基于FSL数据集的去重性能分析

曹 晖，张秦正

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】重复数据删除技术作为一种数据缩减技术，实现了对高度冗余数据集的压缩功能，可以有效地解决存储系统空间浪费所带来的成本开销问题。相较于过去大多针对小规模静态快照或是覆盖时间较短的快照的研究，该文基于从共享用户文件系统选取的覆盖时间较长的大规模快照，从文件、数据块以及用户的角度研究备份数据集的特征，分析不同数据分块方法、策略下去重性能的优缺点，得到最高的重复数据删除率，为未来的重复数据删除系统设计提出建议。

关 键 词 重复数据删除； 重删率； 元数据； 存储

中图分类号 TP391 **文献标志码** A doi:10.3969/j.issn.1001-0548.2018.04.023

Deduplication Performance Analysis Based on FSL Dataset

CAO Hui and ZHANG Qin-zheng

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract As a data reduction technology, the deduplication technology realizes the compression function of highly redundant data sets, and can effectively solve the overhead cost which is caused by the waste of space in the storage system. Compared to the previous studies which were mainly based on small-scale static snapshots or short-coverage snapshots, the highest deduplication ratio can be achieved by using large-scale snapshots with a long-coverage time. The large-scale snapshots are selected from the shared user file system. The characteristics of backup datasets from files, data blocks, and users are studied, and the advantages and disadvantages of different data partitioning methods and strategies are analyzed. The proposed result suggests a reference for future deduplication system design.

Key words data deduplication; deduplication ratio; metadata; storage

随着互联网信息的快速增长，数据所需存储空间也不断增加。仅在过去十几年间，企业存储数据已达到EB级，且数据的增长速度远超存储空间的发展速度，导致数据存储成本大幅增加。IDC(international data corporation)调查发现[1]，仅2006年—2010年间，数据就从161 EB增加到988 EB，并以每年57%的速度飞速增长。IDC预计，到2020年人类所产生的数据总量将达到40 ZB。全球的数字化浪潮所引发的爆炸性数据增长，使得人们对数据的存储需求也大幅提高。

如今的存储系统中有60%是冗余数据，且随着时间的推移这个比例还会越来越高[2]。因此重复数据删除技术受到了越来越多的科研机构及企业的关注[3]。重复数据删除技术可以分析出存储系统中的重复数据并对其进行删除，以减少存储空间达到节省存储成本的目的。

当前对于重复数据删除的研究主要集中在以下4个方面[4]：1) 对重复数据删除率(简称“重删率”)的提高，通过挖掘并删除更多的重复数据，节省存储空间降低存储能耗；2) 提高重复数据删除性能，保证在删除重复数据时不影响数据系统的吞吐率；3) 提高重复数据删除可靠性，保证删除重复数据后留存的数据孤本安全；4) 满足系统的扩展性。

数据重删技术目前广泛应用于数据备份系统和云存储系统中，在对数据进行多次备份后，存储系统中产生大量的重复数据，令该技术有足够的用武之地。衡量重复数据删除技术的标准主要有重复数据删除率和重复数据删除性能两个方面。重复数据删除性能主要取决于重删系统实现所用的技术，而重删率与数据集本身的特点和应用情况有关，如数据变化率、备份策略及保存时间等。

1 重复数据删除技术

重复数据删除技术是一种存储优化技术，可有

收稿日期：2017－07－21；修回日期：2017－11－15

作者简介：曹晖(1995－)，女，主要从事信息安全、重复数据删除方面的研究.

