(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106708606 B
(45)授权公告日 2020.07.07

(21)申请号 201510789816.1

(22)申请日 2015.11.17

(65)同一申请的已公布的文献号
申请公布号 CN 106708606 A

(43)申请公布日 2017.05.24

(73)专利权人 阿里巴巴集团控股有限公司
地址 英属开曼群岛大开曼资本大厦一座四层847号邮箱

(72)发明人 梁永锋

(74)专利代理机构 北京国昊天诚知识产权代理有限公司 11315
代理人 黄熊 李永强

(51)Int.Cl.
G06F 9/46(2006.01)
G06Q 30/06(2012.01)

(56)对比文件

CN 104679590 A,2015.06.03,第一个Map处理的数据大小为64M,第二个Map处理的数据大小为36M.

CN 104391748 A,2015.03.04,执行Map操作,多输出排序、合并、分区,最后检查原始数据文件集中是否还有数据文件未被处理,若无,结束程序,否则,重新将划分好的数据文件再次执行此过程.

CN 103699441 A,2014.04.02,

CN 104978345 A,2015.10.14,

CN 103399927 A,2013.11.20,

US 2011154339 A1,2011.06.23,

CN 103500089 A,2014.01.08,PCT.

郑亚松,王达,叶笑春,崔慧敏,徐远超,范东睿.MALK:一种高效处理大规模键值的MapReduce框架.《计算机研究与发展》.2014,

审查员 刘启军

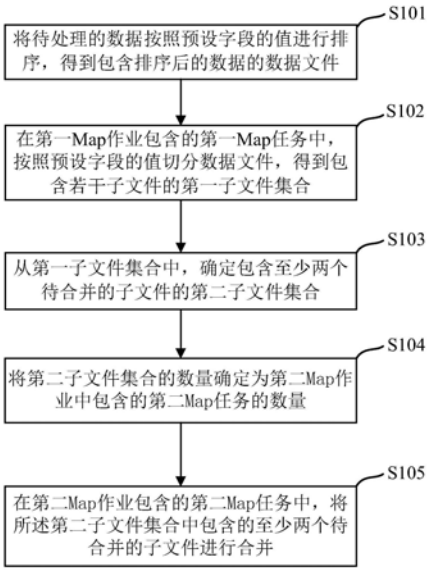
权利要求书3页 说明书9页 附图3页

(54)发明名称

基于MapReduce的数据处理方法及装置

(57)摘要

本申请实施例公开了一种基于MapReduce的数据处理方法及装置,解决现有技术中为MapReduce作业预先配置的任务数无法与实际情况相符的问题。所述方法包括:将待处理的数据按照预设字段的值进行排序,得到包含排序后的数据的数据文件;在第一Map作业包含的第一Map任务中,按照所述预设字段的值切分所述数据文件,得到包含若干子文件的第一子文件集合;从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合;将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量;在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个待合并的子文件进行合并。



CN 106708606 B

1.一种基于MapReduce的数据处理方法,其特征在于,包括:

将待处理的数据按照预设字段的值进行排序,得到包含排序后的数据的数据文件;所述待处理的数据包含多个所述预设字段;

在第一Map作业包含的每个第一Map任务中,按照所述预设字段的值切分所述数据文件,得到包含若干子文件的第一子文件集合;

从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合;

将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量;

在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个待合并的子文件进行合并。

2.根据权利要求1所述的方法,其特征在于,从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合,包括:

确定由至少两个相邻的第一Map任务得到的、且包含的数据在所述预设字段上的值一致的至少两个子文件,得到所述第二子文件集合。

3.根据权利要求1所述的方法,其特征在于,确定所述第二子文件集合之后,将所述第二子文件集合中的子文件进行合并之前,还包括:

将每个第二子文件集合中的包含的至少两个子文件存放于第一预设路径下的同一个子路径中;

将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量,具体包括:

将所述第一预设路径下包含的子路径的数量确定为所述第二Map作业中的Map任务数;

在第二Map作业包含的第二Map任务中,将所述第二子文件集合中的子文件进行合并,具体包括:

在第二Map作业包含的第二Map任务中,读取所述第一预设路径下的同一个子路径中的至少两个子文件并进行合并。

4.根据权利要求1所述的方法,其特征在于,将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量之后,将所述第二子文件集合中的包含的子文件进行合并之前,还包括:

将所述第二Map作业中每个第二Map任务与相应的预设字段的值进行对应;

在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个待合并的子文件进行合并,具体包括:

在第二Map作业包含的每个第二Map任务中,根据该第二Map任务对应的预设字段的值,读取与所述预设字段的值对应的所述第二子文件集合中的至少两个子文件并进行合并。

5.根据权利要求3所述的方法,其特征在于,在第一Map作业包含的第一Map任务中,得到所述第一子文件集合之后,确定所述第二子文件集合之前,还包括:

将在第一Map作业包含的每一个第一Map任务中切分得到的子文件存储于第二预设路径中;

将所述第二子文件集合中的包含的数据在所述预设字段上的值一致的至少两个子文件存放于第一预设路径下的同一个子路径中,具体包括:

将所述第二子文件集合中的包含的数据在所述预设字段上的值一致的至少两个子文

件从所述第二预设路径移到第一预设路径下的同一个子路径中。

6.根据权利要求5所述的方法,其特征在于,将在第一Map作业包含的每一个第一Map任务中切分得到的子文件存储于第二预设路径中,还包括:

确定所述第二预设路径中存放的子文件的文件名中包含该子文件对应的预设字段的值;

