◎博士论坛◎

分布式存储系统的哈希算法研究

黄秋兰,程耀东,陈 刚
HUANG Qiulan, CHENG Yaodong, CHEN Gang

中国科学院 高能物理研究所计算中心,北京 100049
Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China

HUANG Qiulan, CHENG Yaodong, CHEN Gang. Research on hash algorithm for distributed storage system. Computer Engineering and Applications, 2014, 50(1):1-4.

Abstract: Considering the uniform data distribution in physical storage and efficient data positioning in distributed storage system, this paper studies different hash algorithms and proposes how to measure the merits of hash algorithm in distributed storage system. Based on experiments, the detail comparative analysis of various hash algorithms are shown in aspect of hash value distribution, hash conflict and computational efficiency and application scenarios of those algorithms are pointed out. In addition, the optimal scheme combining with distributed storage file system is demonstrated. Experimental results show that Davies-Meyer algorithm has a good uniform distribution and high computational efficiency which is suitable for distributed storage systems.

Key words: distributed storage system; hash algorithm; hash distribution; hash conflict; Davies-Meyer algorithm

摘 要: 针对分布式存储系统中如何实现数据在物理存储上的均匀分布和高效定位的问题,对多种哈希算法展开研究,提出了衡量分布式存储系统哈希算法优劣的标准;从散列分布性、哈希冲突和计算效率等多个维度对这些哈希算法进行分析比较,指出各种哈希算法的应用场景;结合分布式存储系统的应用,给出最优的哈希算法选择。实验结果证明,Davies-Meyer算法具有很好的均匀分布性和很高的计算效率,很适合分布式存储系统的应用。

关键词: 分布式存储系统;哈希算法;散列分布性;哈希冲突;Davies-Meyer算法
文献标志码: A **中图分类号:** TP393 doi:10.3778/j.issn.1002-8331.1306-0307

1 引言

信息技术的快速发展,个人用户、科学计算、互联网等应用产生了海量的数据。爆炸式增长的数据即将从PB级向EB级迈进,这些数据的存储和高速访问对分布式存储系统在可用性、可扩展性及IO访问性能上提出了新的挑战[1]。首先,随着数据信息总量的扩大,存储系统为了满足需求必须不断地动态扩大存储规模。这使得存储系统必须能够支持新的存储节点不断加入,确保数据在各个存储节点的均匀分布,满足存储空间以及网络带宽的负载均衡。其次,在海量的数据信息中,如何高效查找定位目标数据成为提高系统性能的关键。存储系统必须实现高效的数据定位,最大限度地减少平均响应时间,提高系统的IO性能。这两个技术问题可通过引入哈希算法来解决,以实现均匀的数据分布和高效的数据定位。

本文针对分布式存储系统对多种哈希算法展开研究,文章的贡献主要体现在两个个方面:(1)提出了分布式存储系统中衡量哈希算法优劣的三个指标,为评价哈希算法的好坏提供了测度标准;(2)根据提出的哈希算法优劣的衡量标准,对多种哈希算法从理论和软件模拟两方面进行了比较分析,指出这些哈希算法的应用场景。最后结合分布式存储系统的应用,给出最优的哈希算法选择。

基金项目: 国家自然科学基金(No.11205179);中国科学院知识创新工程基金重大项目(No.KJCX1-YW-17)。
作者简介: 黄秋兰(1982—),女,博士研究生,助理研究员,研究领域为海量数据存储技术;程耀东,博士,副研究员,研究领域为海量数据存储技术和大数据;陈刚,博士,研究员,研究领域为海量数据存储技术和高性能计算。E-mail:huangql@ihep.ac.cn
收稿日期: 2013-06-26 **修回日期:** 2013-08-15 **文章编号:** 1002-8331(2014)01-0001-04
CNKI网络优先出版:2013-09-26, http://www.cnki.net/kcms/detail/11.2127.TP.20130926.1644.004.html

2 哈希算法优劣的衡量标准

分布式存储系统中哈希算法的随机性取决于系统中文件、目录的命名和哈希函数的选择。存储系统中文件和目录的命名规则相同,区分英文字符的大小写,包括的字符有:字母、数字、“.”、“_”(下划线)和“-”(连字符)。系统中的文件和目录利用哈希函数计算哈希值,通过哈希值决定存储系统中数据的分布和定位,因此,哈希算法是保证数据均匀分布和高效定位的关键。

