(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106874348 A
(43)申请公布日 2017.06.20

(21)申请号 201611221215.1

(22)申请日 2016.12.26

(71)申请人 贵州白山云科技有限公司
地址 100015 北京市朝阳区酒仙桥北路甲10号电子城IT产业园201号楼E座5层

(72)发明人 陈闯 张炎泼

(74)专利代理机构 北京瑞思知识产权代理事务所(普通合伙) 11341
代理人 李涛

(51)Int.Cl.
G06F 17/30(2006.01)

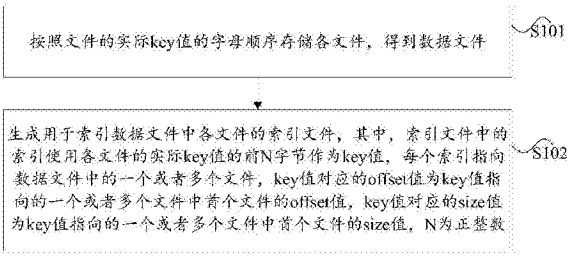
权利要求书2页 说明书8页 附图4页

(54)发明名称

文件存储和索引方法、装置及读取文件的方法

(57)摘要

本发明提供了一种文件存储和索引方法、装置及读取文件的方法，其中，该文件存储和索引方法包括：按照文件的实际key值的字母顺序存储各文件，得到数据文件；生成用于索引数据文件中各文件的索引文件，其中，索引文件中的索引使用各文件的实际key值的前N字节作为key值，每个索引指向数据文件中的一个或者多个文件，key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值，key值对应的size值为key值指向的一个或者多个文件中首个文件的size值。通过本发明，解决了Haystack系统采用的索引方案对内存资源消耗大的问题，降低了索引系统对内存资源的消耗。



CN 106874348 A

1.一种文件存储和索引方法,其特征在于包括:

按照文件的实际key值的字母顺序存储各文件,得到数据文件;

生成用于索引所述数据文件中各文件的索引文件,其中,所述索引文件中的索引使用各文件的实际key值的前N字节作为key值,每个索引指向所述数据文件中的一个或者多个文件,所述key值对应的offset值为所述key值指向的一个或者多个文件中首个文件的offset值,所述key值对应的size值为所述key值指向的一个或者多个文件中首个文件的size值,N为正整数。

2.根据权利要求1所述的方法,其特征在于,所述索引文件中的offset字段和size字段是通过512字节对齐的。

3.根据权利要求1所述的方法,其特征在于,生成用于索引所述数据文件中各文件的索引文件还包括:

按照key值前缀分层存储所述索引文件的索引,其中,所述key值前缀对应的分层中存储的索引的key值为截去所述key值前缀的简短key值,其中,所述key值前缀的字节长度小于N。

4.根据权利要求3所述的方法,其特征在于,

所述索引文件的索引的offset值是以所述索引所在分层为偏移范围的层内offset值,所述层内offset值的字节数是根据分层的最大层地址空间确定的。

5.根据权利要求1至4中任一项所述的方法,其特征在于,所述方法还包括:

将所述数据文件中的所有文件映射到bloomfilter中,以使读取所述数据文件中的文件时通过快速搜索所述bloomfilter来判断将要读取的文件是否可能存在。

6.一种文件存储和索引装置,其特征在于包括:

数据文件存储模块,用于存储数据文件,其中,所述数据文件是按照文件的实际key值的字母顺序存储各文件所得到的;

索引文件生成模块,用于生成用于索引所述数据文件中各文件的索引文件,其中,所述索引文件中的索引使用各文件的实际key值的前N字节作为key值,每个索引指向所述数据文件中的一个或者多个文件,所述key值对应的offset值为所述key值指向的一个或者多个文件中首个文件的offset值,所述key值对应的size值为所述key值指向的一个或者多个文件中首个文件的size值,N为正整数。

7.根据权利要求6所述的装置,其特征在于,所述索引文件生成模块,还用于按照key值前缀分层存储所述索引文件的索引,其中,所述key值前缀对应的分层中存储的索引的key值为截去所述key值前缀的简短key值,其中,所述key值前缀的字节长度小于N。

8.根据权利要求7所述的装置,其特征在于,

所述索引文件的索引的offset值是以所述索引所在分层为偏移范围的层内offset值,所述层内offset值的字节数是根据分层的最大层地址空间确定的。

9.根据权利要求6至8中任一项所述的装置,其特征在于,所述装置还包括:

映射模块,用于将所述数据文件中的所有文件映射到bloomfilter中,以使读取所述数据文件中的文件时通过搜索所述bloomfilter来判断将要读取的文件是否可能存在。

10.一种在权利要求6至9中任一项所述的文件存储和索引装置中读取文件的方法,其特征在于包括:

根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实际key值的前N字节对应的索引；

根据所述实际key值，在所述实际key值的前N字节对应的索引指向的一个或者多个文件中匹配文件；

在匹配到key值与所述实际key值一致的文件时，读取该文件。

11.根据权利要求10所述的方法，其特征在于，根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实际key值的前N字节对应的索引包括：

根据所述bloom filter判断将要读取的文件是否可能存在；

在判断结果为可能存在的情况下，根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实际key值的前N字节对应的索引，否则终止读取文件。

文件存储和索引方法、装置及读取文件的方法

技术领域

[0001] 本发明涉及文件存储及索引领域，具体而言，涉及一种文件存储和索引方法、装置及读取文件的方法。

背景技术

[0002] 当今互联网，数据呈现爆炸式增长，社交网络、移动通信、网络视频、电子商务等各种应用往往能产生亿级甚至十亿、百亿级的海量小文件。由于在元数据管理、访问性能、存储效率等方面面临巨大的挑战，海量小文件问题成为了业界公认的难题。

