elasticsearch

——自定义排序

By 刘刚

关于排序打分公式

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \text{ in } q} (tf(t \text{ in } d) \times idf(t)^2 \times t.getBoost() \times norm(t,d))$$

$$queryNorm(q) = \frac{1}{\sqrt{q.getBoost()^2 \times \sum_{t \text{ in } q} (idf(t) \times t.getBoost())^2}}$$

$$norm(t,d) = d.getBoost() \times lengthNorm(field) \times \prod_{field\ f\ in\ d} f.getBoost()$$

$$lengthNorm(f) = \frac{1}{\sqrt{num\ of\ terms\ in\ field\ f}}$$

一：那些奇奇怪怪的排序需求

①希望排序结果按照某个属性散列开

①希望排序结果按照某个属性散列开

永动机乐队第三张唱片《战•歌》预售 135%

来源：乐童音乐
已筹资：¥13,520 剩余：15天

打破底价——金丝楠不再束之高阁 2128%

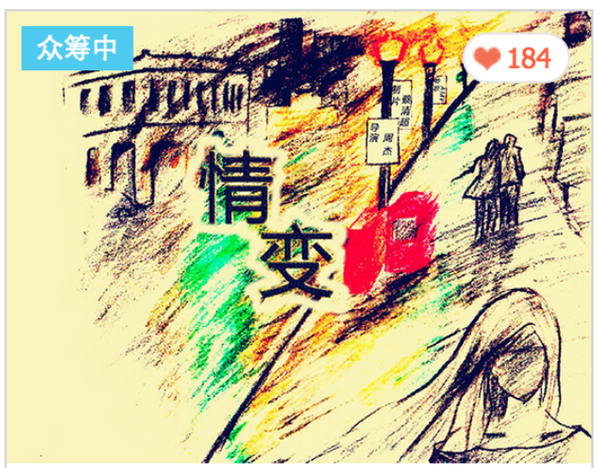
来源：众筹网
已筹资：¥63,822 剩余：3天

Yolanda智能体质管理方案，您健康管理第一站 16673%

来源 淘宝众筹
已筹资：¥1,667,376 剩余：5天

起点圆梦项目邮费差价 2300%