在分布式存储系统中,衡量哈希算法的标准主要有以下几点。

(1)散列分布性:散列在不同区间上的哈希值总数大致相当。

(2)抗冲突性:哈希计算后无法产生多个散列值相同的哈希值,即不同的输入不能出现相同的输出。

(3)计算效率:哈希值的计算效率。

哈希算法的散列分布性越好,则对应的存储系统上的数据分布越均匀,同时计算效率越高,数据定位越快。

3 哈希算法

哈希的英文名为hash,意思为散列,它将任意长度的输入值通过散列算法,变换成固定长度的输出值,这个值就是散列值,即哈希值。哈希值的输出空间一般要比输入空间小很多,不一样的输入也会散列成相同的输出,并且不可能从散列值来唯一确定输入值。因此,可以利用散列值的这一特点来检验数据的完整性。哈希表也叫做散列表,它根据已经设定好的哈希算法和处理数据问题的计算方式,将输入值映射到一个有限的位置空间中,这种存放记录的数组形成的表叫做哈希表,对应的映射函数叫做哈希函数。在算法中所得到的存放空间就是哈希地址,也叫做散列地址。

哈希算法的方式很多,在海量存储系统中主要是对文件名和路径进行哈希,决定数据的分布策略,主流使用的哈希算法有经典的字符串哈希算法、MD4[2]、MD5[3]、SHA-1[4]、Davies-Meyer等。

3.1 字符串哈希算法

常用的字符串哈希算法有BKDRHash、APHash、DJBHash、JSHash、RSHash、SDBMHash、PJWHash、ELFHash等。BKDRHash算法是一种简单快捷的哈希算法,见文献[5]。APHash算法是由Arash Partow提出的一种哈希算法。DJBHash算法是由Daniel J.Bernstein教授提出的一种哈希算法。JSHash算法是由Justin Sobel提出的一种哈希算法。RSHash算法见Robert Sedgwicks的《Algorithms in C》一书。SDBMHash算法是由于在开源项目SDBM(一种简单的数据库引擎)中被应用而得名,它与BKDRHash思想一致,只是种子不同而已。PJWHash算法是基于AT&T贝尔实验室的Peter J.Weinberger的研究而提出的一种哈希算法。ELFHash算法是由于在Unix的Extended Library Function被附带而得名的一种哈希算法,它其实就是PJW Hash的变形,对长字符串和短字符串都很有效,字符串中每个字符都有同样的作用,能够比较均匀地把字符串分布在散列表中。

3.2 MD4算法

MD4算法是哈希算法中较为成熟的算法之一,它是Ronald L.Rivest教授在1990年设计的用于快速计算的哈希函数。MD是Message Digest的缩写,是指消息摘要的意思。MD4算法可以对任意的长度不超过2^{64}的消息进行处理,生成一个128 bit的哈希值。消息在处理前,首先要进行填充,保证Message Digest的填充后的bit位长度是512 bit的整数倍。填充结束后,利用迭代结构和压缩函数来顺序处理每个512 bit的消息分组。MD4本身存在安全性的问题,曾被破译,但是就整个MD4算法来说并没有完全被破译。在此之后Ronald L.Rivest对MD4中存在的漏洞进行了修补与改进。MD4算法为之后的MD5算法、SHA-1算法等提供了很好的理论基础。

3.3 MD5算法

MD5算法是MD4算法的升级版,也是由Ronald L.Rivest提出的一种哈希算法。它与MD4算法相比,安全性有了很大的提升,除了曾被发现假冲突,MD5的加密结果没有发现其他冲突。在MD5算法中原始消息的预处理操作和MD4是完全相同的,都需要进行补位、补长度操作,它们的信息摘要的大小都是128 bit。MD5在MD4的基础上加入了第四轮的计算模式,每一个步骤都是一一对应的固定值,改进了MD4中在第二轮、第三轮计算中的漏洞,完善了访问输入分组的次序,从而减小其对称性和相同性。通过这些变化,使得MD5与MD4相比变得复杂很多,整个运转速度也要比MD4慢一些,但是从整体安全性和抗冲突方面有了很大的提高[6]。

