(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 112650759 A
(43)申请公布日 2021.04.13

(21)申请号 202011644405.0

(22)申请日 2020.12.30

(71)申请人 中国平安人寿保险股份有限公司
地址 518000 广东省深圳市福田区益田路5033号平安金融中心14、15、16、41、44、45、46层

(72)发明人 罗华

(74)专利代理机构 深圳市赛恩倍吉知识产权代理有限公司 44334
代理人 杨毅玲 刘丽华

(51)Int.Cl.
G06F 16/22(2019.01)
G06F 16/27(2019.01)
G06F 16/2455(2019.01)
G06F 16/2453(2019.01)

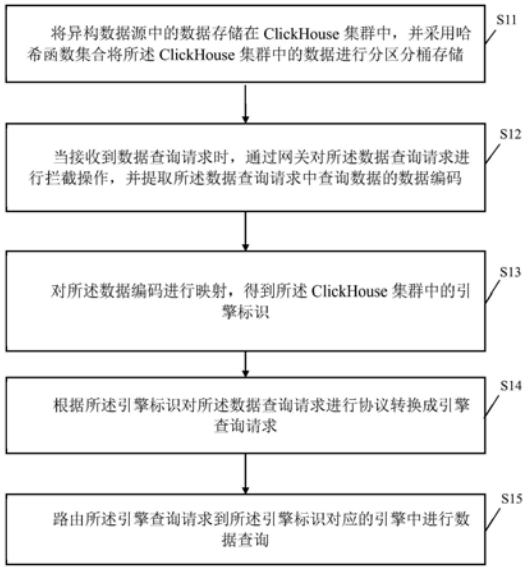
权利要求书2页 说明书13页 附图2页

(54)发明名称

数据查询方法、装置、计算机设备及存储介质

(57)摘要

本发明涉及数据处理技术领域，提供一种数据查询方法、装置、计算机设备及存储介质，包括：将异构数据源中的数据存储在ClickHouse集群中，并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储；当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码；对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求；路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。本发明能够提高异构数据源的数据查询效率。



CN 112650759 A

1.一种数据查询方法,其特征在于,所述方法包括:

将异构数据源中的数据存储在ClickHouse集群中,并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储;

当接收到数据查询请求时,通过网关对所述数据查询请求进行拦截操作,并提取所述数据查询请求中查询数据的数据编码;

对所述数据编码进行映射,得到所述ClickHouse集群中的引擎标识;

根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求;

路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。

2.如权利要求1所述的数据查询方法,其特征在于,所述采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储包括:

从哈希函数集合中任意选取K个哈希函数构造多个哈希函数子集;

根据所述多个哈希函数子集在ClickHouse集群中创建多个分区,并计算每个分区的分区索引;

采用每个哈希函数子集中的每个哈希函数将所述数据映射到哈希桶中;

根据所述哈希桶的个数确定存储所述数据的多个目标分区,并将所述多个目标分区的分区索引进行顺序排序,将所述数据存储于排序在第一的分区索引对应的目标分区中。

3.如权利要求2所述的数据查询方法,其特征在于,所述根据所述多个哈希函数子集在ClickHouse集群中创建多个分区,并计算每个分区的分区索引包括:

计算所述多个哈希函数子集的子集数;

根据所述子集数在ClickHouse集群中创建多个分区,每个分区对应一个哈希函数子集;

确定每个哈希函数子集中每个哈希函数在所述哈希函数集合中的位置索引;

根据每个哈希函数子集中的多个位置索引计算对应所述哈希函数子集的分区的分区索引。

4.如权利要求1所述的数据查询方法,其特征在于,所述根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求包括:

解析所述数据查询请求得到第一IP地址和第一端口地址;

根据所述引擎标识更新所述第一IP地址得到第二IP地址;

根据所述引擎标识更新所述第一端口地址得到第二端口地址;

基于所述第二IP地址及所述第二端口地址生成引擎查询请求。

5.如权利要求1至4中任意一项所述的数据查询方法,其特征在于,所述将异构数据源中的数据存储在ClickHouse集群中包括:

识别所述异构数据源中的数据的数据源;

当所述数据源为第一类型数据源时,匹配与所述第一类型数据源对应的第一引擎,并采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中;

当所述数据源为第二类型数据源时,匹配与所述第二类型数据源对应的第二引擎,并采用所述第二引擎将所述第二类型数据源中的数据存储在ClickHouse集群中。

6.如权利要求5所述的数据查询方法,其特征在于,所述采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中包括:

提取所述第一类型数据源中的数据的第一字段；

将所述第一字段的字段类型映射为所述ClickHouse集群中的第二字段的字段类型；

采用所述第一引擎基于所述第二字段的字段类型将所述第一类型数据源中的数据存储在ClickHouse集群中。

7.如权利要求1至4中任意一项所述的数据查询方法，其特征在于，所述方法还包括：

响应于所述ClickHouse集群中的目标引擎的更换指令，获取所述目标引擎的目标引擎标识；

确定所述目标引擎中的目标数据的目标数据编码；

根据所述目标数据编码及所述目标引擎标识对所述网关进行染色处理。

8.一种数据查询装置，其特征在于，所述装置包括：

存储模块，用于将异构数据源中的数据存储在ClickHouse集群中，并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储；

拦截模块，用于当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码；

映射模块，用于对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；

转换模块，用于根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求；

查询模块，用于路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。

9.一种计算机设备，其特征在于，所述计算机设备包括处理器，所述处理器用于执行存储器中存储的计算机程序时实现如权利要求1至7中任意一项所述的数据查询方法。

10.一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现如权利要求1至7中任意一项所述的数据查询方法。

数据查询方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明涉及数据处理技术领域，具体涉及一种数据查询方法、装置、计算机设备及存储介质。

背景技术

[0002] 在做数据服务的时候，由于数据的特性以及数据量多少的问题会将数据存放到不同的存储引擎进行加工处理。

[0003] 然而，发明人在实现本发明的过程中发现，当需要查询数据时，由于数据存储在不同的存储引擎，则需要开发一套通用路由服务来进行聚合查询，目前没有很好的方案能够解决跨数据库之间数据表的关联聚合分析问题；且查询数据的时候需要集成各种大数据组件，且需要查询者知道底层存储在哪里，而且后续增加组件的时候又需要开发一套组件，系统复杂度高，导致数据查询效率低。

