

基于 ID 特征码的云存储数据分片与备份策略

钱进进, 凌 捷

(广东工业大学 计算机学院, 广东 广州 510006)

摘 要: 提出了一种基于 ID 特征码的云存储数据分片与备份策略, 在查找存储节点时利用文件块 ID 和可用存储节点 ID 的对应关系与随机分配有效存储节点相结合的方法, 提高了主服务器的效率; 将需要备份的数据文件, 在多个云节点上按指定数量建立数据副本进行分片并选择最优的备份区域存储, 减小了建立数据副本的开销, 提高了数据的可靠性和可用性。

关键词: 云存储; ID 特征码; 分片策略; 备份策略

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2013)08-0016-03

Data Slice and Backup Strategy of Cloud Storage Based on ID Signature

QIAN Jin-jin, LING Jie

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: This paper puts forward a cloud storage data backup strategy based on ID signature. By using the correspondence relationship between file block ID and its available storage node ID, and the randomly assigned effective storage node to find storage nodes, the strategy improved the efficiency of a primary server. Besides, creating a copy of backup data file at a specified number of fragments and storing data in the optimal backup area in multiple clouds node can not only reduce the cost but also improve its data reliability and availability.

Key words: cloud storage; ID signature; partitioning strategy; backup strategy

1 引言

云存储系统通过集群应用, 使存储系统中的各个存储节点协同工作, 为用户提供透明的存储服务^[1-2]. 为提高系统处理数据的能力, 实现服务器的负载均衡, 一般采用并行传输和文件分块的方式, 将需存储的大量数据保存到不同的存储节点上. 多数云存储系统中的数据是以文件为存储单位, 容易造成独立节点负载过大, 而且在客户端请求存储文件时, 需要文件存储服务器节点与客户端一对一传输, 文件传输效率低^[3-4]. 文献[5]中提出了分片和备份策略, 并通过取模或哈希再散列的方式分配存储节点, 这样能实现服务器的负载均衡, 但每一个子块的存储节点都需要主服务器进行计算再分配, 这样大

大地增加了主服务器 CPU 的负荷, 降低了系统效率, 并且其备份策略只能确保在备份 n 份的情况下最多能出现 $n-1$ 个节点失效. 文献[6]中提出的备份策略是每个子块分配到一个节点上时, 该节点把此子块发送给其他所有的节点, 也明显地增加了服务器的负荷, 且存在大量的冗余数据. 本文提出一种基于 ID 特征码的数据分片和备份策略, 提高了系统的可靠性和可用性.

2 ID 特征码的设计

考虑到实际应用中大多数单位有多个部门, 每个部门有多个用户, 每个用户又有多个文件, 每个文件又对应多个文件数据块的数据复杂繁多情况, 本文设计一种编写 ID 特征码的方法, 形成文件数据块

收稿日期: 2012-11-30; **收回日期:** 2013-01-12

基金项目: 国家“十二五”科技支撑计划项目(2012BAH27F05); 广东省战略性新兴产业核心技术攻关项目(2012A010701004); 广东省自然科学基金重点项目(S2012020011071)

与存储节点之间的关联关系. 特征码的编码方法: 每个部门对应一个区, 部门 ID 对应区 ID, 每个部门的用户在注册时都会生成一个用户 ID (设为 a_b), a 是部门 ID, b 是用户的注册编号, 此编号是根据注册的时间先后进行自动加 1, 第一位用户的注册编号为 1. 每个用户的文件 ID (设为 a_b_c) 的 a_b 是用户 ID, c 是文件的编号, 此编号是根据用户上传的时间先后进行自动加 1, 第一个上传的文件的编号为 0. 每个文件分块之后形成多个块, 每个块都有一个 ID (设为 $a_b_c_d$), a_b_c 文件 ID, d 为此分块的编号, 文件分块之后的第一个分块的编号为 0, 分块编号依次加 1, 每个编号之间用“_”隔开. 例如公司的技术部 ID 为 i , 技术部的第 j 个用户的第 $k+1$ 个文件分成 p 块, 则每个分块 ID 分别为 $i_j_k_0, i_j_k_1 \dots i_j_k_p-1$, 如图 1 所示.

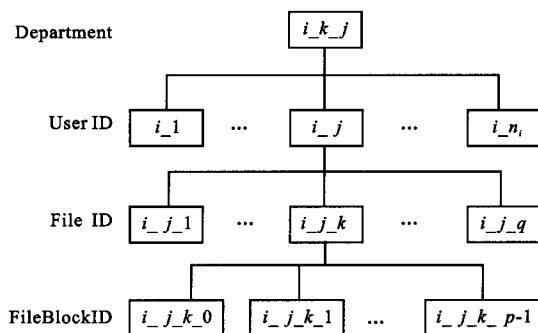


图 1 ID 特征码示例

3 云存储数据分片策略

当客户端发送请求要上传一个文件到云端时, 监听服务器节点先监听每个存储区的可用存储节点, 生成一张索引表并反馈给主服务器节点, 例如某单位有 n 个部门, 且所有存储节点都是可用的, 则监听服务器节点监听之后反馈给主服务器节点的索引表如表 1 所示.

