ES去重聚合评估调研报告

王浩

目前许多大数据查询系统已经使用高效快捷的Elasticsearch。但对于大数据量的去重聚合操作，由于Elasticsearch是分布式系统，主节点的性能无法支持汇聚所有节点结果并去重聚合。故Elasticsearch对于大数据量的去重操作采取的是基数估计方法，是一种概率统计的近似方法，它基于HyperLogLog++(HLL)算法, HLL 会先对我们的输入作哈希运算，然后根据哈希运算的结果中的 bits 做概率估算从而得到基数（数据集中唯一值的个数），具体算法见参考文献算法及分析部分。用此方法目的是以精确值换取了内存消耗。但是此方法只是一种概率估计的方法，与真实值存在差距。本报告基于此问题，重点研究去重误差的变化情况。

此算法特性：

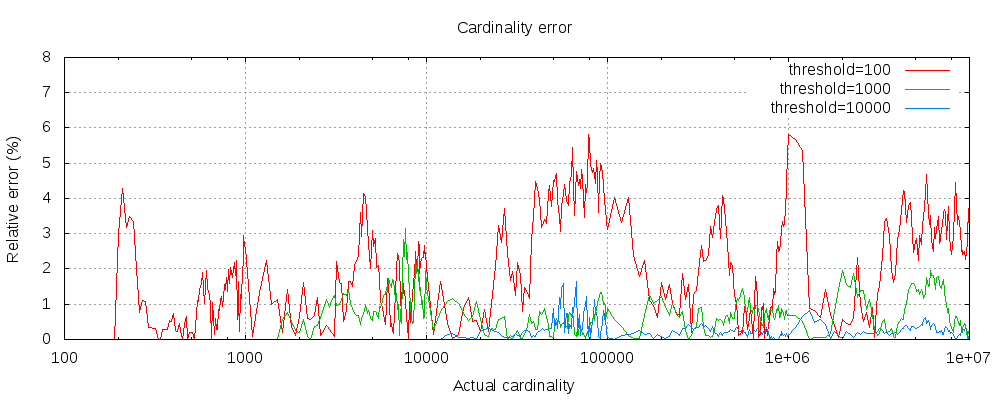
1. 可配置的精度，即认为的基数大小，用来控制内存的使用（更精确=更多内存）。
2. 小的数据集精度是非常高的。
3. 我们可以通过配置参数，来设置去重需要的固定内存使用量。无论数千还是数十亿的唯一值，内存使用量只与你配置的精确度相关。

通过查询时设置precision\_threshold参数（即认为基数唯一值的量）即配置精确阀值，这个阈值定义了在何种基数水平下我们希望得到一个近乎精确的结果。precision\_threshold 接受0-40000之间的数字，更大的值还是会被当40000来处理。对于指定的阀值，HLL 的数据结构会大概使用内存 precision\_threshold \* 8 字节，所以就必须在牺牲内存和获得额外的准确度间做平衡。

在实际应用中，100 的阀值可以在唯一值为百万的情况下仍然将误差维持 5% 以内。但我们使用时不会只设置100的阀值，更多时候考虑设置满阀值40000，降低误差。如图为官方基数误差图。横轴为实际基数，即唯一值数量，纵轴为相对误差百分比，红绿蓝三条线为precision\_threshold阀值，即分别代表定义的多大基数水平。

从图中可以看出，在设置precision\_threshold条件下，实际不同量小于precision\_threshold误差几乎为0。超过之后会产生误差值。设置阀值越高，误差相对越小。

通过调研，有文章提出基数过大的结论。在ElasticSearch中，对于去重计数（COUNT DISTINCT）是基于计数估计（Cardinality），因此如果去重记录数比较大（超过40000），便可能会有误差，误差范围是0~2%[1]。



ES官方提供不同精度阀值设置条件下，基数数据量与误差率的曲线图。

<https://www.elastic.co/blog/count-elasticsearch>

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-metrics-cardinality-aggregation.html>

