海量数据处理

2017-04-06 | 数据结构和算法 | | | 95.5k | 30 | 28

海量数据处理，顾名思义，是指基于海量数据的存储和处理，因为数据量过大，导致要么无法短时间内解决，要么无法一次性装入内存。

总的方法论：

1.时间问题，巧妙算法+合适数据结构（布隆过滤器、散列、位图、堆、数据库、倒排索引、Trie树）
2.空间问题，分而治之，eg:散列映射。

典型方法：

1.散列分治 2.多层划分 3.MapReduce 4.外排序 5.位图（bitmap）6.布隆过滤器 7.Trie树 8.数据库 9.倒排索引 10.simhash()
哈希函数：
哈希函数又叫散列函数，哈希函数的输入域可以是非常大的范围，但是输出域是固定范围。假设为s.
1.典型的哈希函数都拥有无限的输入值域。
2.输入值相同时，访问值一样。
3.输入值不同时，返回值可能一样，也可能不一样。
4.不同输入值得到的哈希值，整体均匀分布在输出域s上。（重要，是评价指标）

MD5 和SHA1是经典的哈希函数算法。

将数据（如一段文字）运算变为另一固定长度值，是散列算法的基础原理。
Map-Reduce
1.Map阶段 -> 把大任务分成子任务
2.Reduce阶段 -> 子任务并发处理，然后合并结果。
难点在于工程上的处理。