发明内容

[0004] 鉴于以上内容，有必要提出一种数据查询方法、装置、计算机设备及存储介质，能够提高异构数据源的数据查询效率。

[0005] 本发明的第一方面提供一种数据查询方法，所述方法包括：

[0006] 将异构数据源中的数据存储在ClickHouse集群中，并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储；

[0007] 当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码；

[0008] 对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；

[0009] 根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求；

[0010] 路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。

[0011] 在一个可选的实施例中，所述采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储包括：

[0012] 从哈希函数集合中任意选取K个哈希函数构造多个哈希函数子集；

[0013] 根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引；

[0014] 采用每个哈希函数子集中的每个哈希函数将所述数据映射到哈希桶中；

[0015] 根据所述哈希桶的个数确定存储所述数据的多个目标分区，并将所述目标分区的分区索引进行顺序排序，并将所述数据存储于排序在第一的在分区索引对应的目标分区中。

[0016] 在一个可选的实施例中，所述根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引包括：

[0017] 计算所述多个哈希函数子集的子集数；

[0018] 根据所述子集数在ClickHouse集群中创建多个分区，每个分区对应一个哈希函数子集；

[0019] 确定每个哈希函数子集中每个哈希函数在所述哈希函数集合中的位置索引；

[0020] 根据每个哈希函数子集中的多个位置索引计算对应所述哈希函数子集的分区的分区索引。

[0021] 在一个可选的实施例中，所述根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求包括：

[0022] 解析所述数据查询请求得到第一IP地址和第一端口地址；

[0023] 根据所述引擎标识更新所述第一IP地址得到第二IP地址；

[0024] 根据所述引擎标识更新所述第一端口地址得到第二端口地址；

[0025] 基于所述第二IP地址及所述第二端口地址生成引擎查询请求。

[0026] 在一个可选的实施例中，所述将异构数据源中的数据存储在ClickHouse集群中包括：

[0027] 识别所述异构数据源中的数据的数据源；

[0028] 当所述数据源为第一类型数据源时，匹配与所述第一类型数据源对应的第一引擎，并采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中；

[0029] 当所述数据源为第二类型数据源时，匹配与所述第二类型数据源对应的第二引擎，并采用所述第二引擎将所述第二类型数据源中的数据存储在ClickHouse集群中。

[0030] 在一个可选的实施例中，所述采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中包括：

[0031] 提取所述第一类型数据源中的数据的第一字段；

[0032] 将所述第一字段的字段类型映射为所述ClickHouse集群中的第二字段的字段类型；

[0033] 采用所述第一引擎基于所述第二字段的字段类型将所述第一类型数据源中的数据存储在ClickHouse集群中。

[0034] 在一个可选的实施例中，所述方法还包括：

[0035] 响应于所述ClickHouse集群中的目标引擎的更换指令，获取所述目标引擎的目标引擎标识；

[0036] 确定所述目标引擎中的目标数据的目标数据编码；

[0037] 根据所述目标数据编码及所述目标引擎标识对所述网关进行染色处理。

[0038] 本发明的第二方面提供一种数据查询装置，所述装置包括：

[0039] 存储模块，用于将异构数据源中的数据存储在ClickHouse集群中，并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储；

[0040] 拦截模块，用于当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码；

[0041] 映射模块，用于对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；

[0042] 转换模块，用于根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求；

[0043] 查询模块，用于路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。

[0044] 本发明的第三方面提供一种计算机设备，所述计算机设备包括处理器，所述处理器用于执行存储器中存储的计算机程序时实现所述数据查询方法。

[0045] 本发明的第四方面提供一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现所述数据查询方法。

[0046] 综上所述，本发明所述的数据查询方法、装置、计算机设备及存储介质，首先将异构数据源中的数据存储在ClickHouse集群中，能解决异构数据源里面数据表的关联聚合以及排序问题，基于ClickHouse的列式存储和MPP并行化查询提升海量数据的查询性能；同时采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储，便于后续进行数据查询时，能够从对应的分区分桶中查询数据，进一步提高数据的查询效率；在接收到数据查询请求时，线通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码，接着对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求，从而转换成对应引擎的HTTP请求方式查询数据，最后路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询，提高了数据查询效率。

附图说明

[0047] 图1是本发明实施例一提供的数据查询方法的流程图。

[0048] 图2是本发明实施例二提供的数据查询装置的结构图。

[0049] 图3是本发明实施例三提供的计算机设备的结构示意图。

具体实施方式

[0050] 为了能够更清楚地理解本发明的上述目的、特征和优点，下面结合附图和具体实施例对本发明进行详细描述。需要说明的是，在不冲突的情况下，本发明的实施例及实施例中的特征可以相互组合。

[0051] 除非另有定义，本文所使用的所有的技术和科学术语与属于本发明的技术领域的技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的，不是旨在于限制本发明。

[0052] 本发明实施例提供的数据查询方法由计算机设备执行，相应地，数据查询装置运行于计算机设备中。

[0053] 图1是本发明实施例一提供的数据查询方法的流程图。所述数据查询方法具体包括以下步骤，根据不同的需求，该流程图中步骤的顺序可以改变，某些可以省略。

[0054] S11，将异构数据源中的数据存储在ClickHouse集群中，并采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储。

[0055] 所述异构数据源中的数据是指来源于不同数据源的数据，即来源于不同存储引擎的数据。

[0056] ClickHouse为开源的数据分析性的数据库。

[0057] 可以事先搭建ClickHouse集群的运行环境，然后将异构数据源中的多个数据整合

在ClickHouse集群中。

[0058] 在一个可选的实施例中，所述将异构数据源中的数据存储在ClickHouse集群中包括：

[0059] 识别所述异构数据源中的数据的数据源；

[0060] 当所述数据源为第一类型数据源时，匹配与所述第一类型数据源对应的第一引擎，并采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中；

[0061] 当所述数据源为第二类型数据源时，匹配与所述第二类型数据源对应的第二引擎，并采用所述第二引擎将所述第二类型数据源中的数据存储在ClickHouse集群中。

