大规模数据聚类分析的并行化

申请号：201110183886.4
申请日：2011-06-30

申请(专利权)人　SAP股份公司
地址　德国瓦尔多夫
发明(设计)人　黎文宪　孙谷飞

主分类号　G06F17/30(2006.01)I
分类号　G06F17/30(2006.01)I
公开(公告)号　102855259A
公开(公告)日　2013-01-02
专利代理机构　北京市柳沈律师事务所 11105
代理人　邵亚丽

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102855259 A
(43)申请公布日 2013.01.02

(21)申请号 201110183886.4

(22)申请日 2011.06.30

(71)申请人 SAP 股份公司
地址 德国瓦尔多夫

(72)发明人 黎文宪 孙谷飞

(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 17/30(2006.01)

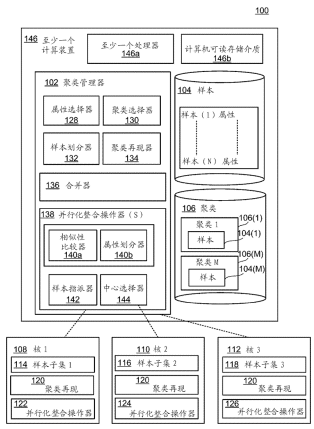
权利要求书 2 页 说明书 20 页 附图 5 页

(54)发明名称

大规模数据聚类分析的并行化

(57)摘要

本发明提供大规模数据聚类分析的并行化处理的方法和系统。聚类选择器可以确定多个样本聚类，以及可以在多个处理核中的每一个处再现所述多个样本聚类。样本划分器可以将存储在数据库中的具有关联属性的多个样本划分为数目相应于所述多个处理核的数目的样本子集，并且可以将所述数目的样本子集中的每一个与所述多个处理核中的对应一个关联。整合操作器可以基于所述多个处理核中的每个对应核处的每个样本子集的每个样本的关联属性，执行所述每个样本相对于在所述对应处理核处再现的多个样本聚类中的每一个的比较。



CN 102855259 A

1. 一种包括记录在计算机可读介质上的指令的计算机系统，该系统包括：

聚类选择器，其被配置为确定多个样本聚类，以及在多个处理核中的每一个处再现所述多个样本聚类；

样本划分器，其被配置为将存储在数据库中的具有关联属性的多个样本划分为数目对应于所述多个处理核的数目的样本子集，并且还被配置为将所述数目的样本子集中的每一个与所述多个处理核中的对应一个相关联；以及

整合操作器，其被配置为基于所述多个处理核中的每个对应核处的每个样本子集中的每个样本的关联属性，执行所述每个样本相对于在所述对应处理核处再现的多个样本聚类中的每一个的比较。

2. 如权利要求 1 所述的系统，其中，所述聚类选择器被配置为通过图形用户界面 (GUI) 从用户接收多个样本聚类的数目。

3. 如权利要求 1 所述的系统，包括合并器，其被配置为合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本来填充所述样本聚类。

4. 如权利要求 1 所述的系统，其中，样本子集的数目等于所述多个处理核的数目，并且其中，每个样本子集包括相等数目的样本。

5. 如权利要求 1 所述的系统，还包括属性划分器，其被配置为将与每个样本关联的属性划分为属性子集，以便在执行所述比较期间对其进行并行处理。

6. 如权利要求 1 所述的系统，其中，所述比较包括在多个处理核中的每一个处执行的、每个样本子集中的每个样本与每个聚类的中心之间的相似性比较。

7. 如权利要求 6 所述的系统，其中，使用包括在每个聚类中的样本的平均属性值来确定每个聚类的中心。

8. 如权利要求 6 所述的系统，其中，所述整合操作器被配置为基于所述比较将样本从第一聚类重新指派到第二聚类。

9. 如权利要求 8 所述的系统，包括合并器，其被配置为合并所述比较的比较结果以及被配置为根据需要使用经合并的比较结果来更新每个聚类的每个中心的值。

10. 如权利要求 9 所述的系统，其中，所述合并器被配置为基于被重新指派的样本的数目来确定每个聚类内样本的稳定性。

11. 一种计算机实现方法，包括：

确定存储在数据库中的具有关联属性的多个样本；

确定多个样本聚类；

在多个处理核中的每一个处再现所述多个样本聚类；

将所述多个样本划分为数目与所述多个处理核的数目相对应的样本子集；

将所述数目的样本子集中的每一个与所述多个处理核中的对应一个相关联；以及

基于在所述多个处理核的每个对应核处的每个样本子集中的每个样本的关联属性，执行所述每个样本相对于在对应处理核处再现的多个样本聚类中的每一个的比较。

12. 如权利要求 11 所述的方法，还包括合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本来填充所述样本聚类。

13. 如权利要求 11 所述的方法，执行所述比较还包括将与每个样本相关联的属性划分为属性子集，以便在执行所述比较期间对其进行并行处理。

14. 如权利要求 11 所述的方法，其中，执行所述比较进一步包括在多个处理核中的每一个处执行每个样本子集中的每个样本与每个聚类的中心之间的相似性比较。

15. 一种计算机程序产品，该计算机程序产品被有形地具体实施在计算机可读介质上并且包括指令，所述指令当被运行时被配置为执行如下步骤：

确定存储在数据库中的具有关联属性的多个样本；

确定多个样本聚类；

在多个处理核中的每一个处再现所述多个样本聚类；

将所述多个样本划分为数目与所述多个处理核的数目相对应的样本子集；

将所述数目的样本子集中的每一个与所述多个处理核中的对应一个相关联；以及

基于在所述多个处理核的每个对应核处的每个样本子集中的每个样本的关联属性，执行所述每个样本相对于在对应处理核处再现的多个样本聚类中的每一个的比较。

16. 如权利要求 15 所述的计算机程序产品，其中，所述指令当被运行时被配置为：合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本来填充所述样本聚类。

17. 如权利要求 15 所述的计算机程序产品，其中，所述指令当被运行时被配置为：将与每个样本关联的属性划分为属性子集，以便在执行所述比较期间对其进行并行处理。

18. 如权利要求 15 所述的计算机程序产品，其中，所述比较包括在多个处理核中的每一个处执行的、每个样本子集中的每个样本与每个聚类的中心之间的相似性比较。

19. 如权利要求 15 所述的计算机程序产品，其中，所述指令当被运行时被配置为：基于所述比较将样本从第一聚类重新指派到第二聚类。

20. 如权利要求 19 所述的计算机程序产品，其中，所述指令当被运行时被配置为：基于所述被重新指派的样本的数目确定每个聚类内的样本的稳定性。

大规模数据聚类分析的并行化

技术领域

[0001] 本说明书涉及并行处理。

背景技术

[0002] 并行处理通常指的是将一个或多个计算任务划分为两个或更多子任务的概念，每个子任务可以在单独的处理器上运行。换句话说，把一个较大的计算任务分成若干子任务，然后将这些子任务分配到两个或多个处理器上执行。与仅使用所述处理器中的一个处理器可能达到的效果相比，通过使用这样的并行处理技术，在许多情况下，可以以更快速并且更有效的方式完成计算任务。

[0003] 然而，实际上，可能存在大量障碍使得难以或者无法执行给定计算任务的并行处理，特别是对于特定类型或者类别的计算任务。举例来说，一般地，要求至少与并行处理关联的计算开销要小。举例来说，对于一项将并行运行的给定计算任务来说，可能需要将与该计算任务相关的数据的部分或者全部复制到将使用的每一个处理器中。更一般来说，可以理解，最好没有为并行处理而进行的数据分割或复制而带来的计算开销。而且，在并行运行的处理器中的任意一个处的延迟或困难可能导致该任务的计算整体上的延迟。而且，因为子任务在两个或更多处理器处完成，所以可能需要计算资源来整合在两个或更多处理器中的每一个处执行的并行处理的结果，以便得到该计算任务整体的统一计算结果。因此，由于可能与在并行处理中处理子任务的划分、计算以及整合相关联的这些计算开销，在许多情况下利用并行处理技术可能是不现实的。

[0004] 举例来说，特定类型的计算任务可能需要对相对来说非常大的数据集的每一元素与相对较小的数据集的每一元素的比较或者其它操作。例如，在一个为了说明的特定例子中，可能出现：需要将一个包括三百万个记录——每一个记录有 300 个属性——的数据集与第二数据集的 100 个记录中的每一个相比较（诸如，举例来说，当希望将三百万个记录中的每一个分组到被认定是最相似的 100 个聚类中的一个中时）。因此，这样的计算将需要三百万乘 300 再乘 100 次单独计算。而且，将数据集划分以使用单独的处理器处理是不可行的，因为该计算的本质是：将第一较大数据集的全部记录和属性与第二较小数据集的个个元素都进行比较。因此，从在这些以及其它类型的计算场景（context）中使用并行处理技术得到显著的益处可能是不可能的或者是行不通的。

发明内容

[0005] 根据一个一般方面，计算机系统可以包括记录在计算机可读介质上的指令。该系统可以包括聚类选择器，其被配置为确定多个样本聚类，以及在多个处理核中的每一个处再现所述多个样本聚类。该系统可以包括样本划分器，其被配置为将存储在数据库中的具有关联属性的多个样本划分为数目相应于所述多个处理核的数目的样本子集，并且还被配置为将所述数目的样本子集中的每一个与所述多个处理核中的对应一个关联。该系统可以包括整合操作器，其被配置为基于所述多个处理核中的每个对应核处的每个样本子集中的

每个样本的关联属性，执行所述每个样本相对于在所述对应处理核处再现的多个样本聚类中的每一个的比较。

[0006] 实施方式可以包括一个或多个下列特征。例如，所述聚类选择器可以被配置为通过图形用户界面 (GUI) 从用户接收的多个样本聚类的数目。所述系统可以包括合并器，其被配置为合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本填充所述样本聚类。样本子集的数目可以等于所述多个处理核的数目，并且每个样本子集可以包括相等数目的样本。所述系统可以包括属性划分器，其被配置为将与每个样本关联的属性划分为属性子集，以供在执行所述比较期间对其进行并行处理。

[0007] 所述比较可以包括在多个处理核中的每一个处执行的、在每个样本子集的每个样本与每个聚类的中心之间的相似性比较。可以使用包括在每个聚类中的样本的平均属性值来确定每个聚类的中心。所述整合操作器可以被配置为基于所述比较将样本从第一聚类重新指派到第二聚类。合并器可以被配置为合并所述比较的比较结果，以及可以被配置为根据需要使用经合并的比较结果来更新每个聚类的每个中心的值。所述合并器可以被配置为基于被重新指派的样本的数目来确定每个聚类内样本的稳定性。

[0008] 根据另一个一般方面，一种计算机实现方法可以包括 :确定存储在数据库中的具有关联属性的多个样本 ;确定多个样本聚类 ;在多个处理核中的每一个处再现所述多个样本聚类。该方法可以包括 :将所述多个样本划分为数目与所述多个处理核的数目对应的样本子集 ;将所述数目的样本子集中的每一个与所述多个处理核中的对应一个关联 ;以及基于在所述多个处理核的每个对应核处的每个样本子集的每个样本的关联属性，执行所述每个样本相对于在对应处理核处再现的多个样本聚类中的每一个的比较。

[0009] 实施方式可以包括一个或多个下列特征。例如，可以合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本填充所述样本聚类。