[0003] 业界的一些知名互联网公司，也对海量小文件提出了解决方案，例如：著名的社交网站Facebook，存储了超过600亿张图片，专门推出了Haystack系统，针对海量小图片进行定制优化的存储。其他的小文件处理方案还有淘宝的TFS等，这些系统的核心思想都是将小文件追加到一个数据文件中，同时生成索引文件，通过索引文件来定位小文件的位置。

[0004] 下面介绍Facebook采用的Haystack的解决方案：

[0005] Facebook的Haystack对小文件的解决办法是，把小文件合起来。将一些小文件的数据依次追加到数据文件中，并且生成索引文件，通过索引来查找小文件在数据文件中的offset和size，对文件进行读取。

[0006] (1)Haystack的数据文件部分：Haystack的数据文件，将每个小文件封装成一个needle，包含文件的key、size、data等数据信息。所有小文件按写入的先后顺序追加到数据文件中。

[0007] (2)Haystack的索引文件部分：Haystack的索引文件保存每个needle的key，以及该needle在数据文件中的offset、size等信息。程序启动时会将索引加载到内存中，在内存中通过查找索引，来定位在数据文件中的偏移量和大小。

[0008] (3)读请求使用索引：将索引文件载入内存，通过查找索引，来定位要读取文件的offset、size，将数据读取出来。

[0009] (4)写请求使用索引：写文件每次添加一个文件，将文件的数据添加到末尾的Needle n。生成索引添加到Needle n index record。

[0010] 由上述的描述可以看出，Facebook的Haystack特点是将文件的完整key都加载到内存中，进行文件定位。机器内存足够大的情况下，Facebook完整的8字节key可以全部加载到内存中。但是现实环境下存在两个问题：

[0011] (1)存储服务器内存不会太大，一般为32G至64G；

[0012] (2)小文件对应的key大小难控制，一般选择文件内容的MD5或SHA1作为该文件的key。

[0013] 假设一台存储服务器有12块4T磁盘，内存为32GB左右。服务器上现需存储大小约为4K的头像、缩略图等文件，约为10亿个。文件的key使用MD5，加上offset和size字段，平均一个小文件对应的索引信息占用28字节。在这种情况下，索引占用内存接近30GB，磁盘仅占用4TB。内存消耗近100％，磁盘消耗只有8％。

[0014] 由此可见，Haystack系统采用的索引方案对内存资源消耗巨大，并且内存资源限制了磁盘资源的利用率，因此，想要获得更大的磁盘资源的利用率需要额外增加内存资源的大量投入。

发明内容

[0015] 本发明提供了一种文件存储和索引方法、装置及读取文件的方法，以至少解决Haystack系统采用的索引方案对内存资源消耗大的问题。

[0016] 根据本发明的一个方面，提供了一种文件存储和索引方法，包括：按照文件的实际key值的字母顺序存储各文件，得到数据文件；生成用于索引所述数据文件中各文件的索引文件，其中，所述索引文件中的索引使用各文件的实际key值的前N字节作为key值，每个索引指向所述数据文件中的一个或者多个文件，所述key值对应的offset值为所述key值指向的一个或者多个文件中首个文件的offset值，所述key值对应的size值为所述key值指向的一个或者多个文件中首个文件的size值，N为正整数。

[0017] 可选地，所述索引文件中的offset字段和size字段是通过512字节对齐的。

[0018] 可选地，生成用于索引所述数据文件中各文件的索引文件还包括：按照key值前缀分层存储所述索引文件的索引，其中，所述key值前缀对应的分层中存储的索引的key值为截去所述key值前缀的简短key值，其中，所述key值前缀的字节长度小于N。

[0019] 可选地，所述索引文件的索引的offset值是以所述索引所在分层为偏移范围的层内offset值，所述层内offset值的字节数是根据分层的最大层地址空间确定的。

[0020] 可选地，所述方法还包括：将所述数据文件中的所有文件映射到bloomfilter中，以使读取所述数据文件中的文件时通过快速搜索所述bloomfilter来判断将要读取的文件是否可能存在。

[0021] 根据本发明的另一个方面，还提供了一种文件存储和索引装置，包括：数据文件存储模块，用于存储数据文件，其中，所述数据文件是按照文件的实际key值的字母顺序存储各文件所得到的；索引文件生成模块，用于生成用于索引所述数据文件中各文件的索引文件，其中，所述索引文件中的索引使用各文件的实际key值的前N字节作为key值，每个索引指向所述数据文件中的一个或者多个文件，所述key值对应的offset值为所述key值指向的一个或者多个文件中首个文件的offset值，所述key值对应的size值为所述key值指向的一个或者多个文件中首个文件的size值，N为正整数。

[0022] 可选地，所述索引文件生成模块，还用于按照key值前缀分层存储所述索引文件的索引，其中，所述key值前缀对应的分层中存储的索引的key值为截去所述key值前缀的简短key值，其中，所述key值前缀的字节长度小于N。

[0023] 可选地，所述索引文件的索引的offset值是以所述索引所在分层为偏移范围的层内offset值，所述层内offset值的字节数是根据分层的最大层地址空间确定的。

[0024] 可选地，所述装置还包括：映射模块，用于将所述数据文件中的所有文件映射到bloom filter中，以使读取所述数据文件中的文件时通过搜索所述bloom filter来判断将要读取的文件是否可能存在。

[0025] 根据本发明的另一个方面，还提供了一种在上述的文件存储和索引装置中读取文件的方法，包括：根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实

际key值的前N字节对应的索引;根据所述实际key值,在所述实际key值的前N字节对应的索引指向的一个或者多个文件中匹配文件;在匹配到key值与所述实际key值一致的文件时,读取该文件。

