

(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



(10) 国际公布号

WO 2023/045295 A1

(43) 国际公布日

2023 年 3 月 30 日 (30.03.2023)

(51) 国际专利分类号:

G06F 16/2455 (2019.01)

京经济技术开发区科创十一街 18 号 C 座 2 层 201 室, Beijing 100176 (CN)。

(21) 国际申请号:

PCT/CN2022/084642

(72) 发明人: 魏秀丽 (WEI, Xiuli); 中国北京市经济技术开发区科创十一街 18 号院 2 号楼 4 层 A402 室, Beijing 100176 (CN)。

(22) 国际申请日:

2022 年 3 月 31 日 (31.03.2022)

(25) 申请语言:

中文

(26) 公布语言:

中文

(74) 代理人: 北京同立钧成知识产权代理有限公司 (LEADER PATENT & TRADEMARK FIRM); 中国北京市海淀区西直门北大街 32 号枫蓝国际 A 座 8F-6, Beijing 100082 (CN)。

(30) 优先权:

202111139049.1 2021 年 9 月 27 日 (27.09.2021) CN

(71) 申请人: 北京沃东天骏信息技术有限公司 (BEIJING WODONG TIANJUN INFORMATION TECHNOLOGY CO., LTD.) [CN/CN]; 中国北京市经济技术开发区科创十一街 18 号院 2 号楼 4 层 A402 室, Beijing 100176 (CN)。北京京东世纪贸易有限公司 (BEIJING JINGDONG CENTURY TRADING CO., LTD.) [CN/CN]; 中国北京市北

(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, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW,

(54) Title: DATA SKEW PROCESSING METHOD, DEVICE, STORAGE MEDIUM, AND PROGRAM PRODUCT

(54) 发明名称: 数据倾斜处理方法、设备、存储介质及程序产品

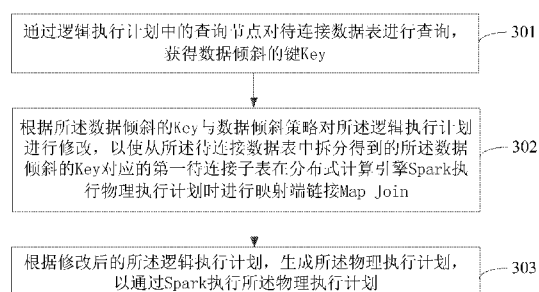


图 3

(57) Abstract: A data skew processing method, a device, a storage medium, and a program product, which relate to the technical field of computers. The method comprises: by means of a query node in a logic execution plan, querying a data table to be connected to obtain a data skew key; modifying the logic execution plan according to the data skew key and a data skew policy, so that a first sub-table to be connected that corresponds to the data skew key obtained by splitting from the data table to be connected executes Map Join when Spark executes a physical execution plan; and generating the physical execution plan according to the modified logic execution plan. According to the method provided in the present embodiment, by modifying the logic execution plan, many problems caused by data skew are prevented at source, thus reducing the limitations on usage scenarios, avoiding dependence on statistical information, and improving the comprehensiveness and accuracy of data skew processing.

(57) 摘要: 一种数据倾斜处理方法、设备、存储介质及程序产品, 涉及计算机技术领域。方法包括逻辑执行计划中的查询节点对待连接数据表进行查询, 获得数据倾斜的键Key, 根据数据倾斜的Key与数据倾斜策略对逻辑执行计划进行修改, 以使从待连接数据表中拆分得到的数据倾斜的Key对应的第一待连接子表在Spark执行物理执行计划时进行映射端链接Map Join, 根据修改后的逻辑执行计划, 生成物理执行计划。本实施例提供的方法通过对逻辑执行计划进行修改, 从根源上规避了数据倾斜带来的诸多问题, 减少了场景限制, 避免了对统计信息的依赖, 提高了数据倾斜处理的全面性和准确性。

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))。

数据倾斜处理方法、设备、存储介质及程序产品

本申请要求于 2021 年 09 月 27 日提交中国专利局、申请号为
5 202111139049.1、申请名称为“数据倾斜处理方法、设备、存储介质及程序产品”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

本发明实施例涉及计算机技术领域，尤其涉及一种数据倾斜处理方法、
10 设备、存储介质及程序产品。

背景技术

对于分布式集群系统，其不同节点负责一定范围的数据存储或数据计算。通常将数据分散度不够，导致大量的数据集中到一台或者几台服务节点上的情况，称为数据倾斜。以分布式计算引擎 Spark 为例，Spark 计算引擎进行洗牌 Shuffle 的时候，需要将各个节点上相同的键 Key 拉取到某个
15 节点上的一个任务 task 来进行处理，整个 Spark 作业的运行进度是由运行时间最长的 task 决定，因此部分 Key 发生数据倾斜后，会使得 Spark 的整体计算效率降低。

20 现有技术中，可以从引擎内核层面，引入自适应查询执行（Adaptive Query Execution, AQE）技术，AQE 针对上述数据倾斜问题使用运行时的统计信息自动优化查询执行，动态的发现倾斜数据的数量，并且把倾斜的分区分成更小的子分区来处理。

然而，实现本发明过程中，发明人发现现有技术中心至少存在如下问题：
25 题：通过 AOE 技术进行数据倾斜的优化，依赖于统计信息的准确度，且仅支持部分场景，例如仅支持同一阶段 Stage 中只有一个链接 Join 的场景，具有局限性。

发明内容

30 本发明实施例提供一种数据倾斜处理方法、设备、存储介质及程序产品，以提高数据倾斜处理的全面性和准确性。

第一方面，本发明实施例提供一种数据倾斜处理方法，包括：

通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key；

5 根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join；

根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。

10 在一种可能的设计中，所述通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的 Key，包括：

针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；

15 若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

在一种可能的设计中，所述根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，包括：

在所述逻辑执行计划中增加以下处理步骤：

20 根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表；

将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

在一种可能的设计中，所述将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，包括：

25 通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

在一种可能的设计中，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和/或所述第二数据表。

30 在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；

所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第

一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：

针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

5 在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

所述根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，还包括：

将所述数据倾斜的 Key 进行分组，得到多个分组；每个分组的总数据量小于第二预设阈值；

10 每个分组对应一个所述第一待连接子表；

所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：

针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

15 在一种可能的设计中，所述通过逻辑执行计划中的查询节点对待连接数据表进行查询之前，还包括：

将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划；
对所述未解析逻辑执行计划进行解析，得到逻辑执行计划；

在所述逻辑执行计划中加入所述查询节点。

20 在一种可能的设计中，所述根据修改后的所述逻辑执行计划，生成所述物理执行计划，包括：

对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划；
对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划；
将所述优化后的逻辑执行计划转换为物理执行计划。

25 在一种可能的设计中，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。

在一种可能的设计中，所述 Map Join 为广播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

第二方面，本发明实施例提供一种数据倾斜处理设备，包括：

30 查询模块，用于通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key；

修改模块，用于根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join；

- 5 生成模块，用于根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。

在一种可能的设计中，所述查询模块具体用于：

针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；

- 10 若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

在一种可能的设计中，所述修改模块具体用于：

在所述逻辑执行计划中增加以下处理步骤：

- 15 根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表；

将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

在一种可能的设计中，所述修改模块具体用于：

- 20 通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

在一种可能的设计中，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和/或所述第二数据表。

- 25 在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；所述修改模块具体用于：针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；所述修改模块还用于：

所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

- 30 针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

在一种可能的设计中，所述设备，还包括：

语法分析模块，用于将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划；

5 解析模块，用于对所述未解析逻辑执行计划进行解析，得到逻辑执行计划；

创建模块，用于在所述逻辑执行计划中加入所述查询节点。

在一种可能的设计中，所述生成模块具体用于：

对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划；

对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划；

10 将所述优化后的逻辑执行计划转换为物理执行计划。

在一种可能的设计中，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。

在一种可能的设计中，所述 Map Join 为广播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

15 第三方面，本发明实施例提供一种数据倾斜处理设备，包括：至少一个处理器和存储器；

所述存储器存储计算机执行指令；

20 所述至少一个处理器执行所述存储器存储的计算机执行指令，使得所述至少一个处理器执行如上第一方面以及第一方面各种可能的设计所述的方法。

第四方面，本发明实施例提供一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机执行指令，当处理器执行所述计算机执行指令时，实现如上第一方面以及第一方面各种可能的设计所述的方法。

25 第五方面，本发明实施例提供一种计算机程序产品，包括计算机程序，所述计算机程序被处理器执行时，实现如上第一方面以及第一方面各种可能的设计所述的方法。

30 第六方面，本发明实施例提供了一种运行指令的芯片，所述芯片包括存储器、处理器，所述存储器中存储代码和数据，所述存储器与所述处理器耦合，所述处理器运行所述存储器中的代码使得所述芯片用于执行上述第一方面以及第一方面各种可能的设计所述的方法。

第七方面，本发明实施例提供了一种计算机程序，当所述计算机程序被处理器执行时，用于执行上述第一方面以及第一方面各种可能的设计所述的方法。

本实施例提供的数据倾斜处理方法、设备、存储介质及程序产品，该方法通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key，根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join，根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。本实施例提供的数据倾斜处理方法通过对逻辑执行计划进行修改，使得对倾斜 Key 的处理从 Reduce Join 优化成 Map Join，从根源上规避了数据倾斜带来的诸多问题，减少了场景限制，避免了对统计信息的依赖，提高了数据倾斜处理的全面性和准确性。

