



(12) 发明专利申请

(10) 申请公布号 CN 116685961 A

(43) 申请公布日 2023. 09. 01

(21) 申请号 202280009263.1

(74) 专利代理机构 北京市金杜律师事务所

(22) 申请日 2022.01.10

11256

专利代理师 姚宗妮

(30) 优先权数据

17/148,644 2021.01.14 US

(51) Int.Cl.

G06F 16/23 (2006.01)

(85) PCT国际申请进入国家阶段日

2023.07.06

(86) PCT国际申请的申请数据

PCT/CN2022/071056 2022.01.10

(87) PCT国际申请的公布数据

WO2022/152085 EN 2022.07.21

(71) 申请人 国际商业机器公司

地址 美国纽约阿芒克

(72) 发明人 K·斯托尔兹 F·拜尔

R·盖塞尔哈特

L·E·奥利维拉·利萨多

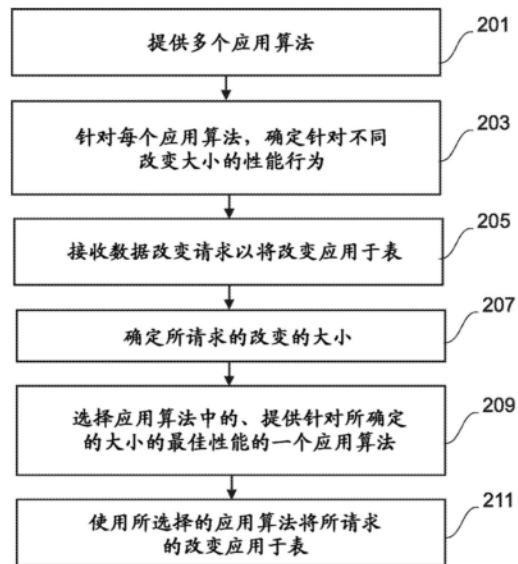
权利要求书4页 说明书10页 附图7页

(54) 发明名称

在目标数据库系统中应用改变

(57) 摘要

在用于动态地选择用于目标数据库系统中的每个改变的应用算法的方法中,处理器提供至少两个应用算法以用于将改变应用于目标数据库系统中的表。处理器针对至少两个应用算法中的每个应用算法,确定每个应用算法的、针对由相应的应用算法应用于该表的改变大小的性能行为。处理器接收用于将改变应用于该表的数据改变请求。处理器确定表的改变的大小。处理器基于每个应用算法的性能行为选择至少两个应用算法中的、提供针对该大小的最佳性能的一个应用算法。处理器使用为所选择的提供针对该大小的最佳性能的应用算法来将该改变应用于该表。



1. 一种计算机实现的方法,包括:

由一个或多个处理器提供至少两个应用算法,所述至少两个应用算法用于将改变应用于目标数据库系统中的表;

由所述一个或多个处理器针对所述至少两个应用算法中的每个应用算法来确定每个应用算法的、针对由相应的所述应用算法应用于所述表的改变大小的性能行为;

由所述一个或多个处理器接收用于将改变应用于所述表的数据改变请求;

由一个或多个处理器确定对所述表的所述改变的大小;

由所述一个或多个处理器基于每个应用算法的所述性能行为来选择所述至少两个应用算法中的、提供针对所述大小的最佳性能的一个应用算法;以及

由所述一个或多个处理器使用所选择的提供针对所述大小的所述最佳性能的所述应用算法来将所述改变应用于所述表。

2. 根据权利要求1所述的计算机实现的方法,其中将所述改变应用于所述表还包括:

由所述一个或多个处理器通过将所述改变应用于所述表来评估所选择的所述应用算法的性能;以及

由所述一个或多个处理器使用所选择的所述应用算法的所述性能和所述表的所述大小来更新所选择的所述应用算法的所述性能行为。

3. 根据权利要求2所述的计算机实现的方法,还包括:

由所述一个或多个处理器针对所述表的每个接收到的数据改变请求来重复所述确定所述大小、所述选择、以及所述应用;以及

由所述一个或多个处理器动态地更新所述表的所述性能行为。

4. 根据权利要求1所述的计算机实现的方法,还包括:

由所述一个或多个处理器确定针对所述目标数据库系统中的每个另外的表的相应的所述性能行为,从而引起所述目标数据库系统中的每个表与相应的性能行为相关联;以及

由所述一个或多个处理器使用与所述目标数据库系统中的特定表相关联的所述性能行为,针对所述特定表的每个接收到的数据改变请求来重复所述确定所述大小、所述选择和所述应用。

5. 根据权利要求1所述的计算机实现的方法,其中确定每个应用算法的所述性能行为包括:

由所述一个或多个处理器将相应的所述应用算法执行预定义数目N次,以用于分别将数据改变应用于所述表,其中每个所应用的数据改变具有大小;

由所述一个或多个处理器针对所述数据改变中的每个数据改变评估指示所述应用算法的所述执行的性能的至少一个性能参数;以及

由所述一个或多个处理器提供表示所述性能行为的N个数据点的数据结构,其中每个数据点指示所述至少一个性能参数和所述数据改变的相关联大小。

6. 根据权利要求5所述的计算机实现的方法,还包括:

其中所述预定数目N小于可配置的最大执行数目;以及

由所述一个或多个处理器通过使用所述N个数据点的内插来构建所述数据结构中的新数据点。

7. 根据权利要求5所述的计算机实现的方法,其中所述性能参数选自自由执行时间和存

储器使用量组成的组。

8. 根据权利要求1所述的计算机实现的方法,其中所述改变的所述大小包括以下至少一项:待插入的记录数目、待删除的记录数目、以及待更新的记录数目。

9. 根据权利要求1所述的计算机实现的方法,其中选择所述应用算法包括:

由所述一个或多个处理器确定两个应用算法的针对所述大小的性能是否相似;以及响应于确定两个应用算法的针对所述大小的所述性能相似,由所述一个或多个处理器选择所述两个应用算法中的任一应用算法。

10. 根据权利要求1所述的计算机实现的方法,其中所选择的所述应用算法选自由基于记录批量加载的应用和基于单独记录加载的应用组成的组。

11. 根据权利要求1所述的计算机实现的方法,还包括:

由所述一个或多个处理器提供源数据库系统中与所述表相关联的源表,其中所述源数据库系统和所述目标数据库系统被配置为在彼此之间同步数据;并且

其中所述数据改变请求响应于所述数据改变被应用于所述源表而被接收,由此复制所述数据改变。

12. 一种计算机程序产品,包括:

一个或多个计算机可读存储介质以及被共同存储在所述一个或多个计算机可读存储介质上的程序指令,所存储的所述程序指令包括:

用于提供至少两个应用算法的程序指令,所述至少两个应用算法用于将改变应用于目标数据库系统中的表;

用于针对所述至少两个应用算法中的每个应用算法来确定每个应用算法的、针对由相应的所述应用算法应用于所述表的改变大小的性能行为的程序指令;

用于接收用于将改变应用于所述表的数据改变请求的程序指令;

用于确定对所述表的所述改变的大小的程序指令;

用于基于每个应用算法的所述性能行为来选择所述至少两个应用算法中的、提供针对所述大小的最佳性能的一个应用算法的程序指令;以及

用于使用所选择的提供针对所述大小的所述最佳性能的所述应用算法来将所述改变应用于所述表的程序指令。

13. 根据权利要求12所述的计算机程序产品,其中将所述改变应用于所述表的所述程序指令还包括:

用于通过将所述改变应用于所述表来评估所选择的所述应用算法的性能的程序指令;以及

用于使用所选择的所述应用算法的所述性能和所述表的所述大小来更新所选择的所述应用算法的所述性能行为的程序指令。

14. 根据权利要求13所述的计算机程序产品,还包括:

用于针对所述表的每个接收到的数据改变请求来重复用于确定所述大小的所述程序指令、用于选择的所述程序指令、以及用于应用的所述程序指令的程序指令;以及

用于动态地更新所述表的所述性能行为的程序指令。

15. 根据权利要求12所述的计算机程序产品,还包括:

用于确定针对所述目标数据库系统中的每个另外的表的相应的所述性能行为、从而引

起所述目标数据库系统中的每个表与相应的性能行为相关联的程序指令;以及

用于使用与所述目标数据库系统中的特定表相关联的所述性能行为,针对所述特定表的每个接收到的数据改变请求来重复用于确定所述大小的所述程序指令、用于选择的所述程序指令、以及用于应用的所述程序指令的程序指令。

16. 根据权利要求12所述的计算机程序产品,其中用于确定每个应用算法的所述性能行为的所述程序指令包括:

用于将相应的所述应用算法执行预定义数目N次以用于分别将数据改变应用于所述表的程序指令,其中每个所应用数据改变具有大小;

