# [笔试题目] 简单总结笔试和面试中的海量数据问题

原创 Eastmount 2015-10-08 06:16:03 5519 收藏 1 版权

分类专栏： 面试工作 文章标签： 海量数据 面试 在线笔记 hash



最近在笔试和面试中遇到了很多关于海量数据的问题，在此进行简单的记录，写一篇方便自己下次学习的处理海量数据的文章及在线笔记，同时也希望对你有所帮助。当然，海量数据最出名的还是七月July，但这里我是想直接从实际题目出发，并参考及摘抄了他们那些大牛的文章及自己的想法进行简单总结记录。

## 一. 原题重现

2015年9月27日百度笔试论述题二选一，其中第一道是关于MapReduce相关的；第二道是搜索引擎中url去重，海量数据集url如何在爬取过程中避免重复爬取过的url。

PS：通常搜索引擎网页去重是通过文档特征提取，再计算相似性或集合Hash实现。

下面是常见的题型：

**1.Hash算法处理海量数据部分**

【题目1】(安卓越 2012) 给定a、b两个文件，各存放50亿个url，每个url各占64字节，内存限制是4G，让你找出a、b文件共同的url?

【题目2】海量日志数据，提取出某日访问百度次数最多的那个IP。

【题目3】有10个文件，每个文件1G，每个文件的每一行存放的都是用户的query，每个文件的query都可能重复。要求你按照query的频度排序。

【题目4】有一个1G大小的一个文件，里面每一行是一个词，词的大小不超过16字节，内存限制大小是1M。返回频数最高的100个词。

**2.Top-K海量数据部分**

【题目1】(360公司 2012) 100万条记录的文本文件，取出重复数最多的前10条。

【题目2】(360公司 2012) 100亿条记录的文本文件，取出重复数最多的前10条。

【题目3】(腾讯公司 2011) 服务器内存1G，有一个2G的文件，里面每行存着一个QQ号（5-10位数），怎么最快找出出现过最多次的QQ号。

【题目4】(腾讯公司 2015 牛客网) 搜索引擎的日志要记录所有查询串,有一千万条查询,不重复的不超过三百万，要统计最热门的10条查询串，内存<1G，字符串长 0-255。

(1) 主要解决思路；(2) 算法及其复杂度分析。

**3.bit海量数据部分**

【题目1】(腾讯公司)给40亿个不重复的unsigned int的整数，没排过序的，然后再给一个数，如何快速判断这个数是否在那40亿个数当中?

【题目2】(July整理) 在2.5亿个整数中找出不重复的整数，注，内存不足以容纳这2.5亿个整数。

## 二. Hash算法处理海量数据

该部分我认为酷辣虫社区的"魂牵梦萦"文章写得非常不错，感觉比July的那篇文章更通俗易懂，所以该部分转载了他的文章。强烈推荐大家阅读原文，网址：

http://www.colabug.com/thread-1148595-1-1.html

**第一部分 概述**

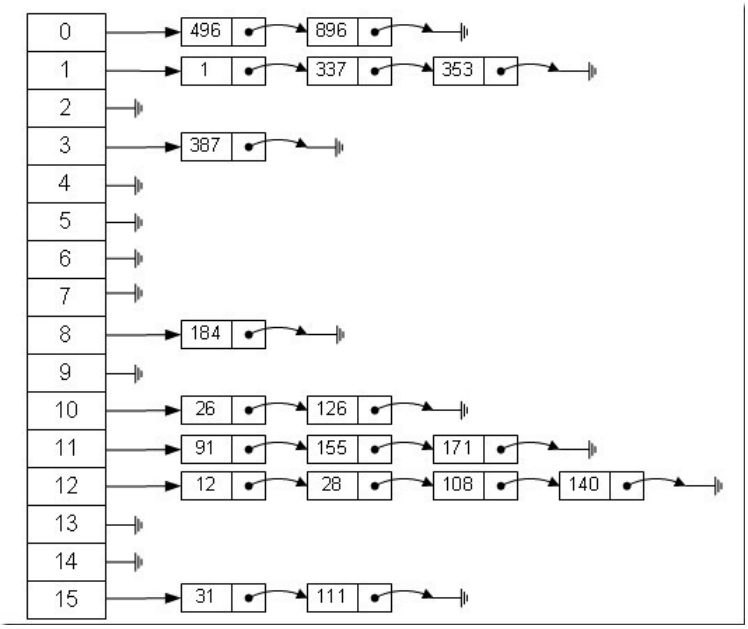
本文将粗略讲述一下Hash算法的概念特性，里边会结合分布式系统负载均衡实例对Hash的一致性做深入探讨。另外，探讨一下Hash算法在海量数据处理方案中的通用性。最后，从源代码出发，具体分析一下Hash算法在MapReduce框架的中的应用。