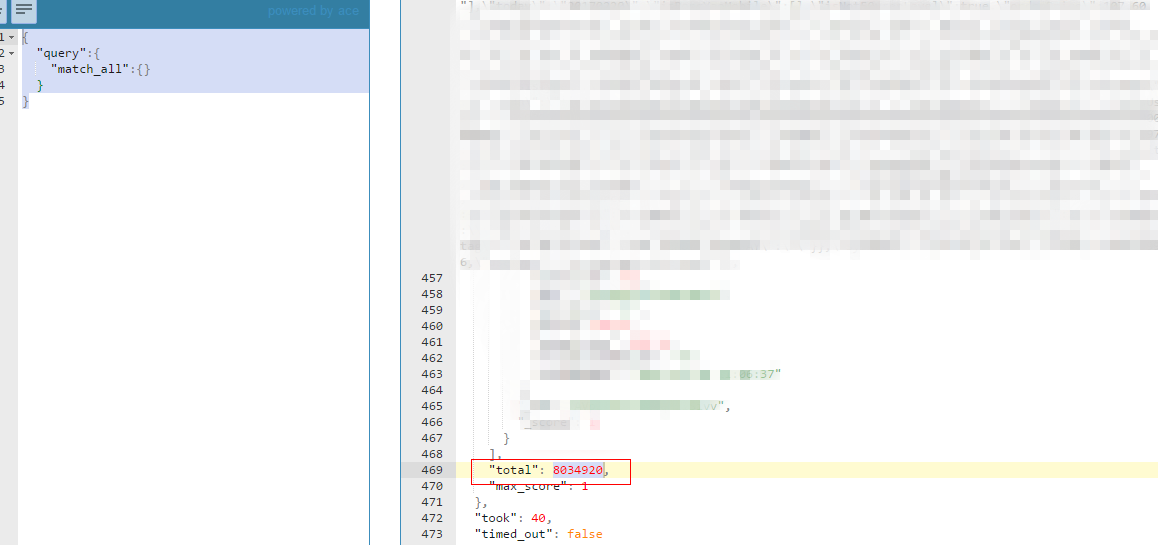
<http://lxw1234.com/archives/2015/12/588.htm>

**数据实验：**

1. 数据量基数大误差情况：

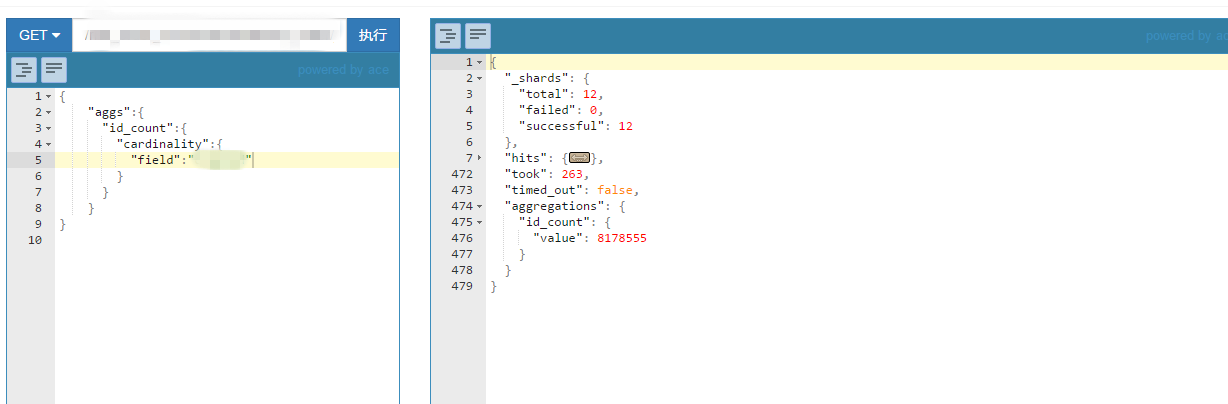
针对线上一组数据，基于某字段进行去重聚合分析。由于每一条记录的此字段值各不相同，即查询全部数据量就为此字段的基数量。

