



(12) 发明专利申请

(10) 申请公布号 CN 103646073 A

(43) 申请公布日 2014. 03. 19

(21) 申请号 201310667847. 0

(22) 申请日 2013. 12. 11

(71) 申请人 浪潮电子信息产业股份有限公司
地址 250014 山东省济南市高新区舜雅路
1036 号

(72) 发明人 郭美思 吴楠

(51) Int. Cl.
G06F 17/30 (2006. 01)

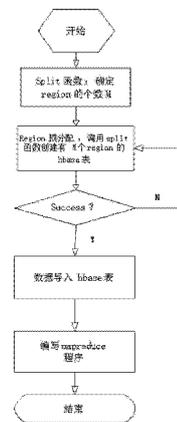
权利要求书1页 说明书6页 附图2页

(54) 发明名称

一种基于 HBase 表的条件查询优化方法

(57) 摘要

本发明提供一种基于 HBase 表的条件查询优化方法, 针对分布式计算框架对数据的分析处理能力及并行计算的特点, 把 HBase 表的条件查询基于 MapReduce 计算框架, 提高查询效率。该优化方法主要通过 Region 预分配、RowKey 设计及 MapReduce 模块来实现。该一种基于 HBase 表的条件查询优化方法和现有技术相比, 提高了条件查询的效率, 为条件查询提供了优化的方法; 该方法的条件查询过程较为简洁, 且支持并行计算, 因此更为高效, 实用性强, 易于推广。



1. 一种基于 HBase 表的条件查询优化方法,其特征在于该方法的具体实施过程为:

根据表格的数据量大小及集群的配置来确定预分配表分区 Region 个数:Region 预分配中,根据导入 HBase 的数据量及分布式集群的规模确定 Region 的个数,然后按数据量、行键 Row Key 的规则预先设计并分配好 Region,HBase 的 Region 随着大小的不断变大会触发一个阈值,一旦触发,Region 会自动分裂,较多的 Region 会保证并发性能;

根据条件查询的应用来设计合理的 RowKey:如果记录的 Row Key 落在某个 Region 的 start key 和 end key 的范围之内,该数据就会存储到这个 Region 上;

根据 Region 预分配、RowKey 和分布式编程框架来提高查询性能:MapReduce 计算框架中,Map 模块需要根据条件查询 HBase 表中符合的记录,根据条件查询的特点及 RowKey 设计出条件查询的优化方案,利用 Scan 中的 StartKey 和 EndKey 参数提高查找性能,同时根据分布式集群参数合理配置 Map 数量,达到优化效果。

2. 根据权利要求 1 所述的一种基于 HBase 表的条件查询优化方法,其特征在于:所述 MapReduce 编程框架是最后得到条件查询结果的处理过程:Map 模块处理了根据条件查询 hbase 表,最终得到查询的记录,再对查询的记录进行处理,得到想要的查询记录格式,并提供遍历的编程接口。

3. 根据权利要求 2 所述的一种基于 HBase 表的条件查询优化方法,其特征在于:所述优化方法的详细步骤为:

首先在表分区 Region 预分配中,根据集群中的环境及配置,设置 zookeeper 的属性,创建预导入的 HBase 表名及列族名称,然后根据编写的 Split 函数创建 HBase 表;其中 Split 函数是根据源数据格式来确定 Split 的个数,通过二维数组来表示 region 的个数;待 Region 预分配结束,可以根据 RowKey 的设计将源数据生成 Hfile 文件;最后将用 completebulkload 命令完成数据的导入,这时数据按照预定的格式导入到了 HBase 表;

编写 MapReduce 程序:该用户编写的 MapReduce 程序是一个作业,用户配置并将一个作业提交到框架中,框架会将这个作业分解成一些列的 Map tasks 和 reduce tasks;框架负责 task 分发和执行;在此查询优化中,需要完成 map 阶段的工作,最后将符合用户的查询结果输出,该输出过程为:

根据用户程序指定的查询条件,首先根据输入的查询参数进行格式的处理,将查询参数拼接成符合 HBase 表中行键 RowKey 的格式,再根据 scan 中的 startKey 和 endKey 来设置查询的起始位置,以及 scan 中的参数设置;然后将根据 initTableMapperJob 函数提交到 Hadoop 框架中,在此过程中框架 split 的每一个数据块对应一个 Map 任务;对于每一个 Map 任务,按照迭代的方式,将每一个数据记录按照 Map 函数中指定的方法进行处理:

在 map 函数中,首先根据 value.getRow() 得到符合条件查询的 RowKey,根据 value.value() 得到第一列的数据;然后将提取的 RowKey 及第一列的数据进行处理转换成用户期待的输出格式,可以根据 substring 函数截取需要的字符串并设定成预定的格式,直接将符合条件的结果输出到指定的文件中,并通过 setOutputPath 函数来设定输出的路径。

一种基于 HBase 表的条件查询优化方法

技术领域

[0001] 本发明涉及通信信息技术领域,具体的说是一种基于 HBase 表的条件查询优化方法。

背景技术

[0002] 随着数据的海量剧增,大数据已然成为了当今的焦点。在大数据所需的技术中,分布式文件系统、分布式数据库等都是适用于大数据的技术。HBase 是一个分布式的、面向列的开源数据库。它是利用 Hadoop HDFS 作为其文件存储系统。随着 HBase 在性能和稳定性持续改善, HBase 逐渐成为在大数据 NoSQL 领域的标准之一。HBase 已经被很多大公司采用,如 Facebook, Twitter, Adobe, Cloudera, IBM 等等。因此,基于 HBase 表的条件查询优化是非常重要的。

[0003] 目前对于 HBase 的查询实现有两种方式:一种是按指定 RowKey 获取唯一一条记录的 Get 方法;另一种是按指定的条件获取一批记录的 Scan 方法。其中实现条件查询功能使用的是 Scan 方法,scan 可以通过 setCaching 与 setBatch 方法提高速度(以空间换时间);同时也可以根据 setStartRow 与 setEndRow 来限定范围。范围越小,性能越高。通过巧妙的 RowKey 设计使批量获取记录集中的元素集中在一起(应该在同一个 Region 下),可以在遍历结果时获得很好的性能。

[0004] 在分布式计算框架中,并行计算在效率上有很大提高,并且分布式文件系统 HDFS 能保存大量的数据,可扩展性强。基于上述原因,提出一种基于 HBase 表格的条件查询优化方法。该方案采用 Region 预分配与设计合理的 RowKey 来将源数据导入到 HBase 表中。并利用 MapReduce 框架来实现 HBase 表的条件查询,提高查询效率,并通过设置集群中相应的配置参数达到优化的效果。

发明内容

[0005] 本发明的技术任务是解决现有技术的不足,提供一种基于 HBase 表的条件查询优化方法。

[0006] 本发明的技术方案是按以下方式实现的,该一种基于 HBase 表的条件查询优化方法,该方法的具体实施过程为:

根据表格的数据量大小及集群的配置来确定预分配表分区 Region 个数;Region 预分配中,根据导入 HBase 的数据量及分布式集群的规模确定 Region 的个数,然后按数据量,Row Key 的规则预先设计并分配好 Region, HBase 的 Region 随着大小的不断变大会触发一个阈值,一旦触发,Region 会自动分裂;较多的 Region 会保证并发性能;

根据条件查询的应用来设计合理的 RowKey:如果记录的 Row Key 落在某个 Region 的 start key 和 end key 的范围之内,该数据就会存储到这个 Region 上;

根据 Region 预分配、RowKey 和分布式编程框架来提高查询性能:MapReduce 计算框架中,Map 模块需要根据条件查询 HBase 表中符合的记录,根据条件查询的特点及 RowKey 设

计算出条件查询的优化方案,利用 Scan 中的 StartKey 和 EndKey 参数提高查找性能,同时根据分布式集群参数合理配置 Map 数量,达到优化效果。