效对数据进行压缩以实现对存储空间的优化。重复数据删除系统对数据指纹进行比对，若存在指纹相同的数据，则仅保存其中一份数据，删除其余重复数据，并将剩余重复数据以索引的形式进行保存[5]。以存储重复数据元数据[6]信息的形式代替实际存储，从而实现提高存储效率、提高传输速度、节省数据存储成本的目的[7]。目前重复数据删除技术已大量应用于数据备份、远程灾备、归档存储中[8]。其中现有的云存储重删系统主要有Lessfs和OpenDedup[9]。

重复数据删除可分为文件级和数据块级的去重。文件级的数据重删技术即单一实例存储(single instance store, SIS)[10]，是以文件大小为粒度，操作简单但效率不高[11]。数据块级重删技术去重长度很小，可缩小到2～128 KB之间。目前针对数据块级数据去重有3种数据块分块算法[12]，即固定长度分块算法、CDC变长分块算法和滑动块分块算法。固定长度分块算法采用预先设置好的分块长度分割文件，分割完成后对其进行弱校验值和MD5算法强校验值检测。CDC算法是应用Rabin指纹[13]将文件切割成大小不一的分块。与固定长度分块算法不同的是，它是基于文件内容对文件进行分块，因此数据块长度是可变的。滑动块相似检测技术结合了可变分块长度和固定分块长度检测技术两者的优点，在现有的研究中发现，对于较大的文件，基于CDC算法的重复数据检测表现较好，滑动块技术更适用于粒度较小的情况。

2 数据集和工具

2.1 数据集概述

目前针对备份数据集的研究所使用的数据集大多涵盖的是周期较短或静态的快照。本文着眼于用户长期真实的有效数据，力求实现对用户实际使用情况的分析，推导结论并为以后的设计及研究工作起到一定的指导作用。本文研究的数据集是FSL(file systems and storage lab)提供的Homes数据集。

Homes数据集上的数据均来自Linux系统软件开发人员，他们在几个联合项目上共同从事开发工作。Homes数据集几乎包含了共享文件系统上每位用户主目录下的文件快照，在快照的收集过程中，利用Fs-hasher工具[14]为每位用户创建一个每日快照(包括整个文件分块和不同平均块长度的分块)。本文选取并分析了从2013年1月22日−2013年6月17日共56.20 TB的数据，该数据使用7种不同长度的分块(2～128 KB)和整个文件分块的方式，以此来细致比对不同分块方式对重复数据删除的影响。为了减少快照收集的时间并降低收集后数据集的存储规模，研究选取了48-bit的MD5哈希算法。Homes数据集共覆盖了33个用户的数据，本文选取了其中最具代表的5个用户，使研究工作能够针对每个用户进行分析。虽然没有收集完整的文件内容，但丰富的元数据和大量的散列可以用以进行广泛的研究。

基于Homes数据集，本文进行了如下分析：1) 比较不同文件大小下的重复数据删除率，寻找重删效率最高的文件大小；2) 分析比较不同的备份策略(增量备份和完全备份)和不同的分块大小对备份数据集重删率的影响，寻找最高效的分块大小和备份策略；3) 基于用户使用特征的分析，对具有相同背景和使用行为的用户备份数据进行分析，为集群存储的设计提供建议；4) 比对不同类型的文件在不同分块大小下与其对应的重删率，分析不同分块下重复率差异的成因，以此对存储系统提供建议。

2.2 数据分析工具

本文使用了开源的Fs-hasher工具包。该工具包主要包含fs-hasher和hf-stat，fs-hasher对文件系统进行扫描并收集包含丰富元数据信息的快照。该工具并不收集文件的实际内容，而是将每个文件按照一定长度进行分块并收集块对应的哈希值。fs-hasher可收集固定长度或可变长的分块以及每一个块的压缩率等，收集到的快照信息存储在哈希文件中。hf-stat工具用于解析由fs-hasher收集的哈希文件，该工具将哈希文件中包含的信息输出为可读形式，包括文件的哈希值、分块后的块哈希值、用户及其所在群组的信息以及文件的元数据信息等。hf-stat还提供了控制和过滤其输出的选项的功能，方便用户对输出信息进行筛选以便后期对特定信息的处理。

3 数据集的分析及结果

重复数据删除系统的关键指标之一是重复数据删除率。重复数据删除率定义为原始数据集的大小除以存储介质的大小。与其他备份数据集的重复删除研究相比，本文所研究的数据集中的重删率相对较高。为实现覆盖周期长的研究目的，将收集每天全部的快照且不删除，以保留尽可能多的历史快照；而在实际备份情景中，为了降低存储成本，许多部署的备份系统会定期删除快照。

