(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 110647497 A
(43)申请公布日 2020.01.03

(21)申请号 201910722847.3

(22)申请日 2019.08.06

(66)本国优先权数据
201910657019.6 2019.07.19 CN

(71)申请人 广东工业大学
地址 510006 广东省广州市番禺区大学城外环西路100号

(72)发明人 吴宗泽 张兴斌 李建中 梁泽逍 李俊彬 黄婷婷

(74)专利代理机构 广州粤高专利商标代理有限公司 44102
代理人 张金福

(51)Int.Cl.
G06F 16/11(2019.01)
G06F 16/182(2019.01)

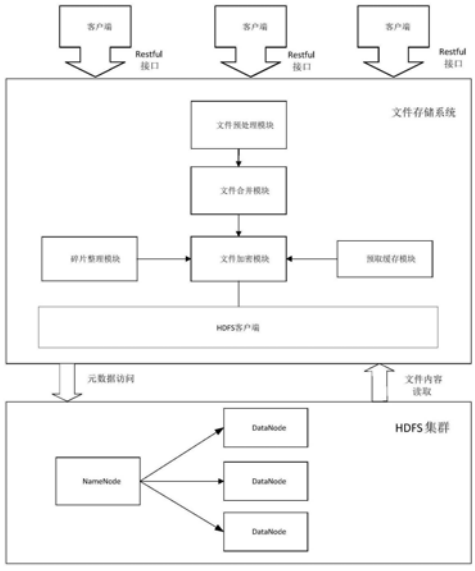
权利要求书2页 说明书9页 附图5页

(54)发明名称

一种基于HDFS的高性能文件存储与管理系统

(57)摘要

本发明提供一种基于HDFS的高性能文件存储与管理系统,为解决HDFS对小文件存储支持的不足提出了文件预处理模块和文件合并模块,为提高文件读取效率提出了预取缓存模块,为提高空间整体利用率提出了碎片整理模块,为保护用户隐私文件提出了数据加密模块。本发明提出一种通用的小文件合并策略和HDFS客户端优化方案,结合了HAR文件归档方法和MapFile方法的优点,以适用于任何类型的文件。具体地,该系统优化了小文件合并策略并提出一种懒更新的碎片整理机制,以提高HDFS集群的空间利用率。同时,基于频率统计的热点文件策略提高了数据的读取速度,基于MD5登录加密和AES文件加密提高了用户账号和隐私文件的安全性。



CN 110647497 A

1.一种基于HDFS的高性能文件存储与管理系统,接收客户端上传的文件,并与HDFS集群进行数据交换,包括HDFS客户端,其特征在于,还包括文件预处理模块、文件合并模块和预取缓存模块,其中:

所述文件预处理模块接收客户端上传的文件并对上传的文件进行大小计算和类型判定,并创建该文件的元数据信息,根据上传的文件的大小分为小文件、中文件和大文件,小文件、中文件传入文件合并队列等待文件合并模块的读取,大文件直接存入HDFS集群;

所述文件合并模块读取文件合并队列中小文件或中文件,根据文件类型判断需要追加到的目标合并文件类型,若属于该合并文件的剩余容量大于等于小文件或中文件的大小,则启动合并操作任务并创建合并文件与小文件或中文件的映射关系,即二级索引,若该合并文件的剩余容量小于小文件或中文件的大小时则将当前合并文件和二级索引作为一个数据块存入HDFS集群中,同时创建一个新的且与该合并文件类型一致的合并文件,将文件合并队列中待合并文件写入新的合并文件中;合并文件存入HDFS集群中;

所述预取缓存模块将用户频繁访问的文件信息映射关系表加载到本地内存中,建立客户端与HDFS集群的DataNode的直接联系。

2.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统,其特征在于,所述元数据信息包括文件名称name、文件类型type、文件大小size和创建日期time。

3.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统,其特征在于,所述类型判定为根据文件扩展名类型对文件进行分类,具体为:

文件扩展名为BMP、GIF、JPG、PNG的文件的类型为IMAGE;

文件扩展名为AVI、MOV、ASF、RM、MPG的文件的类型为VIDEO;

文件扩展名为TXT、DOC、PDF、XLS的文件的类型为TXT、DOC、PDF、XLS;

文件扩展名为RAR、ZIP、ISO、MDF的文件的类型为COMFILE;

文件扩展名为EXE、HTML、MID、TMP的文件的类型为REST。

4.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统,其特征在于,所述根据上传的文件的大小分为小文件、中文件和大文件,具体划分方式为:

大于等于4.3MB的为小文件,大于4.3MB小于等于32MB的为中文件,大于32MB的为大文件。

5.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统,其特征在于,所述合并操作任务具体为:

小文件或中文件以字符数组的形式追加到合并文件末尾,并创建合并文件与小文件的映射关系,即二级索引,其内容包括文件名name,文件类型type,文件大小size、偏移量offset元数据信息。

6.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统,其特征在于,所述预取缓存模块将用户频繁访问的文件信息映射关系加载到本地内存中,具体为:

在小文件的元数据信息中增加一个频率读取标记frequency,用户每发起一次文件的读取指令,相应文件的读取频率数+1,将读取频率最高的前10%的文件作为热点文件,直接将其文件映射关系加载到本地,用户访问热点文件时,可以不经过NameNode,首先在本地内存热点文件目录中根据一级索引找到存放该文件DataNode数据块的具体位置,再根据数据块中二级索引定位到小文件。

7.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统，其特征在于，所述预取缓存模块定期进行热点文件目录的更新。

8.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统，其特征在于，还包括碎片整理模块，所述碎片整理模块采用懒更新机制，当用户索引到某一小文件时，查看该小文件所在的合并文件有效占比是否小于20％，如果是，将用户读取的该小文件合并到系统当前正在运行的同类型合并文件中，并将合并文件中该小文件的数据信息设置为false，只有当合并文件为空时，删除此合并文件中全部的小文件数据信息和NameNode中合并文件的元数据信息，等待HDFS系统进行空间回收。

9.根据权利要求8所述的基于HDFS的高性能文件存储与管理系统，其特征在于，所述碎片整理模块中集成有HDFS集群的状态监控模块，所述状态监控模块对HDFS集群的可用空间进行监控。