[0062] 其中，所述数据可以为库表型数据，也可以为文件型数据。

[0063] 可以在获取数据的同时获取数据的标识信息，根据所述标识信息即可识别出所述数据的数据源。可以与本地数据库进行匹配来确定所述数据源的类型，其中，所述本地数据库中记录了数据源与数据源类型之间的第一映射关系及数据源类型与引擎标识之间的第二映射关系，根据所述第一映射关系可以确定出所述数据的数据源类型，根据所述第二映射关系可以确定采用何种引擎来存储所述数据。

[0064] 示例性的，可以采用ClickHouse的JDBC引擎将处在不同关系型数据库里面的数据整合到ClickHouse集群里面，这样处理可以整合oracle、mysql、PostgreSQL等关系型数据库里面的数据，可以采用ClickHouse的HDFS引擎将处在Hadoop上面的数据文件整合到ClickHouse集群里面。

[0065] 该可选的实施例中，通过采用不同的引擎将异构数据源的数据整合到ClickHouse集群中，对于上层系统的用户而言，不需要关心底层存储引擎，同时能优雅地解决异构数据源里面数据表的关联聚合以及排序问题，还能基于ClickHouse的列式存储和MPP并行化查询提升海量数据的查询性能。

[0066] 在一个可选的实施例中，所述采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中包括：

[0067] 提取所述第一类型数据源中的数据的第一字段；

[0068] 将所述第一字段的字段类型映射为所述ClickHouse集群中的第二字段的字段类型；

[0069] 采用所述第一引擎基于所述第二字段的字段类型将所述第一类型数据源中的数据存储在ClickHouse集群中。

[0070] 将不同关系型数据库中的数据的字段类型与ClickHouse做一个映射，例如mysql里面的int型映射为ClickHouse中Uint16类型，varchar型映射为String类型，json型映射为字符串String型的。

[0071] 该可选的实施例中，根据字段的字段类型将数据存储为ClickHouse集群中统一的字段类型，确保数据格式统一，便于后续进行数据查询。

[0072] 在一个可选的实施例中，所述采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储包括：

[0073] 从哈希函数集合中任意选取K个哈希函数构造多个哈希函数子集；

[0074] 根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引；

[0075] 采用每个哈希函数子集中的每个哈希函数将所述数据映射到哈希桶中；

[0076] 根据所述哈希桶的个数确定存储所述数据的多个目标分区，并将所述目标分区的分区索引进行顺序排序，并将所述数据存储于排序在第一的在分区索引对应的目标分区中。

[0077] 其中，所述哈希函数集合可以包括基于比特取样的敏感哈希函数(Locality Sensitive Hash，LSH)、基于最小独立置换的LSH、基于随机投影的LSH、基于Lattice的LSH以及基于P稳定分布的LSH。

[0078] 示例性的，假设哈希函数集合中有30个哈希函数，每次任意选取出10个哈希函数即可重新形成一个新的哈希函数集合，如此，可以选取出C_{30}^{10}个哈希函数子集，每个哈希函数子集中包括10个哈希函数。

[0079] 使用哈希函数子集中的哈希函数计算所述数据的哈希值，所述哈希值作为哈希桶的标识，最后计算所有哈希桶的个数之和，从而根据哈希桶的个数来确定所述数据可以存储在哪一个分区中。具体实施时，可以将哈希桶的个数越少的哈希桶所在的分区确定为目标分区。当有多个目标分区时，将具有最小分区索引的目标分区确定为最终存储所述数据的目标分区，并存储在哈希桶的桶号索引最小的哈希桶中。将所述目标分区的分区索引及最小的桶号索引连接起来作为所述数据的数据编码。该可选的实施例中，通过哈希函数将数据映射到哈希桶，能够很好的将数据均匀的分散在ClickHouse集群中，能够有效的避免数据倾斜，提高了ClickHouse集群的性能和资源使用率，确保了ClickHouse集群的稳定性；此外，根据哈希桶的个数来确定存储所述数据的目标分区，能够后续查询数据时，能够减少计算量，快速的确定查询的数据所在的目标分区。

[0080] 在一个可选的实施例中，所述根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引包括：

[0081] 计算所述多个哈希函数子集的子集数；

[0082] 根据所述子集数在ClickHouse集群中创建多个分区，每个分区对应一个哈希函数子集；

[0083] 确定每个哈希函数子集中每个哈希函数在所述哈希函数集合中的位置索引；

[0084] 根据每个哈希函数子集中的多个位置索引计算对应所述哈希函数子集的分区的分区索引。

[0085] 示例性的，假设有10个哈希函数子集，则创建10个分区，每个分区对应一个哈希函数子集。第1个哈希函数子集包括3个哈希函数，这3个哈希函数在哈希函数集合中的位置索引分别为2,5,8，则第1个哈希函数子集对应的第1个分区的分区索引可以为258。第2个哈希函数子集包括3个哈希函数，这3个哈希函数在哈希函数集合中的位置索引分别为5,2,9，则第1个哈希函数子集对应的第1个分区的分区索引可以为529。

[0086] 由于选取出C_{30}^{10}个哈希函数子集，则对应创建C_{30}^{10}个分区，对于第一个数据而言，依次使用C_{30}^{10}个哈希函数子集中的每个哈希函数子集中的哈希函数计算第一个数据的哈希值，哈希值作为哈希桶的标识，相同的哈希值具有相同的哈希桶，计算每一个分区中所有哈希桶的个数之和，将哈希桶的个数最小的分区确定为目标分区。如果目标分区有多个，则将目标分区的索引最小的分区确定为第一个数据存储的目标分区。加入存储第一个数据的

目标分区为分区2,哈希桶为2,3,8,则将桶号索引2对应的哈希桶确定为存储第一个数据的哈希桶。

[0087] 该可选的实施例中,根据哈希函数子集中的哈希函数在所述哈希函数集合中的位置索引来计算对应的分区索引,计算效率更高。

[0088] S12,当接收到数据查询请求时,通过网关对所述数据查询请求进行拦截操作,并提取所述数据查询请求中查询数据的数据编码。

[0089] 用户可以在计算机设备显示的页面的搜索框中输入查询关键词来触发数据查询请求。

[0090] 计算机设备响应于所述数据查询请求,发送所述数据查询请求至网关,网关调用拦截函数执行对所述数据查询请求的拦截操作。

[0091] 具体实施时,可以使用每个分区对应的哈希函数子集中的每个哈希函数将所述查询数据映射到哈希桶中,然后根据所述哈希桶的个数确定存储所述数据的查询分区,根据所述目标分区的分区索引及最小的哈希桶的桶索引生成所述数据的数据编码。

