

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2022 年 12 月 1 日 (01.12.2022)



(10) 国际公布号
WO 2022/247201 A1

- (51) 国际专利分类号:
G06F 16/242 (2019.01)
- (21) 国际申请号: PCT/CN2021/134954
- (22) 国际申请日: 2021 年 12 月 2 日 (02.12.2021)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202110572421.1 2021年5月25日 (25.05.2021) CN
- (71) 申请人: 深圳前海微众银行股份有限公司
(**WEBANK CO., LTD.**) [CN/CN]; 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。
- (72) 发明人: 王和平(**WANG, Heping**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。尹强(**YIN, Qiang**); 中

国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。黄山(**HUANG, Shan**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。杨峙岳(**YANG, Zhiyue**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。刘有(**LIU, You**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。杨永坤(**YANG, Yongkun**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。华德义(**HUA, Deyi**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。白乐(**BAI, Le**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。徐嘉杨(**XU, Jiayang**); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。饶进阳(**RAO, Jinyang**); 中国广东

(54) **Title:** DATA QUERY METHOD AND APPARATUS

(54) 发明名称: 一种数据查询方法及装置

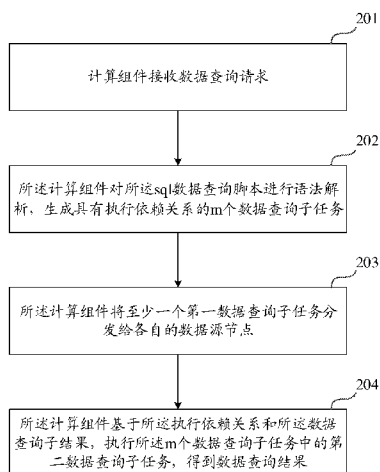


图 2

- 201 A computing assembly receives a data query request
- 202 The computing assembly performs syntax parsing on an SQL data query script to generate m data query sub-tasks having an execution dependency relationship
- 203 The computing assembly distributes at least one first data query sub-task to a respective data source node
- 204 The computing assembly executes, according to the execution dependency relationship and a data query sub-result, a second data query sub-task in the m data query sub-tasks to obtain a data query result

(57) **Abstract:** Embodiments of the present invention provide a data query method and apparatus. The method comprises: a computing assembly receives a data query request, performs syntax parsing on an SQL data query script to generate m data query sub-tasks having an execution dependency relationship, distributes at least one first data query sub-task to a respective data source node, and according to the execution dependency relationship and a data query sub-result, executes a second data query sub-task in the m data query sub-tasks to obtain a data query result. Each data source node only transmits the data query sub-result to the computing assembly, but does not transmit all the data in a database table, and therefore, the amount of data transmitted between the computing assembly and each data source node can be reduced, so that a network resource consumed during data transmission can be reduced. In addition, the solution fully uses the computing capability supported by each data source node to execute the data query sub-tasks, thereby being capable of reducing a computing resource consumed by the computing assembly.

省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。 邸帅(DI, Shuai); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。 卢道和(LU, Daohe); 中国广东省深圳市前海深港合作区前湾一路1号A栋201室, Guangdong 518027 (CN)。

(74) 代理人: 北京同达信恒知识产权代理有限公司 (TDIP & PARTNERS); 中国北京市西城区裕民路18号北环中心A座2002, Beijing 100029 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本发明实施例提供了一种数据查询方法及装置, 该方法包括计算组件接收数据查询请求, 对sql数据查询脚本进行语法解析, 生成具有执行依赖关系的m个数据查询子任务, 并将至少一个第一数据查询子任务分发给各自的数据源节点, 基于执行依赖关系和数据查询子结果, 执行m个数据查询子任务中的第二数据查询子任务, 得到数据查询结果。由于各数据源节点只是传输数据查询子结果给计算组件, 并不是传输数据库表的全量数据, 因此可以减少计算组件与各数据源节点之间的数据传输量, 从而可以降低数据传输过程中所消耗的网络资源。此外, 由于该方案充分利用各数据源节点自身支持的计算能力来执行数据查询子任务, 因此可以降低计算组件所消耗的计算资源。

一种数据查询方法及装置

相关申请的交叉引用

本申请要求在2021年05月25日提交中国专利局、申请号为202110572421.1、申请名称为“一种数据查询方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明实施例涉及金融科技（Fintech）领域，尤其涉及一种数据查询方法及装置。

背景技术

随着计算机技术的发展，越来越多的技术应用在金融领域，传统金融业正在逐步向金融科技转变，但由于金融行业的安全性、实时性要求，也对技术提出的更高的要求。在金融领域，随着金融业务的不断扩展，金融业务数据的数量变多且数据的形式也变得多样，如此针对金融业务数据应用的要求也会变高，在此基础上，数据查询在金融业务上的应用也变得广泛。因此，如何进行及时有效地数据查询操作，以满足金融业务的需求就成为急需解决的问题。

现有方案在针对客户端提交的数据查询脚本中涉及的多数据源进行混合查询计算时，通常基于数据查询引擎工具（比如 Presto 或 openLooKeng），通过各数据源对应的数据源连接器获取各数据源的数据库表的全量数据，并针对各数据源的数据库表的全量数据进行混合计算，即可得到最终的查询结果。然而，这种处理方式一方面在对各数据源的数据库表的全量数据进行混合计算时需要消耗较多的计算资源，另一方面在获取各数据源的数据库表的全量数据时需要消耗较多的网络资源。

综上，目前亟需一种数据查询方法，用以降低数据传输过程中所消耗的网络资源。

发明内容

本发明实施例提供了一种数据查询方法及装置，用以降低数据传输过程中所消耗的网络资源。

第一方面，本发明实施例提供了一种数据查询方法，包括：

计算组件接收数据查询请求；所述数据查询请求中包括结构化查询语言 sql 数据查询脚本；

所述计算组件对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务；所述 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务；

所述计算组件将至少一个第一数据查询子任务分发给各自的数据源节点；所述数据源节点用于执行第一数据查询子任务并得到数据查询子结果；

所述计算组件基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询

子任务中的第二数据查询子任务，得到数据查询结果；所述第二数据查询子任务涉及多个数据源节点。

上述技术方案中，通过对 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务，且该 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务，并将至少一个第一数据查询子任务分发给各自的数据源节点，以使各数据源节点执行各自接收到的第一数据查询子任务，得到数据查询子结果，而无需计算组件从 sql 数据查询脚本所涉及的各数据源节点中获取数据库表的全量数据在计算组件本地进行查询计算。然后，计算组件基于执行依赖关系和数据查询子结果，执行 m 个数据查询子任务中的第二数据查询子任务，即可得到所需的数据查询结果，如此，由于计算组件只是按照执行依赖关系基于数量较少的数据查询子结果（相比数据库表的全量数据，数据查询子结果的数量少很多）进行整合计算，因此可以减轻计算组件的计算压力。基于此，由于各数据源节点只是传输数据查询子结果给计算组件，并不是传输数据库表的全量数据，因此可以大大减少计算组件与各数据源节点之间的数据传输量，从而可以降低数据传输过程中所消耗的网络资源，从而可以解决现有技术中存在需要获取各数据源节点的数据库表的全量数据的问题。此外，由于该方案能够充分利用各数据源节点自身支持的计算能力来执行数据查询子任务，得到数据查询子结果，因此可以降低计算组件所消耗的计算资源，并可以减少针对计算组件的一些计算能力的开发，从而可以减少开发人员的工作量。

可选地，所述计算组件对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务，包括：

所述计算组件按照语法解析规则，生成所述 sql 数据查询脚本的语法树；

所述计算组件从所述语法树中确定仅涉及单一数据源节点的第一数据查询子任务；

所述计算组件针对所述语法树中任一表连接关键词，通过所述表连接关键词对应的第一数据查询子任务的数据查询子结果构建第二数据查询子任务；

所述计算组件根据各第一数据查询子任务和各第二数据查询子任务的执行顺序，确定具有执行依赖关系的 m 个数据查询子任务。

上述技术方案中，通过基于 sql 数据查询脚本的语法树，可以及时准确地确定出第一数据查询子任务，并将第一数据查询子任务分发给对应的数据源节点进行执行，从而可以避免从 sql 数据查询脚本所涉及的各数据源节点中获取数据库表的全量数据。此外，通过表连接关键词对应的第一数据查询子任务的数据查询子结果构建出第二数据查询子任务，以便于计算组件基于执行依赖关系和数据查询子结果来执行该第二数据查询子任务，从而可以及时有效地得到所需的数据查询结果，如此即可减轻计算组件的计算压力。

可选地，所述计算组件按照语法解析规则，生成所述 sql 数据查询脚本的语法树，包括：

所述计算组件按照语法解析规则，依次解析出所述 sql 数据查询脚本中的各关键词；

所述计算组件若确定解析出的表名关键词不符合表名命名规则，则对所述表名关键词按照所述语法解析规则继续解析，直至解析到符合表名命名规则的表名关键词，从而得到所述 sql 数据查询脚本的语法树。

上述技术方案中，在确定解析出的表名关键词不符合表名命名规则时，可对表名关键词进行进一步解析，直至解析到符合表名命名规则的表名关键词，即可得到完整清晰的语法树，并基于该语法树可以拆分出多个数据查询子任务，如此也可以确定出需要数据源节

点执行的数据查询子任务以及需要计算节点执行的数据查询子任务，从而可以为后续及时地确定出各数据源节点需要执行的数据查询子任务提供支持。

可选地，通过如下方式确定数据源节点，包括：

所述计算组件按照语法解析规则中的指定标签名，确定所述 sql 数据查询脚本中涉及的数据源节点；或，

所述计算组件按照语法解析规则中的表名规则，确定所述 sql 数据查询脚本中涉及的数据源节点。

上述技术方案中，通过按照语法解析规则中的指定标签名或照语法解析规则中的表名规则，可以及时准确地确定出 sql 数据查询脚本中涉及的数据源节点。

可选地，在生成具有执行依赖关系的 m 个数据查询子任务之后，还包括：

所述计算组件为所述 m 个数据查询子任务中的至少一个第一数据查询子任务分别标注上对应的数据源节点标识；

所述计算组件将第一数据查询子任务分发给各自的数据源节点，包括：

针对每个第一数据查询子任务，所述计算组件基于所述第一数据查询子任务对应的数据源节点标识，将所述第一数据查询子任务分发给对应的数据源节点。

上述技术方案中，通过对各第一数据查询子任务标注上对应的数据源节点标识，可以确保计算组件及时准确地将各第一数据查询子任务分发给各自对应的数据源节点，并可以避免计算组件在分发第一数据查询子任务时因无法识别第一数据查询子任务的归属而导致第一数据查询子任务分发给不匹配的数据源节点，而无法获取正确的数据查询结果。

