

(12) 发明专利

(10) 授权公告号 CN 101567006 B

(45) 授权公告日 2012. 07. 04

(21) 申请号 200910107773. 9

CN 101272523 A, 2008. 09. 24, 全文.

(22) 申请日 2009. 05. 25

CN 101136029 A, 2008. 03. 05, 全文.

(73) 专利权人 中兴通讯股份有限公司

审查员 李宁

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦A座6层

(72) 发明人 丁奇鹏

(74) 专利代理机构 深圳鼎合诚知识产权代理有限公司 44281

代理人 宋鹰武

(51) Int. Cl.

G06F 17/30 (2006. 01)

(56) 对比文件

JP 特开 2008-146119 A, 2008. 06. 26, 全文.

US 5511190 A, 1996. 04. 23, 全文.

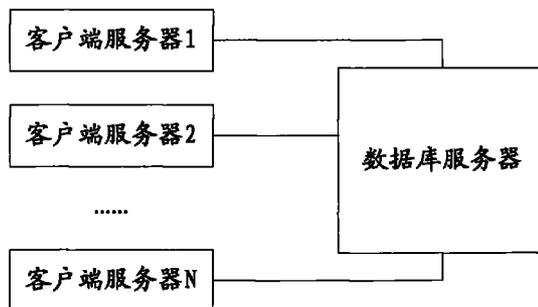
权利要求书 2 页 说明书 5 页 附图 3 页

(54) 发明名称

一种数据库系统及分布式 SQL 语句执行计划重用方法

(57) 摘要

本发明涉及一种数据库系统及分布式 SQL 语句执行计划重用方法,所述系统包括客户端服务器和数据库服务器;所述方法为:客户端接收到用户的 SQL 语句后,对其进行 hash 运算,根据得到的 hash 表下标值在本地 hash 表中查找所述 SQL 语句,找到后读取所述 SQL 语句在服务端执行计划缓存数组中的下标值,将所述 SQL 语句和所述下标值发送给服务端;服务端接收到客户端的请求包后,解析所述请求包,根据得到的 SQL 语句和执行计划缓存数组下标值,找到相应的执行计划并使用所述执行计划执行,执行结束后将结果返回给所述客户端。本发明减轻了数据库服务器的负担,也提高了数据库系统的计算能力使用率。



1. 一种数据库系统,包括客户端服务器和数据库服务器,其特征在于,所述

客户端服务器用于在启动时创建 hash 表,用于对用户发送的结构化查询语言 SQL 语句进行 hash 运算,以及根据得到的 hash 表下标值从所述 hash 表中查找相应 SQL 语句在数据库服务端的执行计划缓存数组的下标值,并在查找到时,读取所述下标值,并将所述 SQL 语句与所述下标值发送给所述数据库服务器;在未查找到时将仅所述 SQL 语句发送给所述数据库服务器;

数据库服务器用于在启动时创建执行计划缓存数组,用于解析并判断所述客户端服务器发送来的请求包中是否携带有合法有效的下标值,并在解析结果中存在合法有效的下标值时,通过所述下标值查找相应的执行计划并使用所述执行计划执行,将执行结果返回给所述客户端服务器;在解析结果中不存在下标值或者下标值非法/失效时,将所述请求包携带的 SQL 语句转换为数据库内部可执行数据结构和代码,得到执行计划,并在所述执行计划缓存数组中为其分配空闲节点,将转换得到的执行计划和所述 SQL 语句存入所述节点,记录所述节点的下标值,使用得到的执行计划执行,将执行结果和所述记录的下标值返回给所述客户端服务器;

所述客户端服务器还用于解析所述数据库服务器返回的响应包,将执行结果返回给用户,用于判断所述响应包中是否携带有下标值;在响应包携带有下标值时,判断本地 hash 表中是否存在当前 SQL 语句,若存在,则将其对应的下标值更新成当前解析得到的下标值;若不存在,则将当前 SQL 语句和解析得到的下标值存入本地 hash 表中。

2. 一种分布式 SQL 语句执行计划重用方法,其特征在于,

客户端在启动时创建 hash 表;

所述客户端接收到用户的 SQL 语句后,对其进行 hash 运算,根据得到的 hash 表下标值在本地 hash 表中查找所述 SQL 语句,找到后读取所述 SQL 语句在数据库服务端执行计划缓存数组中的下标值,将所述 SQL 语句和所述下标值发送给数据库服务端;