确定所述第二预设路径中存放的子文件对应的存储路径名中包含该子文件对应的第一Map任务的ID;

将所述第二子文件集合中的包含的数据在所述预设字段上的值一致的至少两个子文件从所述第二预设路径移到第一预设路径下的同一个子路径中,还包括:

确定所述第一预设路径中的子文件的文件名中包含该子文件对应的第一Map任务的ID;

确定所述第一预设路径中的子文件的对应的子路径名包含该对应的预设字段的值。

7.根据权利要求1所述的方法,其特征在于,所述预设字段是互联网交易平台生成的交易数据中的商家ID。

8.一种基于MapReduce的数据处理装置,其特征在于,包括:

排序单元,用于将待处理的数据按照预设字段的值进行排序,得到包含排序后的数据的数据文件;所述待处理的数据包含多个所述预设字段;

切分单元,用于在第一Map作业包含的每个第一Map任务中,按照所述预设字段的值切分所述数据文件,得到包含若干子文件的第一子文件集合;

第一确定单元,用于从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合;

第二确定单元,用于将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量;

合并单元,用于在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个待合并的子文件进行合并。

9.根据权利要求8所述的装置,其特征在于,所述第一确定单元具体用于:

确定由至少两个相邻的第一Map任务得到的、且包含的数据在所述预设字段上的值一致的至少两个子文件,得到所述第二子文件集合。

10.根据权利要求8所述的装置,其特征在于,所述装置还包括:

第一存储单元,用于将每个第二子文件集合中的包含的至少两个子文件存放于第一预设路径下的同一个子路径中;

所述第二确定单元具体用于:

将所述第一预设路径下包含的子路径的数量确定为所述第二Map作业中的Map任务数;

所述合并单元具体用于:

在第二Map作业包含的第二Map任务中,读取所述第一预设路径下的同一个子路径中的至少两个子文件并进行合并。

11.根据权利要求8所述的装置,其特征在于,所述装置还包括:

对应单元,用于将所述第二Map作业中每个第二Map任务与相应的预设字段的值进行对应;

所述合并单元具体用于：

在第二Map作业包含的每个第二Map任务中，根据该第二Map任务对应的预设字段的值，读取与所述预设字段的值对应的所述第二子文件集合中的至少两个子文件并进行合并。

12.根据权利要求10所述的装置，其特征在于，所述装置还包括：

第二存储单元，用于将在第一Map作业包含的每一个第一Map任务中切分得到的子文件存储于第二预设路径中；

所述第一存储单元具体用于：

将所述第二子文件集合中的包含的数据在所述预设字段上的值一致的至少两个子文件从所述第二预设路径移到第一预设路径下的同一个子路径中。

13.根据权利要求12所述的装置，其特征在于，所述第二存储单元还包括：

第二文件名确定单元，用于确定所述第二预设路径中存放的子文件的文件名中包含该子文件对应的预设字段的值；

第二路径名确定单元，用于确定所述第二预设路径中存放的子文件对应的存储路径名中包含该子文件对应的第一Map任务的ID；

所述第一存储单元还包括：

第一文件名确定单元，用于确定所述第一预设路径中的子文件的文件名中包含该子文件对应的第一Map任务的ID；

第一路径名确定单元，用于确定所述第一预设路径中的子文件的对应的子路径名包含该对应的预设字段的值。

14.根据权利要求8所述的装置，其特征在于，所述预设字段是互联网交易平台生成的交易数据中的商家ID。

基于MapReduce的数据处理方法及装置

技术领域

[0001] 本申请涉及数据仓库技术,特别涉及一种基于MapReduce的数据处理方法及装置。

背景技术

[0002] Hadoop是一个能够对大量数据进行处理的分布式系统基础框架,主要由Hadoop分布式文件系统(Hadoop Distributed File System,HDFS)和映射归约MapReduce组成。其中,MapReduce是一种分布式计算框架,主要用于大规模数据集的并行运算,其主要分为Map(映射)任务和Reduce(归约)任务,Map任务和Reduce任务的处理逻辑分别对应于Map函数和Reduce函数。

[0003] 在一些应用中,利用Hadoop分布式文件系统可以根据一定的规则对大规模数据集中的数据进行合并(聚类)。现有技术中,在进行大规模数据集中的数据合并时,通常是利用一个MapReduce作业来实现,在一个MapReduce作业启动之前,需要预先配置该MapReduce作业中包含的Map任务数和Reduce任务数。其大致过程是:利用Hadoop的数据仓库工具Hive来执行一条sql指令,在HDFS上生成按照一定次序进行排列的数据文件,然后,通过执行Map任务,从HDFS上读取上述数据文件,并按照规则将需要合并的文件存放于同一个路径中;最终,通过执行Reduce任务,将各个路径中存放的需要合并的文件分别作合并。

[0004] 上述现有技术中,由于MapReduce作业包含的任务数均是在MapReduce作业启动之前预先配置好的,而在实际业务运行过程中,待处理的数据量并不是固定的,从而可能导致在MapReduce作业启动之前预先配置的任务数与实际待处理的数据量不匹配,影响机器处理数据的效率。

发明内容

[0005] 本申请实施例的目的是提供一种基于MapReduce的数据处理方法及装置,以解决现有技术中在MapReduce作业启动之前预先配置的任务数与实际待处理的数据量不匹配,影响机器处理数据的效率的问题。

[0006] 为解决上述技术问题,本申请实施例提供的基于MapReduce的数据处理方法及装置是这样实现的:

[0007] 一种基于MapReduce的数据处理方法,包括:

[0008] 将待处理的数据按照预设字段的值进行排序,得到包含排序后的数据的数据文件;