[0026] 可选地,根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实际key值的前N字节对应的索引包括:根据所述bloom filter判断将要读取的文件是否可能存在;在判断结果为可能存在的情况下,根据将要读取的文件的实际key值的前N字节查询所述索引文件中所述实际key值的前N字节对应的索引,否则终止读取文件。

[0027] 通过本发明,采用按照文件的实际key值的字母顺序存储各文件,得到数据文件;生成用于索引数据文件中各文件的索引文件,其中,索引文件中的索引使用各文件的实际key值的前N字节作为key值,每个索引指向数据文件中的一个或者多个文件,key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值,key值对应的size值为key值指向的一个或者多个文件中首个文件的size值的方式,解决了Haystack系统采用的索引方案对内存资源消耗大的问题,降低了索引系统对内存资源的消耗。

附图说明

[0028] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0029] 图1是根据本发明实施例的文件存储和索引方法的流程图;

[0030] 图2是根据本发明实施例的文件存储和索引装置的结构框图;

[0031] 图3是根据本发明实施例的在文件存储和索引装置中读取文件的方法的流程图;

[0032] 图4是根据本发明优选实施例的文件存储和索引结构的示意图;

[0033] 图5是根据本发明优选实施例的读取文件的方法的流程图;

[0034] 图6、图7和图8是根据本发明优选实施例的索引分层示意图;

[0035] 图9和图10是根据本发明优选实施例与相关技术的索引方案的内存消耗对比示意图。

具体实施方式

[0036] 下文中将参考附图并结合实施例来详细说明本发明。需要说明的是,在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互组合。

[0037] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。

[0038] 实施例1

[0039] 在本实施例中提供了一种文件存储和索引方法,图1是根据本发明实施例的文件存储和索引方法的流程图。如图1所示,该流程包括如下步骤:

[0040] 步骤S101,按照文件的实际key值的字母顺序存储各文件,得到数据文件;

[0041] 步骤S102,生成用于索引数据文件中各文件的索引文件,其中,索引文件中的索引使用各文件的实际key值的前N字节作为key值,每个索引指向数据文件中的一个或者多个文件,key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值,key值对应的size值为key值指向的一个或者多个文件中首个文件的size值,N为正整数。

[0042] 在上述步骤中，由于在索引中不再保存文件实际key值，而是仅保存实际key值的前N字节，减少了索引文件的大小；同时，这样的索引不再指向一个文件，而会指向实际key值的前N字节相同的一个或者多个文件；为了能够根据索引中的offset值定位到文件的位置，在存储文件时把文件按照实际key值的字母顺序依次存储到数据文件中，使得实际key值的前N字节相同的一个或者多个文件集中存储在一片连续的位置上，得以使用一个offset值来指示它们的存储位置。可见，在将步骤S102中生成的索引文件加载到内存中之后，相对于相关技术的Haystack系统而言，将会占用更少的内存资源，解决了Haystack系统采用的索引方案对内存资源消耗大的问题，降低了索引系统对内存资源的消耗。

[0043] 在采用步骤S102生成的索引文件索引某一个文件时，根据索引不再能直接索引到某一个确定的文件，而将会索引到一个连续的文件集合；在需要精确读取某一个文件时，只要根据这个文件的实际key值，在文件集合中逐一匹配文件就可能读取到想要的文件。

[0044] 可选地，上述索引文件中的offset字段和size字段是通过512字节对齐的；即如果一个文件是1024字节大小，按照512字节对齐，1024/512＝2，则文件大小可以用2表示，当在索引中得到size是2，用2乘以512字节就可以得到文件的大小是1024字节；之前需要保存的是1024，现在只需要保存2这个数字，至少节省一个字节。并且还可以根据整个数据文件的实际大小计算offset字段和size字段所需使用的字节数，从而可以进一步减小索引所占用的字节数。

[0045] 为了能够进一步减小索引所占用的字节数，考虑到索引文件中存储的key值仍有key值前缀重复的可能行，因此，还可以考虑对索引文件中的索引按照key值前缀进行分层存储，其中，key值前缀对应的分层中存储的索引的key值为截去key值前缀的简短key值，key值前缀的字节长度小于N。在该分层中索引数量越多的情况下，分层后的索引文件占用的字节数相对于原来的索引文件将会更小。

[0046] 在索引文件采用分层存储之后，各分层内的索引的offset值可以进一步优化以减少字节数。可选地，索引文件的索引的offset值是以索引所在分层为偏移范围的层内offset值，该层内offset值的字节数是根据分层的最大层地址空间确定的。由于最大层地址空间必然小于整个数据文件的大小，因此，层内offset值占用的字节数也将小于按照整个数据文件为偏移范围的原始offset值占用的字节数。

[0047] bloom filter是一种二进制向量数据结构，它具有很好的空间和时间效率，被用来检测一个元素是不是集合中的一个成员。如果检测结果为是，该元素不一定在集合中；但如果检测结果为否，该元素一定不在集合中。Bloom filter优点是它的插入和查询时间都是常数，另外它查询元素却不保存元素本身，具有良好的安全性。在本发明中，由于一个索引指向多个文件，因此有必要利用bloom filter，以通过快速搜索文件是否可能存在来避免对不存在文件的查询所造成的资源和时间浪费。可选地，本实施例中还将数据文件中的所有文件映射到bloom filter中，以使读取数据文件中的文件时通过快速搜索bloom filter来判断将要读取的文件是否可能存在。

[0048] 本发明实施例中N的取值优选为4。

[0049] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有

技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端设备(可以是手机，计算机，服务器，或者网络设备等)执行本发明各个实施例所述的方法。