若所述客户端在本地 hash 表中没有查找到所述 SQL 语句,则仅将所述 SQL 语句发送给所述数据库服务端;

所述客户端接收到数据库服务端返回的响应包后,执行如下步骤:

步骤 A:解析并判断所述响应包中是否携带有下标值,若是,则执行步骤 B;否则,执行步骤 C;

步骤 B:根据 hash 表下标值查找本地 hash 表中是否存在当前 SQL 语句,若是,则将其对应的执行计划缓存数组下标值更新成解析得到的下标值;否则,根据所述 hash 表下标值将当前 SQL 语句和解析得到的下标值存入本地 hash 表中;

步骤 C:将执行结果返回给用户;

所述数据库服务端在启动时创建执行计划缓存数组;

所述数据库服务端接收到客户端的请求包后,解析所述请求包,根据得到的 SQL 语句和执行计划缓存数组下标值,找到相应的执行计划并使用所述执行计划执行,执行结束后将结果返回给所述客户端;

若所述数据库服务端从接收到的请求包中没有携带下标值或者携带的下标值非法/失效,则执行如下操作:

步骤 a:将所述 SQL 语句转换为数据库内部可执行数据结构和代码,得到执行计划;

步骤 b :在执行计划缓存数组中为所述 SQL 语句分配一个空闲节点,将所述 SQL 语句及其执行计划存入所述节点,并记录所述节点的下标值;

步骤 c :使用所述执行计划执行,得到执行结果;

步骤 d :将执行结果和记录的下标值返回给所述客户端。

3. 如权利要求 2 所述的分布式 SQL 语句执行计划重用方法,其特征在于,所述 hash 表的记录数大于等于本客户端上可执行的 SQL 语句种类数,且每条记录都具有一个冲突链表;

所述执行计划缓存数组的节点数大于等于数据库系统的 SQL 语句种类数,且每个节点都具有一个指针。

4. 如权利要求 3 所述的分布式 SQL 语句执行计划重用方法,其特征在于,所述客户端将 hash 表下标值相同的 SQL 语句及其执行计划缓存数组下标值存储在所述 hash 表下标值对应记录的冲突链表的不同节点中。

5. 如权利要求 2 或 3 所述的分布式 SQL 语句执行计划重用方法,其特征在于,所述数据库服务端转换得到 SQL 语句的执行计划后,将所述 SQL 语句及其执行计划存入预先申请的结构体中,并使得为该 SQL 语句分配的执行计划缓存数组的节点的指针指向所述结构体;

所述客户端通过遍历所述 hash 表记录的冲突链表来查找其中是否存在当前 SQL 语句。

一种数据库系统及分布式 SQL 语句执行计划重用方法

技术领域

[0001] 本发明涉及数据库系统领域,特别是涉及了一种数据库系统及分布式结构化查询语言 (Structured Query Language, SQL) 语句执行计划重用方法。

背景技术

[0002] 在数据库界,最为普遍接受的语法标准是 SQL92,即 ansix3.135-1992 中“数据库语言 SQL”一节所规定的语法标准,该标准规定了数据库用户操作数据库服务器的接口标准,它是一种纯文本接口,这就意味着:在任何支持 SQL92 标准的数据库中,一定会存在将客户端发来的 SQL 语句文本串转换为内部可执行数据结构和代码的过程,这种转换后的内部数据结构和代码在数据库界通称为“执行计划”,这种转换过程通称为“硬解析”。

[0003] 由于 SQL 语法本身的可扩展、可连接、可嵌套等特性,一个 SQL 语句可能复杂到匪夷所思的程度,SQL 语法解析器要做到通用和兼容,必须各种语句都要支持,所以硬解析是极其复杂和耗时的过程,若每个 SQL 语句执行时都硬解析一次,那么数据库的性能必定是极其低下的。因此,为了提高性能,任何商用数据库都会有执行计划重用功能,即相同语句仅在第一次执行时进行硬解析,解析完成后 SQL 语句及其执行计划被缓存下来,再次执行相同语句时,通过对 SQL 语句进行 hash 运算匹配到上次的执行计划,就可以重用执行计划,省去了大量的硬解析操作。