3.1 单维度分析

3.1.1 基于文件的分析

本文对于文件的分析是基于对整个文件分块的

方式进行的，在不同的数据集中，重复数据删除的执行效果有所不同。Homes数据集中超过98%的文件小于1 MB(3.2.1节将详细说明)，但这些数量巨大的小文件消耗的存储空间占比不足5%。图1为不同文件大小下的重复数据删除率。

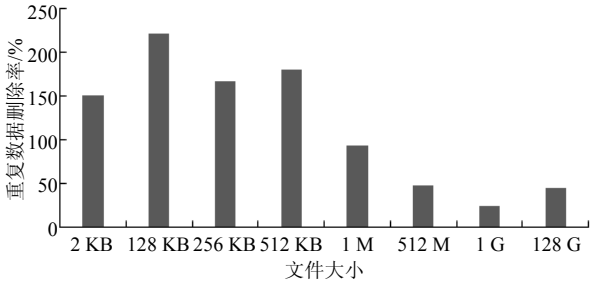

图1　不同文件大小的重复数据删除率

由于统计方法受限，本文只对图中所示的分组进行了计算，但也可以观察到，文件大小超过1 MB的文件重复数据删除率较低；反观大小为128 KB、256 KB和512 KB的文件，其重复数据删除率都较高。因此可以得出，数据集消耗的存储空间主要由具有较低重复数据删除率的大文件占用。因此，以整个文件进行分块对于节省数据集的存储空间的效果不显著。

3.1.2　基于数据块的分析

本文选择了两种较为典型的备份策略：完全备份和增量备份。对于增量备份，本文所用的方法是检测新添加和修改的文件。通过比较两个连续的快照来确定是否新添加了一个文件；通过比对文件中元数据的Mtime来确定自上一个快照以来被修改的文件。图2显示了使用不同备份策略和块大小时的原始重复数据删除率。

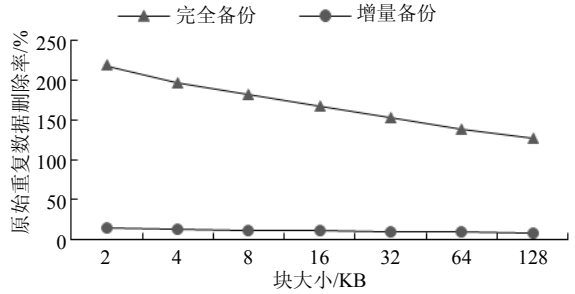

图2　不同备份策略和分块大小下的原始重复数据删除率

完全备份相比增量备份策略有更高的重复数据删除率，因为增量备份会在存储前对比两份快照。然而，较小的分块长度意味着需要更高的元数据开销，因为必须存储更多的散列等信息来用于恢复和检测重复。为了计算各种分块长度的重复删除效率，本文采用了文献[15]提出的方法重新计算有效重复数据删除率。该算法定义为：假设L是原始数据大小，P是数据重删之后的大小，F定义为每个块元数据大小除以分块长度后的值，那么原始重复数据删除率为$D=L/P$，元数据的大小为$F\times(L+P)$，其中$F\times L$是文件表单(Manifest)的大小(即恢复文件原始内容所需的信息)，$F\times P$是哈希索引值的大小，所以实际存储在介质上的总大小即为：$P+F\times(L+P)$。基于此公式，将重复数据删除率公式进行修改，用D'表示实际重复数据删除率：

$$D'=\frac{L}{P+F\times(L+P)}=\frac{D}{1+F\times(D+1)}$$

Homes数据集基于48 bit的MD5哈希算法，每个块的元数据大小为30 byte，运用上述公式计算得到各分块大小的实际重复数据删除率，如图3所示。由图3可以看出完全备份中重复数据删除效果最好的分块长度为32 KB；而对于增量备份，重复数据删除执行效果较好的分块大小为2 KB；这表明最佳分块长度可能取决于备份策略和备份频率。图3还表明，随着分块大小的增大，重复数据删除率下降并不明显，同时较大的分块可以显著减少元数据数量，从而大大减少I/O块索引的数量。