注意点：

1.备份的策略，分布式存储的设计细节，以及容灾策略。

2.任务分配策略与任务进度跟踪的细节设计，节点状态的呈现。

3.多用户权限的控制。

常见海量处理题目解题概念：

1.分而治之。通过哈希函数将大任务分流到机器，或分流成小文件。

2.常用的hashMap或bitmap

难点:通讯、时间和空间的估算。

1.请对10亿个IPV4的ip地址进行排序，每个ip只会出现一次。

10亿 小于 2^32

利用bitmap，若出现则置1，然后将所有值为1 的ip输出,如

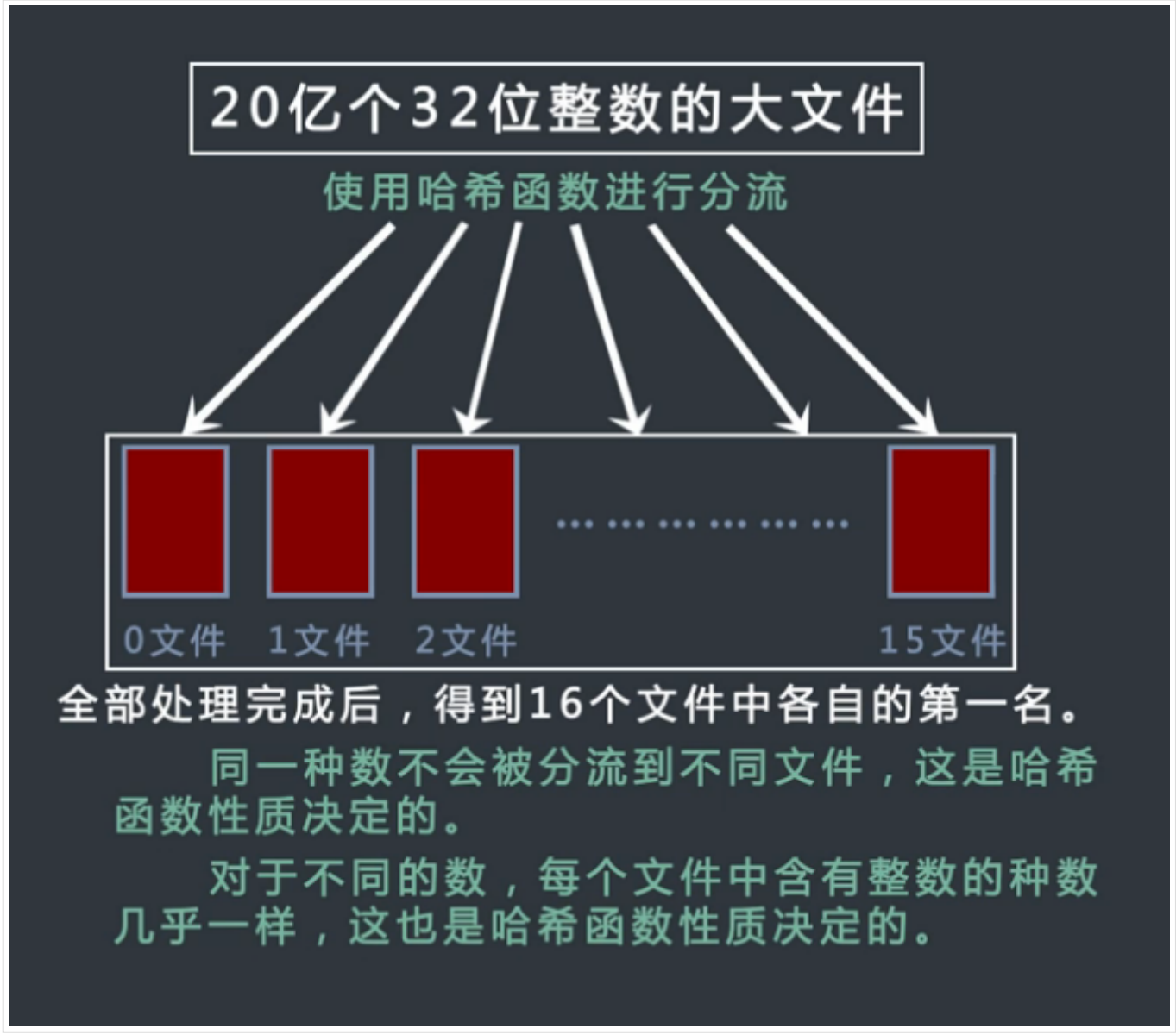
192.168.2.113 ->1921682113 将map中第1921682113处的位置置1，按序输出时，因为值为1，所以输出它