[0010] 而且，执行所述比较可以包括将与每个样本关联的属性划分为属性子集，以便在执行所述比较期间对其进行并行处理。执行所述比较还可以包括在多个处理核中的每一个处执行每个样本子集中的每个样本与每个聚类的中心之间的相似性比较。

[0011] 根据另一个一般方面，一种计算机程序产品可以被有形地具体实施在计算机可读介质上并且可以包括指令，当被运行时所述指令可以被配置为如下 :确定存储在数据库中的具有关联属性的多个样本 ;确定多个样本聚类 ;以及在多个处理核中的每一个处再现所述多个样本聚类。所述指令当被运行时还可以被配置为 :将所述多个样本划分为数目与所述多个处理核的数目对应的样本子集 ;将所述数目的样本子集中的每一个与所述多个处理核中的对应一个关联 ;以及基于在所述多个处理核的每个对应核处的每个样本子集的每个样本的关联属性，执行所述每个样本相对于在对应处理核处再现的多个样本聚类中的每一个的比较。

[0012] 实施方式可以包括一个或多个下列特征。例如，所述指令当被运行时可以被配置为 :合并所述在多个处理核中的每一个处执行的比较的比较结果，以便由此以所述多个样本填充所述样本聚类。

[0013] 所述指令当被运行时可以被配置为 :将与每个样本关联的属性划分为属性子集，以便在执行所述比较期间对其进行并行处理。所述比较可以包括在多个处理核中的每一个处执行的、每个样本子集中的每个样本与每个聚类的中心之间的相似性比较。

[0014] 所述指令当被运行时可以被配置为：基于所述比较将样本从第一聚类重新指派到第二聚类。所述指令当被运行时可以被配置为：基于所述被重新指派的样本的数目确定每个聚类内的样本的稳定性。

[0015] 在附图以及下面的说明中阐述了一个或多个实施例的细节。其他特征将从说明书和附图以及从权利要求中变得明显。

附图说明

[0016] 图 1 是用于对大规模数据聚类分析执行并行处理的系统的框图。

[0017] 图 2 是示出图 1 的系统的操作的更为详细的例子的框图。

[0018] 图 3 是示出图 1 和图 2 的系统的示范性操作的流程图。

[0019] 图 4 是示出在 k 均值聚类算法的场景中使用图 1- 图 3 的系统和操作的流程图。

[0020] 图 5A 和图 5B 是示出与图 1- 图 4 关联的处理技术的计算本质的框图。

具体实施方式

[0021] 图 1 是在聚类分析期间执行并行处理大数据集的系统 100 的框图。在图 1 的例子中，如图所示，聚类管理器 102 可以被配置为分隔相对较大数据集内的多个样本 104 以定义多个聚类 (cluster)106，以及用样本 104 中适合的样本来填充聚类 106 中的每一个。而且，如这里所述，聚类管理器 102 可以被配置为以利用并行处理技术的方式生成聚类 106 以及用样本 104 中适合的样本来填充聚类 106，该并行处理技术被设计为充分利用多个处理核的计算能力，所述多个处理核如图 1 中核 108、110 和 112 所示。这样，可以以高可配置且高效的方式形成聚类 106，并且通过使用这里描述的并行处理技术，可以以比使用核 108、110、112 中的单个核可能达到的效果明显更快地提供聚类 106。

[0022] 正如上所述，并且如这里所详细描述的那样，聚类管理器 102 可以被配置为使系统 100 的用户能够选择或者另外定义样本 104 以供后续对其聚类。举例来说，用户可以从多个可能的样本数据库中选择样本 104，和 / 或可以选择样本 104 为包括来自一个较大数据库内的数据的子集。在下面的许多例子中，样本 104 被描述为企业的多个客户数据记录，其中每个客户用多个预先定义的属性进行描述。举例来说，企业可以维护全部过去的、现在的以及潜在的客户记录，并且可以将这些客户的身份连同多个相关客户属性（诸如像住址 / 地址、年收入、购买历史、职业之类）或者可能公认与企业的能力相关的许多其它潜在客户属性一起以适当的方式 (setting) 中存储（例如，客户关系管理 (CRM) 系统内），以维持高水平的盈利能力以及客户满意度。

[0023] 当然，样本 104 以及相关属性的这些示范性描述应当理解为并非对范围的限制，并且具体来说，可以理解，在其它示范性实施方式中，样本 104 可以表示许多其它类型的数据以及关联属性。举例来说，样本 104 可以是已经或者可以由企业出售的货物或者服务（例如，诸如可以在库存管理系统内找到的）。在其它例子中，样本 104 可以是企业的资源（例如，与设施和 / 或信息技术资产相关的资源）。更一般地，样本 104 可以因此被理解为表示可以使用大数据集或者与大数据集关联的企业的几乎任何方面，该大数据集包括多个记录以及关联属性。甚至更一般地，可以理解，这些数据集可以存在于（并且因此可以从这里所描述的技术中受益）各种非企业环境中，诸如，像包括学校、政府、军队、慈善或者各种其它

场景的环境中。

[0024] 在所有这些环境中，可能期望将样本104中的各个样本分组到多个聚类106中。举例来说，在客户关系管理的场景中，样本104可以包括多个客户数据记录以及关联属性，如上面所述。因此，对系统100的用户来说，可能期望将样本104分组到聚类106中的对应聚类中，其中可以根据用户可能感兴趣的、客户关系管理的某个方面来定义聚类106。

[0025] 举例来说，在一个特定情景中，根据近期将进行大量购买的可能性等级将客户划分到各个聚类106。举例来说，特定准则可以与定义这样的可能性以及对这样的可能性进行评级(rating)相关联。举例来说，共享诸如高年收入、最近的大量购买以及被认为与将来的购买可能性相关的其它基于属性的因素之类的特征的客户可以被分组到第一聚类106(1)内，该第一聚类106(1)将因此包括被图示为样本104(1)的样本子集。同时，反之，具有非常不同属性值（例如，较低年收入、无最近购买历史以及被认为与近期购买的较低可能性相关的其它基于属性的因素）的客户可以被聚类到另外的聚类内，在图1的例子中被示为聚类M106(M)，其由此包括被示为样本104(M)的样本子集。

[0026] 当然，尽管图1的简化例子仅明确地示出两个聚类，但是可以理解，可以形成期望数目的聚类。举例来说，在刚刚提供的图示例子中，可以形成总数目为M的聚类，其中，根据所预测的这里包括的客户将来购买的可能性的等级来定义聚类。当然，可以使用图1的系统100形成许多其它类型的聚类，并且不需要根据刚刚提到的线型分布类型来形成。

[0027] 更一般地来说，可以理解，与聚类106的定义和形成相关的各种概念本身在本领域是公知的，并且因此除了可能对于理解图1的系统100的操作是必要的或者有帮助的内容之外，不在这里更为详细地描述。相反，在这里一般来说针对图1的系统100中出于并行化与针对样本104形成、定义、填充以及以其它方式管理聚类106相关联的计算过程的目的而实现的特定示范性技术，并且更具体地说，针对在以针对使用可用核108、110、112来充分利用各种并行处理技术的方式执行聚类106的这样的管理中聚类管理器102的各种特征和功能，来提供图1的系统100的更进一步的描述。

[0028] 对于此点，可以理解，核108、110、112意欲表示几乎任何已知或者将来的并行处理平台或者场景。举例来说，核108、110、112中的每一个都可以表示独立的服务器或者包括一个或多个处理器的其它计算装置。在另外的或者替换实施方式中，核108、110、112中的每一个可以包括在单个服务器或者其它计算装置内。因此，核108、110、112中的每一个都可以理解为表示或者包括任何多计算平台，在该多计算平台中，多个处理器、中央处理单元(CPU)或者其它处理资源是可用的，包括网络/设备群(cluster)。举例来说，并行处理可以利用现有SMP/CMP（对称多处理/芯片级多处理）服务器。因此，在本说明书中，应当理解，术语“核”表示具有处理能力的单元。

[0029] 因此，可以理解，系统100可以被配置为在刚刚提到的各种并行处理平台和/或未特别提到的其它并行处理平台或者它们的组合中的任意一种平台的场景中操作，并且被配置为充分利用刚刚提到的各种并行处理平台和/或未特别提到的其它并行处理平台或者它们的组合中的任意一种平台的特征和功能。因此，如前所述，聚类管理器102可以被配置为使得能够使用多个处理核——在图1的例子中以核108、110、112表示——来并行化相对较大数目的样本（以及它们各自的属性）到相对较小数目的聚类106的整合(joint)操作。

[0030] 更具体来说，如图所示，样本104可以被划分成多个样本子集114、116、118。如图

所示，样本子集 114、116、118 中的每一个都可以分布到核 108、110、112 中的相应一个。图 1 的例子示出与三个可用核 108、110、112 相应的三个样本子集 114、116、118。当然，更一般地，样本 104 可以被分成任意合适或者期望数目的子集，例如，其数目与可用核的数目对应。在这里所描述的示范性实施方式中，样本 104 可以被划分成多个样本子集，以使得每一个样本子集都包含近似相等数目的样本 104。然而，在其它示范性实施方式中，可以存在样本子集具有不同大小的情况。举例来说，被指派到相对高速处理核的样本子集与相对低速处理核所关联的或者被指派到相对低速处理核的样本子集相比可以被提供有更大数目的样本。

[0031] 与将样本 104 划分为各种样本子集 114、116、118。相反，聚类 106 可以被整体再现，以供可用核 108、110、112 中的每一个处理。具体地说，如图所示，聚类 106 可以在可用核 108、110、112 中的每一个处被整体再现为聚类再现 120。因此，如下面提供的更为详细的例子中所述，可以针对相应的样本子集 114、116、118 并且结合聚类再现 120 运行在各个核 108、110、112 处的（或者与各个核 108、110、112 关联的）并行化整合操作器 (operator)122、124、126。具体地说，举例来说，可以使用并行化整合操作器 122，将示范性子集 114 中的全部样本分别与全部聚类再现 120 相比较或者以其它方式相对于全部聚类再现 120 进行考虑。类似地，可以使用并行化整合操作器 124，将样本子集 116 中的全部样本分别与全部聚类再现 120 进行比较。类似地，类似注释适用于相对于聚类再现 120 的样本子集 118 以及并行化整合操作器 126。

[0032] 通过这样的方式，可以发现，总体上，可以以利用高等级的非常高效的并行化的方式来将全部样本 104 分别与聚类 106 中的每一个进行比较。结果，系统 100 的操作者或者其它用户可以快速得到期望的结果。举例来说，如下面所提供的更为详细的例子中那样，系统 100 的操作者可能接下来能够以期望方式快速定义并且形成聚类 106，以便由此之后以其它传统方式来使用聚类 106。

[0033] 在图 1 的特定例子中，聚类管理器 102 示出为包括属性选择器 128。如上所述，样本 104 可以分别与针对正在讨论的样本定义的或者以其它方式与正在讨论的样本相关的一个或多个属性关联。众所周知，并且如这里详细描述的那样，这些属性的数目和类型以及其可能值或者值的范围，可以依赖于系统 100 的使用场景而变化很大。在系统 100 的一个给定实施方式中，可以出现 :可用属性的仅一个子集或者一部分可能被期望用于相应计算中。另外，或者可替换地，可以出现 :某些属性应当被重视或者被区别对待（例如，视为较重要或较不重要）。因此，属性选择器 128 可以被配置为使系统 100 的用户能够以期望方式选择可用样本属性和 / 或描述可用样本属性的特征。举例来说，尽管在图 1 的例子中未具体示出，但是可以为系统 100 的用户提供合适的图形用户界面，其中，如应当认识到的那样，这样的图形用户界面的形式和格式将依赖于系统 100 的特定使用场景。