3.4 SHA-1算法

SHA-1算法是在MD5的基础上发展而来的,Secure Hash Algorithm(SHA),即安全哈希算法。SHA-1算法是1993年由美国国家技术标准研究所(NIST)开发的[7]。其主要功能是从输入长度不大于2^{64} bit的明文消息中得到长度大小为160 bit的摘要值。SHA-1算法通过计算明文信息得到固定长度的信息摘要,只要原始信息改变,摘要也随之发生改变,而且变化很大。这种发散性可以检测数据的完整性,因此,SHA-1算法在数字签名中有广泛的应用,主要适用于数字签名标准里面定义的数字签名算法。SHA-1的计算方式是基于MD4的算法原理,它的填补和分组模式与MD5算法是一样的,但在算法实现中SHA-1的非线性函数、循环左移运算和加法常数与MD5算法的运算方式有一定的差异,SHA-1的安全性和稳定性比MD5算法更加可靠,且运算速度也有了一定的提高。

3.5 Davies-Meyer算法

Davies-Meyer算法[8-10]是基于对称分组算法的单向散列算法。分组算法在设计和实现上花费很小的代价,

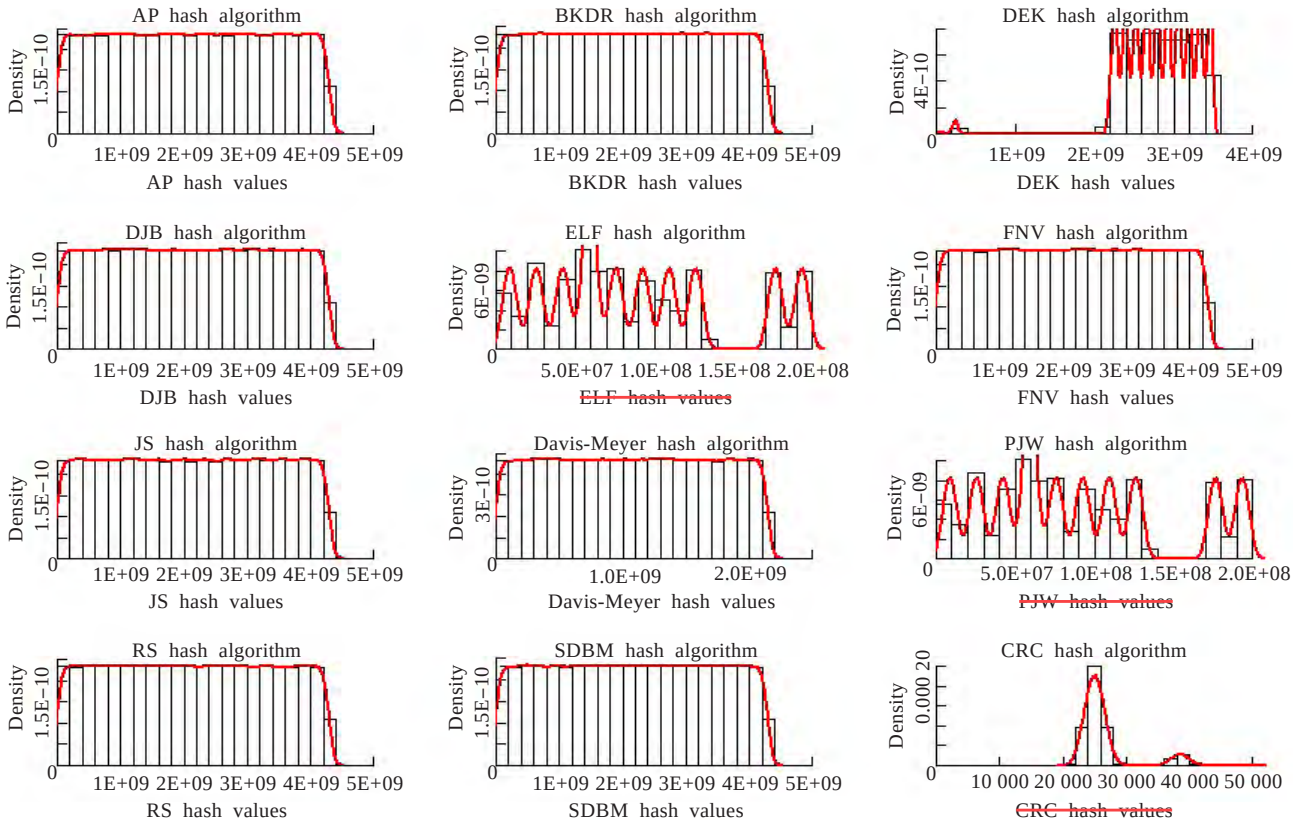

图1　各种哈希算法的散列分布性比较

可以构建一个基于该分组密码的哈希函数。但是用这种方法构造的哈希函数一般比专用的哈希函数速度慢,尤其是当使用的分组密码有较慢的密钥编排方案时。基于对称密码体制的加密算法DES和IDEA改进的Davies-Meyer算法,都具有很好的分布性,并且目前被认为是安全的算法。