来源：中国梦网
已筹资：¥23 剩余：132天



微电影《情变》——生活微启示 12%

来源 淘梦网
已筹资：¥5,870 剩余：7天

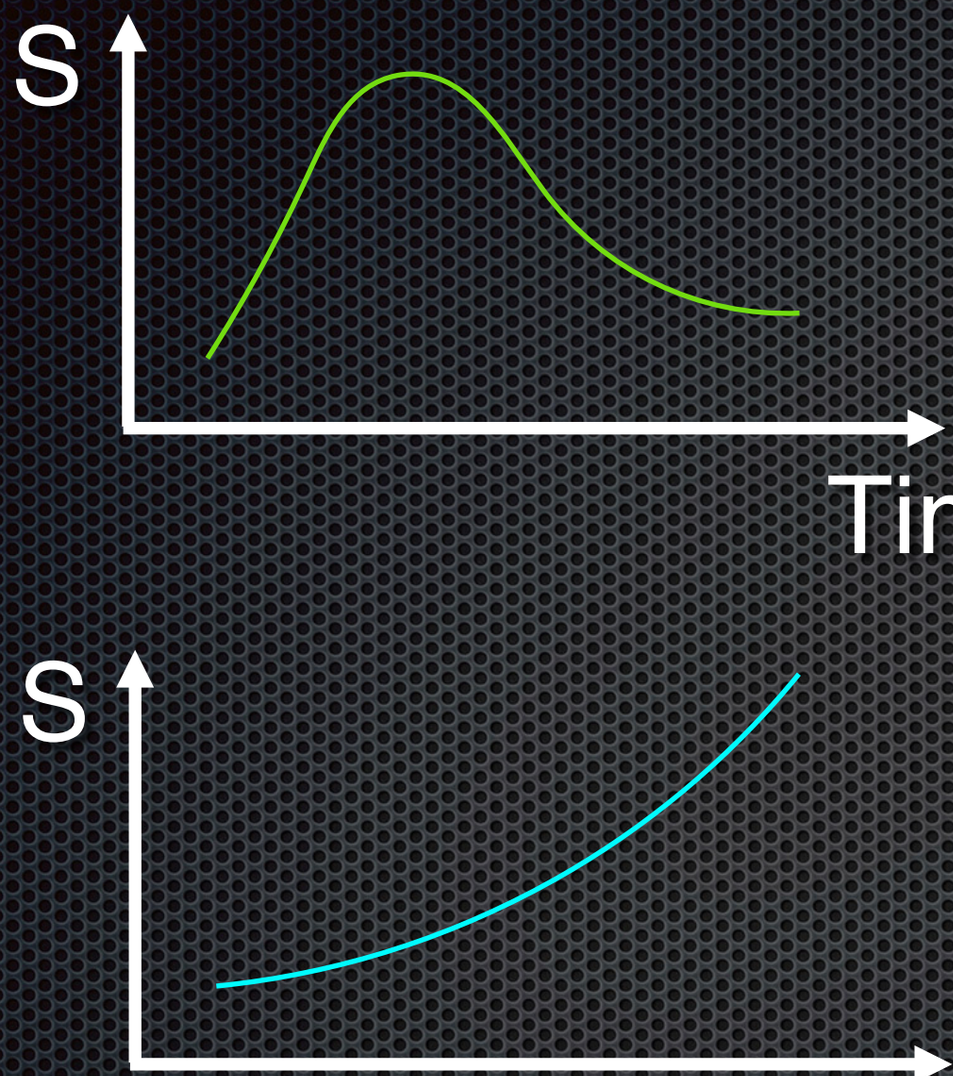
不只是针线活，一把剪刀一台缝纫机便可以开... 133%

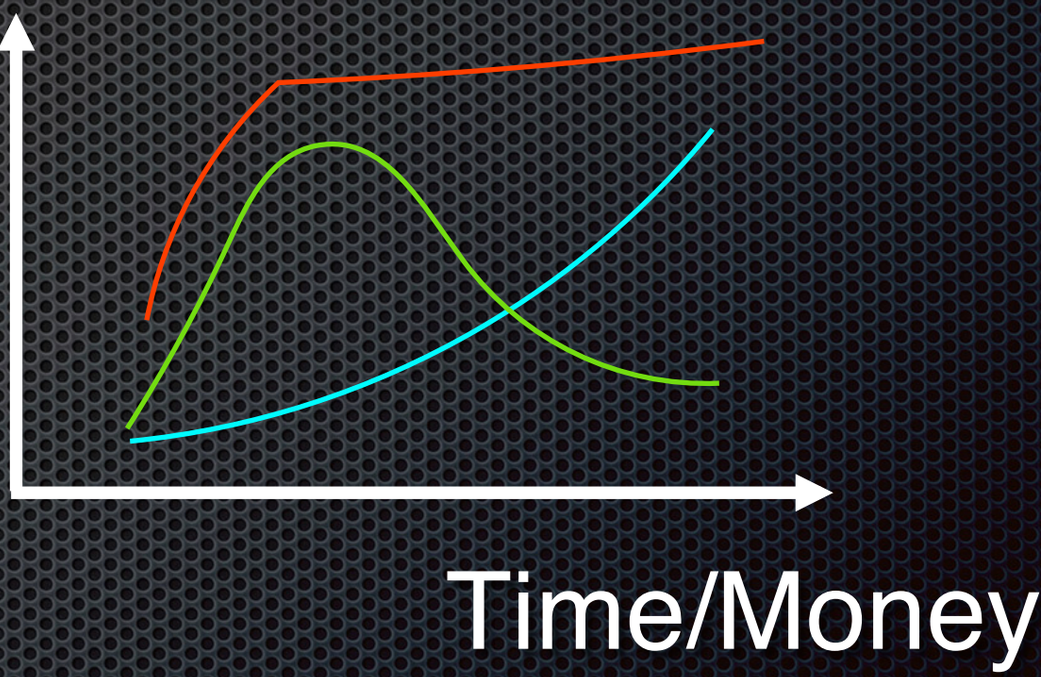
来源 追梦网
已筹资：¥1,333 剩余：5天

一：那些奇奇怪怪的排序需求

②希望能够定制打分函数

②希望能够定制打分函数





一：那些奇奇怪怪的排序需求

③还希望打分函数可以随时配置更改

①希望排序结果按照某个属性散列开

永动机乐队第三张唱片《战•歌》预售 135%

来源：乐童音乐
已筹资：¥13,520 剩余：15天

打破底价——金丝楠不再束之高阁 2128%

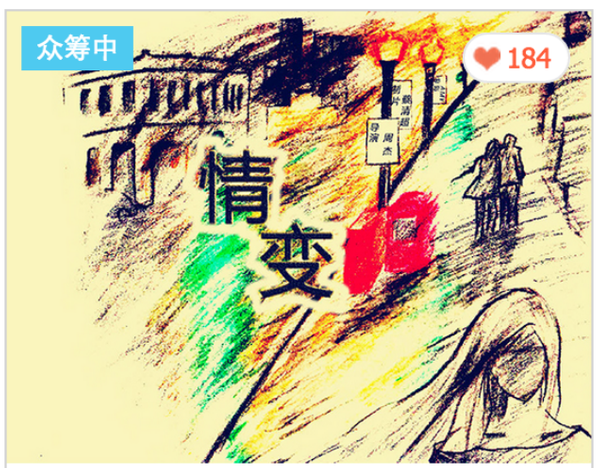
来源：众筹网
已筹资：¥63,822 剩余：3天

Yolanda智能体质管理方案，您健康管理第一站 16673%

来源 淘宝众筹
已筹资：¥1,667,376 剩余：5天

起点圆梦项目邮费差价 2300%

来源：中国梦网
已筹资：¥23 剩余：132天



微电影《情变》——生活微启示 12%

来源 淘梦网
已筹资：¥5,870 剩余：7天

不只是针线活，一把剪刀一台缝纫机便可以开... 133%

来源 追梦网
已筹资：¥1,333 剩余：5天

①希望排序结果按照某个属性散列开

ES 1.3+:

top hits aggregation

聚合，分页

我自己(ES 1.1.1)

用多线程分别取

聚合，分页

top hits aggregation

```
{
    "aggs": {
        "top-tags": {
            "terms": {
                "field": "tags",
                "size": 3
            },
            "aggs": {
                "top_tag_hits": {
                    "top_hits": {
                        "sort": [
                            {
                                "last_activity_date": {
                                    "order": "desc"
                                }
                            }
                        ],
                        "_source": {
                            "include": [
                                "title"
                            ]
                        },
                        "size" : 1
```