15

附图说明

为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

20

图 1 为本发明一实施例提供的两个数据表进行链接操作的流程示意图；

图 2 为现有技术提供的通过散裂化倾斜 Key 进行链接操作的流程示意图；

图 3 为本发明一实施例提供的数据倾斜处理方法的流程示意图；

25

图 4 为本发明又一实施例提供的数据倾斜处理方法的流程示意图；

图 5 为本发明又一实施例提供的两个数据表进行链接操作的流程示意图；

图 6 为现有技术提供的两个数据表进行链接操作的有向无环图；

图 7 为本发明又一实施例提供的两个数据表进行链接操作的有向无环图；

30

图 8 为本发明又一实施例提供的两个数据表进行链接操作的有向无环

图；

图 9 为本发明再一实施例提供的数据倾斜处理方法的流程示意图；

图 10 为本发明一实施例提供的数据倾斜处理设备的结构示意图；

图 11 为本发明一实施例提供的数据倾斜处理设备的硬件结构示意图。

5

具体实施方式

为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。

10 基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

对于分布式集群系统，其不同节点负责一定范围的数据存储或数据计算。通常量数据分散度不够，导致大量的数据集中到了一台或者几台服务节点上，称为数据倾斜。

15 分布式计算引擎 Spark 进行洗牌 Shuffle 的时候，需要将各个节点上相同的 Key 拉取到某个节点上的一个 task 来进行处理，比如按照 Key 进行聚合或链接 Join 等操作。此时如果某个 Key 对应的数据量特别大的话，就会发生数据倾斜。比如大部分 Key 对应 10 条数据，但是个别 Key 却对应了 100 万条数据，那么大部分 task 可能就只会分配到 10 条数据，然后几秒钟
20 就运行完了；但是个别 task 可能分配到了 100 万数据，要运行一两个小时。因此，整个 Spark 作业的运行进度是由运行时间最长的 task 决定的。

图 1 为本发明一实施例提供的两个数据表进行链接操作的流程示意图；如图 1 所示，表 1-1 为学生的数学竞赛成绩，第一列数据包括学生证号 ID，第二列数据为不同 ID 分别对应的数学竞赛成绩，表 1-2 为学生的英语竞赛
25 成绩，第一列数据包括学生证号 ID，第二列数据为不同 ID 分别对应的英语竞赛成绩，从两个表中可以发现，在表 1-1 中 ID 为 001 的学生的成绩条数较多，在两个表进行 Join 时，每个 ID 相当于一个 Key，显然 Key001 对应的数据量较大，在对 Key001 对应的 task 进行处理时，相对于其他 Key 需要耗费更多的时间。因此，可以说 Key001 发生了数据倾斜。当然，此
30 处仅是为了更加形象理解数据倾斜做的示例说明，实际应用中倾斜 Key 对应的数据量判定条件可以根据需要进行设定。

针对上述数据倾斜的问题，现有技术中，通常采用两种方式进行处理。

一种方式是从应用层进行处理：通过 Rand 等技术将倾斜的 Key 进行打散，即将倾斜的 Key 后面加上随机后缀，将原本倾斜的数据进行散裂化。如图 2 所示，表 1-1 和表 2-1 中的 Key001 通过 Rand 技术进行散裂化，可以得到表 1-2 和表 2-1 中的 Key，001-1，001-2 和 001-3。散裂化后的表 1-2 和表 2-2 依据散裂化后的 Key。然而，通过该种方式，一方面会破坏原本的业务逻辑，往往把简单问题复杂化，另一方面，一旦发生拉取失败 Fetch Failure 进行数据重算时，以图 2 中的 Key001 为例，需要重新进行散裂化，上一次散裂得到的 001-1，可能在下一次散裂中变成 001-5，使得同一条数据被分配到不同的数据分区，最终导致数据重复。

另一种方式是从引擎内核层进行处理：Spark 内核引入自适应查询执行技术（Adaptive Query Execution，AQE），AQE 针对上述数据倾斜问题使用运行时的统计信息自动优化查询执行，动态的发现倾斜数据的数量，并且把倾斜的分区分成更小的子分区来处理。然而，通过该种方式处理，一方面 AQE 技术强依赖运行时的统计信息，如果统计信息存在不准确情况会导致数据倾斜误判或遗漏，另一方面，AQE 技术仅支持同一个阶段 Stage 里面只有一个 Join 的场景，对于多表 Join 的场景并不支持，并且对于 Join 后有 Shuffle 的场景也不会触发 AQE 数据倾斜的优化逻辑，再一方面，基于 AQE 处理数据倾斜，本身定位是分区 Partition 粒度的倾斜治理，如果倾斜数据是同一个 Mapper 产生的，则无法解决。

由此可见，无论是从应用层进行的优化，还是基于 Spark 内核引入 AQE 技术进行的优化，都有其缺陷。针对上述技术问题，发明人研究发现，AQE 技术属于对物理执行计划 Physical Plan 进行的优化，在 Spark SQL 整个执行计划中，主要包括：逻辑执行计划 Logical Plan 和物理执行计划 Physical Plan，后者是由前者转化得到的，也即如果从逻辑执行计划入手，对数据倾斜进行优化处理，那么可以从根源上解决数据倾斜的问题，避免了从应用层或物理执行计划层面的缺陷。基于此，本发明实施例提供了一种数据倾斜处理方法，通过对逻辑执行计划进行修改，来有效划分数据分片，将倾斜 key 从 Reduce Join 优化成 Map Join 以提升数据处理能力，从根源上规避了数据倾斜带来的诸多问题，减少了场景限制，避免了对统计信息的依赖，提高了数据倾斜处理的全面性和准确性。

下面以具体地实施例对本发明的技术方案进行详细说明。下面这几个具体的实施例可以相互结合，对于相同或相似的概念或过程可能在某些实施例不再赘述。

图 3 为本发明一实施例提供的数据倾斜处理方法的流程示意图。如图 3 所示，该方法包括：

301、通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key。

本实施例中，待连接数据表可以至少包括两个数据表，例如，可以包括第一数据表和第二数据表，通过逻辑执行计划中的查询节点可以分别对第一数据表和第二数据表分别进行查询，还可以包括第一数据表、第二数据表和第三数据表等三个以上的数据表，在查询过程中可以依次对第一数据表、第二数据表、第三数据表等进行查询。本实施例对待连接数据表的数量以及查询顺序不做限定。

可选地，在一些实施例中，所述通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的 Key，包括：针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

具体的，在通过逻辑执行计划中的查询节点对待连接数据表中的各个数据表进行查询过程中，可以针对每个 Key 进行查询，如图 1 所示，假设表 1-1 和表 1-2 为待连接数据表，在一种可实现方式中，可以首先对表 1-1 中的 Key001 进行查询，再依次查询 Key002 和 Key003 等其余 Key，再对表 2-1 中的 Key001 进行查询，再依次查询 Key002 和 Key003 等其余 Key。在另一种可实现方式中，可以首先将不同 Key 对应的条数进行排序，形成新的序列，按照序列中的顺序，从对应条数多到对应条数少，依次对相应的 Key 进行查询。此种方式有利于尽快找到倾斜的 Key，且如果条数少到一定程度，不足以成为倾斜 Key 的时候，可以及时中止查询，节省运算量。对于具体采用何种查询方式，可根据实际需要确定，本实施例对此不进行限定。

在对每个 Key 进行查询过程中，可以将该 Key 对应的数据量与第一预设阈值进行比较。该 Key 对应的数据量大于第一预设阈值时，将该 Key 确

定为倾斜的 Key。此处的数据量可以为数据的容量的大小，例如多少兆、多少 G，还可以为条数。第一预设阈值可以是根据经验确定的一个固定值。

302、根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join。

本实施例中，通过基于数据倾斜 Key 和数据倾斜策略对逻辑执行计划进行修改，使得从待连接数据表中拆分得到的数据倾斜 Key 对应的第一待连接子表在后续的物理执行计划被 Spark 计算引擎执行时能够进行映射端连接。

映射端链接 Map Join 和归并端链接 Reduce Join 均是为了将待连接数据表进行 Join 操作。也即，不同数据源数据的合并操作。其中，Reduce Join 是在 Map 阶段完成数据的标记，在 Reduce 阶段完成数据的合并。Map Join 是直接在 Map 阶段完成数据的合并，没有 Reduce 阶段。

具体的，如图 1 所示的表 1-1 和表 2-1，以此为例，对 Reduce Join 过程示例说明，在 Reduce Join 的 Map 阶段，会将输入数据统一封装为一个 Bean，此 Bean 包含了表 1-1 和表 2-1 的所有公共和非公共属性，相当于进行了全外连接，并新增加一个属性，文件名称，以区分数据是来自于表 1-1 还是表 2-1，便于在 Reduce 阶段数据的处理；Map 输出的 Key 是学生证号 ID，Value 是 Bean。在 Shuffle 阶段，根据 ID 对 Bean 进行排序，所有 ID 相同的数据都会被聚合到同一个 key 下，发往同一个 Reduce task；在 Reduce 过程中，对同一个 ID 下所有的 Bean，首先要区分出来源，是表 1-1 还是表 2-1。而如果对表 1-1 和表 2-1 进行 Map Join 的话，没有 Reduce 过程，所有的工作都在 Map 阶段完成，极大减少了网络传输和输入输出的代价。具体实现过程中：可以将表 1-1 或表 2-1，例如将表 2-1 预先缓存于各个 Map task 节点，然后等到表 1-1 的数据传输过来时，直接用表 1-1 的数据连接已经预存的表 2-1 的数据并输出即可。

本实施例中，以图 1 中的表 1-1 和表 2-1 为例，数据倾斜的 Key 为 001 的话，那么该第一待连接子表，包括来自于表 1-1 的 001 对应的第一子数据表，来自于表 2-1 的 001 对应的第二子数据表。所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算

引擎 Spark 执行物理执行计划时进行映射端链接 Map Join，可以是在 Spark 执行物理执行计划时，将第一子数据表和第二子数据表进行 Map Join，具体的，可以将第一子数据表或第二子数据表，例如将第二子数据表预先缓存于各个 Map task 节点，然后等到第一子数据表的数据传输过来时，直接用第一子数据表的数据连接已经预存的第二子数据表的数据并输出即可。

303、根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。

本实施例中，Spark 对代码中 API 的执行主要包括以下步骤：首先编写 DataFrame/Dataset/SQL 代码；其次，如果编写的代码没有错误，Spark 会将这些代码转换成逻辑执行计划；再次，Spark 会将生成的逻辑执行计划进行一系列优化后，将优化后的逻辑执行计划转换为物理执行计划；最后，Spark 执行该物理执行计划，也即在集群上对弹性分布式数据集（Resilient Distributed Datasets，RDD）进行一系列操作。

本实施例中，在对逻辑执行计划根据数据倾斜的 Key 和数据倾斜策略修改完成后，可以基于该修改后的逻辑执行计划生成物理执行计划。当然，待修改的逻辑执行计划可以是优化后的逻辑执行计划，或者是优化前的逻辑执行计划，本实施例对此不进行限定。

