

# 第二次试验-DGIM 算法

计研 156 班 2015210874 王庆

## 一、实验目的

熟悉 DGIM 算法的运行过程。

## 二、实验内容

a) 对于正整数流 ( 数的范围是 1 到  $2^m$  ), 用 DGIM 算法估计流中大小为 N 的窗口内最近 k (  $1 \leq k \leq N$  ) 个整数的和。

b)  $m = 8$  ,  $N = 100,000,000$  ,  $k = 50000000$

## 三、试验详细过程

a) 使用 random 函数一次产生随机的 1 到  $2^8$  的正整数

b) 当产生的数字超过 100,000,000 个时开始估计, 每新产生 1000,000 个数后, 估计一个值。

c) 具有相同大小的桶的数目 r 可以取 2,3,4..... , 不做限制。

## 四、算法分析

数据流中的每个数据可以用一个时间戳 ( timestamp ) 标志该数据进入流的时间。

DGIM 算法利用桶 ( bucket ) 对滑动窗口进行划分, 每个桶保存以下信息:

- 桶的大小由一个 list 维护, list 的顺序为时间戳从小到大, 桶中超过窗口的值 remove 掉。
- 桶的大小, 即 2 次幂桶编号的倍数。

## 五、算法结果

$N=1000000$  ,  $k=500000$  , 正整数流为 [1,256], 那么需要 9 个桶, 分别为  $2^0$  ,  $2^1$  , ...  $2^8$

```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
0
308436.0 * 2^0 + 154093.0 * 2^1 + 77591.0 * 2^2 + 38389.0 * 2^3 + 19128.0 * 2^4 + 9516.0 * 2^5 + 4792.0 * 2^6 + 2431.0 * 2^7 + 2408.0 * 2^8
313437.0 * 2^0 + 156576.0 * 2^1 + 54215.0 * 2^2 + 39036.0 * 2^3 + 19468.0 * 2^4 + 9675.0 * 2^5 + 4871.0 * 2^6 + 2475.0 * 2^7 + 1687.0 * 2^8
220163.0 * 2^0 + 109887.0 * 2^1 + 55459.0 * 2^2 + 27349.0 * 2^3 + 19812.0 * 2^4 + 9845.0 * 2^5 + 3415.0 * 2^6 + 1736.0 * 2^7 + 1726.0 * 2^8
225305.0 * 2^0 + 112281.0 * 2^1 + 56726.0 * 2^2 + 27920.0 * 2^3 + 13973.0 * 2^4 + 6931.0 * 2^5 + 3486.0 * 2^6 + 1778.0 * 2^7 + 1776.0 * 2^8
197583.0 * 2^0 + 98386.0 * 2^1 + 49825.0 * 2^2 + 28542.0 * 2^3 + 14235.0 * 2^4 + 7059.0 * 2^5 + 3578.0 * 2^6 + 1554.0 * 2^7 + 1558.0 * 2^8
```

$N=100000000$  ,  $k=50000000$  ,正整数流为 $[1,256]$ ,那么需要 9 个桶 ,分别为 $2^0$  ,  $2^1$  ,  
...  $2^8$

```
25346455.0 * 2^0 + 12660062.0 * 2^1 + 6329327.0 * 2^2 + 3166765.0 * 2^3 + 1579850.0 * 2^4 + 792210.0 * 2^5 + 397773.0 * 2^6 + 199093.0 * 2^7 + 196817.0 * 2^8
25847397.0 * 2^0 + 12909944.0 * 2^1 + 6453321.0 * 2^2 + 3229242.0 * 2^3 + 1611150.0 * 2^4 + 807929.0 * 2^5 + 405597.0 * 2^6 + 202989.0 * 2^7 + 200783.0 * 2^8
26347622.0 * 2^0 + 13159597.0 * 2^1 + 6578618.0 * 2^2 + 3291295.0 * 2^3 + 1642459.0 * 2^4 + 823726.0 * 2^5 + 413408.0 * 2^6 + 206939.0 * 2^7 + 204688.0 * 2^8
26847051.0 * 2^0 + 13410399.0 * 2^1 + 6703938.0 * 2^2 + 3353397.0 * 2^3 + 1673717.0 * 2^4 + 839356.0 * 2^5 + 421168.0 * 2^6 + 210788.0 * 2^7 + 208538.0 * 2^8
27346807.0 * 2^0 + 13660232.0 * 2^1 + 6829353.0 * 2^2 + 3415836.0 * 2^3 + 1704902.0 * 2^4 + 855101.0 * 2^5 + 429011.0 * 2^6 + 214678.0 * 2^7 + 212432.0 * 2^8
```

## 六、算法复杂度分析

时间复杂度  $O(N)$  , 空间复杂度  $O(\log_2 N * \log_2 N)$