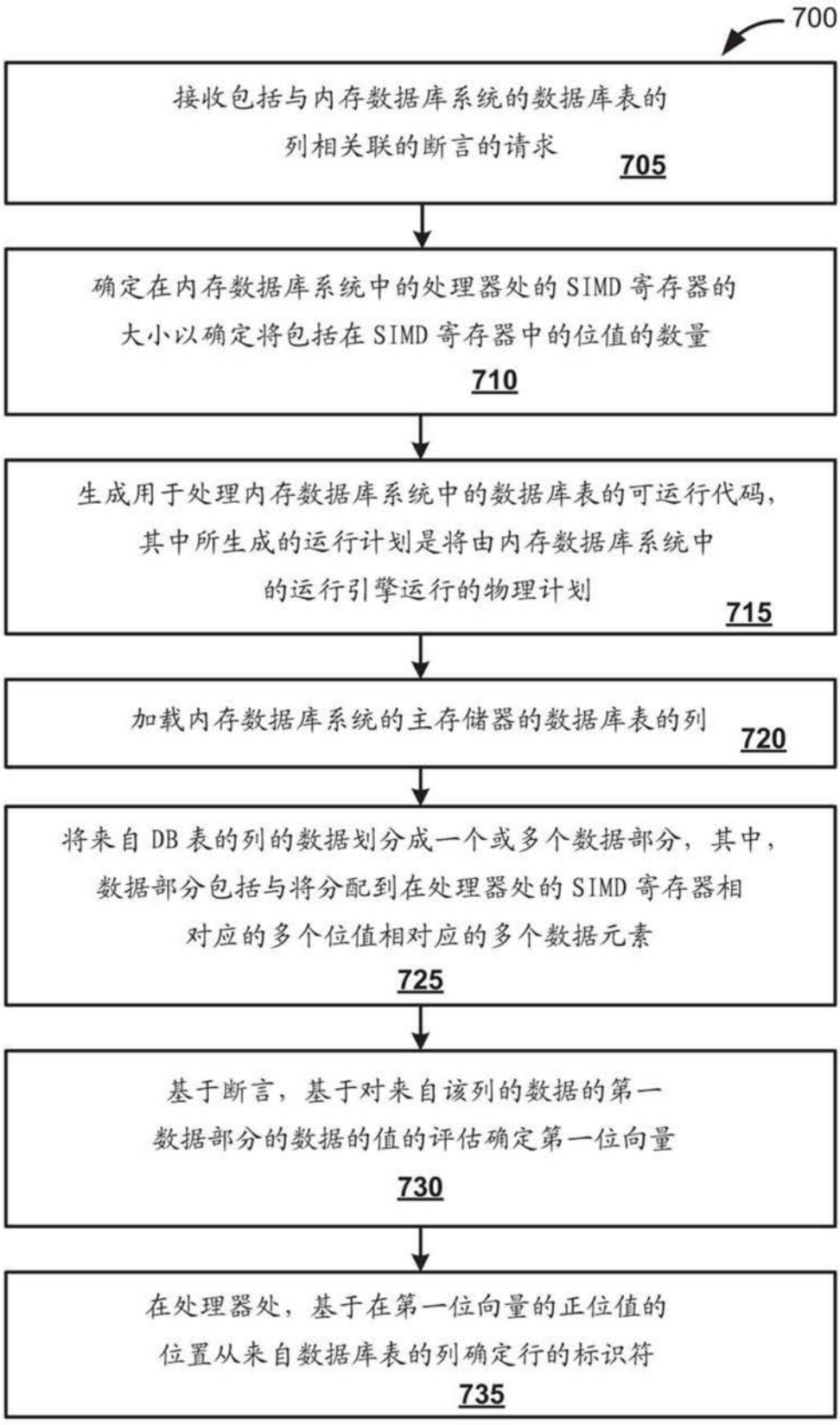

700
接收包括与内存数据库系统的数据库表的列相关联的断言的请求
705
确定在内存数据库系统中的处理器处的SIMD寄存器的大小以确定将包括在SIMD寄存器中的位值的数量
710
生成用于处理内存数据库系统中的数据库表的可运行代码，其中所生成的运行计划是将由内存数据库系统中的运行引擎运行的物理计划
715
加载内存数据库系统的主存储器的数据库表的列
720
将来自DB表的列的数据划分成一个或多个数据部分，其中，数据部分包括与将分配到在处理器处的SIMD寄存器相对应的多个位值相对应的多个数据元素
725
基于断言，基于对来自该列的数据的第一数据部分的数据的值的评估确定第一位向量
730
在处理器处，基于在第一位向量的正位值的位置从来自数据库表的列确定行的标识符
735

