基于聚类的重复数据去冗算法的研究

刘 赛[1],聂庆节[1],刘 军[1],王 超[2],李 静[2]
(1.南瑞集团公司,江苏 南京 210003;
2.南京航空航天大学 计算机学院,江苏 南京 211106)

摘 要: 数据的损坏和丢失会带来无法弥补的损失,数据备份系统可以将损失降到最低程度。随着收集的数据量的迅速增加,备份系统需要备份与恢复的数据也迅速增加,然而备份文件之间的相似度超过 60%,全部存储在硬盘上十分浪费存储空间,故提出了一种基于 K-medoids 聚类的 DELTA 压缩方法,用来去除备份数据中的重复数据。该方法首先对文件进行切割分块,通过对文件块进行两两 DELTA 压缩,得出各自压缩文件的大小,作为两个文件块之间的相似度。通过得到的相似度进行 K-medoids 聚类,作为 DELTA 压缩前的预处理步骤。然后根据 K-medoids 的聚类结果,合并小文件块之后再进行 DELTA 压缩。测试结果表明,该方法提高了压缩率,并减少了 DELTA 压缩中查找指纹的次数,降低了压缩时间。

关键词: DELTA 压缩; 数据压缩; 聚类; K-medoids
中图分类号: TP393 **文献标识码:** A **文章编号:** 1673-629X(2018)02-0125-05
doi: 10.3969/j.issn.1673-629X.2018.02.027

Research on Deduplication Algorithm Based on K-medoids Clustering

LIU Sai[1], NIE Qing-jie[1], LIU Jun[1], WANG Chao[2], LI Jing[2]
(1.NARI Group Corporation, Nanjing 210003, China;
2.School of Computer, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Data damage and loss will lead the irreparable losses which can be minimized by data backup system. With the increasing amount of data collection, data backup system has to deal with more and more data of backup and recovery, but the similarity between the various backup files is more than 60% so that all the data stored in the hard disk will be a waste of storage space. For this, we propose a DELTA compression method based on K-medoids clustering to remove duplicate data from the backup data. It firstly segments and blocks the files, and then obtains the size of each compression file by means of DELTA compression between the two blocks as the similarity of them. K-medoids clustering is performed by the similarity obtained as preprocessing steps before DELTA compression. According to the K-medoids clustering, we merge the small similar file blocks before DELTA compression. The tests show that the proposed method can improve the compression rate, reduce the number of fingerprints in DELTA compression and shorten the compression time.

Key words: DELTA compression; data compression; clustering; K-medoids

0 引 言

随着物联网时代的到来,从传感器节点到企业日常业务中,收集到的数据越来越多[1],收集大量数据组成的信息系统不仅方便了人们的生活,而且为未来决策提供了参考,但是人们也必须面对数据丢失的危险。灾备系统[2]是数据保护最后的防护,用来提高数据的安全性。然而备份存储中心中存在大量的冗余数据[3],比如在多次备份时每次都存储的全量备份中大部分都是重复数据,在各个版本的备份文件中也含有数目巨大的重复数据。备份系统中存在的重复数据导致了需要备份的数据量迅速增加。

备份海量的数据需要使用大量的磁带,而大量的冗余数据浪费了许多磁带。传统的数据压缩[4]方法是采用文件编码中冗余的信息进行压缩,没有全局压缩的思想,文件的压缩效率很差,同时不能进行不同文件间的重复数据压缩。与传统数据压缩算法不同,DELTA 压缩算法[5]是全局压缩技术,针对在不同备份系统中,不同备份文件间进行重复数据检测和消除,利用不同文件之间的重复信息进行冗余数据消除,大大提高了文件的压缩率。由香农的信息论[6]原理可知,在

收稿日期: 2017-03-14 **修回日期:** 2017-07-20 **网络出版时间:** 2017-11-15
基金项目: 国家电网公司总部科技项目(0711-150TL173)
作者简介: 刘 赛(1988-),男,工程师,研究方向为信息安全、云灾备。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20171115.1436.056.html