[0009] 在第一Map作业包含的第一Map任务中,按照所述预设字段的值切分所述数据文件,得到包含若干子文件的第一子文件集合;

[0010] 从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合;

[0011] 将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量;

[0012] 在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个

待合并的子文件进行合并。

[0013] 一种基于MapReduce的数据处理装置,包括:

[0014] 排序单元,用于将待处理的数据按照预设字段的值进行排序,得到包含排序后的数据的数据文件;

[0015] 切分单元,用于在第一Map作业包含的第一Map任务中,按照所述预设字段的值切分所述数据文件,得到包含若干子文件的第一子文件集合;

[0016] 第一确定单元,用于从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合;

[0017] 第二确定单元,用于将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量;

[0018] 合并单元,用于在第二Map作业包含的第二Map任务中,将所述第二子文件集合中包含的至少两个待合并的子文件进行合并。

[0019] 由以上本申请各实施例提供的技术方案可见,在第一Map作业包含的每个第一Map任务中,对排序后的数据文件进行切分,得到包含若干子文件的第一子文件集合;并在此之后,确定第一子文件集合中的待合并的第二子文件集合;并根据确定的第二子文件集合的数量来确定第二Map作业包含的第二Map任务的数量;最终,在第二Map作业包含的每个第二Map任务中,将所述第二子文件集合中的包含的数据在预设字段上的值一致的子文件进行合并。在上述过程中,本申请实施例通过两个Reduce任务为零的MapReduce作业(上述第一Map作业和第二Map作业)来实现数据合并,并且第二Map作业中包含的第二Map任务的数量是根据第一Map作业中确定的第二子文件集合的数量(也就是实际需要进行合并的任务数)来确定的。基于上述内容,当待处理的数据量发生变化时,可以动态地根据第一Map作业得到的第二子文件集合数,来调整第二Map作业包含的第二Map任务的数量,从而解决现有技术中在MapReduce作业启动之前,所预先配置的任务数(Map任务数或Reduce任务数)与实际待处理的数据量不匹配的问题,提升机器处理数据的效率。

附图说明

[0020] 为了更清楚地说明本申请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简单地介绍,显而易见地,下面描述中的附图仅仅是本申请中记载的一些实施例,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。

[0021] 图1为本申请一实施例提供的基于MapReduce的数据处理方法的流程图;

[0022] 图2为本申请一实施例中数据处理过程的数据流图;

[0023] 图3为本申请一实施例中第一Map作业的流程图;

[0024] 图4为本申请一实施例提供的基于MapReduce的数据处理装置的模块示意图。

具体实施方式

[0025] 为了使本技术领域的人员更好地理解本申请中的技术方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,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，都应当属于本申请保护的范围。

[0026] 本申请基于MapReduce来实现互联网数据平台上的数据聚类，互联网数据平台可以例如是电子商务平台(E-Business Platform)或第三方支付平台(Third party payment Platform)等。以电子商务平台为例，随着业务发展，每天会产生大量的交易数据。该电子商务平台上的某些商家可以有相应的ERP(Enterprise Resource Planning，企业资源计划)软件平台，在实际应用中，每个商家的ERP软件平台上的交易数据需要与该电子商务平台上的交易数据保持一致。目前的通用做法是，电子商务平台定时或不定时地根据一定时间内的交易数据，生成与每个商家对应的交易明细文件(每个商家对应于一个交易明细文件)，并将生成的这些交易明细文件存放于一个分布式文件系统HDFS中，从而各个商家的ERP软件平台可以从上述HDFS中获取各自的交易明细文件。目前，随着电子商务平台上交易数据量及商家数量的攀升，如何通过MapReduce和分布式文件系统HDFS进行数据聚类，以高效地生成每个商家的交易明细文件，是本申请的技术方案旨在解决的问题。

[0027] 图1为本申请一实施例提供的基于MapReduce的数据处理方法的流程图。图2为本申请一实施例中数据处理过程的数据流图。配合参照图1和图2所示，本申请实施例中，所述方法包括如下步骤：

[0028] S101：将待处理的数据按照预设字段的值进行排序，得到包含排序后的数据的数据文件。

[0029] 本实施例中，本方法可以采用Hadoop系统，并利用分布式文件系统HDFS来提取数据和存储数据。其中，所述预设字段是根据实际的数据合并(聚类)需求而预先指定的用来作为数据排序操作的依据。

[0030] 以电子商务平台上的交易数据的合并为例。电子商务平台可以将一段时间内产生的交易数据存放在数据仓库(Hadoop)中，可以利用Hadoop的数据仓库工具Hive来执行一条Sql语句(指令)，从而对数据仓库中存放的无序数据进行有序排列。

[0031] 上述sql语句例如是：

[0032] “INSERT OVERWRITE DIRECTORY‘DIR1’SELECT C1 C2…FROM T DISTRIBUT BY C1 SORT BY C1”。

[0033] 其中，对于一笔交易数据而言，可以包含多个字段，例如，C1：商家ID，C2：商品名称/ID，C3：交易金额。通过执行上述sql语句，可以将数据仓库中存储的交易数据按照预设字段(C1：商家ID)的值进行排序，得到包含排序的交易数据的数据文件，并可以将得到的数据文件存放在HDFS中的存储路径DIR1下。