用于针对所述数据改变中的每个数据改变评估指示所述应用算法的所述执行的性能的至少一个性能参数的程序指令;以及

用于提供表示所述性能行为的N个数据点的数据结构的数据结构的程序指令,其中每个数据点指示所述至少一个性能参数和所述数据改变的相关联大小。

17. 一种计算机系统,包括:

一个或多个计算机处理器;

一个或多个计算机可读存储介质;

程序指令,被共同地存储在所述一个或多个计算机可读存储介质上,用于由所述一个或多个计算机处理器中的至少一个计算机处理器执行,所存储的所述程序指令包括:

用于提供至少两个应用算法的程序指令,所述至少两个应用算法用于将改变应用于目标数据库系统中的表;

用于针对所述至少两个应用算法中的每个应用算法来确定每个应用算法的、针对由相应的所述应用算法应用于所述表的改变大小的程序指令;

用于接收用于将改变应用于所述表的数据改变请求的程序指令;

用于确定对所述表的所述改变的大小的程序指令;

用于基于每个应用算法的所述性能行为来选择所述至少两个应用算法中的、提供针对所述大小的最佳性能的一个应用算法的程序指令;以及

用于使用所选择的提供针对所述大小的所述最佳性能的所述应用算法,将所述改变应用于所述表的程序指令。

18. 根据权利要求17所述的计算机系统,其中将所述改变应用于所述表的所述程序指令还包括:

用于通过将所述改变应用于所述表来评估所选择的所述应用算法的性能的所述程序指令;以及

用于使用所选择的所述应用算法的所述性能和所述表的所述大小来更新所选择的所述应用算法的所述性能行为的程序指令。

19. 根据权利要求18所述的计算机系统,还包括:

用于针对所述表的每个接收到的数据改变请求来重复用于确定所述大小的所述程序指令、用于选择的所述程序指令、以及用于应用的所述程序指令的程序指令;以及

用于动态地更新所述表的所述性能行为的程序指令。

20. 根据权利要求17所述的计算机系统,还包括:

用于确定针对所述目标数据库系统中的每个另外的表的相应的所述性能行为、从而引

起所述目标数据库系统中的每个表与相应的性能行为相关联的程序指令;以及

用于使用与所述目标数据库系统中的特定表相关联的所述性能行为,针对所述特定表的每个接收到的数据改变请求来重复用于确定所述大小的所述程序指令、用于选择的所述程序指令、以及用于应用的所述程序指令的程序指令。

## 在目标数据库系统中应用改变

### 背景技术

[0001] 本发明涉及数据库系统领域,并且更具体地,涉及提供用于动态地选择用于在目标数据库系统中应用改变的应用算法的方法。

[0002] 复制是在多于一个位置中维持定义的数据集的过程。它可以涉及将指定的改变从一个源位置复制到目标位置,并且使两个位置中的数据同步。源和目标可以在分布式网络中的相同机器或不同机器上的逻辑服务器中。存在用于将数据从一个系统移动到另一系统的若干方法。然而,这些方法可能需要进一步改进。

### 发明内容

[0003] 本发明的实施例的各方面公开了一种用于动态地选择要用于将改变应用于目标数据库系统的应用算法的计算机实现的方法、计算机程序产品和计算机系统。如果本发明的实施例不相互排斥,则它们可以彼此自由组合。

[0004] 处理器提供用于将改变应用于目标数据库系统的多个应用算法。处理器针对所提供的每个应用算法,确定应用算法的、针对由应用算法应用于目标数据库系统的表的不同的改变大小的性能行为。处理器接收用于将改变应用与表的数据改变请求。处理器确定对表的所请求改变的大小。处理器使用所确定的性能行为来选择应用算法中的、提供针对所确定的大小的最佳性能的一个应用算法。处理器使用所选择的应用算法将所请求的改变应用于所述表。

### 附图说明

[0005] 图1示出了根据本发明实施例的数据处理系统。

[0006] 图2是根据本发明实施例的用于在目标数据库系统中应用改变的方法的流程图。

[0007] 图3是根据本发明实施例的用于在目标数据库系统中应用改变的方法的流程图。

[0008] 图4是根据本发明实施例的用于在目标数据库系统中应用改变的方法的流程图。

[0009] 图5是根据本发明实施例的用于在目标数据库系统中应用改变的方法的流程图。

[0010] 图6A是根据本发明的实施例的用于确定应用算法的性能行为的方法的流程图。

[0011] 图6B是根据本发明的实施例的用于确定应用算法的性能行为的方法的流程图。

[0012] 图6C是示出根据本发明的实施例的应用算法的性能行为的曲线。

[0013] 图7表示适于实现根据本发明的实施例的一个或多个方法步骤的计算机化系统。

### 具体实施方式

[0014] 本发明的不同实施例的描述将出于说明的目的而呈现,但并不旨在是详尽的或限于所公开的实施例。在不脱离所描述的实施例的范围的情况下,许多修改和改变对于本领域普通技术人员来说是显而易见的。这里使用的术语被选择来最好地解释实施例的原理、实际应用或对在市场中找到的技术的技术改进,或者使得本领域普通技术人员能够理解这里公开的实施例。

[0015] 目标数据库系统可以是数据分析系统的一部分。数据分析系统包括目标数据库系统和源数据库系统。数据分析系统可以例如是数据仓储系统或主数据管理系统。该数据分析系统可以实现数据仓储、主数据管理或使用源和目标数据库系统的另一种技术,其中,该目标数据库系统包括目标数据库,该目标数据库被配置成用于接收/包括该源数据库系统的相应源数据库的内容的副本。源数据库系统可以经由连接来连接至目标数据库系统。该连接例如可以是TCP/IP连接或者经由源数据库系统和目标数据库系统之间的连接实现数据通信的另一连接。源数据库系统可以例如是事务引擎,并且目标数据库系统可以是分析引擎。例如,源数据库系统可以是在线事务处理(OLTP)系统,目标数据库系统可以是在线分析处理(OLAP)系统。源数据库系统可以包括源数据集,并且目标数据库系统可以包括目标数据集。源数据集可以是源数据库的一部分,并且目标数据集可以是目标数据库的一部分。源和目标数据集可按相同或不同的格式存储。格式可以在加密、压缩、面向行的存储对面向列的存储等方面不同。例如,源数据集可以以面向行的格式来存储,目标数据集可以以面向列的格式来存储,即,目标数据集可以按列而不是按行来存储。源数据集的内容可通过一个或多个数据库事务而改变。

[0016] 数据分析系统可以被配置来将发生在源数据库系统的源表中的改变复制到目标数据库系统,使得所述改变可以被应用于对应于源表的目标数据库系统的目标表上。应用改变可以例如包括在目标数据库系统的一个或多个表中插入一个或多个记录、更新一个或多个记录、和/或删除一个或多个记录。数据记录或表的记录是相关数据项的集合,诸如特定用户的姓名、出生日期和类别。记录表示实体,其中,实体是指关于其信息被存储在记录中的用户、对象或概念。为此,可以提供多个应用算法(其还可以被称为更新策略),其中每个应用算法指定要执行的复制操作的序列,以便将改变应用于目标数据库系统。例如,应用算法可以包括基于增量加载的算法和基于批量加载的算法。例如,基于增量加载的算法可要求日志记录的每个记录改变在目标数据库系统中单独应用。增量基于加载的算法对小数据集尤其有利,因为大块的开销可能高。基于批量加载的应用算法可例如要求将日志记录的所记录改变分段成批。然后可以经由批量加载接口将那些批次应用于目标数据库系统。基于批量加载的应用算法可以有利地用于大数据集。然而,设置批量加载的开销可能过高,并且不应当花费在仅由几行组成的小尺寸块中。

[0017] 所以,取决于要在目标数据库系统上应用的改变,应用算法可以具有不同的性能。然而,准确并以系统方式选择合适的算法可能不是直截了当的动作。本发明可以通过提供用于动态地选择用于每次改变的应用算法的最优和有效的方法来解决该问题。该方法在时间方面可以是高效的,因为它可节省否则将不适当的应用算法所需的附加时间。该方法可以是最优的,因为这些决策可以基于准确和最新的性能行为。

[0018] 每个应用算法的性能行为例如可以由数据结构来表示。该数据结构可以包括数据点 $p_1, p_2 \dots p_N$  ( $N \geq 2$ ), 其中,每个数据点 $p_i$  ( $i = 1, \dots, N$ ) 包括改变大小 $s_i$ 值和 $m$  ( $m \geq 1$ ) 个性能参数值 $l_1^i \dots l_m^i$ 。该数目 $N$ 可以反映改变的数目。在一个示例中,数量 $N$ 可以是基于时间的,例如,数量 $N$ 可以由最后两小时内的所有操作引起的改变的数量。例如,数据点 $p_1$ 可被定义为 $p_1 = (s_1, l_1^1 \dots l_m^1)$ , 数据点 $p_2$ 可被定义为 $p_2 = (s_2, l_1^2 \dots l_m^2)$ , 等等。例如,如果性能参数的数量是1,即 $m = 1$ , 数据结构可以是二维结构。二维结构的数据点可以限定二维空间,该二维