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

任意非随机生成的文件中都包含冗余信息，可以利用统计建模等方法，得到一个字符或者短语出现的频率，用最短的字符表示出现频率最高的数据。利用这种统计建模构造压缩算法的编码方式包括游程编码[7]、熵编码[8]以及字典编码[9]。利用这些编码方式，较长的字符串可以用较短的字符串信息表示。另外，Zhang等[10]介绍了线性回归原理的分段常数逼近算法(PMC-MR)[11]及其改进算法(PMC-MENAN)[12]，其算法思想是利用时间相关性，在压缩过程中确定数据差距限值，用分段函数的函数式拟合原始数据，同时记录获取最大、最小以及两者差值。当差值超过设定的最大误差值时，用序列的持续时间和数值大小来替代。Wang等[13]提出了基于重复数据消除的备份数据加密方法，根据分块内容的哈希值生成分块对称密钥，用于将明文分块与密文分块一一对应，采用分块对称密钥方法保证了用户私钥与身份识别口令不同时，数据库中的备份数据无法被外界解密破解，同时也保证了文件的压缩效率。Kang等[14]提出面向容灾的备份数据透明加密机制，着重解决备份数据的安全问题，采用一种层叠式文件系统来面向容灾的备份数据进行透明加密，利用虚拟磁盘透明加密的方法，当数据写入时进行加密，数据读取时进行解密，保护了备份数据的安全性和机密性。

DELTA压缩通过保存一份参考文件，采用内容无关的方法从文件中选取特征集，在系统中找出一份与新文件具有高度相似性的参考文件，再计算相似文件之间的DELTA压缩编码，在备份系统中直接存取DELTA压缩编码，不再存储源文件。采用DELTA压缩算法可节省存储空间、减少需要传输的数据量，提高归档的效率。DELTA算法进行重复数据删除时有较高的压缩率，但是随着压缩数据规模变大，所需存储的指纹块数也随之增长。大量的指纹块存储在硬盘上，读取磁盘数据所需时间远大于读取内存所需时间，导致查询时间过久无法用于实际生产环境。采用K-medoids聚类[15]进行预处理的DELTA压缩算法可以提高压缩率，同时减少了查找指纹的次数。

1 相关技术

1.1 重复数据删除技术

重复数据删除可以划分成两种类型，第一种是删除相同数据，即仅仅保留一份相同的数据，然后用指针指向其他位置出现的相同数据。第二种是查找到相似度高的数据，采用DELTA压缩算法对相似数据编码存储，节约存储空间。DELTA压缩的核心思想是将重复的数据标记出来，并存储一份，当再次遇到这部分数据之后，用指针指向其存储位置，以减少需备份的文件。

把文件看作是由字符、数字、字节等按一定顺序组成的字符串。DELTA压缩算法可以视为对字符串从左至右进行编码，当出现一个字典里面不存在的字符串时，向DELTA压缩编码中添加(ADD,L,S)命令字符串，表示在该位置添加长度为L的字符串S，如果发现该字符串已经存储过一份，则添加(COPY,L,O)命令字符串，表示从R中复制长度为L，偏移量为O的字符串到该位置。具体实现通过定义DELTA压缩函数(记为D)，它的输入为一个待压缩文件V和一个参考文件R，输出为一个DELTA文件$\Delta(R,V)$。当备份系统进行DELTA压缩时，首先从文件中选择适当字串，通过计算每个字串的指纹得到指纹集合，利用指纹集合中相同指纹的个数来确定两个文件的相似度。

DELTA压缩算法针对R和V之间的冗余数据，当R中已存在该数据时，保存$\Delta(R,V)$的复制命令，将其指向R中对应数据的链接，针对R中不存在的数据，用添加命令显式添加。R和V相似度较高时，$\Delta(R,V)$比V小得多；R和V完全相同时，$\Delta(R,V)$就是一个$<C,R>$命令。因此，通过保存$\Delta(R,V)$来替代V可大大节省存储空间。

1.2 K-medoids聚类

K-medoids聚类是一种基于划分的聚类算法。与K-means算法以簇内所有样本点的均值作为簇聚类中心不同，K-medoids聚类是以簇中实际样本点作为簇的聚类中心，减少K-means算法对少数误差点的敏感性。K-medoids聚类首先任意选择K个不同的数据对象作为初始簇中心，其他数据对象根据其与簇中心的距离，将它们分配到距离最近的簇，然后在每个簇内部顺序选择一个非簇中心点的数据代替簇中心点，选择代价最小的点作为簇中心点，然后反复迭代，用非簇中心点来代替簇中心点，当一个中心点被某个非中心点替代时，除了未被替换的中心点外，其余各点将被重新分配。当聚类中心和各个点所属聚类簇都不再变化时，算法结束。K-medoids聚类算法使用最接近聚类中心的对象作为簇中心，增强了算法的鲁棒性，具有数据抗干扰性强、聚类结果与输入顺序无关以及对小数据集聚类效果明显等特点。