[0007] 所述 MapReduce 编程框架是最后得到条件查询结果的处理过程:Map 模块处理了根据条件查询 hbase 表,最终得到查询的记录,再对查询的记录进行处理,得到想要的查询记录格式,并提供遍历的编程接口。

[0008] 所述优化方法的详细步骤为:

首先在表分区 Region 预分配中,根据集群中的环境及配置,设置 zookeeper 的属性,创建预导入的 HBase 表名及列族名称,然后根据编写的 Split 函数创建 HBase 表;其中 Split 函数是根据源数据格式来确定 Split 的个数,通过二维数组来表示 region 的个数;待 Region 预分配结束,可以根据 RowKey 的设计将源数据生成 Hfile 文件;最后将用 completebulkload 命令完成数据的导入,这时数据按照预定的格式导入到了 HBase 表。

[0009] 编写 MapReduce 程序:该用户编写的 MapReduce 程序是一个作业,用户配置并将一个作业提交到框架中,框架会将这个作业分解成一些列的 Map tasks 和 reduce tasks;框架负责 task 分发和执行;在此查询优化中,需要完成 map 阶段的工作,最后将符合用户的查询结果输出,该输出过程为:

根据用户程序指定的查询条件,首先根据输入的查询参数进行格式的处理,将查询参数拼接成符合 HBase 表中行键 RowKey 的格式,再根据 scan 中的 startKey 和 endKey 来设置查询的起始位置,以及 scan 中的参数设置;然后将根据 initTableMapperJob 函数提交到 Hadoop 框架中,在此过程中框架 split 的每一个数据块对应一个 Map 任务;对于每一个 Map 任务,按照迭代的方式,将每一个数据记录按照 Map 函数中指定的方法进行处理。

[0010] 在 map 函数中,首先根据 value.getRow() 得到符合条件查询的 RowKey,根据 value.value() 得到第一列的数据;然后将提取的 RowKey 及第一列的数据进行处理转换成用户期待的输出格式,可以根据 substring 函数截取需要的字符串并设定成预定的格式,直接将符合条件的结果输出到指定的文件中,并通过 setOutputPath 函数来设定输出的路径。

[0011] 本发明与现有技术相比所产生的有益效果是:

本发明的一种基于 HBase 表的条件查询优化方法具有并行计算能力,通过对 RowKey 的合理设计及预分配 region,并实现了条件查询的 map 接口,达到优化的目的:Hadoop 分布式集群的 HDFS 提供了充分的存储能力,map 任务可以通过数据就近获取来减少数据传输消耗的资源;在 Hadoop 分布式集群中,可以设置 map 数量及 HBase 中相应的参数来并行处理作业,提高了条件查询的效率,为条件查询提供了优化的方法;该方法的条件查询过程较为简洁,且支持并行计算,因此更为高效,实用性强,易于推广。

附图说明

[0012] 附图 1 为本发明 HBase 表条件查询优化的执行流程图。

[0013] 附图 2 为 Hregion Server 的结构图。

[0014] 附图 3 为无 reduce 阶段的计算框架执行流程图。

具体实施方式

[0015] 下面结合附图对本发明的一种基于 HBase 表的条件查询优化方法作以下详细说明。

[0016] HBase 是一个分布式的、面向列的开源数据库。它是利用 Hadoop HDFS 作为其文件存储系统。随着 HBase 在性能和稳定性持续改善, HBase 逐渐成为在大数据 NoSQL 领域的标准之一。因此, 基于 HBase 表的查询优化是非常重要的。针对 HBase 表的条件查询优化, 本发明提供一种基于 HBase 表的条件查询优化方法, 主要涉及建表时 Region 的预分配, RowKey 的设计, 查询时方案设计三方面。根据表格的数据量大小及集群的配置来确定预分配 Region 个数, 并根据 RowKey 的合理设计(根据条件查询设计合理的 RowKey)将源数据均匀分布到这些 Region 中。为 HBase 表的条件查询做好准备工作, 并提供合理的查询环境; 根据条件查询的应用来设计合理的 RowKey ;Map 阶段 :根据 Region 预分配、RowKey 和分布式编程框架来提高查询性能, 达到优化效果, 该阶段是根据条件查询传入的参数按照 Scan 的查询结果做加工处理, 将符合条件查询的记录处理成用户期待的格式, 因为没有 Reduce 阶段, 因此可以直接将结果输出到输出目录中, 减少传输的带宽限制。