10.根据权利要求1所述的基于HDFS的高性能文件存储与管理系统，其特征在于，还包括数据加密模块，所述数据加密模块采用MD5加密算法对用户密码进行数据加密，同时将直接存入HDFS集群的大文件与经所述文件合并模块合并后的合并文件上传至HDFS时进行AES加密操作。

一种基于HDFS的高性能文件存储与管理系统

技术领域

[0001] 本发明涉及计算机科学领域，更具体地，涉及一种基于HDFS的高性能文件存储与管理系统。

背景技术

[0002] 随着移动互联网的普及和应用，全球数据信息呈爆发式增长，人工智能、机器学习、云计算等新兴产业迅速发展，传统计算机存储系统的发展已无法满足大量数据产生和应用的需求，因此分布式文件存储系统应运而生，目前常用的分布式文件系统有Hadoop、Lustre、MogileFS、GoogleFS等，Hadoop具有可靠性高、扩展性强、存储速度快、容错性高等一系列优势，而HDFS正是Hadoop的分布式文件系统。

[0003] 但大数据的计算和存储并不是庞大信息流面临的唯一问题，随着移动应用的出现，产生了大量的小文件，如图片、视频、文档等，这些文件的体积大多大于1KB，小于10MB。面对这些海量的小文件，Hadoop分布式文件系统在设计之初并没有考虑到小文件存储带来的内存浪费等问题。因为Hadoop中一个最小的存储单元叫做Block，默认存储文件阈值为64MB，小于64MB的文件会占据节点空间但不占满整个空间，其次，一般一个文件的元数据大概要占250bytes的内存空间，默认3个副本会占用大约368bytes的内存空间，当HDFS集群中有大量小文件时，大量的元数据会导致NameNode内存消耗过大。

[0004] 面对HDFS的小文件存储问题，许多企业和学者提出了不同的解决方案：

[0005] 1)Hadoop Archive(HAR)文件归档

[0006] HAR将集群中大量的小文件合并成大文件，减少了NameNode的内存占用，生成的归档包括两层索引和数据文件。HAR文件归档存在两个问题：读取文件要遍历两层索引，相比直接读取效率可能会降低，其次，文件一旦创建便不能被修改，如果要在合并文件上进行增加或删除操作，只能再次创建新的文件。

[0007] 2)SequenceFile方法

[0008] SequenceFile序列化文件的核心思想是key存储文件的名字，value存储文件的内容，处理小文件时，将它们合并成大文件再存储起来。SequenceFile压缩了文件，节省了磁盘空间，但SequenceFile在文件合并时没有建立小文件到大文件的映射关系，查询时需要对整个SequenceFile文件进行遍历，检索效率低下。

[0009] 3)MapFile方法

[0010] MapFile是排序后的SequenceFile方法，由data和index组成，Index是文件的数据索引，记录了每条记录的key值和该记录在文件中的偏移量。MapFile将索引文件加载到内存，通过索引映射关系快速找到Record所在文件的位置，但存储index数据会消耗一部分内存。

[0011] 同时还有专门为地理数据存储设计的WebGIS优化方案，为PPT和视频小文件设计的BlueSky优化方案，基于HBase和Avro的小文件存储策略。这些方法有的适用于特定的场景和数据，有的实现方式太过复杂而无法实现广泛的应用。

[0012] 针对HDFS的安全性问题,Hadoop引入了加密空间,但是存在加密算法单一、不支持迭代目录加密、使用权限高和不提供应用级加密的缺点。

发明内容

[0013] 本发明提供一种基于HDFS的高性能文件存储与管理系统,能够存储大量小文件、安全性高的分布式文件。

[0014] 为解决上述技术问题,本发明的技术方案如下:

[0015] 一种基于HDFS的高性能文件存储与管理系统,接收客户端上传的文件,并与HDFS集群进行数据交换,包括HDFS客户端、文件预处理模块、文件合并模块和预取缓存模块,其中:

[0016] 所述文件预处理模块接收客户端上传的文件并对上传的文件进行大小计算和类型判定,并创建该文件的元数据信息,根据上传的文件的大小分为小文件、中文件和大文件,小文件、中文件传入文件合并队列等待文件合并模块的读取,大文件直接存入HDFS集群;对小文件按扩展名进行划分,将相似类型的小文件追加在一个同类型的合并文件中,在各类型文件数量和大小均衡的情况下,可成倍提高文件检索效率,因为此时系统只需检索存储该类型小文件的合并文件即可。

[0017] 所述文件合并模块读取文件合并队列中小文件或中文件,根据文件类型判断需要追加到的目标合并文件类型,若属于该合并文件的剩余容量大于等于小文件或中文件的大小,则启动合并操作任务并创建合并文件与小文件或中文件的映射关系,即二级索引,若该合并文件的剩余容量小于小文件或中文件的大小时则将当前合并文件和二级索引作为一个数据块存入HDFS集群中,同时创建一个新的且与该合并文件类型一致的合并文件,将文件合并队列中待合并文件写入新的合并文件中;合并文件存入HDFS集群中;用户查找一个小文件时,需要遍历该文件类型一级索引下所有的合并文件和与合并文件有映射关系的二级索引下的所有小文件。

[0018] 所述预取缓存模块将用户频繁访问的文件信息映射关系表加载到本地内存中,建立客户端与HDFS集群的DataNode的直接联系,减少系统遍历文件所消耗的时间,节约访问成本。

[0019] 优选地,所述元数据信息包括文件名称name、文件类型type、文件大小size和创建日期time。

[0020] 优选地,所述类型判定为根据文件扩展名类型对文件进行分类,具体为:

[0021] 文件扩展名为BMP、GIF、JPG、PNG的文件的类型为IMAGE;

[0022] 文件扩展名为AVI、MOV、ASF、RM、MPG的文件的类型为VIDEO;

[0023] 文件扩展名为TXT、DOC、PDF、XLS的文件的类型为TXT、DOC、PDF、XLS;

[0024] 文件扩展名为RAR、ZIP、ISO、MDF的文件的类型为COMFILE;