[0034] 聚类选择器 130 可以被配置为使系统 100 的用户能够定义聚类 106 的期望本质或者以其它方式描述聚类 106 的期望本质的特性。举例来说，例如，依赖于正在使用的相关聚类算法或者其它因素，聚类选择器 130 可以使系统 100 的用户能够定义将要计算的聚类 106 的数目。另外，或者可替换地，聚类选择器 130 可以使系统 100 的用户能够进一步描述聚类 106 的特性。举例来说，用户可以定义聚类 106 的最大大小，或者聚类 106 相互之间的相对大小，或者聚类 106 的任意其它特征或者特性。与属性选择器 128 一样，可以由聚类选择器

130 提供合适的图形用户界面，以供系统 100 的用户在执行刚刚提到的聚类 106 的并行化时使用。

[0035] 举例来说，如图 1 中所示，聚类 106 可以包括 M 个聚类，在图 1 中示出为第一聚类 106(1) 以及第 M 聚类 106(M)。然后，在该例子中，可以认为聚类选择器 130 使系统 100 的用户能够定义参数 M，以使得如所描述的那样，可以将全部 M 个聚类复制为在可用核 108、110、112 处的聚类再现 120。

[0036] 样本划分器 132 可以被配置为执行将样本 104 划分为样本子集 114、116、118。如上所述，样本划分器 132 可以通过将样本 104 划分为一定数目的样本子集——该数目等于可用核的任意数目——来执行，其中样本子集中的每一个可以彼此大小近似相等。然而，如上所述，样本划分器 132 还可以与图形用户界面或者可以使系统 100 的用户能够以更加定制化的方式配置样本子集的其它输入技术相关联。举例来说，样本子集 114、116、118 可以大小不同。在其它例子中，可以基于样本子集的指定参数属性来定义及划分样本子集，而非通过样本 104 的简单分割来定义和划分。

[0037] 聚类再现器 134 可以被配置为在核 108、110、112 中的每一个处将聚类 106 再现为聚类再现 120。相关再现技术本身是公知的，因此这里不再进一步详细描述。

[0038] 合并器 136 可以被配置为整合、同步、聚合 (aggregate) 或者以其它方式合并来自核 108、110、112 的处理结果。举例来说，合并器 136 可以被配置为合并作为聚类管理器 102 的较大操作集的部分的中间处理结果以及合并最终结果集。在并行处理的场景中这样的合并操作本身是公知的，并且可以包括，举例来说，以来自每个相关的处理核的结果填充公共数据库或者其它存储器，和 / 或执行每个相关处理核的关联处理 (具体来说，执行可以仅在中央处理器处执行的数据的处理，这样的处理的例子是公知的和 / 或将在下面详细提供)。

[0039] 举例来说，在图 1 的例子中，并行化整合操作器 138 可以被配置为执行上面针对操作器 122、124、126 所述的多种类型的操作。一般来说，例如，如前所述，这样的整合操作可以包括将样本 104(或者其子集) 与聚类 106 中的每一个单独比较。下面的例子讨论了相似性比较以及相关处理，作为这样的整合操作的一个例子。然而，将认识到，也可以由聚类管理器 102 执行其它类型的整合操作以及相关处理。

[0040] 虽然如此，但是在图 1 的特定例子中，为了说明起见，并行化整合操作器 138 可以包括比较器 140。比较器 140 可以被配置为例如，用于将样本 104(或者其子集) 分别地与聚类 106 中的每一个进行比较。

[0041] 基于这些比较的结果，样本指派器或者样本重新指派器 142 可以被配置为将样本中的每一个与聚类 106 中给定的一个关联。随后，中心选择器 144 可以被配置为分析如此形成的聚类以及所包含的样本，以便由此确定新的或者更新的中心或者与每个聚类关联的其它度量。随后，比较器 140 可以通过重复将单个样本与新定义的聚类中的每一个进行比较来以迭代方式继续，以使得样本重新指派器 142 可以因此根据需要针对当前的聚类定义重新指派样本。该迭代过程可以继续直到例如聚类到达所定义的稳定程度 (例如，根据在新迭代中被重新指派的样本数目或者百分比小于一定的阈值来定义的，和 / 或基于这些迭代到达某一阈值的预定义次数定义的)。

[0042] 如上所述，在图 1 的系统 100 的特定示范性实施方式中，比较器 140 可以使用相似性比较器 140a 来执行所述比较，所述相似性比较器 140 被配置为比较每个单个样本 (例

如，样本子集的每个单个样本）与聚类 106 中的每一个的相似程度。这些相似性测量本身是公知的，并且对于本领域技术人员来说应当是清楚的，因此除了对于理解图 1 的系统 100 的操作是必要的或者有帮助的内容之外不对其详细描述。

[0043] 然而，一般说来，会认识到，可以针对样本 104 的各种属性（或者其子集）来执行这些相似性测量。例如，如上所述，属性选择器 128 可以被配置为使系统 100 的用户能够定义样本 104 的属性（和 / 或其特征或值），以使得相似性比较器 140A 可以由此被配置为基于所定义的属性（或者其特征或值）来执行每个单个样本与聚类 106 中的每一个之间的相似性测量。

[0044] 在许多情况下，如下面详细描述的那样（例如，针对图 2），可以出现 :样本 104 中的每一个与相对较大数目的属性关联，这些属性被定义或者指定用于与聚类 106 中的每一个的相似性比较中。因此，可能期望在较大的并行处理相对于聚类 106 的样本本身的场景内，但是针对正在讨论的属性，使用并行处理。换句话说，例如，以与样本划分器 132 可以被配置为将样本 104 划分为样本子集 114、116、118 几乎一样的方式，属性划分器 140b 可以被配置为划分或者以其它方式分割或者指定所选择的属性。因此，会认识到，通过已经被指定用于相似性比较的样本的属性子集的并行处理的使用，可以加速相似性比较器 140a 的相似性比较。正如所述，下面将例如参考图 2 提供对于在单个样本和聚类 106 之间的相似性比较的场景中使用样本属性的这种并行处理的示范性技术。

[0045] 在图 1 的例子中，聚类管理器 102、样本 104 和聚类 106 被示出为通过使用至少一个计算装置 146——其可以包括或者合并至少一个处理器 146a 以及计算机可读存储介质 146b——来实现。在此上下文中，一般会认识到，图 1 示出系统 100 的特征和功能可以通过使用利用计算机可读存储介质 146b 存储并且由至少一个处理器 146a 运行的指令或者其它代码来识别。

[0046] 具体地说，例如，图 1 示出单个计算装置 146 ;然而，会认识到，至少一个计算装置 146 可以表示多个计算装置，其中每个计算装置都可以使用也许如这里所述并行运行的两个或更多处理器。举例来说，在一些示范性实施方式中，至少一个处理器 146a 可以由一个或多个核 108、110、112 中的一个或多个表示，而在其它示范性实施方式中，至少一个计算装置 146 可以表示与服务器或者容纳核 108、110、112 的其它计算机通信的中央计算机。

[0047] 因此，尽管聚类管理器 102 及其组件被示出为包括在至少一个计算装置 146 中或者结合至少一个计算装置 146 运行，但是会认识到，可以使用多个不同的计算装置——例如，一个或多个核 108、110、112 来运行聚类管理器 102 中的部分或者全部。也就是说，举例来说，在一些实施方式中，聚类管理器 102 的部分可以在第一计算装置以及关联的处理器上运行，而聚类管理器 102 的其它部分可以使用一个或多个单独的计算设备 / 处理器来运行。

[0048] 举例来说，计算装置 146 可以表示中央计算装置，该中央计算装置由系统 100 的用户访问并且运行聚类管理器 102 的组件，诸如像属性选择器 128、聚类选择器 110、样本划分器 132、聚类再现器 134 和合并器 136。同时，如上所述并且如可以从图 1 的例示会认识到的那样，并行化整合操作器 138 可以被实例化为相应核 108、110、112 中的每一个处的并行化整合操作器 122、124、126 和 / 或作为相应核 108、110、112 中的每一个处的并行化整合操作器 122、124、126 以其它方式运行。在其它实施例中，会认识到，至少一个计算装置 146 可

以表示在其中执行这里所描述的并行处理的核中的一个。

[0049] 系统 100 的架构上的这些变化或者其它结构对本领域技术人员来说是清楚的，并且因此在这里没有更详细地进行描述。而且，许多其它变化也是可能的并且应当是清楚的。举例来说，可以使用或许在通过网络彼此通信的不同计算装置上运行的两个或更多组件来运行聚类管理器 102 的任意单个组件。反之，可以使用单个组件运行聚类管理器 102 的两个或更多组件。通过举例在这里描述了许多其它实施方式，或者许多其它实施方式本来也是明显的。

[0050] 图 2 是图 1 的系统 100 的更详细的实施方式的框图。具体地说，如上所述（例如，针对属性划分器 140b），图 2 示出了这样的示范性实施方式：其中，聚类管理器 102 以另外并行处理样本子集的属性子集来补充样本 104 的子集的并行处理。

[0051] 具体地说，在图 1 的例子中，核 108、110、112 被示出并且大致上描述为表示可以被配置为彼此并行处理的几乎任意类型、数量的多个核或者可以被配置为彼此并行处理的多个核的几乎任意组合。在更具体的图 2 的例子中，服务器 202、204、206 被示出为各自包括至少两个处理核。换句话说，如图所示，服务器 202、204、206 表示多核服务器。在示出的该具体例子中，如图所示，服务器 202 包括核 208、210，而服务器 204 包括核 212、214，且服务器 206 包括核 216、218。

[0052] 因此，在操作中，聚类管理器 102 可以被配置为将样本 104 划分为样本子集 220、222 和 224，然后它们可以被分别指派给服务器 202、204、206，如图所示。举例来说，样本划分器 132 可以被配置为以上面针对图 1 所描述的方式将样本 104 划分为样本子集 220、222、224。对于此点，会认识到，上面针对图 1 描述的聚类管理器 102 的许多特征和功能可以在图 2 的场景中类似地执行。举例来说，属性选择器 128 可以接收对与将用于聚类生成过程中的样本 104 中的每一个相关联的各种属性的选择，而聚类选择器 130 和聚类再现器 134 可以也被配置为执行它们的相应功能（例如，尽管在图 2 的例子中未具体示出，但是选择聚类 106 的数目和 / 或特征，并且复制全部聚类 106 以便将其与服务器 202、204、206 关联）。类似注释应用于针对在图 2 的场景中在执行对应功能时使用合并器 136 和并行化整合操作器 138 时合并器 136 和并行化整合操作器 138 的操作。因此，将认识到，聚类管理器 102 的在图 2 中的功能与其在图 1 中的功能是相似的，因此在这里不再详细重复了。