[0017] 该方法的具体实施过程为 :

根据表格的数据量大小及集群的配置来确定预分配表分区 Region 个数 :Region 预分配中, 根据导入 HBase 的数据量及分布式集群的规模确定 Region 的个数, 然后按数据量, Row Key 的规则预先设计并分配好 Region, 可以大幅降低 Region Split 的次数, 甚至不 Split。HBase 的 Region 随着大小的不断变大会触发一个阈值, 一旦触发, Region 会自动分裂。较多的 Region 会保证并发性能。因此, 根据分布式集群环境的规模来合理的确定 region 的个数, 编程实现符合应用的 Region 个数, 提高并发性能。

[0018] 根据条件查询的应用来设计合理的 RowKey :如果记录的 Row Key 落在某个 Region 的 start key 和 end key 的范围之内, 该数据就会存储到这个 Region 上。如果在某个时段内, 很多数据的 row key 都处在某个特定的 row key 范围内。那这个特定范围 row key 对应的 region 会非常繁忙, 而其他的 region 很可能非常的空闲, 导致资源浪费, 影响性能。因此, 在 RowKey 设计时要参考导入数据的应用及尽量保证单位时间内写入数据的 row key 对于 region 呈均匀分布。

[0019] 根据 Region 预分配、RowKey 和分布式编程框架来提高查询性能 :MapReduce 计算框架中, Map 模块需要根据条件查询 HBase 表中符合的记录, 根据条件查询的特点及 RowKey 设计出条件查询的优化方案。利用 Scan 中的 StartKey 和 EndKey 参数提高查找性能, 同时根据分布式集群参数合理配置 Map 数量, 保证达到优化的效果。

[0020] 所述 MapReduce 编程框架, 用于大规模数据集(大于 1TB)的并行运算, 是最后得到条件查询结果的处理过程。Map 模块处理了根据条件查询 hbase 表, 最终得到查询的记录, 再对查询的记录进行处理, 得到想要的查询记录格式。不使用 Reducer, 可以减少集群中带宽的限制。将处理结果输出到指定的输出目录中。该过程可以设置多个 map 数量, 提高了处理效率, 大大提升了性能。MapReduce 框架简化了并行处理程序时的编程麻烦, 提供了遍历的编程接口。

[0021] 所述优化方法的详细步骤为 :

首先在 Region 预分配中, 根据集群中的环境及配置, 设置 zookeeper 的属性, 创建预导入的 HBase 表名及列族名称, 然后根据编写的 Split 函数创建 HBase 表。其中 Split 函数

是根据源数据格式来确定 Split 的个数,通过二维数组来表示 region 的个数。待 Region 预分配结束,可以根据 RowKey 的设计将源数据生成 Hfile 文件,其中由于对数据分析并合理的设计 RowKey,这就保证了每个 region 中的数据都是均匀的,不会发生哪个 Region 数据很多或者哪个 Region 数据很少的现象。最后将用 completebulkload 命令完成数据的导入,这时数据按照预定的格式导入到了 HBase 表。

[0022] 上述技术方案中的 ZooKeeper 是 Hadoop 的正式子项目,它是一个针对大型分布式系统的可靠协调系统,提供的功能包括:配置维护、名字服务、分布式同步、组服务等。ZooKeeper 的目标就是封装好复杂易出错的关键服务,将简单易用的接口和性能高效、功能稳定的系统提供给用户。

[0023] 根据 HBase 表来条件查询优化方法是通过编写 MapReduce 程序实现的。用户编写的 MapReduce 程序是一个作业,用户配置并将一个作业提交到框架中,框架会将这个作业分解成一些列的 Map tasks 和 reduce tasks。框架负责 task 分发和执行。在此查询优化中,需要完成 map 阶段的工作,最后将符合用户的查询结果输出。