[0025] 文件扩展名为EXE、HTML、MID、TMP的文件的类型为REST。文件类型的划分是为了改进HAR文件归档方法的不足,因为如果将大量不同类型的小文件追加到一个合并文件中,用户查找这个文件时需要遍历两层索引,会使得文件读取效率降低,如果对小文件按扩展名进行划分,将相似类型的小文件追加在一个同类型的合并文件中,在各类型文件数量和大小均衡的情况下,可成倍提高文件检索效率,因为此时系统只需检索存储该类型小文件的

合并文件即可。

[0026] 优选地，所述根据上传的文件的大小分为小文件、中文件和大文件，具体划分方式为：

[0027] 大于等于4.3MB的为小文件，大于4.3MB小于等于32MB的为中文件，大于32MB的为大文件；

[0028] 由于HDFS中一个标准存储块的可用体积是64MB，但并非小于这个体积的文件都称为小文件，都需要进行合并，小文件判定阈值设计得高会导致文件没有合并的必要，阈值设置得太低会增加一个合并文件内的小文件数量，增加索引遍历的时间，进而影响系统的读取效率。通过对大量实验数据的分析和读取性能的考虑，该设计将文件的大小划分为3档，即小于或等于4.35MB的文件为小文件，大于4.35MB而小于或等于32MB的文件为中文件，这两种类型的文件都需要传入文件合并队列等待文件合并模块的读取。当文件大于32MB时，系统判定用户上传文件不为小文件，直接传入HDFS进行存储。尽量让大小相似的小文件存储在一个合并文件中，避免个别中型或大型文件的出现占用大量剩余空间，出现这种情况可能会导致当前合并文件因为容量不足而重新创建新的合并文件，影响NataNode数据块的空间利用率，同时，由于每个合并文件的数据块大小均小于64MB，用户访问小文件时不存在跨数据块读取的问题，进一步提高了读取效率。

[0029] 优选地，所述合并操作任务具体为：

[0030] 小文件或中文件以字符数组的形式追加到合并文件末尾，并创建合并文件与小文件的映射关系，即二级索引，其内容包括文件名name，文件类型type，文件大小size、偏移量offset元数据信息。

[0031] 优选地，所述预取缓存模块将用户频繁访问的文件信息映射关系加载到本地内存中，具体为：

[0032] 在小文件的元数据信息中增加一个频率读取标记frequency，用户每发起一次文件的读取指令，相应文件的读取频率数+1，将读取频率最高的前10％的文件作为热点文件，直接将其文件映射关系加载到本地，用户访问热点文件时，可以不经过NameNode，首先在本地内存热点文件目录中根据一级索引找到存放该文件DataNode数据块的具体位置，再根据数据块中二级索引定位到小文件。因为省略了与NameNode的通信过程，且需要遍历的索引仅为文件索引的10％，其次由于直接从本地内存进行读取，故极大提升了检索效率。所述预取缓存模块定期进行热点文件目录的更新

[0033] 优选地，还包括碎片整理模块，在HDFS中，用户操作层发出删除指令后，系统会执行两个步骤的操作，首先遍历NameNode中该文件的元数据信息，其次，查找存放该文件内容的Blocks，修改该数据块的可用标志位为无效，将删除的对象放入回收站。如果用户发出强制删除的命令，则调用delete()操作强制删除目标文件，同时删除NameNode中该文件的索引信息和元数据信息。对于小文件的删除可参考HDFS的删除机制，即在小文件的元数据信息中增加一个可用标志位available，用户调用RESTFUL接口发出删除指令后，并不立即进行删除操作，而是将操作记录保存在delete.log中，只有当用户再次请求该小文件时，首先遍历delete.log中的该小文件的信息，若此文件已经被删除，则依次遍历该文件类型合并文件的一级索引以及DataNode中的二级索引，将该小文件元数据中的可用标志位修改为false，如果是跳过回收站命令，则直接删除该小文件的索引信息及元数据。同时查看此合

并文件的有效利用率是否小于20％，如果是，则将该合并文件合并进系统当前正在运行的同类型合并文件中，删除该合并文件在NameNode中的元数据信息，等待HDFS系统回收资源。也可集成HDFS的状态监控模块，当系统整体可用空间小于30％时启动碎片整理模块。碎片整理模块启动后，整个系统的存储空间可能面临资源的重新分布，其中包括小文件元数据信息的重写、索引的重新构建及小文件的再次合并等操作，因此非常消耗系统资源，且为了保护用户的文件数据，一般只有当整体存储空间不足时启动该模块的功能。

[0034] 优选地，还包括数据加密模块，所述数据加密模块采用MD5加密算法对用户密码进行数据加密，当用户登录时，系统把用户输入的密码进行MD5Hash运算，然后再和保存在数据库中密码的MD5值进行比较，进而确定输入的密码是否正确，由于MD5加密是不可反推的，系统在并不知道用户密码明码的情况下就可以确定用户登录系统的合法性，这样可以避免密码被具有管理员权限的用户知道。为了进一步增强用户密码的复杂性，可以对用户输入的密码进行加盐操作。同时将直接存入HDFS集群的大文件与经所述文件合并模块合并后的合并文件上传至HDFS时进行AES加密操作。因为AES加密算法可以利用软件快速实现，适合加解密大量的数据，由于其采用对称密码体制，加解密的效率非常高，保证了用户的访问效率。至于AES的安全性，经过广泛的研究可以得到一个结论：现如今还没能找到比进行过暴力破解获取密钥来攻击AES更为行之有效的手段。因此，保证AES加密安全性的一个重要手段就是保证加密及解密密钥不被窃取。

[0035] 与现有技术相比，本发明技术方案的有益效果是：

[0036] 该发明重点改进了Hadoop在处理小文件过程中存储性能和读写效率较低的问题。首先，小文件经过预处理模块，创建文件元数据信息并进入合并队列，其次，文件合并模块根据文件类型和实际容量进行合并操作并创建小文件的二级索引，最后，系统将上传的大文件或合并后的小文件存入HDFS，在文件写入HDFS之前需要在本地进行AES加密操作，本发明设置了预取和缓存模块，并定期更新热点文件索引。其优点主要表现在以下几个方面：