可选地，在所述计算组件对所述 sql 数据查询脚本进行语法解析之前，还包括：

所述计算组件确定所述 sql 数据查询脚本可执行。

上述技术方案中，在对 sql 数据查询脚本进行语法解析之前，通过确定 sql 数据查询脚本是否可执行，可以确保正确的 sql 数据查询脚本能够被成功执行而获取正确的数据查询结果，并可以避免 sql 数据查询脚本因不可执行而无法获取所需的数据查询结果。

可选地，通过下述方式确定所述 sql 数据查询脚本是否可执行：

所述计算组件通过设定的 sql 数据查询脚本校验规则，对所述 sql 数据查询脚本的语法和/或参数进行校验，从而确定所述 sql 数据查询脚本是否可执行。

上述技术方案中，通过设定的 sql 数据查询脚本校验规则，对 sql 数据查询脚本的语法和/或参数进行校验，可以及时准确地确定 sql 数据查询脚本的语法和/或参数是否正确，从而可以及时有效地确定 sql 数据查询脚本是否可成功执行。

第二方面，本发明实施例还提供了一种数据查询装置，包括：

接收单元，用于接收数据查询请求；所述数据查询请求中包括结构化查询语言 sql 数据查询脚本；

处理单元，用于对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务；所述 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务；将至少一个第一数据查询子任务分发给各自的数据源节点；所述数据源节点用于执行第一数据查询子任务并得到数据查询子结果；基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询子任务中的第二数据查询子任务，得到数据查询结果；所述第二数据查询子任务涉及多个数据源节点。

可选地，所述处理单元具体用于：

按照语法解析规则，生成所述 sql 数据查询脚本的语法树；

从所述语法树中确定仅涉及单一数据源节点的第一数据查询子任务；

针对所述语法树中任一表连接关键词，通过所述表连接关键词对应的第一数据查询子任务的数据查询子结果构建第二数据查询子任务；

5 根据各第一数据查询子任务和各第二数据查询子任务的执行顺序，确定具有执行依赖关系的 m 个数据查询子任务。

可选地，所述处理单元具体用于：

按照语法解析规则，依次解析出所述 sql 数据查询脚本中的各关键词；

10 若确定解析出的表名关键词不符合表名命名规则，则对所述表名关键词按照所述语法解析规则继续解析，直至解析到符合表名命名规则的表名关键词，从而得到所述 sql 数据查询脚本的语法树。

可选地，所述处理单元具体用于：

按照语法解析规则中的指定标签名，确定所述 sql 数据查询脚本中涉及的数据源节点；
或，

15 按照语法解析规则中的表名规则，确定所述 sql 数据查询脚本中涉及的数据源节点。

可选地，所述处理单元还用于：

在生成具有执行依赖关系的 m 个数据查询子任务之后，为所述 m 个数据查询子任务中的至少一个第一数据查询子任务分别标注上对应的数据源节点标识；

所述处理单元具体用于：

20 针对每个第一数据查询子任务，基于所述第一数据查询子任务对应的数据源节点标识，将所述第一数据查询子任务分发给对应的数据源节点。

可选地，所述处理单元还用于：

在对所述 sql 数据查询脚本进行语法解析之前，确定所述 sql 数据查询脚本可执行。

可选地，所述处理单元具体用于：

25 通过设定的 sql 数据查询脚本校验规则，对所述 sql 数据查询脚本的语法和/或参数进行校验，从而确定所述 sql 数据查询脚本是否可执行。

第三方面，本发明实施例提供一种计算设备，包括至少一个处理器以及至少一个存储器，其中，所述存储器存储有计算机程序，当所述程序被所述处理器执行时，使得所述处理器执行上述第一方面任意所述的数据查询方法。

30 第四方面，本发明实施例提供一种计算机可读存储介质，其存储有可由计算设备执行的计算机程序，当所述程序在所述计算设备上运行时，使得所述计算设备执行上述第一方面任意所述的数据查询方法。

附图说明

35 为了更清楚地说明本发明实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简要介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域的普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 为本发明实施例提供的一种数据查询系统架构的示意图；

图 2 为本发明实施例提供的一种数据查询方法的流程示意图;

图 3 为本发明实施例提供的一种基准语法树的示意图;

图 4 为本发明实施例提供的一种针对 sql 数据查询脚本的语法树示意图;

图 5 为本发明实施例提供的一种依赖树的结构示意图;

5 图 6 为本发明实施例提供的另一种依赖树的结构示意图;

图 7 为本发明实施例提供的一种数据查询装置的结构示意图;

图 8 为本发明实施例提供的一种计算设备的结构示意图。

具体实施方式

10 为了使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明作进一步地详细描述,显然,所描述的实施例仅仅是本发明的一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例,都属于本发明保护的范围。

15 为了便于理解本发明实施例,首先以图 1 中示出的系统结构为例说明适用于本发明实施例的数据查询系统架构。该数据查询系统架构可以应用于参与涉及多数据源的混合查询等。如图 1 所示,该数据查询系统架构可以包括客户端 100、计算组件 200 和至少一个数据源节点(比如数据源节点 301、数据源节点 302 和数据源节点 303 等)。其中,计算组件 200 可以包括任务接收校验模块 201、任务解析编排模块 202 和任务分发模块 203。客户端 100 与计算组件 200 进行连接,每个数据源节点可以与计算组件 200 进行连接,比如,可以
20 可以通过有线方式连接,或者可以通过无线方式连接,本发明实施例对此并不作限定。此外,每个数据源节点可以包括至少一个数据库引擎。示例性地,比如数据源节点 301 可以包括 SparkEngine、HiveEngine 等数据库引擎;数据源节点 302 可以包括 HbaseEngine、HiveEngine 等数据库引擎;数据源节点 303 可以包括 SparkEngine、HbaseEngine 等数据库引擎。

其中,客户端 100 用于在接收到用户提交的数据查询脚本后,将该数据查询脚本发送
25 给计算组件 200。其中,客户端 100 可以为终端上的客户端软件,该终端可以为手机、笔记本电脑、台式电脑、平板电脑等,本发明实施例对此并不作限定。

计算组件 200 用于针对客户端 100 发送的结构化查询语言 sql 数据查询脚本进行语法解析,获取数据源信息,若该数据源信息涉及多个数据源,则可以将该数据查询脚本拆分为多个 sql 数据查询子脚本,并将需要发送给数据源节点执行的至少一个 sql 数据查询子脚本
30 分发给各自对应的数据源节点。然后,按照该多个 sql 数据查询子脚本的查询依赖关系,将各数据源节点执行各自对应的数据查询子脚本后所得到的数据查询子结果进行整合处理,以便得到最终的目标数据查询结果。此外,在具体实施过程中,任务接收校验模块 201 首先对接收到的 sql 数据查询脚本进行校验,在校验成功后,通过任务解析编排模块 202 对该 sql 数据查询脚本进行语法解析,获取数据源信息,若该数据源信息涉及多个数据源,
35 则可以将该数据查询脚本拆分为多个 sql 数据查询子脚本,并通过任务分发模块 203 将该多个 sql 数据查询子脚本分发给各自对应的数据源节点。其中,计算组件(比如 Linkis 等) 200 提供了强大的连通、复用、编排、扩展和治理管控能力。通过计算组件 200 可以将应用层和引擎层解耦,简化了复杂的网络调用关系,降低了整体复杂度,同时节约了整体开发和维护成本。

针对每个数据源节点，该数据源节点在接收到对应的 sql 数据查询子脚本后，执行该 sql 数据子查询脚本，得到数据查询子结果，并将数据查询子结果进行存储，比如将数据查询子结果存储至上下文 context 中。

需要说明的是，上述图 1 所示的结构仅是一种示例，本发明实施例对此不做限定。

5 基于上述描述，图 2 示例性的示出了本发明实施例提供的一种数据查询方法的流程，该流程可以由数据查询装置执行。

如图 2 所示，该流程具体包括：

步骤 201，计算组件接收数据查询请求。

10 步骤 202，所述计算组件对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务。

步骤 203，所述计算组件将至少一个第一数据查询子任务分发给各自的数据源节点。

步骤 204，所述计算组件基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询子任务中的第二数据查询子任务，得到数据查询结果。

15 上述步骤 201 中，计算组件可以接收用户在终端上的客户端上的数据查询请求。比如，用户可以通过手机上的客户端提交数据查询请求，或者可以通过笔记本电脑上的 web（World Wide Web，万维网）界面提交数据查询请求，或者，可以通过计算组件所在系统提供的服务界面上直接实时编辑 sql 数据查询脚本，以生成数据查询请求，并将该数据查询请求在该服务界面进行上传。其中，数据查询请求可以包括 sql 数据查询脚本。

20 上述步骤 202 中，计算组件对 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务。其中，m 个数据查询子任务中至少可以包括一个仅涉及单一数据源节点的第一数据查询子任务；每个数据查询子任务中可以包括 sql 数据查询子脚本，也即是说，数据源节点在执行第一数据查询子任务时，执行的是第一数据查询子任务中的 sql 数据查询子脚本；或者计算组件在执行第二数据查询子任务时，执行的是第二数据查询子任务中的 sql 数据查询子脚本。具体地，计算组件按照语法解析规则，生成 sql 数据查询脚本的语法树，并从该语法树中确定出仅涉及单一数据源节点的第一数据查询子任务。再针对语法树中任一表连接关键词，通过表连接关键词对应的第一数据查询子任务的数据查询子结果构建第二数据查询子任务。然后，根据各第一数据查询子任务和各第二数据查询子任务的执行顺序，确定具有执行依赖关系的 m 个数据查询子任务。其中，表连接关键词可以包括 left join、right join、inner Join/join、full join 等。如此，通过基于 sql 数据查询脚本的语法树，可以及时准确地确定出第一数据查询子任务，并将第一数据查询子任务分发给对应的数据源节点进行执行，从而可以避免从 sql 数据查询脚本所涉及的各数据源节点中获取数据库表的全量数据。此外，通过表连接关键词对应的第一数据查询子任务的数据查询子结果构建出第二数据查询子任务，以便于计算组件基于执行依赖关系和数据查询子结果来执行该第二数据查询子任务，从而可以及时有效地得到所需的数据查询结果，如此即可减轻计算组件的计算压力。