2.请对10亿人的年龄进行排序

年龄在0-200之间，使用计数排序

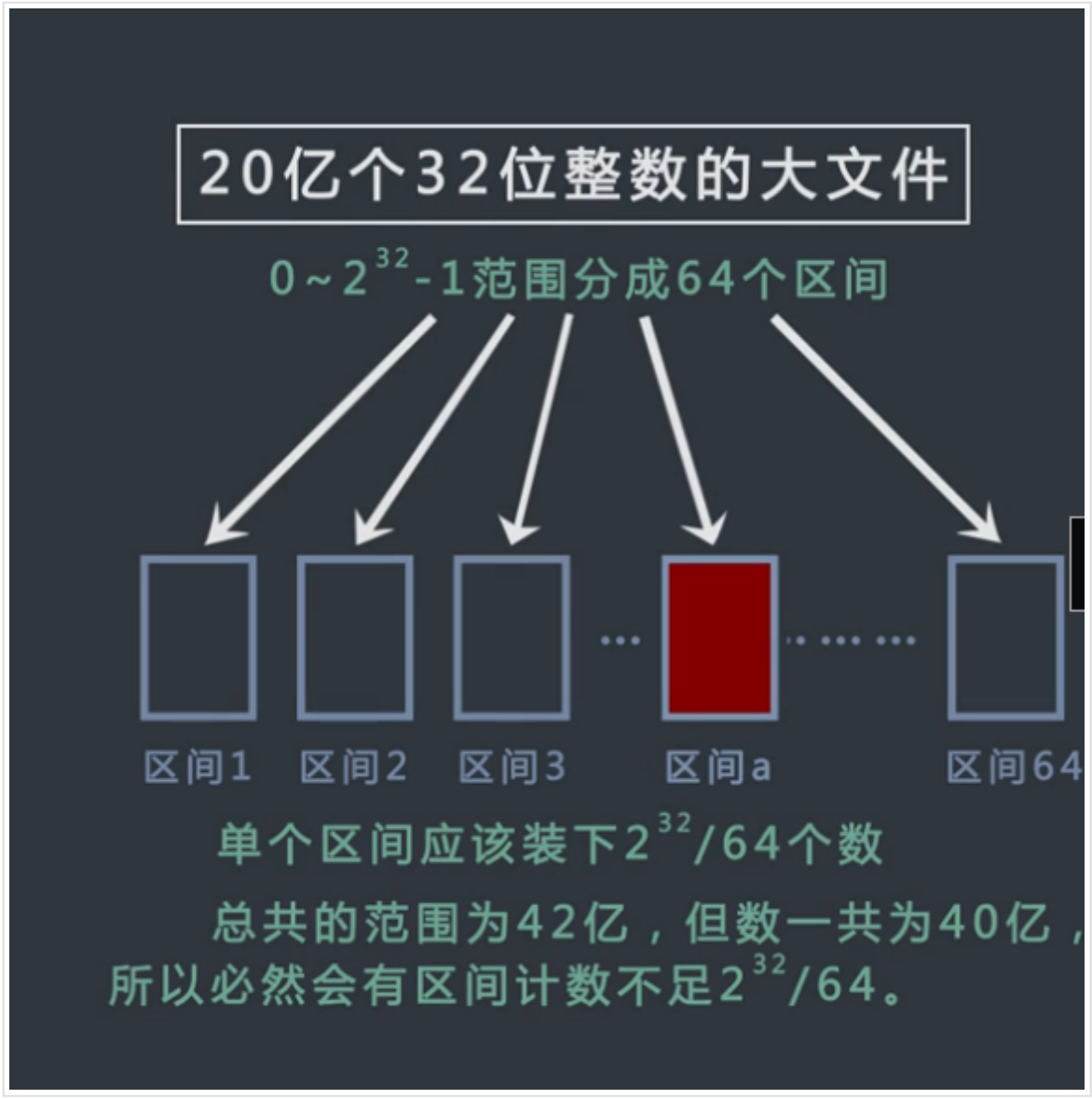
3.有一个包含20亿个全是32位整数的大文件，在其中找到的出现次数最多的数，但是内存限制只有2G.