4　哈希算法性能比较

根据前文对各种哈希算法的原理和特点分析,结合海量存储系统的应用,根据衡量哈希算法的标准,从软件模拟方面对多种哈希算法进行分析比较。文中根据Linux系统的命名规则,利用所有的字母、数字、“.”、“_”(下划线)和“-”(连字符),随机产生10万个长短不一的字符串作为输入,分别采用字符串哈希算法、MD4、MD5、SHA-1和Davies-Meyer算法统计哈希值的分布和冲突情况。

4.1　散列分布性

本文通过随机产生10万个文件名,软件模拟上文提到的哈希算法,将这些哈希算法的哈希值,利用直方图显示,比较这些哈希算法的散列分布性,如图1所示。图1中给出了12种哈希算法的散列值分布图,横坐标为哈希值的大小,纵坐标为哈希值的分布频率,即密度。从图1可知,大多数哈希算法对数据的散列分布性都很均匀,分布图表现为圆角梯形,而ELF哈希算法、PJW哈希算法和CRC算法效果最差,分布图表现为不规则图形。

为进一步对各种数据均匀分布的哈希算法进行比较,图2给出了各种哈希算法哈希值的曲线拟合,图中横坐标为哈希值,纵坐标为哈希值的分布频率。从图中可以看出,各种字符串哈希算法如AP算法、BKDR算法、RS算法等在数据散列分布性上性能差异不大,都能很好地保证均匀的数据分布,在图中表现为拟合曲线几乎重合在一起。并且,在保证数据均匀分布的同时,数据的散列区间也比较大,可以有效减小哈希冲突,而Davies-Meyer算法在数据的散列区间上就相对较差,这一特性会加大哈希值冲突的概率。

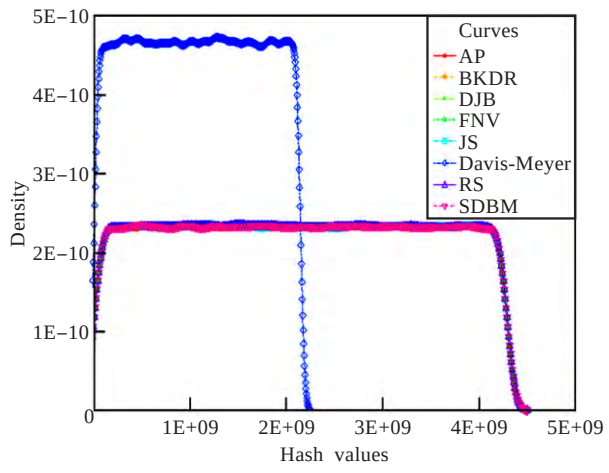

图2　各种哈希算法的散列分布曲线图

4.2 哈希冲突及运行效率

同样对随机产生的10万个文件名进行哈希计算，统计各种哈希算法的哈希冲突次数和运行时间，如表1所示。

表1 哈希算法的哈希冲突及运行时间统计

哈希函数	冲突数	运行时间/s
BKDRHash	2	392
APHash	3	529
DJBHash	2	421
JSHash	2	424
RSHash	2	431
SDBMHash	3	425
PJWHash	23	740
ELFHash	24	771
MD4	5 370	1
MD5	5 029	1
SHA-1	4 745	1
Davies-Meyer	4 660	<1

从表1的统计结果可知，各种字符串哈希算法在哈希冲突上表现出很大的优势，但是运行效率与MD4、MD5、SHA-1和Davies-Meyer算法相比效率很低。反之，MD4、MD5、SHA-1和Davies-Meyer算的运行效率很高，并且Davies-Meyer算法的运行效率最高，哈希冲突次数也相对好点，因此该算法是分布式存储系统中比较受推崇的算法之一。