aggregation种类

所需要散列的field

排序

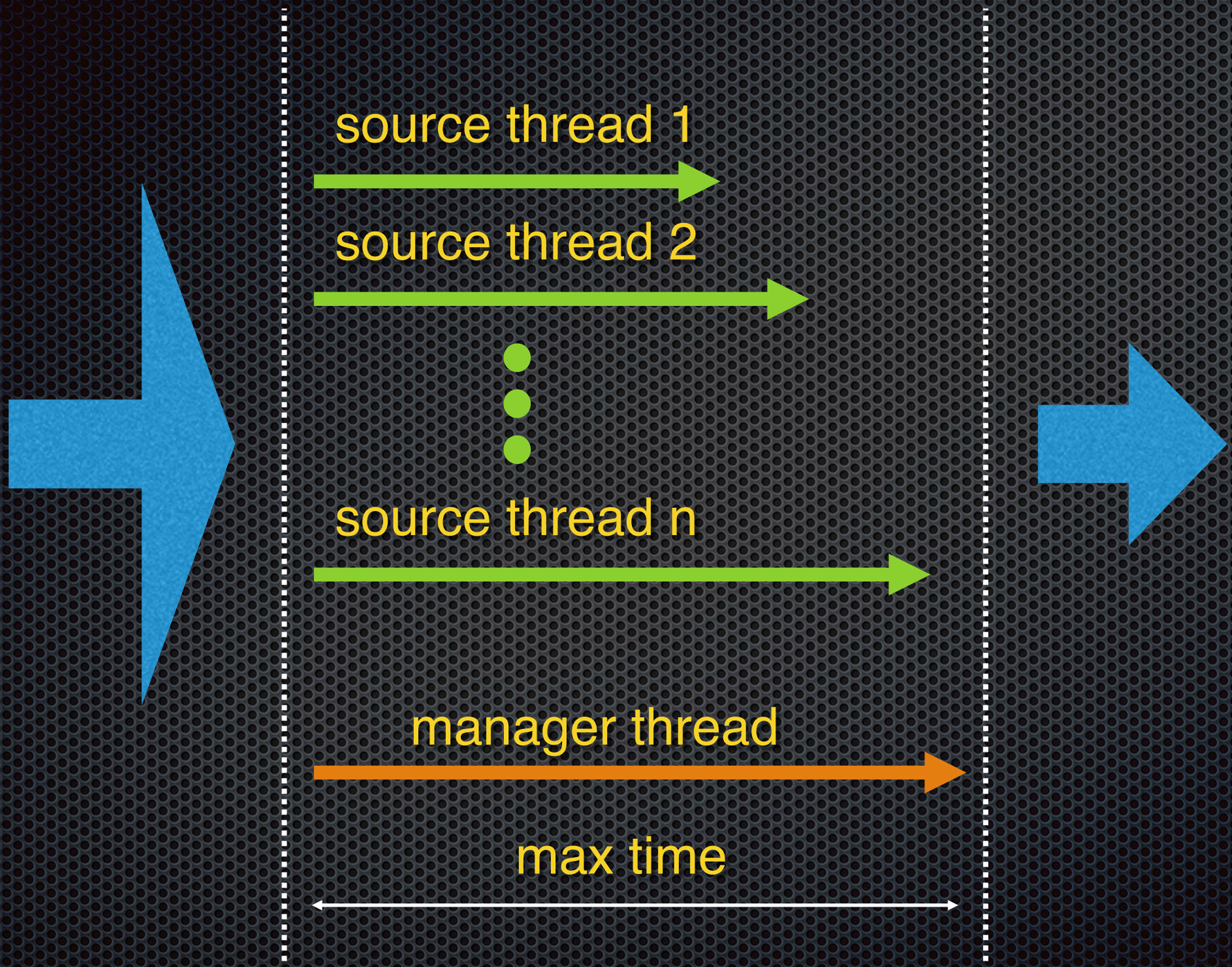
top hits aggregation


```
"aggregations": {
  "top-tags": {
     "buckets": [
        {
           "key": "windows-7",
           "doc_count": 25365,
           "top_tags_hits": {
              "hits": {
                 "total": 25365,
                 "max_score": 1,
                 "hits": [
                    {
                       "_index": "
                       "_type": "
                       "_id": "602
                       "_score": 1
                       "_source":
                          "title":
                       },
                       "sort": [
                          13701432
                       ]
                    }
                 ]
              }
           }
        },
```

```
{
   "key": "linux",
   "doc_count": 18342,
   "top_tags_hits": {
      "hits": {
         "total": 18342,
         "max_score": 1,
         "hits": [
            {
               "_index": "stack"
               "_type": "questio
               "_id": "602672",
               "_score": 1,
               "_source": {
                  "title": "Ubun
               },
               "sort": [
                  1370143379747
               ]
            }
         ]
      }
   }
},
```

```
{
   "key": "windows",
   "doc_count": 18119,
   "top_tags_hits": {
      "hits": {
         "total": 18119,
         "max_score": 1,
         "hits": [
            {
               "_index": "stack",
               "_type": "question",
               "_id": "602678",
               "_score": 1,
               "_source": {
                  "title": "If I change my computers date
e affected?"
               },
               "sort": [
                  1370142868283
               ]
            }
         ]
      }
   }
}
```