[0092] S13,对所述数据编码进行映射,得到所述ClickHouse集群中的引擎标识。

[0093] 计算机设备中存储有ClickHouse集群中引擎的引擎标识及对应引擎中的数据的数据编码之间的映射关系,根据该映射关系可以确定所述查询数据对应的引擎标识。

[0094] S14,根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求。

[0095] 不同的查询用户可能输入的查询请求的协议不同,为便于快速从ClickHouse集群中查询到查询数据,则需要根据引擎标识进行协议转换。

[0096] 在一个可选的实施例中,所述根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求包括:

[0097] 解析所述数据查询请求得到第一IP地址和第一端口地址;

[0098] 根据所述引擎标识更新所述第一IP地址得到第二IP地址;

[0099] 根据所述引擎标识更新所述第一端口地址得到第二端口地址;

[0100] 基于所述第二IP地址及所述第二端口地址生成引擎查询请求。

[0101] 示例性的,假设网关拦截到的数据查询请求是用SQL语句编写的,例如,select* from T1 where account_code＝'AC001',然后提取查询数据AC001的数据编码,再确定数据编码对应的查询引擎的查询引擎标识。再根据所述查询引擎标识确定IP地址和端口,进而转换成对应引擎的HTTP请求方式查询数据。

[0102] 该可选的实施例中,在查询数据时,在网关做个协议转换,全部转换成通用的HTTP协议请求,以便方便集成新的大数据组件同时对于上层指标数据的查询用户使用透明。

[0103] S15,路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询。

[0104] 将前端传过来的请求经过网关后转换成HTTP请求,然后路由到引擎标识对应的引擎,这样转换成通用的HTTP协议之后使得查询数据更通用而且适配性更好,只需要前端用户接入网关即可轻松实现指标数据的查询,解耦系统架构,降低了使用者的学习成本以及系统的复杂度。

[0105] 在一个可选的实施例中,所述方法还包括:

[0106] 响应于所述ClickHouse集群中的目标引擎的更换指令,获取所述目标引擎的目标引擎标识;

[0107] 确定所述目标引擎中的目标数据的目标数据编码；

[0108] 根据所述目标数据编码及所述目标引擎标识对所述网关进行染色处理。

[0109] 当在ClickHouse集群中添加了某个目标引擎时，则会触发目标引擎的更换指令。

[0110] 网关染色是指将新添加的目标引擎的引擎标识及目标引擎中的数据的数据编码关联存储在网关中的redis里面。

[0111] 该可选的实施例中，通过对网关进行染色处理，使得需要更换引擎时，只需要在网关中修改对应的信息即可，如此，在查询的时候便不需要传递具体的引擎标识，减少了前端查询的改动量，进一步提高了查询的效率。

[0112] 在一个可选的实施例中，所述方法还包括：提取查询到的数据中的多个数据字段；确定所述多个数据字段中的目标数据字段；对所述目标数据字段进行脱敏处理；返回脱敏处理的数据。

[0113] 计算机设备中预先配置有脱敏配置表，所述脱敏配置表中记录了需要进行脱敏处理的字段。

[0114] 为了提高数据的安全性，在查询到数据之后，将查询到的数据的数据字段与脱敏配置表中的多个进行脱敏处理的字段进行逐一匹配，当匹配成功时，对匹配成功的数据字段进行脱敏处理。

[0115] 需要强调的是，为进一步保证上述目标引擎的引擎标识及目标引擎中的数据的数据编码的关联关系的私密性和安全性，上述目标引擎的引擎标识及目标引擎中的数据的数据编码的关联关系可存储于区块链的节点中。

[0116] 本发明实施例所述的数据查询方法，首先将异构数据源中的数据存储在ClickHouse集群中，能解决异构数据源里面数据表的关联聚合以及排序问题，基于ClickHouse的列式存储和MPP并行化查询提升海量数据的查询性能；同时采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储，便于后续进行数据查询时，能够从对应的分区分桶中查询数据，进一步提高数据的查询效率；在接收到数据查询请求时，线通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码，接着对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识；根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求，从而转换成对应引擎的HTTP请求方式查询数据，最后路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询，提高了数据查询效率。

[0117] 图2是本发明实施例二提供的数据查询装置的结构图。

[0118] 在一些实施例中，所述数据查询装置20可以包括多个由计算机程序段所组成的功能模块。所述数据查询装置20中的各个程序段的计算机程序可以存储于计算机设备的存储器中，并由至少一个处理器所执行，以执行(详见图1描述)数据查询的功能。

[0119] 本实施例中，所述数据查询装置20根据其所执行的功能，可以被划分为多个功能模块。所述功能模块可以包括：存储模块201、拦截模块202、映射模块203、转换模块204、查询模块205及染色模块206。本发明所称的模块是指一种能够被至少一个处理器所执行并且能够完成固定功能的一系列计算机程序段，其存储在存储器中。在本实施例中，关于各模块的功能将在后续的实施例中详述。

[0120] 所述存储模块201，用于将异构数据源中的数据存储在ClickHouse集群中，并采用

哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储。

[0121] 所述异构数据源中的数据是指来源于不同数据源的数据，即来源于不同存储引擎的数据。

[0122] ClickHouse为开源的数据分析性的数据库。

[0123] 可以事先搭建ClickHouse集群的运行环境，然后将异构数据源中的多个数据整合在ClickHouse集群中。

[0124] 在一个可选的实施例中，所述存储模块201将异构数据源中的数据存储在ClickHouse集群中包括：

[0125] 识别所述异构数据源中的数据的数据源；

[0126] 当所述数据源为第一类型数据源时，匹配与所述第一类型数据源对应的第一引擎，并采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中；

[0127] 当所述数据源为第二类型数据源时，匹配与所述第二类型数据源对应的第二引擎，并采用所述第二引擎将所述第二类型数据源中的数据存储在ClickHouse集群中。