[0037] (1)文件预处理模块对从客户端传入的小文件按扩展名类型进行分类，使得相同类型的小文件合并在一起，系统在读取某个小文件时只需要遍历与其文件类型一致的合并文件的二级索引即可，假设在各种文件类型数量大小均衡的情况下与HAR文件归档方法相比，检索效率可提高N倍，N为划分的基本合并文件类型。

[0038] (2)本发明将上传的文件大小划分为三个等级，小于或等于4.3MB的为小文件，主要针对图片、文档、PDF或由用户输入的文字等，这类小文件在网络中大量存在，大于4.3MB且小于等于32MB的为中文件，主要针对小视频、日志、压缩文件等，大于32MB的为大文件，主要针对ISO、MOV等可能需要分块存储的文件。这样划分的依据是为了应对网络上越来越多出现的小文件，最大限度的利用合并文件的存储容量，避免文件大小分布不均而导致某些合并文件存储容量浪费的问题，由于每个合并文件均小于64MB，用户进行读取操作时不存在跨数据块访问的问题，从而进一步提高了读取速度。

[0039] (3)考虑到某些小文件会有被删除的可能，但HDFS中并没有相应的垃圾回收机制，因为只有直接存入HDFS的文件才会在NameNode中创建元数据和索引信息，将其标记为不可用会删除其元数据信息，但合并文件中仅有一个小文件也会占用64MB的存储空间，所在合并文件的元数据仍然会被保留在NameNode中，浪费存储空间。因此本设计改进了HAR文件归档方法中文件一旦创建不能删除的不足，同时兼顾到系统性能，本发明提出了一种懒更新

的碎片整理机制，首先将用户的删除操作记录在delete.log中，当用户请求访问已删除的小文件时，遍历delete日志，并将该小文件元数据的可用标志位设置为false，同时检查小文件所在合并文件的有效容量，若小于20％，重新将此合并文件合并进系统中当前同类型的合并文件，并删除该合并文件在NameNode中的元数据信息。

[0040] (4)为了提高文件的读取效率，本发明提出了一种文件预取和缓存机制，利用热点文件的概念，用户每读取一次文件，便将该文件元数据信息中的frequency属性加1，定时统计频率最高的10％文件作为热点文件，系统启动时便将热点文件的索引信息，即小文件与DataNode中数据块的映射关系加载到本地内存中，用户读取时首先从热点文件中进行查找，接着依次遍历一级索引和二级索引，极大提高了文件读取的效率。

[0041] (5)数据加密模块主要是为了保证用户账户及文件的安全性而设置的，本设计采用MD5加密算法对用户输入的密码进行加密，使得系统在并不知道用户密码明码的情况下就可以验证用户登录的合法性，文件数据则使用加解密效率高的AES算法，弥补了Hadoop加密算法的不足。

附图说明

[0042] 图1为一种基于HDFS的高性能文件存储与管理系统整体架构图。

[0043] 图2为一种基于HDFS的高性能文件存储与管理系统中一个完整的文件上传过程示意图。

[0044] 图3为一种基于HDFS的高性能文件存储与管理系统中一个完整的文件读取过程示意图。

[0045] 图4为一种基于HDFS的高性能文件存储与管理系统中一个完整的文件删除过程示意图。

[0046] 图5为一种基于HDFS的高性能文件存储与管理系统中一个完整的用户登录及文件数据加密步骤示意图。

具体实施方式

[0047] 附图仅用于示例性说明，不能理解为对本专利的限制；

[0048] 为了更好说明本实施例，附图某些部件会有省略、放大或缩小，并不代表实际产品的尺寸；

[0049] 对于本领域技术人员来说，附图中某些公知结构及其说明可能省略是可以理解的。

[0050] 下面结合附图和实施例对本发明的技术方案做进一步的说明。

[0051] 实施例1

[0052] 一种基于HDFS的高性能文件存储与管理系统，如图1，包括HDFS客户端、文件预处理模块、文件合并模块、预取缓存模块、碎片整理模块和数据加密模块，其中：

[0053] 所述文件预处理模块接收客户端上传的文件并对上传的文件进行大小计算和类型判定，并创建该文件的元数据信息，根据上传的文件的大小分为小文件、中文件和大文件，小文件、中文件传入文件合并队列等待文件合并模块的读取，大文件直接存入HDFS集群；

[0054] 所述元数据信息包括文件名称name、文件类型type、文件大小size和创建日期time,如下:

[0055]
```
public class File{
  String name;
  String path;
long length;
long offset;
  String type;
int blockNum;
```

[0056]
```
  Date time;
  String creator;
  Boolean available;
int frequency;
}
```

[0057] 所述类型判定为根据文件扩展名类型对文件进行分类,具体为:

[0058] 文件扩展名为BMP、GIF、JPG、PNG的文件的类型为IMAGE;

[0059] 文件扩展名为AVI、MOV、ASF、RM、MPG的文件的类型为VIDEO;

[0060] 文件扩展名为TXT、DOC、PDF、XLS的文件的类型为TXT、DOC、PDF、XLS;

[0061] 文件扩展名为RAR、ZIP、ISO、MDF的文件的类型为COMFILE;

[0062] 文件扩展名为EXE、HTML、MID、TMP的文件的类型为REST。

[0063] 所述根据上传的文件的大小分为小文件、中文件和大文件,具体为:

[0064] 大于等于4.3MB的为小文件,大于4.3MB小于等于32MB的为中文件,大于32MB的为大文件。

[0065] 所述文件合并模块读取文件合并队列中小文件或中文件,根据文件类型判断需要追加到的目标合并文件类型,若属于该合并文件的剩余容量大于等于小文件或中文件的大小,则启动合并操作任务并创建合并文件与小文件或中文件的映射关系,即二级索引,若该合并文件的剩余容量小于小文件或中文件的大小时则将当前合并文件和二级索引作为一个数据块存入HDFS集群中,同时创建一个新的且与该合并文件类型一致的合并文件,将文件合并队列中待合并文件写入新的合并文件中;合并文件存入HDFS集群中;