空间可以例如由两个轴的曲线表示,一个轴与改变的大小相关联并且另一个轴与性能参数的值相关联。例如,如果性能参数的数量是2个(即, $m=2$ ),则数据结构可以是三维结构。三维结构的数据点可以定义三维空间,三维空间例如可以由三个轴的立方体来表示,一个轴与改变的大小相关联,而另外两个轴分别与性能参数的值相关联。由此,数据结构的数据点可以定义 $(m+1)$ 维空间。在接收到在目标数据库系统中应用改变的新请求时,所请求的改变的大小 $s_t$ 例如可用于在 $(m+1)$ 维空间中找到不同的最近点,并且选择与那些最近点中的为该大小 $s_t$ 提供最佳性能的一个相关联的应用算法。不同的最近点意味着每个最近点与各自的不同的应用算法程序相关联。性能参数的值可以例如取决于列的数量、列的数据类型、硬件配置(诸如CPU速度)、存储器的类型、CPU高速缓存的大小等。

[0019] 根据一个实施例,该方法还包括通过应用所请求的改变来评估所选择的应用算法的性能,并且使用所评估的性能和所确定的大小来更新所选择的应用算法的性能行为。例如,可以针对该请求的具有大小 $s_t$ 的改变来评估 $m$ 个性能参数。这可导致与所选应用算法相关联的数据结构中的新点 $p_t = (s_t, l_1^t \dots l_m^t)$ 。该实施例可以是有利的,因为它动态地更新所选择的应用算法的性能行为。这可以实现自调谐系统。

[0020] 根据一个实施例,该方法还包括为表的每个接收到的数据改变请求重复确定大小的步骤、选择应用算法的步骤和应用改变的步骤。这可以使得能够动态地更新表的性能行为。

[0021] 根据一个实施例,该方法还包括:对该目标数据库系统的每个另外的表执行该性能行为确定步骤,从而引起该目标数据库系统的每个表与对应的性能行为相关联;以及对使用与目标数据库系统的特定表相关联的性能行为的所述特定表的每个接收到的数据改变请求重复确定所述大小的步骤、选择所述应用算法的步骤和应用所述改变的步骤。该实施例可以是有利的,因为它使得应用算法的性能行为也取决于表。这可能尤其有利,因为对于具有少数列的给定表的改变的给定大小,相同的应用算法可以是最高效的算法,但是对于具有更多列的另一表,对于相同的改变大小,这可能是不太高效的,或者这些列可以具有不同的数据类型。

[0022] 根据一个实施例,确定应用算法的性能行为包括:执行应用算法用于分别向表应用数据改变的预定义数目 $N$ 次,其中每个所应用的数据改变具有大小;针对所述数据改变中的每个数据改变评估指示所述应用算法的所述执行的性能的至少一个性能参数;以及提供表示所述性能行为的 $N$ 个数据点的数据结构,其中每个数据点指示所评估的性能参数和所述数据改变的相关联大小。该实施例可以是有利的,因为在数据分析系统的初始设置时,可能还没有可用的测量。该实施例可以实现提供初始测量的应用算法的初始执行,该初始测量可以稍后被细化。在一个示例中,应用算法的初始执行可以是具有虚拟数据的虚拟执行,例如,1000行可以插入到表中并且可以测量执行时间。这些1000行的改变可以被回滚而不被提交。在另一示例中,应用算法的执行可由 $N$ 个接收到的改变请求引起。应用算法可以在预定算法中随机地或经由循环或某个其他方案来选择。这样做直到有足够的测量 $N$ 个点可用。

[0023] 根据一个实施例,该数量 $N$ 小于可配置的最大执行次数(例如3次)。该方法还包括通过使用 $N$ 个数据点的内插来构建数据结构中的新数据点。例如,可以使用用于插入多个行的每个应用算法来执行2个(即, $N=2$ ) INSERT(插入)。两个INSERT的行数可以是不同的,并



且具有更大的多样性,例如,100行和700行(例如,但不是100行和101行)。可以针对两个INSERT中的每一个来评估(多个)性能参数。这限定了第一基线,例如,以线性曲线的形式,随后能够在该线性曲线上应用外推。样条可用于近似曲线或简单的线性内插。

[0024] 根据一个实施例,性能参数是执行时间和存储器使用量中的任一个。

[0025] 根据一个实施例,改变的大小包括待插入的记录的数量、待删除的记录的数量、和/或待更新的记录的数量。

[0026] 根据一个实施例,选择应用算法包括确定两个应用算法的针对所确定大小的性能是否相似,并且如果它们相似,则选择这两个应用算法中的任一个。两个应用算法可以在它们的性能行为的交叉点处具有类似的性能。这些交叉点可能具有非常重要的特征:这些交叉点是性能方面的无断点。用算法A或算法B处理“n”行的性能是相同的。这意味着,选择哪个算法无关紧要。稍微偏离交叉点不会在性能曲线中产生显著的“跳跃”。例如,如果算法A对于小于1000行较好且算法B对于大于1000行较好,那么针对990行选择B仍非常接近于算法A。当从算法A切换到算法B时,这可以具有平滑过渡的极有帮助的意义。

[0027] 根据一个实施例,应用算法是基于记录批量加载的应用和基于单独记录加载的应用中的任一个。

[0028] 根据一个实施例,该方法还包括在提供源数据库系统中与该表相关联的源表,其中该源数据库系统和目标数据库系统被配置成用于在彼此之间同步数据;并且其中响应于所述数据改变被应用于所述源表而接收所述数据改变请求,由此复制所述数据改变。

[0029] 本发明的实施例的实现方式可以采取多种形式,并且随后参照附图讨论示范性实现方式细节。

[0030] 图1是根据本发明的实施方式的数据处理系统(或数据分析系统)100的框图。根据本发明的实施例,数据处理系统100可以被配置成用于使用数据同步系统102在源数据库系统101与目标数据库系统103之间进行数据同步。源数据库系统101可以是例如在线事务处理(OLTP)系统。目标数据库系统103可以是例如在线分析处理(OLAP)系统。源数据库系统101和目标数据库系统103之间的通信可例如经由TCP/IP通信层来执行。

[0031] 源数据库系统101包括源数据库106的一个或多个源表105和事务恢复日志107。事务恢复日志107的条目或日志记录描述在源数据库系统101处对源表105的行或记录的改变。图1示出了日志记录130的示例内容。日志记录130可包括时间戳、LRSN和属性改变。更具体地,事务恢复日志107中的日志记录可例如包含定义(1)正被改变的表、(2)正被改变的行中的键列的值、(3)被改变的行的所有列的旧值和新值、以及(4)引起改变的事务(工作的单元)的信息。根据定义,插入是新的数据记录,并且因此不具有旧值。对于删除改变,根据定义不存在新的数据记录,仅存在旧的数据记录。因此,用于插入的行的事务日志记录可仅包含新的列值,而用于删除的行的事务日志记录可仅包含旧的列值。更新行的事务日志记录可包含所有行列的新值和旧值。事务恢复日志中的日志记录的顺序可反映事务的改变操作的顺序,并且事务提交记录的顺序可反映完成事务的顺序。事务日志记录中的行操作的类型可例如为删除、插入或更新。

[0032] 数据同步系统102包括日志读取器104。尽管被示为数据同步系统102的一部分,但在另一示例中,日志读取器104可以是源数据库系统101的一部分。日志读取器104可读取事务恢复日志107的日志记录,并将它们提供给改变记录分类器120。改变记录分类器120可基

于日志记录的改变来对其进行分类(例如,以确定改变的大小),使得算法选择模块121可基于日志记录的分类来选择应用算法108.1至108.3中的一个,并且可基于所选择的应用算法来执行更新。应用算法选择和使用所述应用算法的改变应用可以例如定期地(例如每小时)执行,或者可以在事务恢复日志107中保存预定义量的日志记录(例如,1000个日志记录)时自动执行,例如所述量的日志记录可用于定义新的改变请求。在每次迭代中,可仅处理相对于先前提供的日志记录的新添加的日志记录。所选择的应用算法可以例如包括基于批量加载的更新策略或一个或多个增量更新策略,分别对应于基于批量加载的算法108.3和增量算法108.1和108.2。可以针对基于批量加载的更新策略和增量更新策略不同地执行同步。