[0004] 可以看出,对 SQL 语句进行 hash 运算的过程十分关键,该运算的速度和结果离散度直接决定了执行计划重用的速度。如图 1 所示,在传统的数据库中,SQL 语句匹配的过程都是在数据库服务端进行的,客户端仅负责发送语句和接收结果;而服务端不仅需要对所有接收到的 SQL 语句进行 hash 运算,并且还要对第一次执行的 SQL 语句进行硬解析并存储,当服务端连接的客户端非常多、执行语句非常频繁时,hash 匹配运算对服务端也将产生巨大的压力,离散度越高的 hash 算法会导致服务端的中央处理器 (CPU) 负荷越高,这个问题在内存数据库中体现得尤其明显,因为内存数据库几乎没有 I/O 的负担,所以瓶颈一定是体现在 CPU 密集型操作上,而 hash 运算就是一种典型的 CPU 密集型操作。

[0005] 这种完全集中于服务端的 SQL 匹配模式有以下弊端:

[0006] 1、所有 hash 运算压力完全放在服务端,客户端除了发送语句和接收结果外什么都不做,造成等待结果期间客户端计算能力的闲置和浪费;

[0007] 2、全部在服务端进行 CPU 密集型的 hash 运算将造成服务端的性能瓶颈,不利于客户端数量的扩展,也影响了后续其余处理步骤的效率。

发明内容

[0008] 本发明所要解决的技术问题是,提供一种数据库系统及分布式 SQL 语句执行计划重用方法,本发明减轻了数据库服务器的负担,也提高了数据库系统的计算能力使用率。

[0009] 本发明公开了一种数据库系统,包括客户端服务器和数据库服务器,所述

[0010] 客户端服务器用于在启动时创建 hash 表,用于对用户发送的 SQL 语句进行 hash

运算,以及根据得到的 hash 表下标值从所述 hash 表中查找相应 SQL 语句在服务端的执行计划缓存数组的下标值,并在查找到时,读取所述下标值,并将所述 SQL 语句与所述下标值发送给所述数据库服务器;在未查找到时将仅所述 SQL 语句发送给所述数据库服务器;

[0011] 数据库服务器用于在启动时创建执行计划缓存数组,用于解析并判断所述客户端服务器发送来的请求包中是否携带有合法有效的下标值,并在解析结果中存在合法有效的下标值时,通过所述下标值查找相应的执行计划并使用所述执行计划执行,将执行结果返回给所述客户端服务器;在解析结果中不存在下标值或者下标值非法/失效时,硬解析所述请求包携带的 SQL 语句,并在所述执行计划缓存数组中为其分配空闲节点,将硬解析得到的执行计划和所述 SQL 语句存入所述节点,记录所述节点的下标值,使用得到的执行计划执行,将执行结果和所述记录的下标值返回给所述客户端服务器。

[0012] 所述客户端服务器还用于解析所述数据库服务器返回的响应包,将执行结果返回给用户,用于判断所述响应包中是否携带有下标值;在响应包携带有下标值时,判断本地 hash 表中是否存在当前 SQL 语句,若存在,则将其对应的下标值更新成当前解析得到的下标值;若不存在,则将当前 SQL 语句和解析得到的下标值存入本地 hash 表中。

[0013] 本发明还在所述数据库系统的基础上公开了一种分布式 SQL 语句执行计划重用方法,所述方法包括:

[0014] 客户端接收到用户的 SQL 语句后,对其进行 hash 运算,根据得到的 hash 表下标值在本地 hash 表中查找所述 SQL 语句,找到后读取所述 SQL 语句在服务端执行计划缓存数组中的下标值,将所述 SQL 语句和所述下标值发送给服务端;

[0015] 服务端接收到客户端的请求包后,解析所述请求包,根据得到的 SQL 语句和执行计划缓存数组下标值,找到相应的执行计划并使用所述执行计划执行,执行结束后将结果返回给所述客户端。

[0016] 若所述客户端在本地 hash 表中没有查找到所述 SQL 语句,则仅将所述 SQL 语句发送给所述服务端。

[0017] 若所述服务端从接收到的请求包中没有携带下标值或者携带的下标值非法/失效,则执行如下操作:

[0018] 步骤 a:硬解析所述 SQL 语句,得到执行计划;

