# HashMap&Bitmap

1.Hash扩容算法在多线程情况有什么问题？

2.如何在3亿个整数（0~2亿）中判断某一个数是否存在？内存限制500M，一台机器。

分治：

布隆过滤器

Bit：bitMap，位图；

计算机最小的单位：bit

1Btye=8bit

Integer 4Byte

Long 8Byte

Float 4Byte

Double 8Byte

Char 2Byte

扩容算法在多线程时的问题：

1.多线程put操作，get会死循环()。这个确实可以优化掉，比如扩容的时候我们新开一个数组，不要使用共享的那个数组

2.多线程put可能会导致get取值错误

为什么会出现死循环呢？

上节课我们讲到hash冲突时我们会采用了链式结构来保存冲突的值。如果我们在遍历这个链表时本身是这样的1->2->3->null

如果我们遍历到3本身应该是null的，这时候刚好有人把这个null给计算出了值，null => 1->3，这下就完了 原来的3本来是要指向null结束的 这下又变成指向1，而这个1又刚好指向3 ，这样就一直循环下去了

既然Hash扩容是线程非安全的，那我们该怎么用？

1.使用了hash的系统尽可能的单线程操作

2.如果是多线程环境则要注意加锁。比如jdk里面就可以使用ConCurrentHashMap

Jdk1.7：分段锁

类型基础：

计算中最小的内存单位是bit，只可以表示0，1

1Byte = 8bit

1int = 4byte 32bit

Float = 4byte 32bit

Long=8byte 64bit

Char 2byte 16bit

Int a = 1,这个1在计算中是怎么存储的？

0000 0000 0000 0000 0000 0000 0000 0001 toBinaryString (1) =1

2 << 1 = 2\*2

2 << 2 = 2 \* 4

运算符基础：

左移 << ： 8 << 2 = > 8\*4=32

8: 0000 0000 0000 0000 0000 0000 0000 1000

<<2: 0000 0000 0000 0000 0000 0000 0010 0000 => 2^5= 2\*2\*2\*2\*2=32

右移 >>:8 >> : 8 / 4 = 2

8: 0000 0000 0000 0000 0000 0000 0000 1000

<<2: 0000 0000 0000 0000 0000 0000 0000 0010 => 2^1=2

8 / 4 => 8 >> 2

8\*4 => 8 << 2

位与 &:同位上的两个数都是1则位1，否则为0

位或 |:同为上的两个数只要有一个为1 则为1，否则为0

通过以上知识我们可以知道 一个int占32个bit位。假如我们用这个32个bit位的每一位的值来表示一个数的话是不是就可以表示32个数字，也就是说32个数字只需要一个int所占的空间大小就可以了，瞬间就可以缩小空间32倍。

比如假设我们有N{2，3，64}个数中最大的是MAX，那么我们只需要开int[MAX /32+1]个int数组就可以存储完这些数据，具体可以看以下结构：

Int a : 0000 0000 0000 0000 0000 0000 0000 0000 这里是32个位置，我们可以利用每个位置的0或者1来表示该位置的数是否存在,这样我们就可以得到以下存储结构：具体我们可以画个图来展示

Data[0]:0~31 32位

Data[1]:32~63 32位

Data[2]:64~95 32位；

Data[MAX /32+1]