本实施例提供的数据倾斜处理方法，通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key，根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join，根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。本实施例提供的数据倾斜处理方法通过对逻辑执行计划进行修改，使得对倾斜 Key 的处理从 Reduce Join 优化成 Map Join，从根源上规避了数据倾斜带来的诸多问题，减少了场景限制，避免了对统计信息的依赖，提高了数据倾斜处理的全面性和准确性。

图 4 为本发明又一实施例提供的数据倾斜处理方法的流程示意图。如图 4 所示，在上述实施例的基础上，例如在图 3 所示实施例的基础上，本实施例中对如何对逻辑修改计划进行修改进行了详细说明，该方法包括：

401、通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得

数据倾斜的键 Key。

本实施例中，步骤 401 与上述实施例中步骤 301 相类似，此处不再赘述。

402、在所述逻辑执行计划中增加根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表的处理步骤，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join。

403、在所述逻辑执行计划中增加将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表的处理步骤，得到修改后的所述逻辑执行计划。

在一些实施例中，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和/或所述第二数据表。也就是说，待连接数据表中的任何一个数据表都有可能被判定存在数据倾斜的 Key。如图 1 所示，两个数据表中只有一个数据表存在数据倾斜的 Key，即表 1-1 中的 Key001。而如图 5 所示，两个数据表中均存在数据倾斜的 Key，即表 1-1 中数据倾斜的 Key 是 001，表 4-1 中数据倾斜的 Key 是 003。本实施例中对数据倾斜的 Key 的来源以及数量不做限定。

为了更加形象的对本实施例中的对逻辑执行计划的修改操作，以下结合图 5 对待连接数据表表 1-1 和表 4-1 的连接过程进行示例说明。

如图 5 所示，表 1-1 中的数据倾斜的 Key 是 001，表 4-1 中的数据倾斜的 Key 是 003。因此，在对表 1-1 进行拆分后，得到包括数据倾斜 Key001 和 003 对应数据的表 1-1-1，和包括数据非倾斜 Key002 的表 1-1-2；在对表 4-1 进行拆分后，得到包括数据倾斜 Key001 和 003 对应数据的表 4-1-1，和包括数据非倾斜 Key002 的表 4-1-2，也就是说，在对待连接数据表表 1-1 和表 4-1 根据数据倾斜的 Key 进行拆分后，得到表 1-1-1 和表 4-1-1 组成的第一待连接子表，以及由表 1-1-2 和表 4-1-2 组成的第二待连接子表。图 5 中的表 1-1 和表 4-1 仅为示例说明，为了讲述数据表连接过程仅展示了 001 至 003,3 个 Key。在实际的数据表中，Key 的数据量可以达到万数计或千万级别，也就是说，数据非倾斜的 Key 会很多，数据倾斜 Key 的占比会非常小，在此情况下，第一待连接子表中的表 1-1-1 和表 4-1-1 均作为小表在

Spark 执行物理执行计划时可以被执行 Map Join，而第二待连接子表中的表 1-1-2 和表 4-1-2 均作为大表在 Spark 执行物理执行计划时可以被执行 Reduce Join。Map Join 和 Reduce Join 的实现过程可以参照步骤 302 的描述，此处不再赘述。

5 第一待连接子表表 1-1-1 和表 4-1-1 进行 Map Join 后，得到表 5-1，第二待连接子表表 1-1-2 和表 4-1-2 进行 Reduce Join 后，得到表 5-2，表 5-1 和表 5-2 进行合并后，得到最终的数据表表 5。在一些实施例中，所述将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，包括：通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述
10 第二待连接子表进行合并，得到最终的数据表。

在一些实施例中，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。可选地，所述 Map Join 为广播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

实际应用中，第一待连接子表的 Map Join 的方式有多种，以下结合图
15 7 至图 8 对本实施例中逻辑执行计划进行修改后，第一待连接子表在 Spark 执行物理执行计划时进行 Map Join 的方式进行示例说明，并且结合图 6，对采用本实施例提供的数据倾斜处理方法前后的优势对比进行说明。

为了保证数据倾斜的 Key 对应的第一待连接子表均能实现 Map Join，在一些实施例中，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。如图 8 所示，以图 5 中待连接数据表表 1-1-1 和表 4-1-1 包
20 括两个数据倾斜 Key 为例，每个数据倾斜的 Key 对应一个第一待连接子表，即在对表 1-1 进行拆分时，可以拆分得到两个第一待连接子表，第一个第一待连接子表仅包括 Key001 的数据，而另一个待连接子表仅包括 Key002 的数据。在后续的 Map Join 过程中，如图 8 所示，Stage1 和 Stage2 针对一个第一待连接子表，Stage3 和 Stage4 针对另一个第一待连接子表，具体的，
25 均通过 BroadcastHashJoin 实现数据倾斜的 Key 的 Map Join。而在图 8 所示的有向无环图中 Stage5 和 Stage6 则是针对第二待连接子表，在经过 Stage5

和 Stage6 的处理后，通过 SortMergeJoin 实现数据非倾斜的 Key 的 Reduce Join。在 Stage7 中，将执行了 BroadcastHashJoin 的两个第一待连接子表和一个执行了 SortMergeJoin 的第二待连接子表进行合并 Union，得到最终的数据表。

5 为了能够减少合并操作的计算量，且减少用于处理第一待连接子表的计算节点的数量，在一些实施例中，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；所述根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，还包括：将所述数据倾斜的 Key 进行分组，得到多个分组；每个分组的总数据量小于第二预设阈值；每个分组对应一个所述第一待连接子表；所述从所述待连接数据表中拆分得到的所述数据
10 倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。以图 5 中待连接数据表表 1-1-1 和表 4-1-1 包括两个数据倾斜 Key 为例，可以将数据倾斜的 Key001 和 003 基于第二
15 预设阈值进行分组，例如，假设 001 和 003 对应的数据量的总和小于第二预设阈值，则可以将 001 和 003 如图 5 中所示分到一个分组，而且待连接数据表有且仅有这一个分组。如图 7 所示，针对该一个分组，可以得到一个第一待连接子表，针对该一个待连接子表，可以通过执行 Stage1 和 Stage2 的处理后，进行 BroadcastHashJoin 来实现第一待连接子表的 Map Join。而
20 在图 7 所示的有向无环图中 Stage3 和 Stage4 则是针对第二待连接子表，在经过 Stage3 和 Stage4 的处理后，通过 SortMergeJoin 实现数据非倾斜的 Key 的 Reduce Join。在 Stage5 中，将执行了 BroadcastHashJoin 的第一待连接子表和执行了 SortMergeJoin 的第二待连接子表进行合并 Union，得到最终的数据表。本实施例中，第二预设阈值的设定可以根据经验进行设定，本
25 实施例对此不做限定。

可以理解，如果待连接数据表中的数据倾斜的 Key 有多个，例如 100 个，则可以通过多种方式，将这 100 个数据倾斜的 Key 基于第二预设阈值进行分组。在一种可实现方式中，可以将数据倾斜的 Key 按照 Key 的编号数值进行排序，然后针对排序中的第一个 Key 的数据量基于第二预设阈值
30 进行判断，若小于第二预设阈值，则将第一个和第二个 Key 的数据总量基于第二预设阈值进行判断，若仍小于第二预设阈值，则将第一个、第二个

和第三个 Key 的数据总量基于第二预设阈值进行判断，直至超过第二预设阈值，则将第 N-1 个之前的 Key 划入一个分组内，从第 N 个开始继续进行上述步骤的判断。直至遍历排序中的所有 Key。在另一种可实现方式中，可以将数据倾斜的 Key 按照每个 Key 对应的数据量进行排序。然后再基于第二预设阈值对排序中的 Key 进行分组。本实施例中对此不进行限定，可根据实际需要进行选择。

图 6 为现有技术中提供的两个数据表进行链接操作的有向无环图。图 7 为本发明又一实施例提供的两个数据表进行链接操作的有向无环图。图 8 为本发明又一实施例提供的两个数据表进行链接操作的有向无环图。如图 6 所示，阶段 1Stage1 和阶段 2Stage2 均依次包括以下步骤：范围 Range、投影 Project、交换 Exchange、自定义读取 CustomShuffleReader 和排序 Sort；阶段 3Stage3 包括排序合并链接 SortMergeJoin。如图 7 所示，阶段 1Stage1 包括以下步骤：范围 Range、过滤 Filter、投影 Project、交换 Exchange、自定义读取 CustomShuffleReader 和广播交换 BroadcastExchange；阶段 2Stage2 包括以下步骤：范围 Range、过滤 Filter、投影 Project、交换 Exchange、自定义读取 CustomShuffleReader；阶段 3Stage3 和阶段 4Stage4 均包括范围 Range、过滤 Filter、投影 Project、交换 Exchange、自定义读取 CustomShuffleReader 和排序 Sort；阶段 5 包括以下步骤：广播表链接 BroadcastHashJoin、排序合并链接 SortMergeJoin、合并 Union 和自适应 Spark 执行计划 AdaptiveSparkPlan。

下面还是以图 5 所示的表 1-1 和表 4-1 为例对图 6 至图 8 进行示例说明，如图 6 所示，两个数据表进行链接，通常是对表 1-1 通过 Stage1 的处理，对表 4-1 通过 Stage2 的处理后，在 Stage3 进行 SortMergeJoin 实现 Reduce Join。如图 7 所示，对表 1-1-2 进行 Stage3 的处理，对表 4-1-1 进行 Stage4 的处理后，进行 Stage5 中 SortMergeJoin 实现 ReduceJoin，得到表 5-2，对表 1-1-1 和表 4-1-2 中其中一个表进行 Stage1 的处理，对另一个表进行 Stage2 的处理后，进行 Stage5 中的 BroadcastHashJoin 实现 MapReduce，得到表 5-1，最后将表 5-1 和表 5-2 进行 Union 后，进行自适应 Spark 执行计划的执行。图 8 相对于图 7 来说是将数据倾斜的数据拆分为了多组子表，也即在数据倾斜 Key 数量太多的情况下将表 1-1-1 再进行细化拆分后得到多个下一级子表，相对应对表 4-1-2 也拆分得到多个下一级子表，然后针

对每组下一级子表（包括表 1-1-1 的一个下一级子表，以及该下一级子表对应的表 4-1-2 的下一级子表），进行 Stage1 和 Stage2 的处理后进行 BroadcastHashJoin 实现多个 MapJoin。最终通过 Stage7 的处理将多个 BroadcastHashJoin 得到的表和 SortMergeJoin 得到的表进行合并后，进行自
5 适应 Spark 执行计划的执行。