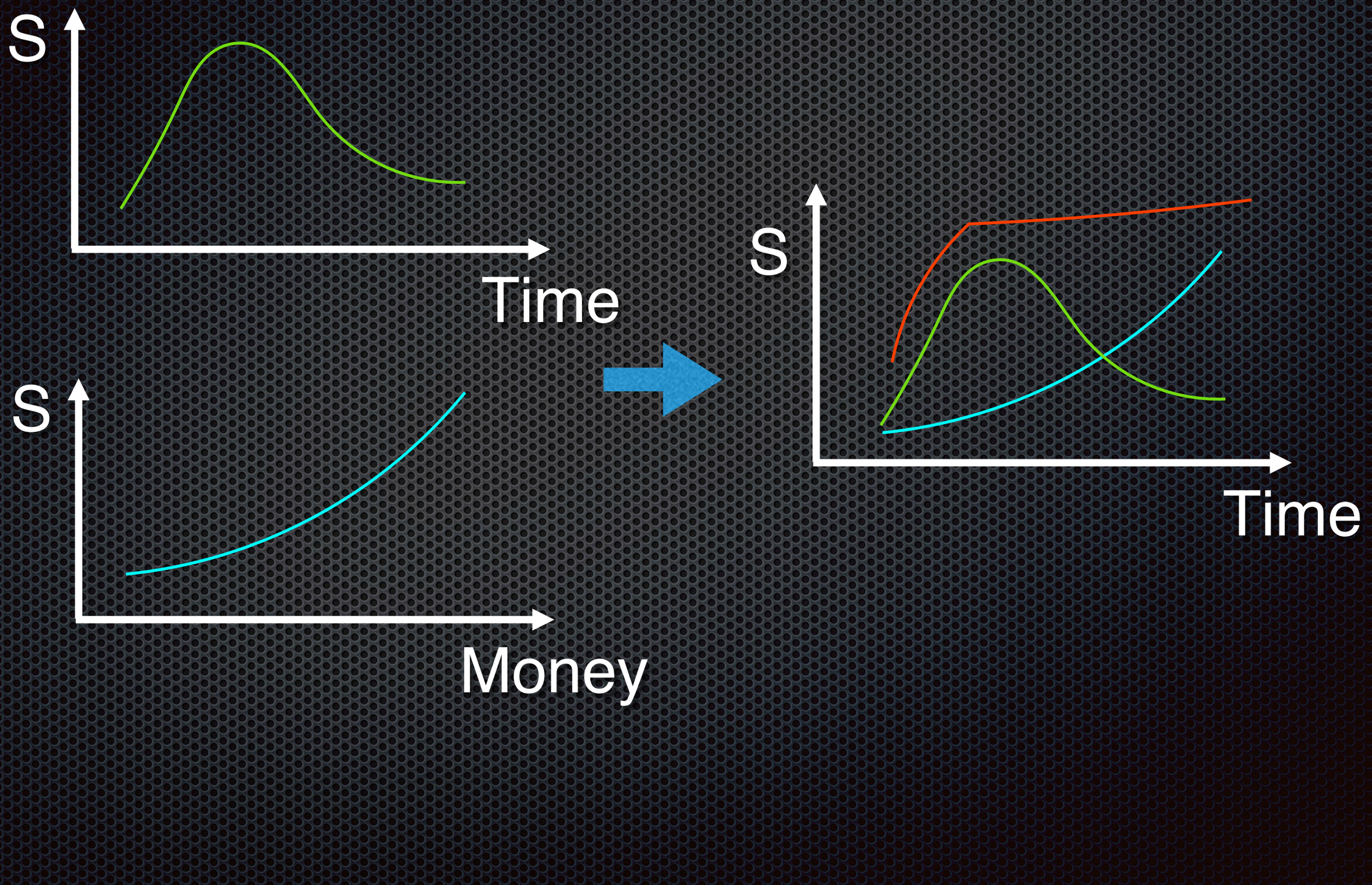
用多线程分别取


source thread 1
source thread 2
source thread n
manager thread
max time
聚合，分页

效率(base on 1.4 beta1 and 200M docs)

	top hits aggregation	用多线程分别取
10*n	60ms	10ms
100*n	200ms	80ms
1000*n	3s	1000ms
10000*n	30s/failed	10s/overtime

