位图法

x>>5 表示x除以32得到的商

1<<(x & 0X1F)) 表示x除以32得到的余数

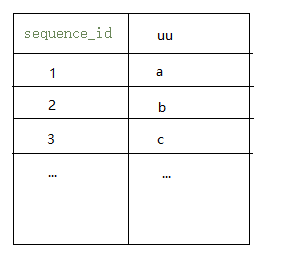
1、原理

位图是通过将数组下标与应用中的一些值关联映射，数组中该下标所指定的位置上的元素可以用来标识应用中值的情况（是否存在或者数目 或者计数等），位图数组中每个元素在内存中占用1位，所以可以节省存储空间。位图是一种非常简洁快速的数据结构，它能同时使存储空间和速度最优化。如可用一个10位长的字符串来表示一个所有元素都小于10的简单的非负整数集合，例如，可以用如下字符串表示集合{1,2,4,5,8} ，对应位置数字存在标记为1，否则标记为0。

2、位图实现

思路：

1.使用uu和递增的序列建立一个映射关系，把这个关系表保存到hbase，记为表ald\_user\_id\_map；



2.我们使用一个数组Array[Int]来表示位图，以此表示每天用户的访问情况

例如：bitmap=Array(1,1,…)

我们知道一个Int可以表示成32位，bitmap(0)=1表示(对应二进制)

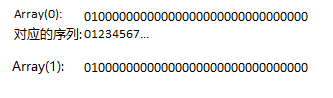
01000000000000000000000000000000

所以bitmap=Array(1,1,…)还原成位图，即

01000000000000000000000000000000

01000000000000000000000000000000

…



根据表ald\_user\_id\_map来解读这个位图的含义就是，uu为1和33的用户来访问过

3、具体代码实现

有两个scala文件SaveUU2Hbase.scala和SaveBitArr.scala

1. SaveUU2Hbase

保存uu和递增序列到ald\_user\_id\_map，hbase表结构设计如下

Rowkey：ak+uu

列族1：ak，列：ak

列族2：user\_id，列1：user\_id，列2：sequence\_id

列族3：ifo，列：day

列族4：time，列：update

这里还用到了zookeeper做增量，记录最后一条的插入的sequence\_id

2. SaveBitArr

这个的作用是用来保存每天的位图