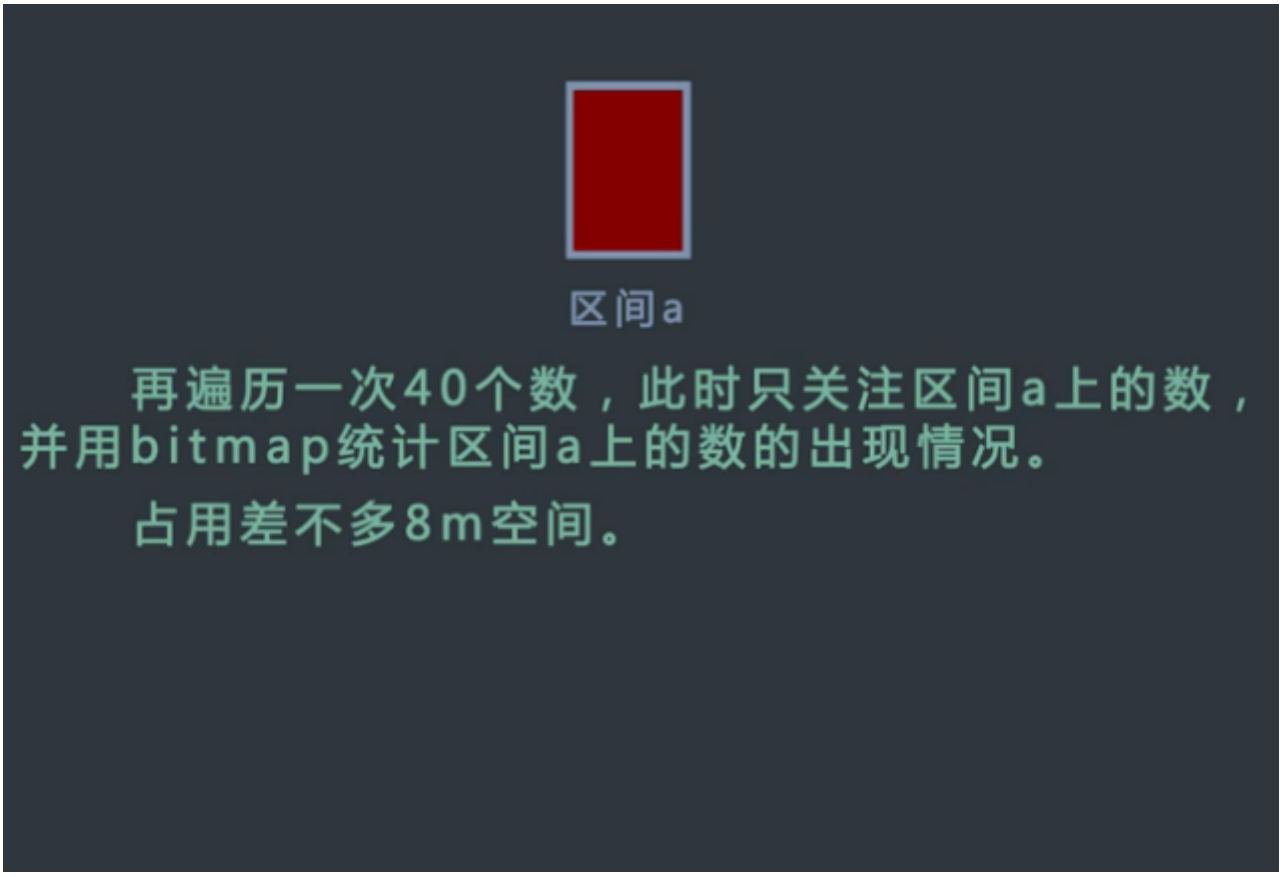
利用hash函数分流



4.32位无符号整数的范围是0~4294967295,现在有一个正好包含40亿个无符号整数的文件，所以在整个范围中必然有没出现过的数，可以使用最多10M的内存，只用找到一个没出现过的数即可，该如何找？

关键词：大区间化成很多个小区间。


20亿个32位整数的大文件
0~2^32-1范围分成64个区间
区间1
区间2
区间3
区间a
区间64
单个区间应该装下2^32/64个数
总共的范围为42亿，但数一共为40亿，
所以必然会有区间计数不足2^32/64。