②希望能够定制打分函数



定制打分函数

```
"function_score": {
    "(query|filter)": {},
    "boost": "boost for the whole query",
    "FUNCTION": {},
    "boost_mode":"(multiply|replace|...)"
}
```

```
"function_score": {
    "(query|filter)": {},
    "boost": "boost for the whole query",
    "functions": [
        {
            "filter": {},
            "FUNCTION": {},
            "weight": number
        },
        {
            "FUNCTION": {}
        },
        {
            "filter": {},
            "weight": number
        }
    ],
    "max_boost": number,
    "score_mode": "(multiply|max|...)",
    "boost_mode": "(multiply|replace|...)"
```

field_value_factor

```
{"explain": true,
"query": {
    "function_score": {
        "functions": [
            {
            "field_value_factor": {
                    "field": "total_money",
                    "factor": 1.2,
                    "modifier": "sqrt"

                },
                  "weight":12
            }
        ],
        "boost_mode":"avg"
    }
}
}
```

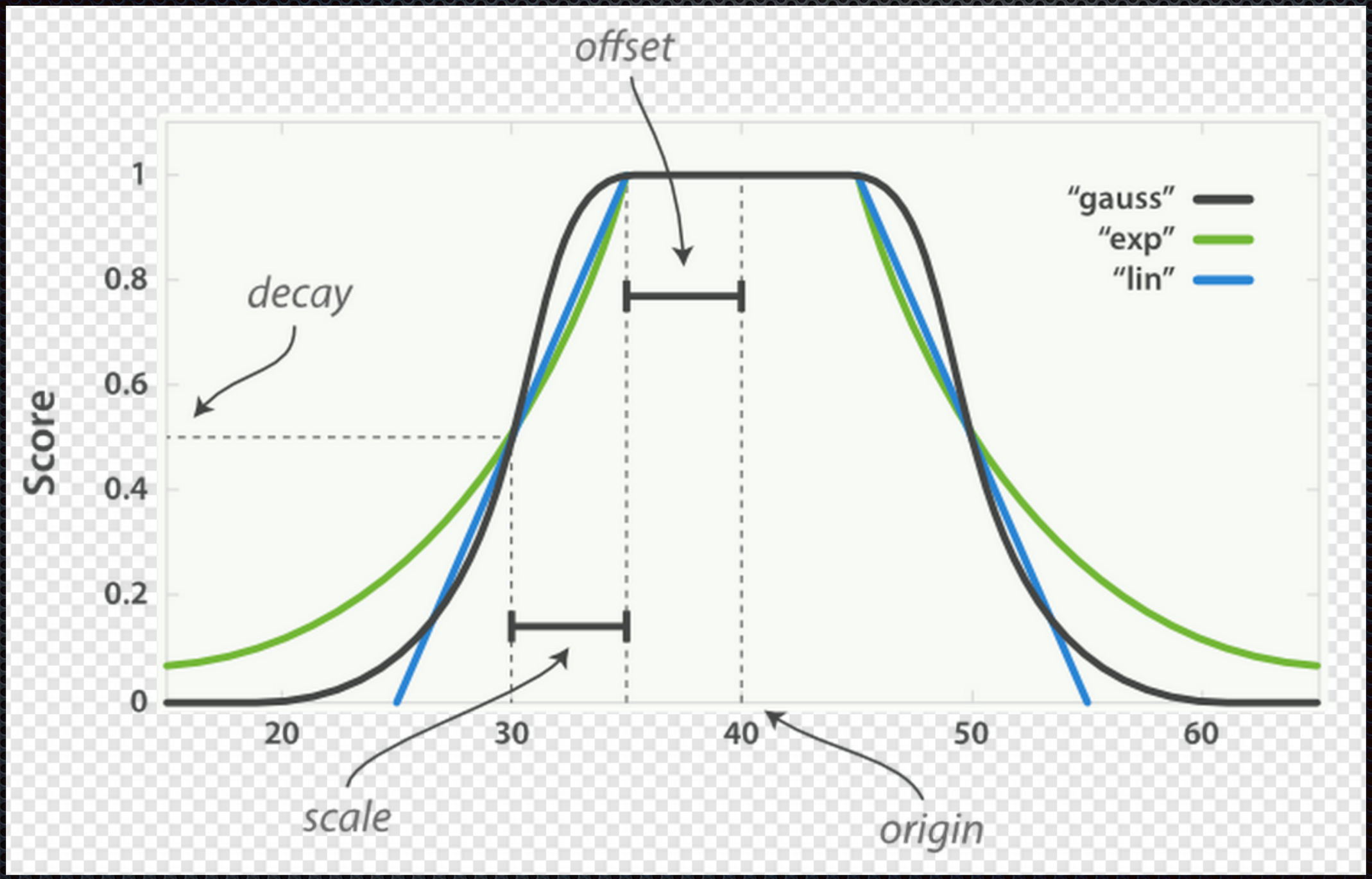
```
sqrt(1.2 * doc['total_money'].value)
```

```
"_explanation": {
    "value": 146969.89,
    "description": "function score, product of:",
    "details": [
        {
            "value": 146969.89,
            "description": "avg of",
            "details": [
                {
                    "value": 1,
                    "description": "ConstantScore(*:*), product of:",
                    "details": [
                        {
                            "value": 1,
                            "description": "boost"
                        },
                        {
                            "value": 1,
                            "description": "queryNorm"
                        }
                    ]
                },
                {
                    "value": 293938.78,
                    "description": "Math.min of",
                    "details": [
                        {
                            "value": 293938.78,
                            "description": "product of:",
                            "details": [
                                {
                                    "value": 24494.898,
                                    "description": "field value function: sqrt(doc['total_money'].value * factor=1.2)"
                                },
                                {
                                    "value": 12,
                                    "description": "weight"
                                }
                            ]
                        },
                        {
                            "value": 3.4028235e+38,
                            "description": "maxBoost"
                        }
                    ]
                }
            ]
        },
        {
            "value": 1,
            "description": "queryBoost"
```