首先查询全部数据量为8034920条。（附录1）

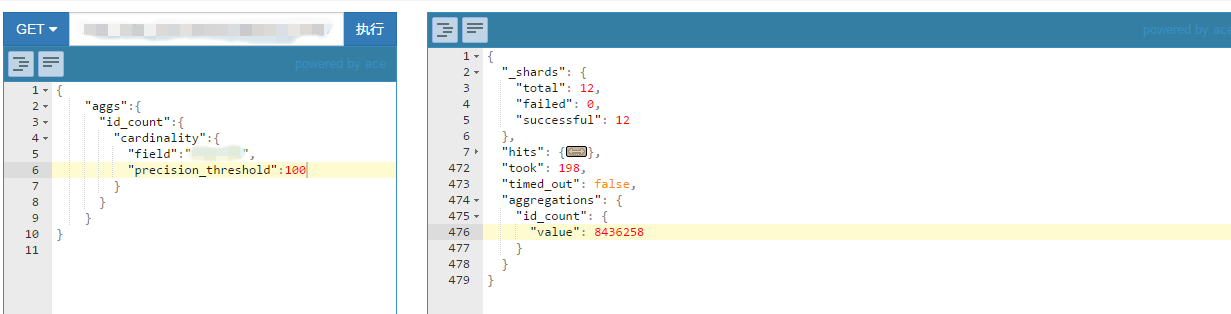


进行聚合去重分析时：

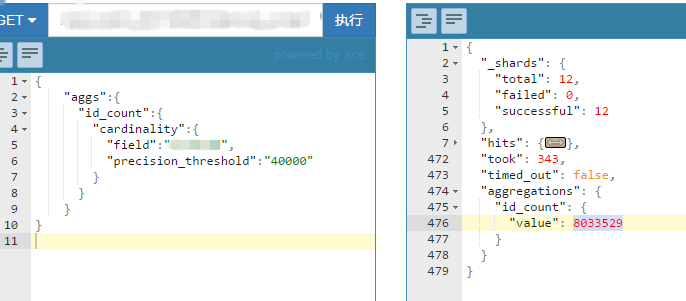
1. 不设阀值，查询数据量为8178555条，误差143635条，误差率1.7876%。耗时0.2s（附录2）