[0128] 其中，所述数据可以为库表型数据，也可以为文件型数据。

[0129] 可以在获取数据的同时获取数据的标识信息，根据所述标识信息即可识别出所述数据的数据源。可以与本地数据库进行匹配来确定所述数据源的类型，其中，所述本地数据库中记录了数据源与数据源类型之间的第一映射关系及数据源类型与引擎标识之间的第二映射关系，根据所述第一映射关系可以确定出所述数据的数据源类型，根据所述第二映射关系可以确定采用何种引擎来存储所述数据。

[0130] 示例性的，可以采用ClickHouse的JDBC引擎将处在不同关系型数据库里面的数据整合到ClickHouse集群里面，这样处理可以整合oracle、mysql、PostgreSQL等关系型数据库里面的数据，可以采用ClickHouse的HDFS引擎将处在Hadoop上面的数据文件整合到ClickHouse集群里面。

[0131] 该可选的实施例中，通过采用不同的引擎将异构数据源的数据整合到ClickHouse集群中，对于上层系统的用户而言，不需要关心底层存储引擎，同时能优雅地解决异构数据源里面数据表的关联聚合以及排序问题，还能基于ClickHouse的列式存储和MPP并行化查询提升海量数据的查询性能。

[0132] 在一个可选的实施例中，所述采用所述第一引擎将所述第一类型数据源中的数据存储在ClickHouse集群中包括：

[0133] 提取所述第一类型数据源中的数据的第一字段；

[0134] 将所述第一字段的字段类型映射为所述ClickHouse集群中的第二字段的字段类型；

[0135] 采用所述第一引擎基于所述第二字段的字段类型将所述第一类型数据源中的数据存储在ClickHouse集群中。

[0136] 将不同关系型数据库中的数据的字段类型与ClickHouse做一个映射，例如mysql里面的int型映射为ClickHouse中Uint16类型，varchar型映射为String类型，json型映射为字符串String型的。

[0137] 该可选的实施例中，根据字段的字段类型将数据存储为ClickHouse集群中统一的字段类型，确保数据格式统一，便于后续进行数据查询。

[0138] 在一个可选的实施例中，所述采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储包括：

[0139] 从哈希函数集合中任意选取K个哈希函数构造多个哈希函数子集；

[0140] 根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引；

[0141] 采用每个哈希函数子集中的每个哈希函数将所述数据映射到哈希桶中；

[0142] 根据所述哈希桶的个数确定存储所述数据的多个目标分区，并将所述多个目标分区的分区索引进行顺序排序，将所述数据存储于排序在第一的分区索引对应的目标分区中。

[0143] 其中，所述哈希函数集合可以包括基于比特取样的敏感哈希函数(Locality Sensitive Hash，LSH)、基于最小独立置换的LSH、基于随机投影的LSH、基于Lattice的LSH以及基于P稳定分布的LSH。

[0144] 示例性的，假设哈希函数集合中有30个哈希函数，每次任意选取出10个哈希函数即可重新形成一个新的哈希函数集合，如此，可以选取出C_{30}^{10}个哈希函数子集。

[0145] 使用哈希函数子集中的哈希函数计算所述数据的哈希值，所述哈希值作为哈希桶的标识，最后计算所有哈希桶的个数之和，从而根据哈希桶的个数来确定所述数据可以存储在哪一个分区中。

[0146] 具体实施时，可以将哈希桶的个数越少的哈希桶所在的分区确定为目标分区。当有多个目标分区时，将具有最小分区索引的目标分区确定为最终存储所述数据的目标分区，并存储在哈希桶的桶号索引最小的哈希桶中。将所述目标分区的分区索引及最小的桶号索引连接起来作为所述数据的数据编码。

[0147] 该可选的实施例中，通过哈希函数将数据映射到哈希桶，能够很好的将数据均匀的分散在ClickHouse集群中，能够有效的避免数据倾斜，提高了ClickHouse集群的性能和资源使用率，确保了ClickHouse集群的稳定性；此外，根据哈希桶的个数来确定存储所述数据的目标分区，能够后续查询数据时，能够减少计算量，快速的确定查询的数据所在的目标分区。

[0148] 在一个可选的实施例中，所述根据所述多个哈希函数子集在ClickHouse集群中创建多个分区，并计算每个分区的分区索引包括：

[0149] 计算所述多个哈希函数子集的子集数；

[0150] 根据所述子集数在ClickHouse集群中创建多个分区，每个分区对应一个哈希函数子集；

[0151] 确定每个哈希函数子集中每个哈希函数在所述哈希函数集合中的位置索引；

[0152] 根据每个哈希函数子集中的多个位置索引计算对应所述哈希函数子集的分区的分区索引。

[0153] 示例性的，假设有10个哈希函数子集，则创建10个分区，每个分区对应一个哈希函数子集。第1个哈希函数子集包括3个哈希函数，这3个哈希函数在哈希函数集合中的位置索引分别为2,5,8，则第1个哈希函数子集对应的第1个分区的分区索引可以为258。第2个哈希函数子集包括3个哈希函数，这3个哈希函数在哈希函数集合中的位置索引分别为5,2,9，则第1个哈希函数子集对应的第1个分区的分区索引可以为529。

[0154] 由于选取出C_30^10个哈希函数子集，则对应创建C_30^10个分区，对于第一个数据而言，依次使用C_30^10个哈希函数子集中的每个哈希函数子集中的哈希函数计算第一个数据的哈希值，哈希值作为哈希桶的标识，相同的哈希值具有相同的哈希桶，计算每一个分区中所有哈希桶的个数之和，将哈希桶的个数最小的分区确定为目标分区。如果目标分区有多个，则将目标分区的索引最小的分区确定为第一个数据存储的目标分区。加入存储第一个数据的目标分区为分区2，哈希桶为2,3,8，则将桶号索引2对应的哈希桶确定为存储第一个数据的哈希桶。