[0024] 基于 HBase 表的条件查询优化主要实现了 Map 接口。该模块主要处理流程是:根据用户程序指定的查询条件,首先根据输入的查询参数进行格式的处理,将查询参数拼接成符合 HBase 表中 RowKey 的格式,再根据 scan 中的 startKey 和 endKey 来设置查询的起始位置,以及 scan 中的参数设置,如 Batch 及 Caching 等。然后将根据 initTableMapperJob 函数提交到 Hadoop 框架中,在此过程中框架 split 的每一个数据块对应一个 Map 任务。对于每一个 Map 任务,按照迭代的方式,将每一个数据记录按照 Map 函数中指定的方法进行处理。在 map 函数中,首先根据 value.getRow() 得到符合条件查询的 RowKey,根据 value.value() 得到第一列的数据。然后将提取的 RowKey 及第一列的数据进行处理转换成用户期待的输出格式,可以根据 substring 函数截取需要的字符串并设定成预定的格式,为了保证性能,防止网络带宽的限制,不适用 Reduce 函数,直接将符合条件的结果输出到指定的文件中,可以通过 setOutputPath 函数来设定输出的路径。

[0025] 根据分布式框架可以提高并行度,然后根据搭建的 Hadoop 集群规模设置 map 任务数及 HBase 中的相关读操作的参数配置,如 caching 设置大一些有利于读。Batch 设置大一些有利于一次可以抓取多条数据,通过这些参数的合理设置可以提高性能,达到优化的目的。

[0026] 具体实施方式如附图 1、图 2、图 3 所示,其具体操作过程为:

首先部署分布式集群环境,该集群中的硬件环境是 7 台服务器,每台服务器是 96G 内存,cpu 有 24core,硬盘是 12*2T。操作系统是 centos6.3。按照官方文档在服务器中安装 hadoop 组件。然后将 hdfs、mapreduce 及 hbase 按照正常顺序开启服务。本实例中源数据的格式是 QVW75520121124120403222, 2222,4, 3。第一列前 6 位代表车牌号,第一列中后 8 位代表日期,第一列最后 9 位代表时分秒及毫秒,第二列代表卡口号。条件查询是指给定车牌号码及起始日期时间和结束日期时间,查找在这期间中固定车牌号经过的卡口信息。源数据有 100 亿条数据,提高查询效率是必要的。经过分析,在源数据导入 HBase 时可以将 RowKey 设计成符合此次查询的效果,可以将车辆号码及日期时间作为 RowKey,这样可以很快速的提取出想要查询的记录。基于 HBase 表的条件查询优化方法的执行流程图如附图 1 所示。首先根据 Split 函数得到 region 的个数 M,用 region 预分配程序调用 Split 函数生

成有 M 个 region 的 HBase 表,然后根据设计好的 RowKey 将数据导入到 HBase 表中,最后编写 mapreduce 程序来条件查询达到优化的结果。

[0027] 在 HBase 中最核心的模块,主要负责响应用户 I/O 请求,向 HDFS 文件系统中读写数据。Hregion Server 的结构图如附图 2 所示,HRegionServer 内部管理了一系列 HRegion 对象,每个 HRegion 对应了 Table 中的一个 Region, HRegion 中由多个 HStore 组成。每个 HStore 对应了 Table 中的一个 Column Family 的存储,每个 Column Family 其实就是一个集中的存储单元,因此最好将具备共同 IO 特性的 column 放在一个 Column Family 中,这样最高效。因此,在数据导入到 HBase 时,我们首先根据 region 预分配将数据均匀分布到 M 个 region 中,然后根据应用查询条件合理的设计 RowKey,将全部的数据按照预先制定的 RowKey 导入到 HBase 表中。

[0028] 在优化查询中,首先是客户端与 regionserver 通信一次,会找到 regionserver 的 region,并扫描 region 返回一定数据。这个数据量是由 scan 的 Batch 指定的。而 caching 的作用就是通信一次找到 region,调用扫描 caching 次,也就是说用这两个参数的话,一次通信可以返回的数据为 caching*batch 条。显然这会减少客户端和 rs 的通信量。