[0034] 参照图2所示，例如，Hadoop存储的原始交易数据(以商家ID来标识)包括：

[0035] {商家1、商家2、商家3、商家2、商家n、商家1、商家2、商家3、商家2、商家n、商家1、商家2、商家3、商家2、商家n}；

[0036] 通过数据仓库工具Hive来执行一条Sql语句进行排序，得到的交易数据包括：

[0037] {商家1、商家1、商家1、商家1、商家1、商家1、商家2、商家2、商家2、商家3、商家3、商家3、商家n、商家n、商家n}；

[0038] 在存储路径DIR1中，可以得到多个按照一定的大小进行均分的数据文件，例如，0.TXT，1.TXT，2.TXT(文件名)。当然，在该存储路径DIR1中也可以只存储一个包含排序后的

交易数据的数据文件。

[0039] 值得一提的是，对数据仓库中的原始交易数据进行排序的具体过程并不限于上述内容。

[0040] S102：在第一Map作业包含的第一Map任务中，按照所述预设字段的值切分所述数据文件，得到包含若干子文件的第一子文件集合；其中，每个所述子文件中包含的数据在所述预设字段上的值一致。

[0041] 本实施例中，生成第一Map作业(Reduce任务数为零的MapReduce作业)，该第一Map作业包含多个第一Map任务，并且该第一Map作业的数据来源是上述存储路径DIR1中的数据文件。每个第一Map任务的分片可以默认为例如64M的数据。在每个第一Map任务中，通过调用Map函数来从上述存储路径DIR1中读取数据，以按照上述预设字段的值将DIR1中的数据文件进行切分，得到包含多个子文件的第一子文件集合。

[0042] 参照图3所示，为该步骤S102的具体过程，包括：

[0043] S1021：逐一读取排序得到的数据文件中的数据。

[0044] S1022：判断读取的数据与缓存中存放的上一个数据在预设字段上的值是否一致。

[0045] S1023：若一致，则将读取的数据写入缓存中。

[0046] S1024：若不一致，则将当前缓存的数据全部写入HDFS中(作为一个子文件)。

[0047] S1025：判断第一Map任务是否结束，若是，转而执行步骤S103；若否，返回到步骤S1021。

[0048] 继续参照图2，举例而言，若第一Map作业包含的第一Map任务是：Task0和Task1(Map任务ID)，其中，假设Task0读取的数据包含数据文件0.TXT中的全部数据、及数据文件1.TXT中的部分数据；Task1读取的数据包含数据文件1.TXT中的部分数据、及数据文件2.TXT中的全部数据。

[0049] 通过执行上述步骤S1021～S1025，可以通过Task0得到两个子文件(本文列举的子文件以预设字段的值来标识)例如是：

[0050] 商家1.TXT(文件名)，包含的数据是：{商家1、商家1、商家1、商家1、商家1、商家1}；

[0051] 商家2.TXT(文件名)，包含的数据是：{商家2}。

[0052] 通过Task1可以得到三个子文件例如是：

[0053] 商家2.TXT(文件名)，包含的数据是：{商家2、商家2}；

[0054] 商家3.TXT(文件名)，包含的数据是：{商家3、商家3、商家3}；

[0055] 商家n.TXT(文件名)，包含的数据是：{商家n、商家n、商家n}。

[0056] 可以看出，在该第一Map作业中包含的每个第一Map任务中，可以分别将排序得到的数据文件进行切分，得到多个子文件，并且每个子文件中包含的数据在预设字段上的值是一致的，也就是说，一个子文件中只包含一个商家ID的数据。从而得到的第一子文件集合Q1例如是：

[0057] {商家1.TXT、商家2.TXT、商家2.TXT、商家3.TXT、商家n.TXT、……}。

[0058] 优选地，本申请实施例中，可以切分得到的上述第一子文件集合中的数据存放于HDFS中存储路径DIR2(本文定义为第二预设路径)下。并且，每个子文件在第二预设路径DIR2中的存储路径(子路径)的路径名可以包含该子文件对应的第一Map任务的ID，该第二预设路径DIR2中存放的子文件的文件名中包含该子文件对应的预设字段的值。例如：由

Task0切分得到的“商家1.TXT”存放的存储路径的路径名是DIR2/0,由Task0切分得到的“商家2.TXT”存放的存储路径的路径名是DIR2/0,……。

[0059] 值得一提的是,上述得到的第一子文件集合的存储方式和存储路径的命名方式、文件名的命名方式均并不限于上述内容。

[0060] S103:从所述第一子文件集合中,确定包含至少两个待合并的子文件的第二子文件集合。

[0061] 在上述步骤S102中,由于第一Map作业中包含多个第一Map任务,并且可能导致相邻的两个或两个以上的第一Map任务得到的子文件包含的数据在预设字段上的值一致(例如,Task0得到的子文件“商家2.TXT”和Task1得到的子文件“商家2.TXT”在预设字段(商家ID)上的值同是“商家2”),则可以确定这两个子文件是同一个商家的交易数据文件,即是待合并处理的子文件。

[0062] 一般地,可以判断前一个第一Map任务切分得到的最后一个子文件与后一个第二Map任务切分得到的第一个子文件在上述预设字段上的值是否一致,若一致,则确定这两个子文件属于待合并的子文件。在上述例子中,得到的待合并的文件是由两个相邻的两个第一Map任务分别切分得到的两个子文件。然而,在其他的例子中,待合并的文件也可以是相邻的三个或更多个的第一Map任务切分得到的三个或更多个子文件。在实际应用过程中,一个商家的交易数据可能会非常多,这样可能导致由更多个Map任务来针对这一个商家的交易数据来作切分。通过遍历上述第一子文件集合Q1,可以确定所有待合并的子文件,若将每一组待合并的子文件(至少两个)确定为一个第二子文件集合Q2,则根据遍历得到的多组待合并的子文件,可以得到多个第二子文件集合Q2(第二子文件集合的数量与待合并的子文件的组数相等)。举例而言,得到的第二子文件集合Q2可以包括:

[0063] {存储在DIR2/0下的“商家2.TXT”,存储在DIR2/1下的“商家2.TXT”};

[0064] {存储在DIR2/4下的“商家6.TXT”,存储在DIR2/5下的“商家6.TXT”,存储在DIR2/6下的“商家6.TXT”};

[0065] ……

[0066] S104:将第二子文件集合Q2的数量确定为第二Map作业中包含的第二Map任务的数量。

[0067] 在本申请可选的实施例中,可以将每个第二子文件集合Q2中的包含的至少两个子文件分别存放于第一预设路径DIR3下的同一个子路径中。相应地,当第一Map作业结束,并且将每个第二子文件集合Q2中包含的待合并的子文件存放于第一预设路径DIR3中的各个子路径下之后,可以根据上述DIR3中子路径的数量来确定第二Map作业中Map任务数。

[0068] 在本申请优选的实施例中,可以将每个第二子文件集合Q2中包含的至少两个子文件从第二预设路径DIR2移到第一预设路径DIR3下的同一个子路径中。当然,在替代的方案中,在未确定得到待合并的子文件时,可以不对第一Map任务得到的子文件进行存储,而是在确定得到待合并的子文件时,再分别对待合并的子文件及不需进行合并的子文件分别进行存储,其存储方式和存储路径均不受限制。

[0069] 通过将确定的待合并的第二子文件集合Q2中的子文件从DIR2移动到DIR3中,并且按照预设字段的值,在DIR3下的同一个子路径中存放同一个商家ID的子文件。为了便于实现第二Map作业的合并操作,存储Q2中的子文件的步骤还包括:

[0070] 确定第一预设路径DIR3中的子文件的文件名中包含该子文件对应的第一Map任务的ID;确定第一预设路径DIR3中的子文件的对应的子路径名包含该对应的预设字段的值。

[0071] 在上述例子中,确定得到第二子文件集合Q2后,可以在DIR3中逐一新建一个子路径,例如:“商家2”(存储路径名),并将待合并的原本存放于DIR/0中的子文件“商家2.TXT”和原本存放于DIR/1中的子文件“商家2.TXT”移动到DIR3/商家2中,并且将这两个子文件的文件名分别修改为“0.TXT”、“1.TXT”。依次类推,根据待合并的子文件的数量,得到对应的DIR3的所有子路径。如前所述,DIR3中的存储方式和存储路径的命名方式、文件名的命名方式均并不限于上述内容。

[0072] S105:在第二Map作业包含的第二Map任务中,将第二子文件集合Q2中包含的至少两个待合并的子文件进行合并。

[0073] 在第二Map作业中,以DIR3中的数据作为数据来源,每个第二Map任务将上述DIR3中一个子路径作为一个Map方法的分片,每个第二Map任务通常只需调用一次Map函数,读取当前子路径中的所有待合并的子文件,并按照文件名进行排序,最后对排序后的文件进行合并,得到一个合并后的与某商家ID对应的交易数据文件。

[0074] 至此,可以通过第一Map作业和第二Map作业实现对互联网中的待处理数据按照预设字段的值进行合并。最终,可以将合并得到的各个商家ID的交易数据文件存放于上述第二预设路径DIR2中(因为DIR2中包含不需作合并的其余商家ID对应的交易数据文件),从而可以合并得到电子商务平台上所有商家ID对应的交易数据文件,以供各个商家的ERP软件平台进行提取。

[0075] 在本申请其他可行的实施例中,上述步骤S104之后,步骤S105之前,所述方法还包括:将所述第二Map作业中每个第二Map任务与相应的预设字段的值进行对应。相应地,上述步骤S105可以具体包括:在第二Map作业包含的每个第二Map任务中,根据该第二Map任务对应的预设字段的值,读取与所述预设字段的值对应的所述第二子文件集合中的至少两个子文件并进行合并。

[0076] 在第一Map作业中,可以将确定的第二子文件集合Q2中包含的待合并的子文件存放在同一个大的存储路径中(不切分各个子路径),对于同一个商家ID对应的多个待合并的子文件,在其文件的命名上,可以例如是:“商家2-1.TXT”,“商家2-2.TXT”,……。从而,在第二Map作业中包含的每个第二Map任务中,可以通过预先将每个第二Map任务与相应的预设字段的值进行对应,从而分别指定每个第二Map任务应该读取的第二子文件集合Q2。例如:通过配置,第二Map任务Task20读取的子文件的文件名中包含商家ID“商家2”,第二Map任务Task21读取的子文件的文件名中包含商家ID“商家6”,等等。通过这样的机制,可以通过每个第二Map任务,分别从上述大的存储路径中,按照顺序读取到需要合并的某个商家ID(预设字段)的所有待合并的子文件并作合并处理。本申请替代的实施例并不限于上述列举的内容,不再一一列举。

[0077] 与上述方法流程对应的,本申请的实施例还提供了一种基于MapReduce的数据处理装置。该装置可以通过软件实现,也可以通过硬件或者软硬件结合的方式实现。以软件实现为例,作为逻辑意义上的装置,是通过服务器的中央处理器(Central Process Unit,CPU)将对应的计算机程序指令读取到内存中运行形成的。