35 示例性地，针对某一 sql 数据查询脚本，通过 sql 数据查询脚本进行语法解析，生成 sql 数据查询脚本的语法树，并从该语法树中确定出两个数据查询子任务，即数据查询子任务 a 和数据查询子任务 b。再通过表连接关键词（比如 join）对应的数据查询子任务 a 的执行结果和数据查询子任务 b 的执行结果构建出数据查询子任务 c。其中，数据查询子任务 c 的执行需要依赖数据查询子任务 a 的执行结果以及数据查询子任务 b 的数据查询子结果，

比如数据查询子任务是在数据源节点 a 中执行的, 数据查询子任务 b 是在数据源节点 b 中执行的, 而数据查询子任务 c 是在计算组件中执行的。

进一步地, 在生成 sql 数据查询脚本的语法树时, 可以按照语法解析规则, 依次解析出所述 sql 数据查询脚本中的各关键词, 若确定解析出的表名关键词不符合表名命名规则, 则对表名关键词按照语法解析规则继续解析, 直至解析到符合表名命名规则的表名关键词, 从而得到 sql 数据查询脚本的语法树。基于此, 在确定解析出的表名关键词不符合表名命名规则时, 可对表名关键词进行进一步解析, 直至解析到符合表名命名规则的表名关键词, 即可得到完整清晰的语法树, 并基于该语法树可以拆分出多个数据查询子任务, 如此也可以确定出需要数据源节点执行的数据查询子任务以及需要计算节点执行的数据查询子任务, 从而可以为后续及时地确定出各数据源节点需要执行的数据查询子任务提供支持。

此外, 在对 sql 数据查询脚本进行语法解析之前, 需要确定 sql 数据查询脚本是否可执行, 以便确保正确的 sql 数据查询脚本能够被成功执行而获取正确的数据查询结果, 并可以避免 sql 数据查询脚本因不可执行而无法获取所需的数据查询结果。具体地, 计算组件通过设定的 sql 数据查询脚本校验规则, 对 sql 数据查询脚本的语法和/或参数进行校验, 从而确定 sql 数据查询脚本是否可执行。如此, 可以及时准确地确定 sql 数据查询脚本的语法和/或参数是否正确, 从而可以及时有效地确定 sql 数据查询脚本是否可成功执行。

上述步骤 203 中, 计算组件将至少一个第一数据查询子任务分发给各自的数据源节点, 以使各数据源节点执行接收到的第一数据查询子任务并得到数据查询子结果。具体地, 在生成具有执行依赖关系的 m 个数据查询子任务之后, 计算组件为 m 个数据查询子任务中的至少一个第一数据查询子任务分别标注上对应的数据源节点标识。针对每个第一数据查询子任务, 计算组件基于所述第一数据查询子任务对应的数据源节点标识, 将第一数据查询子任务分发给对应的数据源节点。如此, 通过对各第一数据查询子任务标注上对应的数据源节点标识, 可以确保计算组件及时准确地将各第一数据查询子任务分发给各自对应的数据源节点, 并可以避免计算组件在分发第一数据查询子任务时因无法识别第一数据查询子任务的归属而导致第一数据查询子任务分发给不匹配的数据源节点, 而无法获取正确的数据查询结果。

此外, 可以通过下述方式确定数据源节点: 计算组件按照语法解析规则中的指定标签名, 可以及时准确地确定 sql 数据查询脚本中涉及的数据源节点; 或者, 可以按照语法解析规则中的表名规则, 可以及时准确地确定 sql 数据查询脚本中涉及的数据源节点。

示例性地, 假设有两个第一数据查询子任务, 即数据查询子任务 a 和数据查询子任务 b。其中, 数据查询子任务 a 是发给数据源节点 a (比如 IDC_a) 进行执行, 则可以为数据查询子任务 a 标注上 IDC_a 的标识; 数据查询子任务 b 是发给数据源节点 b (比如 IDC_b) 进行执行, 则可以为数据查询子任务 b 标注上 IDC_b 的标识。然后, 在分发各第一数据查询子任务时, 即可及时准确地将数据查询子任务 a 发送给 IDC_a 对应的数据源节点 a, 可以及时准确地将数据查询子任务 b 发送给 IDC_b 对应的数据源节点 b。

上述步骤 204 中, 计算组件基于执行依赖关系和数据查询子结果, 执行 m 个数据查询子任务中的第二数据查询子任务, 得到数据查询结果。其中, 第二数据查询子任务涉及多个数据源节点。

示例性地, 针对某一 sql 数据查询脚本, 通过 sql 数据查询脚本进行语法解析, 生成具有依赖关系的三个数据查询子任务, 即数据查询子任务 a、数据查询子任务 b 和数据查询

子任务 c。其中，数据查询子任务 c 的执行需要依赖数据查询子任务 a 的执行结果以及数据查询子任务 b 的数据查询子结果。如此，计算组件即可基于数据查询子任务 a、数据查询子任务 b 和数据查询子任务 c 之间的依赖关系，以及数据查询子任务 a 的执行结果和数据查询子任务 b 的执行结果，执行数据查询子任务 c，从而可以准确地得到用户所需的数据查询结果。

基于此，下面对本发明实施例中数据查询方法的实施过程进行具体描述。

Step1: 计算组件针对客户端提交的数据查询脚本进行校验。

其中，计算组件可以设置在一台独立的物理机上，或者可以设置在多个物理服务器构成的服务器集群或者分布式系统上，本发明实施例对此并不作限定。

具体实施过程中，用户在需要进行数据查询时，可以通过终端上的客户端向计算组件中的任务接收校验模块提交数据查询请求，该数据查询请求中可以包括已经预先编辑好的 sql 数据查询脚本。计算组件在接收到数据查询请求后，从数据查询请求中解析出 sql 数据查询脚本。或者，用户可以在计算组件所在系统提供的服务界面上直接实时编辑 sql 数据查询脚本或直接输入已预先编辑好的 sql 数据查询脚本。其中，计算组件所在系统提供的服务界面可以通过终端上的客户端进行显示。然后，计算组件中的任务接收校验模块会对用户提交的 sql 数据查询脚本进行校验，以确定用户提交的 sql 数据查询脚本是否可成功执行，比如，对 sql 数据查询脚本的语法和/或参数等进行校验。

Step2: 计算组件针对校验通过的 sql 数据查询脚本进行语法解析，生成至少一个数据查询子脚本。

具体地，计算组件中的任务解析编排模块在接收到 sql 数据查询脚本后，按照 sql 语法解析规则，对该 sql 数据查询脚本进行语法解析，解析出该 sql 数据查询脚本所涉及的至少一个数据源，再基于该至少一个数据源将该 sql 数据查询脚本划分为多个步骤执行。也即是，基于该至少一个数据源将该 sql 数据查询脚本划分为多个 sql 数据查询子脚本，并针对该多个 sql 数据查询子脚本中需要发送给数据源节点执行的至少一个 sql 数据查询子脚本，将需要发送给数据源节点的至少一个 sql 数据查询子脚本分别生成对应的 sql 数据查询子任务，将该多个 sql 数据查询子脚本中不需要发送给数据源节点的至少一个 sql 数据查询子脚本分别生成对应的 sql 数据查询子任务保存在本地。同时将需要发送给数据源节点的 sql 数据查询子任务标注上对应的数据源标签，比如标注上 IDC (Internet Data Center, 数据中心, 也可称为机房或计算中心) 1 或 IDC2 等，当然，也可以标注上具体执行 sql 数据查询子任务的数据库引擎，比如标注上 Engine1 或 Engine2 等。比如，Engine1 可以用于表示数据库引擎 SparkEngine 或 HiveEngine 等，Engine2 可以用于表示数据库引擎 HiveEngine 或 SparkEngine、等；或者，Engine1 可以用于表示数据库引擎 SparkEngine 或 HbaseEngine 等，Engine2 可以用于表示数据库引擎 HbaseEngine 或 SparkEngine 等；或者，Engine1 可以用于表示数据库引擎 HiveEngine 或 HbaseEngine 等，Engine2 可以用于表示数据库引擎 HbaseEngine 或 HiveEngine 等，本发明实施例对此并不作限定。示例性地，假设有两个 sql 数据查询子任务 (比如 sql 数据查询子任务 A 和 sql 数据查询子任务 B) 需要发送给对应的数据源节点，即，sql 数据查询子任务 A 需要发送给数据源节点 A (比如 IDC1_Engine1)，sql 数据查询子任务 B 需要发送给数据源节点 B (比如 IDC2_Engine2)，则可以为 sql 数据查询子任务 A 标注上数据源标签 IDC1_Engine1，为 sql 数据查询子任务 B 标注上 IDC2_Engine2。

示例性地, 假设某一 sql 数据查询脚本所涉及两个数据源, 即数据源 A 和数据源 B。基于该两个数据源可以将该 sql 数据查询脚本划分为多个数据查询步骤(比如 stage1、stage2 和 stage3) 执行。也即是, 基于该两个数据源将该 sql 数据查询脚本划分为对应的多个 sql 数据查询子脚本, 即, 基于数据源 A 和数据源 B, 从该 sql 数据查询脚本中划分出 sql 数据查询子脚本 A、sql 数据查询子脚本 B 以及 sql 数据查询子脚本 C。基于此, 相比现有方案中需要在数据查询引擎工具中执行整个 sql 数据查询脚本, 并需要读取该 sql 数据查询脚本所涉及的至少一个数据源的数据库表中的全量数据, 而本方案不需要读取各数据源节点的数据库表中的全量数据, 将整个 sql 数据查询脚本拆分为多个 sql 数据查询子脚本, 并将需要各数据源节点执行的至少一个 sql 数据查询子脚本分发给对应的数据源节点, 依靠各数据源节点自身支持的计算能力来执行对应 sql 数据查询子脚本, 可以使得各数据源节点自身支持的计算能力得到充分利用, 降低计算组件所消耗的计算资源, 并可以减少针对计算组件的一些计算能力的开发。

下面对基于 sql 语法解析规则对 sql 数据查询脚本进行解析的具体过程进行描述。

步骤 a、计算组件中的任务解析编排模块完成针对 sql 数据查询脚本的数据源解析。

具体地, 可以通过两种实现方式来获取 sql 数据查询脚本所涉及的数据源信息。

第一种实现方式为: 用户在提交的数据查询请求中指定对应的数据源标签信息, 则可以直接通过数据源标签信息获取到对应的数据源信息。也即是说, 用户需要在数据源标签信息中指定涉及到的数据库表名和数据源信息。