[0033] 日志读取器104可被配置来基于由选择模块121选择的增量更新算法执行事务恢复日志107至目标数据库系统103的日志运送(log shipping)。运送可例如通过发送由事务恢复日志107的日志记录形成的日志记录流来执行。正在运送的日志流可以例如与流ID相关联。日志记录的流可例如是合并日志记录的流。这可在目标数据库系统103处实现日志记录的高效处理。目标数据库系统103可以包括多个算法108.1和108.2,每个算法与对应的增量更新策略相关联。图1出于示例性目的仅示出了三种算法,但其不限于这些类型或数量的算法。目标数据库系统103还包括一个或多个目标表副本113。目标数据库系统包括日志流送接口,用于从源数据库系统101接收日志流。算法108.1和108.2中的每个可被配置为经由日志流送接口接收日志记录的流。算法108.1和108.2中的每个可缓冲所接收的日志记录并且将改变合并成批次以提高当将修改应用于目标数据库114的表副本113时的效率,例如经由批量加载接口。

[0034] 在另一示例中,基于批量加载(快照更新)算法108.3可以基于选择的基于批量加载的更新策略在源数据库系统101和目标数据库系统103之间执行。加载可以是整个表数据或在给定时间点的表的分区集合的加载,并且从源数据库系统101到目标数据库系统103直接执行。目标数据库系统103上的数据可以反映在执行加载时的源数据库系统状态。

[0035] 虽然示出为单独的部件,但是在另一个示例中,数据同步系统102可以是源数据库系统101的一部分或者目标数据库系统103的一部分。在一个示例中,源数据库系统101和目标数据库系统103分别可以在分布式网络中的相同系统上或不同系统上。

[0036] 图2是根据本发明实施例的用于向目标数据库系统的表 $T_g$ 应用改变的方法的流程图。为了说明的目的,在图2中描述的方法可以在图1中示出的系统中实现,但不限于该实施方式。图2的方法可例如由数据同步系统102执行。图2的方法可以例如使得能够将源数据库系统的源表 $T_s$ (对应于 $T_g$ )中做出的改变应用至目标数据库系统,并且因此可以使得能够在源数据库系统与目标数据库系统之间同步。

[0037] 在步骤201中,可以提供多个应用算法,用于在目标数据库系统中应用改变。例如,若干应用算法 $App_1, \dots, App_r$ ,其中 $r \geq 2$ 。例如,应用算法可以包括单个记录应用算法和批量加载应用算法。单个记录应用算法可以单独地应用在每个日志记录中记录的每个改变。每个算法可以不同方式应用改变,从而有效地导致其他应用算法。例如,单个记录应用算法可仅更新自表 $T_g$ 的最后更新以来改变的数据,或者可更新整个表 $T_g$ ,每当在相应的源表 $T_s$ 处发生改变时。这可提供两种不同的单个记录应用算法。

[0038] 可以在步骤203中确定应用算法 $App_1, \dots, App_r$ 中的每个应用算法的性能行为。性能行为可以指示应用算法的性能如何根据由应用算法应用于表 $T_g$ 的改变的大小而改变。为

此,可以使用每个应用算法 $App_1, \dots, App_r$ 将 $N$ 个改变应用于表 $T_g$ 。 $N$ 个改变中的每个改变可具有其自身的大小。例如,大小可以是要插入表 $T_g$ 中的记录的数量和/或要从表 $T_g$ 中删除的记录的数量。对于每个应用的改变,可以针对应用程序 $App_1, \dots, App_r$ 中的每个应用程序评估性能参数。这可得到 $r$ 个数据结构(例如,离散点的曲线 $curv_1, \dots, curv_r$ )。 $r$ 个数据结构中的每一个包括表示与 $m$ 个性能参数的各个测量值相关联的 $N$ 个改变大小 $s_1 \dots s_N$ 的 $N$ 个数据点 $p_1 = (s_1, l_1^1 \dots l_m^1), p_2 = (s_2, l_1^2 \dots l_m^2) \dots p_N = (s_N, l_1^N \dots l_m^N)$ 。例如,如果性能参数的数量是1,则数据点中的每一个可以是包括改变的大小和相应的性能参数值的一对值。性能参数例如可以是执行时间和存储器使用。

[0039] 例如,步骤203可以作为预处理步骤的一部分执行,例如,在数据分析系统的运行时间使用表 $T_g$ 之前离线执行。在另一示例中,步骤203可以在数据分析系统的运行时间执行。改变的数目 $N$ 可被选择得尽可能小,例如 $N=2$ ,因为步骤203可能是资源消耗的,因为它针对改变中的每一个评估每个应用算法。

[0040] 然而,如果改变的数量 $N$ 不足以估计所请求的改变的性能,则可以通过向其添加附加点来进一步扩充所得到的数据结构 $curv_1, \dots, curv_r$ 。附加点的这种添加可以使用例如内插来执行。例如,内插可以是线性或样条内插。因此,数据结构 $curv_1, \dots, curv_r$ 中的每个数据结构可以具有 $N+x$ 个数据点。在另一示例中,数据结构 $curv_1, \dots, curv_r$ 可被处理以便使用数学函数来建模它们的行为。因此,步骤203可以得到分别具有离散点的 $r$ 个数据结构或表示应用算法 $App_1, \dots, App_r$ 的性能行为的 $r$ 个数学函数。

[0041] 可在步骤205中接收数据改变请求以将改变应用于表。例如,响应于检测到源表中的改变,数据改变请求可被发送至数据同步系统,例如,图1的数据同步系统102。

[0042] 响应于接收到数据改变请求,在步骤207中可确定所请求的改变的大小。大小的定义可以与用于确定性能行为的定义相同。例如,所请求的数据改变的大小 $s_t$ 可以是待插入的记录的数量和/或待从表 $T_g$ 中删除的记录的数量。

[0043] 在步骤209中,所确定的大小 $s_t$ 可以用于选择与所确定的大小 $s_t$ 的其他应用算法相比提供最佳性能的应用算法。为此,可以针对所确定的大小 $s_t$ 估计应用算法 $App_1, \dots, App_r$ 中的每个应用算法的性能。可以使用在步骤203中确定的数据结构 $curv_1, \dots, curv_r$ 或数学模型来执行这个估计。例如,在使用具有离散点的数据结构 $curv_1, \dots, curv_r$ 的情况下,可以标识与每个数据结构的确定大小 $s_t$ 最接近的点。这可得到 $r$ 个最近的点。可以比较最近点的性能参数的值,并且可以选择最佳最近点。所选择的最近点可与应用算法中的一个应用算法相关联。应用算法中的该一个应用算法可以是步骤209的所选应用算法。

[0044] 在使用数学模型的情况下,所确定的大小 $s_t$ 可以作为自变量或输入给每个数学模型,以便估计与大小 $s_t$ 相关联的性能。这可得到 $r$ 个性能参数的值的集合。可以将这些相互比较以标识最佳性能和相关联的应用算法。

[0045] 在步骤211中,可以使用所选择的应用算法将所请求的改变应用于表 $T_g$ 。

[0046] 图3是根据本发明实施例的用于向目标数据库系统的表 $T_g$ 应用改变的方法的流程图。为了说明的目的,在图3中描述的方法可以在图1中示出的系统中实现,但不限于该实施方式。例如,图3的方法可由数据同步系统102执行。例如,图3的方法可以使得能够在源数据库系统的源表 $T_s$ (对应于 $T_g$ )中做出的改变应用于目标数据库系统,并且因此可以使得能够在源数据库系统与目标数据库系统之间同步。

[0047] 步骤301至311是图2的步骤201至211。此外,图3的方法还包括更新在步骤303中针对所选择的应用算法确定的性能行为的步骤313。这可以例如通过将接收到的改变相关联的点 $p_t = (s_t, l_1^t \dots l_m^t)$ 添加到表示所选择的应用算法的性能行为的数据结构中来执行。此外,可以针对要应用于表 $T_g$ 的每个数据改变重复步骤305至313。

[0048] 图4是根据本发明实施例的用于向目标数据库系统的多个表应用改变的方法的流程图。为了说明的目的,在图4中描述的方法可以在图1中示出的系统中实现,但不限于该实施方式。例如,图4的方法可由数据同步系统102执行。图4的方法可以例如使得能够在源数据库系统的多个源表(对应于多个目标表)中做出的改变应用至目标数据库系统,并且因此可以使得能够在源数据库系统与目标数据库系统之间同步。

[0049] 可以在步骤401中提供多个应用算法(例如,如在步骤201中所描述的)用于在目标数据库系统中应用改变。

[0050] 可以在步骤403中并且针对多个表中的每个表确定应用算法 $App_1, \dots, App_r$ 中的每个应用算法的性能行为。例如,步骤403可包括:对多个表中的每个表执行图2的步骤203。这可能是有利的,因为它使得应用算法的性能行为也取决于表。