4.3 哈希算法的选择

根据衡量哈希算法优劣的有三个指标：散列分布性、哈希冲突和计算效率，文中讨论的几种主流的哈希算法，各有千秋。字符串哈希算法（除ELF哈希算法和PJW哈希算法）和Davies-Meyer算法具有很好的散列分布性，并且各种字符串算法在哈希冲突上表现较好，而MD4、MD5、SHA-1和Davies-Meyer算法的哈希冲突性能较差；MD4、MD5、SHA-1算法在计算效率上有很大的优势，而字符串哈希算法在这方面有表现出一定的劣势。因此，Davies-Meyer算法和各种字符串哈希算法比较适合于对散列分布性要求高的场景；MD4、MD5、SHA-1及Davies-Meyer算法比较适合于对计算效率有很大需求的应用。在实际应用中根据应用的具体需求进行选择。比如，在分布式存储系统中对哈希函数的哈希冲突的要求不高，这样就需要从算法的运行效率考虑，选择运行效率最高的，则非常有利于数据的快速定位，如Davies-Meyer算法；而在文件搜索过滤中，对哈希冲突要求很高，而计算效率没有很高要求，则字符串哈希算法是比较好的选择。

5 哈希算法在分布式存储系统HepyCloud中的应用

分布式存储系统HepyCloud是中科院高能所自主开发的一套海量数据存储系统，该系统采用key-value技术，实现海量数据的快速存储、定位和高可扩展性，支持EB级存储。系统提出统一布局的思想，对一致性哈希算法[11-15]进行改进。系统中将存储设备抽象为区间管理节点，即将存储设备均匀分布到一个大小为2^m的环上($m=32$)，每个设备负责的区域空间为2^m/设备总数。假设8个节点，$N1$节点负责的区间为[0~536 870 912)，如图3所示。数据的读写通过对文件名进行哈希计算，得到一个整数，通过判断整数所在的区间，从而定位到具体的存储设备，完成数据的读写操作。因此，在数据的访问操作中，哈希算法的好坏，直接影响数据的分布和定位，从而影响系统的性能。

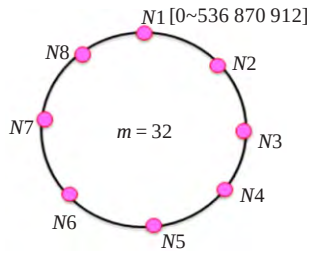

图3 一致性哈希环

HepyCloud系统采用改进的一致性哈希算法，实现数据的均匀分布和快速定位，在对哈希函数的选择时主要从以下两个方面考虑：(1)运行效率；(2)散列均匀。

运行效率指所选择的哈希函数有较高的计算效率，实现数据的快速定位，达到很好的用户体验；散列均匀指所选的哈希函数具有很好的分布性，保证数据在存储设备上的均匀分布。从前文对多种哈希算法的性能分析中，可知，Davies-Meyer算法是一种较好的选择。一方面高效的运行效率，保证了快速定位数据；另一方面均匀的散列分布性，确保了数据均匀分布。

从实际使用看，将改进的一致性哈希和Davies-Meyer算法应用到HepyCloud系统中，实现数据在存储设备上的均匀分布。目前，系统共有23个存储设备，存储容量186 TB，14 478 054个文件，每个设备上的文件数约为629 410（总文件数/设备数）。在数据定位方面，经测试和实际使用其表现与其他分布式文件系统相当，足以满足存储系统的性能要求。

6 结束语

针对如何实现数据在物理存储上的均匀分布和高效定位的问题，对多种哈希算法进行研究，提出了衡量分布式存储系统哈希算法优劣的标准，从散列分布性、哈希冲突和计算效率等多个维度对这些哈希算法进行分析比较。实验证明，各种字符串哈希算法（除ELF哈希算法和PJW哈希算法）和Davies-Meyer算法具有很好的散列分布性，采用这些算法能够有效保证海量数据在物理存储上的均匀分布；MD4、MD5、SHA-1算法在计算效率上有很大的优势，而字符串哈希算法在这方面有

（下转77页）

参考文献:

[1] López C.Watermarking of digital geospatial datasets:a review of technical,legal and copyright issues[J].International Journal of Geographical Information Science,2002,16(6):589-607.

[2] Kang H.A map data watermarking using the generalized square mask[C]//Proc of the International Conference on Information Technology:Coding and Computing,Las Vegas,NV,USA,2001:234-236.

[3] Schulz G,Voigt M.A high capacity watermarking system for digital maps[C]//Proceedings of the 2004 Workshop on Multimedia and Security.[S.l.]:ACM,2004:180-186.