采用本发明实施例之前，即优化前如图 6 所示处理方式中，Spark 任务运行时间是 5.5 分钟，采用本发明实施例之后，即优化后如图 7 或图 8 所示处理方式中，Spark 任务运行时间是 2.1 分钟，整体性能提升 60%。

404、根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通
10 过 Spark 执行所述物理执行计划。

本实施例中步骤 404 与上述实施例中步骤 303 相类似，此处不再赘述。

本发明实施例提供的数据倾斜处理方法，通过对逻辑执行计划进行修改，对数据倾斜的 Key 和非数据倾斜的 Key 进行拆分，使得 Spark 在执行物理执行计划时，能够使数据倾斜的 Key 对应的第一待连接子表进行 Map
15 Join，大大减少了运行时间。避免了数据倾斜的 Key 运行时间过长影响整个 Spark 任务的运行效率。并且该方式没有场景限制，从根源上解决了数据倾斜的问题。

图 9 为本发明再一实施例提供的数据倾斜处理方法的流程示意图。如图 9 所示，在上述实施例的基础上，例如在图 3 所示实施例的基础上，本
20 实施例中对逻辑执行计划的生成过程，以及由修改后的逻辑执行计划生成物理执行计划的过程进行了详细说明。该方法包括：

901、将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划。

902、对所述未解析逻辑执行计划进行解析，得到逻辑执行计划。

25 903、在所述逻辑执行计划中加入所述查询节点。

本实施例中，逻辑执行计划主要是一系列抽象的转换过程。并不涉及执行器或驱动器，他只是将用户的表达式集合转换为最优的版本。具体的，用户的代码首先会被转换成未解析逻辑执行计划(Unresolved Logical Plan)，未解析逻辑计划之所以被称为未解析的，是因为未解析逻辑执行计划并不
30 一定是正确的，未解析逻辑执行计划所引用到的表名或者列名可能存在，也可能不存在。Spark 之后会使用目录 Catalog，一个含有所有数据表 table

和数据框架 `DataFrame` 的元数据仓库，在解析器（`Analyser`）中来解析校对所引用的表名或者列名。假如未解析逻辑执行计划通过了验证，则可以得到逻辑执行计划（`Resolved Logical Plan`）。在逻辑执行计划中增加查询节点（`Query Node`），以通过该查询节点来对待连接数据表进行查询。

5 904、通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 `Key`。

 905、根据所述数据倾斜的 `Key` 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 `Key` 对应的第一待连接子表在分布式计算引擎 `Spark` 执行物理执行计划时进行映射端链接 `Map Join`。

10

 本实施例中步骤 904 和步骤 905 与上述实施例中步骤 301 和步骤 302 相类似，此处不再赘述。

 906、对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划。

15 907、对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划。

 908、将所述优化后的逻辑执行计划转换为物理执行计划。

 具体的，逻辑执行计划修改后，需要进行更新，以便后续步骤应用该更新后的逻辑执行计划。将修改后的逻辑执行计划传递到优化器 `Catalyst Optimizer` 进行优化，然后被经过一些列优化生成优化后的逻辑执行计划。`Spark` 将此逻辑执行计划转化为一个物理执行计划，检查可行的优化策略，并在此过程中检查优化。物理执行计划通过生成不同的物理执行操作，并通过代价模型进行比较分析，从而确定如何在集群上执行逻辑计划。`Spark` 在选择一个物理计划时，`Spark` 将所有代码运行在 `Spark` 的底层编程接口

20

 RDD 上。`Spark` 在运行时执行进一步优化，生成可以在执行期间优化任务或阶段的本地 `Java` 字节码，最终将结果返回给用户。

25

 本发明实施例提供的数据倾斜处理方法，通过在逻辑执行计划优化前加入查询节点，并将查询到的数据倾斜的 `Key`，与数据非倾斜的 `Key` 进行拆分，使得 `Spark` 在执行物理执行计划时，能够将拆分得到的数据倾斜的 `Key` 对应第一待连接子表进行 `Map Join`，从而能够大大节省运行时间。从根源上规避了数据倾斜带来的诸多问题，减少了场景限制，避免了对统计

30

信息的依赖，提高了数据倾斜处理的全面性和准确性。

图 10 为本发明一实施例提供的数据倾斜处理设备的结构示意图。如图 10 所示，该数据倾斜处理设备 100 包括：查询模块 1001、修改模块 1002 以及生成模块 1003。

5 查询模块 1001，用于通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key。

 修改模块 1002，用于根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行
10 计划时进行映射端链接 Map Join。

 生成模块 1003，用于根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。

 本发明实施例提供的数据倾斜处理设备，

 在一种可能的设计中，所述查询模块具体用于：

15 针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；

 若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

 在一种可能的设计中，所述修改模块具体用于：

20 在所述逻辑执行计划中增加以下处理步骤：

 根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表；

 将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

25 在一种可能的设计中，所述修改模块具体用于：

 通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

 在一种可能的设计中，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和/或所述第二数据表。

30 在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；

所述修改模块具体用于：针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

在一种可能的设计中，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；所述修改模块还用于：

5 所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

在一种可能的设计中，所述设备，还包括：

10 语法分析模块，用于将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划。

解析模块，用于对所述未解析逻辑执行计划进行解析，得到逻辑执行计划。

创建模块，用于在所述逻辑执行计划中加入所述查询节点。

在一种可能的设计中，所述生成模块具体用于：

15 对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划；
对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划；
将所述优化后的逻辑执行计划转换为物理执行计划。

在一种可能的设计中，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。

20 在一种可能的设计中，所述 Map Join 为广播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

本发明实施例提供的数据倾斜处理设备，可用于执行上述的方法实施例，其实现原理和技术效果类似，本实施例此处不再赘述。

25 图 11 为本发明一实施例提供的数据倾斜处理设备的硬件结构示意图，该设备可以是计算机，数字广播终端，消息收发设备，游戏控制台，平板设备，医疗设备，健身设备，个人数字助理等。

设备 110 可以包括以下一个或多个组件：处理组件 1101，存储器 1102，电源组件 1103，输入/输出（I/O）接口 1104，以及通信组件 1106。

30 处理组件 1101 通常控制设备 110 的整体操作，诸如与显示，电话呼叫，数据通信，相机操作和记录操作相关联的操作。处理组件 1101 可以包括一个或多个处理器 1105 来执行指令，以完成上述的方法的全部或部分步骤。此外，

处理组件 1101 可以包括一个或多个模块，便于处理组件 1101 和其他组件之间的交互。例如，处理组件 1101 可以包括多媒体模块，以方便多媒体组件 808 和处理组件 1101 之间的交互。

存储器 1102 被配置为存储各种类型的数据以支持在设备 110 的操作。这些数据的示例包括用于在设备 110 上操作的任何应用程序或方法的指令，联系人数据，电话簿数据，消息，图片，视频等。存储器 1102 可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

电源组件 1103 为设备 110 的各种组件提供电力。电源组件 1103 可以包括电源管理系统，一个或多个电源，及其他与为设备 110 生成、管理和分配电力相关联的组件。

I/O 接口 1104 为处理组件 1101 和外围接口模块之间提供接口，上述外围接口模块可以是键盘，点击轮，按钮等。这些按钮可包括但不限于：主页按钮、音量按钮、启动按钮和锁定按钮。

通信组件 1106 被配置为便于设备 110 和其他设备之间有线或无线方式的通信。设备 110 可以接入基于通信标准的无线网络，如 WiFi，2G 或 3G，或它们的组合。在一个示例性实施例中，通信组件 1106 经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中，所述通信组件 1106 还包括近场通信（NFC）模块，以促进短程通信。例如，在 NFC 模块可基于射频识别（RFID）技术，红外数据协会（IrDA）技术，超宽带（UWB）技术，蓝牙（BT）技术和其他技术来实现。

在示例性实施例中，设备 110 可以被一个或多个应用专用集成电路（ASIC）、数字信号处理器（DSP）、数字信号处理设备（DSPD）、可编程逻辑器件（PLD）、现场可编程门阵列（FPGA）、控制器、微控制器、微处理器或其他电子元件实现，用于执行上述方法。

本申请还提供一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机执行指令，当处理器执行所述计算机执行指令时，实现如上数据倾斜处理设备执行的数据倾斜处理方法。

上述的计算机可读存储介质，上述可读存储介质可以是由任何类型的易

失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。可读存储介质可以是通用或专用计算机能够存取的任何可用介质。

一种示例性的可读存储介质耦合至处理器，从而使处理器能够从该可读存储介质读取信息，且可向该可读存储介质写入信息。当然，可读存储介质也可以是处理器的组成部分。处理器和可读存储介质可以位于专用集成电路(Application Specific Integrated Circuits, 简称：ASIC)中。当然，处理器和可读存储介质也可以作为分立组件存在于设备中。

本领域普通技术人员可以理解：实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时，执行包括上述各方法实施例的步骤；而前述的存储介质包括：ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

本发明实施例还提供一种计算机程序产品，包括计算机程序，所述计算机程序被处理器执行时，实现如上数据倾斜处理设备执行的数据倾斜处理方法。

本发明实施例还提供一种运行指令的芯片，所述芯片包括存储器、处理器，所述存储器中存储代码和数据，所述存储器与所述处理器耦合，所述处理器运行所述存储器中的代码使得所述芯片用于执行上述数据倾斜处理设备执行的数据倾斜处理方法。

本发明实施例还提供了一种计算机程序，当所述计算机程序被处理器执行时，用于执行上述数据倾斜处理设备执行的数据倾斜处理方法。

最后应说明的是：以上各实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述各实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分或者全部技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

权 利 要 求 书

1、一种数据倾斜处理方法，其特征在于，包括：

通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key；

5 根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join；

根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过
10 Spark 执行所述物理执行计划。

2、根据权利要求 1 所述的方法，其特征在于，所述通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的 Key，包括：

针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；

15 若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

3、根据权利要求 1 或 2 所述的方法，其特征在于，所述根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，包括：

在所述逻辑执行计划中增加以下处理步骤：

20 根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表；

将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

4、根据权利要求 3 所述的方法，其特征在于，所述将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，包括：