[0053] 至于样本子集 220（要理解类似的注释适用于样本子集 222 和样本子集 224），可以存在：样本子集 220 的每个样本被相对于被复制以与服务器 202 相关联并且由服务器 202 使用的聚类 106 中的每一个都进行比较（例如，样本子集 220 的每个样本将具有一个经判定的相似程度）。换句话说，如这里详细描述的那样，可以将样本子集 220 中的第一样本相对于聚类 106 中的每一个都进行比较，以得到其与聚类 106 中的每一个的相似性。随后，样本子集 220 中的第二样本可以类似地与聚类 106 中的每一个都进行比较，直到样本子集 220 的全部样本都被如此进行了比较为止。在图 2 的示范性实施方式中，假定样本 104 各自与相对较大数目的关联属性相关联，并且假定已经选择了相对较大数目的这些可用属性用于刚刚提到的相似性比较。

[0054] 于是，在图 2 的例子中，这样的一个相对较大的属性池可以被划分为相应的属性子集，例如，与服务器 202 的核 208 相关联的属性子集 226 以及与服务器 202 的核 210 相关联的属性子集 228。具体地说，例如，属性划分器 140b 可以被配置为将待使用的属性集划分

为期望数目的子集，例如，与位于用于处理与正在讨论的属性关联的相应样本的相应多核服务器处的可用核的数目相应的数目。

[0055] 随后，会认识到，基于关联属性的比较可以彼此并行进行，以使得可以以更快且更及时的方式完成整体相似性比较。而且，如上所述并且如图 2 的例子所示，类似注释适用于服务器 204、206。具体来说，如图所示，与样本子集 222 关联的属性可以被划分为属性子集 230 和 232，用于使用服务器 204 的各个核 212、214 对其的并行处理。类似注释适用于分别与服务器 206 的核 216、218 关联的属性子集 234、236。

[0056] 图 3 是示出图 1 的系统 100 和图 2 的系统 200 的示范性操作的流程图 300。在图 3 的例子中，操作 302-312 被示为单独的、顺序的操作。然而，将认识到，在其它示范性实施方式中，可以以部分或者完全重叠或并行的方式实现两个或更多操作 302、312。而且，操作 302-312 可以以不同于图示的次序执行，包括例如，以嵌套的、循环的或者迭代的方式。另外，还可以包括未在图 3 的例子中具体示出的附加操作或者替换操作，和 / 或可以省去一个或多个操作或者其部分。

[0057] 在图 3 的例子中，可以确定存储在数据库中的具有关联属性的多个样本 (302)。举例来说，属性选择器 128 可以被配置为接收对与样本 104 相关联地存储的指定属性的选择。

[0058] 可以确定多个样本聚类 (304)。举例来说，聚类选择器 110 可以被配置为标识、特征化、参数化和 / 或以其它方式标识或者确定聚类 106。

[0059] 多个样本聚类可以在多个处理核中的每一个处被再现 (306)。举例来说，聚类再现器 134 可以被配置为在核 108、110、112 中的每一个处 (或者，在图 2 的例子中，在服务器 202、204、206 中的每一个处) 再现相对于相关样本 104 定义或者标识的全部聚类 106。

[0060] 多个样本可以被划分为数目与多个处理核的数目相应的样本子集 (308)。举例来说，样本划分器 132 可以被配置为将样本 104 划分为样本子集 114、116、118(或者，在图 2 的例子中，划分为样本子集 220、222、224)。

[0061] 该数目的样本子集中的每一个都可以与多个处理核中的相应一个处理核相关联 (310)。举例来说，样本划分器 132 可以被配置为复制或者以其它方式提供样本 104 的样本子集 (例如，图 1 的样本子集 114、116、118，或者图 2 的样本子集 220、222、224)。举例来说，样本划分器 132 可以被配置为将样本子集中的每一个复制到与所述多个处理核中的相应一个处理核 (例如，图 1 的核 108、110、112，或者图 2 的服务器 202、204、206) 相关联的存储器，所述存储器例如可以由相应处理核读取。

[0062] 基于在多个处理核中的每个相应核处的每个样本子集中的每个样本的关联属性，执行所述每个样本相对于在相应处理核处再现的多个样本聚类中的每个样本聚类的比较 (312)。举例来说，并行化整合操作器 138(例如，或者其实例 122、124、126) 可以被配置为执行这样的比较。举例来说，并行化整合操作器 122 可以被配置为基于样本子集 114 的每个样本的属性，将样本子集 114 的每个样本与核 108 所关联的再现聚类 120 中的每一个相比较。当然，类似注释适用于并行化整合操作器 124、126 以及各个样本子集 116、118。

[0063] 在这里描述的具体例子中，所述比较可以包括子集样本中的每一个与在处理核中的每一个处再现的多个聚类中的每一个之间的相似性比较。举例来说，与特定样本子集中的特定样本相关联的属性可以被用于执行与多个再现聚类中的每一个的这种相似性比较，如这里详细描述的那样。具体地说，例如，如上相对于图 2 所述，将用于这种相似性比较的、

相对较大数目的这些样本属性可以被进一步划分为样本属性子集，以供随后在这种相似性比较的场景中的另外的并行化处理中使用。

[0064] 会认识到，可以如上针对图 1 和图 2 所述的那样来执行这样的并行处理的附加方面或者替换方面，或者这样的并行处理的附加方面或者替换方面可以以清楚的其它方式运行。举例来说，在这样的并行处理完成时或者在其中间步骤完成时，可以进行适当的合并操作，以便组合或者以其它方式合并该并行处理（或者其中间操作的）的结果。举例来说，合并器 136 可以被配置为组合与给定子集样本相关联的属性子集 226、228 的并行处理，以便完成该子集样本与给定样本聚类的相似性比较。类似地，合并器 136 可以被配置为合并在图 1 的场景中与样本子集 114、116、118 中的每一个关联的比较结果，或者合并在图 2 的场景中的样本子集 220、222、224 的比较结果。

[0065] 当然，可以根据给定示范性实施方式的特定场景，包括更多附加或者替换操作。下面图 4、图 5A 和图 5B 将提供 k 均值算法的的例子。

[0066] 具体地说，图 4 和图 5A、图 5B 提供了在实现 k 均值算法 (k-means algorithm) 的场景中图 1- 图 3 的系统和操作的实施方式的例子。如上所述，并且众所周知，k 均值算法是一种分析方法 :被设计为将“N”个样本（例如，样本 104）分割为“k”个聚类（例如，聚类 106)，以使得“N”个样本中的每一个都属于具有最近均值的第 k 个聚类。因为 k 均值算法本身是具有许多已知实现领域的公知算法，除了可能对理解这里描述的系统和操作的特征和功能是必要的或者有帮助的内容之外，k 均值算法本身和许多实现领域的例子这里都不详细提供。

[0067] 虽然如此，但是为了说明和示范起见，图 4 示出包含 k 均值算法的完整运行的操作 402-412。在图 4 和图 5A、图 5B 的场景中并且如上所述，参考具体例子或者例子集合，其中“N”个样本 104 可以包括提供能源和相关服务的公用事业公司的客户的大量（例如，3 百万）客户简档。在这样的例子中，客户简档中的每一个都可以具有已定义数目的属性（例如，300 个属性）。举例来说，如在这样的场景中公知的那样，这样的属性可以包括，例如，家庭收入或者与对应客户简档关联的财务特性、能源使用历史、住所特性（例如，关联的客户是住套房 (house) 还是公寓 (apartment)）或者可能与客户关联的以及可能与按时且有利地向其递送公共设施相关的任意其它特性或者属性。

[0068] 而且，还是如上所述，在图 1 和图 2 的系统中，以及一般而言在 k 均值算法中，对于用户来说可以选择以及以其它方式描述将要形成的一定数目的 k 聚类的特性。为了所提供的例子起见，假定 k 均值算法的示范性实施方式将使用与 3 百万个客户简档中的每一个关联的 300 个属性的对应值，将该 3 百万客户简档聚类为 100 个聚类。

[0069] 因此，在图 4 的例子中，操作可以一开始以随机选择 k 个聚类中心开始 (402)。也就是说，正如所述的那样，用户可能想要数目为 k 的聚类，其等于例如 100 个聚类。在这里所描述的 k 均值算法的场景中，每个这样聚类可以相对于其中心来定义，“N”个样本中的每一个都与该中心来比较，以得到其相似度。也就是说，如 k 均值算法中所已知的，并且如下所述，应当形成理想的或者期望的 k ＝ 100 个聚类的最终集合，以使得全部 N ＝ 3 百万个样本被指派给这样的聚类 :其中心在所有中心中与正在讨论的样本最相似。

[0070] 因此，在该场景中，并且如一般在 k 均值算法实施方式的场景中公知的那样，术语“中心”或者“聚类中心”应当理解为指代代表性的属性或者定义的属性，或者属性的集合或

者组合，这些属性可以用于描述聚类的特性，以将聚类相互区分。举例来说，在一个简单例子中，3 百万客户简档的 300 个属性中的一个可以包括对应客户的地理位置（例如，使用位置的邮政编码或者经 / 纬度表示）。在这样的例子中，可以存在 :这些位置属性被指定为 100 个聚类的定义的基础，以使得 3 百万个地理位置中的 100 个可以用于定义该聚类（也即，用于定义聚类中心）。在这样的例子中，使用如下所述的方法，3 百万客户中的每一个都将与对应聚类和聚类中心相关联，该聚类中心在所有聚类中相对于正在讨论的特定客户来说具有最接近的地理位置。通过这样的方式，全部 3 百万客户可以被指派给对应的、地理上定义的聚类。

[0071] 当然，在其它更为详细的例子中，可以相对于属性集合或者组合定义聚类中心。举例来说，可以归一化属性的值，例如，通过给 300 个属性中的每一个赋予 0 到 1 之间的属性值，以使得 3 百万客户中的每一个都具有针对 300 个属性中的每一个的 0 到 1 之间的对应属性值。在这样的例子中，可以选择期望的属性集合或者组合，并且可以使用所选择的属性的归一化值来计算 100 个聚类的中心的对应集合。再有，这样的技术本身是公知的，因此除了对于理解图 4 和图 5A、5B 的例子是必要的或者有帮助的内容之外，在这里不更为详细地进行描述。

[0072] 如上所述，在图 4 的例子中，操作 402 因此表示对所选择的 k 个聚类中心的初始的、最佳猜测的 (best guess) 或者随机的选择，仅仅作为开始图 4 的算法的迭代的手段。举例来说，在上面描述的简化的基于地理的聚类中，操作 402 可以包括随机从 3 百万客户中选择 100 个，并且使用对应的 100 个地理位置作为聚类中心的初始集合。在上面所述的其它例子中，操作 402 可以包括随机选择 100 个客户，然后分析相关的、关联归一化属性值以计算对应的 100 个中心。

[0073] 一旦正如上所述已经选择了聚类中心的初始集合，就可以计算 N ＝ 3 百万个样本中的每一个与 k ＝ 100 个聚类中心之间的相似性 (404)。举例来说，如关于上面图 1- 图 3 的描述应当清楚的那样，操作 404 的计算一般应当需要 3 百万乘 300 再乘 100 次计算（假定将使用全部 300 个属性）。虽然如此，但是使用上面相对于图 1- 图 3 描述的特征和功能，可以以有助于快速且及时执行这样的计算方式来并行化这样的计算。