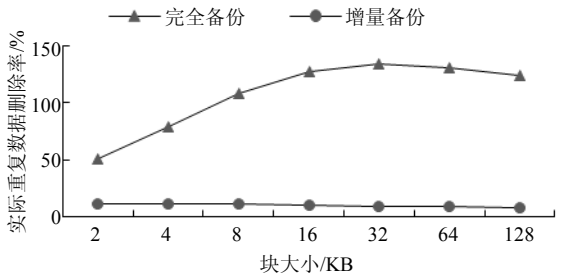

图3　不同备份策略和分块大小下的实际重复数据删除率

3.1.3　基于用户的分析

以往研究往往侧重于从整体分析数据集，而没有从用户的角度进行研究。虽然不同用户组成了数据集整体，但每个用户的数据也具有不同的特性。

在研究用户之间的重复数据时，发现当任意两个用户之间的共享数据占比较大时，这些用户可以分成一个组。将用户自己的数据集定义为S，$|S|$表示该用户数据集中块的个数，用户X和用户Y之间共享的数据是$S_X \wedge S_Y$。用户X和Y的共享数据在用户X和Z的共享数据中的占比为[14]：

$$\frac{|(S_x \cap S_y)\cap(S_x \cap S_z)|}{\min(|S_x \cap S_y|,|S_x \cap S_z|)}$$

本文选取了具有代表性的5个用户(用户4、8、12、22和26)的数据信息。根据上式得出，用户4和12之间共享的块的96%也可以在用户4和26的共享数据中找到。且在这5个用户中，每两个用户的共享数据均在其余共享数据中占比90%以上，说明该组中的用户所共享的数据有极高的相似度，因此在设计重复数据删除系统中，可以将这些用户划分为同一群组，以此来提高系统整体的重复数据删除率。

3.2 多维度分析

3.2.1 基于文件和用户的分析

图4为不同用户不同文件大小的分布。可以看出文件大小多集中在1～64 KB之间，只有极少数为大文件(＞1 MB)。但是存储空间大部分被大型文件(如.vmdk)占据，使得文件的平均值大小要远高于文件集中出现的区间。因此设计备份数据系统时可将若干个较小的文件合并为压缩文件以进一步减少存储空间。

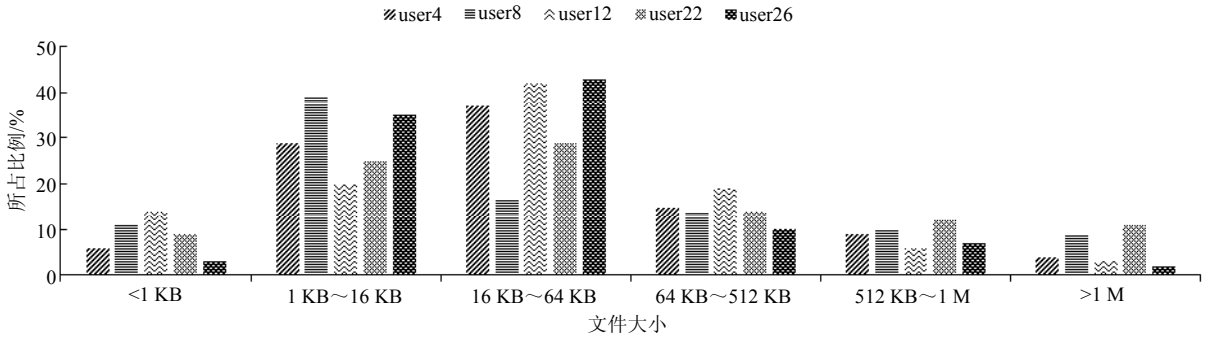

图4 不同用户不同文件大小的分布

3.2.2 基于文件和数据块的分析

本文对不同文件类型的文件进行了研究分析。图5显示了数据集中几种占用存储空间较大的文件类型占用总存储空间的比例，可以看出虚拟机映像(.vmdk)和虚拟硬盘(.vdi)消耗了整个数据大小的近70%。