示例性地, 比如用户提交的数据查询请求中, 包括某一 sql 数据查询脚本: select * from tableA join tableB on tableA.c1=tableB.c2, 以及该 sql 数据查询脚本所涉及的数据源标签信息为 tableA-IDC1_Hive-DB1 和 tableB-IDC2_Hbase-DB1。其中, 该数据源标签信息的含义为: 数据库表名-数据源名-数据库名。即, tableA 用于表示数据库表名, IDC1_Hive 用于表示数据源名, DB1 用于表示 Hive 中的数据库名; tableB 用于表示数据库表名, IDC2_Hbase 用于表示数据源名, DB1 用于表示 Hbase 中的数据库名。

第二种实现方式为: 用户在提交的 sql 数据查询脚本中的数据库表名中指定了对应的数据源信息, 则可以通过解析 sql 数据查询脚本获取对应的数据源信息。也即是说, 需要用户在数据库表名中指定数据源名、数据库名等信息。

示例性地, 比如用户提交的某一 sql 数据查询脚本为: select * from IDC1_Hive.DB1.tableA tableA join IDC2_Hbase.DB1.tableB tableB on tableA.c1=tableB.c2。其中, 用户在 sql 数据查询脚本中指定包含有数据源信息的表名规则为: 数据源名.数据库名.数据库表名。即, 在该 sql 数据查询脚本所涉及的第一个数据源中, IDC1_Hive 用于表示数据源名, DB1 用于表示 Hive 中的数据库名, tableA 用于表示数据库表名; 在该 sql 数据查询脚本所涉及的第二个数据源中, IDC2_Hbase 用于表示数据源名, DB1 用于表示 Hbase 中的数据库名, tableB 用于表示数据库表名。

通过上述两种实现方式获取到 sql 数据查询脚本所涉及的数据源信息后, 会和计算组件中的数据源模块 Datasource (图 1 中未示出) 进行映射, 获取到对应的数据源的真实信息。

步骤 b、计算组件中的任务解析编排模块生成针对 sql 数据查询脚本的依赖树。

在获取到 sql 数据查询脚本所涉及的数据源信息后, 若该数据源信息只涉及一个数据源, 即, 该 sql 数据查询脚本并不涉及数据混合查询, 则不需要生成针对 sql 数据查询脚本

的依赖树，可直接基于该 sql 数据查询脚本生成对应的 sql 数据查询任务，并将该 sql 数据查询任务分发给对应的数据源节点，以便数据源节点执行该 sql 数据查询任务中的 sql 数据查询脚本，得到数据查询结果并进行缓存。

若该数据源信息涉及两个或两个以上数据源，则该 sql 数据查询脚本涉及数据混合查询，需要生成针对 sql 数据查询脚本的依赖树。即，通过基于 sql 数据查询脚本所涉及的多个数据源，将 sql 数据查询脚本进行解析拆分，确定出多个 sql 数据查询子脚本。具体地，计算组件中的任务解析编排模块可以按照 sql 语法解析规则，sql 数据查询脚本进行解析拆分，以此可得到多个 sql 数据查询子脚本。其中，参考图 3，图 3 为本发明实施例提供的一种基准语法树的示意图。sql 语法解析规则采用的是自顶向下的分析方法去做词法分析，词法分析工具将按照 select、from、join、where 等 token 对 sql 数据查询脚本进行解析。基于图 3 所示，从左到右按字符输入，并按照最左推导为终结符，将 sql 数据查询脚本解析为对应的语法树。示例性地，以用户提交的某一 sql 数据查询脚本：select * from IDC1_Hive.DB1. tableA tableA join IDC2_Hbase.DB1. tableB tableB on tableA.c1=tableB.c2 where tableA.id>1 and tableB.id>1 为例进行描述。词法分析工具按照 select、from、join、where 等 token 对该 sql 数据查询脚本进行解析，先解析出对应的数据源信息（比如 IDC1_Hive 和 IDC2_Hbase），再解析出对应的查询条件（比如 where tableA.id>1 and tableB.id>1），可以得到对应的如图 4 所示的语法树。然后，基于该 sql 数据查询脚本对应的语法树，即可生成如图 5 所示的涉及多个数据查询步骤（即多个 sql 数据查询子脚本）的依赖树。基于图 5，计算组件中的任务解析编排模块通过基于 sql 数据查询脚本所涉及的多个数据源，将 sql 数据查询脚本进行解析拆分，确定出多个数据查询步骤，即 Stage1、Stage2 和 Stage3。其中，Stage1 为执行 sql 数据查询子脚本 A（即执行 Select * from IDC1_Hive.DB1. tableA tableA where tableA.id>1）；Stage2 为执行 sql 数据查询子脚本 B（即执行 Select * from IDC2_Hbase.DB1. tableB tableB where tableB.id>1）；Stage3 为执行 sql 数据查询子脚本 C（即执行 Select * from CSTableA a join CSTableB b on a.c1=b.c1）。需要说明的是，Stage1 是在数据源节点 IDC1_Hive 中执行，Stage2 是在数据源节点 IDC2_Hbase 中执行，Stage3 是在计算组件中执行。而且，Stage3 是基于 Stage1 的执行结果 CSTableA 以及 Stage2 的执行结果 CSTableB 在计算组件中执行的。

进一步地，下面通过一条较为复杂的 sql 数据查询脚本来对 sql 数据查询脚本的解析过程进行描述。

示例性地，该较为复杂的 sql 数据查询脚本是将数据源 ES 中 XX 城市（比如城市 A）的成年人与数据源 Hive 中 XX 城市（比如城市 A）的成年人名字相同的人查询出来。其中，该较为复杂的 sql 数据查询脚本为：

```
select * from
(
select name,age,area
  from IDC1_ES.ES1.tableA tableA where area='城市 A'
) as es_tableA
join
(
select tableB.name,tableB.age,tableC.area
```

```

from IDC2_Hive1.hive1. tableB tableB
join IDC2_Hive2.hive2. tableC tableC on tableB.name=tableC.name
where tableC.area='城市 A'
) as hive_tableB

```

```

5 on es_tableA.name= hive_tableB.name
where es_tableA.age>18 and hive_tableB.age>18

```

针对上述较为复杂的 sql 数据查询脚本，词法分析工具按照 select、from、join、where 等 token 对该较为复杂的 sql 数据查询脚本进行解析，先解析出对应的数据源信息（比如 IDC1_ES、IDC2_Hive1 和 IDC2_Hive2），再解析出对应的查询条件（比如 where es_tableA.age>18 and hive_tableB.age>18），可以得到对应的语法树。然后，基于该较为复杂的 sql 数据查询脚本对应的语法树，即可生成如图 6 所示的涉及多个数据查询步骤（即多个 sql 数据查询子脚本）的依赖树。基于图 6，计算组件中的任务解析编排模块通过基于该较为复杂的 sql 数据查询脚本所涉及的多个数据源，将该较为复杂的 sql 数据查询脚本进行解析拆分，确定出多个数据查询步骤，即 Stage1、Stage2、Stage3、Stage4 和 Stage5。具体地，首先通过该较为复杂的 sql 数据查询脚本的外层的 join 条件解析出两个数据查询步骤（即 Stage1 和 Stage*），并接着解析出最外层的 where 条件，再下推到对应的 join 两边的子查询里。然后，接着由于第二个数据查询步骤（即 Stage*）里还存在混合查询的 join，继续解析生成两个数据查询步骤（即 Stage2 和 Stage3），在执行 Stage2 和 Stage3 时将 Stage2 和 Stage3 的混合查询结果存储至 CSTableD，以便 Stage4 基于该 CSTableD 执行 sql 数据查询子脚本 4。最后，Stage5 基于 Stage1 的执行结果以及 Stage4 的执行结果得到最终的目标数据查询结果。其中，Stage1 为执行 sql 数据查询子脚本 1（即执行 select name,age,area from IDC1_ES.ES1.tableA tableA where tableA.area='城市 A' and tableA.age>18）；Stage2 为执行 sql 数据查询子脚本 2（即执行 Select tableB.name,tableB.age from IDC2_Hive1.hive1. tableB tableB where tableB.age>18）；Stage3 为执行 sql 数据查询子脚本 3（即执行 Select tableC.name,tableC.area from IDC2_Hive2.hive2. tableC tableC where tableC.area='城市 A'）；Stage4 为执行 sql 数据查询子脚本 4（即执行 Select B.name,B.age,C.area from CSTableB B join CSTableC C on B.name=C.name）；Stage5 为执行 sql 数据查询子脚本 5（即执行 Select A.name,A.age,A.area from CSTableA A join CSTableD D on A.name=D.name）。需要说明的是，Stage1 是在数据源节点 IDC1_ES 中执行，Stage2 是在数据源 IDC2_Hive 中执行，Stage3 是在 IDC2_Hive 中执行，Stage4 和 Stage5 是在计算组件中执行。而且，Stage4 是基于 Stage2 的执行结果 CSTableB 以及 Stage3 的执行结果 CSTableC 在计算组件中执行的；Stage5 是基于 Stage1 的执行结果 CSTableA 以及 Stage4 的执行结果 CSTableD 在计算组件中执行的。

Step3: 计算组件将至少一个 sql 数据查询子脚本封装为对应的数据查询子任务，并将至少一个数据查询子任务分发给各自对应的数据源节点。

继续以上述图 5 所示的示例为例进行描述，将 Stage1 和 Stage2 分别封装为对应的 sql 数据查询子任务，即 sql 数据查询子任务 A 和 sql 数据查询子任务 B，并为每个 sql 数据查询子任务标注上对应的数据源标签以及数据库引擎标签，比如，为 sql 数据查询子任务 A 标注上 IDC1_Hive，为 sql 数据查询子任务 B 标注上 IDC2_Hbase。然后，将 sql 数据查询子任务 A 分发给对应的数据源节点 IDC1_Hive，将 sql 数据查询子任务 B 分发给对应的数据源节点 IDC2_Hbase。同时，将 sql 数据查询子脚本 C 生成对应的 sql 数据查询子任务 C