[0074] 具体地说，如上相对于图 1 所述，可以将 N ＝ 3 百万个样本划分为数目为 S 的样本子集，该数目 S 等于可用服务器或者处理核的数目。然后，如上所述，可以针对 S 个服务器 / 核中的每一个再现全部 k ＝ 100 个聚类中心。然后，如上所述，可以针对 S 个服务器 / 核中的每一个再现全部 k ＝ 100 个聚类中心。如可以看到的那样，这样的再现可以是实际且直接明了的，假定期望的聚类中心的数目 k 比待分组的样本的数目 N 小，因此 k 个聚类中心的复制不会产生相对可观的开销。

[0075] 尽管没有相对于图 1 具体讨论，但是可以存在 :可以使用描述相对于对应聚类中心中的每一个的相关属性（或者其集合或者组合）的特性的相似性表格，进行将在如此划分的样本子集中的每一个样本子集中的每个样本与对应聚类中心中的每一个之间执行的相似性测量，从而可以确定它们之间的相对相似度。这样的相似性表格本身及其使用是公知的。然而，在图 1- 图 4 的例子中，并且如下面将相对于图 5A 和图 5B 更为详细地描述的那样，这样的相似性表格可以被类似地分割为 S 个部分，并且与相同服务器或者处理核相关联地存储。换句话说，如这里所述，N ＝ 3 百万个样本可以被划分为数目为 S 的样本子集，S

＝可用服务器 / 核，并且对应的相似性表格可以类似地被划分为数目为 S 的相似性表格子集，S ＝可用服务器 / 核。

[0076] 图 5A 和图 5B 用图示出将结合操作 404 执行的计算的本质，以及如这里所述这些计算可以被并行化的方式。具体地说，如图所示，N ＝ 3 百万个样本被示为样本 502、504...506，并且被示为分别与对应聚类中心 508、510...512 相比较。因此，图 5A 概念性地示出上面所述的将针对 N ＝ 3 百万个样本执行相对于 k ＝ 100 个聚类的所述类型的相似性计算或者其它整合操作的资源密集的本质 (resource-intensive nature)。

[0077] 同时，图 5B 概念性地示出可以使用这里所描述的技术并行化操作 404 的资源密集的整合操作的方式。如上所述，参照图 5B 描述的技术可以例如使用具有多个服务器的群和 / 或使用包括多个核的单个服务器来实现。因此，在所提供的示例中，可以认识到，对服务器 / 核或多个服务器 / 多个核的引用应当被理解为指代这些选择中的其中之一或者它们二者，以供实现。具体地说，如图所示，由服务器 514、516...51 8 示出数目为 S 或 C 的可用服务器 / 核。如图所示并且所描述的那样，N ＝ 3 百万个样本可以被划分为对应于并且被指派给可用服务器 / 核 514、516...518 中的每一个的样本子集。具体地说，如图所示，样本 (1 ～ N/S) 或 (1 ～ N/C)(在图 5B 中示出为样本 502...520) 可以被指派给第一服务器 / 核 514。类似地，对应数目的样本以及第二样本子集 (在图 5B 中以样本 522 表示) 将被指派给第二服务器 / 核 516，等等，直到最后的服务器 / 核 518，最后的服务器 / 核 518 将接收最后的样本子集，其包括第 N(1-(1/S)) 或 N(1-(1/C)) 样本 524 到最后的样本 506。从以上内容，可以认识到，在下面的内容中，为了简洁并且仅仅为了注解，一般使用标号“S”自己，但是无论如何标号“S”要被理解为指代多个服务器中的一个或者在单个服务器上运行的多个核中的一个。

[0078] 如图 5B 所示，服务器 / 核 514、516...51 8 中的每一个也将接收全部 k 个聚类中心 508...512，或者已经向服务器 / 核 514、516...518 中的每一个指派了全部 k 个聚类中心 508...512。而且，如图所示以及如上所述，也可以将针对给定样本子集的样本的任意相似性表格项和 / 或样本 - 聚类映射复制或者存储到服务器 / 核 514、516...518 中的对应一个中。举例来说，如图所示，接收样本 502...520 的服务器 / 核 514 将类似地接收针对样本子集 (也即，N 个样本中的第一到第 N/S 样本) 中的相应样本的相应相似性表格项以及样本 - 聚类映射。同时，并且类似地，服务器 / 核 518 将接收针对对应于相关样本子集 (也即，第 N(1-(1/S)... 第 N 样本) 的样本的相似性表格项和样本 - 聚类映射。尽管未具体示出，但是可以认识到，可以针对样本子集中的每一个和关联服务器 / 核执行相关相似性表格项和样本 - 聚类映射的类似的关联和指派。

[0079] 因此，可以认识到，可以在 S 个服务器 / 核上并行化操作 404，否则操作 404 会昂贵，并且此后彼此独立地执行。具体地说，可以在相关样本子集中的每个样本与聚类中心中的每一个之间都进行相似性测量和比较。在该上下文中，如上所述，这样的相似性比较可以包括或者基于与在这里描述的例子中为定义相似性而选择的 M ＝ 300 个属性相关联的计算。

[0080] 举例来说，相对于图 5B，可以认识到，样本 502 可以与 300 个属性的对应值关联，并且可以针对这 300 个属性类似地定义中心 508...512。此后，可以一开始将第一样本 502 与第一中心 508 相比较，以便确定与其的相对相似性。

[0081] 举例来说，可以使用已知的欧式距离计算这样的相似性，如下面的公式 1 的例子所示：

[0082] $$d = \sqrt{\sum_1^M (x_i - x_i')^2}$$ 公式 1

[0083] 举例来说，对于表示为“样本 A”的第一样本 502 以及对于表示为“样本 B”的第一中心 508 来说，可以根据公式 2 来计算公式 1 的欧式距离 d：

[0084] 样本 A = [x_1, x_2, . . . , x_M]

[0085] 样本 B = [x_1', x_2', . . . , x_M']

[0086] $$d(A,B) = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \ldots + (x_M - x_M')^2}$$ 公式 2

[0087] 其中，如所示，对于样本 A、B 中的每一个来说 M 个属性被示为 x_1. . . x_M。

[0088] 因此，在这样的例子中，可以依次计算该欧式距离，作为对于第一样本 502 与聚类 508. . . 512 中的每一个的相对相似性的相似性量度 (measurement)，此后对于指派给第一服务器 / 核 514 的样本子集中的每个剩余样本，直到并且包括相关样本子集的最后一个这样的样本，也即，第 N/S 样本 520，进行这样的计算。类似注释将适用于在剩余服务器 / 核 516. . . 518 处执行的计算。

[0089] 而且，如上所述，例如，相对于属性划分器 140b，可以以类似于针对 N = 3 百万个样本整体的相似性计算的并行化的方式，进一步并行化刚刚提到的相似性计算。具体地说，如上所述，可以存在：服务器 / 核 514、516. . . 518 可以表示图 2 的多核服务器 202、204、206。然后，300 个属性可以被划分为相应的属性子集，用于在与每个这样的多核服务器相关联的多个核处对其的并行化处理。

[0090] 在一个具体例子中，参考图 2，可以存在：样本子集 220 将包括 3 百万客户简档中的一百万个客户简档，而核 208 将与包括 300 个属性中的 150 个属性的属性子集 226 相关联，第二核 210 将包括属性子集 228 中的其余 150 个属性。通过这样的方式，还可以在一定数目的可用属性上进一步并行化公式 1-2 的相似性计算，以便更进一步促进快速和及时的相似性计算。

[0091] 一旦已经计算出每个样本子集中的每一样本与每个聚类中心之间的相似性量度，就可以基于该相似性量度在 k 个聚类中心内以及在 k 个聚类中心之间重新指派样本 (406)。举例来说，相对于图 5B 的第一服务器 / 核 514，如上所述，可以计算样本 502. . . 520 中的每一个与中心 508. . . 512 中的每一个之间的相似性量度。

[0092] 因此，举例来说，第一样本 502 将具有最接近聚类中心 508. . . 512 中之一的相似性量度，并且因此将被指派给该聚类中心。可以针对指派给服务器 / 核 514 的样本子集中的其余样本，直到并且包括 N/S 样本 520，进行类似指派，从而，在操作 406 的结束时，全部样本 502. . . 520 都已经指派给核中心 508. . . 512 中的一个。当然，类似注释适用于将服务器 / 核 516 处的样本子集中的样本重新指派成服务器 / 核 518 处的样本子集 524. . . 506 的样本。因此，可以明确地看到，在图 4 和图 5B 的例子中，以并行化的方式将全部 N = 3 百万个样本与 k = 100 个聚类中心中的每一个都进行了比较，因此这是以高效方式进行计算。

[0093] 在操作 406 之后，可以做出稳定性确定 (408)。举例来说，在一个简单例子中，可以简单地基于图 4 的流程图 400 的迭代的次数来确定 k = 100 个聚类中心的聚类结果的稳定性。也就是说，例如，在假定其稳定性之后可以定义迭代的最大次数。另外，或者可替换

地，可以基于其它衡量标准 (metric) 来判定稳定性。举例来说，稳定性可以被确定为在操作 406 期间被重新指派给 k ＝ 100 个聚类中心的样本的数目。也就是说，如在 k 均值算法的传统实施方式中已知的那样，一旦在流程图 400 的一次迭代的操作 406 期间在聚类之间重新指派了最小数目的样本，就可以进一步假定迭代将对 k ＝ 100 个聚类的样本指派产生微小的影响，和 / 或事实上 N ＝ 3 百万个样本中的大多数或者全部被指派给最相似的聚类中心。

[0094] 在至少部分地基于被重新指派的样本的数目来判定稳定性的情况下，可能需要例如使用合并器 136 合并来自可用服务 / 核的数据。也就是说，参考作为示例的图 1 的例子，可能出现 :单个样本被重新指派到在核 108、110、112 中的每一个处的新的中心。因此，合并器 136 可能合并来自核 108、110、112 的数据，以确定已发生总共三次重新指派。

[0095] 因此，如果确定了期望的稳定程度 (408)，那么流程图 400 的算法就可以完成 (410)。否则 (408)，可能需要或者期望计算正在讨论的 k ＝ 100 个聚类中的更新的聚类中心 (412)。

[0096] 也就是说，如上所述，例如，相对于操作 402，流程图 400 的算法可以一开始以随机选择 k 个聚类中心开始。举例来说，在上述所给定的简化例子中，聚类中心是基于客户的地理位置来定义的，那么操作 402 就可以一开始随机地选择关联地理位置中的 100 个客户。然而，因为这样的选择是随机的，所以可能存在 :如此选择的聚类中心是非代表性的或者要不然的话就是非期望的。举例来说，可能存在 :所有 100 个地理位置都位于彼此非常接近的位置。

[0097] 因此，正如所述的那样，操作 412 可以被设计为计算新的、经更新的聚类中心——其可以更能代表相关样本（例如，客户）的期望特性或者属性的实际的、期望的分布。换句话说，可以确定新的、更新的聚类中心，其使得可能最小化 N ＝ 3 百万个样本中的每一个与至少一个相应聚类中心之间的距离。

[0098] 在 k 均值算法的传统实施方式中，可以通过计算每个样本与其当前或者（如果适用的话）新指派的聚类中心之间的总距离来执行操作 412。然后，可以确定每个聚类中所有样本的均值或者平均数，并且每个这样的计算出的均值在之后都可以用作 k ＝ 100 个聚类中心的新的聚类中心。