## 第二部分 Hash算法

Hash可以通过散列函数将任意长度的输入变成固定长度的输出，也可以将不同的输入映射成为相同的相同的输出，而且这些输出范围也是可控制的，所以起到了很好的压缩映射和等价映射功能。这些特性被应用到了信息安全领域中加密算法，其中等价映射这一特性在海量数据解决方案中起到相当大的作用，特别是在整个MapReduce框架中，下面章节会对这二方面详细说。

话说，Hash为什么会有这种 压缩映射和等价映射功能，主要是因为Hash函数在实现上都使用到了取模。下面看看几种常用的Hash函数：

- 直接取余法：f(x):= x mod maxM ; maxM一般是不太接近 2^t 的一个质数。
- 乘法取整法：f(x):=trunc((x/maxX)*maxlongit) mod maxM，主要用于实数。
- 平方取中法：f(x):=(x*x div 1000 ) mod 1000000); 平方后取中间的，每位包含信息比较多。




此处参考July文章：

Hash就是把任意长度的输入通过散列算法，变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，其实hash就是找到一种数据内容和数据存放地址之间的映射关系。

数组的特点是：寻址容易，插入和删除困难；而链表的特点是：寻址困难，插入和删除容易。那么我们能不能综合两者的特性，做出一种寻址容易，插入删除也容易的数据结构？答案是肯定的，这就是哈希表，哈希表有多种不同的实现方法，我接下来解释的是最常用的一种方法——拉链法，我们可以理解为“链表的数组”，如上图所示。

适用范围：快速查找，删除的基本数据结构，通常需要总数据量可以放入内存。

## 第三部分 Hash算法在海量数据处理方案中的应用

单机处理海量数据的大体主流思想是和MapReduce框架一样，都是采取分而治之的方法，将海量数据切分为若干小份来进行处理，并且在处理的过程中要兼顾内存的使用情况和处理并发量情况。而更加仔细的处理流程大体上分为几步（对大多数情况都使用，其中少部分情况要根据你自己的实际情况和其他解决方法做比较采用最符合实际的方法）：

**第一步：分而治之。**

采用Hash取模进行等价映 射。采用这种方法可以将巨大的文件进行等价分割（注意：符合一定规律的数据要被分割到同一个小文件）变成若干个小文件再进行处理。这个方法针对数据量巨大，内存受到限制时十分有效。

**第二步：利用hashMap在内存中进行统计。**

我们通过Hash映射将大文件分割为小文件后，就可以采用HashMap这样的存储结构来对小文件中的关注项进行

频率统计。具体的做法是将要进行统计的Item作为HashMap的key，此Item出现的次数作为value。

**第三步：对存储在HashMap中的数据根据出现的次数来进行排序。**

在上一步进行统计完毕之后根据场景需求往往需要对存储在HashMap中的数据根据出现的次数来进行排序。其中排序我们可以采用堆排序、快速排序、归并排序等方法。

现在我们来看看具体的例子:

**【题目1】(安卓越 2012) 给定a、b两个文件，各存放50亿个url，每个url各占64字节，内存限制是4G，让你找出a、b文件共同的url?**

思路: 还是老一套，先Hash映射降低数据规模，然后统计排序。

具体做法：

(1) 分析现有数据的规模

按照每个url64字节来算，每个文件有50亿个url，那么每个文件大小为**5G*64=320G**（按照1000换算10亿字节=1GB）。320G远远超出内存限定的4G，所以不能将其全部加载到内存中来进行处理，需要采用**分而治之**的方法进行处理。

(2) Hash映射分割文件

逐行读取文件a，采用hash函数：**Hash(url)%1000**将url分割到1000个小文件中，文件即为f1_1,f1_2,f1_3,...,f1_1000。那么理想情况下每个小文件的大小大约为300M左右。再以相同的方法对大文件b进行相同的操作再得到1000个小文件，记为： f2_1,f2_2,f2_3,...,f2_1000。