图7

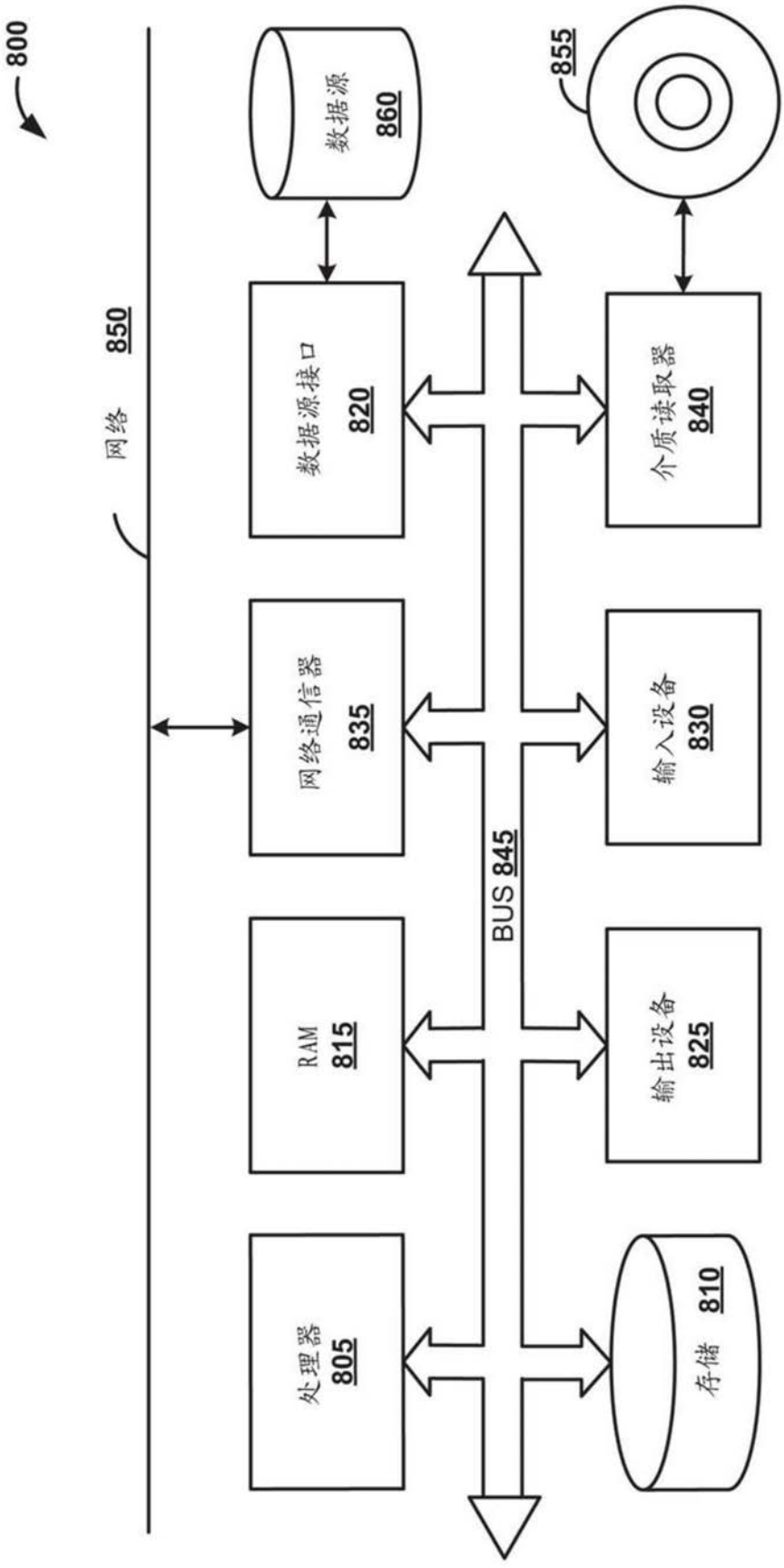

800
网络 850
数据源 860
855
数据源接口 820
介质读取器 840
网络通信器 835
输入设备 830
BUS 845
RAM 815
输出设备 825
处理器 805
存储 810

图8