存储在计算组件本地，并在检测到 sql 数据查询子任务 A 和 sql 数据查询子任务 B 在各自的数据源节点执行完成后（或者各数据源节点可以在执行各自的 sql 数据查询子任务完成后，将执行完成的消息通知计算组件，或者各数据源节点可以在执行各自的 sql 数据查询子任务完成后，将各自的执行结果发送给计算组件），再基于 sql 数据查询子任务 A 的执行结果以及 sql 数据查询子任务 B 的执行结果执行 sql 数据查询子任务 C，得到最终的目标数据查询结果。或者，继续以上述图 6 所示的示例为例进行描述，将 Stage1、Stage2 和 Stage3 分别封装为对应的 sql 数据查询子任务，即 sql 数据查询子任务 1、sql 数据查询子任务 2 和 sql 数据查询子任务 3，并为每个 sql 数据查询子任务标注上对应的数据源标签以及数据库引擎标签，比如，为 sql 数据查询子任务 1 标注上 IDC1_ES，为 sql 数据查询子任务 2 标注上 IDC2_Hive1，为 sql 数据查询子任务 3 标注上 IDC2_Hive2。然后，将 sql 数据查询子任务 1 分发给对应的数据源节点 IDC1_ES，将 sql 数据查询子任务 2 分发给对应的数据源节点 IDC2_Hive1，将 sql 数据查询子任务 3 分发给对应的数据源节点 IDC2_Hive2。同时，将 sql 数据查询子脚本 4 和 sql 数据查询子脚本 5 分别生成对应的 sql 数据查询子任务 4、sql 数据查询子任务 5 保存在计算组件本地，并在检测到 sql 数据查询子任务 1、sql 数据查询子任务 2 和 sql 数据查询子任务 3 在各自的数据源节点执行完成后（或者各数据源节点可以在执行各自的 sql 数据查询子任务完成后，将执行完成的消息通知计算组件，或者各数据源节点可以在执行各自的 sql 数据查询子任务完成后，将各自的执行结果发送给计算组件），再基于 sql 数据查询子任务 2 的执行结果以及 sql 数据查询子任务 3 的执行结果执行 sql 数据查询子任务 4，然后在检测到 sql 数据查询子任务 4 执行完成后，基于 sql 数据查询子任务 1 的执行结果以及 sql 数据查询子任务 4 的执行结果执行 sql 数据查询子任务 5，得到最终的目标数据查询结果。基于此，相比现有方案中的数据查询引擎工具需要去对应的数据源节点中读取对应的数据库表的全量数据后，再基于该读取的数据库表的全量数据进行计算，而本方案利用各数据源节点自身支持的计算能力来执行 sql 数据查询子脚本，有助于降低计算组件所消耗的计算资源，并可以减少读取数据库表的全量数据的传输过程中所消耗的网络资源。

Step4: 每个数据源节点接收对应的数据查询子任务，并执行数据查询子任务中的 sql 数据查询子脚本。

针对每个数据源节点，数据源节点在接收到数据查询子任务后，解析该数据查询子任务，得到对应的 sql 数据查询子脚本，然后执行该 sql 数据查询子脚本。示例性地，继续以上述图 5 所示的示例为例进行描述，数据源节点 IDC1_Hive 在接收到 sql 数据查询子任务 A 后，解析该 sql 数据查询子任务 A，得到 sql 数据查询子脚本 A，然后执行该 sql 数据查询子脚本 A，并将执行后的执行结果存储至上下文 CSTableA 中。数据源节点 IDC2_Hbase 在接收到 sql 数据查询子任务 B 后，解析该 sql 数据查询子任务 B，得到 sql 数据查询子脚本 B，然后执行该 sql 数据查询子脚本 B，并将执行后的执行结果存储至上下文 CSTableB 中。然后，计算组件基于 sql 数据查询子任务 A 的执行结果以及 sql 数据查询子任务 B 的执行结果执行 sql 数据查询子任务 C，得到最终的目标数据查询结果。或者，继续以上述图 6 所示的示例为例进行描述，数据源节点 IDC1_ES 在接收到 sql 数据查询子任务 1 后，解析该 sql 数据查询子任务 1，得到 sql 数据查询子脚本 1，然后执行该 sql 数据查询子脚本 1，并将执行后的执行结果存储至上下文 CSTableA 中。数据源节点 IDC2_Hive1 在接收到 sql 数据查询子任务 2 后，解析 sql 数据查询子任务 2，得到 sql 数据查询子脚本 2，然

后执行该 sql 数据查询子脚本 2，并将执行后的执行结果分别存储至上下文 CStableB。数据源节点 IDC2_Hive2 在接收到 sql 数据查询子任务 3 后，解析 sql 数据查询子任务 3，得到 sql 数据查询子脚本 3，然后执行该 sql 数据查询子脚本 3，并将执行后的执行结果分别存储至上下文 CStableC 中。然后，计算组件基于 sql 数据查询子任务 2 的执行结果以及 sql 数据查询子任务 3 的执行结果执行 sql 数据查询子任务 4，并在检测到 sql 数据查询子任务 4 执行完成后，基于 sql 数据查询子任务 1 的执行结果以及 sql 数据查询子任务 4 的执行结果执行 sql 数据查询子任务 5，得到最终的目标数据查询结果。

上述实施例表明，通过对 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务，且该 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务，并将至少一个第一数据查询子任务分发给各自的数据源节点，以使各数据源节点执行各自接收到的第一数据查询子任务，得到数据查询子结果，而无需计算组件从 sql 数据查询脚本所涉及的各数据源节点中获取数据库表的全量数据在计算组件本地进行查询计算。然后，计算组件基于执行依赖关系和数据查询子结果，执行 m 个数据查询子任务中的第二数据查询子任务，即可得到所需的数据查询结果，如此，由于计算组件只是按照执行依赖关系基于数量较少的数据查询子结果（相比数据库表的全量数据，数据查询子结果的数量少很多）进行整合计算，因此可以减轻计算组件的计算压力。基于此，由于各数据源节点只是传输数据查询子结果给计算组件，并不是传输数据库表的全量数据，因此可以大大减少计算组件与各数据源节点之间的数据传输量，从而可以降低数据传输过程中所消耗的网络资源，从而可以解决现有技术中存在需要获取各数据源节点的数据库表的全量数据的问题。此外，由于该方案能够充分利用各数据源节点自身支持的计算能力来执行数据查询子任务，得到数据查询子结果，因此可以降低计算组件所消耗的计算资源，并可以减少针对计算组件的一些计算能力的开发，从而可以减少开发人员的工作量。

基于相同的技术构思，图 7 示例性的示出了本发明实施例提供的一种数据查询装置，该装置可以执行数据查询方法的流程。

如图 7 所示，该装置包括：

接收单元 701，用于接收数据查询请求；所述数据查询请求中包括结构化查询语言 sql 数据查询脚本；

处理单元 702，用于对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务；所述 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务；将至少一个第一数据查询子任务分发给各自的数据源节点；所述数据源节点用于执行第一数据查询子任务并得到数据查询子结果；基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询子任务中的第二数据查询子任务，得到数据查询结果；所述第二数据查询子任务涉及多个数据源节点。

可选地，所述处理单元 702 具体用于：

按照语法解析规则，生成所述 sql 数据查询脚本的语法树；

从所述语法树中确定仅涉及单一数据源节点的第一数据查询子任务；

针对所述语法树中任一表连接关键词，通过所述表连接关键词对应的第一数据查询子任务的数据查询子结果构建第二数据查询子任务；

根据各第一数据查询子任务和各第二数据查询子任务的执行顺序，确定具有执行依赖关系的 m 个数据查询子任务。

可选地, 所述处理单元 702 具体用于:

按照语法解析规则, 依次解析出所述 sql 数据查询脚本中的各关键词;

若确定解析出的表名关键词不符合表名命名规则, 则对所述表名关键词按照所述语法解析规则继续解析, 直至解析到符合表名命名规则的表名关键词, 从而得到所述 sql 数据查询脚本的语法树。

可选地, 所述处理单元 702 具体用于:

按照语法解析规则中的指定标签名, 确定所述 sql 数据查询脚本中涉及的数据源节点; 或,

按照语法解析规则中的表名规则, 确定所述 sql 数据查询脚本中涉及的数据源节点。

可选地, 所述处理单元 702 还用于:

在生成具有执行依赖关系的 m 个数据查询子任务之后, 为所述 m 个数据查询子任务中的至少一个第一数据查询子任务分别标注上对应的数据源节点标识;

所述处理单元 702 具体用于:

针对每个第一数据查询子任务, 基于所述第一数据查询子任务对应的数据源节点标识, 将所述第一数据查询子任务分发给对应的数据源节点。

可选地, 所述处理单元还用于:

在对所述 sql 数据查询脚本进行语法解析之前, 确定所述 sql 数据查询脚本可执行。

可选地, 所述处理单元 702 具体用于:

通过设定的 sql 数据查询脚本校验规则, 对所述 sql 数据查询脚本的语法和/或参数进行校验, 从而确定所述 sql 数据查询脚本是否可执行。

基于相同的技术构思, 本发明实施例还提供了一种计算设备, 如图 8 所示, 包括至少一个处理器 801, 以及与至少一个处理器连接的存储器 802, 本发明实施例中不限定处理器 801 与存储器 802 之间的具体连接介质, 图 8 中处理器 801 和存储器 802 之间通过总线连接为例。总线可以分为地址总线、数据总线、控制总线等。

在本发明实施例中, 存储器 802 存储有可被至少一个处理器 801 执行的指令, 至少一个处理器 801 通过执行存储器 802 存储的指令, 可以执行前述的数据查询方法中所包括的步骤。

其中, 处理器 801 是计算设备的控制中心, 可以利用各种接口和线路连接计算设备的各个部分, 通过运行或执行存储在存储器 802 内的指令以及调用存储在存储器 802 内的数据, 从而实现数据处理。可选的, 处理器 801 可包括一个或多个处理单元, 处理器 801 可集成应用处理器和调制解调处理器, 其中, 应用处理器主要处理操作系统、用户界面和应用程序等, 调制解调处理器主要处理下发指令。可以理解的是, 上述调制解调处理器也可以不集成到处理器 801 中。在一些实施例中, 处理器 801 和存储器 802 可以在同一芯片上实现, 在一些实施例中, 它们也可以在独立的芯片上分别实现。

处理器 801 可以是通用处理器, 例如中央处理器 (CPU)、数字信号处理器、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现场可编程门阵列或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件, 可以实现或者执行本发明实施例中公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合数据查询方法实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成, 或者用处理器中的硬件及软件模块组合执行完成。

存储器 802 作为一种非易失性计算机可读存储介质，可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块。存储器 802 可以包括至少一种类型的存储介质，例如可以包括闪存、硬盘、多媒体卡、卡型存储器、随机访问存储器(Random Access Memory, RAM)、静态随机访问存储器(Static Random Access Memory, SRAM)、可编程只读存储器(Programmable Read Only Memory, PROM)、只读存储器(Read Only Memory, ROM)、带电可擦除可编程只读存储器(Electrically Erasable Programmable Read-Only Memory, EEPROM)、磁性存储器、磁盘、光盘等等。存储器 802 是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质，但不限于此。本发明实施例中的存储器 802 还可以是电路或者其它任意能够实现存储功能的装置，用于存储程序指令和/或数据。