[0066] 所述预取缓存模块将用户频繁访问的文件信息映射关系加载到本地内存中,建立客户端与HDFS集群的DataNode的直接联系。在小文件的元数据信息中增加一个频率读取标记frequency,用户每发起一次文件的读取指令,相应文件的读取频率数+1,将读取频率最高的前10%的文件作为热点文件,直接将其文件映射关系加载到本地,用户访问热点文件时,可以不经过NameNode,首先在本地内存热点文件目录中根据一级索引找到存放该文件DataNode数据块的具体位置,再根据数据块中二级索引定位到小文件。所述预取缓存模块定期进行热点文件目录的更新。

[0067] 所述碎片整理模块在所述系统对小文件执行删除时，若合并文件的有效占比小于20％，将合并文件中有效的小文件合并到系统当前正在运行的同类型合并文件中，合并任务完成后，删除该合并文件中全部的小文件数据信息和NameNode中合并文件的元数据信息。

[0068] 所述数据加密模块采用MD5加密算法对用户密码进行数据加密，同时将直接存入HDFS集群的大文件与经所述文件合并模块合并后的合并文件上传至HDFS时进行AES加密操作。

[0069] 所述合并操作任务具体为：

[0070] 小文件或中文件以字符数组的形式追加到合并文件末尾，并创建合并文件与小文件的映射关系，即二级索引，其内容包括文件名name，文件类型type，文件大小size、偏移量offset元数据信息。

[0071] 所述预取缓存模块将用户频繁访问的文件信息映射关系加载到本地内存中，具体为：

[0072] 所述碎片整理模块中集成有HDFS集群的状态监控模块，所述状态监控模块对HDFS集群的可用空间进行监控。

[0073] 在具体实施过程中，本实施例的系统实现完整的文件上传过程、完整的文件读取过程、完整的文件删除过程、完整的用户登录及文件数据加密步骤，如图2；

[0074] 一个完整的文件上传过程，如图2，包括以下步骤：

[0075] S01：用户调用RESTFUL接口请求文件写入操作，携带MultipartFile和目标路径(destPath)参数进入文件预处理模块；

[0076] S02：文件预处理模块根据传入的文件信息创建文件元数据信息，包括统计文件大小，根据扩展名进行类别划分，更新上传时间等；

[0077] S03：文件预处理模块的逻辑判断单元比较文件元数据信息的size属性，若size<＝4.3MB或4.3MB<size<＝32MB，则判定为S文件或M文件，传入文件合并队列，否则，文件内容直接写入HDFS中；

[0078] S04：文件合并模块从合并队列中取出下一个待合并的文件，根据其元数据信息的type属性判断要追加到的合并文件类型，若当前系统中此类型合并文件的剩余容量大于或等于待合并的小文件大小，系统启动合并操作，即将待合并的小文件追加到合并文件末尾，同时创建小文件与合并文件的映射关系，即二级索引。若此类型合并文件的剩余容量小于待合并的小文件大小，系统将当前合并文件写入HDFS，并重新创建一个同大小同类型的合并文件，启动合并操作将待合并的小文件写入新的合并文件。同时监听合并队列读取下一个小文件的数据信息；

[0079] S05：HDFS中NameNode接收到文件存储信号，对文件信息进行备份，对大于64MB的文件切分存储到DataNode上，由于合并文件大小都小于64MB，可以直接分配给一个DataNode进行存储。DataNode确认写入文件的数据信息，将写入结果报告给NameNode，以便形成与文件信息一一对应的元数据；

[0080] S06：NameNode将文件写入结果返回给客户端，等待客户端确认，此时写入文件操作完成，关闭相应的数据流。

[0081] 一个完整的文件读取过程，如图3包括以下步骤：

[0082] S11:用户调用RESTFUL接口请求文件读取命令,携带等待读取的文件信息(FileInfo)和Http响应体(HttpResponse)进入文件预处理模块;

[0083] S12:根据待读取的文件信息获得文件存储数据,如文件名和扩展名,根据文件名在本地热点文件目录查找该文件,如果查找到,则获得该文件的位置信息,客户端直接与DataNode建立连接;

[0084] S13:若在S12中,本地缓存目录未找到待读取的文件的位置信息,则继续遍历非合并文件的一级索引,如果仍未找到,则根据文件扩展名判断合并文件类型,依次遍历该合并文件类型下的所有二级索引;

[0085] S14:若在S13中,NameNode一级索引或DataNode二级索引中均未找到文件的位置信息,则将异常结果返回给客户端,若检索得到文件信息,客户端与该DataNode建立连接;

[0086] S15:DataNode运行read()操作,将文件所在位置的Blocks中的数据接连传送给Client,并给该文件元数据信息的frequency属性+1;

[0087] S16:客户端将文件读取结果返回给用户,读取文件操作完成,关闭相应的数据流。

[0088] 一个完整的文件删除过程,如图4,包括以下步骤:

[0089] S21:用户调用RESTFUL接口发出删除操作指令,程序携带指令参数如文件名、目标路径或删除标记进入删除模块;

[0090] S22:系统将用户的删除操作记录在delete.log日志中,其中包括是否强制删除文件等信息;

[0091] S23:用户请求某小文件时,首先遍历delete.log目录,如果是已删除小文件,则检索该小文件类型合并文件的一级索引和对应的二级索引,将该小文件元数据信息的可用标记位设置为false,如果是强制删除,则删除该小文件的元数据信息。

[0092] S24:系统调用碎片整理模块统计合并文件中小文件的容量占比,若合并文件中可用小文件占比小于20%时,将此合并文件中可用的小文件合并进系统中正在运行的同类型合并文件中,同时删除合并文件的元数据信息,释放存储资源。系统也可集成HDFS的状态监控模块,当整体空间可利用率小于30%时,启动碎片整理程序;

[0093] S25:在编辑日志中记录删除结果,并将结果汇报给客户端,删除操作完成后,关闭相应数据流。

[0094] 一个完整的用户登录及文件数据加密步骤,如图5,包括以下步骤:

[0095] S31:用户输入账号和密码尝试登录,程序调用Login方法启动登录流程;