[0050] 实施例2

[0051] 在本实施例中还提供了一种文件存储和索引装置，该装置用于实现上述实施例及优选实施方式，已经进行过说明的不再赘述。如以下所使用的，术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现，但是硬件，或者软件和硬件的组合的实现也是可能并被构想的。

[0052] 图2是根据本发明实施例的文件存储和索引装置的结构框图，如图2所示，该装置包括：数据文件存储模块21和索引文件生成模块22，其中，

[0053] 数据文件存储模块21，用于存储数据文件，其中，数据文件是按照文件的实际key值的字母顺序存储各文件所得到的；索引文件生成模块22，耦合至数据文件存储模块21，用于生成用于索引数据文件中各文件的索引文件，其中，索引文件中的索引使用各文件的实际key值的前N字节作为key值，每个索引指向数据文件中的一个或者多个文件，key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值，key值对应的size值为key值指向的一个或者多个文件中首个文件的size值，N为正整数。

[0054] 可选地，索引文件生成模块还用于按照key值前缀分层存储索引文件的索引，其中，key值前缀对应的分层中存储的索引的key值为截去key值前缀的简短key值，其中，key值前缀的字节长度小于N。

[0055] 可选地，索引文件的索引的offset值是以索引所在分层为偏移范围的层内offset值，层内offset值的字节数是根据分层的最大层地址空间确定的。

[0056] 可选地，上述文件存储和索引装置还包括：映射模块，用于将数据文件中的所有文件映射到bloom filter中，以使读取数据文件中的文件时通过搜索bloom filter来判断将要读取的文件是否可能存在。

[0057] 需要说明的是，上述各个模块是可以通过软件或硬件来实现的，对于后者，可以通过以下方式实现，但不限于此：上述模块均位于同一处理器中；或者，上述模块分别位于多个处理器中。

[0058] 本发明实施例中N的取值优选为4。

[0059] 实施例3

[0060] 在本实施例中提供了一种在上述的文件存储和索引装置中读取文件的方法，图3是根据本发明实施例的在文件存储和索引装置中读取文件的方法的流程图，如图3所示，该流程包括如下步骤：

[0061] 步骤S301，根据将要读取的文件的实际key值的前N字节查询索引文件中实际key值的前N字节对应的索引；

[0062] 步骤S302，根据实际key值，在实际key值的前N字节对应的索引指向的一个或者多个文件中匹配文件；

[0063] 步骤S303，在匹配到key值与实际key值一致的文件时，读取该文件。

[0064] 可选地，在步骤S301中，在查询索引之前，还可以根据bloom filter判断将要读取的文件是否可能存在；在判断结果为可能存在的情况下，根据将要读取的文件的实际key值

的前N字节查询索引文件中实际key值的前N字节对应的索引，否则终止读取文件。

[0065] 本发明实施例中N的取值优选为4。

[0066] 实施例4

[0067] 为了使本发明实施例的描述更加清楚，下面结合优选实施例进行描述和说明。

[0068] 在本优选实施例中提供了一种文件存储和索引结构和方法，图4是根据本发明优选实施例的文件存储和索引结构的示意图，如图4所示，其中，layer是分层文件，将相同的key前缀分为一层。Index是索引文件，对小文件进行定位。data是数据文件，其中的每个needle都是一个小文件。

[0069] 图5是根据本发明优选实施例的读取文件的方法的流程图，在图5中示出了通过匹配索引前缀，定位小文件的具体位置，然后通过读取完整的key，来查看文件的key是否匹配，如果不匹配再继续顺序查找下一个needle的详细流程。

[0070] 本优选实施例提供的文件存储和索引方案包括下列步骤：

[0071] 步骤1：压缩前缀优化，减少key、offset、size占用空间；

[0072] (1)数据文件组织：

[0073] 与Facebook的Haystack类似，该系统将多个小文件写入到一个数据文件中，每个needle保存key、size、data等信息。

[0074] (2)索引文件组织：

[0075] 1)索引文件只保存key的前四字节，而非完整的key；

[0076] 2)索引文件中的offset和size字段，通过512字节对齐，节省1个字节；并根据整个数据文件实际大小计算offset和size使用的字节数。

[0077] 步骤2：needle顺序存放，定位小文件位置；

[0078] 数据文件中的needle按照key的字母顺序存放。

[0079] 由于索引文件的key，只保存前四字节，如果小文件key的前四字节相同，不顺序存放needle，则无法根据一个offset找到分散存放的全部needle的具体位置。例如：用户读取的文件key是0x ab cd ef ac ee，但由于索引文件中的key只保存前四字节，只能匹配0x ab cd ef ac这个前缀，此时无法定位到具体要读取的offset。

[0080] 在本优选实施例中，通过needle顺序存放，来解决上述问题：例如：用户读取文件的key是0x ab cd ef ac bb，匹配到0x ab cd ef ac这个前缀，此时offset指向0x ab cd ef ac aa这个needle，第一次匹配未命中。

[0081] 通过存放在needle的header(文件头)中的size，我们可以定位0x abcdef ac bb位置，匹配到正确needle，并将数据读取给用户。

[0082] 步骤3：索引分层优化；

[0083] (1)分层方案

[0084] 参考图6，可以将索引中key值前缀相同的索引分为一层。分层原则是每个分层中的needle数尽量控制在64个左右，并且根据分层要存放的needle数量，选择分层级别。分层级别可以根据需要确定，例如下面给出了一种分层级别的示例：

[0085] 0级：不进行分层；

[0086] 1级：选择needle key第一字节进行分层；

[0087] 2级：选择needle key的前两字节进行分层；

[0088] 分层所用的key值前缀的字节数小于索引中key值的字节长度。

[0089] (2)分层减少Key的占用字节数

[0090] 参考图7,通过分层,只保存一份重复的前缀,节省key的字节数。

[0091] (3)分层减少offset的占用字节数

[0092] 参考图8,优化前的offset,偏移范围为整个数据文件的地址空间。优化后,layer的offset在整个数据文件中进行偏移,而分层下的索引的offset只需在数据文件中的层内进行偏移,根据最大的层地址空间可以计算所需字节数。