[0099] 在图 4 的实施方式中，可以在每个服务器 / 核处并行计算样本中的每一个与它们被新指派到的（如果已经发生了重新指派）聚类的当前中心之间的总距离。随后，可以集中执行新中心的实际计算，并且该算法可以接下来进行操作 404。换句话说，如已知并且如图 4 中所示，相似性计算之后可以如上所述针对操作 404 继续进行，直到达到稳定 (408)，并且流程图 400 的算法完成 (410)。

[0100] 下面提供的伪码部分 1-3 示出了根据图 4 的流程图 400 的示范性实施方式。具体地说，如下所示，伪码 1 示出了用于计算公式 1 和 2 的欧式距离的示范性函数，而伪码 2 示出了使用一定数目的核 C 并行计算距离（例如，相似性）的例子。最后伪码 3 示出了图 4 的 K 均值算法的实施方式，其可以包括伪码部分 1 和 2 的计算。

[0101] 伪码 1

[0102] 1.FUNCTION Euclidean_Distance(Vector x_vec,Vector y_vec)

[0103] 2.BEGIN

```
[0104] 3.for i := 0 to size(x_vec)do
[0105] 4.    distance+ = (x_vec[i]-y_vec[i])^2
[0106] 5.end for
[0107] 6.distance := sqrt(distance)
[0108] 7.return distance
[0109] 8.END
[0110] 伪码 2
[0111] 1.% C :the number of cores
[0112] 2.FUNCTION Euclidean_Distance(Vector x_vec,Vector y_vec)
[0113] 3.BEGIN
[0114] 4.  N := size(x_vec)
[0115] 5.  Vector distance = new Vector[C]
[0116] 6.    On Core 1 :
[0117] 7.       for i := 0 to INT(N/C)do
[0118] 8.           distance[1]+ = (x_vec[i]-y_vec[i])^2
[0119] 9.        end for
[0120] 10.    On Core 2 :
[0121] 11.       for i := INT(N/C)to 2*INT(N/C)do
[0122] 12.           distance[2]+ = (x_vec[i]-y_vec[i])^2
[0123] 13.        end for
[0124] 14.   ... ...
[0125] 15.   On Core C :
[0126] 16.      for i := INT(N(1-1/C))to N do
[0127] 17.          distance[C-1]+ = (x_vec[i]-y_vec[i])^2
[0128] 18.       end for
[0129] 19.   result := sqrt(sum(distance))
[0130] 20.   return result
[0131] END
[0132] 伪码 3
[0133] 1.FUNCTION K_Means_Parallel
[0134] 2.% K :the number of cluster
[0135] 3.% nSamples :the number of samples
[0136] 4.% nDimensions :the number of dimensions
[0137] 5.% [nSamples,nDimensions] := size(inputData)
[0138] 6.% nServers :the number of servers
[0139] 7.
[0140] 8.% Set up the maximum number of iterations
[0141] 9.MAX_ITER := 100
[0142] 10.
```

```
[0143]  11. Matrix center := new Matrix[K][nDimensions]
[0144]  12. center := random select K samples
[0145]  13.
[0146]  14. % Set up storage for the cluster id of samples
[0147]  15. Vector cluster_id := new Vector[nSamples]
[0148]  16. Vector old_cluster_id := new Vector[nSamples]
[0149]  17. old_cluster_id := ones(nSamples)
[0150]  18.
[0151]  19. % Set up storage for the cluster result
[0152]  20. Matrix cluster_result := new Matrix[K][]
[0153]  21.
[0154]  22. while cluster_id ! = old_cluster_id && iter < MAX_ITER do
[0155]  23.   Copy the new centers to S servers
[0156]  24.   old_cluster_id := cluster_id
[0157]  25.   On server 1 :
[0158]  26.      % Set up storage for the similarity between samples on this
server and centers
[0159]  27.       Matrix similarity_on_Server_1 := new Matrix[num_Samples_on_
Server_1][K]
[0160]  28.
[0161]  29.     Matrix sum_on_Server_1 := new Matrix[K][nDimensions]
[0162]  30.      % Compute similarity between the samples on this server and
centers
[0163]  31.     for i := 0 to num_Samples_on_Server_1 do
[0164]  32.         for j := 0 to K do
[0165]  33.             similarity_on_Server_1[i][j] := Euclidean_Distance(
[0166]   Samples_on_Server1[i],center_copy_1[j])
[0167]  34.
[0168]  35.         end for
[0169]  36.       end for
[0170]  37.
[0171]  38.        % Find out the cluster id(with minimum distance)for each
sample
[0172]  39.       for i := 0 to num_Samples_on_Server_1 do
[0173]  40.        id := min_index(similarity_on_Server_1)
[0174]  41.        cluster_id[i] := id
[0175]  42.        cluster_result[id].pushback[i]
[0176]  43.    end for
[0177]  44.
```

```
[0178] 45.    % For each cluster,compute the sum of the corresponding samples on this server
[0179] 46.    for i := 0 to num_Samples_on_Server_1 do
[0180] 47.        if Samples_on_Server1[i].cluster_id == m then
[0181] 48.            sum_on_Server_1[m]+ = Samples_on_Server1[i]
[0182] 49.         end if
[0183] 50.      end for
[0184] 51.
[0185] 52.    On server 2 :
[0186] 53.      % Set up storage for the similarity between samples on this server and centers
[0187] 54.       Matrix similarity_on_Server_2 := new Matrix[num_Samples_on_Server_2][K]
[0188] 55.
[0189] 56.     Matrix sum_on_Server_2 := new Matrix[K][nDimensions]
[0190] 57.      % Compute similarity between the samples on this server and centers
[0191] 58.     for i := 0 to num_Samples_on_Server_2 do
[0192] 59.         for j := 0 to K do
[0193] 60.             similarity_on_Server_2[i][j] := Euclidean_Distance(
[0194]  Samples_on_Server2[i],center_copy_2[j])
[0195] 61.         end for
[0196] 62.      end for
[0197] 63.
[0198] 64.      for i := 0 to num_Samples_on_Server_2 do
[0199] 65.          id := min_index(similarity_on_Server_2)
[0200] 66.          cluster_id[i+num_Sample_on_Server1] := id
[0201] 67.          cluster_result[id].pushback[i+num_Sample_on_Server1]
[0202] 68.       end for
[0203] 69.
[0204] 70.       for i := 0 to num_Samples_on_Server_2 do
[0205] 71.           if Samples_on_Server2[i].cluster_id == m then
[0206] 72.               sum_on_Server_2[m]+ = Samples_on_Server2[i]
[0207] 73.           end if
[0208] 74.        end for
[0209] 75.   ... ...
[0210] 76.   On server S :
[0211] 77.      % Set up storage for the similarity between samples on this server and centers
```

[0212] 78. Matrix similarity_on_Server_S := new Matrix[num_Samples_on_Server_S][K]

[0213] 79.

[0214] 80. Matrix sum_on_Server_S := new Matrix[K][nDimensions]

[0215] 81. % Compute similarity between the samples on this server and centers

[0216] 82. for i := 0 to num_Samples on_Server_S do

[0217] 83. for j := 0 to K do

[0218] 84. similarity_on_Server_S[i][j] := Euclidean_Distance(

[0219] Samples_on_ServerS[i],center_copy_S[j])

[0220] 85. end for

[0221] 86. end for

[0222] 87.

[0223] 88. for i := 0 to num_Samples_on_Server_S do

[0224] 89. id := min_index(similarity_on_Server_S)

[0225] 90. cluster_id[i+nSamples-num_Samples_on_Server_S] := id

[0226] 91. cluster_result[id].pushback[i+nSamples-

[0227] num_Samples_on_Server_S]

[0228] 92. end for

[0229] 93.

[0230] 94. for i := 0 to num_Samples_on_Server_S do

[0231] 95. if Samples_on_ServerS[i].cluster_id == m then

[0232] 96. sum_on_Server_S[m]+ = Samples_on_ServerS[i]

[0233] 97. end if

[0234] 98. end for

[0235] 99.

[0236] 100. % Update the centers

[0237] 101. Matrix sum := new Matirx[K][nDimensions]

[0238] 102. for i := 0 to K do

[0239] 103. sum+ = sum_on_Server_i

[0240] 104. end for

[0241] 105. for i := 0 to K do

[0242] 106. center[i] := sum[i]/size(cluster_result[i])

[0243] 107. end for

[0244] 108.

[0245] 109.end while

[0246] 110.return cluster_result

[0247] 111.END

[0248] 因此,图 1-5B 的特征和功能提供使用这里描述的并行处理技术将样本快速、高效

且及时分组到期望数目的聚类。当然，会认识到，提供的例子仅为了图示的目的，而不意在以某种方式进行限制。举例来说，会认识到，可以在几乎任意如下场景中利用这里描述的技术，在所述场景中希望针对相对较大数据集和相对小得多的数据集来实现这里所述的类型的整合操作。如所述的那样，在这样的场景中，相对较小的数据集的全部都可以被复制到多个可用核中的每一个中，并且较大数据集可以细分为数目对应于可用核的数目的子集。之后，可以进行并行化处理，从而当合并运行的整合操作时，保证在两个数据集的所有组合上的计算结果都被包括在内。因此，在这种整合操作先前局限于传统的串行处理的这样的场景和背景中，可以获得并行处理的好处。

[0249] 这里描述的各种技术的实现方式可以被实施在数字电子电路中，或者实施在计算机硬件、固件、软件，或者它们的组合中。实现方式可以实施为计算机程序产品，即有形地具体实施在信息载体中的计算机程序，信息载体例如在机器可读存储设备中或者在传播的信号中，以供数据处理装置执行或者控制数据处理装置的操作，所述数据处理装置例如可编程处理装置、计算机或多个计算机。计算机程序，诸如上面描述的计算机程序，可以用任何形式的编程语言编写，包括汇编语言或解释语言，并且，它可以被以任何形式部署，包括作为独立的程序或者作为模块、组件、子程序或其他适于在计算环境中使用的单元。计算机程序可以被部署为在一个计算机上执行或在位于一个地点或跨过多个地点分布并被通信网络互连起来的多个计算机上执行。

[0250] 方法步骤可以被一个或多个可编程处理器执行，所述可编程处理器执行计算机程序，以便通过对输入数据操作和产生输出来执行功能。方法步骤还可以被专用逻辑电路执行，或者装置可以被实施为专用逻辑电路，所述专用逻辑电路例如 FPGA（现场可编程门阵列）或 ASIC（专用集成电路）。

[0251] 作为例子，适于执行计算机程序的处理器包括通用和专用微处理器，以及任何类型的数字计算机的任意一个或多个处理器。一般来说，处理器将从只读存储器或随机存取存储器接收指令和数据，或者从两者都接收指令和数据。计算机的元件可以包括至少一个用于执行指令的处理器，和用于存储指令和数据的一个或多个存储器设备。一般来说，计算机还可以包括，或者被可操作地连接，以从一个或多个用于存储数据的海量储存设备接收数据，或把数据传送到海量储存设备，或者二者皆有，所述海量储存设备例如 :磁盘、磁光盘或光盘。适于具体实施计算机程序指令和数据的信息载体包括所有形式的非易失性存储器，作为例子，包括半导体存储器器件，例如 :EPROM、EEPROM 和闪存设备、磁盘，例如内置硬盘或可移动磁盘、磁光盘和 CD-ROM 以及 DVD-ROM 盘。处理器和存储器可以以专用逻辑电路补充，或者被包含在专用逻辑电路中。