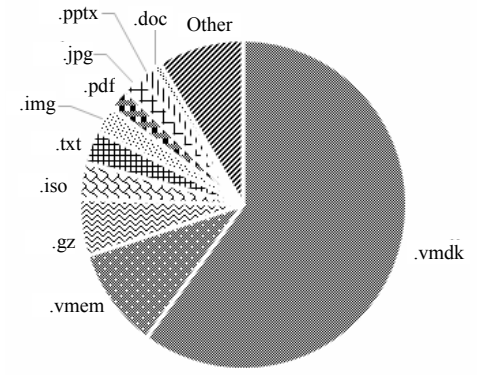

图5 不同文件类型的空间占比

图6显示了文件分块大小下不同文件类型的重复数据删除率。此部分的研究选择了占用磁盘空间最多的几种文件类型。从图中可以看出，不同类型的文件的重复数据删除率有着显著的差异：在8 KB的分块大小下，各文件类型的重删率为23%～65%。此外，不同类型文件的重删率对分块大小也有不同的敏感性。对于某些文件类型，分块大小从8 KB增加到128 KB导致数据重删率大幅下降，如.vmdk类型文件下降25%，.disk类型文件下降44%。但是，对于大多数文件类型，重复数据删除率并不随着分块大小的增大而降低，某些文件类型(如.mp3、.jpg等)的重复数据删除率根本没有变化。造成这种现象的原因如下：1) 压缩文件、MP3和视频等类型的文件在软件开发环境中可能很少被改变，因此分块大小对这几种类型的文件的重删率不会造成太大影响; 2) 当收集数据快照时，有的小文件没有被合并成一个单独的更大的块，这意味着这些小文件本身就是一个块，其大小小于最低分块大小。因此，对于那些较小的文件类型，一旦分块大小超过文件本身的大小，随后增大分块大小将对此类文件不造成影响。

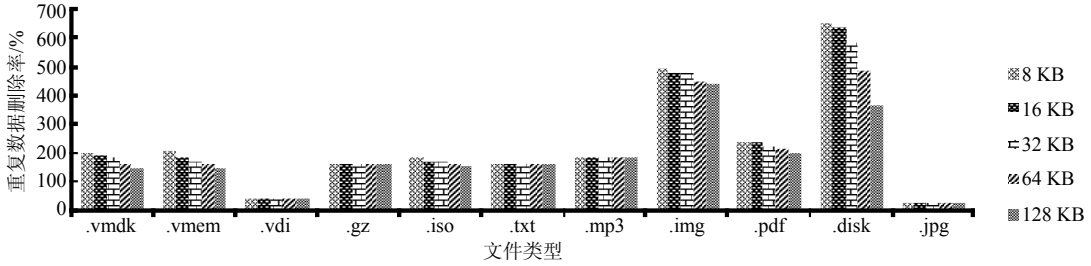

图6 不同分块大小下不同文件类型的重复数据删除率

4 结 束 语

重复数据删除技术能大规模地压缩数据量，从而节省冗余数据所带来的存储开销。不过，目前的数据重删技术大多针对小规模的静态快照或覆盖时间较短的快照，而针对长时间、大规模的数据集的研究较少。本文重在研究一个跨度6个月的公开数据集，并使用Fs-hasher工具包，从文件、数据块以及用户的角度进行研究，研究发现：1) 整个文件分块在数据重删中表现不佳，因为大文件在数据集中占据大部分存储空间且具有很低的重复数据删除率；2) 越小的数据分块并不总能节省存储空间，越小数据分块意味着更高的元数据存储开销，在设计重删系统时采用32 KB或更大的分块大小效果更佳；3) 相似行为的用户拥有更多的相同数据，将这些用户组成一个集群可更大限度地实现重复数据删除；4) 不同文件类型对不同文件分块大小的敏感度不一。设计重复数据删除系统时可将不同类型的文件进行差异性去重。本文的研究对重复数据删除系统的设计提供了建议，未来将对集群存储的重复数据删除进行研究。

参 考 文 献

[1] GANTZ J F, REINSEL D. Extracting value from chaos[R]. [S.l.]: IDC White Paper, 2011.