[0096] S32:系统将用户从页面输入的密码进行MD5加密,并将用户账号和密码与数据库存储的用户信息进行比对,若找到匹配用户,则登录成功,赋予用户特定权限,否则登录失败,尝试重新登录;

[0097] S33:用户在客户端提供待加密的文件路径或手动指定明文F,程序进入文件预处理模块;

[0098] S34:如果是小文件,文件进入合并模块进行合并操作,其次,数据加密模块对待写入HDFS的大文件数据块或合并文件使用AES加密算法进行加密操作,得到密文数据C;

[0099] S35:系统将密文数据C通过网络传输至服务器进行保存,关闭相应的数据流并将存储结果报告给客户端。

[0100] 相同或相似的标号对应相同或相似的部件;

[0101] 附图中描述位置关系的用语仅用于示例性说明，不能理解为对本专利的限制；

[0102] 显然，本发明的上述实施例仅仅是为清楚地说明本发明所作的举例，而并非是对本发明的实施方式的限定。对于所属领域的普通技术人员来说，在上述说明的基础上还可以做出其它不同形式的变化或变动。这里无需也无法对所有的实施方式予以穷举。凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明权利要求的保护范围之内。

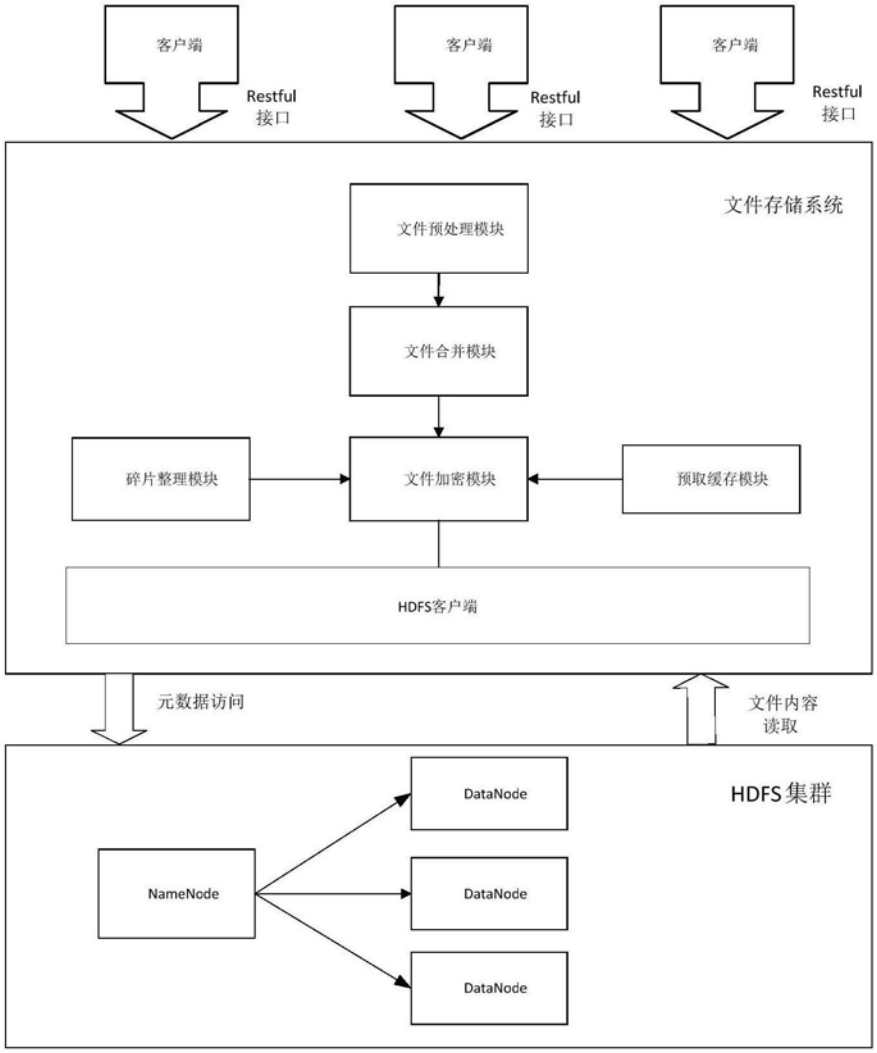

客户端
Restful
接口
客户端
Restful
接口
客户端
Restful
接口
文件存储系统
文件预处理模块
文件合并模块
碎片整理模块
文件加密模块
预取缓存模块
HDFS客户端
元数据访问
文件内容
读取
HDFS集群
DataNode
NameNode
DataNode
DataNode

图1

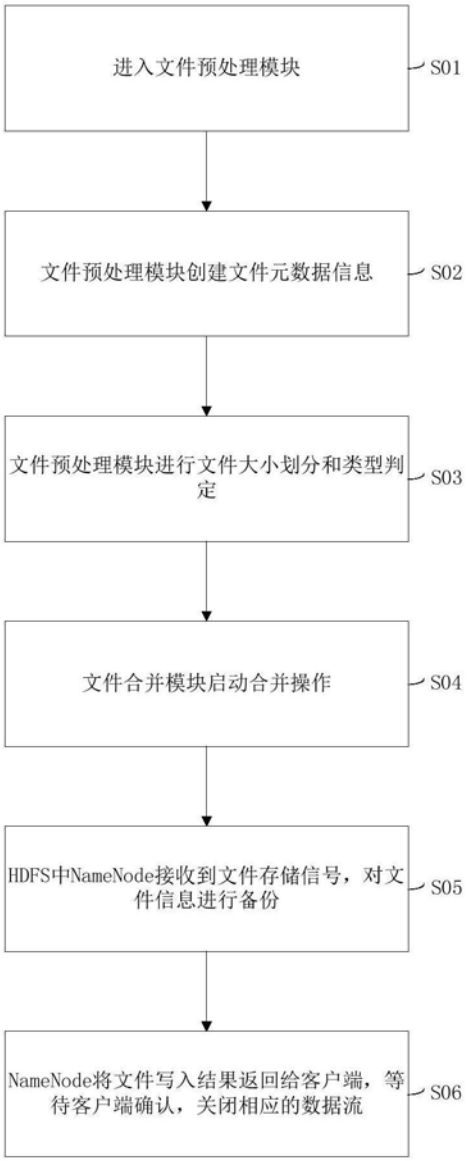

进入文件预处理模块
S01
文件预处理模块创建文件元数据信息
S02
文件预处理模块进行文件大小划分和类型判定
S03
文件合并模块启动合并操作
S04
HDFS中NameNode接收到文件存储信号，对文件信息进行备份
S05
NameNode将文件写入结果返回给客户端，等待客户端确认，关闭相应的数据流
S06