[0051] 在步骤405中,可接收数据改变请求,以分别将一个或多个改变应用于一个或多个表。这些改变可以具有不同的大小,因为它们被应用于不同的表。

[0052] 对于一个或多个表中的每个表,可使用表的性能行为来应用步骤407至411。步骤407至411是图2的步骤207至211。

[0053] 图5是根据本发明实施例的用于向目标数据库系统的多个表应用改变的方法的流程图。为了说明的目的,在图5中描述的方法可以在图1中示出的系统中实现,但不限于该实施方式。例如,图5的方法可由数据同步系统102执行。例如,图5的方法可以使得能够在源数据库系统的多个源表(对应于多个目标表)中做出的改变应用至目标数据库系统,并且因此可以使得能够在源数据库系统与目标数据库系统之间同步。

[0054] 步骤501至511是图4的步骤401至411。此外,图5的方法还包括更新步骤503中针对所选择的应用算法确定的性能行为的步骤513。此外,可以针对要应用于多个表中的一个或多个表的每个改变重复步骤505至513。

[0055] 图6A是根据本发明的实施例的用于确定应用算法的性能行为的方法的流程图。

[0056] 应用算法可以在步骤601中执行预定义数目 $N$ 次,以用于分别将数据改变应用于表,其中每个所应用的数据改变具有大小。

[0057] 对于数据改变中的每个数据改变,可以在步骤603中评估指示应用算法的执行的性能的至少一个性能参数。例如,可以针对 $N$ 个改变中的每个改变的应用来测量应用算法的执行时间。

[0058] 在步骤605中可以提供 $N$ 个数据点的数据结构。数据结构表示应用算法的性能行为。每个数据点指示所评估的性能参数和数据改变的相关联大小。图6C示出了两个应用算法 $App_1$ 和 $App_2$ 的 $N=2$ 个测量数据点的示例。

[0059] 图6B是根据本发明的实施例的用于确定应用算法的性能行为的方法的流程图。

[0060] 图6B的方法包括上述步骤601至605以及步骤607。在步骤607中,可用附加点来扩充所提供的表示应用算法的性能行为的数据结构。这可以例如使用这些数据点之间的内插来执行。例如,这表示在图6C中,其中在每个应用算法的两个数据点之间进行线性插值。附

加点可以是沿着虚线(内插)的任何点。

[0061] 图7表示适于实现根据本发明的实施例的方法步骤的至少一部分的通用计算机化系统700。

[0062] 应当理解,本文描述的方法至少部分是非交互的,并且通过诸如服务器或嵌入式系统的计算机化系统自动化。然而,在示例性实施例中,在此描述的方法可以在(部分)交互系统中实现。这些方法可进一步在软件712、722(包括固件722)、硬件(处理器)705或其组合中实现。在示范性实施例中,本文中描述的方法以软件实现为可执行程序,并且由专用或通用数字计算机(诸如个人计算机、工作站、小型计算机或大型计算机)执行。因此,最通用的系统700包括通用计算机701。

[0063] 在示例性实施方式中,在硬件架构方面,如图7所示,计算机701包括处理器705、耦接至存储器控制器715的存储器(主存储器)710、以及经由本地输入/输出控制器735通信地耦接的一个或多个输入和/或输出(I/O)设备(或外围设备)10、745。输入/输出控制器735可以是但不限于如本领域中已知的一个或多个总线或其他有线或无线连接。输入/输出控制器735可具有为简单起见而省略的额外元件,例如控制器、缓冲器(高速缓冲存储器)、驱动器、中继器和接收器,以实现通信。进一步,本地接口可以包括地址、控制和/或数据连接,以实现上述组件之间的适当通信。如本文所描述的,I/O设备10、745一般可以包括本领域已知的任何通用密码卡或智能卡。

[0064] 处理器705是用于执行软件(特别是存储在存储器710中的软件)的硬件装置。处理器705可以是与计算机701相关联的若干处理器中的任何定制或商用处理器、中央处理单元(CPU)、辅助处理器、基于半导体的微处理器(呈微芯片或芯片组的形式)、或通常用于执行软件指令的任何装置。

[0065] 存储器710可以包括易失性存储器元件(例如,随机存取存储器(RAM,诸如DRAM、SRAM、SDRAM等))和非易失性存储器元件(例如,ROM、可擦可编程只读存储器(EPROM)、电可擦可编程只读存储器(EEPROM)、可编程只读存储器(PROM))中的任何一个或其组合。注意,存储器710可以具有分布式架构,其中不同部件彼此远离定位,但是可以由处理器705访问。

[0066] 存储器710中的软件可以包括一个或多个单独的程序,每个程序包括用于实现逻辑功能(尤其是在本发明的实施例中涉及的功能)的可执行指令的有序列表。在图7的示例中,存储器710中的软件包括指令712,例如,管理数据库(诸如数据库管理系统)的指令。

[0067] 存储器710中的软件通常还应包括合适的操作系统(OS)711。OS711实质上控制其他计算机程序的执行,诸如,用于实现此处所描述的方法的可能的软件712。

[0068] 本文描述的方法可以是源程序712、可执行程序712(目标代码)、脚本或包括要执行的指令712的集合的任何其他实体的形式。当源程序时,则需要经由编译器、汇编器、解释器等翻译程序,该编译器、汇编器、解释器等可以包括在或者可以不包括在存储器710内,以便结合OS 711适当地操作。此外,该方法可以被写为具有数据和方法类的面向对象的编程语言,或者具有例程、子例程和/或函数的过程编程语言。

[0069] 在示范性实施例中,常规键盘750和鼠标755可耦合到输入/输出控制器735。其他输出设备(诸如I/O设备745)可以包括输入设备,例如但不限于打印机、扫描仪、麦克风等。最后,I/O设备10、745可以还包括传送输入和输出两者的设备,例如但不限于,网络接口卡(NIC)或调制器/解调器(用于访问其他文件、设备、系统、或网络)、射频(RF)或其他收发器、

电话接口、网桥、路由器等。I/O设备10、745可以是本领域中已知的任何通用密码卡或智能卡。系统700可进一步包含耦合到显示器730的显示器控制器725。在示例性实施例中，系统700可以还包括用于耦合到网络765的网络接口。网络765可以是用于计算机701和任何外部服务器、客户端等之间经由宽带连接的通信的基于IP的网络。网络765在计算机701与外部系统30之间传输和接收数据，该数据可以被涉及来执行在此所讨论的这些方法的部分或全部步骤。在示例性实施例中，网络765可以是由服务提供商管理的受管理的IP网络。网络765可以以无线方式实现，例如使用无线协议和技术，例如WiFi、WiMax等。网络765也可以是分组交换网络，例如局域网、广域网、城域网、互联网网络或其他类似类型的网络环境。网络765可以是固定无线网络、无线局域网W(LAN)、无线广域网(WWAN)、个人区域网(PAN)、虚拟专用网(VPN)、内联网或其他合适的网络系统，并且包括用于接收和传送信号的设备。

[0070] 如果计算机701是PC、工作站、智能设备等，则存储器710中的软件还可以包括基本输入输出系统(BIOS)722。BIOS是在启动时初始化和测试硬件、启动OS711、并且支持在硬件设备之间的数据传输的基本软件例程的集合。BIOS存储在ROM中，使得当计算机701被激活时可以执行BIOS。

[0071] 当计算机701运行时，处理器705被配置为执行存储在存储器710内的软件712，以向和从存储器710传送数据，并且总体上根据该软件来控制计算机701的操作。本文所描述的方法和OS 711全部或部分(但通常是后者)被处理器705读取，可能在处理器705内被缓冲，然后被执行。

[0072] 当在本文中描述的系统和方法在软件712中实现时，如图7中所示，该方法可以存储在任何计算机可读介质(例如，存储器720)上，用于由任何计算机相关的系统或方法使用或者与任何计算机相关的系统或方法结合。存储器720可包括磁盘存储器，例如HDD存储器。

[0073] 在此描述的程序基于在本发明的具体实施例中对其实施的应用来标识。然而，应当理解，本文中的任何特定程序术语仅为了方便而使用，并且因此本发明不应局限于仅由这样的术语标识和/或暗示的任何特定应用中使用。

[0074] 本发明可以是系统、方法和/或计算机程序产品。计算机程序产品可包括其上具有用于使处理器执行本发明的各方面的计算机可读程序指令的计算机可读存储介质(或多个介质)。

[0075] 计算机可读存储媒体可为可保留和存储供指令执行装置使用的指令的有形装置。计算机可读存储介质可以是，例如但不限于，电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备、或者上述的任意合适的组合。计算机可读存储介质的更具体实例的非穷尽列表包括以下各项：便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EEPROM或闪存)、静态随机存取存储器(SRAM)、便携式紧凑盘只读存储器(CD-ROM)、数字通用盘(DVD)、记忆棒、软盘、诸如穿孔卡之类的机械编码设备或具有记录在其上的指令的槽中的凸出结构、以及上述各项的任何合适的组合。如本文所使用的计算机可读存储媒体不应被解释为暂时性信号本身，例如无线电波或其他自由传播的电磁波、通过波导或其他传输媒体传播的电磁波(例如，穿过光纤电缆的光脉冲)或通过电线发射的电信号。