1. 设置阀值100时，查询数据量为8436258条，误差401338条，误差率4.995%。耗时0.198s。（附录3）



1. 设置最高阀值40000时，查询数据量为8033529条，误差-1391条，误差率0.0173%。耗时0. 319s。（附录4）



查询语句及结果见附录。由上述查询得到与官方提供走势曲线基本一致。

1. 数据量基数与阀值对比误差情况

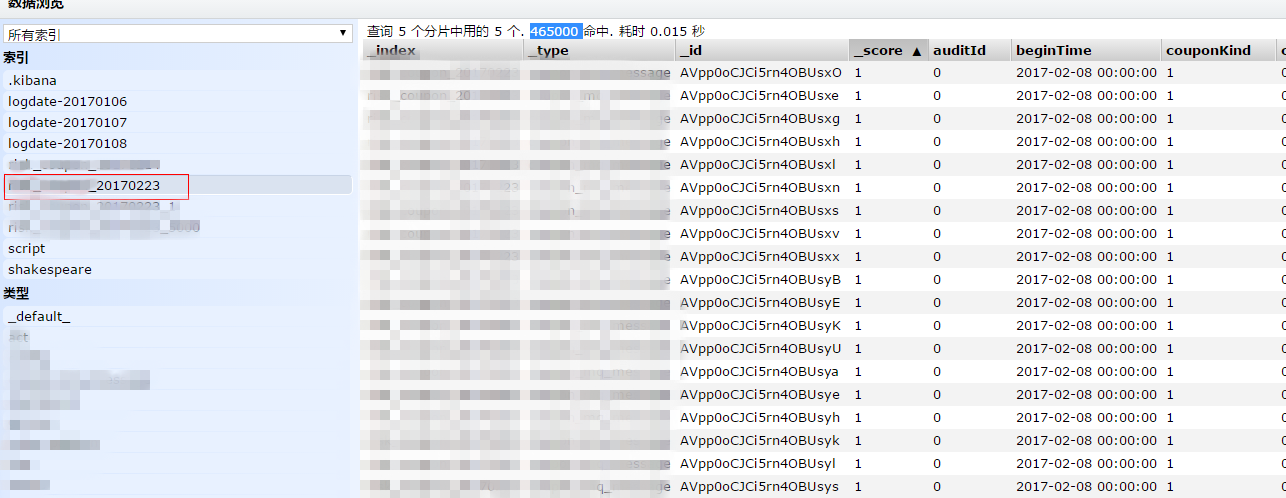
通过本地测试环境，生成数据进行分析。

数据总量：465000 条

基数（唯一值）量：30000条

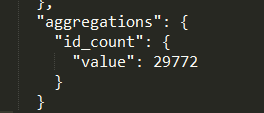
基数字段XX。

数据情况如图。



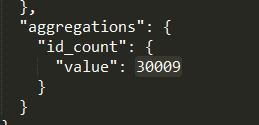
1. 不设置阀值（附录5）

本地环境耗时0.033s，结果29772条。误差0.76%。



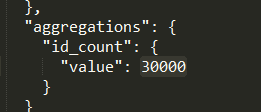
1. 阀值设置小于基数量。设置为10000。（附录6）

本地环境耗时0.085s，结果30009条。误差0.03%。



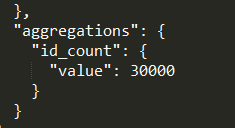
1. 阀值设置等于基数量。设置为30000。（附录7）

本地环境耗时0.102s，结果30000条。误差0%。



1. 阀值设置大于基数量。设置为40000。（附录8）

本地环境耗时0.046s，结果30000条。误差0%。



经过上述实验，验证了基数小于阀值时精确结果。基数大于阀值时会有一定的误差，且阀值越高误差越低。

附 录

(1)

{

"query":{

"match\_all":{}

}

}

(2)

{

"aggs":{

"id\_count":{

"cardinality":{

"field":"XX"

}

}

}

}

(3)

{

"aggs":{

"id\_count":{

"cardinality":{

"field":"XX",

"precision\_threshold":"100"

}

}

}

}

(4)

{

"aggs":{

"id\_count":{

"cardinality":{

"field":"XX",

"precision\_threshold":"40000"

}

}

}

}

(5)

curl -XGET 10.14.133.XX:9200/XXX/XXX/\_search -d '

{"aggs":{

"id\_count":{

"cardinality":{

"field":"XX"

}

}

}

}'

(6)

curl -XGET 10.14.133.XX:9200/XXX/XXX/\_search -d '

{"aggs":{

"id\_count":{

"cardinality":{

"field":"XX",

"precision\_threshold":10000

}

}

}

}'

(7)

curl -XGET 10.14.133.XX:9200/XXX/XXX/\_search -d '

{"aggs":{

"id\_count":{

"cardinality":{

"field":"XX",

"precision\_threshold":30000

}

}

}

}'

(8)

curl -XGET 10.14.133.XX:9200/XXX/XXX/\_search -d '

{"aggs":{

"id\_count":{

"cardinality":{

"field":"XX",

"precision\_threshold":40000

}

}

}

}'

参考文献

1. <http://lxw1234.com/archives/2015/12/588.htm>
2. <http://www.cnblogs.com/richaaaard/p/5319299.htm>
3. <https://qbox.io/blog/elasticsearch-aggregations-1-1-0>
4. <http://blog.csdn.net/heiyeshuwu/article/details/41248379>
5. <http://www.blogs8.cn/posts/ER2G7bf>
6. <http://106.186.120.253/preview/cardinality.html>
7. <http://www.tuicool.com/articles/7v6vyqm>

算法及分析：

1. <http://static.googleusercontent.com/media/research.google.com/fr//pubs/archive/40671.pdf>
2. <https://highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/>
3. <http://blog.codinglabs.org/articles/algorithms-for-cardinality-estimation-part-iv.html>
4. <http://blog.codinglabs.org/articles/algorithms-for-cardinality-estimation-part-iii.html>
5. <http://dataunion.org/13732.html>
6. <http://www.feellin.com/hyperloglogde-he-xin-si-xiang-yuan-li/>