图2

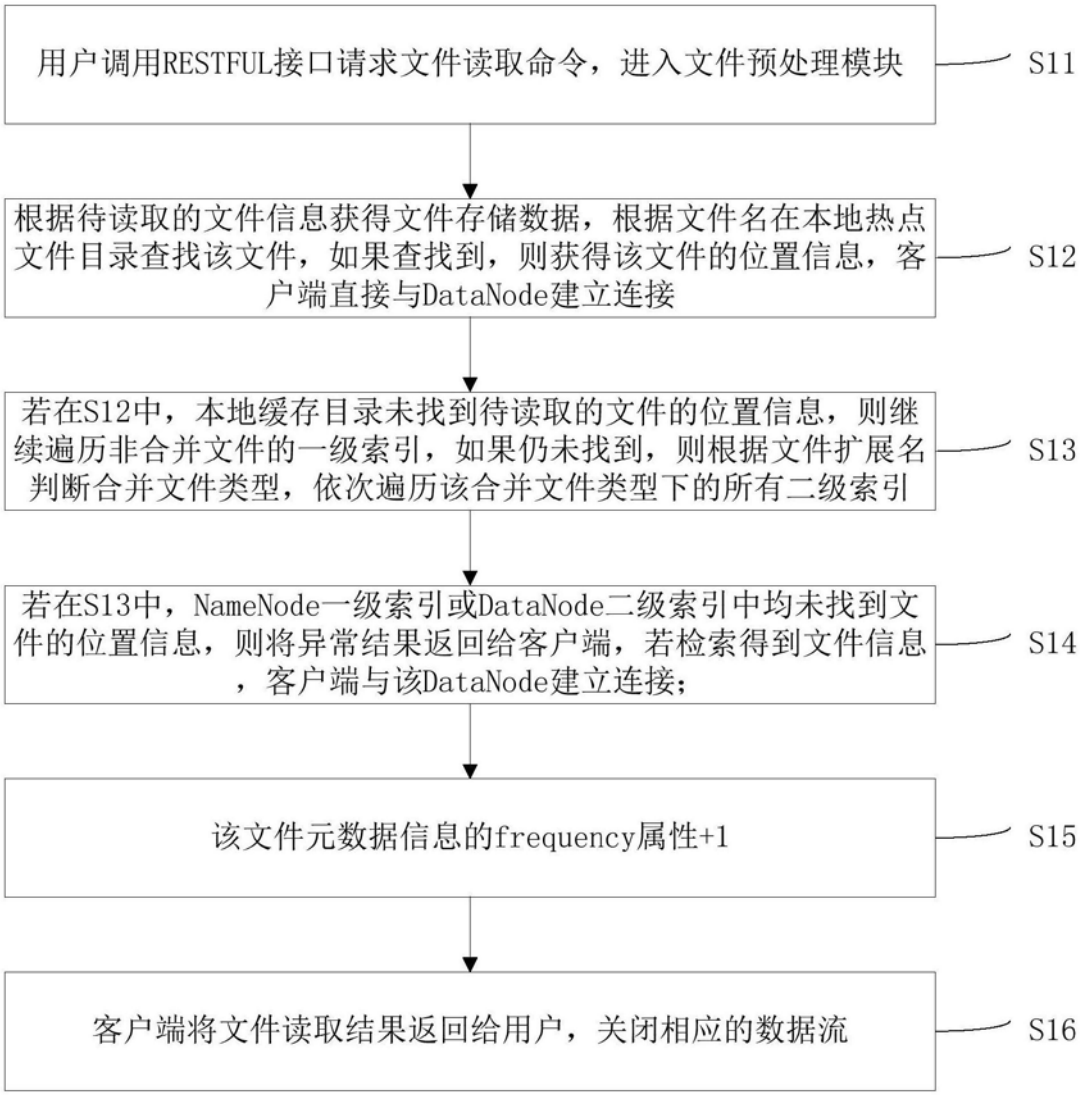

用户调用RESTFUL接口请求文件读取命令，进入文件预处理模块
S11
根据待读取的文件信息获得文件存储数据，根据文件名在本地热点文件目录查找该文件，如果查找到，则获得该文件的位置信息，客户端直接与DataNode建立连接
S12
若在S12中，本地缓存目录未找到待读取的文件的位置信息，则继续遍历非合并文件的一级索引，如果仍未找到，则根据文件扩展名判断合并文件类型，依次遍历该合并文件类型下的所有二级索引
S13
若在S13中，NameNode一级索引或DataNode二级索引中均未找到文件的位置信息，则将异常结果返回给客户端，若检索得到文件信息，客户端与该DataNode建立连接；
S14
该文件元数据信息的frequency属性+1
S15
客户端将文件读取结果返回给用户，关闭相应的数据流
S16