2 基于K-medoids聚类的重复数据去冗算法

对于一个新的数据块，根据其与其他数据块之间的相似度，会出现两种情况，分别如图1中的S_1和S_2所示。图中的每个点都表示一个对应的数据块，两个点之间的距离表示二者之间的相似度。S_1表示该数据块属于一号簇的情况，它到其他簇的距离远大于到一

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

号簇的距离,并且与它相近距离的点也属于一号簇。S_2 表示该数据块不属于任意簇,与其最相似的数据块是属于其他簇的数据块,但是这些数据块都在其他簇的边界。即表示 S_2 也可作为聚类中心,它周围的数据块与它都较相似,也可以达到高效压缩,所以,通过设置阈值的方法无法考虑到各个文件之间的相似性。K-means 算法是采用簇内所有点的均值作为簇聚类中心,无法计算该中心到其他数据块之间的距离(相似度),故无法评价该聚类中心是否合理。K-medoids 聚类算法可以通过不断迭代的方式选择簇内的一个点作为聚类中心,使得每个数据块可以归属到更加合适的聚类簇中,所以采用 K-medoids 聚类算法作为 DELTA 压缩前的预处理。

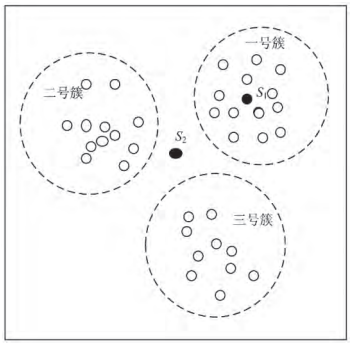

图 1　相似数据聚类模型描述

针对 DELTA 压缩大文件时所需数据指纹较多、查找速度慢等问题,且当文件足够大时,内存将无法存储,所以将大文件切割成小文件块,然后将小文件块两两合并,进行 DELTA 压缩,并把 DELTA 压缩后的文件大小作为文件的相似度,根据得到的相似度进行 K-medoids 聚类。最后,按照聚类结果合并小文件块,再将合并后相似度高的文件进行 DELTA 压缩。基于 K-medoids 聚类的 DELTA 压缩的框架图如图 2 所示。

2.1　K-medoids 聚类处理

对需要压缩的大文件进行切分,采用 1 M 文件大小作为划分单位,之后对划分的文件块两两之间进行 DELTA 压缩,DELTA 压缩后的文件大小存储在临时矩阵 arr_delta [N] [N],将它作为文件块之间的相似度。为方便叙述,现在统一规定本文所用符号。

定义: 将待查重的数据块集合记为 $S=\{S_1,S_2,\cdots,S_n\}$;将聚类前的簇记为 $C=\{C_1,C_2,\cdots,C_k\}$;将聚类后的簇记为 $C'=\{C_1',C_2',\cdots,C_k'\}$;两个相似数据段之间的距离度量记为 $\mathrm{dist}(S_i,S_j)$,两个数据块的相似度的评价函数记为 $\mathrm{delta}(S_i,S_j)$。

文中选用 K-medoids 聚类算法对元数据进行聚类,以相似度矩阵中所保存的压缩率(相似度)信息作为聚类依据,查重数据块标号组为 $S=\{S_1,S_2,\cdots,S_n\}$。然后,对集合 S 中的数据对象进行聚类,将数据段分为 K 类 $C'=\{C_1',C_2',\cdots,C_k'\}$。