decay functions

```
{"explain": true,
"query": {
    "function_score": {
        "functions": [
            {
            "gauss": {
                "start_time": {
                    "origin": "2014-09-17",
                    "scale": "10d",
                    "offset": "5d",
                    "decay" : 0.5
                }
            }
            }
        ],
        "boost_mode":"replace"
    }
}
}
```

decay functions



decay functions


```
"value": 1,
"description": "function score, product of:",
"details": [
    {
        "value": 1,
        "description": "Math.min of",
        "details": [
            {
                "value": 1,
                "description": "Function for field start_time:",
                "details": [
                    {
                        "value": 1,
                        "description": "exp(-0.5*pow(MIN[Math.max(Math.abs(1.410487971E12(=doc value) -
                      1.410912E12(=origin))) - 4.32E8(=offset), 0)],2.0)/5.384830386217238 4E17)"
                    }
                ]
            },
            {
                "value": 3.4028235e+38,
                "description": "maxBoost"
            }
        ]
    },
    {
        "value": 1,
        "description": "queryBoost"
    }
]
```

```
{"explain": true,
"query": {
    "function_score": {
        "functions": [
            {
            "field_value_factor": {
                    "field": "total_money",
                    "factor": 1.2,
                    "modifier": "none"

                },
                 "weight":0.00012
            },
            {
            "gauss": {
                 "start_time": {
                     "origin": "2014-09-17",
                     "scale": "10d",
                     "offset": "5d",
                     "decay" : 0.5
                   }
                }
            }
        ],

        "score_mode": "avg",
        "boost_mode":"replace"
    }
}
```

```
"_explanation": {
    "value": 36000.062,
    "description": "function score, product of:",
    "details": [
        {
            "value": 36000.062,
            "description": "Math.min of",
            "details": [
                {
                    "value": 36000.062,
                    "description": "function score, score mode [avg]",
                    "details": [
                        {
                            "value": 72000,
                            "description": "function score, product of:",
                            "details": [
                                {
                                    "value": 1,
                                    "description": "match filter: *:*"
                                },
                                {
                                    "value": 72000,
                                    "description": "product of:",
                                    "details": [
                                        {
                                            "value": 600000000,
                                            "description": "field value function: (doc['total_money'].value * factor=1.2)"
                                        },
                                        {
                                            "value": 0.00012,
                                            "description": "weight"
                                        }
                                    ]
                                }
                            ]
                        },
                        {
                            "value": 0.124922864,
                            "description": "function score, product of:",
                            "details": [
                                {
                                    "value": 1,
                                    "description": "match filter: *:*"
                                },
                                {
                                    "value": 0.124922864,
                                    "description": "Function for field start_time:",
                                    "details": [
                                        {
                                            "value": 0.124922864,
                                            "description": "exp(-0.5*pow(MIN[Math.max(Math.abs(1.408983286E12(=doc value) - 1.410912E
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}
```