[0076] 本文中所述的计算机可读程序指令可以经由网络(例如，互联网、局域网、广域网和/或无线网络)从计算机可读存储介质下载到相应的计算/处理设备，或者下载到外部

计算机或外部存储设备。网络可以包括铜传输电缆、光传输纤维、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配器卡或网络接口接收来自网络的可读程序指令,并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0077] 用于执行本发明的操作的计算机可读程序指令可以是汇编指令、指令集架构 (ISA) 指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或以一种或多种程序设计语言的任何组合编写的源代码或目标代码,这些程序设计语言包括面向对象的程序设计语言 (诸如 Smalltalk、C++ 等)、以及常规的过程式程序设计语言 (诸如“C”程序设计语言或类似程序设计语言)。计算机可读程序指令可以完全地在用户计算机上执行、部分在用户计算机上执行、作为独立软件包执行、部分在用户计算机上部分在远程计算机上执行或者完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可通过任何类型的网络 (包括局域网 (LAN) 或广域网 (WAN)) 连接至用户计算机,或者可连接至外部计算机 (例如,使用互联网服务提供商通过互联网)。在一些实施例中,包括例如可编程逻辑电路、现场可编程门阵列 (FPGA) 或可编程逻辑阵列 (PLA) 的电子电路可以通过利用计算机可读程序指令的状态信息来使电子电路个性化来执行计算机可读程序指令,以便执行本发明的各方面。

[0078] 下面将参照根据本发明实施例的方法、装置 (系统) 和计算机程序产品的流程图和/或框图描述本发明。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0079] 这些计算机可读程序指令可被提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器以产生机器,使得经由计算机或其他可编程数据处理装置的处理器执行的指令创建用于实现在流程图和/或框图的或多个框中指定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置、和/或其他设备以特定方式工作,从而,其中存储有指令的计算机可读存储介质包括包含实现流程图和/或框图中的或多个框中规定的功能/动作的方面的指令的制品。

[0080] 也可以把计算机可读程序指令加载到计算机、其他可编程数据处理装置、或其他设备上,使得在计算机、其他可编程装置或其他设备上执行一系列操作步骤,以产生计算机实现的处理,使得在计算机、其他可编程装置或其他设备上执行的指令实现流程图和/或框图中的或多个框中规定的功能/动作。

[0081] 附图中的流程图和框图示出了根据本发明的不同实施例的系统、方法和计算机程序产品的可能实现方式的架构、功能和操作。对此,流程图或框图中的每个框可表示指令的模块、段或部分,其包括用于实现指定的逻辑功能的一个或多个可执行指令。在一些替代实施例中,框中所指出的功能可不按图中所指出的次序发生。例如,取决于所涉及的功能,连续示出的两个块实际上可以基本上同时执行,或者这些块有时可以以相反的顺序执行。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作或执行专用硬件与计算机指令的组合作用的专用的基于硬件的系统来实现。