[0155] 该可选的实施例中，根据哈希函数子集中的哈希函数在所述哈希函数集合中的位置索引来计算对应的分区索引，计算效率更高。

[0156] 所述拦截模块202，用于当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码。

[0157] 用户可以在计算机设备显示的页面的搜索框中输入查询关键词来触发数据查询请求。

[0158] 计算机设备响应于所述数据查询请求，发送所述数据查询请求至网关，网关调用拦截函数执行对所述数据查询请求的拦截操作。

[0159] 具体实施时，可以使用每个分区对应的哈希函数子集中的每个哈希函数将所述查询数据映射到哈希桶中，然后根据所述哈希桶的个数确定存储所述数据的查询分区，根据所述目标分区的分区索引及最小的哈希桶的桶索引生成所述数据的数据编码。

[0160] 所述映射模块203，用于对所述数据编码进行映射，得到所述ClickHouse集群中的引擎标识。

[0161] 计算机设备中存储有ClickHouse集群中引擎的引擎标识及对应引擎中的数据的数据编码之间的映射关系，根据该映射关系可以确定所述查询数据对应的引擎标识。

[0162] 所述转换模块204，用于根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求。

[0163] 不同的查询用户可能输入的查询请求的协议不同，为便于快速从ClickHouse集群中查询到查询数据，则需要根据引擎标识进行协议转换。

[0164] 在一个可选的实施例中，所述根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求包括：

[0165] 解析所述数据查询请求得到第一IP地址和第一端口地址；

[0166] 根据所述引擎标识更新所述第一IP地址得到第二IP地址；

[0167] 根据所述引擎标识更新所述第一端口地址得到第二端口地址；

[0168] 基于所述第二IP地址及所述第二端口地址生成引擎查询请求。

[0169] 示例性的，假设网关拦截到的数据查询请求是用SQL语句编写的，例如，select* from T1 where account_code＝’AC001’，然后提取查询数据AC001的数据编码，再确定数据编码对应的查询引擎的查询引擎标识。再根据所述查询引擎标识确定IP地址和端口，进而转换成对应引擎的HTTP请求方式查询数据。

[0170] 该可选的实施例中，在查询数据时，在网关做个协议转换，全部转换成通用的HTTP协议请求，以便方便集成新的大数据组件同时对于上层指标数据的查询用户使用透明。

[0171] 所述查询模块205，用于路由所述引擎查询请求到所述引擎标识对应的引擎中进

行数据查询。

[0172] 将前端传过来的请求经过网关后转换成HTTP请求,然后路由到引擎标识对应的引擎,这样转换成通用的HTTP协议之后使得查询数据更通用而且适配性更好,只需要前端用户接入网关即可轻松实现指标数据的查询,解耦系统架构,降低了使用者的学习成本以及系统的复杂度。

[0173] 所述染色模块206,用于响应于所述ClickHouse集群中的目标引擎的更换指令,获取所述目标引擎的目标引擎标识;确定所述目标引擎中的目标数据的目标数据编码;根据所述目标数据编码及所述目标引擎标识对所述网关进行染色处理。

[0174] 当在ClickHouse集群中添加了某个目标引擎时,则会触发目标引擎的更换指令。

[0175] 网关染色是指将新添加的目标引擎的引擎标识及目标引擎中的数据的数据编码关联存储在网关中的redis里面。

[0176] 该可选的实施例中,通过对网关进行染色处理,使得需要更换引擎时,只需要在网关中修改对应的信息即可,如此,在查询的时候便不需要传递具体的引擎标识,减少了前端查询的改动量,进一步提高了查询的效率。

[0177] 在一个可选的实施例中,所述计算机设备还可以提取查询到的数据中的多个数据字段;确定所述多个数据字段中的目标数据字段;对所述目标数据字段进行脱敏处理;返回脱敏处理的数据。

[0178] 计算机设备中预先配置有脱敏配置表,所述脱敏配置表中记录了需要进行脱敏处理的字段。

[0179] 为了提高数据的安全性,在查询到数据之后,将查询到的数据的数据字段与脱敏配置表中的多个进行脱敏处理的字段进行逐一匹配,当匹配成功时,对匹配成功的数据字段进行脱敏处理。

[0180] 需要强调的是,为进一步保证上述目标引擎的引擎标识及目标引擎中的数据的数据编码的关联关系的私密性和安全性,上述目标引擎的引擎标识及目标引擎中的数据的数据编码的关联关系可存储于区块链的节点中。

[0181] 本发明实施例所述的数据查询装置,首先将异构数据源中的数据存储在ClickHouse集群中,能解决异构数据源里面数据表的关联聚合以及排序问题,基于ClickHouse的列式存储和MPP并行化查询提升海量数据的查询性能;同时采用哈希函数集合将所述ClickHouse集群中的数据进行分区分桶存储,便于后续进行数据查询时,能够从对应的分区分桶中查询数据,进一步提高数据的查询效率;在接收到数据查询请求时,线通过网关对所述数据查询请求进行拦截操作,并提取所述数据查询请求中查询数据的数据编码,接着对所述数据编码进行映射,得到所述ClickHouse集群中的引擎标识;根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求,从而转换成对应引擎的HTTP请求方式查询数据,最后路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询,提高了数据查询效率。

[0182] 参阅图3所示,为本发明实施例三提供的计算机设备的结构示意图。在本发明较佳实施例中,所述计算机设备3包括存储器31、至少一个处理器32、至少一条通信总线33及收发器34。

[0183] 本领域技术人员应该了解,图3示出的计算机设备的结构并不构成本发明实施例

的限定,既可以是总线型结构,也可以是星形结构,所述计算机设备3还可以包括比图示更多或更少的其他硬件或者软件,或者不同的部件布置。

[0184] 在一些实施例中,所述计算机设备3是一种能够按照事先设定或存储的指令,自动进行数值计算和/或信息处理的设备,其硬件包括但不限于微处理器、专用集成电路、可编程门阵列、数字处理器及嵌入式设备等。所述计算机设备3还可包括客户设备,所述客户设备包括但不限于任何一种可与客户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互的电子产品,例如,个人计算机、平板电脑、智能手机、数码相机等。