[0093] 此外,在本优选实施例中,还通过bloomfilter避免不存在文件的访问。

[0094] 在内存中,将存在的文件映射到bloom filter中,只需要通过快速搜索,就可以排除掉不存在文件。时间复杂度为O(k),k为一个元素需要的bit位数。经验表明,当k为9.6时,误报率为1%,如果k再增加4.8,误报率会降低到0.1%。

[0095] 下面将以Haystack为参考说明本发明优选实施例的有益效果。

[0096] (1)通过前缀压缩,带来的内存节省对比

[0097] 参考图9,横轴表示文件数,纵轴表示索引文件需要的内存大小,短虚线表示传统的Haystack的内存消耗量,长虚线表示通过本发明实施例进行前缀压缩后的内存消耗量。从图9可以看出在文件数量为10亿的情况下,使用facabook的Haystack消耗的内存为26G多,使用本优选实施例提供的压缩前缀的索引方案消耗的内存为9G多,内存使用降低了2/3。

[0098] (2)再次通过索引分层,带来的内存节省对比

[0099] 参考图10,横轴表示文件数,纵轴表示索引文件需要的内存大小,短虚线表示传统的Haystack的内存消耗量,长虚线表示通过本发明实施例进行前缀压缩后的内存消耗量,实线表示通过本发明实施例进行前缀压缩并索引分层后的内存消耗量。从图10可以看出,在进行索引分层后,从优化之前的9G多内存消耗,进一步降低到4G多,又节省了1半的内存消耗。

[0100] 在试验本优选实施例提供的文件存储和索引方案后,小文件的整体性能有显著提高,每秒请求数(RequestPerSecond,简称为RPS)提升一倍多,机器的输入输出(Input/Output,简称为IO)使用率减少了将近一倍。同时,因为优化了最小存储单元,碎片降低80%。使用该系统我们可以为用户提供更快速地读写服务,并且节省了集群的资源消耗。

[0101] 实施例5

[0102] 在本实施例中提供了一种软件,该软件用于执行上述实施例及优选实施方式中描述的技术方案。

[0103] 实施例6

[0104] 本发明的实施例还提供了一种存储介质。在本实施例中,上述存储介质可以被设置为存储用于执行以下步骤的程序代码:

[0105] 步骤S101,按照文件的实际key值的字母顺序存储各文件,得到数据文件;

[0106] 步骤S102,生成用于索引数据文件中各文件的索引文件,其中,索引文件中的索引使用各文件的实际key值的前N字节作为key值,每个索引指向数据文件中的一个或者多个文件,key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值,key值对应的size值为key值指向的一个或者多个文件中首个文件的size值,N为正整数。

[0107] 可选地，在本实施例中，上述存储介质可以包括但不限于：U盘、只读存储器(Read-Only Memory，简称为ROM)、随机存取存储器(Random Access Memory，简称为RAM)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0108] 可选地，本实施例中的具体示例可以参考上述实施例及可选实施方式中所描述的示例，本实施例在此不再赘述。

[0109] 实施例7

[0110] 本发明的实施例还提供了一种存储介质。在本实施例中，上述存储介质可以被设置为存储用于执行以下步骤的程序代码：

[0111] 步骤S301，根据将要读取的文件的实际key值的前N字节查询索引文件中实际key值的前N字节对应的索引；

[0112] 步骤S302，根据实际key值，在实际key值的前N字节对应的索引指向的一个或者多个文件中匹配文件；

[0113] 步骤S303，在匹配到key值与实际key值一致的文件时，读取该文件。

[0114] 可选地，在本实施例中，上述存储介质可以包括但不限于：U盘、只读存储器(Read-Only Memory，简称为ROM)、随机存取存储器(Random Access Memory，简称为RAM)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0115] 可选地，本实施例中的具体示例可以参考上述实施例及可选实施方式中所描述的示例，本实施例在此不再赘述。

[0116] 显然，本领域的技术人员应该明白，上述的本发明的各模块或各步骤可以用通用的计算装置来实现，它们可以集中在单个的计算装置上，或者分布在多个计算装置所组成的网络上，可选地，它们可以用计算装置可执行的程序代码来实现，从而，可以将它们存储在存储装置中由计算装置来执行，并且在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤，或者将它们分别制作成各个集成电路模块，或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样，本发明不限制于任何特定的硬件和软件结合。

[0117] 以上所述仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。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。