[2] MCKNIGHT J, ASARO T, BABINEAU B. Digital archiving: End-user survey and market forecast 2006-2010[R]. Milford: The Enterprise Strategy Group, 2006.

[3] 王国华. 高效重复数据删除技术研究[D]. 广州: 华南理工大学, 2014.
WANG Guo-hua. Research on technologies for high-effect data deduplication[D]. Guangzhou: South China University of Technology, 2014.

[4] 李映刚. 重复数据删除技术在图片文件系统中的应用[D]. 成都: 成都理工大学, 2013.
LI Ying-gang. The application of deduplication technology in picture file system[D]. Chengdu: Chengdu University of Technology, 2013.

[5] 张宗华, 屈英, 叶志佳, 等. 基于多特征匹配和Bloomfilter的重复数据删除算法[J]. 深圳大学学报, 2016, 33(5): 531-535.
ZHANG Zong-hua, QU Ying, YE Zhi-jia, et al. Deduplication based on multi-feature matching and bloom filter[J]. Journal of Shenzhen University, 2016, 33(5): 531-535.

[6] 王龙翔, 董小社, 张兴军, 等. 内容分块算法中预期分块长度对重复数据删除率的影响[J]. 西安交通大学学报, 2016, 50(12): 73-78.
WANG Long-xiang, DONG Xiao-she, ZHANG Xing-jun, et al. Influence of expected chunk size on deduplication ratio in content defined chunking algorithm[J]. Journal of Xi'An Jiaotong University, 2016, 50(12): 73-78.

[7] 敖莉, 舒继武, 李明强. 重复数据删除技术[J]. 软件学报, 2010, 21(5): 916-929.
AO Li, SHU Ji-wu, LI Ming-qiang. Data deduplication techniques[J]. Journal of Software, 2010, 21(5): 916-929.

[8] 尚颖丹. 面向文件级重复数据删除的稀疏索引技术[D]. 长沙: 国防科学技术大学, 2012.
SHANG Ying-dan. Sparse indexing for file-level de-duplication[D]. Changsha: National University of Defense Technology, 2012.

[9] 徐奕奕, 唐培和. 基于分数阶Fourier变换的云存储系统重复数据删除算法[J]. 计算机科学, 2015,42(7): 174-177.
XU Yi-yi, TANG Pei-he. Duplicate data remove algorithm of cloud storage system based on fractional fourier transform[J]. Computer Science, 2015, 42(7): 174-177.

[10] 吴鹏，史芳芳. 删除重复数据的一种数据备份方案[J]. 通信管理与技术, 2012(5): 58-60.
WU Peng, SHI Fang-fang. A data backup scheme for deleted duplicated data[J]. Communications Management and Technology, 2012(5): 58-60.

[11] 卞琛, 于炯, 修位蓉. 基于回归检测的滑动块重复数据删除算法[J]. 新疆大学学报, 2017, 34(3): 259-266.
BIAN Chen, YU Jiong, XIU Wei-rong. A sliding blocking algorithm with regressrion-checking for duplicate data detection[J]. Journal of Xinjiang University, 2017, 34(3): 259-266.

[12] 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.
FU Yin-jin, XIAO Nong, LIU Fang. Research and development on key techniques of data deduplication[J]. Journal of Computer Research and Development, 2012, 49(1): 12-20.

[13] RABIN M. Fingerprinting by random polynomials[R]. Cambridge: Technical Report, 1981.

[14] ZHEN S, GEOFF K, SONAM M, et al. A long-term user-centric analysis of deduplication patterns[C]//MASS Storage Systems and Technologies. Santa Clara, CA: IEEE, 2016.

[15] WALLACE G, DOUGLIS F, QIAN H, et al. Characteristics of backup workloads in production systems[C]//Usenix Conference on File and Storage Technologies. Berkeley: USENIX Association, 2012: 262-289.

编 辑 蒋 晓