[0019] 步骤 b:在执行计划缓存数组中为所述 SQL 语句分配一个空闲节点,将所述 SQL 语句及其执行计划存入所述节点,并记录所述节点的下标值;

[0020] 步骤 c:使用所述执行计划执行,得到执行结果;

[0021] 步骤 d:将执行结果和记录的下标值返回给所述客户端。

[0022] 所述客户端接收到服务端返回的响应包后,执行如下步骤:

[0023] 步骤 A:解析并判断所述响应包中是否携带有下标值,若是,则执行步骤 B;否则,执行步骤 C;

[0024] 步骤 B:根据 hash 表下标值查找本地 hash 表中是否存在当前 SQL 语句,若是,则将其对应的执行计划缓存数组下标值更新成解析得到的下标值;否则,根据所述 hash 表下标值将当前 SQL 语句和解析得到的下标值存入本地 hash 表中;

[0025] 步骤 C:将执行结果返回给用户。

[0026] 所述方法还包括:

- [0027] 所述客户端在启动时创建 hash 表；
- [0028] 所述服务端在启动时创建执行计划缓存数组。
- [0029] 所述 hash 表的记录数大于等于本客户端上可执行的 SQL 语句种类数，且每条记录都具有一个冲突链表；
- [0030] 所述执行计划缓存数组的节点数大于等于数据库系统的 SQL 语句种类数，且每个节点都具有指针。
- [0031] 所述客户端将 hash 表下标值相同的 SQL 语句及其执行计划缓存数组下标值存储在所述 hash 表下标值对应记录的冲突链表的不同节点中。
- [0032] 所述服务端硬解析得到 SQL 语句的执行计划后，将所述 SQL 语句及其执行计划存入预先申请的结构体中，并使得为该 SQL 语句分配的执行计划缓存数组的节点的指针指向所述结构体；
- [0033] 所述客户端通过遍历所述 hash 表记录的冲突链表来查找其中是否存在当前 SQL 语句。
- [0034] 本发明将对 SQL 语句进行 hash 运算的过程分散到所有客户端，从而将运算负荷平均分布到整个数据库系统中所有成员机器，不仅大大减轻了数据库服务器的负担，极大地提高了数据库服务器的处理能力，也提高了整个数据库系统的计算能力使用率。

附图说明

- [0035] 图 1 是传统的执行计划重用方法示意图；
- [0036] 图 2 是本发明所述系统示意图；
- [0037] 图 3 是本发明所述方法总体流程示意图；
- [0038] 图 4 是本发明所述方法实施例一流程示意图。

具体实施方式

- [0039] 以下结合附图和优选实施例，对本发明做进一步详细说明。
- [0040] 如图 2 所示，是本发明所述数据库系统示意图，包括客户端服务器和数据库服务器，客户端服务器与数据库服务器直接以 TCP/IP 协议通信，客户端服务器与数据库服务器之间的关系可以是一对多、多对一或者多对多，本实施例采用多对一模式，即包括 N 个客户端服务器和一个数据库服务器的模式，其中
- [0041] 客户端服务器用于在启动时创建记录数大于等于本客户端上可执行的 SQL 语句种类数的 hash 表；用于对用户发送的 SQL 语句进行 hash 运算，以及根据得到的 hash 表下标值从 hash 表中查找相应 SQL 语句在服务端的执行计划缓存数组的下标值，并在查找到时，读取所述下标值，并将所述 SQL 语句与所述下标值发送给数据库服务器；在未查找到时将仅所述 SQL 语句发送给数据库服务器；还用于解析数据库服务器返回的响应包，将执行结果返回给用户，用于判断响应包中是否携带有下标值；在响应包携带有下标值时，判断本地 hash 表中是否存在当前 SQL 语句，若存在，则将其对应的下标值更新成当前解析得到的下标值；若不存在，则将当前 SQL 语句和解析得到的下标值存入本地 hash 表中。
- [0042] 数据库服务器用于在启动时创建节点数大于等于数据库系统的 SQL 语句种类数的执行计划缓存数组，用于解析并判断所述客户端服务器发送来的请求包中是否携带有合