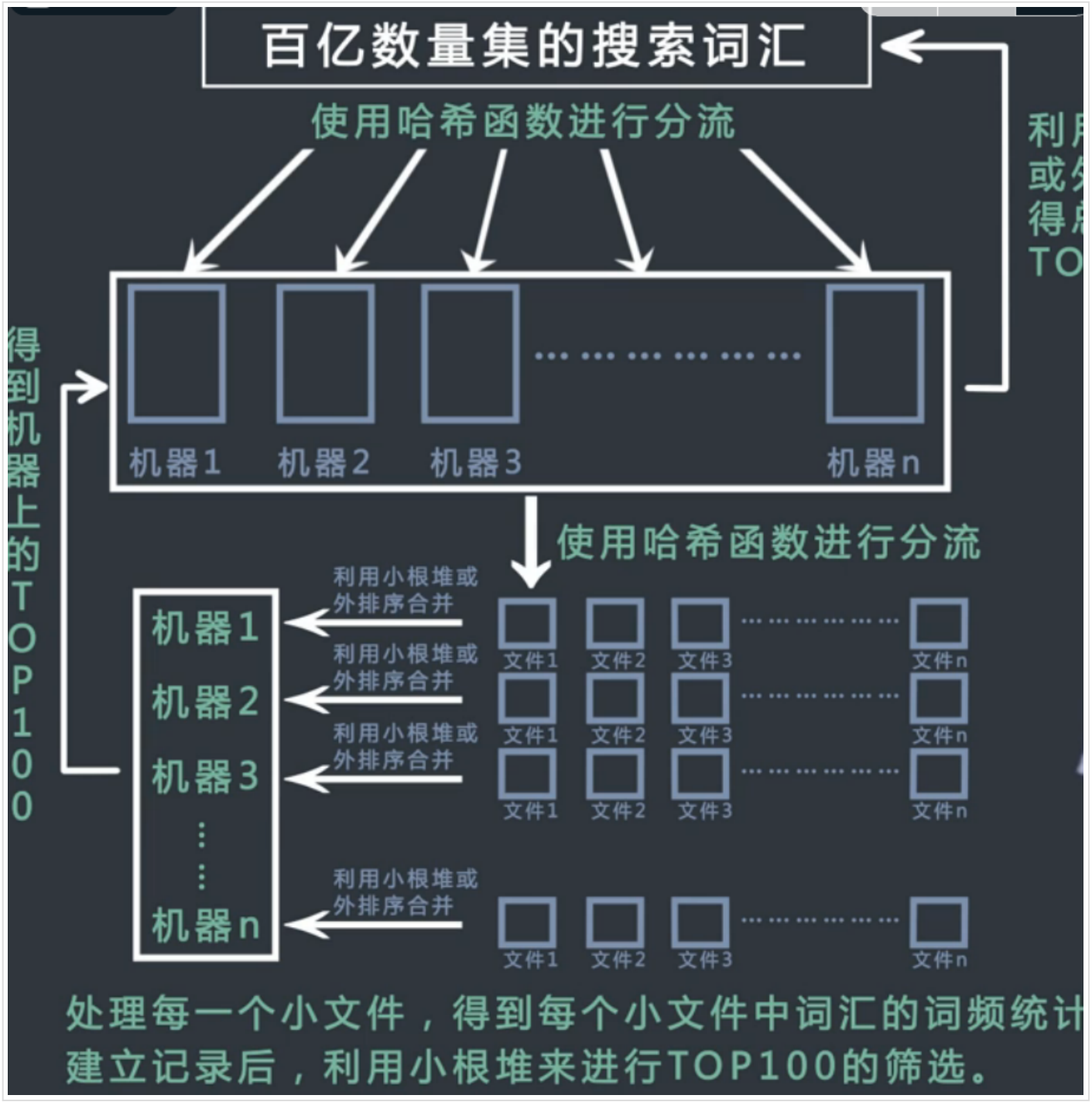
总结：

1.根据内存限制决定区间大小，根据区间大小，得到有多少个变量，来记录每个区间的数出现的次数。

2.统计区间上的数的出现次数，找到不足的区间。

3.利用bitmap对不满的区间，进行这个区间上的数的词频统计。

5.某搜索公司一天的用户搜索词汇是海量的额，假设有百亿的数据量，请设计一种求出每天最热100词的可行方法。