[0029] 在编写 MapReduce 中,为了减少带宽的限制,不用 reduce 函数。无 reduce 阶段的计算框架执行流程图如附图 3 所示。MapReduce 算法运行过程中有一个主控程序,称为 master。主控程序会产生很多作业程序,称为 worker。并且把 M 个 map 任务这些 worker,让它们去完成。被分配了 map 任务的 worker 读取并处理相关的输入数据,将分析出的键/值 (key/value)。由于没有 reduce 函数, map() 函数产生的中间结果键/值 (key/value) 直接输出到输出文件中。

[0030] Map 函数的编写首先对起始日期和结束日期进行解析,用 SimpleDateFormat 类中的 parse 函数将输入的参数解析成日期的格式,再将解析的 Date 设置成想要的格式,用 SimpleDateFormat 类中的 format 函数,这时可以将起始日期及车配号码组合成 RowKey 的格式。在本次条件查询中,RowKey 被设计成车牌号码 + 日期时间的组合,这样可以根据车牌号码及起始日期和时间将条件查询中的起始位置确定出来,在根据车牌号码及结束日期和时间将条件查询中的结束位置确定出来,将这两个值分别赋予 scan 中的 startkey 和 endkey。由于在车辆号码固定的情况下,查找它经过的卡口信息可能都在一个 region 中,返回数据时先放到客户端进行缓存的,通过 caching 配置项可以设置 HBase scanner 一次从服务端抓取的数据条数。通过将其设置成一个合理的值,可以减少 scan 过程中 next() 的时间开销,代价是 scanner 需要通过客户端的内存来维持这些被 cache 的行记录。在本试验中,有 7 台服务器,每台服务器都有 96G 内存,因此,可以将 cache 设置为较大的值来提高性能。然后我们根据 initTableMapperJob(sourceTable, scan, Mapper.class, Text.class, Text.class, job) 函数开始执行任务,其中 sourceTable 指导入到 HBase 中的表,即要查询的表格,scan 中通过设置 startkey、endkey 和相应的 caching 和 batch 大小去查询表格的符合记录。Mapper.class 中主要有 map 函数的执行过程,该过程将符合条件的记录进行处理,在输出的记录中格式为:QVW755 2012-11-24 10:23 22222。根据 value.getRow() 函数得到 RowKey,再根据 value.value() 得到卡口信息,对上述信息进行处理得到想要的输出格式,可以对 RowKey 进行 substring,得到日期信息,时间信息,然后根据输出格式将其转化为预期的效果,由于没有 reduce 阶段,因此,可以直接将 map 阶段的结果

输出到指点的文件即可,可以根据 `setOutputPath` 来设置输出的目录。最后对整个集群中的配置文件进行优化调整,将 `mapreduce` 任务相关的参数,如 `map` 任务数等。`hbase` 相关的参数,对于读请求先到 `Memstore` 中查数据,查不到就到 `BlockCache` 中查,再查不到就会到磁盘上读,并把读的结果放入 `BlockCache`。一个 `Regionserver` 上有一个 `BlockCache` 和 `N` 个 `Memstore`,它们的大小之和不能大于等于 `heapsize * 0.8`,否则 `HBase` 不能启动。默认 `BlockCache` 为 0.2,而 `Memstore` 为 0.4。对于注重读响应时间的系统,可以将 `BlockCache` 设大些,比如设置 `BlockCache=0.4`,`Memstore=0.39`,以加大缓存的命中率。