[0078] 图4为本申请一实施例提供的基于MapReduce的数据处理装置的模块示意图。其

中，该装置中各单元的功能与上述方法中各步骤的功能类似，故该装置可以参照上述方法实施例的具体内容。该装置包括：

[0079] 排序单元101，用于将待处理的数据按照预设字段的值进行排序，得到包含排序后的数据的数据文件；

[0080] 切分单元102，用于在第一Map作业包含的第一Map任务中，按照所述预设字段的值切分所述数据文件，得到包含若干子文件的第一子文件集合；

[0081] 第一确定单元103，用于从所述第一子文件集合中，确定包含至少两个待合并的子文件的第二子文件集合；

[0082] 第二确定单元104，用于将所述第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量；

[0083] 合并单元105，用于在第二Map作业包含的第二Map任务中，将所述第二子文件集合中包含的至少两个待合并的子文件进行合并。

[0084] 本申请实施例中，所述第一确定单元103具体用于：

[0085] 确定由至少两个相邻的第一Map任务得到的、且包含的数据在所述预设字段上的值一致的至少两个子文件，得到所述第二子文件集合。

[0086] 本申请实施例中，所述装置还包括：

[0087] 第一存储单元，用于将每个第二子文件集合中的包含的至少两个子文件存放于第一预设路径下的同一个子路径中；

[0088] 所述第二确定单元104具体用于：

[0089] 将所述第一预设路径下包含的子路径的数量确定为所述第二Map作业中的Map任务数。

[0090] 所述合并单元105具体用于：

[0091] 在第二Map作业包含的第二Map任务中，读取所述第一预设路径下的同一个子路径中的至少两个子文件并进行合并。

[0092] 本申请实施例中，所述装置还包括：

[0093] 对应单元，用于将所述第二Map作业中每个第二Map任务与相应的预设字段的值进行对应；

[0094] 所述合并单元105具体用于：

[0095] 在第二Map作业包含的每个第二Map任务中，根据该第二Map任务对应的预设字段的值，读取与所述预设字段的值对应的所述第二子文件集合中的至少两个子文件并进行合并。

[0096] 本申请实施例中，所述装置还包括：

[0097] 第二存储单元，用于将在第一Map作业包含的每一个第一Map任务中切分得到的子文件存储于第二预设路径中；

[0098] 所述第一存储单元具体用于：

[0099] 将所述第二子文件集合中的包含的数据在所述预设字段上的值一致的至少两个子文件从所述第二预设路径移到第一预设路径下的同一个子路径中。

[0100] 本申请实施例中，所述第二存储单元还包括：

[0101] 第二文件名确定单元，用于确定所述第二预设路径中存放的子文件的文件名中包

含该子文件对应的预设字段的值；

[0102] 第二路径名确定单元，用于确定所述第二预设路径中存放的子文件对应的存储路径名中包含该子文件对应的第一Map任务的ID；

[0103] 所述第一存储单元还包括：

[0104] 第一文件名确定单元，用于确定所述第一预设路径中的子文件的文件名中包含该子文件对应的第一Map任务的ID；

[0105] 第一路径名确定单元，用于确定所述第一预设路径中的子文件的对应的子路径名包含该对应的预设字段的值。

[0106] 本申请实施例中，所述预设字段是互联网交易平台生成的交易数据中的商家ID。

[0107] 综上，由以上本申请各实施例提供的技术方案可见，在第一Map作业包含的每个第一Map任务中，对排序后的数据文件进行切分，得到包含若干子文件的第一子文件集合；并在此之后，确定第一子文件集合中的待合并的第二子文件集合；并根据确定的第二子文件集合的数量来确定第二Map作业包含的第二Map任务的数量；最终，在第二Map作业包含的每个第二Map任务中，将所述第二子文件集合中的包含的数据在预设字段上的值一致的子文件进行合并。在上述过程中，本申请实施例通过两个Reduce任务为零的MapReduce作业(上述第一Map作业和第二Map作业)来实现数据合并，并且第二Map作业中包含的第二Map任务的数量是根据第一Map作业中确定的第二子文件集合的数量(也就是实际需要进行合并的任务数)来确定的。基于上述内容，当待处理的数据量发生变化时，可以动态地根据第一Map作业得到的第二子文件集合数，来调整第二Map作业包含的第二Map任务的数量，从而解决现有技术中在MapReduce作业启动之前，所预先配置的任务数(Map任务数或Reduce任务数)与实际待处理的数据量不匹配的问题，例如，在现有技术中一个MapReduce作业启动之前，预先配置的Map任务数是N，Reduce任务数是M，而一般情况下，Map任务数远大于Reduce任务数，这样便可能导致预设的Reduce任务数过少，进而导致一个Reduce任务需要对两个或两个以上的商家数据进行合并，从而影响数据处理的效率。本申请实施例通过上述两个Map作业，来确保在第二Map作业中，每个第二Map任务都只是针对一个商家的数据进行合并，从而提升机器处理数据的效率。

[0108] 为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

[0109] 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。

[0110] 本发明是参照根据本发明实施例的方法、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。

[0111] 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。

[0112] 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。

[0113] 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0114] 本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请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，本申请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。

[0115] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0116] 本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[0117] 以上所述仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。

