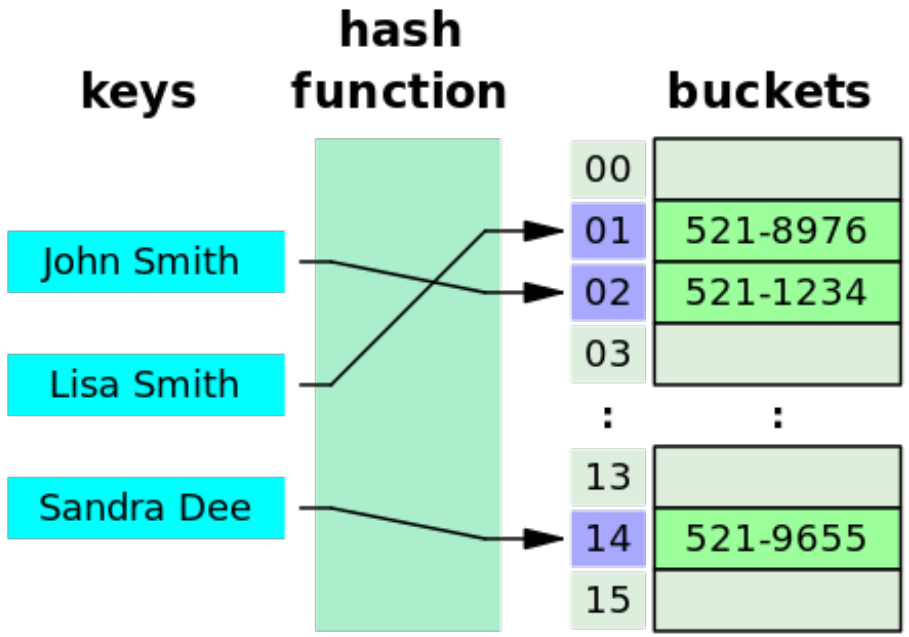
Hashing的应用场景和经典故事

陈巍
161180012
2018.5.28

散列表

哈希表（Hash table，也叫散列表），是根据关键码值而直接进行访问的数据结构。也就是说，它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。这个映射函数叫做散列函数，存放记录的数组叫做散列表。Hash Table的查询速度非常的快，几乎是O(1)的时间复杂度。



散列函数

- 除法散列法

  通过取k除以m的余数，来将关键字k映射到m个槽的某一个里面去，即h(k)=k mod m。

- 乘法散列法

  用关键字k乘上常数A(0<A<1),并抽取出kA的小数部分，然后用m乘以这个值，再取结果的底，即h(k)= $\lfloor m(kA \bmod 1) \rfloor$

- 斐波那契法
- 等等

解决碰撞

哈希表中，一个关键字可能可能对应了多个值，这就造成了冲突。
在开放寻址法中我们可以通过下面几种方法解决。

- 线性探查

  $h(k,i)=(h'(k)+i) \bmod m, i=0,1,\ldots,m-1$

  给定关键字k，先探查的槽是T[h'(k)]，然后是T[h'(k)+1],直至T[m-1]，线性探查实现较为容易，但随着时间推移，平均查找时间会不断增加。

- 二次探查

  $h(k,i)=(h'(k)+c1i+c2i^2) \bmod m$

  初始的探查位置为T[h'(k)]，后续的探查位置要在此基础上加一个偏移量。

- 双重散列

  $h(k,i)=(h1(k)+ih2(k)) \bmod m$

  与前面两个探查不一样的是，这里的探查序列以两种方式依赖于关键字k。

- 建立公共溢出区

广泛使用的哈希算法

（1） MD4

MD4(RFC 1320)是 MIT 的 Ronald L. Rivest 在 1990 年设计的，MD 是 Message Digest 的缩写。它适用在32位字长的处理器上用高速软件实现--它是基于 32 位操作数的位操作来实现的。

（2） MD5

MD5(RFC 1321)是 Rivest 于1991年对MD4的改进版本。它对输入仍以512位分组，其输出是4个32位字的级联，与 MD4 相同。MD5比MD4来得复杂，并且速度较之要慢一点，但更安全，在抗分析和抗差分方面表现更好。

（3）SHA-1及其他

SHA1是由NIST NSA设计为同DSA一起使用的，它对长度小于264的输入，产生长度为160bit的散列值，因此抗穷举(brute-force)性更好。SHA-1 设计时基于和MD4相同原理,并且模仿了该算法。

哈希算法的应用

（1）文件校验

我们比较熟悉的校验算法有奇偶校验和CRC校验，这2种校验并没有抗数据篡改的能力，它们一定程度上能检测并纠正数据传输中的信道误码，但却不能防止对数据的恶意破坏。

MD5 Hash算法的"数字指纹"特性，使它成为目前应用最广泛的一种文件完整性校验和Checksum算法，不少Unix系统有提供计算md5 checksum的命令。