表 1 可用存储节点索引表

区 ID	1	...	i	...	n
存储节点 ID	$i_1 i_2 \dots i_{m_1}$	...	$i_1 i_2 \dots i_{m_i}$	...	$n_1 n_2 \dots n_{m_n}$
可用性	1 1 ... 1	...	1 1 ... 1	...	1 1 ... 1

当可用性值全为 1 时, 则所有的存储节点都是可用的, 当有些存储节点的可用性值为 0 时则此存储节点不可用. 主服务器先据文件信息对文件进行切块计算, 并根据文件 ID 信息给每个子块分配一个 ID, 再根据监听服务器反馈回来的索引表, 找到各个分块相应的存储节点, 将文件信息、块信息和相关存储节点信息写入数据库, 然后服务器将文件块信息 (块数、块大小、块 ID 等) 及对应的存储节点地址

返回给客户端, 当客户端根据返回的信息后将文件分块、并行上传到对应的存储节点, 存储节点每接收到一个文件块后都会向主服务器发送一条确认信息, 主服务器将数据库中该文件的状态信息改为 1, 表示已经上传成功, 当 i 区的第 j 个用户 (用户编号为 i_j) 要上传一个文件时, i 区的可用存储服务器节点数为 m_i (设此区的所有存储节点都是可用的), 则此存储区的存储服务器节点的编号从 i_1 到 i_m_i , 假设 i 区的第 j 个用户 (编号为 i_j) 的第 $k+1$ 个文件 (编号为 i_j_k) 的切分为 p 块, 则此文件的每个分块的编号依次为 $i_j_k_0, i_j_k_1 \dots i_j_k_p-1$, 则此文件第一块存储在节点编号为 $i_j \bmod m_i$ 的节点上, 第二块存储在节点编号为 $i_j+1 \bmod m_i$ 的节点上, 第 p 块则存储在节点编号为 $i_j + ((p+j-2) \bmod m_i) + 1$ 的节点上, 如图 2 所示.

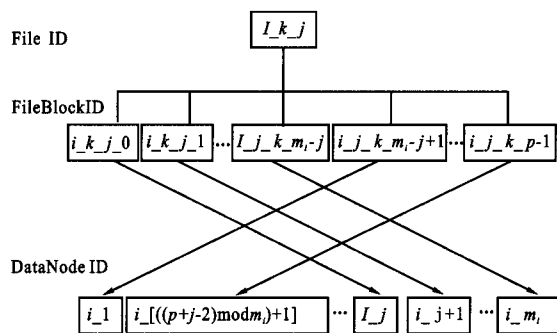


图 2 文件块 ID 对存储节点 ID 的对应关系

当该区域中存在失效节点时, 根据索引表记录可用性值为 0 的存储节点 ID, 当有文件的子块根据计算需要存储在失效节点上时, 可以把这些子块可以随机存储在该区有效节点上. 分片策略中的分块大小可以在程序中任意设定, 本文以 1M 为单位对文件进行分块, 最后不足 1M 的也作为一块, 在用户下载一个文件时, 发送请求给主服务器, 主服务器根据文件 ID 从数据库中找到该文件的块信息和相应的存储节点的地址, 然后从各个存储节点并行将文件块下载到用户计算机, 最后在用户计算机上将文件块整合成一个完整的文件.

4 云存储数据备份策略

一般的云存储数据备份策略中, 数据资源通常是以完整的数据文件为单位在节点间进行拷贝, 其研究重点在于副本数目的设置、副本的数据一致性维护等, 而对大规模副本复制时会引起大量的时间开销、带宽开销且对负载均衡问题考虑得较少. 基于分片的数据备份是通过创建同一个数据文件块的多个副本, 存放在不同的存储节点来提高数据资源的