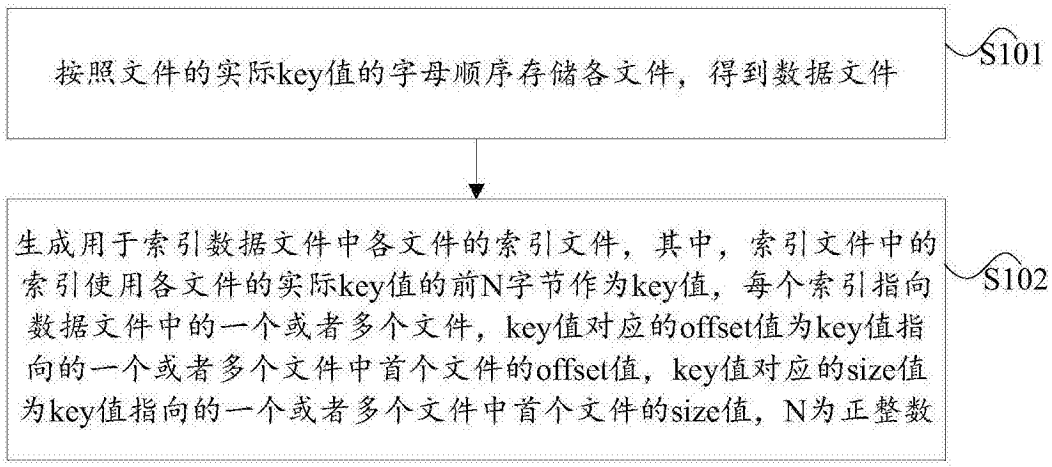

按照文件的实际key值的字母顺序存储各文件，得到数据文件
S101
生成用于索引数据文件中各文件的索引文件，其中，索引文件中的索引使用各文件的实际key值的前N字节作为key值，每个索引指向数据文件中的一个或者多个文件，key值对应的offset值为key值指向的一个或者多个文件中首个文件的offset值，key值对应的size值为key值指向的一个或者多个文件中首个文件的size值，N为正整数
S102

图1

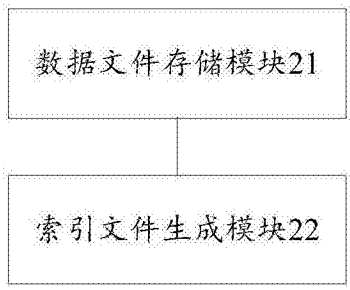

数据文件存储模块21
索引文件生成模块22

图2

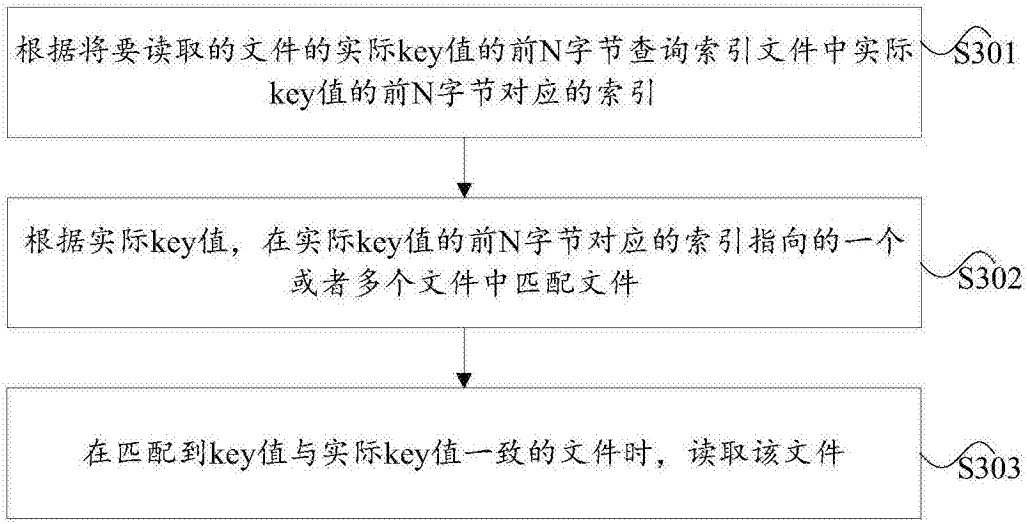

根据将要读取的文件的实际key值的前N字节查询索引文件中实际key值的前N字节对应的索引
S301
根据实际key值，在实际key值的前N字节对应的索引指向的一个或者多个文件中匹配文件
S302
在匹配到key值与实际key值一致的文件时，读取该文件
S303

图3

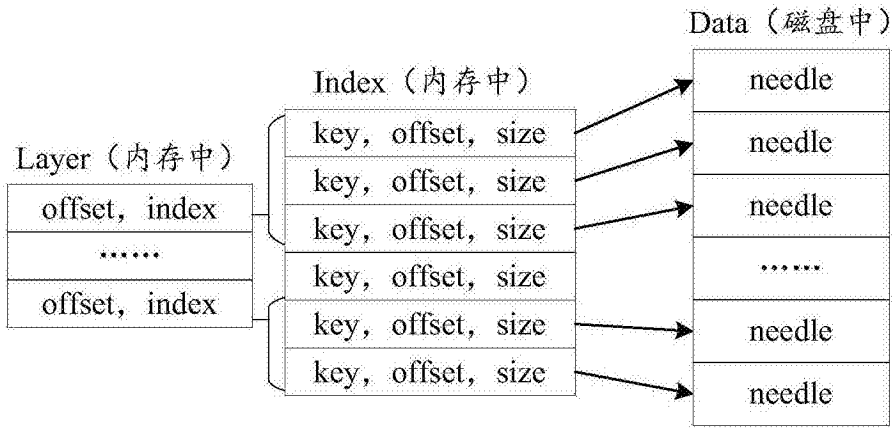

Data（磁盘中）
Index（内存中）
Layer（内存中）
offset，index
……
offset，index
key，offset，size
key，offset，size
key，offset，size
key，offset，size
key，offset，size
key，offset，size
needle
needle
needle
……
needle
needle