[0252] 为了提供和用户的交互，实现方式可以在具有显示设备和键盘以及定点设备的计算机上实施，显示设备例如阴极射线管 (CRT) 或液晶显示器 (LCD) 监视器，用于向用户显示信息，键盘和指示设备例如鼠标或轨迹球，用户利用它们可以提供到计算机的输入。其他种类的设备也可以被用来提供和用户的交互 ;例如，提供给用户的反馈可以是任何形式的感觉反馈，例如视觉反馈、听觉反馈或触觉反馈，并且，可以以任何形式接收来自用户的输入，包括声音、语音或触觉输入。

[0253] 实现方式可以被在包括后端组件或包括中间件组件或包括前端组件的计算系统中实施，或者在这些后端、中间件、前端组件的任意组合中实施，后端组件例如数据服务器，

中间件组件例如应用服务器，前端组件例如具有图形用户界面，或 Web 浏览器的客户端计算机，通过图形用户界面或 Web 浏览器，用户可以和实现方式进行交互。可以利用数字数据通信的任何形式或介质互连组件，数字数据通信介质例如通信网络。通信网络的例子包括：局域网 (LAN) 和广域网 (WAN)，例如因特网。

[0254] 虽然如这里所描述的那样已经示出了所描述的实现方式的某些特征，但是本领域普通技术人员现在应当想到很多修改、替换、变化或等同物。因此应当理解，所附权利要求旨在覆盖落入实施例的实质精神内的所有这样的修改和变化。

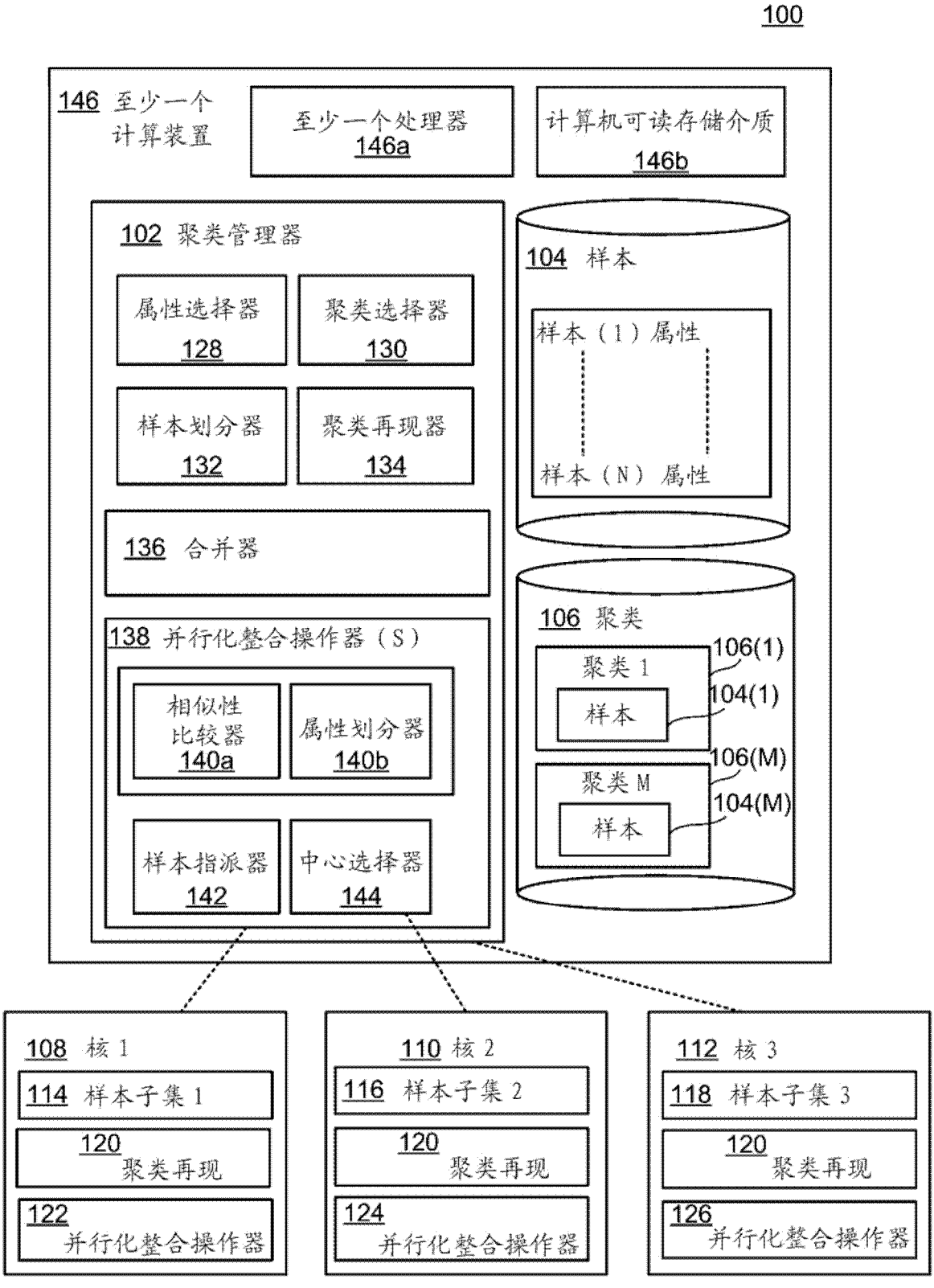

100
146 至少一个计算装置
至少一个处理器 146a
计算机可读存储介质 146b
102 聚类管理器
属性选择器 128
聚类选择器 130
样本划分器 132
聚类再现器 134
136 合并器
138 并行化整合操作器（S）
相似性比较器 140a
属性划分器 140b
样本指派器 142
中心选择器 144
104 样本
样本（1）属性
样本（N）属性
106 聚类
聚类 1
样本
106(1)
104(1)
聚类 M
样本
106(M)
104(M)
108 核 1
114 样本子集 1
120 聚类再现
122 并行化整合操作器
110 核 2
116 样本子集 2
120 聚类再现
124 并行化整合操作器
112 核 3
118 样本子集 3
120 聚类再现
126 并行化整合操作器

图 1

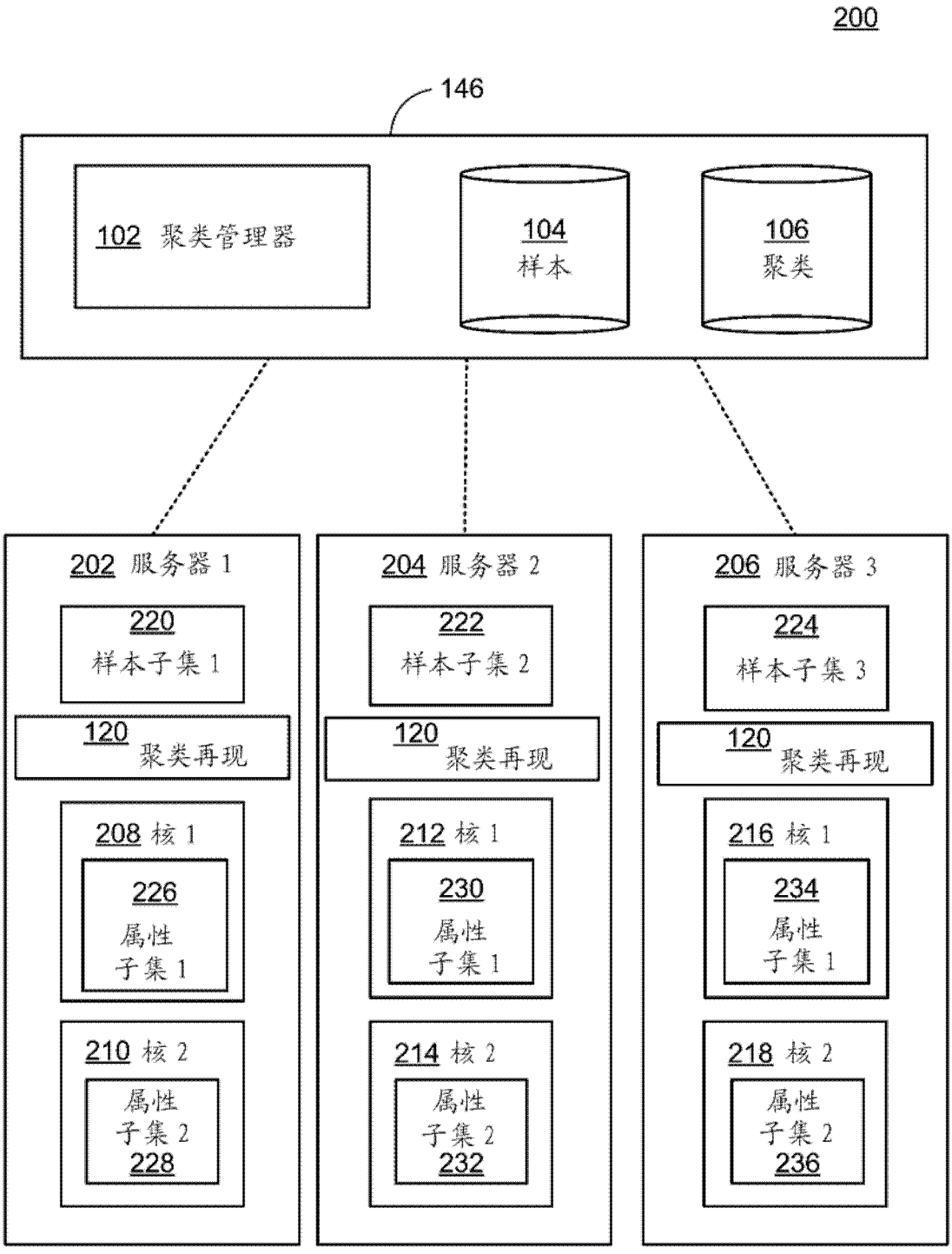

200
146
102 聚类管理器
104 样本
106 聚类
202 服务器 1
220 样本子集 1
120 聚类再现
208 核 1
226 属性子集 1
210 核 2
属性子集 2 228
204 服务器 2
222 样本子集 2
120 聚类再现
212 核 1
230 属性子集 1
214 核 2
属性子集 2 232
206 服务器 3
224 样本子集 3
120 聚类再现
216 核 1
234 属性子集 1
218 核 2
属性子集 2 236