基于相同的技术构思，本发明实施例还提供了一种计算机可读存储介质，其存储有可由计算设备执行的计算机程序，当所述程序在所述计算设备上运行时，使得所述计算设备执行上述数据查询方法的步骤。

本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

本发明是参照根据本发明的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

尽管已描述了本发明的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例作出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本发明范围的所有变更和修改。

显然，本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样，倘若本发明的这些修改和变型属于本申请权利要求及其等同技术的范围之内，则本发明也意图包含这些改动和变型在内。

权 利 要 求

1、一种数据查询方法，其特征在于，包括：

计算组件接收数据查询请求；所述数据查询请求中包括结构化查询语言 sql 数据查询脚本；

5 所述计算组件对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务；所述 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务；

所述计算组件将至少一个第一数据查询子任务分发给各自的数据源节点；所述数据源节点用于执行第一数据查询子任务并得到数据查询子结果；

10 所述计算组件基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询子任务中的第二数据查询子任务，得到数据查询结果；所述第二数据查询子任务涉及多个数据源节点。

2、如权利要求 1 所述的方法，其特征在于，所述计算组件对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务，包括：

15 所述计算组件按照语法解析规则，生成所述 sql 数据查询脚本的语法树；

所述计算组件从所述语法树中确定仅涉及单一数据源节点的第一数据查询子任务；

所述计算组件针对所述语法树中任一表连接关键词，通过所述表连接关键词对应的第一数据查询子任务的数据查询子结果构建第二数据查询子任务；

20 所述计算组件根据各第一数据查询子任务和各第二数据查询子任务的执行顺序，确定具有执行依赖关系的 m 个数据查询子任务。

3、如权利要求 2 所述的方法，其特征在于，所述计算组件按照语法解析规则，生成所述 sql 数据查询脚本的语法树，包括：

所述计算组件按照语法解析规则，依次解析出所述 sql 数据查询脚本中的各关键词；

25 所述计算组件若确定解析出的表名关键词不符合表名命名规则，则对所述表名关键词按照所述语法解析规则继续解析，直至解析到符合表名命名规则的表名关键词，从而得到所述 sql 数据查询脚本的语法树。

4、如权利要求 1 所述的方法，其特征在于，通过如下方式确定数据源节点，包括：

所述计算组件按照语法解析规则中的指定标签名，确定所述 sql 数据查询脚本中涉及的数据源节点；或，

30 所述计算组件按照语法解析规则中的表名规则，确定所述 sql 数据查询脚本中涉及的数据源节点。

5、如权利要求 1 所述的方法，其特征在于，在生成具有执行依赖关系的 m 个数据查询子任务之后，还包括：

35 所述计算组件为所述 m 个数据查询子任务中的至少一个第一数据查询子任务分别标注上对应的数据源节点标识；

所述计算组件将至少一个第一数据查询子任务分发给各自的数据源节点，包括：

针对每个第一数据查询子任务，所述计算组件基于所述第一数据查询子任务对应的数据源节点标识，将所述第一数据查询子任务分发给对应的数据源节点。

6、如权利要求 1 所述的方法，其特征在于，在所述计算组件对所述 sql 数据查询脚本

进行语法解析之前，还包括：

所述计算组件确定所述 sql 数据查询脚本可执行。

7、如权利要求 6 所述的方法，其特征在于，通过下述方式确定所述 sql 数据查询脚本是否可执行：

5 所述计算组件通过设定的 sql 数据查询脚本校验规则，对所述 sql 数据查询脚本的语法和/或参数进行校验，从而确定所述 sql 数据查询脚本是否可执行。

8、一种数据查询装置，其特征在于，包括：

接收单元，用于接收数据查询请求；所述数据查询请求中包括结构化查询语言 sql 数据查询脚本；

10 处理单元，用于对所述 sql 数据查询脚本进行语法解析，生成具有执行依赖关系的 m 个数据查询子任务，所述 m 个数据查询子任务中至少包括一个仅涉及单一数据源节点的第一数据查询子任务；将至少一个第一数据查询子任务分发给各自的数据源节点；所述数据源节点用于执行第一数据查询子任务并得到数据查询子结果；基于所述执行依赖关系和所述数据查询子结果，执行所述 m 个数据查询子任务中的第二数据查询子任务，得到数据查询结果；所述第二数据查询子任务涉及多个数据源节点。

15 9、一种计算设备，其特征在于，包括至少一个处理器以及至少一个存储器，其中，所述存储器存储有计算机程序，当所述程序被所述处理器执行时，使得所述处理器执行权利要求 1 至 7 任一权利要求所述的方法。

20 10、一种计算机可读存储介质，其特征在于，其存储有可由计算设备执行的计算机程序，当所述程序在所述计算设备上运行时，使得所述计算设备执行权利要求 1 至 7 任一权利要求所述的方法。