图4

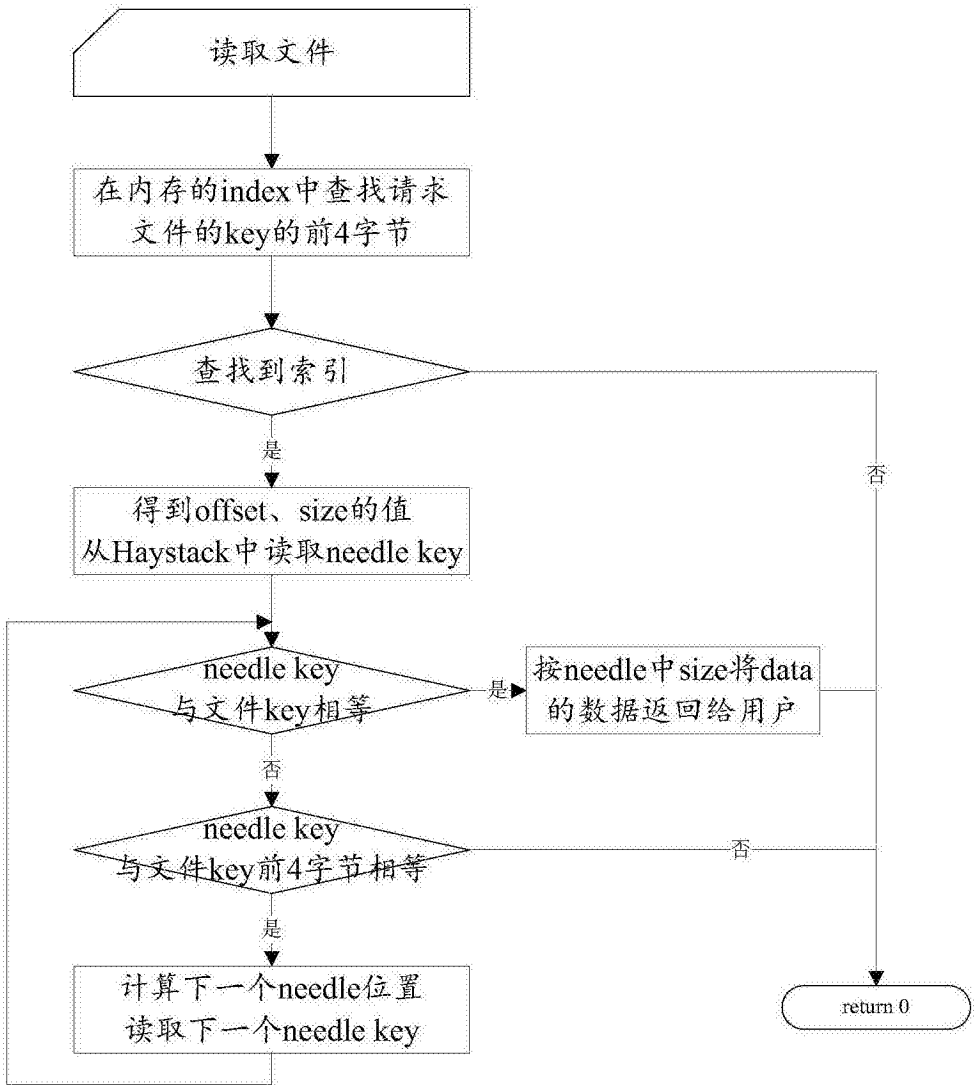

读取文件
在内存的index中查找请求文件的key的前4字节
查找到索引
是
否
得到offset、size的值
从Haystack中读取needle key
needle key
与文件key相等
是
按needle中size将data的数据返回给用户
否
needle key
与文件key前4字节相等
否
是
计算下一个needle位置
读取下一个needle key
return 0

图5

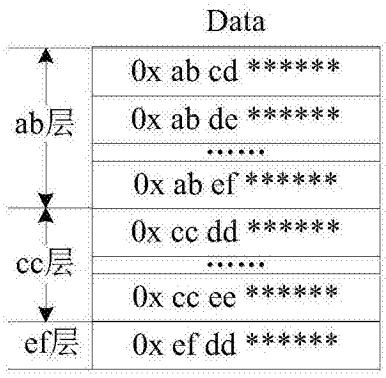

Data
ab层
cc层
ef层
0x ab cd ******
0x ab de ******
……
0x ab ef ******
0x cc dd ******
……
0x cc ee ******
0x ef dd ******

图6

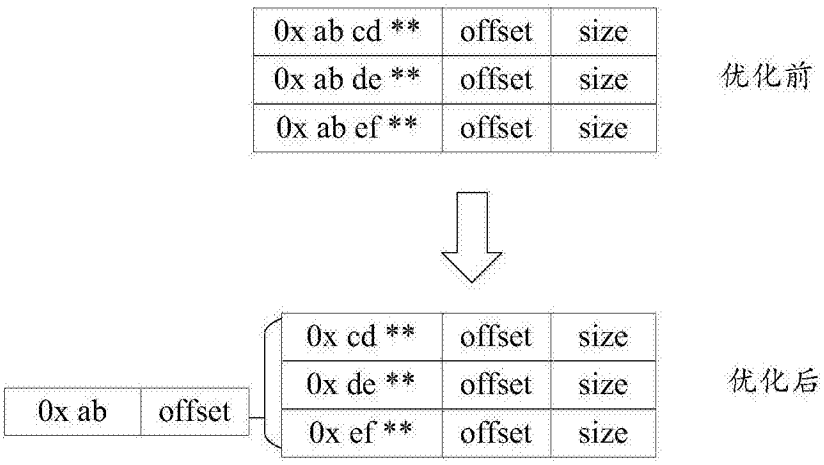

0x ab cd ** offset size
0x ab de ** offset size
0x ab ef ** offset size
优化前
0x ab offset
0x cd ** offset size
0x de ** offset size
0x ef ** offset size
优化后