经过一番折腾后我们将大文件进行了分割并且将相同url都分割到了这2组小文件中下标相同的两个文件中,其实我们可以将这2组文件看成一个整体：

f1_1& f2_1， f1_2& ,f2_2, f1_3& f2_3,..., f1_1000& f2_1000

那么我们就可以将问题转化成为求这1000对小文件中相同的url就可以了。接下来，求每对小文件中的相同url，首先将每对对小文件中较小的那个的url放到**HashSet结构**中，然后遍历对应这对小文件中的另一个文件，看其是否存才刚刚构建的HashSet中，如果存在说明是一样的url，将这url直接存到结果文件就ok了。如果存在大文件接着hash划分即可。

**【题目2】 海量日志数据，提取出某日访问百度次数最多的那个IP**

思路：当看到这样的业务场景，我们脑子里应该立马会想到这些海量网关日志数据量有多大？这些IP有多少中组合情况，最大情况下占多少存储空间？解决这样的问题前我们最重要的先要知道数据的规模，这样才能从大体上制定解决方案。所以现在假设这些这些**网关日志量有3T**。下面大体按照我们上面的步骤来对解决此场景进行分析：

（1）首先，从这些海量数据中过滤出指定一天访问百度的用户IP，并逐个写到一个大文件中。

（2）**采用“分而治之”的思想用Hash映射将大文件进行分割降低数据规模。**

按照IP地址的Hash(IP)%1024值，把海量IP日志分别存储到1024个小文件中，其中Hash函数得出值为分割后小文件的编号。

（3）逐个读小文件，对于每一个小文件**构建一个IP为key，出现次数为value的HashMap。**

对于怎么利用HashMap记录IP出现的次数这个比较简单，因为我们可以通过程序读小文件将IP放到HashMap中key的之后可以先判断此IP是否已经存在如果不存在直接放进去，其出现次数记录为1，如果此IP已经存储则过得其对应的value值也就是出现的次数然后加1就ok。最后，按照IP出现的次数采用排序算法对HashMap中的数据进行**排序**，同时记录当前出现次数最多的那个IP地址。

（4）走到这步，我们可以得到1024个小文件中出现次数最多的IP了，再采用常规的排序算法找出总体上出现次数最多的IP就ok了。

这个我们需要特别地明确知道一下几点内容：

第一：我们通过Hash函数:**Hash(IP)%1024**将大文件映射分割为了1024个小文件，那么这1024个小文件的大小是否均匀？另外，我们采用HashMap来进行IP频率的统计，内存消耗是否合适?

首先是第一个问题，被分割的小文件的大小的均匀程度是取决于我们使用怎么样的Hash函数 ，对本场景而言就是： Hash(IP)%1024。设计良好的Hash函数可以减少冲突，使数据均匀的分割到1024个小文件中。但是尽管数据映射到了另外一些不同的位置，但数据还是原来的数据，只是代替和表示这些原始数据的形式发生了变化而已。

另外，看看第二个问题：用HashMap统计IP出现频率的内存使用情况。

要想知道HashMap在统计IP出现的频率，那么我们必须对IP组合的情况有所了解。32Bit的IP最多可以有**2^32**种的组合方式，也就是说去所有IP最多占**4G存储空间**。在此场景中，我们已经根据IP的hash值将大文件分割出了1024个小文件，也就是说这4G的IP已经被分散到了1024个文件中。那么在Hash函数设计合理最perfect的情况下针对每个小

文件的HashMap占的内存大小最多为4G/1024+存储IP对应的次数所占的空间，所以内存绝对够用。

第二：Hash取模是一种**等价映射**，换句话说通过映射分割之后相同的元素只会分到同一个小文件中去的。就本场景而言，相同的IP通过Hash函数后只会被分割到这1024个小文件中的其中一个文件。

**【题目3】有10个文件，每个文件1G，每个文件的每一行存放的都是用户的query，每个文件的query都可能重复。要求你按照query的频度排序。**

**【题目4】 有一个1G大小的一个文件，里面每一行是一个词，词的大小不超过16字节，内存限制大小是1M。返回频数最高的100个词。**

像例子3和例子4这些场景都可以用我们的一贯老招数解决：先Hash映射降低数据规模，然后统计加载到内存，最后排序。具体做法可以参考上面2个例子。

最后简述July的处理方法，为后面的论述作下铺垫。

**时间：Bloom filter、Hash、bit-map、堆、倒排索引**

**空间：分而治之、hash映射**

**集群：分布式、并行计算**

例：300万个查询字符串中统计最热门的10个查询

若10亿个则先划分小，1000个小文件中，再hashmap通过数量，归并top10，而该题数据300万较小，够内存中处理，HashTable+堆实现<key,value>对应<字符串,次数>的top10获取。