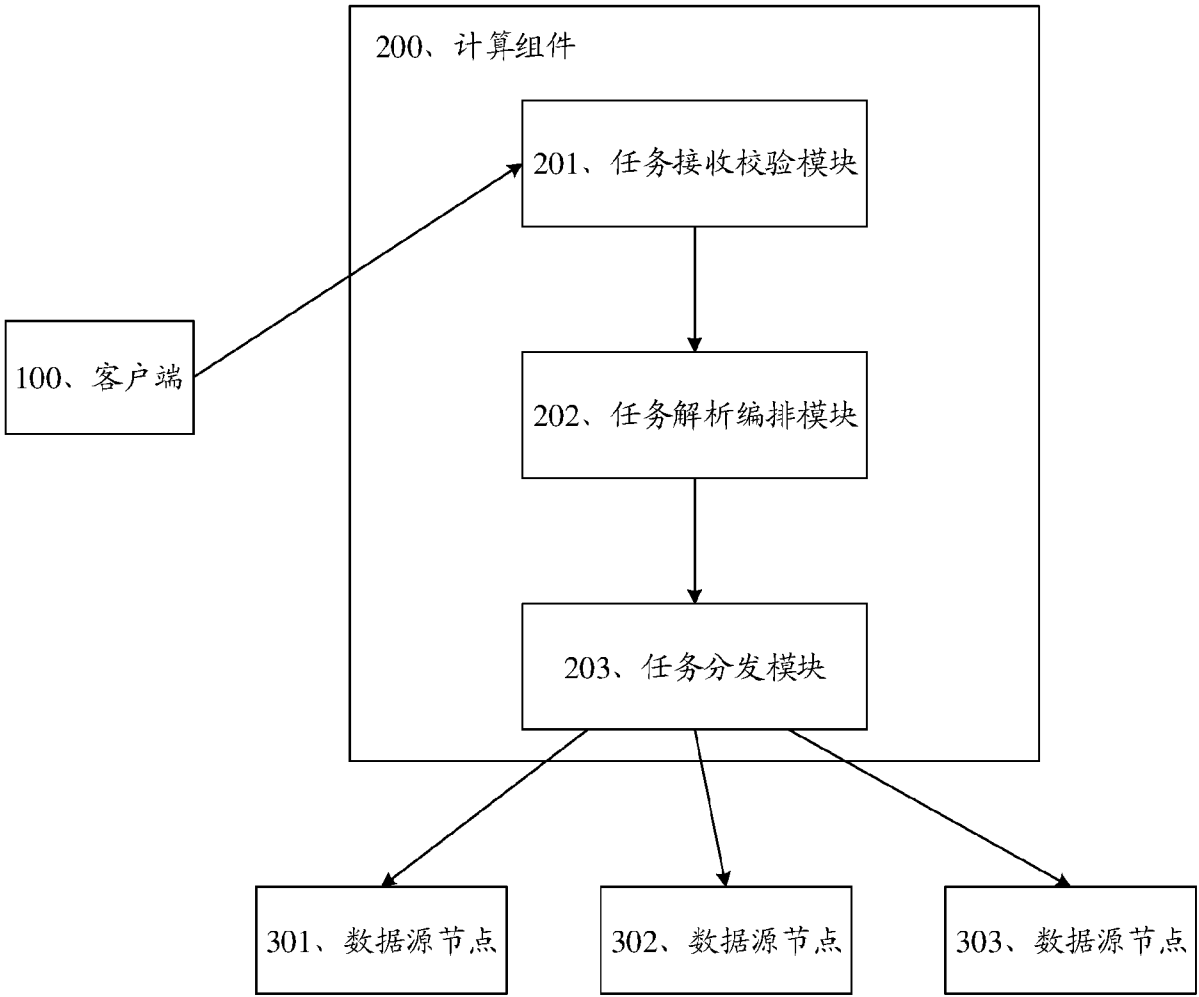


图 1

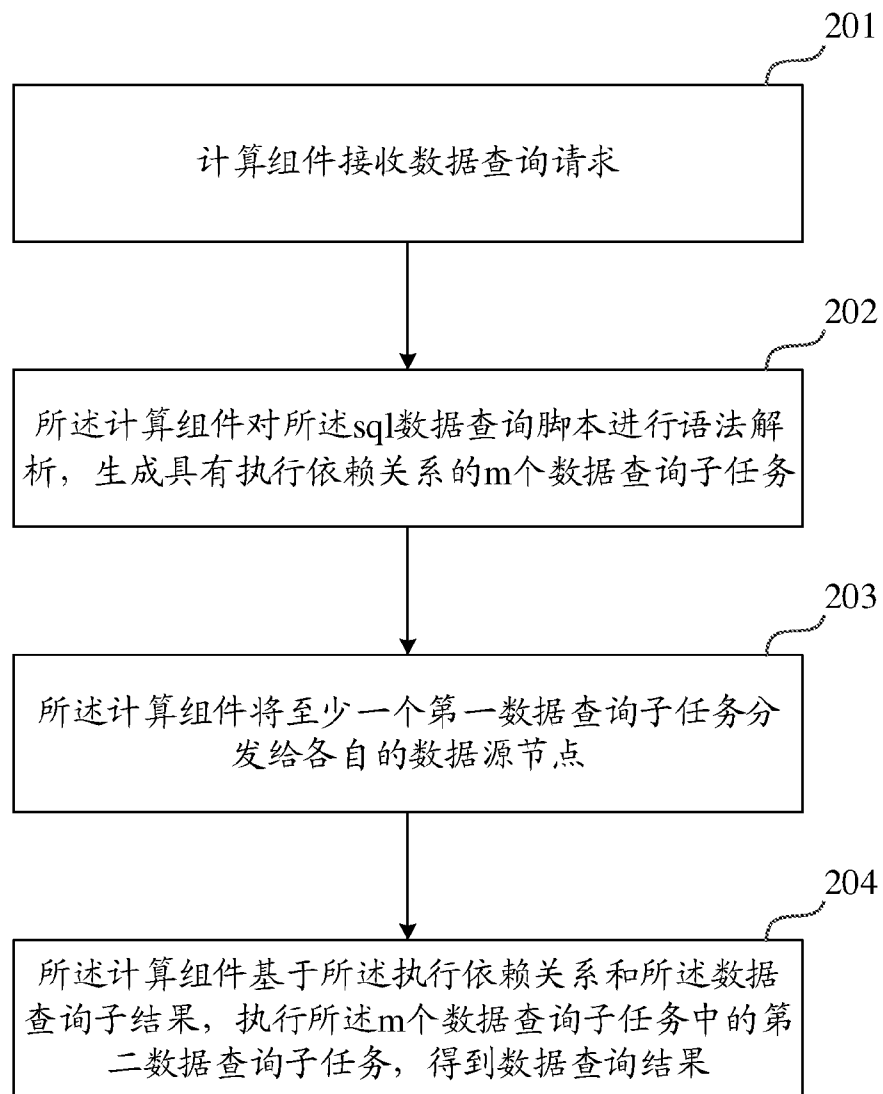


图 2

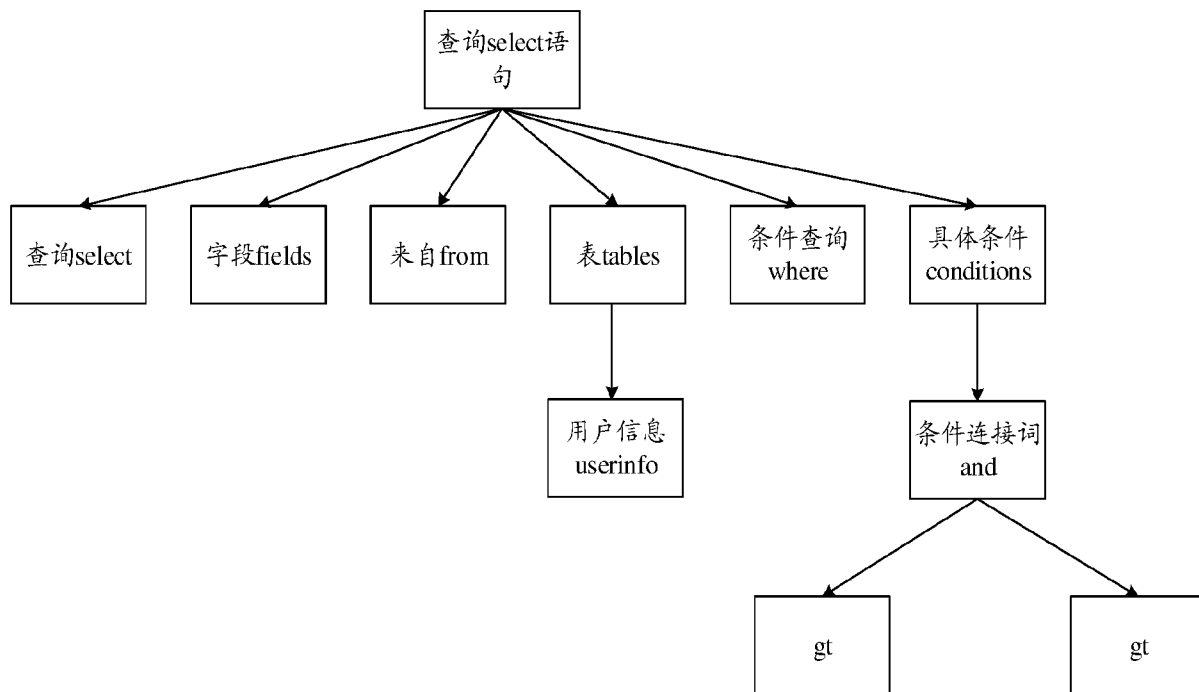


图 3

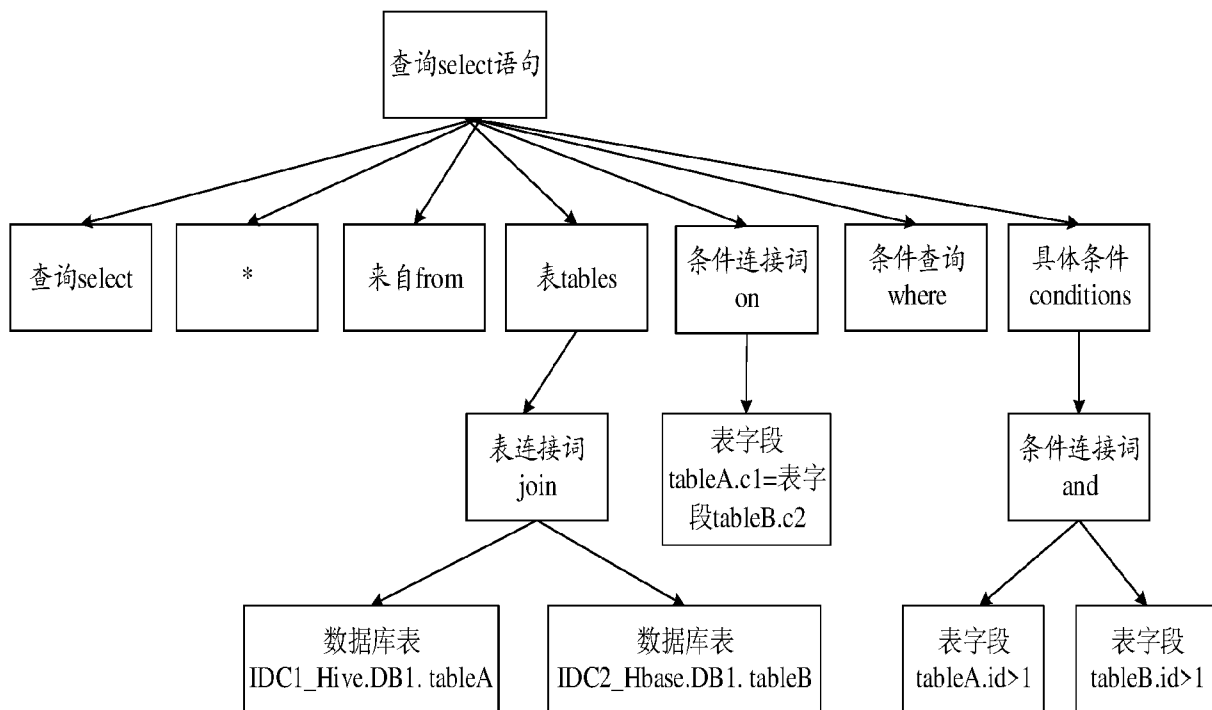


图 4

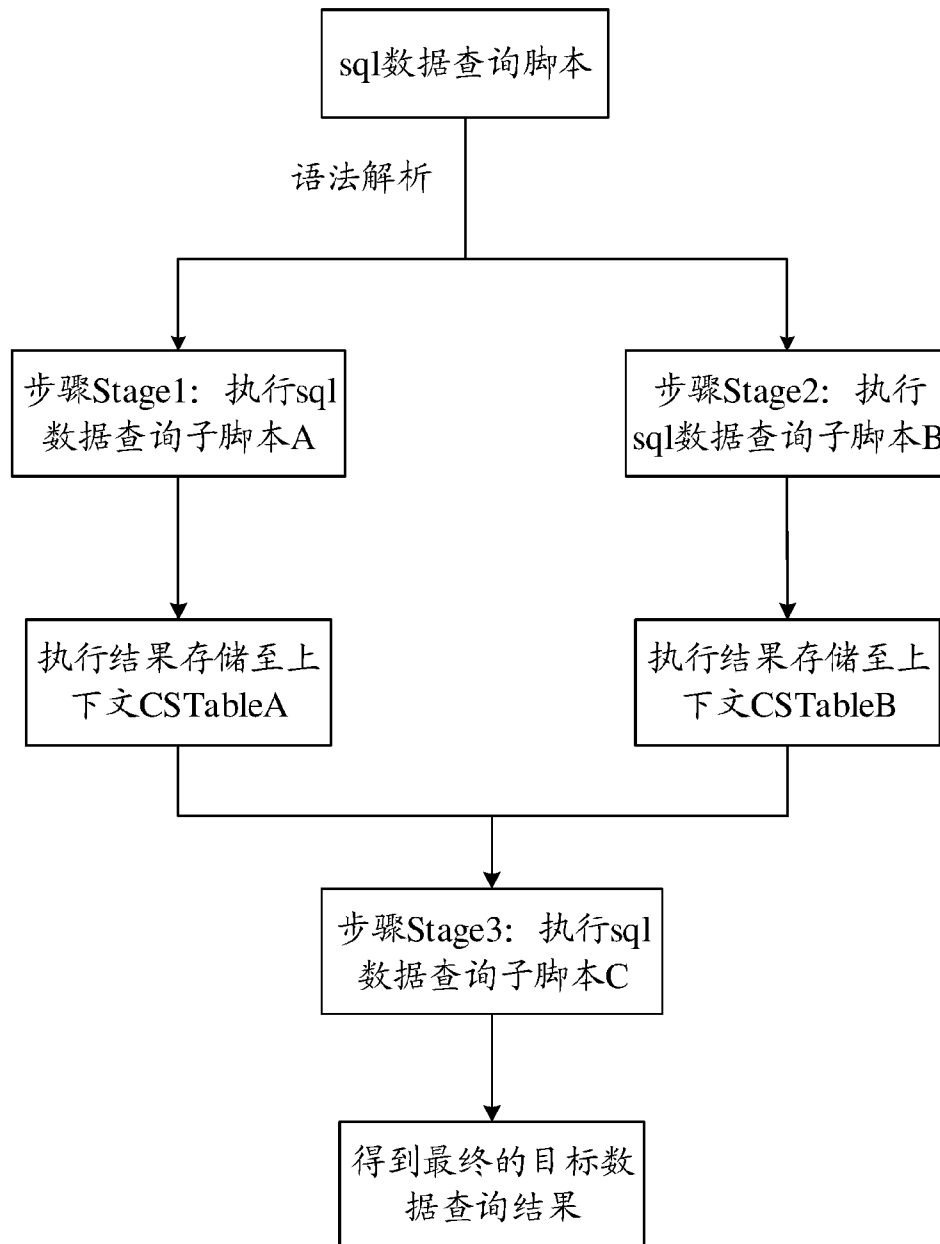


图 5

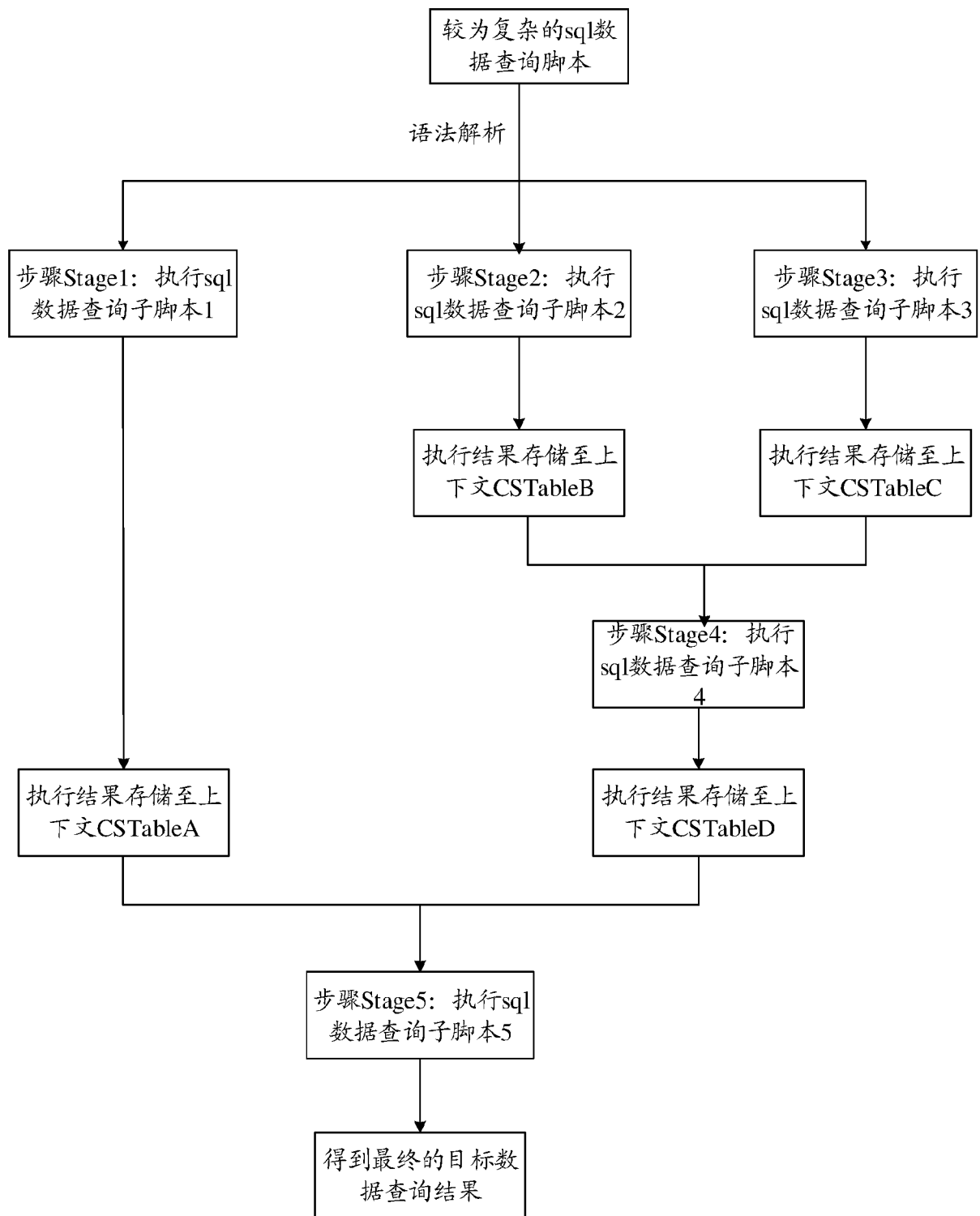


图 6

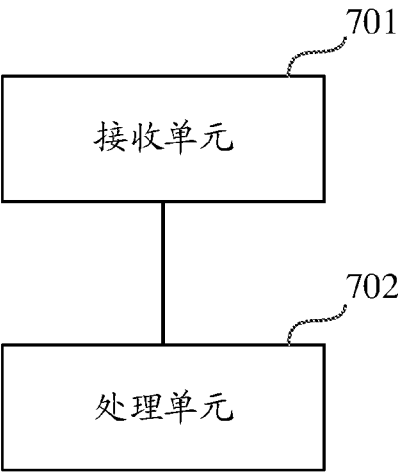


图 7

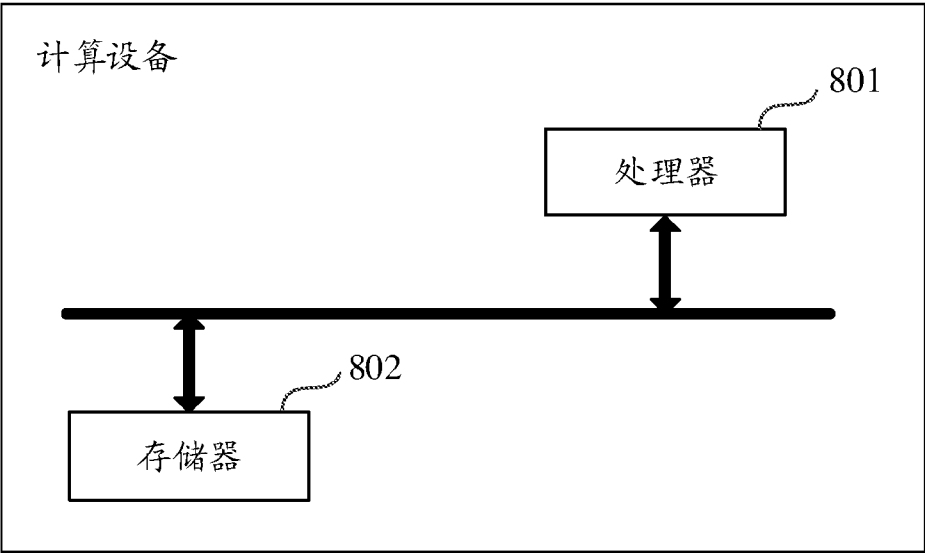


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/134954

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/242(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC, IEEE: 数据, 查询, 子查询, 查询子, 依赖, 关系, 解析, 语法, 数据源, 多个, data, query, sub, dependency, relationship, data source, grammar, analysis, multiple

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 113177062 A (SHENZHEN QIANHAI WEBANK CO., LTD.) 27 July 2021 (2021-07-27) claims 1-10	1-10
X	CN 112699141 A (YIDU CLOUD (BEIJING) TECHNOLOGY CO., LTD.) 23 April 2021 (2021-04-23) description, paragraphs [0025]-[0108]	1-10
X	CN 110674177 A (QI AN XIN TECHNOLOGY GROUP CO., LTD. et al.) 10 January 2020 (2020-01-10) description, paragraphs [0031]-[0117]	1-10
A	CN 111190924 A (UNI-LEDGER TECHNOLOGY (BEIJING) CO., LTD.) 22 May 2020 (2020-05-22) entire document	1-10
A	CN 111930770 A (BEIJING KINGSOFT CLOUD NETWORK TECHNOLOGY CO., LTD.) 13 November 2020 (2020-11-13) entire document	1-10
A	US 2014280326 A1 (LOOKER DATA SCIENCES INC.) 18 September 2014 (2014-09-18) entire document	1-10

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

16 February 2022

Date of mailing of the international search report

25 February 2022

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088, China

Authorized officer

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/134954

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	113177062	A	27 July 2021	None			
CN	112699141	A	23 April 2021	None			
CN	110674177	A	10 January 2020	None			
CN	111190924	A	22 May 2020	None			
CN	111930770	A	13 November 2020	None			
US	2014280326	A1	18 September 2014	US	2018165329	A1	14 June 2018