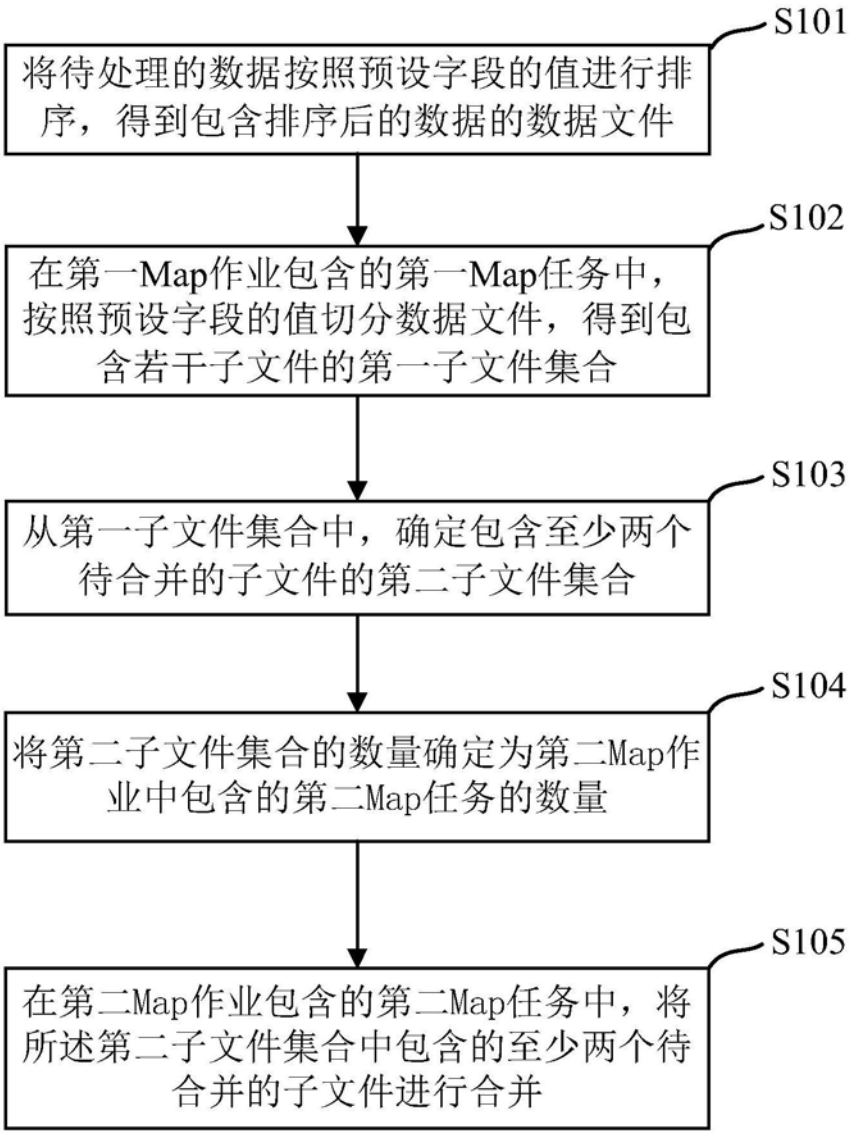

S101
将待处理的数据按照预设字段的值进行排序，得到包含排序后的数据的数据文件
S102
在第一Map作业包含的第一Map任务中，按照预设字段的值切分数据文件，得到包含若干子文件的第一子文件集合
S103
从第一子文件集合中，确定包含至少两个待合并的子文件的第二子文件集合
S104
将第二子文件集合的数量确定为第二Map作业中包含的第二Map任务的数量
S105
在第二Map作业包含的第二Map任务中，将所述第二子文件集合中包含的至少两个待合并的子文件进行合并