**参考文章：**

[NoSQL] 海量数据解决思路之Hash算法

十一 从头到尾解析Hash表算法 - by:七月

[C/C++] 海量数据处理利器 STL中哈希表 hash_map（C++）

## 三. Top-K问题解决

最经典的Top-K问题我介绍的是2012年360的php面试方向题目：

**【题目1】(360公司 2012) 100万条记录的文本文件，取出重复数最多的前10条。**

示例文本：

098
123
234
789
.....
234
678
654
123

**【题目2】(360公司 2012) 100亿条记录的文本文件，取出重复数最多的前10条。**

刚才是100万的数据，你的计算机可以单批正常处理，现在有100亿的数据，假设由于你的计算机内存、cpu限制，无法单批处理 ...

处理方法主要参考July中关于Top-K海量数据的介绍，方法如下：

1.100万的直接用hash存储**key(值),value(次数)**。同时建一个10个元素的数组，一个整数记录数据中最小次数，循环一次没有出现过的插入hash表中，value记1，如已存则value加1，value时大于数组中的最小次数则进行替换，改写整数值。

PS：假设一条记录64字节，100万应该为64MB（10亿字节=1GB，1000换算时），内存此时肯定够用；而100亿条此时需要640GB内存，显然分治实现。

2.100亿类似100万时的处理方法，对**数据进行切片**，可以都切为100万的记录，对100万最前10，不同在于这前10也存入hash，如果key相同则合并value，显然100亿的数据分割完后的处理结果也要再进行类似的处理，hash表不能过长，原理其实也就是map和reduce。

简单总结步骤如下：

**(1) 分而治之**

**(2) HashMap<key,value>=<字符串,次数>**
**(3) 数据合并类似MapReduce**
**(4) 排序输出TopK，可以采用高效的堆排、快排 、归并排序等**

**【题目3】(腾讯公司 2011) 服务器内存1G，有一个2G的文件，里面每行存着一个QQ号（5-10位数），怎么最快找出出现过最多次的QQ号。**

解救方法类似，通过hash方法将qq分配到10个文件（硬盘）中，相同qq在同一文件中；再统计每个文件里面qq出现次数，通过hashmap(qq, qq_count)实现；最后计算10个文件中最大的访问qq数即可。

**【题目4】(腾讯公司 2015 牛客网) 搜索引擎的日志要记录所有查询串,有一千万条查询,不重复的不超过三百万，要统计最热门的10条查询串，内存<1G，字符串长 0-255。**

**(1) 主要解决思路；(2) 算法及其复杂度分析。**

首先一千万条查询记录，每条字符串长0~255，而限制内存< 1G，所以不能把一千万条记录全部放进内存中处理，经计算，1千万条记录的最大占用空间大小为256Byte*10^7=0.25KB * 10^7=2.5*10^6KB，而1G = 1024M = 1024*1024KB = 1.024*1.024*10^6KB，可以使用hash分割将1千万条记录分成25个记录块，Hash(字符串记录)%25，使得相同的字符串记录在相同的记录块中，再使用哈希表来计算出40万条记录重复次数最大的前10条记录，哈希表的key是记录字符串，值是重复次数。这样25次访问完1千万条记录，将会得到250条记录，然后使用Map存储这250条记录，key是重复次数，值是记录字符串，比较函数是greater函数对象，从大到小存储在Map中，前10条即是最热门的10条查询串。复杂度分析：2N + klogk ~ O(N)。

但又好像需要hash+堆完成，建立10个最小堆。参考：牛客网

**参考July文章：**

教你如何迅速秒杀掉：99%的海量数据处理面试题
十道海量数据处理面试题与十个方法大总结
360 php 面试题 - oschina

## 四. bit处理海量数据

最经典的就是腾讯的面试题，同时这部分也简单包括Bloom Filter和Bit-map介绍，此部分都是摘抄July文章内容，后又原文地址，大家可以去膜拜下七月大神。

**【题目1】(腾讯公司)给40亿个不重复的unsigned int的整数，没排过序的，然后再给一个数，如何快速判断这个数是否在那40亿个数当中?**

**方案一：**申请512M内存，一个bit位代表一个unsigned int值，读40亿个数，设置相应bit位，读入要查询位，查看bit是否为1，若为1表示存在否则表示不存在。

**方案二：**这个问题在《编程珠玑》里有很好的描述，大家可以参考下面的思路，探讨一下：

又因为2^32为40亿多，所以给定一个数可能在，也可能不在其中；这里我们把40亿个数中的每一个用32位的二进制来表示。假设这40亿个数开始放在一个文件中，然后将这40亿个数分成两类：