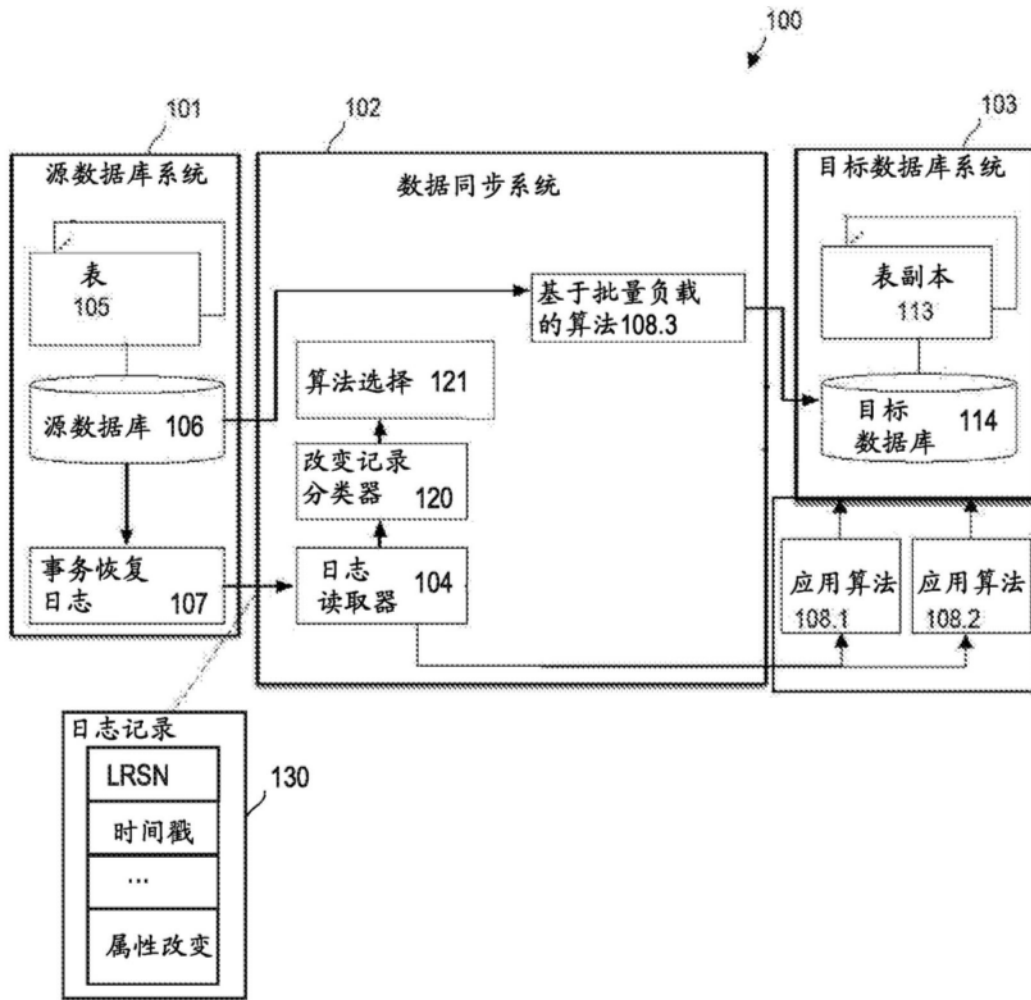


图1



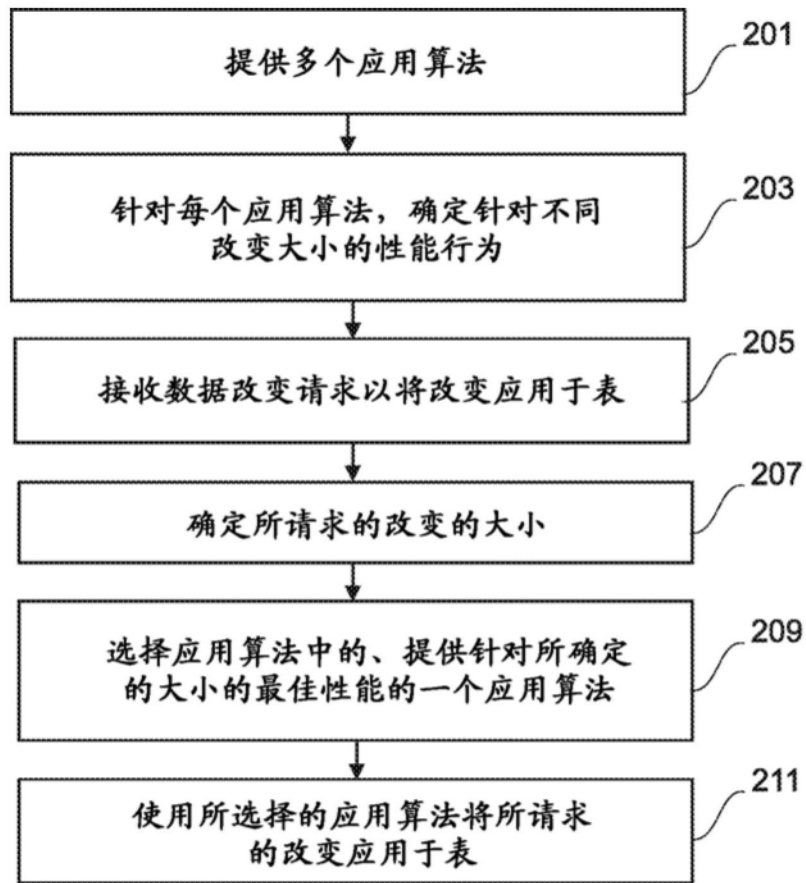


图2

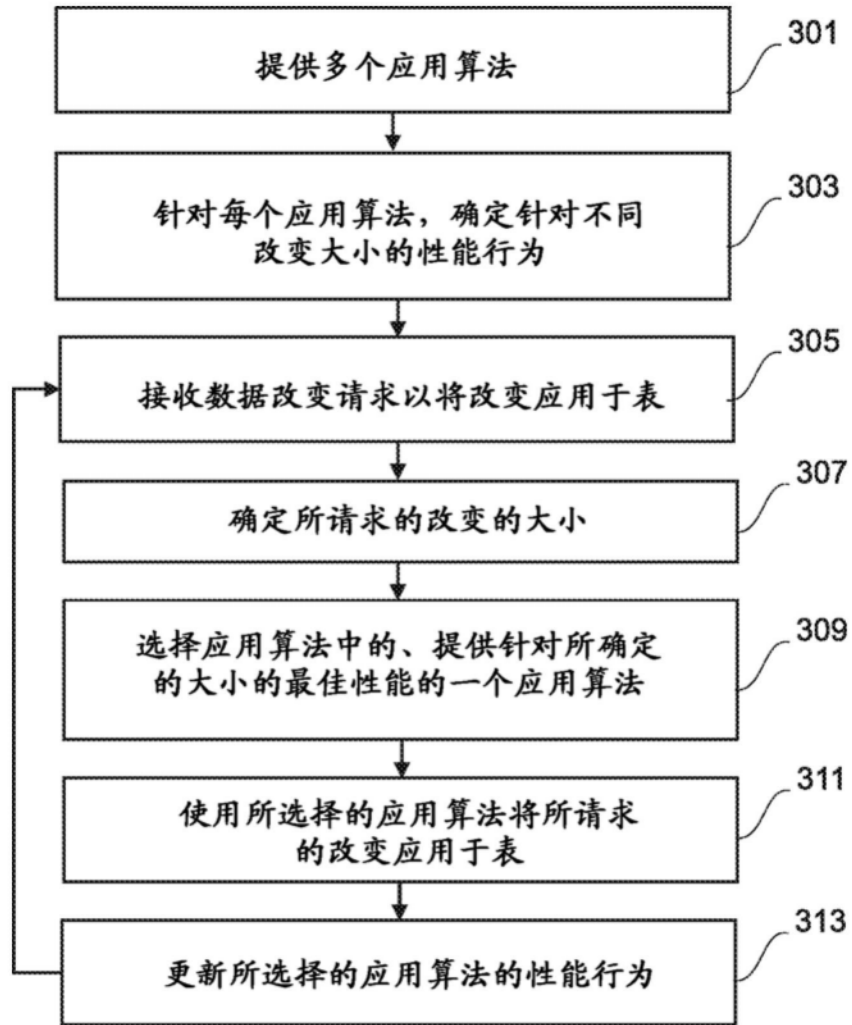


图3

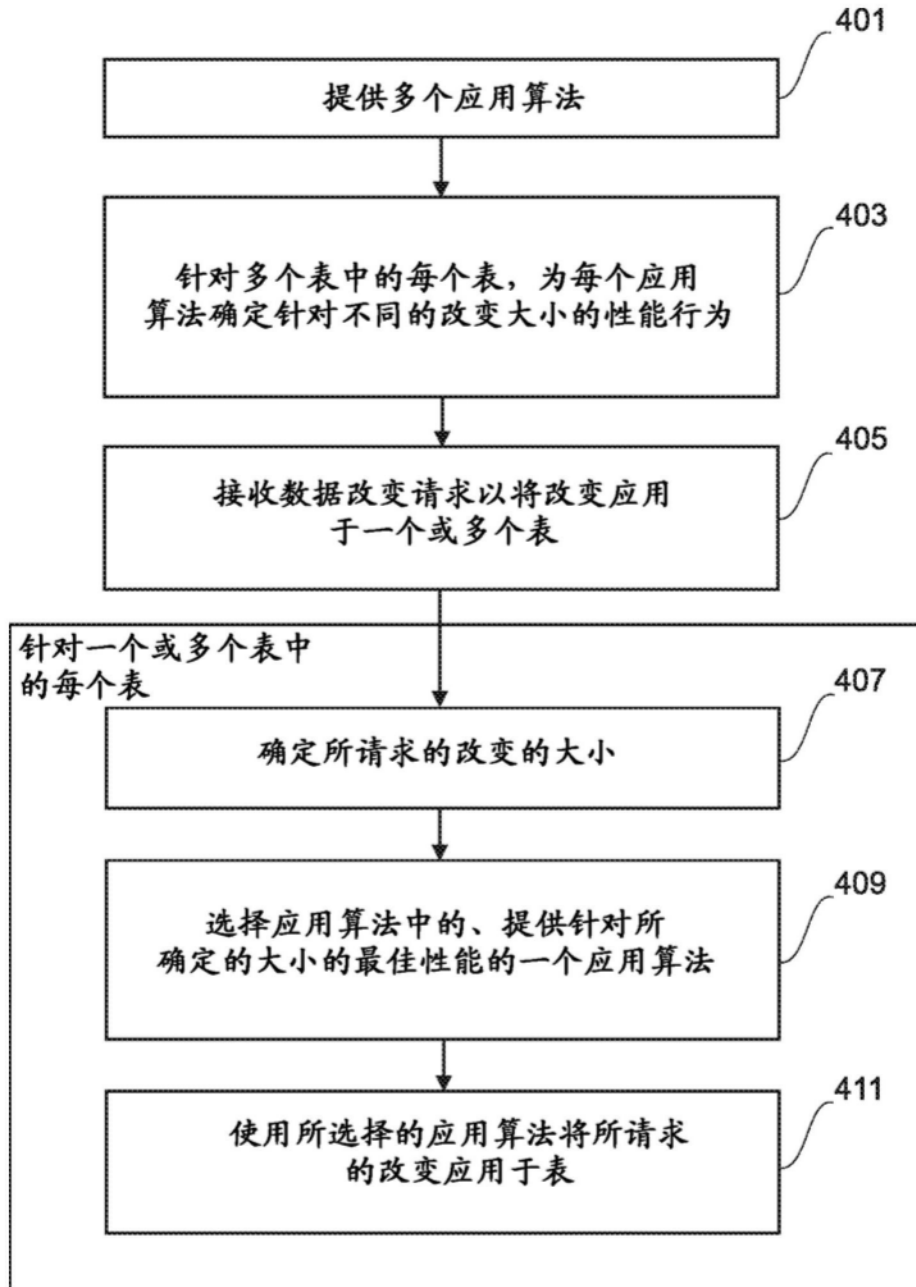


图4

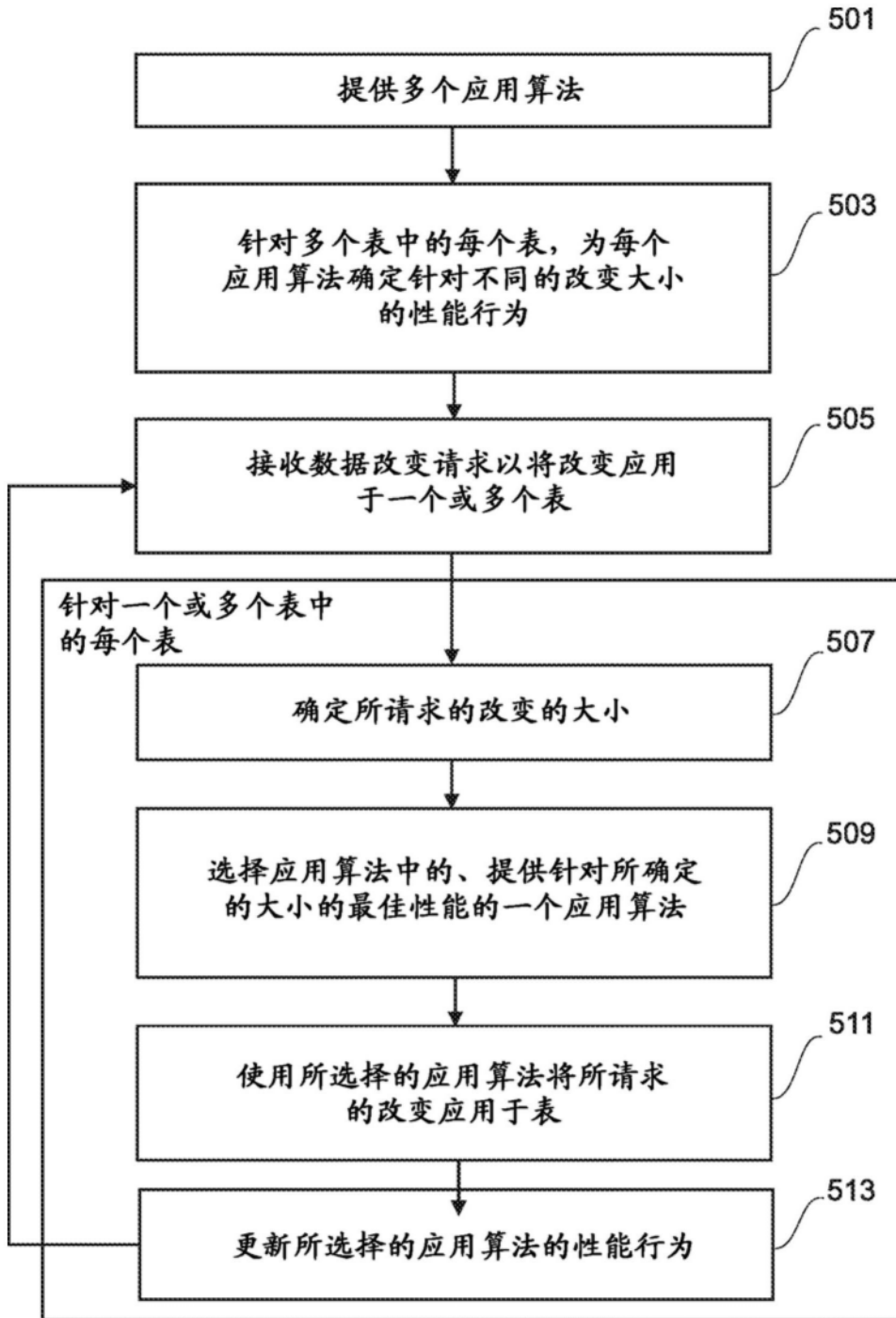