图1

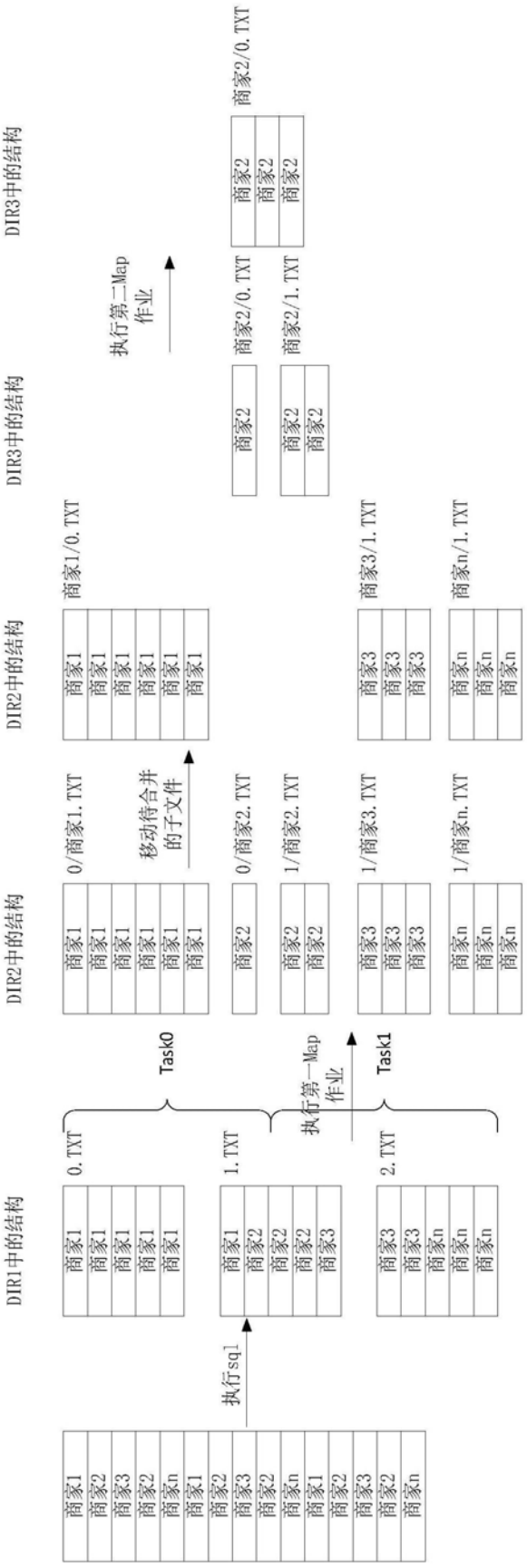

DIR1中的结构
DIR2中的结构
DIR2中的结构
DIR3中的结构
DIR3中的结构
执行sql
0.TXT
1.TXT
2.TXT
Task0
Task1
执行第一Map作业
0/商家1.TXT
0/商家2.TXT
1/商家2.TXT
1/商家3.TXT
1/商家n.TXT
移动待合并的子文件
商家1/0.TXT
商家3/1.TXT
商家n/1.TXT
执行第二Map作业
商家2/0.TXT
商家2/1.TXT
商家2/0.TXT

图2

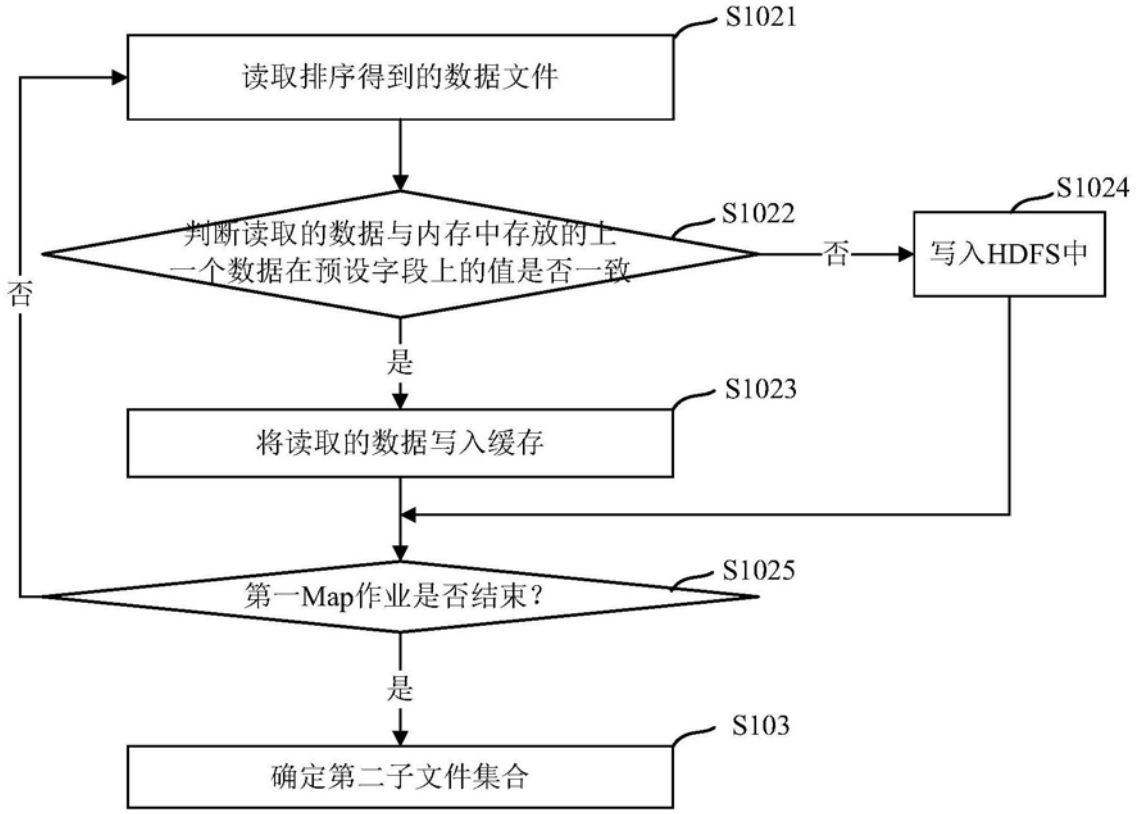

S1021
读取排序得到的数据文件
S1022
判断读取的数据与内存中存放的上一个数据在预设字段上的值是否一致
否
S1024
写入HDFS中
否
是
S1023
将读取的数据写入缓存
S1025
第一Map作业是否结束？
是
S103
确定第二子文件集合

图3

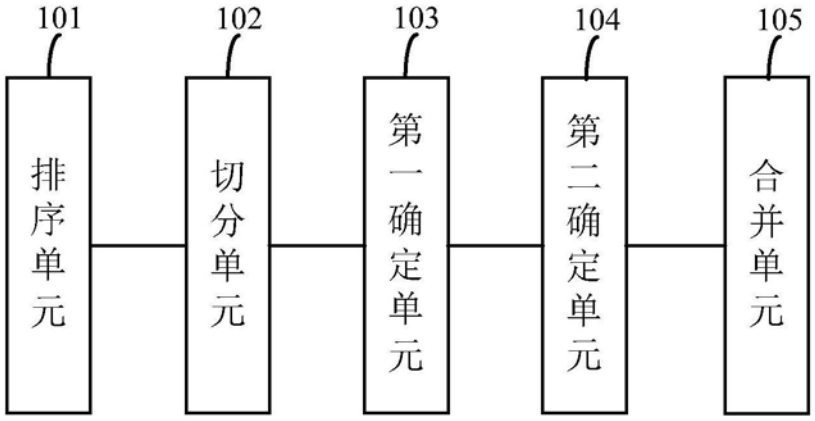

101
排序单元
102
切分单元
103
第一确定单元
104
第二确定单元
105
合并单元

图4