图3

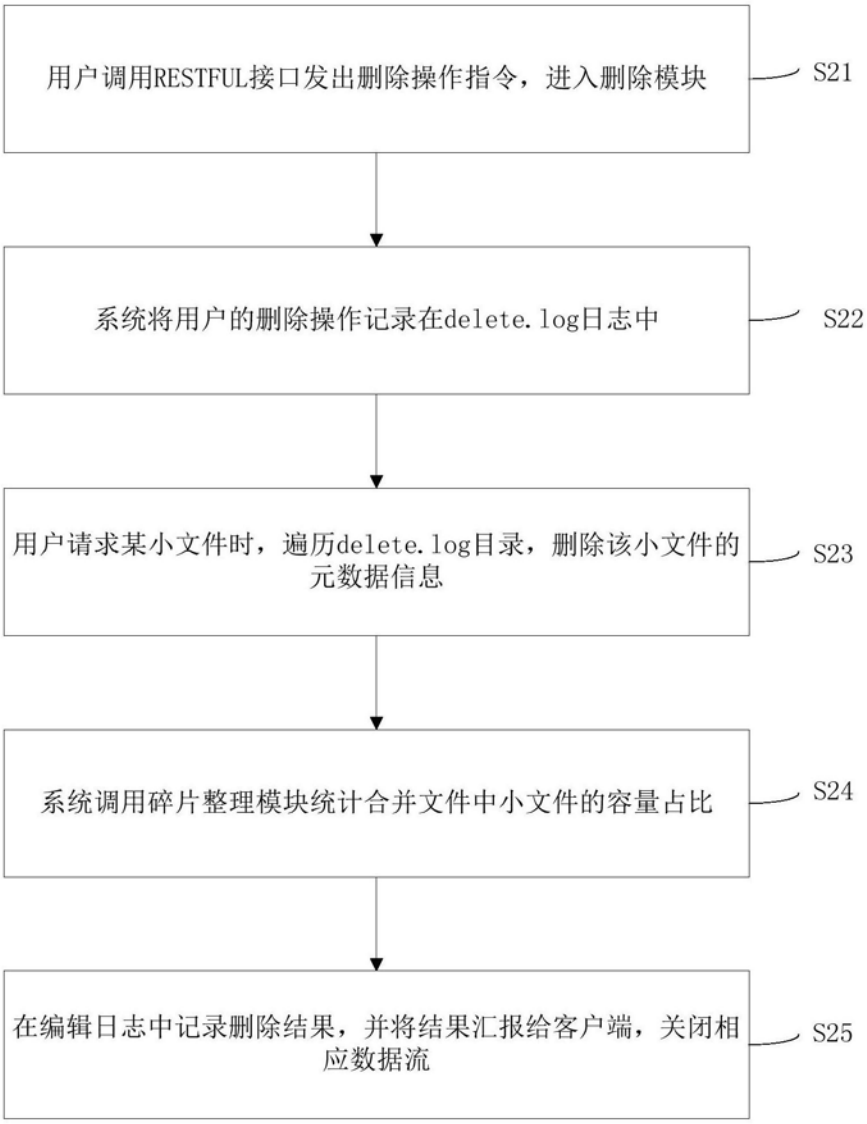

用户调用RESTFUL接口发出删除操作指令，进入删除模块
S21
系统将用户的删除操作记录在delete.log日志中
S22
用户请求某小文件时，遍历delete.log目录，删除该小文件的元数据信息
S23
系统调用碎片整理模块统计合并文件中小文件的容量占比
S24
在编辑日志中记录删除结果，并将结果汇报给客户端，关闭相应数据流
S25

图4

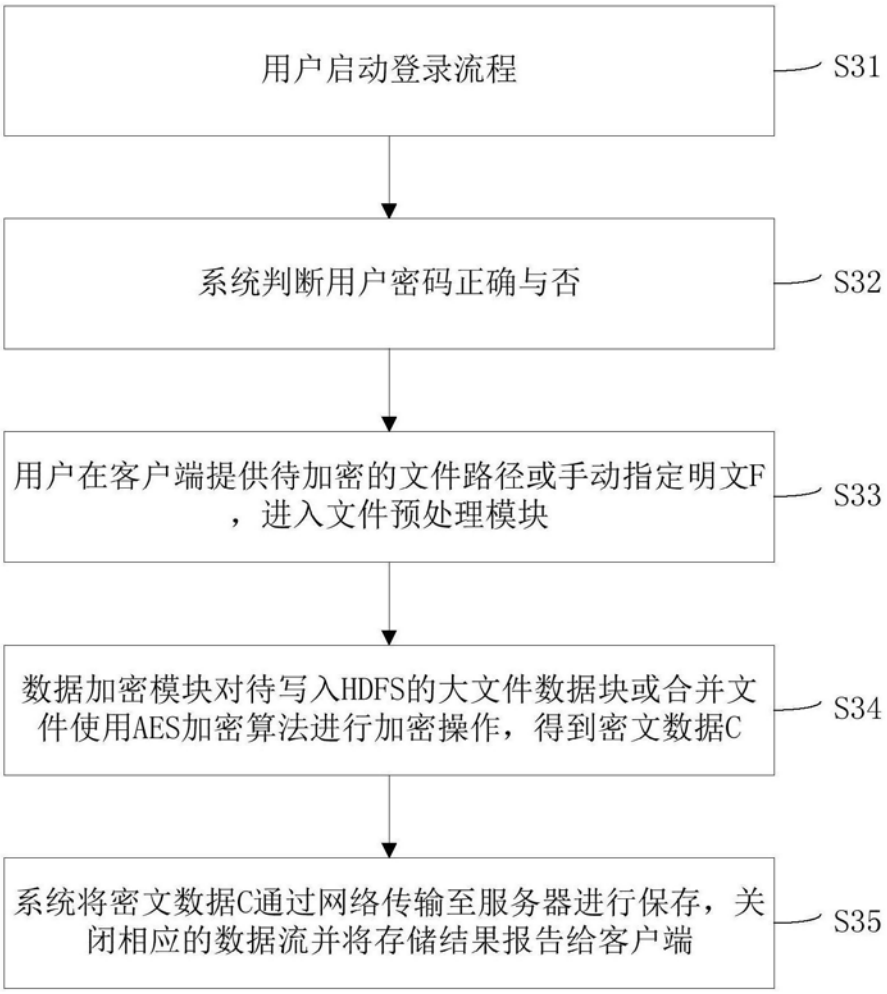

用户启动登录流程
S31
系统判断用户密码正确与否
S32
用户在客户端提供待加密的文件路径或手动指定明文F，进入文件预处理模块
S33
数据加密模块对待写入HDFS的大文件数据块或合并文件使用AES加密算法进行加密操作，得到密文数据C
S34
系统将密文数据C通过网络传输至服务器进行保存，关闭相应的数据流并将存储结果报告给客户端
S35

图5