法有效的下标值,并在解析结果中存在合法有效的下标值时,通过所述下标值查找相应的执行计划并使用所述执行计划执行,将执行结果返回给客户端服务器;在解析结果中不存在下标值或者下标值非法/失效时,硬解析所述请求包携带的 SQL 语句,并在所述执行计划缓存数组中为其分配空闲节点,将硬解析得到的执行计划和所述 SQL 语句存入所述节点,记录所述节点的下标值,使用得到的执行计划执行,将执行结果和所述记录的下标值返回给客户端服务器。

[0043] 如图 3 所示,是本发明优选实施例中 hash 表和 execution plan 缓存数组的结构示意图,本实施例中,

[0044] 客户端启动时创建的 hash 表的记录数为 M,每条记录都具有一个冲突链表,本实施例中,假设记录 M-1 的冲突链表中存储有两个 SQL 语句及其下标值,其余的都为空,客户端将 hash 运算得到的 hash 表下标值相同的 SQL 语句及其在数据库服务器的 execution plan 缓存数组的下标值存储在运算得到的 hash 表下标值记录的冲突链表的不同节点中;当需要查找时,则根据所述运算得到的 hash 表下标值定位到相应的记录,然后再遍历该记录的冲突链表即可。

[0045] 数据库服务器启动时创建的 execution plan 缓存数组的节点数为 N,每个节点都具有 void* 指针,数据库服务器硬解析 SQL 语句后,将得到的执行计划和 SQL 语句先存入一个预先申请的结构体中,然后使得为该 SQL 语句分配的数组节点的指针指向所述结构体。

[0046] 如图 4 所示,是本发明所述方法的实施例流程图,本实施例中,假设客户端和服务端都已启动,即客户端已创建了一个通常意义上的 hash 表(hash 表在计算机业界有通用的定义和现成的代码,本发明中不再详细叙述),表名为 hash_tab,hash_tab 的最大记录数为 n;hash_tab 的每条记录都含有一个冲突链表,hash 表下标值相同的 SQL 语句及其 execution plan 缓存数组下标值被存放在此链表的一个节点中;服务端也已创建了一个长度为 max_saved_plans 的数组,数组名为 array_server;具体包括如下步骤:

[0047] 步骤 401:客户端收到一个 SQL 语句,设长度为 SQL_len;

[0048] 步骤 402:对所述 SQL 语句进行 hash 运算,得到 hash 值 hash_val,用 hash_val 对 n 取模,得到 hash_tab 下标值 hash_tab_foot,并定位到 hash_tab 的 hash_tab_foot 的冲突链表头节点 head_node;

[0049] 这里的 hash 算法的应适用于长度不定的字符串处理并且适应于本机 CPU 处理能力。

[0050] 步骤 403:从 head_node 开始遍历冲突链表,将本次待执行的 SQL 语句和存储在每一个节点的 SQL 语句进行比较;

[0051] 步骤 404:判断是否找到,若找到,则执行步骤 405;否则,执行步骤 406;

[0052] 本步骤中,若该 SQL 语句是第一次执行,则找不到,否则,可以找到;

[0053] 步骤 405:取出其中存储的下标值 achieved_node_foot,将 SQL 语句和 achieved_node_foot 打包发送给服务端,并等待服务端返回响应包;

[0054] 步骤 406:将 SQL 语句打包发送给服务端,并等待服务端返回响应包;

[0055] 步骤 407:服务端收到客户端的请求包;

[0056] 步骤 408:解析并判断是否携带了下标值,若是,则执行步骤 409;否则,执行步骤 410;

[0057] 步骤 409 :判断解析得到的下标值 `achieved_node_foot` 是否合法有效,若是,则执行步骤 412 ;否则,执行步骤 410 ;

[0058] 步骤 410 :对解析得到的 SQL 语句进行硬解析,得到执行计划 `exe_plan` ;

[0059] 硬解析的算法可以采用通用的 `flex` 和 `yyac` 配合的方式实现。

[0060] 步骤 411 :从 `array_server` 中分配一个空闲节点,设为 `achieved_node`,其下标值为 `achieved_node_foot` ;将所述 SQL 语句和得到的执行计划存入预先申请的结构体中,并使得 `achieved_node` 执行该结构体,记录 `achieved_node` 下标值 `achieved_node_foot` ;

[0061] 步骤 412 :使用 `exe_plan` 执行,得到执行结果 ;