假设我们要判断100是否在列表中，那么我们就可以这样来计算：

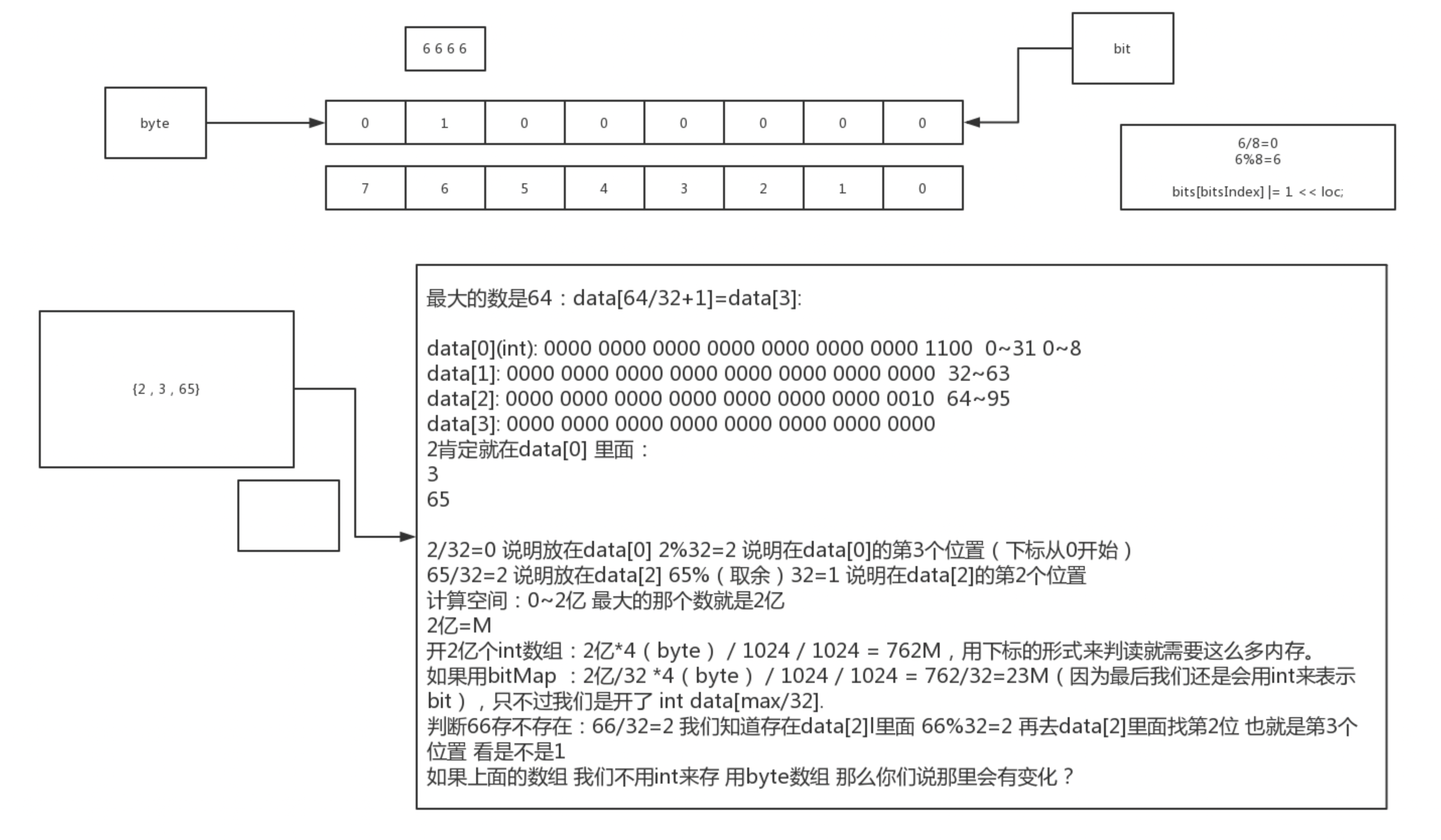
65/32=2 =>定位到data[2],65%32=1 定位到data[2]的第2位(注意这里从右边开始数)。我们再来看data[2]的第二位是否是1，如果是1则列表中存在65，否则不存在。

这就是bitMap算法的核心思想。

假设序列:2 3 65

我们开一个byte数组，或者int数组都可以。Byte数组的话 那就需要2个，因为byte为8bit，int的话只需要1个就行。

我们以byte数组为例。



BitMap算法分析：

时间复杂度：

空间复杂度：

能解决的问题：

1.数据判重

2.对没有重复的数据排序，既然处理不了重复的数据 那么也处理不了Hash冲突，假如我们只有10个数（0~10亿）如果用bitmap你还是要开10亿/32个空间，我们直接用hashMap 或者一个10个空间的数组是不是就更好了。

3.根据1和2可以扩展出很多其他的应用，比如找不重复的数，统计数据等

打标签问题。

BitMap算法的缺点：

1.数据不能重复：数据只有0和1 也就是有或者没有 不知道有多个

2.数据量少时相对于普通的hash没有优势

3.无法处理字符串：hash冲突