图5

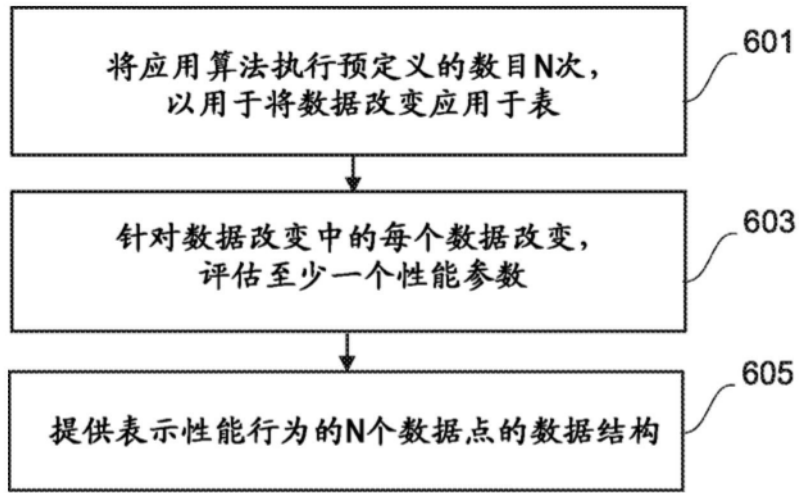


图6A

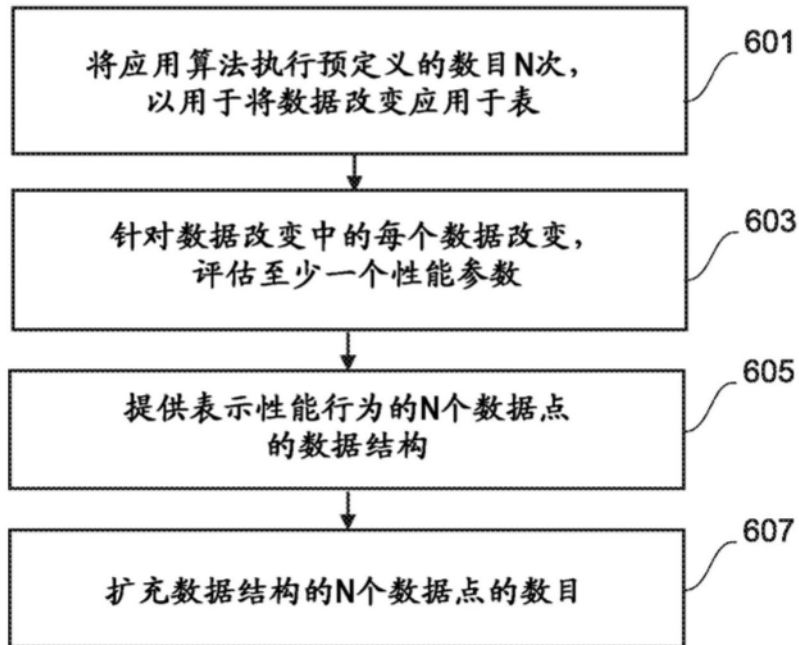


图6B

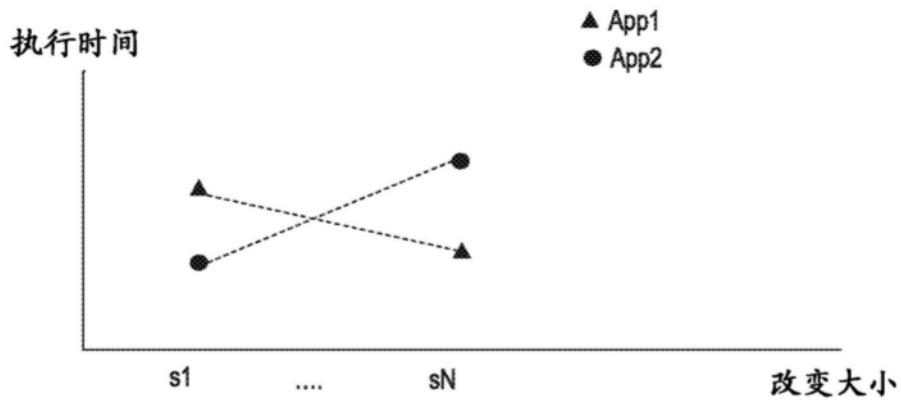


图6C

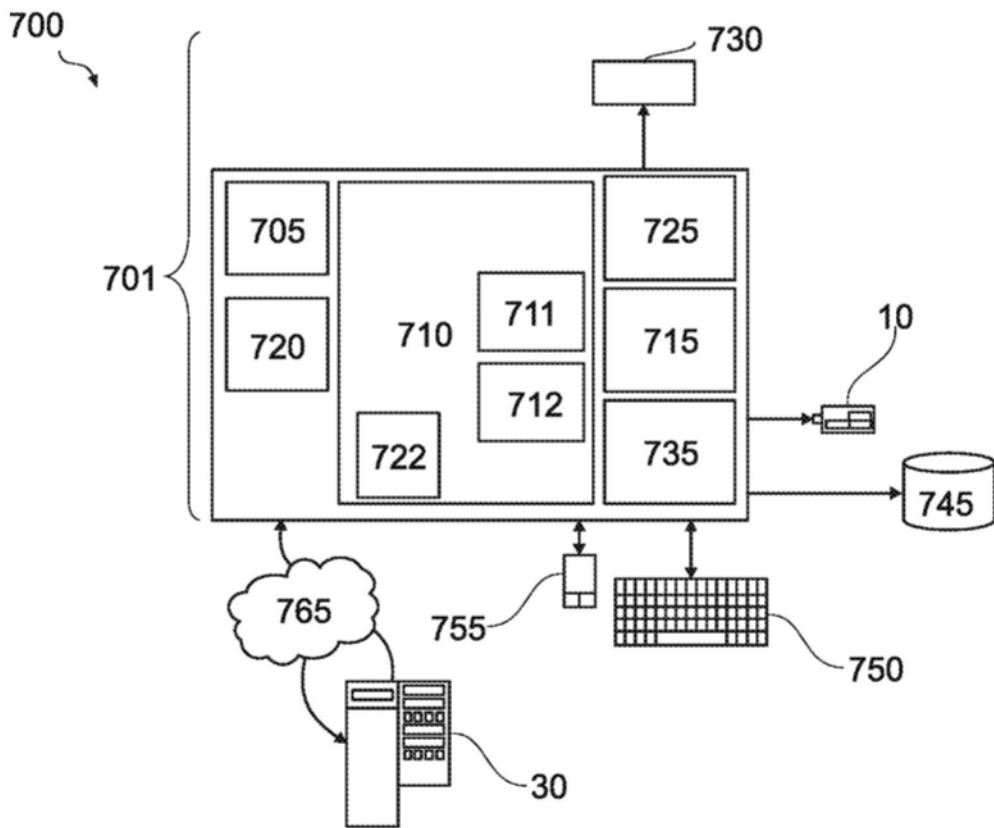


图7