通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

5、根据权利要求 3 所述的方法，其特征在于，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和
30 /或所述第二数据表。

6、根据权利要求 5 所述的方法，其特征在于，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；

5 所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：

针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

7、根据权利要求 5 所述的方法，其特征在于，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

10 所述根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，还包括：

将所述数据倾斜的 Key 进行分组，得到多个分组；每个分组的总数据量小于第二预设阈值；

每个分组对应一个所述第一待连接子表；

15 所述从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join，包括：

针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

20 8、根据权利要求 1 至 7 任一项所述的方法，其特征在于，所述通过逻辑执行计划中的查询节点对待连接数据表进行查询之前，还包括：

将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划；
对所述未解析逻辑执行计划进行解析，得到逻辑执行计划；

在所述逻辑执行计划中加入所述查询节点。

25 9、根据权利要求 1 至 8 任一项所述的方法，其特征在于，所述根据修改后的所述逻辑执行计划，生成所述物理执行计划，包括：

对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划；
对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划；
将所述优化后的逻辑执行计划转换为物理执行计划。

30 10、根据权利要求 3 至 7 任一项所述的方法，其特征在于，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。

11、根据权利要求 10 所述的方法，其特征在于，所述 Map Join 为广

播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

12、一种数据倾斜处理设备，其特征在于，包括：

5 查询模块，用于通过逻辑执行计划中的查询节点对待连接数据表进行查询，获得数据倾斜的键 Key；

修改模块，用于根据所述数据倾斜的 Key 与数据倾斜策略对所述逻辑执行计划进行修改，以使从所述待连接数据表中拆分得到的所述数据倾斜的 Key 对应的第一待连接子表在分布式计算引擎 Spark 执行物理执行计划时进行映射端链接 Map Join；

10 生成模块，用于根据修改后的所述逻辑执行计划，生成所述物理执行计划，以通过 Spark 执行所述物理执行计划。

13、根据权利要求 12 所述的设备，其特征在于，所述查询模块具体用于：

15 针对所述待连接数据表的每个 Key，将所述 Key 对应的数据量与第一预设阈值进行对比；

若所述 Key 对应的数据量大于所述第一预设阈值，则将所述 Key 确定为数据倾斜的 Key。

14、根据权利要求 12 或 13 所述的设备，其特征在于，所述修改模块具体用于：

20 在所述逻辑执行计划中增加以下处理步骤：

根据所述数据倾斜的 Key，将所述待连接数据表的数据进行拆分，得到所述第一待连接子表和数据非倾斜的 Key 对应的第二待连接子表；

将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

25 15、根据权利要求 14 所述的设备，其特征在于，所述修改模块具体用于：

通过 Union 算子，将连接后的所述第一待连接子表和连接后的所述第二待连接子表进行合并，得到最终的数据表。

30 16、根据权利要求 14 所述的设备，其特征在于，所述待连接数据表包括第一数据表和第二数据表，所述数据倾斜的 Key 来自于所述第一数据表和/或所述第二数据表。

17、根据权利要求 16 所述的设备，其特征在于，所述数据倾斜的 Key 为多个，所述第一待连接子表为多个，所述数据倾斜的 Key 与所述第一待连接子表一一对应；

所述修改模块具体用于：针对每个数据倾斜的 Key，所述数据倾斜的 Key 对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

18、根据权利要求 16 所述的设备，其特征在于，所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

所述修改模块还用于：

所述数据倾斜的 Key 为多个，所述第一待连接子表为至少一个；

针对每个分组，所述分组对应的第一待连接子表在 Spark 执行物理执行计划时进行 Map Join。

19、根据权利要求 12 至 18 任一项所述的设备，其特征在于，还包括：

语法分析模块，用于将结构化查询语言 SQL 文本解析成语法树，生成未解析逻辑执行计划；

解析模块，用于对所述未解析逻辑执行计划进行解析，得到逻辑执行计划；

创建模块，用于在所述逻辑执行计划中加入所述查询节点。

20、根据权利要求 12 至 19 任一项所述的设备，其特征在于，所述生成模块具体用于：

对修改后的所述逻辑执行计划进行更新，得到更新后的逻辑执行计划；
对所述更新后的逻辑执行计划进行优化，得到优化后的逻辑执行计划；
将所述优化后的逻辑执行计划转换为物理执行计划。

21、根据权利要求 14 至 18 任一项所述的设备，其特征在于，所述第二待连接子表 Spark 执行物理执行计划时进行归并端链接 Reduce Join。

22、根据权利要求 21 所述的设备，其特征在于，所述 Map Join 为广播表链接 BroadcastHashJoin，所述 Reduce Join 为排序合并链接 SortMergeJoin。

23、一种数据倾斜处理设备，其特征在于，包括：至少一个处理器和存储器；

所述存储器存储计算机执行指令；

所述至少一个处理器执行所述存储器存储的计算机执行指令，使得所

述至少一个处理器执行如权利要求 1 至 11 任一项所述的数据倾斜处理方法。

24、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有计算机执行指令，当处理器执行所述计算机执行指令时，实现如权利要求 1 至 11 任一项所述的数据倾斜处理方法。

5 25、一种计算机程序产品，包括计算机程序，其特征在于，所述计算机程序被处理器执行时实现权利要求 1 至 11 任一项所述的数据倾斜处理方法。

10 26、一种运行指令的芯片，其特征在于，所述芯片包括存储器、处理器，所述存储器中存储代码和数据，所述存储器与所述处理器耦合，所述处理器运行所述存储器中的代码使得所述芯片用于执行上述权利要求 1 至 11 任一项所述的数据倾斜处理方法。