script score

```
"script_score": {
    "lang": "lang",
    "params": {
       "param1": value1,
       "param2": value2
    },
   "script": "_score * doc['my_numeric_field'].value / pow(param1, param2)"
}
```

定制打分函数

```
{
  "query" : {
    "function_score" : {
      "functions" : [ {
        "script_score" : {
          "script" : "
          total_money  = doc['total_money'].value;
          start_money = doc['start_money'].value;
          if (total_money < threshold) {
               return total_money * start_money.power(2) / Math.sqrt(target)
          };
          return total_money * (1 - discount) * start_money / target;",
          "lang" : "groovy",
          "params" : {
            "target" : 100000,
            "discount" : 0.1,
            "threshold" : 80
          }
        }
      } ],
      "score_mode" : "max",
      "boost_mode" : "replace"
    }
  },
  "explain" : true
}
```

定制打分函数

```
"_explanation": {
        "value": 2024999940,
        "description": "function score, product of:",
        "details": [
            {
                "value": 2024999940,
                "description": "Math.min of",
                "details": [
                    {
                        "value": 2024999940,
                        "description": "script score function, computed with script:\"
                        total_money  = doc['total_money'].value;
                        start_money = doc['start_money'].value;
                        if (total_money < threshold) {
                            return total_money * start_money.power(2) / Math.sqrt(target) };
                            return total_money * (1 - discount) * start_money / target;
                       \" and parameters: \n{target=100000, threshold=80, discount=0.1}"
                    },
                    {
                        "value": 3.4028235e+38,
                        "description": "maxBoost"
                    }
                ]
            },
            {
                "value": 1,
                "description": "queryBoost"
            }
```

定制打分函数

```
{"explain": true,
"query": {
    "function_score": {
        "functions": [
            {
            "script_score": {
            "lang" : "groovy",
                "params": {
                "threshold": 80,
                "discount": 0.1,
                "target": 1000000
                },
              "script_id": "testforchinaesuser"
            }
        }
        ],

        "score_mode": "avg",
        "boost_mode":"replace"
    }
}
}
```

定制打分函数

http://10.18.6.169:9201/_scripts/groovy/testforchinaesuser

form-data | x-www-form-urlencoded | raw | JSON

```
{"script": "
 total_money   = doc['total_money'].value;
 start_money = doc['start_money'].value;
 if (total_money < threshold) {
 return total_money * start_money / target };
 return total_money * (1 - discount) * start_money / target;"
}
```

不足：JAVA API还不支持scrip_id

效率(base on 1.4 beta1 and 200k docs)

	1Field	2Field	3Field
加减运算	60ms	120ms	170ms
乘除运算	80ms	120ms	180ms
幂次运算	80ms	120ms	180ms

[ES-sort-share] shutdown

By 刘刚