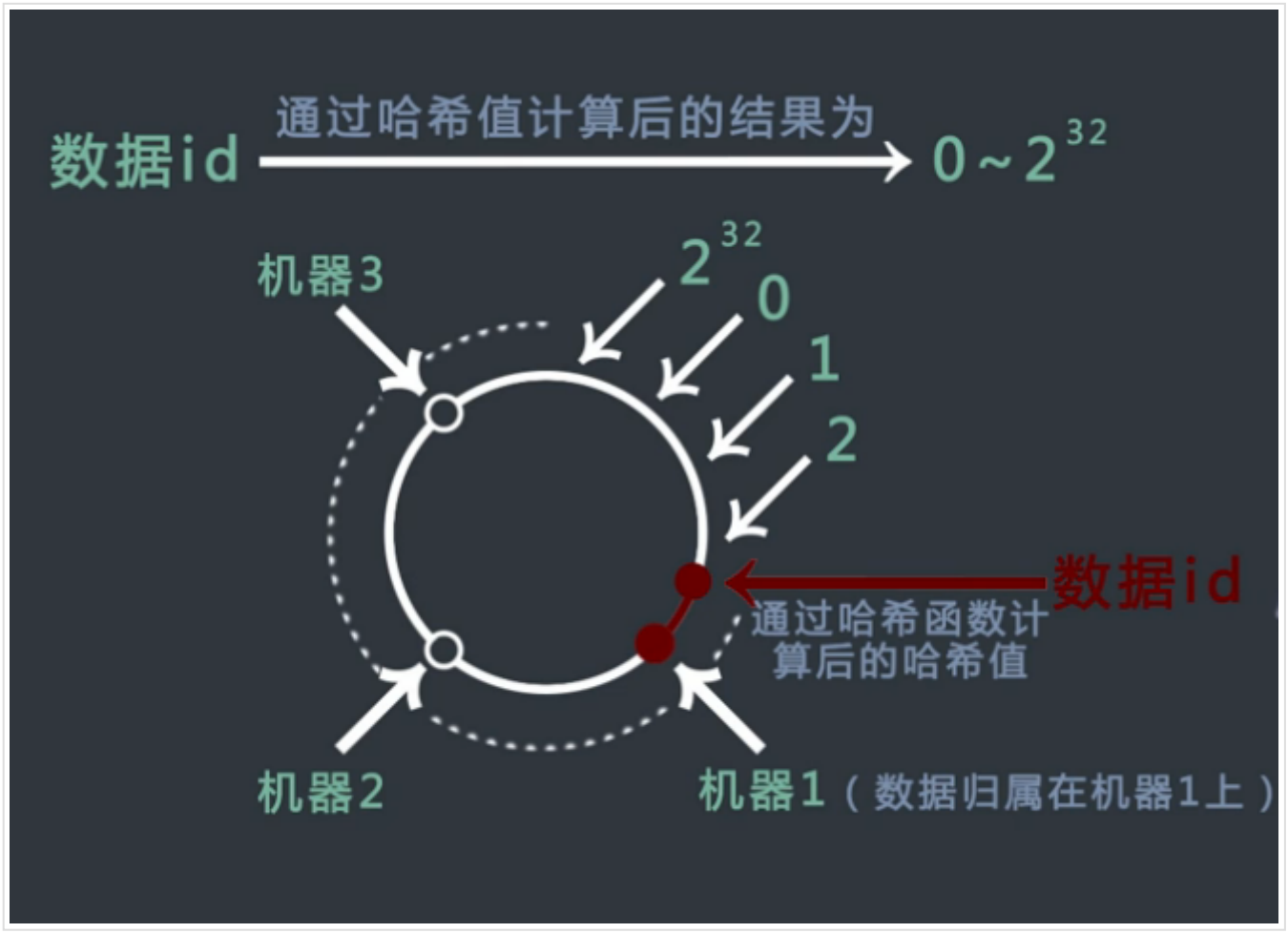
6.工程师常使用服务器集群来设计和实现数据缓存，以下是常见的策略。

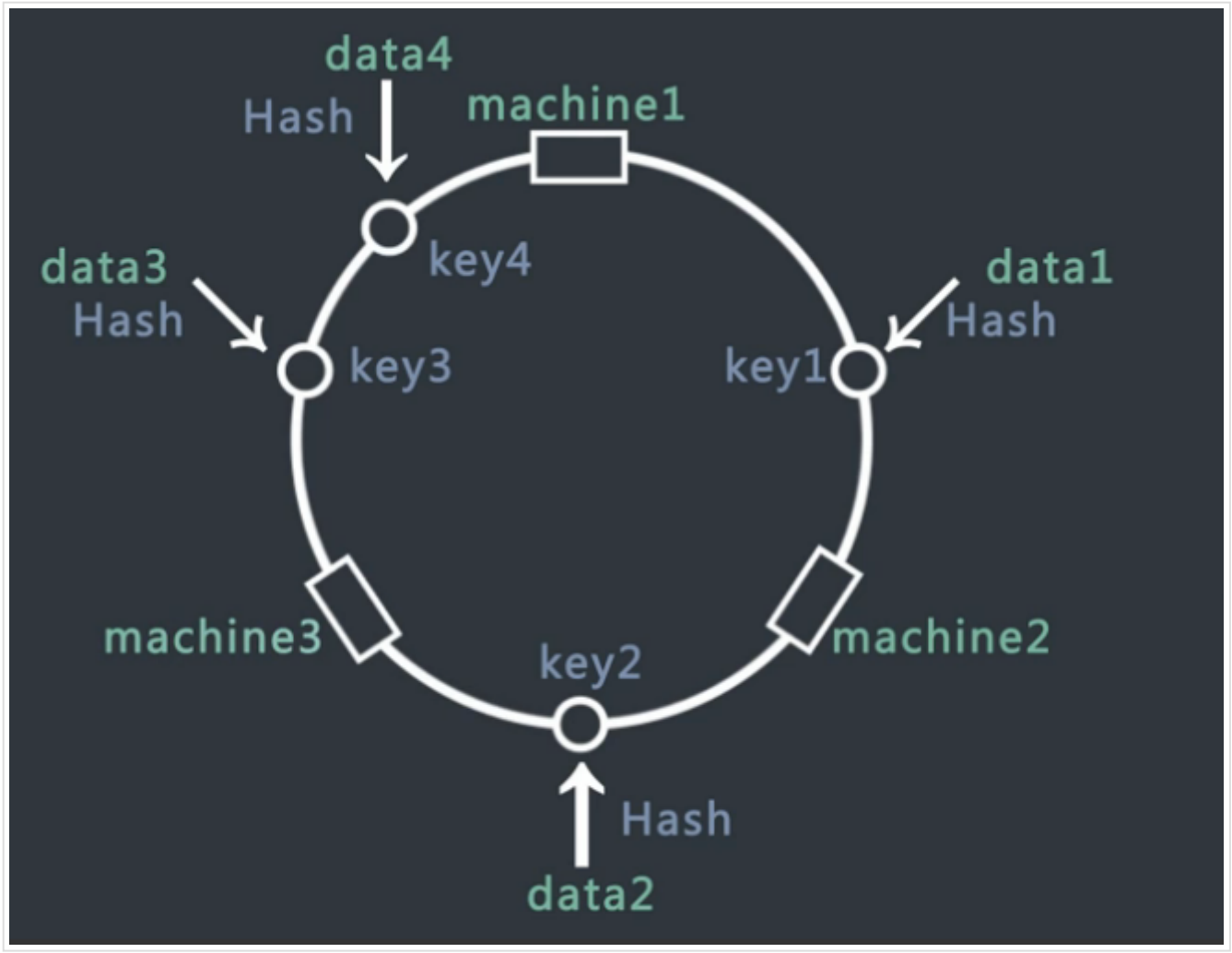
1.无论是添加、查询还算删除数据，都先将数据的ID通过哈希函数转换成一个哈希值，记为key.
2.如果目前机器有N台，则计算key%N的值，这个值就是该数据所属的机器编号，无论是添加，删除还算查询操作，都只是在这台机器上进行。请分析这个缓存策略可能带来的问题，并提出改进的方案。

潜在问题：如果增加或删除机器，数据迁移的代价很大。根据哈希函数得到的哈希值结果%N，当机器数N发生变化时，所有数据必须重新计算哈希值，以及对新的机器数M取余，来决定各自数据的归属。