27、一种计算机程序，其特征在于，当所述计算机程序被处理器执行时，用于执行上述权利要求 1 至 11 任一项所述的数据倾斜处理方法。

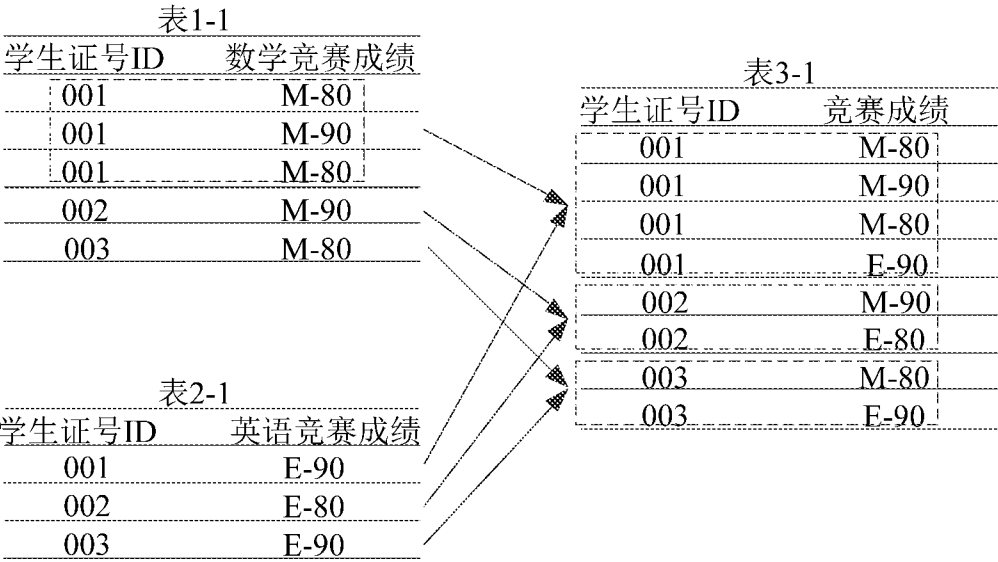


图 1

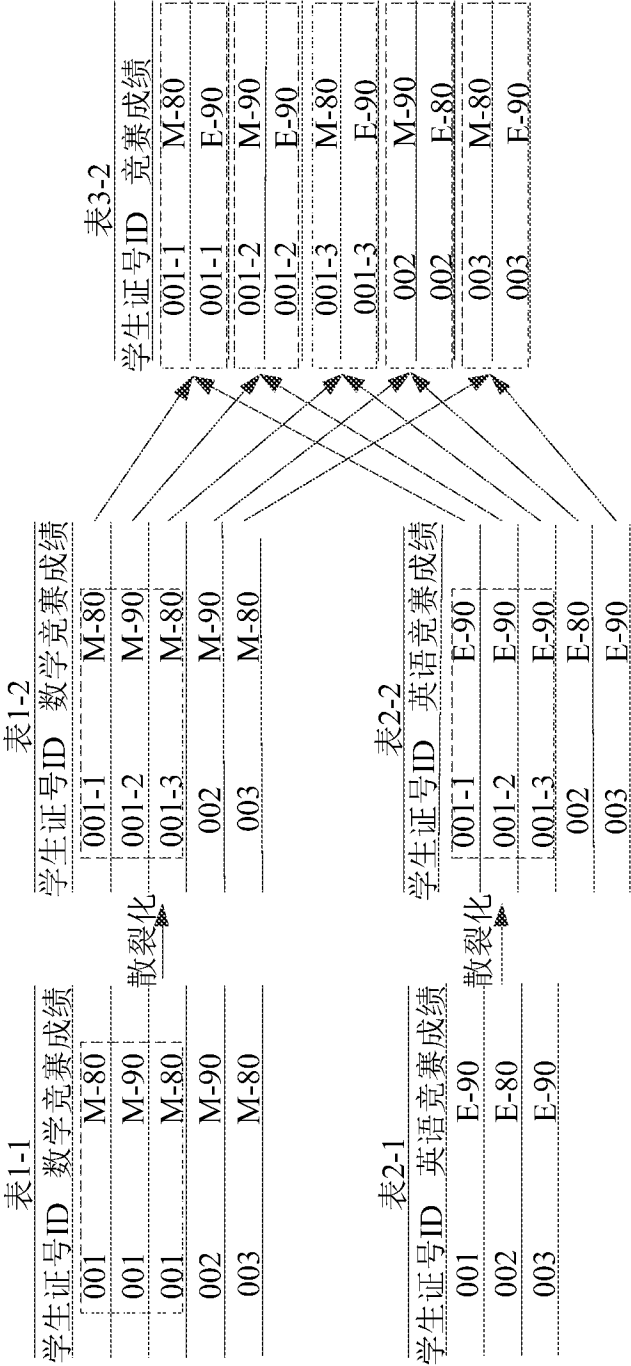


图2

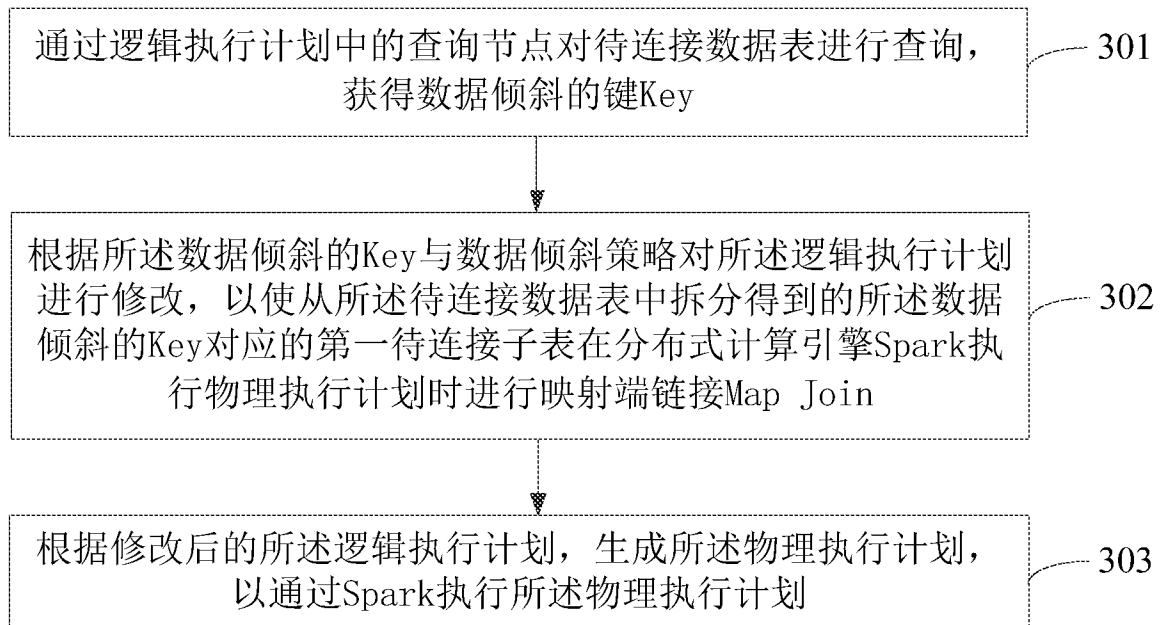


图 3

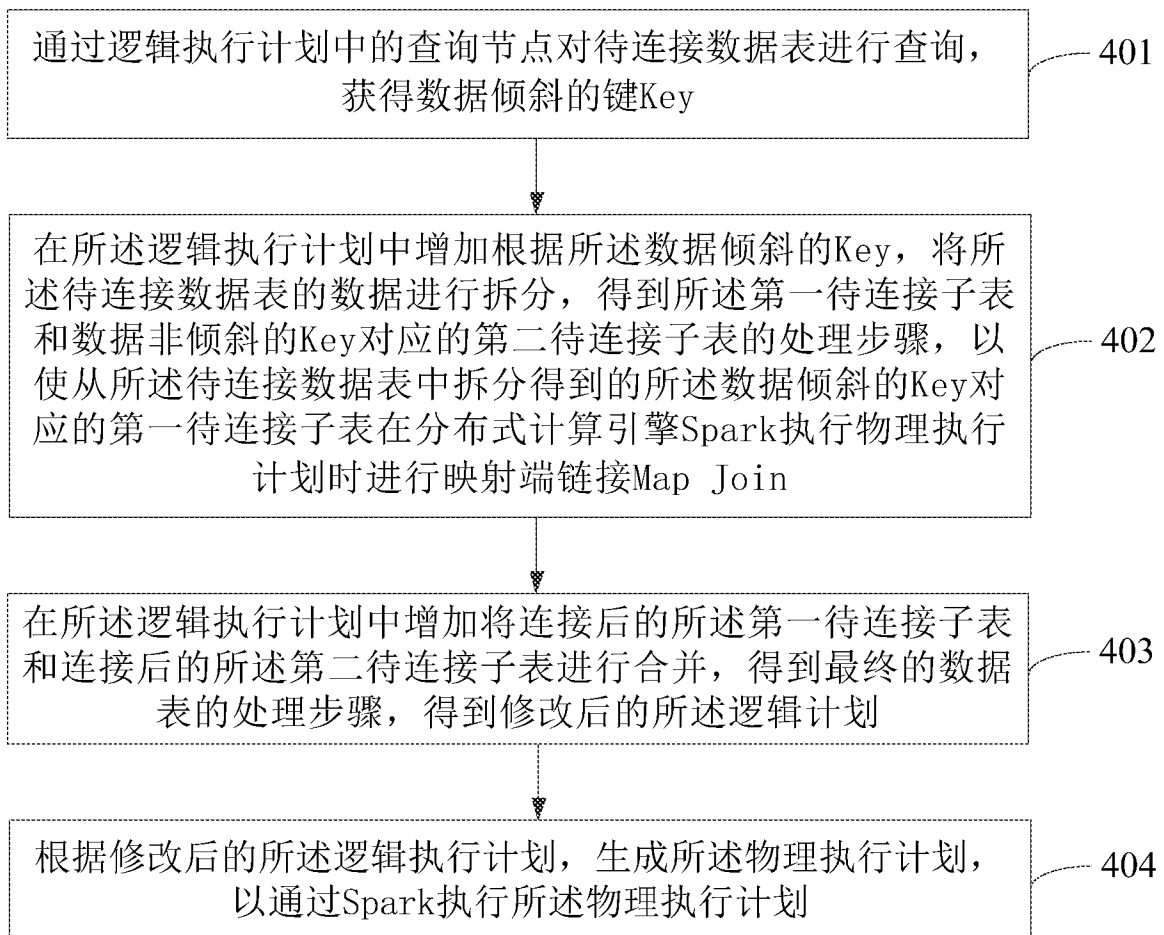


图 4

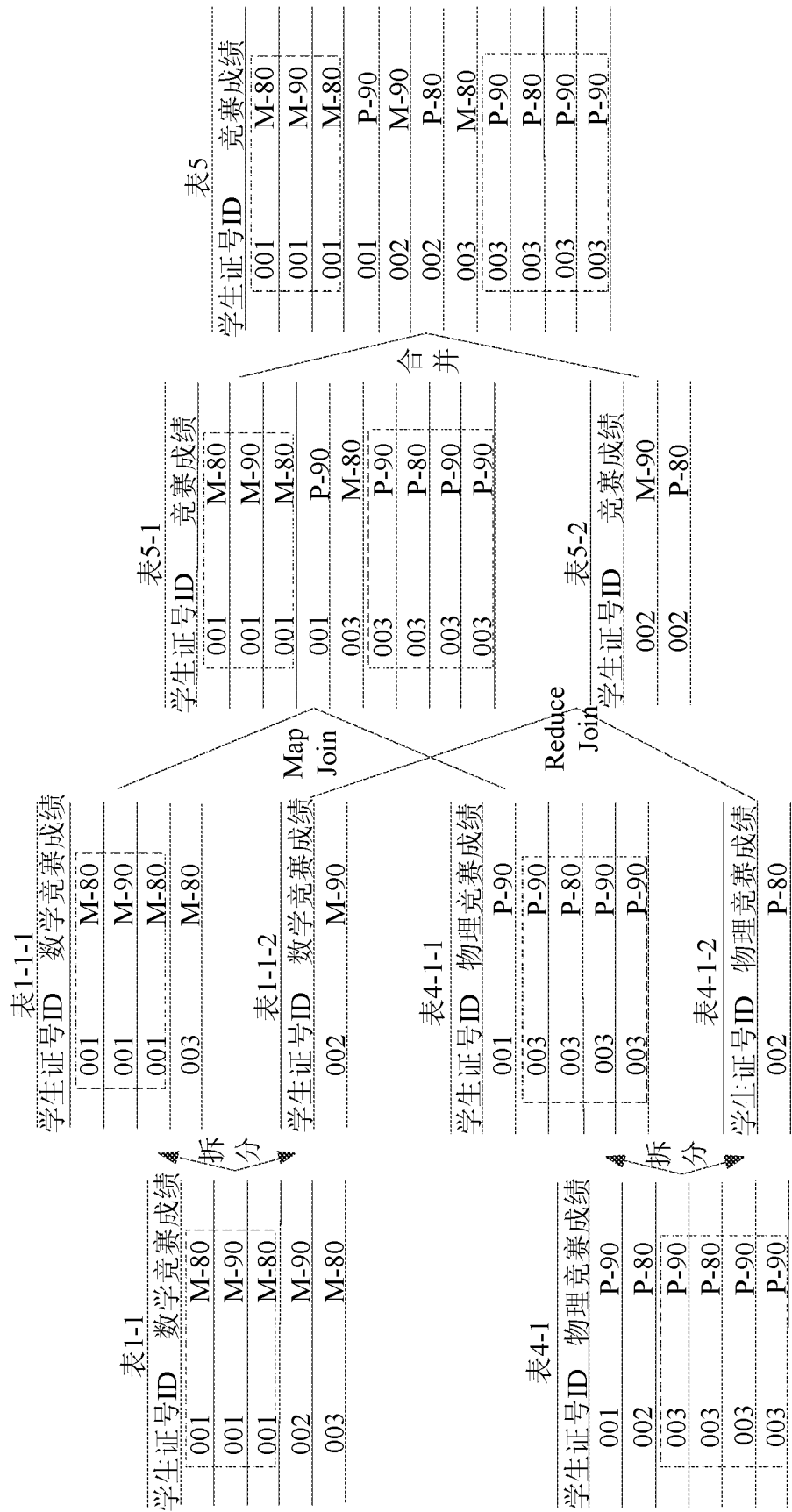


图5

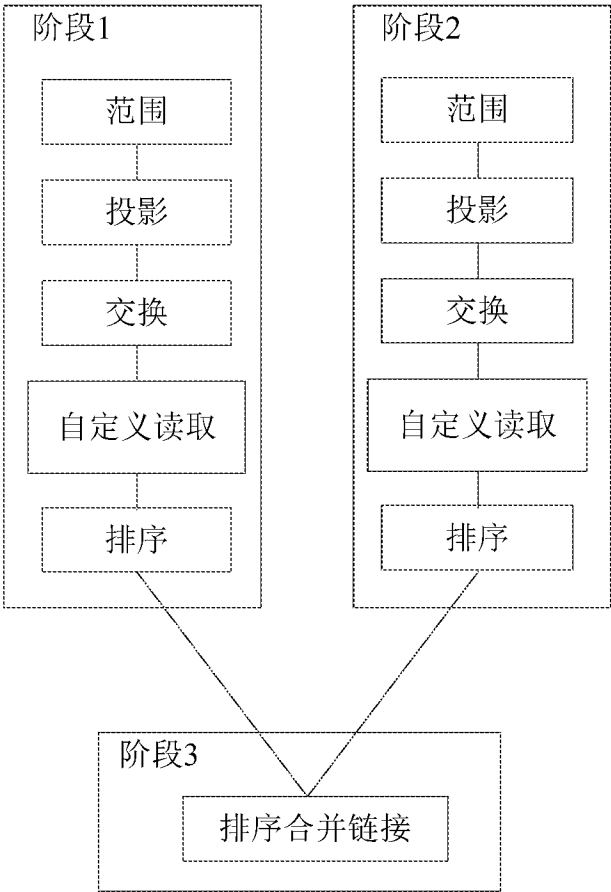


图 6

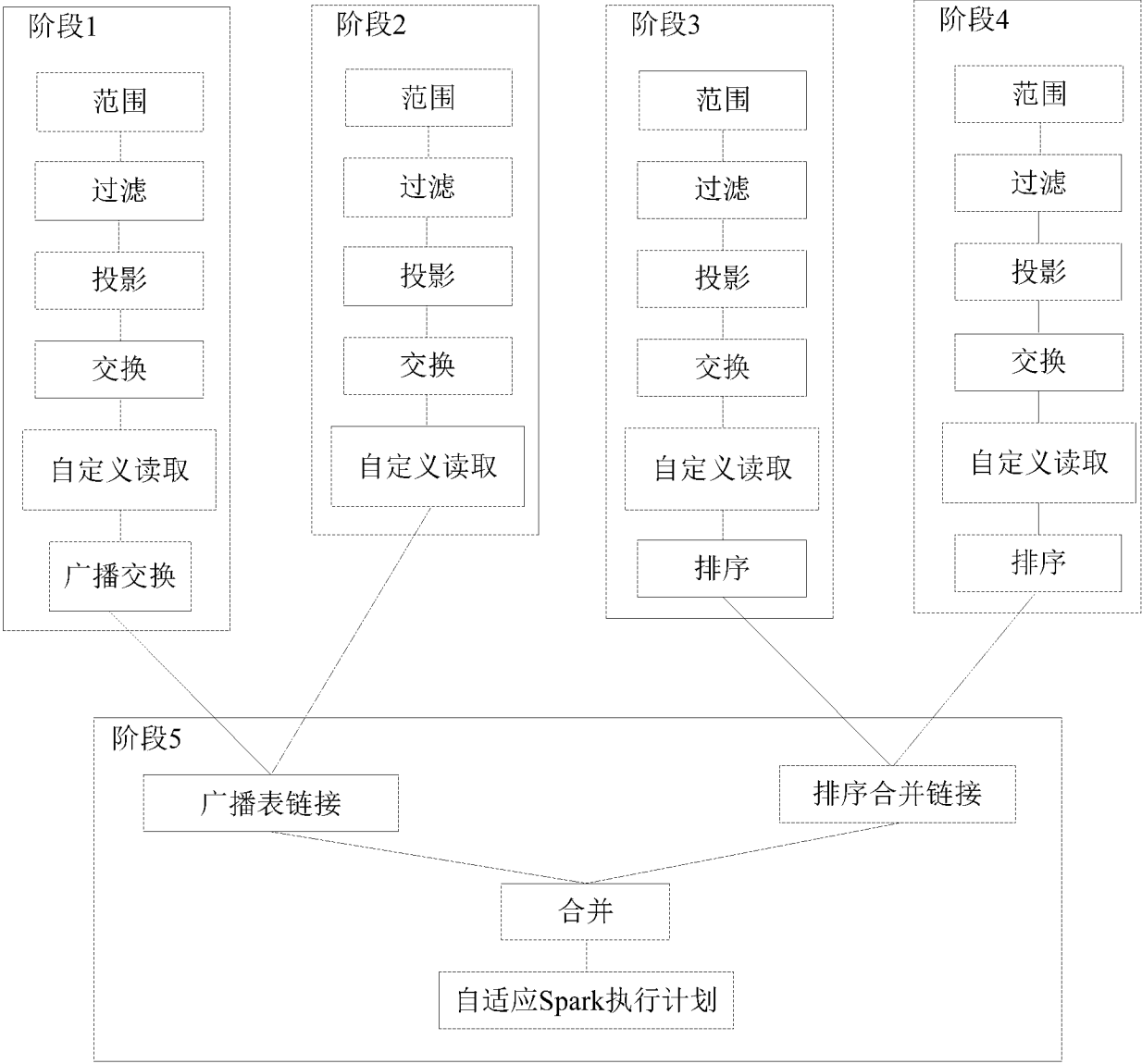


图 7

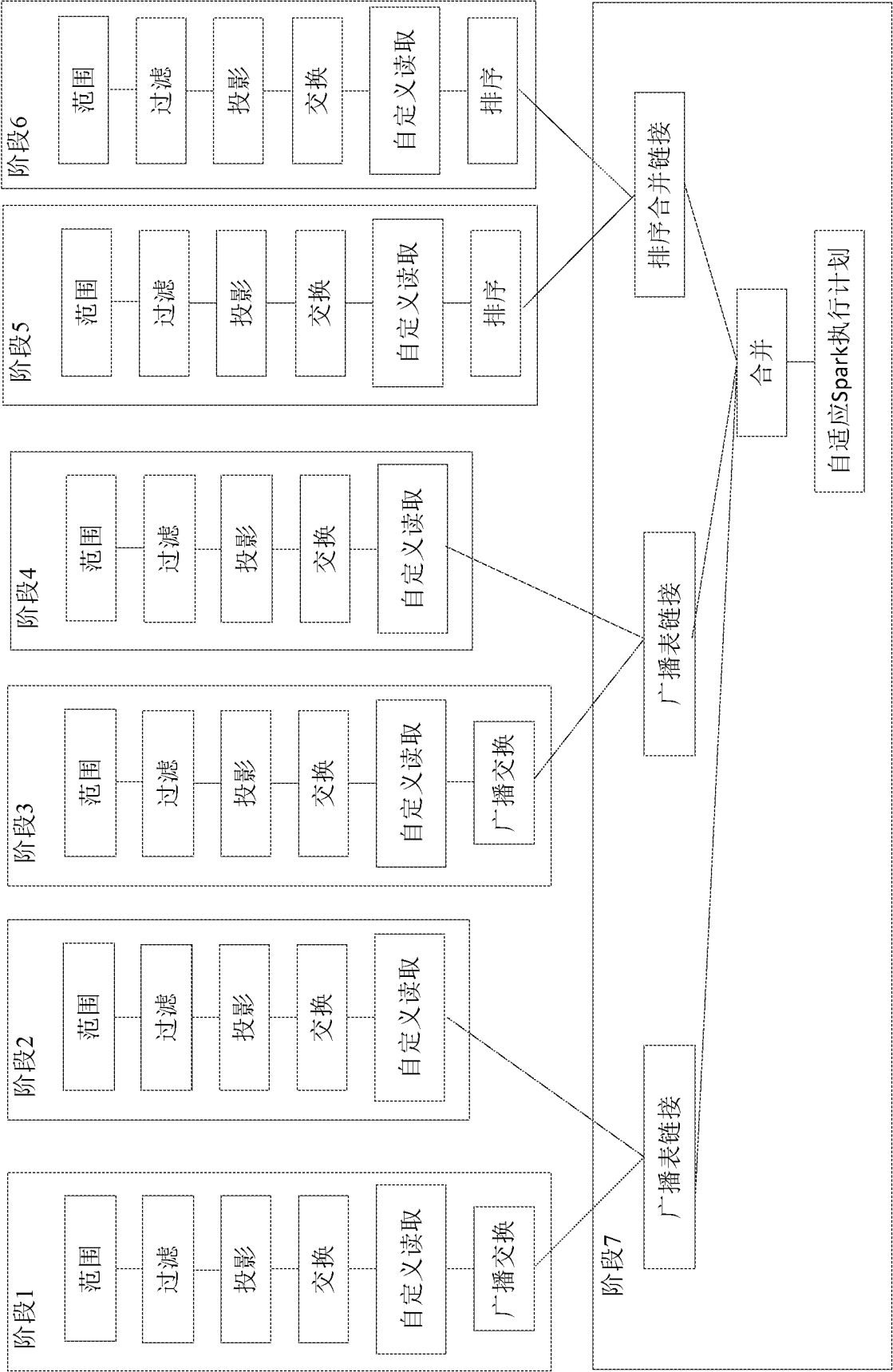


图8

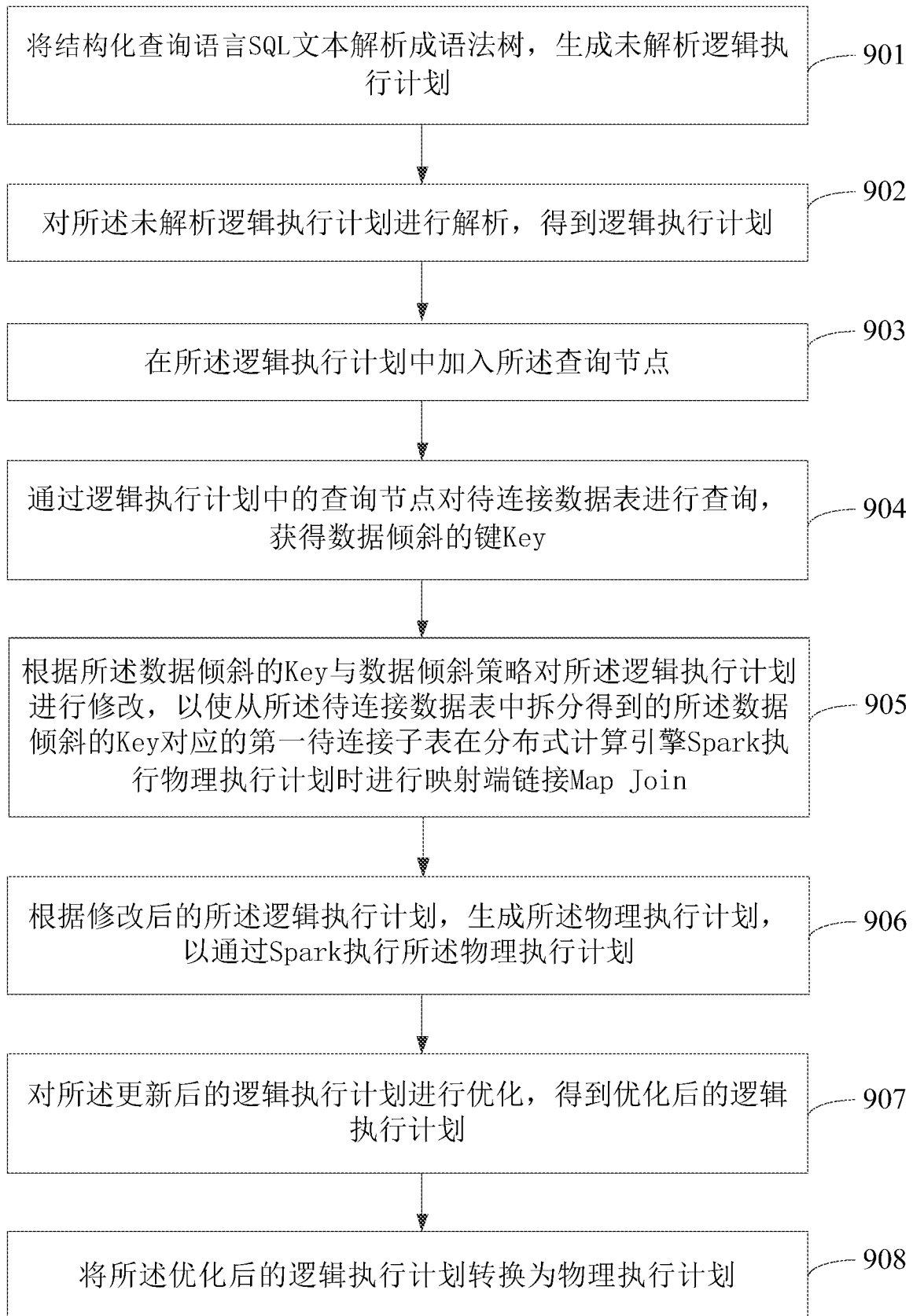


图 9

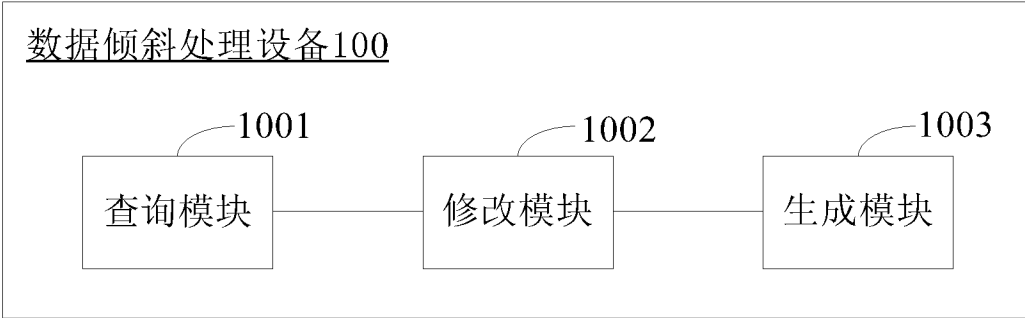


图 10

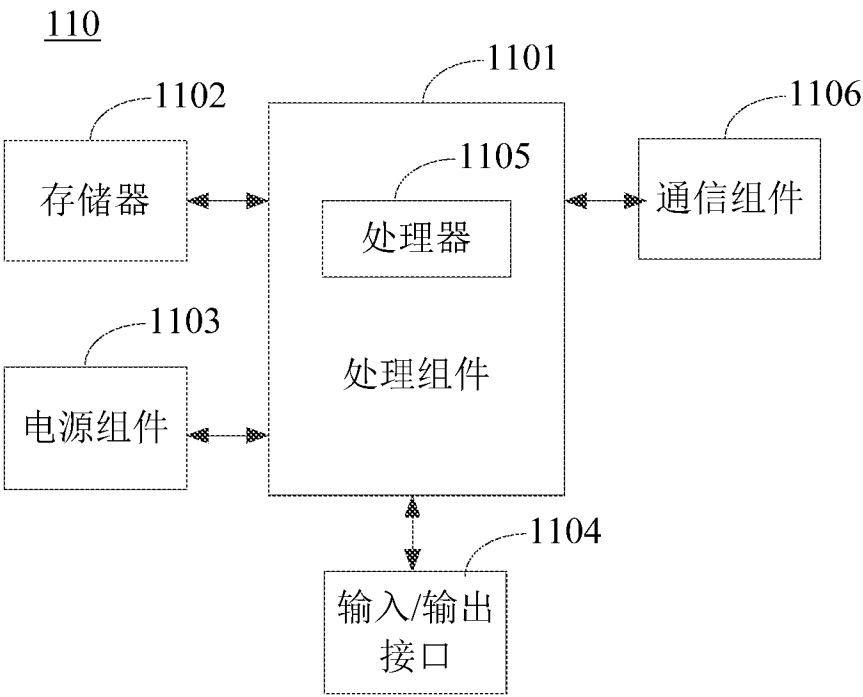


图 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/084642

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/2455(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)

CNABS, CNTXT, SIPOABS, DWPI: 分布式, 数据, 倾斜, 均衡, 映射, 规约, 连接, 链接, 物理, 逻辑, 计划, 键, 阈值, 阈值;
spark, data, skew, incline, balance, map, reduce, join, physical, logical, plan, key, threshold**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 113821541 A (BEIJING WODONG TIANJUN INFORMATION TECHNOLOGY CO., LTD. et al.) 21 December 2021 (2021-12-21) description, paragraphs [0033]-[0131]	1-27
Y	CN 106874322 A (ALIBABA GROUP HOLDING LTD.) 20 June 2017 (2017-06-20) description, paragraphs [0022]-[0078]	1-27
Y	CN 109299131 A (BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.) 01 February 2019 (2019-02-01) description, paragraphs [0050]-[0180]	1-27
A	CN 105975463 A (WUHAN ANTIAN INFORMATION TECHNOLOGY CO., LTD.) 28 September 2016 (2016-09-28) entire document	1-27