[0031] 以上所述仅为本发明的实施例而已,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

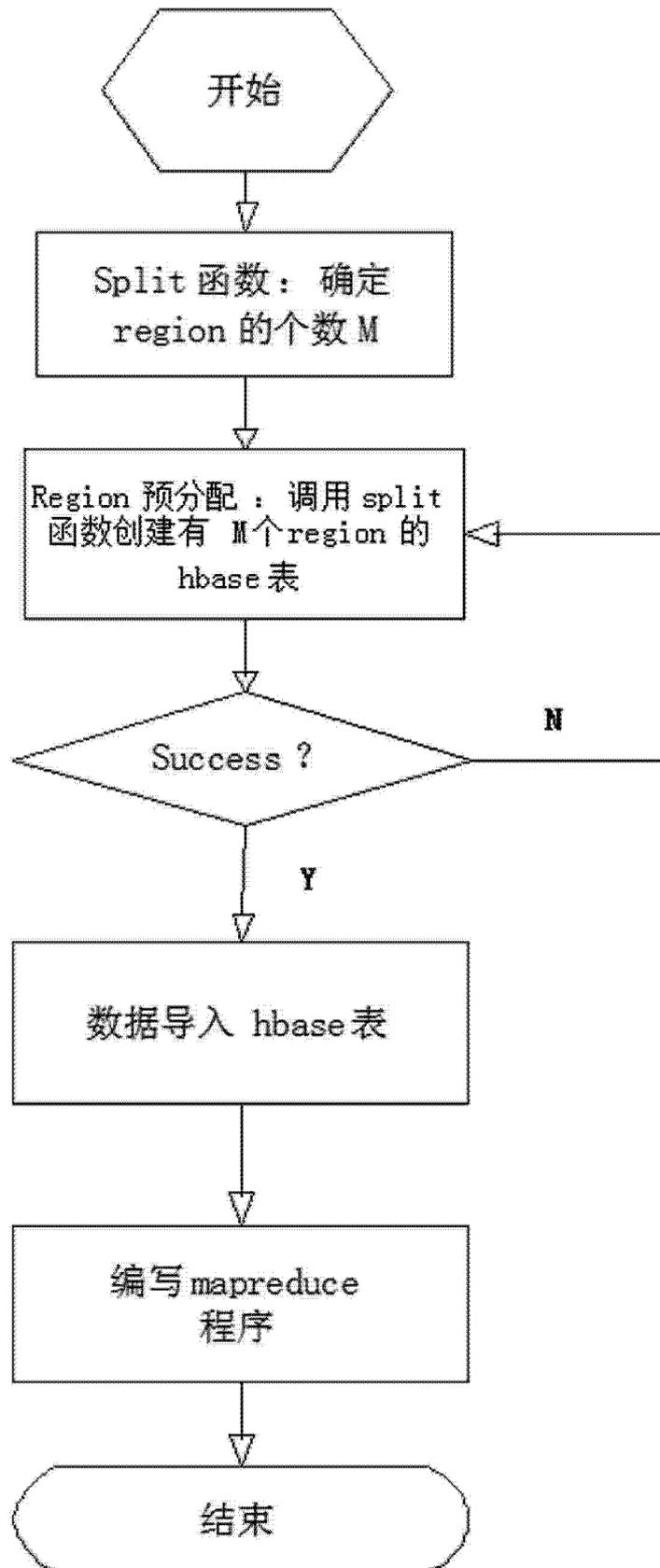