[0185] 需要说明的是,所述计算机设备3仅为举例,其他现有的或今后可能出现的电子产品如可适应于本发明,也应包含在本发明的保护范围以内,并以引用方式包含于此。

[0186] 在一些实施例中,所述存储器31中存储有计算机程序,所述计算机程序被所述至少一个处理器32执行时实现如所述的数据查询方法中的全部或者部分步骤。所述存储器31包括只读存储器(Read-Only Memory,ROM)、可编程只读存储器(Programmable Read-Only Memory,PROM)、可擦除可编程只读存储器(Erasable Programmable Read-Only Memory,EPROM)、一次可编程只读存储器(One-time Programmable Read-Only Memory,OTPROM)、电子擦除式可复写只读存储器(Electrically-Erasable Programmable Read-Only Memory,EEPROM)、只读光盘(Compact Disc Read-Only Memory,CD-ROM)或其他光盘存储器、磁盘存储器、磁带存储器、或者能够用于携带或存储数据的计算机可读的任何其他介质。

[0187] 进一步地,所述计算机可读存储介质可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序等;存储数据区可存储根据区块链节点的使用所创建的数据等。

[0188] 本发明所指区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链(Blockchain),本质上是一个去中心化的数据库,是一串使用密码学方法相关联产生的数据块,每一个数据块中包含了一批次网络交易的信息,用于验证其信息的有效性(防伪)和生成下一个区块。区块链可以包括区块链底层平台、平台产品服务层以及应用服务层等。

[0189] 在一些实施例中,所述至少一个处理器32是所述计算机设备3的控制核心(Control Unit),利用各种接口和线路连接整个计算机设备3的各个部件,通过运行或执行存储在所述存储器31内的程序或者模块,以及调用存储在所述存储器31内的数据,以执行计算机设备3的各种功能和处理数据。例如,所述至少一个处理器32执行所述存储器中存储的计算机程序时实现本发明实施例中所述的数据查询方法的全部或者部分步骤;或者实现数据查询装置的全部或者部分功能。所述至少一个处理器32可以由集成电路组成,例如可以由单个封装的集成电路所组成,也可以是由多个相同功能或不同功能封装的集成电路所组成,包括一个或者多个中央处理器(Central Processing unit,CPU)、微处理器、数字处理芯片、图形处理器及各种控制芯片的组合等。

[0190] 在一些实施例中,所述至少一条通信总线33被设置为实现所述存储器31以及所述至少一个处理器32等之间的连接通信。

[0191] 尽管未示出,所述计算机设备3还可以包括给各个部件供电的电源(比如电池),优选的,电源可以通过电源管理装置与所述至少一个处理器32逻辑相连,从而通过电源管理装置实现管理充电、放电、以及功耗管理等功能。电源还可以包括一个或一个以上的直流或

交流电源、再充电装置、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。所述计算机设备3还可以包括多种传感器、蓝牙模块、Wi-Fi模块等，在此不再赘述。

[0192] 上述以软件功能模块的形式实现的集成的单元，可以存储在一个计算机可读取存储介质中。上述软件功能模块存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可以是个人计算机，计算机设备，或者网络设备等)或处理器(processor)执行本发明各个实施例所述方法的部分。

[0193] 在本发明所提供的几个实施例中，应该理解到，所揭露的装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述模块的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。

[0194] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的，作为模块显示的部件可以是或者也可以不是物理单元，既可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0195] 另外，在本发明各个实施例中的各功能模块可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能模块的形式实现。

[0196] 对于本领域技术人员而言，显然本发明不限于上述示范性实施例的细节，而且在不背离本发明的精神或基本特征的情况下，能够以其他的具体形式实现本发明。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本发明的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本发明内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外，显然“包括”一词不排除其他单元或，单数不排除复数。本发明陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

[0197] 最后应说明的是，以上实施例仅用以说明本发明的技术方案而非限制，尽管参照较佳实施例对本发明进行了详细说明，本领域的普通技术人员应当理解，可以对本发明的技术方案进行修改或等同替换，而不脱离本发明技术方案的精神和范围。