图7

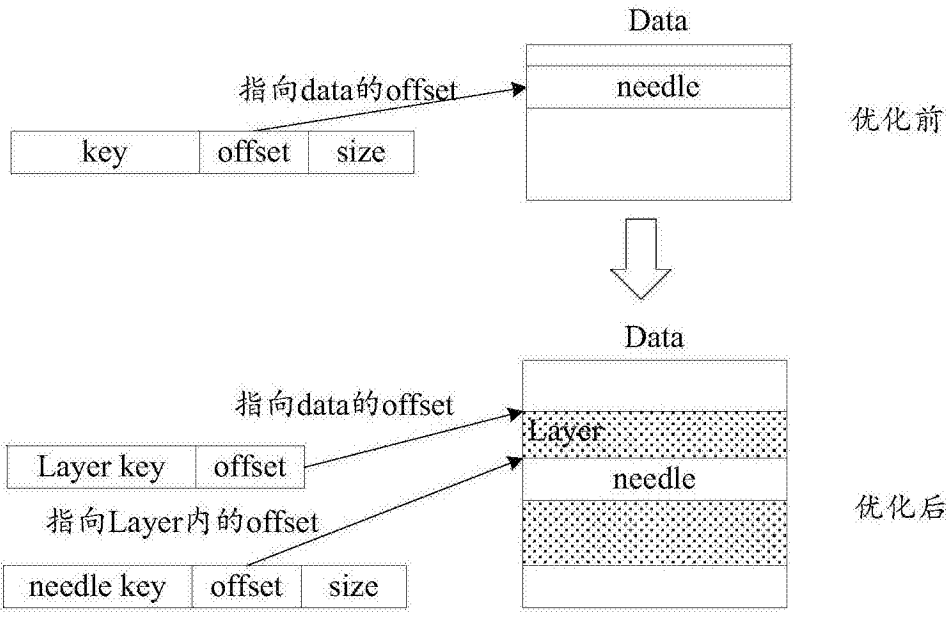

Data
指向data的offset
key offset size
needle
优化前
Data
指向data的offset
Layer key offset
指向Layer内的offset
needle key offset size
Layer
needle
优化后

图8

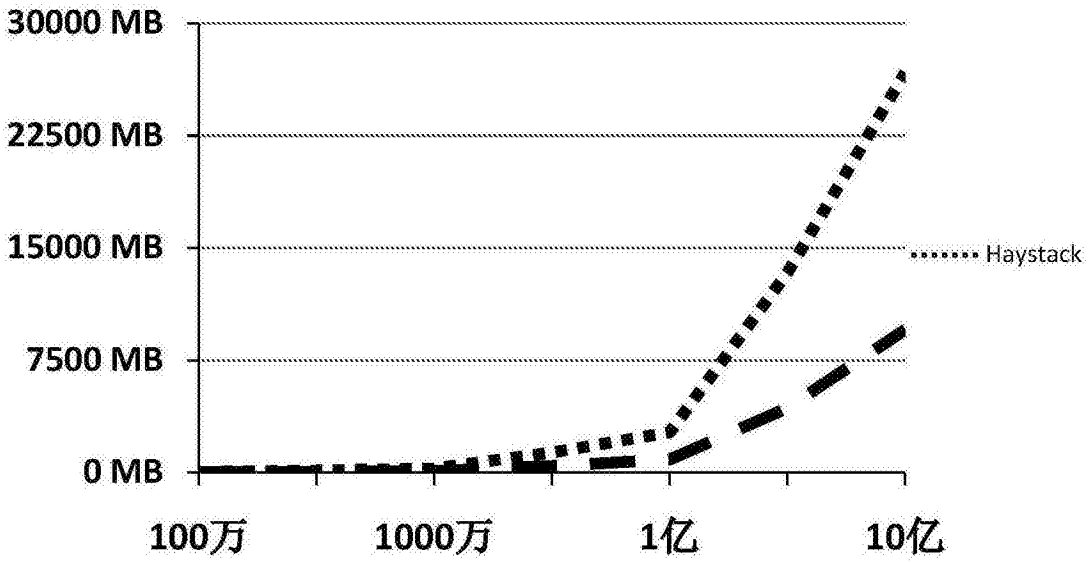

30000 MB
22500 MB
15000 MB
7500 MB
0 MB
100万
1000万
1亿
10亿
Haystack

图9

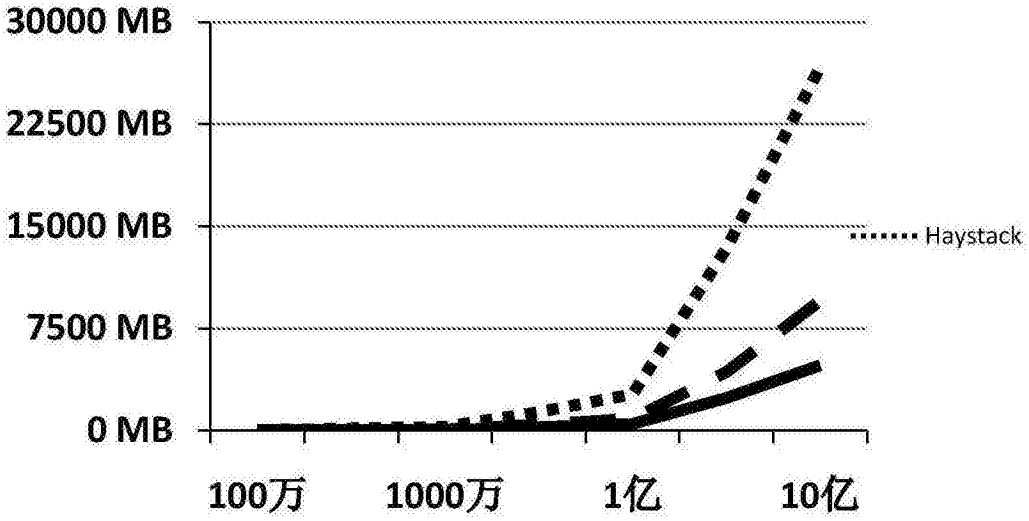

30000 MB
22500 MB
15000 MB
7500 MB
0 MB
100万
1000万
1亿
10亿
Haystack

图10