图 1

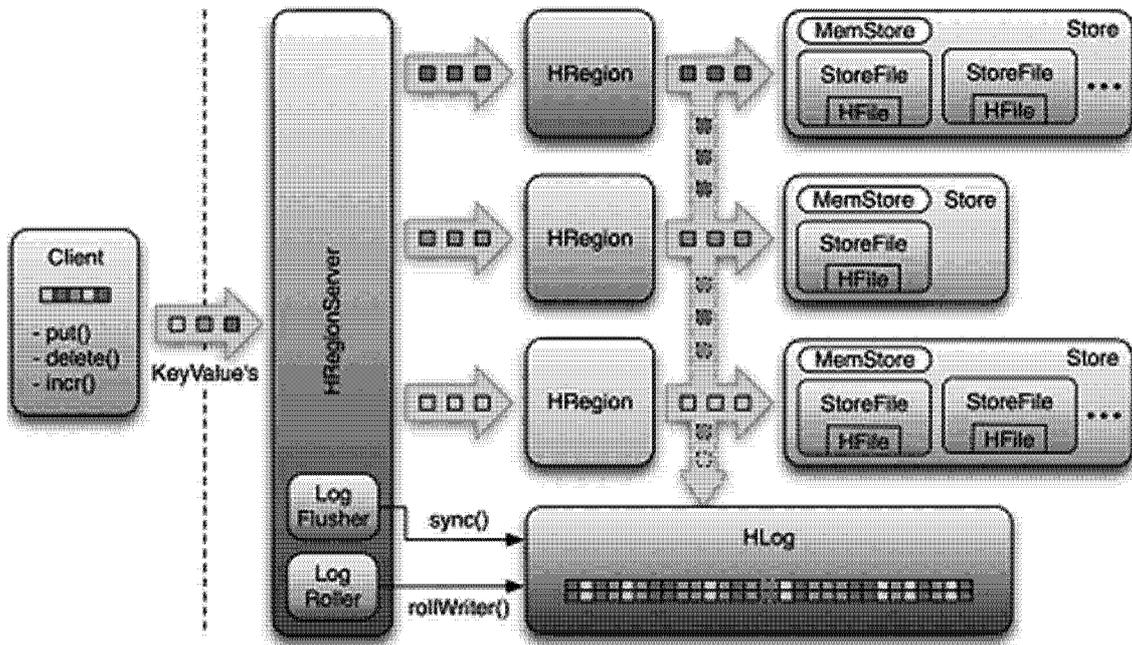


图 2

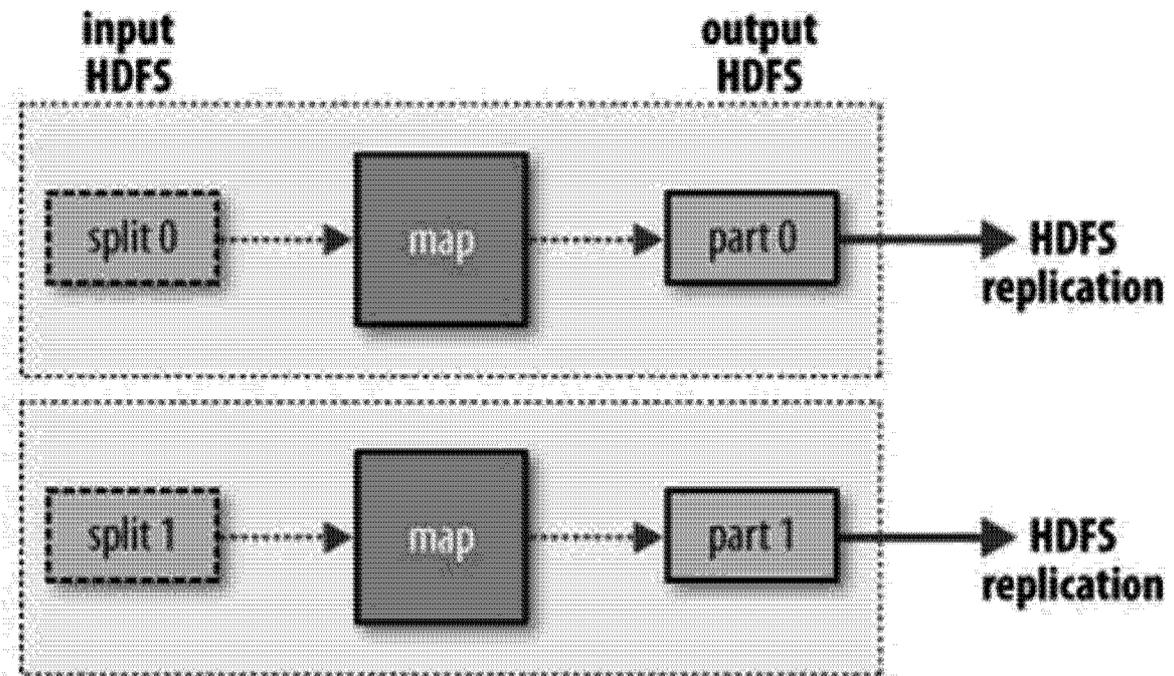


图 3