[4] Sonnet H,Isenberg T,Dittmann J,et al.Illustration watermarks for vector graphics[C]//Proceedings of the 11th Pacific Conference on Computer Graphics and Applications.[S.l.]:IEEE,2003:73-82.

[5] 车森,邓术军.基于双重网格的矢量地图数字水印算法[J].海洋测绘,2008,28(1):13-17.

[6] 杨成松,朱长青,陶大欣.基于坐标映射的矢量地理数据全盲水印算法[J].中国图象图形学报,2010,15(4):684-688.

[7] 陈晓光,李岩.针对二维矢量图形数据的盲水印算法[J].计算机应用,2011,31(8):2174-2177.

[8] 闵连权.一种鲁棒的矢量地图数据的数字水印[J].测绘学报,2008,37(2):262-267.

[9] Shao Chengyong,Wang Xiaotong,Jin Liangan,et al.A robust algorithm for watermarking 2D vector maps with low shape-distortions[J].Journal of China Ordnance,2006,2(3).

[10] 杨成松,朱长青.基于常函数的抗几何变换的矢量地理数据水印算法[J].测绘学报,2011,40(2):256-262.

[11] 许德合,朱长青,王奇胜.利用QIM的DFT矢量空间数据盲水印模型[J].武汉大学学报:信息科学版,2010,35(9):1100-1103.

[12] 王奇胜,朱长青,许德合.利用DFT相位的矢量地理空间数据水印方法[J].武汉大学学报:信息科学版,2011,36(5):523-526.

[13] 李媛媛,许录平.矢量图形中基于小波变换的盲水印算法[J].光子学报,2004,33(1):97-100.

[14] 杨成松,朱长青.基于小波变换的矢量地理空间数据数字水印算法[J].测绘科学技术学报,2007,24(1):37-39.

[15] 范铁生,孟瑶,房肖冰.基于B-spline矢量图形数字水印方法[J].计算机工程与应用,2007,43(17):69-70.

(上接4页)

一定的劣势,不利于数据的快速定位。海量数据存储系统采用哈希算法实现均匀的数据分布和高效的数据定位,这就要求哈希算法既有很好的分布性,又要满足高效的计算速度。实验结果证明,Davies-Meyer算法很适合分布式存储系统的应用。最后,结合分布式存储系统HepyCloud的应用,进一步验证在分布式存储系统中Davies-Meyer算法是一种较好的选择。一方面其高效的运行效率,保证了快速定位数据;另一方面其均匀的散列分布性,确保了数据均匀分布。

参考文献:

[1] 董继光,陈卫卫,田浪军,等.大规模云存储系统副本布局研究[J].计算机应用,2012,32(3):620-624.

[2] 黎琳.MD4算法分析[J].山东大学学报,2007,42(4).

[3] Rivest R.RFC1321 The MD5 message-digest algorithm[S].1992.

[4] Eastlake D,Jones P.RFC 3174 US secure hash algorithm1(SHA1)[S].2001.

[5] Kernighan B W,Ritchie D M.The C programming language[M].北京:机械工业出版社,2004.

[6] 王小云,张金清.MD5 报文摘要算法的各圈函数碰撞分析[J].计算机工程与科学,1996,18(2):15-22.

[7] 张邵兰.几类密码哈希函数的设计和安全性分析[D].北京:北京邮电大学,2011.

[8] Wintemitz R.A secure one-way hash function built from DES[C]//Proceedings of the IEEE Symposium on Information Security and Privacy.[S.l.]:IEEE Press,1984:88-90.

[9] 余秦勇,陈林,童斌.一种无中心的云存储架构分析[J].通信技术,2012,45(8):123-127.

[10] 李正.杂凑函数结构研究现状及新的结构设计[D].济南:山东大学,2010.

[11] 周敬利,周正达.改进的云存储系统数据分布策略[J].计算机应用,2012,32(2):309-312.

[12] 杨彧剑,林波.分布式存储系统中一致性哈希算法的研究[J].电脑知识与技术,2011(8):5295-5296.

[13] 沈琦.基于Chord的高性能文件存储技术的研究与设计[D].杭州:浙江大学,2007.

[14] Devine R.Design and implementation of DDH:a distributed dynamic hashing algorithm[C]//Proceedings of 4th International Conference on Foundations of Data Organizations and Algorithms,1993.

[15] Lewin D.Consistent hashing and random trees:algorithms for caching in distributed networks[D].Cambridge,Massachusetts:Massachusetts Institute of Technology,1998.