可用性、系统的可靠性和数据资源的使用效率,可以实现负载均衡和降低访问延迟。

本文提出的云存储数据备份策略基于分片策略,当客户端发出请求给主服务器后,主服务器根据文件块信息和监控节点反馈的索引表信息,副本数可以在程序中任意设定,这里选择副本数为 3,分别选取三个区,每个区存储一份,当 i 区的第 j 个用户(用户编号为 i_j)要上传一个 ID 为 i_j_k 的文件时,主服务器先根据文件信息进行切块计算,文件块 ID 分别为 $i_j_k_0, i_j_k_1 \dots i_j_k_p-1$,第一个副本存储在 ID 为 $((i-1) \bmod n)+1$ 的区,再根据索引表信息查找存储节点失效率最低的两个区(即最优的两个备份区域),两个副本存储的区 ID 设为 $((f_1-1) \bmod n)+1, ((f_2-1) \bmod n)+1$,最后根据文件块 ID 和索引表中对应区的可用存储节点 ID 选取文件块对应的存储节点. 设此三个区的可用存储节点数分别为 $m_{((i-1) \bmod n)+1}, m_{((f_1-1) \bmod n)+1}$ 和 $m_{((f_2-1) \bmod n)+1}$,且所有节点都是可用的,则 $((i-1) \bmod n)+1$ 区第 p 块存储在节点 ID 为 $((i-1) \bmod n)+1_[(p+j-2) \bmod m_{((i-1) \bmod n)+1}+1]$ 的存储节点上, $((f_1-1) \bmod n)+1$ 区第 p 块存储在节点 ID 为 $((i \bmod n)+1_[(p+j-2) \bmod m_{((f_1-1) \bmod n)+1}+1])$ 的存储节点上, $((f_2-1) \bmod n)+1$ 区第 p 块存储在节点 ID 为 $((i+1) \bmod n)+1_[(p+j-2) \bmod m_{((f_2-1) \bmod n)+1}+1]$ 的存储节点上,当每个区中存在失效节点时,各个文件子块的存储方法和分片策略中的存储方法类似,再把三个区的相应存储节点地址和文件子块信息返回给客户端,并将相关信息写入数据库. 客户端收到返回信息后,即将文件分块、并行上传到对应的存储节点,由于每个区都存有一个副本,每个块要传到三个区的相应存储节点,当用户下载文件时,发送请求给主服务器,主服务器根据监控节点服务器反馈回来的索引表和数据库表中的文件信息,首先选取本区的可用存储节点,若本区中有失效节点并且此文件的文件块存储在此节点上,则查找数据库中该文件块信息,找到其他两个区的此文件块的可用存储节点,并发送信息给存储节点服务器,不同区的存储节点服务器并行工作,且根据索引表信息选择最优的备份存储区域,这样实现了存储服务器的负载均衡,提高了数据可用性与传输效率。

5 实验结果分析

本文实验环境选取主服务器 1 台、数据存储节

点 12 台,将存储节点分为三个区,分别存放 3、4 和 5 台存储节点服务器,其中主服务器(元数据服务器)配置环境为操作系统 Windows Server 2003、CPU 为 Intel Core2 2.83 GHz、内存 4 GB RAM,数据库为 MySQL. 数据存储节点配置环境为操作系统 Windows Server 2003、CPU 为 Intel Core i3 3.3 GHz、内存 2 GB RAM、硬盘 320 GB.

实验中对几个不同大小的数据文件在上传过程进行了仿真,比较了本文的方法与一般的方法在完成整个上传过程所花费的时间开销和能承受失效的节点数的最大值,仿真结果如图 3、图 4 所示。