				CA	2904788	A1	18 September 2014
				CA	3112807	A1	18 September 2014
				US	2014280332	A1	18 September 2014
				WO	2014144938	A2	18 September 2014
				AU	2014233506	A1	01 October 2015
				EP	2973050	A2	20 January 2016
				CN	105283870	A	27 January 2016
				JP	2016517600	W	16 June 2016
				HK	1215314	A0	19 August 2016
				US	2019155855	A1	23 May 2019
				US	2021294862	A1	23 September 2021

国际检索报告

国际申请号

PCT/CN2021/134954

A. 主题的分类

G06F 16/242(2019.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, CNKI, WPI, EPODOC, IEEE: 数据, 查询, 子查询, 查询子, 依赖, 关系, 解析, 语法, 数据源, 多个, data, query, sub, dependency, relationship, data source, grammar, analysis, multiple

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 113177062 A (深圳前海微众银行股份有限公司) 2021年7月27日 (2021 - 07 - 27) 权利要求1-10	1-10
X	CN 112699141 A (医渡云北京技术有限公司) 2021年4月23日 (2021 - 04 - 23) 说明书第[0025]-[0108]段	1-10
X	CN 110674177 A (奇安信科技集团股份有限公司 等) 2020年1月10日 (2020 - 01 - 10) 说明书第[0031]-[0117]段	1-10
A	CN 111190924 A (中思博安科技北京有限公司) 2020年5月22日 (2020 - 05 - 22) 全文	1-10
A	CN 111930770 A (北京金山云网络技术有限公司) 2020年11月13日 (2020 - 11 - 13) 全文	1-10
A	US 2014280326 A1 (LOOKER DATA SCIENCES INC.) 2014年9月18日 (2014 - 09 - 18) 全文	1-10

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2022年2月16日

国际检索报告邮寄日期

2022年2月25日

ISA/CN的名称和邮寄地址

中国国家知识产权局(ISA/CN)

中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

授权官员

刘剑

电话号码 86-(10)-53961304

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/134954

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	113177062	A	2021年7月27日	无			
CN	112699141	A	2021年4月23日	无			
CN	110674177	A	2020年1月10日	无			
CN	111190924	A	2020年5月22日	无			
CN	111930770	A	2020年11月13日	无			
US	2014280326	A1	2014年9月18日	US	2018165329	A1	2018年6月14日
				CA	2904788	A1	2014年9月18日
				CA	3112807	A1	2014年9月18日
				US	2014280332	A1	2014年9月18日
				WO	2014144938	A2	2014年9月18日
				AU	2014233506	A1	2015年10月1日
				EP	2973050	A2	2016年1月20日
				CN	105283870	A	2016年1月27日
				JP	2016517600	W	2016年6月16日
				HK	1215314	A0	2016年8月19日
				US	2019155855	A1	2019年5月23日
				US	2021294862	A1	2021年9月23日