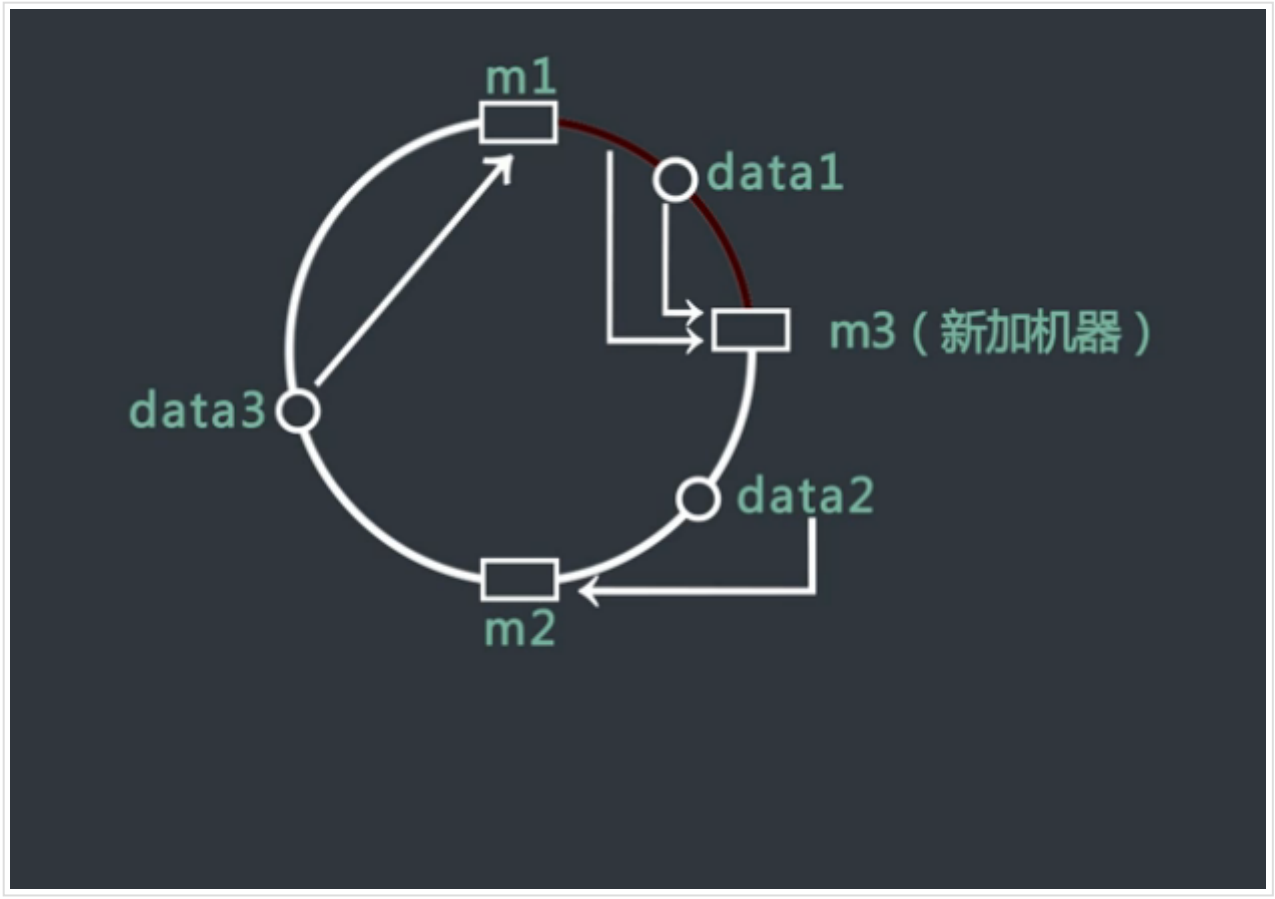
解决方法：

一致性哈希算法：




data4
Hash
machine1
key4
data3
Hash
key3
data1
Hash
key1
machine3
key2
machine2
Hash
data2

增加机器



扫一扫，关注我的微信公众号！

bigdata

感谢生命中有你们 【软技能 代码之外的生存指南】读书笔记

网友跟贴

0人参与

抵制低俗，文明上网，登录发贴

406458561 | 退出

发表跟贴

最新 最热

网易云跟贴，有你更精彩