[0062] 步骤 413 :将执行结果和 `achieved_node_foot` 打包返回给客户端 ;

[0063] 步骤 414 :根据下标值 `achieved_node_foot` 找到 `array_server` 的 `achieved_node_foot` 节点,读取对应的执行计划 `exe_plan` ;

[0064] 步骤 415 :使用读取的 `exe_plan` 执行,得到执行结果 ;

[0065] 步骤 416 :将执行结果返回给客户端 ;

[0066] 步骤 417 :客户端接收到服务端的响应包 ;

[0067] 步骤 418 :解析并判断响应包中是否携带有下标值,若是,则执行步骤 419 ;否则,执行步骤 420 ;

[0068] 步骤 419 :判断 `hash_tab` 的 `hash_tab_foot` 的冲突链表中是否存在当前 SQL 语句,若是,则执行步骤 420 ;否则,执行步骤 421 ;

[0069] 本步骤与步骤 403 采用的方法相同,都是通过遍历所述冲突链表,用当前 SQL 语句和存储在所述冲突链表的每一个节点的 SQL 语句进行比较,若找到相同的,则存在 ;否则,不存在。

[0070] 步骤 420 :将该 SQL 语句对应的下标值更新成解析出的下标值 `achieved_node_foot`,执行步骤 422 ;

[0071] 步骤 421 :将当前 SQL 语句和 `achieved_node_foot` 存入 `hash_tab` 的 `hash_tab_foot` 记录的冲突链表中 ;

[0072] 步骤 422 :将解析出的执行结果返回给用户。

[0073] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改,等同替换,改进等,均应包含在本发明的保护范围之内。