A	CN 110673794 A (ZTE CORP.) 10 January 2020 (2020-01-10) entire document	1-27
A	CN 107066612 A (ZHENGZHOU YUNHAI INFORMATION TECHNOLOGY CO., LTD.) 18 August 2017 (2017-08-18) entire document	1-27



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

10 May 2022

Date of mailing of the international search report

02 June 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

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/084642**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 105930479 A (LETV HOLDINGS (BEIJING) CO., LTD. et al.) 07 September 2016 (2016-09-07) entire document	1-27
A	US 10691597 B1 (MIFRONTIERS CORP.) 23 June 2020 (2020-06-23) entire document	1-27

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/084642

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	113821541	A	21 December 2021	None			
CN	106874322	A	20 June 2017	None			
CN	109299131	A	01 February 2019	EP	3654201	A1	20 May 2020
				US	2020151170	A1	14 May 2020
				CN	109299131	B	29 May 2020
				US	11120018	B2	14 September 2021
CN	105975463	A	28 September 2016	None			
CN	110673794	A	10 January 2020	WO	2021052169	A1	25 March 2021
				CN	110673794	B	17 December 2021
CN	107066612	A	18 August 2017	None			
CN	105930479	A	07 September 2016	None			
US	10691597	B1	23 June 2020	None			

国际检索报告

国际申请号

PCT/CN2022/084642

A. 主题的分类

G06F 16/2455 (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))