1.最高位为0
2.最高位为1

并将这两类分别写入到两个文件中，其中一个文件中数的个数<=20亿，而另一个>=20亿（这相当于折半了）；与要查找的数的最高位比较并接着进入相应的文件再查找。再然后把这个文件为又分成两类:

1.次最高位为0
2.次最高位为1

并将这两类分别写入到两个文件中，其中一个文件中数的个数<=10亿，而另一个>=10亿（相当于折半）；与要查找的数的次最高位比较并接着进入相应的文件再查找。

.....

以此类推，就可以找到了,而且时间复杂度为O(logn)，方案2完。

**【题目2】(July整理) 在2.5亿个整数中找出不重复的整数，注，内存不足以容纳这2.5亿个整数。**

方案一：采用2-Bitmap（每个数分配2bit，00表示不存在，01表示出现一次，10表示多次，11无意义）进行，共需内存2^32 * 2 bit=1 GB内存，还可以接受。然后扫描这2.5亿个整数，查看Bitmap中相对应位，如果是00变01，

01变10，10保持不变。所描完事后，查看bitmap，把对应位是01的整数输出即可。

方案二：也可采用与海量日志中找IP次数最多的类似方法，进行划分小文件的方法。然后在小文件中找出不重复的整数，并排序。然后再进行归并，注意去除重复的元素。

**什么是Bit-map**

所谓的Bit-map就是用一个bit位来标记某个元素对应的Value， 而Key即是该元素。由于采用了Bit为单位来存储数据，因此在存储空间方面，可以大大节省。

来看一个具体的例子，假设我们要对0-7内的5个元素(4,7,2,5,3)排序（这里假设这些元素没有重复）。那么我们就可以采用Bit-map的方法来达到排序的目的。要表示8个数，我们就只需要8个Bit（1Bytes），首先我们开辟1Byte的空间，将这些空间的所有Bit位都置为0(如下图：)



然后遍历这5个元素，首先第一个元素是4，那么就把4对应的位置为1（可以这样操作 p+(i/8)|(0×01<<(i%8)) 当然了这里的操作涉及到Big-ending和Little-ending的情况，这里默认为Big-ending），因为是从零开始的，所以要把第五位置为一（如下图）：



然后再处理第二个元素7，将第八位置为1，接着再处理第三个元素，一直到最后处理完所有的元素，将相应的位置为1，这时候的内存的Bit位的状态如下：



然后我们现在遍历一遍Bit区域，将该位是一的位的编号输出（2，3，4，5，7），这样就达到了排序的目的。BitMap可进行数据的快速查找，判重，删除，一般来说数据范围是int的10倍以下。

基本原理及要点：使用bit数组来表示某些元素是否存在，比如8位电话号码。

**参考July文章：**

海量数据处理之Bloom Filter详解

十七道海量数据处理面试题与Bit-map详解

http://taop.marchtea.com/06.07.html

## 总结：

这类题的方法涉及到很多知识，包括：

**Bloom Filter、Hash、Bit-Map、堆(Heap)、分而治之、数据库索引、倒排索引、外排序、Trie树(字典树)、MapReduce**

每一个内容都非常难，建议大家去阅读July的文章，大神就是大神，确实讲得非常好，自己收获也颇多。还有为什么这么多获得图灵奖的都是搞算法的，最后你看看他们做的东西吧。最近屠呦呦获得了中国第一个非文学的诺贝尔奖，也多么希望什么时候中国能获得以下图灵奖啊！

罗伯特·弗洛伊德1978年获奖，堆排序算法和Floyd-Warshall算法的创始人。

迪科斯彻1972年获奖，荷兰人。最短路径算法、银行家算法、提出信号量PV原语、解决"哲学聚餐"问题等。

查尔斯·霍尔1980年获奖，快速排序QuickSort算法之父、霍尔逻辑、程序语言定义。

约翰·麦卡锡1971年获奖，人工智能之父，α-β搜索法、LISP发明者。

肯·汤普逊和丹尼斯·里奇1983年获奖，设计了B语言、C语言、Go语言，创建Unix操作系统。

最后希望文章对你有所帮助，这是一篇我自己记录海量数据处理的一篇在线笔记，希望以后能用到。

(By：Eastmount 2015-10-8 清晨6点 http://blog.csdn.net/eastmount/)