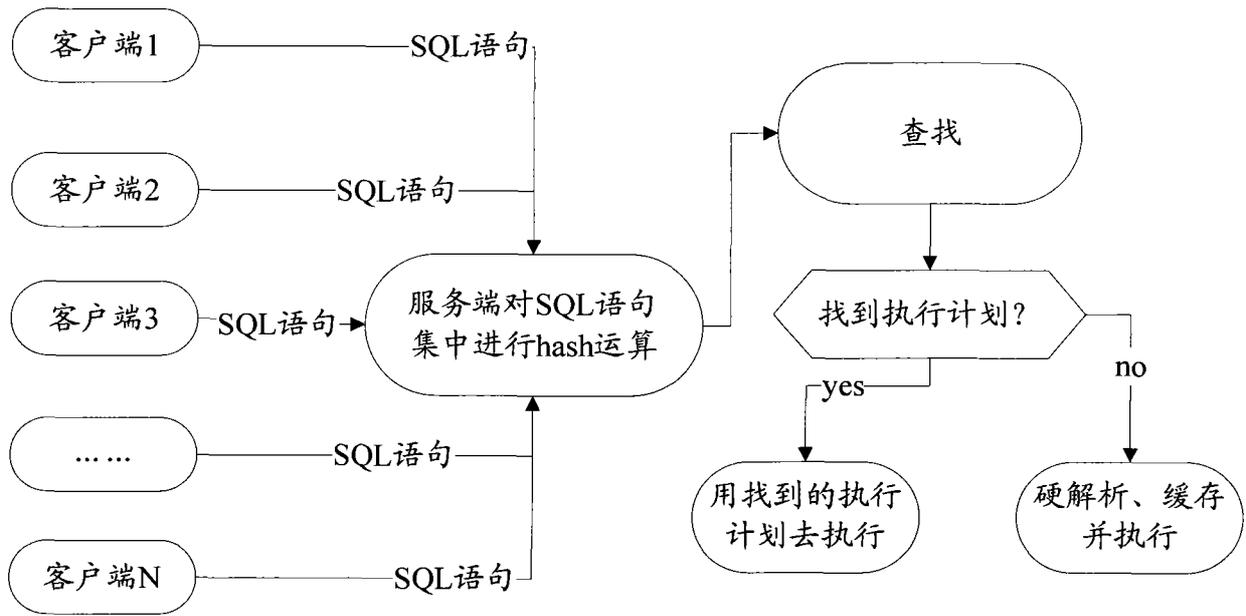


图 1

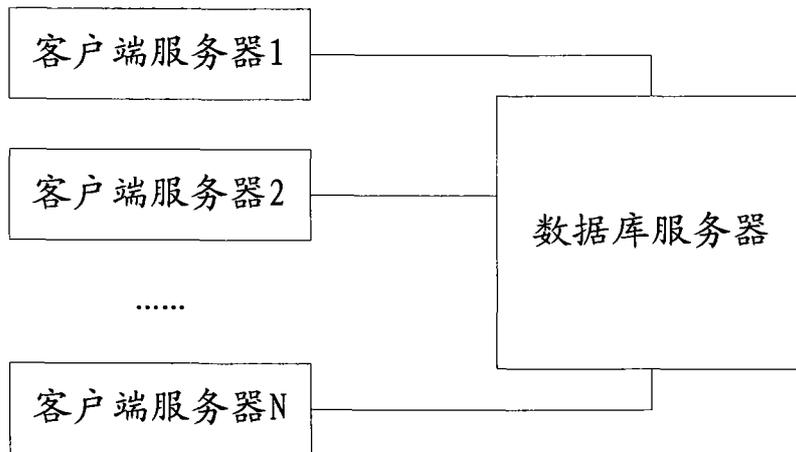


图 2

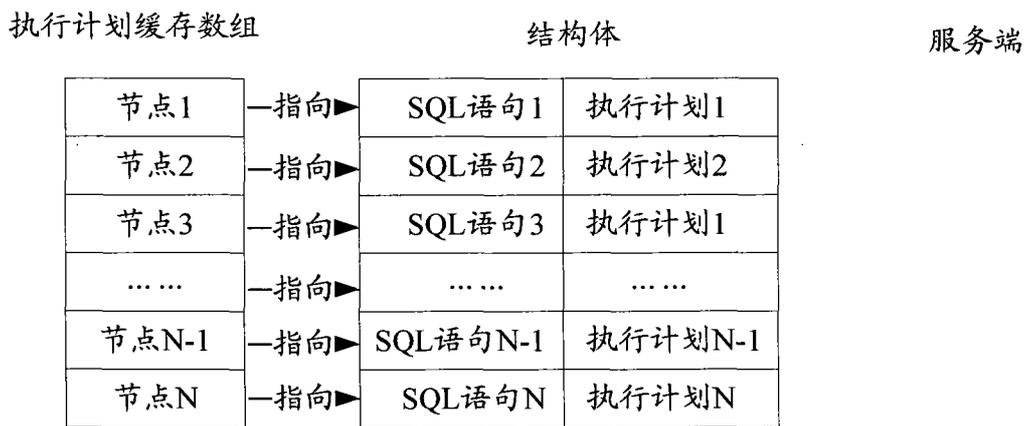
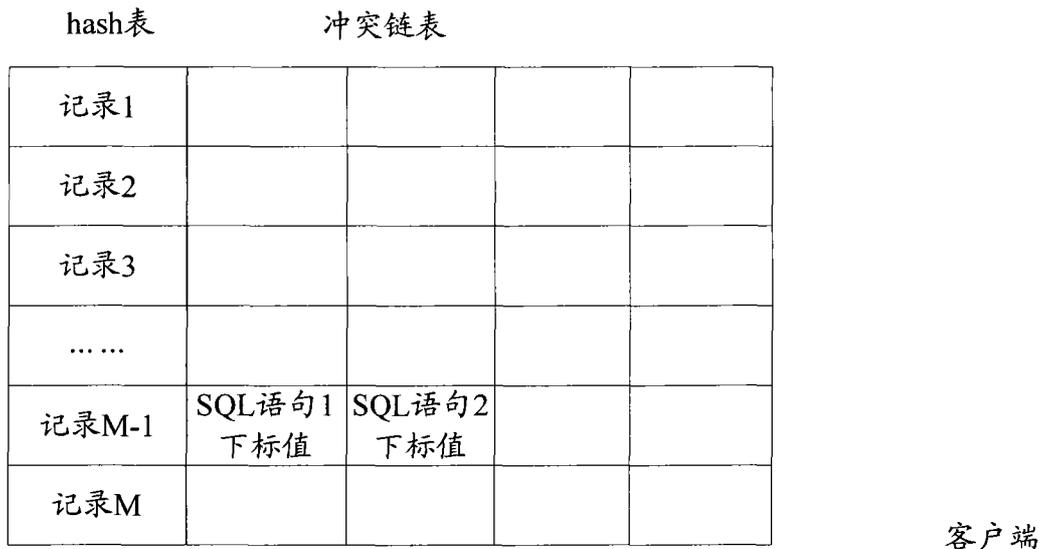