CNABS, CNTXT, SIPOABS, DWPI: 分布式, 数据, 倾斜, 均衡, 映射, 规约, 连接, 链接, 物理, 逻辑, 计划, 键, 阈值, 阈值; spark, data, skew, incline, balance, map, reduce, join, physical, logical, plan, key, threshold

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 113821541 A (北京沃东天骏信息技术有限公司 等) 2021年12月21日 (2021 - 12 - 21) 说明书第[0033]-[0131]段	1-27
Y	CN 106874322 A (阿里巴巴集团控股有限公司) 2017年6月20日 (2017 - 06 - 20) 说明书第[0022]-[0078]段	1-27
Y	CN 109299131 A (百度在线网络技术北京有限公司) 2019年2月1日 (2019 - 02 - 01) 说明书第[0050]-[0180]段	1-27
A	CN 105975463 A (武汉安天信息技术有限责任公司) 2016年9月28日 (2016 - 09 - 28) 全文	1-27
A	CN 110673794 A (中兴通讯股份有限公司) 2020年1月10日 (2020 - 01 - 10) 全文	1-27
A	CN 107066612 A (郑州云海信息技术有限公司) 2017年8月18日 (2017 - 08 - 18) 全文	1-27
A	CN 105930479 A (乐视控股北京有限公司 等) 2016年9月7日 (2016 - 09 - 07) 全文	1-27

☒ 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5月10日

国际检索报告邮寄日期

2022年6月2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

受权官员

熊沐阳

电话号码 (86-27)59371887

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
A	US 10691597 B1 (MIFRONTIERS CORP) 2020年6月23日 (2020 - 06 - 23) 全文	1-27

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2022/084642

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	113821541	A	2021年12月21日	无			
CN	106874322	A	2017年6月20日	无			
CN	109299131	A	2019年2月1日	EP	3654201	A1	2020年5月20日
				US	2020151170	A1	2020年5月14日
				CN	109299131	B	2020年5月29日
				US	11120018	B2	2021年9月14日
CN	105975463	A	2016年9月28日	无			
CN	110673794	A	2020年1月10日	WO	2021052169	A1	2021年3月25日
				CN	110673794	B	2021年12月17日
CN	107066612	A	2017年8月18日	无			
CN	105930479	A	2016年9月7日	无			
US	10691597	B1	2020年6月23日	无			