图 2

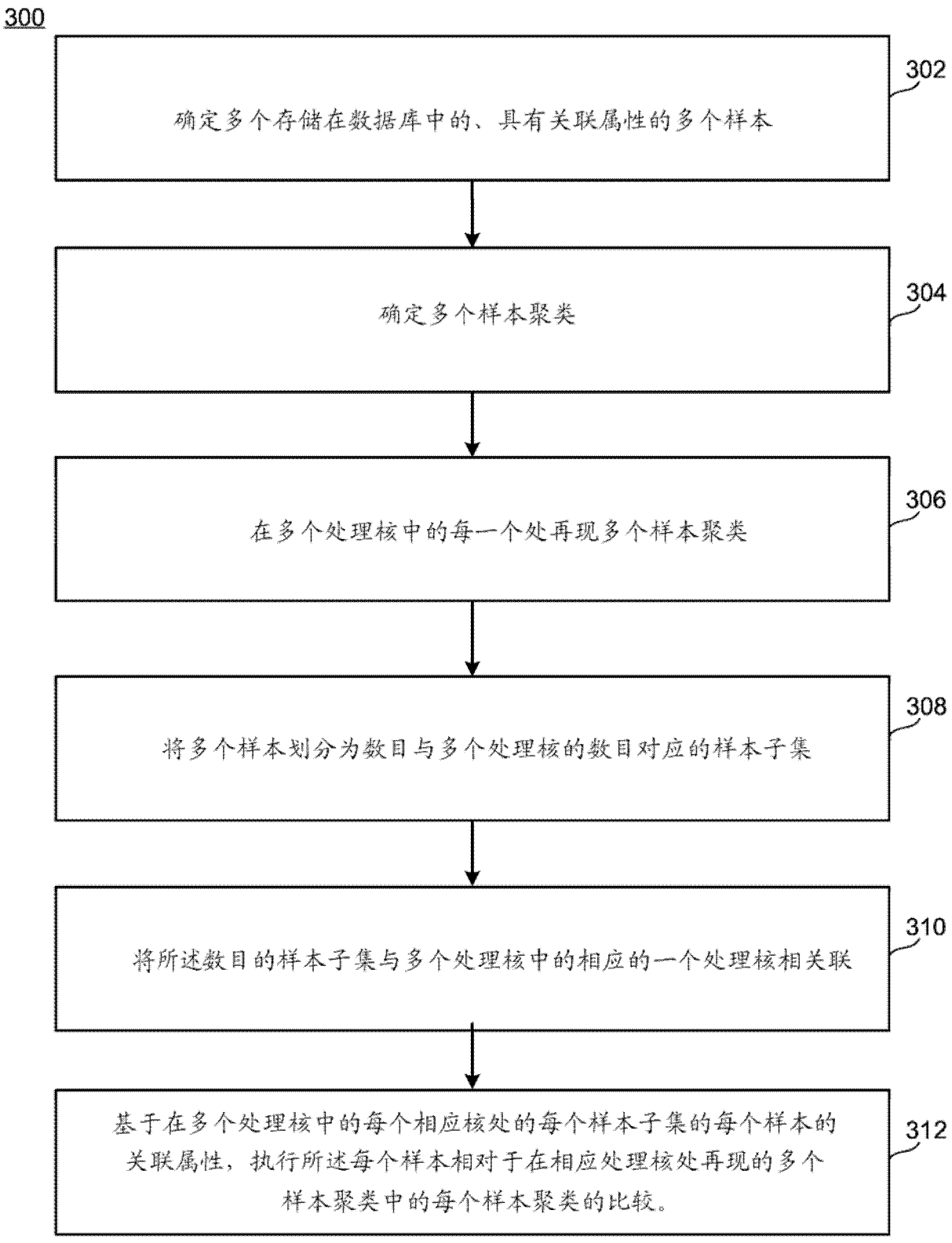

300
确定多个存储在数据库中的、具有关联属性的多个样本
302
确定多个样本聚类
304
在多个处理核中的每一个处再现多个样本聚类
306
将多个样本划分为数目与多个处理核的数目对应的样本子集
308
将所述数目的样本子集与多个处理核中的相应的一个处理核相关联
310
基于在多个处理核中的每个相应核处的每个样本子集的每个样本的关联属性，执行所述每个样本相对于在相应处理核处再现的多个样本聚类中的每个样本聚类的比较。
312

图 3

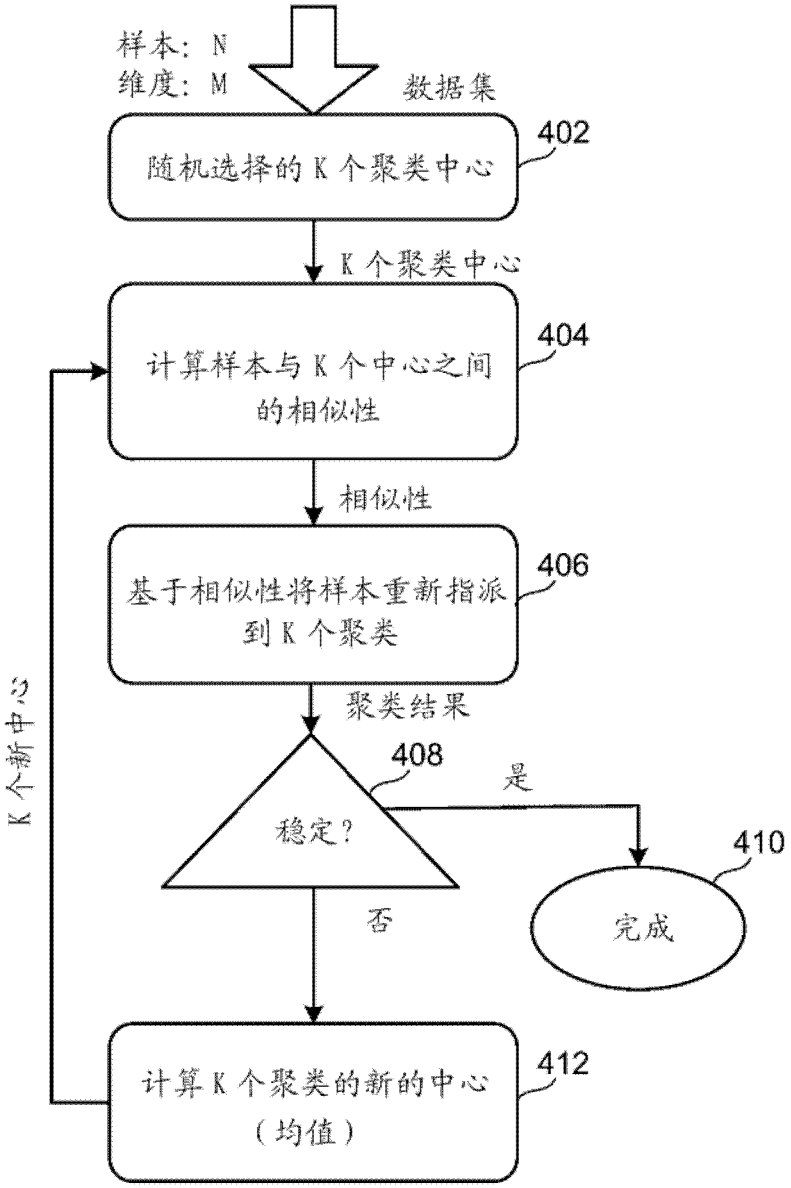

样本：N
维度：M
数据集
随机选择的K个聚类中心
402
K个聚类中心
计算样本与K个中心之间的相似性
404
相似性
基于相似性将样本重新指派到K个聚类
406
聚类结果
408
稳定？
是
410
完成
否
计算K个聚类的新的中心（均值）
412
K个新中心

图 4

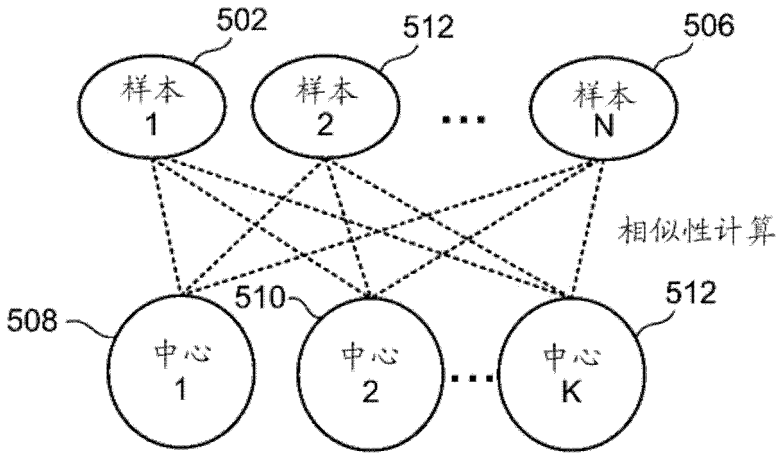

502
512
506
样本 1
样本 2
样本 N
相似性计算
508
510
512
中心 1
中心 2
中心 K

图 5A

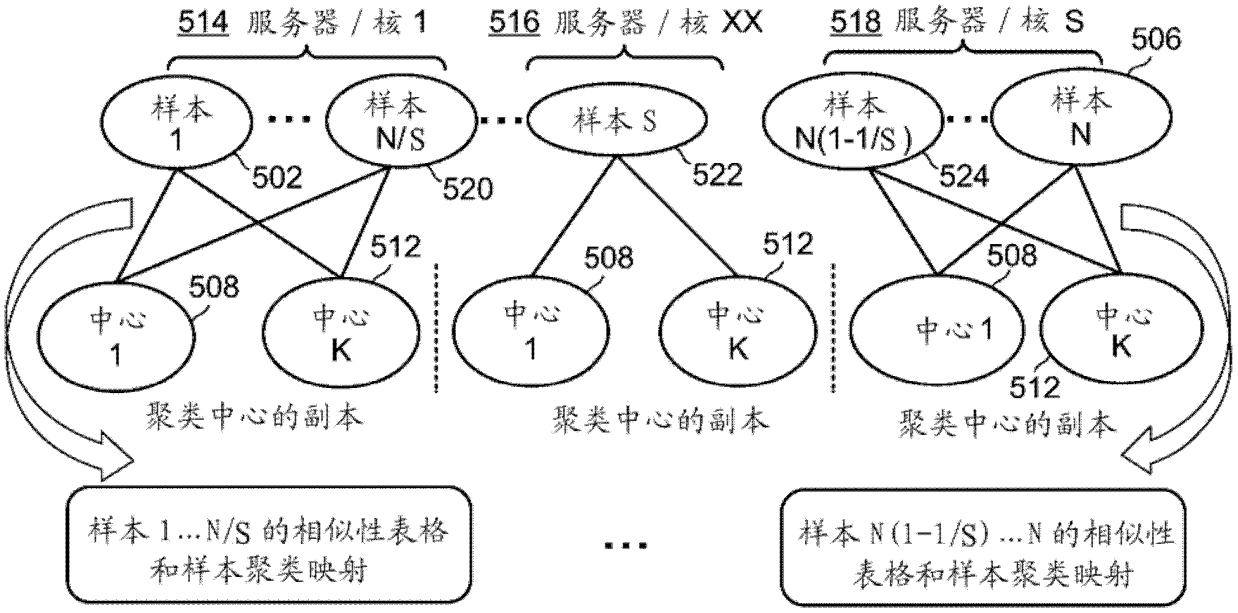

514 服务器 / 核 1
516 服务器 / 核 XX
518 服务器 / 核 S
506
样本 1
样本 N/S
样本 S
样本 N(1-1/S)
样本 N
502
520
522
524
512
508
中心 1
中心 K
聚类中心的副本
聚类中心的副本
聚类中心的副本
样本 1...N/S 的相似性表格和样本聚类映射
样本 N(1-1/S)...N 的相似性表格和样本聚类映射

图 5B