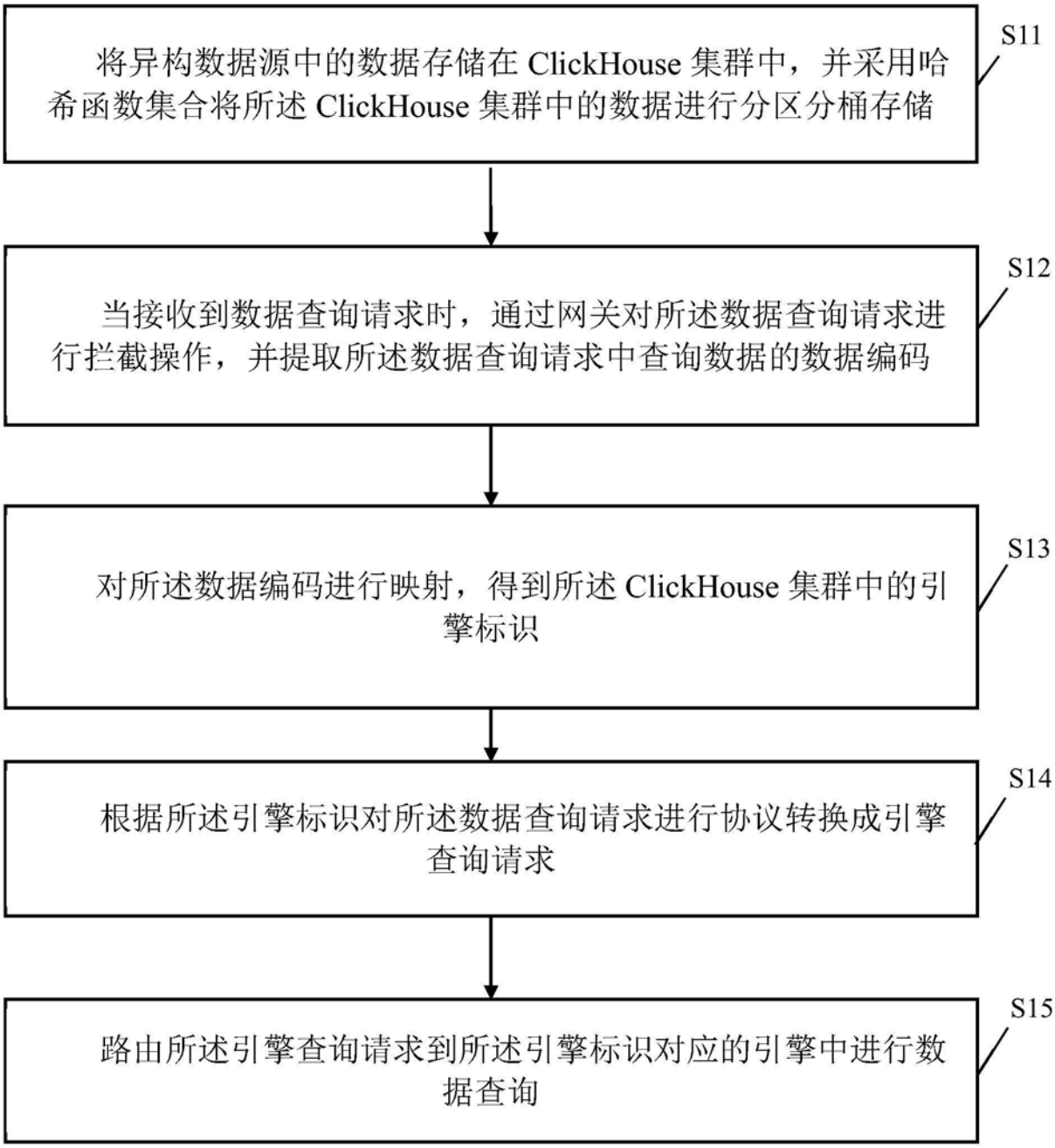

S11
将异构数据源中的数据存储在 ClickHouse 集群中，并采用哈希函数集合将所述 ClickHouse 集群中的数据进行分区分桶存储
S12
当接收到数据查询请求时，通过网关对所述数据查询请求进行拦截操作，并提取所述数据查询请求中查询数据的数据编码
S13
对所述数据编码进行映射，得到所述 ClickHouse 集群中的引擎标识
S14
根据所述引擎标识对所述数据查询请求进行协议转换成引擎查询请求
S15
路由所述引擎查询请求到所述引擎标识对应的引擎中进行数据查询

图1

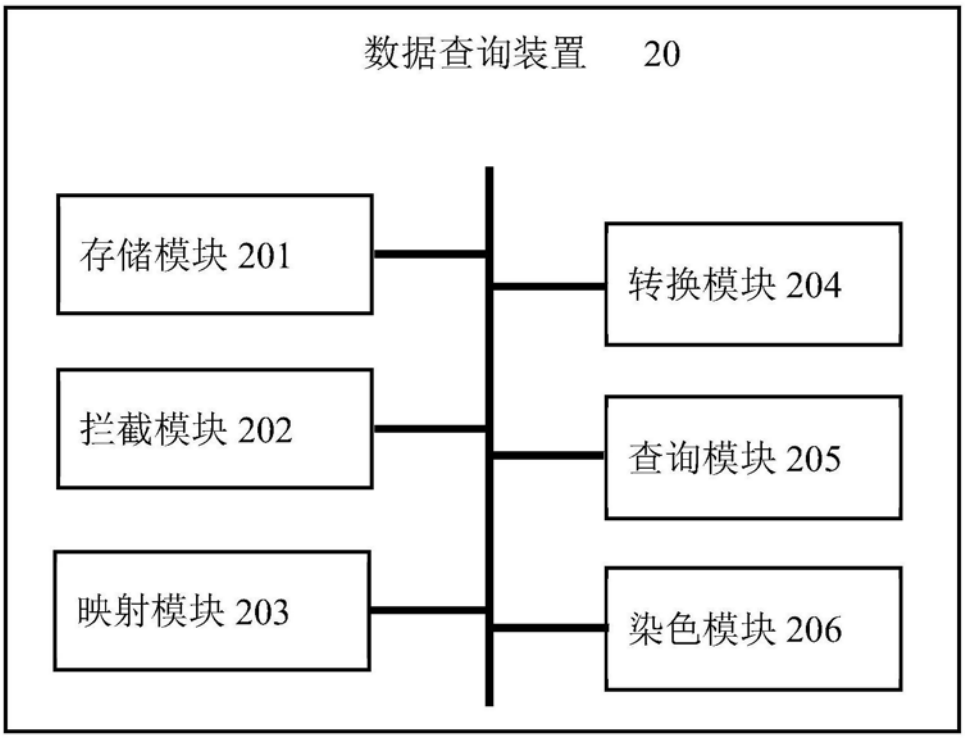

数据查询装置 20
存储模块 201
拦截模块 202
映射模块 203
转换模块 204
查询模块 205
染色模块 206

图2

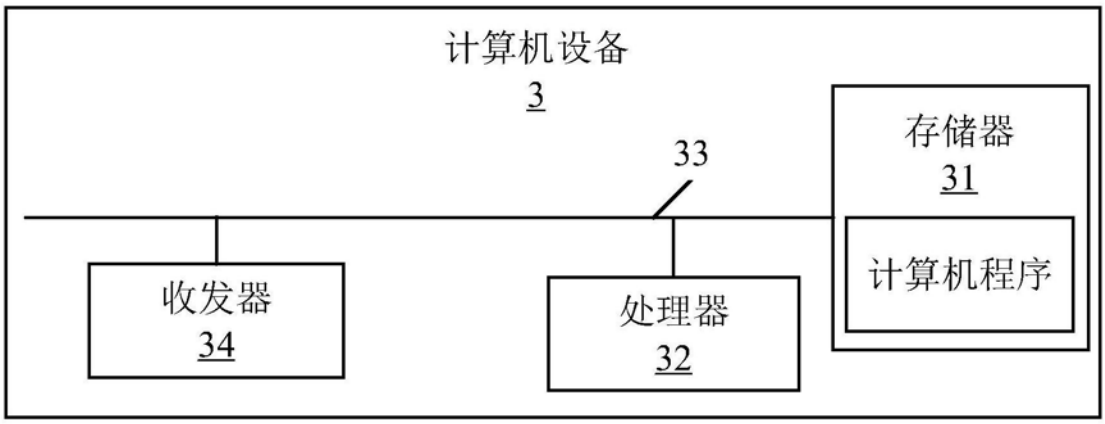

计算机设备
3
33
存储器
31
计算机程序
收发器
34
处理器
32

图3