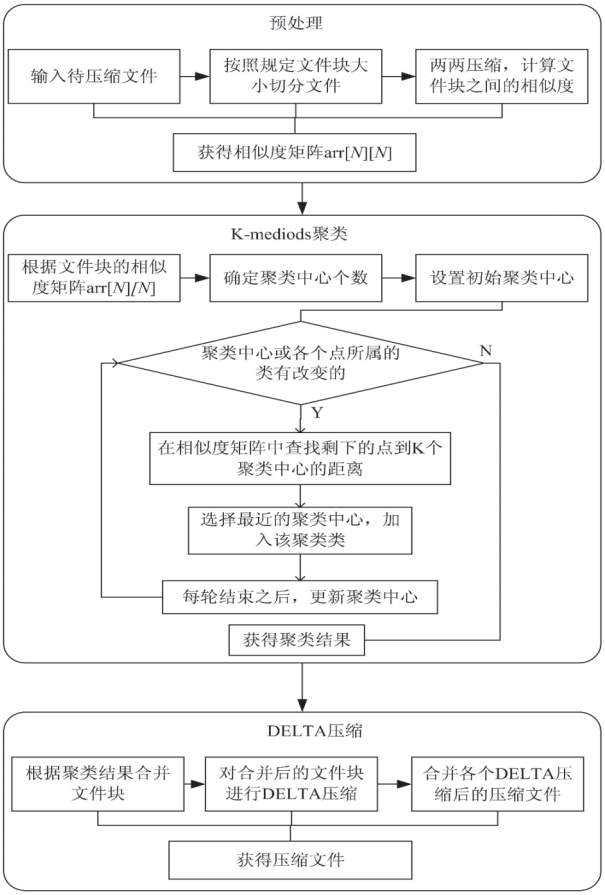

图 2　基于 K-medoids 聚类的 DELTA 压缩框架图

两个相似数据块之间的相似度表示为二者的 DELTA 距离,即:

$$\mathrm{dist}(S_i,S_j)=\mathrm{delta}(S_i,S_j) \tag{1}$$

整个聚类的流程为: 在 S 中任意选择 K 个聚类中心点,分别用 $\{m_1,m_2,\cdots,m_k\}$ 表示。将代表剩余数据块的点分配给距它最近的簇中,获得聚类簇 $C=\{C_1,C_2,\cdots,C_k\}$。然后对每一个簇 $C_i,i\in\{1,2,\cdots,k\}$,遍历簇中的第 i 个非中心点对象 S_j,用式(2)计算当 S_j 代替中心点 m_i 的总代价。

$$\mathrm{cost}=\sum_{i=1}^{l_i}\mathrm{dist}(m_i,s_j)=\sum_{i=1}^{l_i}\mathrm{delta}(m_i,s_j) \tag{2}$$

选择簇内最小的总代价点作为新簇的中心点。迭代以上步骤,直到各个簇的中心点不再变化。最终获得的 K 个簇 $C'=\{C_1',C_2',\cdots,C_k'\}$,即每个簇中的数据块之间的相似度都较高。

2.2　聚类后 DELTA 压缩

经过以上步骤的预处理,得到了各个相似的数据块的聚类集合,根据聚类结果将文件块合并。合并之后的大文件内部之间的相似度较高,即通过合并相似文件块之后,$|\Delta(R,V)|$ 小于甚至远小于 $|V|$,考虑通过 $\Delta(R,V)$ 替代存储 V,可以达到压缩文件大小的目的。根据聚类后的结果进行去冗余,对文件整体进

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

行数据压缩。在重复数据去冗阶段再次使用 DELTA 压缩算法消除重复数据。

对于聚类后的待压缩文件，首先采用内容无关的方法从文件中选取特征集，根据可分配内存的大小，确定产生中间指纹数量与文件大小。设定一个滑动窗口的大小，不断向前移动滑动窗口，计算移动窗口下的数据指纹，在特征数据库中搜索一个与它高度相似的参考文件，找到该参考文件后，根据压缩函数 D 进行压缩。为了提高检索速度，降低查找时间，采用 Hash 函数映射成超级特征或超级指纹集。若超级指纹相匹配，则两个文件的相似度较大。通过以上算法找到了与该文件具有极大相似性的文件，然后通过函数 D 对有序符号串进行编码，利用 ADD 编码命令(其命令格式为(ADD,L,S))，表示在 V 中的指定位置添加长度为 L 的字符串 S；COPY 编码命令(其命令格式为(COPY,L,O))，表示从 R 中复制长度为 L，偏移量为 O 的字符串到 V 中的指定位置。获得经过 DELTA 压缩后的文件 $\Delta(R,V)$，最后存储压缩文件。基于 K-medoids 聚类的重复数据去冗算法的整体流程如图 3 所示。

```
Input: 输入大备份文件(待压缩文件)
Output: 输出已经压缩好的文件
(1) 输入大备份文件;
(2) 按照文件大小=1 M 切割该文件;
(3) for i=1: M
{
for j=1: M
{
第 i 个文件和第 j 个文件合并
进行 DELTA 压缩
将压缩后的文件存储到临时数组 arr_delta [N] [N]
}
}
(4) 确定聚类个数 K
(5) while 聚类中心或者聚类中的点有变化
do 根据 arr_delta [N] [N] 确定各个点所属类
for i←1 to K
do 更新聚类中心
end
end
输出聚类情况
(6) 根据聚类情况，合并文件块
(7) 将合并后的文件块进行 DELTA 压缩
```