图 3

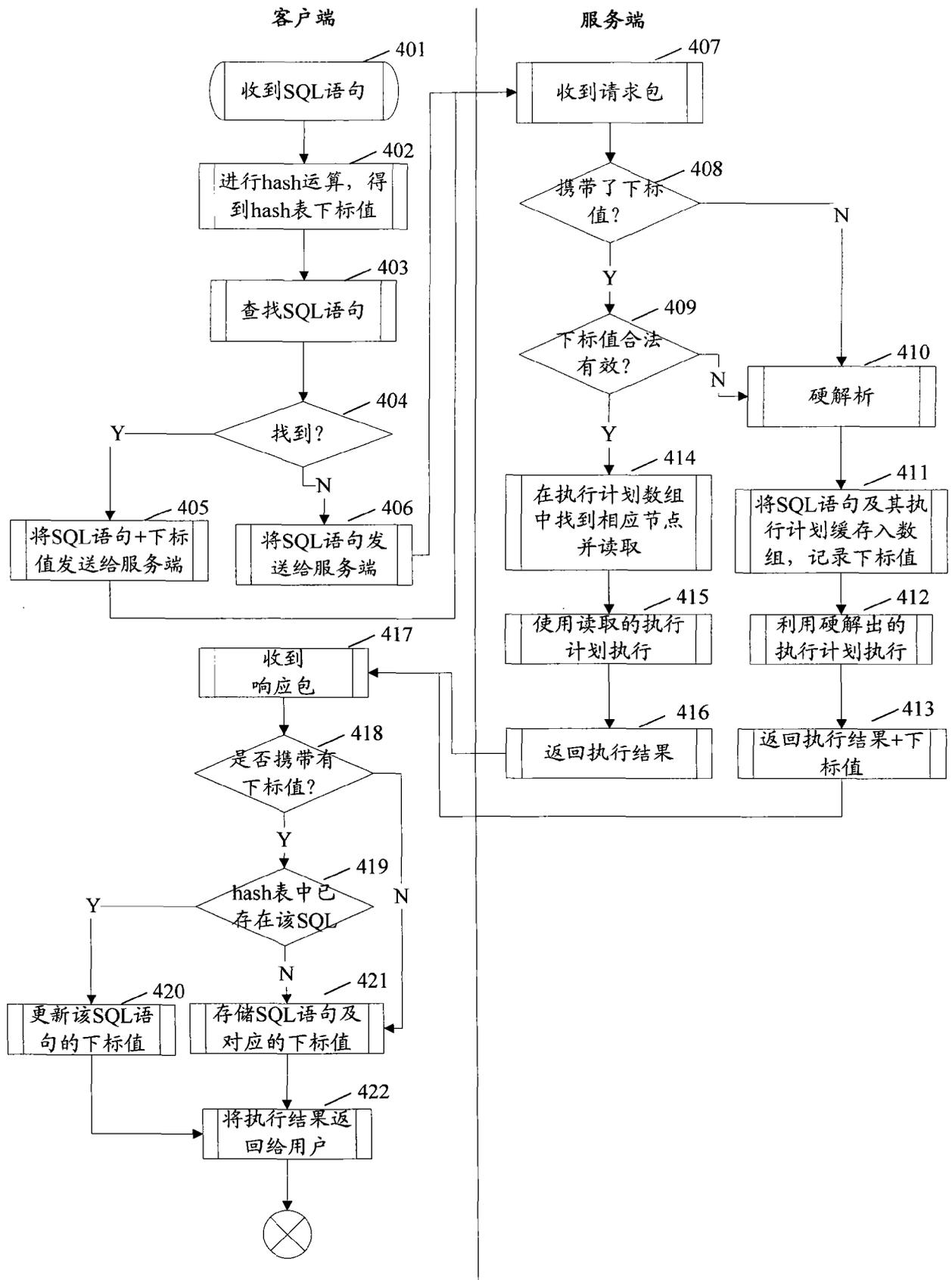


图 4