（2） 数字签名

Hash 算法也是现代密码体系中的一个重要组成部分。由于非对称算法的运算速度较慢，所以在数字签名协议中，单向散列函数扮演了一个重要的角色。 对 Hash 值，又称"数字摘要"进行数字签名，在统计上可以认为与对文件本身进行数字签名是等效的。

（3）加密算法

哈希算法也是非常常见的加密算法之一。他和对称算法以及非对称算法最大的区别是，它不是用来做数据传输，而是对数据是否被篡改加以验证，防止不法分子篡改数据。它的特点是无论原文多长都会变成固定长度的字符串，只能加密不能解密（只能单向运算）。对于不同的输入，理论上会生成不同的输出（部分算法已出现大规模碰撞，碰撞就是指不同明文相同密文的情况）。

哈希算法的历史

哈希的想法在不同的地方独立出现。
1953年1月，Hans Peter Luhn撰写了一份IBM内部备忘录，首先提出了散列表这一技术，以及用于解决碰撞的链接方法。
差不多同一时间， Gene Amdahl首先提出了开放寻址的想法。
Carter和Wegman于1979年引入了全域散列函数类的概念。
Fredman、Komlos和Szemeredi针对静态关键字集合，提出了完全散列方案。
Dietzfelbinger等人后来又将这一方法扩展至动态关键字集合上，其处理插入和删除操作的平均期望时间为O(1)。

MD5

MD5即Message-Digest Algorithm 5，用于确保信息传输完整一致。是计算机广泛使用的哈希算法之一。将数据（如汉字）运算为另一固定长度值，是哈希算法的基础原理，MD5的前身有MD2、MD3和MD4。

MD5算法具有以下特点：

1、压缩性：任意长度的数据，算出的MD5值长度都是固定的。

2、容易计算：从原数据计算出MD5值很容易。

3、抗修改性：对原数据进行任何改动，哪怕只修改1个字节，所得到的MD5值都有很大区别。

4、强抗碰撞：已知原数据和其MD5值，想找到一个具有相同MD5值的数据（即伪造数据）是非常困难的。

MD5的作用是让大容量信息在用数字签名软件签署私人密钥前被"压缩"成一种保密的格式（就是把一个任意长度的字节串变换成一定长的十六进制数字串）。

2004年8月17日在美国加州圣巴巴拉举行了一次国际密码学学术年会（Crypto’2004）,当晚来自中国山东大学的王小云教授做了关于破译MD5、HAVAL-128、 MD4和RIPEMD算法的报告。

MD5发展历史

MD2

Rivest在1989年开发出MD2算法。在这个算法中，首先对信息进行数据补位，使信息的字节长度是16的倍数。然后，以一个16位的检验和追加到信息末尾，并且根据这个新产生的信息计算出散列值。后来，Rogier和Chauvaud发现如果忽略了检验将和MD2产生冲突。MD2算法加密后结果是唯一的（即不同信息加密后的结果不同）。

MD4

为了加强算法的安全性，Rivest在1990年又开发出MD4算法。MD4算法同样需要填补信息以确保信息的比特位长度减去448后能被512整除。然后，一个以64位二进制表示的信息的最初长度被添加进来。信息被处理成512位迭代结构的区块，而且每个区块要通过三个不同步骤的处理。Den boer和Bosselaers以及其他人很快的发现了攻击MD4版本中第一步和第三步的漏洞。Dobbertin向大家演示了如何利用一部普通的个人电脑在几分钟内找到MD4完整版本中的冲突。毫无疑问，MD4就此被淘汰掉了。尽管MD4算法在安全上有个这么大的漏洞，但它对在其后才被开发出来的好几种信息安全加密算法的出现却有着不可忽视的引导作用。

MD5

1991年，Rivest开发出技术上更为趋近成熟的MD5算法。它在MD4的基础上增加了"安全-带子"的概念。虽然MD5比MD4复杂度大一些，但却更为安全。这个算法很明显的由四个和MD4设计有少许不同的步骤组成。在MD5算法中，信息-摘要的大小和填充的必要条件与MD4完全相同。2004年8月17日在美国加州圣巴巴拉举行了一次国际密码学学术年会,当晚来自中国山东大学的王小云教授做了关于破译 MD5、HAVAL-128、 MD4和RIPEMD算法的报告，公布了MD系列算法的破解结果。宣告了固若金汤的世界通行密码标准MD5的堡垒轰然倒塌，引发了密码学界的轩然大波。破解MD5之后，2005年2月，王小云教授又破解了另一国际密码SHA－1。