图 3 基于 K-medoids 聚类的重复数据去冗算法

3 实验验证

为了验证基于 K-medoids 聚类的 DELTA 压缩算法的有效性，利用一台配置为 Intel Core i5-3470，2 GB 内存，SATA 5 600 转磁盘的台式计算机作为测试环境进行测试。利用 Oracle 数据库的测试数据作为待压缩数据，测试基于 K-medoids 聚类的 DELTA 压缩效果。通过在测试集上直接使用 DELTA 压缩与先聚类再进行 DELTA 压缩两种方式进行对比，来验证该算法的有效性。

首先，定义压缩率为 DELTA 压缩文件与待压缩文件大小的比率，即压缩率为:

$$\text{compress\_rate}=\frac{\text{经过压缩之后的文件大小}}{\text{总文件大小}} \quad (3)$$

式(3)表明压缩率的值越小越好。DELTA 压缩的算法思想表明待压缩文件与同一个参考文件之间的相似性越高，压缩效果就越好，换言之，文件内含有的相似数据越多，文件的压缩率越高。接下来进行实验验证。

3.1 压缩性能测试

为了验证基于 K-medoids 聚类方法的 DELTA 压缩方法的性能，对比源文件大小、经过 DELTA 压缩后的文件大小以及聚类之后再经过 DELTA 压缩后的文件大小。

首先将大文件切割为每个文件大小为 1 M 的小文件，作为初始的源文件。然后将切割后的每个文件直接进行 DELTA 压缩，作为另一组对比实验。将切割后的文件块两两合并，然后对合并后的文件进行 DELTA 压缩，将 DELTA 压缩后的文件大小作为文件间的相似度。再根据文件间的相似度利用 K-medoids 对文件块进行聚类，根据聚类结果进行文件块合并，然后再进行 DELTA 压缩，作为最后一组对比实验。

针对 Oracle 数据库测试数据进行测试，选取测试数据的前 30 个文件块进行描述。

由式(3)可得，直接通过 DELTA 压缩的压缩率约为 50%；经过 K-medoids 聚类预处理之后再经过 DELTA 压缩的压缩率约为 25%。

结果如图 4 所示。

直接经过 DELTA 压缩，压缩率在 50%，经过 DELTA 压缩算法可以使得文件压缩率降低 40%以上，在此基础上，经过 K-medoids 聚类预处理之后再进行 DELTA 压缩的方式，即在 DELTA 压缩之前依据相似度进行聚类，又可提高 50%左右的压缩率。

由此可证，通过聚类后的文件有很大程度上的相似，可大大提高文件内的相似度。对于相似度越高的文件，DELTA 压缩的效率越高。

在第 2 节中，分析得出重复文件删除的效率瓶颈在于当检测重复指纹时需要大量的读磁盘操作，尤其当指纹数量超过内存大小时，检索算法需要大量的 I/O 操作将指纹从磁盘中读入到内存，使得程序效率大

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

大降低。将大文件切割成小文件,再按照相似度相近的原则进行聚类,然后以数据指纹可以在内存上可接受为原则进行合并。再对合并之后的文件进行 DELTA 压缩,通过对数据 DELTA 压缩前的预处理步骤来提高数据的压缩率。经实验验证,效果有明显提高。