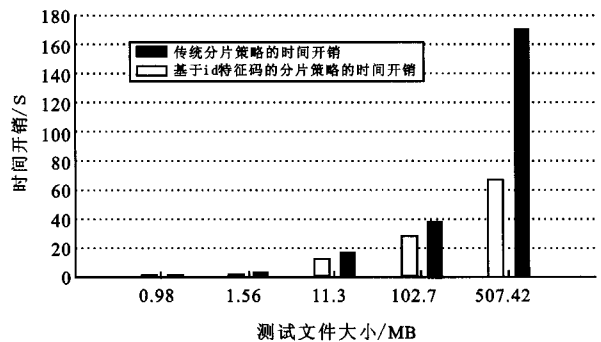


图3 不同大小文件分片存储的时间开销

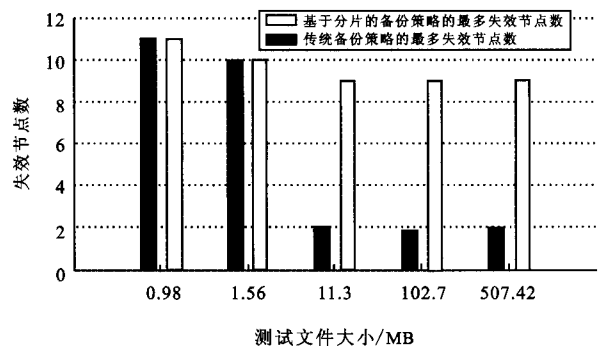


图4 不同大小的文件分片备份后能承受失效的节点数

6 结束语

针对当前云存储系统中分片备份策略存在的问题,本文提出了一种基于 ID 特征码的云存储分片和备份策略,在文件上传云端过程中利用了文件块 ID 与可用存储节点 ID 的对应关系,若存储区中存在失效节点时再采取随机选取存储节点,与一般的方法相比,降低了查找存储节点的时间开销,在备份时选择最优的备份存储区并采用异区域备份,每个备份都存储在不同的区,即使其中任何两个区的所有服务器出现宕机的情况,文件块在系统中都还至少存储一份备份,当各个部门的上传和下载文件块数相近时,服务器能达到最好的负载均衡状态,提高了云存储数据的可靠性和可用性。(下转第 22 页)

表 1 不同代数下两种算法检测率 %

代数	原克隆选择算法	本文算法
200	32	35
400	51	53
600	65	68
800	72	77
1 000	76	83

从表 1 可以看出,随着代数的增加,本文提出算法的检测率也随之提高.另外,和原克隆选择算法相比,本文提出算法的检测率有所提高.

对异常抗原类型的检测执行如下步骤:找出测试数据向量和检测器集中亲和力小于阈值 Y 所有检测器,排序后选取最小的 K 个,通过检查各检测器的类型得出该数据的类型.其中 $Y=0.06$ $K=3$.

代数为 5 000 时,异常数据类型检测效果如表 2 所示.

表 2 各类型异常数据检测率

	NORMAL	DOS	R2L	U2R	PROBE
检测率/%	96.30	98.60	96.55	86.38	97.92

从表 2 看出:DOS 类型异常数据检测率最高,U2R 类型检测率最低.各个类型的检测率都达到了较好的效果,因此本文提出的算法可以很好地识别异常数据类型.

5 结束语

本文提出在基于人工免疫的入侵检测系统中,通过对成熟检测器进行标志聚类学习,使检测器具具备非己抗原的类型特征,同时对成熟检测器集中的过度聚类问题进行了解决,优化了检测器集,提高了检测器对非己空间的覆盖率.实验表明达到了较好

的检测率同时能够对异常数据的类型有很好识别.

参考文献:

- [1] Dasgupta D, Gonzalez F. An immunity-based technique to characterize intrusions in computer networks [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(3): 281—291.
- [2] Stephanie F, Steven A, Anil S. Computer innunology [J]. Communications of the ACM, 1997, 40(10):88—96.
- [3] Kim J, Bentley P J. Towards an artificial immune systemfor network intrusion detection;an investigation of dynamic clonal selection[C]//Evolution Computation CEC 2002. Hawaii,2002:1015—1020.
- [4] 符海东,王枫.基于数理统计及人工免疫的入侵检测模型[J].计算机工程与设计,2008,29(12):3037—3039.
- [5] 邓荣.基于人工免疫的网络入侵检测模型[J].煤炭技术,2011,30(4):105—107.
- [6] Forrest S, Perelson A, Cherukuri R. Self-nonself discrimination in acomputer[C]// Proc. of 1994 IEEE Computer Society Symposium on Research in Security and Privacy. Los Almitos, CA, USA: IEEE Computer Society, 1994: 202—212.
- [7] 姜懿庭.基于人工免疫系统的检测器生成算法改进[J].云南民族大学学报:自然科学版,2012,21(2):141—144.
- [8] 肖军弼,黄波涛.基于改进基因库的检测器生成算法[J].计算机系统应用,2011,20(4):251—253.

作者简介:

池 静 女,(1973—),硕士,副教授.研究方向为数据挖掘.
杨振宇 男,(1988—),硕士研究生.研究方向为网络安全.
张 婷 女,(1986—),硕士研究生.研究方向为网络安全.

(上接第 18 页)

参考文献:

- [1] 刘鹏.云计算[M].北京:电子工业出版社,2010
- [2] 王鹏.云计算的关键技术与应用实例[M].北京:人民邮电出版社,2010
- [3] Tom White,周敏,曾大聘,等. Hadoop 权威指南[M].北京:清华大学出版社,2011
- [4] Zhang Hong, Lu Song feng, Zhao You qiao, et al. Research on file—slice model for data security storage [J]. Computer Engineering and Application, 2012, 48 (18): 66—70.

- [5] 王鹏.走进云计算[M].北京:人民邮电出版社,2009: 221—224.
- [6] Huang Xiaotao, Li Zhiyong. Replication strategy based on file—partition for P2P network [J]. Application Research of Computers,2008, 25(1): 171—173.

作者简介:

钱进进 女,(1988—),硕士研究生.研究方向为网络与信息安全.
凌 捷 男,(1964—),博士.研究方向为网络与信息安全.