**题目要求：**

100GB url 文件，使用 1GB 内存计算出出现次数 top100 的 url 和出现的次数。

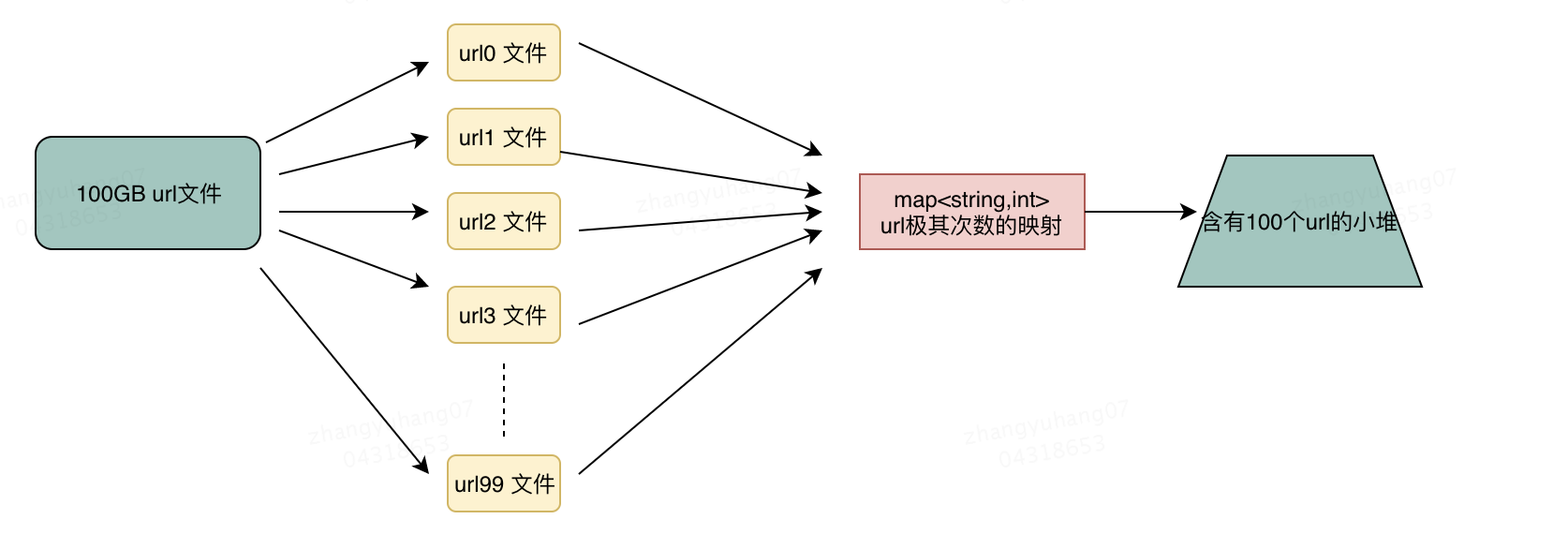
**整体思路**

1.要统计topK首先要做的是将每一个url出现的次数统计出来，但是问题在于内存只有1GB，而url文件有100GB，文件无法一次性加载到内存，而且要使用暴力遍历的方式解决的话，每次遇到一个url都要遍历整个文件，效率是十分底下的。这种方式显然不可取。

2.换一种思路，我们可以采用哈希分割的方式，采用哈希映射将相同的url放入同一个文件中，那么获得一个url出现的次数就可以只遍历当前文件来解决。

**分割过程：**

1. 可以将100GB的文件分割为100个小文件，创建100个文件，命名为 url0，url1.... url99。
2. 遍历整个大文件，读取每一个url，将url的每一个字符转化为ASCLL码，并进行相加取和 % 100，将该url写入对应的小文件中。
3. 依次遍历每一个小文件，统计出不同的url出现的次数，并将url及其出现的次数写入 unordered\_map<string,int>。
4. 如果遍历的小文件的大小大于1GB，就采用类似的哈希分割，则使用另一种哈希方法继续划分数据。
5. 统计出每一url出现的次数后，因为我们要求找出topK的url，因此需要建立一个小堆，里面共有100个url，然后遍历map，每当拿到的url出现的次数，大于堆顶url出现的次数，就进行替换，然后调节堆。
6. 所有的map中所有的url遍历完后，堆中剩余的就是前topK的url。



**时间复杂度分析：**

文件分割：O(N);

遍历所有小文件统计次数：O(N);

构建堆求topK: O(N\*(logK));

**过程中用到的C++STL容器**：unorderede\_map（统计每一个url出现的次数），priority\_queue(优先级队列，相当于是堆)

**设计思路：**

**1、采用生产者-消费者模式（阻塞队列），对读写任务可控，从而读内存使用可控，防止出现 超出内存限制的问题；**

**2、使用多线程读/写，提高效率；**