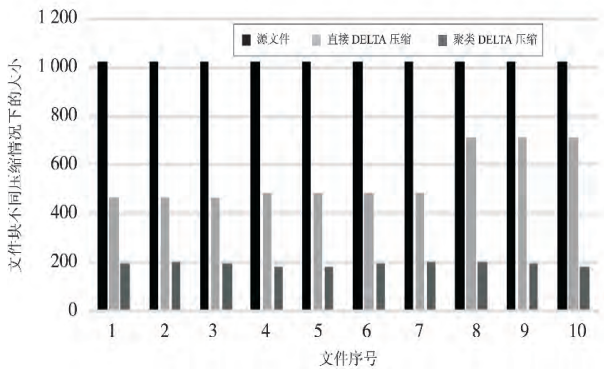

图 4　部分文件块压缩对比

3.2　聚类合并性能测试

在数据压缩中,可以采取对小文件直接压缩、随机合并文件压缩(一般采取直接按照序号合并)、聚类合并后压缩三种方式,对比这三种方式的压缩情况,如图 5 所示。

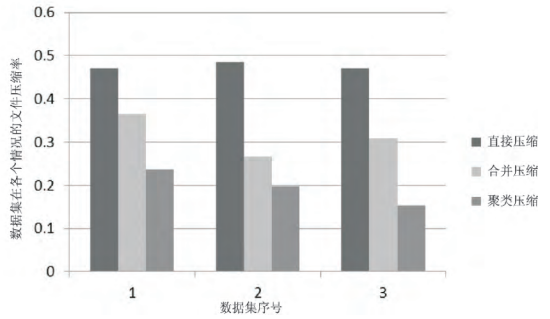

图 5　部分数据集在不同方式下的压缩对比

经过实验证明,直接对小文件进行压缩时,得到的压缩率最小; 对小文件合并之后再进行压缩,可以提高压缩率; 根据文件块相似度差异的大小先聚类再进行合并,可以大大提高文件压缩率。通过聚类将小文件进行合并,提高了合并后文件的相似度,之后再进行 DELTA 压缩,减少了查找相似文件块的时间,并且提高了文件的压缩率。采用文件切分、聚类、合并、再压缩的方式,可以保证数据指纹的数量在一定的限度内(保证数据指纹存储在内存中),减少磁盘的 I/O 操作,有助于大文件的压缩。从测试结果来看,聚类后压缩可以很好地达到压缩效果。这也证明了聚类后压缩去冗的有效性。

4　结束语

针对大文件在进行 DELTA 压缩时数据指纹过多的问题,提出了一种基于 K-medoids 聚类算法的 DELTA 压缩方法。测试结果表明,该算法可有效地减少数据指纹数量、提高压缩率,使重复数据删除的效果更加明显。下一步将研究如何设置最优的聚类中心个数,以进一步提高压缩率,降低压缩时间。

参考文献:

[1] 蒋　鹏,吴建峰,吴　斌,等.基于自适应最优消零的无线传感器网络数据压缩算法研究[J].通信学报,2013,34(2):1-7.

[2] 毛　波,叶阁焰,蓝琰佳,等.一种基于重复数据删除技术的云中云存储系统[J].计算机研究与发展,2015,52(6):1278-1287.

[3] 马发勇,厉启鹏,马志斌,等.电力调度 SCADA 系统中历史数据压缩及存储策略[J].电网技术,2014,38(4):1109-1114.

[4] QUAN W,XU C,GUAN J,et al.Scalable name lookup with adaptive prefix bloom filter for named data networking[J].IEEE Communications Letters,2014,18(1):102-105.

[5] 付印金,肖　侬,刘　芳.重复数据删除关键技术研究进展[J].计算机研究与发展,2012,49(1):12-20.

[6] 刘厚贵,邢　晶,霍志刚,等.一种支持海量数据备份的可扩展分布式重复数据删除系统[J].计算机研究与发展,2013,50:64-70.

[7] MEISTER D,BRINKMANN A.Multi-level comparison of data deduplication in a backup scenario[C]//SYSTOR 2009: the Israeli experimental systems conference.New York,NY,USA: ACM,2009.

[8] 张沪寅,周景才,陈毅波,等.用户感知的重复数据删除算法[J].软件学报,2015,26(10):2581-2595.

[9] BILIMOVIC A.Boundary value problems for nonlinear fractional differential equations[J].Acta Mathematica Scientia,2015,31(4):1337-1346.

[10] 屈志伟.无线传感器网络数据压缩算法综述[J].科技创新与应用,2015(32):90.

[11] AHMAD B,NTOUYAS S.Existence of solutions for fractional differential inclusions with nonlocal Riemann-Liouville integral boundary conditions[J].Boundary Value Problems,2014,139(3):451-465.

[12] MIN J,YOON D,WON Y.Efficient deduplication techniques for modern backup operation[J].IEEE Transactions on Computers,2011,60(6):824-840.

[13] 王　灿,秦志光,冯朝胜,等.面向重复数据消除的备份数据加密方法[J].计算机应用,2010,30(7):1763-1766.

[14] 康潇文,杨英杰,杜　鑫.面向容灾的备份数据透明加密机制[J].计算机工程,2009,35(20):131-133.

[15] 祁　兰,毛燕琴,沈苏彬.一种传感数据的压缩和高效存储方案[J].计算机技术与发展,2016,26(11):177-181.

?1994-2018 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net